

ПРОЕКТИРОВАНИЕ КОРПОРАТИВНОЙ СЕРВИСНОЙ ШИНЫ БАНКА

Шапранов Артём Вадимович, магистрант направления 710400 «Программная инженерия», Кыргызский государственный технический университет им. И. Раззакова, Кыргызстан, 720044, г. Бишкек, пр. Ч.Айтматова 66, e-mail: shapranov.a199@gmail.com.

Мусина Индира Рафиковна, к.т.н., доцент, Кыргызский государственный технический университет им. И. Раззакова, Кыргызстан, 720044, г. Бишкек, пр. Ч.Айтматова 66, e-mail: musina-indira@yandex.ru.

Аннотация. В статье приведены результаты анализа проблем, возникающих при добавлении отдельных автономных подсистем к уже существующей автоматизированной банковской системе (АБС). Показана текущая архитектура взаимодействия всех подсистем. Предлагается использование сервис-ориентированной архитектуры на основе корпоративной банковской сервисной шины, которая позволяет обеспечить бесперебойное взаимодействие всех модулей автоматизированной системы банка. Приводится обоснование выбора новой архитектуры АБС с использованием сервисной шины. Предлагаются модули шины, которые позволят решить проблемы текущей архитектуры банка.

Ключевые слова: автоматизированная банковская система, архитектура компьютерных систем, сервис-ориентированная архитектура, сервисная шина, проектирование информационных систем.

DESIGNING THE BANK'S ENTERPRISE SERVICE BUS

Shapranov Artem Vadimovich, Master's student of the direction 710400 "Software engineering", Kyrgyz State Technical University (KSTU) named after I.Razzakov, Kyrgyzstan, 720044, Bishkek, 66 Ch.Aitmatov Ave., e-mail: shapranov.a199@gmail.com

Musina Indira Rafikovna, Candidate of Technical Sciences, Professor, KSTU named after I.Razzakov, Kyrgyzstan, 720044, Bishkek, 66 Ch.Aitmatov Ave., e-mail: musina-indira@yandex.ru

Abstract. The article presents the results of the analysis of the problems that arise when adding separate autonomous subsystems to the already existing automated banking system (ABS). The current architecture of interaction of all subsystems is shown. It is proposed to use a service-oriented architecture based on the corporate banking bus, which allows for uninterrupted interaction of all modules of the bank's automated system. The rationale for choosing a new ABS architecture using a service bus is given. Bus modules are offered that will solve the problems of the current architecture of the bank.

Keywords: automated banking system, computer system architecture, service-oriented architecture, service bus, design of information systems.

Введение

Сейчас невозможно себе представить управление работой банка, будь он коммерческим или государственным, без автоматизированной банковской системы (АБС), которая представляет собой комплекс аппаратно-программных средств, реализующих мультивалютную информационную систему, обеспечивающую современные финансовые и управленческие технологии в режиме реального времени при транзакционной обработке данных [1]. В настоящее время, в условиях пандемии, становится актуальной добавление новых автоматизированных функций, которые обеспечивают бесперебойное дистанционное обслуживание клиента. Это дает возможность предоставления клиенту банка управлять всеми доступными ему продуктами банка (кредитами, депозитами, счетами, банковскими картами), не приходя в отделения. Подобные услуги позволяют банку быть конкурентоспособным в кредитно-финансовой системе государства.

Однако с увеличением новых автоматизируемых функций, растет и количество подсистем (а чаще это автономные подсистемы), которые надо включить в систему управления деятельностью банка. А это, в свою очередь, приводит к ряду проблем, связанных с корректной работой взаимосвязанных модулей, сопровождением программ, обеспечением безопасности и т.п.

Таким образом, возникает необходимость в обеспечении эффективного автоматического взаимодействия всех подсистем АБС.

Наилучшим решением в этом случае является переход на сервис-ориентированную

архитектуру [2 - 4] путем создания для банка собственной корпоративной сервисной шины, которая позволит обеспечить бесперебойное взаимодействие всех интегрируемых подсистем в одном центре, объединяя существующие источники информации и предоставляя централизованный обмен данными между различными программными модулями.

