CUSTIS®

Авторизация на основе атрибутов: как мы перестали раздавать роли и занялись политиками

Антон Лапицкий

Архитектор приложений

Joker

20 октября 2018 года

Антон Лапицкий

CUSTIS®

- Архитектор приложений в CUSTIS.
  Более 8 лет занимаюсь промышленной разработкой на Java в банковской и образовательной сферах, а также для госсектора

{О чем поговорим}

CUSTIS®

- Ролевая vs атрибутивная модель на реальном примере
- Стандарт XACML
- Фреймворк EasyABAC
  - Утилиты
  - Устройство
  - API
  - Производительность

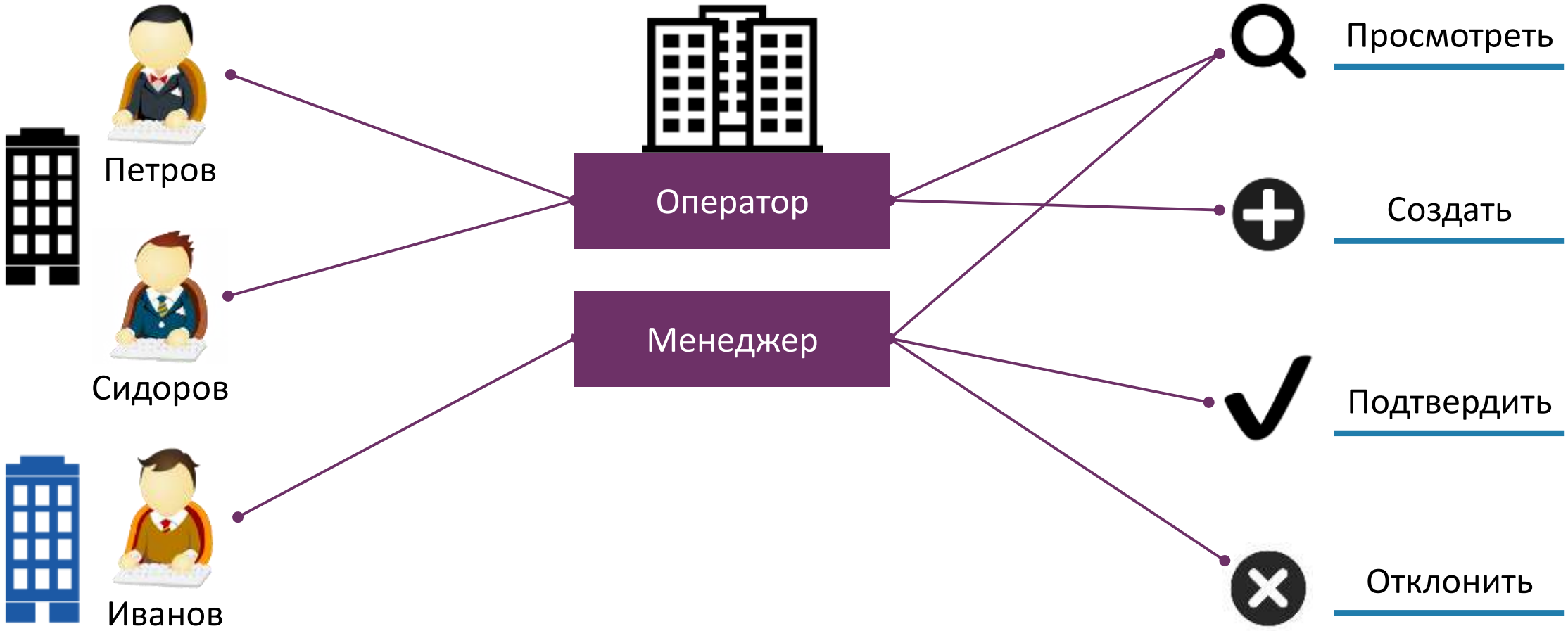
{Задача}

CUSTIS®

- Магазин продает бытовую технику оптом
- У магазина несколько филиалов, планируется расширение
- Оператор принимает заказы
- Менеджер подтверждает или отклоняет заказы

{Модель доступа v. 1 (RBAC)}

CUSTIS®



{Модель доступа v. 1 (RBAC)}

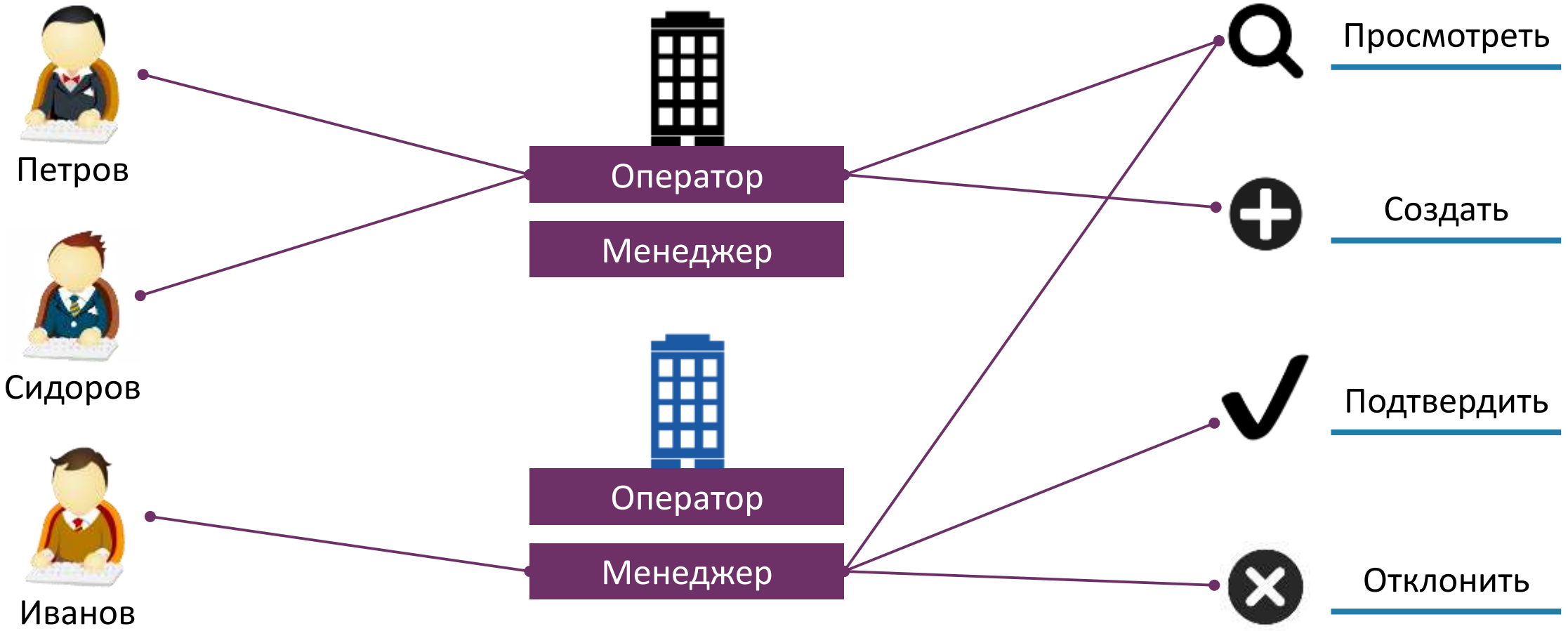
CUSTIS®

```
void checkView() {
        User user = AuthenticationContext.currentUser();

        if (user.hasRole(ROLE_MANAGER.name())
                || user.hasRole(ROLE_OPERATOR.name())) {
            return;
        }

        throw new NotPermittedException("not permitted");
}
```

{Модель доступа v. 2 (RBAC)}



{Модель доступа v. 2 (RBAC)}

CUSTIS®

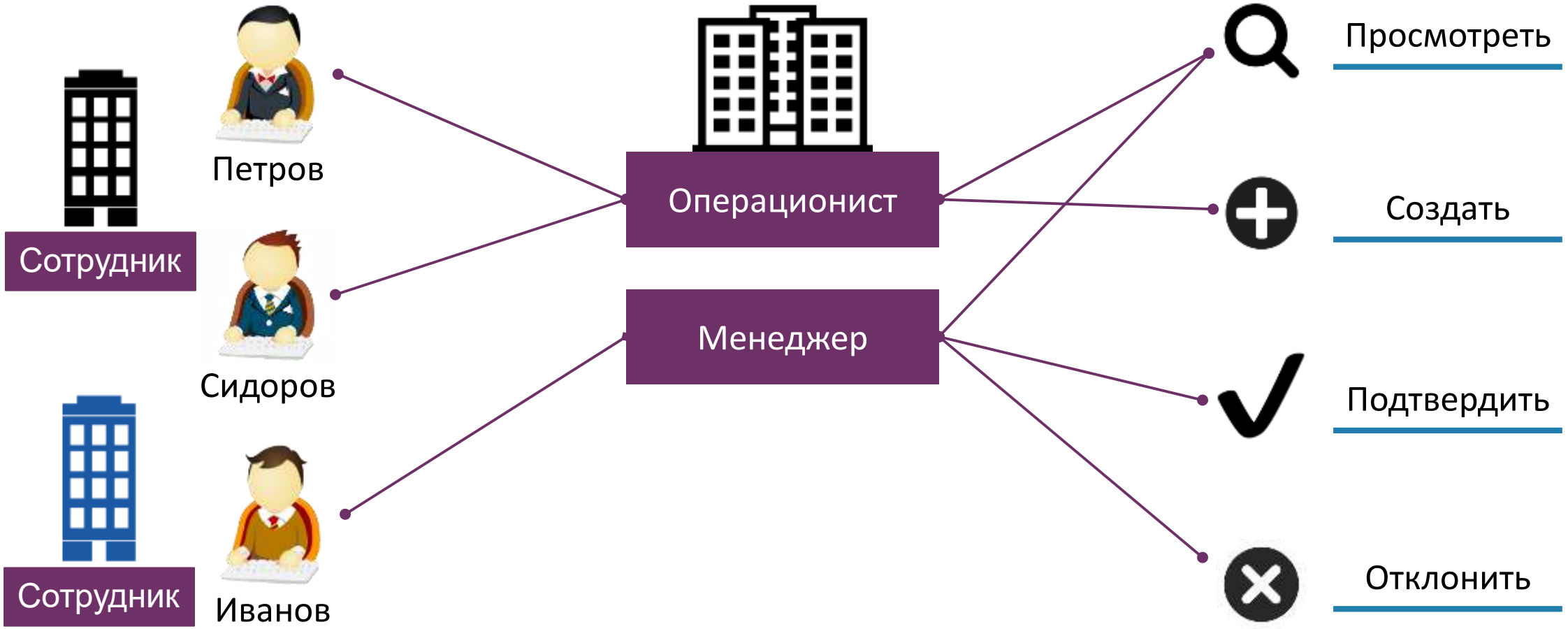
```
void checkView(Order order) {
        User user = AuthenticationContext.currentUser();

        if (user.hasRole(ROLE_MANAGER.ofBranch(order.getBranchId()))
                || user.hasRole(ROLE_OPERATOR.ofBranch(order.getBranchId()))) {
            return;
        }

        throw new NotPermittedException("not permitted");
}
```

