slido.com
#70247

arquivei
.com.br

Plataforma de Dados

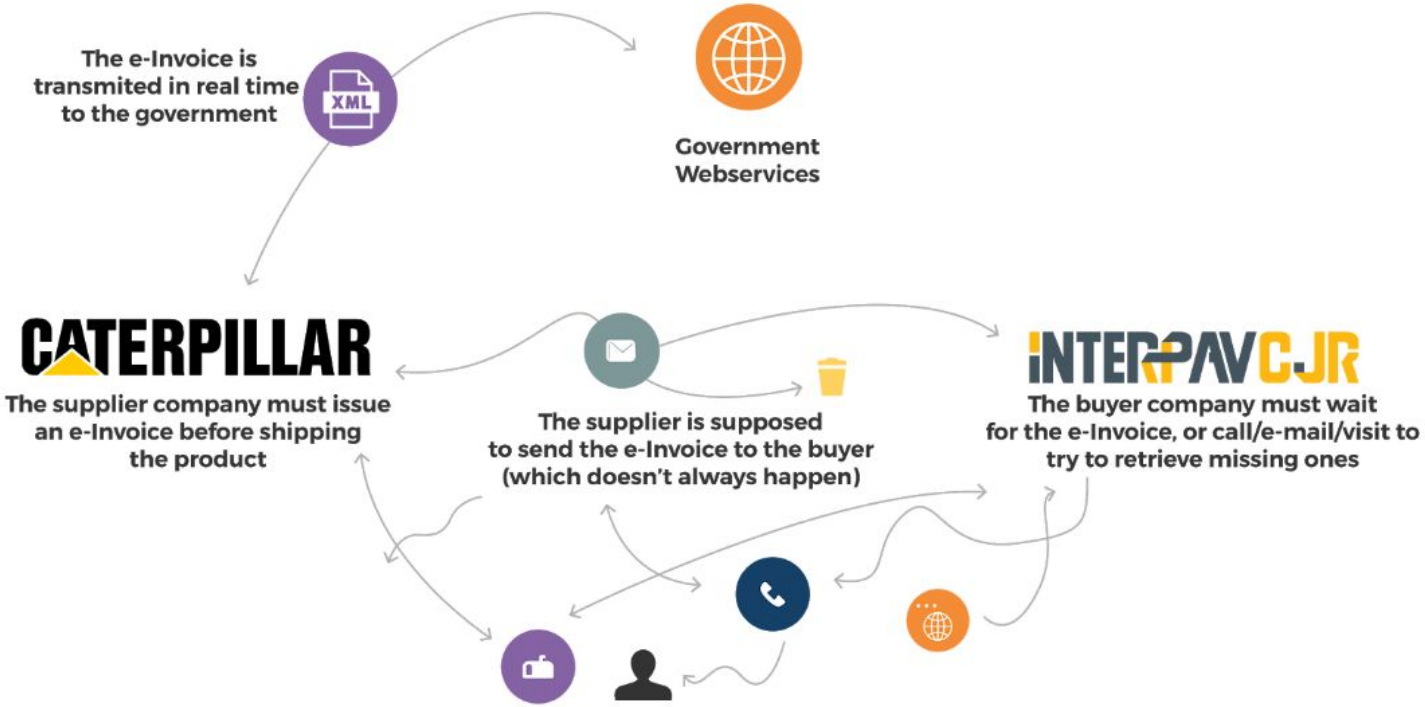
arquivei .com.br

Roadmap

- Apresentação
- Filas
- Streaming
- Plataforma de Dados

Eu

Arquivei


The e-Invoice is transmited in real time to the government
XML
Government Webservices
CATERPILLAR
The supplier company must issue an e-Invoice before shipping the product
The supplier is supposed to send the e-Invoice to the buyer (which doesn't always happen)
INTERPAVC.JR
The buyer company must wait for the e-Invoice, or call/e-mail/visit to try to retrieve missing ones

