

**Архитектура  
АБС «Профиль»**

---

## Содержание

Архитектура АБС «Профиль» .....	1
Содержание .....	2
1 Общие сведения .....	3
2 Термины и сокращения .....	3
3 Архитектурные принципы и задачи .....	4
3.1 Многоканальная архитектура .....	4
3.2 Сервис-ориентированная архитектура .....	4
3.3 Гибкие графики обработки транзакций .....	5
3.4 Непрерывная обработка транзакций в режиме реального времени .....	5
3.5 Отчетность и аналитика в режиме реального времени .....	5
4 Компоненты .....	6
4.1 Клиентский уровень .....	6
4.2 Сети .....	7
4.3 Система обмена сообщениями .....	7
4.4 Серверы приложений .....	7
4.5 База данных .....	7
4.6 Стандартные методы и форматы обмена сообщениями .....	7
4.7 Менеджер транспорта сообщений .....	8
4.8 Потоки обработки .....	8
4.9 Открытый банкинг / внутренние сервисы .....	9
4.9.1 Хранимые процедуры управления сетью .....	9
4.9.2 Хранимые процедуры транзакционного сервиса .....	9
4.9.3 Удаленные вызовы процедур .....	9
4.9.4 SQL (язык запросов) .....	9
4.9.5 HTML (язык гипертекстовой разметки) .....	9
4.10 Встроенный язык программирования .....	10
4.11 Архитектура сервера локальной сети (LAN) .....	10
4.12 Пользовательские интерфейсы .....	10
4.13 Менеджер операционной системы .....	11
5 Список изменений .....	12

## 1 Общие сведения

Данный документ определяет состав структурных элементов и их интерфейсов, формирующих архитектуру Автоматизированной банковской системы «Профиль», далее АБС «Профиль».

## 2 Термины и сокращения

Таблица 1 – Список терминов и сокращений

Термины и сокращения	Определения
API	Application Programming Interface — программный интерфейс приложения. Описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.
IFX	Interactive Financial Exchange - стандарт (спецификация) интерактивного обмена финансовой информацией, основанный на XML
JDBC	Java DataBase Connectivity - платформенно независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД, реализованный в виде пакета java.sql, входящего в состав Java SE
LAN	Local Area Network – локальная вычислительная сеть
LMS	Логическая многосайтовая конфигурация (Logical Multi-Site) - все экземпляры в конфигурации LMS логически эквивалентны в том смысле, что в условиях покоя все экземпляры имеют не только одно и то же состояние базы данных, но и один и тот же путь в пространстве состояний
MIS	Management Information Systems - информационная система, используемая для принятия решений, а также для координации, контроля, анализа и визуализации информации в организации
ODBC	Open Database Connectivity - программный интерфейс доступа к базам данных на основе спецификаций Call Level Interface
RPC	Remote procedure call - класс технологий, позволяющих программам вызывать функции или процедуры в другом адресном пространстве
SMP	Symmetric multiprocessing - архитектура многопроцессорных компьютеров, в которой два или более одинаковых процессора сравнимой производительности подключаются единообразно к общей памяти и выполняют одни и те же функции
SOA	Service-oriented architecture (сервис-ориентированная архитектура) - подход к разработке программного обеспечения, базирующийся на обеспечении удаленного по стандартизированным протоколам использования распределённых, слабо связанных легко заменяемых компонентов (сервисов) со стандартизированными интерфейсами.
SOAP	Simple Object Access Protocol - протокол обмена структурированными сообщениями в распределённой вычислительной среде для реализации удалённого вызова процедур и обмена сообщениями в формате XML
TCP/IP	Transmission Control Protocol/Internet Protocol - сетевая модель, описывающая процесс передачи цифровых данных
WAN	Wide Area Network - глобальная вычислительная сеть
WSDL	Web Services Definition Language - язык описания веб-сервисов и доступа к ним, основанный на языке XML

Термины и сокращения	Определения
Главная книга/ГК	Основной сводный регистр бухгалтерского учета, в котором отражаются данные за отчетный год по всем счетам бухгалтерского учета, применяемым кредитной организацией

### 3 Архитектурные принципы и задачи

АБС «Профиль» является профильной системой для работы финансовых организаций, включающей как функциональный, так и технический уровень, обеспечивающий изоляцию приложения от технологических изменений.