{Модель доступа v. 3 (RBAC)}

CUSTIS®



{Модель доступа v. 3 (RBAC)}

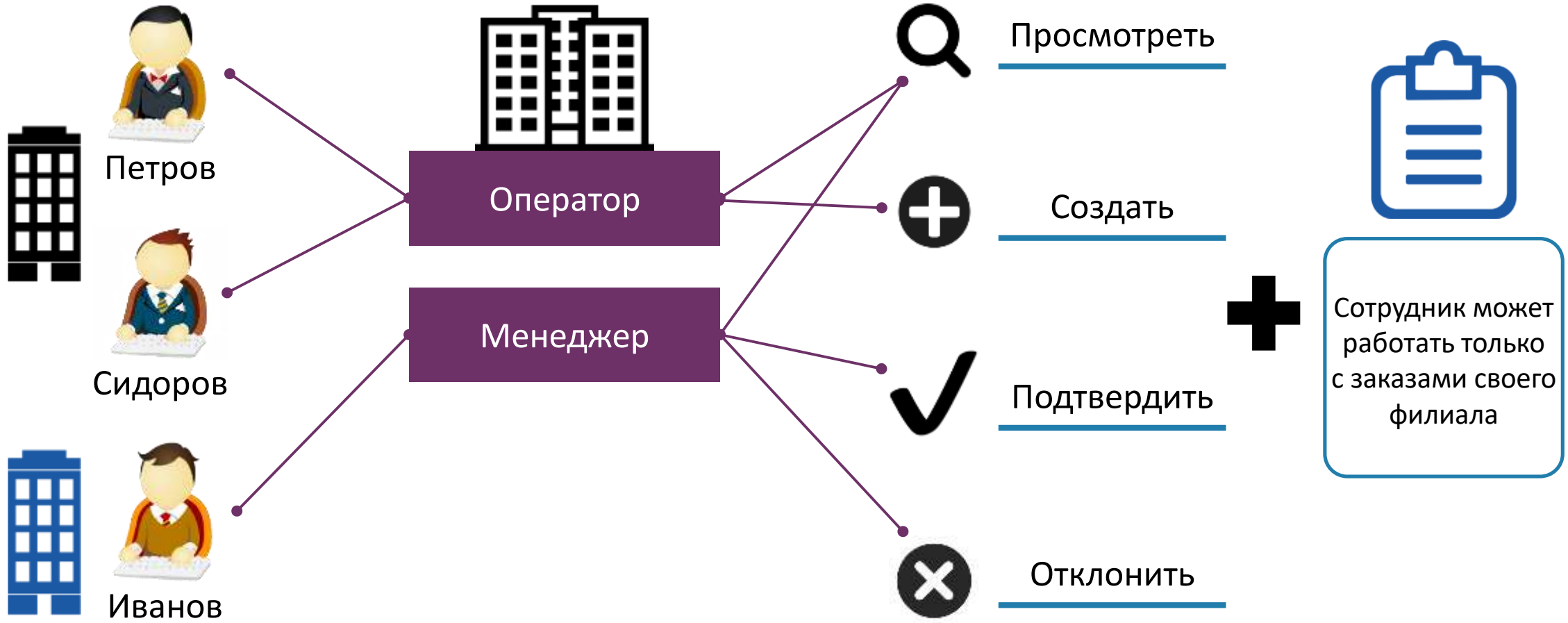
CUSTIS®

```
void checkView(Order order) {
        User user = AuthenticationContext.currentUser();
        checkBranchRole(user, order.getBranch());
        if (user.hasRole(ROLE_MANAGER.name())
                        || user.hasRole(ROLE_OPERATOR.name())) {
            return;
        }
        throw new NotPermittedException("not permitted");
}

void checkBranchRole(User user, Branch branch) {
        if (user.hasRole(ROLE_USER.ofBranch(branch.getId()))) return;
        throw new NotPermittedException("not permitted");
}
```

{Модель доступа v. 4 (RBAC)}

CUSTIS®



{Модель доступа v. 4 (RBAC)}

CUSTIS®

```
void checkView(Order order) {
        User user = AuthenticationContext.currentUser();

        checkUserBranch(user, order.getBranch());
        if (user.hasRole(ROLE_MANAGER.name())
                                || user.hasRole(ROLE_OPERATOR.name())) {
            return;
        }
        throw new NotPermittedException("not permitted");
}

void checkUserBranch(User user, Branch branch) {
        if (user.getBranch().getId().equals(branch.getId())) return;
        throw new NotPermittedException("not permitted");
}
```

{Сравнение моделей доступа}

CUSTIS®

		
2 Роли	Простота	Нет разделения между филиалами
2 Роли для каждого филиала	Простота	Масштабируемость Искусственные роли
2 Бизнес-роли + Роль Сотрудник филиала	Корректные бизнес-роли	Масштабируемость Искусственные роли
2 Бизнес-роли + Бизнес-логика в коде	Масштабируемость	Hardcode бизнес-логики

{Как должно работать}

CUSTIS®

КОНТЕКСТ

- Субъект
- Действие
- Объект
- Окружение

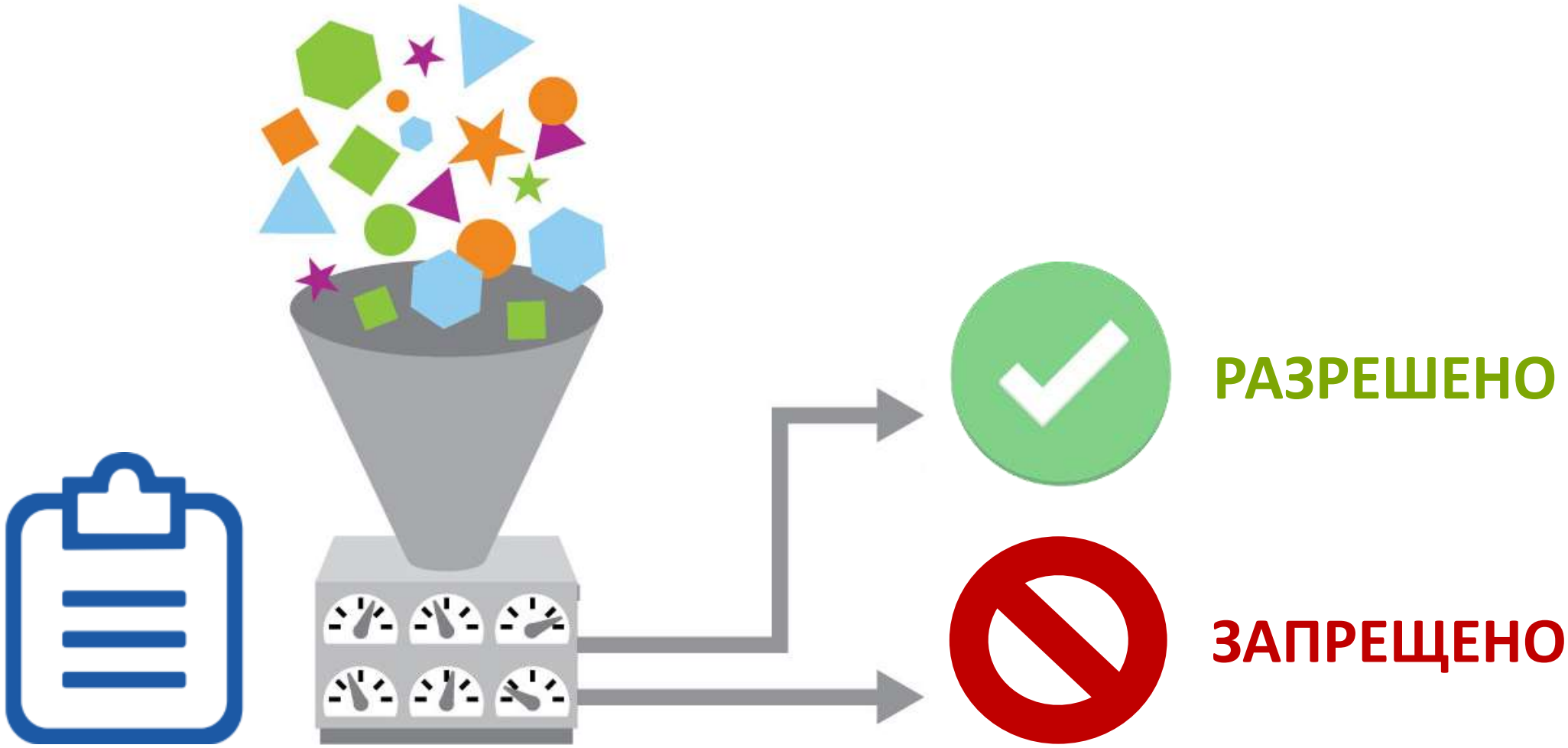
{Как должно работать}

CUSTIS®

ПРАВИЛА

{Как должно работать}

CUSTIS®



CUSTIS®

XACML

“eXtensible Access Control Markup Language”

CUSTIS®

XACML

“eXtensible Access Control Markup Language”

Разрешить просмотр заказа с 09:00 до 17:00

XACML

“eXtensible Access Control Markup Language”

```
<Policy PolicyId="SamplePolicy"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
   <Rule RuleId="LoginRule" Effect="Permit">
      <Target>
         <Actions>
            <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
               <ActionAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                          AttributeId=“OrderAction"/>
            </ActionMatch>
         </Actions>
      </Target>
      <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-greater-than-or-equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
               <EnvironmentAttributeSelector DataType="http://www.w3.org/2001/XMLSchema#time"
                                             AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"/>
            </Apply>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">09:00:00</AttributeValue>
         </Apply>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
               <EnvironmentAttributeSelector DataType="http://www.w3.org/2001/XMLSchema#time"
                                             AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"/>
            </Apply>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">17:00:00</AttributeValue>
         </Apply>
      </Condition>
   </Rule>
   <Rule RuleId="FinalRule" Effect="Deny"/>
</Policy>
```

