

Логическая репликация, особенности и обоснование. Сопровождение распределенных систем. Особенности и прелести PSQL

Докладчик: Александр Куров

Разработчик: Владимир Архипов

Инструменты создания и администрирования единого территориально-распределенного информационного пространства

1 Логическая репликация



Бизнес задача, обоснование, архитектура и примеры использования

2 Синхронизация с использованием PSQL



Использование PSQL и ESOE в алгоритмически сложных задачах синхронизации данных между базами данных

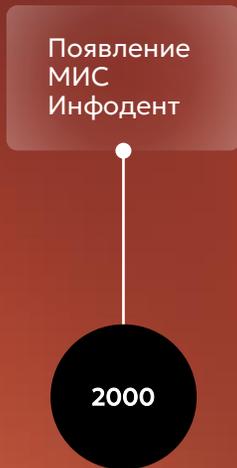
3 Дедубликация данных с использованием PSQL



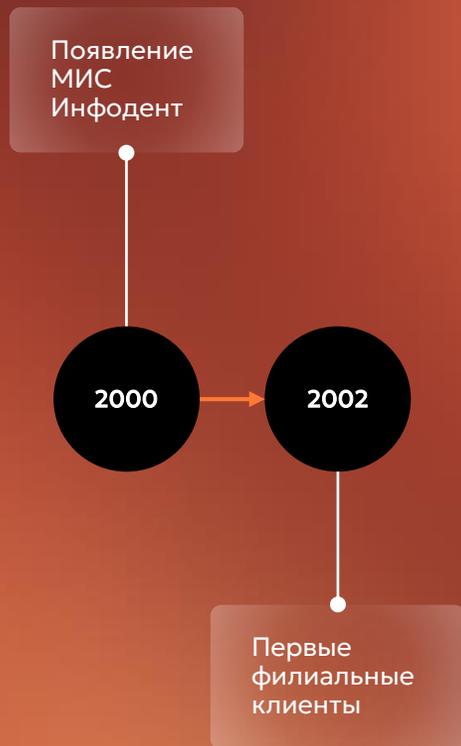
Унификация справочной информации в территориально распределенной системе. Опыт решения задачи средствами СУБД

История развития наших инструментов репликации и синхронизации

История развития наших инструментов репликации и синхронизации



История развития наших инструментов репликации и синхронизации



История развития наших инструментов репликации и синхронизации



История развития наших инструментов репликации и синхронизации



История развития наших инструментов репликации и синхронизации



История развития наших инструментов репликации и синхронизации



История развития наших инструментов репликации и синхронизации



История развития наших инструментов репликации и синхронизации



История развития наших инструментов репликации и синхронизации



История развития наших инструментов репликации и синхронизации

Переход на Firebird 2.5
(on external, parameterized
statements)

Кол-во филиальных сетей
превысило 100.

Кол-во филиалов
превысило 300.



2011

История развития наших инструментов репликации и синхронизации



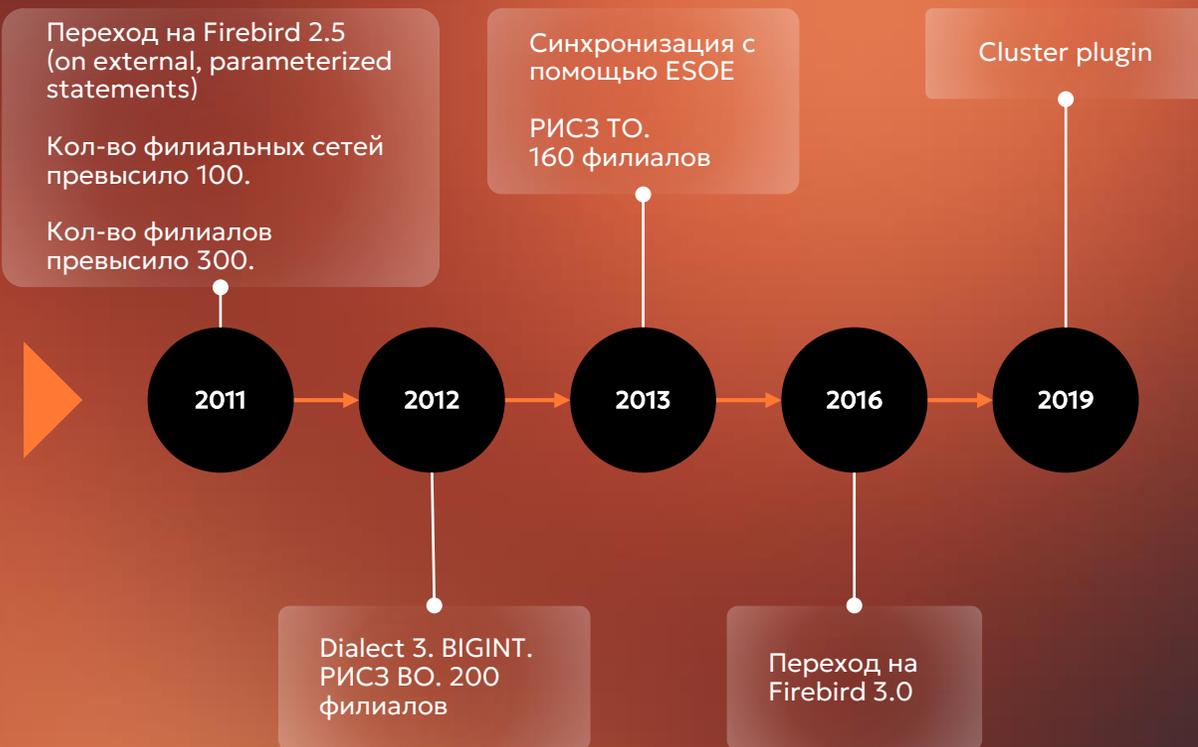
История развития наших инструментов репликации и синхронизации



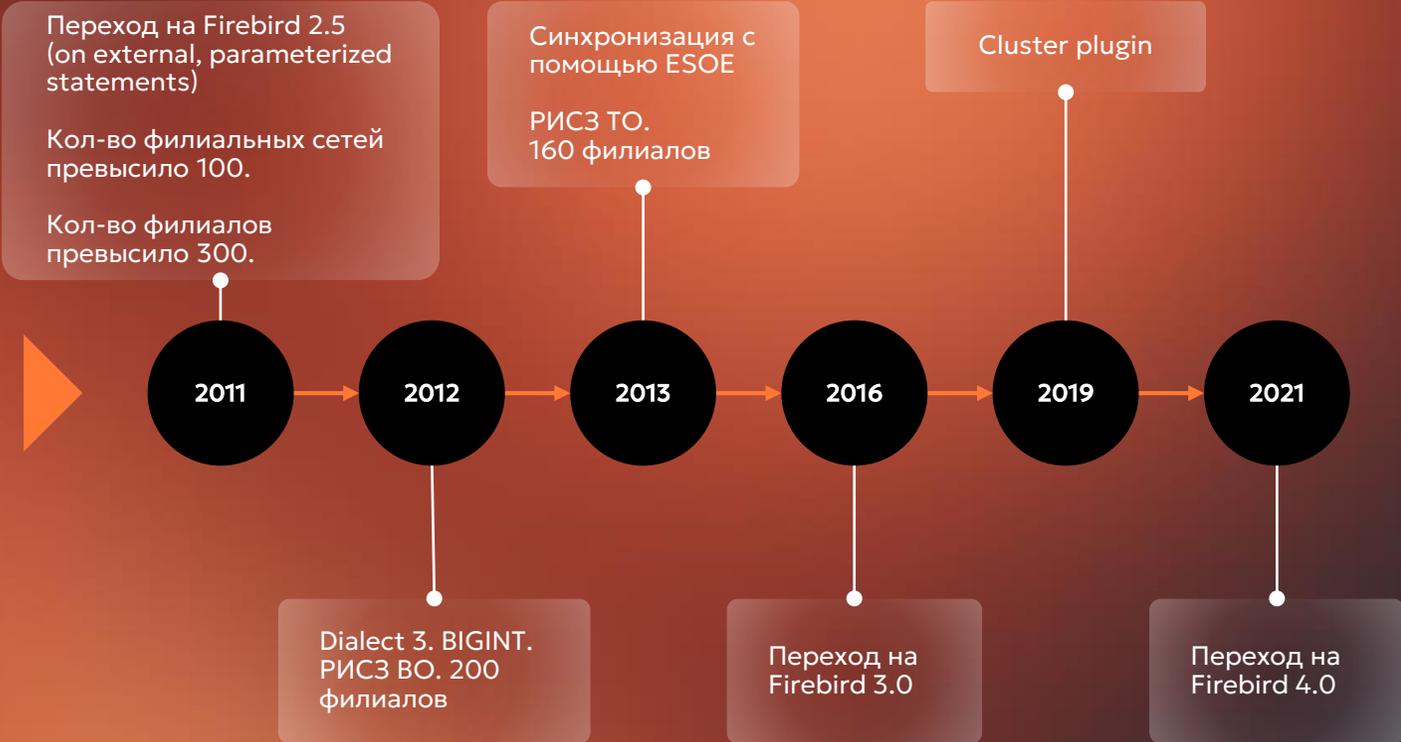
История развития наших инструментов репликации и синхронизации



История развития наших инструментов репликации и синхронизации



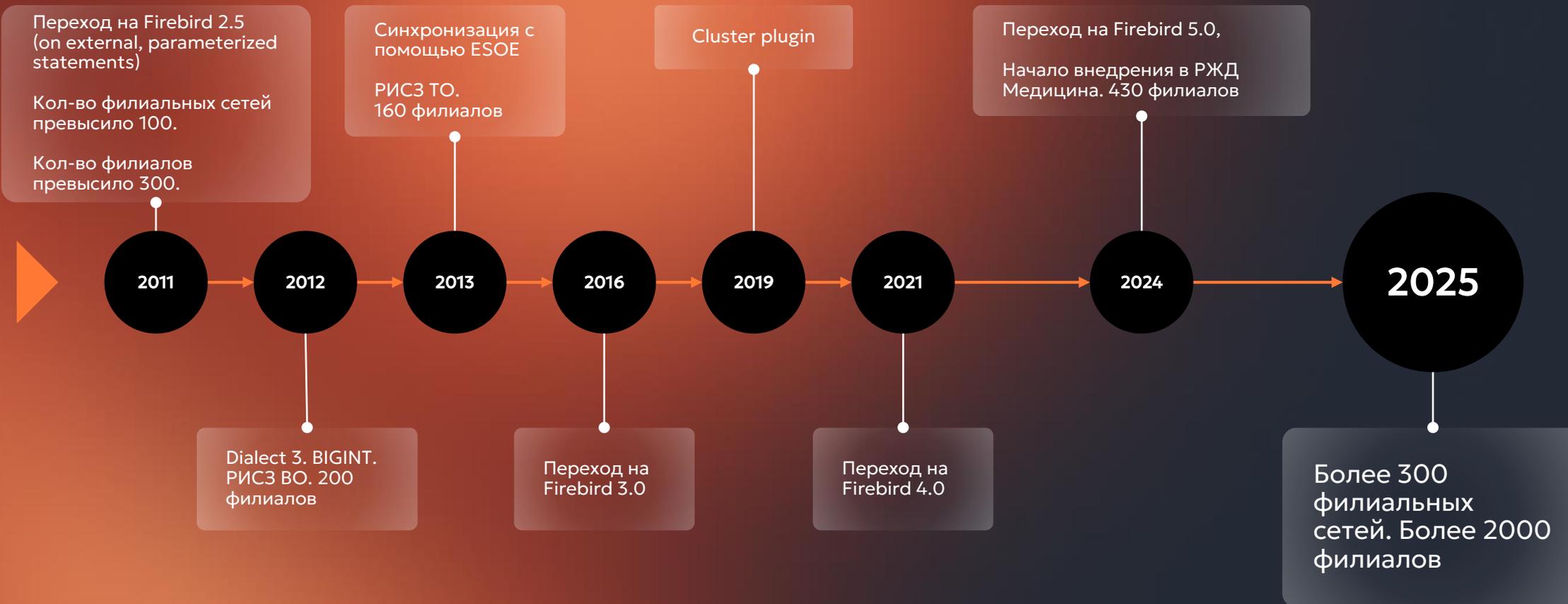
История развития наших инструментов репликации и синхронизации



История развития наших инструментов репликации и синхронизации



История развития наших инструментов репликации и синхронизации



Бизнес требования к системе репликации

Бизнес требования к системе репликации

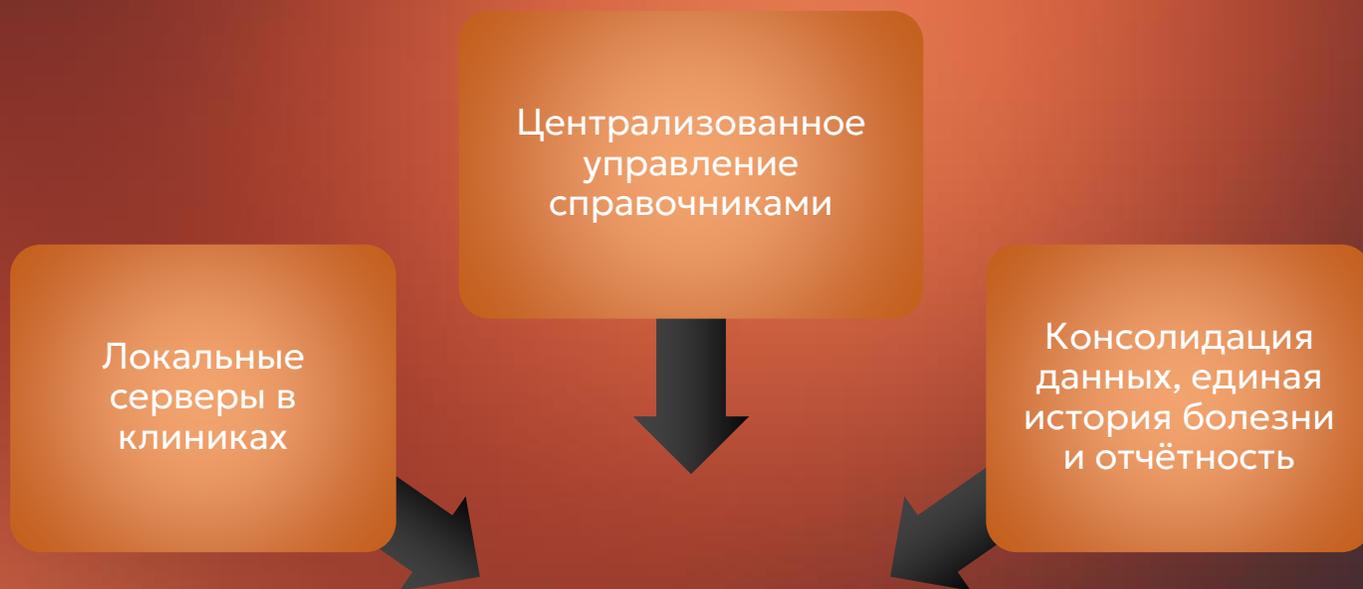
Локальные
серверы в
клиниках

A diagram consisting of a rounded orange rectangle on the left containing the text 'Локальные серверы в клиниках'. A dark grey arrow points from the right side of this rectangle towards the right side of the slide.

Бизнес требования к системе репликации



Бизнес требования к системе репликации



Бизнес требования к системе репликации



Бизнес требования к системе репликации



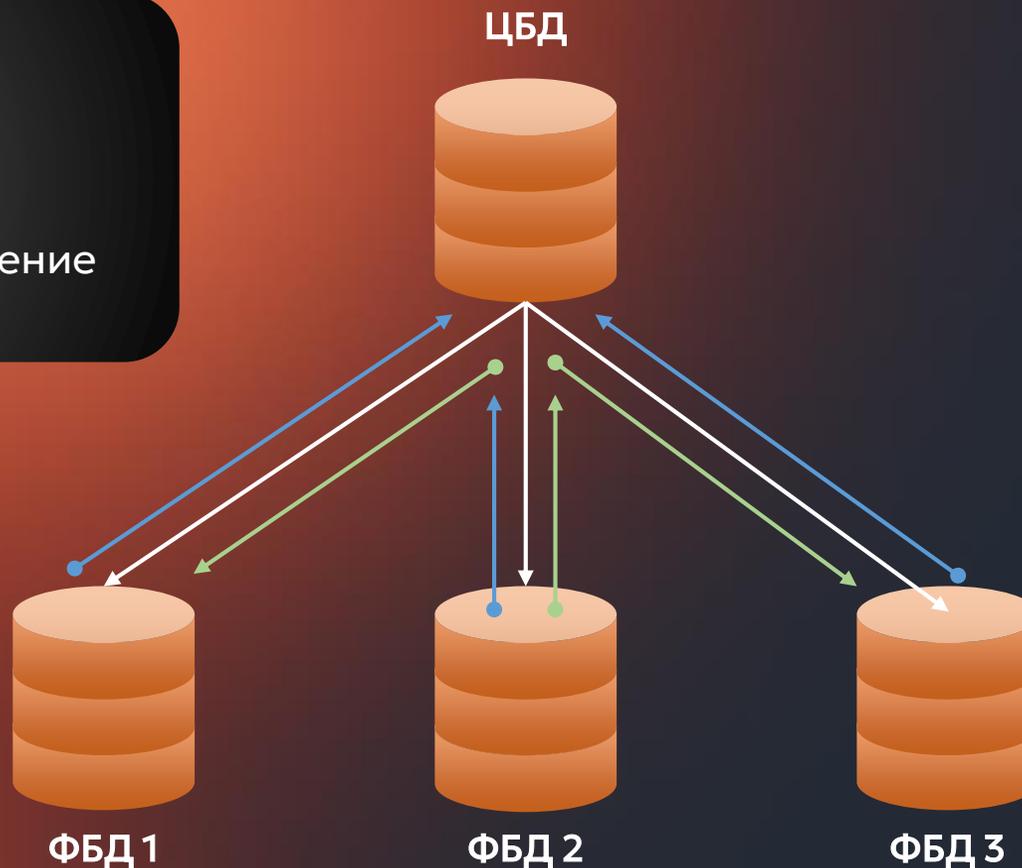
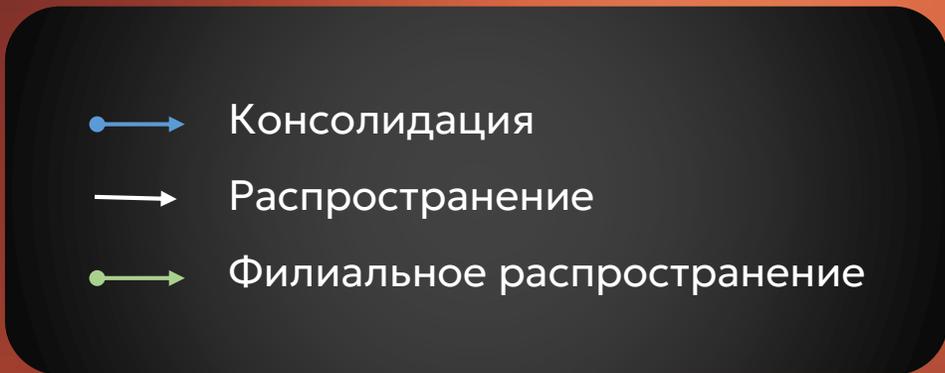
Бизнес требования к системе репликации



Бизнес требования к системе репликации



Сценарии репликации



Виды репликации

Виды репликации

Блочная репликация на уровне
системы хранения данных

Физическая репликация
на уровне СУБД

Виды репликации

Блочная репликация на уровне
системы хранения данных

Физическая репликация
на уровне СУБД

Логическая репликация

Виды репликации

Блочная репликация на уровне
системы хранения данных

Физическая репликация
на уровне СУБД

Логическая репликация

Репликация
триггерами

Виды репликации

Блочная репликация на уровне
системы хранения данных

Физическая репликация
на уровне СУБД

Логическая репликация

Репликация
триггерами

Использование
журналов СУБД

Виды репликации

Блочная репликация на уровне
системы хранения данных

Физическая репликация
на уровне СУБД

Логическая репликация

Репликация
триггерами

Использование
журналов СУБД

Использование программного
обеспечения класса CDC
(change data capture)

Виды репликации

Блочная репликация на уровне системы хранения данных

Физическая репликация на уровне СУБД

Логическая репликация

Репликация триггерами

Использование журналов СУБД

Использование программного обеспечения класса CDC (change data capture)

Прикладная репликация

Репликация на триггерах

Недостатки

- Дополнительная нагрузка от срабатывания триггеров
- Не реплицируется DDL
- Замедление работы при росте журналов репликации
- Сложность реализации по сравнению с готовыми решениями
- Настройка сложных схем репликации требует высокой экспертизы

