

Как устроена поставка данных в Яндексе и почему мы больше не боимся перекладывать JSON'ы

Andrei Tserakhau @yandex

Agenda

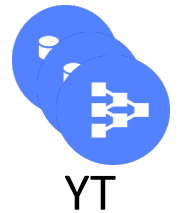
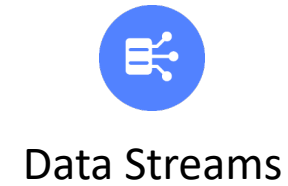
- Что мы такое
- Зачем мы такое
- Как мы такое

Зачем

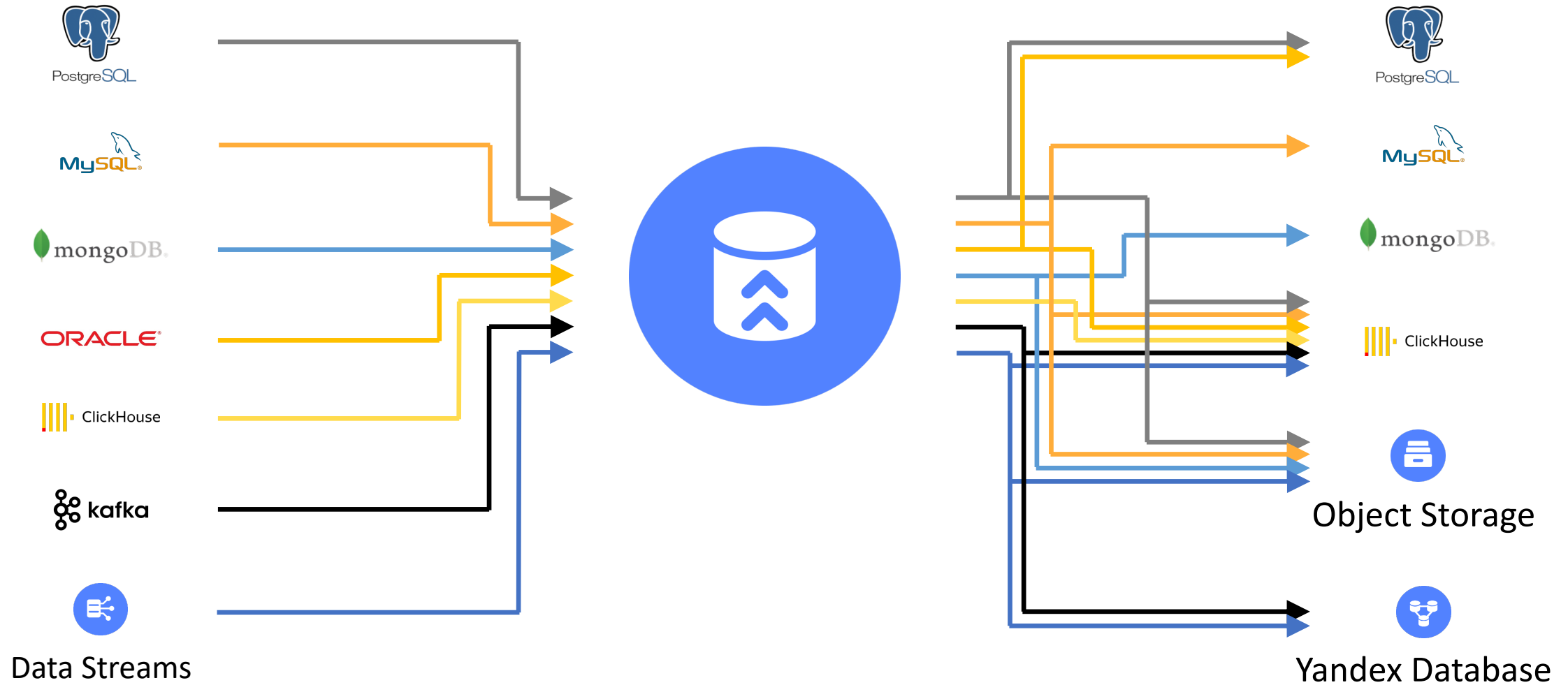
- Ландшафт технологий яндекса:



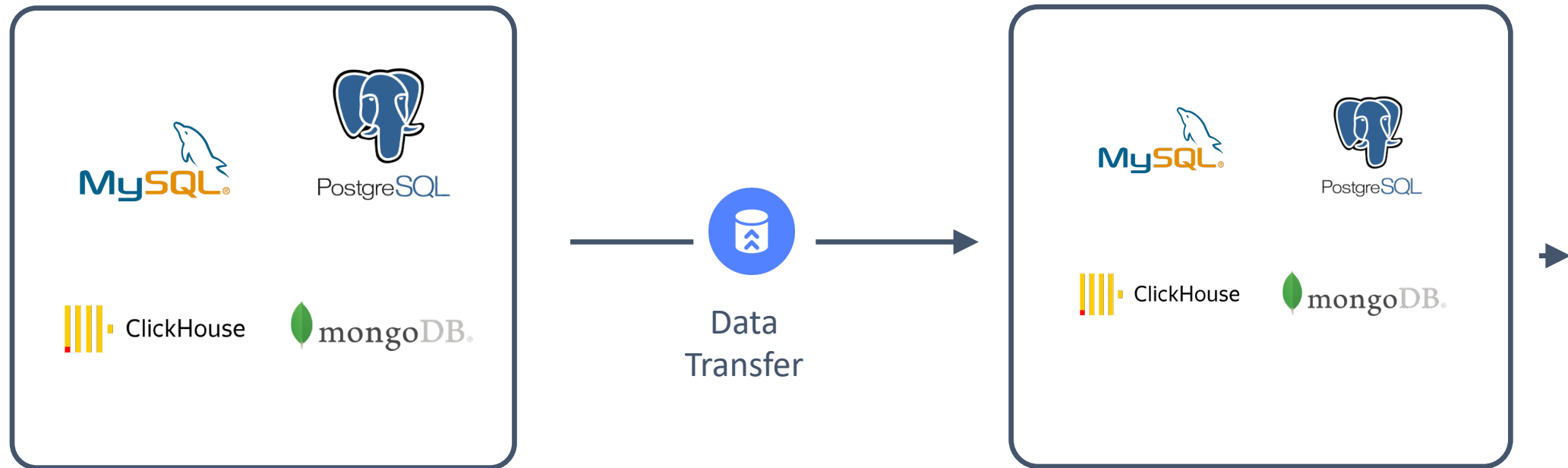
ORACLE®



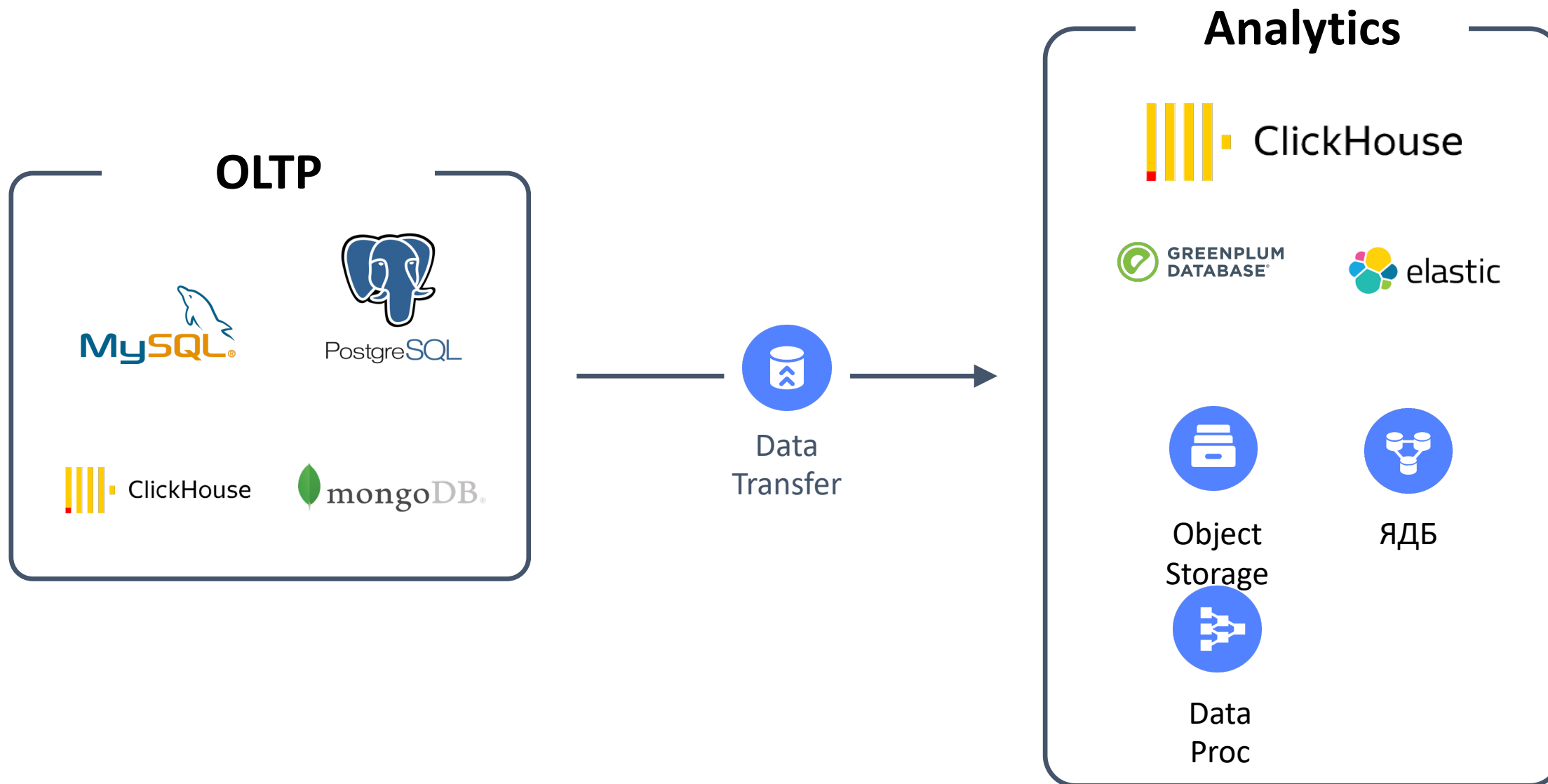
Одна боль – одно решение



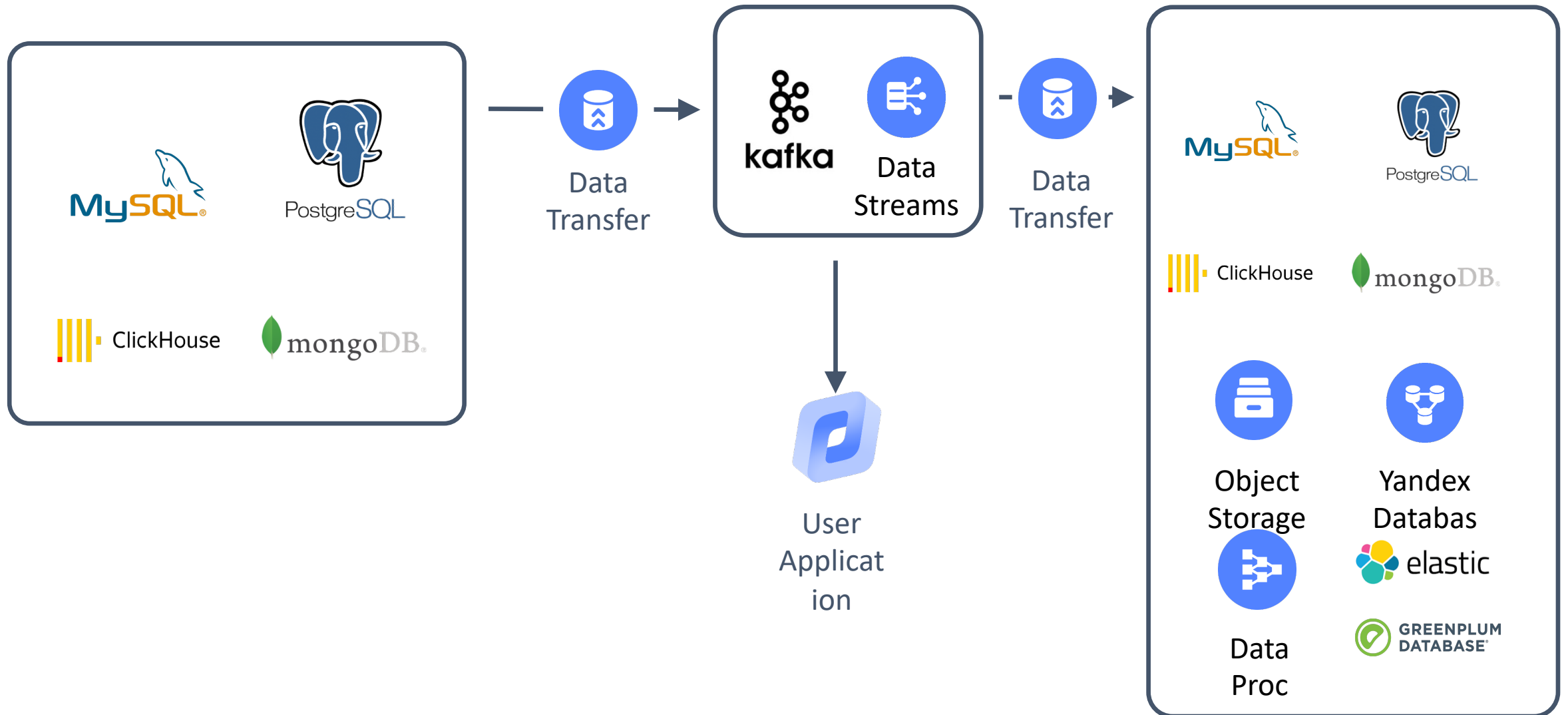
Миграция баз данных



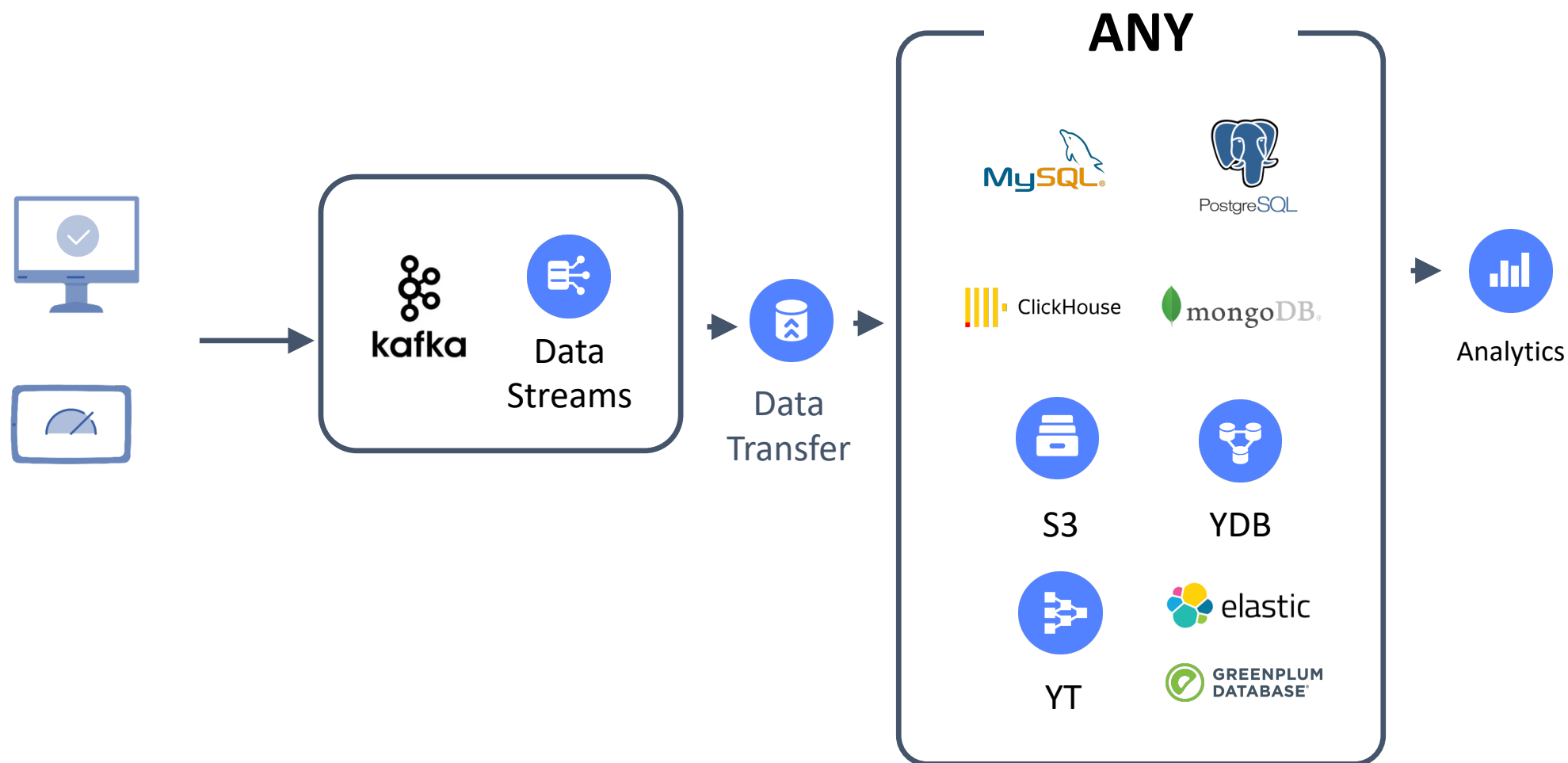
Аналитика



Change Data Capture



Поставка сырых данных



Задачи

- Миграция данных гомогенных хранилищ
- Миграция данных гетерогенных хранилищ
- Распределение читающей нагрузки на разные хранилища
- Консолидация данных
- Фрагментация данных
- Поставка сырых данных
- Процессинг данных

Решение

- Сервис логического переноса данных между различными системами обработки и хранения данных