Разработка и внедрение собственной корпоративной шины обслуживания банка позволит решить ряд проблем.

Проблема 1-дублирование функционала. Зачастую, при добавлении новых автономных подсистем в АБС происходит так, что дублируются функции, т.е. различные процессы (модули) используют аналогичные функции, которые вызывают друг друга. Невозможно объединить все функции в одной системе. Проблема в том, что если вам нужно подкорректировать логику одной функции, то приходится изменения вносить во все подсистемы, которые используют эту функцию.

Проблема 2-длительная реакция на ошибки в системе. Наличие нескольких автономных подсистем усложняет поиск проблем и замедляет скорость реакции на ошибки в системе.

Проблема 3-удлиняется время, затраченное на реализацию аналогичного функционала. Когда разработчик получает задачу разработать подсистему с функциональностью, аналогичной в уже реализованной подсистеме, он должен вначале понять используемый алгоритм функции, чтобы реализовать то же самое в своей подсистеме.

Анализ текущей архитектуры. На примере «Халык Банк Кыргызстан» был проведен анализ текущей архитектуры АБС и ее взаимодействие со всех автономных подсистемами банка. Анализ показал, что в архитектуре есть ряд недостатков. На рисунке 1 показана архитектура взаимодействия между внутренними банковскими подсистемами.

АБС включает в себя ядро и собственные «доработки» разработчиков, которые позволяют обеспечить взаимодействие подсистемы с ядром. Внутренние корпоративные автономные подсистемы взаимодействуют с АБС только через модули собственных модификаций. Для интеграции с внешними системами (процессинговыми центрами, терминальными сетями, поставщиками услуг) разработаны интеграционные уровни, позволяющие АБС получать и отправлять данные в формате, доступном для АБС. Функции интеграционных слоев часто дублируются, меняются только интерфейсы, в зависимости от протокола взаимодействия с внешней системой. Поэтому, когда меняется логика взаимодействия с внешними системами, приходится менять ее в нескольких внутренних подсистемах, а это занимает немало времени. Из-за того, что центральным звеном обработки данных является АБС, нагрузка на ее сервер увеличивается, что может привести к сбоям в работе банка.

Способы решения проблемы. Проблему большой нагрузки на АБС можно решить двумя способами.

Первый способ заключается в вертикальном масштабировании системы АБС, т. е. в увеличении производительности за счет модернизации серверного оборудования. Преимущество этого метода заключается в том, что нет необходимости тратить время на разработку новых решений: архитектура программного обеспечения останется неизменной. Недостатком этого метода является то, что со временем нагрузка будет увеличиваться, и через некоторое время банк снова столкнется с этой же проблемой.

Второй метод – это горизонтальное масштабирование, то есть разделение нагрузки на несколько серверов. В этом случае нагрузка на АБС будет распределена между несколькими серверами. Однако реализовать это в рамках АБС невозможно, поскольку АБС является монолитным программным продуктом.

Для распределения нагрузки можно использовать несколько протоколов обмена информацией: 1) обмен файлами определенной структуры 2) общие (шлюзовые) таблицы 3) обмен информацией с использованием сервисов [2].

Наиболее надежным, безопасным, отказоустойчивым вариантом является обмен информацией с использованием сервисов, т.к. это взаимодействие происходит моментально в отличие от двух других, также взаимодействие через сервисы безопаснее т.к. в настоящее время разработано множество механизмов авторизации, аутентификации в сервис-ориентированной архитектуре. Для более эффективного сервисного взаимодействия необходимо использовать сервисную шину, которая будет маршрутизировать и обрабатывать запросы и позволит иметь одну контролируемую точку входа для выполнения множества операций.

