

Л.М. Алимжанова<sup>1</sup>, Н.Х. Маулина<sup>2</sup>, Ж.С. Жасболат<sup>1</sup>, А.Д. Нұрланбек<sup>2</sup>

<sup>1</sup>Международный университет информационных технологий, Алматы, Казахстан

<sup>2</sup>Академия логистики и транспорта, Алматы, Казахстан

E-mail: nurgul.maulina.79@mail.ru

## АНАЛИЗ ЦИФРОВЫХ ИНСТРУМЕНТОВ И ТЕХНОЛОГИИ ДЛЯ ЭФФЕКТИВНОЙ КОММУНИКАЦИИ ВНУТРЕННЕЙ БАЗЫ ДАННЫХ С ВНЕШНИМИ РЕСУРСАМИ КОМПАНИЙ

**Аннотация.** Современные бизнес-приложения редко живут изолированно. Они должны быть связаны, чтобы создать интегрированное решение, из которого организация может извлечь выгоду. Для больших компаний сложные задачи интеграции решает корпоративная служебная шина. Чтобы иметь возможность адаптироваться к рыночным изменениям и соответствовать соглашению об уровне обслуживания, организации сосредотачиваются на гибкости и скорости реагирования. В этой статье будет рассматриваться универсальное программное решение для эффективной коммуникации в условиях взаимодействия внутренней базы данных и нескольких фронтальных систем.

**Ключевые слова.** Корпоративная служебная шина, база данных, мобильное приложение, интеграция, сервис-ориентированная архитектура (SOA), архитектура, управляемая событиями (EDA), соглашение об уровне обслуживания (SLA).

### Введение.

В последние годы интеграция Enterprise Service Bus (ESB) в крупных компаниях становится все более популярной, поскольку компании стремятся улучшить свою ИТ-инфраструктуру и оптимизировать свои бизнес-процессы. ESB — это инструмент промежуточного программного обеспечения, который обеспечивает связь между различными приложениями и системами, что делает его важным компонентом любой современной ИТ-инфраструктуры.

ESB действует как центральный узел для всех коммуникаций между различными приложениями, что делает его мощным инструментом для компаний, которые хотят интегрировать свои бизнес-процессы. Используя ESB, организации могут воспользоваться преимуществами как SOA, так и EDA, не делая выбора между ними.

SOA фокусируется на создании многократно используемых сервисов, которые можно использовать в различных приложениях. Этот подход упрощает разработку и управление сложными системами, разбивая их на более мелкие и более управляемые компоненты. EDA, с другой стороны, представляет собой подход, который фокусируется на потоке событий между различными приложениями. Он позволяет обмениваться данными и обрабатывать данные в режиме реального времени, что очень важно в современной быстро меняющейся бизнес-среде.

30-секундное SLA — это соглашение о предоставлении услуг, в котором указывается, что запрос клиента должен быть разрешен в течение максимум 30 секунд. Это особенно важно в таких секторах, как электронная коммерция, финансы и телекоммуникации, где быстрое время отклика имеет важное значение для поддержания удовлетворенности и лояльности клиентов.

Соблюдая этот стандарт, предприятия могут повысить качество обслуживания клиентов, повысить эффективность работы и повысить свою репутацию. Таким образом, предприятия во всех отраслях должны уделять первоочередное внимание внедрению

систем и процессов, которые позволяют им соблюдать 30-секундное соглашение об уровне обслуживания. Что подчеркивает важность внедрения данного программного решения для более эффективной коммуникации Базы данных и фронтальной системы.

### **Материалы и методы.**

Одним из ключевых преимуществ использования ESB в крупной компании является возможность создания гибкой и масштабируемой ИТ-инфраструктуры. С помощью ESB компании могут легко добавлять новые приложения и системы в свою ИТ-среду, не беспокоясь о проблемах совместимости. Это облегчает компаниям адаптацию к изменяющимся потребностям бизнеса и сохранение конкурентоспособности на быстро меняющемся рынке.

И подобная система дает нам возможность реализовать асинхронное взаимодействие между этими системами.

Чтобы понять это понятие, необходимо сначала понять его антитезу, синхронное взаимодействие. Последний представляет собой рабочий механизм в форме запрос-ответ-запрос, основанный на структуре ответ-ответ-запрос и так далее.

