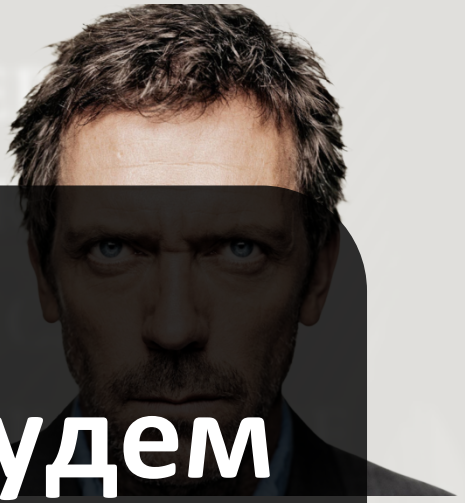
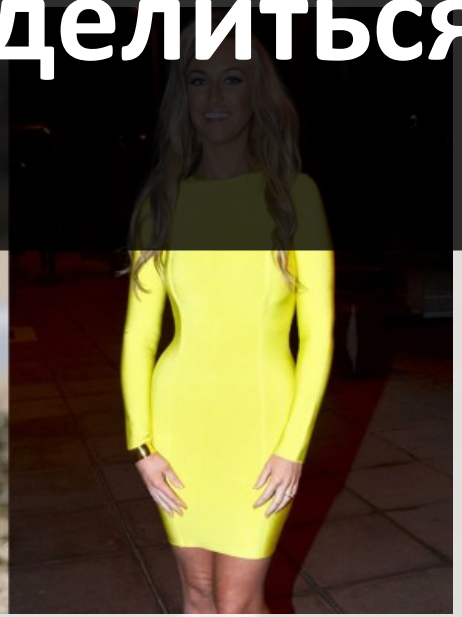
Правдивая история об использовании SQL Server Change Data Capture

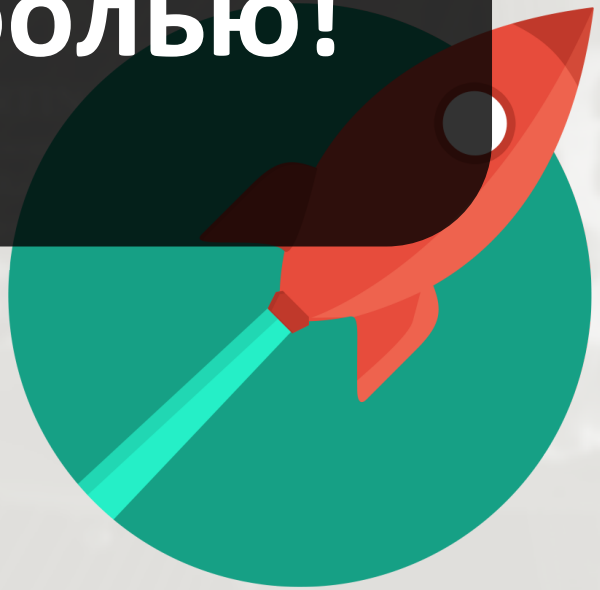
Автор:
Михалев Сергей

Сегодня не будет!



Сегодня мы будем делиться болью!





