Ростелеком

Из Oracle в Postgres по российским рельсам. Неочевидные нюансы

Левченко Никита Андреевич
Плотников Юрий Александрович

Технологии возможностей

Импортозамещение

Директива правительства перехода на отечественное ПО

Основные направления импортозамещения: Офисное ПО, Системный стек, ПО работы с данными, Прикладное ПО

Определение приоритетных ИС для перевода на отечественный стек

Внедрение отечественного решения не нарушает бизнес-процессы.

Постоянный процесс тестирования совместимости отечественных решений, в том числе с различными интеграциями

Ростелеком
Технологии возможностей

Почему Postgres Pro?

- Major Contributors
- Авторские русскоязычные курсы и русскоязычная документация
- Техническая поддержка
- Сертифицированный ФСТЭК дистрибутив
- Дополнительные фичи и модули
- Реестр отечественного ПО

Ростелеком
Технологии возможностей


С самого начала у нас была
какая-то архитектура

Классификация ИС по уровню доступности в соответствии с Технической политикой

Office Production

- Режим работы – 8/5
- SLA – 95%
- RTO – 24 часа
- RPO – 7 суток
- Георезерв – нет

Business Operational

- Режим работы – 8/5
- SLA – 98%
- RTO – 24 часа
- RPO – 24 часа
- Георезерв – нет

Business Critical

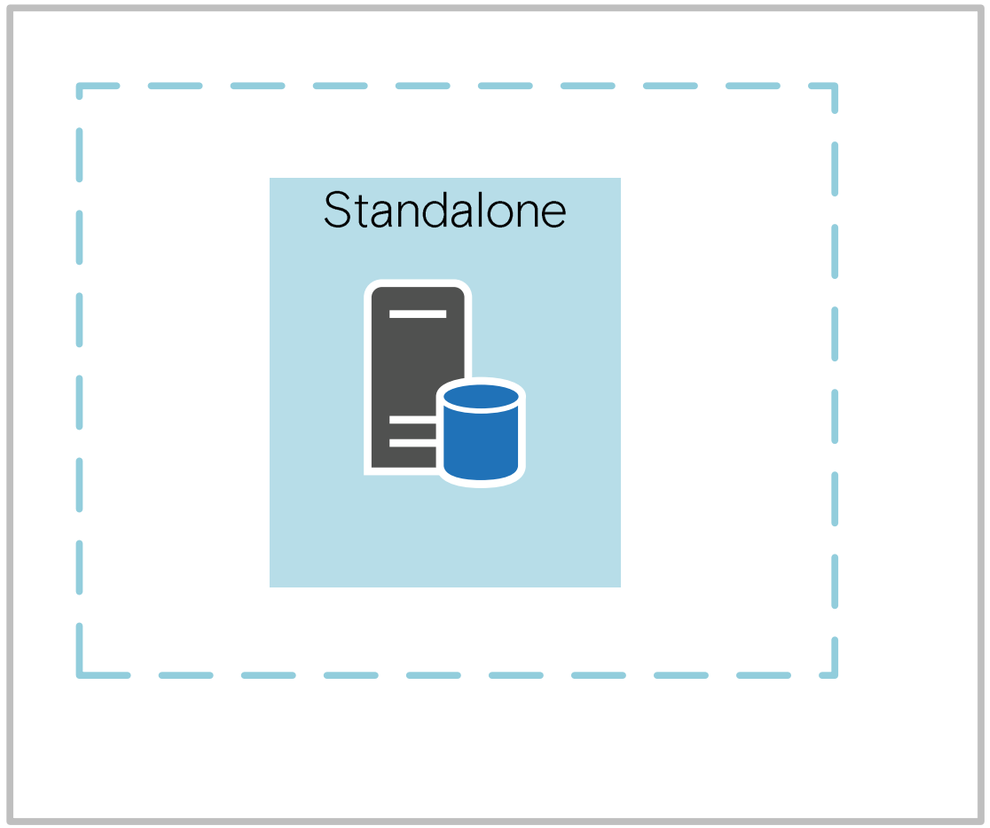
- Режим работы – 24/7/365
- SLA – 99,5%
- RTO – 1 час
- RPO – 12 часов
- Георезерв – да

Mission Critical

- Режим работы – 24/7/365
- SLA – 99,995%
- RTO – 30 минут
- RPO – 3 часа
- Георезерв – да

