Служба контроля доступа

Служба контроля доступа (СКД) является частью подсистемы управления доступом (ПУД). Она контролирует доступ к доменным объектам и глобальным операциям в системе. Кроме того, она осуществляет протоколирование этих событий.

В основе реализации системы лежит принцип передачи маркеров доступа (authorization или access tokens). Маркер доступа содержит в себе информацию о субъекте, объекте и типе доступа. Он формируется только в том случае, если данный тип доступа к данному объекту разрешён данному субъекту. (Это пуристский подход, который в целях оптимизации быстродействия следует несколько усложнить – см. далее.) Для выполнения любой базовой операции с объектом (чтение, запись, удаление) требуется наличие маркера доступа на соответствующий тип доступа к этому объекту. Создаваться маркеры доступа могут только через СКД по запросу бизнес- или (реже) DAO-слоя системы. Хорошей идеей выглядит хранение маркеров доступа в контексте текущего запроса (транзакции) вместо явной передачи их через параметры методов.

Маркеры доступа на выполнение базовых операций являются простыми. Кроме них, СКД умеет формировать универсальные маркеры, которые могут выступать заместителями (proxies) простых маркеров при проверке прав доступа. Эти маркеры используются, например, для выполнения операций от имени системы.

В рамках предложенной схемы понятие субъектов доступа должно быть расширено по сравнению с описанием настройки прав доступа. Субъектами доступа для СКД являются не только пользователи, но и компоненты системы, осуществляющие изменения объектов, в том числе конфигурируемые действия и бизнес-процессы.

Реализация такой службы создаст единый компонент, контролирующий доступ к защищаемым объектам системы. Доступ к объектам в обход компонента будет невозможен; для выполнения каких-либо операций (даже чтения объектов) от имени системы код должен будет получить универсальный маркер доступа, обратившись в СКД. Теоретически, СКД в соответствии с какими-либо своими настройками может и не предоставить такой маркер, т.е. запретить доступ. Можно также рассмотреть возможность дескриптивного назначения маркера доступа классам или методам (через аннотации или beans.xml).

Ещё одним преимуществом предложенной архитектуры является простота реализации протоколирования событий. Маркер доступа агрегирует всю информацию, необходимую для внесения записи в протокол работы. Протоколирование может осуществляться, например, одновременно с проверкой маркера доступа перед выполнением базовой операции.

# Оптимизация производительности

Существенным недостатком предложенной схемы является необходимость выполнения проверок прав отдельными запросами в БД (при создании маркера доступа), что может негативно сказаться на времени отклика системы. Схема БД при этом построена таким образом, чтобы позволять формирование эффективных запросов самих объектов одновременно с проверкой прав доступа к ним. Таким образом, необходимо отказаться от жёсткого следования принципу проведения проверки в момент формирования маркера доступа.

Итак, некоторые простые маркеры доступа могут формироваться без проверки соответствующих разрешений пользователя на доступ – отложенные (deferred) маркеры. Получив такой маркер, базовый сервис (DAO) обязан позаботиться о включении в запрос соответствующих проверок. Отложенные маркера наиболее важны для эффективной фильтрации коллекций, которая осуществляется путём встраивания фильтров по правам в запросы к БД. Возможно использование отложенных маркеров и для других операций. Это означает, что часть важнейшей функции СКД по проверке прав пользователей эта служба делегирует базовым сервисам управления доменными объектами и коллекциями.

Ещё одним способом оптимизации производительности (а зачастую, и увеличения удобства использования) являются групповые маркеры доступа. Они содержат список объектов доступа (их идентификаторов) либо список типов доступа. Такие списочные проверки легко реализуются единым запросом в БД, существенно сокращая количество обращений к ней, а бизнес-сервисы получают возможность ранней проверки прав доступа к множеству объектов, которые понадобятся для выполнения одного запроса.

# Посмотрим на примерах

* DomainObjectService получает запрос на чтение доменного объекта.  
  Он обращается в СКД, запрашивая маркер доступа на чтение этого объекта (без указания пользователя; СКД может взять его из контекста EJB), затем вызывает DomainObjectDao. (Следует заметить, что при пуристском подходе проверка прав доступа произойдёт при формировании маркера, в случае неудачи будет выброшено исключение, и обработка запроса завершится. В оптимизированном варианте проверка на этом этапе не производится.) DomainObjectDao проверяет полученный маркер доступа на применимость к своей операции и, исходя из того, что маркер является отложенным, формирует SQL для чтение объекта в БД, добавляя в него проверки на права доступа пользователя. Если подходящей записи не нашлось, то либо идентификатор неправильный, либо доступ запрещён, и тогда уже бросается исключение.
* ActionService (сервис обработки конфигурируемых действий) получает запрос на выполнение действия «Создать новый документ».  
  Он обращается к СКД с запросом маркера доступа на выполнение действия. Этот маркер – не отложенный (проверка прав осуществляется сразу), но универсальный: он замещает любые другие маркеры доступа. Субъектом доступа в нём является действие, а не пользователь, вызвавший оное. Получив маркер, сервис передаёт управление классу, указанному в конфигурации действия. Этот класс в соответствии с реализованной в нём бизнес-логикой вызывает DomainObjectDao, который запрашивает маркер доступа на системное действие. Поскольку имеющийся у нас сложный маркер замещает любое действие, требование DomainObjectDao удовлетворено, и тот создаёт доменный объект в БД.
* DomainObjectService получает запрос на сохранение доменного объекта «Поручение». При этом в объекте «Документ» у нас сконфигурировано вычисляемое поле «Срок исполнения», которое вычисляется специальным классом как максимальное значение поля «Срок исполнения» из всех дочерних доменных объектов «Поручение». Разумеется, вычисление этого поля в родительском документе должно производиться при изменении объекта «Поручение».  
  Итак, вначале всё идёт так же, как при чтении объекта. Вопрос о том, является ли маркер доступа на запись объекта отложенным, здесь не важен. После сохранения в БД объекта «Поручение» вызывается код нашего специального класса для вычисления поля (через точку расширения, видимо). И поскольку он будет читать множество объектов, и даже менять один (документ), а вовсе не факт, что у сохранившего поручение пользователя есть права на все эти операции, класс должен будет обратиться к СКД за универсальным маркером доступа. При наличии такого маркера класс сможет производить любые манипуляции с объектами системы (действовать от её имени), но именно он становится субъектом этих операций, и это будет отражаться в протоколе.
* Activiti исполняет service task бизнес-процесса.  
  Поскольку бизнес-процесс действует от имени системы, инфраструктура взаимодействия с Activiti в ходе подготовки среды исполнения для класса, реализующего этот task (JavaDelegate), запрашивает универсальный маркер доступа. Субъектом доступа провозглашается данный бизнес-процесс. Любой базовый сервис, проверяя соответствие маркера своей операции, получает положительный ответ и может выполнять запрошенные действия.
* CollectionService получает запрос на чтение списка (коллекции) объектов.  
  Он обращается в СКД, запрашивая доступ на чтение коллекции объектов (отдельный тип доступа!) Предоставленный маркер является отложенным, значит, исключение по нехватке прав здесь невозможно. CollectionServiceDao, получив маркер и определив, что он отложенный, встраивает в запросы проверки прав доступа для субъекта (пользователя), указанного в маркере доступа.

# А теперь – слайды!

Схема взаимодействия СКД с различными компонентами системы:

