Microsoft

skype™

Архитектура платформы баз данных и опыт администрирования PostgreSQL в Skype

Алексей Плотников

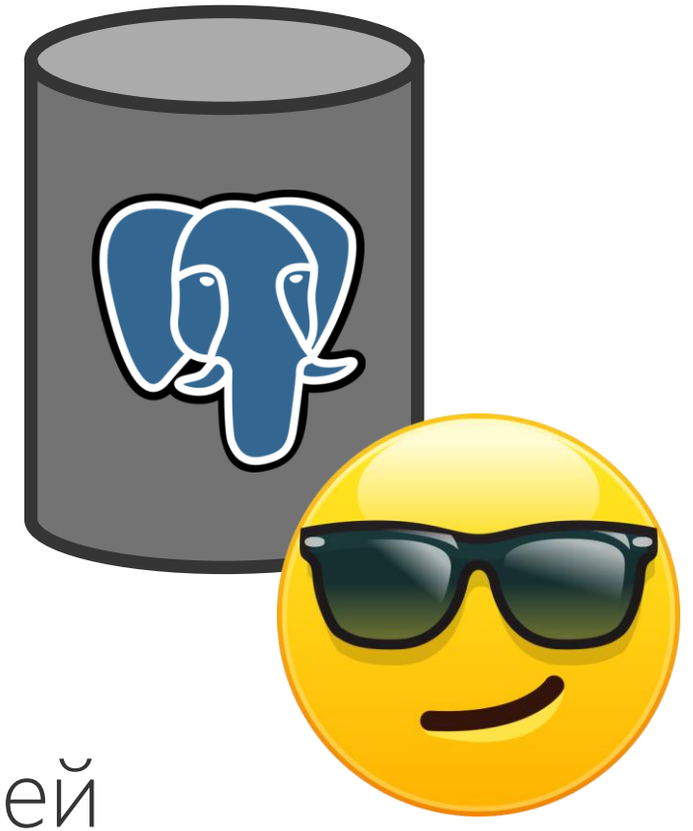
PgConf.Russia 2017

Кто?

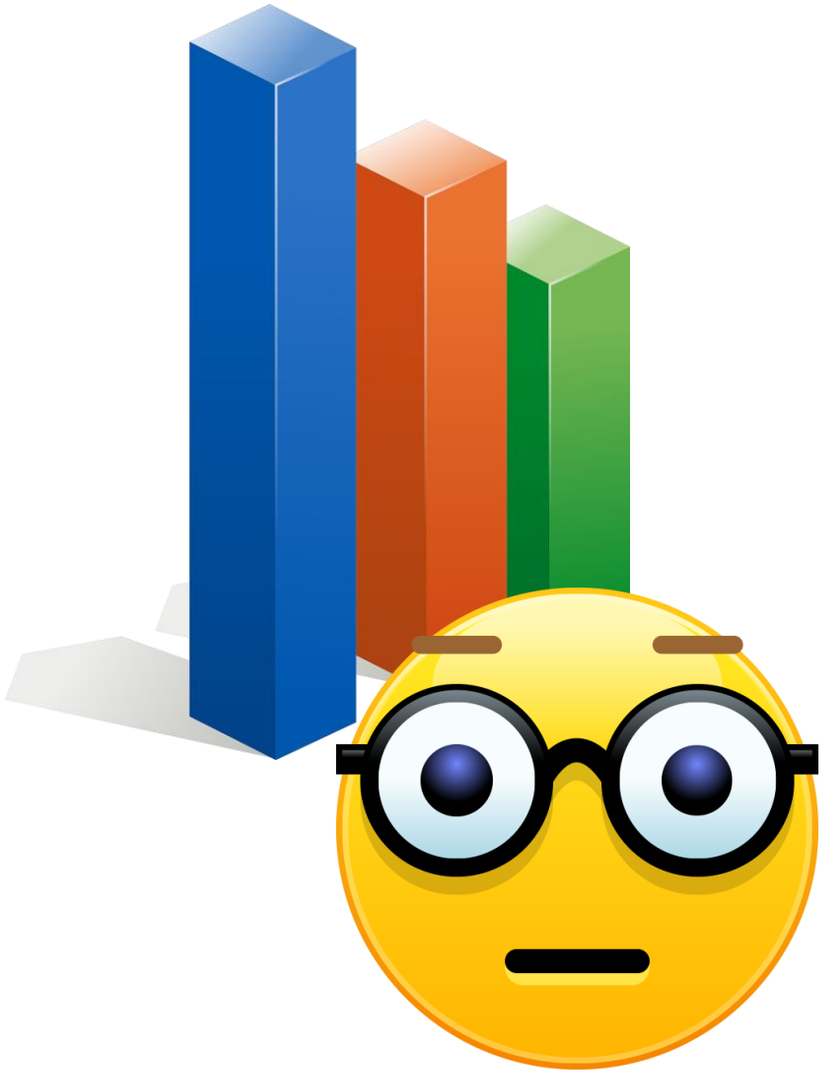
- Пользователь PostgreSQL с версии 8.2
- Senior Service Engineer @ Skype
- Skype Database Platform team

PostgreSQL в Skype

- Skype for Consumer
- История
  - Выбор PostgreSQL
  - Быстрый рост
  - Microsoft
- Когда используется PostgreSQL
- Сотни миллионов активных пользователей



