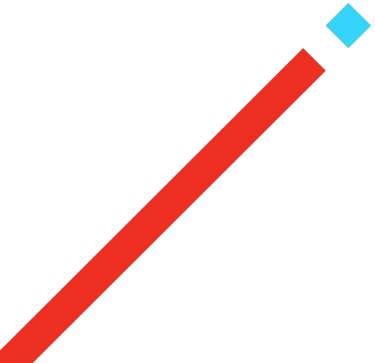
Breaking logs

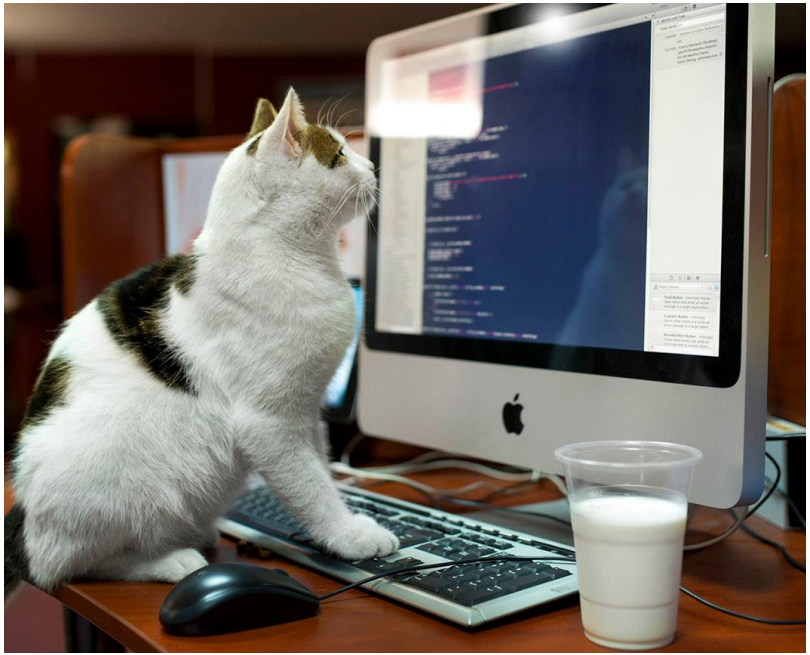
Сломай логи



alfalab

Привет!

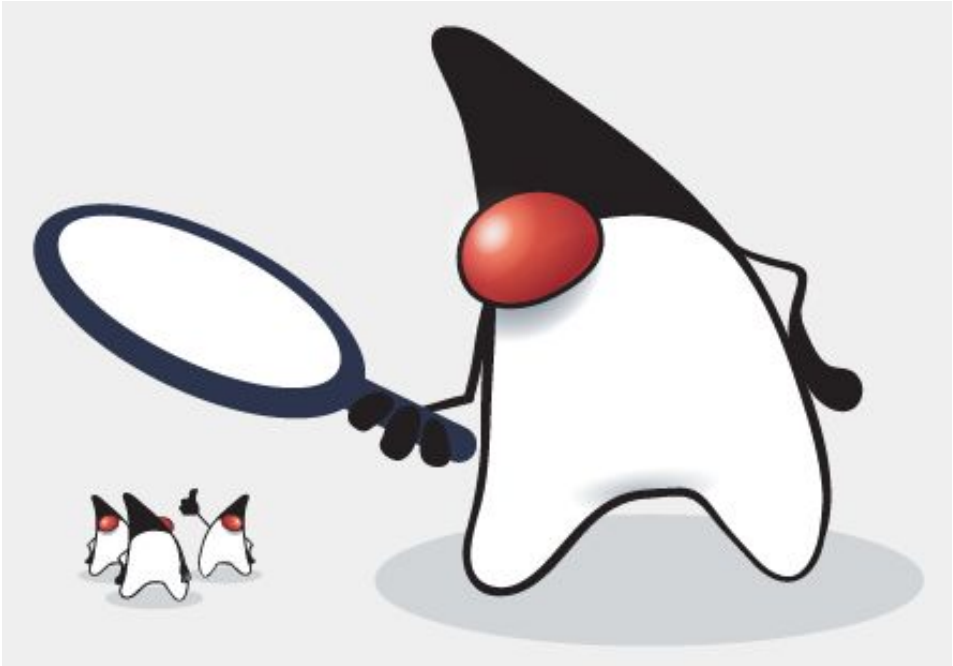
Я разработчик в Альфа-Лаборатории



+

alfalab

Что мы делаем в лабе?



#microservices

Что мы делаем в лабе?

#research

Что мы делаем в лабе?

#proofOfConcept

По результатам пилота:

Внедряем в prod то, что приносит value

Структура доклада

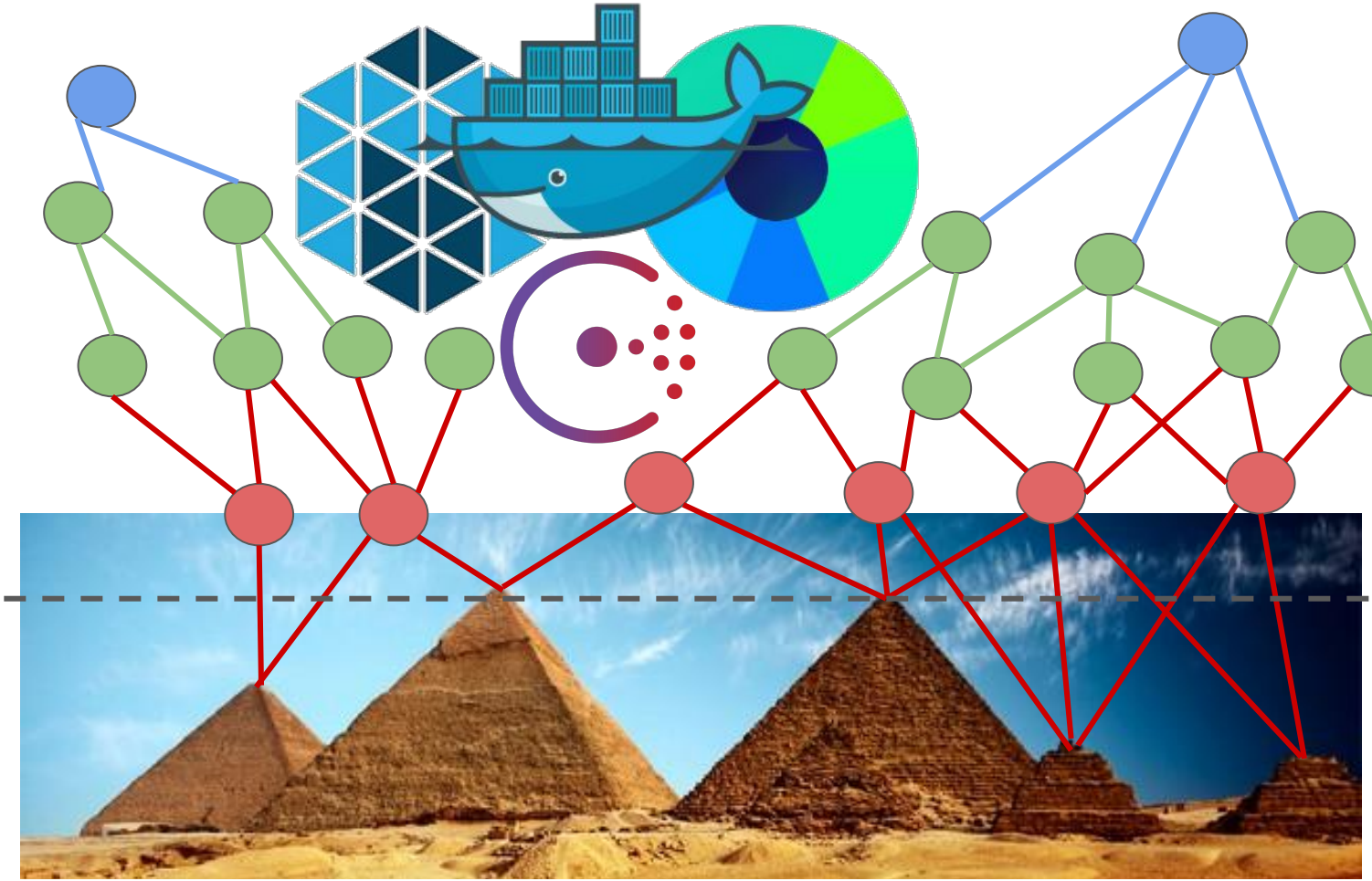
- Архитектура приложений
- Система логирования
- Проблемы
- Решение
- Демо
- Задача анализа логов

MicroServices

Front

Middle

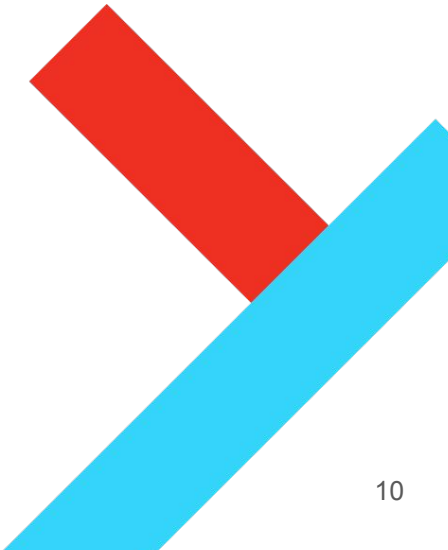
Back



Пример

Приложение Dashboard:

- Информация о клиенте
- Последние транзакции



Уже есть

Возвращает информацию о клиенте

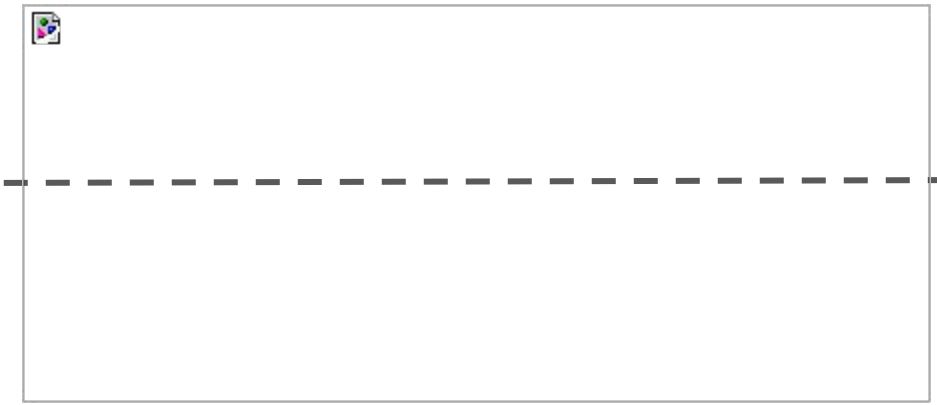
Уже есть

Возвращает информацию о клиенте

Список транзакций по счетам

Создаём

Переиспользуем



Создаём

Переиспользуем

Создаём

Переиспользуем

Service Discovery, Service Registration, Routing

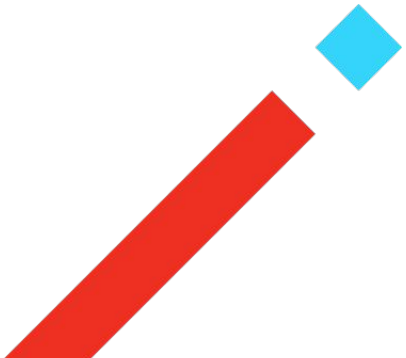
Что получается?

Много проектных приложений

Много инфраструктурных приложений

Все они активно пишут логи

Сейчас в день ~100 гб логов



Сейчас в день ~100 гб логов

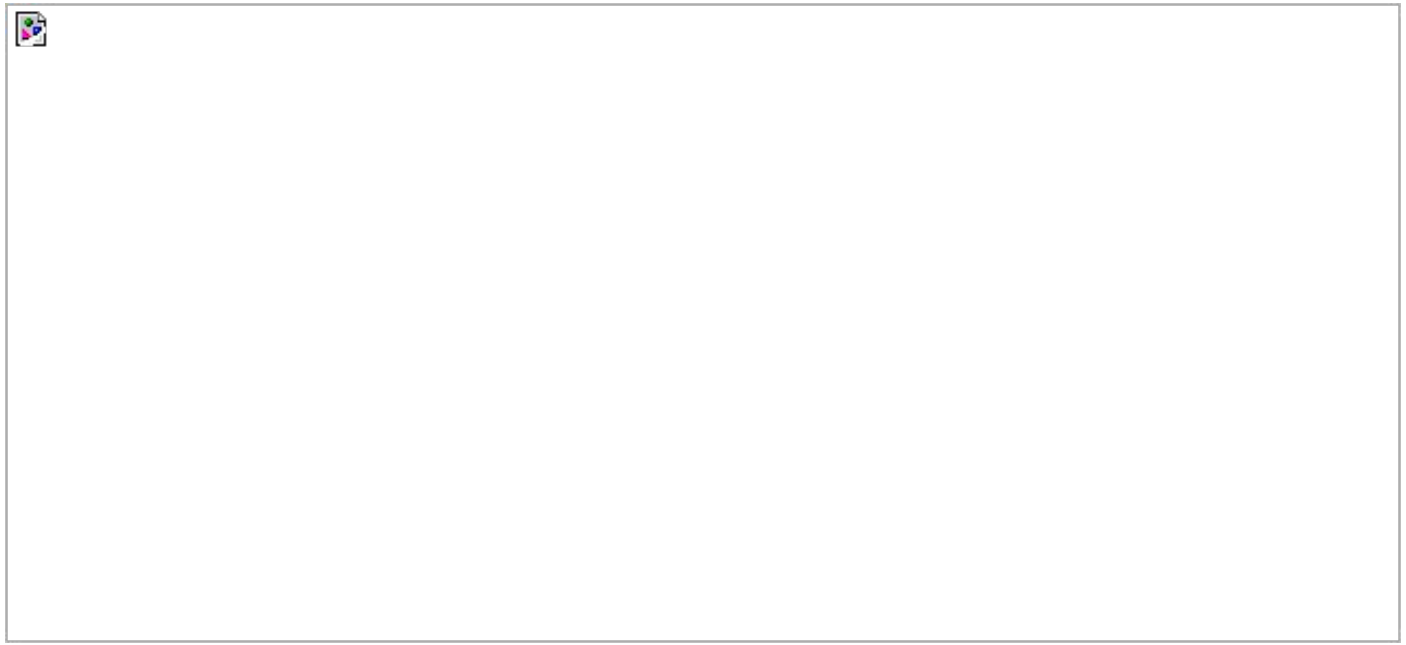
Может уменьшить количество логов?

Приложения должны логировать все вызовы

Inbound/Outbound сообщения

Нельзя уменьшить количество логов - требование отдела безопасности

Как выглядит простая система логирования?



Минимум - централизация логов

Хотим поиск

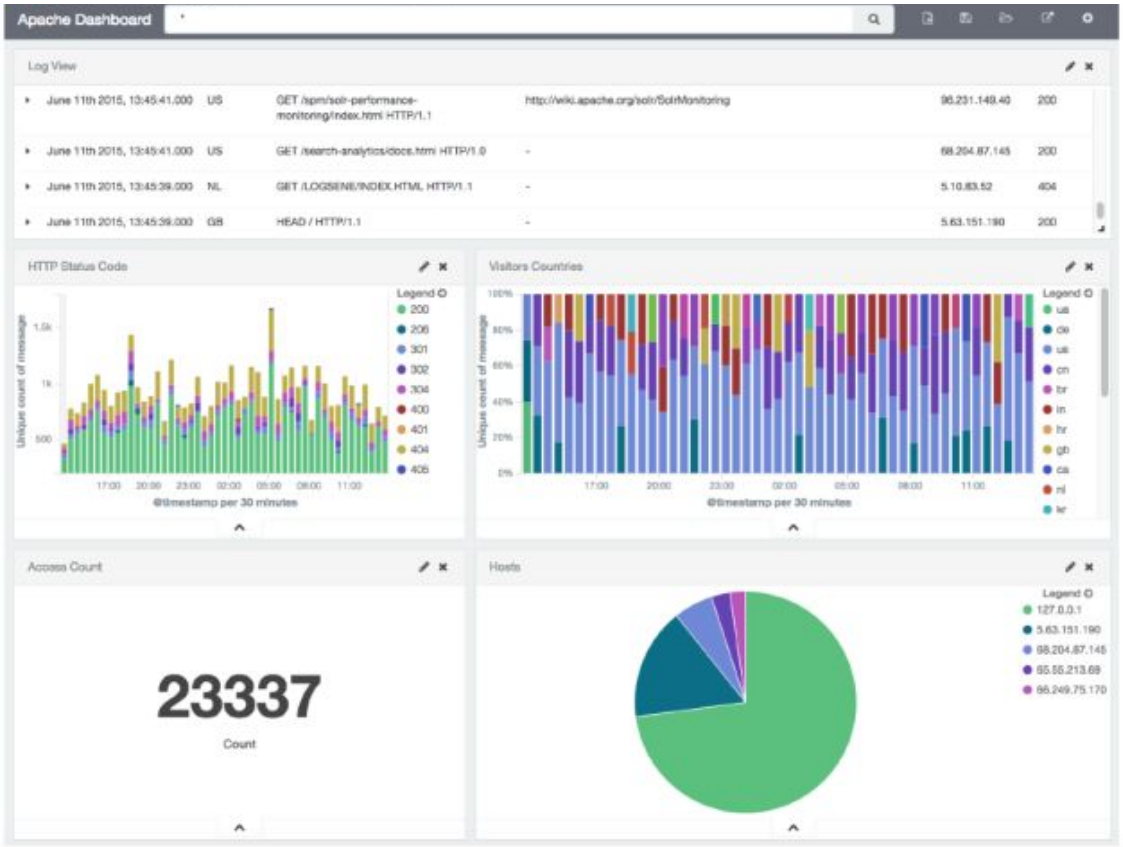
Чтобы искать сообщения с NPE по всем логам!

Хочешь поиск?

Есть же grep!

tail 300 -f /var/log/syslog | grep ...

Хотим дашборд



Что нужно для дашборда?

Поиск и аналитика

Что нужно для дашборда?

Поиск и аналитика

elastic

Что нужно для дашборда?

Поиск и аналитика

elastic

Визуализация

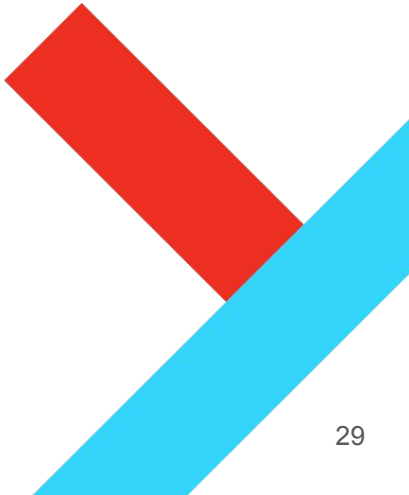
Что нужно для дашборда?

Поиск и аналитика

elastic

Визуализация

Всё готово можно в продакшн?



