



Простые решения
для сложных проектов

Что такое СУБД «Енисей»?

«Енисей» — это документо-ориентированная, высокопроизводительная, распределённая NoSQL СУБД с поддержкой multi-master replication.

Отличительными особенностями СУБД «Енисей» являются:

- 01 простота в разработке и эксплуатации
- 02 высокая производительность
- 03 экономичность
- 04 кроссплатформенность

СУБД «Енисей» спроектирована для применения в высоконагруженных системах реального времени. Она позволяет хранить миллиарды документов, предоставляет инструментарий для работы с такими объемами данных.

У СУБД «Енисей» есть несколько вариантов использования.

Варианты использования СУБД «Енисей»



Мощный и отказоустойчивый файловый сервер



Распределенная коммуникационная инфраструктура для вашего решения



Отказоустойчивое документное хранилище позволяет строить решения, способные хранить десятки миллиардов документов без деградации производительности



Поисковый индекс для документного хранилища. Многократно ускоряет процессы поиска документов и формирование ETL выгрузок из хранилища



Хранение ГЕО данных в соответствии с требованиями RFC 7946 GeoJSON



Часть проекта **TrioData** — технологического стека, на базе которого можно строить высоконагруженные решения для хранения и обработки больших массивов данных

«Енисей» как СУБД

В настоящее время документные хранилища всё более востребованы, поскольку в ряде случаев удобнее и эффективнее реляционной модели хранения. JSON давно стал стандартом в индустрии, ведь именно данный формат является основой RESTful API, которые активно используются в подавляющем большинстве информационных систем.

Безусловно, удобнее хранить ваши документы в нативном формате в специализированном хранилище, предоставляющем широкий набор инструментов для работы с данными в этом формате.

Есть множество кейсов, когда гораздо удобнее использовать специализированные СУБД для хранения документов, как в JSON так и в бинарном виде.

Функциональная архитектура СУБД «Енисей»



Например, организация документного подхода к хранению данных. Если предметная область позволяет оперировать самодостаточными сущностями, коими как раз и являются документы, гораздо удобнее реализовать хранение именно по такому принципу. Например, в виде документа можно сохранить информацию о клиенте, вы получите не реляционную структуру с набором таблиц и необходимостью их джойна (JOIN) при построении запросов, а самодостаточный документ с любым количеством уровней вложенности, скорость его выборки из БД будет многократно превышать скорость работы с реляционной СУБД.

Ваша система уже работает с JSON, в этом формате обмениваются данными сервисы. REST является основным архитектурным стилем в современных системах, используемым для API как внешнего, так и для микросервисного взаимодействия. Так как работа с «Енисей» осуществляется через обычный HTTP и именно через RESTful и не требует специализированных языков построения запросов, то применение нашей СУБД в современных системах, как минимум в качестве оперативной БД, представляется наиболее выгодным решением по сравнению с реляционными СУБД. Ваши разработчики скажут спасибо!

«Енисей» — это append only СУБД, т.е. данные всегда добавляются, даже если вы обновляете какой-то документ. Скорость работы в разы выше по сравнению с реляционными решениями, так как полностью исключаются ситуации блокировки. Индексы строятся асинхронно, что также существенно ускоряет работу.

Стоит сказать про дисковое пространство. Большинство современных реляционных СУБД создают довольно много служебных файлов, очень большого размера, часто отношение полезных данных к служебным бывает 1:6, иногда выше в зависимости от СУБД. В «Енисей» объем хранилища чуть больше, чем суммарный объем полезных данных в нем.

Хранение больших массивов документов в реляционных структурах, как правило, крайне затратно. Если ваша система с ними интенсивно работает, разработчикам придется постоянно искать баланс между количеством индексов (которые перестраиваются после каждой вставки или обновления данных и, как правило, синхронно) и требованиями бизнеса к скорости и удобству работы с данными. Не забываем про джойны (JOIN) и «многоэтажные» SQL-запросы, с которыми придется работать, добиваясь требуемой производительности.

«Енисей» работает с БД, где миллиарды документов, без деградации скорости записи, чтения и обновления данных.



Представления (Views)

Для сложных выборок есть высокопроизводительные представления (Views), которые можно писать на нескольких языках:

