

**ИМПОРТОНЕЗАВИСИМАЯ АРХИТЕКТУРА КОРПОРАТИВНОЙ
АНАЛИТИЧЕСКОЙ СИСТЕМЫ УПРАВЛЕНЧЕСКОГО УЧЕТА**

Аннотация. В работе предложена архитектура корпоративной аналитической системы управленческого учета, рассчитанная на эксплуатацию без критической зависимости от зарубежных коммерческих платформ и облачных сервисов. В качестве исходного случая рассматривается организация, в которой фактические данные формируются в 1С, сведения о трудозатратах и проектных бюджетах ведутся в Timetta, а прогнозные показатели вводятся руководителями проектов в электронных таблицах. Основная проблема состояла не только в разрозненности источников, но и в отсутствии единого набора правил сопоставления проектов, статей бюджета, подразделений и валют. Для решения задачи сформулированы критерии импортонезависимости: локальное или контролируемое развертывание, открытые интерфейсы обмена, возможность выгрузки данных без участия поставщика и заменяемость отдельных компонентов. Предлагаемый стек включает Astra Linux, Postgres Pro, 1С, Timetta, «МойОфис Частное Облако» и Visiology в локальном варианте развертывания. Интеграционные адаптеры взаимодействуют с центральным приложением через REST API; прямой доступ внешних модулей к базе данных не используется. Описаны правила загрузки, нормализации, версионирования и агрегации данных, а также ограничения подхода. Количественные эффекты внедрения в статье не заявляются, поскольку для их подтверждения требуется отдельное измерение до и после промышленного запуска.

Ключевые слова: управленческий учет; импортонезависимая архитектура; корпоративная аналитика; интеграция данных; Postgres Pro; 1C; Timetta; МойОфис; Visiology.

Abstract. The paper presents an architecture for a corporate management accounting analytics system designed to operate without critical dependence on foreign commercial platforms or mandatory external cloud services. The case is based on an organization where accounting data are generated in 1C, project labor and budget data are maintained in Timetta, and forecast figures are entered by project managers in spreadsheets. The main difficulty is not limited to data fragmentation: the source systems use different identifiers, classifiers, currencies, and update cycles. The paper therefore defines practical criteria of import independence, including controlled deployment, open integration interfaces, independent data export, and component replaceability. The proposed stack combines Russian products and vendor-neutral open-source components: Astra Linux, Postgres Pro, 1C, Timetta, MyOffice Private Cloud, and an on-premises Visiology installation. Integration adapters communicate with the central application through a documented REST API; direct database access from external modules is excluded. The paper describes data loading, validation, mapping, versioning, aggregation, access control, and audit procedures. Quantitative implementation effects are intentionally not reported because they require a separate before-and-after measurement methodology.

Keywords: management accounting; import-independent architecture; corporate analytics; data integration; Postgres Pro; 1C; Timetta; MyOffice; Visiology.

Введение

После 2022 года вопрос замены зарубежных корпоративных платформ перестал быть исключительно задачей ИТ-службы. Для систем учета и аналитики он напрямую связан с непрерывностью управленческих процессов: если организация не может получить обновления, продлить лицензию или восстановить

доступ к облачному сервису, под угрозой оказывается не только техническая инфраструктура, но и регулярное формирование бюджета, прогнозов и управленческой отчетности. Российские исследования и опубликованные кейсы миграции показывают, что наиболее сложной частью таких проектов становится не установка новой платформы, а перенос логики учета, справочников, интеграций и накопленных пользовательских практик [1-3].

В этой связи термин «импортонезависимость» требует более точного определения. Его нельзя сводить к происхождению каждого программного компонента. Современная прикладная система почти неизбежно использует библиотеки и протоколы, созданные международным сообществом. Для корпоративного заказчика существеннее другое: наличие исходного кода или локального поставщика, отсутствие обязательного обращения к зарубежному лицензионному серверу, возможность развернуть решение в контролируемой инфраструктуре и гарантированно выгрузить данные в открытом формате. Именно это понимание используется далее.

