Разработка и применение

систем разграничения доступа на базе атрибутов

Михаил Польгун

8+ лет разработки систем ИБ

Лид трех команд

mpolgun@ptsecurity.com

@mpolgun

@MikePolgun

Обеспечиваем практическую кибербезопасность

20+
лет опыта
исследований
и разработок

1800+
инженеров по ИБ,
разработчиков, аналитиков
и других специалистов

250+
экспертов в нашем
исследовательском
центре безопасности

200+
обнаруженных
уязвимостей
нулевого дня в год

200+
аудитов безопасности
корпоративных систем
делаем ежегодно

50%
всех уязвимостей
в промышленности и телекомах
обнаружили наши эксперты

создаем продукты
и решения

проводим аудиты
безопасности

расследуем
инциденты

исследуем
угрозы

План

1. Зачем нам это понадобилось
2. Немного теории
3. Как мы это делали

MaxPatrol 10 (2019 год)

- On-prem-решение
- Микросервисная архитектура
- Около 40 микросервисов single instance
- Очень много экспертного контента
  (например, база уязвимостей, правила обработки событий)
- Выпускается в одной редакции — SIEM

Схема MaxPatrol 10

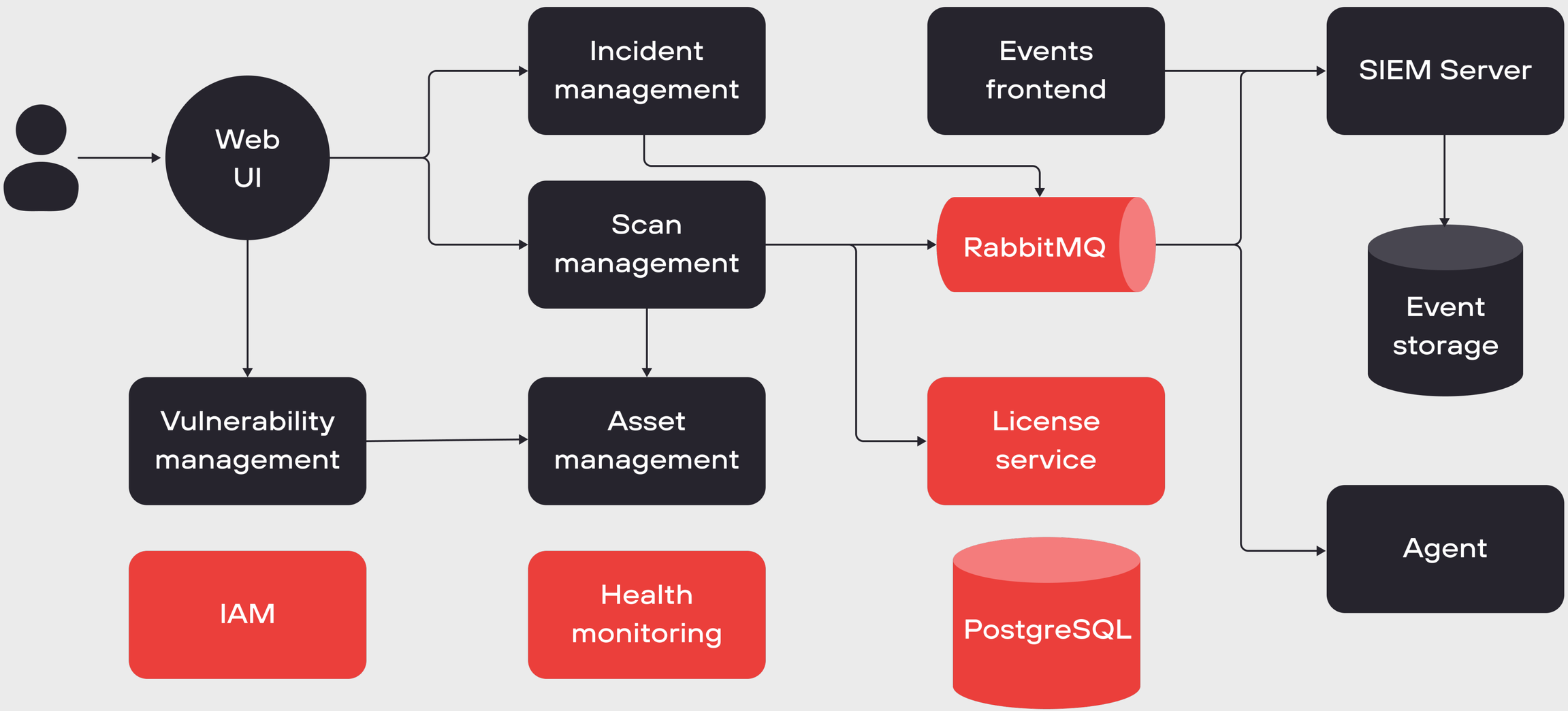


Схема MaxPatrol 10

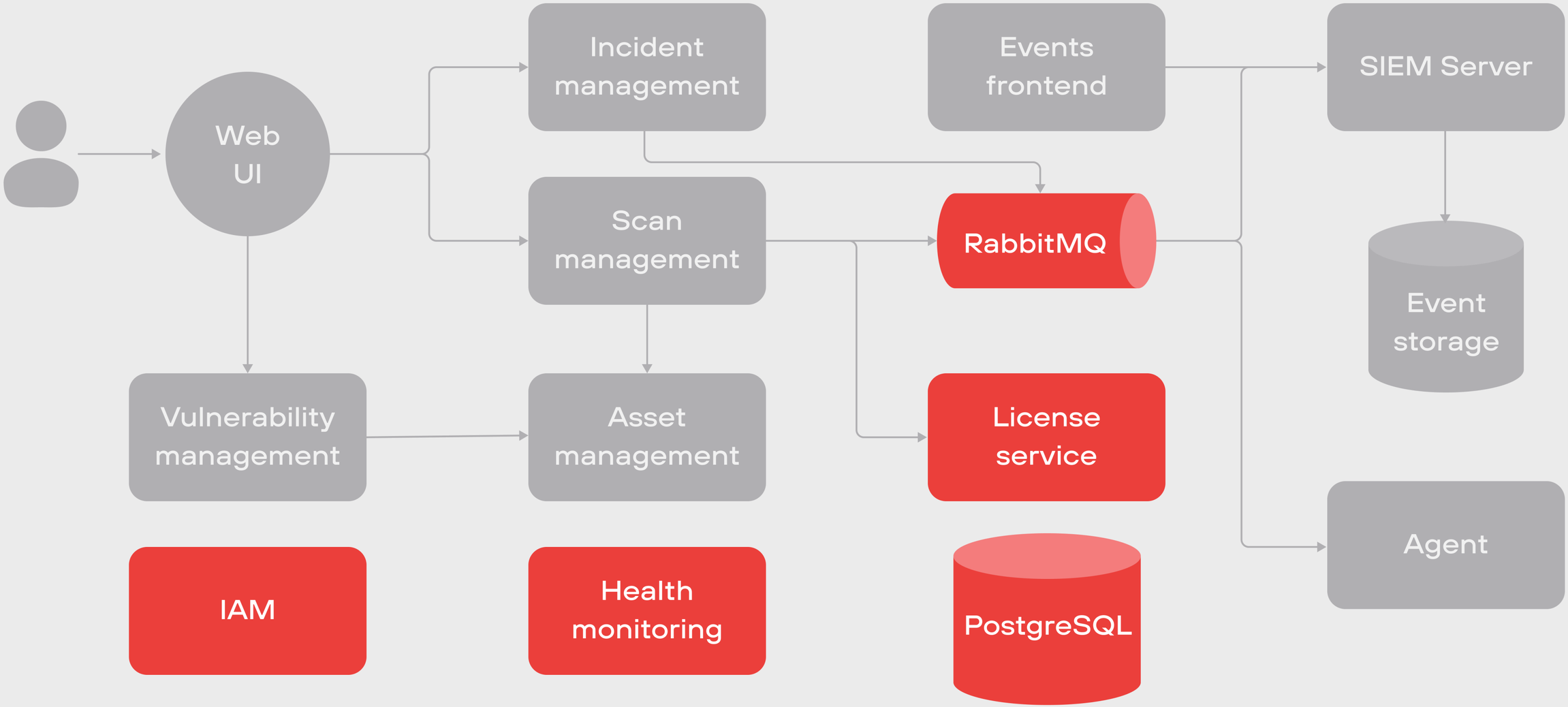


Схема MaxPatrol 10

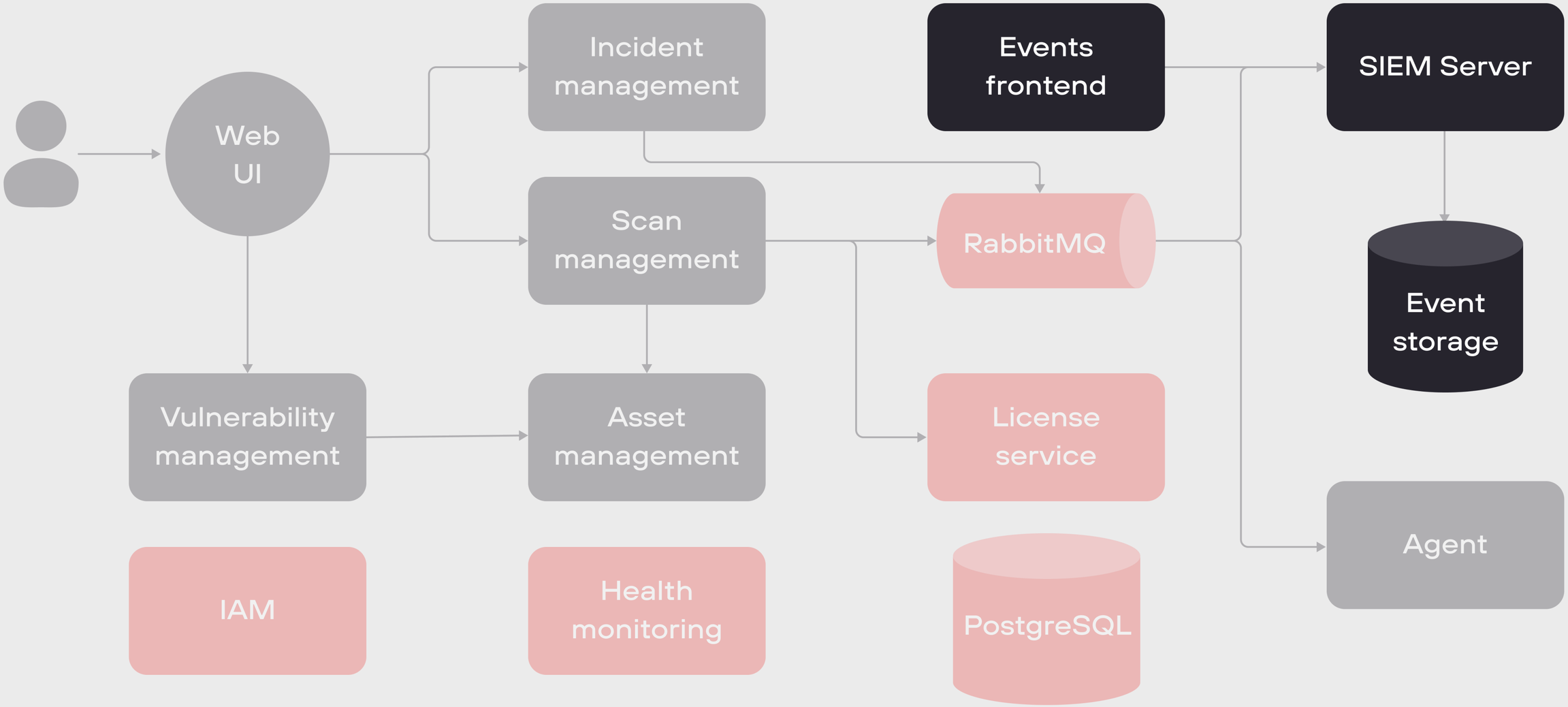


Схема MaxPatrol 10

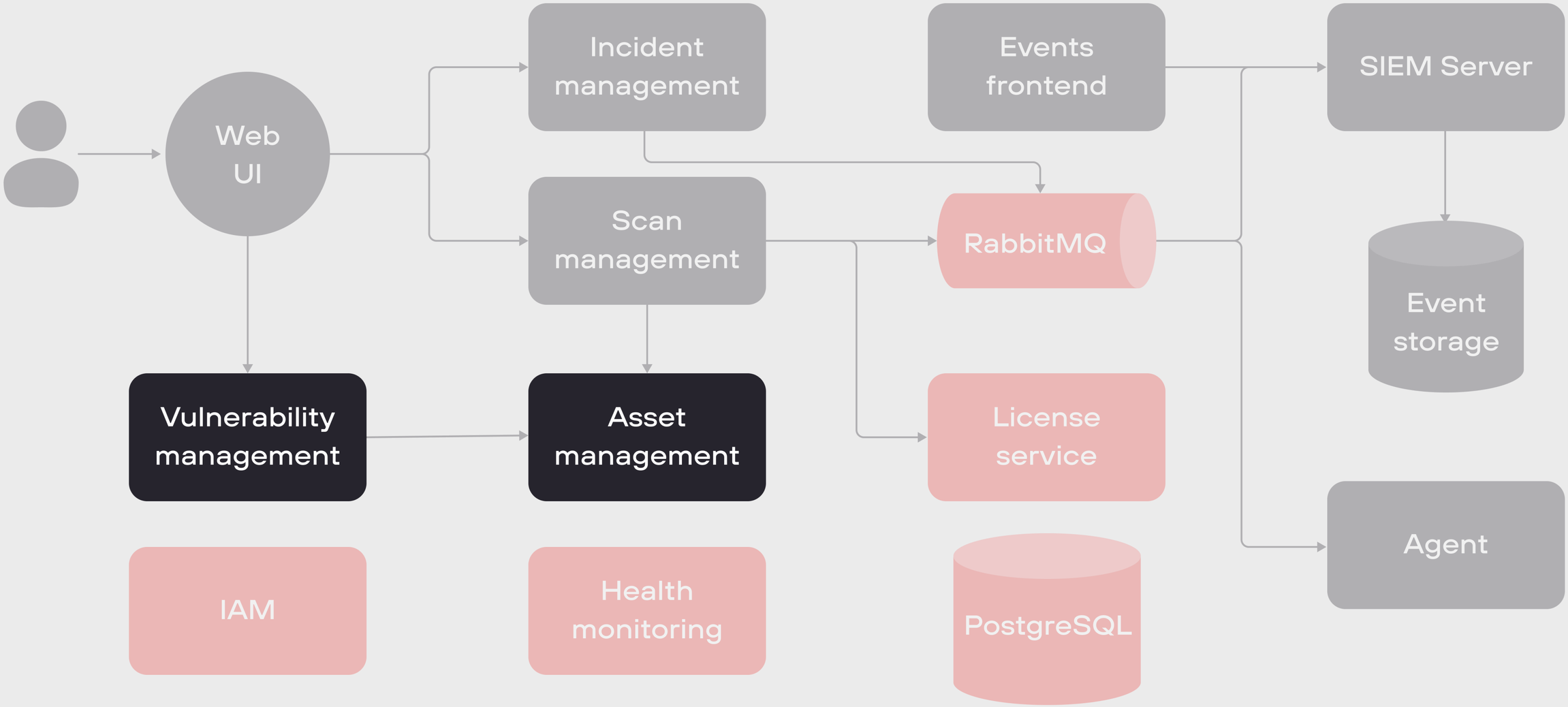


Схема MaxPatrol 10

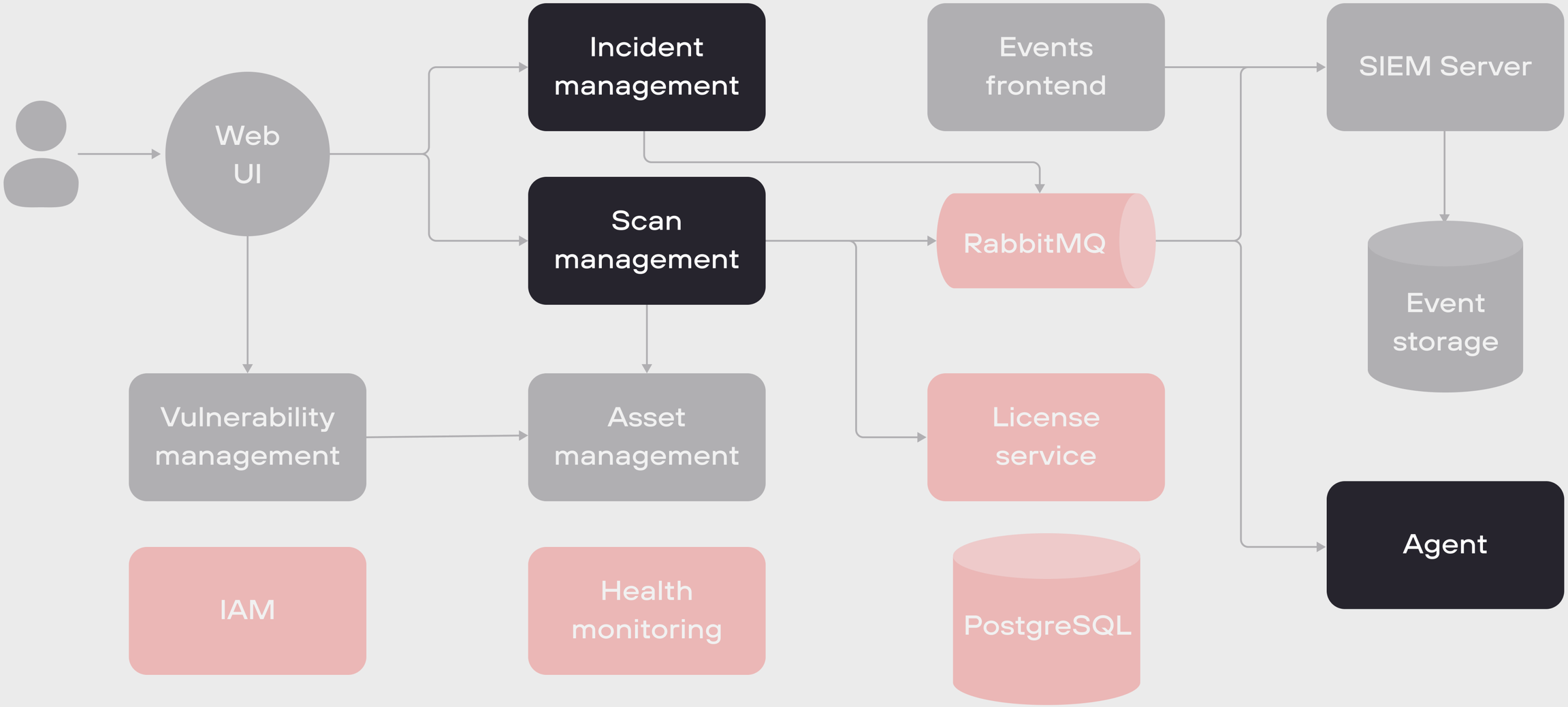
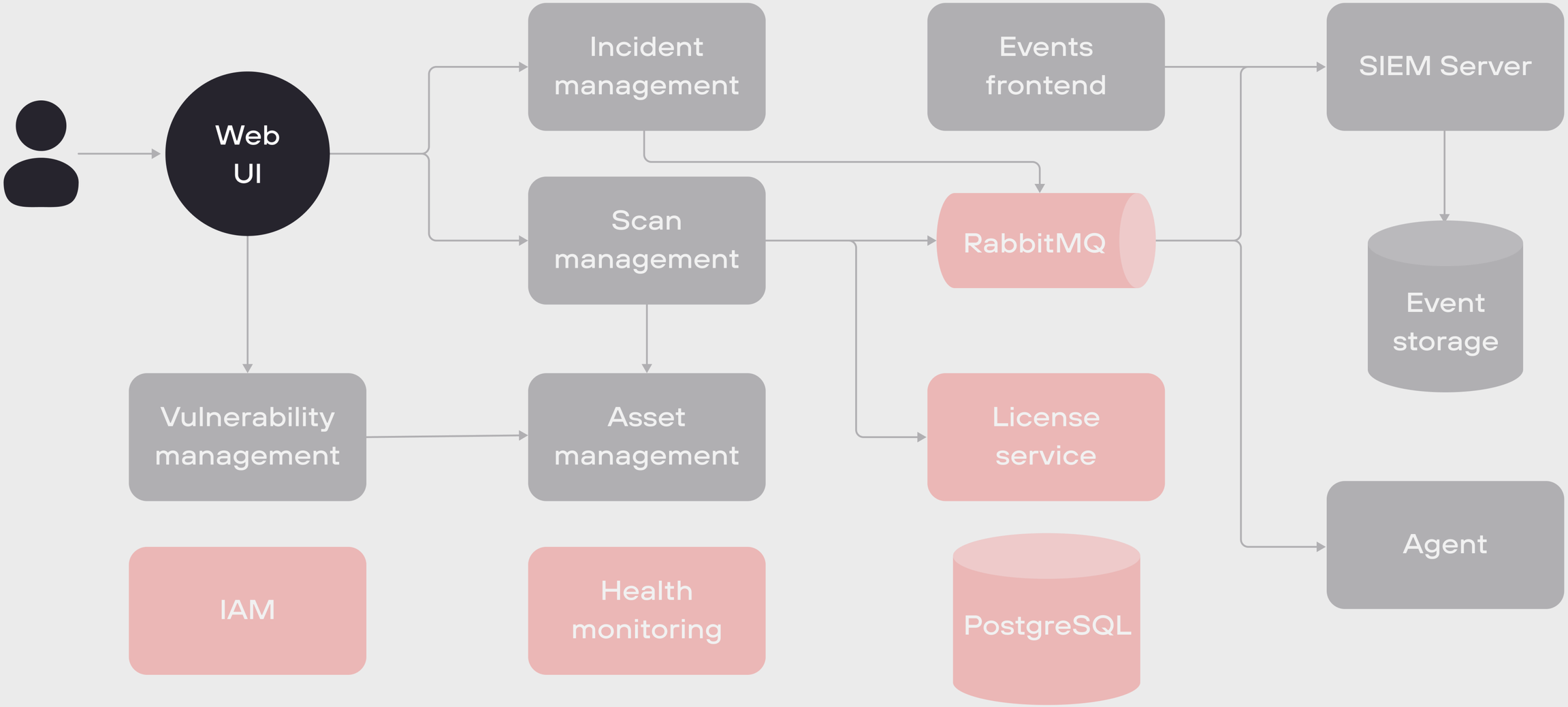