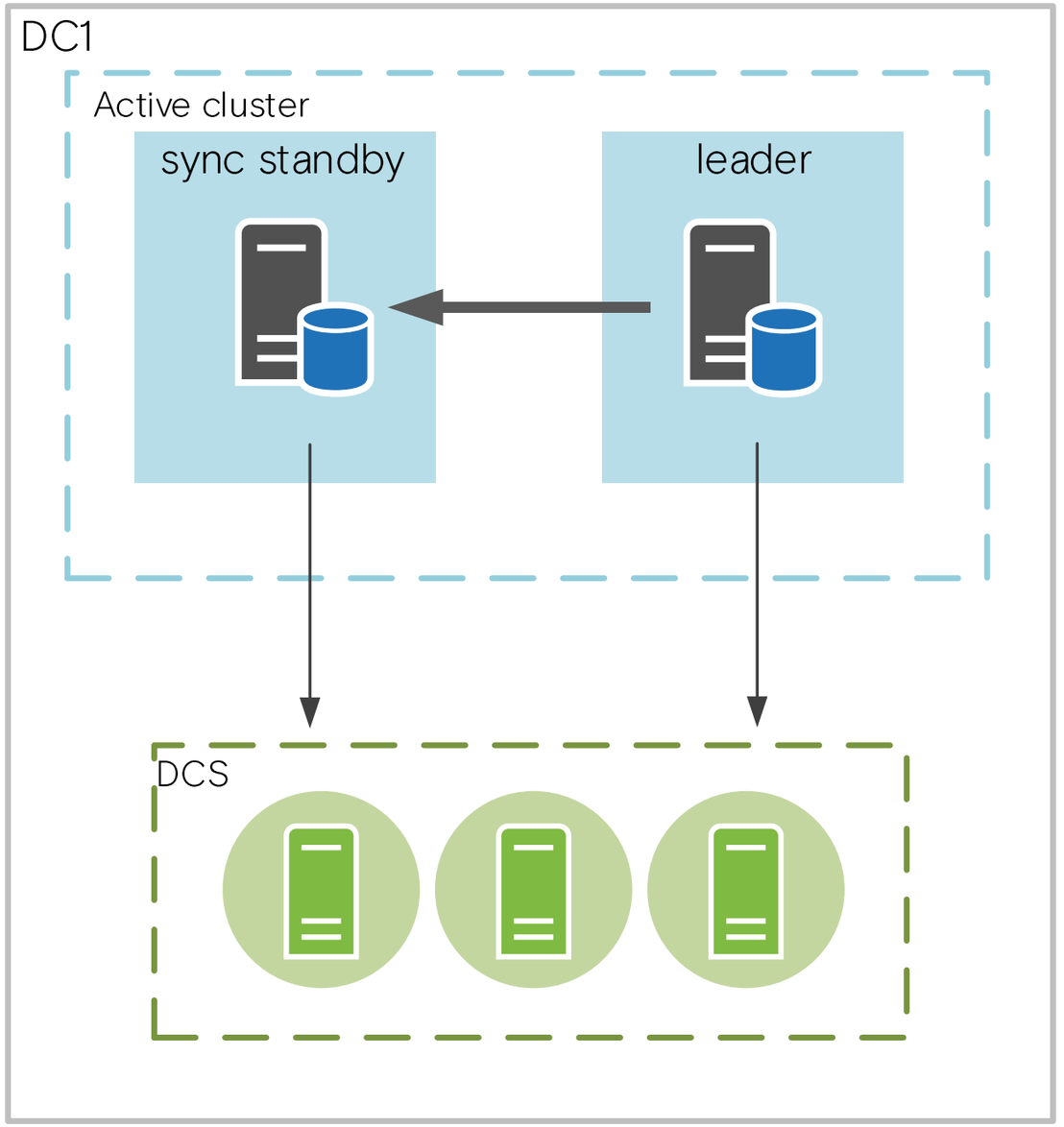
Ростелеком
Технологии возможностей

Office Production



Ростелеком

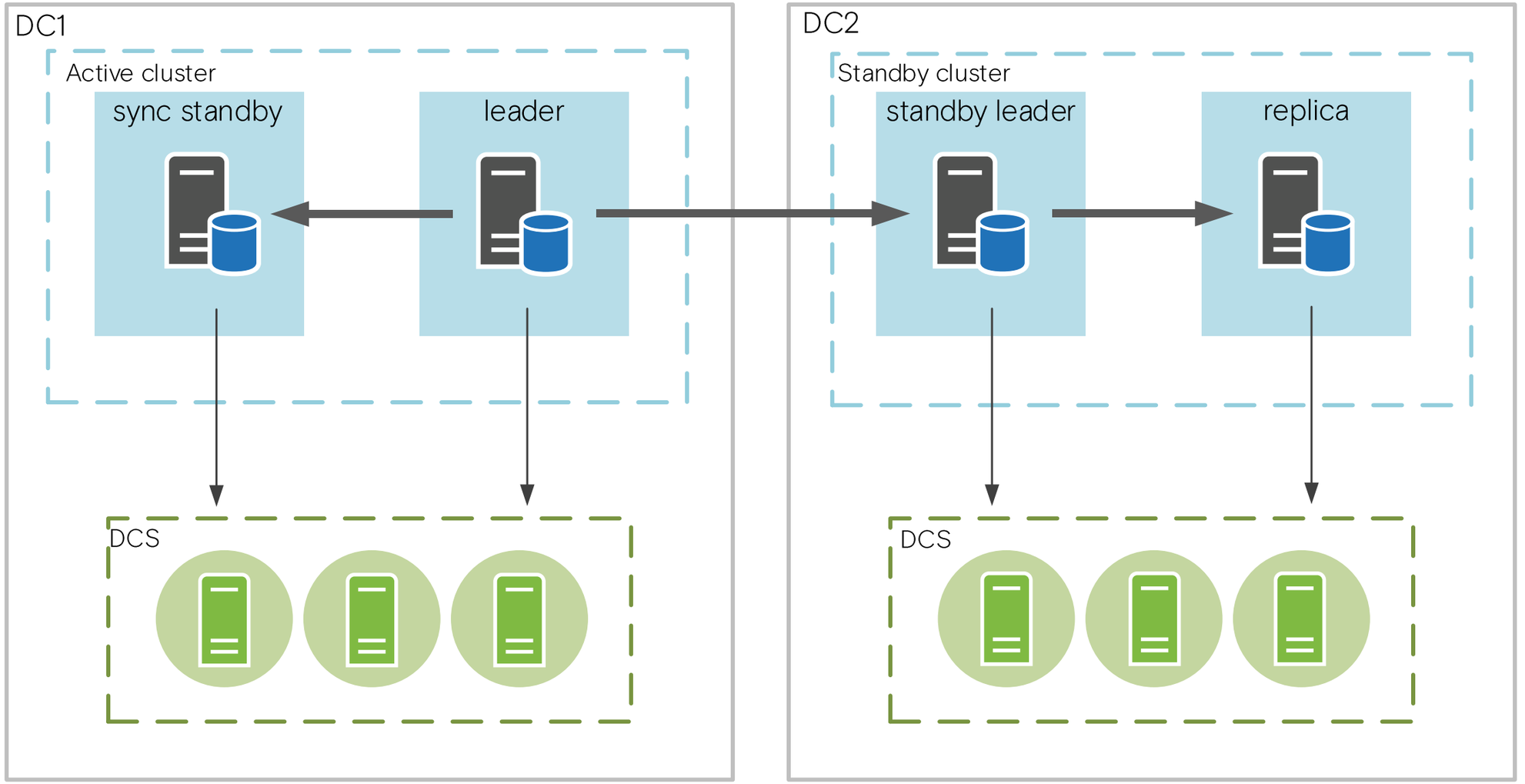
Business Operational



Ростелеком

Business Critical

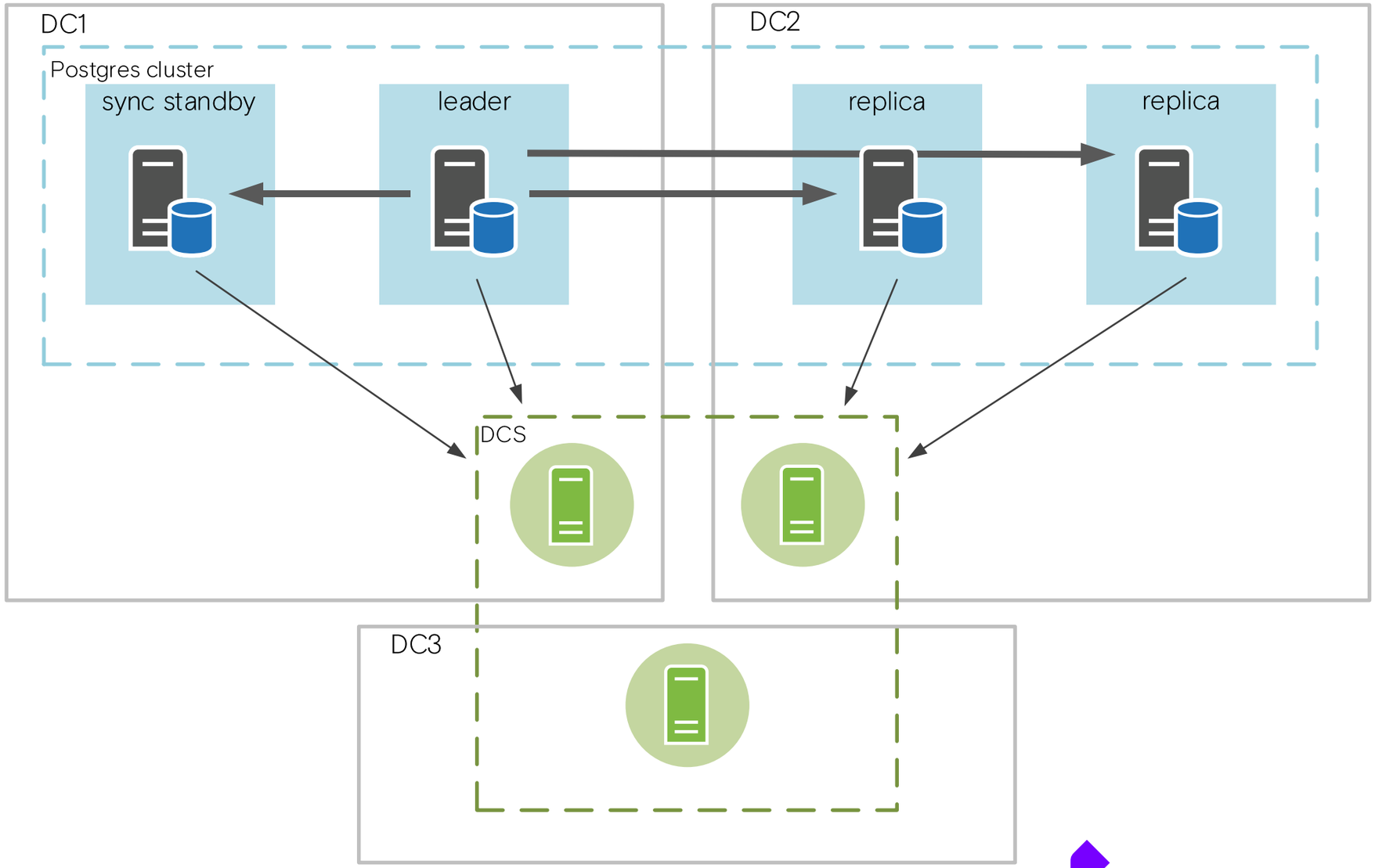
- Каскадная репликация
- Асинхронная репликация на резервной площадке
- Выделенный NFS на каждую площадку
- Синхронизация NFS между площадками
- Приложение Active/Passive
- Автоматизированное переключение площадок скриптом