Предметом работы является архитектура аналитической системы управленческого учета для проектно-ориентированной организации. В исходной ситуации сведения распределены между бухгалтерской системой, системой учета рабочего времени и проектных бюджетов, а также электронными таблицами. Финансовый отдел сводит данные вручную, устраняет расхождения в названиях проектов и статей, проверяет валюты и периоды, после чего готовит итоговые отчеты. Такой процесс зависит от конкретных сотрудников и плохо масштабируется при росте числа проектов и юридических лиц.

Цель работы состоит в том, чтобы предложить архитектуру, которая сохраняет существующие источники данных, но переносит правила их согласования и консолидации в отдельный управляемый контур. Для достижения цели решаются четыре задачи: формулируются критерии импортонезависимости; уточняются функциональные и нефункциональные требования; обосновывается

выбор технологических компонентов; описываются потоки данных, модель хранения и порядок эксплуатации. Работа не претендует на создание нового метода бухгалтерского учета. Ее результатом является архитектурное решение и набор проектных принципов, которые могут быть применены в аналогичных организациях.

Материалы и методы

Исследование выполнено в форме проектного кейса. Исходными материалами послужили описания действующих процессов управленческого учета, состав используемых систем, примеры отчетных форм и требования финансовых специалистов. Анализ проводился на уровне архитектуры предприятия и не включал обработку персональных данных сотрудников или коммерческих показателей конкретных проектов.

Методика включала последовательное выполнение пяти шагов. Сначала были выделены источники данных и определены владельцы информации. Затем зафиксированы проблемные места ручной консолидации: неодинаковые идентификаторы проектов, разная детализация статей, неполные периоды и повторная загрузка одних и тех же записей. На третьем шаге сформированы критерии выбора программных компонентов. После этого разработана логическая архитектура и определены границы модулей. Завершающим шагом стала проверка архитектуры на типовых сценариях: ежемесячная загрузка факта из 1С, получение трудозатрат из Timetta, прием прогноза из корпоративной таблицы, пересчет отчета и раскрытие агрегированной суммы до первичных записей.

При сравнении вариантов использовались не только функциональные возможности продукта, но и эксплуатационные характеристики: способ развертывания, наличие документированного API, возможность резервного копирования, экспорт данных, поддержка российских операционных систем и наличие поставщика, работающего в российской юрисдикции. Такой подход

позволяет отличить реальную заменяемость компонента от формального наличия отечественного аналога.

Исходная ситуация и требования к системе

В рассматриваемой организации бухгалтерский факт формируется в нескольких базах 1С. Учет времени, загрузки сотрудников и части проектных бюджетов ведется в Timetta. Прогноз по отдельным доходам и расходам вводится руководителями проектов в электронные таблицы. Каждая система решает свою задачу, однако единого справочника проектов и статей управленческого учета нет. Один проект может называться по-разному в 1С и Timetta, а в таблице использовать сокращенное обозначение. Аналогичная проблема возникает с подразделениями и контрагентами.

Ручная консолидация строится вокруг нескольких файлов, поддерживаемых финансовым отделом. В них хранятся таблицы соответствия, курсы валют и формулы план-фактного анализа. При изменении структуры источника часть формул приходится корректировать. Отдельная сложность связана с повторной загрузкой: если данные за прошлый месяц были уточнены, специалист должен определить, какие строки заменить, а какие оставить. Ошибка на этом этапе может привести либо к дублированию суммы, либо к потере корректировки.

Требования были разделены на функциональные и архитектурные. Функциональные требования описывают действия пользователей и правила обработки информации. Архитектурные требования задают ограничения, связанные с технологической независимостью, безопасностью и сопровождением. Основные требования приведены в таблице 1.