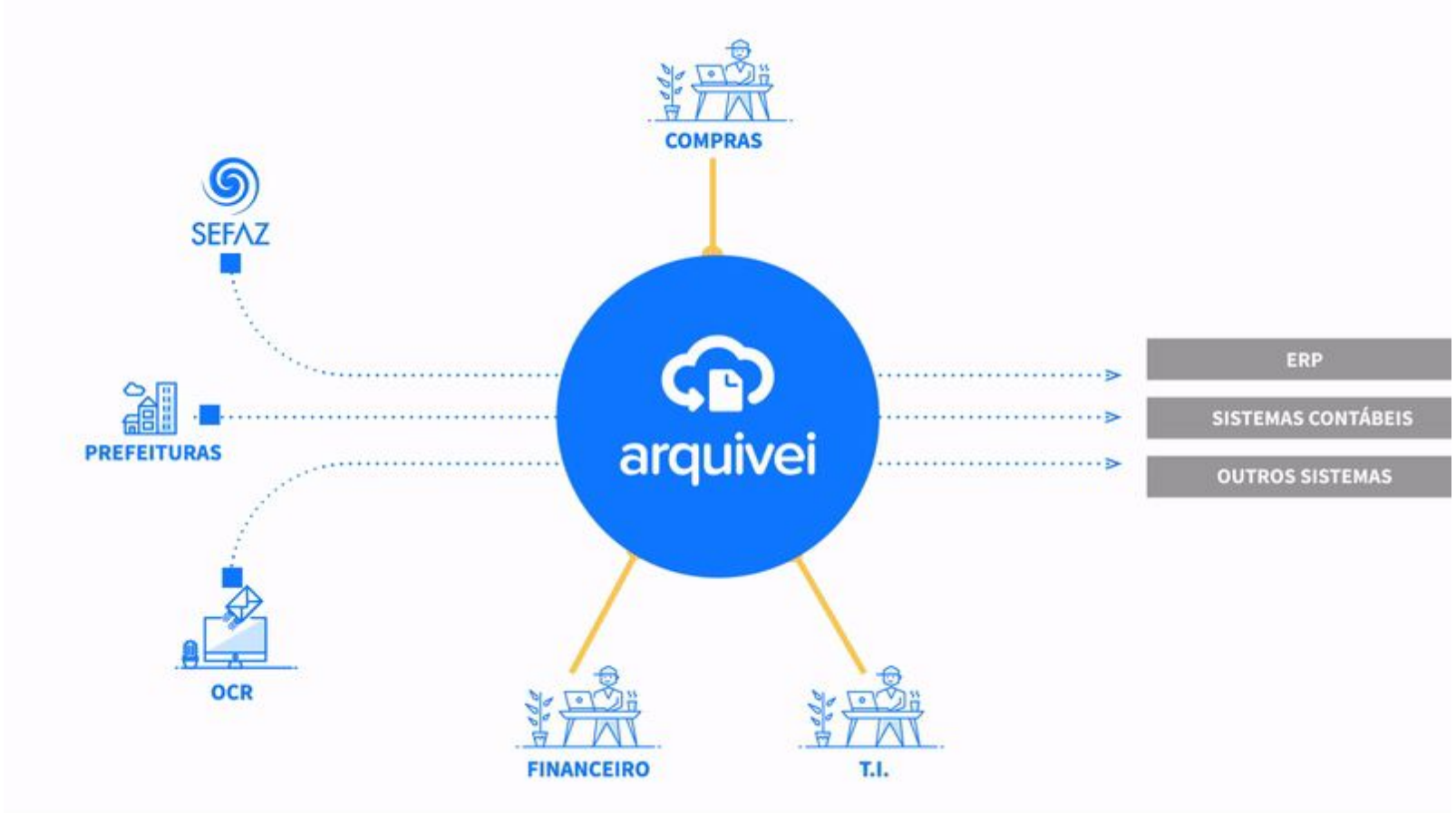
Arquivei

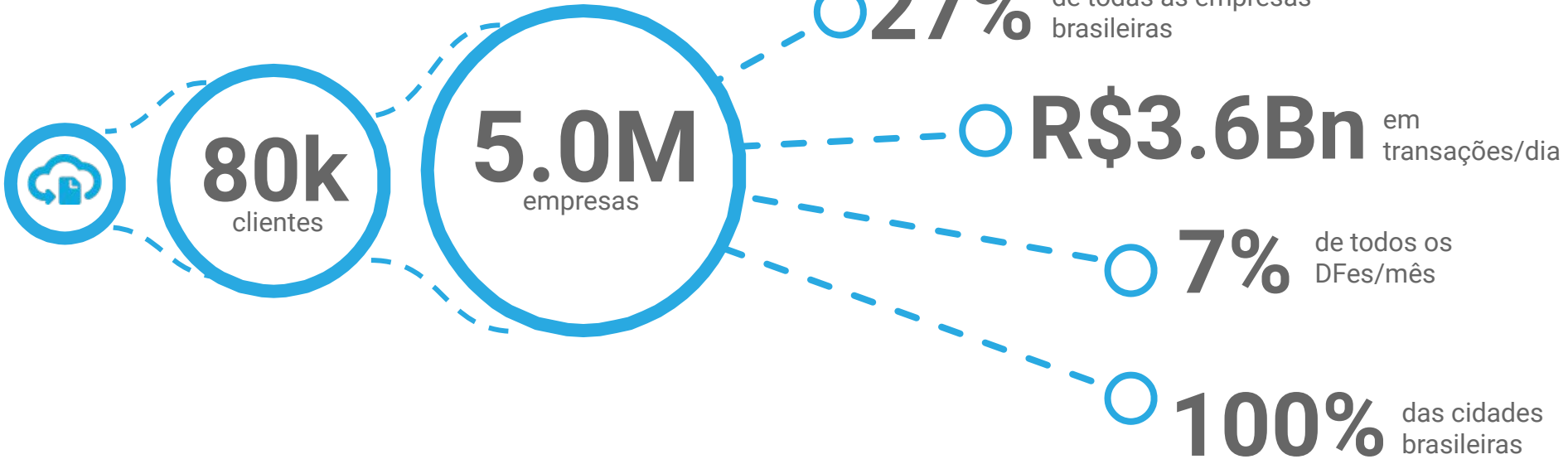

The e-Invoice is transmited in real time to the government
XML
Government Webservices
The buyer authorizes Arquivei to access e-invoices issued by all of their suppliers
CATERPILLAR
The supplier company simply issues the e-Invoice to the goverment
arquivei
XML
Arquivei automatically fetches, stores in the cloud and analyses the e-invoices in real time
INTERPAVE JR
The buyer company must simply have an Arquivei account