Ростелеком

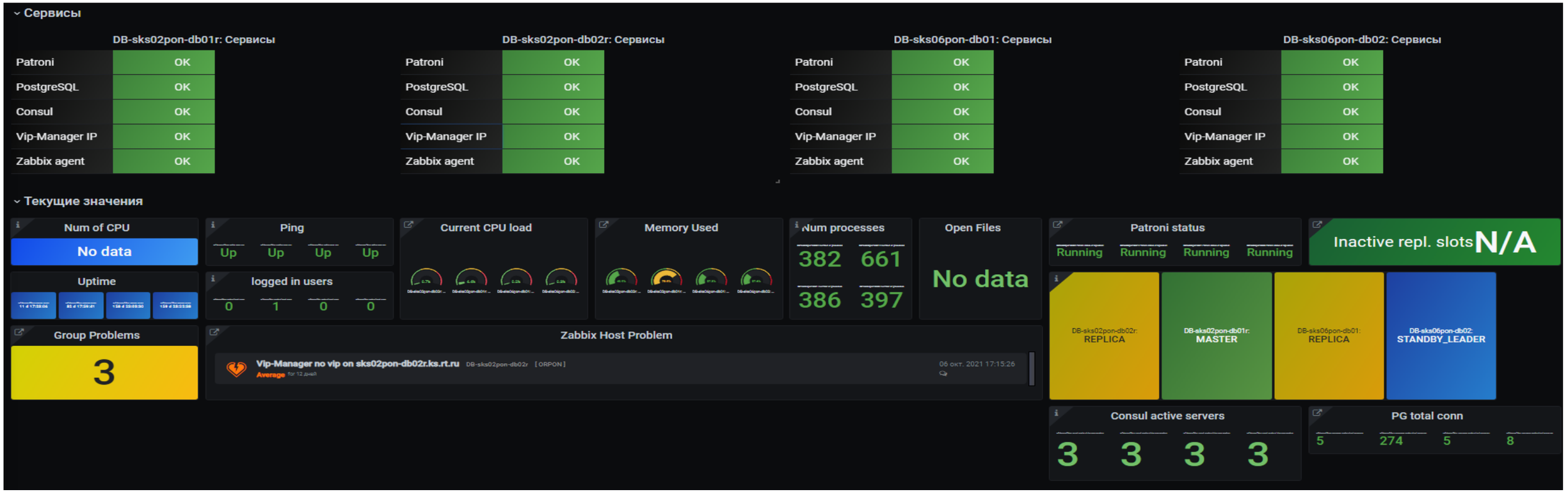
Mission Critical

- Дорогостоящая инфраструктура
- На соседней площадке sync/async реплики
- Выделенный NFS на двух основных площадках
- Синхронизация NFS между площадками
- Низкий latency сети
- QoS
- Приложение Active/Active
- Autofailover площадок