Статистика платформы БД



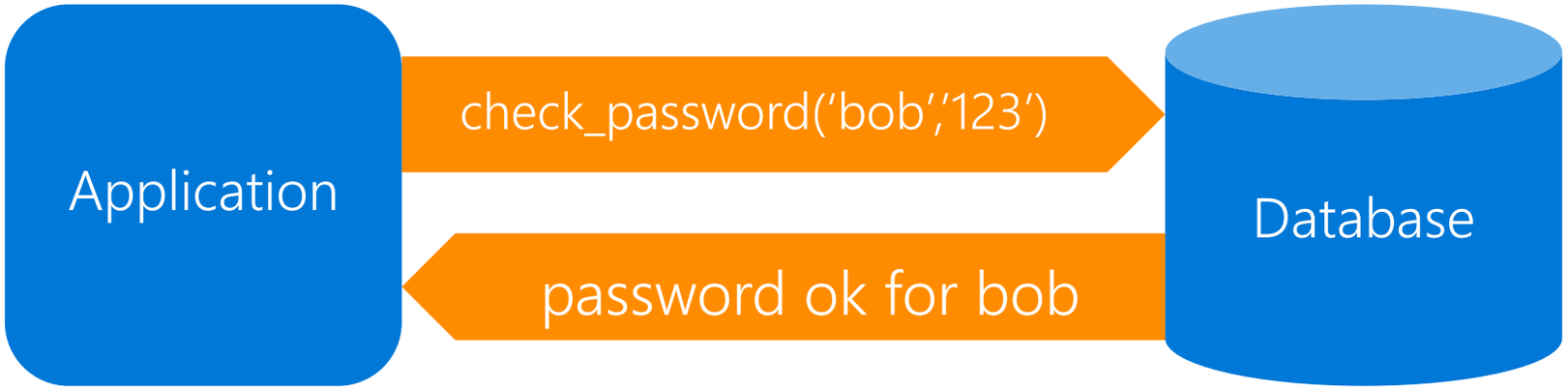
- 160 логических баз данных
- 2000+ физических экземпляров БД
- Около 1000 серверов
- Больше 200k транзакций в секунду
- Почти 500 TB объем данных

Инфраструктура платформы БД

- 2 датацентра
- Hyper-V виртуализация
- SAN
- Debian Linux
- PostgreSQL 9.4

Логическая архитектура

- Database as a Service
- Stored Procedure API
- Логическая точка доступа

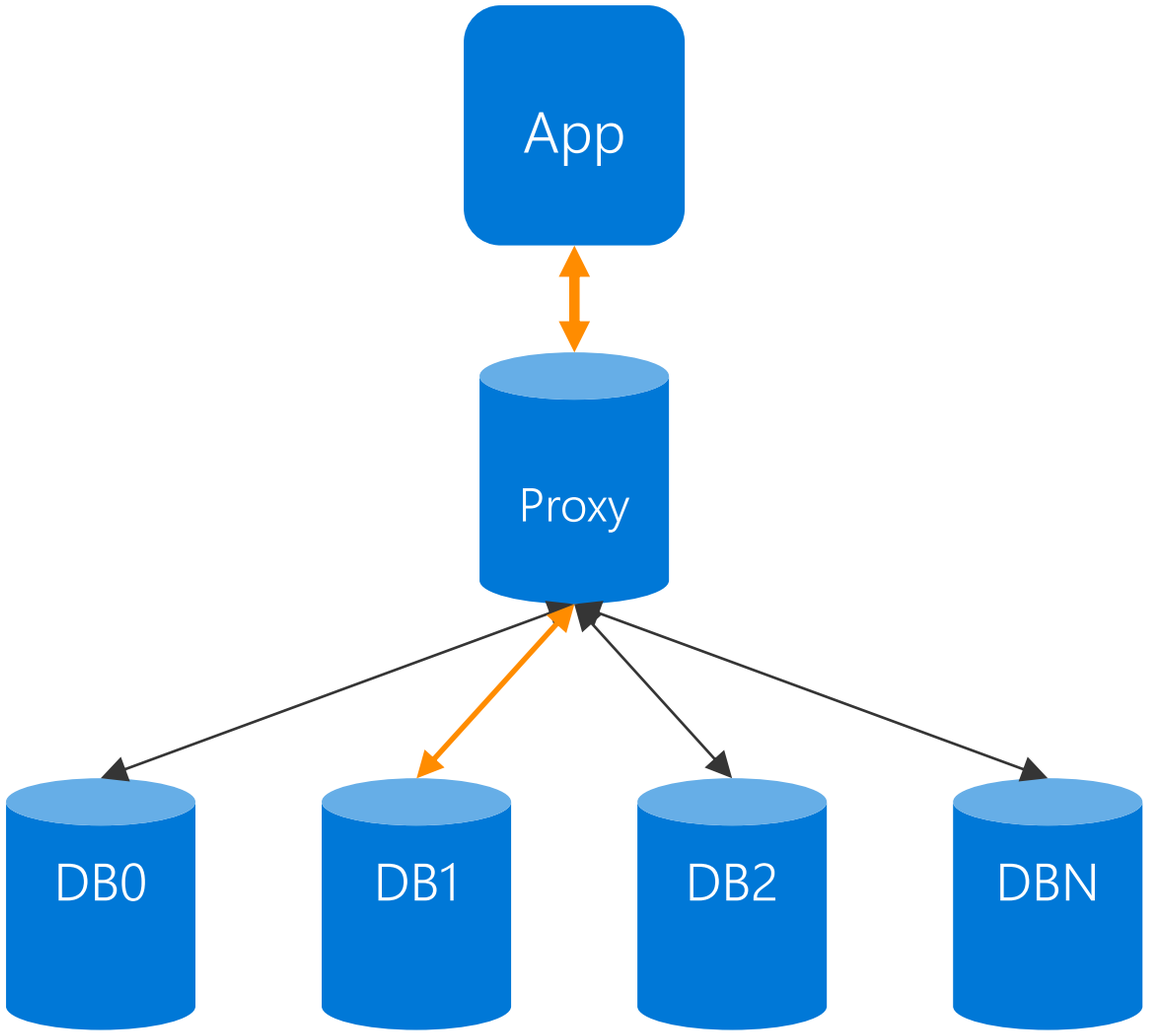


Хранимые процедуры

- Существенное ограничение, но имеет много плюсов
  - Бизнес-логика, касающаяся работы с данными, выполняется в БД
  - Упрощенная модель безопасности
  - Прозрачная разработка и обслуживание БД
- Статистика
  - Около 20 тысяч "логических" функций
  - Больше миллиона строк кода