Arquivei


Estratégia
Inteligência
Produtividade
Controle e Compliance

Democratizando a Informação





arquivei .com.br

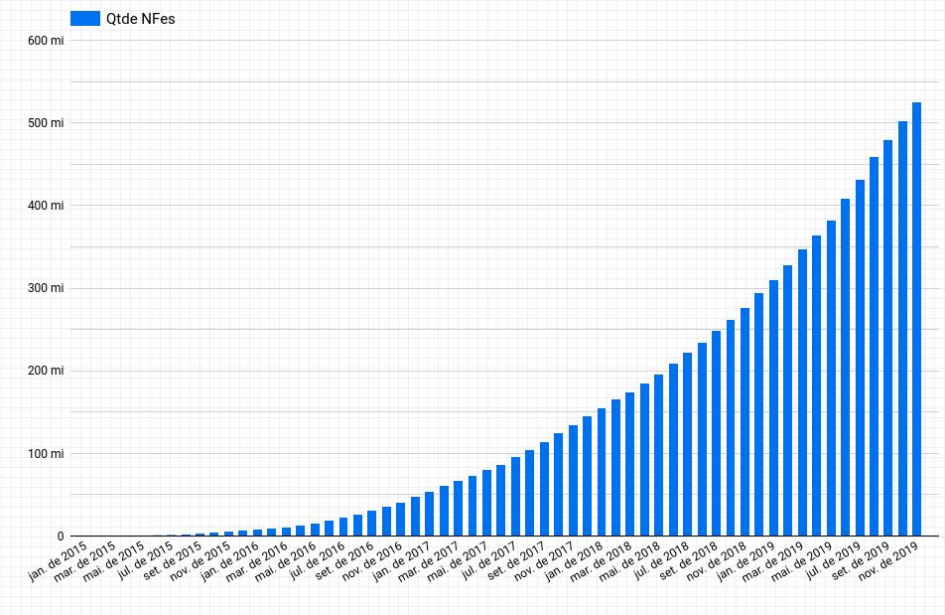
Temos vagas: arquivei.com.br/vagas

Desafio

Problema inicial

Problema inicial

- Extrair 600M de XMLs de um PSQL
  - Contagens
  - Histórico
  - Em tempo real...

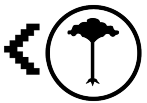


Problema inicial

- Extrair 50M de XMLs de um PSQL
  - Contagens
  - Histórico
  - Em tempo real...