Ростелеком

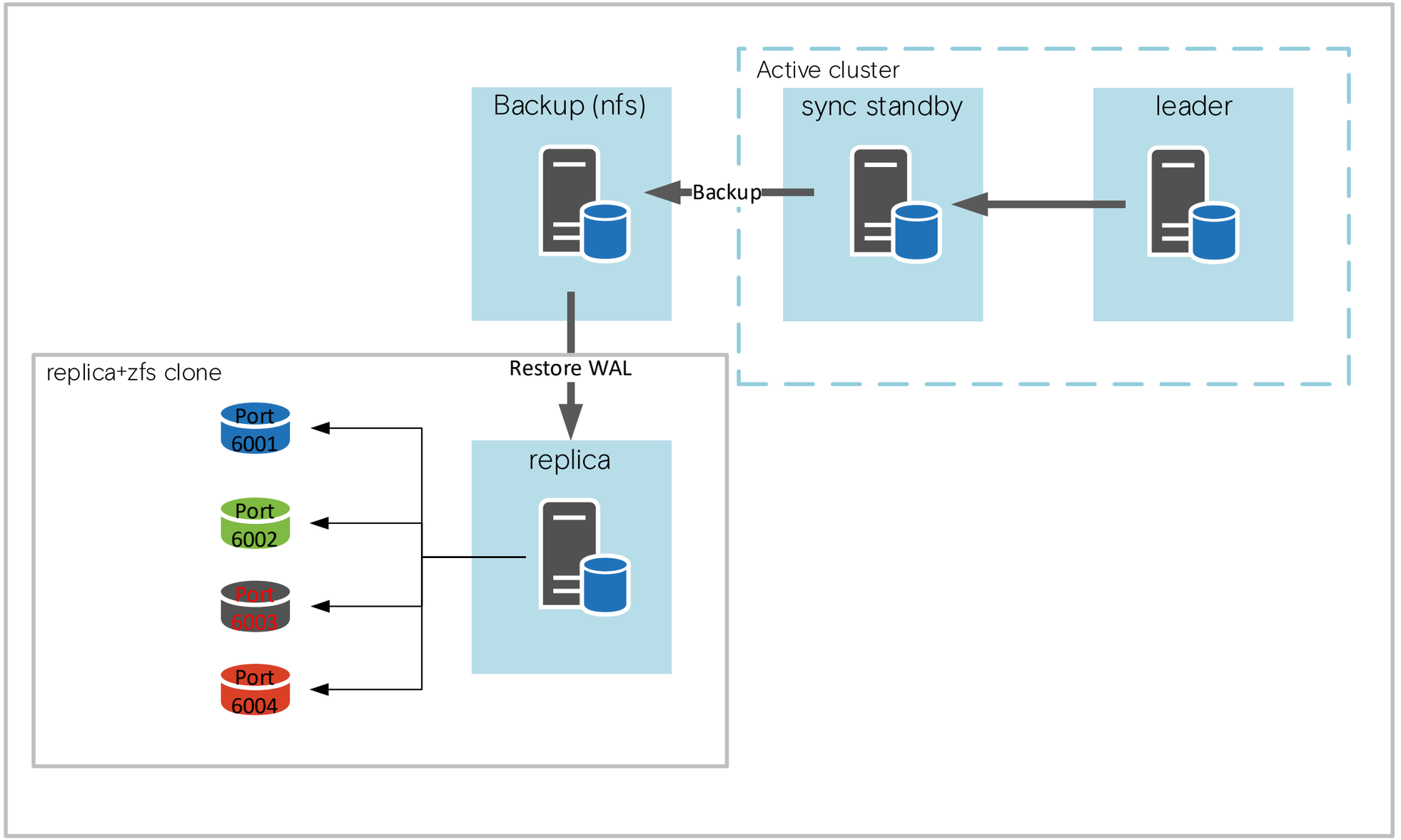
Диагностическая платформа



- отслеживание изменений текущих и ретроспективных показателей производительности;
- получение информации об исполняемых запросах SQL;
- отслеживание ресурсоёмких запросов, с целью их оптимизации;
- сбор показателей ОС и СУБД для проведения проактивного углубленного анализа производительности;

Ростелеком

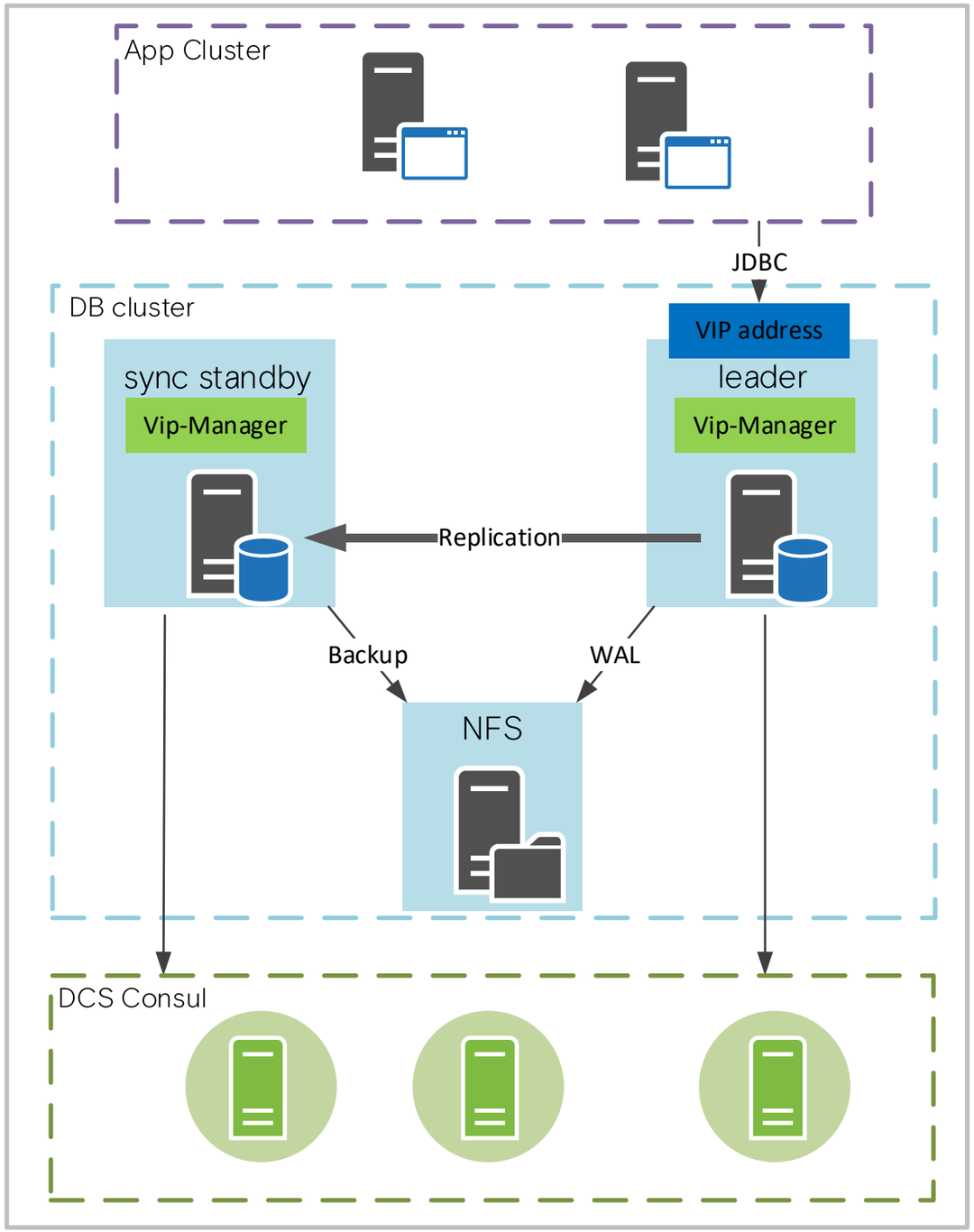
Техническое решение по быстрому созданию зон тестирования и разработки