01 JavaScript

03 CoffeeScript

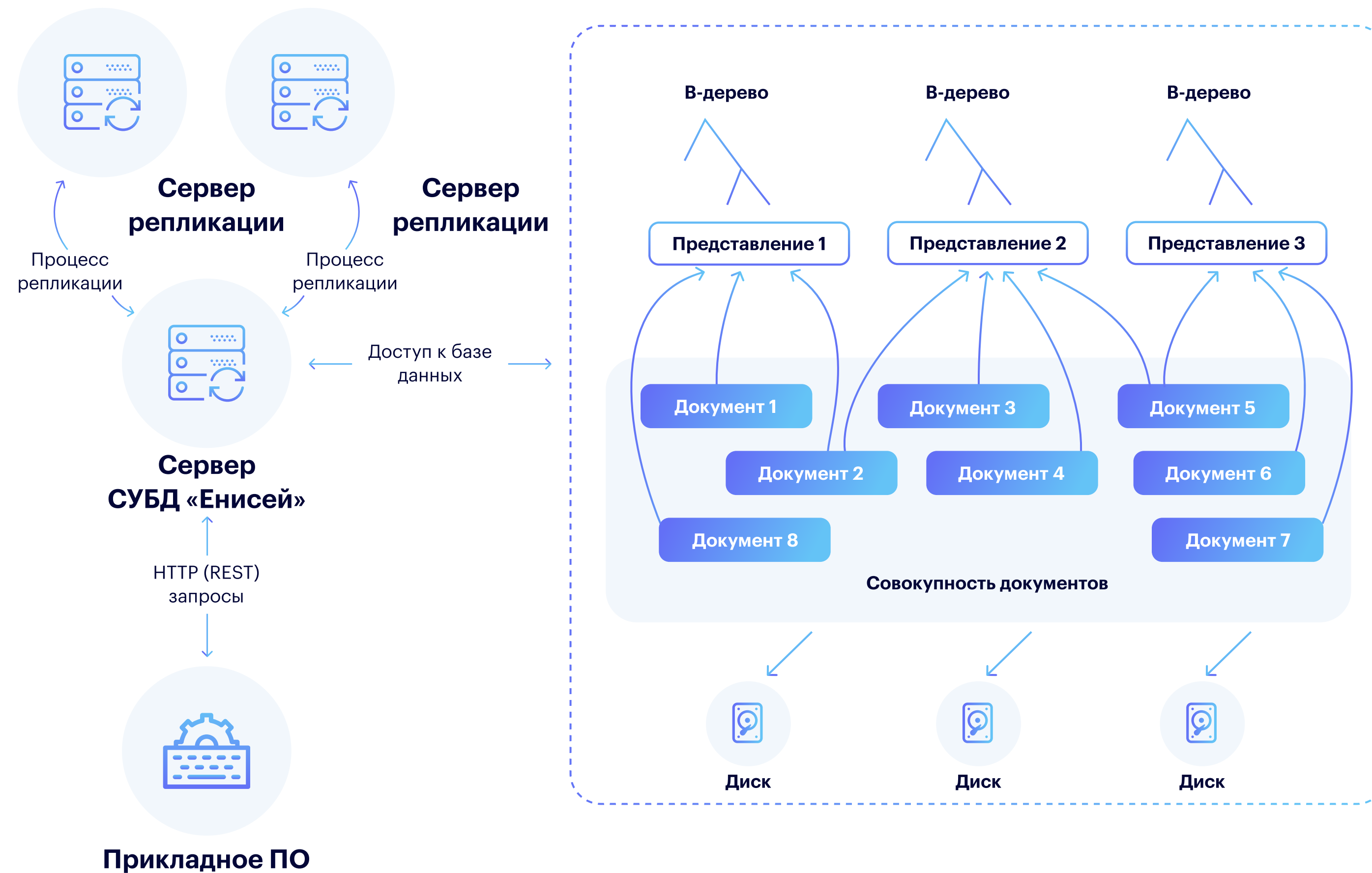
02 Erlang

04 Elixir

Внутри представления разработчик может писать любую логику, на привычном ему языке (в вашей команде точно есть разработчики, знакомые с JavaScript). Тут важно — любая логика в пределах возможности языка программирования. Это не SQL-запросы в несколько «этажей», это простой и понятный любому разработчику код. В работе представлений используется модель MapReduce, что позволяет вам реализовать очень сложные выборки данных.

Таким образом, вы сохраняете документы в нативном формате вашей системы, работаете с ними через простой и понятный HTTP протокол командами GET и POST, PUT, DELETE, и разработчикам не нужно изучать специализированный язык для работы с СУБД.

Разработчики могут писать представления и фильтры к запросам на обычном языке программирования, используя очень мощный движок выполнения запросов с поддержкой MapReduce.