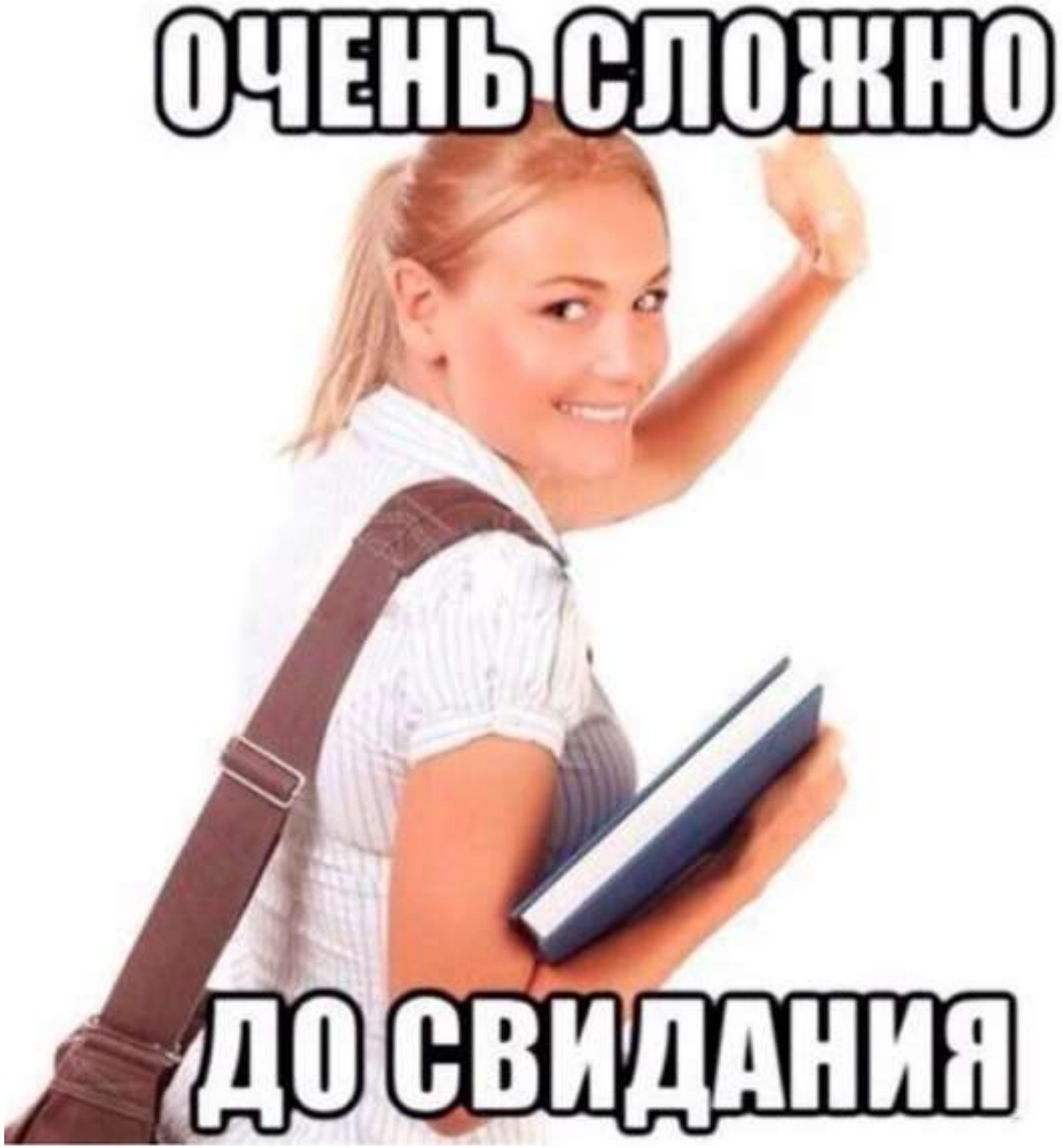
XACML

“eXtensible Access Control Markup Language”

```
<Policy PolicyId="SamplePolicy"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
   <Rule RuleId="LoginRule" Effect="Permit">
      <Target>
         <Actions>
            <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
               <ActionAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                          AttributeId=“OrderAction"/>
            </ActionMatch>
         </Actions>
      </Target>
      <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-greater-than-or-equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
               <EnvironmentAttributeSelector DataType="http://www.w3.org/2001/XMLSchema#time"
                                             AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"/>
            </Apply>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">09:00:00</AttributeValue>
         </Apply>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
               <EnvironmentAttributeSelector DataType="http://www.w3.org/2001/XMLSchema#time"
                                             AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"/>
            </Apply>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">17:00:00</AttributeValue>
         </Apply>
      </Condition>
   </Rule>
   <Rule RuleId="FinalRule" Effect="Deny"/>
</Policy>
```

XACML

“eXtensible Access Control Markup Language”