Ростелеком

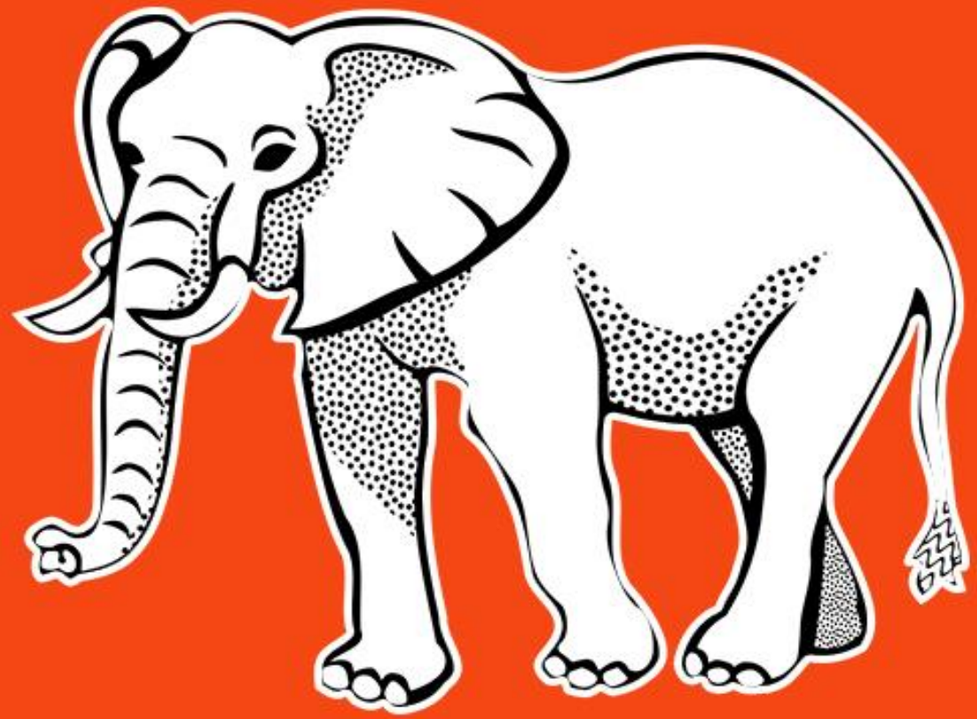
Проект СКМ Business Operational

- App cluster: JDK 8, Glassfish 5, Nginx, Spring boot 2.3, React
- DB Cluster: Postgres Pro Standard 13, Patroni, Vip-Manager, Consul agent, pg_probackup
- NFS Storage: NFS-server
- DCS Consul: Consul server
- ОС: РЕД ОС 7.2



Ростелеком

Хабр https://habr.com/ru/company/rostelecom/blog/566624/



Ростелеком

Команда линейного развития

- Существует сравнительно давно;
- Выработаны подходы, сформированы командные ценности;
- Основная задачи – обеспечить долгосрочное развитие проекта в соответствии с запросами бизнеса.

Команда миграции

- Сформирована на период проекта миграции;
- Основная задача:
  - Выполнить миграцию системы на новое окружение в заданный срок;
  - Обеспечить сходимость данных;
- Проектные ценности:
  - Обеспечить ***схожее*** функционирование системы после миграции;
  - «***Не мешать***» команде развития в процессе разработки миграции (по возможности)

Ростелеком
Технологии возможностей

База данных, интеграции

Объем миграции, объекты:
число схем = 4;
таблиц = 157;
представлений = 76;

Размер мигрируемой БД:
271 GB;

Отчеты: 18 отчетов, сложные скрипты формирования
(среднее число - 180 строк SQL кода на 1 отчет);

Число интеграций (Oracle, Greenplum): 9
Число таблиц, обновляемых по интеграционным связям = 44.

Технологические стеки

До миграции

- RHEL 7
- Oracle 19c EE;
- JDK 8;
- Glassfish 4;
- Spring Boot 1.5;
- React.

После миграции

- РЕД ОС 7.2
- PostgresPro Std 13;
- JDK 8;
- Glassfish 5;
- Spring Boot 2.3;
- React.

Миграция БД, данных

- Создание БД (ora2pg)
- Соответствие типов (было -> стало)
- Создание репликатора (на основе OracleFDW)
- Периодическая репликация (pgCron)
- Отслеживание изменений, вносимых командой линейного развития

Модуль репликации - требования

- Односторонняя репликация данных Oracle -> PG
- Лучше не вносить изменения в БД Oracle
- Репликация – временно, на период разработки миграции
- Онлайн – нет необходимости, лаг 1-2 дня - допустим
- Просто и быстро
- Частые доработки, переиспользование
- Выбор – самописный репликатор на основе Oracle FDW

Ростелеком
Технологии возможностей

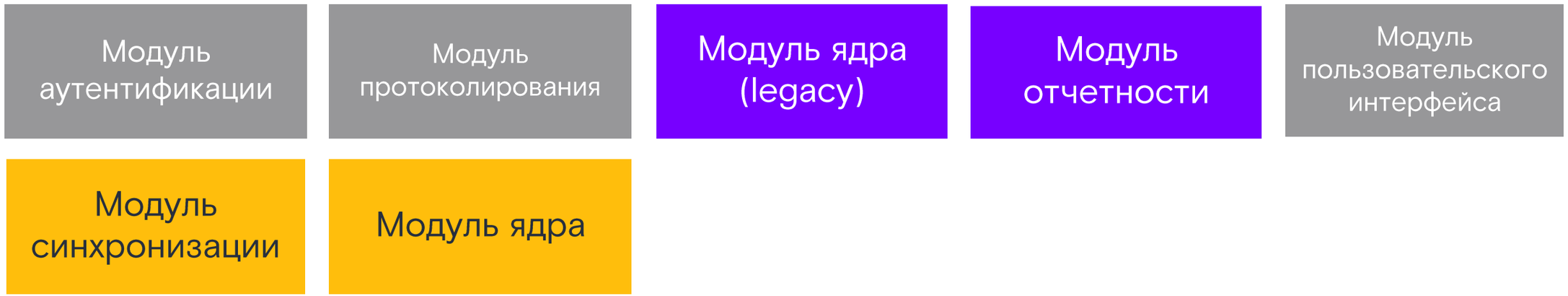
Модуль репликации - реализация

- Удаление внешних ключей
- Таблица «метаданных»
- Репликация интеграционного слоя
- Репликация основного слоя
- Восстановление внешних ключей, синхронизация sequence

