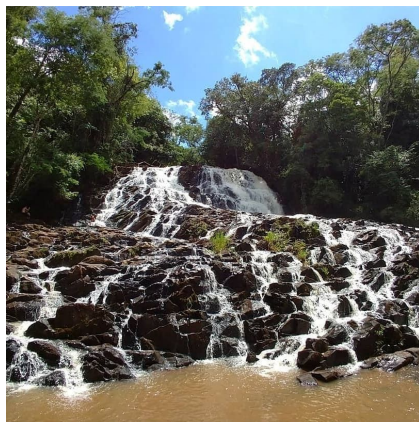


Beam: ETL Agnóstico

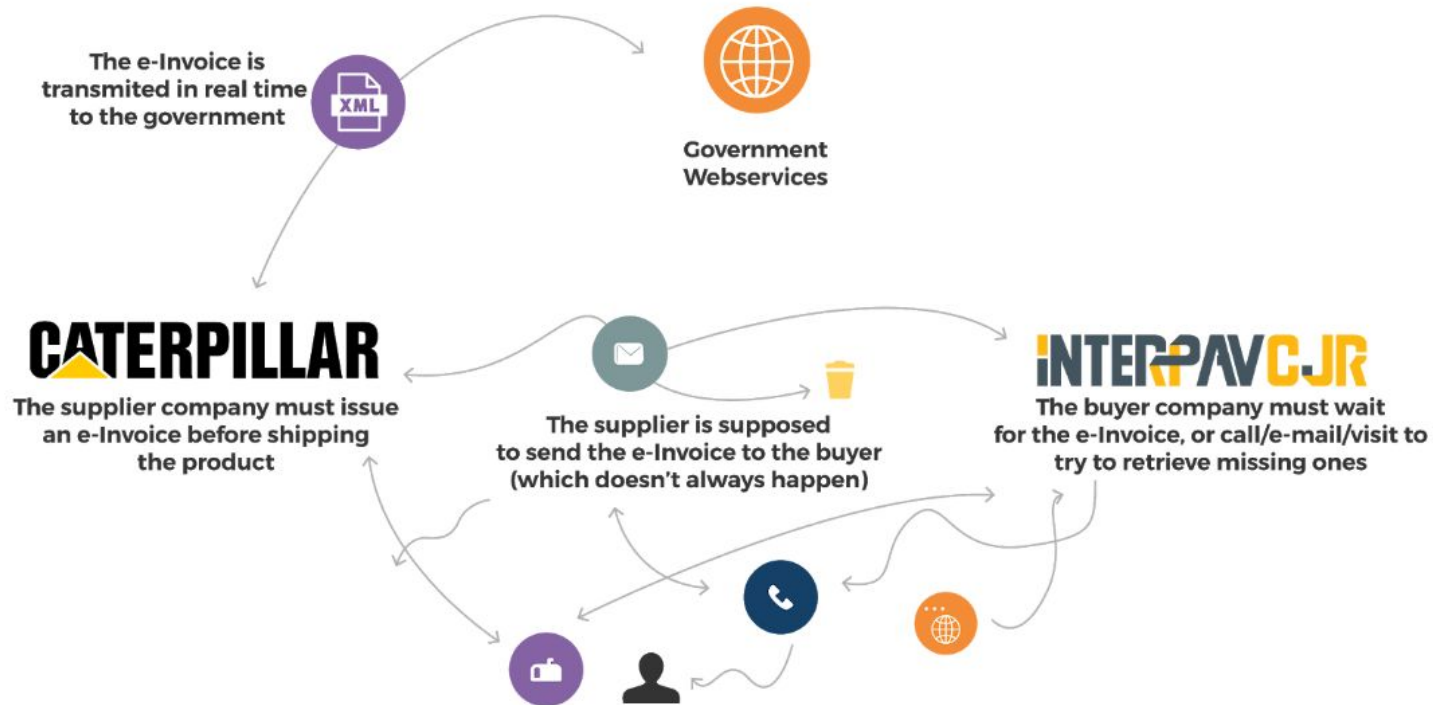
Roadmap

- ▶ Dataflow model
- ▶ Apache Beam aka Dataflow
- ▶ Google Dataflow





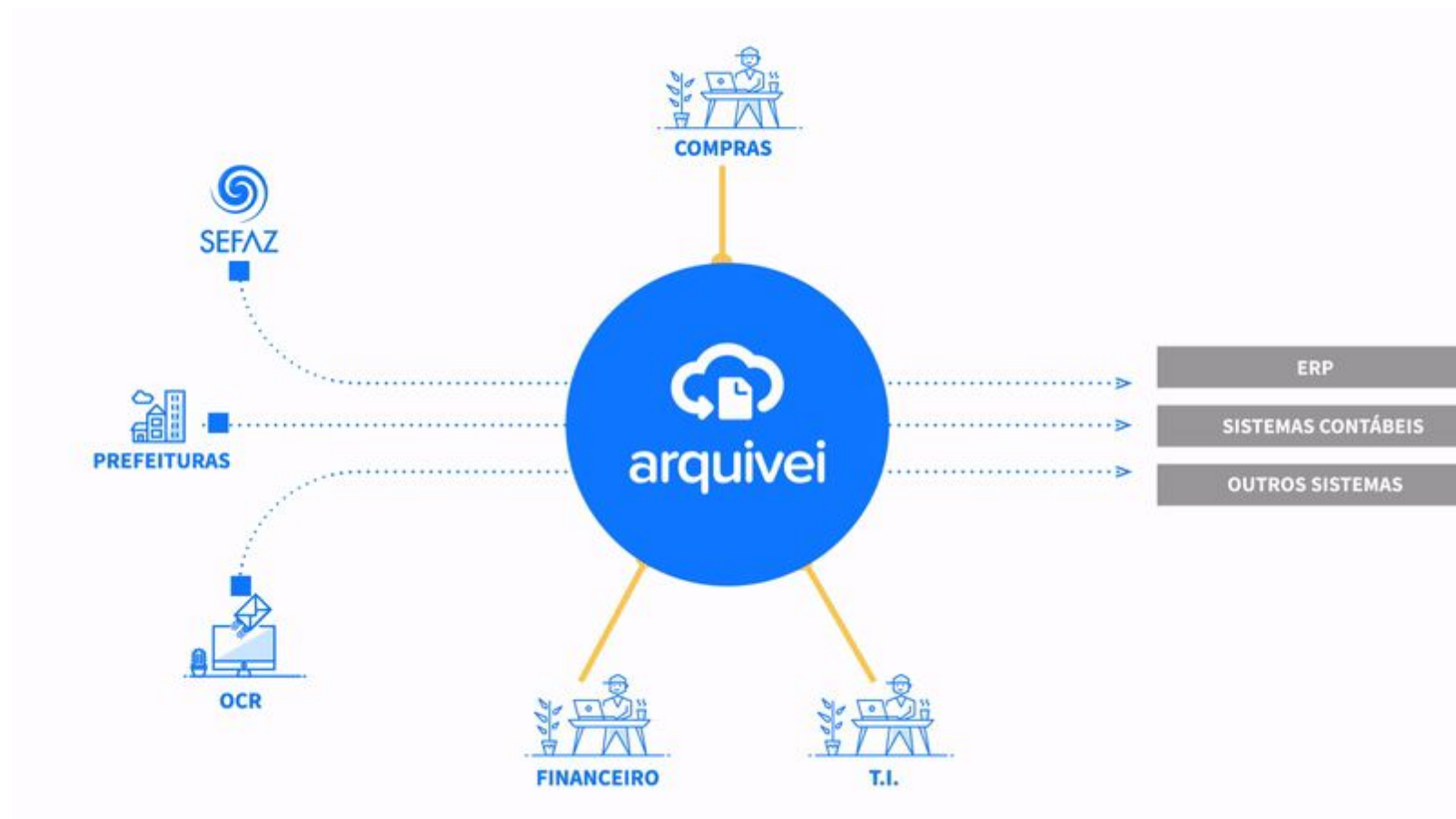
Arquivei

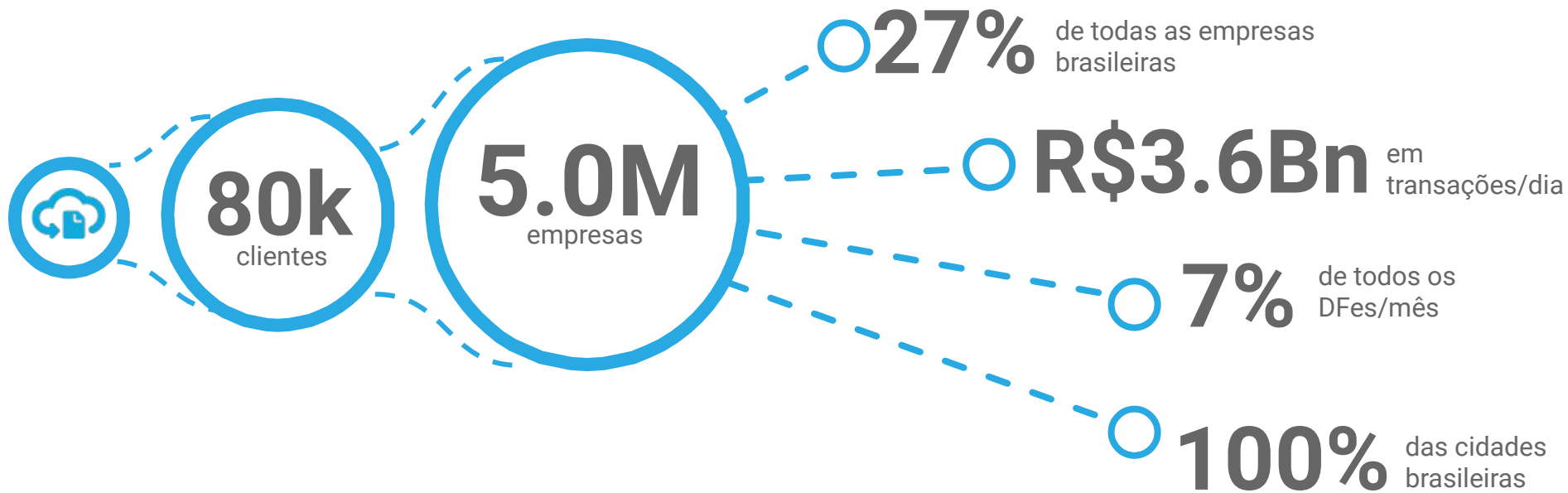