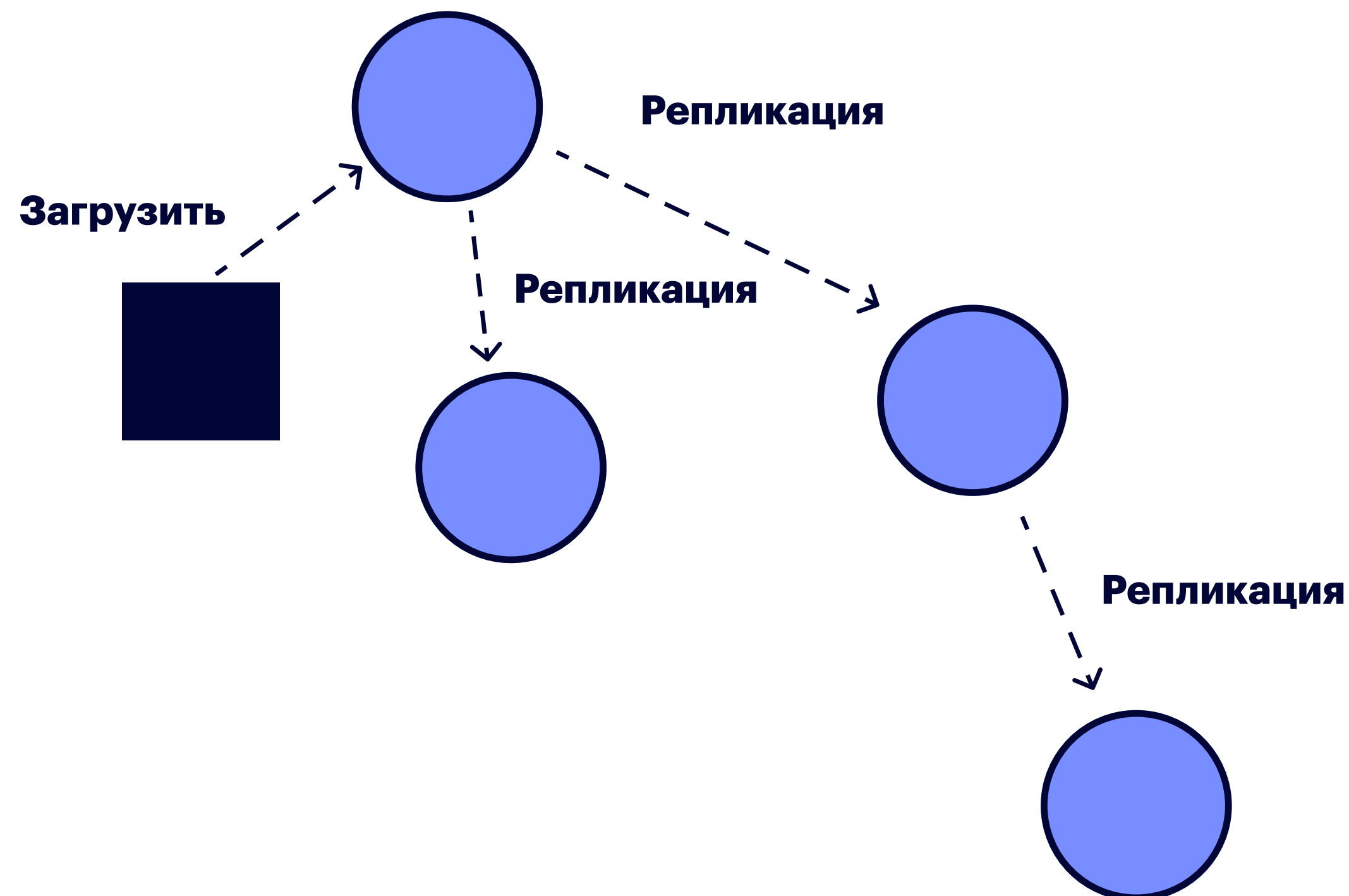


Представления могут работать с массивами документов, осуществляя очень сложные выборки из них и обработку данных.

Также можно строить запросы на Mango — языке запросов, основанном на JSON и не требующем знаний JavaScript/MapReduce. Эта возможность будет особенно полезна тем, у кого есть опыт работы с MongoDB.

Репликация

«Енисей» реализует модель multi-master-репликации. Каждая нода БД является мастером, можно одновременно писать данные в неё и читать их. Вы можете реализовать достаточно сложные механики репликации данных между нодами, настраивая, какие документы должны реплицироваться, а какие — нет. Все очень гибко.



Благодаря тому, что СУБД «Енисей» разработана на языке Erlang (о нём подробно см. ниже), скорость репликации и надежность ее работы очень высоки. «Енисей» также обеспечивает решение конфликтов между версиями документов.

Для того чтобы решать конфликты версий в БД, используется механизм MVCC (Multi-Version Concurrency Control).

«Енисей» как сервер приложений

В СУБД «Енисей» есть такая фишка как «проектный документ», — приложение, описанное в JSON-документе. Приложение может включать в себя несколько функций:

- 01 Views
- 02 Validate
- 03 Show
- 04 Filters
- 05 Updates

В них мы можете описать бизнес-логику обработки, фильтрации, представления и трансформации данных. Кратко о функциях:

- Views — это представления, о которых мы написали выше. Позволяют вам комбинировать данные из разных документов, применять к ним map и reduce функции, стандартные или написанные вами.
- Filters — логика фильтрации данных, при выборке из БД. Функции данного типа могут быть использованы в настройке репликации. Таким образом вы можете строить сколь угодно сложные сценарии репликации данных между нодами БД.
- Updates — это функции, которые создают, или обновляют документы в БД.

- **Validate** — это функция валидатор данных. Если необходимо осуществить форматно-логический контроль документа перед его записью в БД, логика описывается в данной функции.
- **Show** — функции трансформации данных в другие форматы, функция может возвращать HTML-страницу, XML-документ или любой другой из Content-Types.

