Акционерное общество
Национальный межбанковский процессинговый центр «Хумо»

Дата: «12» февраля 2026г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Внедрение инструмента токенизации

Наименование:

«Внедрение инструмента токенизации»

Заказчик: АО «Национальный Межбанковский Процессинговый Центр» (НМПЦ)

Контактное лицо: Тоиров А. (Главный специалист отдела по закупкам, +998781132407 / 7733, tender@nmpc.uz)

Согласовано:

Заместитель председателя правления
по ИТ и технологической инфраструктуре _____ Самигуллин Д.Р.

Начальник управления цифровыми
продуктами и проекта _____ Бибик П.М.

Директор ДИБ _____ Гафуров А.А

Разработал:

Ведущий специалист _____ Ташкариев Д.Р.

Начальник отдела архитектурных решений _____ Володикова Е.О.

Техническое задание на проведение анализа рынка

Наименование проекта: Внедрение инструмента токенизации
Класс ИТ-решения: Token Management Provider

Термины и сокращения

Оглавление

1.Общие сведения

1.1. Наименование системы

Система токенизации

1.2. Основания для проведения работ

- Наличие в текущем решении функциональных гэпов, препятствующих устойчивому развитию организации на рынке республики Узбекистан
- Запрос коммерческого блока Организации на проведение анализы рынка и выбор технологического решения

2. Назначение системы и цели проведения работ

2.1. Назначение системы

Предоставление функционала токенизации и детокенизации платежных инструментов в рамках платежей, проводимых в рамках платежной системы HUMO и платежного приложения HumoPay.

2.2. Цели проведения работ

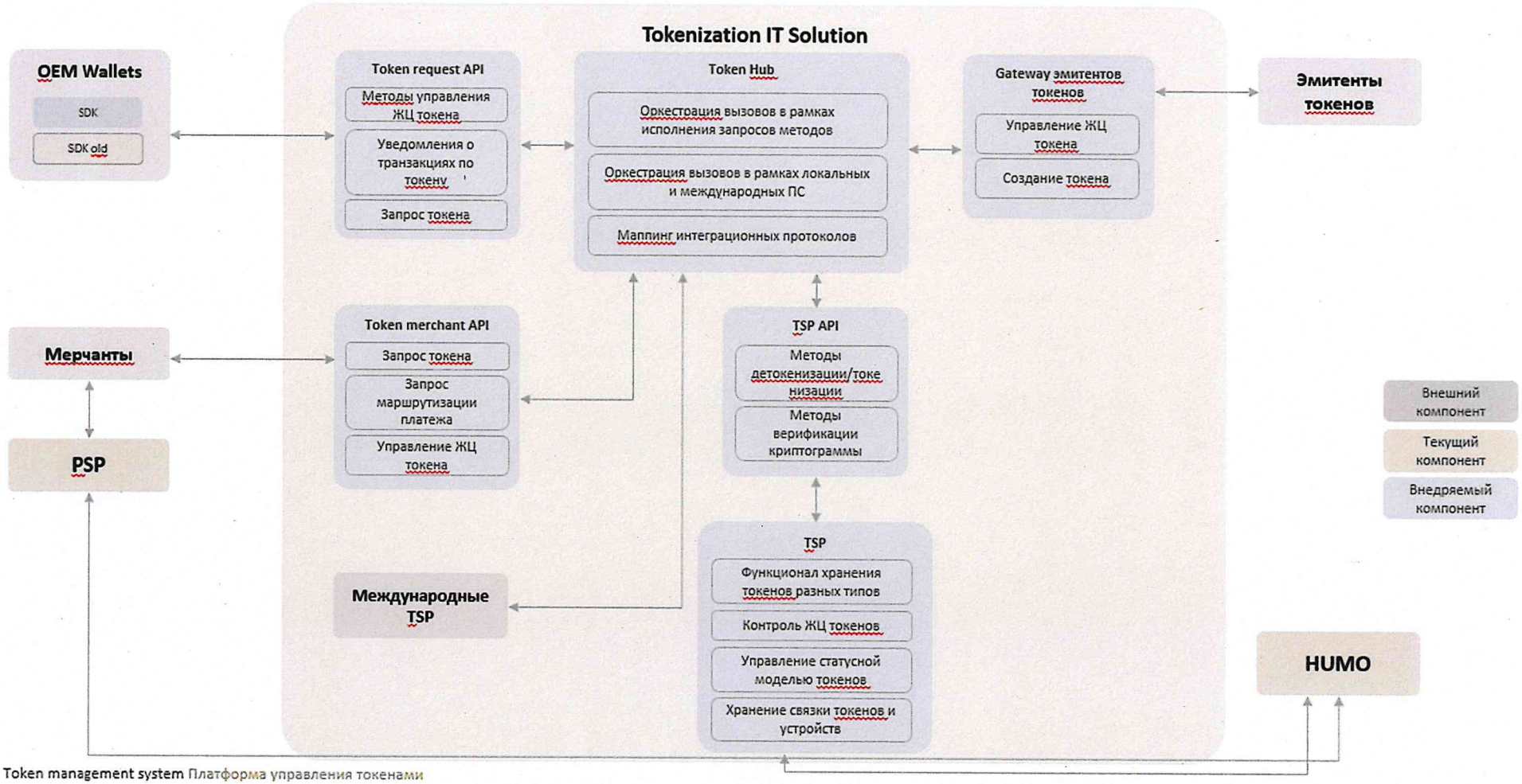
- Внедрение системы управления токенами в рамках функций, предоставляемых АО «Национальный межбанковский процессинговый центр» (HUMO) на рынке Узбекистана
- Доработка функций, предоставляемых текущей TSP

3. Характеристика объектов автоматизации

1. Процесс эмиссии токена платежной карты
2. Процесс эмиссии токена кошелька
3. Процесс управления жизненным циклом токенов
4. Процесс управления токенами из виртуальных платежных кошельков

Количество одновременных запросов на токенизацию/детокенизацию rps-1500/rpm-3000;
Количество запросов на формирование токена rps-500/rpm-1500
Количество активных кошельков – более 12млн;
Глубина хранения данных – 5 лет;
Поддерживаемая ОС Microsoft Windows Server 2022 Standard или Centos/Ubuntu.