Problema inicial

Filas



Primeiro modelo

- Pub/Sub (SQS)
  - Barato
  - Gerenciado
  - Simples
  - Escalável

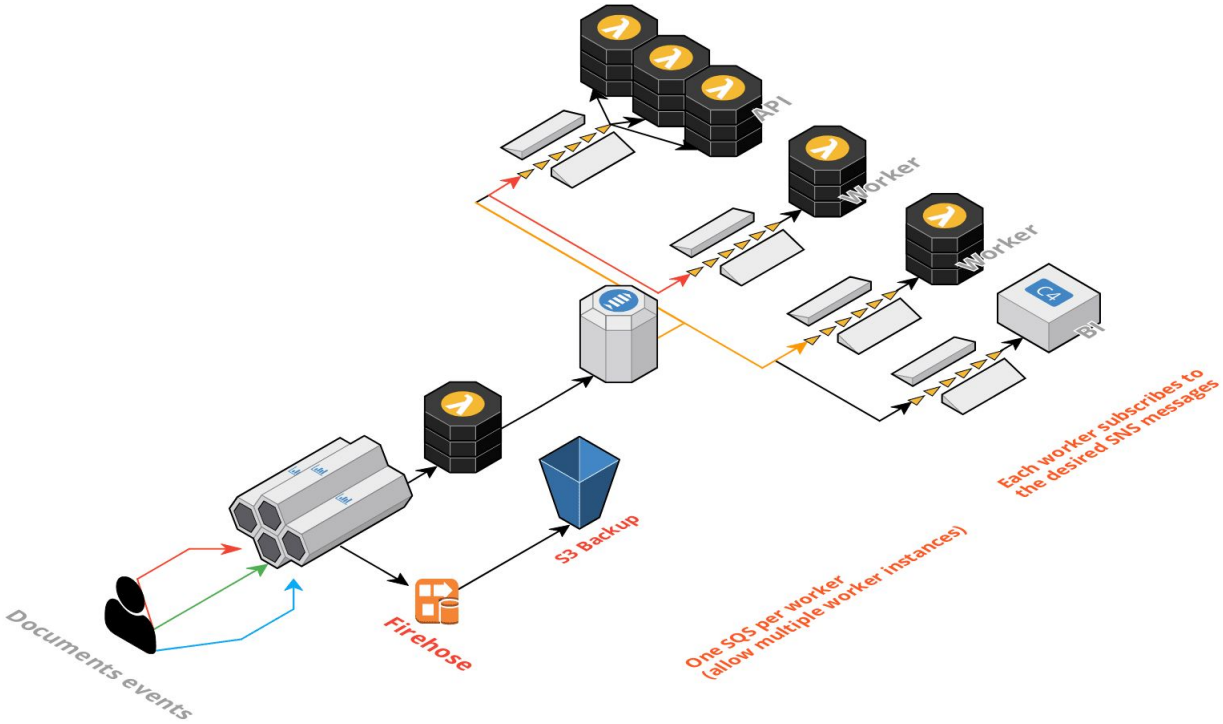
Primeiro modelo

- Pub/Sub
  - Barato
  - Gerenciado
  - Simples
  - Escalável
  - Problemas:
    - Ordem/Concorrência
    - Reprocessamentos

Segundo modelo

- Requisitos do pipeline de dados
  - Pouca infra
  - Deploys fáceis para produtores e consumidores
    - Desacoplado
    - Independente

Segundo modelo



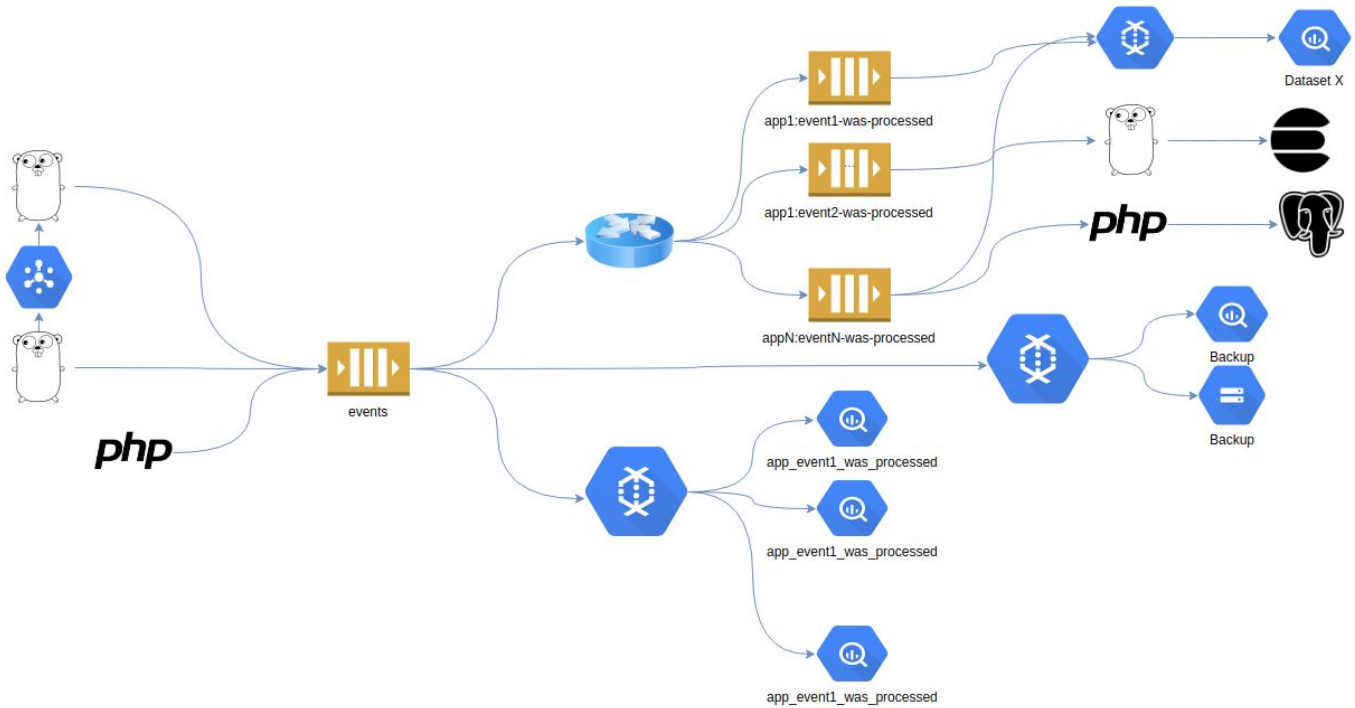
Terceiro modelo

- Apache Kafka
  - Baixíssima latência
  - Escalável
  - “Barato”