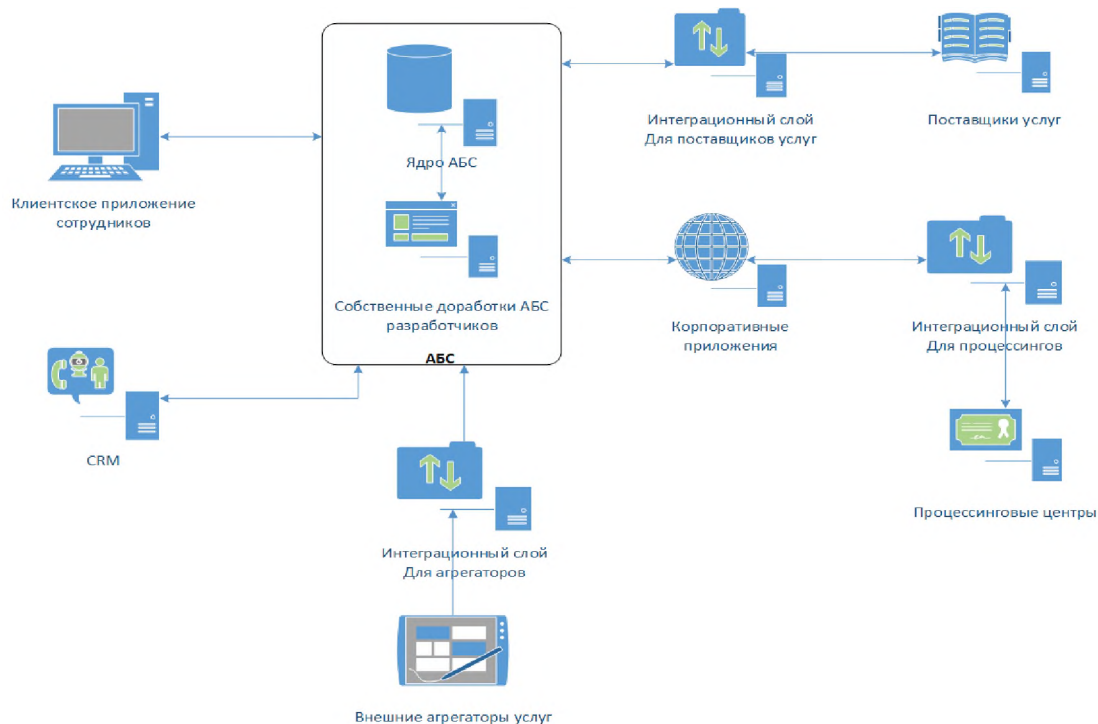


Рис. 1. Текущая архитектура АБС и ее взаимодействие со всеми системами (модель AS-IS)

На сегодняшний день существует несколько решений реализации корпоративной сервисной шины (КСШ) от крупных компаний. Все они разрабатывались на протяжении нескольких лет и имеют множество версий, каждая из которых имеет свое назначение. Минусами таких систем, в нашем случае является:

- 1) высокая стоимость;
- 2) системы тяжеловесны и ресурсоемки;
- 3) необходимы затраты на обучение сотрудников т.к. системы являются конструкторами, в которых для настройки интеграций необходимо выполнить большое количество интуитивно непонятных действий;
- 4) решение продается не узкоспециализировано, а целым пакетом, состоящим из большого набора программных продуктов, предоставляющий излишний функционал, который нигде не будет использован.