Основой для построения архитектуры системы является обеспечение:

- Надежности - достигается благодаря аппаратным средствам высокой доступности, журнализации изменений базы данных и обработки транзакций;
- Скорости работы - обеспечение высокой производительности при обработке транзакций и использование пакетной обработки;
- Непрерывности - обеспечение актуального состояния базы данных клиентов / счетов в режиме реального времени за счет репликации изменений в нескольких независимых узлах, гарантируя, что в случае сбоя можно немедленно перенаправить систему на локальное или удаленное место резервного копирования, которое содержит самое последнее, актуальное состояние базы данных;
- Устойчивости - стабильность работы приложения опирается на использование резервных экземпляров системы и многосайтовую архитектуру для обеспечения доступности системы даже в случае сбоя или во время плановых обновлений;
- Адаптивности - позволяет дорабатывать и настраивать систему под потребности клиента и изменяющиеся условия рынка, регулярные поставки обновлений обеспечивают соответствие требованиям регулятора;
- Масштабируемости - обеспечивается за счет поддержки нескольких платформ, сервис-ориентированной архитектуры и высокой пропускной способности транзакций.

#### 3.1 Многоканальная архитектура

Внутренняя логика обработки транзакций (включая авторизацию, обновление счетов и отражение в Главной книге) является общей для всего приложения, независимо от источника транзакции, типа интерфейса или классификации продукта.

Независимая от каналов архитектура системы упрощает добавление и поддержку каналов, автоматически наследуя существующие в системе процессы авторизации, обновления, балансировки и расчетов. Это также делает возможной одновременную обработку транзакций из нескольких каналов, включая пакетную и онлайн обработку.

#### 3.2 Сервис-ориентированная архитектура

**АБС** «Профиль» - это система с поддержкой сервис-ориентированной архитектуры (SOA), имеющаяся стандартная библиотека сервисов предоставляет широкий перечень банковских функций, что позволяет любому из каналов (в т.ч. онлайн-банкингу, мобильному банкингу, колл-центру или операционному офису) подключаться через общий набор бизнес-сервисов.

Система основана на обмене сообщениями стандарта IFX (Interactive Financial eXchange), использует протокол SOAP и язык WSDL (Web Services Definition Language). Кроме того, на уровне сервисов используются стандартные серверы приложений, включая WebSphere, WebLogic или JBoss.

---

### **3.3 Гибкие графики обработки транзакций**

Благодаря этому система поддерживает широкий набор стандартных периодов с очень гибким синтаксисом, а также неограниченное количество определяемых пользователем бизнес-календарей и графиков.

---

### **3.4 Непрерывная обработка транзакций в режиме реального времени**

Система реализует архитектуру Logical Multi-Site, которая поддерживает несколько копий базы данных в географически разнесенных местах. Система может практически мгновенно переключиться на удаленный сайт, восстанавливаясь в точке последней транзакции базы данных. Для любого производственного узла может быть назначено до 16 независимых систем резервного копирования, и каждая из них может иметь еще 16 резервных копий. Logical Multi-Site поддерживает смешанные конфигурации, включая удаленные машины на различных аппаратных платформах, различных операционных системах и даже разных версий приложений самой системы. Эта архитектура позволяет обеспечивать непрерывную доступность системы, даже в случае планового технического обслуживания или обновления операционной системы или приложения.

---

### **3.5 Отчетность и аналитика в режиме реального времени**

Благодаря внедрению высокопроизводительной нереляционной базы данных, система поддерживает формирование аналитической отчетности для оценки эффективности бизнес-операций (MIS), обработку специальных запросов. Процессы формирования отчетов могут запускаться непосредственно на основной производственной системе или на любой из реплицированных копий в режиме реального времени, не влияя на производительность основной системы.

---

## 4 Компоненты

На следующей схеме показана взаимосвязь между базой данных, приложением, сервисными модулями, серверами приложений, системой обмена сообщениями, сетью и внешними устройствами (то есть клиентами).

Схема 1 – Компоненты системы



---

### 4.1 Клиентский уровень

Различные типы клиентов могут использовать предоставляемые услуги, устанавливая соединения и обмениваясь сообщениями с соответствующими сервисными модулями. Клиентами могут выступать (включая, но не ограничиваясь):

- приложения колл-центра;
- клиенты внутрибанковской системы;
- клиенты ODBC (Open Database Connectivity) /JDBC (Java Database Connectivity);
- сторонние системы автоматизации филиалов;
- банкоматы;
- точки самообслуживания.