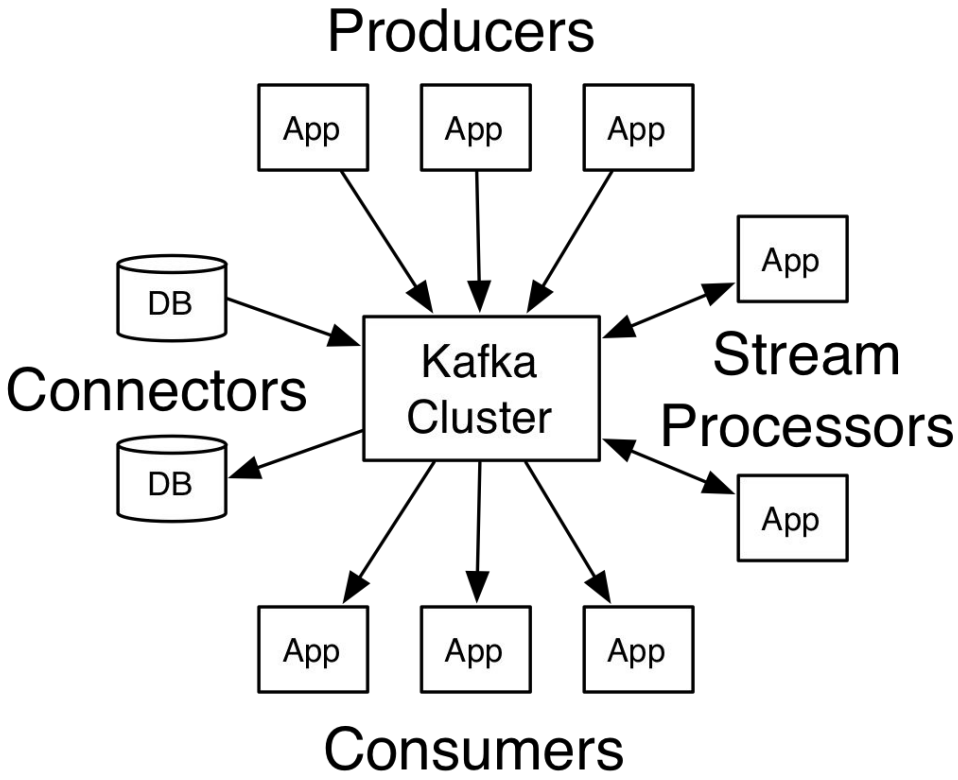
Terceiro modelo

- Apache Kafka
  - Baixíssima latência
  - Escalável
  - “Barato”
  - Problemas
    - Overhead de Infra
    - Curva de aprendizado
      - Escalabilidade
      - Modelos de produção
      - Modelos de consumo
    - Libs limitadas para PHP

