Никитин А.А. Метод проверки прав доступа пользователя на сервер через gRPC на примере образовательной платформы // Международный журнал информационных технологий и энергоэффективности. – 2025. – Т. 10 № 4(54) с. 23–28

Международный журнал информационных технологий и энергоэффективности

Сайт журнала: http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.021

МЕТОД ПРОВЕРКИ ПРАВ ДОСТУПА ПОЛЬЗОВАТЕЛЯ НА СЕРВЕР ЧЕРЕЗ GRPC НА ПРИМЕРЕ ОБРАЗОВАТЕЛЬНОЙ ПЛАТФОРМЫ

Никитин А.А.
ФГБОУ ВО "МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)", Москва, Россия, (125993, Москва, Волоколамское ш., д. 4), e-mail: lyosha-2001@mail.ru

При разработки различных образовательных платформ, одну из центральных ролей в проектировании играет проверка ролей и прав доступа для пользователей. В данной работе рассматривается реализация одного из методов проверки прав доступа на сервере, где клиент и сервер общаются под средствами gRPC запросов. gRPC требует заранее определенных контрактов, по которым строиться генерация кода: на стороне сервера и клиента. В свою очередь, по кодогенарации строятся запросы в виде объектов, которые могут обрабатываться на стороне сервера.

Ключевые слова: gRPC, права доступа, веб-сервис, микросервисы, база данных.

A METHOD FOR VERIFYING USER ACCESS RIGHTS TO THE SERVER VIA GRPC USING THE EXAMPLE OF AN EDUCATIONAL PLATFORM

Nikitin A.A.
MOSCOW AVIATION INSTITUTE (NATIONAL RESEARCH UNIVERSITY), Moscow, Russia, (125993, Moscow, Volokolamskoye shosse, 4), e-mail: lyosha-2001@mail.ru

When developing various educational platforms, one of the central roles in design is the verification of roles and access rights for users. In this paper, we consider the implementation of one of the methods for verifying access rights on a server, where the client and server communicate using gRPC requests. gRPC requires pre-defined contracts for code generation: on the server and client sides. In turn, code generation builds queries in the form of objects that can be processed on the server side.

Keywords: gRPC, access rights, web service, microservices, database.

Общение клиента и сервера строиться на обмене данных между ними: клиент отправляет запрос, а сервер принимает и обрабатывает, в конце отдавая ответ. При общении учитывается множество факторов, но можно выделить основные: протокол передачи данных, формат данных, производительность. Обобщая, можно говорить о двух подходах между общением: gRPC, REST. Каждый из них является уникальным, например, gRPC является более быстрым, имеет типизацию запросов при описании контрактов для взаимодействия с сервером, но на данный момент не поддерживается браузером напрямую, поэтому требует дополнительных узлов для обработки запросов, приводящих данные к нужному протоколу, при этом данный узел может выступать в роли балансировщика запросов. Несмотря на данный минус, gRPC набирает популярность при проектировании интернет-сервисов.

Интернет-сервисы призваны решать потребительские проблемы для пользователей и должны подходить к ним с разных сторон, например, для образовательной платформы

необходимо рассмотреть пользователей в роли автора курсов, а также в роли потребителя или учащегося, проходящего этот курс, также будет преимуществом создание администратора платформы, но не обязательным, так как можно заниматься администрированием под средством обращения к серверу на прямую или же к базе данных.

Исходя из вышесказанного можно сказать, что интернет-сервис, обладающий определенной бизнес-логикой, работающей по-разному для различных пользователей, должен обладать определенной ролевой моделью с описанием прав доступа к ним.

Описание прав доступа по ролевой модели играет важную роль при проектировании. Для описание необходимо понимать, что оно должно быть представлено в виде табличных данных (Таблица 1), где описываются действия и разрешение для пользователей, а также дополнительные условия. Иными словами, права доступа необходимо задокументировать, это важно не только для серверной разработки, но и для клиентской.

Таблица 1 - Примеры прав доступа для курсов образовательной платформы

	Курсы образовательной платформы		
	Учащийся	**Автор**	**Администратор**
Создание	-	+	+
Чтение	+	+	+
Обновление	-	+ (своего курса)	+ (всех курсов)
Удаление	-	+ (своего курса)	+ (всех курсов)

Права доступа должны выдаваться на серверной части приложения: обычно если проект является монолитным, то есть два подхода: первый - создание прав доступа под средствам миграции данных и второй - написание прав доступа непосредственно в кодовой базе приложения, но оптимальным подходом является накатывание прав доступа с миграциями для базы данных, но есть сложность, когда становиться много данных в миграции, велик шанс внесения ошибочных данных, поэтому данный подход расширяться через описание прав доступа и внесением их в базу данных в коде.