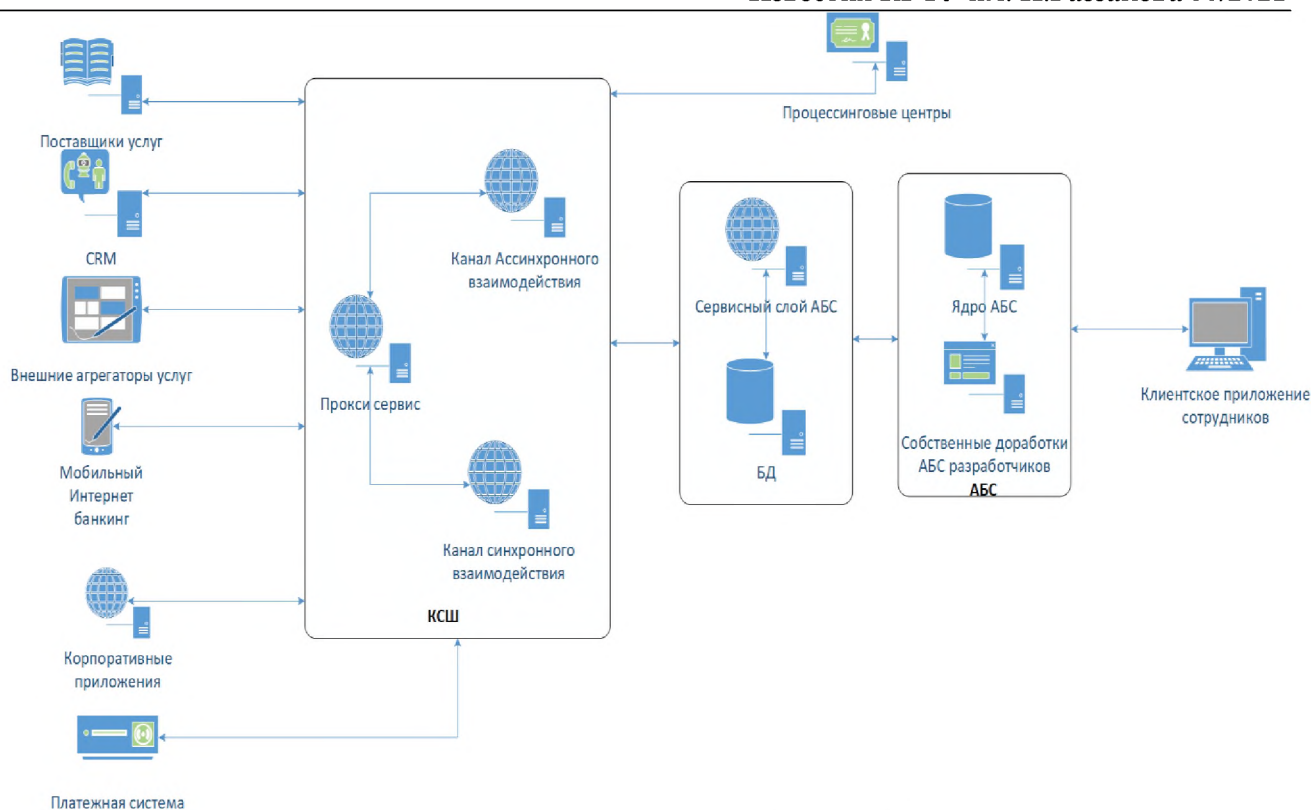


Рис. 2. Целевая архитектура (модель TO-BE)

Целевая архитектура. Для децентрализации обработки данных АБС, необходимо построить сервисный слой АБС, который будет иметь свою собственную базу данных и периодически синхронизировать свои данные с базой данных АБС. Таким образом децентрализуется обработка данных, касающихся счетов клиентов. При этом снизится нагрузка на АБС, но прямой доступ к операциям с любыми счетами банка опасен тем, что система может провести операцию не только с клиентским счетом, но и с внутрибанковскими счетами. Также сервисный слой АБС не позволяет проводить операции с картами, т.к. подобными операциями занимаются сервисы процессинговых центров. Сервисная шина позволит предоставлять доступ системам только к ограниченному списку операций, а также унифицировать общие функции, избежав дублирования функционала. На рисунке 2 представлена предлагаемая целевая архитектура, которая позволит решить все описанные выше проблемы.

Как показано на рисунке 2, посредником между внешними и внутренними транзакционными и информационными системами будет корпоративная сервисная шина, которая будет осуществлять маршрутизацию запросов систем и проверять возможность выполнения системой указанной операции.

Описание корпоративной сервисной шины. Для корректной работы корпоративная сервисная шина должна включать в себя следующие подсистемы:

1. подсистему авторизации, которая будет осуществлять проверку доступов внешних систем к запрашиваемым операциям;
2. подсистему логирования, которая будет фиксировать ход обработки запроса;
3. подсистему синхронного взаимодействия, позволяющую выполнять операции в синхронном режиме;
4. подсистему асинхронного взаимодействия, позволяющую выполнять операции в асинхронном режиме;
5. подсистему маршрутизации, которая отвечает за перенаправление запросов в целевые сервисы.