Arquivei



Democratizando a Informação

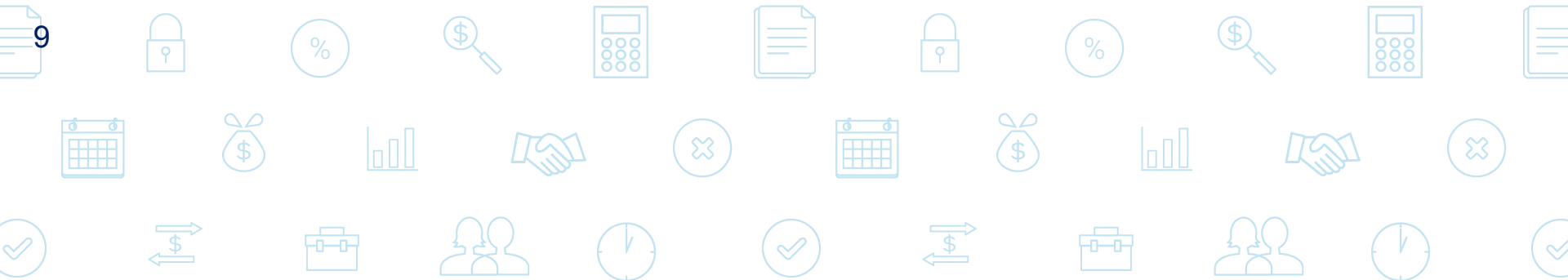






arquivoi

Temos vagas: arquivoi.com.br/vagas



Os vários dataflows

Dataflow model

- ▶ 2003 - GFS
- ▶ 2004 - MapReduce
- ▶ 2006 - Hadoop
- ▶ 2009 - Spark
- ▶ 2010 - Apache Flume
- ▶ 2013 - Google MillWheel
- ▶ **2015 - Dataflow model**



Dataflow model

- ▶ 2003 - GFS
- ▶ 2004 - MapReduce
- ▶ 2006 - Hadoop
- ▶ 2009 - Spark
- ▶ 2010 - Apache Flume
- ▶ 2013 - Google MillWheel
- ▶ **2015 - Dataflow model**
- ▶ 2019 - Palmeiras continua sem mundial