Схема MaxPatrol 10

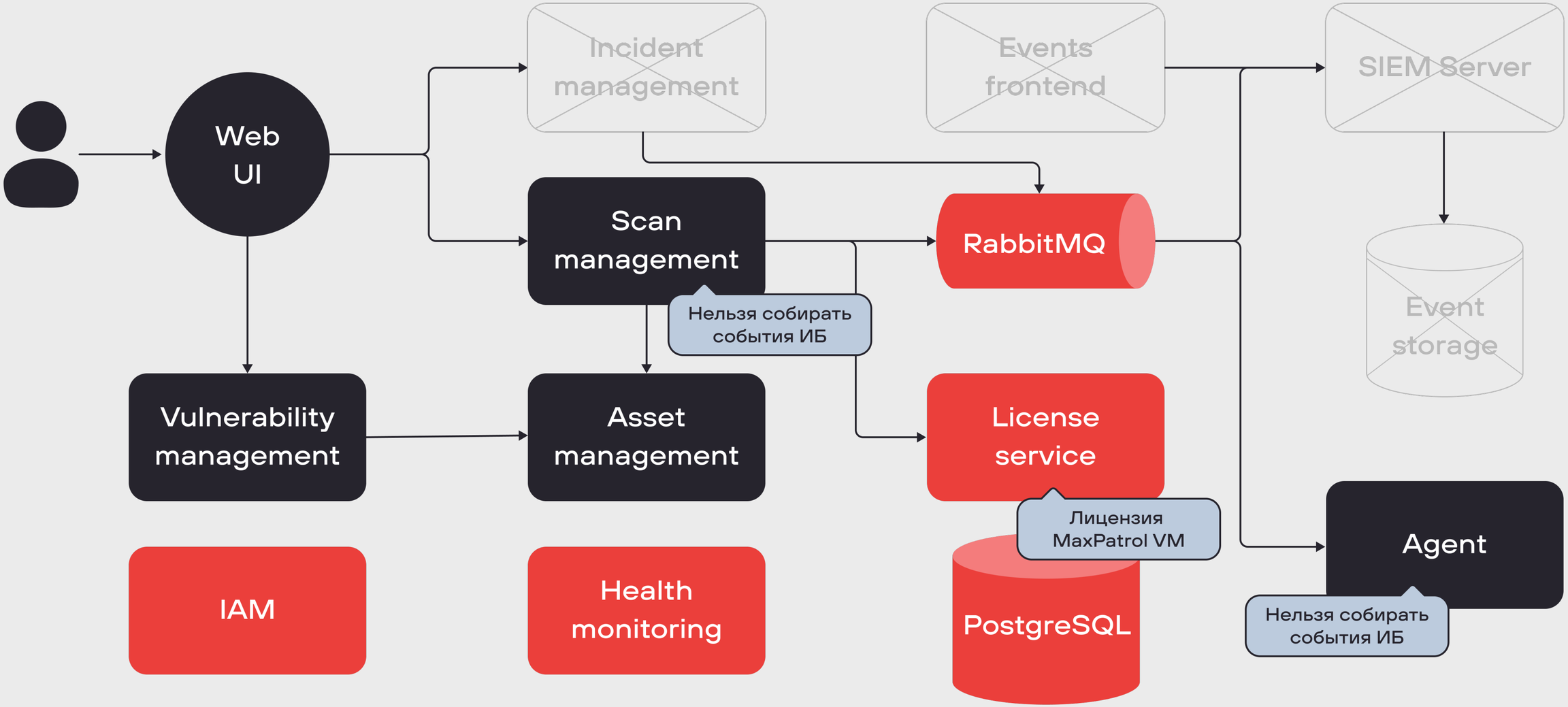


MaxPatrol 10 (2020 год)

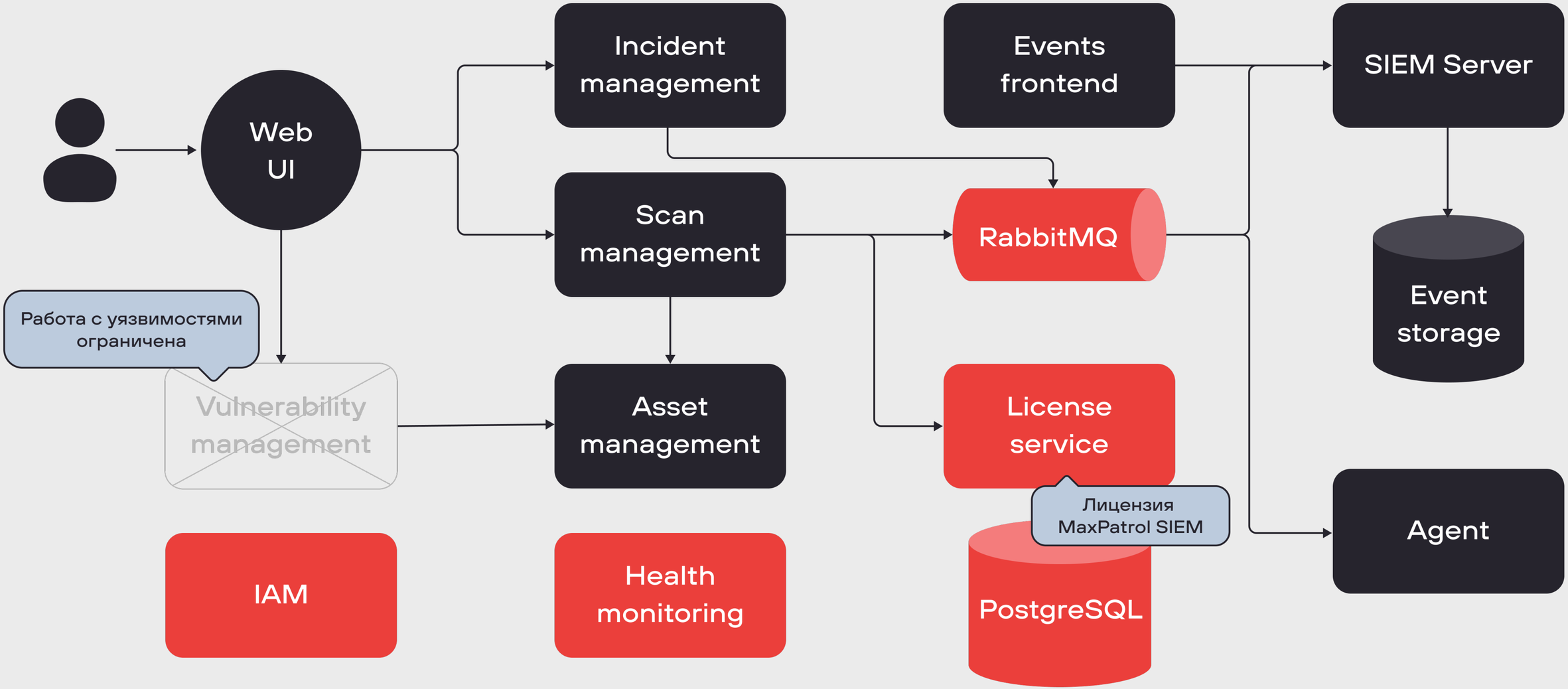
На подходе новая редакция — VM

- Уметь выпускать разные вариации одного продукта
- Заказчик может работать с разными редакциями на одной установке
- **В зависимости от редакции:**
  - Часть экспертизы не доступна
  - Часть набора функций отключается
- Тип редакции влияет на действия пользователя

MaxPatrol VM (в составе MaxPatrol 10)

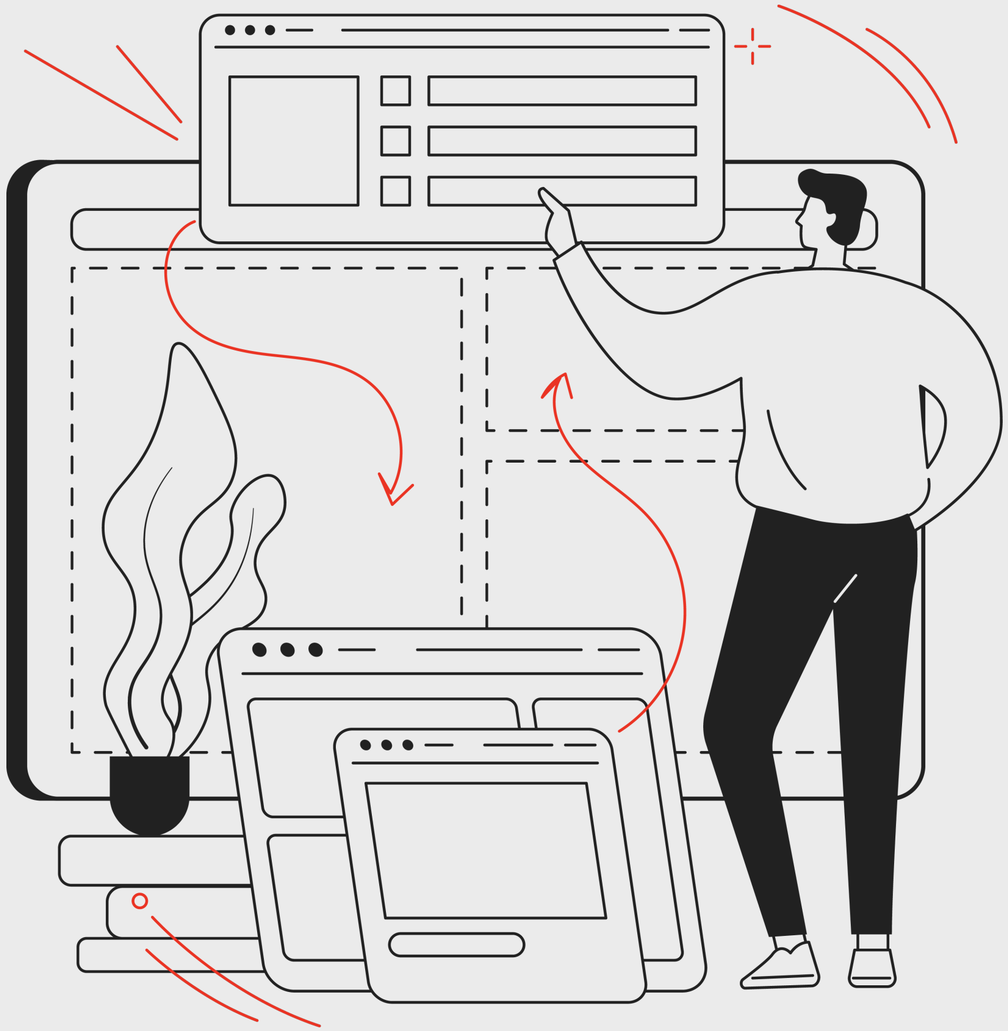


MaxPatrol SIEM (в составе MaxPatrol 10)