Рассмотрим подробнее каждую из подсистем.

Подсистема авторизации. Эта подсистема должна выполнять следующие функции: аутентификацию внешней системы; генерацию токена; авторизацию внешней системы; деактивацию токена.

Подсистема логирования. Функции этой подсистемы: фиксация лога -функция, которую будет вызывать внутренняя подсистема; классификация лога – определяет класс лога (уведомление, критическая ошибка, информационный лог); отправка уведомления о критической ошибке.

Подсистема маршрутизации должна выполнять следующие функции: авторизация внешней системы; поиск маршрута – функция поиска целевого сервиса; перенаправление запроса; обработка ответа – функция проверки и конвертации ответа целевого сервиса для внешней системы; формирование ответа внешней системе.

Подсистемы синхронного и асинхронного взаимодействия могут выполнять одни и те же запросы. Разница заключается в способах обработки ответа запрашиваемой системой. На рисунке 3 представлена UML - диаграмма [5, 6] последовательности, отображающая взаимодействие внешней системы с подсистемами КСШ. Из рисунка видно, что внешняя система ожидает выполнения операции и ответа о статусе выполнения до тех пор, пока операция не будет выполнена.

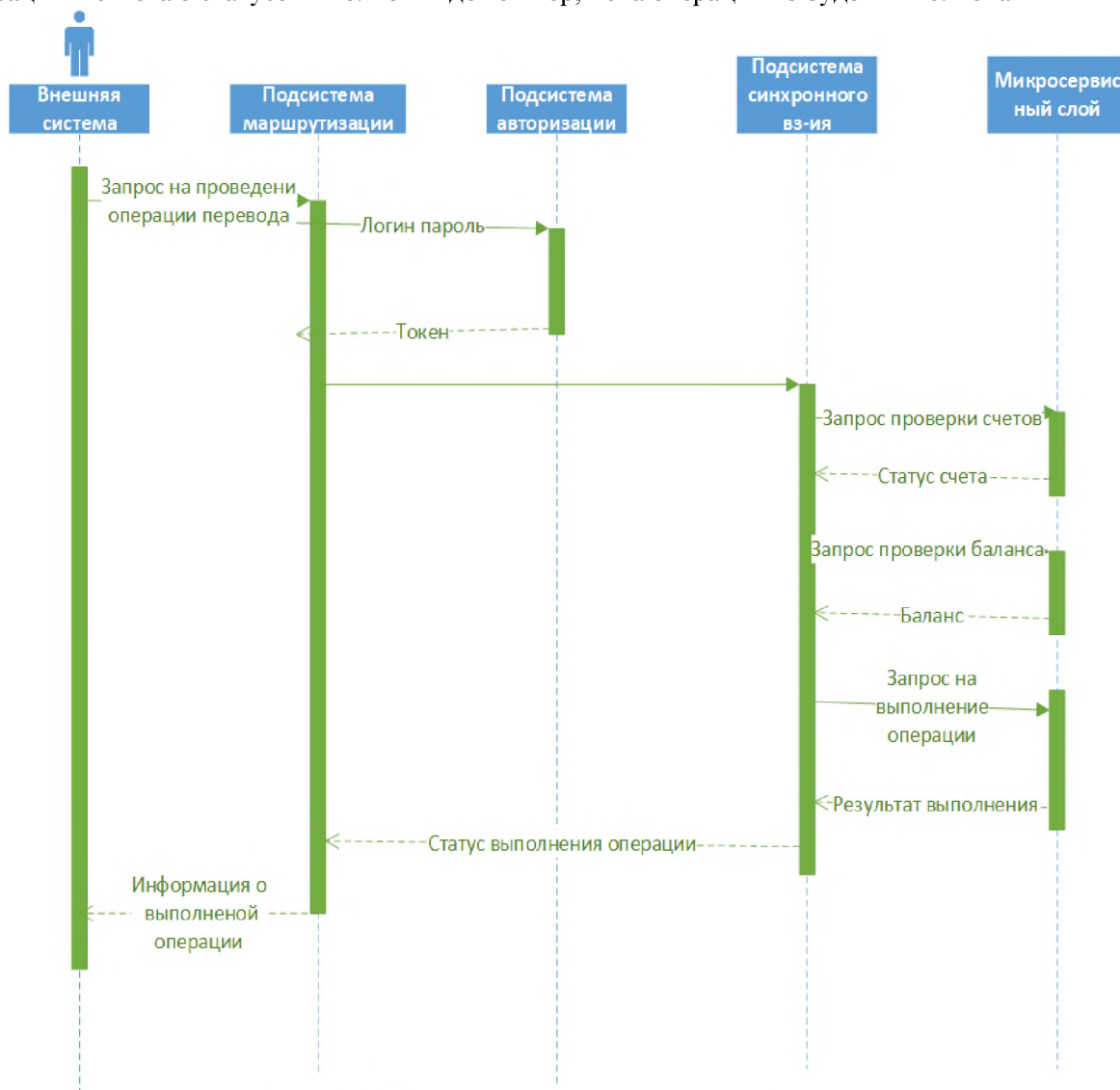


Рис. 3. Подсистема синхронного взаимодействия

На рисунке 4 представлена диаграмма последовательности, которая отображает взаимодействие внешней системы с подсистемами КСШ при асинхронном взаимодействии. В системе асинхронного взаимодействия внешняя система отправляет запрос на выполнение операции и получает ответ, о том, что операция принята в обработку. Для того, чтобы получить информацию о статусе выполнения операции, внешней системе, необходимо спустя некоторое время отправить запрос на проверку статуса операции. Так работа внешней системы никак не приостанавливается во время выполнения операций.