Готовые решения



Oracle GoldenGate



Amazon Kinesis
Data Firehose

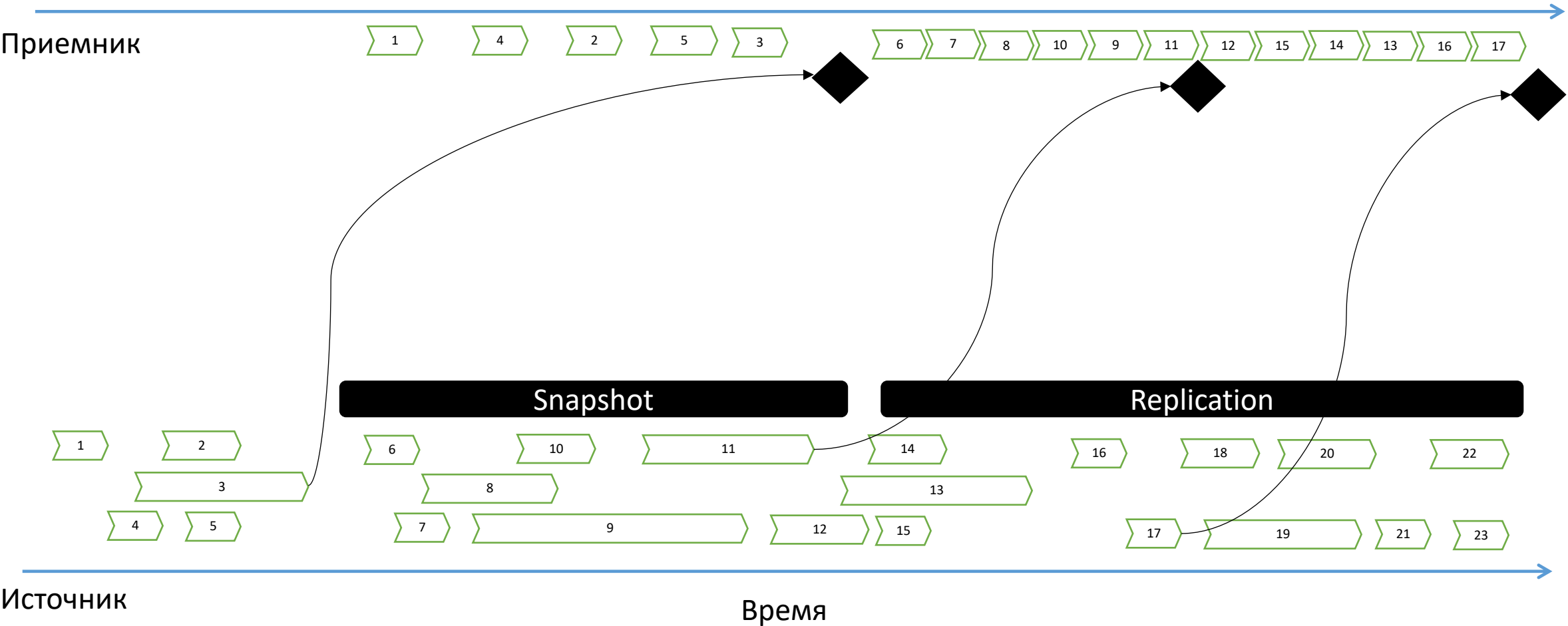


AWS Database
Migration Service



- Мы не ставим дополнительных агентов на хосты
- Не зависим от других технологий
- Все коннекторы в одном сервисе
- Повышенные гарантии для гомогенных трансферов
- Транзакционность *
- Управляемый сервис
- Работаем в режиме serverless