При использовании микросервесной архитектуры права доступа можно хранить в отдельном пользовательском микросервисе, где права доступа передаются от микросервиса с бизнес-логикой, но появляется масса сложностей с поддержанием такого подхода: возникает большая связность между микросервисами и запускать, например, отдельно микросервис курсов становиться невозможным без пользовательского, также возникает проблема с поддержанием согласованности данных, оптимально же рассматривается подход, когда каждый микросервис самостоятельно накатывает права доступа в свою отведенную базу данных и дополняет определенными сведениями, а вся пользовательская информация запрашивается с пользовательского микросервиса или же передается через клиентскую часть, но возникает проблема согласованности данных.

Рассмотренный случай при монолитном проекте можно рассмотреть и для каждого микросервиса в отдельности, что позволит запускать каждый микросервис отдельно, из

минусов необходимо выделить, что может происходить дублирование логики между микросервисами.

Цель работы: описать метод проверки прав доступа пользователей через gRPC на примере образовательной платформы.

Для достижения цели необходимо решить задачи:

- рассмотреть образовательную платформу [1];
- описать метод проверки прав доступа.

В работе [1] описана архитектура образовательной платформы по изучению различных авиационных материалов. Необходимо подчеркнуть, что образовательная платформа имеет микросервисную архитектуру, поделенную на 5 микросервисов: работа с курсами, пользователями, справочных материалов, интерактивной карты и интеграции с чат-ботом. Каждый микросервис включает в себя работу с пользователями: вся необходимая информация запрашивается с пользовательского микросервиса и сохраняется в базе каждого отдельного микросервиса, а также реализуется работа проверка прав доступа. Каждый микросервис использует чистую архитектуру [2]. Верхнеуровневая реализация проверки показана на Рисунке 1.

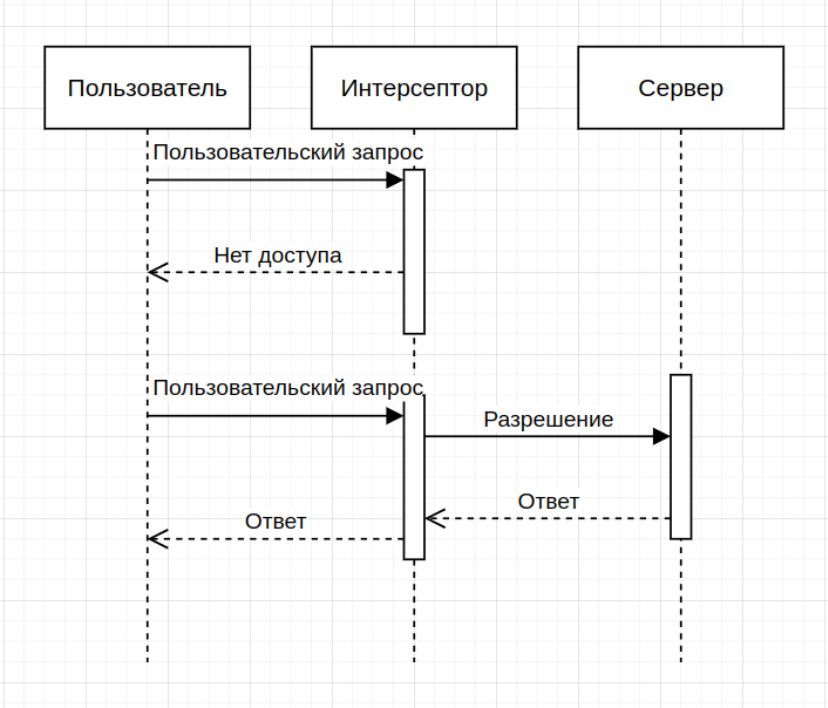

Рисунок 1 - Проверка прав доступа для пользователя

Микросервисы образовательной платформы пишутся на высокопроизводительном языке программирования golang и gRPC для быстрого взаимодействия между ними. Golang и gRPC разрабатываются и проектируются в рамках одной организации google, что привело к тому, что golang хорошо поддерживает gRPC, позволяя пользоваться всеми преимуществами: кодогенрацией, интерсепторами и т.д.