Таким образом, вы можете имплементировать в БД бизнес-логику приложения, вызывая его посредством RESTful API, напрямую из UI, по имени функции. Существует комплекс кейсов, в которых такой подход может сэкономить вам много времени и средств.

```
{
  "_id": "_design/ddoc1",
  "_rev": "5-fb926fc19806221b83c8d9939a13ebf2",
  "views": {
    "view1": {
      "reduce": "_sum",
      "map": "function (doc) {\n  emit(doc.kind,
doc.price);\n}"
    }
  },
  "shows": {
    "book_title": "function(doc, req) {\n  return
'<h1>' + doc.title + '</h1>';\n}"
  },
  "filters": {
    "food_only": "function(doc, req) {return
(doc.kind === \"food\");}",
    "drink_only": "function(doc, req) {return
(doc.kind === \"drink\");}"
  },
  "validate_doc_update": "function(newDoc, oldDoc,
userCtx, secObj) {if (oldDoc) {\n      if
(oldDoc.name !== newDoc.name) {\n
throw({forbidden: 'Usernames can not be
changed.'});\n      }\n    }\n  }",
  "language": "javascript"
}
```

«Енисей» как интеграционная шина



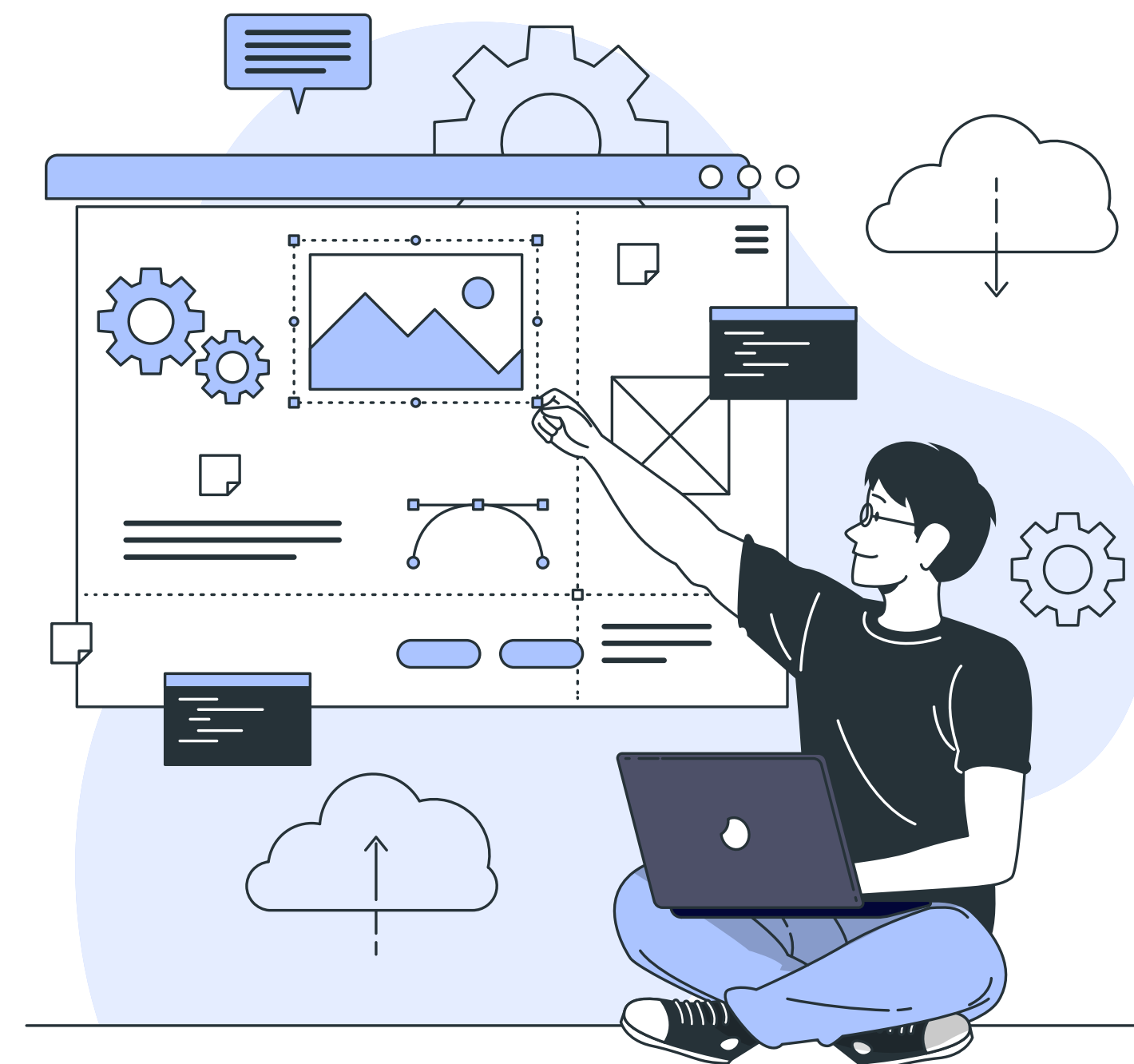
Вы можете использовать «Енисей» как шину, через которую компоненты вашей инфраструктуры обмениваются данными. Компоненты и сервисы пишут данные в БД. Благодаря функциям листенеров обработчики внутри компонентов могут перехватывать необходимые документы и обрабатывать их.