Кластер БД

- Два уровня
  - Шарды с данными
  - Прокси БД
- Удаленный вызов ХП

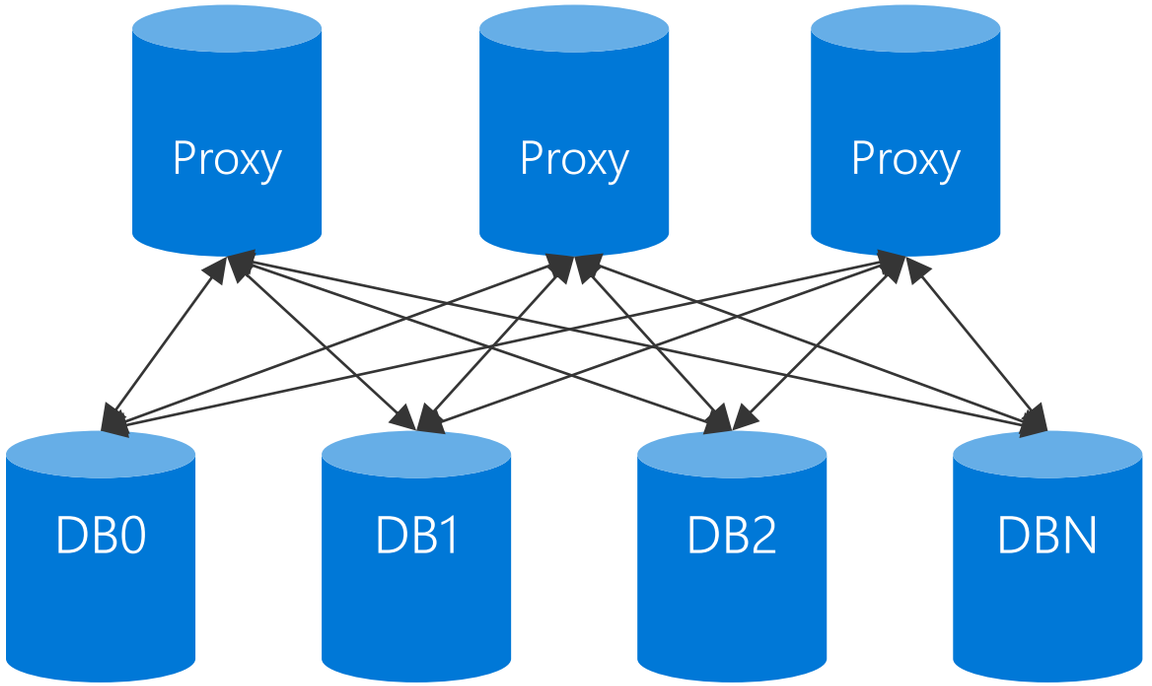


PL/Proxy

- Процедурный язык для удаленного вызова ХП
  - Open source https://plproxy.github.io/
- Прост в настройке и использовании
  - Прокси функции
  - Правила для выбора удаленной БД
  - Удаленная функция с такой же подписью
- Поддержка шардинга
  - Динамический выбор БД
  - Хэширование аргументов
  - Неограниченная масштабируемость
  - Количество шардов - степень двойки
  - Самый большой кластер – 256 шардов
  - Решардинг
- Горизонтальные RPC

Масштабирование прокси БД

- Легко масштабировать
  - Нет данных, только функции и конфигурация
  - Идентичные копии
- Балансировка нагрузки
  - DNS Round-Robin
  - Автоматическое управление

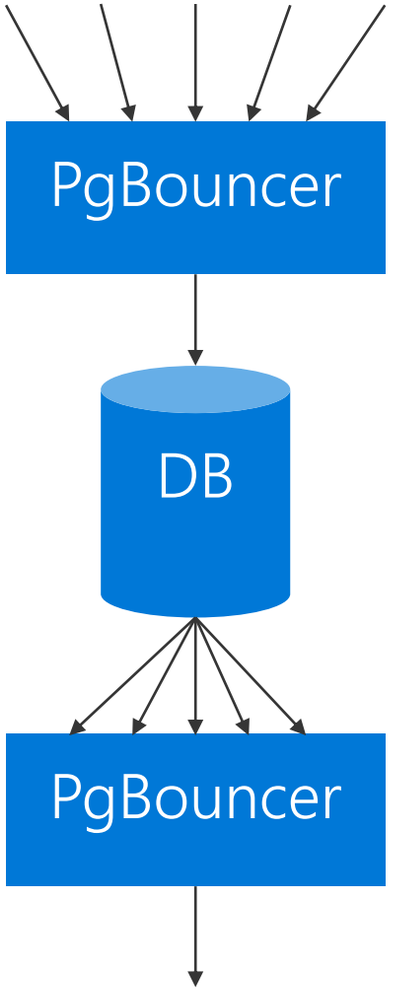


PgBouncer

- Очень эффективный и быстрый pooler подключений к БД
- Помогает сократить количество соединений
- Экономит ресурсы ОС
- Минимизирует время подключения
- Использует мало ресурсов, не обрабатывает пакет полностью
- Session, transaction, statement режимы
- Open source https://pgbouncer.github.io/