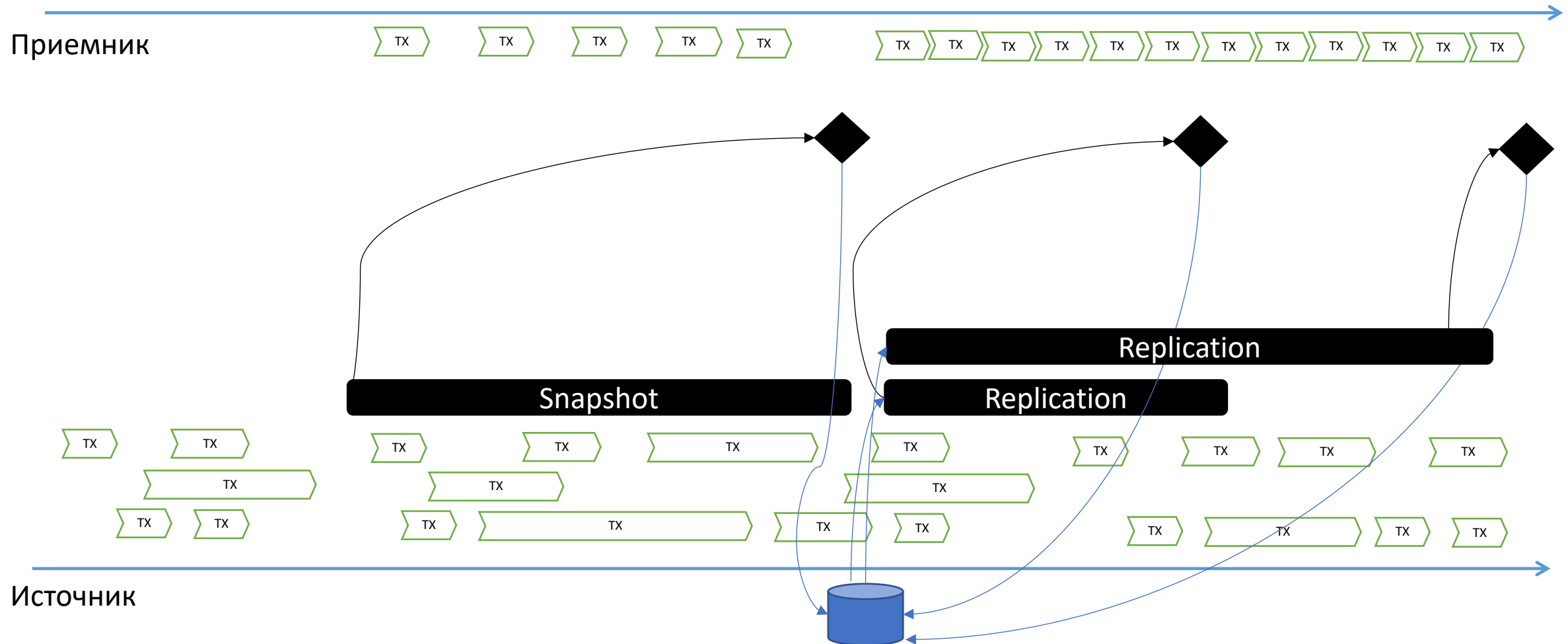
Логический перенос данных



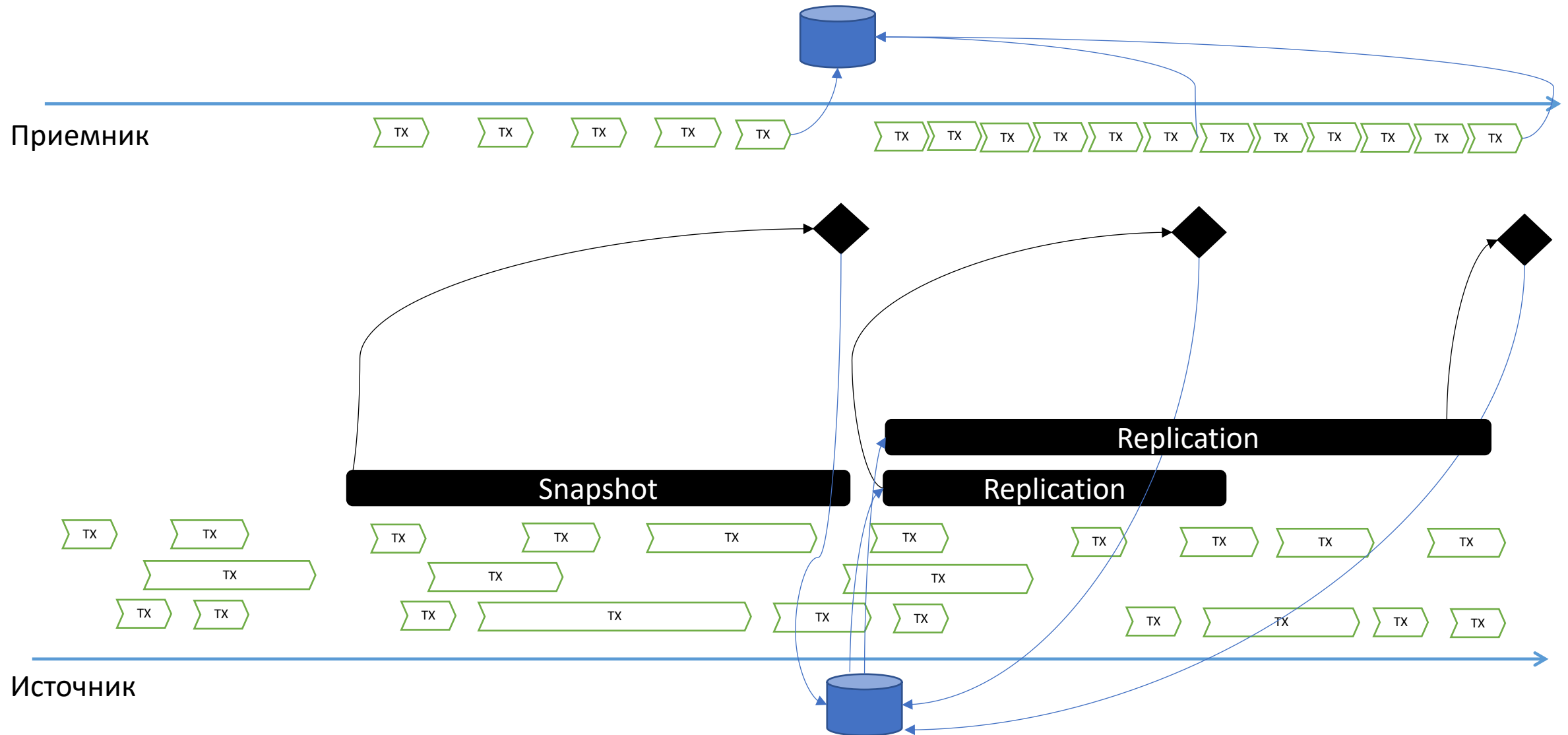
Целостность данных

- Семантика поставки и транзакционность
- Реализация At-Least-Once