Любая распределенная система состоит из сервисов и компонентов, взаимодействующих между собой. В классическом случае мы пишем для каждого RESTful API и погружаемся в увлекательный процесс отладки взаимодействия. По мере развития нашей системы мы вынуждены поддерживать API, обеспечивать поддержку старых версий.

Если использовать в качестве связующего звена «Енисей», сервисы вашей системы будут взаимодействовать через БД. Так как «Енисей» — это мульти-мастер СУБД, передача данных между сервисами будет обеспечена посредством репликации данных между нодами базы. Языку Erlang посвящена отдельная страница ниже, настоятельно рекомендуем ознакомиться с ней для полного понимания возможностей СУБД «Енисей».

Благодаря многопоточности скорость репликации у СУБД «Енисей» намного выше, чем традиционное взаимодействие между компонентами и сервисами через RESTful API.

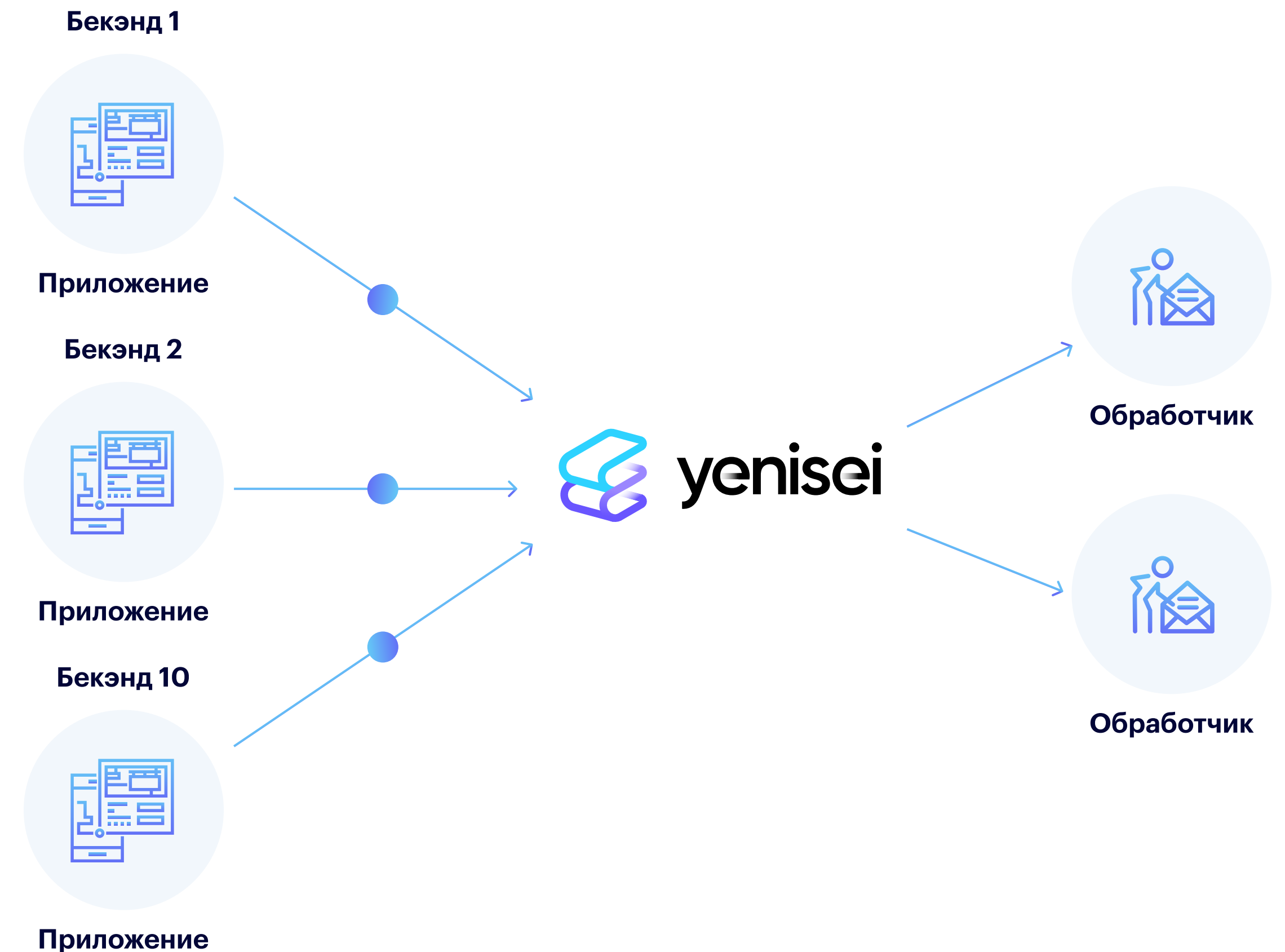
Для разработчиков все те же привычные механизмы, но внутри мощь и скорость multi-master репликации.



«Енисей» как очередь сообщений

Вы можете использовать «Енисей» как очередь сообщений, ваши приложения и сервисы могут писать данные в БД, как описано выше, обработчики будут получать их благодаря механизму листенеров.