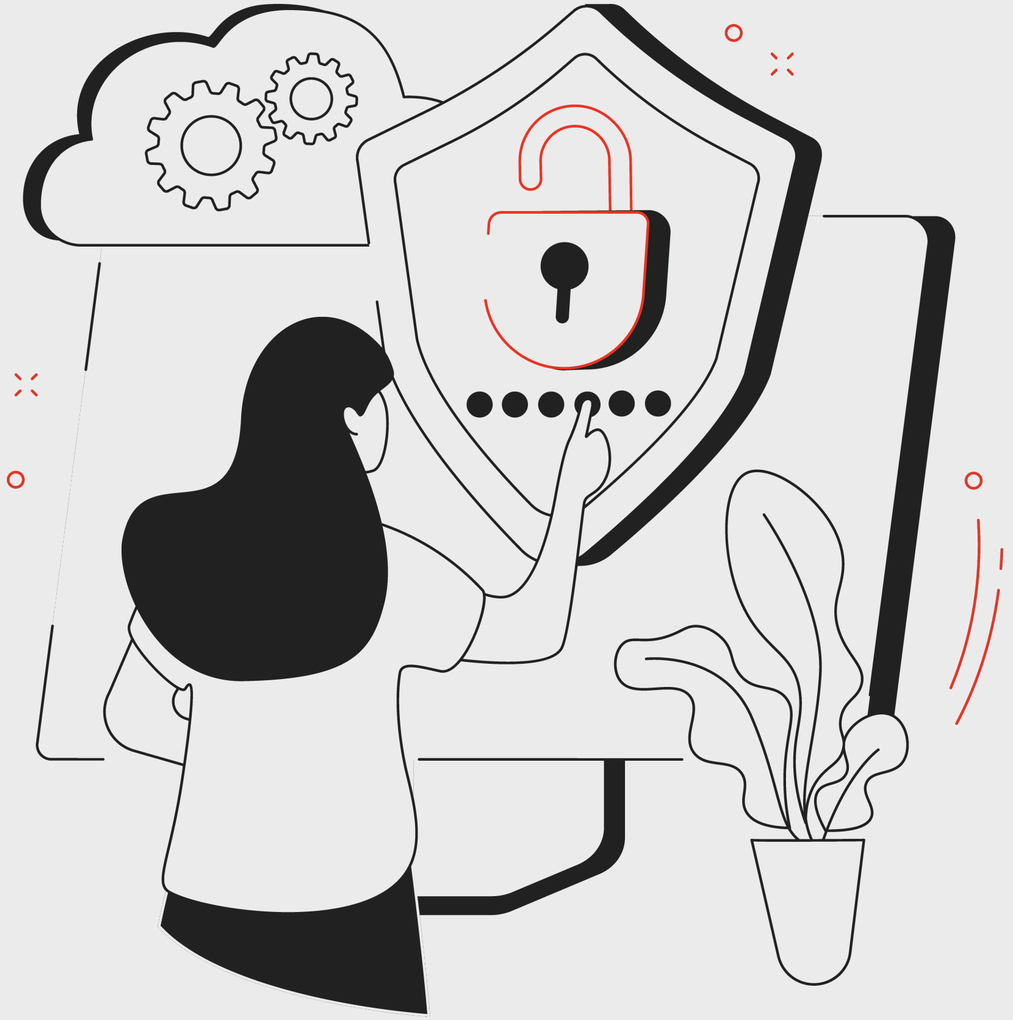


Функции и экспертиза

зависят от редакции
и прав пользователя



Нам поможет разграничение доступа



Разграничение доступа

Ваня

Финансы

Бухгалтерия

IT

Разграничение доступа

Таня

Финансы

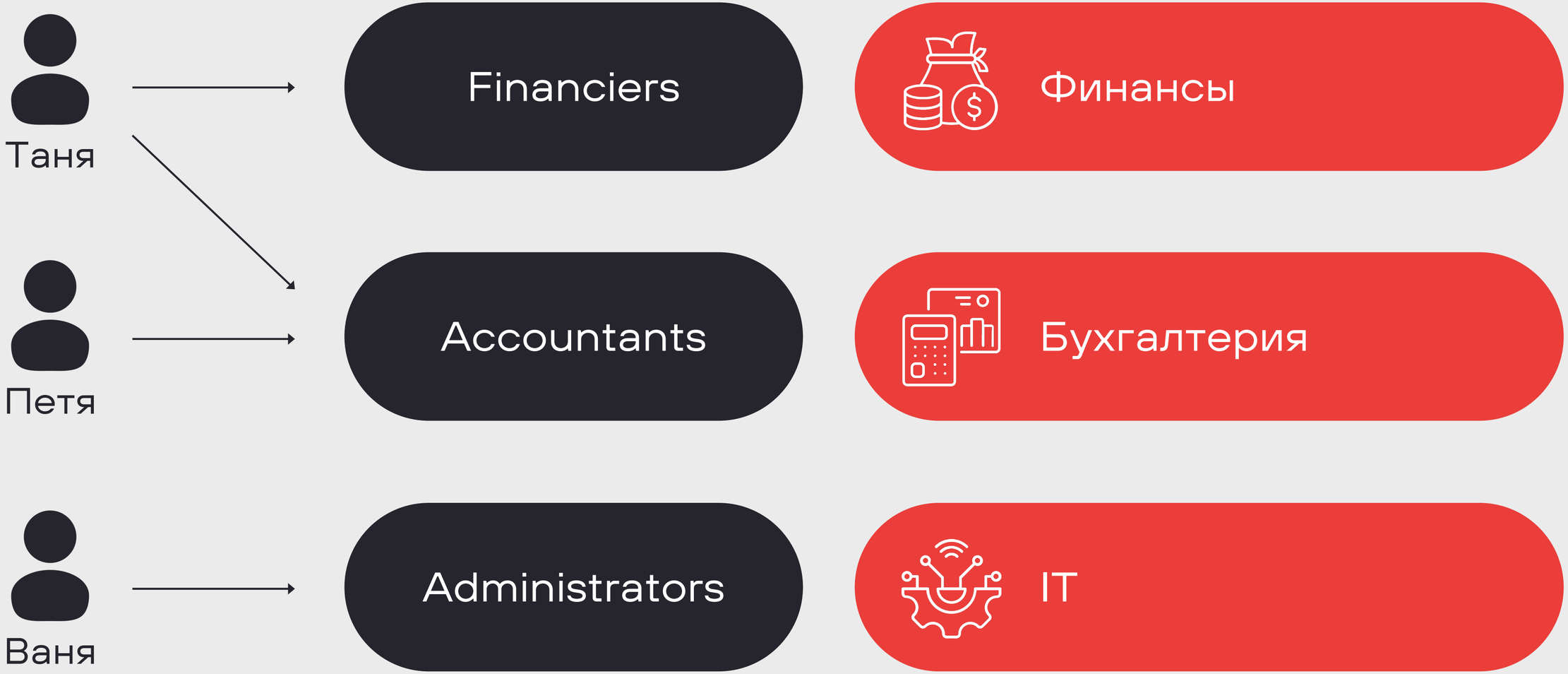
Петя

Бухгалтерия

Ваня

IT

Role-based access control



Проблемы RBAC

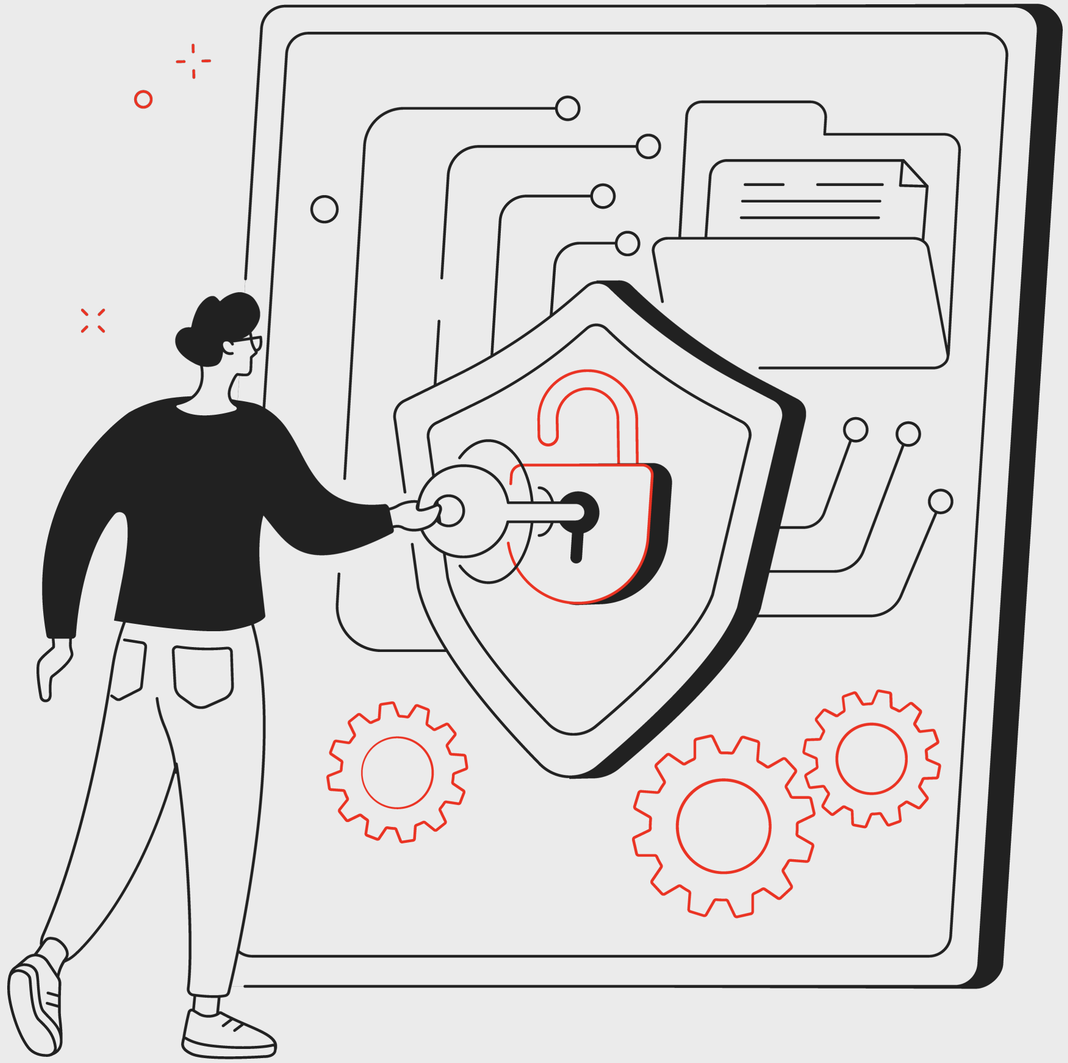
- Количество ролей растет

Декартово произведение ролей

	Moscow	Saints-Petersburg	Novosibirsk
Financiers	Financiers Moscow	Financiers Saints-Petersburg	Financiers Novosibirsk
Accountants	Accountants Moscow	Accountants Saints-Petersburg	Accountants Novosibirsk
Administrators	Administrators Moscow	Administrators Saints-Petersburg	Administrators Novosibirsk

Проблемы RBAC

- Количество ролей растет
- Нельзя ограничить доступ к данным



Правило !

Бухгалтер может выпускать квартальный отчет

по своему офису

в рабочее время

Attribute-based access control

Субъект

имя, должность, офис, рабочие часы

Ресурс

тип, название

Действие

название

Среда

город, страна, время

Attribute-base access control

Правило !

Бухгалтер может выпускать квартальный отчет в рабочее время

Условия

- Субъект.Должность = *Бухгалтер*
- Ресурс.Тип = *Отчет*
- Среда.Время in Субъект.Рабочие часы

Attribute-base access control

- Разграничивает доступ к данным
- Правила не меняются при росте количества атрибутов

XACML

eXtensible Access Control Markup Language

- С 2003 года разрабатывается OASIS
- Версия 3.0 — самая актуальная (2013 год)
- Основан на XML


PolicySet
Policy
Rule

Rule

Target — логическое выражение. Состоит из атрибутов и констант

Condition — логическое выражение, вычисляемое динамически

Effect — результат вычисления правила

Permit | Deny | Not applicable | Indeterminate

Obligation — действие, которое необходимо выполнить

Advice — действие, которое рекомендуется выполнить

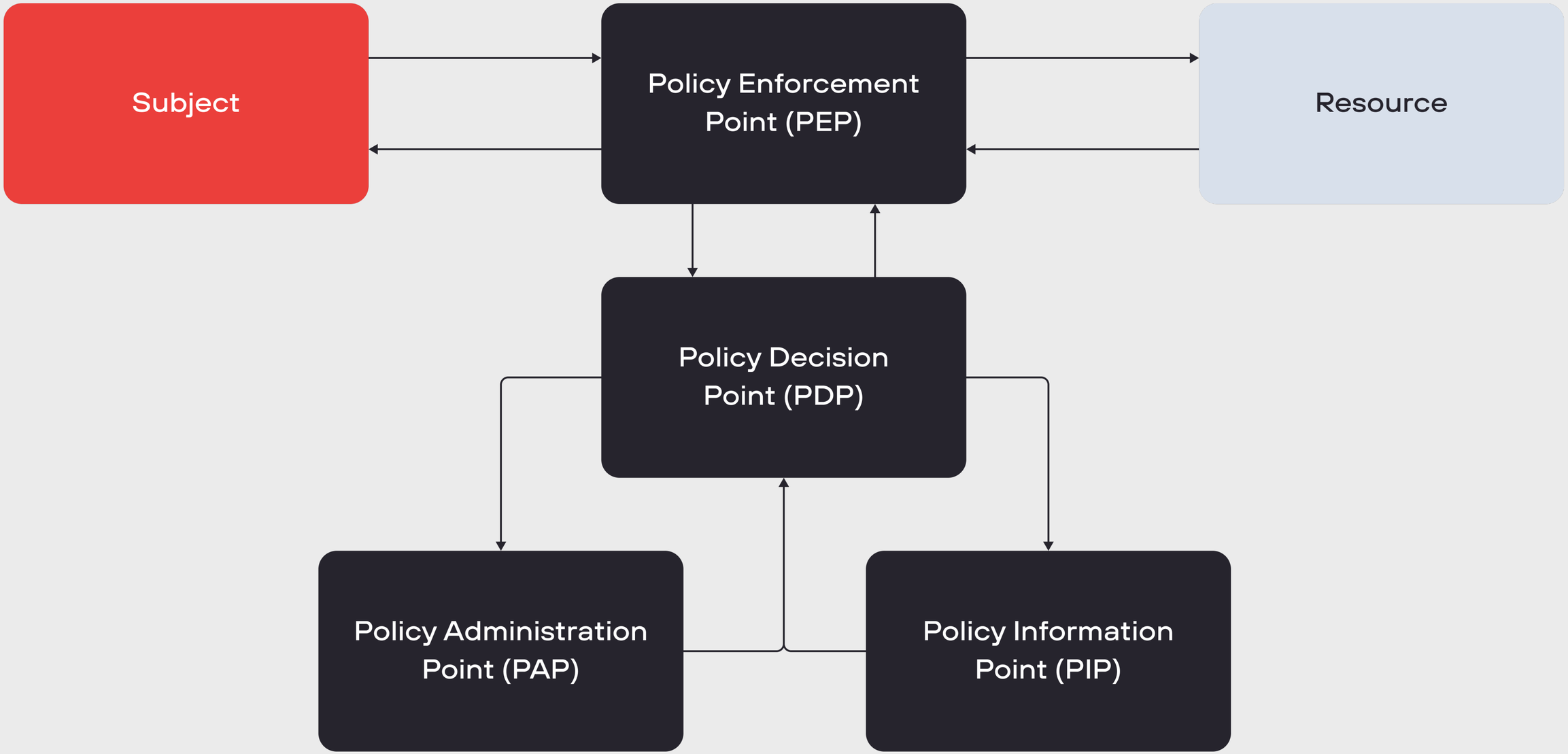
Policy

- **Target** — логическое выражение. Состоит из атрибутов и констант
- **Rules** — список правил
- **Obligation**
- **Advice**
- **Rule-combining algorithm**

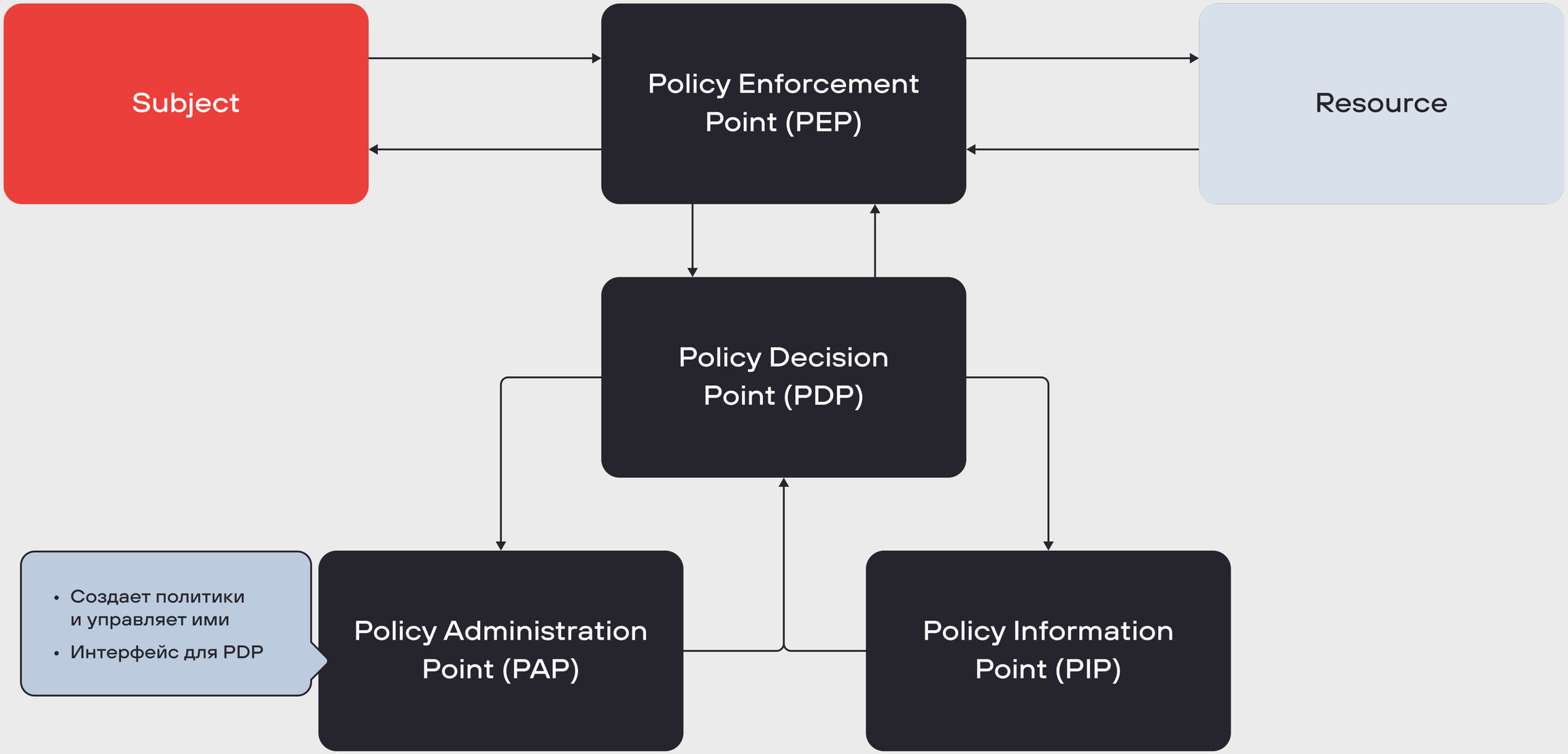
Rule-combining algorithm

Алгоритм	Пример
First-applicable	[Not applicable, Permit, Not applicable] → Permit [Not applicable, Deny, Permit] → Deny [Not applicable, Not applicable, Not applicable] → Not applicable
Deny — Unless — Permit	[Deny, Permit, Not applicable] → Permit [Deny, Deny, Not applicable] → Deny [Not applicable, Not applicable, Not applicable] → Deny
Permit — Unless — Deny	[Permit, Deny, Permit] → Deny [Permit, Permit, Not applicable] → Permit [Not applicable, Not applicable, Not applicable] → Permit

