

ТРАНЗАКЦИОННЫЕ СИСТЕМЫ В СФЕРЕ ФИНАНСОВЫХ ТЕХНОЛОГИЙ

Кохан А.Д.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Научный руководитель: Василькова А.Н. – ст. преподаватель кафедры ИПиЭ

Аннотация. В статье рассмотрена роль транзакционных систем в работе современных финансовых учреждений, описаны компоненты таких систем и принципы их функционирования и взаимодействия друг с другом и с внешними системами.

Ключевые слова: финансовые технологии, транзакционная система, Java, база данных, брокер сообщений, сервер приложений.

Введение. Стремительное развитие финансовых технологий является ключевым двигателем цифровой трансформации мировой экономики, и его устойчивый рост в условиях цифровизации, глобальной конкуренции и повышенных требований к безопасности напрямую зависит от активного внедрения и интеграции новейших технологий. Развитие этой сферы оказывает глубокое и многогранное положительное влияние на мир, выходящее далеко за рамки просто удобств клиентов банков. Это стало возможным благодаря развитию автоматизированных банковских систем, которые на данный момент являются основой рабочего процесса практически всех финансовых учреждений, что признают такие большие исследовательские компании как Gartner и IBM [1].

Основная часть. Gartner определяет основную банковскую систему как внутреннюю автоматизированную систему, которая обрабатывает ежедневные банковские операции и производит обновление счетов и других финансовых записей [2]. Основные банковские системы обычно включают в себя возможности обработки депозитов, займов и кредитов, с возможностью подключения к стандартным системам бухгалтерского учёта и инструментам отчетности.

В настоящее время многие крупные финансовые учреждения могут самостоятельно разработать свою систему и не прибегать к решениям других компаний-разработчиков. В некоторых случаях финансовое учреждение может купить лицензию на уже готовый продукт, такие решения называются коробочный программный продукт. Это значит, что техническим специалистом учреждения придётся самостоятельно интегрировать продукт в свой процессинг и произвести некоторые настройки, прежде чем он начнёт выполнять свою функцию. Он поставляется на условиях «как есть», со стандартными для всех покупателей функциями [3]. В отличие от заказного программного продукта, само появление которого обусловлено требованием конкретного заказчика. И в отличие от проектного программного продукта, продажа которого может, по требованию заказчика, сопровождаться проектной доработкой или разработкой функций, дополняющих стандартные возможности. Распространённым примером таких продуктов являются Oracle FLEXCUBE, SAP Transactional Banking [4] и Temenos Transact, на примере которого далее в этой статье будет подробнее рассмотрено устройство подобных систем.

Комплексно архитектуры систем обычно характеризуют как монолитную или микросервисную. Монолитная архитектура – это традиционная модель программного обеспечения, которая представляет собой единый модуль, работающий автономно и независимо от других приложений. Микросервисная архитектура представляет собой метод организации архитектуры, основанный на ряде независимо развёртываемых служб. У этих служб есть собственная бизнес-логика и база данных с конкретной целью [5].

Рассматриваемую в статье систему Temenos Transact правильнее было бы характеризовать как монолитную, но с оговоркой на то, что это модульный монолит. Модульный монолит – это архитектурный подход, который сочетает в себе аспекты как монолитной, так и модульной парадигм проектирования [6]. Вместо того чтобы разбивать приложение на отдельные микросервисы или распределенные компоненты, что может привести к усложнению процесса развертывания и затратам на связь, модульный монолит организует кодовую базу в виде четко определенных модулей в рамках одного приложения. Эти модули обычно организованы на основе бизнес-возможностей или принципов проектирования, основанных на предметной области. Далее будет рассмотрено как это реализовано в системе.

Автоматизированная банковская система работает на сервере, расположенном либо в центре обработки данных финансового учреждения, либо предоставляемого, по принципу IaaS (инфраструктура как услуга), другой организацией, например Amazon Web Services [7]. Такой сервер может работать под управлением операционной системы Windows Server или множества серверных дистрибутивов Linux, например RedHat или Oracle [8]. При необходимости можно использовать и десктопные версии операционных систем, что запустить систему локально на рабочем компьютере или ноутбуке. Это может потребоваться в процессе разработки, т.к. содержать отдельные сервера может быть экономически невыгодно.

Система Temenos Transact традиционно использует серверы приложений для функционирования. Сервер приложений – это тип сервера, на котором размещается программное обеспечение, используемое для исполнения бизнес-логики. В системах, связанных с финансовыми услугами или мониторингом, код системы содержит бизнес-логику, которая позволяет ей исполнять свои функции. Бизнес-логика – это логика, применяемая для того, чтобы помочь системе выполнять критически важные операции, такие как транзакции или вычисления, на основе бизнес-правил. Обычно она определяет вычислительные задачи и порядок их выполнения [9]. Примерами таких веб-серверов являются IBM WebSphere и RedHat JBoss. Систему Temenos Transact традиционно разворачивают с помощью RedHat JBoss Enterprise Application Platform, который способен исполнять Java код, добавленный к нему с помощью jar, war и ear архивов. JBoss может применяться как на Windows, так и на Linux за счёт дублирования скриптов в bash и bat формат и за счёт использования виртуальной машины Java.