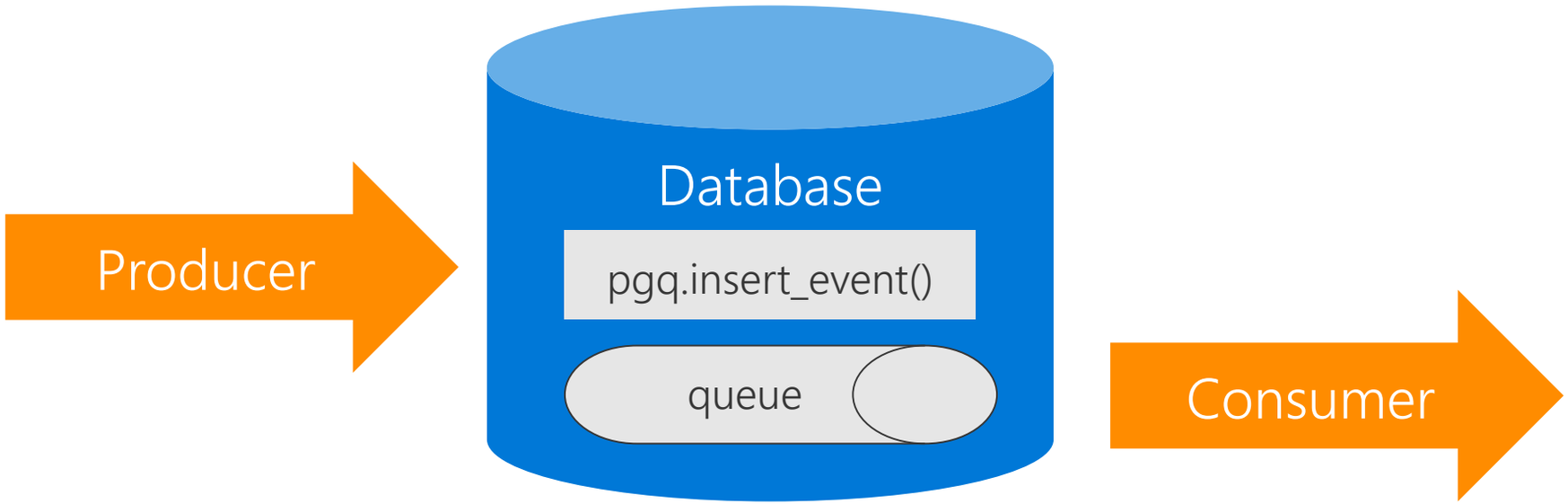
PgBouncer

- Эффективная работа с DNS
  - IP caching
  - Внутренний DNS Round-Robin
  - Проверка изменений зоны
- Используем для входящих и для исходящих соединений



PgQ

- Система очередей PostgreSQL, написанная на PL/pgSQL и C
- Асинхронная обработка событий, основанная на механизме транзакционных снимков