4. Схема прикладной архитектуры ИТ-решения



5. Требования к системе

5.1. Общие требования к Системе

1. Система должна быть установлена on-premise (Закон Республики Узбекистан, от 02.07.2019 г. № ЗРУ-547) (см. приложение 1);

2. Система должна быть представлена готовым коробочным решением (не заказная разработка), за исключением потребности в разработке выделенных интеграционных компонент у поставщика решения, позволяющих интегрироваться с действующим ландшафтом АО «Национальный межбанковский процессинговый центр» (HUMO).

3. Система должна поддерживать следующие типы интеграций с ИТ-ландшафтом:
   - Kafka
   - Rabbit MQ
   - REST
   - ISO 8583
   - SOAP
   - TCP интеграция с СУБД

В случае отсутствия возможности поддержать указанные протоколы, вендор должен предоставить готовое промышленное решение в качестве альтернативы текущим типам интеграции (например, собственная интеграционная шина).

4. Система должна иметь возможность модульной поставки (итерационное внедрение в разные промежутки времени независимых функциональных компонент программного обеспечения);
5. Система должна иметь управляемую ролевую модель прав доступа в админ панели Системы;
6. Соответствие техническим требованиям, представленным в данном документе;
7. Соответствие требованиям информационной безопасности, представленным в данном документе;

5.2. Требования к функциям, выполняемым Системой

1. Предоставление API для поддержки интеграции с локальным TSP Организации через безопасный API;
2. Поддержка функционала хаба токенизации для оркестрации вызовов в локальные TSP и международные (Visa, Mastercard, UnionPay, Мир)
3. Предоставление функционала простого масштабирования для подключения к новым типам кошельков без значтельных конфигураций;
4. Поддержка SDK интеграции с кошельками Apple Pay, Google Pay, Samsung Pay, Huawei Pay, MirPay и другими OEM Pay-сервисами.
5. Поддержка push provisioning — добавление карт платежной системы HUMO в кошельки напрямую из мобильного приложения HUMO Pay и мобильных приложений банков Узбекистана.
6. Предоставление методов передачи криптограмм платежным приложениям – держателям виртуальных кошельков.
7. Предоставление методов точечного и массового управления сроком действия токена.
8. Предоставление функционала поддержки совместимости API при применении обновлений на платежных кошельках (ApplePay, SamsungPay), применение необходимых обновлений на стороне менеджера токенов без участия HUMO.

9. Предоставление методов точечного и массового обновления информации по токенам из мобильного кошелька
10. Предоставление методов точечного и массового обновления информации по всем связанным токенам при обновлении данных по карте (срок действия, FPAN) в рамках менеджера токенов.
11. Предоставление методов обновления персональных данных владельца токена при перевыпуске карты/ при проведении действий с токеном (в т.ч. смене статуса).
12. Предоставление функционала ручного и автоматизированного управления статусами токенов;
13. Поддержка жизненного цикла токена: создание, активация, приостановка, удаление, перевыпуск.
14. Поддержка токенов типов: токена платежной карты и токен виртуального кошелька.
15. Предоставление методов запроса результата транзакций по токенам.
16. Функционал получения и сохранения связки токена и девайса в хранилище токенов.
17. Возврат всех видов токенов владельца кошелька.
18. Поддержка моно и кобейдж карт.
19. Функционал приостановления активного токена из приложения (один токен или массовое управление).
20. Функционал точечного и массового возобновления действия приостановленных токенов.
21. Массовое приостановление действия токенов по всем картам держателя по всем банкам.
22. Маскирование персональных данных карты в приложении кошелька.
23. Предоставление маскированных данных платежной карты в приложении при аутентификации.
24. Наличие многоуровневой системы мониторинга и уведомлений о попытках несанкционированного доступа к кошельку.
25. Возможность интеграции с внутренней CRM и системой мониторинга HUMO.
26. Поддержка HCE (Host Card Emulation) и SDK для HUMO Pay-приложения.
27. Возможность отслеживания статистики по активным токенам, транзакциям, отказам, статусам.
28. Поддержка административной панели для управления токенами и правами пользователей.
29. Генерация детализированных отчётов для технических и коммерческих подразделений.
30. Наличие SLA-договорённостей (доступность системы не менее 99.95%).
31. Обеспечение технической поддержки и обновлений на срок не менее 3 лет.

6. Нефункциональные требования

6.1. Технологические требования

1. Возможность самостоятельно развернуть сервисы аутентификации в рамках поставляемого решения, соответствующего требованиям к Информационной безопасности
2. Отсутствие в предлагаемом продукте технологических компонент, требующих покупки дополнительных лицензий и компонентов или явное описание их стоимости в коммерческом предложении
3. ОС Microsoft Windows Server 2022 Standard или Centos/Ubuntu
4. Поддержка развертывания системы на технологической платформах Kubernetes
5. Стек технологий для реляционных СУБД – PostgreSQL/Oracle
6. Стек технологий для нереляционных СУБД – Redis/ClickHouse/MariaDB/MongoDB/Cassandra
7. Рекомендуемый стек технологий для объектных СУБД – Minio (бесплатное решение с открытым исходным кодом) и аналоги
8. Возможность соответствия стандартам и принципам единой интеграционной среды Банка:
   a. возможность публикации/получения доменных событий из шины событий (RabbitMQ, Apache Kafka);
   b. возможность создания контракта API с обратной совместимостью, либо возможность версионирования контракта API;
   c. возможность выполнения синхронных запросов посредством протокола HTTP/S;
   d. формат данных в запросах и ответах – JSON;
   e. доступ к конечным точкам защищен авторизацией;
   f. формат сообщений об ошибках должен соответствовать спецификации JSON.API;
   g. возможность документирования контрактов на основании The OpenAPI Specification (Swagger).
9. При использовании в закупаемом коробочном решении технологий с открытым исходным кодом, требуется удостовериться, что данные технологии экспертно поддержаны развитым сообществом (комьюнити;
10. Развитое сообщество (комьюнити) в рамках поставляемого решения
11. Для опциональных компонентов: поставка компонента отдельным сервисом для возможности переиспользования.
12. Максимальное переиспользование существующих сервисов АО «Национальный межбанковский процессинговый центр» (HUMO) при интеграции с ИТ-ландшафтом;
13. Документация к системе с описанием функциональных возможностей, архитектуры и модели хранения данных с подробным описанием и справочником терминов системы
14. Готовность поддержать параметры RTO/RPO в соответствии с запросом Заказчика
15. Предоставление сайзингов инфраструктуры и архитектуры развертывания