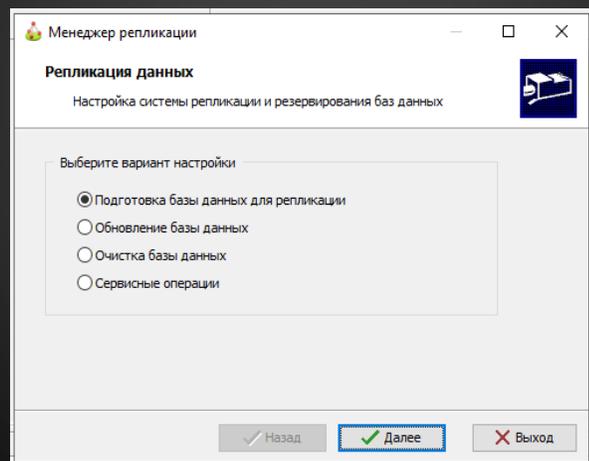
Преимущества

- Возможность выборочной репликации отдельных таблиц или записей из таблиц
- Независимость от внешних решений и/или встроенных инструментов СУБД
- Возможна реализация в любой СУБД, где есть триггеры
- Поддержка гетерогенных сред
- Возможность репликации из представлений

Архитектура ПО для настройки и выполнения репликации

1. Мастер настройки

Подготовка базы данных к репликации



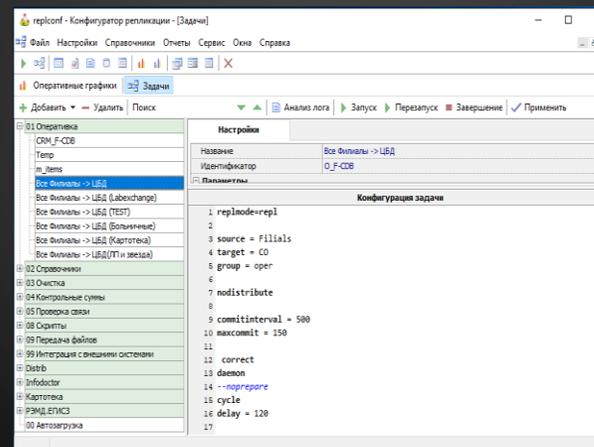
2. Конфигуратор репликации

Настройка потоков репликации



3. Сервер репликации

Передача данных



Подготовка базы данных

- 1 Создание специальной роли REPL
Создание журналов репликации и задание идентификаторов баз данных

repl\$changelog

PK	NN	Поле	Комментарий
☺	✓	ID	Первичный ключ
	✓	TABLERNAME	Наименование таблицы
	✓	REPLID	Идентификатор суррогатного ключа
	✓	REPLGRPID	Идентификатор базы данных
	✓	OP	Тип операции: 0 - добавление, 1 - изменение, 2 - удаление
	✓	MODIFYDATE	Дата и время изменения
	✓	OWNGRPID	Идентификатор базы данных владельца
	✓	FROMGRPID	Идентификатор базы данных источника
	✓	TOGRPID	Идентификатор целевой базы данных. Если 0, то всем

repl\$changelogdet

PK	NN	Поле	Комментарий
☺	✓	ID	Идентификатор записи журнала
☺	✓	REPLGRPID	Идентификатор целевой базы данных
	✓	DTYPE	Тип обработки записи журнала
		MODIFYDATE	Дата и время обработки

- 2 Создание суррогатных первичных ключей

- 3 Создание триггеров

Настройка потоков репликации

1 Регистрация баз данных и создание групп баз данных

2 Создание групп таблиц

- Оперативные
- Справочные
- Звезда
- Другие

3 Создание задач репликации

- Репликация
- Дистрибуция
- Очистка
- Другие

The image shows two overlapping screenshots of the 'repconf' configuration tool. The top screenshot displays the 'Группы репликации' (Replication Groups) configuration window. On the left, a tree view shows a hierarchy starting with 'all', followed by 'labportal', which contains sub-groups 'labportal_feedback' and 'labportal_feedback_invoice'. The main panel shows the configuration for the 'labportal' group, with fields for 'Название' (Name) and 'Идентификатор' (Identifier) both set to 'labportal'. The bottom screenshot shows the 'Задачи' (Tasks) configuration window. A tree view on the left shows a hierarchy including 'Репликация Lab Portal' with sub-tasks like 'labportal_arc_sprav', 'labportal_cash', 'labportal_feedback' (highlighted in blue), 'labportal_feedback_invoice', and 'script_labportal_daily'. The main panel shows the configuration for the 'labportal_feedback' task, including a 'Конфигурация задачи' (Task Configuration) section with the following settings:

```
1 replmode = repl
2
3 noprepere
4 source = labportal_db
5 target = cash_db
6 group = labportal_feedback
7
8 maxcommit = 100
9 commitinterval = 300
10
11 cycle
12 delay = 60 -- 5 min
13 ftime = 22:00
14
15 daemon
16 btime = 8:00
17
18 logfile=log\cash_from_labportal#.txt
```