Таблица 1. Основные требования к системе

Группа	Требование	Практическое значение
Данные	Единое хранилище факта, плана и прогноза	Отчет строится из согласованного набора данных, а не из нескольких пользовательских файлов.
Интеграция	Обмен через документированные HTTP-интерфейсы или контролируемый импорт файлов	Источник можно заменить без переписывания всей системы.
Качество	Валидация формата, периода, валюты и обязательных аналитик	Некорректная запись отклоняется до включения в отчетность.
Сопоставление	Централизованные правила соответствия проектов и статей	Правила изменяются финансовым администратором и применяются одинаково ко всем загрузкам.
Аудит	Хранение истории загрузок и версий	Можно определить источник показателя и восстановить состояние отчета на выбранную дату.
Аналитика	План-фактный анализ и раскрытие агрегатов до первичных записей	Руководитель видит не только итог, но и состав суммы.
Развертывание	Работа внутри контролируемой инфраструктуры	Нет обязательной зависимости от зарубежного облака или внешнего лицензионного сервера.

Группа	Требование	Практическое значение
Безопасность	Разграничение доступа по подразделениям и ролям	Пользователь получает только те данные, которые нужны для выполнения его функций.

К системе не предъявлялось требование полностью заменить 1С или Timetta. Напротив, архитектура должна использовать данные действующих систем и не дублировать их функции. Центральное приложение не ведет бухгалтерские документы и не подменяет систему управления проектами. Оно отвечает за прием данных, их нормализацию, хранение правил соответствия и подготовку управленческой отчетности.

Критерии импортонезависимости

Для выбора стека был сформирован набор критериев, применяемых к каждому критичному компоненту. Первый критерий - возможность эксплуатации без постоянного доступа к инфраструктуре иностранного поставщика. Компонент может использовать открытые библиотеки, но его запуск, обновление конфигурации и доступ к данным не должны зависеть от зарубежного облачного аккаунта.

Второй критерий - контролируемость данных. Организация должна располагать резервной копией и иметь возможность выгрузить сведения в распространенном формате. Закрытый сервис, из которого нельзя получить полный набор данных и историю изменений, не может считаться приемлемым даже в том случае, если его поставщик зарегистрирован в России.

Третий критерий - заменяемость. Интеграция строится вокруг собственных контрактов центрального API и канонической модели данных. Формат конкретного источника преобразуется внутри адаптера. Если организация меняет

BI-платформу или табличный редактор, правила управленческого учета и структура хранилища сохраняются.

Четвертый критерий - наличие локальной поддержки и документации. В корпоративной эксплуатации важна не только доступность дистрибутива, но и возможность получить обновления безопасности, консультацию по совместимости и исправление ошибок. По этой причине для операционной системы и СУБД предпочтение отдано продуктам с российским правообладателем и официальной поддержкой.

Пятый критерий относится к открытости интеграции. Используются HTTP, REST, OData и файловые форматы, которые можно обработать независимо от конкретного поставщика. При этом открытый протокол сам по себе не гарантирует качественную интеграцию: должны быть определены ограничения частоты запросов, правила повторной загрузки и обработка ошибок.

Выбор технологического стека

Выбор компонентов проводился не по принципу максимального числа функций, а по соответствию конкретным ролям в архитектуре. Итоговый вариант приведен в таблице 2. Он представляет собой сочетание российских продуктов и открытых компонентов, разворачиваемых в контролируемой инфраструктуре.

Таблица 2. Компоненты предлагаемого стека

Назначение	Выбранный компонент	Причина выбора	Ограничение
Операционная система	Astra Linux	Российская ОС, доступна для корпоративного и защищенного развертывания [10].	Требуется предварительная проверка совместимости прикладных пакетов.

Назначение	Выбранный компонент	Причина выбора	Ограничение
СУБД	Postgres Pro	Российский правообладатель, включение в реестр отечественного ПО, поддержка PostgreSQL-совместимого стека [8].	Лицензирование и редакция выбираются с учетом требований к отказоустойчивости.
Учетный источник	1С:Предприятие	Существующая система учета; платформа предоставляет REST-интерфейс на основе OData [4].	Состав доступных объектов зависит от конфигурации и настроек публикации.
Проекты и трудозатраты	Timetta	Российская корпоративная система; API построен на OData v4 [5, 6].	Облачная модель остается внешней зависимостью и требует регламентированной выгрузки данных.
Корпоративные таблицы	МойОфис Частное Облако	Совместная работа с документами внутри инфраструктуры организации; предусмотрен HTTPS API [7].	Для аналитической загрузки нужен утвержденный шаблон таблицы и процедура экспорта.
BI	Visiology On-Prem	Российская BI-платформа с локальным вариантом развертывания [9].	Лицензионные и ресурсные требования необходимо оценить на пилоте.