Terceiro modelo



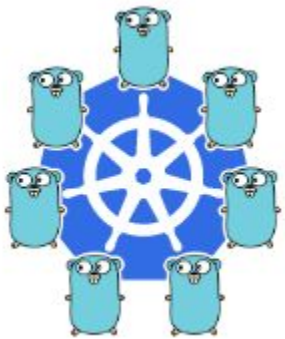
v3.0: Kafka Streams



Streaming

v1.0: Golang

- Linguagem simples
- Leve
- Escalável
- Deploy K8s



v2.0: Google Dataflow

- Produto GCP
- Cluster gerenciado
- Autoscaling

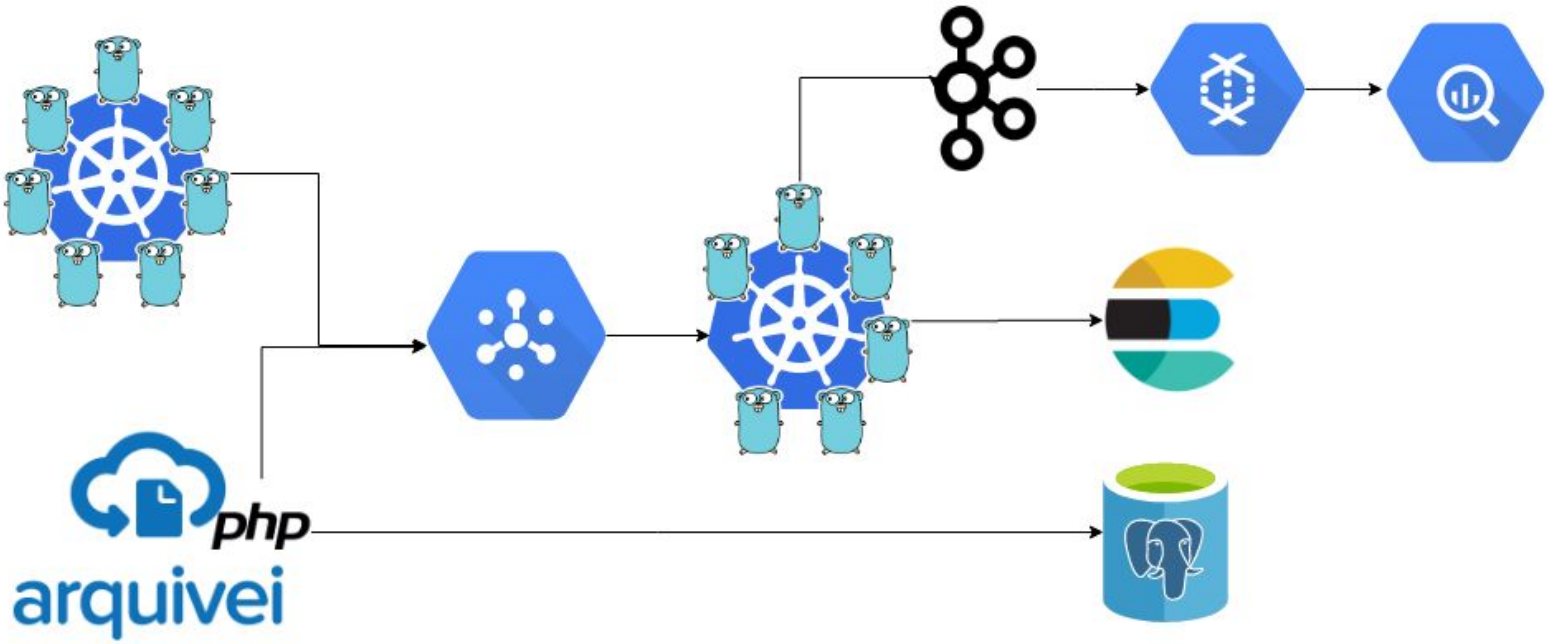
Dataflow model

- Akidau
- Bounded: batches
- Unbounded: streaming
- Embarrassingly parallel

Apache Beam

- Implementação do modelo
- Linguagens
  - Java
  - Python
  - Go (experimental)
  - Portability Framework
- I/Os prontos
  - GCP
  - AWS
  - Elasticsearch, Kafka, etc