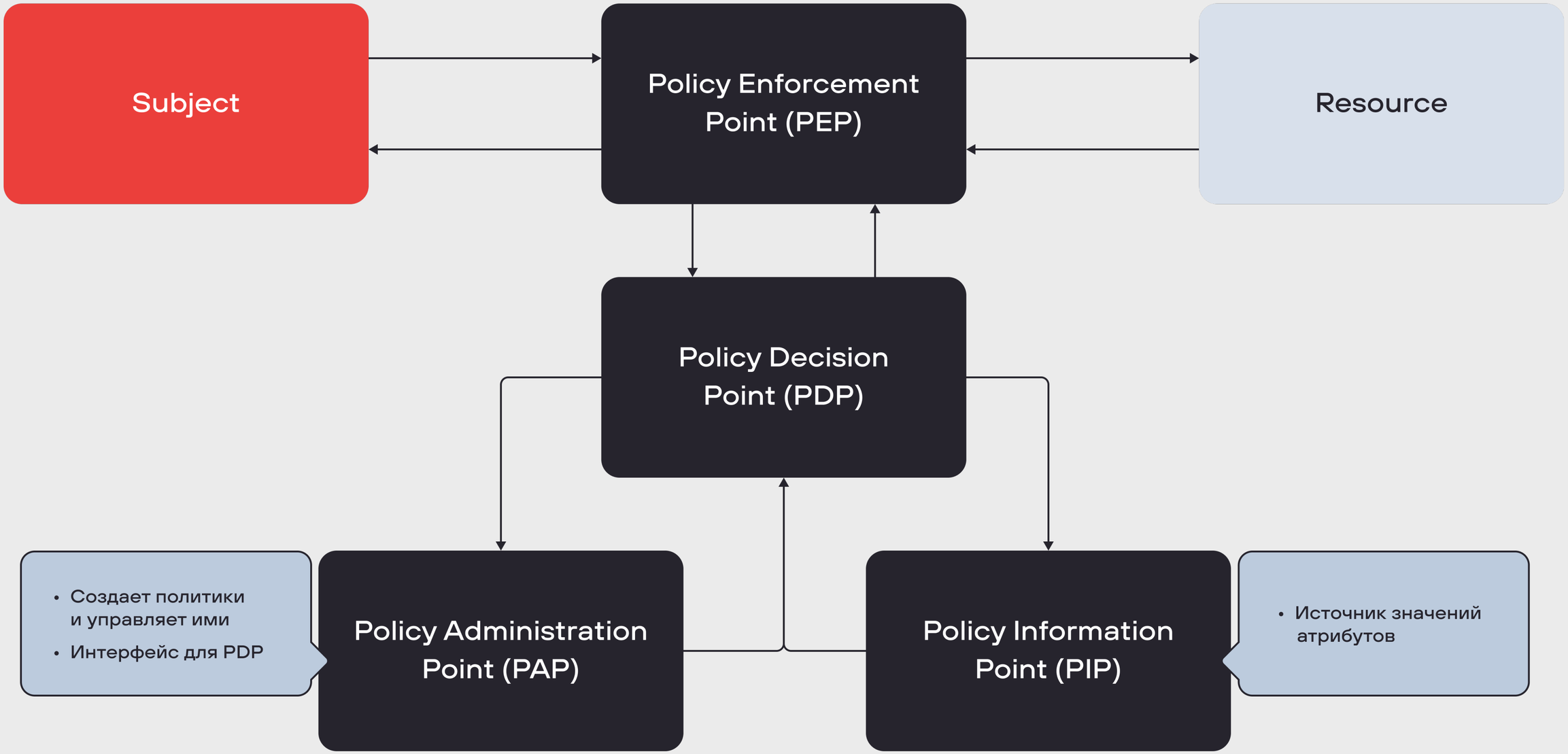
Компоненты XACML



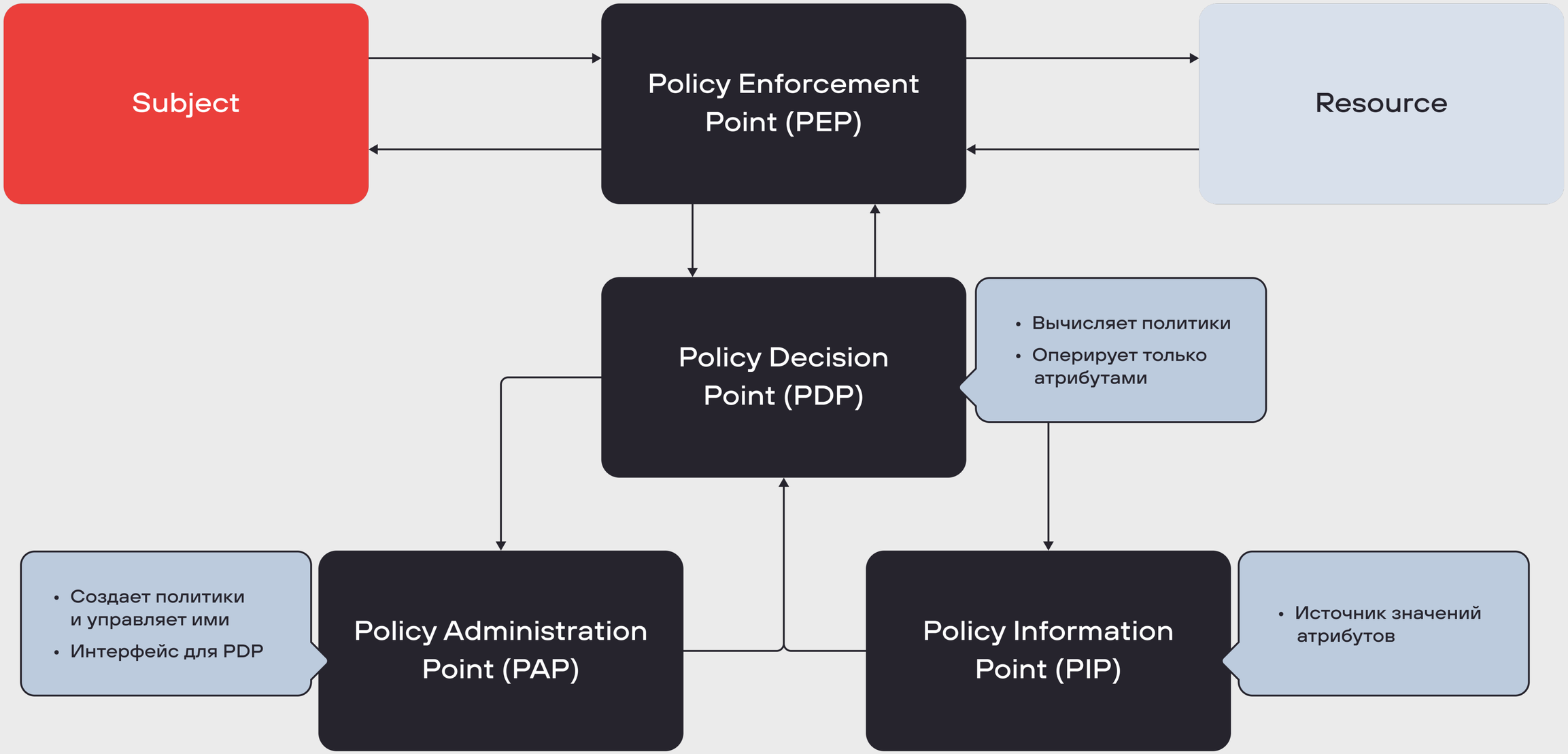
Компоненты XACML



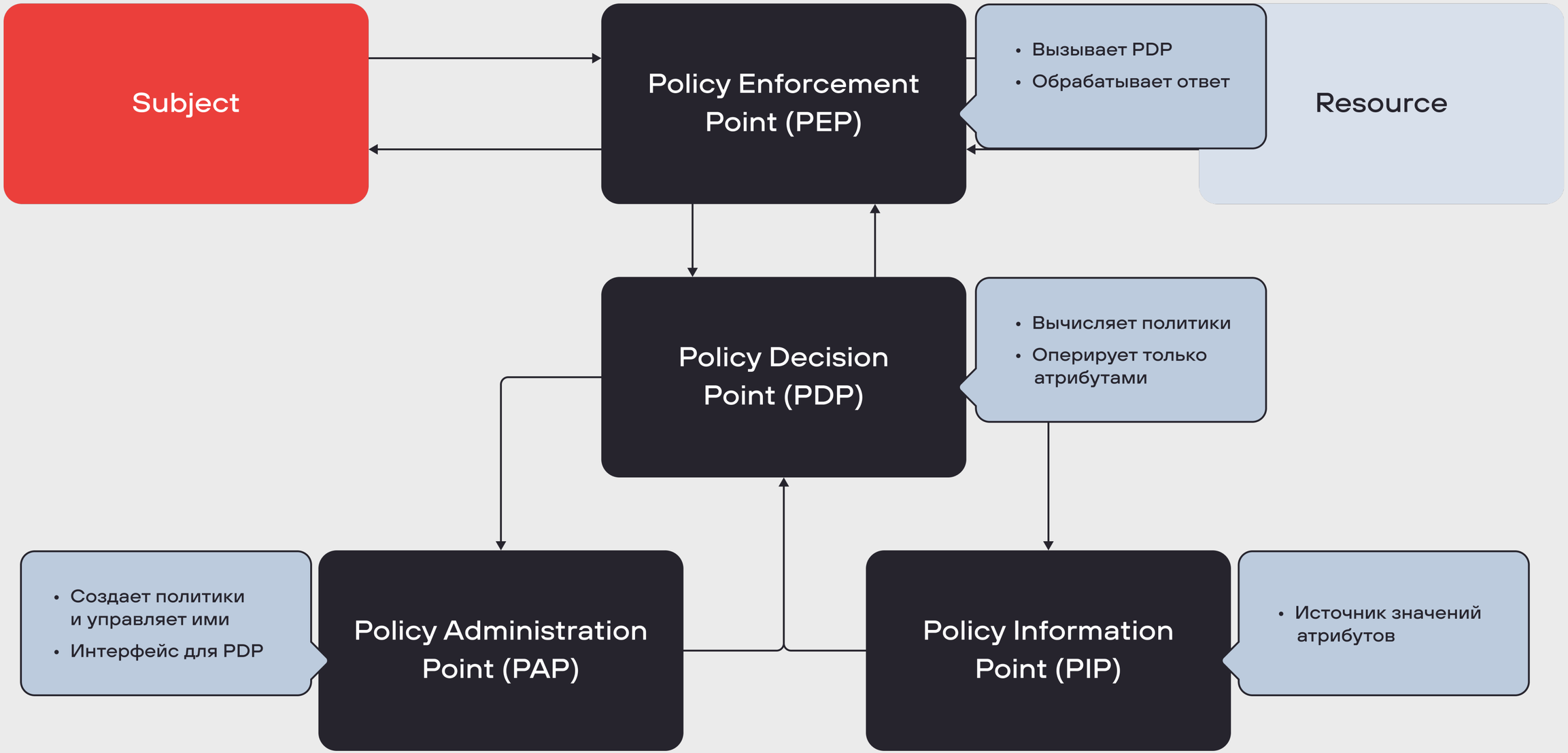
Компоненты XACML



Компоненты XACML



Компоненты XACML



Компоненты XACML

PAP	Policy Administration Point	• Создает и администрирует политики • Интерфейс получения политик
PIP	Policy Information Point	Источник значений атрибутов
PDP	Policy Decision Point	• Вычисляет политики • Оперирует только атрибутами
PEP	Policy Enforcement Point	Вызывает PDP и обрабатывает ответ

Взаимодействие компонентов XACML

Client

Policy Enforcement Point (PEP)

Obligations service

Policy Decision Point (PDP)

Resource

Subject

Environment

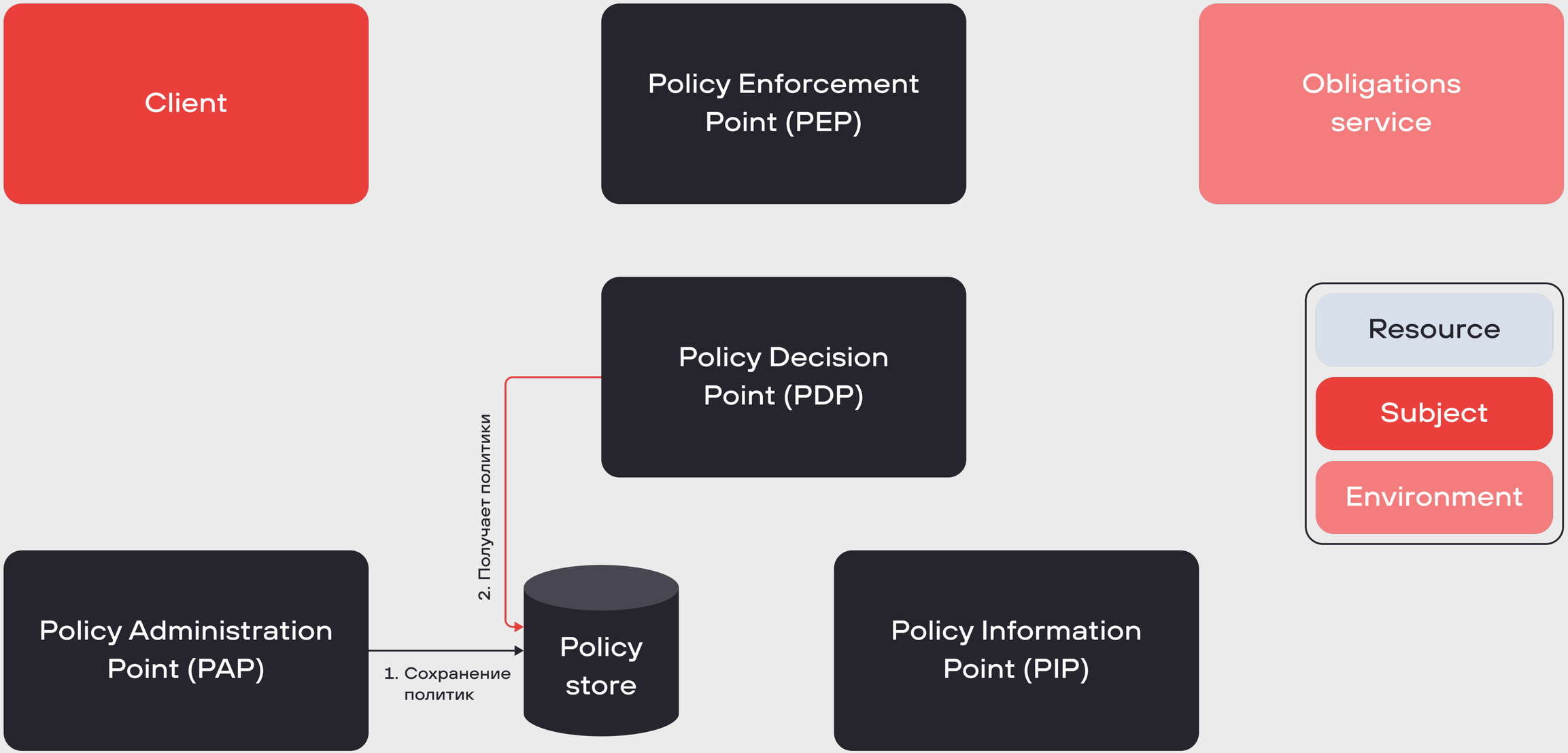
Policy Administration Point (PAP)

1. Сохранение политик

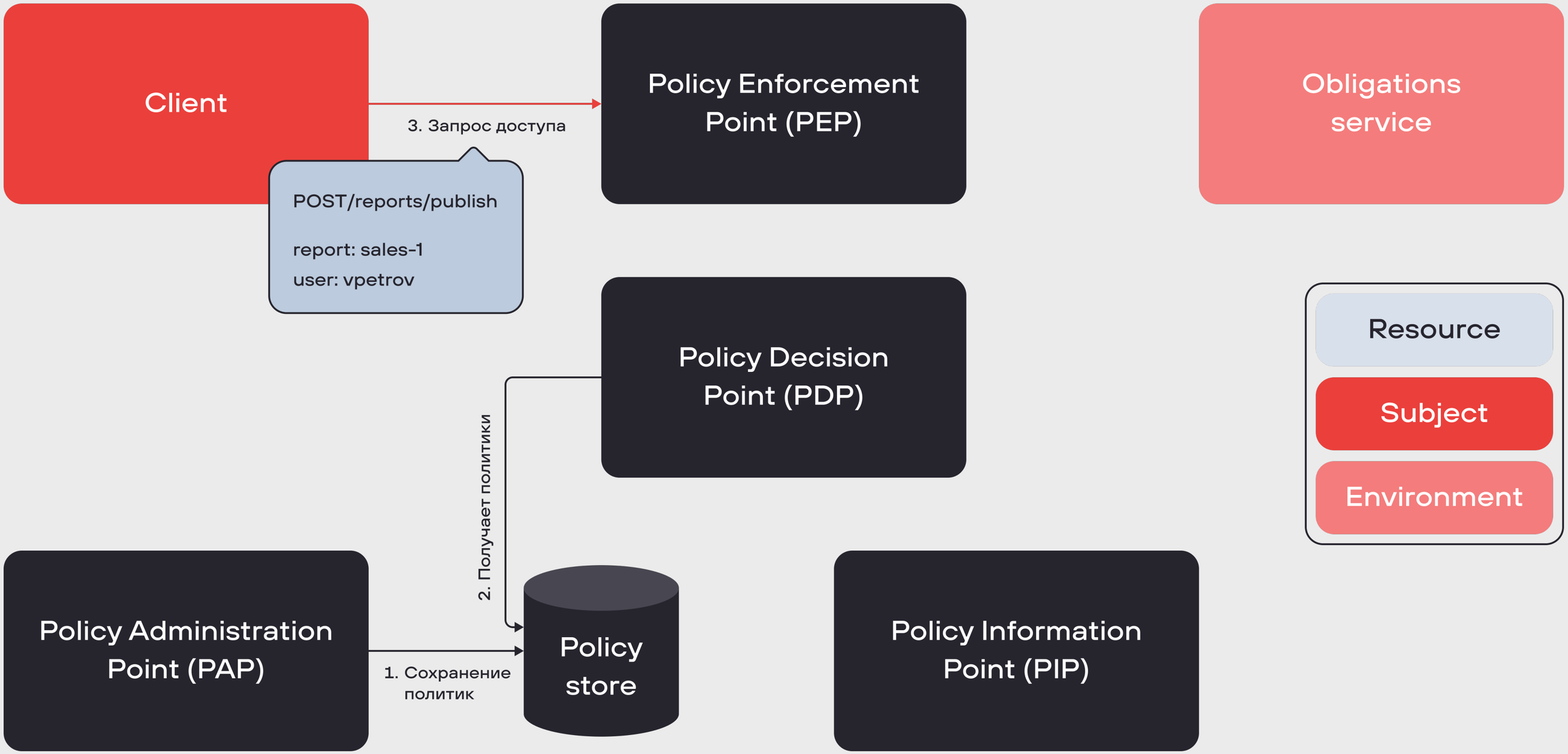
Policy store

Policy Information Point (PIP)