Формат логов

От syslog:

```
Sep 21 23:48:17 Ilyas-MacBook-Pro ilyasergeev[22839]: Hello, joker.
```

Для ElasticSearch:

```
{
      "message" => "Hello, joker.",
      "@version" => "1",
      "@timestamp" => "2016-09-21T23:49:16.4752",
      "host" => "Ilyas-MacBook-Pro"
}
```

Индексы ElasticSearch

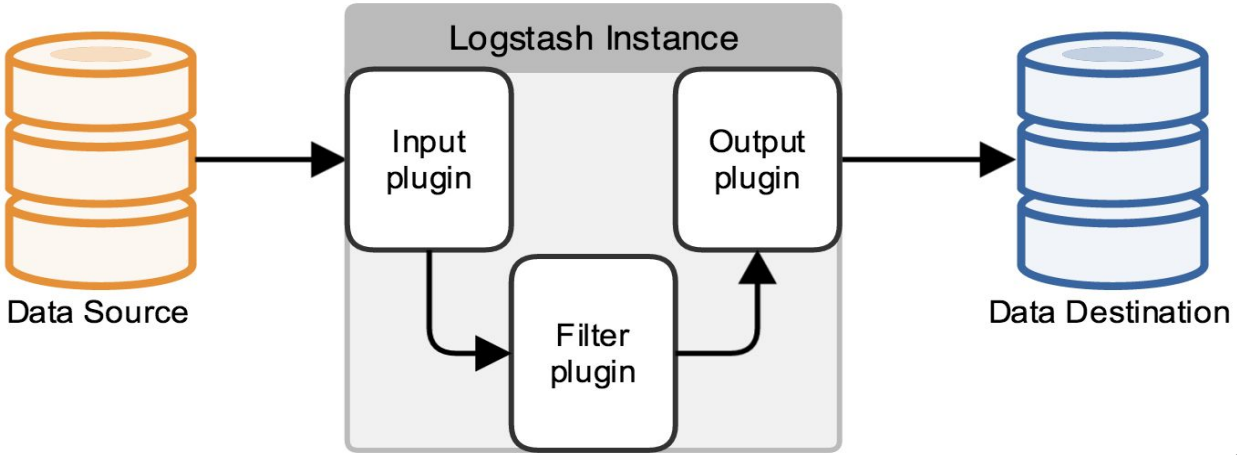
Автоматизировать добавление индексов в ElasticSearch

Создавать индексы на каждый день

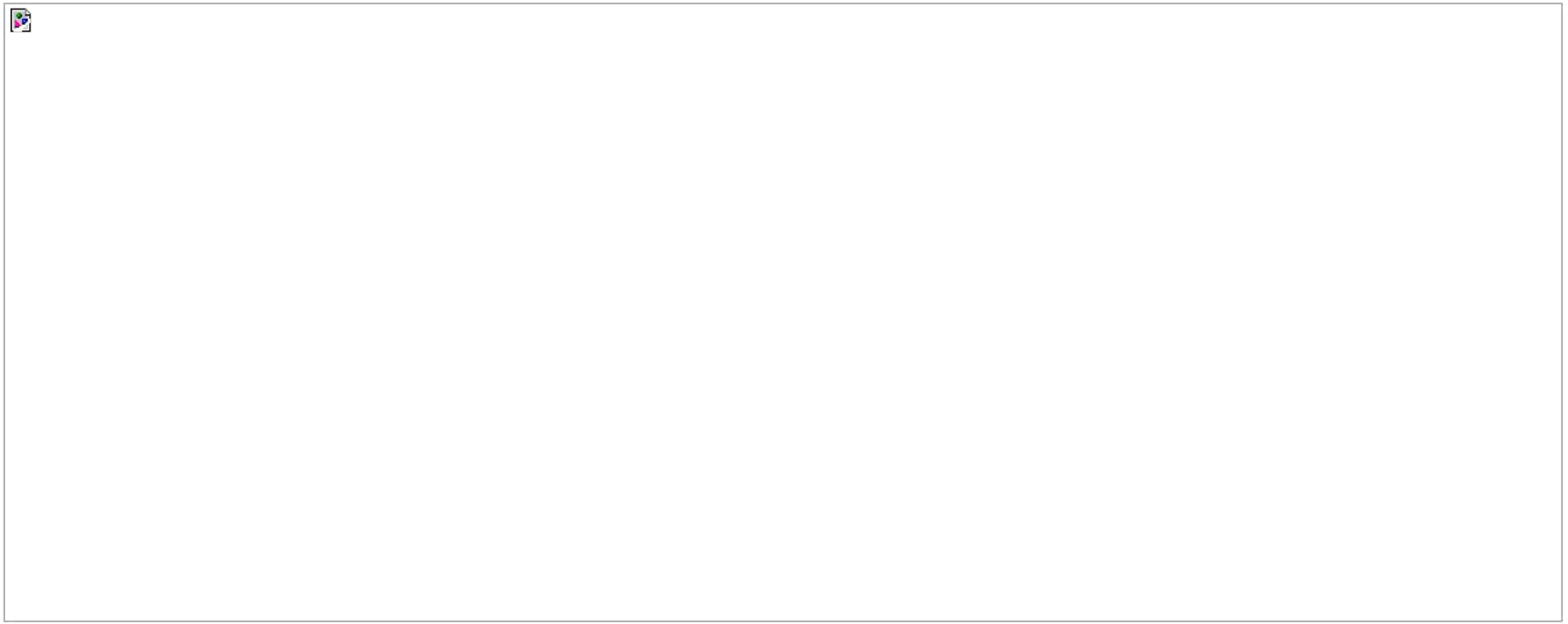
Logstash

Real-Time pipeline для данных

Много плагинов



Как теперь выглядит система логирования?



Требования отдела безопасности

Нужно хранить логи 5 лет

База отдела безопасности

Где у нас сейчас хранятся логи

В индексах ElasticSearch

Может тут хранить пять лет?

Индексы

Открытые / Закрытые

Открытые - можно читать, писать
Закрытые - просто хранят информацию

Индексы

Открытые / Закрытые

Открытые - можно читать, писать
Закрытые - просто хранят информацию

```
curl -XPOST 'localhost:9200/my_index/_close'
curl -XPOST 'localhost:9200/my_index/_open'
```

Открытые индексы потребляют ресурсы кластера

Храним только открытые индексы

Период хранения логов

45 дней

45 дней

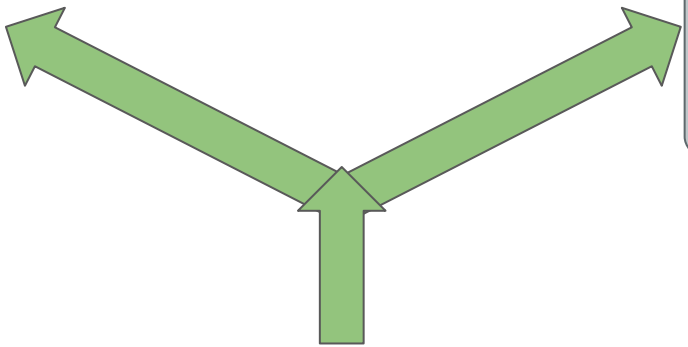
5 лет

периодическая
выгрузка логов

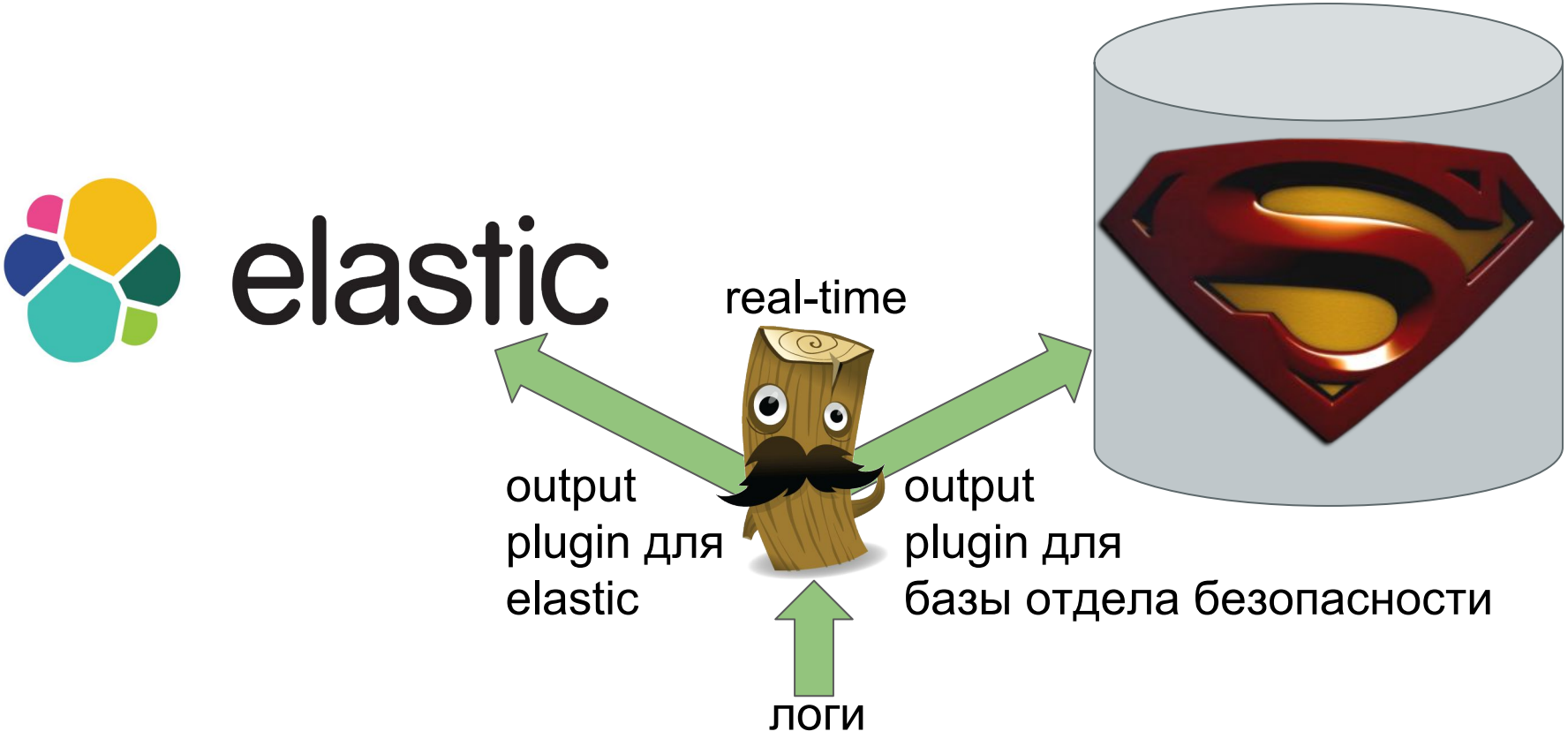
велосипед

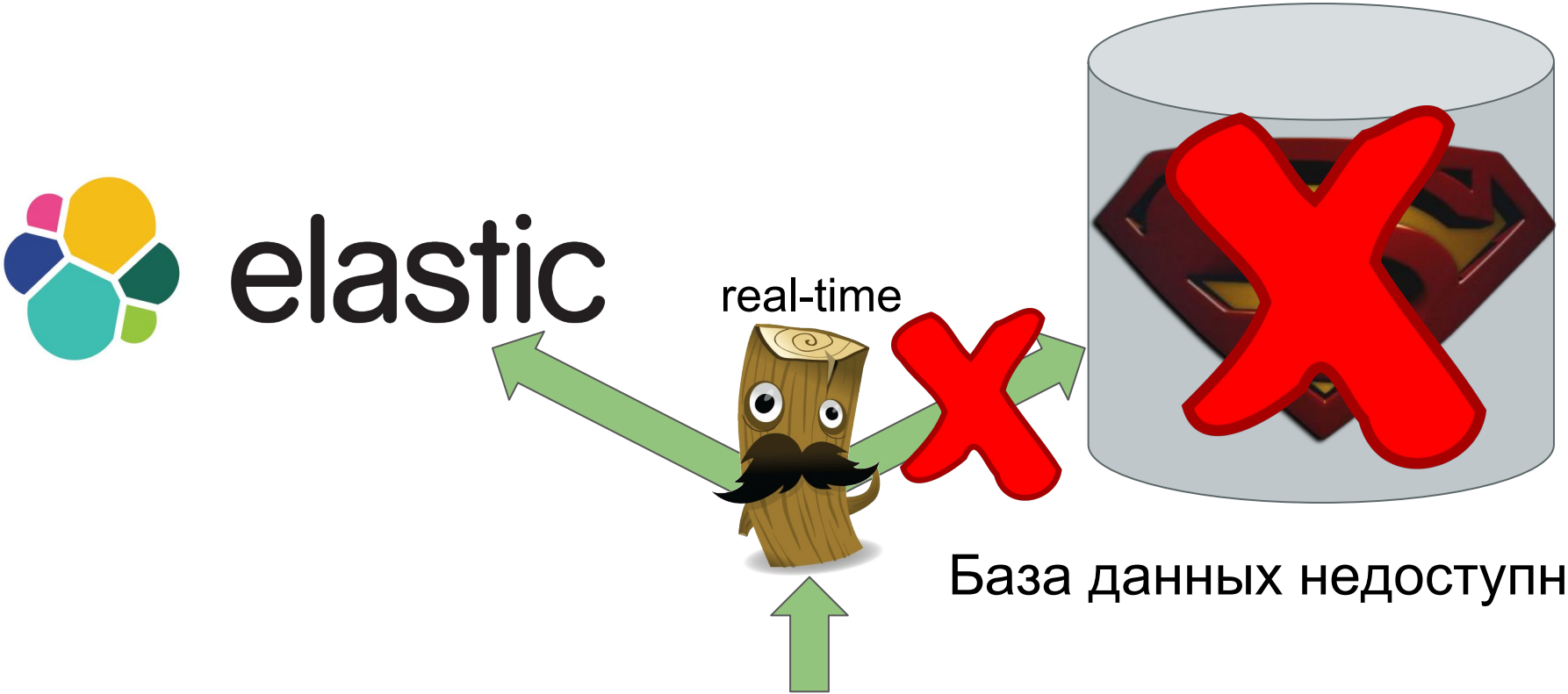
одноразовое решение

elastic



логи


elastic
real-time
output
plugin для
elastic
output
plugin для
базы отдела безопасности
логи


elastic
real-time
База данных недоступна

Решение в общем случае

Давайте добавим ещё один слой абстракции

?

elastic

APACHE Spark™

Логи хранятся n дней

Логи хранятся n дней

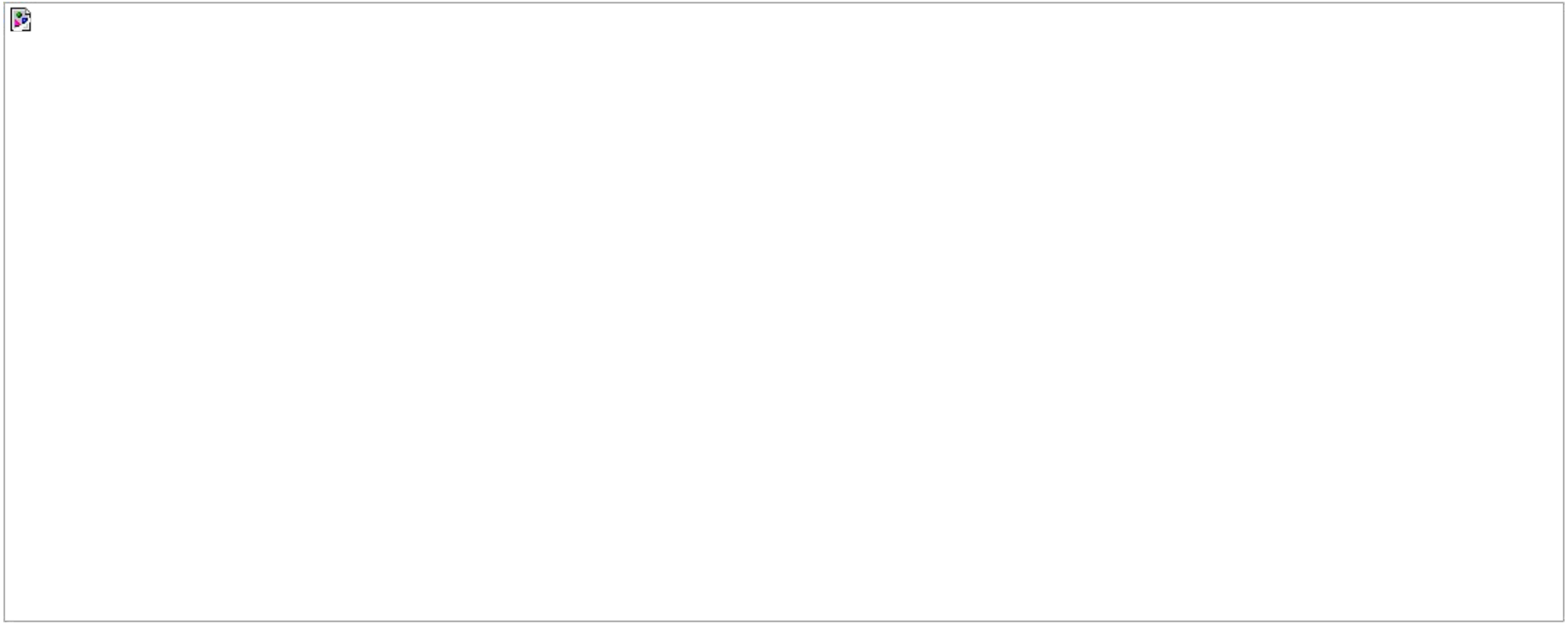
Легко добавлять потребителей

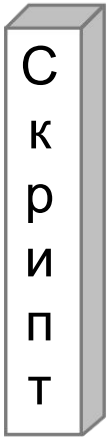
elastic

APACHE Spark™

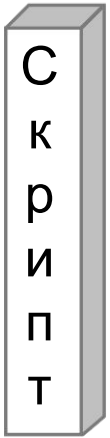
Стоит ли добавлять промежуточную систему ради решения одной проблемы?