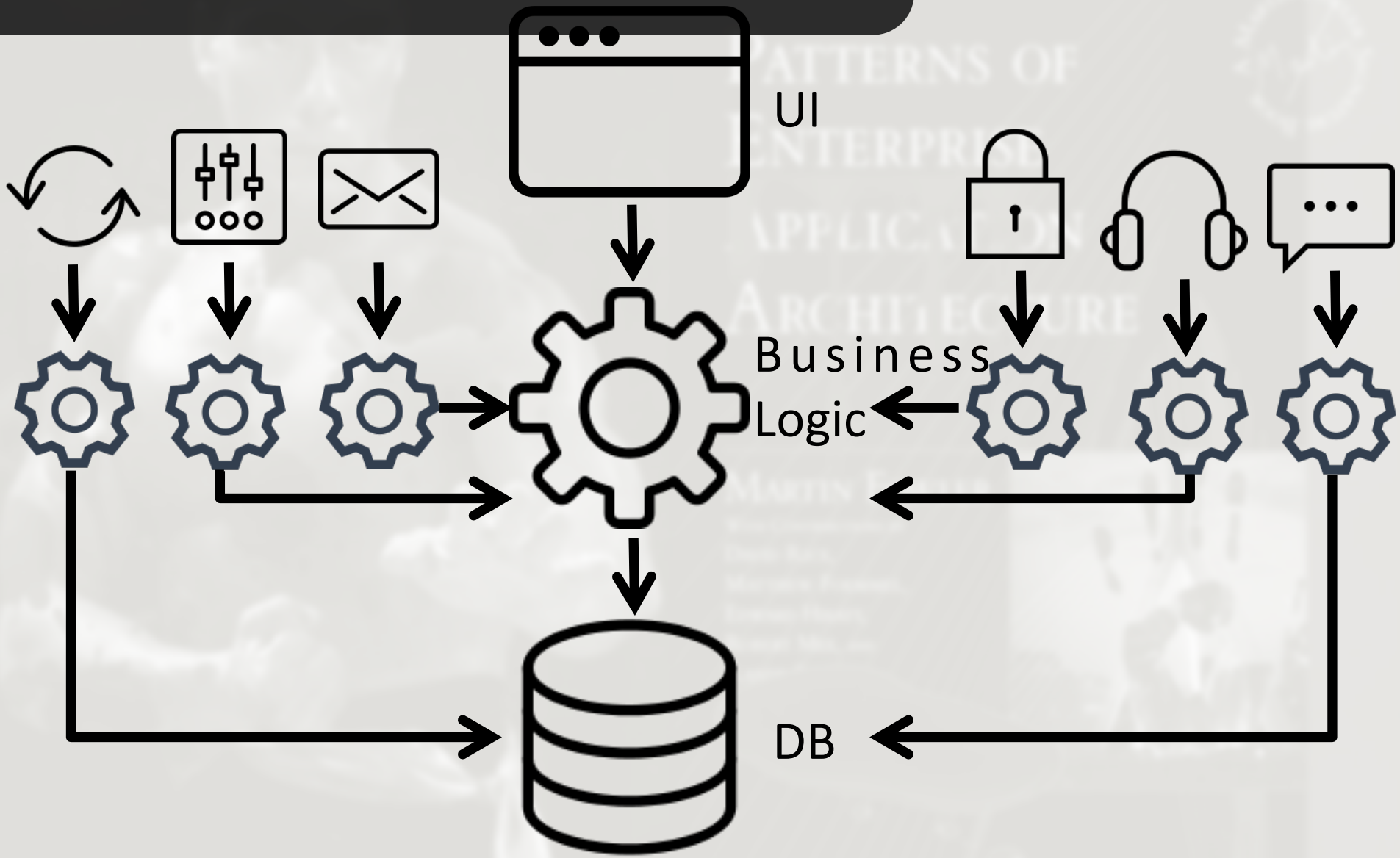
Где же больше крови?!

Разминка

Нам нужно научится делиться болью! ☺

Больно?

База – единственная шина данных



Больше информации

1) Это аутсорс
2) Разрабатывается со времен .Net 1.0, Asp .Net 1.0 и SQL Server 2000
3) Команда разработчиков порядка 20 человек
4) Уже давно никто не знает всё приложение целиком
5) Нет доменной модели
6) Тяжело написать юнит тест без прямого взаимодействия с базой данных
7) Большая часть бизнес логики в одной сборке
8) Очень мало автоматических тестов

А теперь больно?

Задача

Традиционные способы

Polling (опрос)

Triggers

- DML triggers;
- DML triggers + Service Broker;

Недостатки:

- Могут слишком сильно нагружать базу;
- Работают синхронно;
- Замедляют основную транзакцию;

И закипела работа

А через какое-то время ...

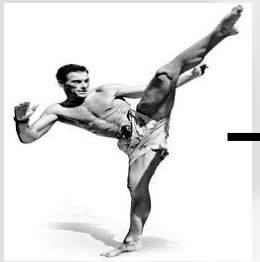
Команда показывает, что они сделали и обсуждают дизайн с заказчиком.


Заказчик
Триггера – это зло.
А можно тоже самое но без них?!

Команда отвечает


Заказчик
Ну тогда нужно переделать.
Больновато, да?