Примерный сценарий синхронного взаимодействия можно проиллюстрировать на примере процесса получения кредита через банковское мобильное приложение. Первоначально заявитель заполняет форму общей информации (Форма 1) и ожидает, что система проведет проверку биографических данных на основе этой информации по черным спискам. После этого заявитель приступает к заполнению второй формы (Форма 2), содержащей информацию о своих доходах и расходах. Затем система ожидает, пока вторая система загрузит и найдет свой реестр должников, после чего будут выполнены определенные операции. Наконец, как только все эти шаги выполнены, заявитель перенаправляется на главную страницу, где он может подать заявку на получение кредита. В случае, если условия внутреннего кредита не соответствуют заранее установленным критериям системы, заявителю может быть отказано.

В конечном итоге это приводит к сценарию, в котором пользователь или клиент должен выдержать период ожидания примерно от 4 до 7 минут после завершения первого и второго шагов только для того, чтобы получить отказ на начальном этапе. На первый взгляд, этот период может показаться не критичным с точки зрения клиента. Однако важно признать, что это может привести к потере производительности для организации в двух случаях: а именно, не закрытие сделок и потеря части клиентов, как показывает клиентская статистика. В среднем 58 клиентов закрывают окно загрузки в течение 2-минут, а 20% пользователей не возвращаются для завершения процесса.

Следовательно, был задуман альтернативный подход в виде асинхронного взаимодействия. В этой методологии системы работают независимо друг от друга, выполняя операции в своем собственном темпе и обрабатывая данные, полученные от клиента, в индивидуальном порядке. Все проверки выполняются на верхнем фронтальном уровне после получения всей информации от клиента. В результате пользователь, успешно прошедший все проверки, может получить одобрение всего за 30 секунд и ответ о том, что предложение будет предоставлено в течение 10 минут. Что соответствует соглашению об уровне обслуживания (SLA) в 30 секунд.

### **Результаты и обсуждения.**

В сфере современных технологий для бизнеса становится все более важным использовать преимущества, предоставляемые общей базой данных и служебной шиной, для облегчения беспрепятственного взаимодействия между системами. Этого можно добиться путем имитации заявок, примером которых может служить процесс утверждения кредита. Перед выдачей кредита выполняются все необходимые проверки и условия, и в

случае их успешного выполнения необходимые данные собираются и передаются в базу данных.

	'GKS'	DOGOVOR	FROMDATE	P_DATA_INS	P_DATE_UPD	P_STATUS	P_PATH	PSSDOK	P_CLI_CODE	ID
1	GKS	EM23950-003393346	28.03.2023	28.03.2023 14:59:25	inserted				000950.00301477	345929
2	GKS	EM23200-003392948	28.03.2023	28.03.2023 14:59:27	inserted					345930
3	GKS	EM23980-003393487	28.03.2023	28.03.2023 14:59:35	inserted				000980.00144290	345931

Рисунок 1 - Отражение заявки во внутренней базе данных для фронтальных систем

Фронтальные системы (Рис-2) работают параллельно над заявкой отталкиваясь от статуса заявки. На момент этапа по “созданию/обновлению клиента” одна из систем регистрирует клиента, вторая формирует отчеты по сверке адресов, третья система обновляет реестр займов, связанных по ИИН клиента. Все эти внешние системы получают данные из базы данных, запросы кэшируются на сервисной шине и после выполнения отражают на заявке. Есть конечно системы, которые ждут перехода на этап “создания договора” в базе, но время прохождения каждого этапа сокращается до времени выполнения самой затратной операции, что дает нам меньше времени обработки, если бы системы работали поочередно по системе запроса и ответа, когда ответ приходит после выполнения всех операций.