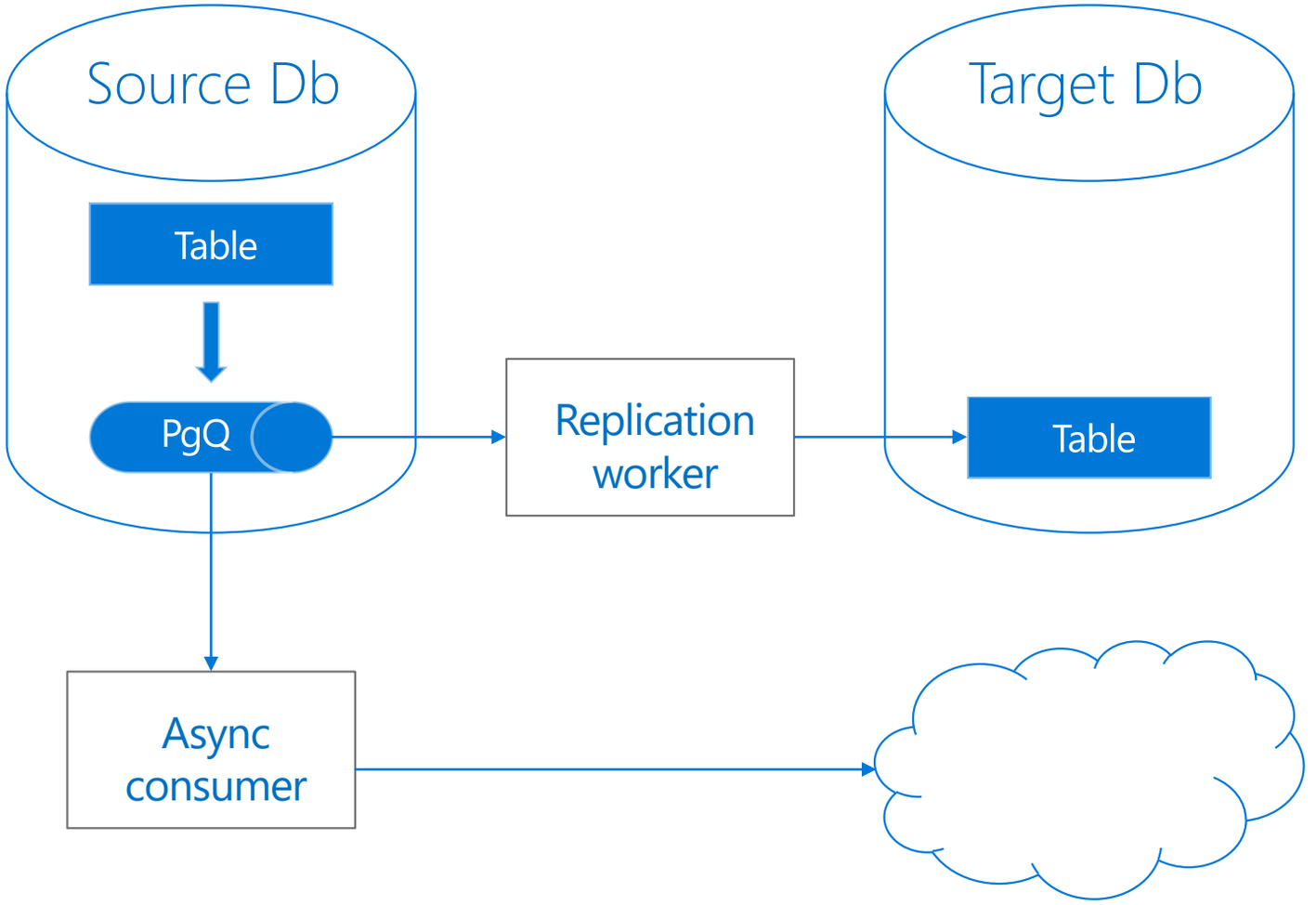


PgQ компоненты

- Producers
  - Приложение напрямую через вызов функции
  - Insert/Update/Delete/Truncate триггер на таблице
- Consumers
  - Каждый консумер гарантировано увидит эвент как минимум один раз
  - Слежение за обработанными эвентами или их повторная обработка
  - Кооперативные консумеры и каскадные консумеры
- Ticker pgqd
  - Внешний служебный даемон
  - Генерирует ticks для «нарезания» групп эвентов – «батчей» для повышения производительности
  - Выполняет обслуживание PgQ очередей
- Таблицы очередей

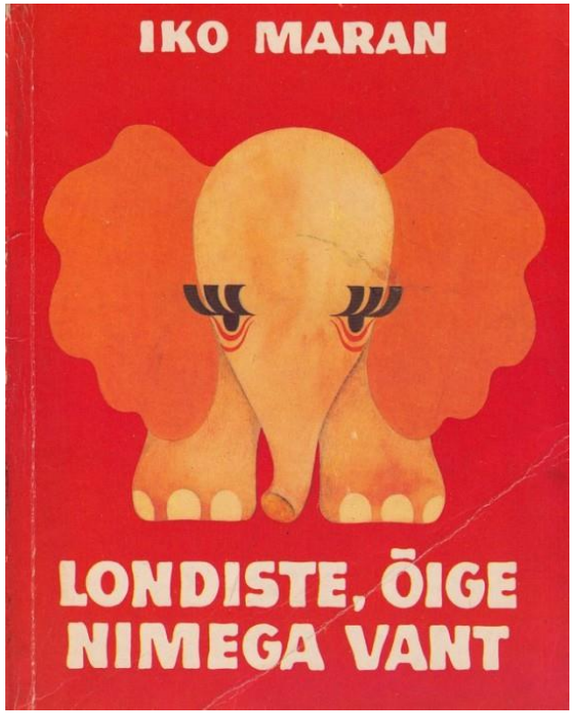
PgQ примеры использования

- Асинхронная обработка данных
- Репликация



Londiste3

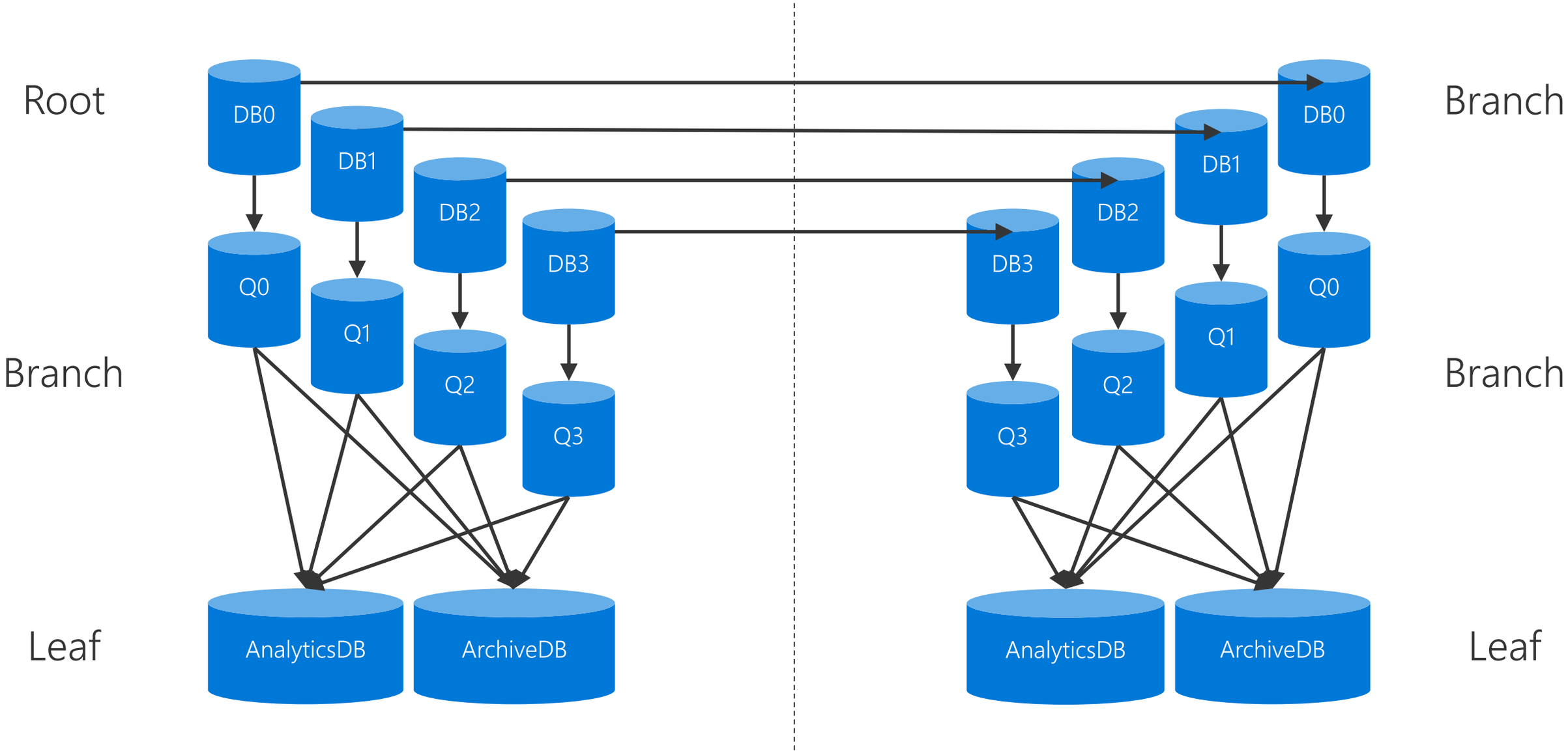
- Система репликации, основанная на использовании триггеров
- Асинхронная логическая репликация
- Использует PgQ в качестве транспортного механизма
- Сложный PgQ consumer
- Таблицы и sequence
  - Primary key
  - Автоматическое создание при добавлении
  - Нет синхронизации изменений структуры