---

## 4.2 Сети

Система использует набор протоколов TCP/IP для подключения к глобальной сети (WAN).

---

## 4.3 Система обмена сообщениями

Система обмена сообщениями обеспечивает связь между сетью и сервером приложения. Сообщения представляют собой атомарные транзакции приложения, которые буферизируются и передаются к доступным серверам по принципу FIFO. Компонент транспортировки сообщений отвечает за прием сообщений от клиента, их буферизацию по мере необходимости, маршрутизацию их к доступному серверу, и, в конечном итоге, возвращает ответ от сервера клиенту. Система поддерживает свою собственную систему доставки сообщений (менеджер транспорта сообщений), а также интерфейсы с MQSeries от IBM.

---

## 4.4 Серверы приложений

Многopotочность достигается за счет одновременно активных серверов. Каждый экземпляр сервера автоматически обрабатывает сообщение. Хотя большинство сообщений являются атомарными, если требуется учитывать контекст, он сохраняется таким образом, что сервер, обрабатывающий продолжение сообщения, может восстановить нужный контекст. Количество одновременно активных серверов настраивается в зависимости от объема транзакций. Дополнительные возможности включают защиту транзакций и откат/восстановление.

---

## 4.5 База данных

Система использует функциональность нереляционной базы данных и представлена обширным системным репозиторием, управляемым DATA-QWIK. Этот репозиторий содержит все метаданные, описывающие компоненты приложения, а также схему компонентов.

База данных поддерживает высокоскоростную обработку транзакций, используя разреженные массивы и разделители столбцов для оптимизации хранения и сохранения гибкости. Столбцы могут расширяться до 32K, а отдельные таблицы могут расширяться до аппаратных ограничений.

Для обеспечения надежности системы используется журналирование: создается один или несколько файлов, которые собирают дополнительную информацию об обновлениях базы данных в соответствии с критериями, заданными администратором системы, обеспечивая средства восстановления или замены баз данных, в которых возникли проблемы с целостностью.

---

## 4.6 Стандартные методы и форматы обмена сообщениями

Стандартные форматы сообщений определены для каждого из сервисных модулей. Они включают в себя клиентские сообщения запроса и ответные сообщения сервера об успешной обработке / ошибке.

Поскольку не все клиенты соответствуют этим форматам, система предоставляет механизм, позволяющий осуществлять преобразование сообщений на сервере.

Система предоставляет общий, стандартный формат сообщений, используя подход к определению поля "длина-значение" (LV), обеспечивающий поля переменной длины с неограниченным

размером. Типы полей, такие как массивы двоичных данных (blob), поддерживаются в формате LV. Стандартный формат включает в себя описание заголовков для сообщений как клиента, так и сервера.

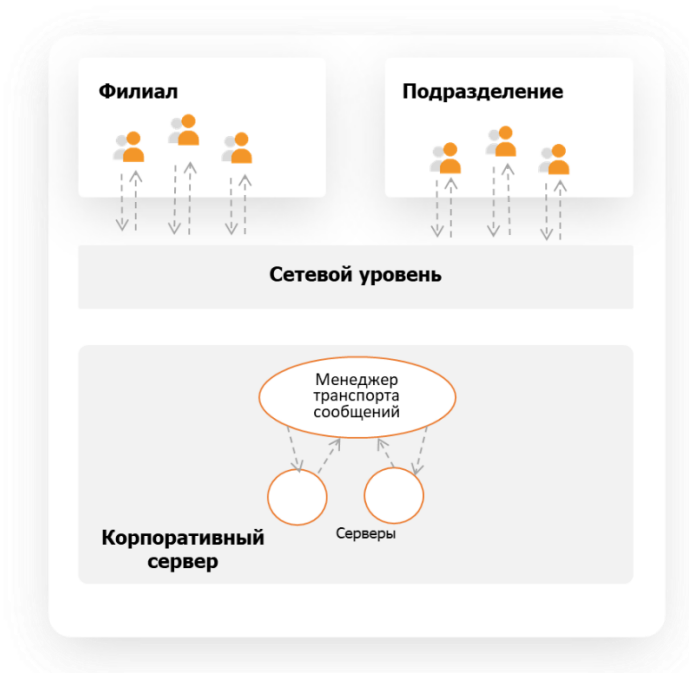
---

## 4.7 Менеджер транспорта сообщений

Система в настоящее время поддерживает различные методы распределенной обработки, которые требуют транспортировки сообщений между процессами. К ним относятся стандартные сообщения для финансовых транзакций, удаленный вызов процедур (RPC), SQL и монитор обработки транзакций (Transaction Processing, TP).