Apache Beam

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



Google Dataflow

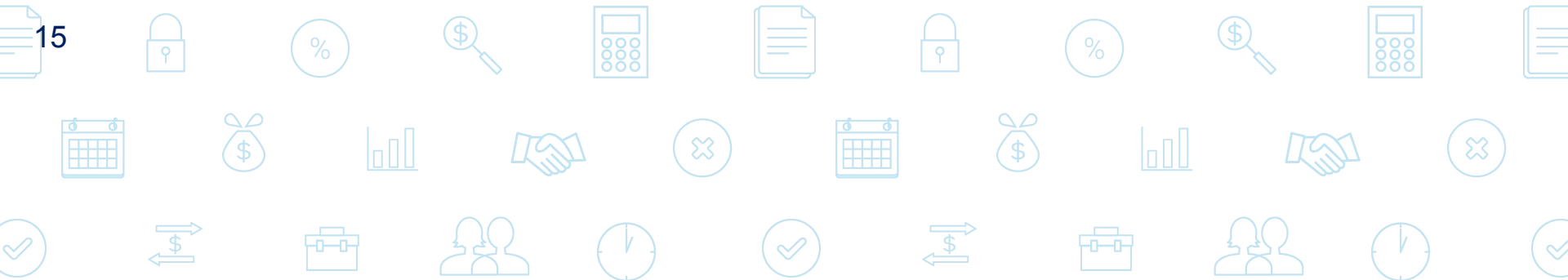
- ▶ Produto GCP
- ▶ Cluster gerenciado
- ▶ Autoscaling



Google Dataflow

► Produto GCP





O modelo

Dataflow model

- ▶ Akidau
- ▶ Bounded: batches
- ▶ Unbounded: streaming

Dataflow model

- ▶ Akidau
- ▶ Bounded -> Unbounded
- ▶ Unbounded -> Bounded



Dataflow model

- ▶ Akidau
- ▶ FlumeJava (batch)
+
- ▶ MillWheel (streaming)
- ▶ Batches existentes
 - Alta latência
- ▶ Streamings existentes
 - Tolerância a falhas, escalabilidade, latência
 - Complexidade (janelamento)



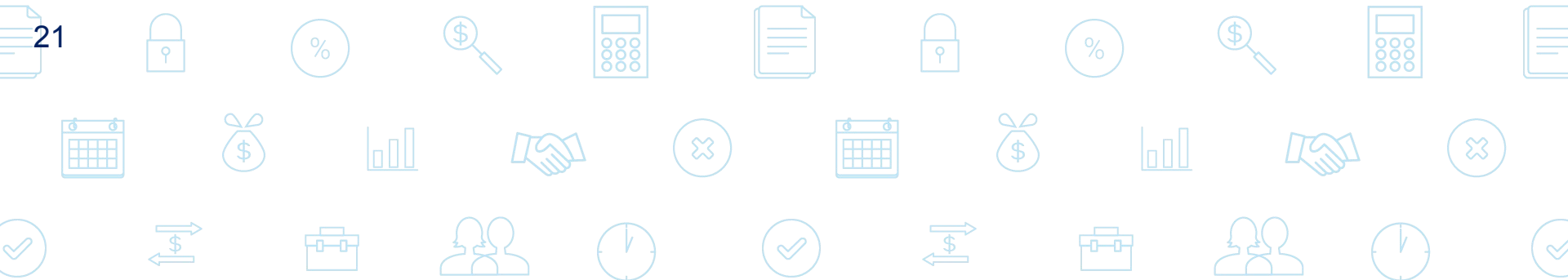
Tempo

- ▶ Event time
- ▶ Processing time

4 dimensões

- ▶ **Quais** resultados computados?
- ▶ **Onde** serão computados?
 - em event time
- ▶ **Quando** serão materializados?
 - em processing time
- ▶ **Como** refinar dados recentes mais tarde?