Сервер репликации



Выполнение штатной
репликации

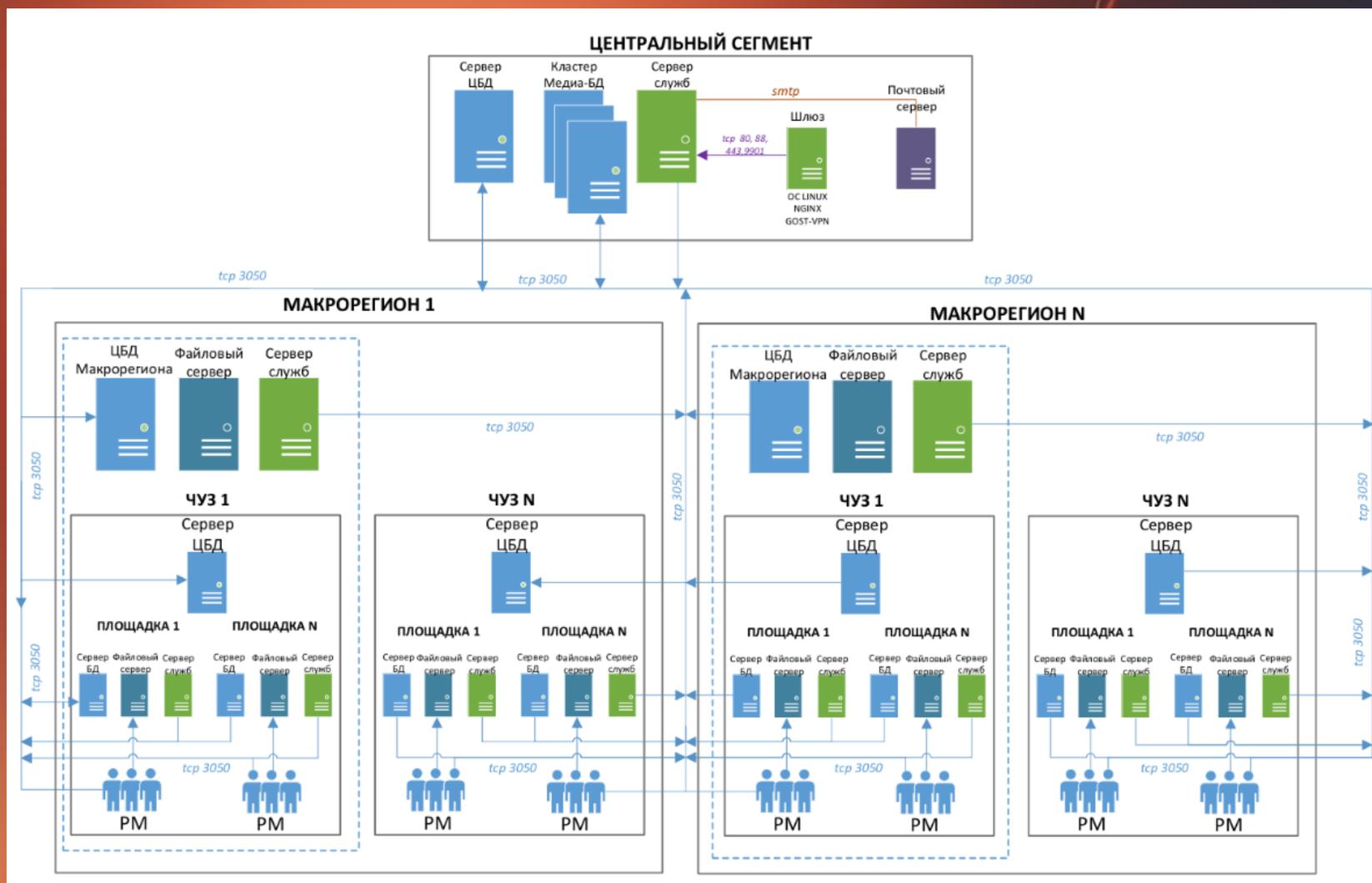


Оптимизация трафика,
подготовка журнала
репликации



Разрешение коллизий

Пример многоуровневой иерархической репликации

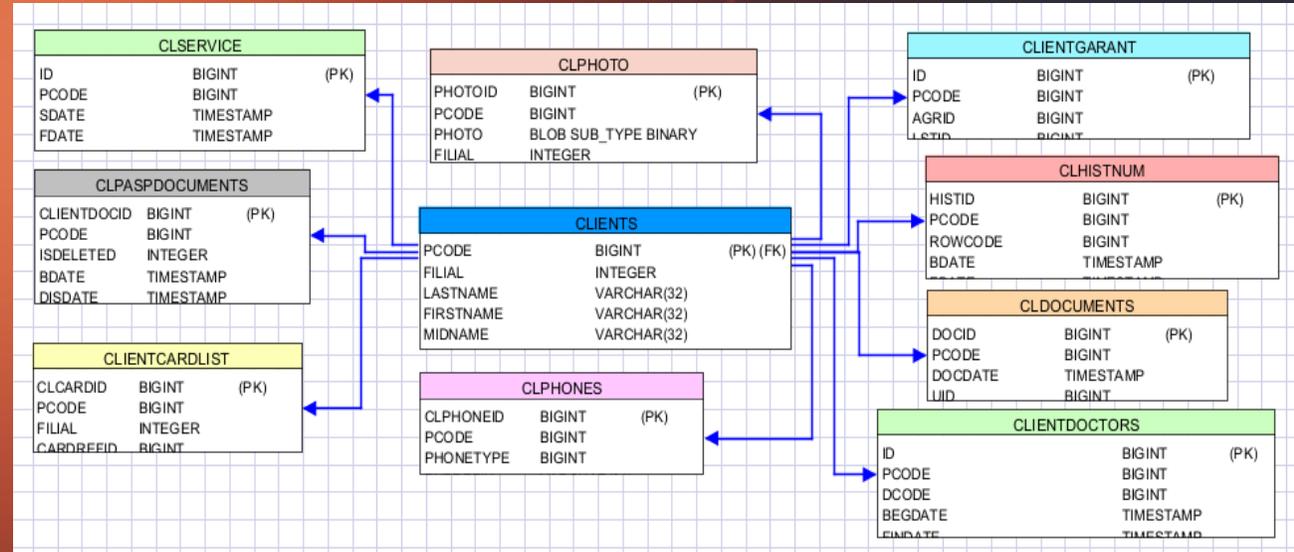


Когда логики логической репликации не хватает

Примеры комплексных сущностей



- Карточки регистрации пациента
- Листы временной нетрудоспособности
- Задачи CRM
- Направления
- и т.д.