- Реализация At-Most-Once
- Реализация Exactly-Once
- Как этим управлять и зачем это может быть нужно

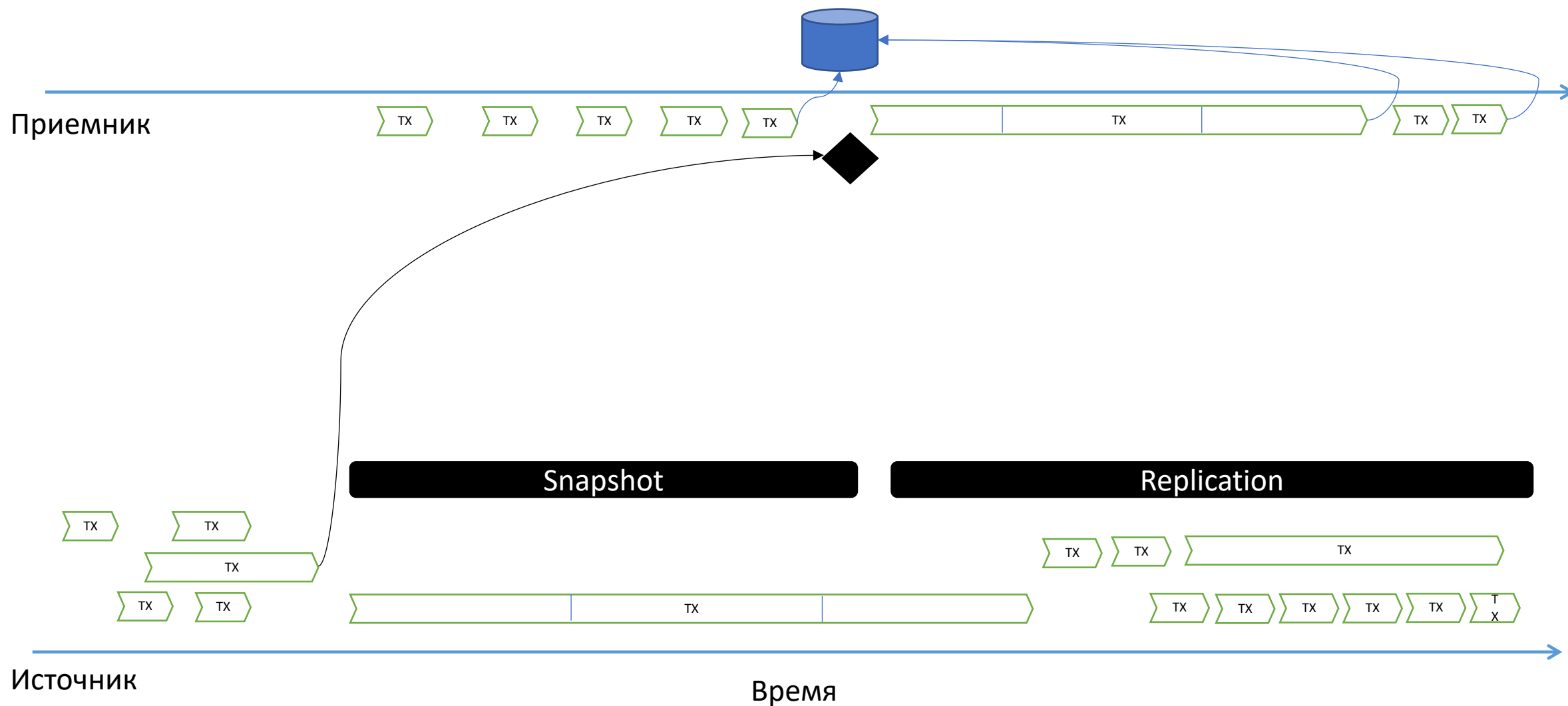
Семантика поставки (at least once)