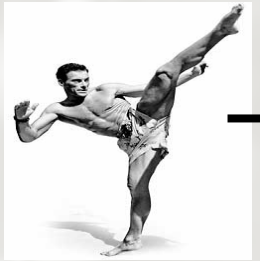
Transaction Log



Vandam1

1 INSERT

7 COMMIT



Vandam2

3 UPDATE

Log Buffer

Финиш транзакции 9

Старт транзакции 2 4

Все изменения в лог 8

Transaction Log

6 Информация об изменении в каждой строке записывается в кеш лога

Все измененные (dirty) страницы в data файл(ы)

p1 p2 p3 p4 p5 p6 p9

Buffer Cache

10

5

Data File(s)

Измененные страницы помечаются как dirty

В кеш с диска считываются страницы

Database Recovery Model

FULL | **BULK** | **SIMPLE**

1) Если изменения о всех строках попадают в Transaction Log, то когда они удаляются?

Только при log backup

При log backup, для bulk операций после checkpoint-a.

*Сразу после checkpoint

2) Зачем хранить изменения в transaction log, что это дает?

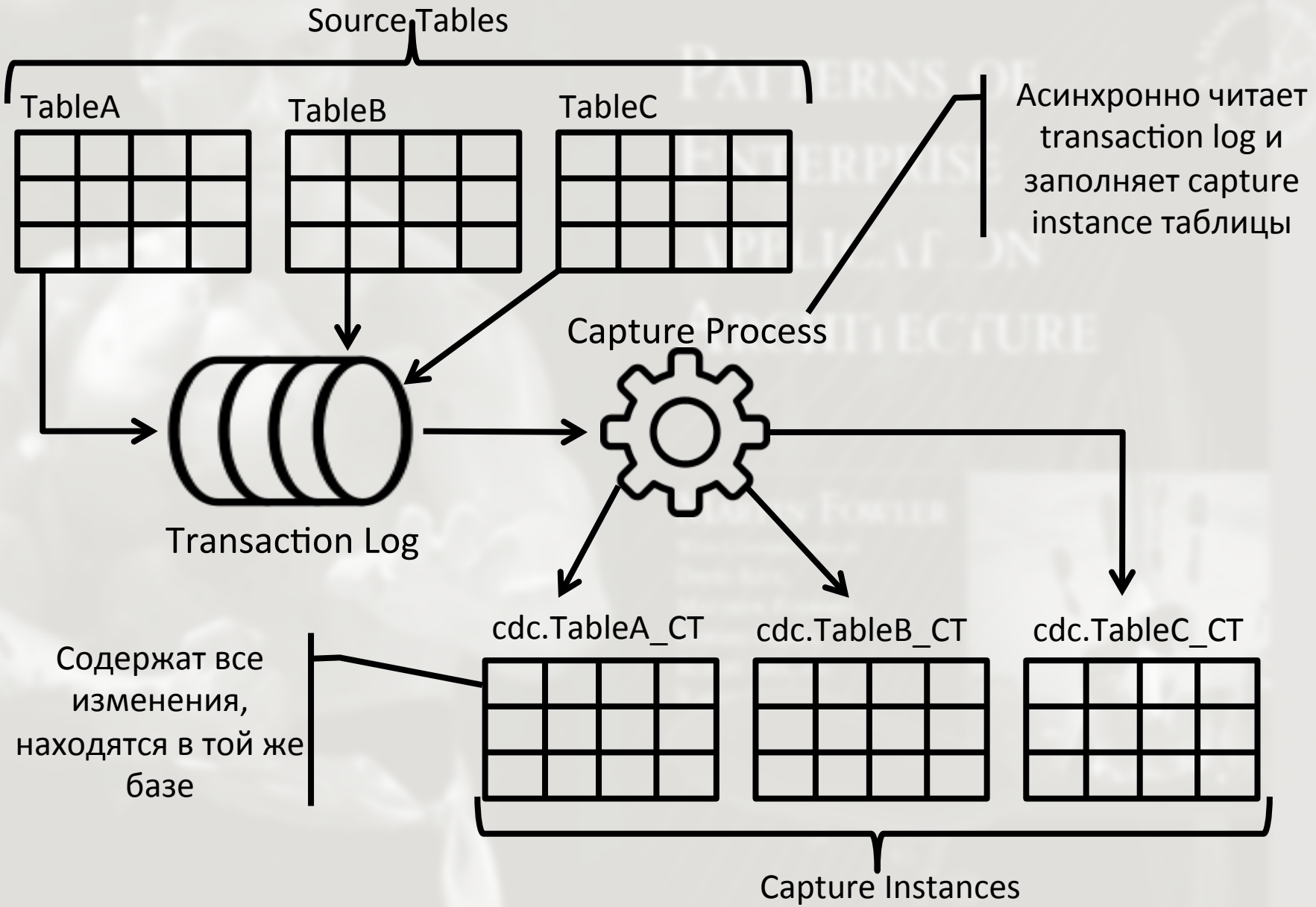
Point-in-time восстановление

Point-in-time восстановление, если в промежутке нет bulk операций

Нет возможности делать backup/restore у transaction log.

Database Mirroring
Log Shipping
Always On

Change Data Capture



Capture Instances

```
insert into users
(first_name,last_name,create_time)
values('Van', 'Dam', getdate())
```

```
select *
from cdc.DBO_USERS_CT
```

__$start_lsn	__$end_lsn	__$seqval	__$operation	__$update_mask	id	first_name	last_name	create_time
0x0000004300000067001A	NULL	0x0000043000000670018	2	0x0F	1	Van	Dam	2014-12-07 01:08:23.557

```
update users
set first_name = 'Sergey',
    last_name = 'Mikhalev'