В средах с большим количеством транзакций эти методы обработки приводят к огромному потоку сообщений, в связи с чем возникает потребность отделить задачу управления потоками сообщений между процессами от задачи обработки этих сообщений серверами баз данных. Слой программного обеспечения, разработанный для выполнения этой задачи, называется Менеджер транспорта сообщений (МТС). МТС маршрутизирует входящий и исходящий поток сообщений между клиентами системы и серверами.

Схема 2 – Менеджер транспорта сообщений



---

## 4.8 Потоки обработки

Разделение операций на потоки позволяет лучше использовать общие вычислительные ресурсы, особенно на многопроцессорных машинах с общей памятью (SMP), и позволяет увеличить скорость их обработки.

Система имеет современный диспетчер задач, который поддерживает многопоточную обработку (в том числе для процессов пакетной обработки операций). Система поддерживает конфигурацию SMP и выполняет параллельную обработку задач в рамках одной физической машины. Благодаря использованию стандартной архитектуры, система может разделять большие пакеты на несколько одновременно обрабатываемых потоков.



---

## 4.9 Открытый банкинг / внутренние сервисы

Система отвечает на запросы баз данных и осуществляет функциональную обработку запросов от других приложений с помощью набора программных интерфейсов (API). Доступные сервисы Системы описаны ниже.

---

### 4.9.1 Хранимые процедуры управления сетью

Процедуры управления сетью контролируют доступ к системе, осуществляют контроль входа и выхода клиентов из системы, а также логирование событий пользователей в системе.

---

### 4.9.2 Хранимые процедуры транзакционного сервиса

Службы данного класса позволяют клиентам выполнять операции с профильными счетами: обрабатывают инициированные клиентом запросы на дебетовые и кредитные транзакции по финансовым счетам в системе.

---

### 4.9.3 Удаленные вызовы процедур

Службы данного класса предоставляет клиенту механизм для запуска публичного процедурного кода М на сервере Системы. Клиентские приложения могут запускать процедуры для выполнения функций программного уровня, которые не могут быть выполнены на клиенте. Сюда входят такие процедуры, как проведение корректирующих проводок или получение сложных процентных ставок.

---

### 4.9.4 SQL (язык запросов)

Процедуры данного класса предоставляют средства для взаимодействия с базой данных Системы используя обширное множество стандартных операторов языка запросов SQL. Обеспечивается основная функциональность SQL, включая такие команды, как INSERT, UPDATE, SELECT и DELETE. А также включены дополнительные специфические для Системы расширения и квалификаторы для обеспечения более широкого функционала для взаимодействия с базой данных.

Словарь DATA-QWIK управляет доступом к базе данных системы: наименования таблиц, включая все проверки внешних ключей, триггеры и правила заполнения, а также обновления внешних индексов.

---

### 4.9.5 HTML (язык гипертекстовой разметки)

Сервисы класса HTML обеспечивают доступ клиента к функциональности Системы через веб-браузер и отвечает за сопоставление полей в базе данных и полей на клиенте. Сервисы преобразуют исходный HTML-код в выполняемую процедуру, включающую SQL для доступа к данным. HTML-страница каталогизируется, сохраняя имя процедуры, на которую можно ссылаться при следующем обращении, а также переменные, поименованные в форме для возможности их использования в качестве локальных переменных в системе.

---

## 4.10 Встроенный язык программирования

Встроенный язык программирования - это высокоуровневый язык генерации кода со встроенной абстрактной схемой базы данных. Среда разработки построена на объектно-ориентированном подходе (используются классы, методы, определение свойств объектов класса, а также инкапсуляция данных, наследование и полиморфизм).

Встроенный язык программирования позволяет разработчикам создавать модульный код, связанный с объектами с определенными свойствами и операциями, которые эти объекты могут выполнять. Объектно-ориентированное программирование фокусируется на проектировании данных (т.е. объектов), поэтому такой подход позволяет разработчикам сконцентрироваться на конечном продукте или приложении, а не на инструментах, используемых для выполнения задачи.