Требования к механизму синхронизации данных



- Сравнение данных в исходной и целевой БД перед синхронизацией
- Рассматривать множество записей в зависимых таблицах как единое целое
- Реализовывать произвольную дополнительную логику на любом этапе синхронизации

Реализация логики на клиенте VS на уровне СУБД

	Клиент / Сервер приложений	Firebird
Гибкость и удобство разработки	++	+
Масштабируемость	+	-
Тестирование и отладка	+	+-
Изоляция логики	+	-
Единая логика	+-	++
Производительность	+-	+
Упрощенная транзакционная модель	-	++
Простота доставки патчей	-	+

Логика на уровне СУБД целесообразна для высокопроизводительных операций, требующих обеспечения единой логики и не требующих распределения нагрузки

Логика синхронизации

Исходная БД

Целевая БД

Исходная БД

Целевая БД

1	A
2	B



2	C
3	D



1	A
2	B

1	A
2	B
3	D

Копирование UPDATE OR INSERT из исходной в целевую

Копирование UPDATE OR INSERT из исходной в целевую со сравнением MODIFYDATE (без удаления целевых записей, отсутствующих в исходной)

1	A
2	B



2	C
3	D



1	A
2	B
3	D

1	A
2	B
3	D

Двухстороннее копирование UPDATE OR INSERT в целевую или исходную со сравнением MODIFYDATE

1	A
2	B



2	C
3	D



1	A
2	B

1	A
2	B

Копирование UPDATE OR INSERT из исходной в целевую со сравнением MODIFYDATE + удаление целевых записей, отсутствующих в исходной

Универсальная синхронизация данных через ESOE

Упрощённый пример обновления данных в целевой БД через ESOE

```
1 execute block as
2 declare id type of column table_name.id;
3 declare foreign_id type of column table_name.foreign_id;
4 declare field01 type of column table_name.field01;
5 declare field02 type of column table_name.field02;
6 declare modifydate type of column table_name.modifydate;
7 begin
8   for execute statement 'select id, foreign_id, field01, field02, modifydate from table_name where (foreign_id = 1)'
9     into id, foreign_id, field01, field02, modifydate do
10    begin
11      execute statement
12        ('update or insert into table_name(id, foreign_id, field01, field02, modifydate)
13         values (?, ?, ?, ?, ?)')
14        (id, foreign_id, field01, field02, modifydate)
15        on external 'external_server:external_db'
16        as user current_user role current_role password '';
17    end
18 end
```

Универсальная синхронизация данных через ESOE

Для универсального решения нужно просто формировать текст execute block динамически

```
-- Синхронизация из исходной в целевую
execute statement
'execute block as ' || declare_list ||
'begin ' ||
'for execute statement 'select ' || fields_list ||
' from ' || source_table_name ||
iif(condition > '', ' where (' || replace(condition, ' ', ' ') || ')', '') ||
iif(source_condition > '', iif(condition > '', ' and ', ' where ') || keyname || ' in (' || :source_condition || ')', '') || '' ||
iif(source_db > '', ' on external ' || source_db || ' as user current_user role current_role password ' || source_pass || ', ' ||
' into ' || fields_list || ' do begin ' ||
'execute statement (' || iif(sync_mode in (1, 2, 3, 4), 'update or ', '') || 'insert into ' || target_table_name || '(' || fields_list || ') ' ||
' values (' || params_list || ') ' || iif(merge_key_fields > '', ' matching (' || merge_key_fields || ')', '') ||
'' || fields_list || ') ' ||
iif(target_db > '', ' on external ' || target_db || ' as user current_user role current_role password ' || target_pass || ', ' || '' || ';' ||
'end end';
```

FS_SYNC_TABLE_DATA_COMPARE

IO	Название	Комментарий
IN	TABLE_NAME	Наименование таблицы
IN	CONDITION	Условие where
IN	KEYNAME	Наименование ключевого поля
IN	SOURCE_DB	Исходная БД
IN	TARGET_DB	Целевая БД
IN	SOURCE_PASS	Пароль к исходной БД
IN	TARGET_PASS	Пароль к целевой БД
OUT	KEYVALUE	Значение ключевого поля
OUT	SOURCE_MODIFYDATE	Дата модификации записи исходной БД
OUT	TARGET_MODIFYDATE	Дата модификации записи целевой БД

FS_SYNC_TABLE_DATA

IO	Название	Комментарий
IN	SOURCE_TABLE_NAME	Наименование таблицы исходной БД
IN	CONDITION	Условие отбора данных из таблицы (по умолчанию вся таблица)
IN	SOURCE_DB	Путь к исходной базе данных
IN	TARGET_DB	Путь к целевой базе данных
IN	SOURCE_PASS	Пароль к исходной базе данных
IN	TARGET_PASS	Пароль к целевой базе данных
IN	SYNC_MODE	Режим синхронизации:
IN	EXCLUDE_FIELD_LIST	Список полей, которые необходимо исключить из копирования или синхронизации
IN	KEYNAME	Название ключевого поля для режима двухсторонней синхронизации (обязательно)
IN	MERGE_KEY_FIELDS	Список ключевых полей в условии MATCHING для режима копирования или синхронизации
IN	TARGET_TABLE_NAME	Наименование таблицы целевой БД