Мы разработали уникальный метод блокировки документов, предназначенный для контроля того, что один документ будет обработан одним обработчиком. Обработчик может заблокировать документ, взяв его в работу, тем самым, не допустив параллельной обработки одного документа несколькими обработчиками.



Во многих случаях может быть удобно не расширять стек, добавляя в него новые компоненты, например, очереди сообщений, требующие специальных знаний, опыта и эксплуатационных затрат. Благодаря СУБД «Енисей» вы можете реализовать аналогичные функции, решая с их помощью задачи очередей обработки документов.

Простота

Одним из немаловажных аргументов в пользу СУБД «Енисей» является простота начиная с процесса установки и настройки. Так как «Енисей» представляет собой документо-ориентированную СУБД, выполняющую строго данную функцию и связанные с ней задачи, её очень просто установить и настроить. Её настройки «из коробки» достаточны для подавляющего большинства проектов. Можно с уверенностью сказать, что 95% пользователей не будут менять настройки по умолчанию.

Еще вам не нужно знать SQL или другой язык взаимодействия с СУБД. Вам нужно знать HTTP, а это всего несколько всем знакомых команд: GET, POST, PUT, DELETE, — вот собственно и все, что вы должны знать и уметь, чтобы начать работу с СУБД «Енисей».

СУБД имеет RESTful-интерфейс, может быть интегрирована в вашу систему фактически без затрат, как любой другой сервис. Если говорить о пороге входа, то он попросту нулевой.

Для написания представлений (views) вы можете использовать JavaScript, входящий в тройку самых популярных языков программирования в мире. Представления необходимы вам для построения сложных выборок данных из БД и фильтров. И в них вы можете использовать любую логику, в рамках возможностей JavaScript. Далеко не каждая СУБД предоставляет такую возможность. Поддерживается CoffeeScript, Erlang и Elixir.

В ряде случаев вы можете имплементировать логику работы с данными прямо в СУБД в виде представлений или приложений (design documents), которые вызываются посредством RESTful API, вы можете реализовать всю бизнес-логику или её существенную часть непосредственно в СУБД «Енисей».

UI может обращаться напрямую к СУБД. Это существенно упрощает разработку мобильных приложений или, например, клиентов для ваших систем.

Для того чтобы сделать разработку вашей системы с СУБД «Енисей» еще проще, мы предоставляем вам в комплекте фреймворк «Ангара». В нём мы реализовали ODM (Object-Document Mapping) по аналогии с ORM для реляционных СУБД (помните Hibernate?). Инструмент крайне необходим для сокращения сроков разработки и упрощения разработки решений. «Ангара» предоставляет разработчику множество классов, которые можно использовать в своих проектах.

Простота – это экономия и эффективность. Решение, отлично выполняющее свои задачи, не пытающееся выполнять функции других продуктов или решать не свои задачи, по определению будет работать надежнее и эффективнее, чем комбайны (сравнение «Енисей» с многофункциональными реляционными комбайнами есть в отдельной статье).

Erlang

Рассматривать «Енисей» в отрыве от языка, на котором он был написан неправильно, так как Erlang сам по себе предоставляет нашим клиентам целый ряд уникальных возможностей. Ведь именно благодаря им, он широко применяется в настоящее время в высоконагруженных решениях.

Кстати, серверная часть WhatsApp, которым многие пользуются ежедневно написана именно на нем, Facebook написал на нем серверную часть чата, Amazon применяет его в EC2. Erlang стал применяться для разработки облачных систем ещё до того, как сформировалось само понятие облачных вычислений.

В ядре СУБД «Енисей» находится мощный «движок», обеспечивающий репликацию данных между нодами в несколько потоков. Здесь и раскрывается мощь Erlang как языка, который был целенаправленно разработан для применения в распределённых, отказоустойчивых, параллельных системах реального времени и применяется в системах видеостриминга и коммуникационных проектах.

Для сравнения: большинство популярных современных языков программирования были разработаны без предварительного определения их области применения.

Erlang имеет конкретную область применения, плюс феноменальную надёжность и скорость.

Отличительной особенностью языка является применение легковесных процессов в соответствии с моделью акторов. Такой подход позволяет выполнять одновременно миллионы таких процессов, у каждого из которых могут быть скромные требования по памяти. Erlang с самого начала проектировался для распределенных вычислений и масштабируемости. Распределение вычислений встроено в синтаксис и семантику языка, поэтому построение системы можно вести, абстрагируясь от конкретного места вычислений.

За счет встроенных возможностей распределенных вычислений объединение в кластер, балансировка нагрузки, добавление узлов и серверов, повышение надежности требуют лишь небольших затрат на дополнительный код.

Благодаря Erlang «Енисей» является решением высокой доступности, пригодным для применения в системах реального времени.