Назначение	Выбранный компонент	Причина выбора	Ограничение
Прикладной слой	Веб-приложение на открытом стеке	Исходный код и сборка находятся у организации; нет обязательной привязки к внешнему облаку.	Сопровождение ложится на внутреннюю команду или интегратора.

Astra Linux выбрана как базовая операционная система для серверного контура. В рамках статьи не утверждается, что переход на нее автоматически решает вопросы безопасности. Профиль защиты, состав пакетов, порядок обновлений и резервного копирования должны определяться в проектной документации. Однако наличие российского поставщика и официальной документации снижает риск прекращения сопровождения по внешним причинам [10].

Postgres Pro используется для хранения нормализованных данных, справочников и журналов загрузки. Выбор обусловлен совместимостью с PostgreSQL и наличием российской поддержки. При этом переносимость приложения не должна зависеть от специфических расширений одной редакции СУБД. В прикладном коде целесообразно использовать стандартный SQL, а специализированные возможности подключать только там, где их польза подтверждена нагрузочными испытаниями.

Платформа 1С сохраняется как источник регламентированного и части управленческого факта. Платформа предоставляет автоматический REST-интерфейс на основе OData, что позволяет получать данные по HTTP без прямого доступа к внутренним таблицам конфигурации [4]. Это важно для сопровождения: изменения структуры информационной базы не должны приводить к несанкционированным запросам к физическим таблицам.

Timetta в исходном контуре уже используется для учета рабочего времени и экономики проектов. Компания-разработчик позиционирует продукт как российскую корпоративную систему, а API поддерживает чтение и изменение сущностей по OData v4 [5, 6]. Вместе с тем облачный сервис, даже российский, остается внешним элементом. Поэтому архитектура предусматривает регулярное получение полной выгрузки необходимых сущностей и хранение идентификаторов источника. Если доступ к сервису временно отсутствует, ранее загруженные данные остаются доступны для отчетности.

Google Sheets исключен из целевой схемы. Для совместного ввода прогнозов предлагается «МойОфис Частное Облако», развернутое в инфраструктуре организации. Руководители проектов продолжают работать с табличной формой, но документ хранится внутри корпоративного контура. Интеграционный адаптер получает файл через официальный HTTPS API, проверяет его версию и обрабатывает только утвержденные листы и столбцы [7]. Такой вариант менее удобен, чем произвольное чтение пользовательской таблицы, зато делает формат загрузки управляемым.

В качестве аналитической платформы выбран локальный вариант Visiology. Он подключается к подготовленным витринам данных, а не к транзакционным таблицам. Это разграничение позволяет изменять структуру внутреннего хранения без одновременной переделки всех отчетов. Встроенный интерфейс центрального приложения используется для настройки правил и контроля загрузок; интерактивные управленческие панели строятся в BI-системе.

Архитектура системы

Архитектура построена вокруг центрального приложения и набора интеграционных адаптеров. Термин «микросервисная архитектура» здесь сознательно не используется: для рассматриваемого масштаба нет необходимости дробить систему на большое число независимо развертываемых сервисов.

Достаточно модульного приложения с четко отделенными зонами ответственности. При росте нагрузки отдельные адаптеры и обработчики могут быть вынесены в самостоятельные процессы без изменения контрактов.

Центральное приложение включает внешний REST API, модуль авторизации, службу загрузок, модуль сопоставления справочников, расчетный модуль и административный интерфейс. Только центральное приложение имеет право записывать данные в Postgres Pro. Адаптеры 1С, Timetta и МойОфис передают пакеты через API и получают идентификатор операции. Это исключает ситуацию, когда разные модули по-разному трактуют структуру таблиц или обходят общую валидацию.