Преимущества использования ESOE

- Оптимально с точки зрения скорости выполнения и утилизации ресурсов, так как нет других потребителей кроме СУБД. При реализации на клиенте или сервере приложений текущие запросы к БД в любом случае остались бы
- Минимальная нагрузка на каналы связи, так как взаимодействие только между серверами СУБД
- Управление пулом подключений на стороне СУБД
- Простота переиспользования без необходимости использовать дополнительные библиотеки. Одни и те же процедуры вызываются в клиентском приложении при интерактивной синхронизации и в службе репликации при фоновой синхронизации по преднастроенным сценариям
- ESOE также используется в обработке запросов web-сервиса к филиальным базам данных, выборках данных из внешних БД, кластеризации медиа-БД и других задачах

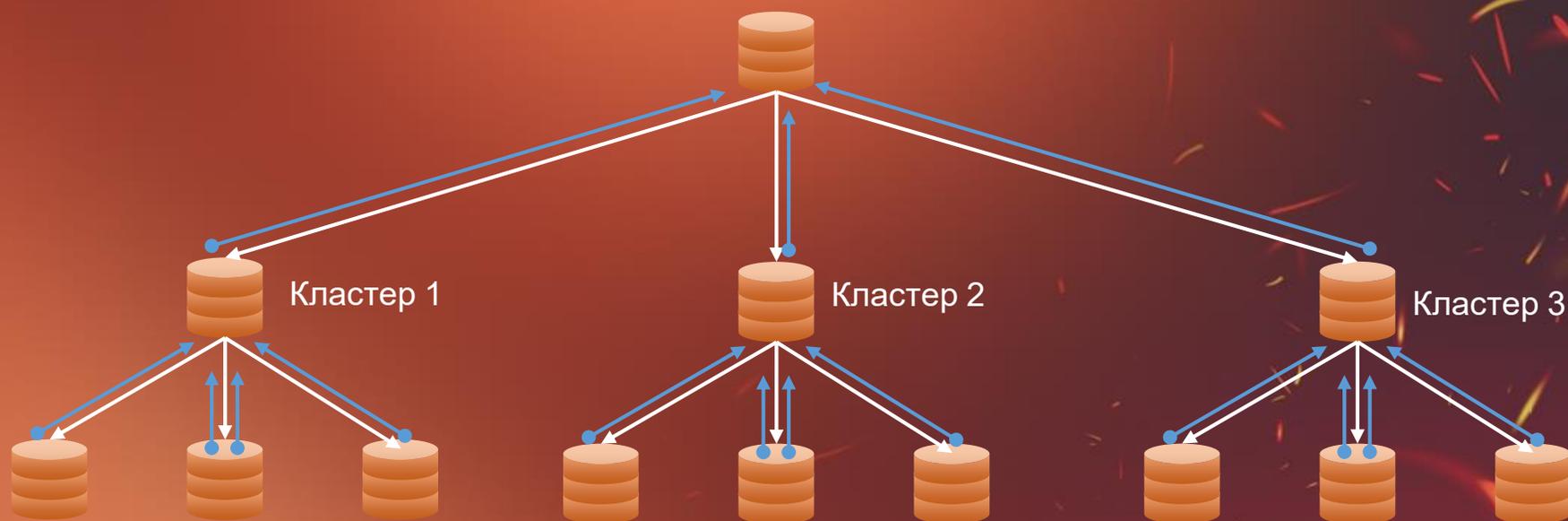
Цифровая трансформация бизнеса в работающей единой информационной системе

Задача

Быстрая информатизация территориальных кластеров без унификации НСИ между кластерами

Проблема

Что потом с этим делать?



Универсальный механизм дедубликации данных

Дано:

- Более 1200 таблиц БД
- Более 350 справочных таблиц
- Метаданные и связи между таблицами описаны

Представление связей между таблицами в реляционном виде

PK	NN	Поле	Комментарий
✓		TABLE_NAME	Название таблицы
✓		FIELD_NAME	Название поля
✓		MAIN_TABLE_NAME	Название основной таблицы
✓		MAIN_FIELD_NAME	Название поля в основной таблице
✓		MAIN_CONDITION	Условие по основной таблице
		CONDITION	Условие, при котором поле ссылается на данную основную

Реализация хранимой процедуры S_GET_DEPENDENT_TABLES для рекурсивного поиска зависимых таблиц

Реализация интерфейса для формирования заданий на дедубликацию

Реализация хранимой процедуры S_RECORD_JOIN для универсального объединения двух произвольных записей таблицы и каскадной перекодировки внешних ключей в зависимых таблицах

Рассылка заданий на дедубликацию по базам данных единой информационной системы

Выполнение заданий по дедубликации в каждом территориальном подразделении

Эффективное использование ресурсов благодаря всем вычислениям на уровне СУБД

Выводы и заключение

Это работает

Текущий инструмент репликации прост в настройке и администрировании, легко масштабируется и выдерживает нагрузку нескольких сотен одновременных потоков репликации.

Благодаря многолетнему опыту эксплуатации и отладки на крупных объектах достигнута высокая стабильность

Использование PSQL – не антипатерн

Синхронизация с помощью ESOE дополняет логическую репликацию и позволяет решать сложные бизнес задачи с небольшими затратами на настройку и последующее администрирование.

Благодарим за внимание!

29 МАЯ | МОСКВА

FIREBIRD  CONF

2025