В качестве базы данных для хранения сущностей системы может использоваться любая реляционная система управления базами данных, например Oracle Database, Microsoft SQL Server или PostgreSQL. В базе данных системы присутствует более 60 тысяч таблиц и более 40 тысяч представлений. Такое большое количество данных обусловлено не только сложностью вычислений финансовых показателей, но и вызвано необходимостью распределения данных одной сущности по нескольким физическим таблицам для облегчения работы с данными и обеспечения возможности ленивой загрузки. Ленивая загрузка – это способ извлечения данных из базы при котором полный набор данных извлекается лишь при необходимости, предоставляя базово лишь часть данных или только идентификатор записи [10]. Также причинами большого количества данных в некоторых случаях становятся ошибки разработчиков системы, например, некоторые сущности дублируются в разных таблицах, какие-то таблицы перестают применяться из-за устаревания механизмов, которые их использовали, но при этом не удаляются из системы, а продолжают накапливаться [11]. Когда продукт долго разрабатывается большой командой, возникают архитектурные ошибки, которые приводят к негативным последствиям в виде увеличенного объёма системы и понижения производительности.

Для обмена сообщениями между исполняемыми модулями системы, которыми, в данном примере, являются java архивы используются брокеры сообщений. Брокер сообщений – это программное обеспечение, которое позволяет приложениям, системам и службам взаимодействовать и обмениваться информацией, даже если они написаны на разных языках или реализованы на разных платформах, путем перевода сообщений между официальными

протоколами обмена сообщениями [12]. Temenos Transact использует Apache ActiveMQ или Artemis для обмена сообщениями через очереди и по принципу темы и подписчика. Однако существует возможность при необходимости подключить сторонний брокер, который совершенно не обязательно должен работать внутри JBoss. Так, например, некоторые микросервисы, которые компания Temenos разработала, в последнее время требуют разворачивания брокера сообщений Apache Kafka.

Заключение. Для обработки данных в финансовых учреждениях используются различные программное обеспечение, которое было разработано по проекту для конкретной организации или разработанной для продажи по лицензии одной из технологических компаний. Такое программное обеспечение называется автоматизированной банковской системой. В качестве примера, в этой статье, была разобрана транзакционная система Temenos Transact и набор технологий, который используется в совокупности с ней. За исключением систем управления базой данных, которая может использовать любой другой набор технологий, почти все компоненты экосистемы Temenos Transact работают с применением технологий Java. Причиной тому является баланс между небольшой сложностью разработки, надёжностью, безопасностью и высокой производительностью набора технологий Java. За 30 лет своего существования Java зарекомендовала себя надёжным решением, к которому существует множество решений, что не только упрощает разработку, но и обеспечивает простоту интеграции с любыми прочими технологиями.

Список литературы

1. What is core banking? // IBM [Электронный ресурс]. Режим доступа: <https://www.ibm.com/think/topics/core-banking>. Дата доступа: 30.01.2026
2. Core Banking System // Gartner [Электронный ресурс]. Режим доступа: <https://www.gartner.com/en/information-technology/glossary/core-banking-systems>. Дата доступа: 30.01.2026
3. King, B. Bank 4.0: Banking Everywhere, Never at a Bank / Brett King // Wiley. – 2018. – Vol. 1. – P. 18.
4. Автоматизированные банковские системы // SOWARE [Электронный ресурс]. Режим доступа: <https://soware.ru/products/temenos-transact/alternatives>. Дата доступа: 01.02.2026
5. Ричардсон, К. Микросервисы. Паттерны разработки и рефакторинга / Крис Ричардсон // Пупер. – 2021. – Vol. 1. – P. 52.
6. What Is a Modular Monolith? // Geeks for Geeks [Электронный ресурс]. Режим доступа: <https://www.geeksforgeeks.org/system-design/what-is-a-modular-monolith/>. Дата доступа: 01.02.2026
7. What are the types of cloud computing? // AWS Amazon [Электронный ресурс]. Режим доступа: <https://aws.amazon.com/types-of-cloud-computing/>. Дата доступа: 02.02.2026
8. Ванденбринк, Р. Linux для сетевых инженеров / Роб Ванденбринк // Пупер. – 2023. – Vol. 1. – P. 31.
9. What is an application server? // IBM [Электронный ресурс]. Режим доступа: <https://www.ibm.com/think/topics/application-server>. Дата доступа: 02.02.2026
10. Lazy Loading в Java // Хабр [Электронный ресурс]. Режим доступа: <https://habr.com/ru/companies/otus/articles/819357/>. Дата доступа: 06.02.2026
11. Кэмпбелл, Л. Базы данных. Инжиниринг надёжности / Лейн Кэмпбелл, Черити Мейджорс // Пупер. – 2021. – Vol. 1. – P. 278.
12. What is a message broker? // IBM [Электронный ресурс]. Режим доступа: <https://www.ibm.com/think/topics/message-brokers>. Дата доступа: 06.02.2026

UDC 334.024

ONLINE TRANSACTION PROCESSING IN MODERN FINANCIAL TECHNOLOGY

Kokhan A.D.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vasilkova A.N. – senior lecturer of department of Engineering Psychology and Ergonomics

Annotation. Current article examines the role of transactional systems in the workflow of modern financial institutions, describing the components of such systems and the principles of their functioning and interaction with each other and with external systems.

Keywords. financial technologies, online transaction processing, application programming interface, Java, database, message broker, application server.