{Реализации XACML}

CUSTIS®

WSO2 Identity Server

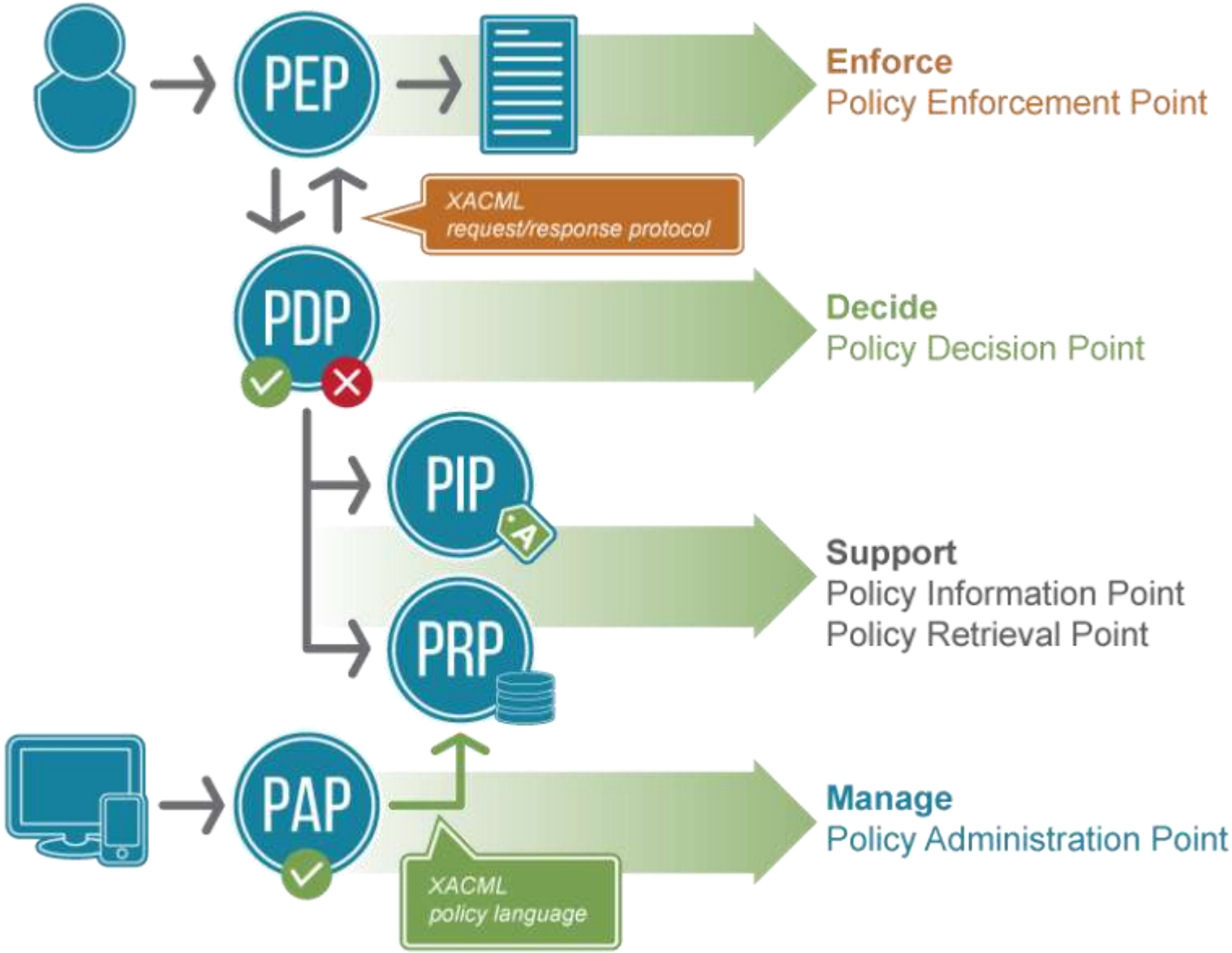
AUTHZ FORCE

WSO2

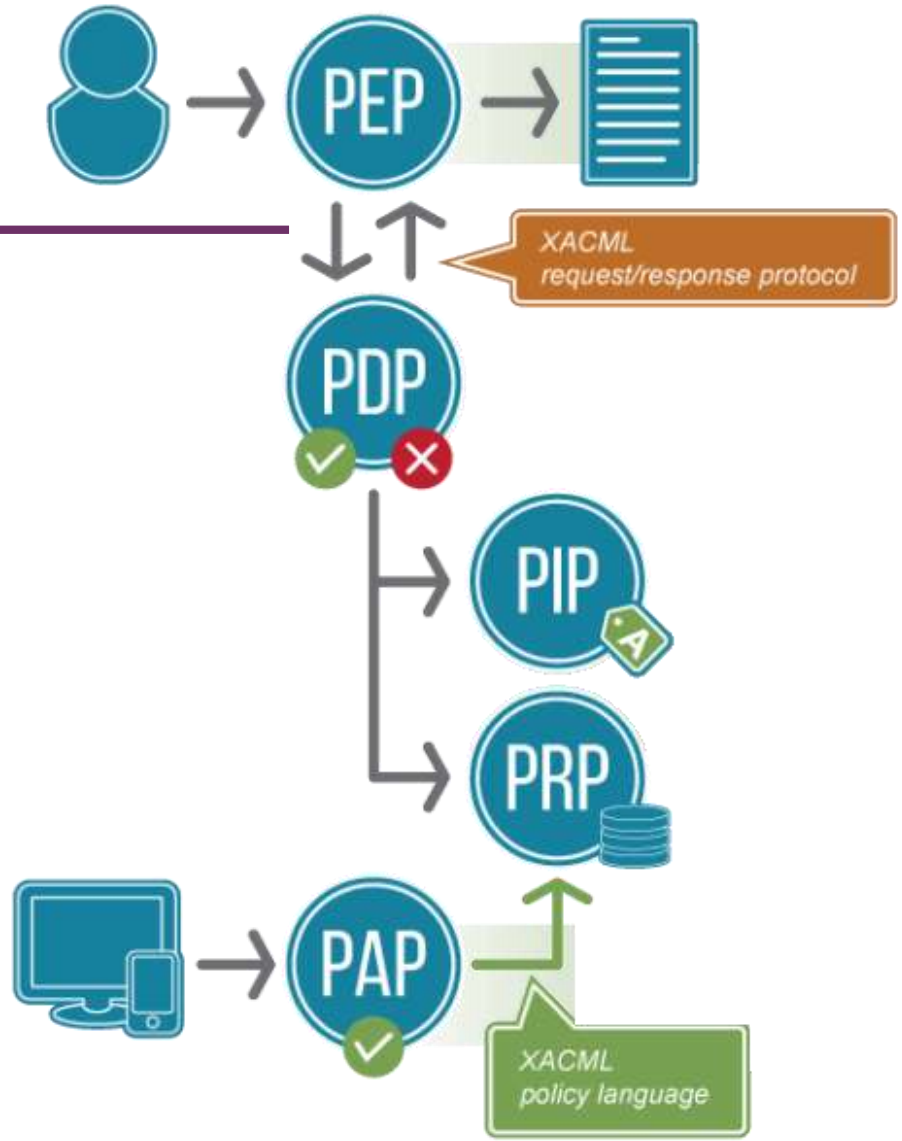
BALANA

{Архитектура XACML}

CUSTIS®



{Архитектура XACML}

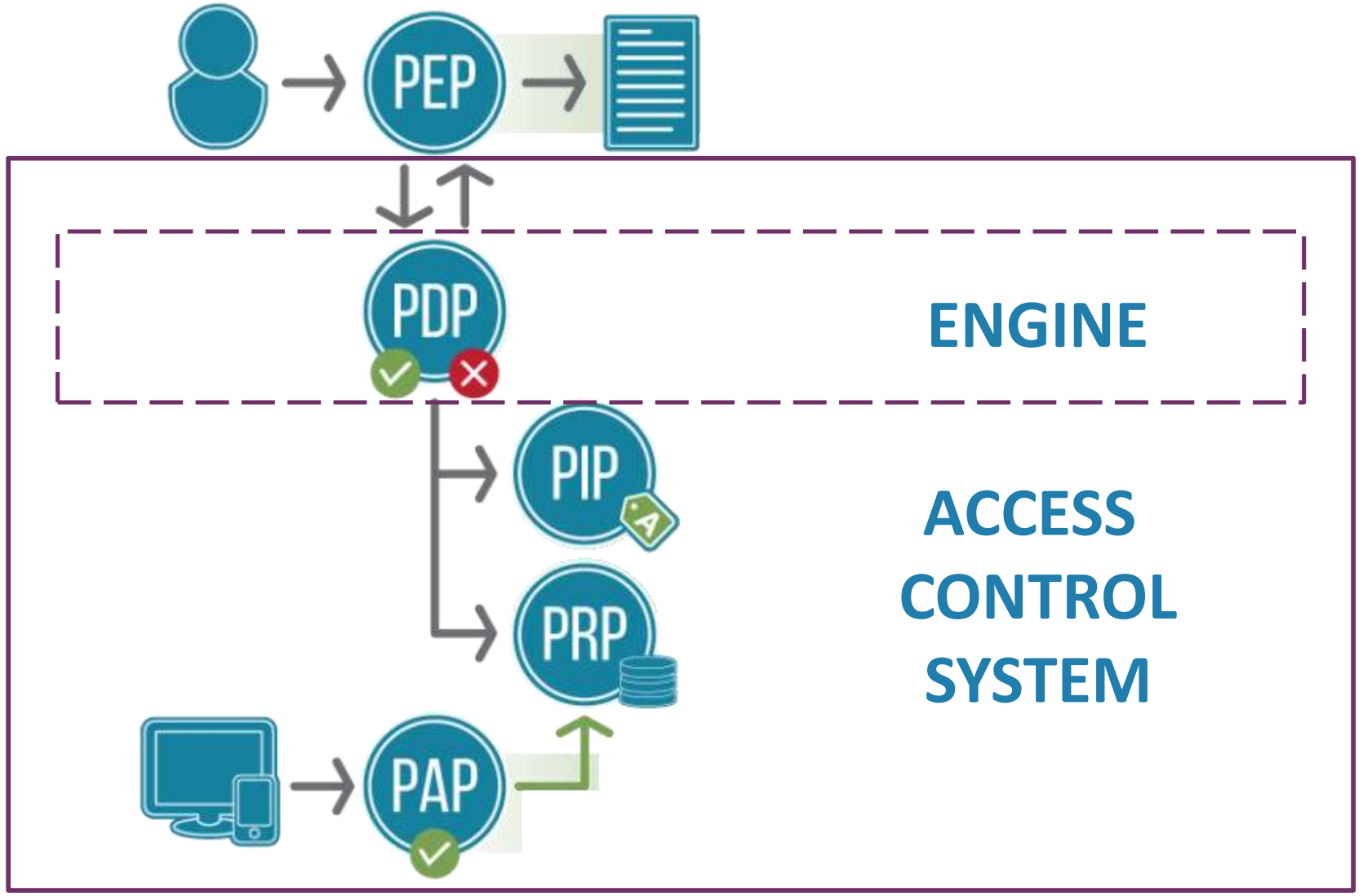


APPLICATION

ACCESS CONTROL SYSTEM

{Архитектура XACML}

CUSTIS®



{Проблемы существующих решений}

CUSTIS®

- Плохие инструменты тестирования политики доступа
- Непонятно, почему политика дала Permit или Deny
- Скорость обработки сильно зависит от структуры политик
- Нет инструментов оптимизации запросов

CUSTIS®

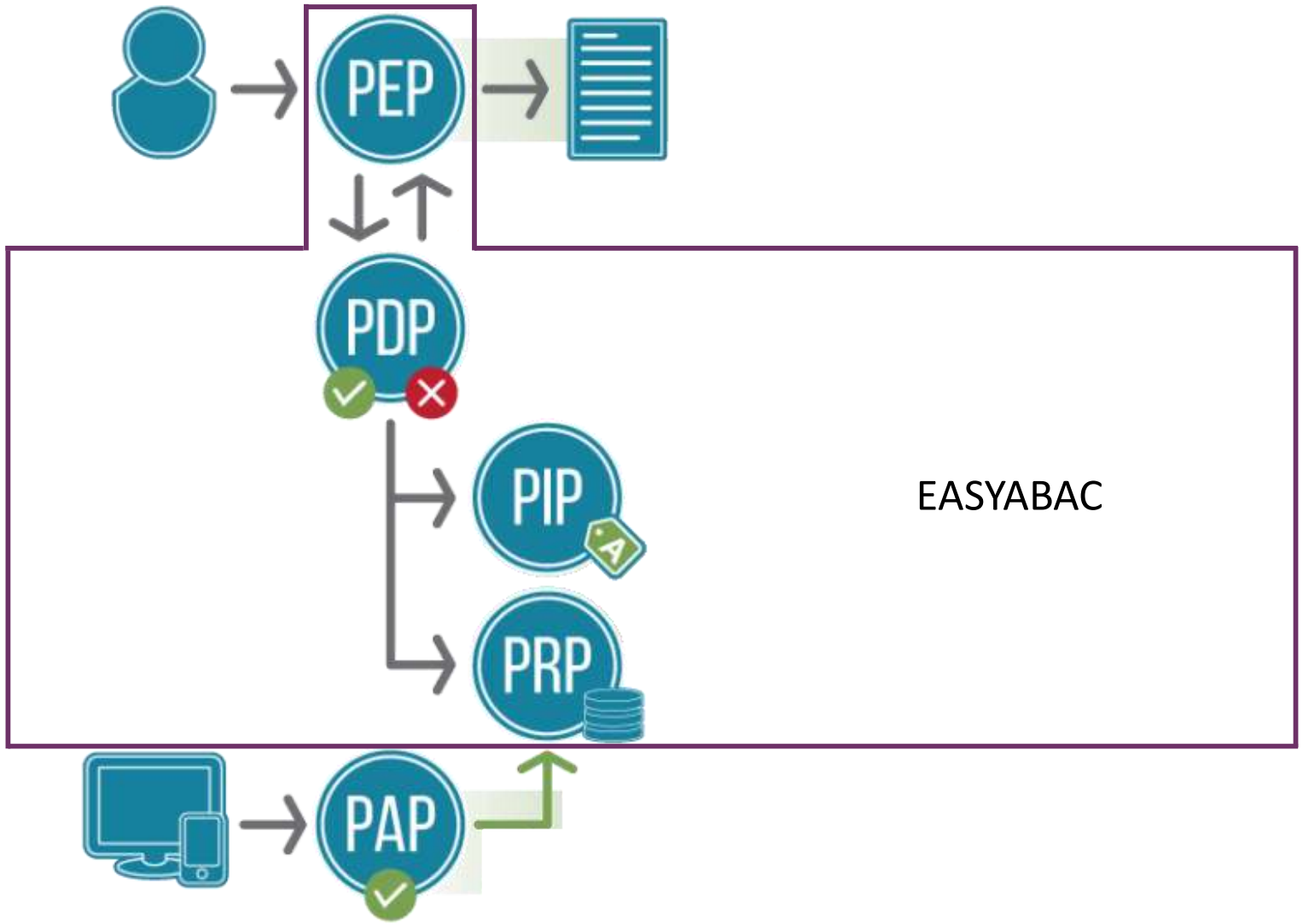
{Easyabac}

{Easyabac:goals}

CUSTIS®

- Упростить создание политик
- Упростить тестирование политик
- Сделать удобный Java API
- Оптимизировать запросы

{Easyabac и XACML}



{Easyabac:model}

CUSTIS®

```
subject:
  attributes:
  – id: id
  – id: role
    title: Роль сотрудника
    allowableValues:
      – OPERATOR
      – MANAGER
  – id: branchId
    title: ИД филиала
  – id: maxOrderAmount
    title: Максимальный заказ
    type: int
```

```
customer:
  title: Клиент
  attributes:
  – id: id
  – id: branchId
    title: ИД филиала
```

```
order:
  title: Заказ
  actions:
  – view
  – create
  – approve
  – reject
  attributes:
  – id: id
    title: ИД заказа
  – id: amount
    title: Сумма заказа
    type: int
  – id: branchId
    title: ИД филиала
  – id: customerId
    title: ИД клиента
```

{Easyabac:model}

CUSTIS®

```
subject:
  attributes:
  – id: id
  – id: role
    title: Роль сотрудника
    allowableValues:
      – OPERATOR
      – MANAGER
  – id: branchId
    title: ИД филиала
  – id: maxOrderAmount
    title: Максимальный заказ
    type: int
```

```
customer:
  title: Клиент
  attributes:
  – id: id
  – id: branchId
    title: ИД филиала
```

```
order:
  title: Заказ
  actions:
  – view
  – create
  – approve
  – reject
  attributes:
  – id: id
    title: ИД заказа
  – id: amount
    title: Сумма заказа
    type: int
  – id: branchId
    title: ИД филиала
  – id: customerId
    title: ИД клиента
```