7. Требования к информационной безопасности

Требования к соответствию стандартам ИБ

1. Для соответствия требованиям сертификации и стандартам ИБ программное решение обязано иметь актуальные сертификаты соответствия требованиям регуляторов в каждой стране присутствия, где планируется его использование в финансовом секторе.
2. Решение обязано соответствовать местным требованиям по защите персональных данных, информационной безопасности и специфическим нормативам для финансового сектора: закон «О кибербезопасности» и закон «О защите информации в автоматизированной банковской системе».
3. В качестве доказательства соответствия соответствующим требованиям и стандартам используется отчет или сертификат о соответствии соответствующим требованиям от независимой организации.
4. Если решение работает с персональными данными (ПД), то решение обязано соответствовать требования закона РУз «О персональных данных», ЗРУ-547-сон 02.07.2019 во всех областях применения соответствующего закона.
5. Решение обязано пройти комплексное тестирование на проникновение (анализ защищенности, аудит безопасности) с датой проведения не более полугода от момента внедрения соответствующего решения. Доказательством проведения тестирования является отчет о проведенном тестировании. Отчет должен содержать раздел или приложение к отчету о проверке исправления найденных уязвимостей.

7.1. Документация и процедуры

Качественная документация является краеугольным камнем безопасности программного обеспечения. Без четкого понимания архитектуры системы, ее компонентов и их взаимодействия невозможно обеспечить должный уровень защиты. Техническая документация должна быть живым документом, который постоянно обновляется и отражает текущее состояние системы.

Особое внимание следует уделять руководствам по установке и настройке. Статистика показывает, что значительная часть инцидентов безопасности происходит именно из-за неправильной конфигурации систем. Подробные инструкции и чек-листы помогают минимизировать человеческий фактор и обеспечить безопасность с самого начала эксплуатации системы.

Эксплуатационная документация должна включать не только стандартные процедуры, но и планы действий в нештатных ситуациях.

Требования к документации

Для соответствия требованиям к документации программное обеспечение обязано поставляться с полным комплектом технической документации. Техническая документация должна включать в себя не меньше следующих пунктов:

- Руководство по установке;
- Руководство по настройке;
- Руководство по эксплуатации;
- Архитектура решения, включающая компоненты системы;
- Руководство по ролевой модели и ее настройке;
- Документация по доступным API и интеграциям с внешними системами;

7.2. Управление уязвимостями

Современные подходы к разработке программного обеспечения требуют внедрения процессов непрерывного анализа безопасности на всех этапах жизненного цикла. Статический анализ кода (SAST) позволяет выявлять потенциальные уязвимости еще на этапе написания кода, до того, как они попадут в продуктивную среду. Использование нескольких инструментов SAST обеспечивает более полное покрытие и снижает вероятность пропуска критических уязвимостей.

Динамический анализ (DAST) дополняет статический анализ, позволяя находить уязвимости, которые могут проявляться только во время выполнения программы. Регулярное проведение DAST-сканирования помогает выявлять проблемы, связанные с конфигурацией окружения и взаимодействием компонентов системы.

Независимый security-аудит кода является дополнительным уровнем защиты, позволяющим найти уязвимости, которые могли быть пропущены автоматизированными инструментами. Опытные аудиторы способны выявлять сложные логические ошибки и потенциальные проблемы безопасности, требующие глубокого понимания архитектуры приложения.

7.3. Управление версиями и обновлениями

Поддержание актуальности используемых компонентов и библиотек является критически важным аспектом безопасности. Злоумышленники активно используют известные уязвимости в устаревших версиях программного обеспечения, поэтому своевременное обновление компонентов становится одной из приоритетных задач.

Гарантированная поддержка в течение 5 лет дает пользователям уверенность в том, что они будут получать необходимые обновления безопасности и техническую поддержку на протяжении длительного периода. Это особенно важно для критически важных систем, где процесс миграции на новые версии может быть сложным и требовать значительных ресурсов.

Требования к технической поддержке

Для соответствия требованиям к технической поддержке разработчик решения обязан предоставлять гарантированную техническую поддержку решения в течение не менее 5 лет после внедрения. Техническая поддержка должна оказываться как минимум в случае инцидентов информационной безопасности. Уровень технической поддержки определяется в каждом конкретном случае ответственным лицом со стороны информационной безопасности.

Разработчик решения обязан гарантировать SLA для исправления инцидентов информационной безопасности на следующих уровнях критичности:

- Критический уровень - не более месяца с момента оповещения;
- Высокий уровень - не более 3 месяцев с момента оповещения;
- Средний уровень - не более 6 месяцев с момента оповещения;
- Низкий уровень - не более 12 месяцев с момента оповещения;

Разработчик решения обязан предоставлять отчеты об устранении и исправлении инцидентов информационной безопасности.

7.4. Безопасный цикл разработки (Secure SDLC)

Внедрение практик безопасной разработки

Безопасный цикл разработки программного обеспечения является фундаментальным элементом создания защищенных программных решений. В современных условиях, когда скорость вывода продукта на рынок имеет критическое значение, особенно важно интегрировать практики безопасности непосредственно в процесс разработки, а не добавлять их постфактум. Внедрение принципов "Security by Design" и "Security by Default" должно начинаться с самых ранних этапов проектирования системы.

Ключевым аспектом безопасной разработки является автоматизация процессов безопасности. Современные инструменты позволяют интегрировать проверки безопасности непосредственно в конвейер CI/CD, обеспечивая непрерывный контроль качества кода с точки зрения безопасности. При этом важно настроить эти инструменты таким образом, чтобы они не создавали лишних препятствий для разработки, но эффективно выявляли реальные проблемы безопасности.

Требования к процессу разработки ПО

Для соответствия требованиям разработки безопасного ПО - процесс безопасной разработки должен включать как минимум следующие этапы:

- Разработка требований безопасности и анализ рисков на этапе согласования технического задания;
- Анализ безопасности архитектуры решения на этапе разработки архитектуры проекта;
- Внедрение и обслуживание конвейера безопасной разработки на этапе разработки решения;
- Комплексное тестирование на проникновение или анализ защищенности на этапе развертывания решения;

Разработчик решения обязан проводить и предоставлять отчет о комплексном тестировании на проникновение решения не менее одного раза в год с момента внедрения решения. Отчет должен содержать раздел или приложение к отчету о проверке исправления обнаруженных уязвимостей. В случае отсутствия изменений в решении на уровне кода в течение одного года с последнего тестирования безопасности проведение комплексного тестирования на проникновение решения не требуется.