Каскадная репликация

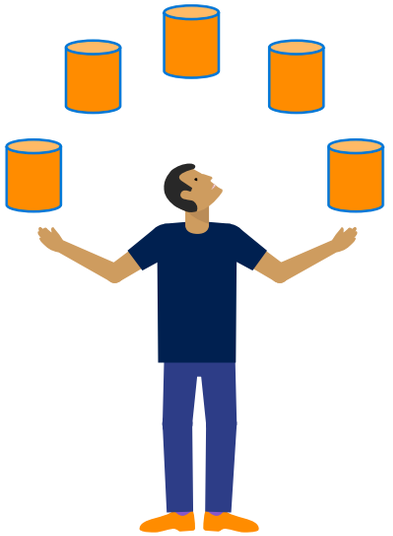
- «Цепочки» из реплик
- PgQ позволяет реплицировать структуру и данные очередей
- Типы узлов каскада (nodes)
  - Root
  - Branch
  - Leaf
- Queue nodes

Пример топологии каскадов кластера



Управление топологией каскада

- Онлайн изменения топологии
- Смена провайдера
- Takeover
  - Ситуация, когда один узел берет на себя функции другого
  - Можно «забрать» всех консумеров
- Switchover
  - Переключение главного узла каскада
  - Deny триггеры запрещают изменения на остальных узлах
  - Перенаправление трафиика
- Failover
- Resurrect
  - «Возвращает» старый root в каскад в качестве branch
  - Нереплицированные эвенты в JSON файле



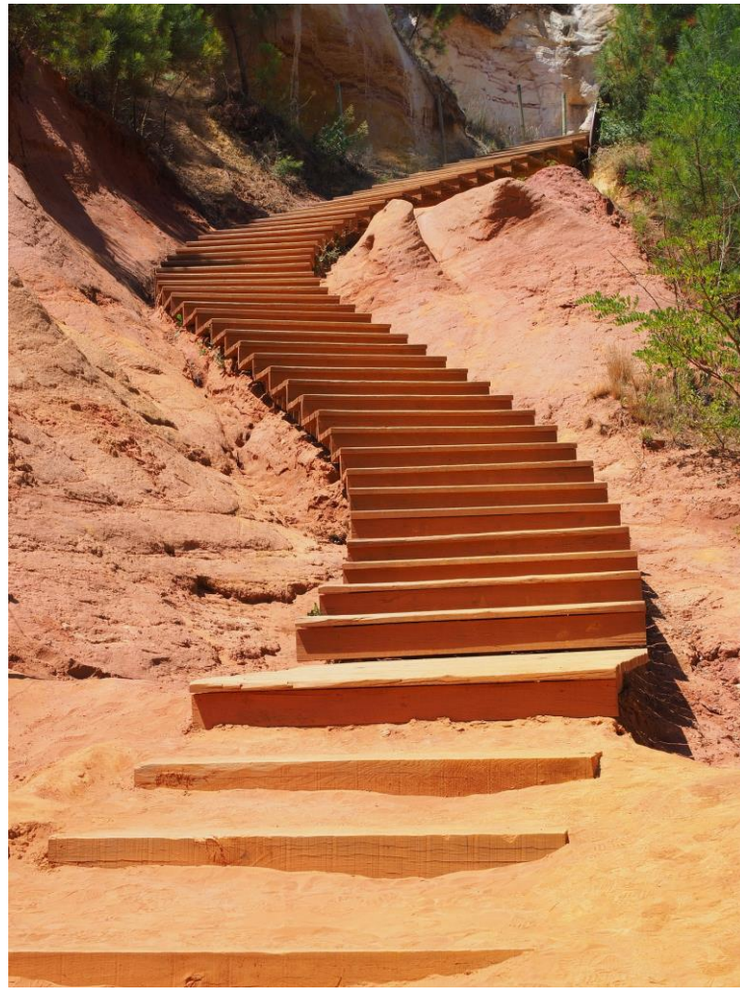
Дополнительные возможности

- Handlers – методы обработки эвентов
  - Секционирование данных
  - Разделение и слияние данных шардов
  - Пропуск определеных столбцов
  - Исправление ошибок декодирования UTF8
- Проверка и синхронизация данных
- Объединение каскадов