Мы рассказали о нашей системе логирования



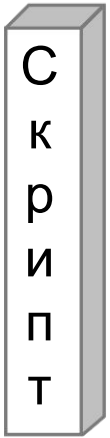


Скрипт отдела поддержки

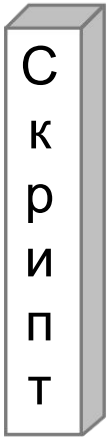


Скрипт отдела поддержки
для мониторинга количества заходов в приложения
использует API

elastic

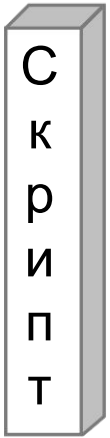


Делает периодические запросы



Делает периодические запросы

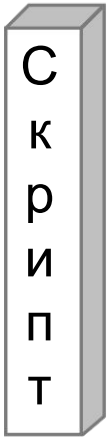
Полнотекстовый поиск и агрегации



Делает периодические запросы

Полнотекстовый поиск и агрегации

Запрос по всем индексам

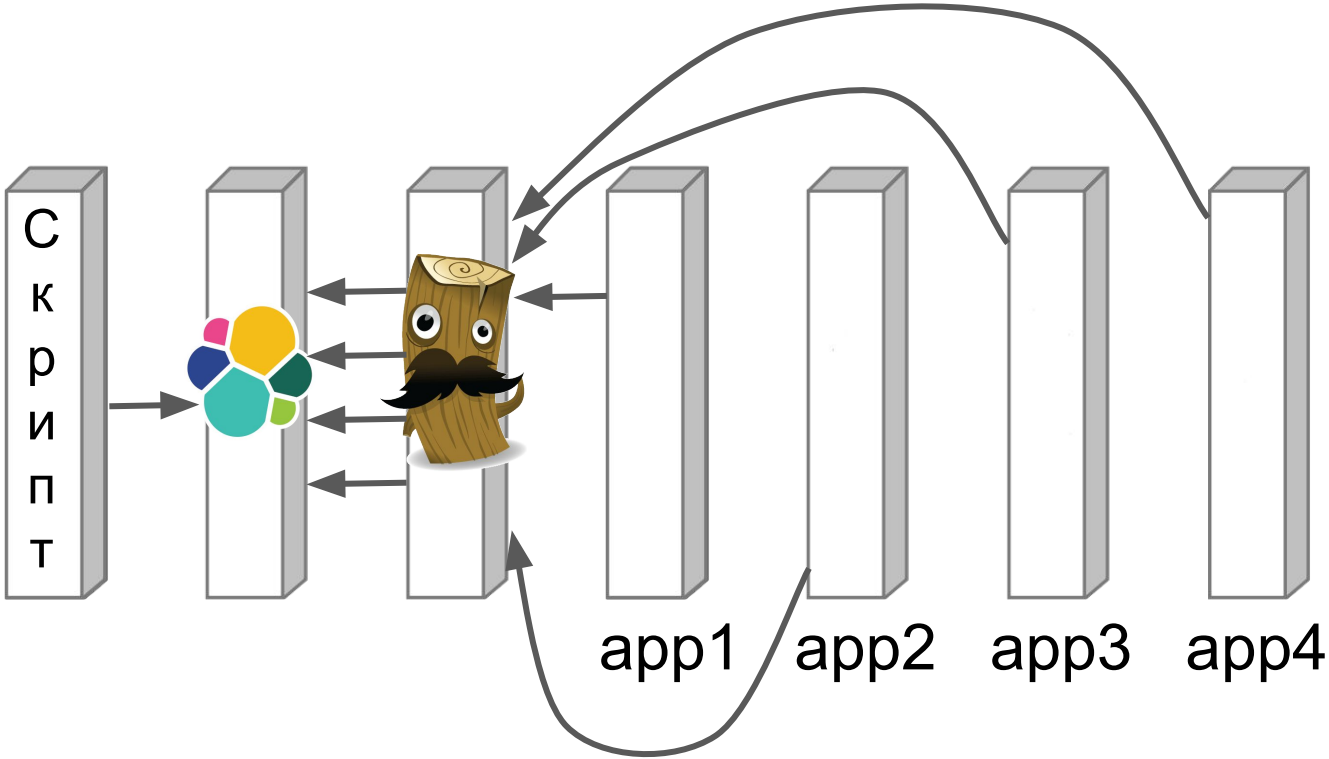


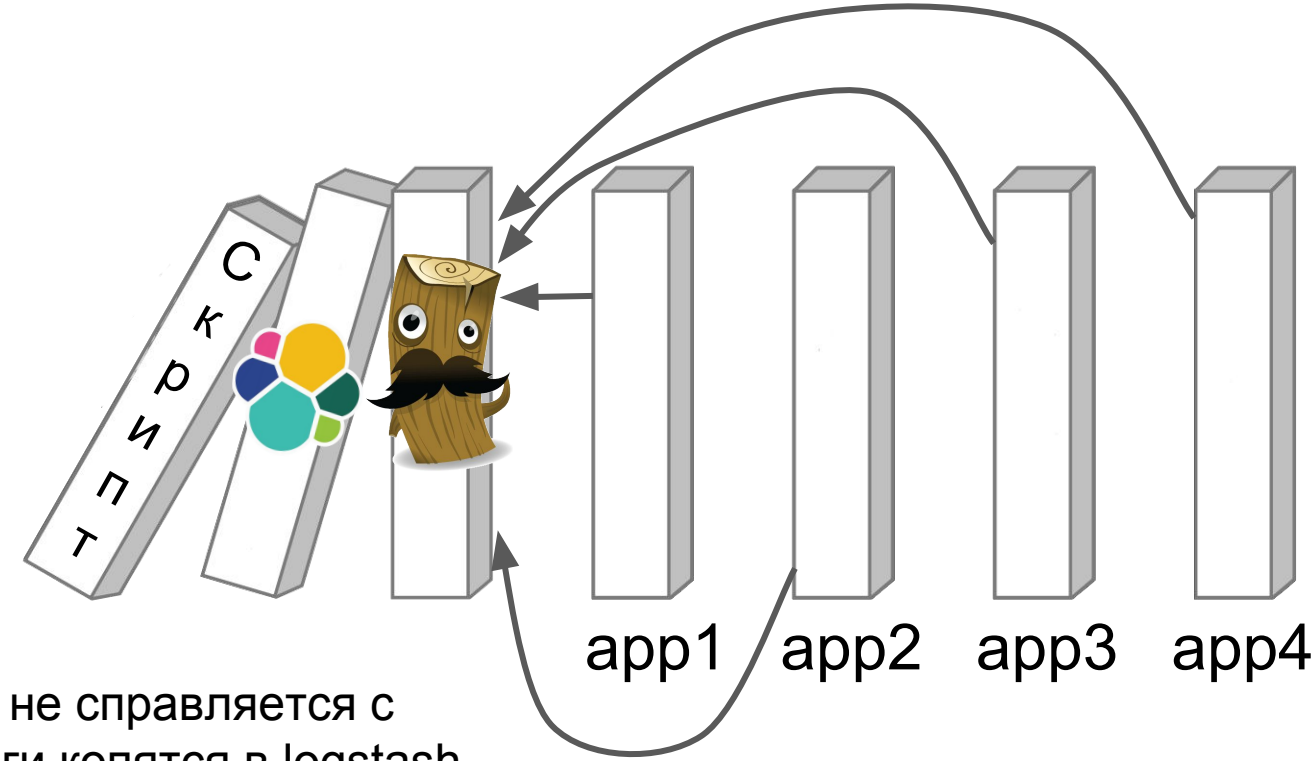
Делает периодические запросы

Полнотекстовый поиск и агрегации

Запрос по всем индексам

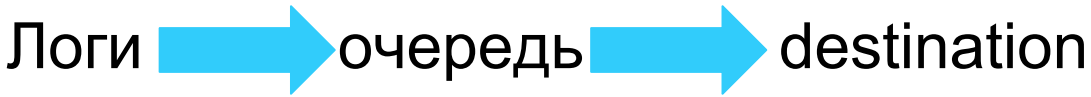
Период - 10 секунд


Скрипт
app1
app2
app3
app4



ElasticSearch не справляется с нагрузкой, логи копятся в logstash

Как вообще пишутся логи?



Очередь

Сообщения добавляются в очередь если не могут быть доставлены в destination

Очередь

Сообщения добавляются в очередь если не могут быть доставлены в destination

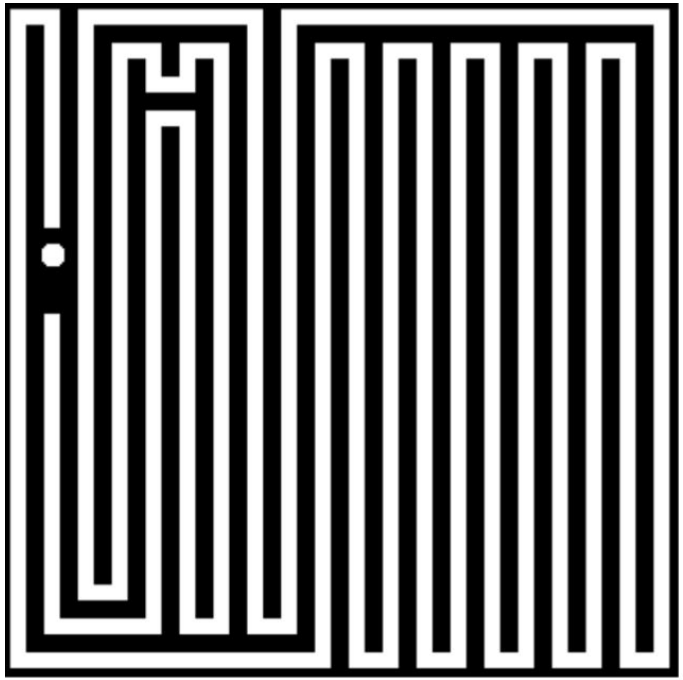
Сообщения не теряются

Очередь

Сообщения добавляются в очередь если не могут быть доставлены в destination

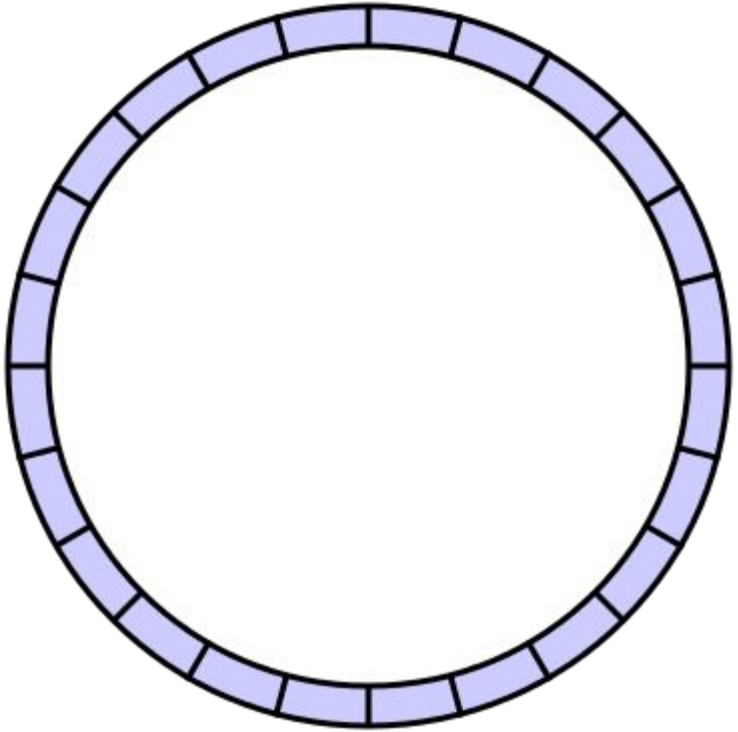
Сообщения не теряются

Ограничение по размеру - объём свободной памяти



Ещё одно сообщение и OutOfMemory

Ring-buffer



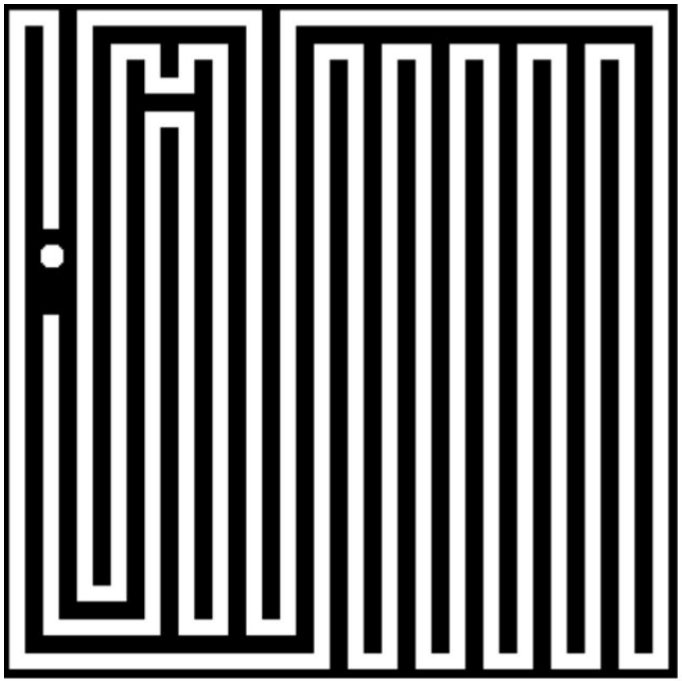
Фиксированный размер

Данные перезаписываются

Ring-buffer

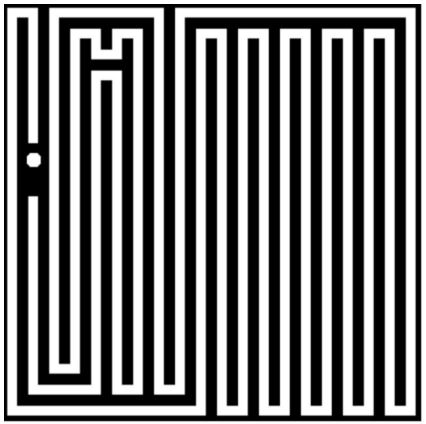
Фиксированный размер

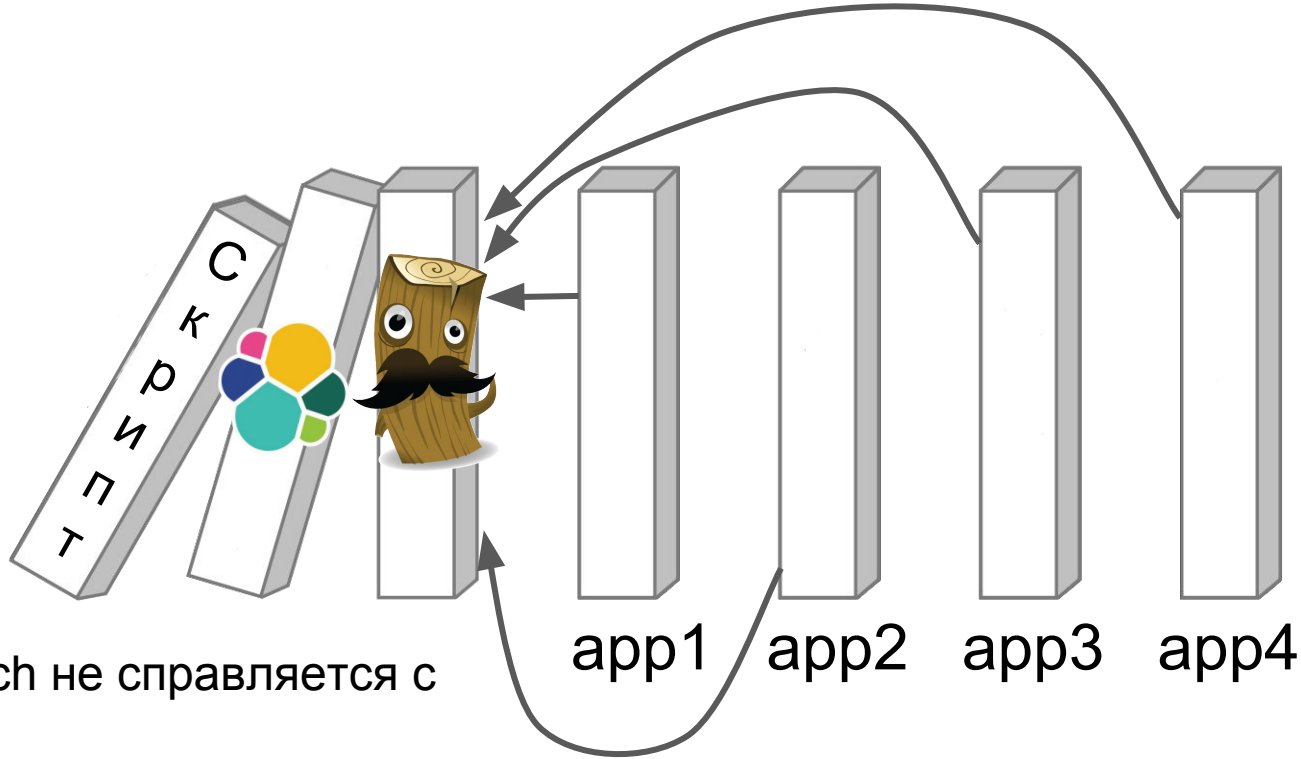
Данные перезаписываются



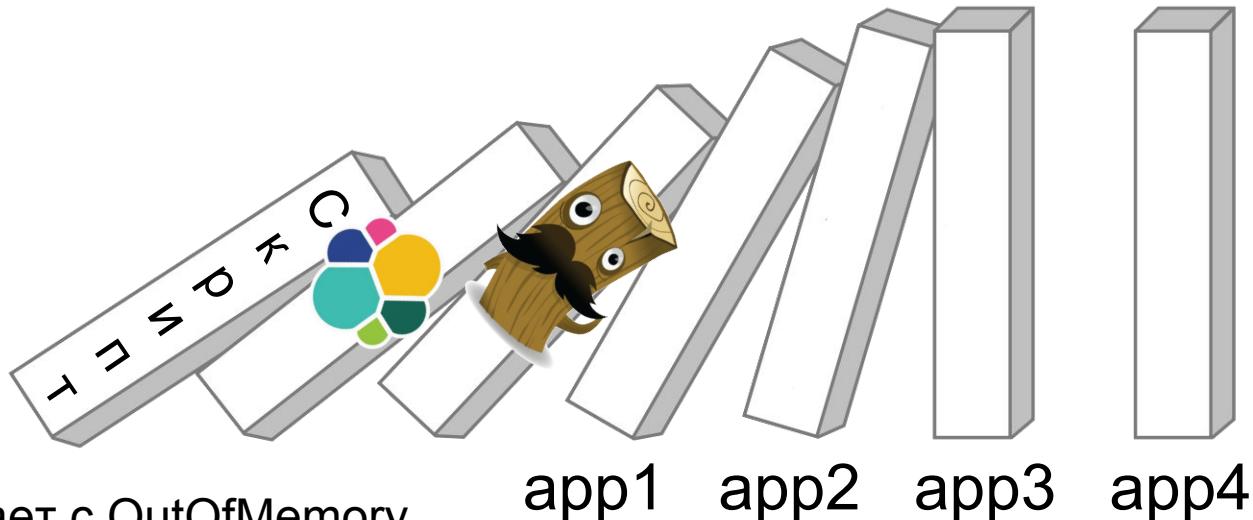
или

Требование отдела безопасности:
Логи не должны теряться, поэтому





ElasticSearch не справляется с нагрузкой
Логи копятся в Logstash



Logstash падает с OutOfMemory
Логи копятся в приложениях

Используем Ring-buffer

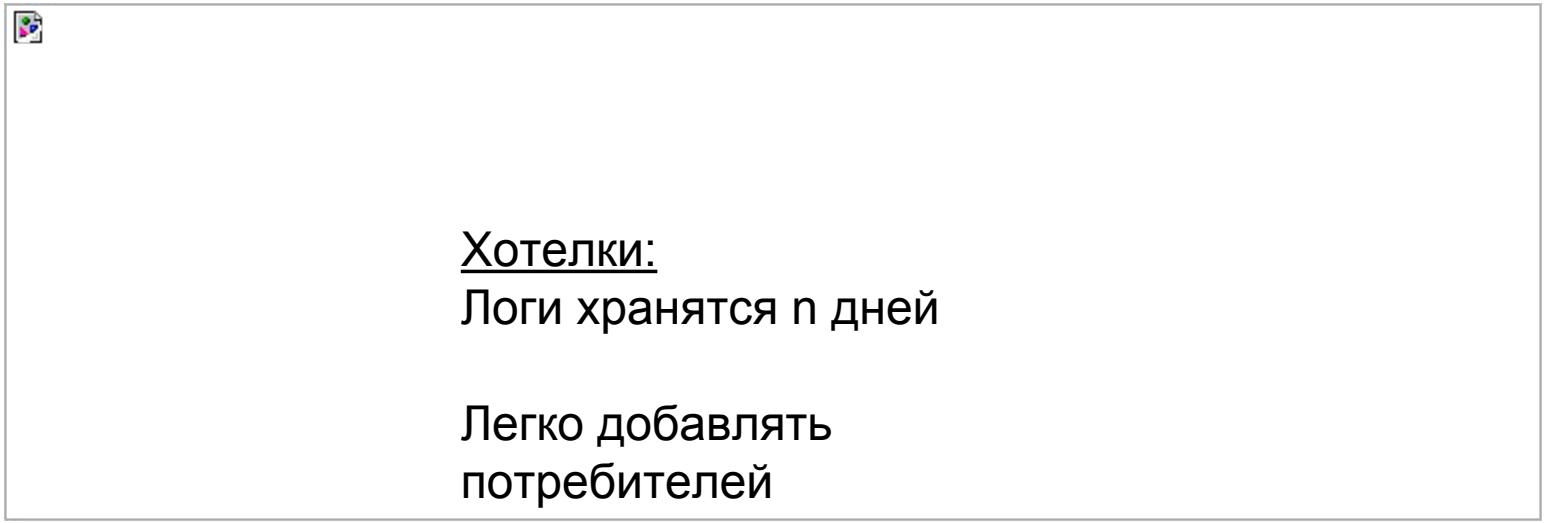
Что делать если destination недоступен?