7.5. Практики безопасного кодирования

Обеспечение безопасности кода требует комплексного подхода, включающего как технические, так и организационные меры. Команды разработки должны следовать установленным стандартам кодирования, которые включают в себя требования по безопасности. Это включает
использование проверенных библиотек и компонентов, правильную обработку ошибок, безопасную работу с данными пользователя и корректную реализацию криптографических механизмов.

Требования к кодированию

Для соответствия требованиям к кодированию команды разработки должны следовать установленным стандартам кодирования, которые включают в себя требования по безопасности.

Разработчик обязан покрыть исходный код решения функциональными тестами на уровне 70% от всего функционала.

Для разработки решения должны использоваться операционные системы и образы контейнеров с актуальными обновлениями безопасности на момент разработки решения. Соответствующие версии обновлений безопасности должны быть обновлены на момент внедрения решения.

Версии используемых компонентов и библиотек не должны содержать общеизвестных уязвимостей (CVE) критического и высокого уровня, с оценкой по версии CVSS больше 6.9.

Разработчик решения обязан предоставить и поддерживать Software Bill of Materials (SBOM), содержащий список используемых зависимостей с используемыми версиями этих зависимостей.

Разработчик решения обязан поддерживать процедуру управления уязвимостями. Процедура должно включать в себя список имеющихся уязвимостей по уровню критичности, а также график исправления соответствующих уязвимостей в соответствии с установленным SLA на исправление.

7.6. Тестирование безопасности в процессе разработки

Тестирование безопасности должно проводиться на всех этапах разработки, начиная с модульного тестирования и заканчивая полным тестированием безопасности системы.

Первым уровнем защиты выступает статический анализ кода (SAST), который выполняется непосредственно на этапе разработки и позволяет выявлять потенциальные уязвимости еще до компиляции кода.

После развертывания приложения производится его динамический анализ (DAST), позволяющий оценить безопасность системы в рабочем состоянии. Дополнительную глубину тестированию придает интерактивный анализ безопасности приложений (IAST), объединяющий преимущества статического и динамического подходов.

Особое внимание уделяется тестированию конфигураций на соответствие требованиям безопасности, что позволяет выявить потенциальные проблемы на уровне настроек системы.

Для обнаружения нестандартных уязвимостей применяется фаззинг- тестирование, которое помогает найти проблемы, не выявляемые традиционными методами анализа.

Требования к конвейеру разработки (CI/CD)

Для соответствия требованиям к конвейеру разработки (CI/CD) тестирование безопасности в процессе разработки должно проводиться на всех этапах разработки, начиная с модульного тестирования и заканчивая полным тестированием безопасности системы.

Конвейер разработки обязан включать в себя как минимум следующие этапы тестирования безопасности:

- Secret Scanning: Автоматический поиск утечек конфиденциальных данных в исходном коде.
- Static Application Security Test (SAST): Анализ кода для выявления уязвимостей на уровне синтаксиса и структуры.
- Software Composition Analysis (SCA): Проверка используемых библиотек и компонентов на известные уязвимости.
- IaC Scanning: Анализ кода конфигурации инфраструктуры для обнаружения небезопасных настроек.
- Container Scanning: Проверка контейнеров на наличие уязвимых образов или конфигураций.

- Dynamic Application Security Test (DAST): Тестирование работающего приложения для выявления уязвимостей в реальном времени.

7.7. Документирование и аудит

Документирование в процессе безопасной разработки играет ключевую роль в обеспечении прозрачности и поддержании высокого уровня защиты.

Фундаментом документации служит подробная архитектурная документация, в которой детально описываются все механизмы безопасности системы. На её основе формируются и постоянно.

Обновляются результаты анализа рисков, позволяющие оценивать потенциальные угрозы и планировать меры защиты.

Все проведенные тесты безопасности фиксируются в соответствующих протоколах, обеспечивая возможность аудита и анализа эффективности принятых мер.

История изменений и патчей безопасности тщательно документируется, что позволяет отслеживать эволюцию системы защиты и оценивать эффективность внесенных изменений.

Завершающим элементом документации является подробное руководство по настройке безопасной конфигурации, обеспечивающее корректное развертывание и эксплуатацию системы.

7.8. Реагирование на инциденты

В современной разработке критически важно иметь отлаженные процедуры реагирования на инциденты безопасности. В основе этого процесса лежат четко определенные процедуры эскалации проблем безопасности, позволяющие оперативно привлекать необходимых специалистов к решению возникающих угроз.

При обнаружении критических уязвимостей запускается процесс экстренного выпуска патчей, обеспечивающий минимальное время реакции на возникающие угрозы. Важным элементом является своевременная и прозрачная коммуникация с пользователями о выявленных проблемах безопасности, что помогает поддерживать доверие к продукту и минимизировать потенциальный ущерб.

После каждого инцидента проводится тщательный анализ произошедшего с целью извлечения уроков и предотвращения подобных ситуаций в будущем. На основе полученного опыта регулярно производится обновление процессов разработки, что обеспечивает постоянное совершенствование системы безопасности.

7.9. Требования к защите данных

Хранение данных

В современных информационных системах безопасное хранение данных является критически важным аспектом. Особое внимание необходимо уделять защите конфиденциальной информации, включая персональные данные пользователей, финансовую информацию и корпоративные секреты.

Шифрование данных в состоянии покоя с использованием стандарта AES-256 обеспечивает необходимый уровень защиты для хранения персональных данных. При работе с платежной информацией критически важно применять методы обезличивания данных банковских карт, что существенно снижает риски при возможной компрометации системы. Для защиты аутентификационных данных необходимо использовать современные алгоритмы хеширования, которые обеспечивают надежную защиту паролей пользователей от различных видов атак.