{Easyabac:model}

CUSTIS®

```
subject:
  attributes:
  – id: id
  – id: role
    title: Роль сотрудника
    allowableValues:
      – OPERATOR
      – MANAGER
  – id: branchId
    title: ИД филиала
  – id: maxOrderAmount
    title: Максимальный заказ
    type: int
```

```
customer:
  title: Клиент
  attributes:
  – id: id
  – id: branchId
    title: ИД филиала
```

```
order:
  title: Заказ
  actions:
  – view
  – create
  – approve
  – reject
  attributes:
  – id: id
    title: ИД заказа
  – id: amount
    title: Сумма заказа
    type: int
  – id: branchId
    title: ИД филиала
  – id: customerId
    title: ИД клиента
```

{Easyabac:policy}

CUSTIS®

```
order:
  title: Заказ
  actions:
  – view
  – create
  – approve
  – reject
  attributes:
  – id: id
    title: ИД заказа
  – id: amount
    title: Сумма заказа
    type: int
  – id: branchId
    title: ИД филиала
  – id: customerId
    title: ИД клиента
```

```
– title: Ограничение на изменение заказа
  accessToActions: [order.create, order.approve, order.reject]
  rules:
   – title: Только в рабочее время для своего филиала
     operation: AND
     conditions:
      – env.time >= 09:00
      – env.time <= 17:00
      – order.branchId == subject.branchId
```

```
– title: Менеджер
  accessToActions: [order.approve, order.reject]
  rules:
   – title: Доступ менеджера
     operation: AND
     conditions:
     – subject.role in 'MANAGER'
     – subject.maxOrderAmount > order.amount
```

{Easyabac:policy}

CUSTIS®

```
order:
  title: Заказ
  actions:
  – view
  – create
  – approve
  – reject
  attributes:
  – id: id
    title: ИД заказа
  – id: amount
    title: Сумма заказа
    type: int
  – id: branchId
    title: ИД филиала
  – id: customerId
    title: ИД клиента
```

```
– title: Ограничение на изменение заказа
  accessToActions: [order.create, order.approve, order.reject]
  rules:
   – title: Только в рабочее время для своего филиала
     operation: AND
     conditions:
      – env.time >= 09:00
      – env.time <= 17:00
      – order.branchId == subject.branchId
```

```
– title: Менеджер
  accessToActions: [order.approve, order.reject]
  rules:
   – title: Доступ менеджера
     operation: AND
     conditions:
     – subject.role in 'MANAGER'
     – subject.maxOrderAmount > order.amount
```

{Easyabac:policy}

CUSTIS®

```
order:
  title: Заказ
  actions:
  – view
  – create
  – approve
  – reject
  attributes:
  – id: id
    title: ИД заказа
  – id: amount
    title: Сумма заказа
    type: int
  – id: branchId
    title: ИД филиала
  – id: customerId
    title: ИД клиента
```

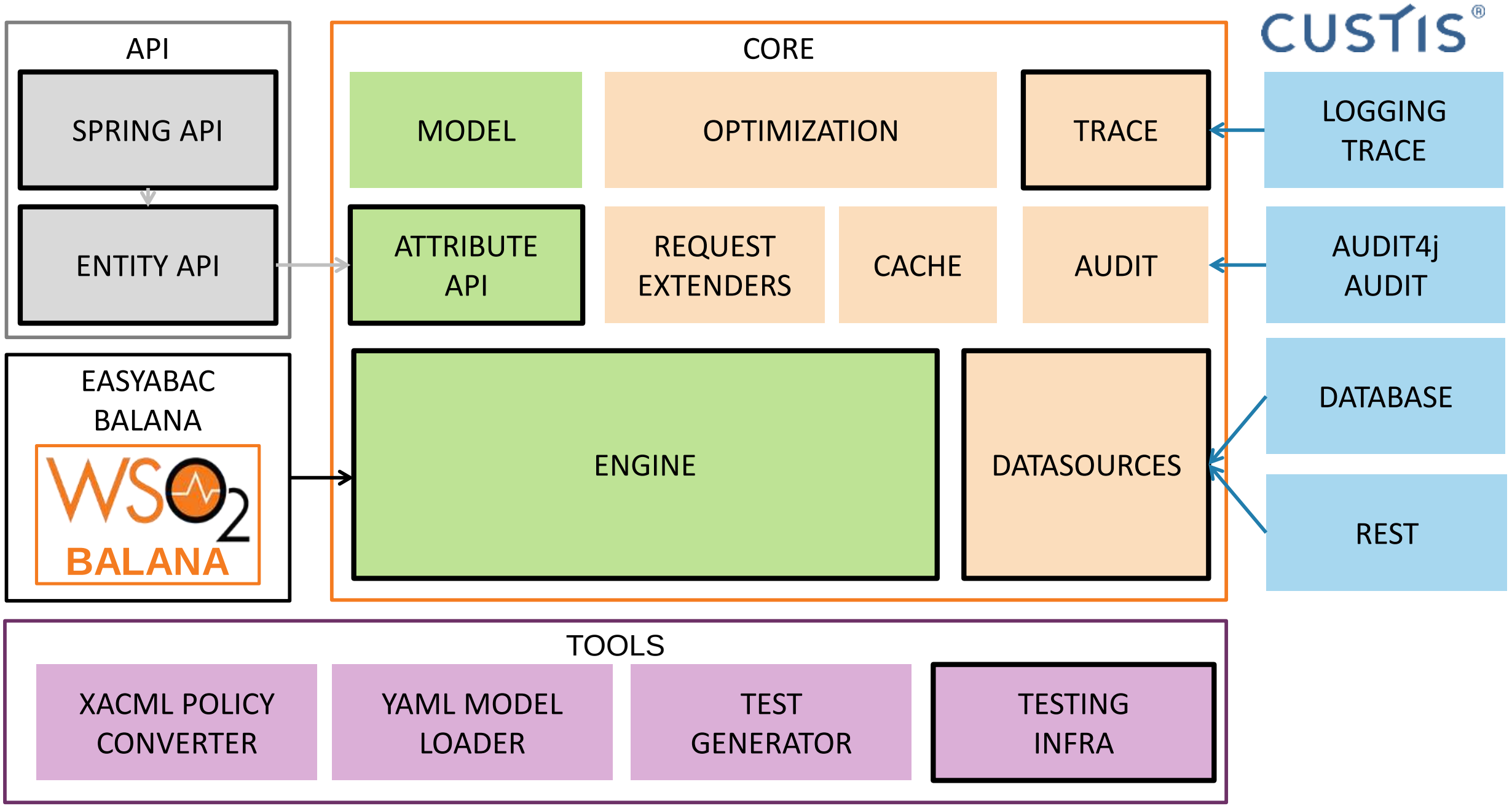
```
– title: Ограничение на изменение заказа
  accessToActions: [order.create, order.approve, order.reject]
  rules:
   – title: Только в рабочее время для своего филиала
     operation: AND
     conditions:
      – env.time >= 09:00
      – env.time <= 17:00
      – order.branchId == subject.branchId
```

```
– title: Менеджер
  accessToActions: [order.approve, order.reject]
  rules:
   – title: Доступ менеджера
     operation: AND
     conditions:
     – subject.role in 'MANAGER'
     – subject.maxOrderAmount > order.amount
```

{Easyabac:policy:restrictions}

CUSTIS®

- Нет групп политик
- Простые условия
- Только базовые функции (==, >, <, in и т. д.)
- Только основные типы данных (нет поддержки rfc822, uri и т. д.). Строки – основной тип данных


API
SPRING API
ENTITY API
EASYABAC
BALANA
WSO2
BALANA
CORE
MODEL
OPTIMIZATION
TRACE
ATTRIBUTE
API
REQUEST
EXTENDERS
CACHE
AUDIT
ENGINE
DATASOURCES
CUSTIS
LOGGING
TRACE
AUDIT4j
AUDIT
DATABASE
REST
TOOLS
XACML POLICY
CONVERTER
YAML MODEL
LOADER
TEST
GENERATOR
TESTING
INFRA

CUSTIS®

{Easyabac:tools}

{Easyabac:tools:test}

CUSTIS®

```
public class OrderAuthTest extends EasyAbacBaseTestClass {

    public OrderAuthTest () {
        super(model); // модель для тестирования
    }

    @Parameters(name = "{index}: resource({0}) and action({1}). Expected = ({2})")
    public static List<Object[]> data() throws Exception {
        // тестовые данные
    }
}
```

{Easyabac:tools:test}

CUSTIS®

```
@RunWith(Parameterized.class)
public abstract class EasyAbacBaseTestClass {

    @Parameterized.Parameter
    public Object resource;

    @Parameterized.Parameter(value = 1)
    public Object action;

    @Parameterized.Parameter(value = 2)
    public boolean expectedPermit;

    @Parameterized.Parameter(value = 3)
    public TestDescription testDescription;
....
}
```

{Easyabac:tools:test}

CUSTIS®

order_0.yaml

```
expectedResult: PERMIT
action:
   id: order.action
   value: order.approve
attributes:
  order:
    amount: 500
    branchId: branchId_0
  subject:
    role: MANAGER
    branchId: branchId_0
    maxOrderAmount: 600
```

CUSTIS®

{Easyabac:core}

{Easyabac:core:datasources}

CUSTIS®

```
public abstract class Datasource {
    private final Set<Param> params;
    private final String returnAttributeId;
    private final long expire;

    private Attribute returnAttribute;


    abstract public List<String> find()
                        throws EasyAbacDatasourceException;

}

public class Param {
    private final String name;
    private final String attributeParamId;
    private String value;
    private Attribute attributeParam;
}
```

{Easyabac:core:datasources}

CUSTIS®

```
public abstract class Datasource {
    private final Set<Param> params;
    private final String returnAttributeId;
    private final long expire;

    private Attribute returnAttribute;


    abstract public List<String> find()
                    throws EasyAbacDatasourceException;

}
```

Уже есть реализации для

- БД – DatabaseDatasource
- REST – RESTDatasource

{Easyabac:core:trace}

CUSTIS®

```
– title: Ограничение на изменение заказа
  accessToActions: [order.create, order.approve, order.reject]
  rules:
   – title: Только в рабочее время для своего филиала
     operation: AND
     conditions:
      – env.time >= 09:00
      – env.time <= 17:00
      – order.branchId == subject.branchId
```

```
– title: Менеджер
  accessToActions: [order.approve, order.reject]
  rules:
   – title: Доступ менеджера
     operation: AND
     conditions:
     – subject.role in 'MANAGER'
     – subject.maxOrderAmount > order.amount
```

{Easyabac:core:trace}

CUSTIS®

– **title:** Ограничение на изменение заказа
  accessToActions: [order.create, order.approve, order.reject] MATCH=TRUE, ACTION=order.approve
  rules:
   – **title:** Только в рабочее время для своего филиала
     operation: AND RESULT=TRUE
     conditions:
      – env.time >= 09:00 RESULT=TRUE, env.time = 15:43
      – env.time <= 17:00 RESULT=TRUE, env.time = 15:43
      – order.branchId == subject.branchId RESULT=TRUE,
          order.branchId = Moscow
          subject.branchId = Moscow