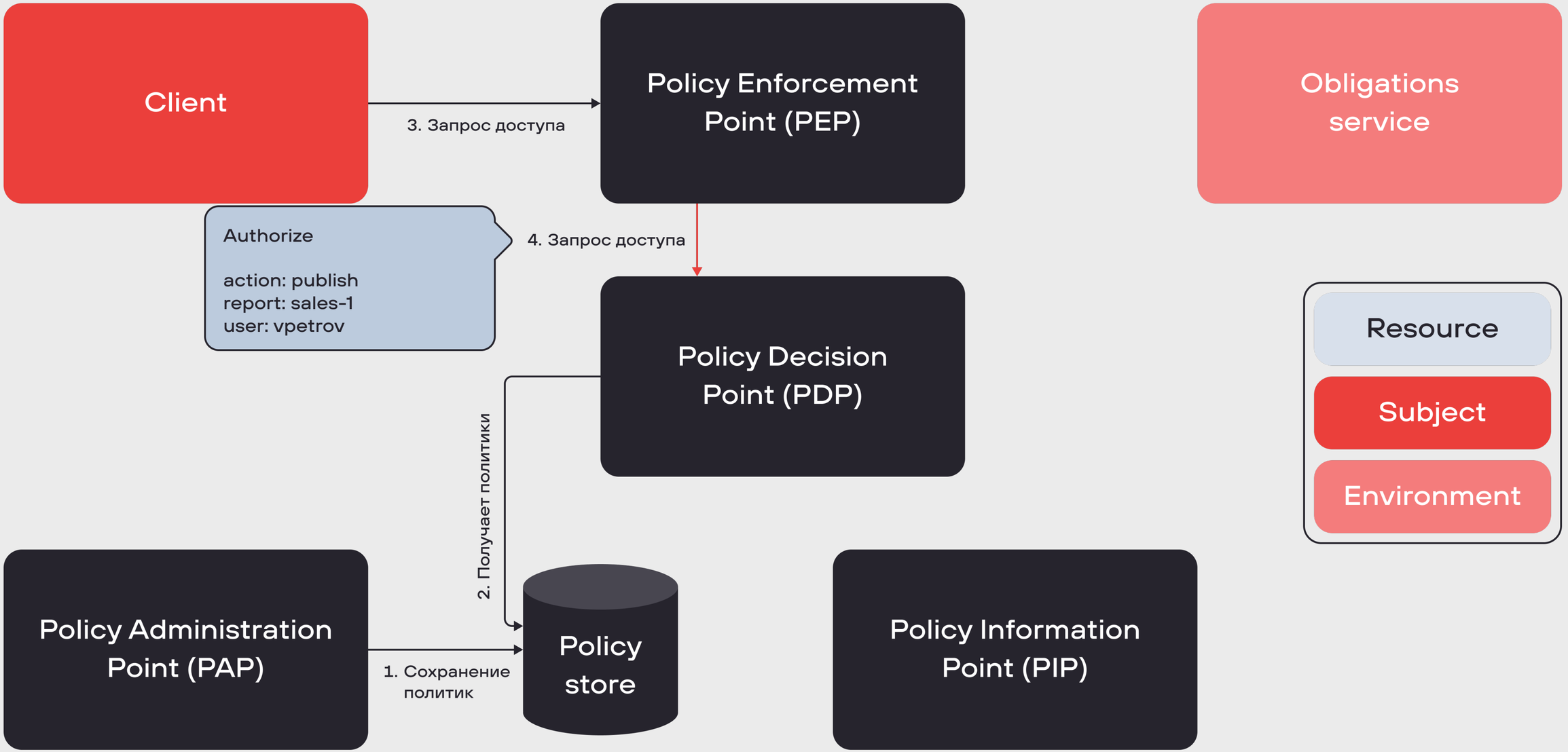
Взаимодействие компонентов XACML



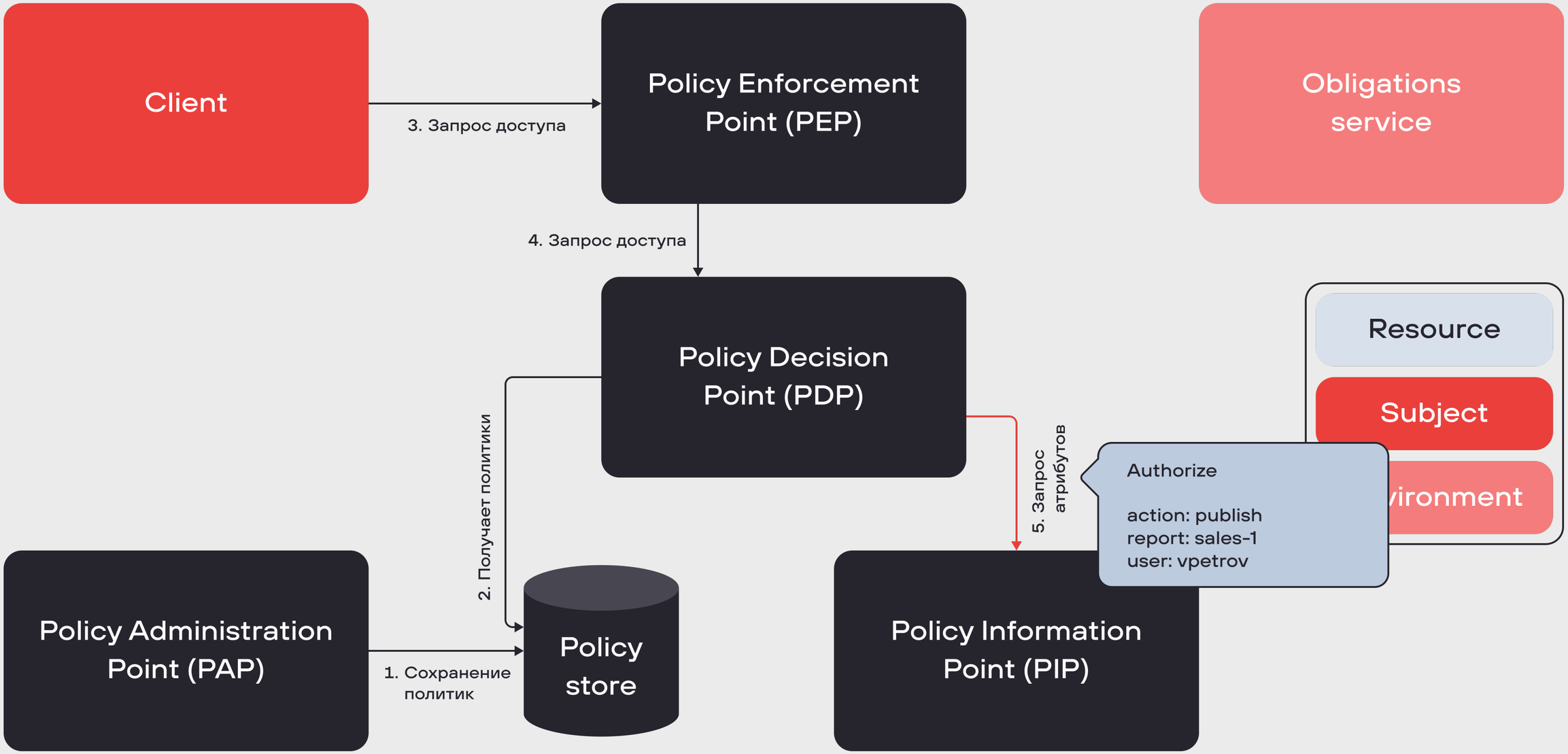
Взаимодействие компонентов XACML



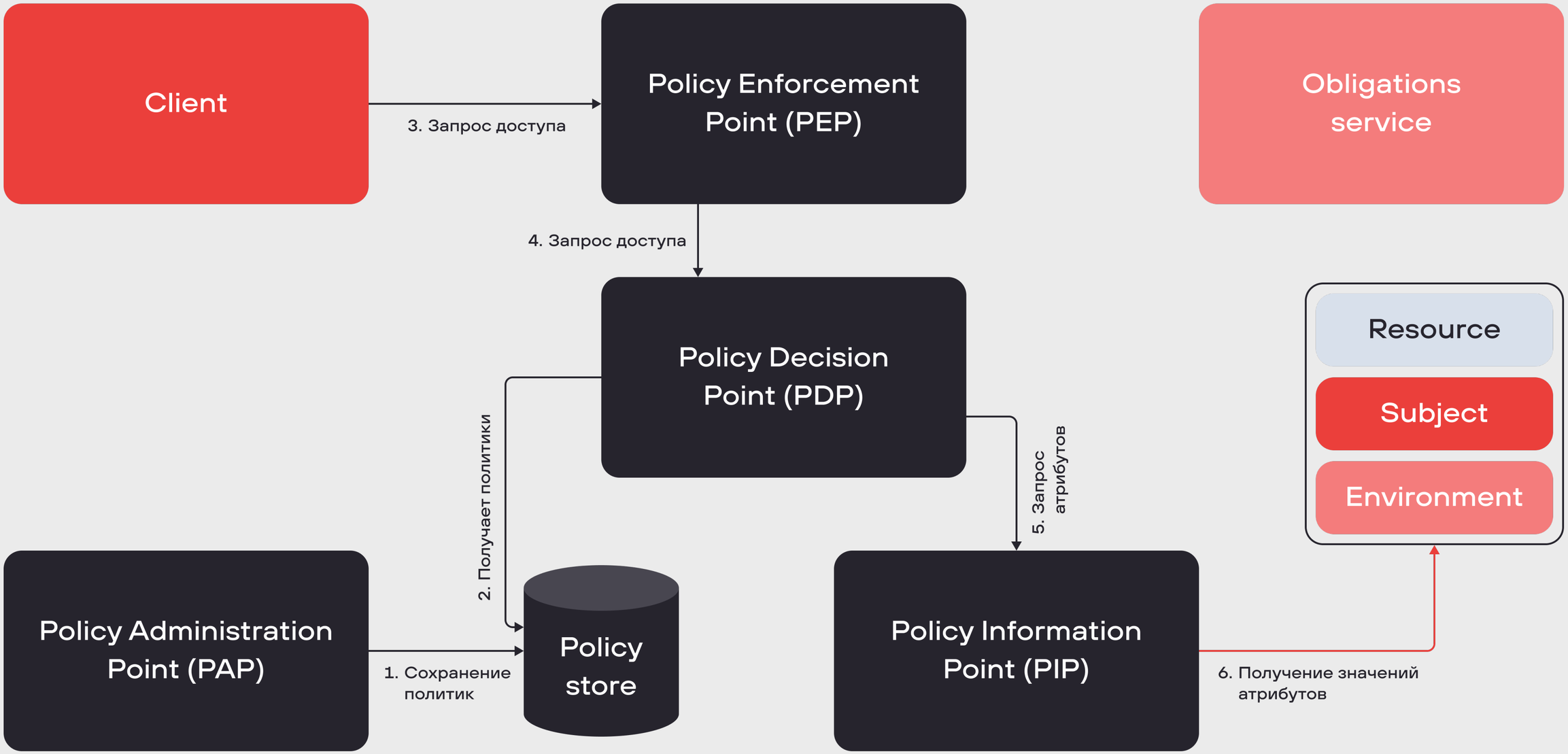
Взаимодействие компонентов XACML



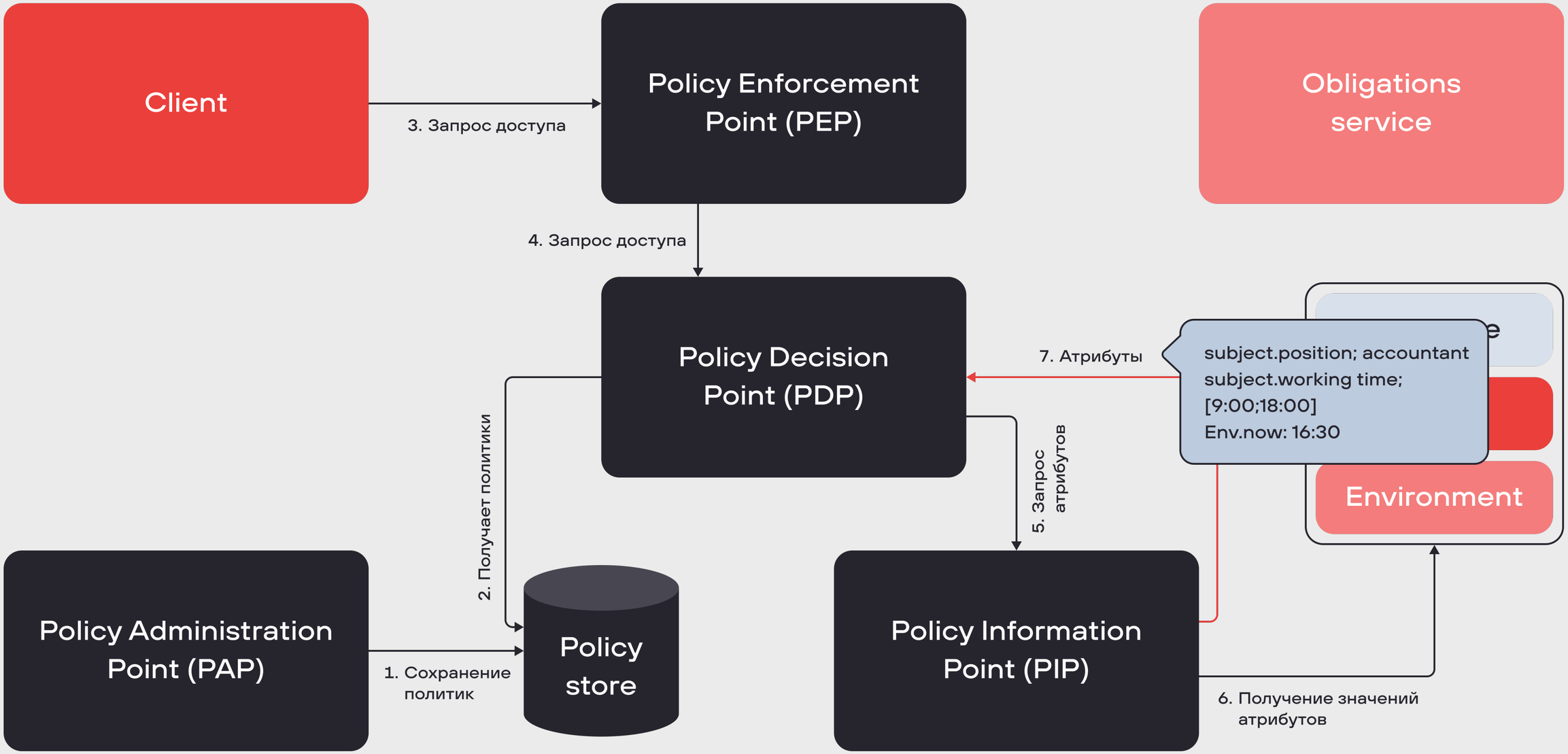
Взаимодействие компонентов XACML



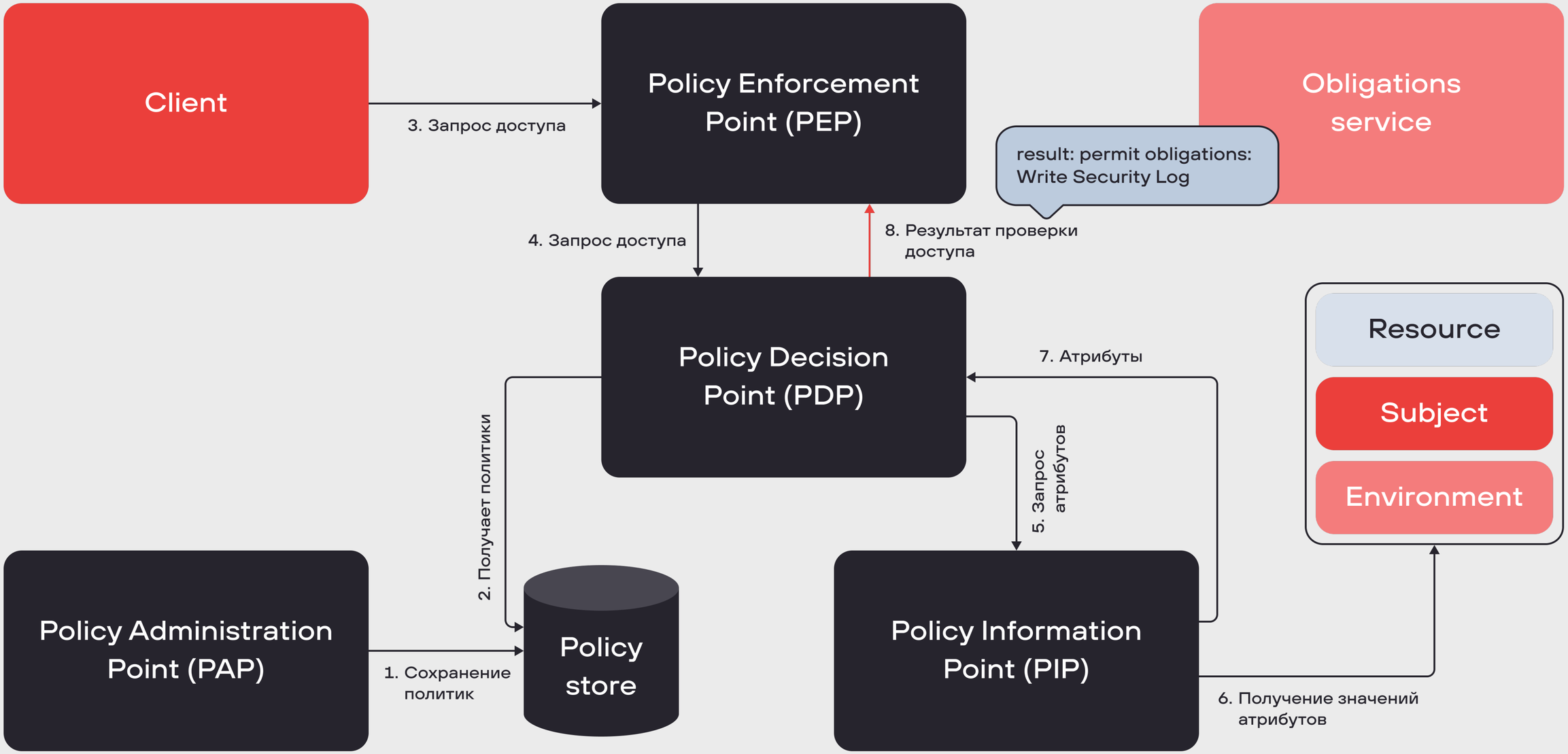
Взаимодействие компонентов XACML



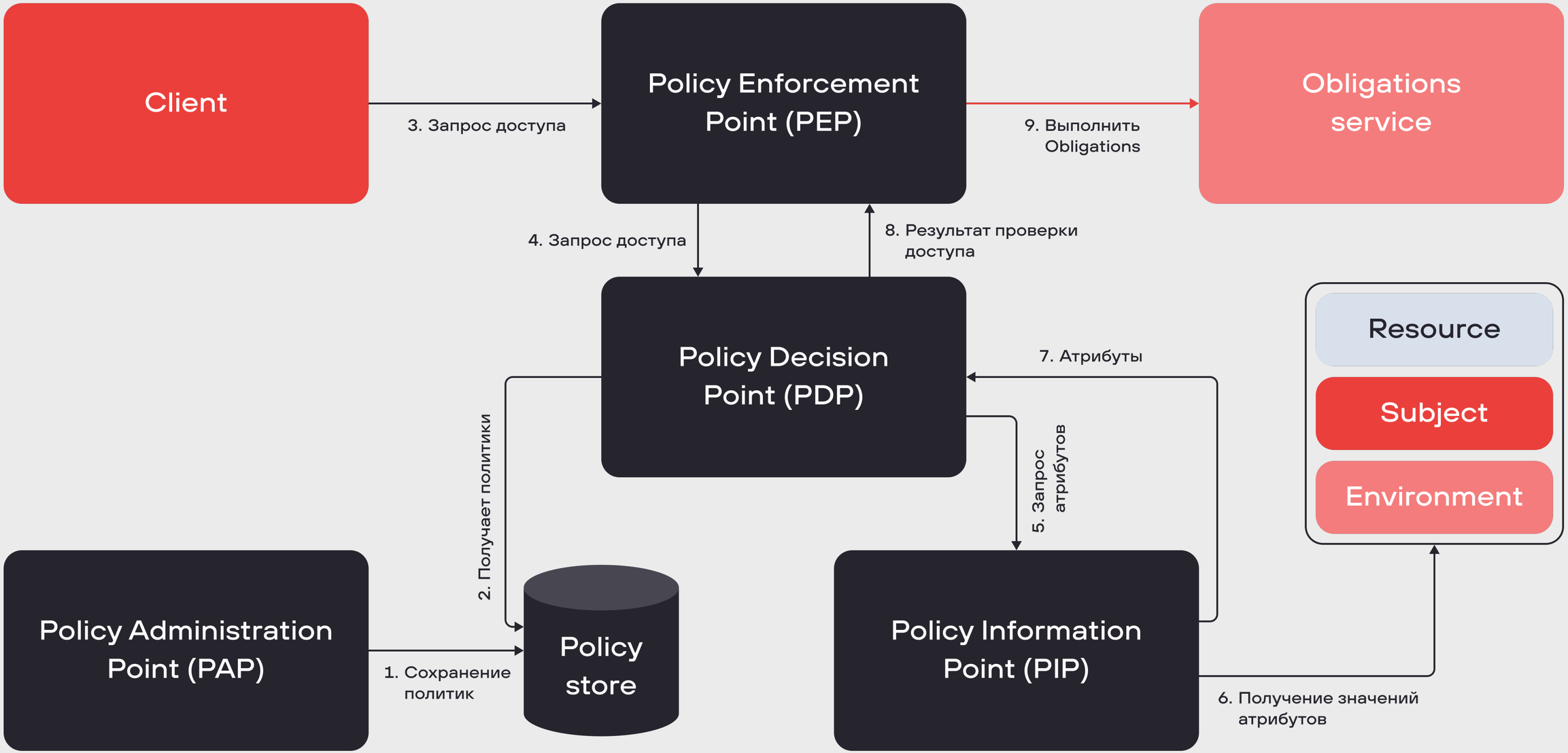
Взаимодействие компонентов XACML



Взаимодействие компонентов XACML

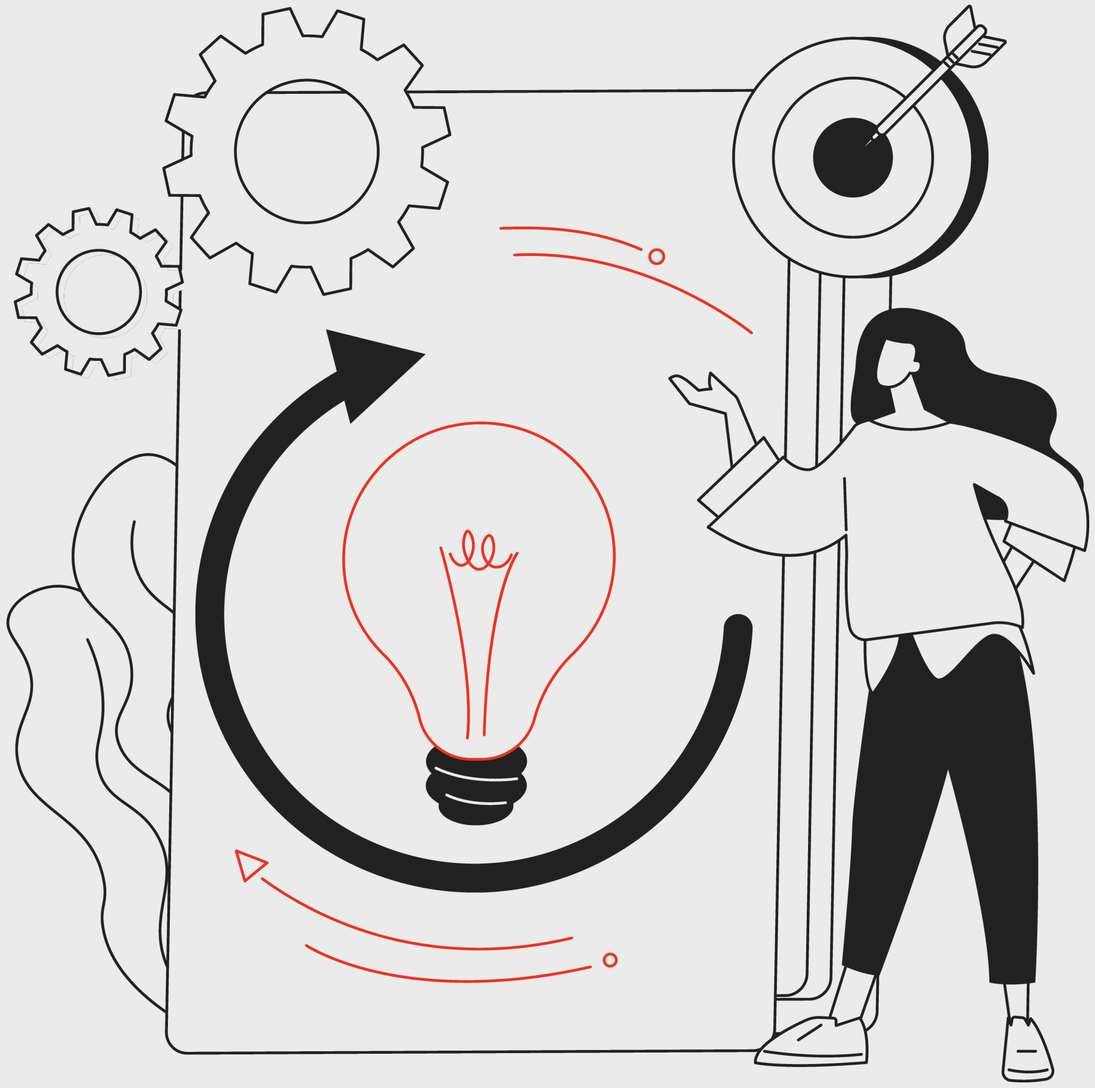


Взаимодействие компонентов XACML



Плюсы XACML

- Отраслевой стандарт
- Несколько реализаций (как опенсорсных, так и проприетарных)