При обработке электронных документов система должна обеспечивать полное

протоколирование всех операций и сохранение истории взаимодействия с документами. Особое внимание следует уделять управлению криптографическими ключами и сертификатами, включая своевременную проверку их валидности и поддержку процедур отзыва. Использование специализированных сервисов управления секретами, таких как HashiCorp Vault и аналогами, позволяет централизованно и безопасно хранить криптографические ключи и другую чувствительную информацию.

Внедрение систем мониторинга и аудита доступа к хранилищам данных должно быть неотъемлемой частью решения для своевременного выявления подозрительной активности. Важным аспектом является разработка и внедрение политик управления жизненным циклом данных, включая их безопасное уничтожение по истечении срока хранения. При использовании облачных хранилищ необходимо обеспечить дополнительный уровень защиты данных, независимый от мер безопасности провайдера. Реализация механизмов восстановления доступа к зашифрованным данным должна быть тщательно продумана и документирована для обеспечения непрерывности бизнес-процессов.

Требования к хранению данных

Для соответствия требованиям к хранению данных программное решение обязано обеспечивать хранение персональных данных в зашифрованном виде. Данные в состоянии покоя должны шифроваться с использованием стандарта AES-256.

В случае работы с данными банковских карт решение обязано хранить такие данные в обезличенном виде. Решение обязано обеспечивать хранение паролей пользователей в хешированном виде. Решение должно использовать BCrypt в качестве алгоритма для хеширования данных.

Если решение работает с ЭДО, то оно обязано хранить историю взаимодействия с подписываемыми документами. Решение обязано проводить валидацию сертификатов и цепочек доверия систем, в том числе проверку истечения срока действия. Также должна быть реализована возможность отзыва и смены сертификатов и ключей.

Решение обязано обеспечивать возможность безопасного хранения секретов и криптографических ключей с использованием внешних сервисов для хранения секретов, например HashiCorp Vault и аналоги.

7.10. Шифрование и криптография

Особое внимание следует уделять управлению криптографическими ключами. Даже самые надежные алгоритмы шифрования становятся бесполезными при компрометации ключей. Поэтому внедрение процедур регулярной ротации ключей и безопасного их хранения является критически важным. Использование аппаратных модулей безопасности (HSM) для хранения критичных ключей становится стандартной практикой в финансовом секторе.

При работе с системами электронного документооборота реализация механизмов электронной подписи требует особого внимания к процессам валидации сертификатов и проверки цепочек доверия. Необходимо обеспечить надежное хранение истории подписанных документов и возможность долгосрочной проверки подписей, учитывая возможное истечение срока действия сертификатов.

Требования к защищенности каналов связи

Для соответствия требованиям к защищенности каналов связи программное решение обязано проводить коммуникацию с пользователем, компонентами и внешними системами с помощью защищенных протоколов шифрования актуальных версий. На момент написания это TLSv1.3 с использованием стандарта AES-256.

Решение обязано использовать для шифрования каналов коммуникации сертификаты от доверенных центров сертификации.

Доверенными считаются внутренний центр сертификации Организации и корневые центры сертификации.

Решение обязано предоставлять возможность настройки параметров безопасности каналов связи, в том числе смену доверенных корневых сертификатов.

7.11. Управление доступом

Реализация эффективной системы управления доступом является краеугольным камнем информационной безопасности. Современные требования к парольным политикам должны учитывать как необходимость обеспечения высокого уровня безопасности, так и удобство использования для конечных пользователей. Использование парольных менеджеров и единой системы аутентификации (SSO) помогает найти баланс между этими требованиями.

Многофакторная аутентификация (далее МФА) стала обязательным требованием для систем, работающих с конфиденциальной информацией. При этом важно предоставить пользователям выбор между различными вариантами второго фактора – от аппаратных токенов до биометрических данных. Это повышает удобство использования системы и снижает риск отказа от использования МФА.

Внедрение ролевой модели доступа требует тщательного планирования и регулярного аудита. Практика показывает, что со временем в системах накапливаются избыточные права доступа, что увеличивает риски несанкционированного доступа к данным. Автоматизированные системы контроля и регулярный пересмотр прав помогают поддерживать принцип минимально необходимых привилегий.

Требования к ролевой модели и авторизации

Для соответствия требованиям к ролевой модели и авторизации программное обеспечение обязано предоставлять управление доступом на основе ролей (Role-Based Access Control - RBAC).

Решение обязано обеспечивать ролевую модель доступа с детальным разграничением прав пользователей в соответствии с доступным функционалом. Как минимум решение обязано предоставлять разделение обычных пользователей и администратора системы.

Решение обязано давать аутентифицированному пользователю доступ только к функционалу и данным в соответствии с его ролью в ролевой модели.

Решение обязано предоставлять возможность конфигурировать роли учетных записей со стороны администратора.

Решение обязано предоставлять возможность администратору отзывать имеющиеся сессии у пользователей.

7.12. Работа с тестовыми данными

Обеспечение безопасности при работе с тестовыми данными часто недооценивается, что может привести к серьезным инцидентам безопасности. Использование реальных данных в тестовых средах создает дополнительные риски утечки конфиденциальной информации. Поэтому процесс обезличивания данных должен быть тщательно проработан и автоматизирован.

При этом важно сохранять референтную целостность данных после обезличивания, чтобы тестовые сценарии оставались релевантными. Современные инструменты обезличивания позволяют сохранять статистические характеристики данных и их взаимосвязи, что критично для качественного тестирования.

Управление тестовыми средами требует особого внимания к вопросам безопасности. Часто в тестовых средах используются менее строгие настройки безопасности, что может быть использовано злоумышленниками как точка входа в систему. Поэтому важно обеспечить надежную изоляцию тестовых сред и регулярную очистку тестовых данных.

Требования к тестовым данным

Для соответствия требования к тестовым данным в процессе разработки запрещается использовать реальные данные вплоть до ввода решения в эксплуатацию.

Процесс обезличивания данных должен быть тщательно проработан и автоматизирован. Разработчик должен сохранять референтную целостность данных после обезличивания, чтобы тестовые сценарии оставались релевантными.

В процессе разработки должны быть реализованы функции для обезличивания и минимизации обработки персональных данных на тестовых стендах.

7.13. Интеграция и совместимость

Логирование

Построение эффективной системы логирования является критически важным элементом обеспечения безопасности. Логи безопасности должны предоставлять полную картину происходящего в системе, позволяя восстановить последовательность событий при расследовании инцидентов. При этом важно соблюдать баланс между детальностью логирования и производительностью системы.