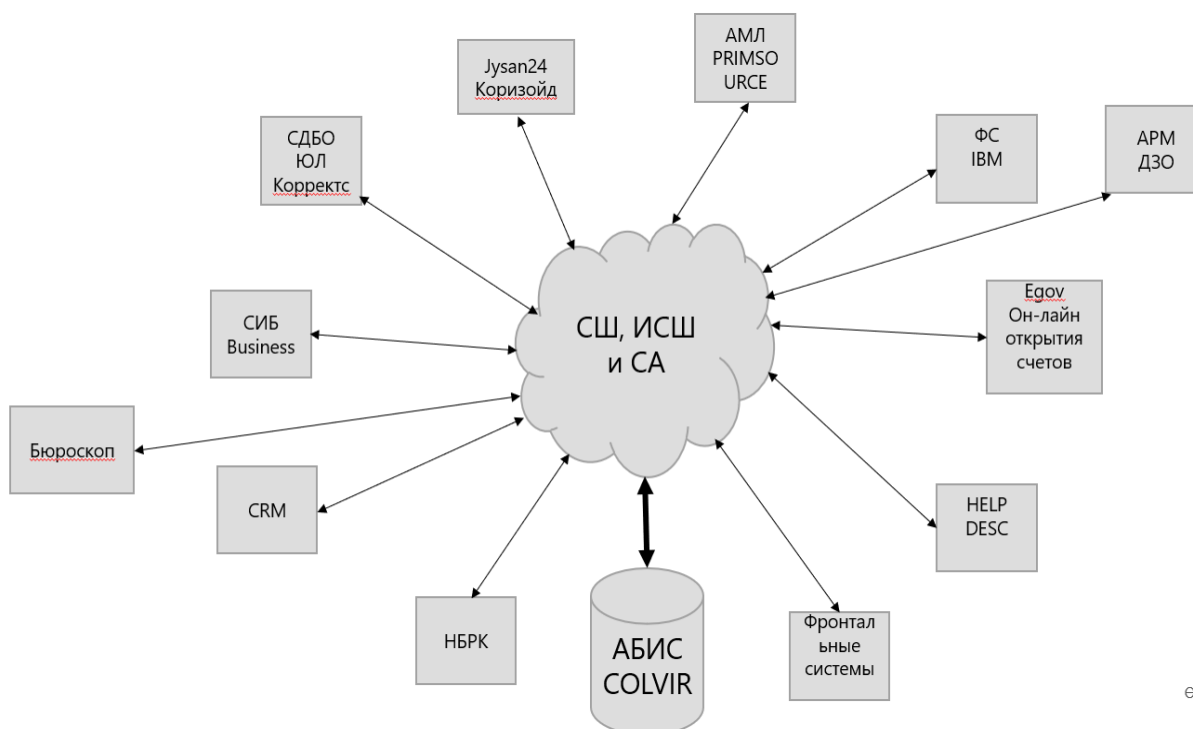


Рисунок 2 - Внешние ресурсы и база данных связанные единой корпоративной сервисной шиной

Заявка на выдачу займа проходит по всем нужным операциям (Рис-3), которые можно разделить на 3 главных этапа:

- 1) Создание/обновление клиента.
- 2) Создание займа и присвоение всех параметров обслуживания.
- 3) Регистрация и Актуализация (Проходит уже при подписании договора).

Обрабатываются все заявки и обновляются планировщиками, которые обновляют статус заявки и определяют дальнейшие операции по ней.