Разделение потоков поставки и потребления

Логи хранятся n дней

Легко добавлять потребителей

Выбор промежуточный системы


Хотелки:
Логи хранятся n дней
Легко добавлять
потребителей

~~База данных~~

Хотелки:
Логи хранятся n дней

Легко добавлять потребителей

Очередь

<u>Хотелки:</u>
Логи хранятся n дней

Легко добавлять
потребителей

Какие есть варианты очередей?

С сохранением сообщений на диск:

ActiveMQ, RabbitMQ, Kestrel, NSQ, Kafka ..

Без:

NATS, ruby-nats ..

Почему kafka?

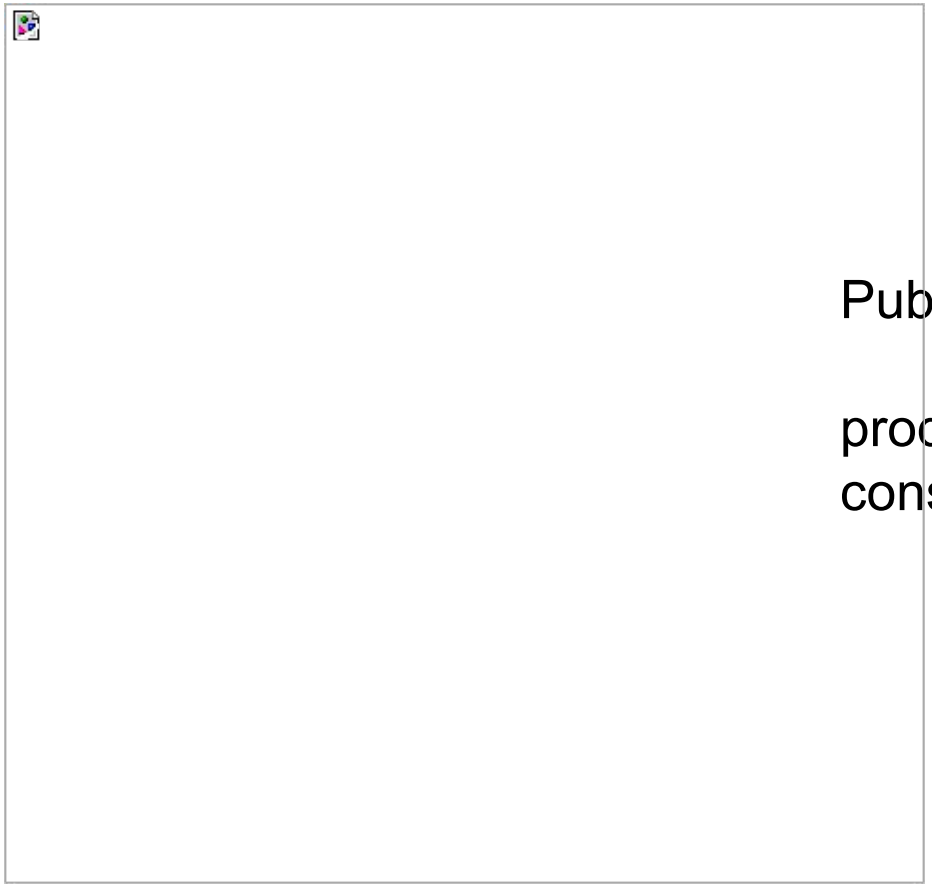
Добавление потребителей не влияет на производительность

Почему kafka?

Добавление потребителей не влияет на производительность

Потребитель должен реализовать Kafka Consumer
или воспользоваться готовым приложением, например:

Верхнеуровневая архитектура



Publish-Subscribe

producer пишет в topic
consumer читает из topic

Чтение из kafka

Нужно знать:

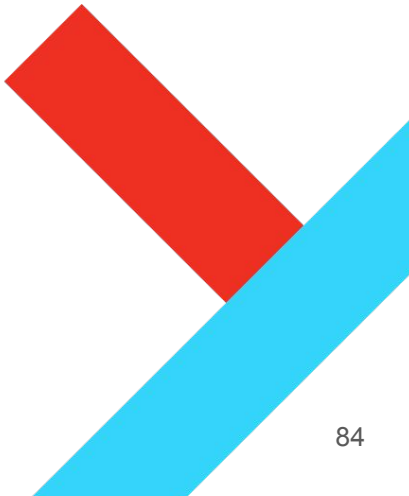
- Адрес сервера kafka
- Название Topic

Чтение из kafka

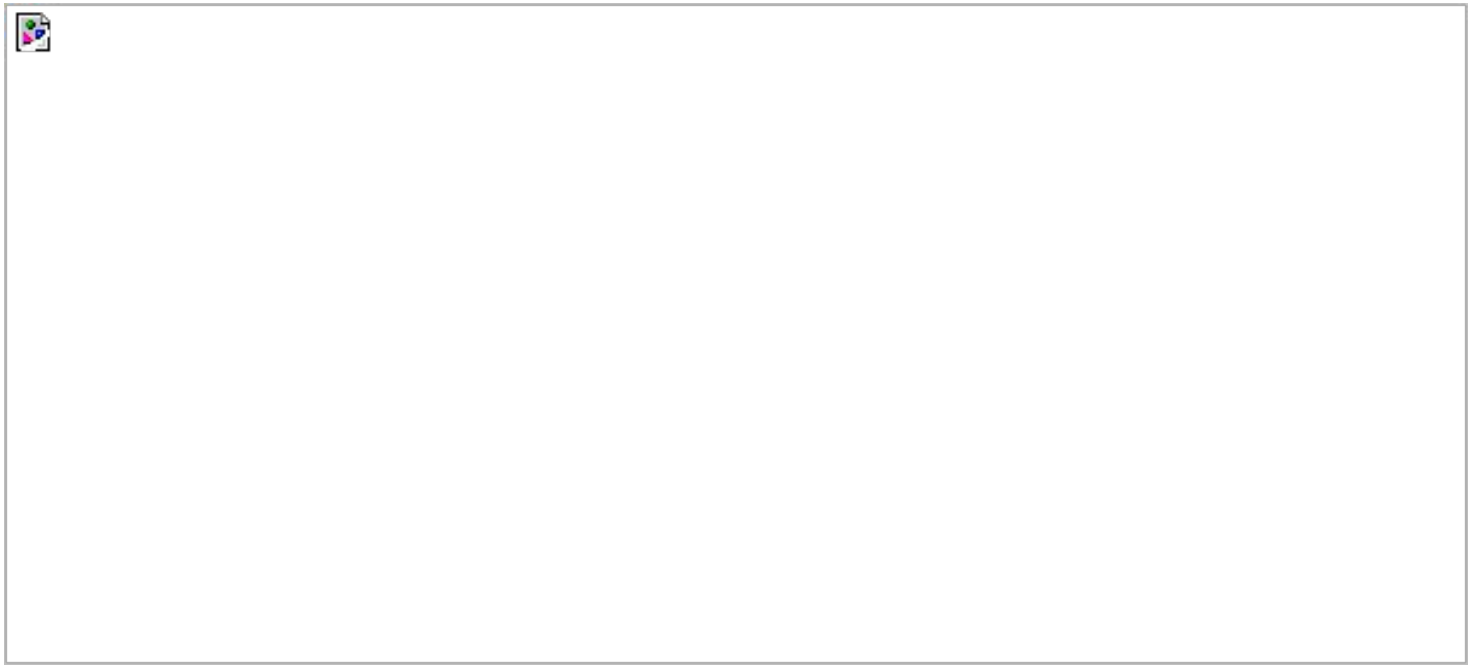
Особенности:

- Offset - индекс с которого начать чтение
- Compaction и Retention

Kafka - это файл который читается в один поток?



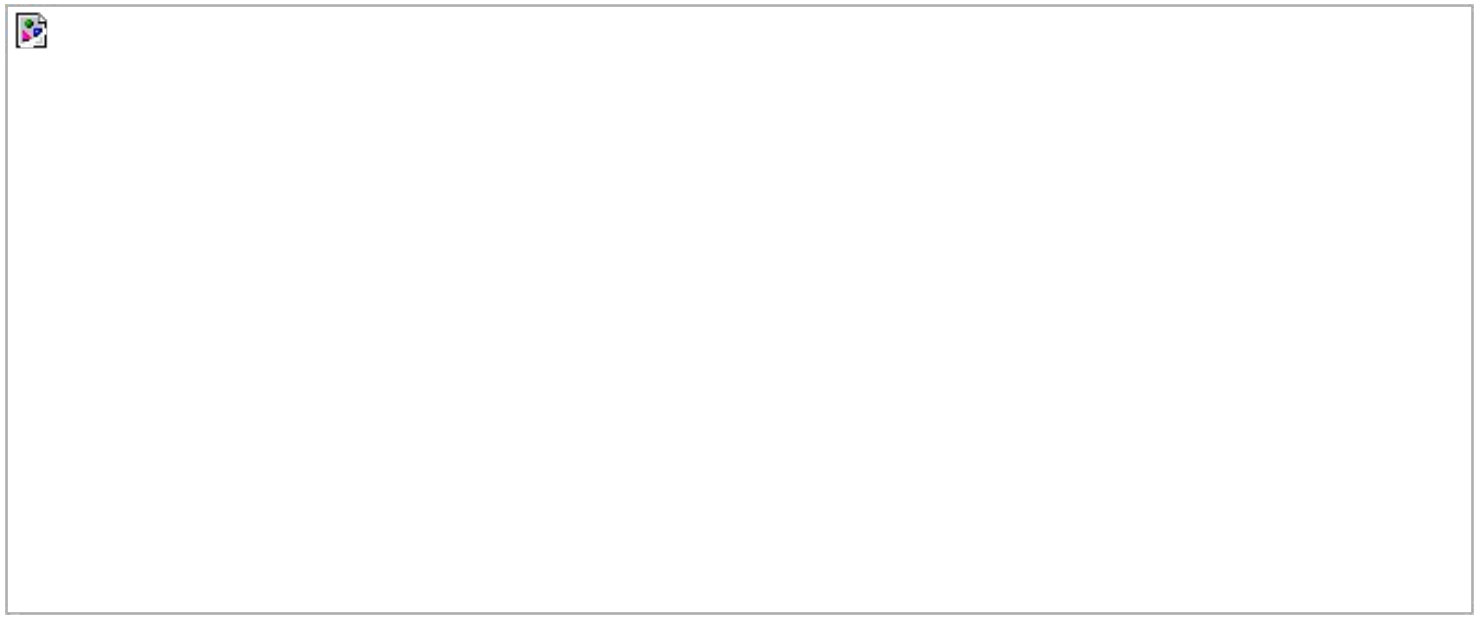
Topic делится на partitions



Параллельное чтение логов

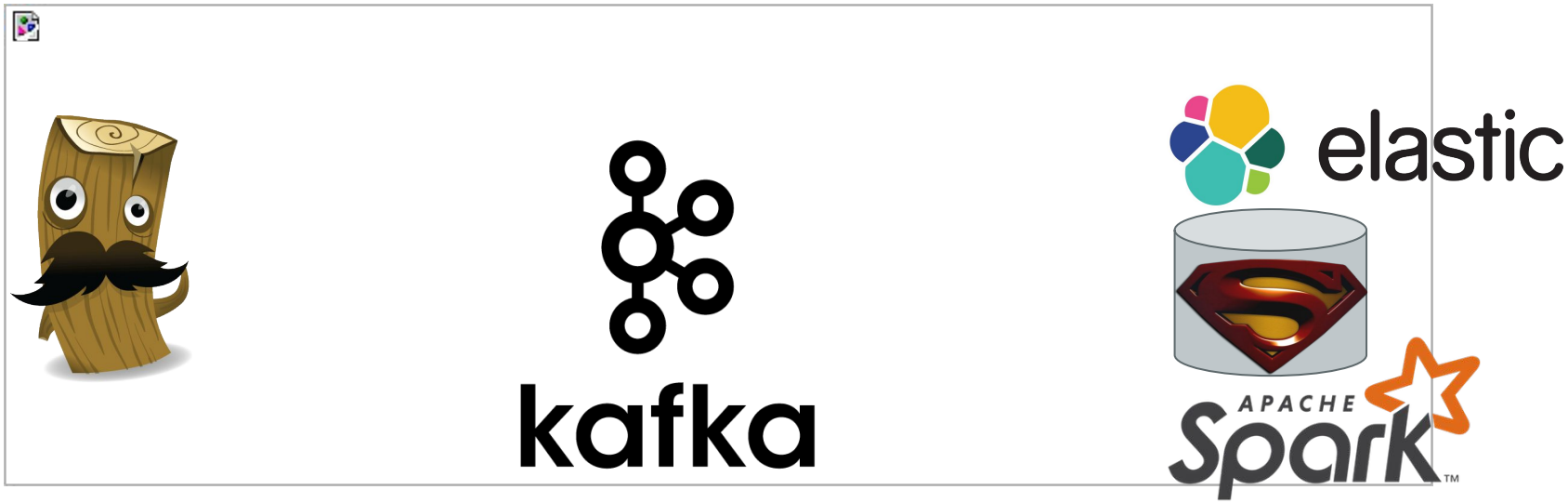
Параллельное чтение логов
Балансировка сообщений в consumer group

Потребление логов из kafka



Решение проблем

~~Один потребитель логов~~
~~Потребитель влияет на поставщика~~



А что с производительностью?

Увеличение потока потребления за счёт добавления partitions

Увеличение входного потока - ?

Partition

Дешёвая операция записи

Параллельная запись в kafka

Partition == unit of parallelism

Partition по ключу

Логи хранятся n дней

Легко интегрироваться

Производительность

Логи хранятся n дней

Легко интегрироваться

Производительность

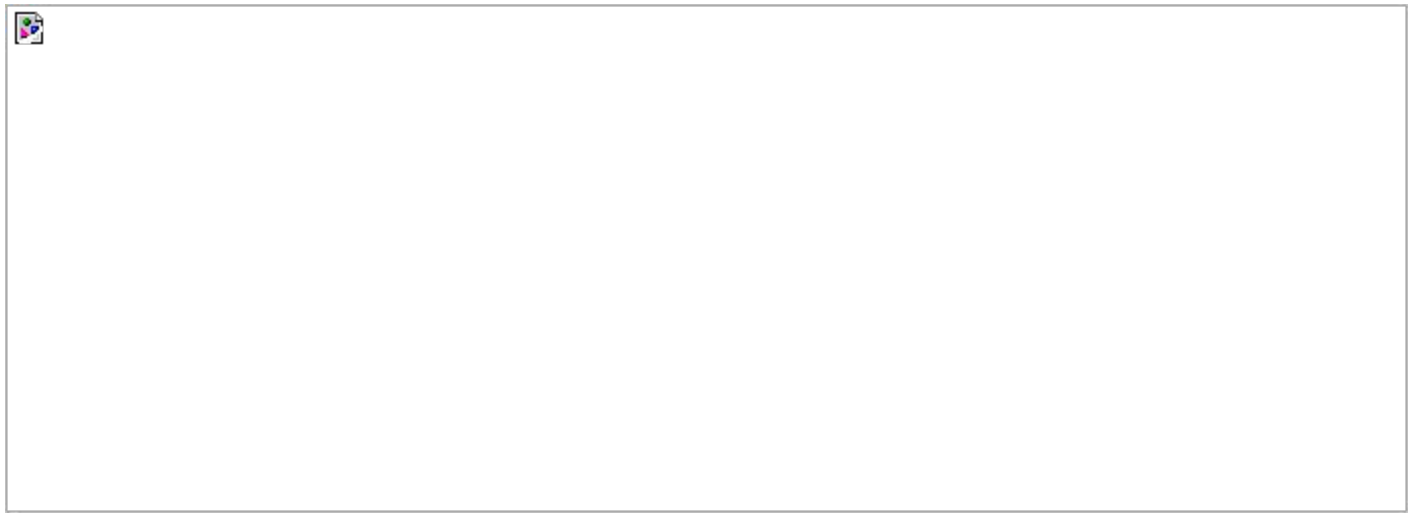
А что с надёжностью?