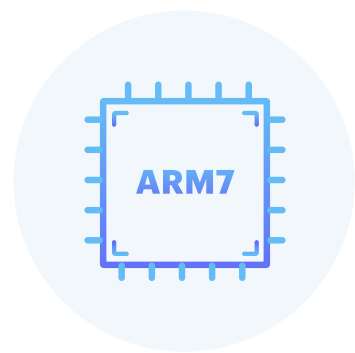


Кроссплатформенность

Для СУБД «Енисей» подойдут несколько платформ:



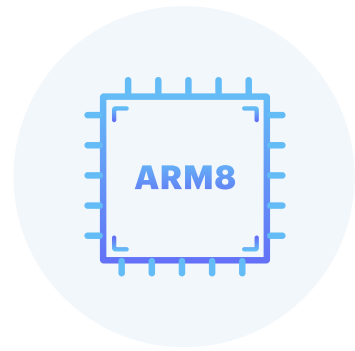
Linux ОС, включая основные
русские



ARM7



Windows



ARM8



Android



Версия, написанная на JavaScript,
работающая в браузере

Таким образом, с «Енисей» вы можете реализовать распределённые системы, объединяющие компоненты на разных ОС и платформах.

«Енисей» — это единственная СУБД с таким широким выбором платформ.

Кроссплатформенность позволяет существенно сэкономить, особенно на разработке клиентских компонентов вашей инфраструктуры, терминалов, мобильных приложений.

С СУБД «Енисей» можно быстро начать разработку в условиях нечетких требований, постепенно развивая функционал сервисов. База данных может хранить как старые, так и новые сообщения или документы.

Для прототипирования систем это уникальная возможность, серьезно сокращающая затраты.