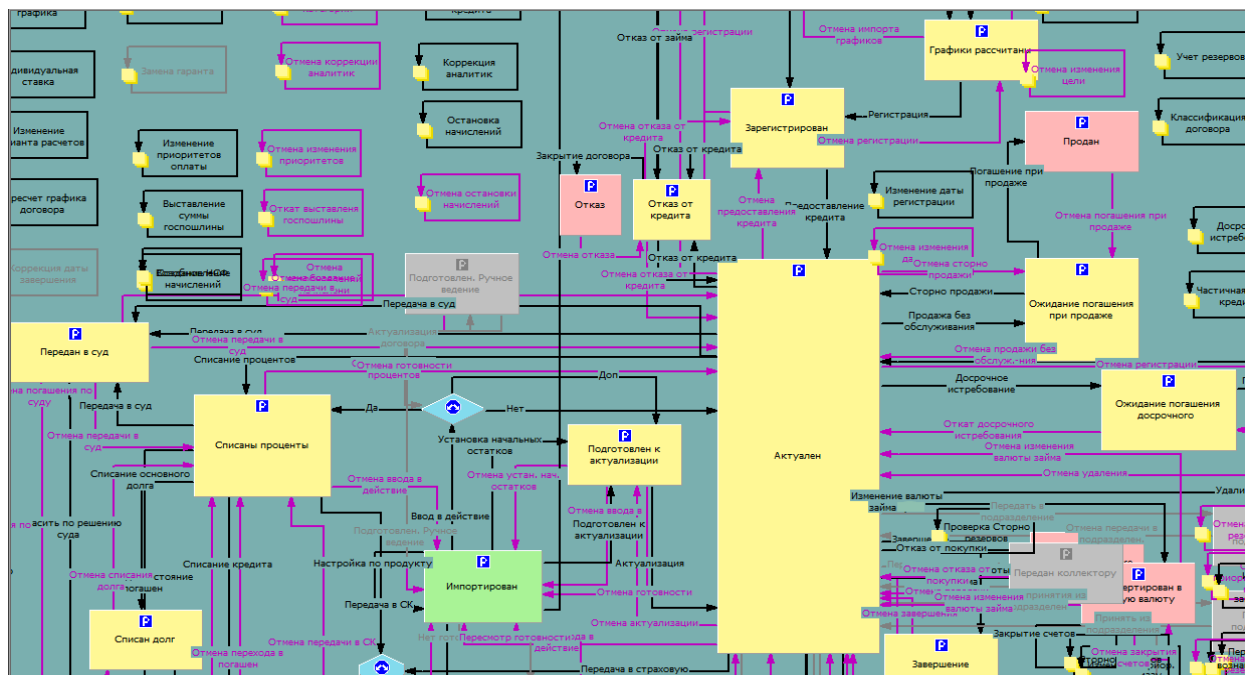


Рисунок 3 - Бизнес-процесс выдачи займа и этапы прохождения заявки на создание займа

Обработчик (Рис-4) заявок решает на котором этапе находится заявка, очередность операции и корректность выполнения для пропуска заявки на следующий шаг и т.д. Один обработчик может несколько раз проводить операции на заявке в случае технических ошибок, как блокировка объекта, сбой связи при отправке данных и отсутствие необходимых данных.

Связь на основе сообщений с несколькими получателями. В качестве более гибкого подхода вы также можете использовать механизм публикации/подписки, чтобы ваши сообщения от отправителя были доступны для дополнительных микрослужб подписчика или внешних приложений. Таким образом, это помогает вам следовать принципу открытости/закрытости в службе отправки. Таким образом, в будущем можно добавить дополнительных подписчиков без необходимости изменять службу отправителя. Когда вы используете связь публикации/подписки, вы можете использовать интерфейс шины событий для публикации событий любому подписчику.

При использовании асинхронной связи, управляемой событиями, микросервис публикует событие интеграции, когда что-то происходит в ее домене, и другой микросервис должен знать об этом, например, изменение цены в микросервисе каталога продуктов. Дополнительные микросервисы подписываются на события, чтобы получать их асинхронно. Когда это произойдет, получатели могут обновить свои собственные объекты домена, что может привести к публикации большего количества событий интеграции. Эта система публикации/подписки выполняется с использованием реализации шины событий.

```
create or replace procedure Z_011_CRE_XLOAN_BY_JOBID (JOBID IN Z_011_tranche_REQ.p_job_id$type) as
begin
  for req in (select *
              from Z_011_tranche_REQ z
              where (z.p_status not in ('done', 'Cancelled') or
                    z.p_status is null)
                    and z.p_dfrom >= trunc(sysdate)
                    and z.id is not null
                    and z.p_lloan_dep_id_id is null
                    AND Z.P_PRODUCT_CODE not IN ('CL1.24A', 'CL1.24R')
                    AND Z.p_job_id = jobid) loop
    declare
      v_stat      varchar2(30);
      v_cli_code  varchar2(30);
      p_cli_dep_id number(10);
      p_cli_id    number(10);
      v_istat     integer := 0;
      v_err_txt   varchar2(4000);
      v_dea_dep_id number(10);
      v_dea_id    number(10);
      v_trf_cat   number(10);
      V_CLI_IIN_BIN varchar2(12);
    begin
      c_pkgconnect.pOpenLink(upper(nvl(req.p_usercode, c_fprm('Z_011BUS_USRMP'))), 1);

      update Z_011_tranche_REQ set p_job_lock = 1 where id = req.id;

      select r.p_status, r.p_cli_code, r.p_cli_iin_bin
             into v_stat, v_cli_code, v_cli_iin_bin
             from z_011_loan_req r
             where r.prefer = req.p_request_ref;

      select /*R.DEA_ID,
             R.PAR_ID,
             substr(T_PkgDeaPrm.fGetShowValue(R.DEA_ID, R.PAR_ID, null), 1, 250) as PARVAL,*/
             R.PARVALUE
             into v_trf_cat
             from T_DEAPRMRUL R, T_DEAPRMDSC, C_VALLIST
```

Рисунок 4 - Реализация алгоритма обработки планировщика

В случаях, когда заявка не может обработаться сама в течении дня то бот, который проверяет состояния заявок каждые 10 минут сообщает мониторинговой группе администраторов, которые в свою очередь оперативно доводят заявку до успешного конечного состояния.