Стандартизация форматов логов (CEF, Syslog) обеспечивает возможность их эффективной обработки в централизованных системах мониторинга безопасности. Это особенно важно в крупных организациях, где необходимо обеспечить единый подход к анализу событий безопасности от различных систем и приложений.

Защита самих логов от модификации становится критически важной задачей, особенно в свете требований регуляторов и необходимости использования логов как доказательной базы при расследовании инцидентов.

Требования к журналированию

Для соответствия требованиям к журналированию решение обязано проводить полный аудит действий пользователей в системе.

Решение обязано поддерживать детальное журналирование всех операций, включающее в себя как минимум следующие пункты о событии безопасности:

- Время в формате timestamp (ISO8601);
- IP-адрес;
- Тип события (event_type);
- Оценка критичности события (severity);
- Имя учетной записи (user_id);

Решение обязано структурировать данные журналирования в форматы JSON или CEF.

Решение обязано предоставлять возможность администратору конфигурировать уровень журналирования от вывода всех событий до вывода только критических событий.

7.14. Безопасность API

Безопасность API становится все более критичной в современных распределенных системах. Внедрение OAuth 2.0 и JWT обеспечивает гибкий и безопасный механизм аутентификации и авторизации, позволяя эффективно управлять доступом к различным ресурсам системы. При этом важно правильно настроить время жизни токенов и обеспечить возможность их оперативного отзыва при компрометации.

Контроль доступа к API требует комплексного подхода, включающего не только ограничение по IP-адресам, но и внедрение механизмов защиты от атак типа "брутфорс" и DDoS. Rate limiting и квоты использования API помогают защитить систему от перегрузки и потенциальных атак на отказ в обслуживании.

Мониторинг использования API и выявление аномальной активности позволяет оперативно реагировать на потенциальные угрозы безопасности. Современные системы мониторинга API способны выявлять подозрительные паттерны использования и автоматически блокировать вредоносную активность.

Требования к аутентификации и безопасности API

Для соответствия требованиям к аутентификации и безопасности API программное обеспечение обязано поддерживать многофакторную аутентификацию с поддержкой стандарта TOTP. Аутентификация с поддержкой стандарта TOTP должна работать с общедоступными клиентскими приложениями для многофакторной аутентификации, например: Google Authenticator и аналоги.

Если ПО имеет доступы к Автоматизированной Банковской Системе (АБС) или связано с работой с электронным документооборотом (ЭДО), то решение обязано поддерживать стандарты аппаратных токенов или биометрические данные в многофакторной аутентификации.

В случае поддержки нескольких стандартов многофакторной аутентификации решение обязано предоставлять пользователям возможность выбирать между различными вариантами стандартов.

Решение обязано поддерживать функционал парольной политики для локальной аутентификации внутри решения. Парольная политика должна быть настраиваемой. Парольная политика по умолчанию должна содержать следующие настраиваемые пункты:

Установить минимальную длину пароля в 14 символов;

Установить обязательный набор символов Большой буквы, маленькой буквы и цифр, спецсимволы необязательны;

Отключить обязательное требование к периодическому сбросу пароля;

Запретить использование общеизвестных, часто используемых паролей;

Запретить повторное использование паролей;

Обязательное требование к двухфакторной аутентификации;

Решение обязано поддерживать возможность блокировки учетных записей пользователей со стороны администратора системы.

В случае работы решения с API, решение обязано поддерживать аутентификацию по стандарту OAuth 2.0 с использованием токенов JWT- токенов в качестве ключей.

Решение обязано поддерживать возможность интеграции с внешними системами аутентификации, как минимум Active Directory (LDAP).

7.15. Конфигурация и развертывание

Безопасное конфигурирование и развертывание программных решений является важнейшим фактором обеспечения их защищенности в производственной среде. Пра-

вильная настройка параметров безопасности и соблюдение отраслевых стандартов позволяет минимизировать риски несанкционированного доступа и компрометации системы.

Процесс развертывания должен быть автоматизирован и воспроизводим, что позволяет избежать человеческих ошибок и обеспечить идентичность конфигурации во всех средах. При этом все параметры конфигурации, влияющие на безопасность, должны быть четко документированы и проверяемы.

Особое внимание следует уделять процессу обновления системы и ее компонентов. Необходимо обеспечить возможность безопасного обновления без прерывания работы сервисов и с минимальными рисками потери данных. Регулярное обновление компонентов системы критически важно для устранения известных уязвимостей.

Система управления конфигурацией должна предусматривать механизмы контроля версий и возможность отката изменений в случае обнаружения проблем. Это особенно важно при внедрении критических обновлений безопасности, когда необходимо иметь план отката в случае непредвиденных ситуаций.

Управление учетными данными сервисных аккаунтов требует особого внимания, так как компрометация таких учетных записей может привести к полной компрометации системы. Необходимо обеспечить регулярную ротацию учетных данных и строгий контроль доступа к ним.

Внедрение стандартов безопасности, таких как CIS Benchmark, должно быть неотъемлемой частью процесса развертывания. При этом важно регулярно проверять соответствие настроек актуальным версиям стандартов и своевременно вносить необходимые изменения.

Система мониторинга и журналирования должна охватывать все аспекты конфигурации и развертывания, включая изменения настроек, обновления компонентов и действия с учетными записями. Это позволяет обеспечить прозрачность процессов и возможность аудита.

Критически важно обеспечить безопасность процесса доставки обновлений, включая проверку целостности обновлений и использование защищенных каналов связи. Необходимо предусмотреть механизмы верификации источника обновлений и проверки их подлинности.

Управление конфигурацией должно учитывать требования по масштабируемости и отказоустойчивости системы, обеспечивая при этом неизменный уровень безопасности при увеличении нагрузки или изменении инфраструктуры.

Процессы конфигурации и развертывания должны быть интегрированы с общей системой управления безопасностью организации, обеспечивая единый подход к обеспечению защиты информации на всех уровнях.

Требования к конфигурации и развертыванию

Для соответствия требованиям к конфигурации и развертывания программное решение обязано соответствовать минимальным стандартам, определенным в CIS Benchmark для каждой из применимых сред развертывания. Версия CIS Benchmark должна быть актуальной версии на момент внедрения решения.