На уровне развертывания выделяются четыре логические зоны: источники, интеграционный контур, прикладной контур и аналитический контур. Источники остаются владельцами первичных данных. Интеграционный контур отвечает за извлечение и преобразование формата. Прикладной контур хранит каноническую модель и правила учета. Аналитический контур получает только подготовленные витрины. Сетевые соединения между зонами ограничиваются необходимыми направлениями и портами.

Интеграция с 1С

Адаптер 1С обращается к опубликованному OData-интерфейсу. Набор извлекаемых объектов определяется конфигурацией: это могут быть документы, регистры накопления или специально подготовленные представления. Для каждого юридического лица хранится отдельная настройка подключения, но формат пакета, передаваемого в центральную систему, остается единым.

Загрузка выполняется инкрементально. Адаптер запрашивает записи, измененные после времени последней успешной синхронизации, и передает их порциями. Каждая запись сопровождается внешним идентификатором, датой изменения, идентификатором организации и типом объекта. Центральное

приложение не считает дату запроса достаточным признаком уникальности: если запись в 1С была изменена задним числом, новая версия заменяет ранее загруженное состояние, а история изменения сохраняется в журнале.

Для снижения зависимости от структуры типовой или доработанной конфигурации рекомендуется формировать в 1С отдельный интеграционный слой. Он должен отдавать только необходимые поля и не раскрывать внутренние объекты, не относящиеся к управленческой отчетности. Такой подход требует начальной разработки, но упрощает дальнейшие обновления конфигурации.

Интеграция с Timetta

Из Timetta загружаются проекты, сотрудники в обезличенном или минимально необходимом составе, табели, ставки и плановые трудозатраты. API позволяет строить выборки по OData v4 [5]. Однако аналитические запросы не следует выполнять непосредственно из BI-системы: это создает лишнюю нагрузку на внешний сервис и делает отчет зависимым от его доступности. Данные сначала помещаются в центральное хранилище.

При загрузке табелей учитывается статус согласования. Черновые записи могут использоваться для оперативного прогноза, но не должны смешиваться с утвержденным фактом. В канонической модели статус хранится отдельным атрибутом. Отчет либо фильтрует только утвержденные записи, либо явно показывает две величины. Это правило устраняет распространенную ситуацию, когда сумма меняется после закрытия периода без понятного объяснения.

Для защиты от потери связи с внешним сервисом сохраняется журнал синхронизации и контрольная выгрузка ключевых справочников. Архитектура не гарантирует работу всех функций Timetta при недоступности сервиса, но управленческая отчетность по уже полученным данным продолжает формироваться.

Интеграция с корпоративными таблицами

Прогнозные сведения часто удобнее вводить в таблице, чем в специализированной форме. Полный отказ от таблиц в рассматриваемом случае нецелесообразен: руководители проектов работают с небольшим числом показателей и привыкли видеть бюджет по месяцам. Проблема заключалась не в самом табличном интерфейсе, а в отсутствии контроля версий и формата.

В целевой схеме для каждого подразделения создается документ по утвержденному шаблону в «МойОфис Частное Облако». В шаблоне фиксируются идентификатор проекта, статья бюджета, валюта, период, сумма и комментарий. Дополнительные пользовательские колонки допускаются, но не участвуют в автоматической загрузке. Изменение обязательной части шаблона оформляется как новая версия.

Адаптер получает документ через API, сохраняет техническую копию файла, вычисляет контрольную сумму и передает строки в центральное приложение. Если файл уже загружался и не изменился, повторная обработка не выполняется. При ошибке пользователь получает протокол с номером строки и причиной отклонения: неизвестный проект, некорректный период, отсутствующая валюта или сумма в неверном формате. Такая обратная связь важнее попытки автоматически исправить любую ошибку, поскольку финансовый специалист должен понимать, какие данные вошли в отчет.

Нормализация и правила сопоставления

После приема пакет попадает в промежуточную область хранения. На этом этапе исходные значения сохраняются без изменения. Затем выполняется нормализация: даты приводятся к единому часовому поясу и формату, валюты - к кодам справочника, строки очищаются от технических пробелов, а числовые значения проверяются на допустимый диапазон. Запись, не прошедшая обязательную проверку, не включается в расчетную витрину.