Учитывая, что один обработчик способен работать над 2000 заявками, необходимо обеспечить независимую обработку каждого запроса путем изоляции соответствующего процесса. Как показано на рис. 4, значения параметров очищаются при переходе к новому циклу запроса. Процессор выбирает соответствующие приложения из совокупного пула на основе связанного идентификатора и инициирует выделенный обработчик для каждого отдельного процесса. Чтобы приспособиться к выбору приложений, необходимо установить индивидуальные конфигурации.

### Заклучение.

В современную эпоху предприятия работают в среде, где скорость и эффективность имеют первостепенное значение для успеха. Одним из ключевых показателей, используемых для измерения эффективности бизнеса, является соглашение об уровне обслуживания (SLA). SLA — это соглашение между поставщиком услуг и клиентом, определяющее качество обслуживания, которое будет предоставлено в течение

определенного периода времени. В последние годы появился новый стандарт в виде 30-секундного SLA, который быстро становится эталоном для предприятий в различных отраслях.

Возможность интегрировать эти данные в соответствующие системы, участвующие в процессе подачи заявки на кредит и регистрации, облегчается за счет использования служебной шины. Благодаря этому механизму все клиентские системы могут получить доступ к информации и данным, необходимым для завершения процесса утверждения кредита, что повышает эффективность работы.

Данное программное решение дает возможность крупным бизнес-процессам придерживаться 30-секундного SLA стандарта и дальнейшие вопросы будут стоять в минимизировании рисков после предварительного одобрения заявки. И оптимизация скорости обработки.

В заключение, очевидно, что внедрение программного решения для эффективной связи с базой данных и интерфейсными системами может оказать значительное влияние на эффективность и производительность организации. С ростом зависимости от принятия решений на основе данных крайне важно иметь оптимизированную систему, обеспечивающую беспрепятственный обмен данными между базами данных и интерфейсными приложениями. Преимущества такого решения включают повышенную точность данных, более быстрое время отклика и повышенную масштабируемость. Однако важно тщательно оценить различные программные решения и выбрать то, которое соответствует конкретным нуждам и требованиям организации. Таким образом, предприятия могут эффективно использовать технологии для оптимизации своей деятельности и получения конкурентного преимущества в сегодняшнем быстро развивающемся цифровом ландшафте.

## ЛИТЕРАТУРА

[1] Нагмех Никнежад, Вайда Исмаил, Имран Гани, Бехзад Назари, Махади Бахари Аб Разак Бин Че Хуссин. Понимание сервис-ориентированной архитектуры (SOA): систематика, 2020 г.

[2] «Улучшение эффективной связи между внутренними базами данных и внешними ресурсами в контексте аналитики больших данных», Ю. Чен, Дж. Лю и К. Чжан. Журнал бизнес-исследований, 2021.

[3] «Интеграция внутренних и внешних баз данных для эффективной коммуникации в системах здравоохранения» М.Ю. Зайеда и Х. Эль-Мараги. Журнал медицинских систем, 2020.

[4] «Структура для эффективной связи между внутренними базами данных и внешними ресурсами в контексте бизнес-аналитики» С. Сюн, Л. Се и Ю. Ли. Журнал управления корпоративной информацией, 2021 г.

[5] «Проектирование интеллектуальной системы связи для эффективной интеграции внутренних баз данных и внешних ресурсов в электронной коммерции» С. Ван, С. Чжоу и Дж. Лю. Журнал бизнес-исследований, 2021.

[6] «Новый подход на основе данных для эффективной связи между внутренними базами данных и внешними ресурсами в контексте управления цепочками поставок» Х. Ву, Ю. Сюй и Ю. Чжао. Журнал бизнес-исследований, 2020.

## REFERENCES\*

[1] Nagmeh Niknezhad, Vajda Ismail, Imran Gani, Behzad Nazari, Mahadi Bahari Ab Razak Bin Che Hussin. Ponimanie servis-orientirovannoj arhitektury (SOA): sistematika, 2020 g.



[2] «Uluchshenie jeffektivnoj svjazi mezhdv vnutrennimi bazami dannyh i vneshnimi resursami v kontekste analitiki bol'shih dannyh», Ju. Chen, Dzh. Lju i K. Chzhan. Zhurnal biznes-issledovaniy, 2021.