entity	delete_stmt	insert_stmt	count_stmt
mcs.tarif_mounting	delete from {entity};	insert into mcs.tarif_mounting (branch_name, ka	select count(*) from {entity};
db2l.bis3k_client	delete from {entity} where snap:	insert into db2l.bis3k_client (master_id, snapshot	select count(*) from {entity} wl
db2l.itc_charges	delete from {entity} where snap:	insert into db2l.itc_charges (master_id, snapshot_	select count(*) from {entity} wl
db2l.cms_channel_elabor	delete from {entity} where snap:	insert into db2l.cms_channel_elaboration (maste	select count(*) from {entity} wl
db2l.r12_contragent	delete from {entity} where snap:	insert into db2l.r12_contragent (master_id, snaps	select count(*) from {entity} wl
db2l.future_point_rtk_herr	delete from {entity} where snap:	insert into db2l.future_point_rtk_hermes (master_	select count(*) from {entity} wl
db2l.mdm6_client	delete from {entity} where snap:	insert into db2l.mdm6_client (master_id, snapsh	select count(*) from {entity} wl
db2l.misr12_hfct_curr_rate	delete from {entity} where snap:	insert into db2l.misr12_hfct_curr_rate (master_id,	select count(*) from {entity} wl
mcs.address_promo	delete from {entity};	insert into mcs.address_promo (id, id_channel, a	select count(*) from {entity};
mcs.branch_ref	delete from {entity};	insert into mcs.branch_ref (id, parent_id, name, c	select count(*) from {entity};
mcs.capacity_ref	delete from {entity};	insert into mcs.capacity_ref (id, description, start	select count(*) from {entity};
mcs.cfo_ref	delete from {entity};	insert into mcs.cfo_ref (id, code, name, short_nai	select count(*) from {entity};
mcs.cms_branch	delete from {entity};	insert into mcs.cms_branch (id, name) select id,	select count(*) from {entity};
mcs.cms_branch_branch_	delete from {entity};	insert into mcs.cms_branch_branch_ref_mapping	select count(*) from {entity};

fk_create_text	fk_enabled
ALTER TABLE IF EXISTS mcs.channel_history_data_mart ADD CON	[ ]
ALTER TABLE IF EXISTS mcs.channel_history_data_mart ADD CON	[ ]
ALTER TABLE IF EXISTS mcs.channel_history_data_mart ADD CON	[ ]
ALTER TABLE IF EXISTS mcs.channel_history_data_mart ADD CON	[ ]
ALTER TABLE IF EXISTS mcs.hermes_results ADD CONSTRAINT sys	[v]
ALTER TABLE IF EXISTS mcs.channel ADD CONSTRAINT channel_e	[v]
ALTER TABLE IF EXISTS mcs.task ADD CONSTRAINT task_rev_man	[v]
ALTER TABLE IF EXISTS mcs.task ADD CONSTRAINT task_parent_t	[v]
ALTER TABLE IF EXISTS mcs.channel_aud ADD CONSTRAINT chan	[v]

Адаптация приложений



Проектное решение – доработать приложения для возможности запуска на обоих СУБД:

- Единая кодовая база
- Общие артефакты (jar, war-файлы)

Сложность – адаптация SQL запросов для работы в обоих СУБД одновременно

SQL-запросы в приложениях (1)

ORM (JPA, Hibernate)

```
@Entity
@Table(schema = "MCS", name = "ADDRESS")
public class Address extends AbstractAddress{

    @Id
    @SequenceGenerator(name = "ADDRESS_SEQ", sequenceName = "ADDRESS_SEQ", initialValue = 1, allocationSize = 1)
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "ADDRESS_SEQ")
    @NotNull
    @Access(AccessType.PROPERTY)
    @Column(name = "ID")
    private Long id;

    @Column(name = "FIAS_CODE_STREET")
    private String fiasCodeStreet;
```

Hibernate Query Language

```
@Query("select regionId from Address group by regionId order by regionId asc")
fun findAllRegionIds(): List<Long?>

@Query(
    "select distinct a from Channel ch join ch.addressBnew a where a.regionId is not null and ch.lastMileType=1 and " +
            "ch.closed=false and a.yLatitude is not null and a.yLongitude is not null and a.closePointRTKId is not null"
)
fun findAllForGetFuturePoint():List<Address>
```

SQL-запросы в приложениях (2)

QueryDSL

```
sqlString
    .leftJoin( join: "MCS.ADDRESS address ON address.ID = channel.ADDRESS_B_ID")
    .leftJoin( join: "MCS.REGION_REF region ON region.ID = address.REGION")
    .where(
        expr: "(" + fillInClause( colName: "region.ID", rfList as List<Any>)
                + orValue
                + "("
                + fillInClause(
            colName: "contractProvider.BRANCH_ID",
            mrfList as List<Any>?
        )
                + orValue
                + fillInClause(
            colName: "channel.BRANCH_REF_ID",
            mrfList as List<Any>?
        )
                + ")" + ")"
    )
```

Запросы «в чистом виде»

```
case
  when cp.elaboration_tech_sol_lifetime is not null then
  case
    when fprtk.construction_year is not null
    or fcprtk.ground_cost is not null
    or fprtk.fabric_length is not null then
    case
      when
      -- 1 условие
      (add_months(trunc(to_date(cast(fprtk.construction_year as varchar(50)), 'YYYY'), 'YYY
      12) - interval '1' second <= add_months(cast(cp.elaboration_tech_sol_lifetime as date
      12)
      and fcprtk.ground_cost < (cp.ELABORATION_INSTALL_FEE + cp.ELABORATION_MONTHLY_FEE + c
      and ((months_between(cast(add_months( trunc(to_date(cast(fprtk.construction_year as v
      cast(cp.elaboration_tech_sol_lifetime as date)) * s_pa_c_ca_exp.COST) + fcprtk.ground
      or
```

Проблемы SQL в JDK приложениях

- Общие проблемы

  Типы полей (date, number, boolean), регистрозависимость, квотированные идентификаторы, null в string), строгая типизация (fk и т.д.)