where id = 1
```

```
select *
from cdc.DBO_USERS_CT
```

__$start_lsn	__$end_lsn	__$seqval	__$operation	__$update_mask	id	first_name	last_name	create_time
0x0000004300000067001A	NULL	0x0000043000000670018	2	0x0F	1	Van	Dam	2014-12-07 01:08:23.557
0x0000043000000760003	NULL	0x0000043000000760002	3	0x06	1	Van	Dam	2014-12-07 01:08:23.557
0x0000043000000760003	NULL	0x0000043000000760002	4	0x06	1	Sergey	Mikhalev	2014-12-07 01:08:23.557

```
delete users
where id = 1
```

```
select *
from cdc.DBO_USERS_CT
```

__$start_lsn	__$end_lsn	__$seqval	__$operation	__$update_mask	id	first_name	last_name	create_time
0x0000004300000067001A	NULL	0x0000043000000670018	2	0x0F	1	Van	Dam	2014-12-07 01:08:23.557
0x0000043000000760003	NULL	0x0000043000000760002	3	0x06	1	Van	Dam	2014-12-07 01:08:23.557
0x0000043000000760003	NULL	0x0000043000000760002	4	0x06	1	Sergey	Mikhalev	2014-12-07 01:08:23.557
0x0000004300000007C0004	NULL	0x0000004300000007C0002	1	0x0F	1	Sergey	Mikhalev	2014-12-07 01:08:23.557

Log sequence number (LSN)

Номер внутри транз.

Операция

Маска изменившихся полей

Capture Process

Capture Job

1) Используя sp_replcmds, читает transaction logs и записывает изменения в capture instance.

2) Может работать при любом Recovery Mode, но если simple log truncation произойдет только после прочитывания, а не после checkpoint.

3) Обрабатывает 1000 транзакций за один цикл с 5 секундным таймаутом между циклами.