[3] «Integracija vnutrennih i vneshnih baz dannyh dlja jeffektivnoj kommunikacii v sistemah zdavoohranenija» M.Ju. Zajeda i H. Jel'-Maragi. Zhurnal medicinskih sistem, 2020.

[4] «Struktura dlja jeffektivnoj svjazi mezhdv vnutrennimi bazami dannyh i vneshnimi resursami v kontekste biznes-analitiki» S. Sjun, L. Se i Ju. Li. Zhurnal upravlenija korporativnoj informaciej, 2021 g.

[5] «Proektirovanie intellektual'noj sistemy svjazi dlja jeffektivnoj integracii vnutrennih baz dannyh i vneshnih resursov v jelektronnoj kommercii» S. Van, S. Chzhou i Dzh. Lju. Zhurnal biznes-issledovaniy, 2021.

[6] «Novyj podhod na osnove dannyh dlja jeffektivnoj svjazi mezhdv vnutrennimi bazami dannyh i vneshnimi resursami v kontekste upravlenija seepochkami postavok» H. Vu, Ju. Sjuj i Ju. Chzhao. Zhurnal biznes-issledovaniy, 2020.

**Лаура Әлімжанова**, т.ғ.к., доцент, Халықаралық ақпараттық технологиялар университеті, Алматы, Қазақстан, l.alimzhanova@iitu.edu.kz

**Нүргүл Маулина**, э.ғ.к., ассистент-профессор, Логистика және көлік академиясы, Алматы, Қазақстан, nurgul.maulina.79@mail.ru

**Жансая Жасболат**, магистрант, Халықаралық ақпараттық технологиялар университеті, Алматы, Қазақстан, zhans1335@gmail.com

**Айгерім Нұрланбек**, сениор-лектор, Логистика және көлік академиясы, Алматы, Қазақстан, aika9008@mail.ru

## КОМПАНИЯЛАРДЫҢ СЫРТҚЫ РЕСУРСТАРЫМЕН ІШКІ ДЕРЕКТЕР БАЗЫНЫҢ ТИІМДІ БАЙЛАНЫСЫ ҮШІН ЦИФРЛЫҚ ҚҰРАЛДАР МЕН ТЕХНОЛОГИЯЛАРДЫ ТАЛДАУ

**Андатпа.** Заманауи іскерлік қосымшалар оқшауланған жағдайда сирек өмір сүреді. Олар ұйым пайда алатын кешенді шешім жасау үшін байланыстырылуы керек. Ірі компаниялар үшін күрделі интеграциялық тапсырмалар Enterprise Service Bus арқылы өңделеді. Нарық өзгерістеріне бейімделу үшін ұйымдар икемділік пен жауаптылыққа назар аударады. Бұл мақалада ішкі деректер қоры мен бірнеше фронттық жүйелердің өзара әрекеттесу контекстінде тиімді байланыс үшін әмбебап бағдарламалық шешім қарастырылады.

**Түйінді сөздер.** Enterprise Service Bus, Database, мобильді қосымша, интеграция, қызметке бағытталған архитектура, оқиғаға негізделген архитектура, қызмет деңгейі келісімі (SLA).

**Laura Alimzhanova**, candidate of technical sciences, docent, International University of Information Technology, Almaty, Kazakhstan, l.alimzhanova@iitu.edu.kz

**Nurgul Maulina**, candidate of economic sciences, assistant professor, Academy of logistics and transport, Almaty, Kazakhstan, nurgul.maulina.79@mail.ru

**Zhansaya Jasbolat**, master's student, International University of Information Technology, Almaty, Kazakhstan, zhans1335@gmail.com

**Aigerim Nurlanbek**, senior lecturer, Academy of logistics and transport, Almaty, Kazakhstan, aika9008@mail.ru

## ANALYSIS OF DIGITAL TOOLS AND TECHNOLOGIES FOR EFFECTIVE COMMUNICATION OF THE INTERNAL DATABASE WITH EXTERNAL RESOURCES OF COMPANIES

**Annotation.** Modern business applications rarely live in isolation. They must be linked to create an integrated solution that the organization can benefit from. For large companies, complex integration tasks are handled by Enterprise Service Bus. To be able to adapt to market changes, organizations are focusing on flexibility and responsiveness. This article will consider a universal software solution for effective communication in the context of the interaction of an internal database and several front-end systems.

**Keywords.** Enterprise Service Bus, Database, mobile application, integration, service-oriented architecture, event-driven architecture, service level agreement (SLA).

\*\*\*\*\*