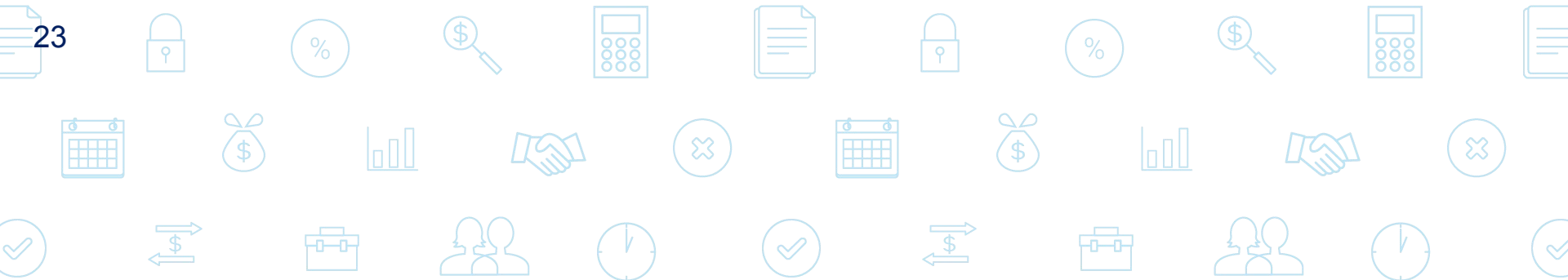
O que?



Primitivas

- ▶ ParDo == Map
- ▶ GroupByKey == Reduce





Onde?



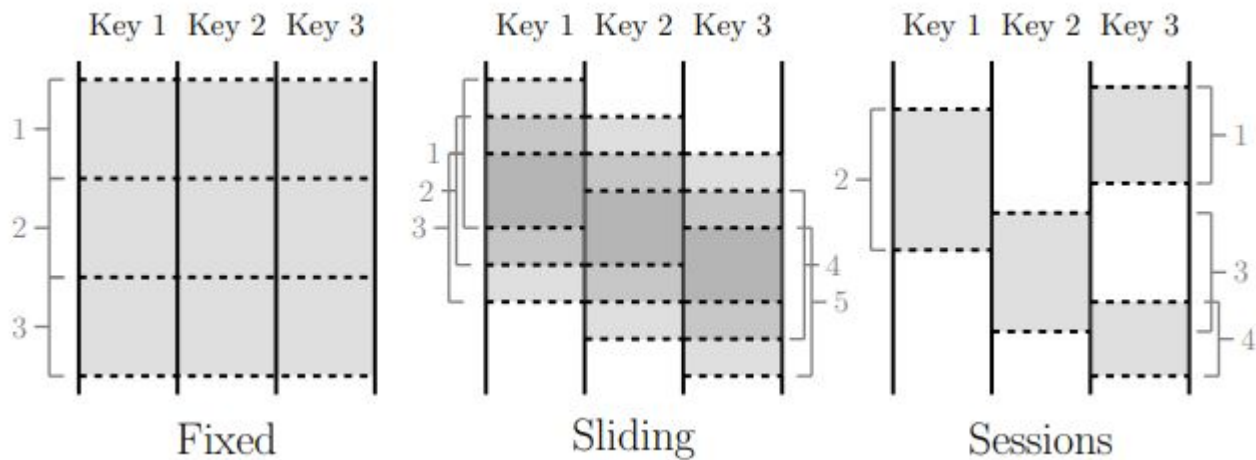
Windows

- ▶ Divisão do dataset para processamento
- ▶ Event time
- ▶ Dois tipos
 - Aligned
 - se aplica a todo dataset
 - Unaligned
 - se aplica a um subset



Windows

- Janelas de tempo



Primitivas

- ▶ AssignWindows
- ▶ MergeWindows
- ▶ GroupByKeyAndWindow



Primitivas

$(k_1, v_1, 13:02, [0, \infty)),$
 $(k_2, v_2, 13:14, [0, \infty)),$
 $(k_1, v_3, 13:57, [0, \infty)),$
 $(k_1, v_4, 13:20, [0, \infty))$

↓ *GroupByKey*

↓ *GroupAlsoByWindow*

↓ *AssignWindows*(
 Sessions(30m))

$(k_1, v_1, 13:02, [13:02, 13:32)),$
 $(k_2, v_2, 13:14, [13:14, 13:44)),$
 $(k_1, v_3, 13:57, [13:57, 14:27)),$
 $(k_1, v_4, 13:20, [13:20, 13:50))$

$(k_1, [(v_1, [13:02, 13:32)),$
 $(v_3, [13:57, 14:27)),$
 $(v_4, [13:20, 13:50))]),$
 $(k_2, [(v_2, [13:14, 13:44))])$