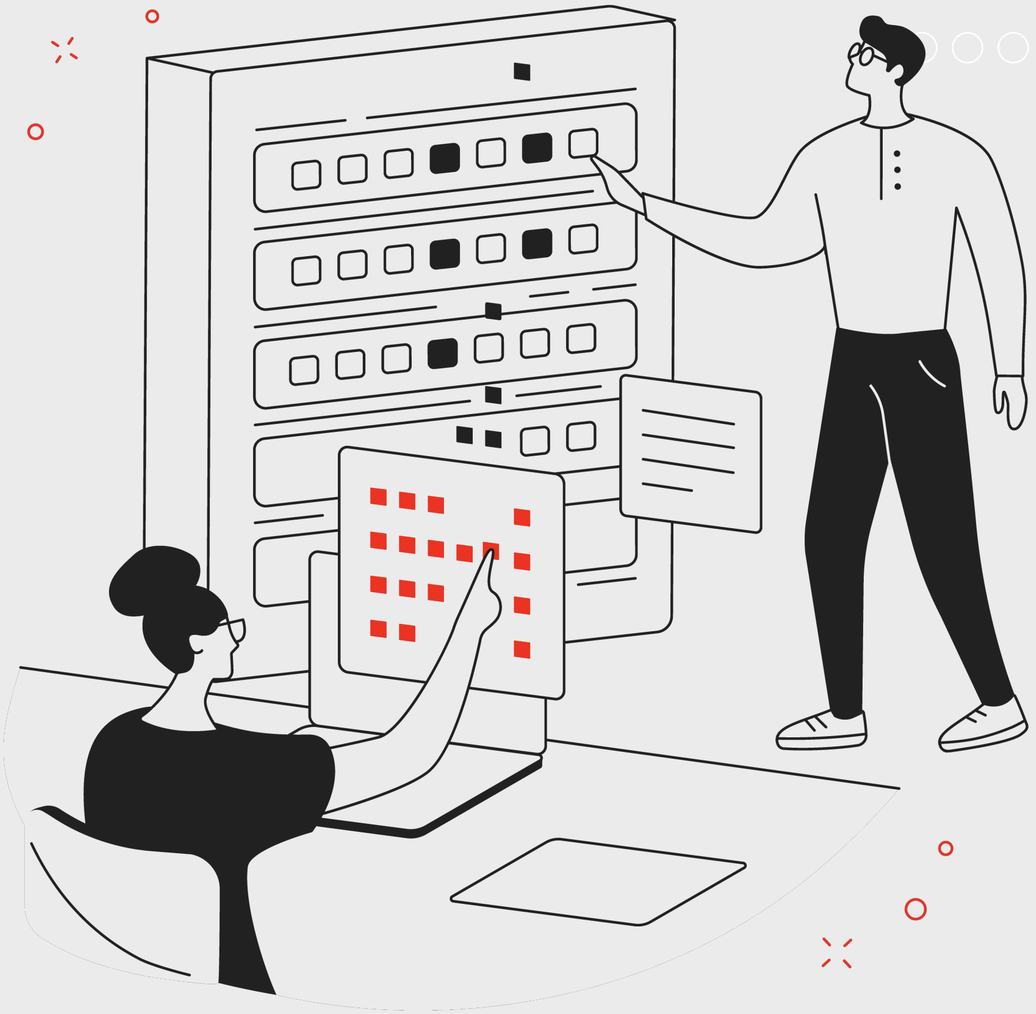
Недостатки XACML

Все на XML :)

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
  xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17
  http://docs.oasis-open.org/xacml/3.0/xacml-core-v3-schema-wd-17.xsd"
  PolicyId="urn:oasis:names:tc:xacml:3.0:example:SimplePolicy1"
  Version="1.0"
  RuleCombiningAlgId="identifier:rule-combining-algorithm:deny-overrides">
  <Description>
    Medi Corp access control policy
  </Description>
  <Target/>
  <Rule
    RuleId= "urn:oasis:names:tc:xacml:3.0:example:SimpleRule1"
    Effect="Permit">
    <Description>
      Any subject with an e-mail name in the med.example.com domain
      can perform any action on any resource.
    </Description>
    <Target>
      <AnyOf>
        <AllOf>
          <Match
            MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string"
              >med.example.com</AttributeValue>
          <AttributeDesignator
            MustBePresent="false"
            Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
            AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
            DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"/>
          </Match>
        </AllOf>
      </AnyOf>
    </Target>
  </Rule>
</Policy>
```

Недостатки XACML

- Все на XML :)
- Стандарт не определяет, как работать со множеством ресурсов

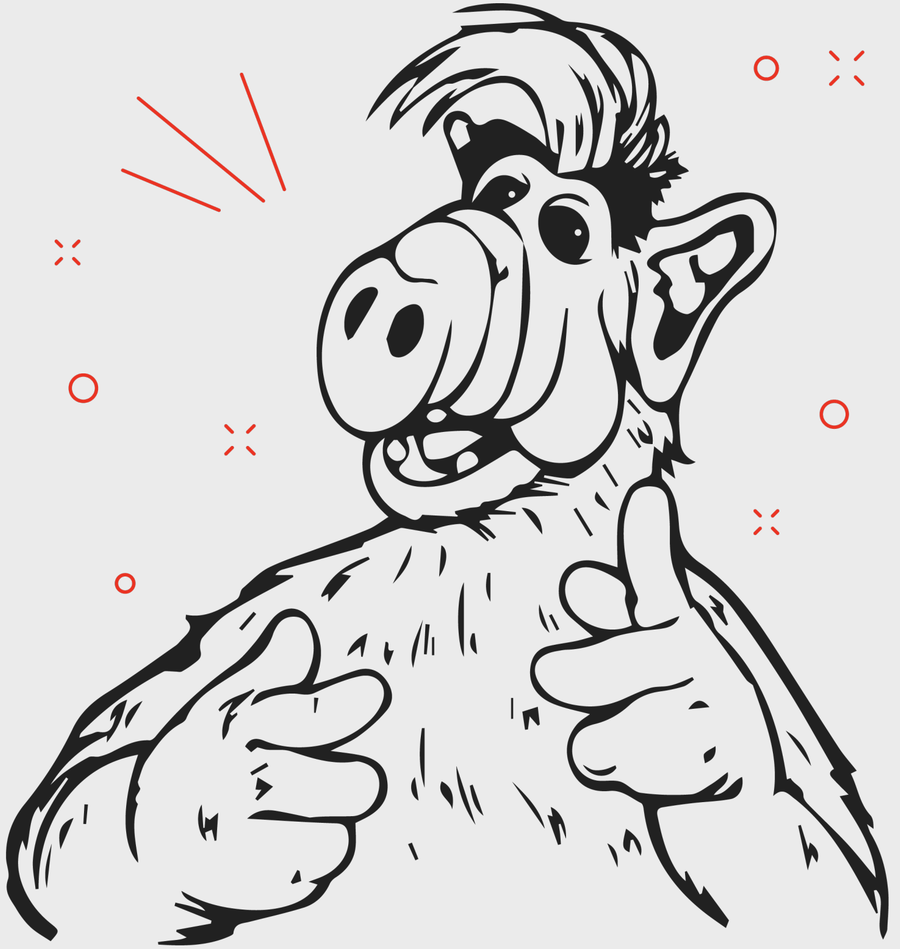


Недостатки XACML

- Все на XML :)
- Страдает latency в стандартной схеме
- Сложно обеспечить целостность данных

Axiomatics ALFA

Abbreviated Language For Authorization



Пример политики

```
policy report{
    target clause resource-type == "report"
    apply firstApplicable

    rule publishReport{
        target clause action == "publish" and user.role == "accountant"
        permit
        condition user.id == owner
    }
}
```

Недостатки ALFA

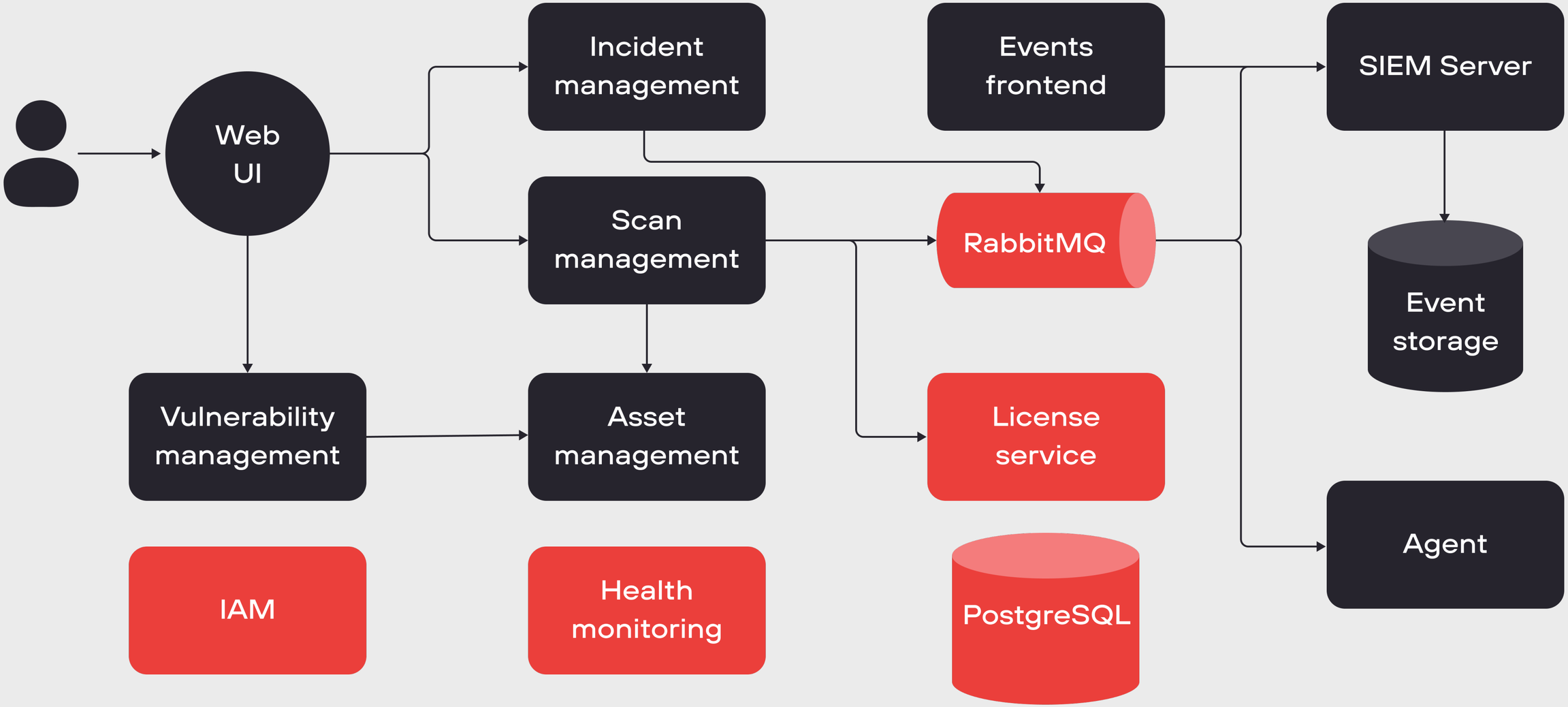
Неявная
параметризация
получения атрибутов

Недостатки ALFA

- Неявная параметризация получения атрибутов
- Возможности индексации политик ограничены

От теории к практике

Схема MaxPatrol 10



ABAC для разделения продукта

- Выделение редакций через политики доступа
- Возможность формирования новых редакций с минимальным изменением кодовой базы

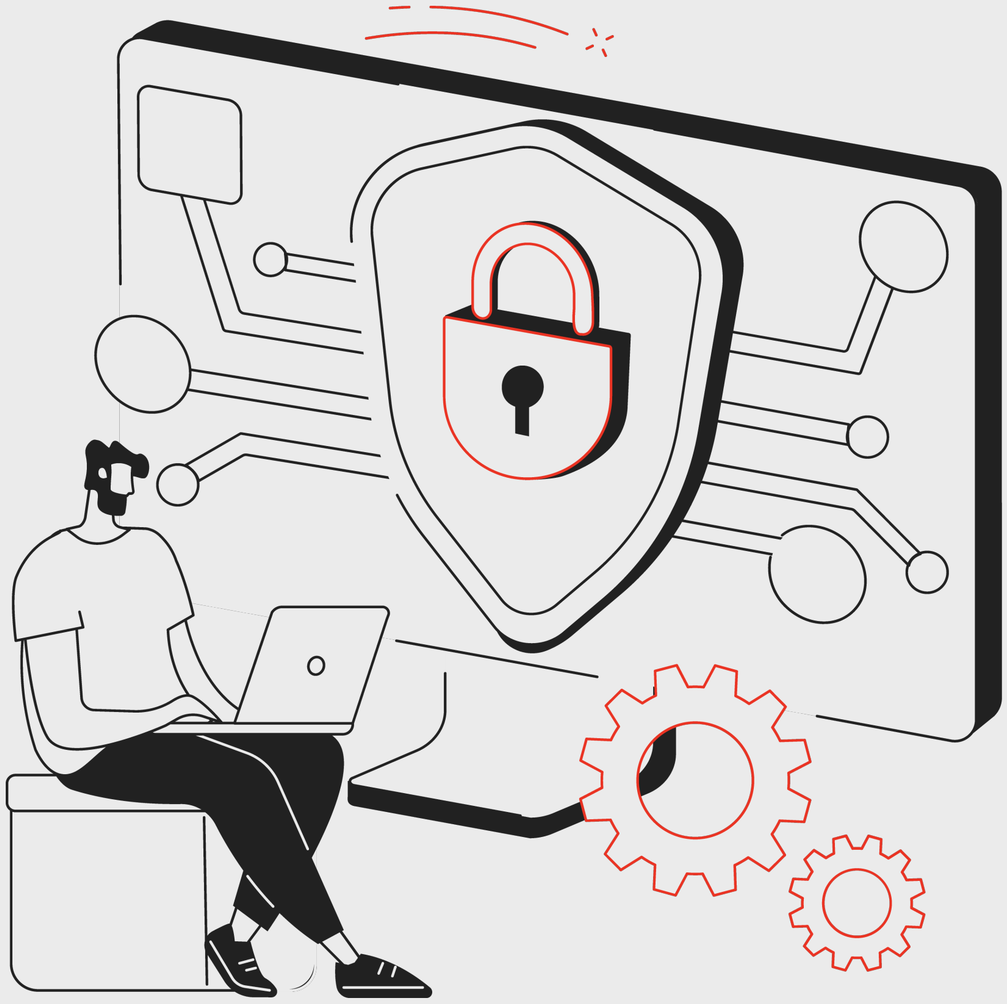


Схема MaxPatrol 10

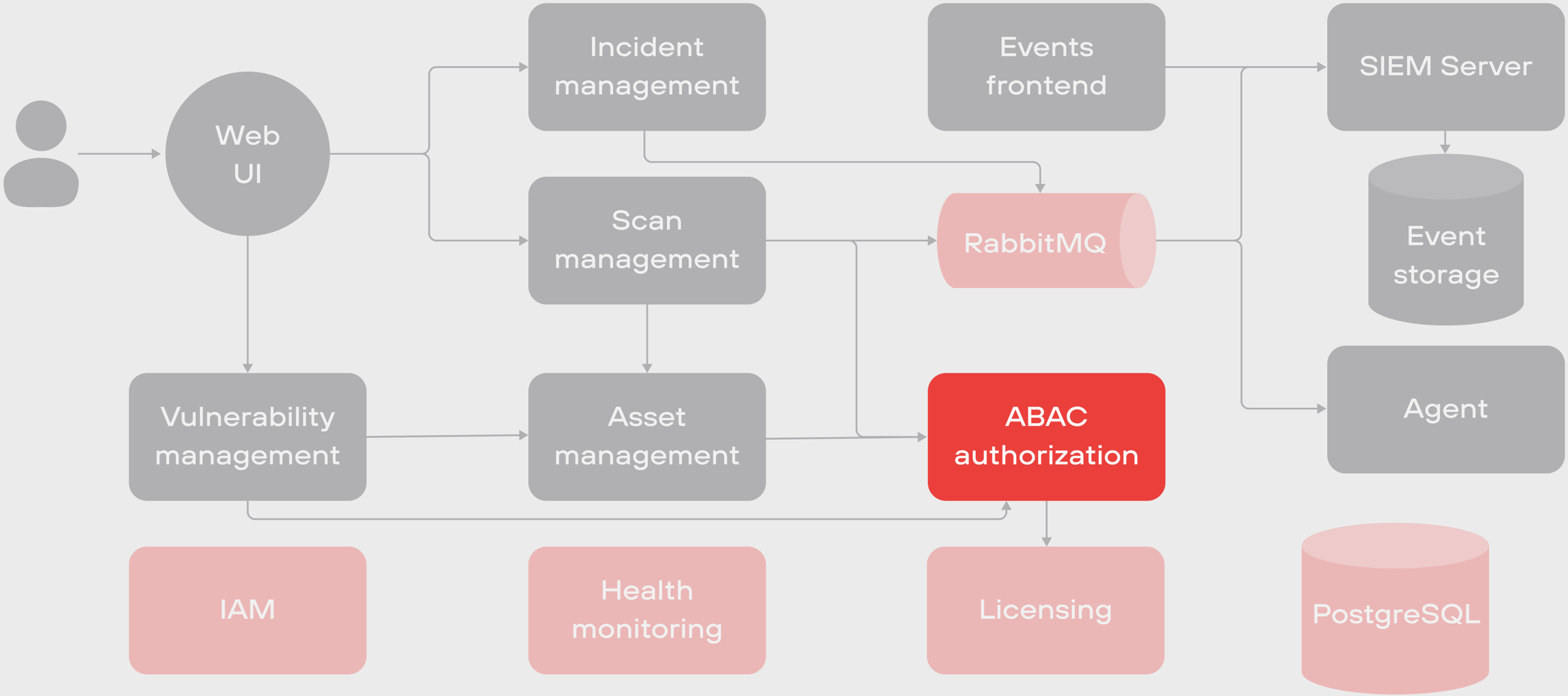


Схема атрибутов ресурса

```yaml
resources:
    report:
      idType: Uuid
      actions:
        read:
        publish:
attributes:
    actions:
      type: String
    "report[].id":
      type: Uuid
    "report[].owner":
      type: String

```

P-ALFA

```
policy report{
    target clause resource-type == "report"
    apply firstApplicable

    rule publishReport{
        target clause action == "publish" and user.role == "accountant"
        permit
        condition user.id == owner
    }
}
```

```
policy report
resource = "report"

rule publishReport
    target clause action == "publish" and user.role == "accountant"
    condition report[id = @id].owner = user.id
```

Фильтр типа