Надёжность

1. Репликация
2. Failure recovery

Репликация

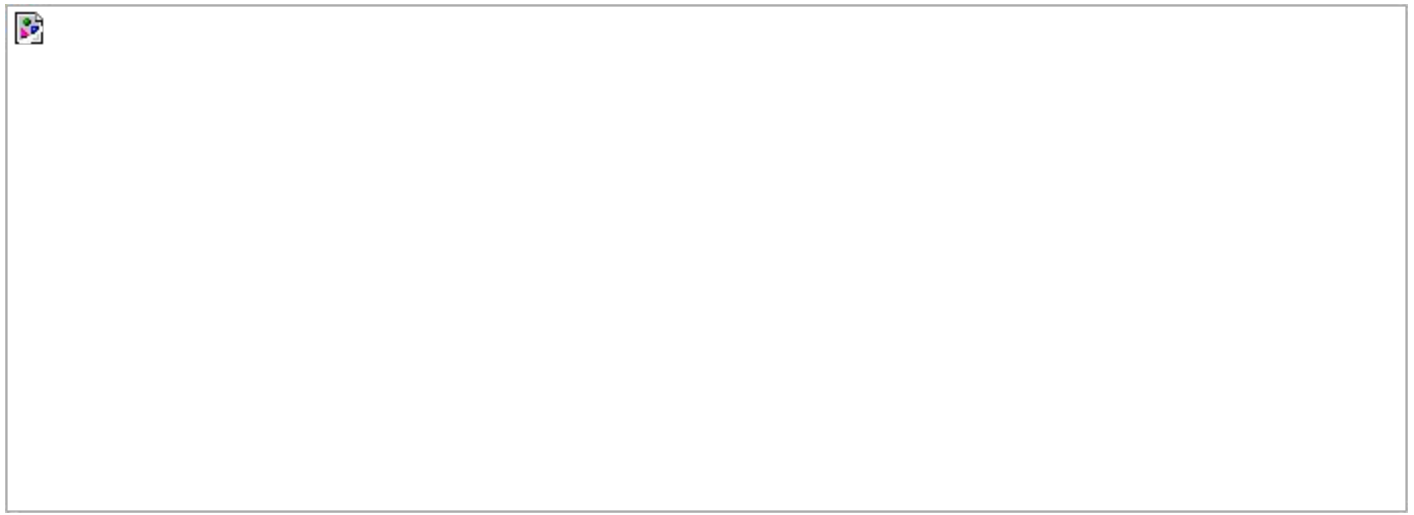
Запрос метаданных + push в leader



Репликация

Запрос метаданных + push в leader

In-sync replica set (ISR)

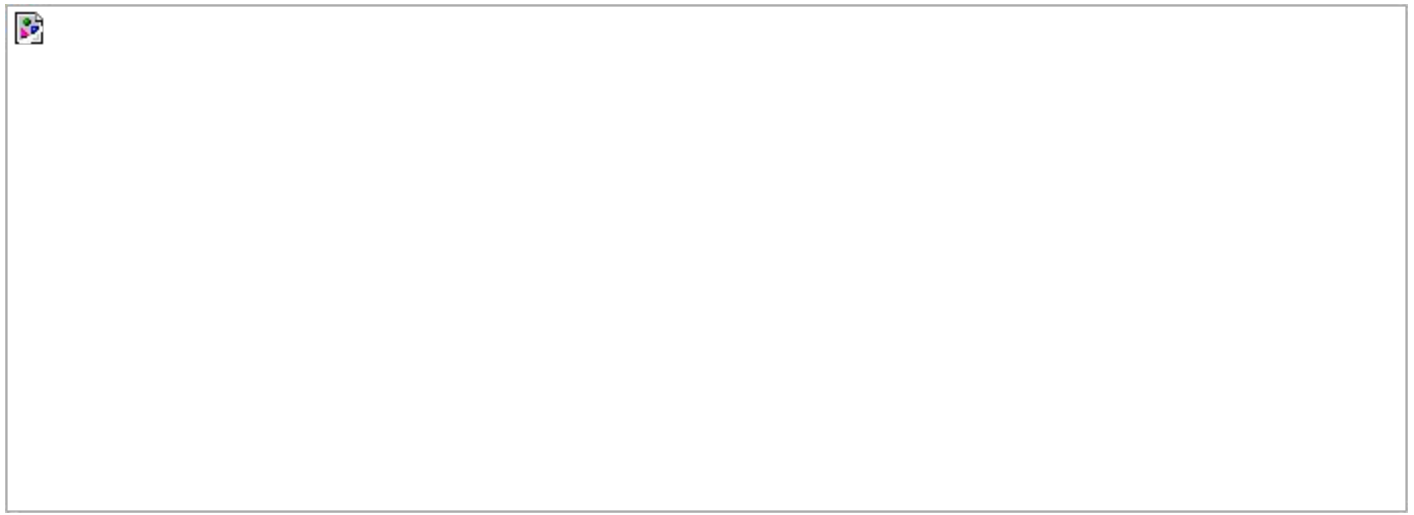


Репликация

Запрос метаданных + push в leader

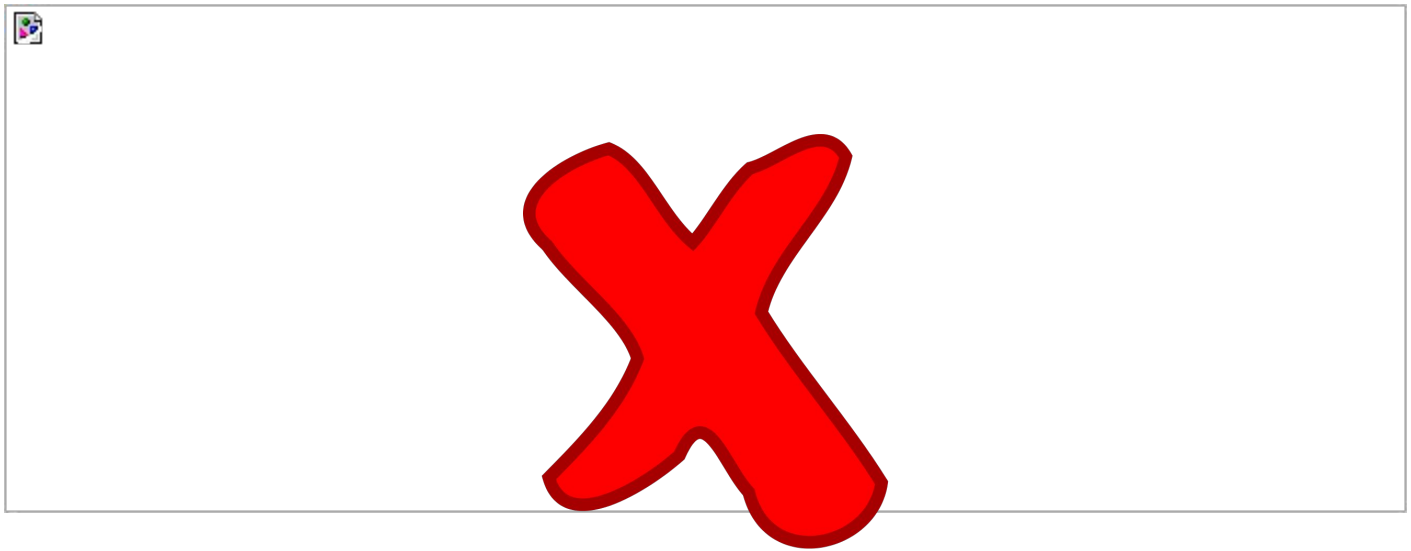
In-sync replica set (ISR)

Commit сообщения



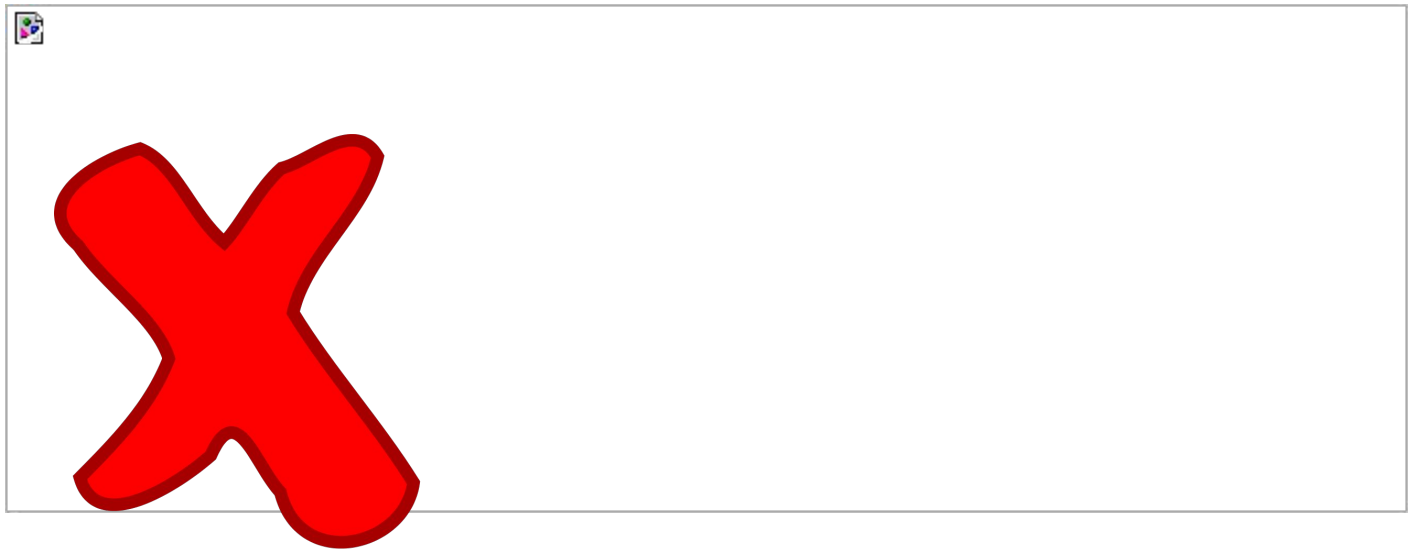
Failure Recovery

Quorum или all (x failures with x+1 ISR)



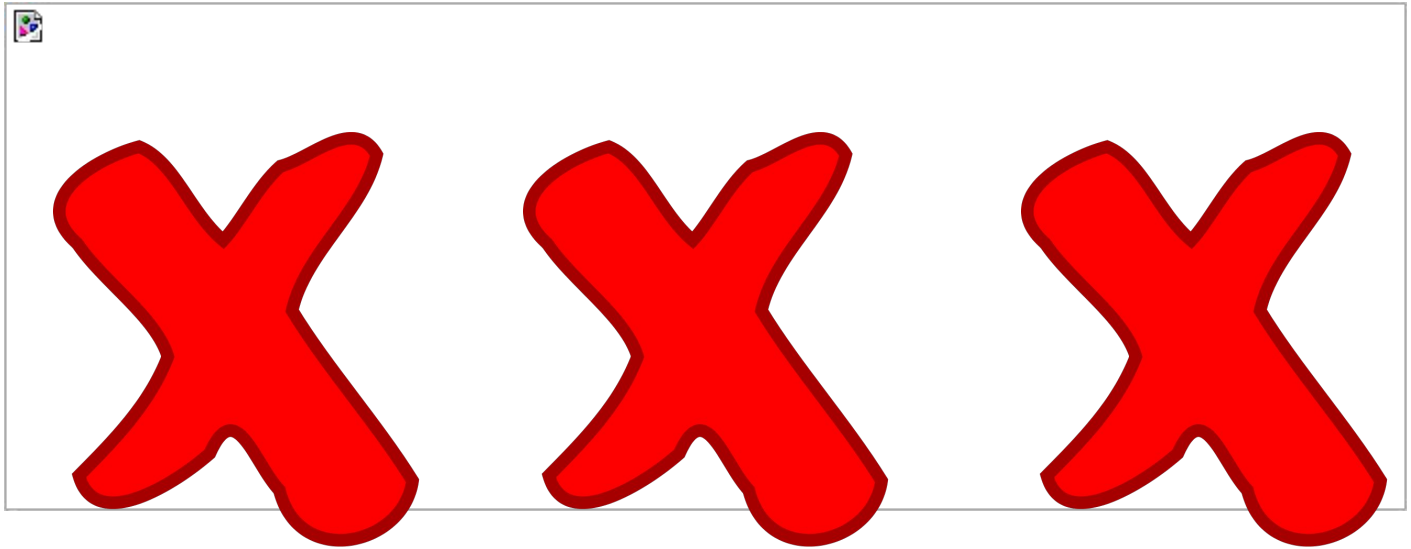
Failure Recovery

Выбор лидера с помощью Zookeeper



Failure Recovery

Все сломались -
дожидаемся возвращения любой реплики

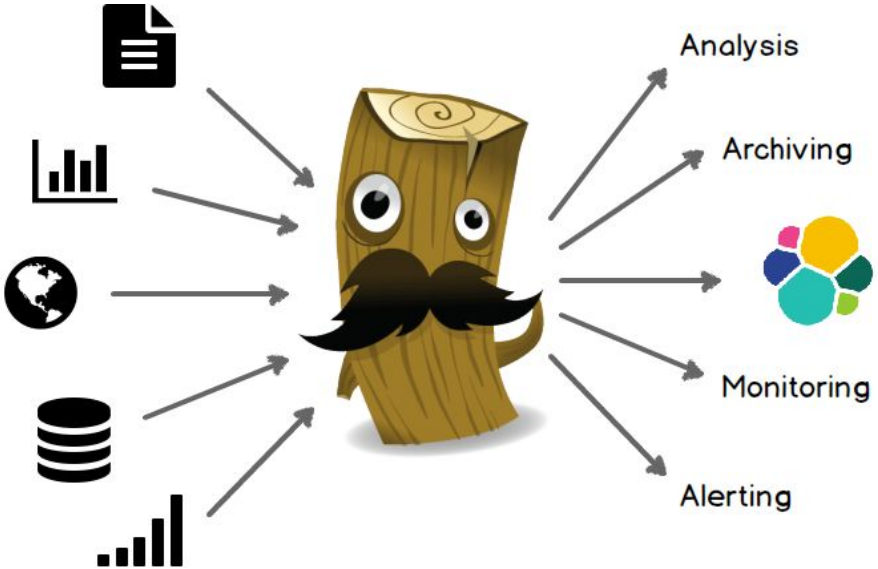


Логи хранятся n дней
Легко интегрироваться
Производительность
Репликация
Failure recovery

Логи хранятся n дней
Легко интегрироваться
Производительность
Репликация
Failure recovery

А как же надёжность поставки логов?

Поставка логов



Real-Time pipeline

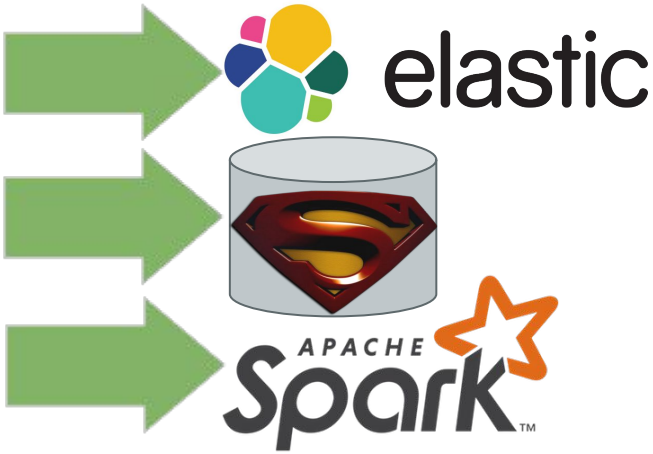
Хотелки

Потоки поставки | Промежуточная система | Потоки потребления

Производительность

Failure Recovery

Сохранение на диск



Хотелки

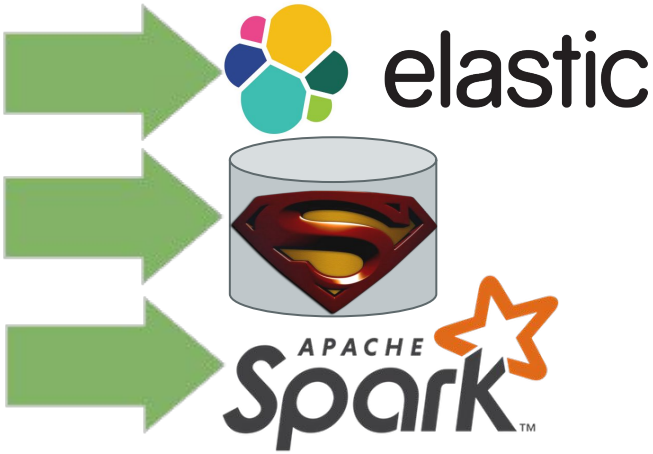
Потоки поставки | Промежуточная система | Потоки потребления

Производительность

Failure Recovery

Сохранение на диск

Логи инфраструктуры



RSyslog

Rocket-Fast System for Log Processing

Базируется на syslog

RSyslog

Rocket-Fast System for Log Processing

Базируется на syslog

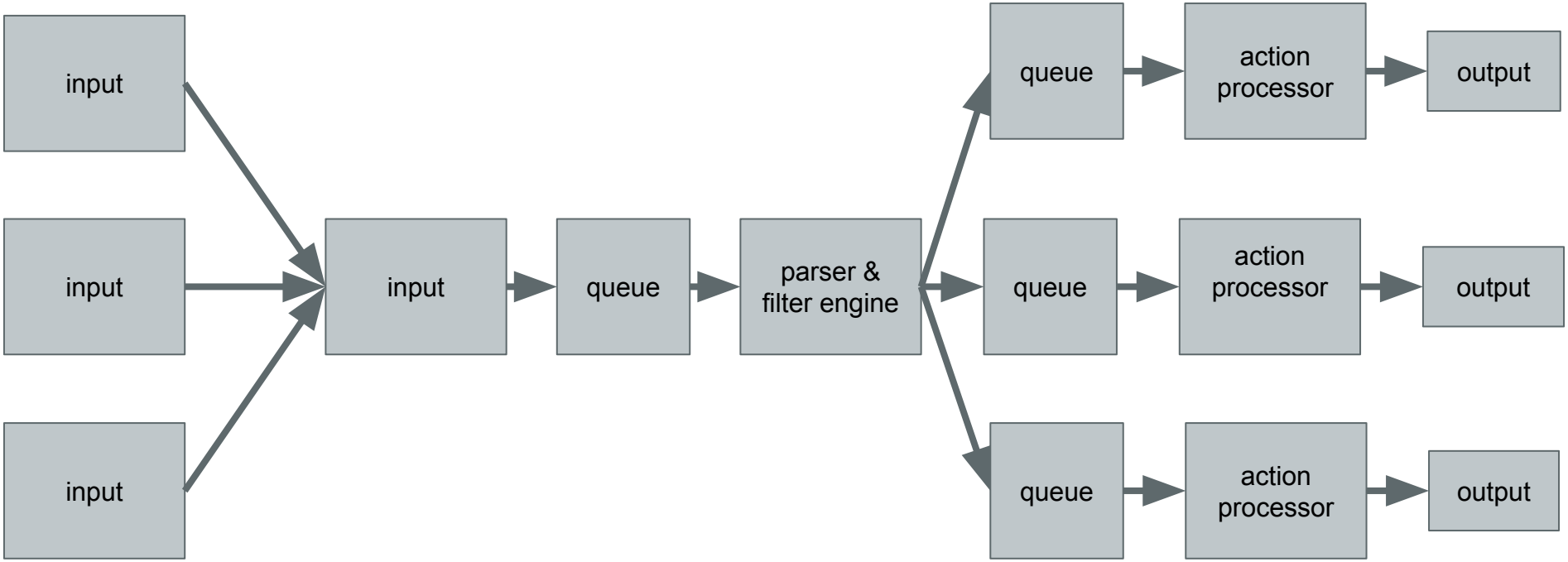
Модуль omkafka - Apache Kafka producer

RSyslog

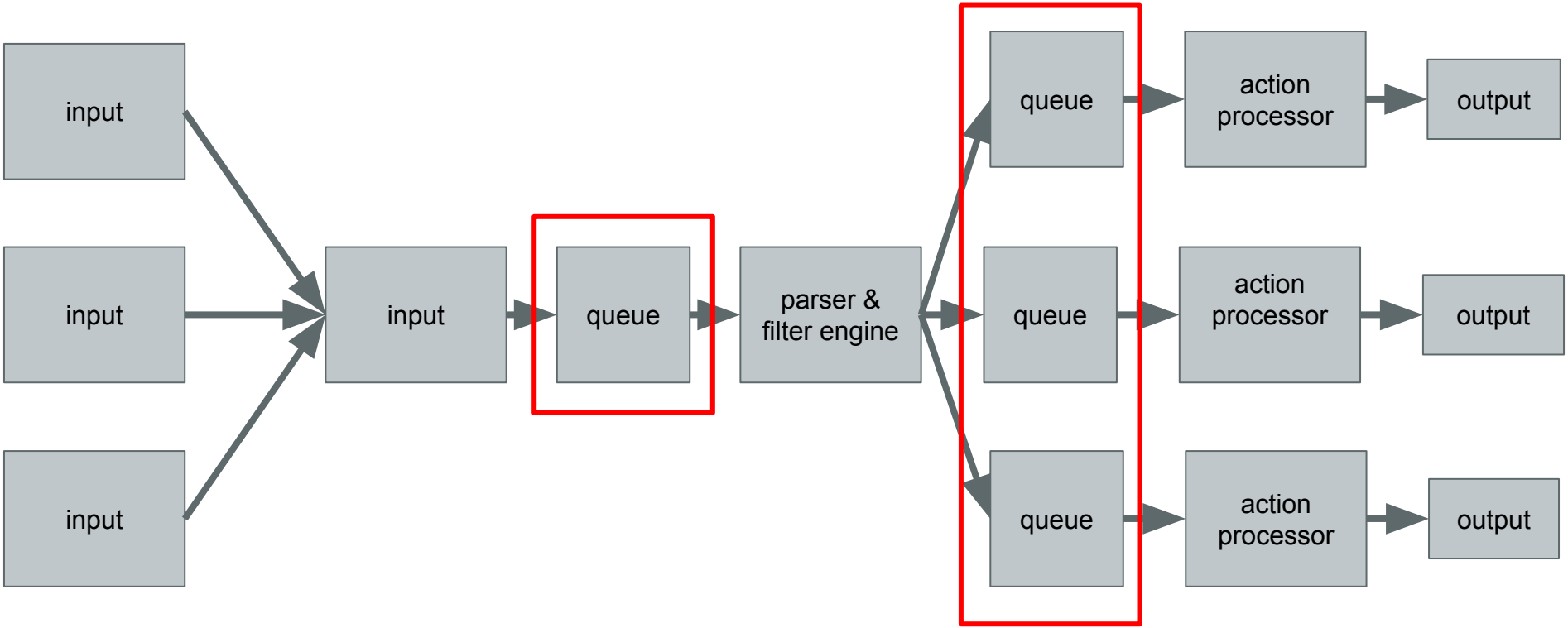
Rocket-Fast System for Log Processing

Базируется на syslog
Модуль omkafka - Apache Kafka producer
Очереди

RSyslog


input
input
input
input
queue
parser &
filter engine
queue
action
processor
output
queue
action
processor
output
queue
action
processor
output