При использовании gRPC сначала описываются контракты. Контракты могут выступают в роли документации, но на данный момент документация создается с кодогенерацией и

комментариев в контрактах (рисунок 2). Которые потом мигрирует в код микросервиса, это необходимо для автоматизации и быстрого написания кодовой базы.

```
// UserInfo структура для получения информации о текущем пользователе
message UserInfo {
  optional int64 id = 1; // Уникальный идентификатор
  string portal_code = 2; // Код пользователя для телефонной книги
  string firstname = 4; // Имя пользователя
  string lastname = 5; // Фамилия пользователя
  repeated string email = 6; // Адрес электронной почты пользователя
  repeated string phone = 7; // Телефон пользователя
  string avatar = 8; // Ссылка на аватар пользователя
  string position  = 9; // Должность пользователя из телефонной книги
  repeated permission.Role roles = 10; // Список ролей пользователя
}
```

UserInfo

UserInfo структура для получения информации о текущем пользователе

Field	Type	Label	Description
id	int64	optional	Уникальный идентификатор
portal_code	string		Код пользователя для телефонной книги
firstname	string		Имя пользователя
lastname	string		Фамилия пользователя
email	string	repeated	Адрес электронной почты пользователя
phone	string	repeated	Телефон пользователя
avatar	string		Ссылка на аватар пользователя
position	string		Должность пользователя из телефонной книги
roles	permission.Role	repeated	Список ролей пользователя

Рисунок 2 - Пример контракта для пользователя и сгенерированной документации

Frontend при использовании gRPC использует proxy серверы, например, envoy. Данный сервер перехватывает запросы перед отправкой на сервер и конвертирует запрос в нужный формат данных понятный для gRPC. Envoy способен балансировать запросы на сервер, что делает общение через gRPC актуальным.

Проверка прав доступа заключается в том, что при каждом запросе на сервер необходимо проверять доступ пользователя к ресурсу. На сервере для этого используется перехватчик (middleware, interceptor), который будет перехватывать запрос от клиента и проверять доступ (Рисунок 3).

Схема выглядит довольно простой, пользователь хочет получить результаты определенной бизнес-логики, для этого пользователь отправляет запрос с определенными метаданными о себе, например, это может быть JWT токен или UUID пользователя для того, чтобы однозначно определить пользователя в системе. При отправке запроса, он проходит через перехватчика приложения, где сначала вытаскиваются метаданный пользователя,

отправляются в обработчик для пользователя и в контекст программы вставляется пользователь, если же не удалось определить пользователя в системе отдается ошибка в получении ответа от бизнес-логики.

После идет проверка прав доступа: из контекста достается пользователь с определенными ролями, а также из запроса вытаскивается вся необходимая информация на какой метод бизнес-логики должен дойти данный запрос: данная информация представлена в виде названия метода. Данные о ролях пользователя и названия метода передаются в обработчик для прав доступа, где если есть в базе данных строчка роль и название метода, то выдается разрешение для дальнейшего запроса в бизнес-логику, которая отдает ответ.