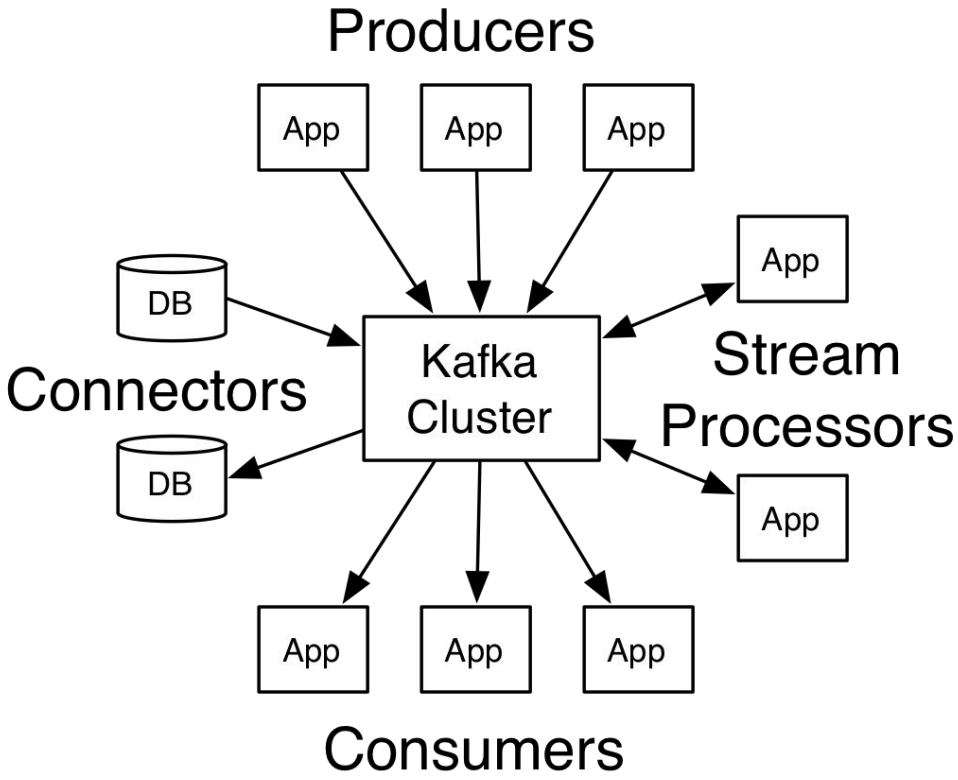
Arquitetura de dados atual



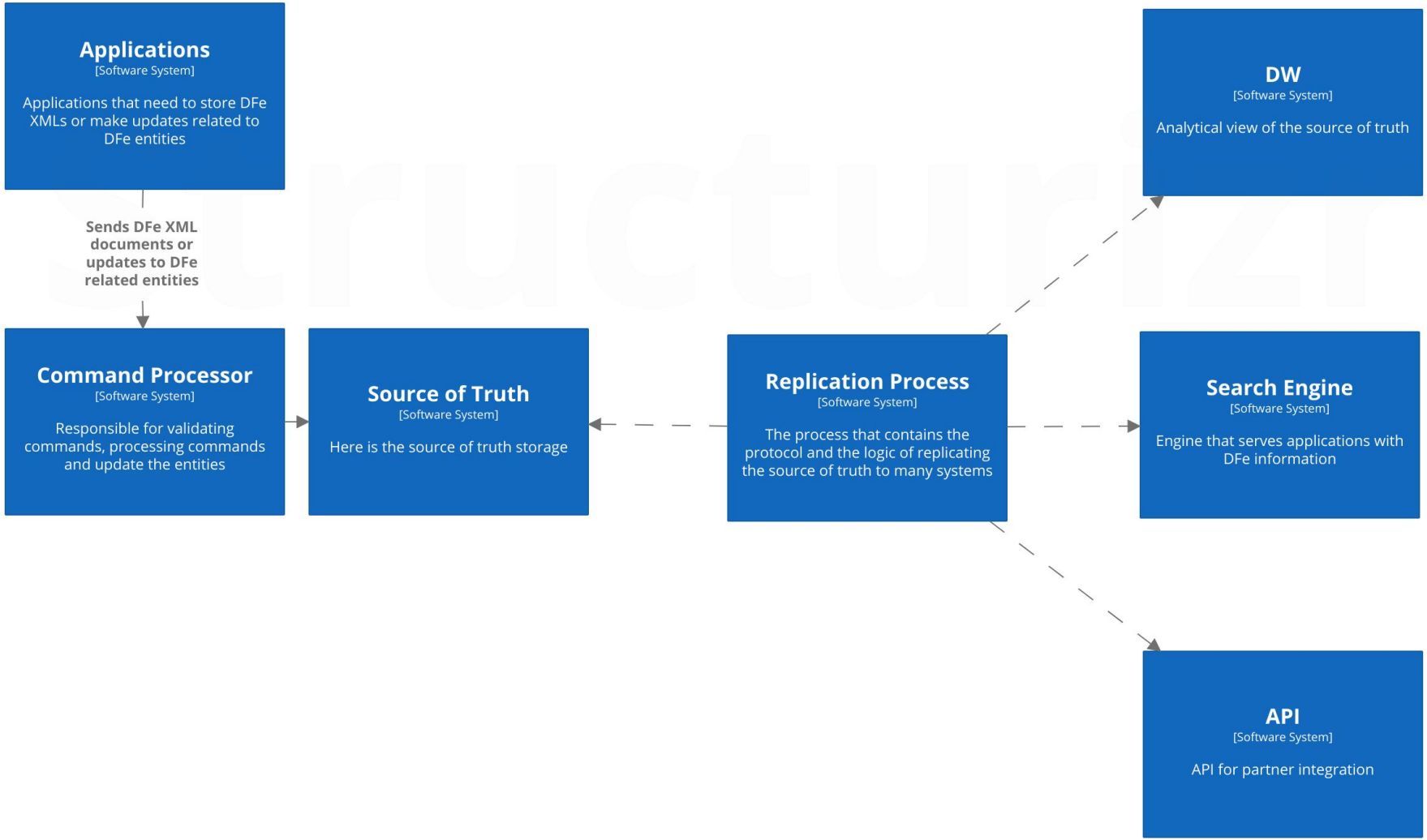
Arquitetura atual

- Problemas:
  - Concorrência
  - Rastreabilidade
  - Consistência
    - Escrita em vários bancos
    - Armazenamento de histórico

v3.0: Kafka Streams



Plataforma de Dados


Applications
[Software System]
Applications that need to store DFe XMLs or make updates related to DFe entities
Sends DFe XML documents or updates to DFe related entities
Command Processor
[Software System]
Responsible for validating commands, processing commands and update the entities
Source of Truth
[Software System]
Here is the source of truth storage
Replication Process
[Software System]
The process that contains the protocol and the logic of replicating the source of truth to many systems
DW
[Software System]
Analytical view of the source of truth
Search Engine
[Software System]
Engine that serves applications with DFe information
API
[Software System]
API for partner integration