```
policy report
resource = "report"

rule publishReport
    target clause action == "publish" and user.role == "accountant"
    condition report[id = @id].owner = user.id
```

Значения атрибутов

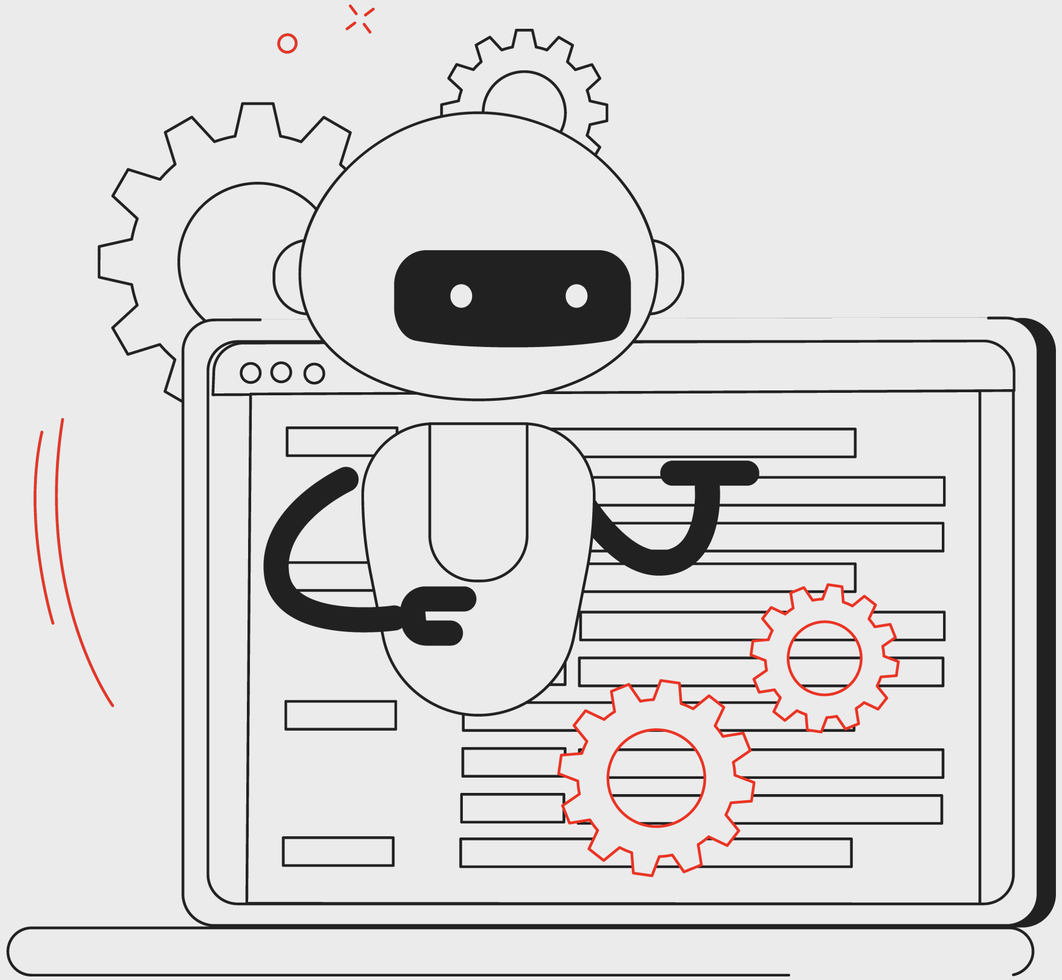
```
policy report
resource = "report"

rule publishReport
    target clause action == "publish" and user.role == "accountant"
    condition report[id = @id].owner = user.id
```

Группа атрибутов

report[] — группа атрибутов

```
class Report
{
    public Guid Id { get; }
    public string Owner { get; }
}
```

Фильтр



1

report[id = @id].owner == user.id

2

any(report[id = @id, owner = user.id])

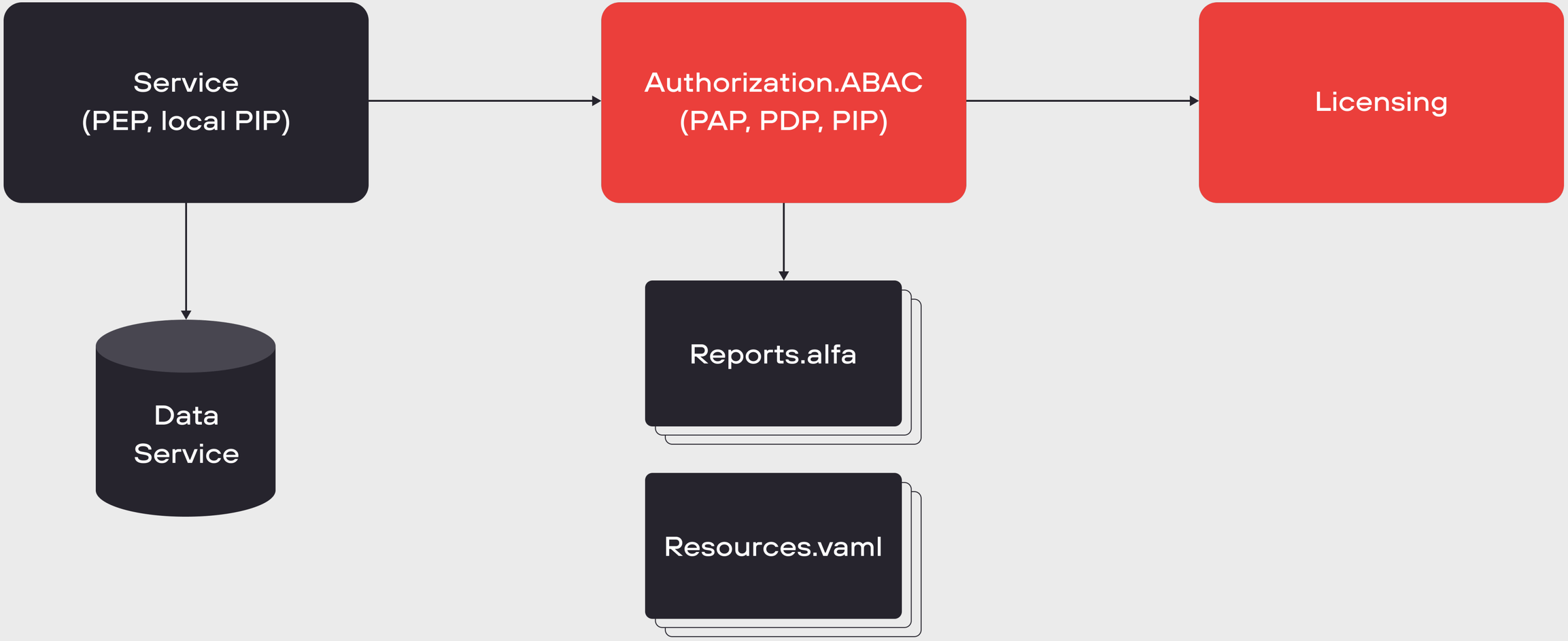
Доработки XACML

- Микросервис может зарегистрировать в PEP локальный PIP
- Новый результат проверки — Deferred

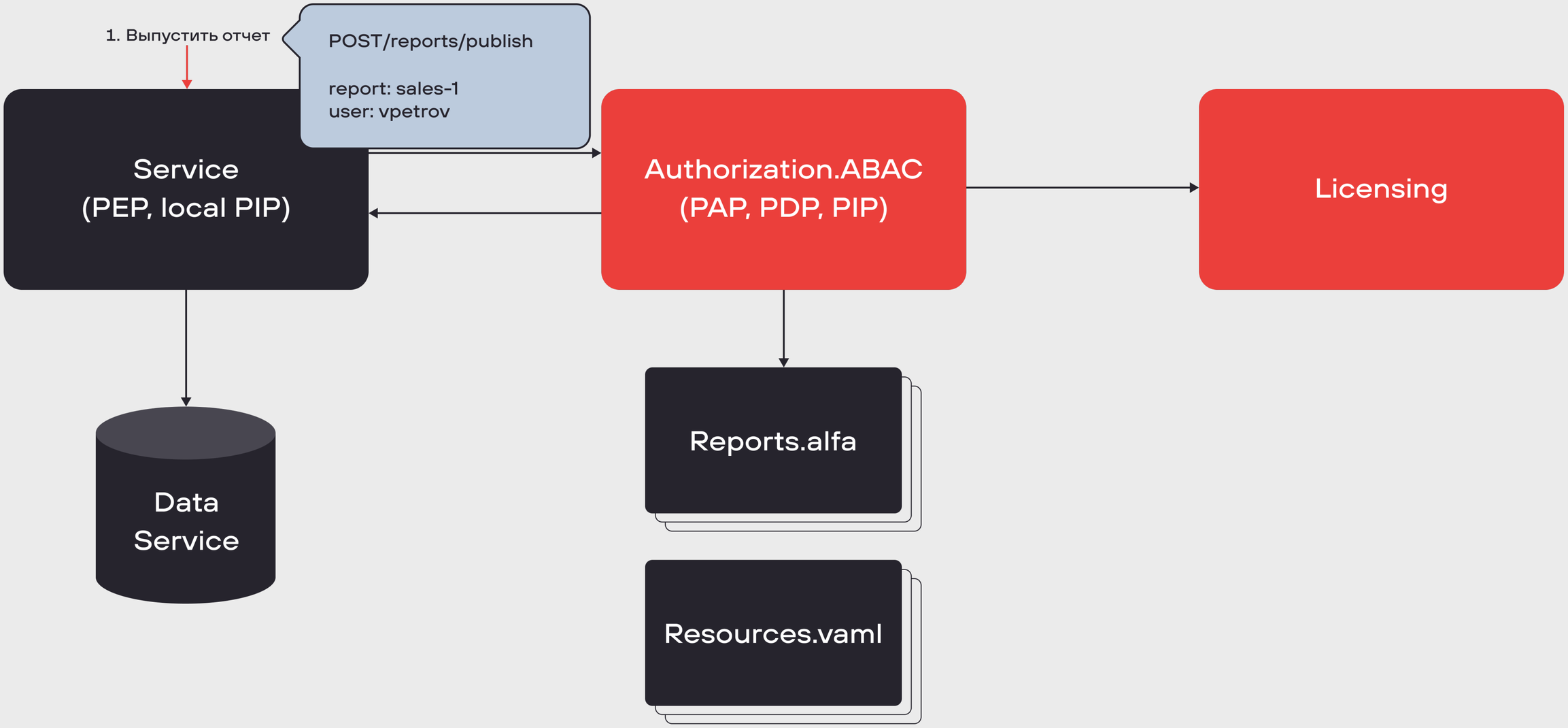
Компоненты XACML

PAP	Policy Administration Point	• Создает и администрирует политики • Интерфейс получения политик
PIP	Policy Information Point	Источник значений атрибутов
PDP	Policy Decision Point	• Вычисляет политики • Оперирует только атрибутами
PEP	Policy Enforcement Point	Вызывает PDP и обрабатывает ответ

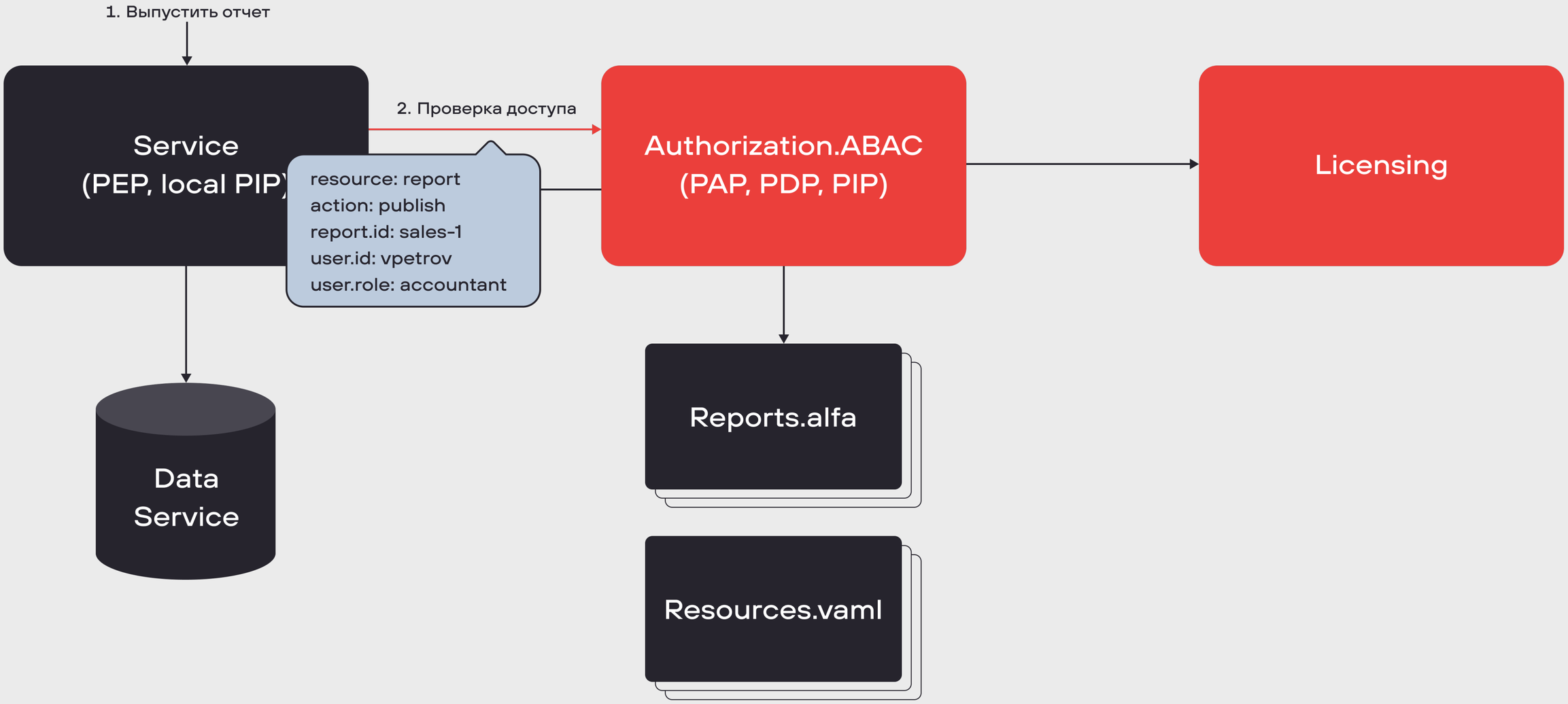
Проверка авторизации



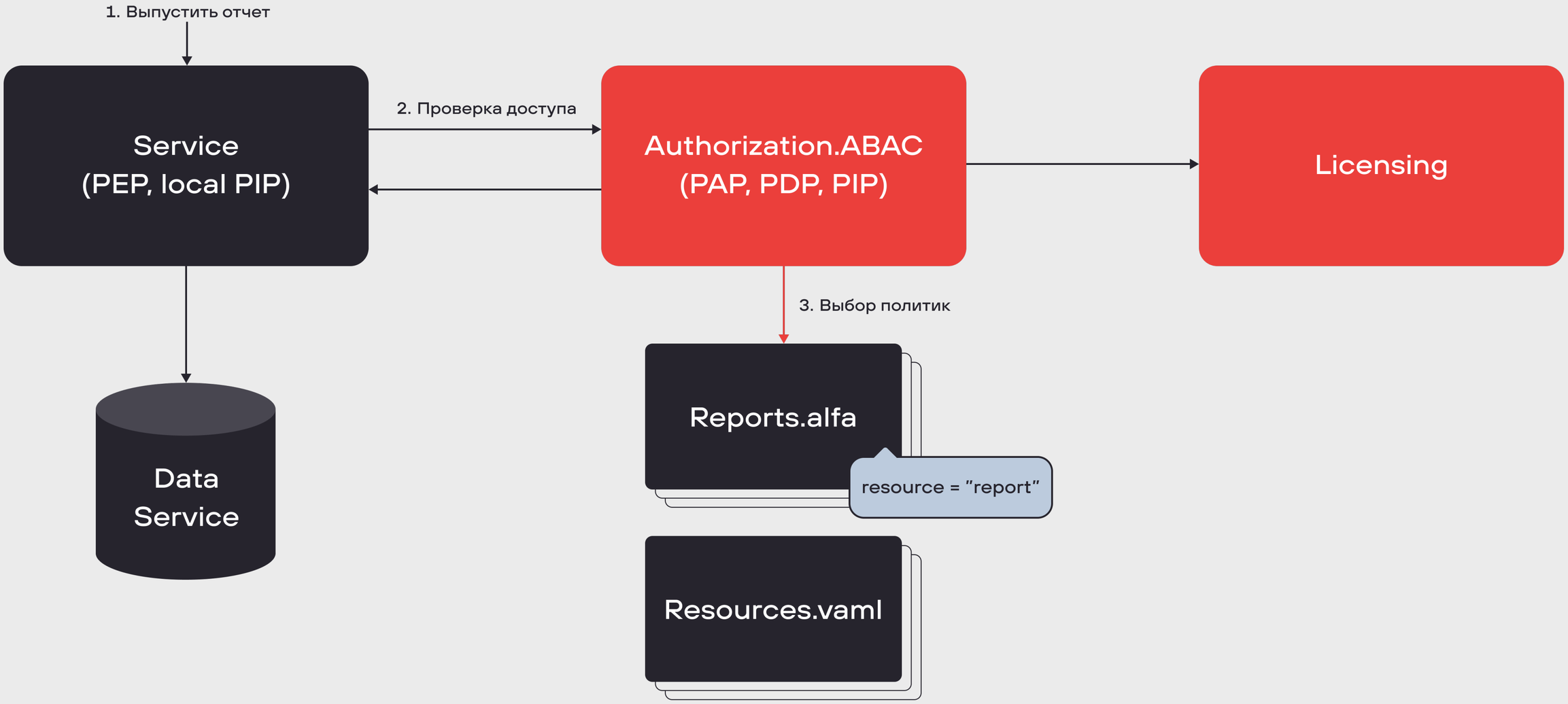
Проверка авторизации



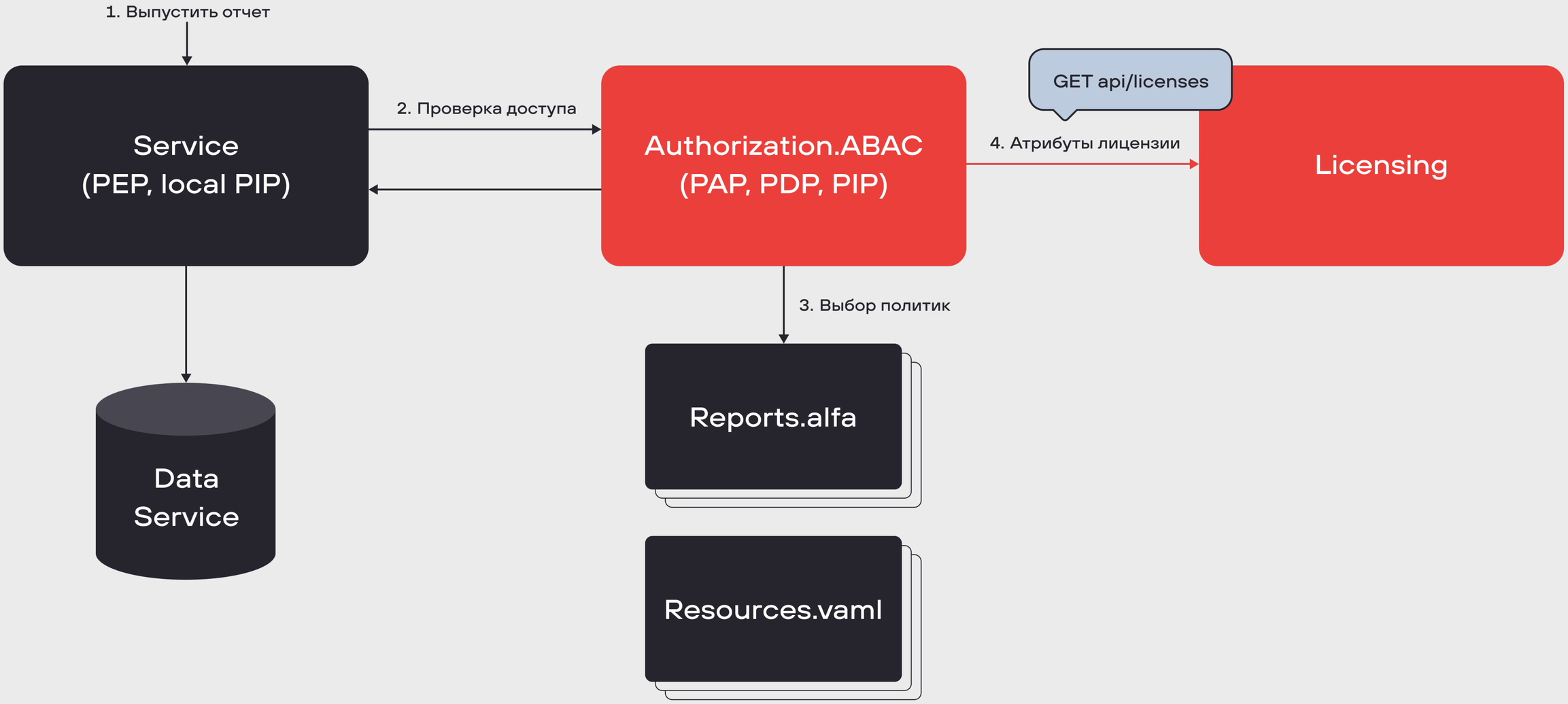
Проверка авторизации



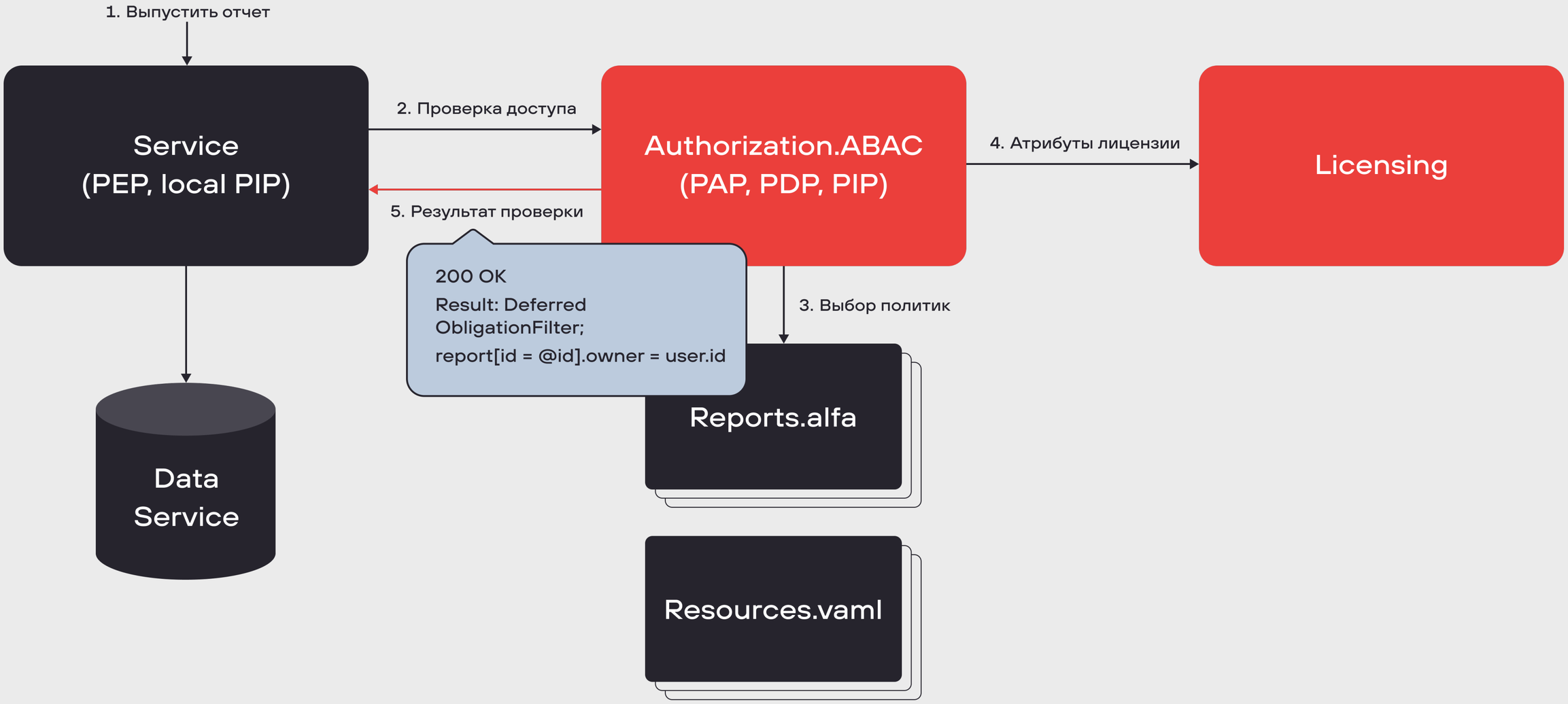
Проверка авторизации



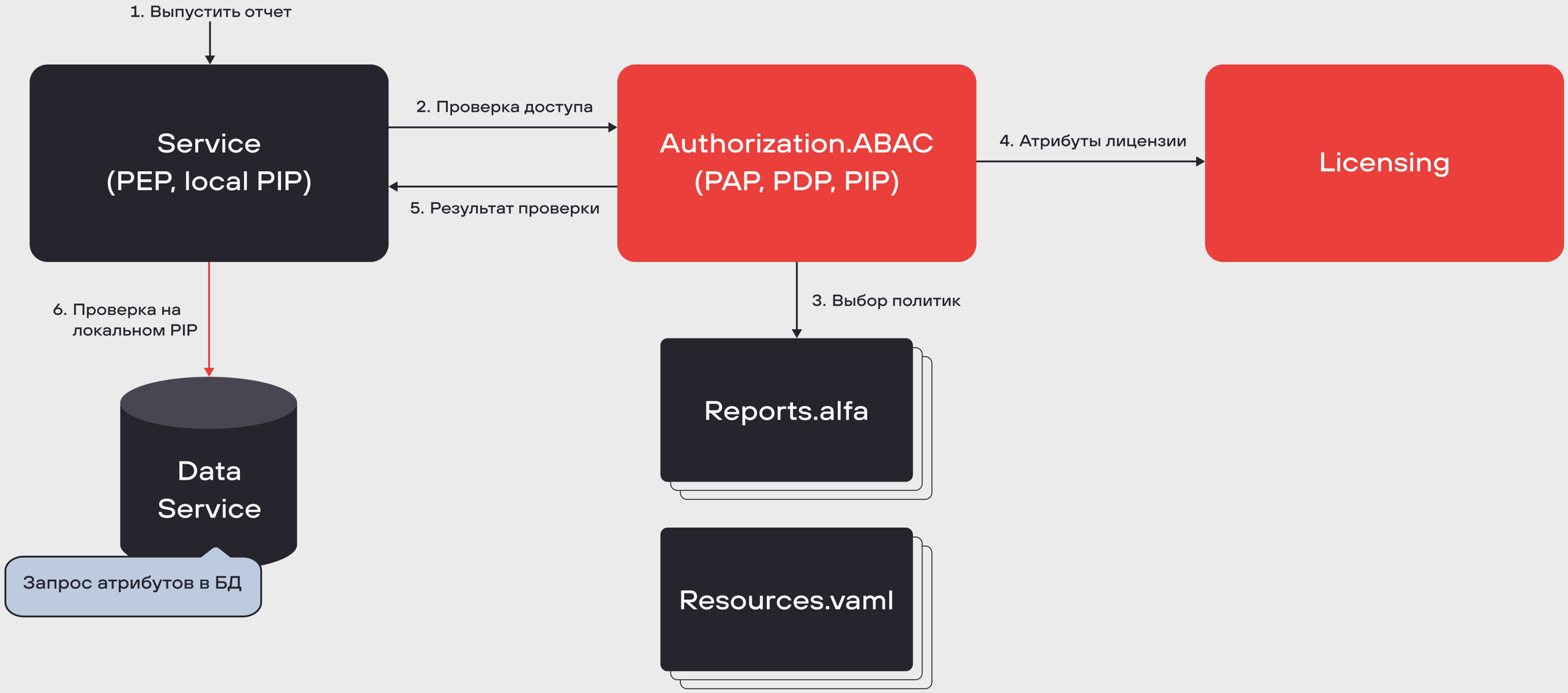
Проверка авторизации



Проверка авторизации



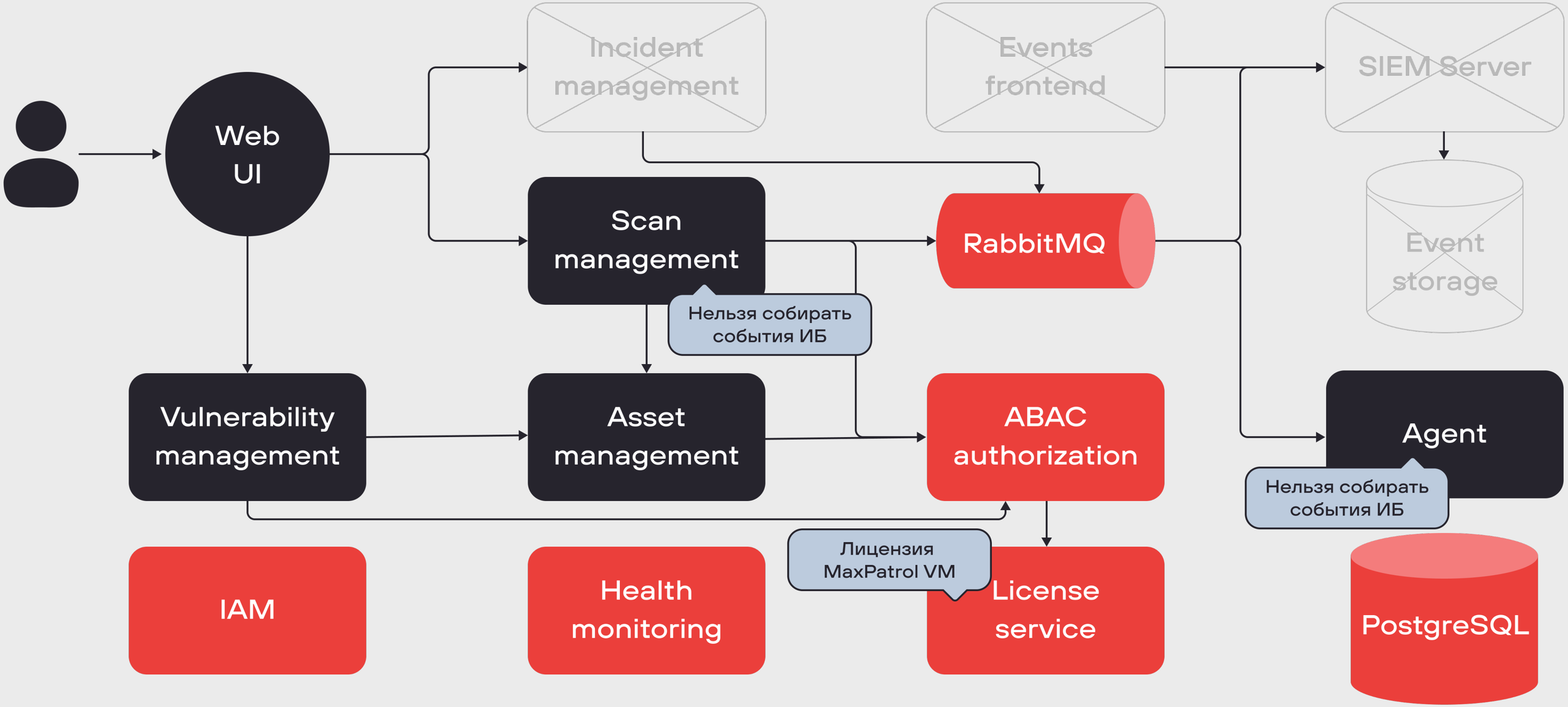
Проверка авторизации



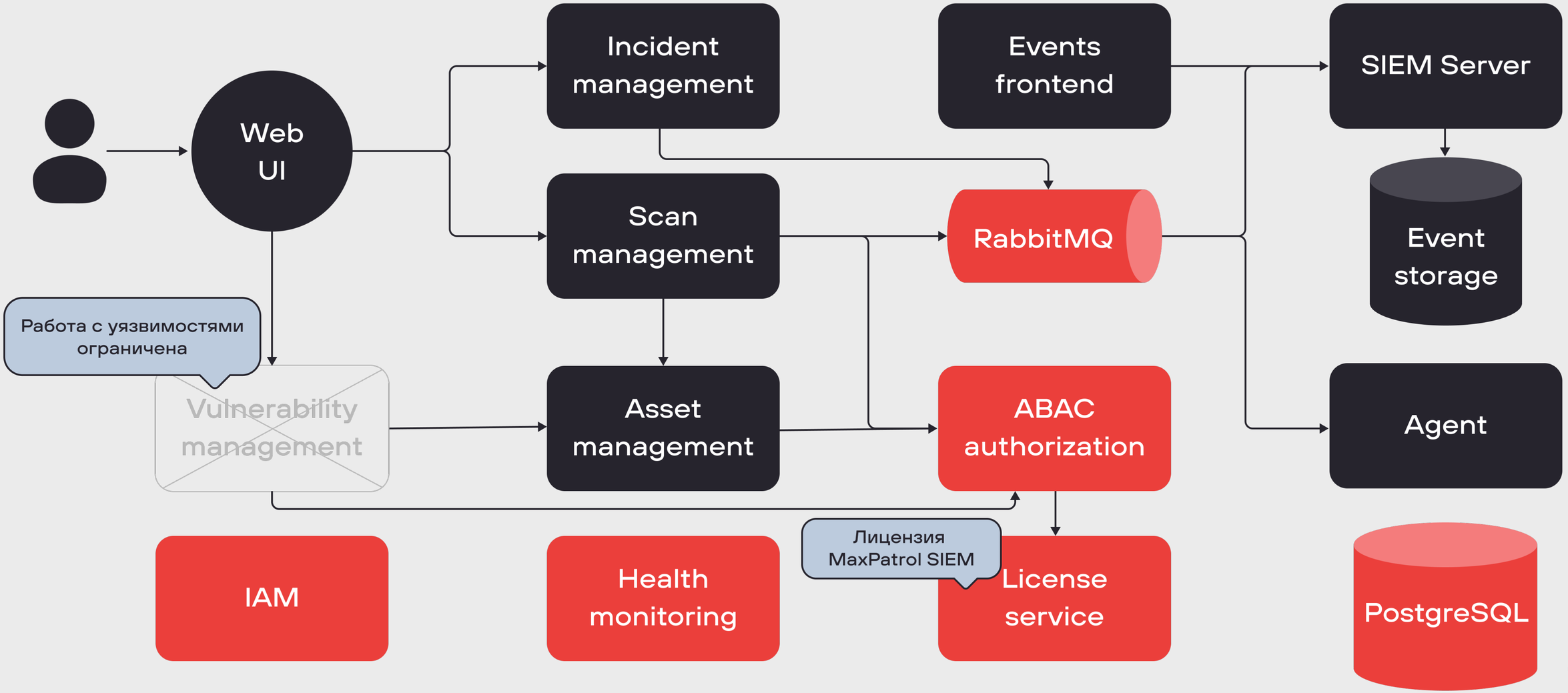
Итоги

- Получилось сделать расширяемый механизм управления редакциями
- Механизм учитывает разрешения пользователя

MaxPatrol VM (в составе MaxPatrol 10)

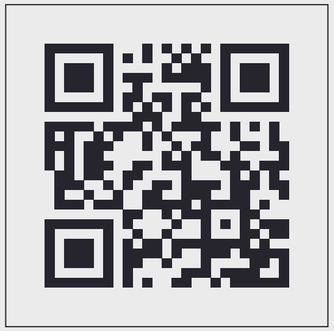


MaxPatrol SIEM (в составе MaxPatrol 10)



habrahabr.ru/
company/pt

t.me/
positive_technologies



vk.com/
ptsecurity

t.me/
positive_investing

Вопросы

Дополнения

Регистрация локального PIP

```
config.AddAbac(
    c => c
        .AddPostgreSql(
            connectionString: ctx => ctx.GetInstance<IConfiguration<PgSqlSettings>>().ConnectionString
            builder =>
            {
                builder
                    .AddResourceType(
                        resourceName: AuthorixationResources.Reports,
                        cfg: cfg.FromTable(table: "reports", idColumn: "id"))
                    .AddAttributeGroup(
                        groupPrefix: "report[]",
                        cfg: cfg => cfg.FromTable("reports")
                            .AddProperty(
                                property: "id",
                                column: "id",
                                attributeConfig: ac => ac.WithType(ItemType.Uuid))
                            .AddProperty(
                                property: "owner",
                                column: "report->>'ownerId")
                    )
            }));
```

Запрос в базу

```
policy report
resource = "report"

rule publishReport
    target clause action == "publish" and user.role == "accountant"
    condition report[id = @id].owner = user.id
```

```
SELECT id
FROM reports
WHERE id = @id AND report->'owner' = @user_id
```

PIP с произвольной логикой

```
/// <summary>
/// Интерфейс для реалзиции PIP
/// </summary>
public interface IInformationPoint
{
    /// <summary>
    /// Получить значения поддерживаемых атрибутов
    /// </summary>
    Task<AttributeValueResult> GetAttributeValuesAsync(AttributeRequest attributeRequest, CancellationToken token);

    /// <summary>
    /// Возвращает список поддерживаемых атрибутов
    /// </summary>
    Task<IReadOnlyList<AttributeInfo>> GetSupportedAttributesAsync(CancellationToken token);
}
```

Запрос в PDP

```
try
{
    await _enforcementPoint.AuthorizeAsync(
        authorizationRequets: AuthorizationRequets.Create(
            resourceType: "report",
            action: "publish",
            cgf: cfg => cfg.WithId(id: reportId.ToGuid())),
        cancellationToken: cancellationToken);
}
catch (AccessDeniedException ex)
{
    // Доступ запрещен
}
catch (IntermediateException ex)
{
    throw new ServiceUnavailableException(message: "Unable to get uathorization decision");
}
```

Фильтрация данных

```
try
{
    var resources = new List<Report>
    {
        new Report { Id = "allowed-id1" },
        new Report { Id = "denied-id1"  }
    };

    var authContext = await _enforcementPoint.AuthorizeAsync(
        authorizationRequets: AuthorizationRequets.Create(
            resourceType: "report",
            action: "publish"),
        cancellationToken: cancellationToken);

    // Доступ разрешен...
    var filtered = container.GetInstance<ICollectionFilter>()
        .Filter(
            filteringCollection: resources,
            resourceType: "report",
            idSelector => x => x.Id,
            authContext: authContext);
    // filtered содержит только allowed-id1
}
catch (AccessDeniedException ex)
{
    // Доступ запрещен
}
```