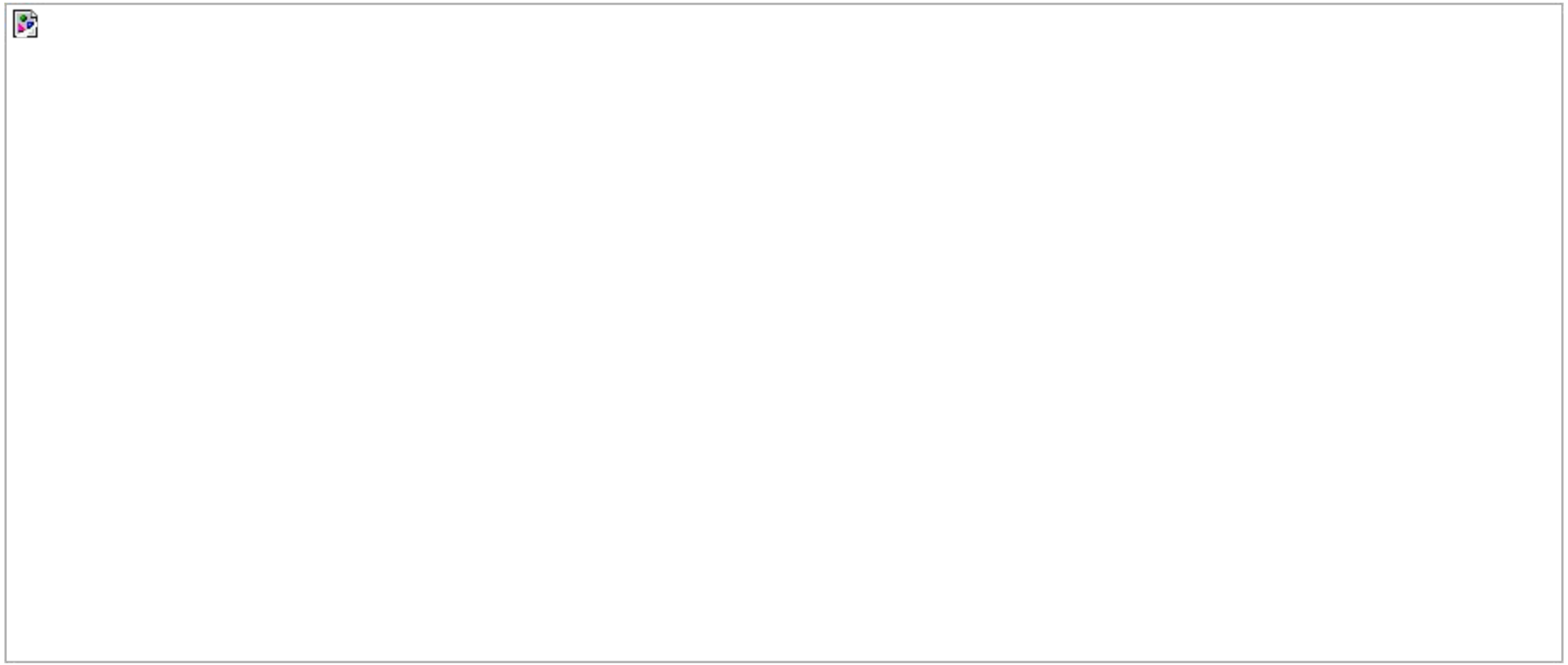
RSyslog


input
input
input
input
queue
parser &
filter engine
queue
queue
queue
action
processor
action
processor
action
processor
output
output
output

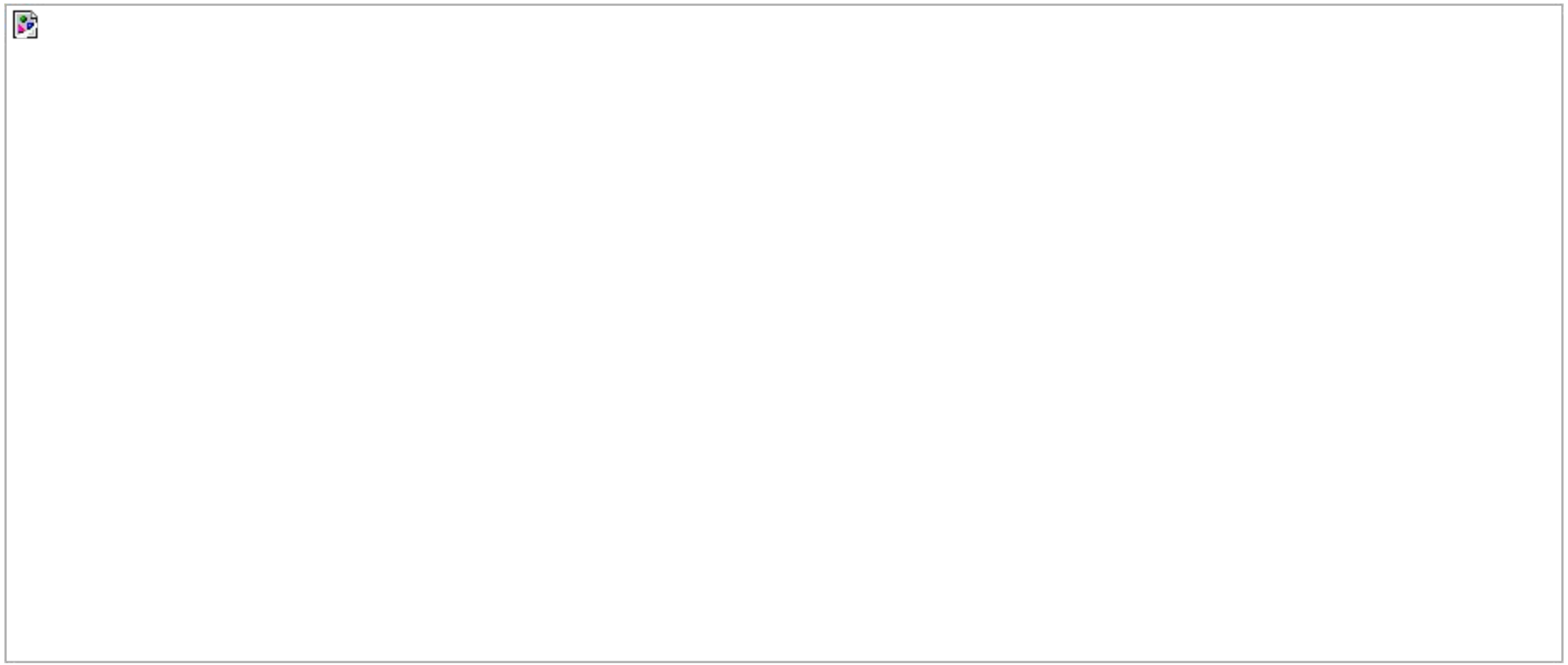
Что получается?

RSyslog

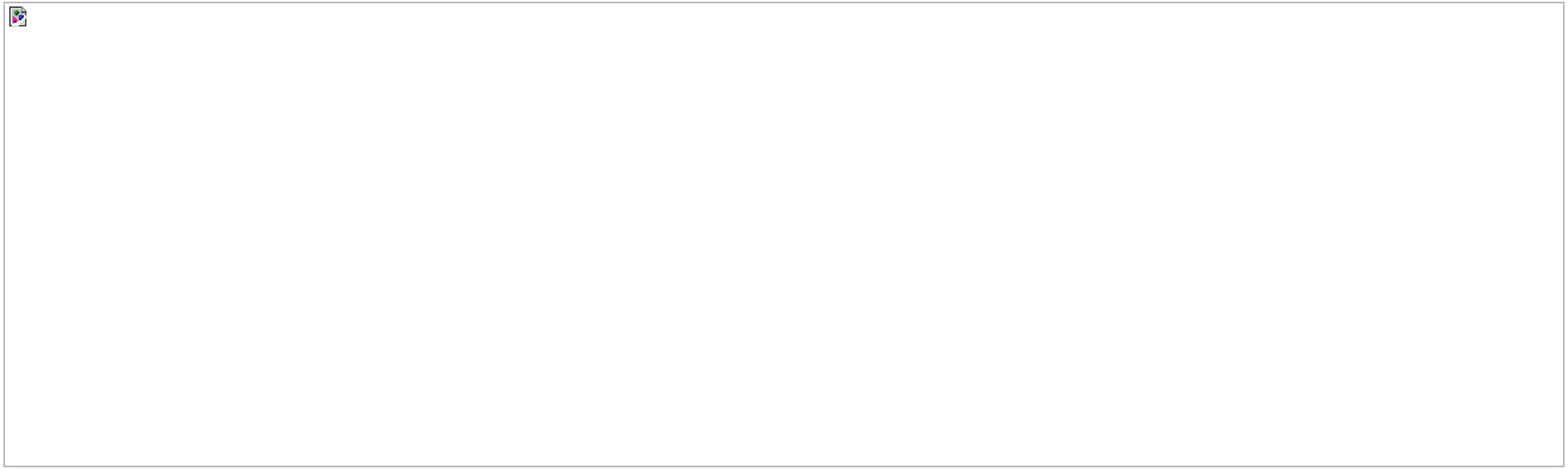
Новая архитектура



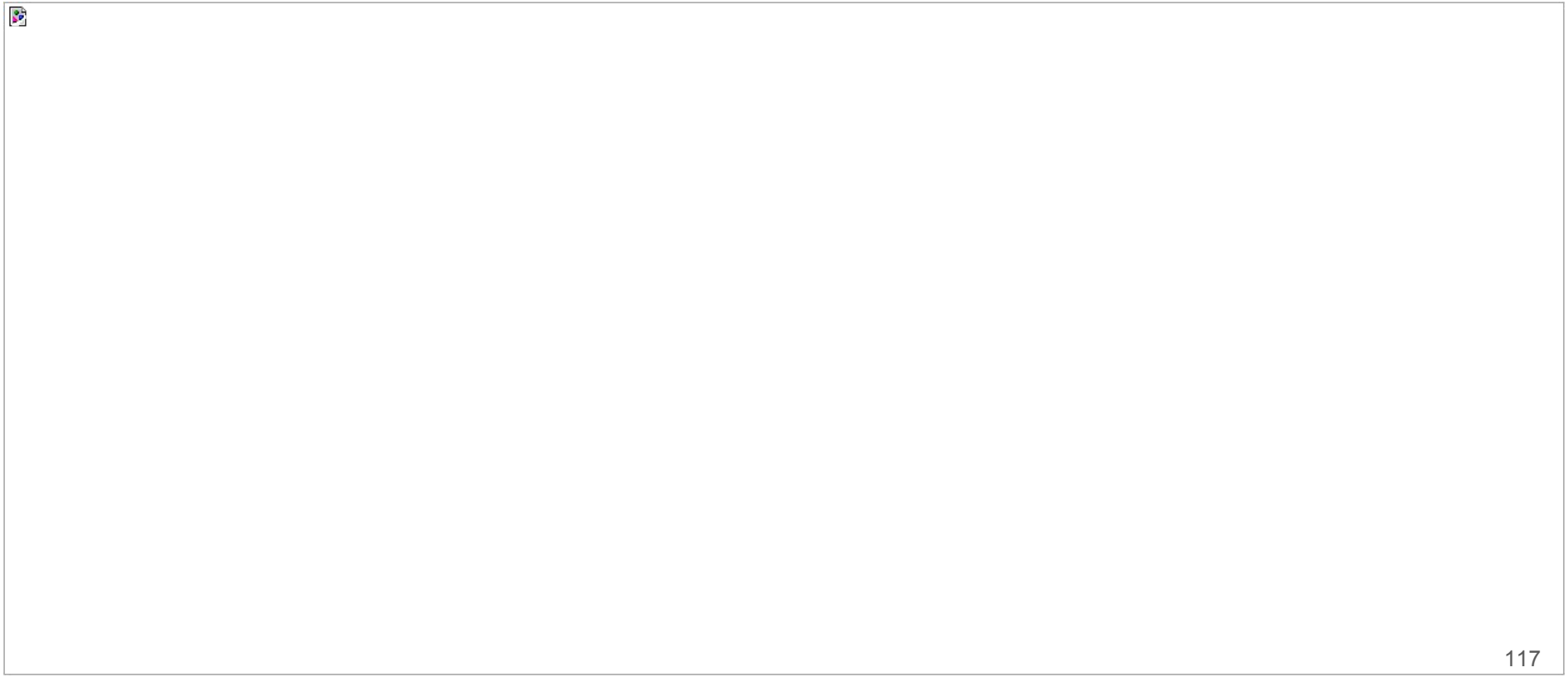
Новая архитектура



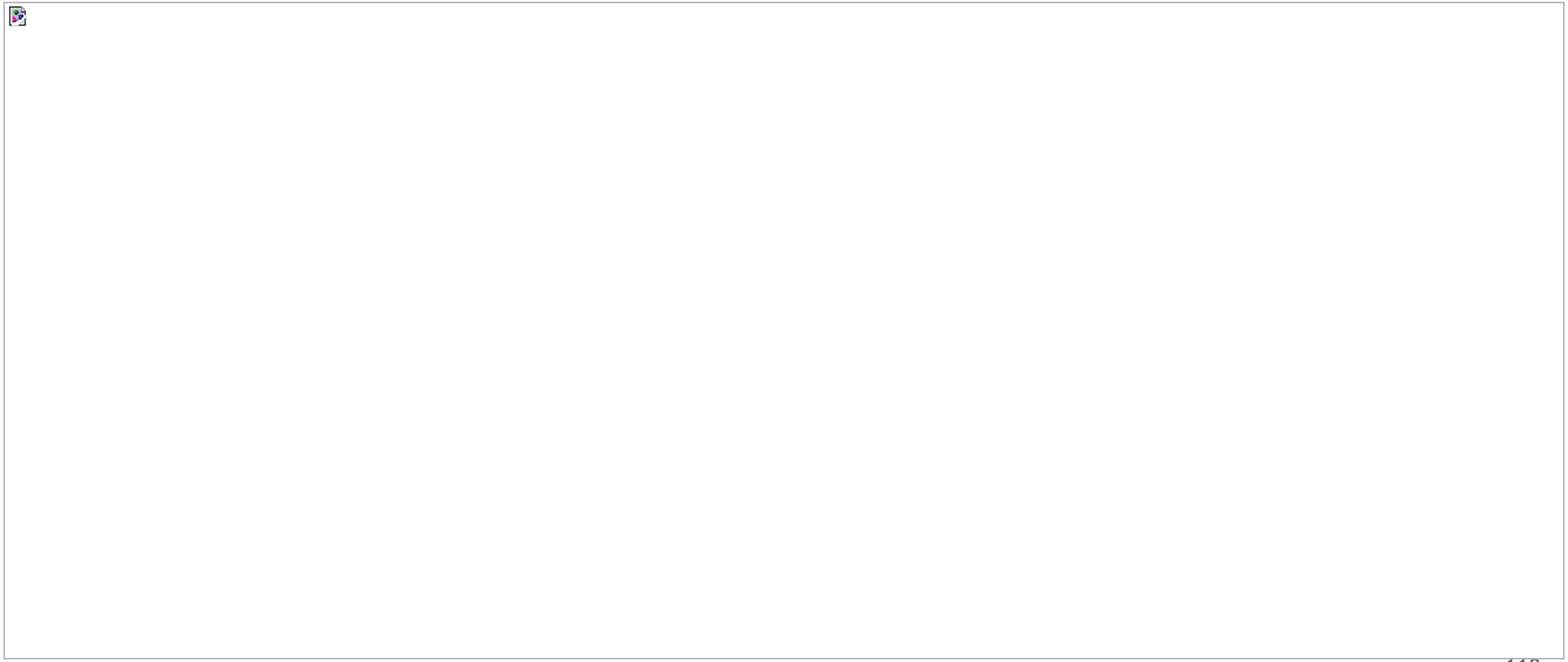
Новая архитектура



Новая архитектура



Архитектура 1.0



Но мы же банк..

Мы используем Red Hat Linux на серверах

Но мы же банк..