Ключевым элементом является справочник соответствий. Он связывает внешние идентификаторы проекта, подразделения и статьи с внутренними сущностями системы. Правило может быть точным, когда известен внешний идентификатор, или условным, когда требуется анализ нескольких полей. Условные правила применяются осторожно: совпадение по части названия удобно на этапе первичной настройки, но в промышленной эксплуатации должно быть заменено явным соответствием.

Изменение правила не должно незаметно менять уже опубликованные отчеты. Поэтому каждое правило имеет период действия и номер версии. При пересчете закрытого периода система использует версию, действовавшую на момент публикации, если финансовый администратор явно не инициировал перерасчет. Это решение усложняет модель, но делает результат воспроизводимым.

Версионирование и идемпотентность загрузок

Для каждой загрузки создается объект «Пакет импорта». Он содержит источник, время запуска, диапазон данных, контрольную сумму, статус и статистику обработки. Входящая запись идентифицируется комбинацией источника, внешнего идентификатора и версии. Повторная передача того же пакета не создает дубликаты. Если внешний источник прислал новую версию записи, предыдущая версия переводится в историческое состояние, а актуальная становится доступной для расчета.

Идемпотентность особенно важна для интеграций по сети. Адаптер может не получить ответ из-за разрыва соединения и повторить запрос, хотя центральная система уже обработала пакет. Идентификатор операции и контрольная сумма позволяют вернуть прежний результат без повторной вставки данных.

Логическая модель данных

В основе модели находятся сущности «Организация», «Подразделение», «Проект», «Статья бюджета», «Период» и «Валюта». Фактические операции хранятся отдельно от плановых и прогнозных значений, поскольку они имеют разное происхождение и правила изменения. Для построения отчетов формируется единая витрина, где тип значения задается признаком «план», «факт» или «прогноз».

Первичная запись содержит сумму в исходной валюте, валютный код, дату операции, источник и ссылку на пакет импорта. Сумма в валюте управленческой отчетности рассчитывается по курсу, действующему согласно учетной политике. Курс не подменяется при каждом открытии отчета: использованное значение сохраняется вместе с результатом пересчета. Это позволяет объяснить расхождение между отчетами, сформированными в разные даты.

Для раскрытия агрегатов используется связь между витриной и первичными записями. Пользователь видит сумму по проекту и статье, а затем может перейти к списку операций, из которых она сформирована. Если часть суммы была введена прогнозом из таблицы, в детализации отображается имя документа и номер строки; для 1С указывается идентификатор объекта источника; для Timetta - идентификатор табеля или бюджетной записи.

Расчетный и аналитический контур

Расчетный модуль формирует витрины по расписанию и по требованию финансового администратора. Витрина не должна содержать сложную бизнес-логику, скрытую в BI-отчете. Основные правила - отбор статусов, преобразование валют, распределение общих затрат и расчет отклонений - выполняются централизованно. BI-платформа отвечает за фильтрацию, визуализацию и навигацию по уже согласованным показателям.

Для руководителя формируются отчеты по выручке, прямым и косвенным затратам, маржинальности, исполнению бюджета и загрузке сотрудников. Набор показателей зависит от учетной политики организации. В статье намеренно не приводится универсальная формула маржинальности: в проектном бизнесе состав прямых затрат и правила распределения накладных расходов могут различаться. Эти правила должны быть описаны в методике управленческого учета и только затем реализованы в системе.

Visiology подключается к отдельной схеме витрин с правами на чтение [9]. Доступ к административным таблицам, журналам и настройкам сопоставления через BI не предоставляется. Для оперативного контроля загрузок используется собственный веб-интерфейс: в нем отображаются время последней синхронизации, число принятых и отклоненных строк, а также список ошибок.

Доступ и безопасность

Модель доступа строится на ролях и организационной области видимости. Финансист подразделения может просматривать и корректировать правила, относящиеся к его подразделению, но не имеет доступа к данным других юридических лиц. Финансовый директор получает консолидированный доступ. Технический администратор управляет подключениями и расписаниями, но не должен изменять финансовые показатели через интерфейс.