Cleanup Job

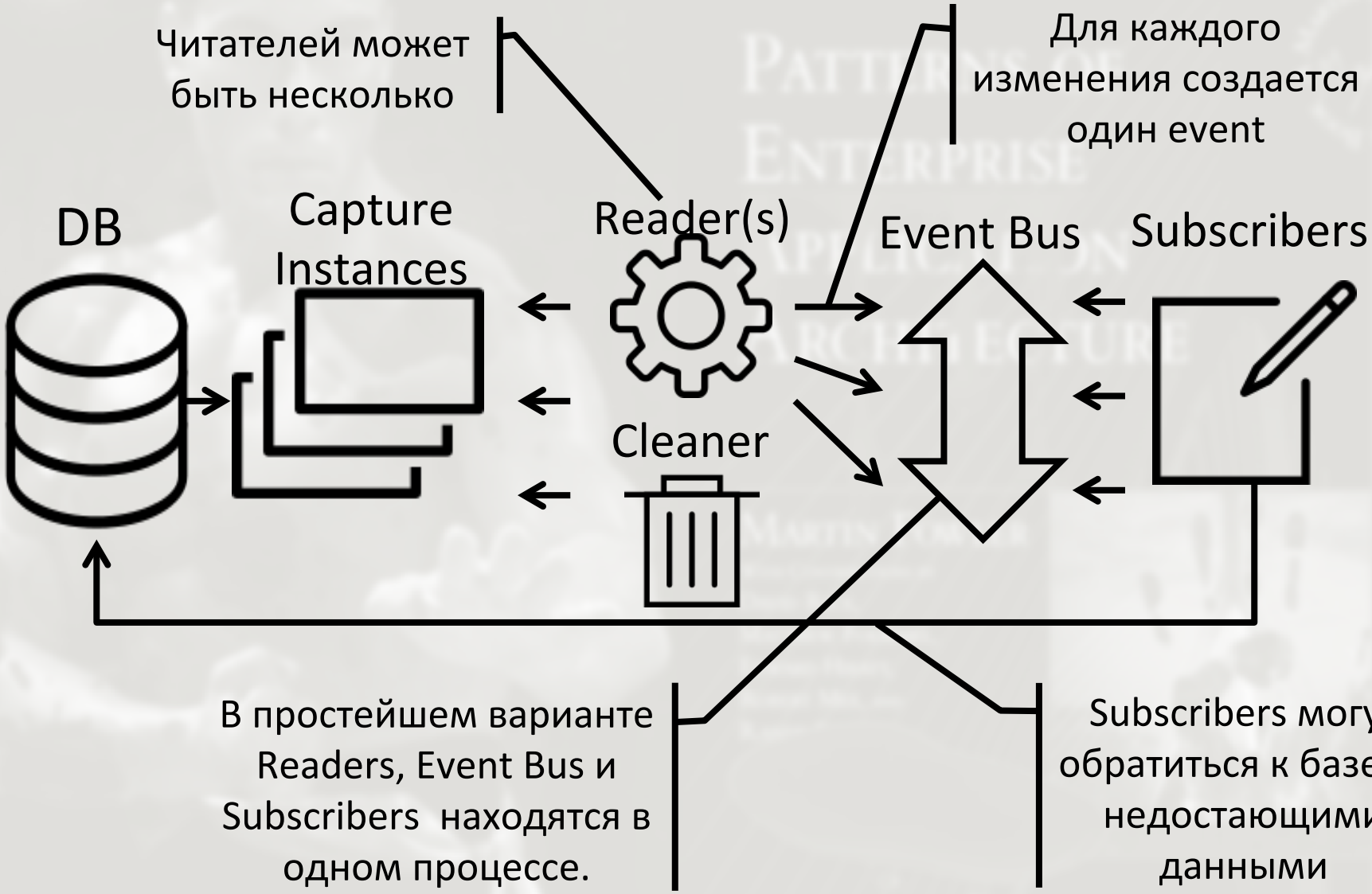
1) Очищает capture instance таблицы от старых записей.

2) По умолчанию, стартует в 2 часа ночи и удаляет все записи старше 3 дней.

Готовность к работе

В целом все просто!