– **title:** Менеджер
  accessToActions: [order.approve, order.reject] MATCH=TRUE, ACTION=order.approve
  rules:
   – **title:** Доступ менеджера
     operation: AND RESULT=FALSE
     conditions:
     – subject.role in 'MANAGER‘ RESULT=FALSE, subject.role = OPERATOR
     – subject.maxOrderAmount > order.amount RESULT=N/A

CUSTIS®

{Easyabac:api}

{Example}

CUSTIS®

```
public interface AuthService {

    AuthResponse authorize(List<AuthAttribute> attributes);

    Map<RequestId, AuthResponse>
                        authorizeMultiple(Map<RequestId, List<AuthAttribute>> attributes);
}
```

```
void checkOrder (Order order, List<OrderAction> operations) throws NotPermittedException {
        // TODO реализовать через EasyAbac
}
```

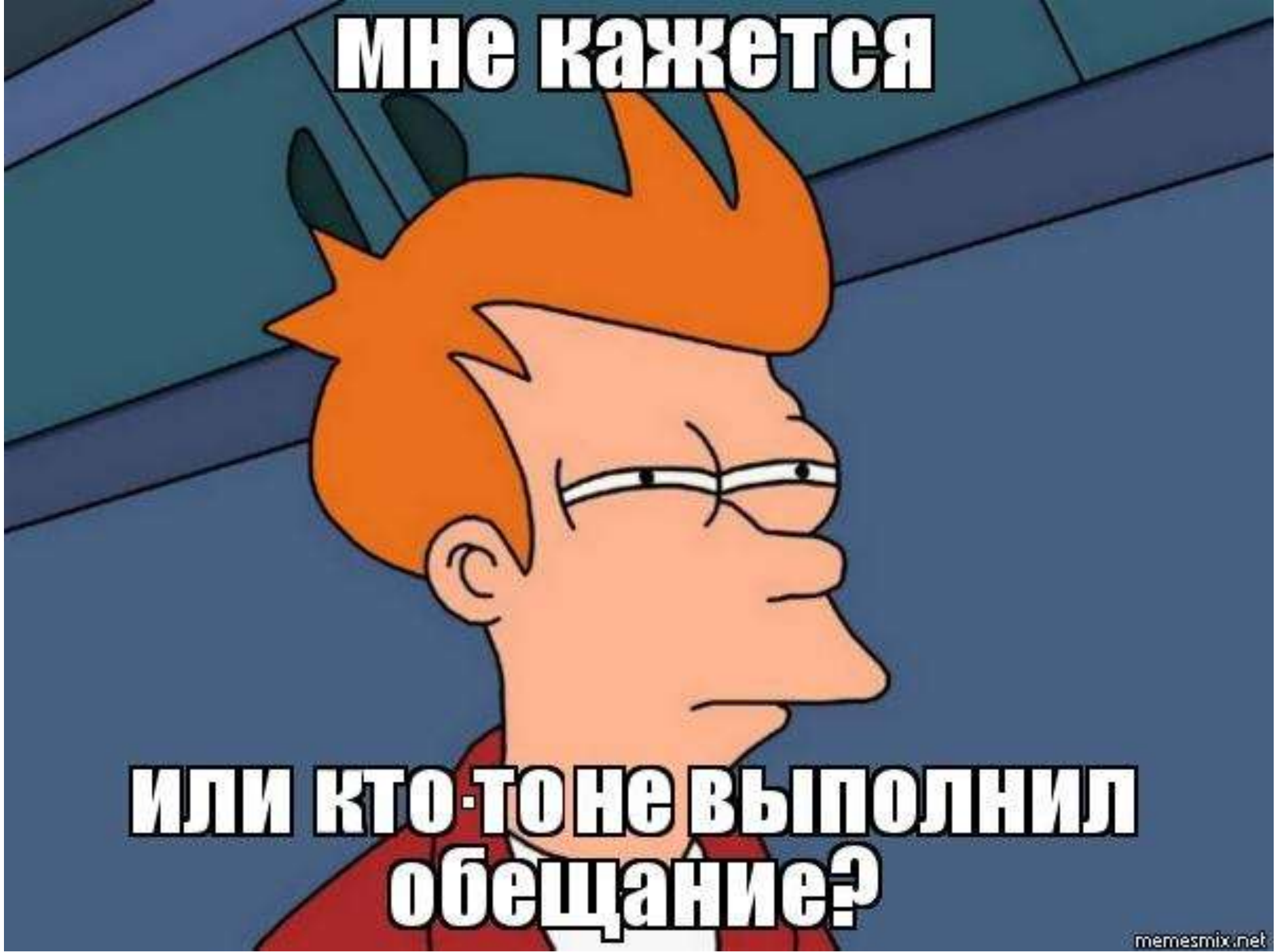
{Example}

CUSTIS®

```
public void checkOrder (Order order, List<OrderAction> operations)
                                                        throws NotPermittedException {
    Map<RequestId, List<AuthAttribute>> attributes = operations.stream()
            .map(operation -> extract(order, operation))
            .collect(Collectors.toMap(o -> RequestId.newRandom(), o -> o));

    Map<RequestId, AuthResponse> results = authService.authorizeMultiple(attributes);

    for (AuthResponse result : results.values()) {
        if (result.getDecision() != AuthResponse.Decision.PERMIT) {
            throw new NotPermittedException("Not permitted");
        }
    }
}
```


мне кажется
или кто-то не выполнил обещание?
memesmix.net

{Example}

CUSTIS®

```
public void checkOrder (Order order, List<OrderAction> operations)
                                                    throws NotPermittedException {
    Map<RequestId, List<AuthAttribute>> attributes = operations.stream()
            .map(operation -> extract(order, operation))
            .collect(Collectors.toMap(o -> RequestId.newRandom(), o -> o));

    Map<RequestId, AuthResponse> results = authorizationService.authorizeMultiple(attributes);

    for (AuthResponse result : results.values()) {
        if (result.getDecision() != AuthResponse.Decision.PERMIT) {
            throw new NotPermittedException("Not permitted");
        }
    }
}
```

{Easyabac:api:entity}

CUSTIS®

```
public interface EntityPermissionChecker<T, A> {

   void ensurePermitted(T entity, A operation) throws NotPermittedException;
   void ensurePermittedAny(T entity, List<A> operations) throws NotPermittedException;
   void ensurePermittedAll(T entity, List<A> operations) throws NotPermittedException ;
   void ensurePermittedAll(Map<T, A> operationsMap) throws NotPermittedException ;

   List<A> getPermittedActions(T entity, List<A> operations);
   Map<T, List<A>> getPermittedActions(Map<T, List<A>> operationsMap);
   Map<T, List<A>> getPermittedActions(List<T> entities, List<A> operations);
}
```

{Flashback:rbac_v4}

CUSTIS®

```
void checkView(Order order) {
    User user = AuthenticationContext.currentUser();
    checkUserBranch(user, order.getBranch());
    if (user.hasRole(ROLE_MANAGER.name()) ||
                        user.hasRole(ROLE_OPERATOR.name())) {
        return;
    }
    throw new NotPermittedException("not permitted");
}

void checkUserBranch(User user, Branch branch) {
    if (user.getBranch().getId().equals(branch.getId())) return;
    throw new NotPermittedException("not permitted");
}
```

{Easyabac:api:entity}

CUSTIS®

```
EasyAbac easyAbac = EasyAbacBuilderHelper.defaultBuilder(…).build();

EntityPermissionChecker<Order, OrderAction> permissionChecker =
EntityPermissionCheckerHelper.newPermissionChecker(easyAbac);


    public void checkView(Order order) {
        permissionChecker.ensurePermitted(order, OrderAction.VIEW);
    }
    …
}
```

{Easyabac:api:entity:attr}

CUSTIS®

```
class Order implements AttributeAuthEntity {

    @Override
    public List<AuthAttribute> getAuthAttributes()  {
            // список атрибутов сущности
    }
}
```

```
enum OrderAction implements AttributiveAuthAction {

    @Override
    public AuthAttribute getAuthAttribute() {

            // идентификатор действия

    }
}
```

{Easyabac:api:entity:attr}

CUSTIS®

```
@AuthorizationEntity(name = "order")
public class Order {

    @AuthorizationAttribute
    private String id;

    @AuthorizationAttribute(id = "amount")
    private int orderAmount;

    @AuthorizationAttribute
    private String branchId;

    private String customerId;
    …
}
```

```
@AuthorizationAction(entity = "order")
public enum Action {

    VIEW("view"), CREATE("create"),
    APPROVE("approve"), REJECT("reject");

    @AuthorizationActionId
    private String id;
….
}
```

CUSTIS®

{Easyabac:api}

+

{Easyabac:api:spring}

CUSTIS®

```
@Indexed
public interface PermissionChecker<T, A> {

}


@Configuration
@EnablePermissionCheckers("custis.easyabac.demo.permissionchecker")
public class EasyABACConfiguration {

}
```

{И что можно...}

CUSTIS®

- Проверять любые результаты (permit, deny, …)

```
void ensureIndeterminate(Order order, OrderAction action)
                                                    throws NotExpectedResultException;

void ensureDeniedAll(Order order, List<OrderAction> action)
                                                    throws NotExpectedResultException;
```

{И что можно...}

CUSTIS®

- Сделать методы под конкретное действие

```
void ensureDeniedRead(Order order) throws NotExpectedResultException;

void ensurePermittedReadOrApprove(Order order)
                                        throws NotExpectedResultException;
```

{И что можно...}

CUSTIS®

- Возвращать **boolean**, а не исключение

```
boolean isIndeterminate(Order order, OrderAction action);

boolean isDeniedAll(Order order, List<OrderAction> action);

boolean isDeniedRead(Order order);

boolean isPermittedReadOrApprove(Order order);
```

{И что можно...}

CUSTIS®

- Менять входные параметры

```
boolean isDeniedAll(List<Order> order, OrderAction action);

boolean isDeniedAll(OrderAction action, List<Order> order);

boolean isDeniedAll(Map<OrderAction Order> order);

boolean isDeniedAll(Map<OrderAction, List<Order>> order);

....
```

{И что можно...}

CUSTIS®

- Группировать результаты через методы get*

```
List<OrderAction> getDeniedActions(Order order, List<OrderAction> actions);

List<Order> getPermittedResources(List<Order> orders, OrderAction action);

Map<Order, List<OrderAction>> getDeniedActions(
                                        List<Order> orders, OrderAction action);

Map<OrderAction, List<Order>> getDeniedResources(
                                      List<OrderAction> actions, List<Order> orders);
```