Примеры использования Londiste

- Копирование данных из онлайн БД во внутренние для различной аналитики и архивации
- И наоборот
- Read-only базы данных для распределения нагрузки
- Создание копий БД в другом датацентре
  - Восстановление при сбоях
  - Тяжелые DDL
  - Обслуживание БД

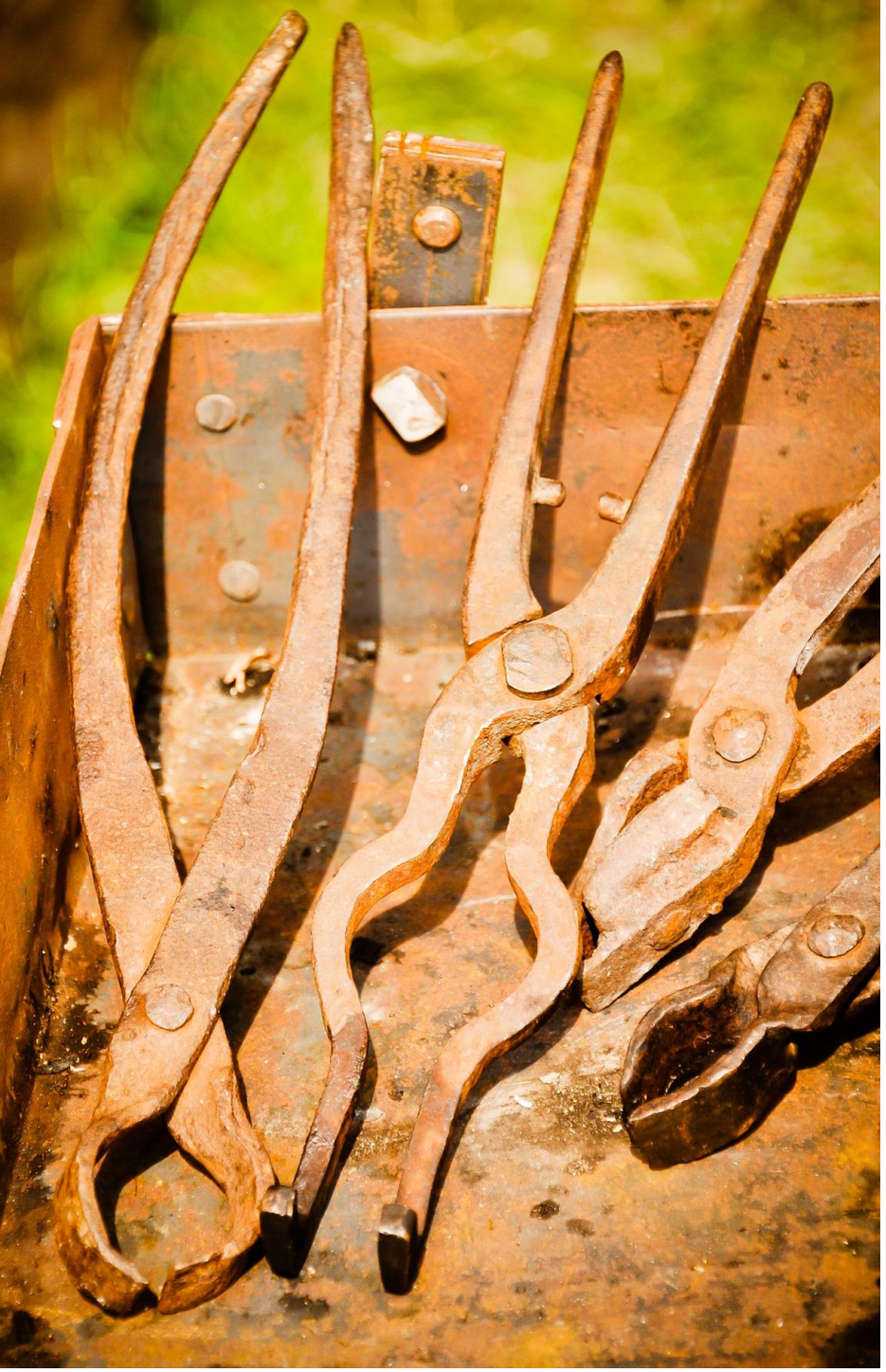
Обновление major версии PostgreSQL



- Предварительное тестирование
- Две копии БД, основная и вторичная
- Londiste репликация между ними
- Обновляем вторичную
  - pg_upgrade --link
- Переключаемся на новую версию
  - londiste3 takeover
  - Перенаправляем приложения
  - Проверяем
- В случае проблем, переключаемся назад
- Обновляем основную БД и переключаемся тем же способом

Сложности использования Londiste и PgQ

- Требуют внимания
- Обработка некорректных данных
- Лаг
- Различия в структуре баз данных
- Сложности с большими батчами