Аутентификация интегрируется с корпоративной службой каталогов через стандартный протокол, поддерживаемый инфраструктурой организации. Сервисные учетные записи адаптеров отделяются от пользовательских. Секреты подключений не хранятся в исходном коде или таблицах конфигурации в открытом виде. Все действия, влияющие на правила сопоставления и опубликованные периоды, журналируются.

Резервное копирование охватывает базу данных, конфигурацию приложения и исходные файлы, полученные из МойОфис. Проверка восстановления должна

выполняться регулярно. Наличие резервной копии без периодической процедуры восстановления не считается достаточной мерой.

Развертывание и сопровождение

Для пилотного контура приложение и СУБД могут быть размещены на двух виртуальных машинах. В промышленной эксплуатации рекомендуется отделить базу данных от прикладного сервера, использовать резервный экземпляр, централизованный сбор журналов и мониторинг очередей загрузки. Конкретная схема отказоустойчивости зависит от допустимого времени простоя и объема данных.

Обновление адаптера не должно требовать остановки всей системы. Версии API поддерживаются в течение переходного периода, а изменение контракта сопровождается миграцией. Для каждой интеграции создается набор тестовых данных, позволяющий проверить обработку новых, измененных и ошибочных записей. Эта практика особенно важна при обновлении конфигурации 1С или шаблона корпоративной таблицы.

Оценка решения и ограничения

Предложенная архитектура устраняет несколько исходных проблем. Правила соответствия перестают храниться в отдельных пользовательских файлах; повторная загрузка контролируется на уровне пакетов и внешних идентификаторов; аналитические отчеты получают единый источник данных; зарубежный сервис Google Sheets исключается из целевой схемы. При этом существующие системы 1С и Timetta продолжают выполнять свои основные функции, поэтому проект не требует одновременной замены всего ИТ-ландшафта.

Сильной стороной решения является явная граница между источником и канонической моделью. В опубликованных кейсах крупных внедрений 1С и миграции аналитических платформ также отмечается значение единой модели данных и поэтапного переноса функциональности [3, 11]. В рассматриваемой

архитектуре эта идея реализована через адаптеры и централизованные справочники.

Вместе с тем решение нельзя считать полностью независимым от внешних факторов. Timetta используется как облачный сервис, и его недоступность ограничит получение новых табелей и бюджетов. Риск снижается регулярной синхронизацией и возможностью замены адаптера, но не исчезает. Если требования организации запрещают использование любого внешнего SaaS, потребуется выбор локально развертываемой системы управления проектами.

Вторая группа ограничений связана с открытым прикладным стеком. Наличие исходного кода дает контроль, но одновременно увеличивает ответственность за обновления, тестирование и устранение уязвимостей. Коробочная система управленческого учета может быть выгоднее для организации без собственной команды разработки. На российском рынке существуют продукты, которые объединяют несколько видов учета и отчетности в одной платформе, например Digital Q.Accounting [12]. Выбор между собственной разработкой и готовым решением должен основываться на стоимости жизненного цикла, а не только на стоимости лицензии.

Третье ограничение относится к пользовательскому вводу в таблицах. Утвержденный шаблон снижает число ошибок, но не устраняет их полностью. Чем сложнее форма, тем выше риск, что руководители проектов будут вести параллельные версии или изменять структуру документа. Поэтому таблица должна содержать только те показатели, которые действительно требуют ручного прогноза. Остальные данные следует получать из систем-источников.

В исходном варианте статьи приводились проценты сокращения трудозатрат и ускорения отчетности. Эти значения исключены из переработанной версии, поскольку не сопровождались описанием методики измерения. Для корректной оценки после пилота необходимо зафиксировать базовый период, измерить время

выполнения одинаковых операций, число ручных корректировок и количество ошибок. Только после этого можно делать количественный вывод об эффекте внедрения.