Решение обязано представлять возможность конфигурации уровня безопасности системы.

Решение обязано предоставлять функционал обновления компонентов системы, ее функционала и зависимостей. Функционал обновления должен быть включен в состав компонентов системы, а не являться отдельной от нее сущностью. Функционал обновления должен включать в себя журналирование всех событий обновлений системы.

Решение обязано предоставлять возможность отката до предыдущих версий без риска потери имеющихся данных и настроек.

Если решение использует сервисные учетные записи в компонентах системы, то

решение обязано предоставлять возможность регулярной ротации учетных данных для сервисных учетных записей

7.16. Обеспечение безопасности системы

Управление привилегиями

Принцип минимальных привилегий является фундаментальным требованием информационной безопасности, однако его практическая реализация часто сталкивается с серьезными вызовами. В современных системах, где количество компонентов и их взаимосвязей постоянно растет, корректное управление привилегиями становится критически важным фактором безопасности. Запуск сервисов под выделенными пользователями и точное определение необходимых системных вызовов помогает минимизировать потенциальный ущерб в случае компрометации отдельных компонентов системы. Практика показывает, что большинство серьезных инцидентов безопасности связано именно с избыточными привилегиями, которые позволяют злоумышленникам расширить начальную точку проникновения.

Регулярный аудит административных аккаунтов и служебных учетных записей является необходимым элементом поддержания безопасности системы. Особое внимание следует уделять своевременной деактивации учетных записей уволенных сотрудников и регулярной ротации учетных данных служебных аккаунтов. История знает немало примеров, когда именно неактуальные, но активные учетные записи становились причиной серьезных инцидентов.

7.17. Изоляция и защита

Современные подходы к изоляции компонентов через контейнеризацию предоставляют мощные инструменты для обеспечения безопасности, но требуют глубокого понимания и правильной настройки. Использование security-hardened базовых образов и регулярное сканирование контейнеров на уязвимости становится обязательной практикой. При этом важно помнить, что контейнеризация – это не серебряная пуля, и без правильной настройки сетевой изоляции и управления ресурсами она может создать ложное чувство безопасности.

Внедрение Web Application Firewall (WAF) требует тщательной настройки и постоянной адаптации правил под меняющиеся угрозы. Современные WAF должны не только защищать от известных атак из списка OWASP Top 10, но и использовать элементы машинного обучения для выявления новых векторов атак. Геолокационные ограничения и поведенческий анализ запросов помогают выявлять подозрительную активность еще до того, как она превратится в полноценную атаку.

Требования к изоляции и виртуализации

Для соответствия требованиям к изоляции и виртуализации решение обязано включать изоляцию каждого компонента системы используя контейнеризацию или виртуализацию. Решение обязано предоставлять сетевой доступ для пользователей только к необходимым для работы с системой компонентам решения.

В случае использования контейнеризации решение обязано использовать distroless-образы контейнеров. Эти образы должны пройти проверку на наличие уязвимостей контейнера и не содержать зарегистрированные уязвимости с оценкой по CVSS больше 6.9.

В случае использования контейнеризации решение обязано обеспечивать работу своих компонентов без необходимости предоставления повышенных или административных привилегий в операционной системе, обслуживающей решение.

В случае использования виртуализации в настройках гипервизора должна быть отключена опция использования общей памяти (shared memory) между виртуальными машинами.

В случае использования виртуализации образ виртуальной машины обязан соответствовать требованиям безопасности по CIS Benchmark. Версия CIS Benchmark должна быть актуальна на момент внедрения решения.

В случае использования виртуализации необходимо применить актуальные security-патчи на момент ввода в эксплуатацию.

7.18. Устойчивость и восстановление

Резервное копирование

Эффективная стратегия резервного копирования должна учитывать не только технические аспекты создания резервных копий, но и бизнес- требования к доступности данных. Определение правильных метрик RPO (Recovery Point Objective) и RTO (Recovery Time Objective) требует тесного взаимодействия между ИТ-специалистами и бизнес-подразделениями. Часто организации недооценивают реальную стоимость простоя систем и потери данных, что приводит к недостаточным мерам по их защите.

Шифрование резервных копий и контроль доступа к ним часто остаются без должного внимания, создавая потенциальную точку утечки конфиденциальной информации. История знает немало случаев, когда злоумышленники получали доступ к данным именно через недостаточно защищенные резервные копии. Поэтому важно обеспечить такой же уровень защиты резервных копий, как и основных данных.

Регулярное тестирование процедур восстановления является критически важным элементом обеспечения отказоустойчивости. Практика показывает, что без регулярных тестов восстановления многие резервные копии могут оказаться бесполезными в реальной кризисной ситуации. Важно проводить полное тестирование восстановления в условиях, максимально приближенных к реальным сценариям отказа систем.

Требования к резервному копированию и восстановлению

Для соответствия требованиям к резервному копированию и восстановлению решение обязано иметь документированные процедуры резервного копирования и восстановления данных с гарантированным временем восстановления (RTO) не более 4 часов для критичных компонентов. Под критичными компонентами понимаются компоненты, без которых система не будет функционировать.

Решение обязано поддерживать функционал резервного копирования данных системы. Функционал резервного должен быть настраиваемым, как минимум используемый протокол, адрес доставки для резервного копирования. Решение обязано ограничивать возможность доступа к уже сделанным резервным копиям системы.

Решение обязано предоставлять функционал восстановления данных, например, из резервной копии.

7.19. Отказоустойчивость

Построение отказоустойчивой архитектуры требует глубокого понимания как технических аспектов, так и бизнес-процессов организации. Реализация Active-Active кластеризации и автоматического переключения при сбоях должна учитывать специфику конкретных приложений и их требования к согласованности данных. Особое внимание следует уделять тестированию механизмов переключения и восстановления нормальной работы после сбоев.

Защита от программ-вымогателей становится все более актуальной задачей в свете участившихся атак на корпоративные системы. Комплексный подход к защите должен включать не только технические меры, такие как системы обнаружения вредоносного

ПО, но и организационные меры по повышению осведомленности пользователей и реагированию на инциденты.

Требования к отказоустойчивости

Для соответствия требованиям к отказоустойчивости решение обязано обеспечивать минимальный уровень доступности, определяемый в каждом конкретном случае ответственным лицом со стороны информационной безопасности Организации.