В дополнение к преимуществам, присущим объектно-ориентированному подходу, среда программирования обеспечивает следующие преимущества:

- множество экземпляров: например, можно одновременно открыть несколько депозитных счетов при условии, что вы однозначно идентифицируете объекты, определяющие эти счета,
- обратная совместимость с существующим исходным кодом, при соблюдении синтаксиса языка,
- собственная библиотека запуска кода, обеспечивающая независимость от платформы,
- автоматизированное выделение и очистка памяти,
- выявление большинства ошибок компиляции до выполнения программы,
- легкая адаптация к изменяющимся средам (т.е. разработчики могут легко добавлять новые методы и переменные экземпляра без какого-либо влияния на клиентское приложение).

---

## 4.11 Архитектура сервера локальной сети (LAN)

Сервер LAN содержит распределенную базу данных. Этот сервер управляет распределением таблиц и промежуточным хранением. Функциональность позволяет выполнять автономные операции в локальной сети с промежуточными транзакциями, лимитами и локальными подмножествами центральной базы данных.

Прикладное программное обеспечение, расположенное в локальной сети, использует услуги открытой банковской архитектуры Системы на сервере компании. Другие системы локальной сети могут быть интегрированы с Системой по той же методологии.

---

## 4.12 Пользовательские интерфейсы

UI Системы состоит из браузерной версии, толстого клиента, веб-приложений и сервисов, которые полностью обеспечивают банковский сервис.

Пользовательский интерфейс Системы построен на базе промышленных стандартов и технологии с открытым исходным кодом на платформе Java Enterprise Edition (Java EE).

Система использует трехуровневую архитектуру с веб-браузером, сервером приложений и базой данных. Архитектура и дизайн основаны на масштабируемых технологиях с открытым исходным кодом, таких как комбинированный фреймворк Tiles/Struts/Spring Java.

Apache Struts предоставляет расширяемую среду разработки с открытым исходным кодом. Ядром Struts является гибкий слой управления, основанный на стандартных технологиях, таких как Java Servlets, JavaBeans, ResourceBundles и Extensible Markup Language (XML). Spring Workflow обеспечивает инфраструктуру для реализации процессов навигации в веб-приложениях. Совместимость с другими банковскими системами также обеспечивается благодаря кроссплатформенным и надежным интеграционным возможностям платформы Java EE.

Архитектура веб-решений Системы предоставляет экономически эффективное, масштабируемое и отказоустойчивое решение на основе браузера и XML, которое является простым в настройке и внедрении, и предоставляет банковским клиентам решение, которое обеспечивает гибкость, ценность и окупаемость инвестиций.

---

## 4.13 Менеджер операционной системы

Компании могут конфигурировать Систему в соответствии со спецификой своей бизнес-модели. Логические части объединяются в конкретные конфигурации для удовлетворения следующих бизнес-потребностей:

- отдельные банки: все функции выполняются в одном месте,
- финансовые группы: компания организована как центральный офис с распределенными филиалами и региональными представительствами,
- сервисные компании: организация, которая предоставляет услуги по обработке данных нескольким организациям или группам.

АБС «Профиль» может работать на различных платформах и операционных системах. Применение процедурного языка и баз данных, используемых в Системе, в основном не зависит от платформы. Имеющиеся зависимости решаются с помощью слоя обработки, разработанного для этой цели, тем самым сокращая затраты на обслуживание/ поддержку, предоставляя возможность использование преимуществ каждой операционной системы.

Широкий спектр выбора обеспечивает максимальную гибкость при определении операционной среды и возможность комбинации платформ (например, используя одну платформу для своей основной производственной среды, другую - для создания отчетов в режиме реального времени, а третью - для разработки и тестирования).

Менеджер операционной системы предоставляет:

- полностью скомпилированный код и исполняемые файлы, разработанные для каждой операционной системы;
- использование объектных библиотек и компоновщиков платформы;
- использование принтеров данной платформы, буферизации печати и других средств вывода;
- соответствие стандартам вызова платформы для интеграции с другими языками;
- полностью совместимая реализация со стандартными продуктами в той же операционной среде;
- файлы сценариев для управления окружением, включая запуск, остановку и определение параметров окружения.

---

## 5 Список изменений

Версия	Дата	Внесенные изменения	Исполнитель
1.0	20.02.2023	Создание документа	Александрова А.А.