Общая архитектура



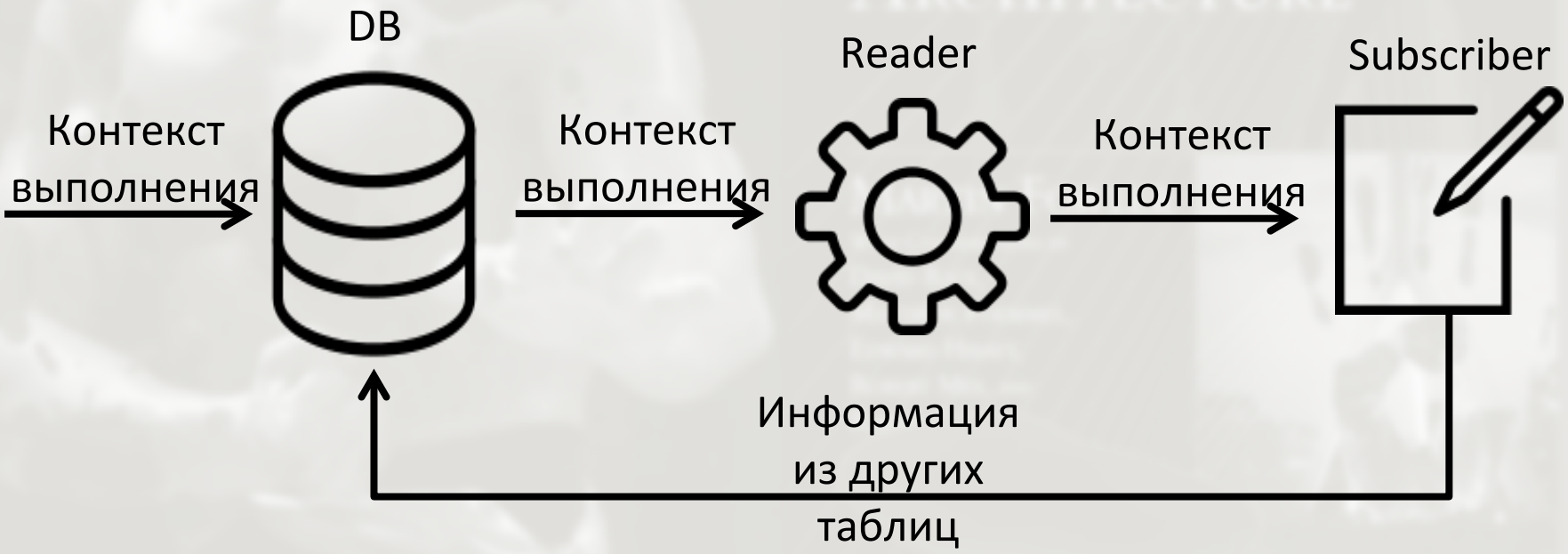
Первый соперник

Необходимо иметь возможность передавать контекст изменений.

Контекст изменений

Дополнительная информация необходимая для обработки subscriber-ом события:

1) Из других таблиц.
2) Контекста выполнения, например, пользователь, запустивший операцию изменения.



Контекст изменений: подход в лоб

Добавить в каждую таблицу контекст выполнения. Т.е. например, добавить колонку user_id.

Недостатки:

1) Необходимо менять схему ВСЕХ таблиц.
2) Для bulk операций придется дублировать море информации.

Понятно, что это очень плохо и мы так делать не будем!

Связывание через transactionId

1) Добавим таблицу TRANSACTION_CONTEXT.

```
CREATE TABLE TRANSACTION_CONTEXT
(
    user_id int NOT NULL
)
```

2) Начнем следить за изменениями в этой таблице.
3) Вместе с основными изменениями в той же транзакции будем осуществлять вставку в TRANSACTION_CONTEXT

```
select tc.user_id, a.*,
from cdc.DBO_TABLE_CT a
inner join cdc.LSN_TIME_MAPPING altm
    on i.__$start_lsn = iltm.start_lsn
outer apply
(
    select top 1 user_id
    from cdc.DBO_TRANSACTION_CONTEXT_CT tc
    inner join cdc.LSN_TIME_MAPPING tcltm
        on tc.__$start_lsn = tcltm.start_lsn
        and tcltm.tran_id = altm.tran_id
) tc
```

Второй круг

Обработка ошибок.

Обработка ошибок

Необходимо обеспечить возможность

1) Повторного запуска обработки упавшего изменения.
2) Сохранения изменения при его падении.

А можно ли продолжать обработку других изменений при падении отдельного изменения?

Failed сервис

1) Создадим отдельную таблицу для учета повторных попыток.

```
CREATE TABLE CDC_FAILED_CHANGES
(
    capture_instance sysname NOT NULL,
    failed_lsn binary (10) NOT NULL,
    retry_count int NOT NULL,
    is_processed bit NOT NULL
)
```

2) Теперь Clean Up сервис не удаляет упавшие изменения.