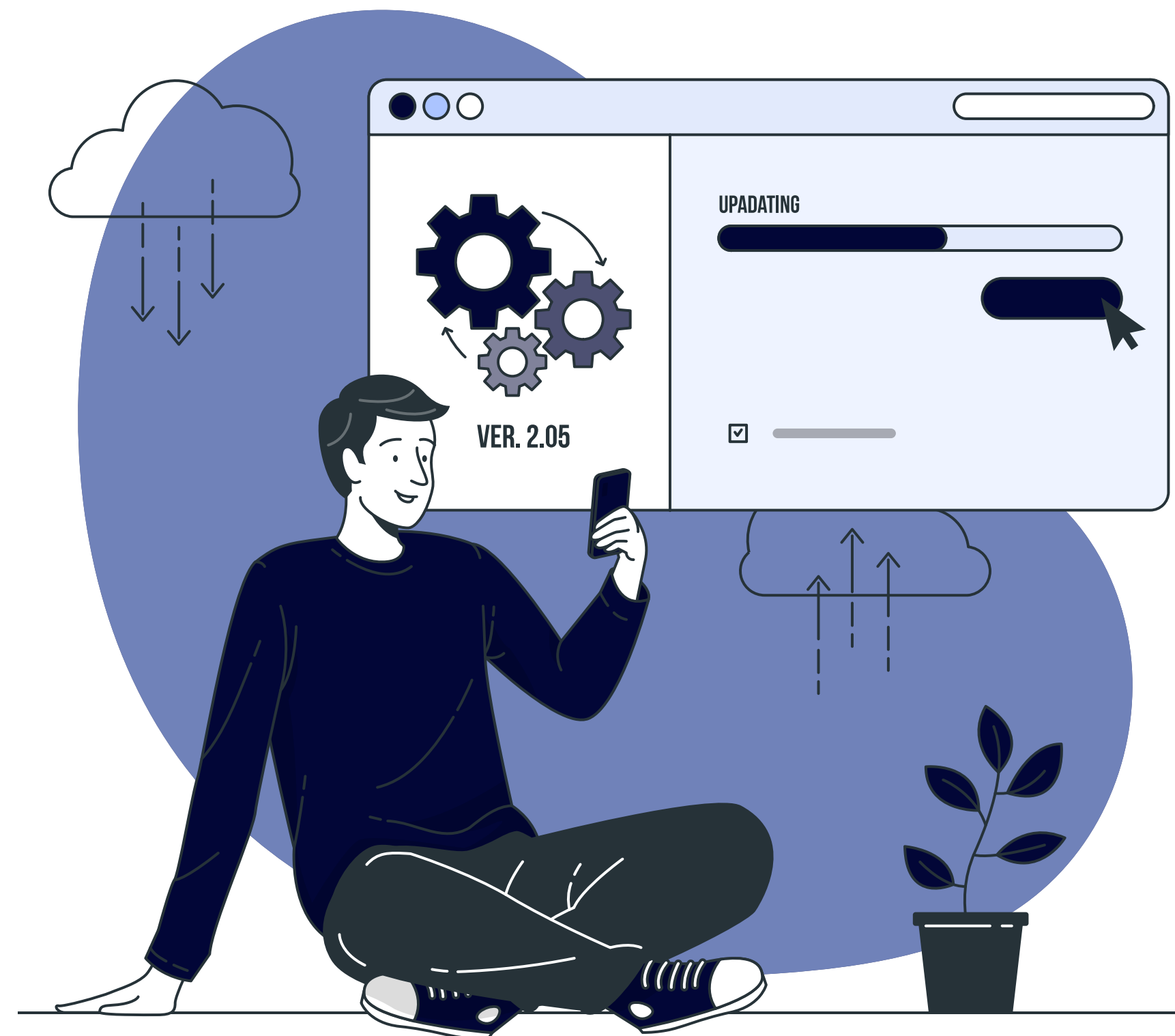


Сравнение «Енисей» и PostgreSQL

	Енисей	PostgreSQL
Схема данных	Нет фиксированной схемы данных	Фиксированная схема данных
Способ хранения данных	Документы	Таблицы
Поддержка ACID	✓	✓
Поддержка транзакций	×	✓
Специализированный язык запросов	×	✓
RESTful API для работы с СУБД	✓	×
Мульти-мастер репликация	✓	Нет. Но возможно через сторонние решения
Асинхронная репликация	✓	×
Настройка правил репликации	✓	×
Поддержка MapReduce	✓	×
Валидаторы данных	✓	×
Сложные фильтры данных	✓	×
Представления (Views) на обычных языках программирования	✓	×
Мультиплатформенность	✓	×
MVCC	✓	✓
Append-only	✓	×

Основное отличие — модель хранения данных. У PostgreSQL — реляционная структура, у «Енисей» — документы. Каждая модель имеет свои преимущества и недостатки, и выбор СУБД зависит от ваших целей и задач.

«Енисей» — это высокопроизводительная СУБД, предназначенная для применения в системах, ориентированных на высокую нагрузку. Время отклика «Енисей» не деградирует по мере роста размера БД, благодаря append-only режиму и особенностям реализации СУБД.



Сравнение «Енисей» и MongoDB

	Енисей	MongoDB
Формат хранения данных	JSON с хорошим сжатием	BSON
Организация хранения	Документы	Коллекции документов
Протокол взаимодействия	RESTful	Проприетарный протокол
Язык запросов	RESTful API, собственного языка взаимодействия нет	MQL
Мультиплатформенность, в том числе мобильные версии	Windows, Linux (в том числе отечественные), ARM7, ARM8, Android, JavaScript версия	Windows, Linux, мобильных версий нет
Поддержка от вендора на территории РФ	✓	×
Поддержка MapReduce	✓	✓
Поддержка MVCC	✓	✓
Поддержка multi-master replication	✓	×
Настройка правил репликации	✓	×
Возможность писать функции в БД, например для валидации данных	✓	×
Поддержка ACID	✓	✓
Хранение бинарных файлов (фото, видео и другие)	✓	✓
Append-only	✓	×

Миграция с Oracle

При миграции с решений Oracle и в ряде случаев Microsoft, чаще всего приходится пересматривать архитектуру слоя хранения данных. На рынке нет решений, способных один-к-одному заменить Oracle, иначе они бы активно продавались и заменяли столь дорогостоящее решения уже давно.

Один из подходов, который широко применяется при такой миграции - это разделение функций, которые ранее выполняла одна СУБД Oracle. на две СУБД: реляционную и документную. Что позволит сократить объем данных, что хранится в одной БД, разделив его на специализированные решения.

Во многих проектах, с которыми мы сейчас работаем как консультанты, в БД Oracle хранили и JSON и реляционное представление данных и сейчас мы помогаем клиентам провести рефакторинг архитектуры и вывести хранение документов и бинарных данных в Енисей.

Отказоустойчивость и высокая доступность

«Енисей» предоставляет вам несколько вариантов обеспечения отказоустойчивости и распределения нагрузки:

- 01 кластер
- 02 multi-master репликация
- 03 шардирование

В случае эксплуатации с другим нашим решением — Imaginarium вы получаете реплицируемые дисковые массивы, как дополнительную возможность обеспечения отказоустойчивости системы. Комбинация вариантов и их тонкая настройка под вашу систему позволяют добиться невероятной производительности.

Преимущества «Енисея» по сравнению с «ванильным» CouchDB

- ✓ **Проверенная и переработанная кодовая база.** Исправлено множество багов, удален многолетний легаси код. Это позволяет сократить время реакции на запросы клиентов по багфиксу или наращиванию функционала.
- ✓ **Переработанная обработка ошибок СУБД.**
- ✓ **Поддержка от вендора.** У нас большой опыт работы с Енисеем, мы можем помочь с внедрением и решить возникающие вопросы, принять участие в разработке архитектуры слоя данных.
- ✓ **Документация на русском языке.**
- ✓ **Представления (Views) на Erlang с отладкой.** Практически, работа как в IDE — подсветка ошибок компиляции и рантайма.
- ✓ **Представления (Views) на Elixir.**
- ✓ **UI для работы с фильтрами, валидаторами, обновлениями.**
- ✓ **Поддержка LDAP в самом Енисее без использования дополнительных прокси авторизации.**
- ✓ **Липкие сессии (Sticky sessions).** Очень актуально при динамической работе с данными в кластере, т.к. это реализует дополнительный уровень обеспечения консистентности данных.
- ✓ **Контроль доступа к данным. С использованием дискреционной и мандатной модели разграничения.**
- ✓ **Блокировки, позволяющие применять Енисей в качестве базы для реализации очередей обработки данных.**
- ✓ **Swagger-документация ко всем методам API.**
- ✓ **Индексатор.** Отдельный модуль для хранения метадаты документов и быстрого поиска, и встроенный индексатор для наполнения поискового индекса.