CUSTIS®

{Easyabac:demo}

CUSTIS®

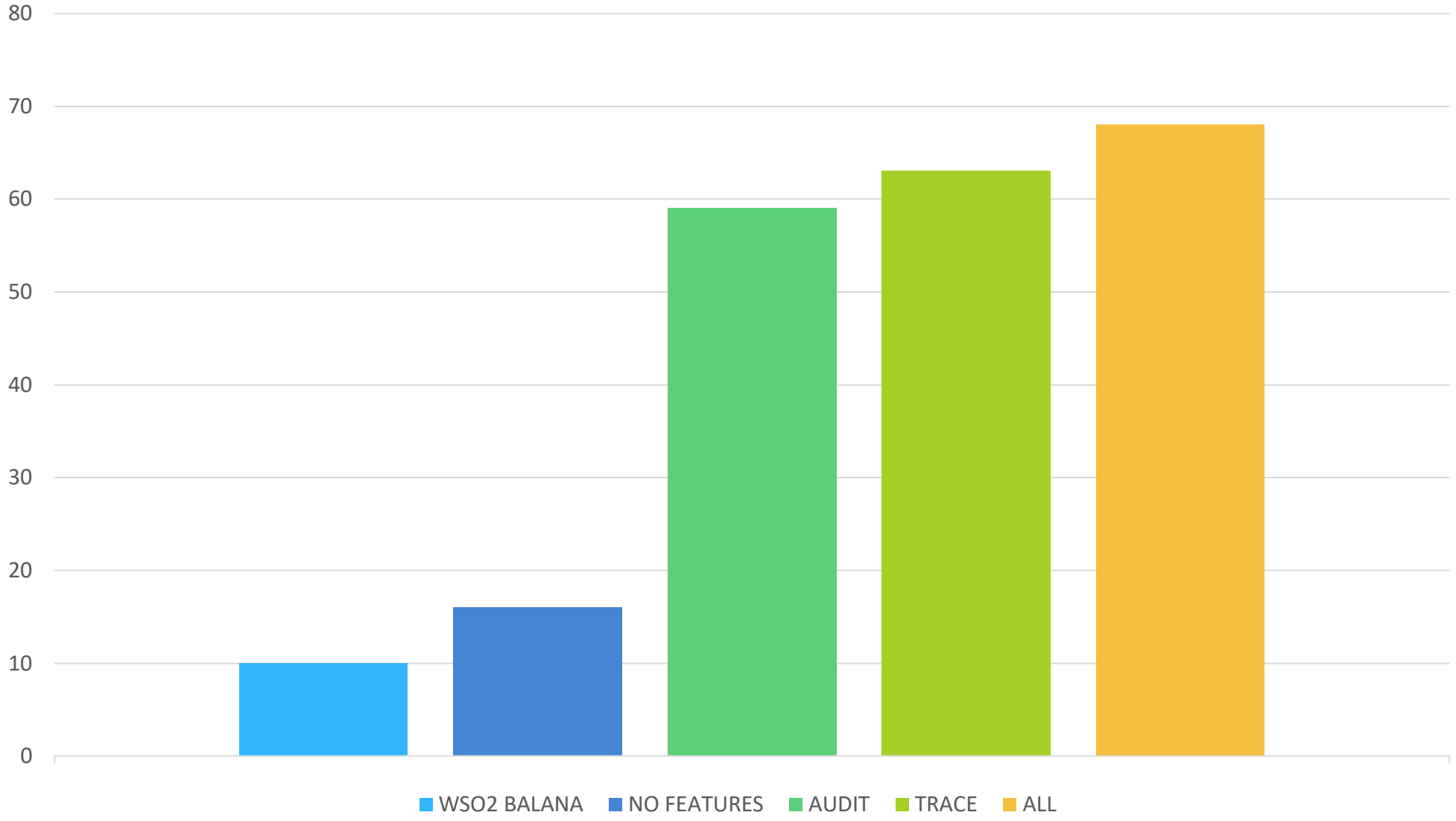
{Easyabac:perf}

{Easyabac core features}

- WSO2 BALANA
- NO FEATURES
- AUDIT
- TRACE
- ALL FEATURES

Milliseconds per operation

CUSTIS®

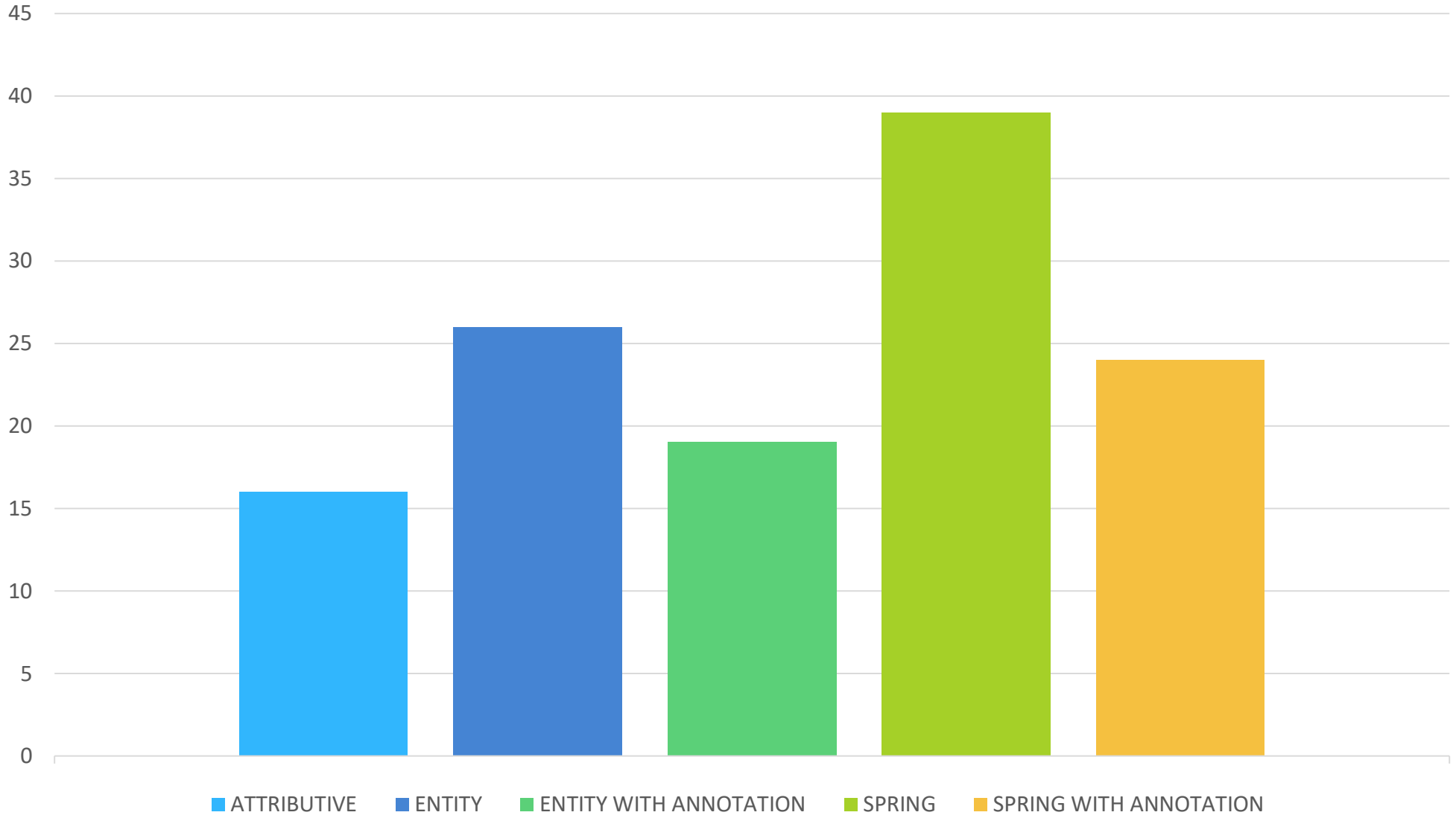


{Easyabac API}

- ATTRIBUTIVE
- ENTITY
- ENTITY WITH ANNOTATION
- SPRING
- SPRING WITH ANNOTATION

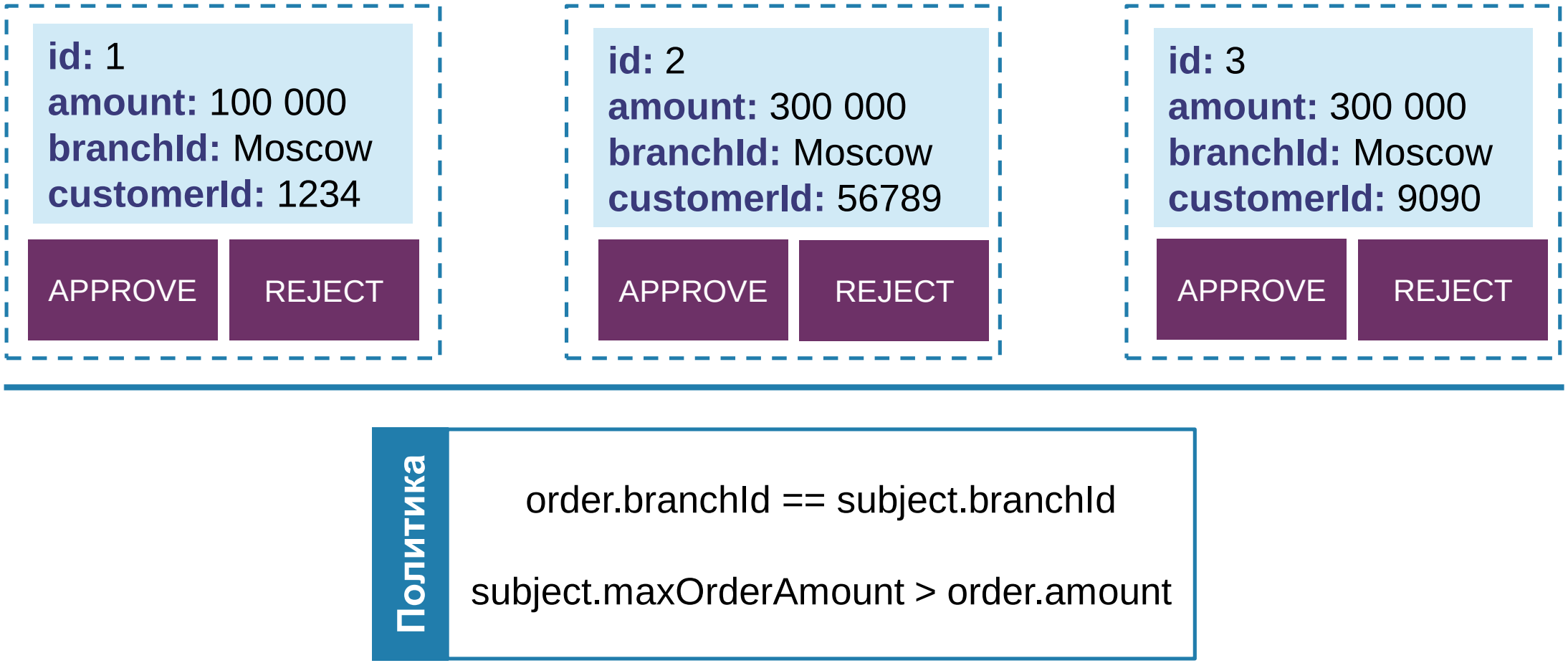
Milliseconds per operation

CUSTIS®



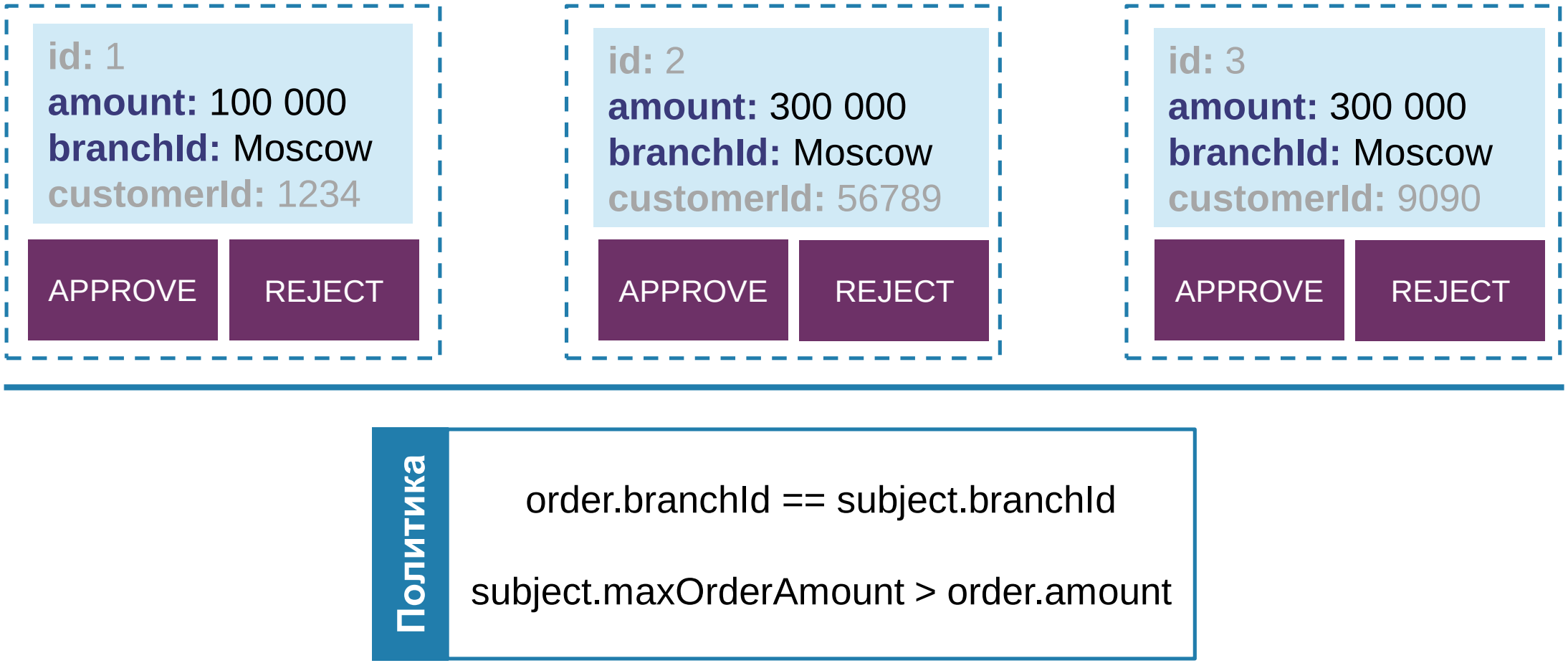
{Оптимизация мультизапросов}

CUSTIS®



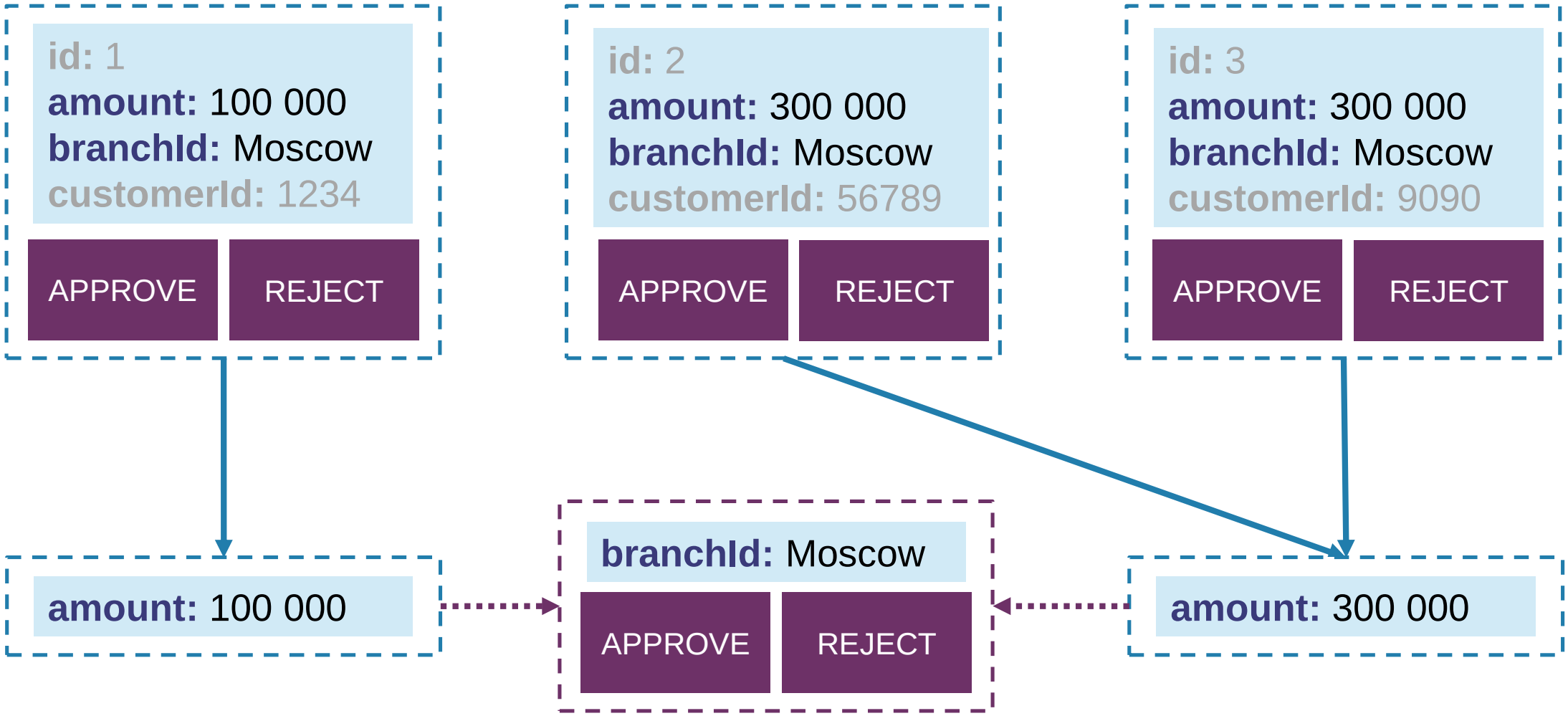
{Оптимизация мультизапросов}

CUSTIS®



{Оптимизация мультизапросов}

CUSTIS®



{Оптимизация мультизапросов}