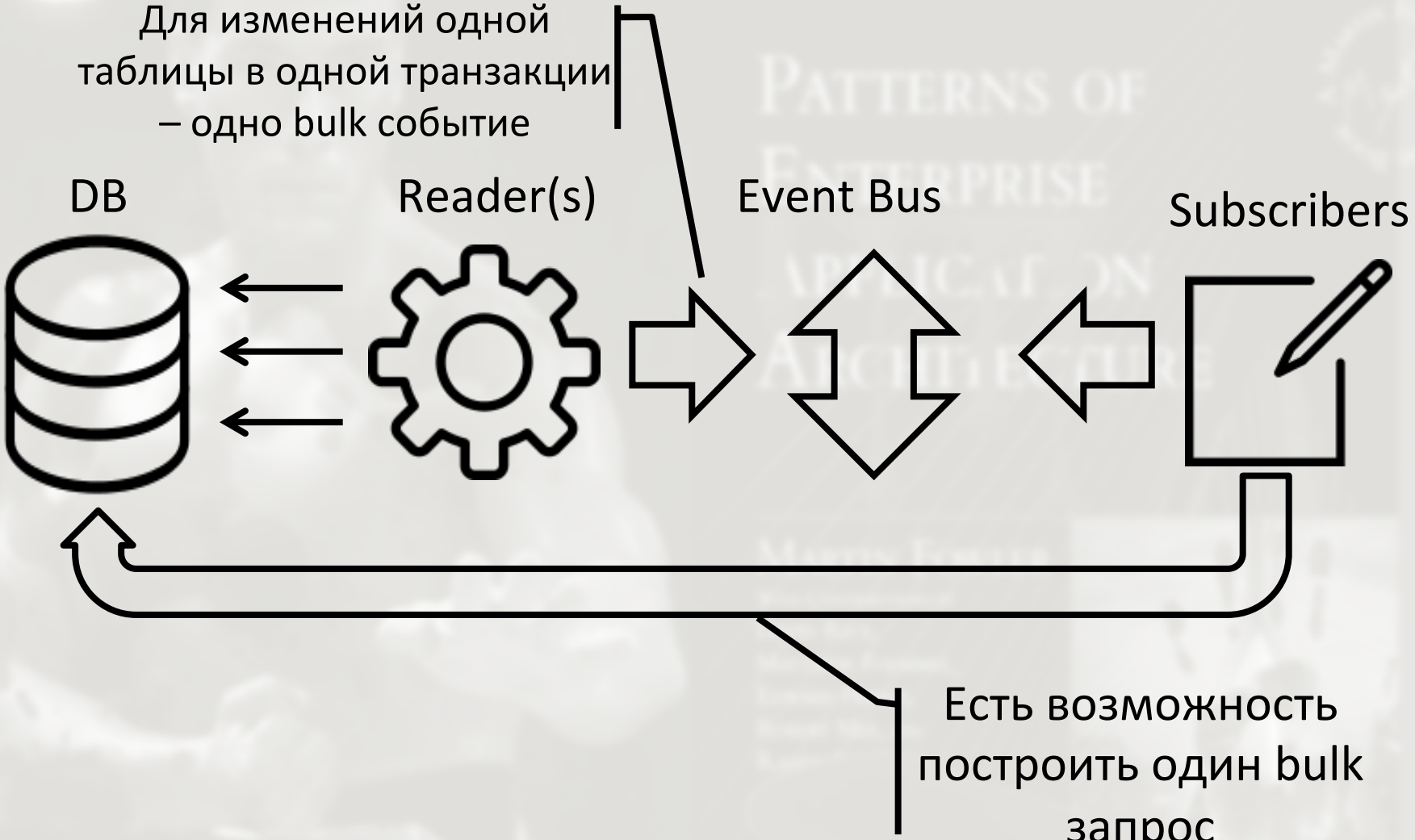
3) Повторным запуском занимается отдельный сервис, чтобы не отнимать время у основных обрабатывающих сервисов.

Третий круг

Слишком медленная обработка событий.

Row-by-row события

Bulk события



1/8 финала

Слишком частный пересчет.

Слишком частный пересчет

Когда реакция на изменения – это пересчет какого-то значения. И это значение зависит от большого количество параметров, которые часто меняются.

Polling (опрос) может быть быстрее!