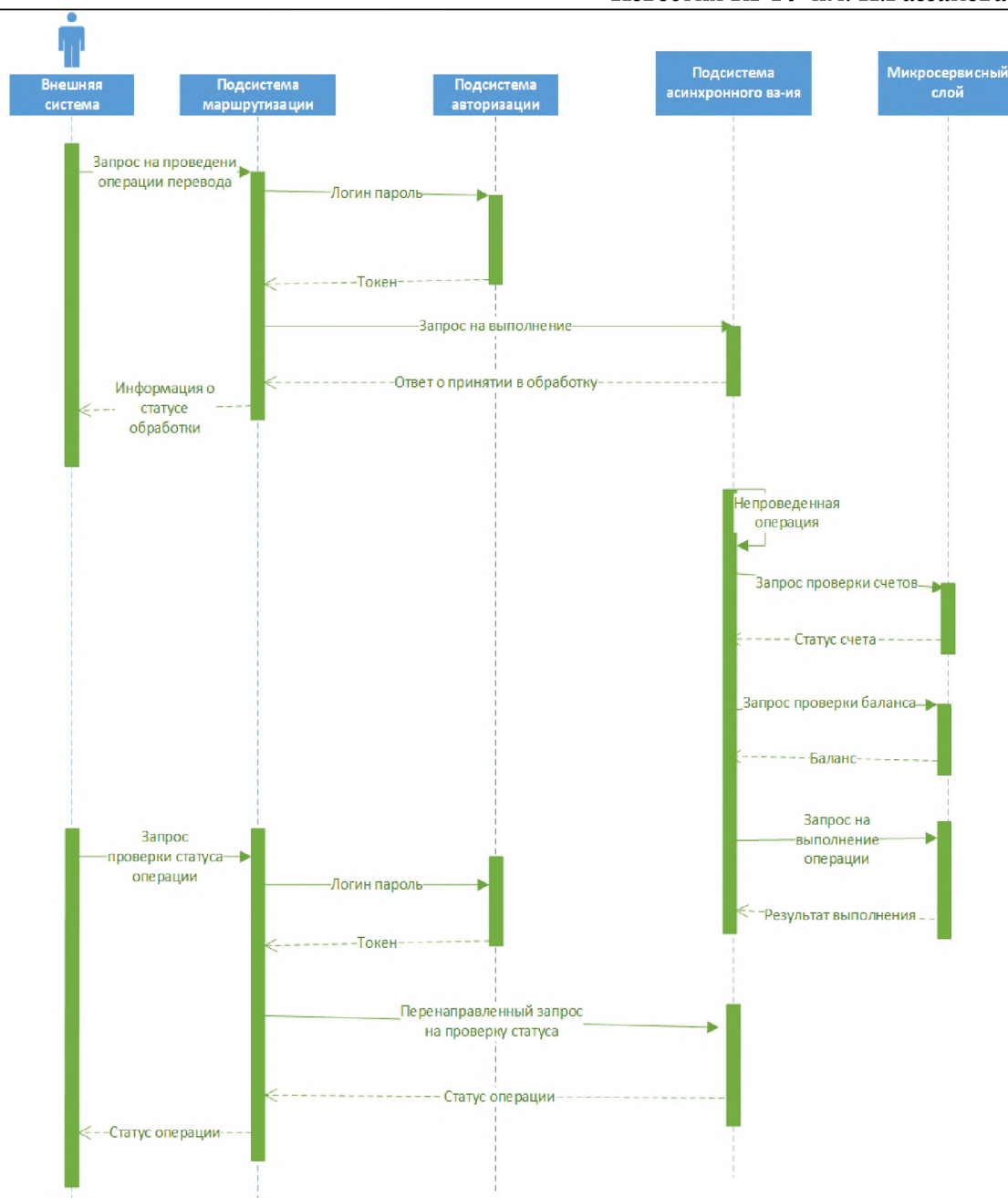


Рис. 4. Подсистема асинхронного взаимодействия

В случае долгой обработки запроса внешние системы, как правило, ожидают ответ не более одной минуты, после чего закрывают соединение, при этом запрос продолжает обрабатываться. Так система, отправляющая запрос не получает обратной связи от сервиса и данные теряются. Реализация подсистем синхронного и асинхронного взаимодействия поможет снизить нагрузку на сеть и избежать возможную потерю данных.

Предварительные исследования на прототипе подсистемы показали, что пропускная способность системы, используя асинхронное взаимодействие увеличится на 1000 обращений в секунду, при этом внешняя система, гарантировано получит корректный ответ о статусе операции, без потери информации.

Принцип построения информационных сервисов. Банк осуществляет выпуск и поддержку двух типов карт – Visa и Элкарт. Обработкой и хранением основного количества информации по этим типам карт осуществляют два, никак не связанных друг другом, процессинговых центра, откуда банку в различных бизнес-процессах, необходимо получать информацию о картах клиента. Для корректной работы внутренних и внешних подсистем банка необходимо взаимодействие с обоими процессинговыми центрами. Для достижения этого требуется унифицировать взаимодействие, поскольку интерфейсы и логика работы у разных процессинговых центров отличается. Объединение

