Добрый день уважаемые члены и председатель комиссии. Меня зовут Дерид Артем Александрович и тема моего дипломного проекта – разработка библиотеки для межсетевого связывания микросервисов для платформы .net core.

Проблема обновление ПО крупных предприятий является важной в наше время. Это связано с тем что такие системы сложны, зачастую они имеют повышенные требования к производительности, надёжности и безопасности. Разработка таких систем ведётся долгое время, их законченная монолитная структура осложняет модернизацию и расширение их функционала.

Моя библиотека создавалась для решения проблем коммуникации систем крупных предприятий, построенных на платформе .NET, одна часть которых находится в публичном доступе, а другая скрыта за межсетевым экраном.

+++ Сетевое устройство системы после внедрения.

На данном слайде мы можем увидеть условное устройство системы после внедрения моего продукта. Библиотека позволит создать ещё один безопасный и гибкий канал связи для доступа к внутренним сервисам, обработки ими информации и возвращение результата отправителю.

+++ Основные модули системы

Используя статический анализ и генерацию кода, я реализовал основные модули моей системы, такие как анализатор типов, шину связи и инструменты соединения, которые позволяют создавать слепок любого сервиса, работающий идентично внутреннему для стороннего разработчика.

Анализатор типов позволяет автоматически создать классы декораторы над любыми существующими сервисами. Используя их система создаёт объекты близнецы, которые взаимодействуют друг с другом посредством специальных протоколов передачи сообщений и имеют поведение, благодаря которому их можно считать единым экземпляром класса.

+++ Алгоритм работы анализатора типов

На данном слайде изображен алгоритм работы анализатора типов. Анализатор изучает все поля и свойства, методы и события, а также удаляет информацию о всех закрытых или служебных частях сервиса. После этого анализатор создаёт специальный тип содержащий все метаданные для дальнейшей работы.

Коммуникация между частями сервиса происходит посредством следующего важнейшего узла системы, а именно шины связи с сервером сообщений RabbitMQ.

+++ Чертёж 1

На данном слайде мы видим алгоритм подключения обоих частей сервиса к серверу сообщений, создание декораторов для внутренней и внешней части сервиса, их запуск и подключение к очереди сообщений RabbitMQ.

+++ Чертёж 2

На следующем слайде изображён алгоритм отправки сообщений между частями сервиса после начальной инициализации. Данный алгоритм является бесконечным циклом и выполняется всё время жизни программы. Он отслеживает изменение полей, вызовы методов, отправку сообщений о событиях, а также имеет механизмы защиты от потери данных при сбоях или высоких нагрузках системы.

+++ Устройство тестового стенда

Надежности системы так же уделено должное внимание. С самого начального этапа разработки мы имели тесты поведения системы в нестандартных ситуациях, модульные тесты, интеграционные тесты, а также тесты производительности и нагрузки.

Для последних был создан специальный тестовый стенд, который с помощью аппаратных и программных средств модулировал работу целевой системы после внедрения нашей библиотеки.

+++ Следующие этапы разработки системы

На данный момент мы имеем законченный продукт, который протестирован и готов к поставкам для заказчиков. Хорошим показателем качества системы является востребованность моей библиотеки, а также договоры о поставке системы в крупную компанию в западной Европе.

Так же мы имеем планы по модернизации системы, дальнейшей её оптимизации и внедрению на других предприятиях.

Резюмируя всё вышесказанное, я считаю, что я выполнил все цели, которые были поставлены мной при начале разработки дипломного проекта. Продукт является востребованным и выгодным для крупных предприятий и уже завоевал доверие и позволил заключить контракты на поставку и модернизацию систем преприятий.

Спасибо за внимание. Я готов ответить на любые интересующие вас вопросы.