Семантика поставки (exactly once)



Транзакционная целостность



Разные базы

OLTP

- Postgres
- Mysql
- Oracle
- MS SQL

NoSQL

- MongoDB

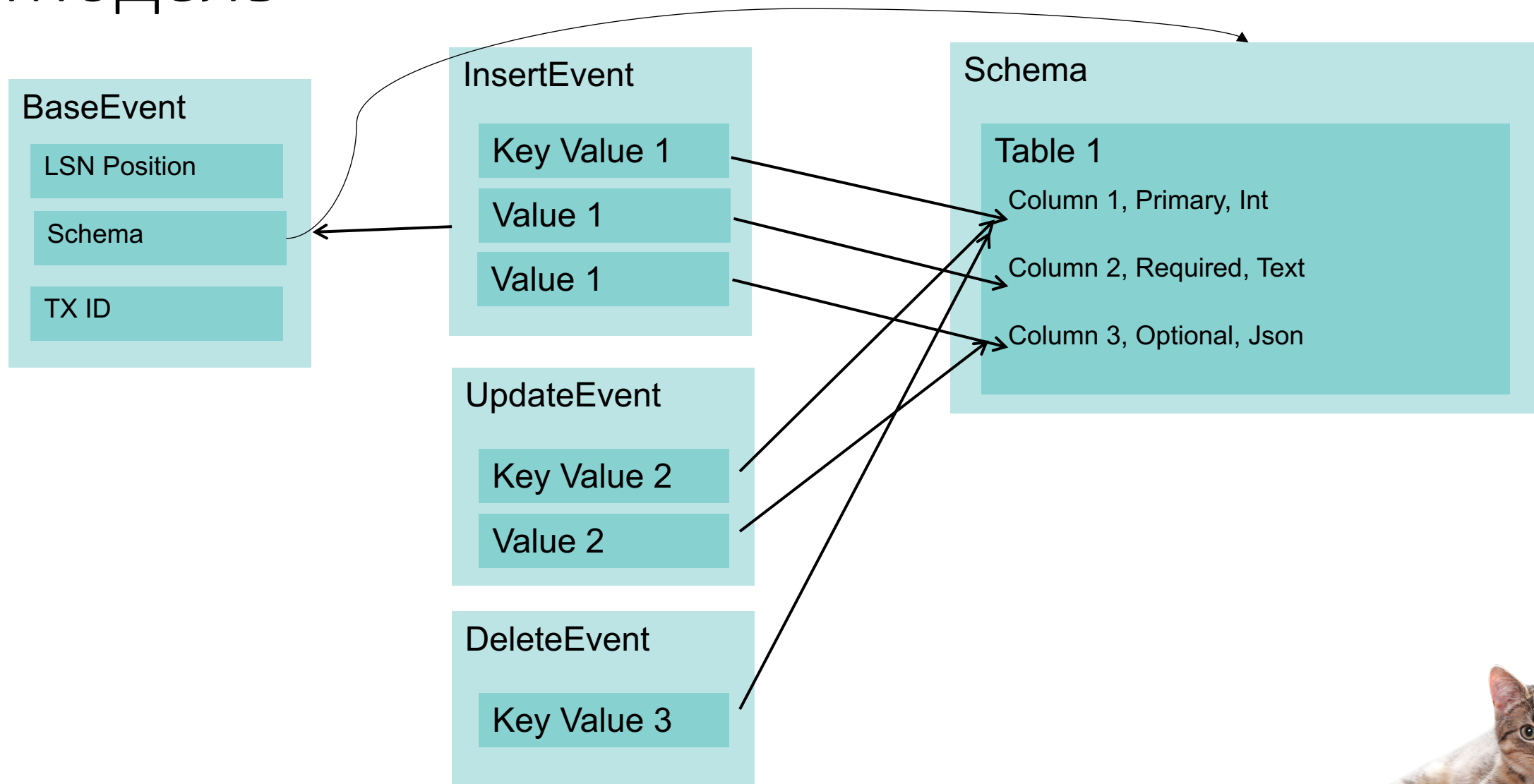
Message Bus

- Kafka
- YDS
- EventHub
- Kinesis

Analytical

- Clickhouse
- S3
- Hadoop
- YT

Модель



Форматы данных и система типов

[illegible]