Мы используем Red Hat Linux на серверах

В стандартных репозиториях нету версии 8.7.+

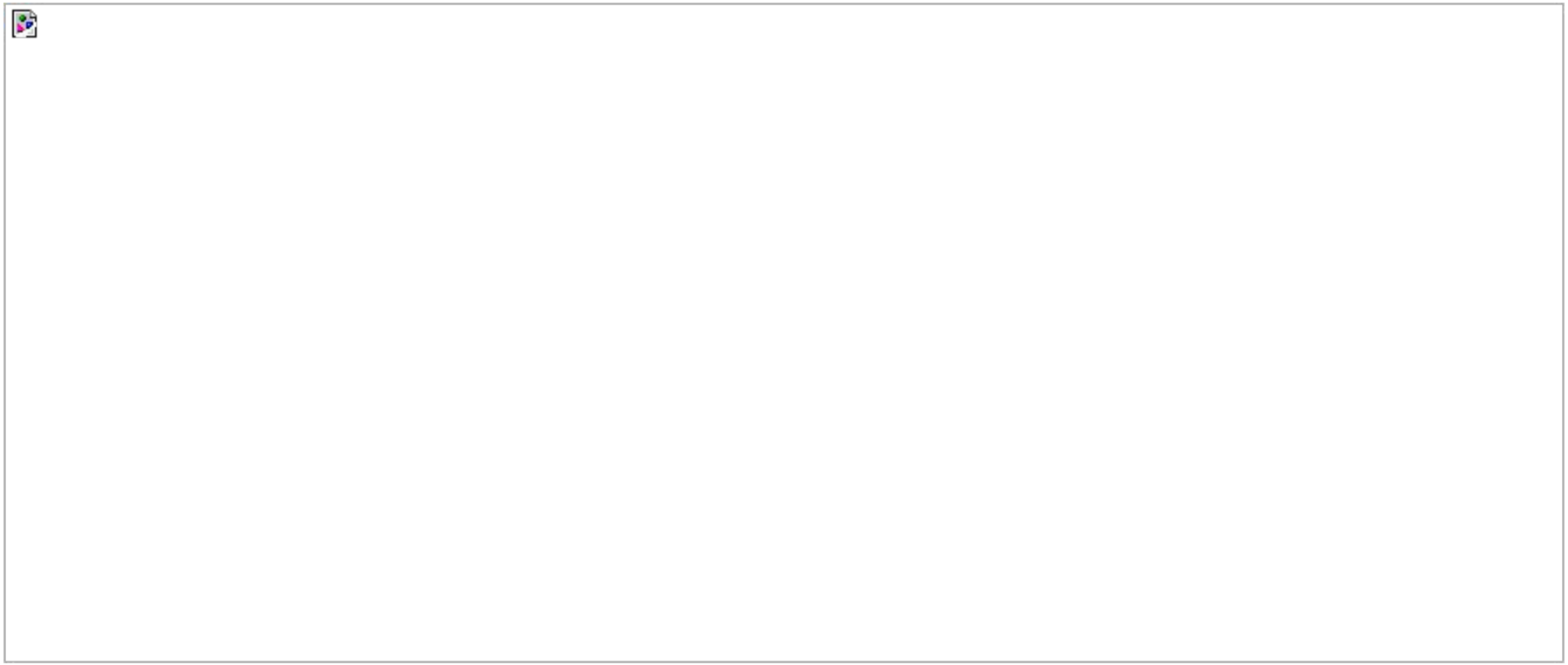
Но мы же банк..

Мы используем Red Hat Linux на серверах

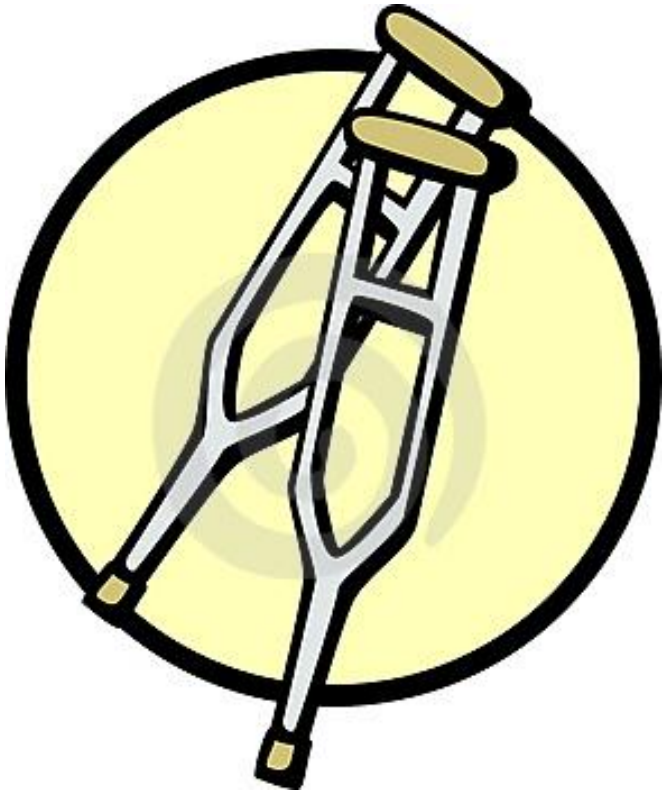
В стандартных репозиториях нету версии 8.7.+

Только в ней появился модуль omkafka

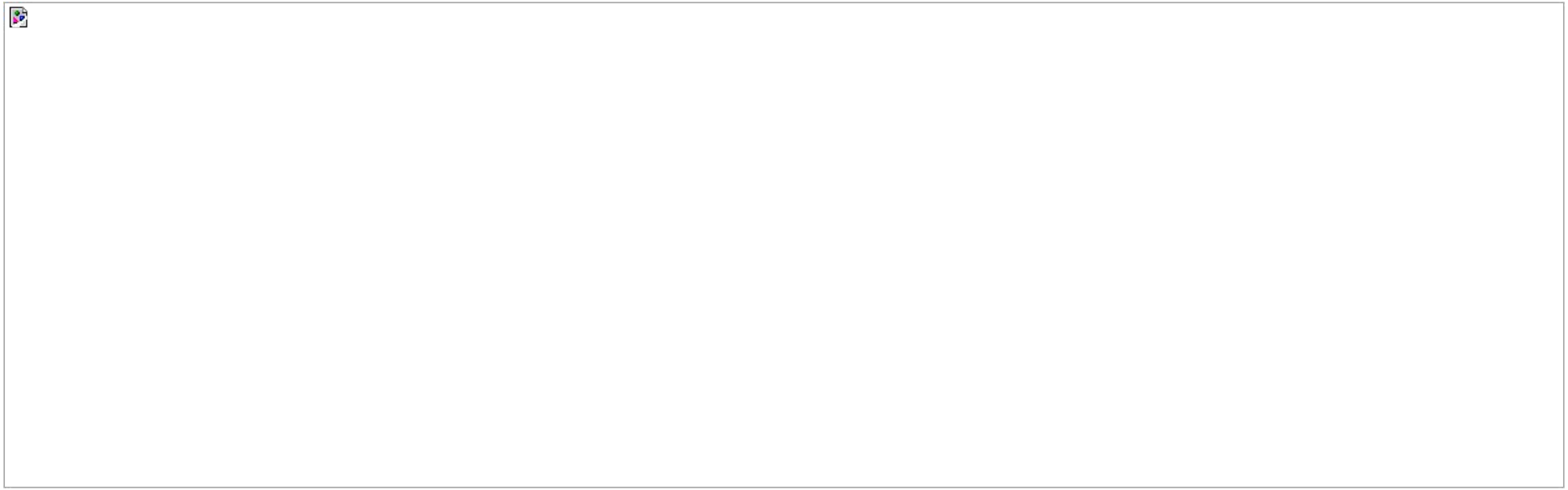
Архитектура 1.0



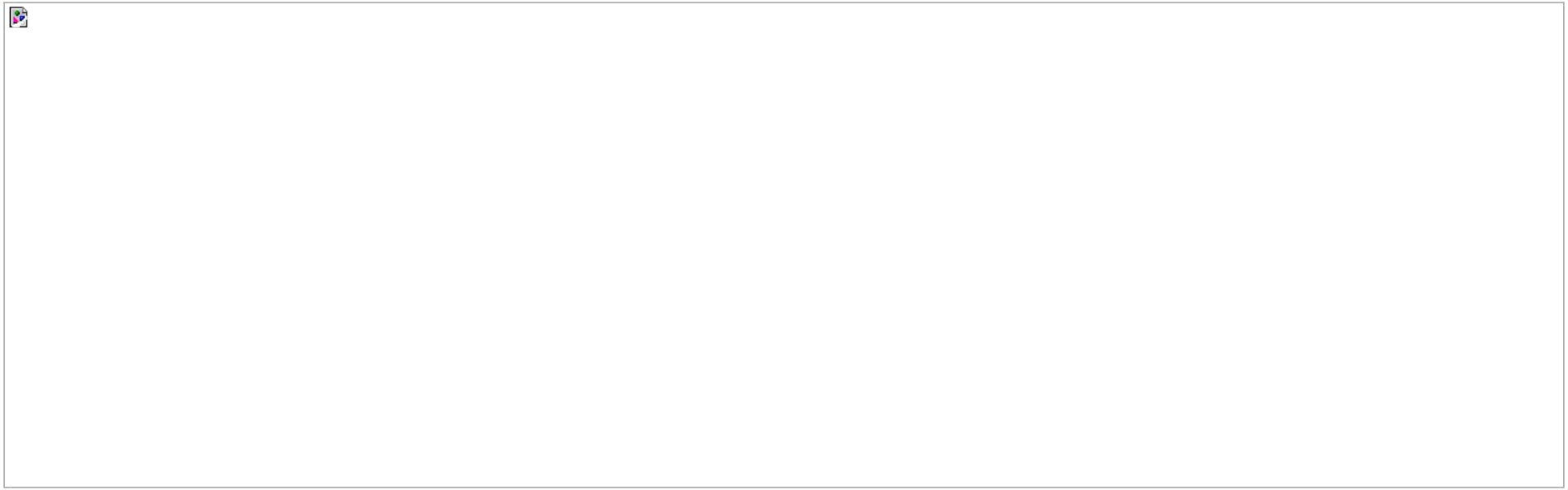
Известный архитектурный приём



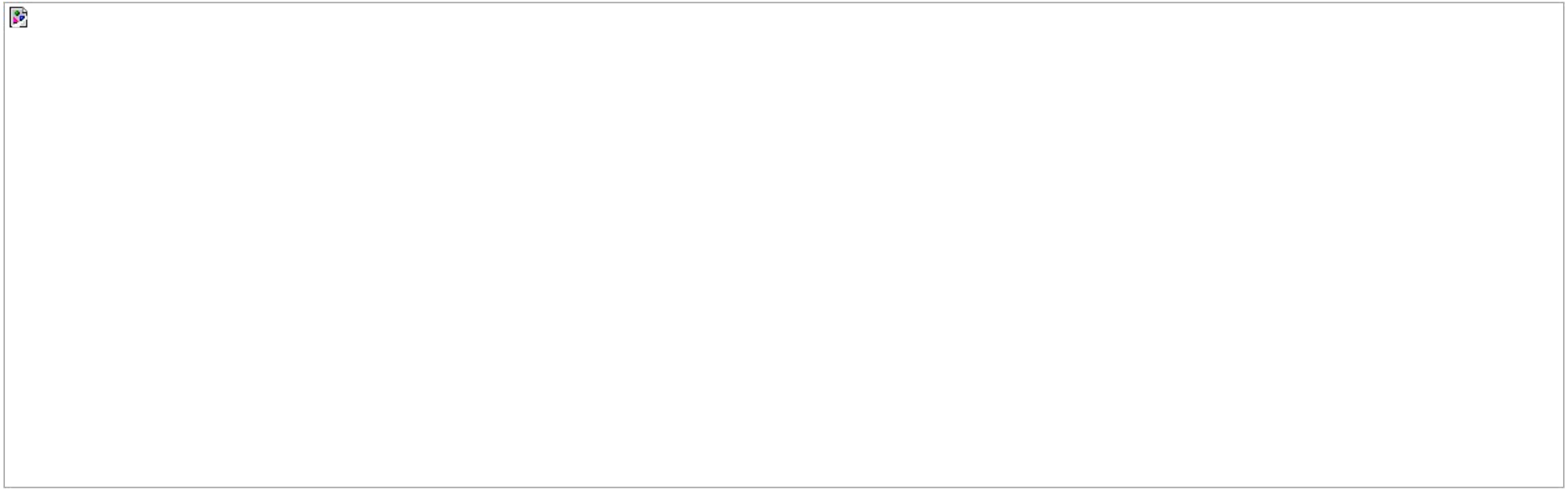
Архитектура 2.0



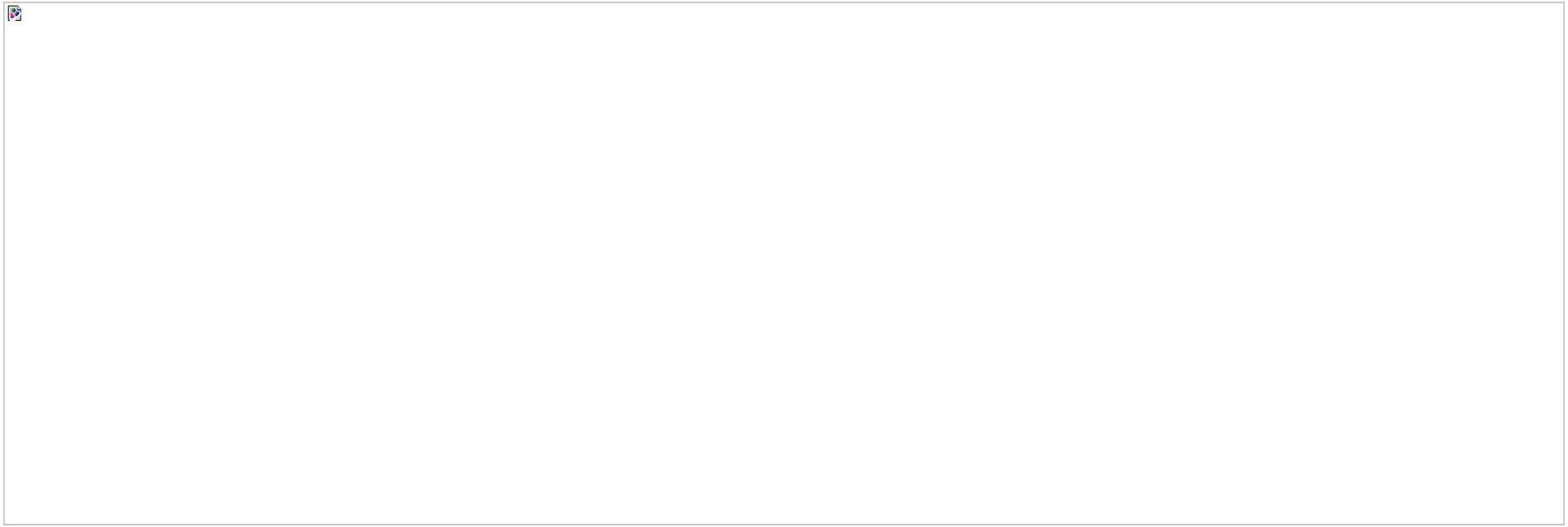
Архитектура 2.0



Архитектура 2.0



Архитектура 2.0

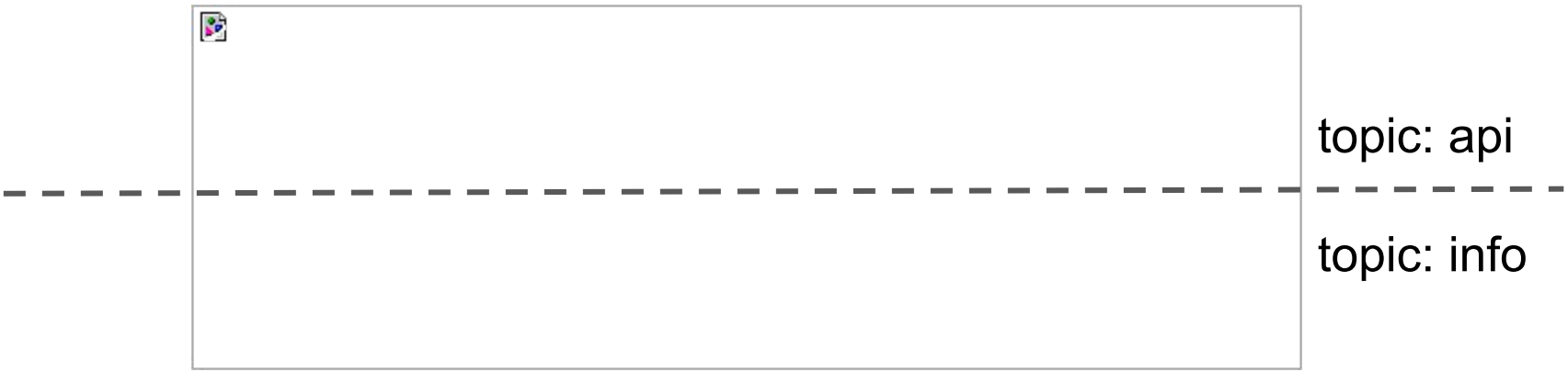


Демо

Логи DashboardAPI отправляются в топик api

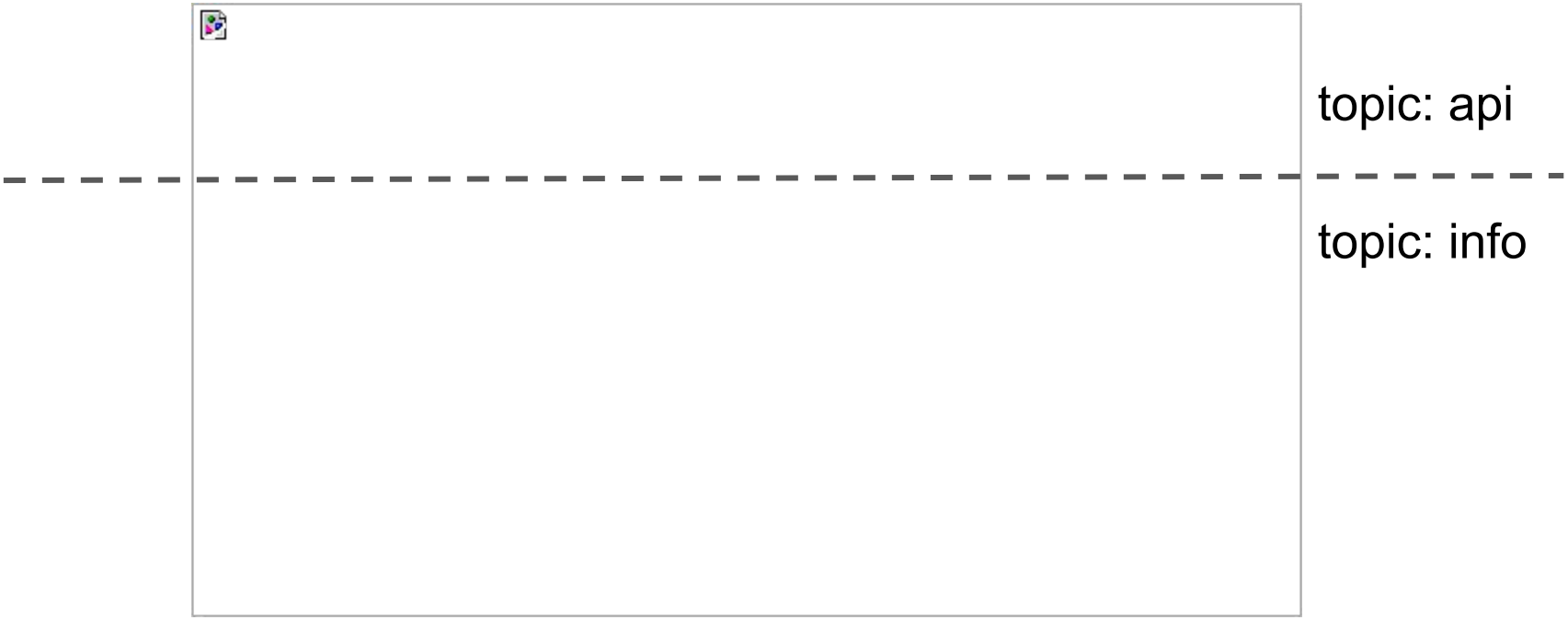
Partition по ключу DashboardAPI

Демо

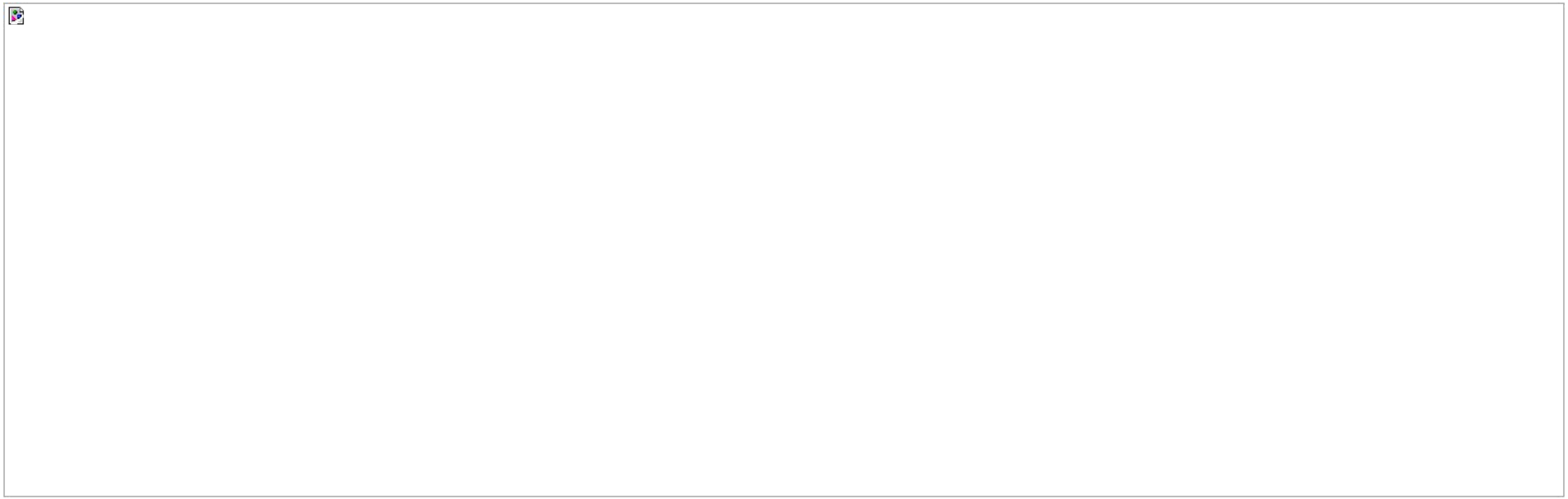


Логи CustomerInfo и TransactionsList отправляются в топик info
Partition по ключу названия сервиса