Решение обязано сохранять работоспособность при отказе некритичных для работы системы компонентов.

7.20. Работа с уязвимостями

Тестирование и сканирование

Регулярное проведение тестов на проникновение должно быть неотъемлемой частью процесса обеспечения безопасности. При этом важно понимать, что формальный подход к тестированию может создать ложное чувство безопасности. Тесты должны учитывать реальные сценарии атак и постоянно адаптироваться к новым угрозам. Привлечение разных команд пентестеров и использование различных методологий помогает получить более полную картину защищенности системы.

Автоматизированное сканирование на уязвимости должно проводиться непрерывно, а не только перед релизами или аудитами. Современные инструменты сканирования позволяют интегрировать проверки безопасности непосредственно в процесс разработки, что помогает выявлять и устранять уязвимости на ранних этапах.

Управление исправлениями

Эффективное управление процессом устранения уязвимостей требует четкой приоритизации и установленных SLA. При этом важно понимать, что не все уязвимости требуют немедленного исправления – необходимо учитывать реальный риск их эксплуатации в конкретном окружении. Правильная оценка критичности уязвимостей помогает оптимально распределять ресурсы на их устранение.

Процесс внедрения исправлений должен быть тщательно документирован и протестирован. История знает немало случаев, когда поспешное внедрение патчей безопасности приводило к серьезным сбоям в работе систем. Поэтому важно найти баланс между скоростью устранения уязвимостей и стабильностью работы систем.

7.21. Мониторинг и контроль

Построение эффективной системы мониторинга безопасности требует комплексного подхода, включающего сбор и анализ различных метрик безопасности. При этом важно не только собирать данные, но и правильно их интерпретировать, выявляя тренды и потенциальные проблемы до того, как они приведут к реальным инцидентам.

Регулярная отчетность по состоянию безопасности должна быть адаптирована под различные уровни управления – от технических специалистов до высшего руководства. Важно представлять информацию в форме, понятной конкретной аудитории, четко обозначая риски и необходимые действия по их минимизации. При этом отчетность должна быть не просто формальностью, а реальным инструментом принятия решений по улучшению безопасности системы.

8. Требования к поставщику решения

8.1. Требования к поставщику

1. Наличие экспертизы и понимание методологии автоматизируемой предметной области, релевантного опыта команды разработки и внедрения, референсов в соизмеримых по масштабу проектах;
2. Наличие не менее двух подтвержденных успешных внедрений в платежных системах/процессинговых центрах/банках, соизмеримых и более крупных по отношению к АО «Национальный межбанковский процессинговый центр» (может быть подтверждено справкой о наличии опыта и приложением актов приемки-услуг, подписанных заказчиками соответствующих проектов);
3. Наличие штата сотрудников с релевантным опытом и экспертизой в реализации и внедрении соответствующих предметной области решений (может быть подтверждено резюме сотрудников проектной команды с указанием названия проекта, сроков начала и окончания проекта и выполняемого функционала);
4. Готовность провести референс-визит для демонстрации внедренного проекта в контуре заказчика (заказчиков);
5. Подтверждение наличия сопроводительной документации;
6. Готовность внедрить и передать на сопровождение Систему в срок 6 месяцев.

8.2. Требования к предоставляемой поставщиком информации

В случае заинтересованности в участии в данной активности, просим компанию-претендента, в том числе, предоставить ответ на запрос в следующем составе:

1. Общая информация, с указанием наименования, даты основания, страны-учредителя, опыта в индустрии финансовых услуг
7. Полное наименование предлагаемых продуктов и их версии, состав и стоимости модулей, покрывающий полностью функциональность запроса;
8. Информация для оценки бюджета владения продуктом (коммерческое предложение);
9. В явном виде указать, какие требования поддерживаются предлагаемой системой без доработки, а какие не реализованы в текущей версии системы. Приложить оценочную стоимость и сроки доработок, в разрезе каждой недостающей функциональности;
10. Рекомендованные требования к серверным мощностям, отдельно указать требования к дискам (ssd (быстрые) или sas (обычные);
11. Отразить параметры и диапазоны от чего считается масштабирование;
12. Требования к сети и др. инфраструктуре;
13. Рекомендованную схему развёртывания; SLA по сопровождению, если есть разные варианты 8/5, 24/7 и описать основные виды сопровождения;
14. Описание рисков, рекомендации и предложения для лучшего планирования и реализации активности;
15. Объем ресурсов АО «Национальный межбанковский процессинговый центр», которые потребуются для дальнейшего сопровождения продукта;
16. Предоставить презентационные материалы, раскрывающие возможности системы;

17. Справку о наличии кадровых ресурсов и комплект резюме проектной команды, которую поставщик готов мобилизовать на проект, с указанием сроков реализации проектов, подтверждающих релевантный опыт, названия проекта и выполняемых в рамках проекта функций.

9. Перечень услуг по технической поддержке ПО

1. Наличие SLA-договорённостей (доступность системы не менее 99.95%).
2. Обеспечение технической поддержки и обновлений на срок не менее 3 лет.

10. Совокупная стоимость владения

Ниже приведены обязательные пункты коммерческого предложения, требуемые к предоставлению в рамках анализа рынка:

№ п/п	Статья	Расшифровка (что включает)
1	Лицензии	Необходимо: - предоставить расшифровку стоимости лицензии, информацию об используемой СУБД и технические требования к инфраструктуре (информация о количестве необходимых ядер, оперативной памяти, iops и пр.). - указать политику лицензирования (по пользователям, ядрам и т.д.) - указать порядок обновления (при наличии)
2	Внедрение	Указать порядок внедрения, количество затрачиваемых чел/дн.
3.1	Техническая поддержка 1 год	Описать условия и порядок - включено обновление ПО; - порядок постановки на сопровождение по доработкам, дозакупленным модулям и т.д.
3.2	Техническая поддержка на 3 года	Описать условия и порядок - включено обновление ПО; - порядок постановки на сопровождение по доработкам, дозакупленным модулям и т.д.
3.3	Техническая поддержка на 5 лет	Описать условия и порядок - включено обновление ПО; - порядок постановки на сопровождение по доработкам, дозакупленным модулям и т.д.
4	Обучение	Указать о необходимости обучения, стоимости и необходимом объеме