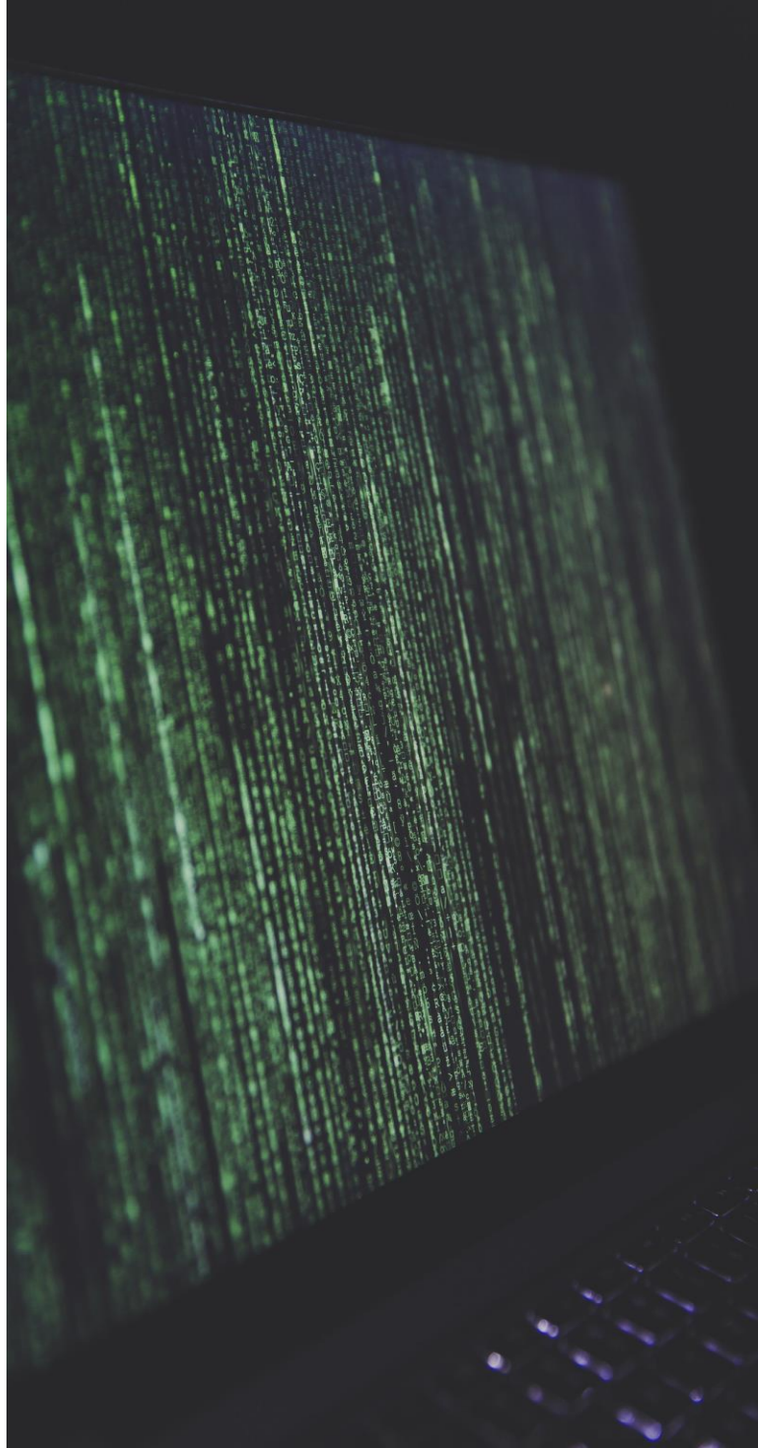


Skytools3

- Пакет утилит и технологий для работы с PostgreSQL
- PgQ, Londiste и другие
- Open source проект
- Python framework

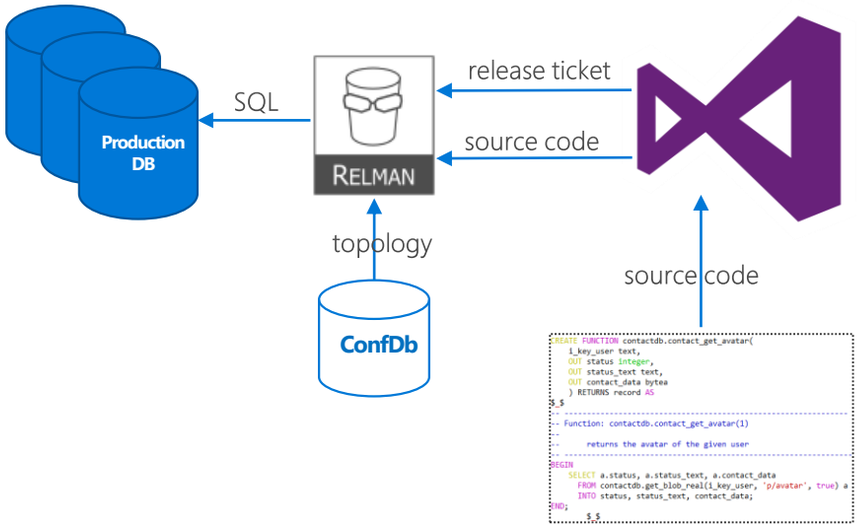
Мониторинг

- Агент для сбора информации о БД
  - Структура
  - Настройки
  - Конфигурация и статус репликации
  - Статистика использования
- Агент для работы с логами
  - Отправка логов на центральное хранилище
  - Ошибки
  - Сбор статистики
- Центральная база данных конфигурации
  - Информация для автоматизации процессов
  - Генерирование сообщений в системах мониторинга



Управление обновлениями кода БД

- до 200 релизов БД ежемесячно
- Relman
  - Система автоматизации выкатки кода
  - Декларативное описание объектов
  - Поддержка работы со всеми компонентами платформы
  - Интеграция с Visual Studio Team Services
- Автоматическая выкатка релизов
  - Проверки на соответствие требованиям
  - В более чем 90% случаев
  - Остальные при помощи DBA
- Влияние на DBA комманду



Среды разработки и тестирования

- Цель – 100% соответствие Production
- Регулярный процесс синхронизации
- Возможность создания новых сред по востребованию
- Radoslav Glinsky - «Test environment on demand»

Мои контактные данные

- Skype: agent_persik
- E-mail: aleksei.plotnikov@skype.net