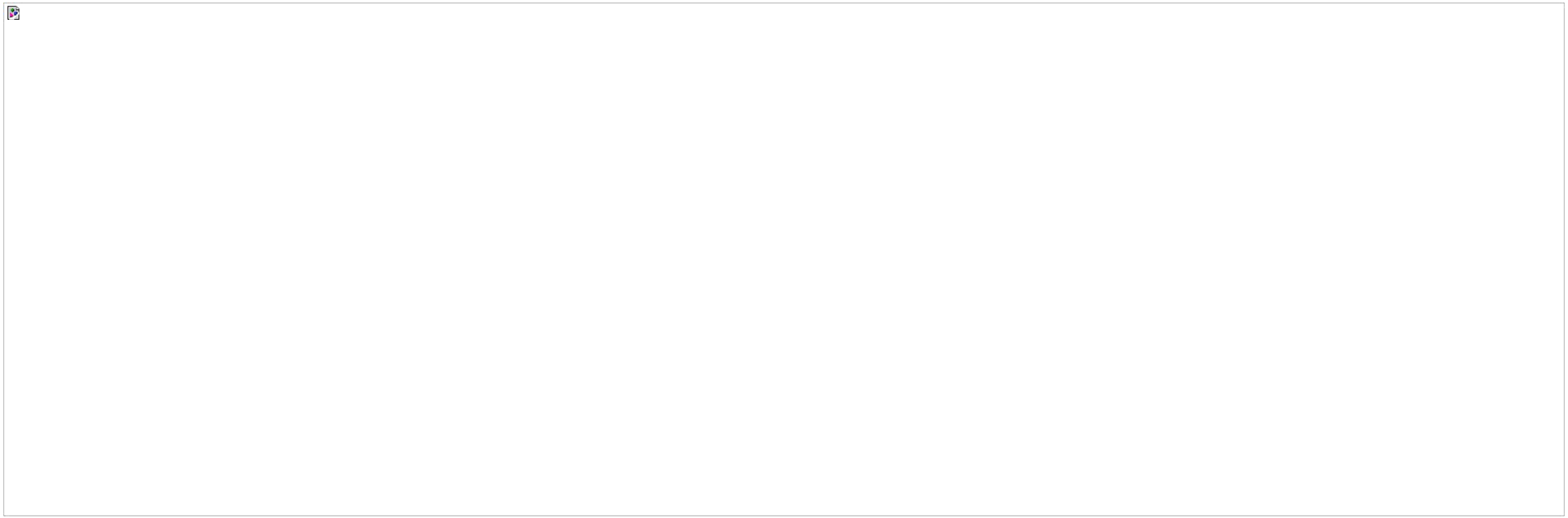
Демо



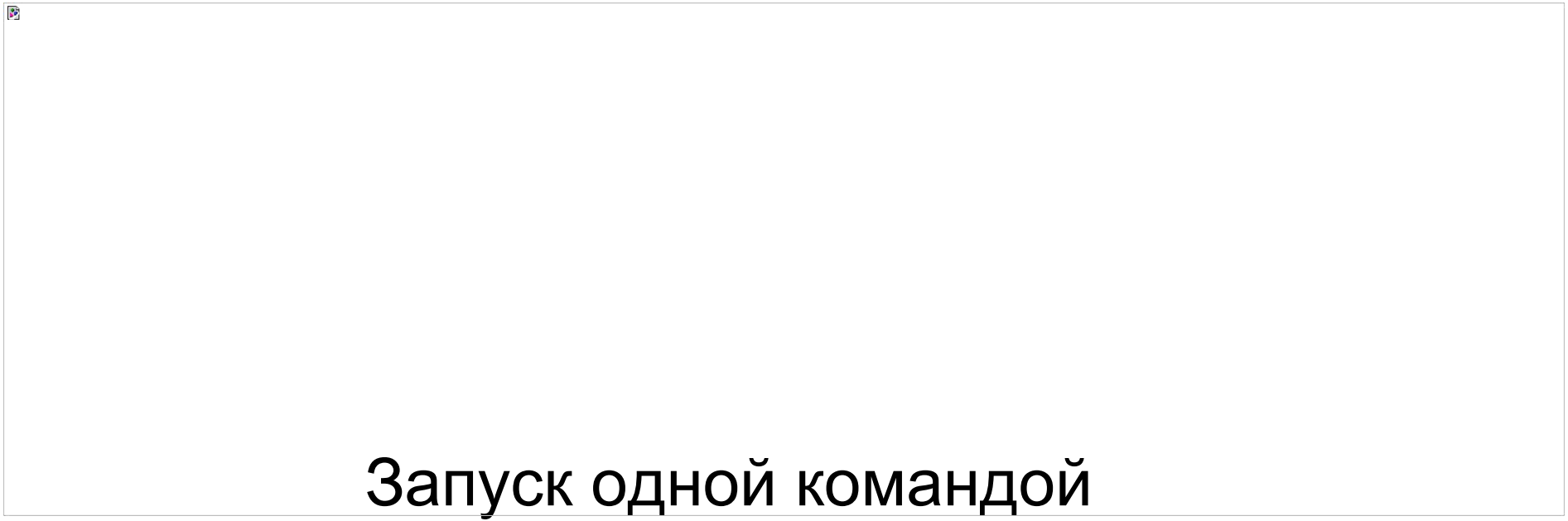
Демо



Демо



Демо



Запуск одной командой

./compose-logs.sh up -d

Новая задача

Мониторинг количества ошибок в секунду от приложений CustomerInfo и TransactionsList

Как делают stream processing

1. Kafka producer и consumer

Как делают stream processing

1. Kafka producer и consumer
2. Прикрутить полноценный stream processing framework

Stream processing инструменты

Apache Spark
Apache Storm
Apache Flink
Apache Samza
Google’s DataFlow
AWS Lambda
...

Что нужно большим фрэймворкам

- Распределять джобы в пуле машин

Что нужно большим фрэймворкам

- Распределять джобы в пуле машин
- Управлять ресурсами в кластере

Что нужно большим фрэймворкам

- Распределять джобы в пуле машин
- Управлять ресурсами в кластере
- Паковать код и отправлять на исполняющие машины

Stream processing инструменты

Apache Spark

Apache Storm

Apache Flink

...

Kafka Streams

Kafka Streams

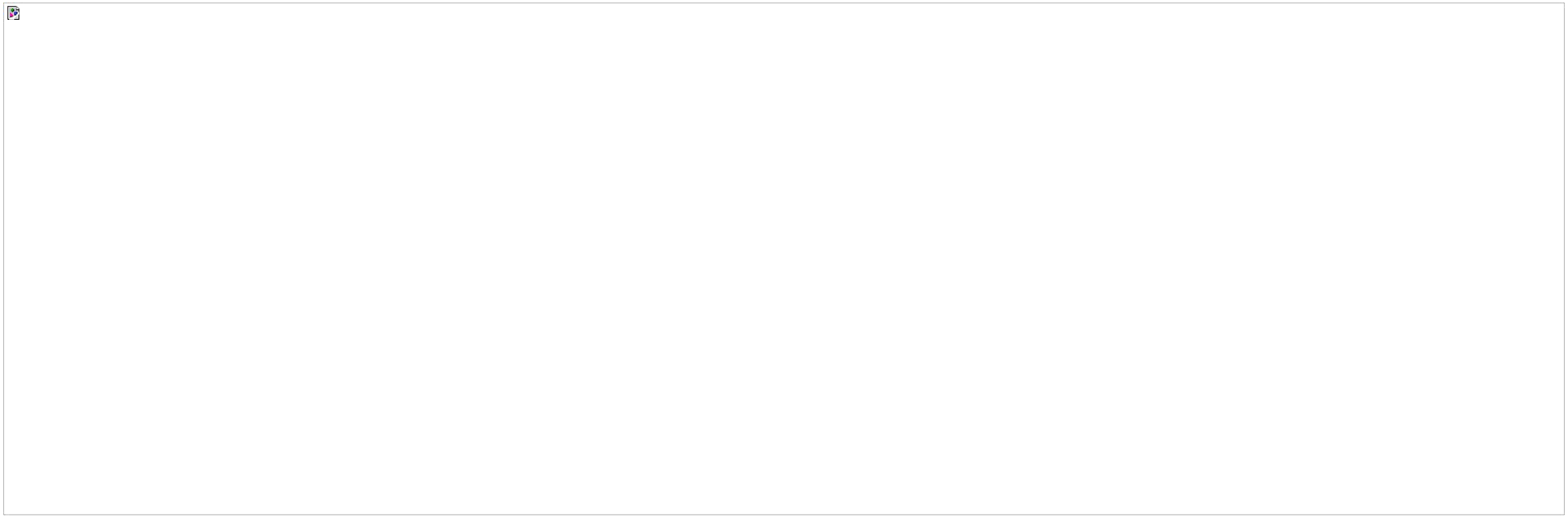
Event-at-a-time processing

Уже в Kafka 0.10 release

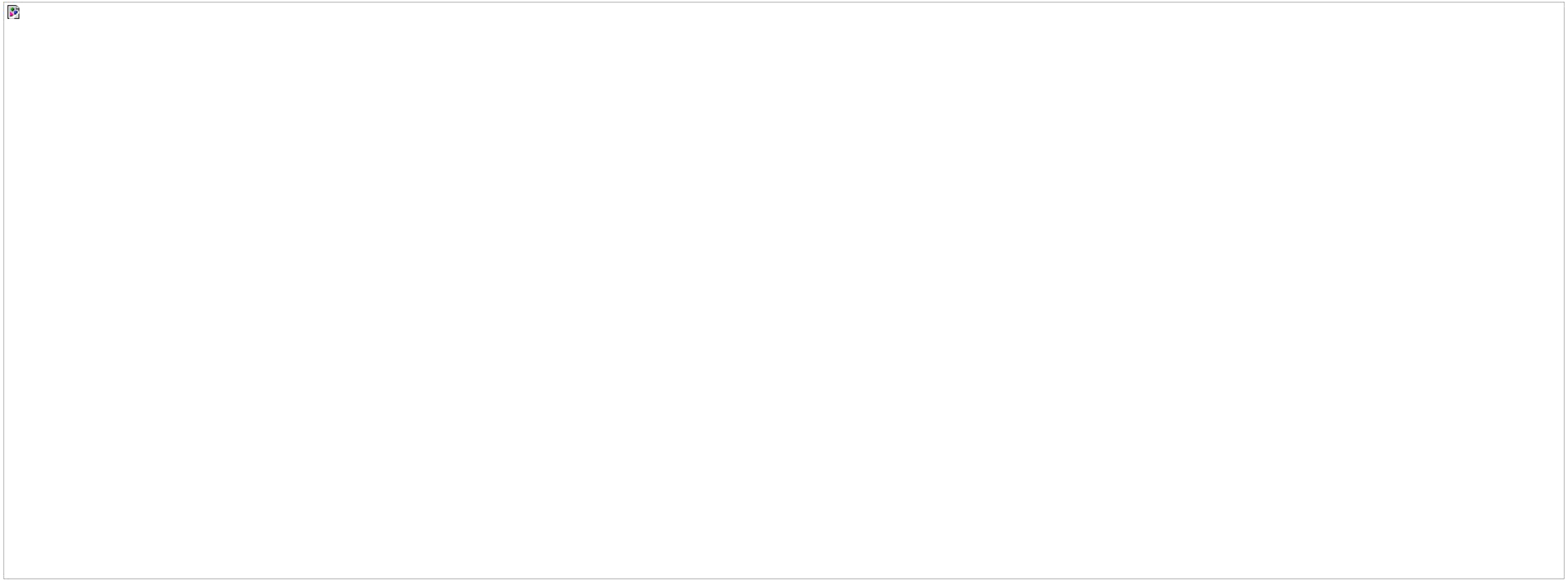
Для приложения - просто библиотека

Нет дополнительной инфраструктуры

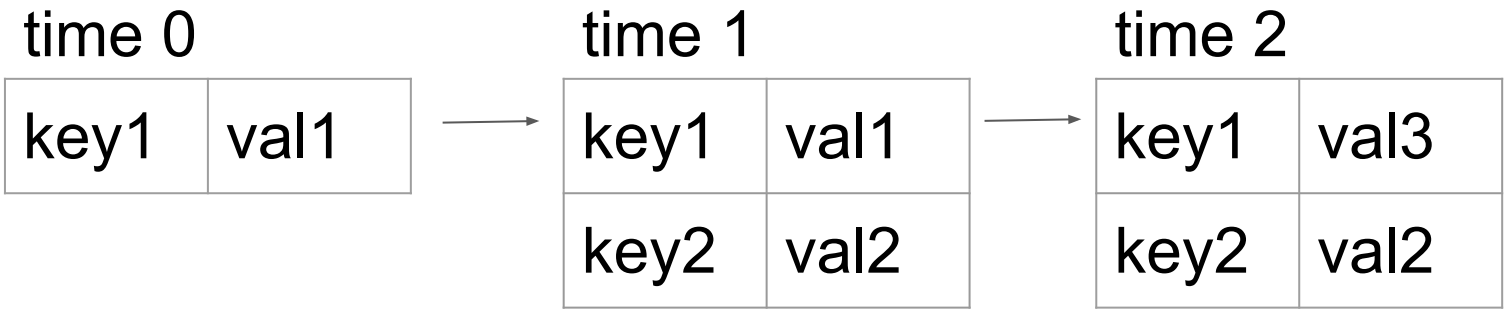
Демо



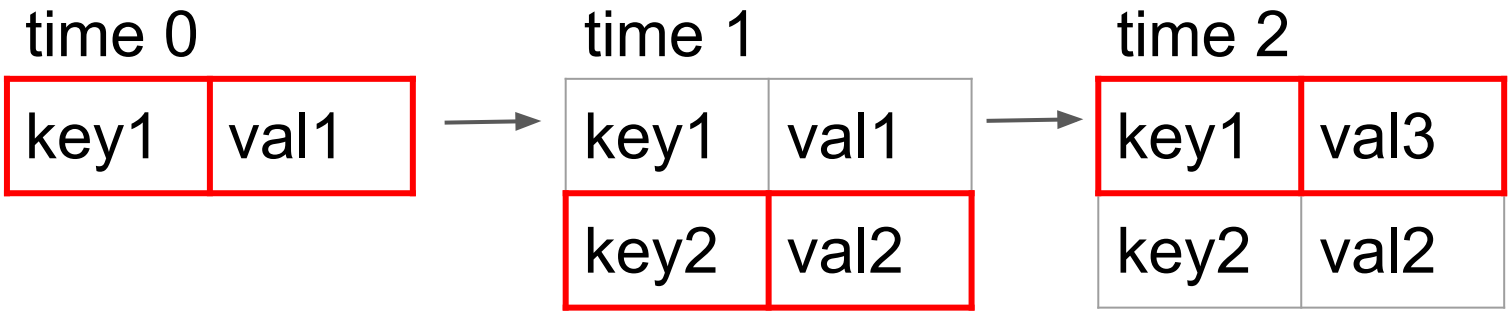
Приложение с Kafka Streams



Таблицы и стримы



Таблицы и стримы



put (key1, val1)
put (key2, val2)
put (key1, val3)

Таблицы и стримы

put (key1, val1)		key1 => val1
put (key2, val2)	==	key2 => val2
put (key1, val3)		key1 => val3

Таблицы и стримы

put (key1, val1)		key1 => val1
put (key2, val2)	==	key2 => val2
put (key1, val3)		key1 => val3

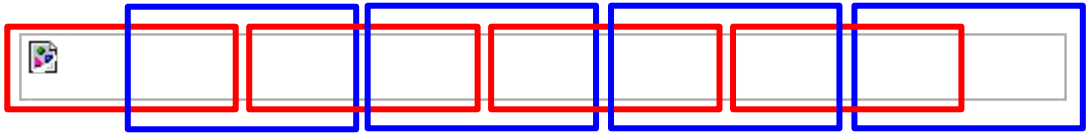
Таблица - это кэш актуальных значений для ключей в стриме

Стрим - changelog таблицы

Окно

Поток сообщений

Окно



Считаем количество сообщений
в каждом окне

Итого

Для большой децентрализованной системы нужно логирование и оно должно быть централизованным.

Итого

Если у вас много разных потребителей, которым нужен доступ к системе логирования, то следует задуматься о промежуточной системе, где логи будут храниться некоторое время.

Итого

У нас получилась система позволяющая:

- легко добавлять новых потребителей
- хранящая логи в течении некоторого временного интервала
- позволяющая делать real-time аналитику без дополнительной инфраструктуры

Итого

Демо системы логирования и Kafka Streams
github.com/sergeevii123/joker.logs

Для связи:
sergeevii123@gmail.com

Вопросы