Практическую проверку целесообразно провести на одном юридическом лице и ограниченном наборе отчетов. Критериями успешности пилота могут быть: полная трассировка суммы до источника; отсутствие дубликатов при повторной загрузке; воспроизводимость отчета на закрытую дату; обработка корректировок прошлых периодов; формирование протокола ошибок; восстановление данных из резервной копии. Эти критерии проверяют свойства архитектуры и не зависят от субъективной оценки интерфейса.

Заключение

В работе предложена модульная архитектура корпоративной аналитической системы управленческого учета, ориентированная на снижение критической зависимости от зарубежных коммерческих платформ. Импортнезависимость трактуется как совокупность проверяемых свойств: контролируемое развертывание, доступность данных, открытые интерфейсы, локальная поддержка и заменяемость компонентов. Такое определение позволяет использовать открытые технологии без необоснованного утверждения, что весь программный код имеет отечественное происхождение.

Целевая схема сохраняет 1С и Timetta как действующие источники, заменяет Google Sheets на «МойОфис Частное Облако», использует Postgres Pro для центрального хранения, Astra Linux для серверного контура и Visiology для аналитических панелей. Все внешние адаптеры передают данные через единый API и не имеют прямого доступа к базе. Центральное приложение выполняет валидацию, сопоставление справочников, версионирование и подготовку витрин.

Предложенное решение подходит организациям, где структура управленческого учета существенно отличается от типовых конфигураций и

требуется интеграция нескольких источников. Его внедрение оправдано только при наличии компетенций по сопровождению и формализованной методики учета. Следующим этапом работы должна стать пилотная реализация с измерением трудозатрат и качества данных до и после запуска. Без такого измерения оценка экономического эффекта останется предположением.

Список литературы

1. Панина В. З., Панина Э. К. Импортозамещение систем оперативного и управленческого учета наукоемкого предприятия как фактор технологического суверенитета: кейс нефтесервисной компании «Технологии ОФС» // Управленческий учет. 2025. № 7. С. 184-193.

2. Гудзь С. С., Цуканова О. А. Рынок ВІ-технологий в России: анализ перехода на отечественные решения в контексте импортозамещения // Бизнес-информатика. 2025. Т. 19, № 3. С. 85-100.

3. Синельников М., Ведяков В. Опыт ВТБ по миграции SAP BW/4HANA: что помогло уложиться в сроки и сохранить функциональность // Хабр. 10.12.2025. URL: <https://habr.com/ru/companies/vtb/articles/975144/> (дата обращения: 22.07.2026).

4. 1С. REST-интерфейс платформы «1С:Предприятие». URL: <https://v8.1c.ru/platforma/rest-interfeys/> (дата обращения: 22.07.2026).

5. Timetta. Общие сведения об API. URL: <https://timetta.com/ru/docs/api/main> (дата обращения: 22.07.2026).

6. Timetta. О компании. URL: <https://timetta.com/ru/company> (дата обращения: 22.07.2026).

7. МойОфис. «МойОфис Частное Облако 3». Руководство по работе с API. Версия 3.2. 17.12.2024. URL: https://support.myoffice.ru/upload/iblock/a5b/0gmofnb94hgl3ahe5xtgbn55o3qc69gt/MyOffice_CO_PGS_3.2_API_Guide.pdf (дата обращения: 22.07.2026).

8. Postgres Professional. СУБД Postgres Pro Certified. URL: <https://postgrespro.ru/products/postgrespro/certified> (дата обращения: 22.07.2026).
9. Visiology. Корпоративная платформа управления данными и бизнес-аналитики. URL: <https://ru.visiology.su/> (дата обращения: 22.07.2026).
10. Группа Астра. Российская операционная система Astra Linux. URL: <https://astralinux.ru/> (дата обращения: 22.07.2026).
11. 1С:Консалтинг. В «Газпром Интернэшнл Лимитед» внедрили аналитические витрины данных. 2023. URL: <https://consulting.1c.ru/cases/321893.html> (дата обращения: 22.07.2026).
12. Диасофт. Digital Q.Accounting - централизованный бухгалтерский учет и отчетность для бизнеса. URL: <https://fm.diasoft.ru/platform/q-accounting/> (дата обращения: 22.07.2026).