CUSTIS®

- Зависит от конкретной задачи
- На небольших запросах не требуется
- Настраиваемый порог свертки

{Easyabac:roadmap}

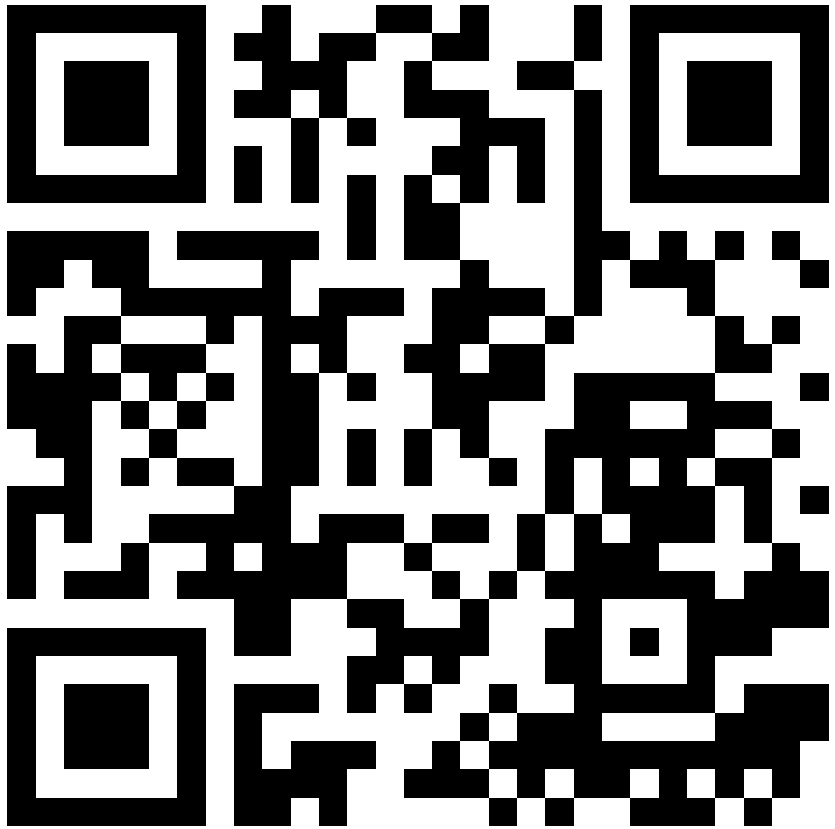
- Загрузка модели из различных источников
- Standalone EasyABAC
- Ограничение доступа к данным
- Упрощенный расчет политик
- Дополнительные расширения (Active Directory, …)

{Summary}

- Атрибутивный подход к авторизации
- Стандарт XACML и его реализации
- Open-source фреймворк EasyAbac

{Easyabac:source}

CUSTIS®



GitHub: CUSTIS-public/EasyABAC

Twitter: @easyabac

CUSTIS®

СПАСИБО ЗА ВНИМАНИЕ!

Антон Лапицкий

Архитектор приложений

custis.ru

E-mail: alapitskiy@custis.ru

Twitter: @anton_lapitskiy