логики работы с обоими процессинговыми центрами в одно приложение сервиса может повлечь за собой ряд проблем.

- 1) Проблемы при доработке нового функционала. Доработка логики взаимодействия с одним процессинговым центром может сказаться на логике работы с другим.
- 2) Количество тестов при внедрении. Помимо проблемы, описанной выше, после доработки сервиса, появляется необходимость тестировать те функции, которые не были изменены, но на них могут сказаться внесенные изменения. Увеличенное количество тестов замедляет выпуск обновления сервиса.
- 3) Недоступность сервиса на момент обновления. В случае, когда все описанные проблемы будут решены, в момент обновления сервис будет недоступен и не сможет передать информацию на обработку ни в один процессинговый центр.

Для решения перечисленных выше проблем необходимо разбить каждый сервис взаимодействия с процессинговыми центрами на три независимых сервиса:

1. Сервис маршрутизатор – осуществляет обработку запроса от банковской системы и дальнейшую маршрутизацию;
2. Сервис работы с информацией по картам Visa – выполняет необходимую логику и работает только с сервисами процессингового центра карт Visa;
3. Сервис работы с информацией по картам Элькарт - выполняет необходимую логику и работает только с сервисами процессингового центра карт Элькарт.

На рисунке 5 представлена UML-диаграмма компонентов, описывающая набор сервисов необходимых для получения списка карт из двух процессингов.