Divisão de responsabilidades

- Ingestão
  - Responsável pela coleta de dados
- **Fonte de Verdade**
  - **Responsável pela consistência**
- Disponibilização
  - Responsável pelas *views*

SOT: BigTable

- Escalável
- Gerenciado
- Integra bem com BigQuery

SOT: BigTable

- Escalável
- Gerenciado
- Integra bem com BigQuery
- Não-transacional
- **Fonte de Verdade**
  - **Consistente**

Worker SOT

- É responsável por manter consistência
- Como?
  - *“A mesma chave de acesso nunca será processada por threads diferentes ao mesmo tempo”*

Golang X Kafka Streams

- Golang
  - Flexível
  - Concorrência ok
  - Integração com Kafka
    - Sarama
    - SegmentIO
    - Goduck

Golang X Kafka Streams

- Golang
  - Flexível
  - Concorrência ok
  - Integração com Kafka
    - Sarama
    - SegmentIO
    - Goduck
- Kafka Streams
  - Flexível, Concorrente, Kafka tbm

Golang X Kafka Streams

- Pró Go
  - XP do time
  - Reaproveitamento de código
  - Linguagem
- Pró KStreams
  - Facilidade de manter consistência
  - Desempenho com Kafka

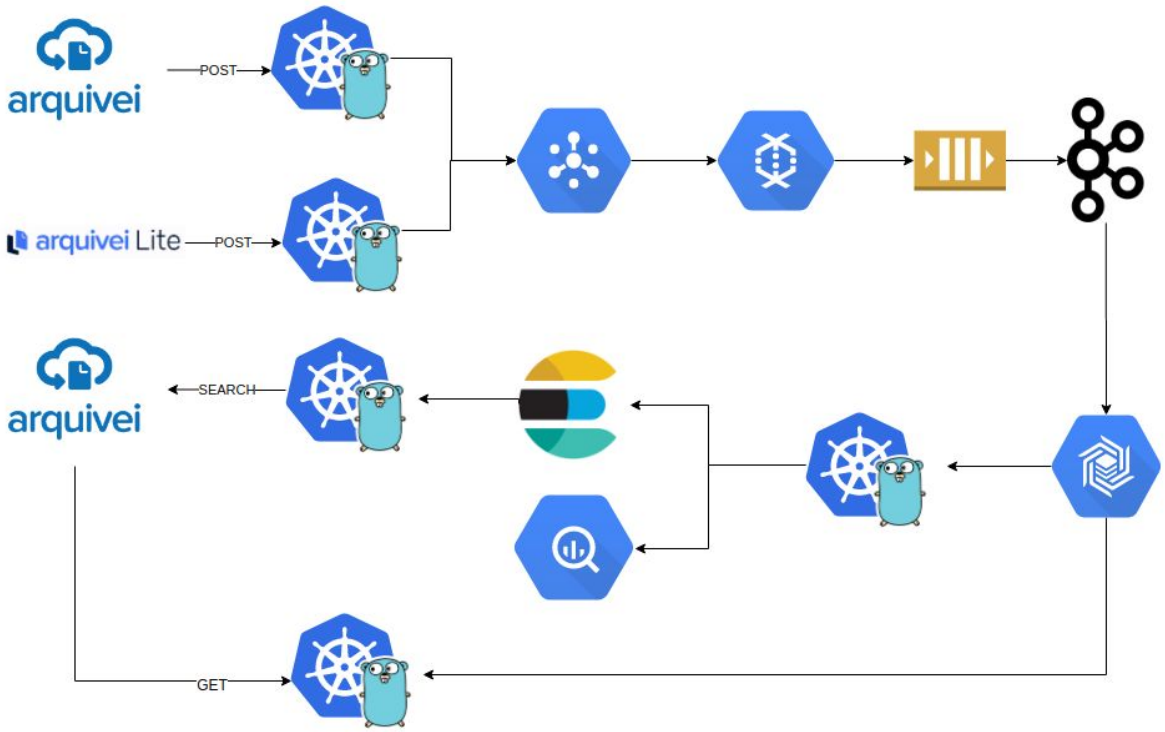
Golang X Kafka Streams

- Pró Go
  - XP do time
  - Reaproveitamento de código
  - Linguagem
- Pró KStreams
  - Facilidade de manter **consistência**
  - Desempenho com Kafka

Divisão de responsabilidades

- Golang
  - Command processor: APIs
- Google Dataflow
  - Batches
  - Streams: Embarrassingly parallel
- Kafka Streams
  - Streams: Consistency-first

Plataforma de Dados



Perguntas?

FAQ

- Contatos:
  - @leonardobarbie (twitter)
  - leonardoam94@gmail.com
- Códigos: github.com/arquivei