Полиморфные типы

```
type ColumnType interface {  
    SRID() int  
    Valiadte() error  
    Cast(value Value) (Value, error)  
}
```

```
type GeometryValue interface {  
    SRID() int  
    Type() ColumnType  
    GeoJson() string  
    EWKB() []byte  
    EWKT() string  
}
```

```
type MySQLValue interface {  
    MySQLValue() []byte  
}
```

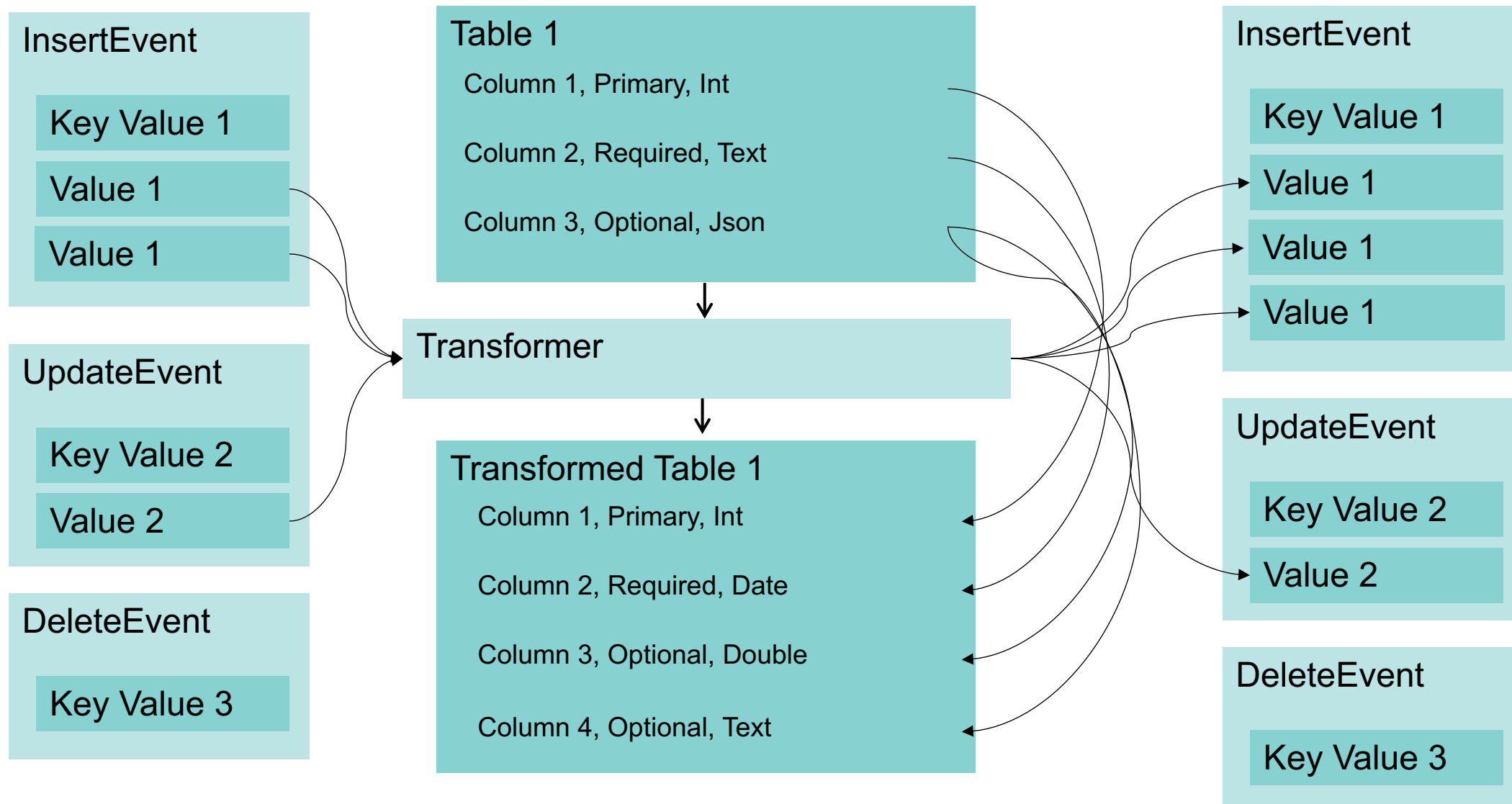
```
type MySQLGeometryValue struct {  
    SRID() int  
    GeoJson() string  
    EWKB() []byte  
    EWKT() string  
    MySQLValue() []byte  
}
```

```
type PGGeometryValue struct {  
    SRID() int  
    GeoJson() string  
    EWKB() []byte  
    EWKT() string  
    PGValue() []byte  
}
```

```
type PGValue interface {  
    PGValue() []byte  
}
```



Трансформации



Спасибо!

Вопросики пожалуйста, очень хочу ответить на вопросики