- ORM (JPA, Hibernate)

  SpringBoot 1.5 (NamingStrategy), SpringBoot 2.3 + Glassfish 5

- Hibernate Query Language, hibernate envers

- QueryDSL

  Boolean (case “expression” when “constant” then …)

- Запросы «в чистом виде»

  Адаптация для работы в обоих СУБД, сложные функции, Orafce.

«Общие» проблемы

Типы полей, нюансы

- String + null
- Date – содержит время в Oracle, не содержит в PG

  Number vs Numeric:

  > Без указания точности и масштаба создаёт столбец, в котором можно сохранять числовые значения любой точности и масштаба в пределах, поддерживаемых системой. В столбце этого типа входные значения не будут приводиться к какому-либо масштабу, тогда как в столбцах numeric с явно заданным масштабом значения подгоняются под этот масштаб. (Стандарт SQL утверждает, что по умолчанию должен устанавливаться масштаб 0, т. е. значения должны приводиться к целым числам. Однако мы считаем это не очень полезным. Если для вас важна переносимость, всегда указывайте точность и масштаб явно)

- Boolean

  select … from some_table where boolean_field

  select … from some_table where boolean_field = '1'

  ~~select … from some_table where boolean_field = 1~~

Адаптация для работы в обоих СУБД (1)

- Greatest, least

Заметьте, что функции GREATEST и LEAST не описаны в стандарте SQL, но часто реализуются в СУБД как расширения. В некоторых других СУБД они могут возвращать NULL, когда не все, а любой из аргументов равен NULL.

- months_between, add_months, to_date, TO_CHAR(..., 'DDD')

MONTHS_BETWEEN returns number of months between dates date1 and date2. If date1 is later than date2, then the result is positive. If date1 is earlier than date2, then the result is negative. If date1 and date2 are either the same days of the month or both last days of months, then the result is always an integer. Otherwise Oracle Database calculates the fractional portion of the result based on a 31-day month and considers the difference in time components date1 and date2.

Orafce – Oracle’s compatibility functions and packages

Адаптация для работы в обоих СУБД (2)

- LISTAGG (DISTINCT)
- Древовидные запросы, CTE
- SEQUENCE

Ora: select sequence.nextval from dual

Pg: select nextval(‘sequence’)

- LIMIT vs ROWNUM = FETCH NEXT ROWS ONLY
- JOB -> pgCron, @Sheduled
- in (...many values...) -> in (select ...)

Проблемы SQL в JDK приложениях

- Общие проблемы

  Типы полей (date, number, boolean), регистрозависимость, квотированные идентификаторы, null в string), строгая типизация (fk и т.д.)

- ORM (JPA, Hibernate)

  SpringBoot 1.5 (NamingStrategy), SpringBoot 2.3 + Glassfish 5

- Hibernate Query Language, hibernate envers

- QueryDSL

  Boolean (case “expression” when “constant” then …)

- Запросы «в чистом виде»

  Адаптация для работы в обоих СУБД, сложные функции, Orafce.

Интересные запросы (1)

В какой базе мы находимся?

```
select coalesce(cast(null||'ora' as varchar(100)),'pg') db_type from dual
```

Двойной TO_DATE

```
with dates as (
  select to_date('19.02.2020', 'DD.MM.YYYY') dt from dual
  union all
  select to_date('21.02.2020', 'DD.MM.YYYY') dt from dual
)
select TO_DATE(dt, 'YYYY-MM-DD HH24:MI:SS') from dates
```

Ограничение по периоду

```
… trunc(bdf2.PERIOD, 'MM') = (SELECT ADD_MONTHS(trunc(SYSDATE, 'MM'), -1) FROM dual)
```

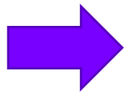
Интересные запросы (2)

Несколько ROWNUM

```
select * from
  ( select ... and rownum = 1
    union all
   select ... and rownum = 1 )
and rownum = 1
```

Двойной агрегат

```
select COUNT(max(some_field))
from
... group by ....
```

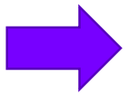


```
select count(*) from
(
  select max(some_field) from ...
  group by ....
) q
```

Интересные запросы (3)

Запрос с максимумом

```
SELECT HR.* FROM SOME_TABLE HR
    WHERE HR.ADDRESS_ID =
  (SELECT AD.ID FROM SOME_TABLE2 AD
        WHERE AD.ID =
    (SELECT CH.ADDRESS_B_ID
       FROM SOME_TABLE_3 CH
            WHERE CH.ID = 517425
     )
)
    AND HR.PROCESS_DATE =
(SELECT MAX(HR.PROCESS_DATE) FROM
SOME_TABLE)
    AND ….
```



```
SELECT * FROM
(
  SELECT
    DENSE_RANK() over
      (partition by HR.ADDRESS_ID
       order by HR.PROCESS_DATE desc) DENSE_R,
       HR.*
  FROM SOME_TABLE HR
…
) Q where DENSE_R = 1
```

Ростелеком

Спасибо за внимание!

Всегда на связи

Технологии возможностей