Четвертьфинал

Deployment

Deployment

1) Включить CDC на уровне базы

```
exec sys.sp_cdc_enable_db
```

2) Включить CDC для каждой наблюдаемой таблицей.

```
exec  sys.sp_cdc_enable_table
        @source_schema = N‘<schema name>',
        @source_name = N‘<table name>',
        @role_name = null,
        @supports_net_changes = 0
```

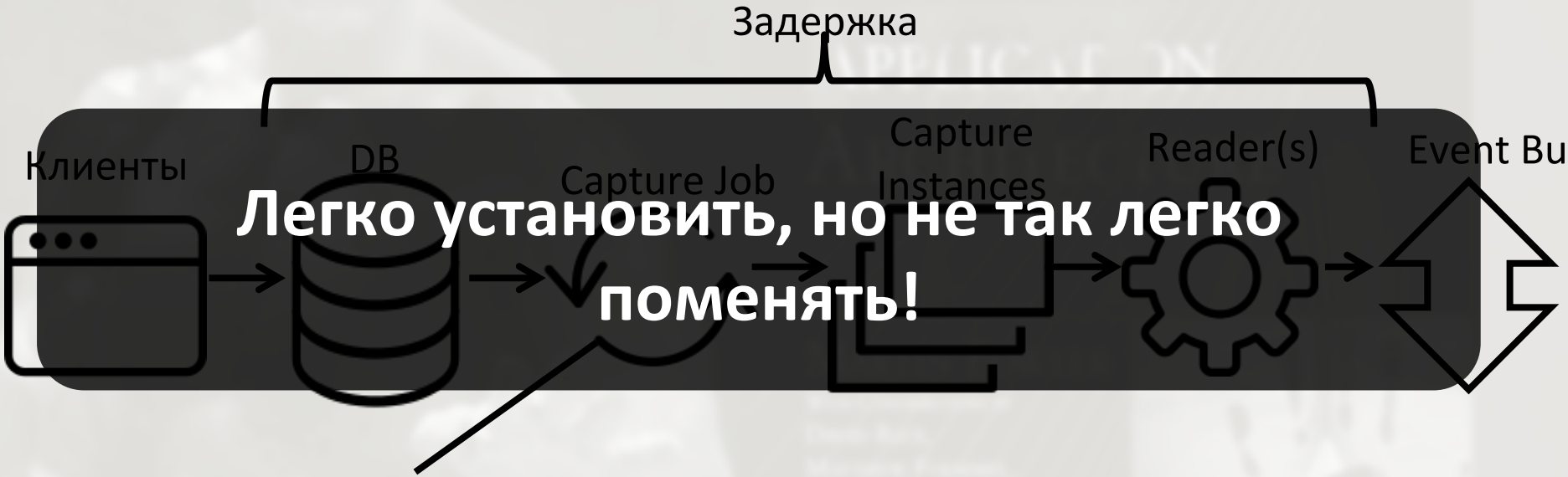
3) Удалить стандартную cleanup SQL job.

4) Установить Cleanup, Failed и Reader сервисы.

В этом прелесть CDC – он очень просто устанавливается!

Alter table?

1) После включения CDC уже не сделать изменений в таблице.
2) А для того, чтобы выключить CDC нужно быть уверенным, что все изменения уже подхвачены.




```
select start_time, scan_phase,
       tran_count, command_count,
       empty_scan_count
from sys.dm_cdc_log_scan_sessions
where start_time > @date
order by start_time desc
```

Полуфинал

Insert и Delete – это не всегда insert и delete!

Где мои обновления???

1) CDC читает transaction Log.
2) Но SQL в некоторых случаях, меняет операцию UPDATE на пару INSERT и DELETE. Например для оптимизации больших операций.
3) И update mask – тоже неверный.

__$start_lsn	__$end_lsn	__$seqval	__$operation	__$update_mask
0x00000043000000067001A	NULL	0x00000043000000670018	2	0x0F
0x00000043000000760003	NULL	0x00000043000000760002	3	0x06
0x00000043000000760003	NULL	0x00000043000000760002	4	0x06

Получается, что нельзя слепо полагаться на поля __$operation и __$update_mask.

Приходится обрабатывать все изменения в транзакции, искать есть ли пара команд INSERT и DELETE, самостоятельно понимать какие поля поменялись.

И это больно … ☹

Финал

Беседа с заказчиком! ☺

Реальный смысл

Это первый шаг к уходу от монолитной системы.

Финал


В целом CDC хорошая функциональность
Но её нужно поддерживать
Поэтому оставим текущий вариант!


Это вам не Голливуд!!!
Это кровавый Enterprise!!!


The End!