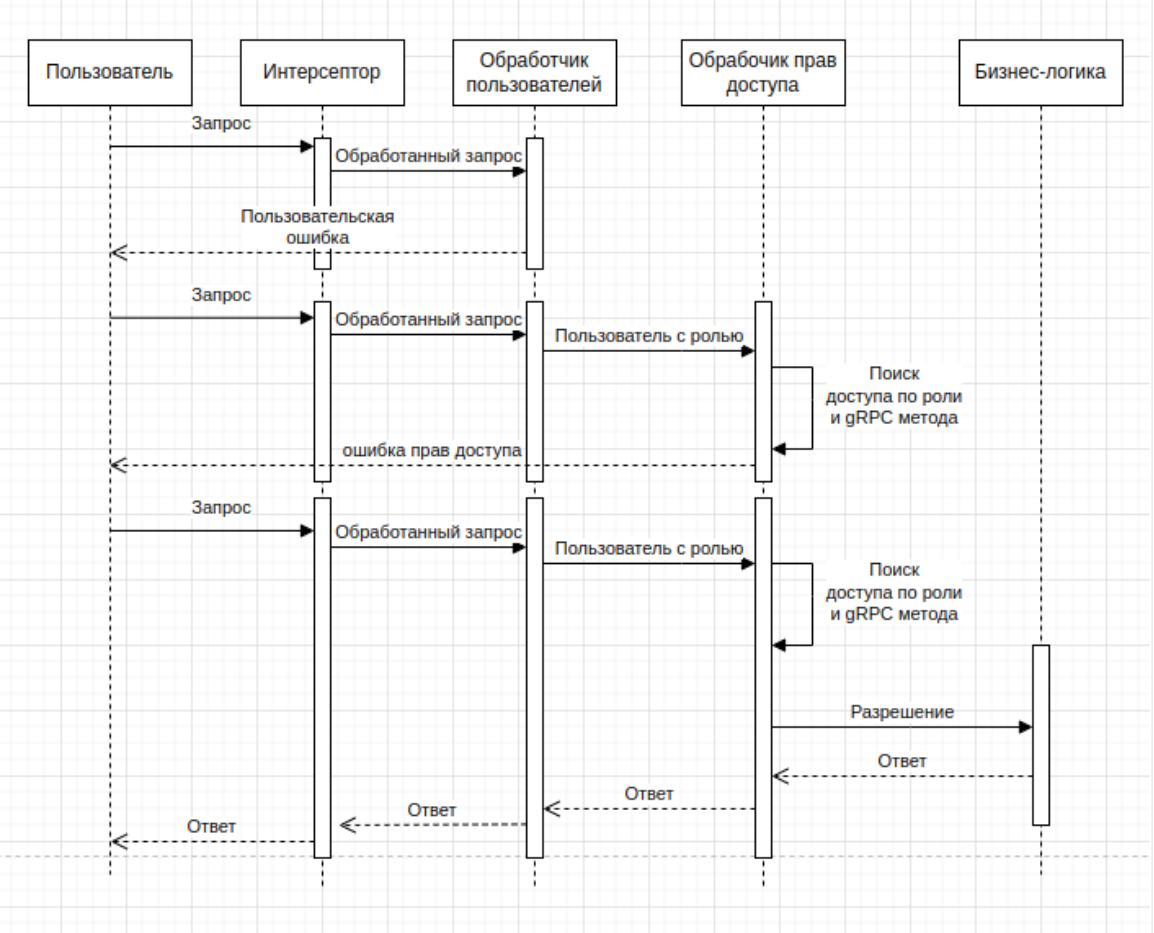

Рисунок 3 - Обработка прав доступа

Для внесения прав доступа пользователям образовательной платформы в БД сначала выделяются роли: студент, автор курса. Для корректного внесения в БД необходимо использовать паттерн проектирование builder (строитель) [3], который в зависимости от роли будет перед развертыванием приложения накатывать миграции в БД. Строитель берет описанные название gRPC методов из кодогенерации и распределяет права доступа для ролей. Для каждой роли может быть собственный стек методов. Например, в рамках образовательной платформы бизнес-логика должна обрабатывать CRUD для курсов, где студент имеет право только на прохождение курсов, а автор на создание и удаление собственного курса.

Для написания обработчика необходимо учесть, что при кодогенерации gRPC создает сервисы, который описаны в контрактах, поэтому необходимо внимательно подходить к их написанию. Каждый сервис имеет набор собственных методов. Отталкиваясь от сервиса,

необходимо построить путь при перехвате запроса: перехват запроса с метаданными о пользователе, получения пользовательских данных и его авторизация, переход к нужному gRPC сервису, затем переход к нужному обработчика перед использованием бизнес-метода, а дальше заключающий этап проверки – нахождение прав доступа в БД.

Выводы: образовательная платформа по изучению авиационных материалов имеет небольшое количество микросервисов, значит обработка пользователей для каждого микросервиса не вызовет большое количество проблем, а наоборот решит главные вопросы: поддержание актуальности данных, возможность развертывания вне зависимости от других микросервисов, проверка прав доступа через gRPC под требования определенной бизнес-логики, что позволит разграничивать зоны ответственности.

Надо отметить, что в методе проверки прав доступа используется builder для создания прав и ролей пользователей в БД, а также используется interceptor для проверки прав доступа. Данный реализация позволит ускорить разработку приложения, а также упростит работу с пользователями.

Список литературы

1. Никитин, А.А. Архитектура высоконагруженного интернет-сервиса: образовательная платформа для изучения авиационных материалов / Никитин А.А. / МАИ, г. Москва – 1 с.
2. Мартин, Роберт. Чистая архитектура. Искусство разработки программного обеспечения / Роберт Мартин ; перевел с английского А. Киселев. - Санкт-Петербург [и др.] : Питер, 2021. - 350 с. : ил. - (Серия "Библиотека программиста"). - Пер. изд.: Clean architecture. A craftsman's guide to software structure and design / Robert C. Martin. - 2018.
3. Гамм, А. Паттерны объектно-ориентированного проектирования / А. Гамм, М. Хел, Н. Джонсо, В. Д. - СПб. : Питер, 2021. - 448 с.

References

1. Nikitin, A.A. Architecture of a highly loaded Internet service: an educational platform for studying aviation materials / Nikitin A.A. / MAI, Moscow – p.1
2. Martin, Robert. Clean architecture. The Art of Software Development / Robert Martin; translated from English by A. Kiselyov. - St. Petersburg [and others] : Peter, 2021. 350 p. : ill. (Series "Programmer's Library"). - Per. ed.: Clean architecture. A craftsman's guide to software structure and design / Robert C. Martin. - 2018.
3. Gamm, A. Patterns of object-oriented design / A. Gamm, M. Khel, N. Jones, V. D. - St. Petersburg : Peter, 2021. - p. 448