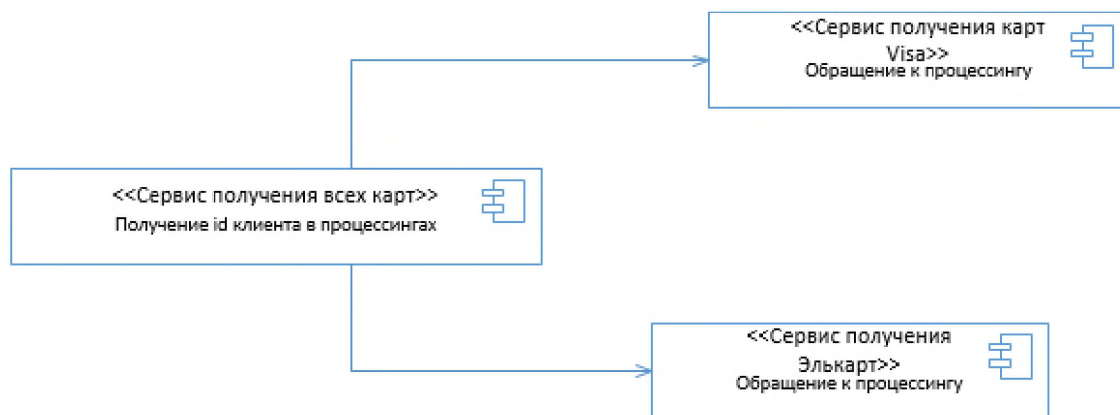


Рис. 5. Набор сервисов для получения списка карт

Подобный прием построения архитектуры поможет избежать описанные выше проблемы, поскольку логика разбита между тремя сервисами, и каждый из них выполняет конкретные и понятные функции. Кроме того, при изменении логики работы одного сервиса логика другого сервиса не будет изменена случайным образом, что уменьшает количество тестов и ускоряет выпуск обновлений сервисов.

Заключение

В результате исследования был проведен анализ текущей архитектуры АБС, взаимодействия всех ее подсистем, а также связь с внешними системами. Спроектирована новая целевая архитектура АБС основе КСШ. Идентифицированы новые подсистемы, которые должна включать в себя корпоративная сервисная шина.

Внедрение предложенной в работе банковской корпоративной сервисной шины в АБС позволит:

- 1) Перейти на микросервисную архитектуру [3], которая позволяет избежать дублирования функционала в различных подсистемах АБС;
- 2) предоставить единую точку интеграции банка с различными видами систем, что позволит увеличить количество способов предоставления банковских услуг;
- 3) минимизировать риски возникновения сбоев.

Список используемой литературы

1. Бретт Кинг. Банк 4.0: Новая финансовая реальность /Кинг Бретт. — М.: Олимп-Бизнес, 2020. - 164 с.
2. Биберштейн Н., Боуз С. Компас в мире сервис-ориентированной архитектуры (SOA)/ Н. Биберштейн Н., С. Боуз. – М.: Кудиц-Пресс, 2007.-256 с.
3. Ньюмен С. Н93 Создание микросервисов / С. Ньюмен. — СПб.: Питер, 2016. — 304 с.: ил. — (Серия «Бестселлеры O'Reilly»).
4. Шаппелл Д.А. ESB - Сервисная Шина Предприятия / Дэвид А. Шаппелл. – СПб.: БХВ-Петербург, 2008.-370 с.
5. Гради Буч, Джеймс Рамбо, Ивар Якобсон. Введение в UML от создателей языка. 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2010. – 496 с.
6. Фаулер Мартин. UML. Основы. Краткое руководство по стандартному языку объектного моделирования / Мартин Фаулер. - М.: Символ-плюс, 2011. - 192 с.
7. Мусина, И. Р. Проектирование программных средств системы поддержки принятия управленческого решения в условиях неопределенностей / И. Р. Мусина, Т. Н. Хоменко // Известия Кыргызского государственного технического университета им. И. Раззакова. – 2014. – № 32-1. – С. 40-46.