$(k_1, [([v_1, v_4], [13:02, 13:50)),$
 $([v_3], [13:57, 14:27))]),$
 $(k_2, [([v_2], [13:14, 13:44))])$

↓ *MergeWindows*(
 Sessions(30m))

↓ *ExpandToElements*

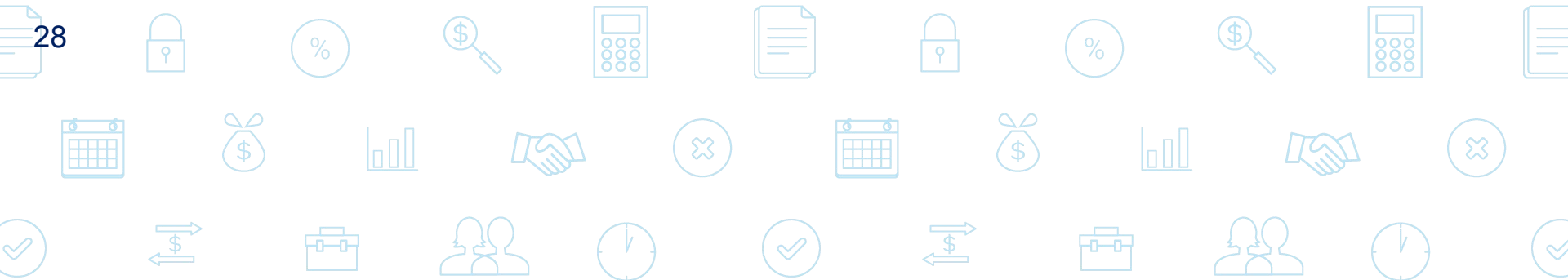
↓ *DropTimestamps*

$(k_1, v_1, [13:02, 13:32)),$
 $(k_2, v_2, [13:14, 13:44)),$
 $(k_1, v_3, [13:57, 14:27)),$
 $(k_1, v_4, [13:20, 13:50))$

$(k_1, [(v_1, [13:02, 13:50)),$
 $(v_3, [13:57, 14:27)),$
 $(v_4, [13:02, 13:50))]),$
 $(k_2, [(v_2, [13:14, 13:44))])$

$(k_1, [v_1, v_4], 13:50, [13:02, 13:50)),$
 $(k_1, [v_3], 14:27, [13:57, 14:27)),$
 $(k_2, [v_2], 13:44, [13:14, 13:44))$





Quando?



Triggers

- ▶ Determina quando um GroupByKeyAndWindow ocorre
- ▶ Emite um Pane
 - Window processada



Triggers

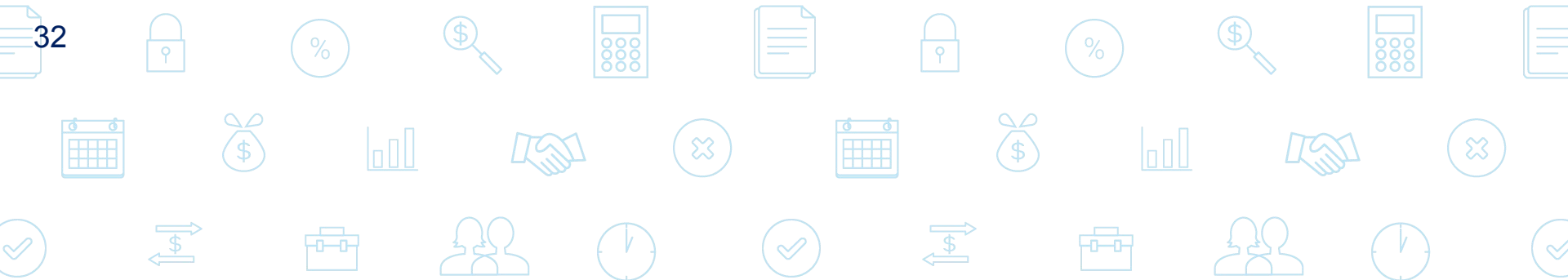
- ▶ Determina quando um GroupByKeyAndWindow ocorre
- ▶ Emite um Pane
 - Window processada
- ▶ Late data
 - Atraso: event time vs processing time



Tipos de Triggers

- ▶ Baseado em Processing Time
- ▶ Baseado em conteúdo
- ▶ Baseado em sinais externos
- ▶ Triggers compostos





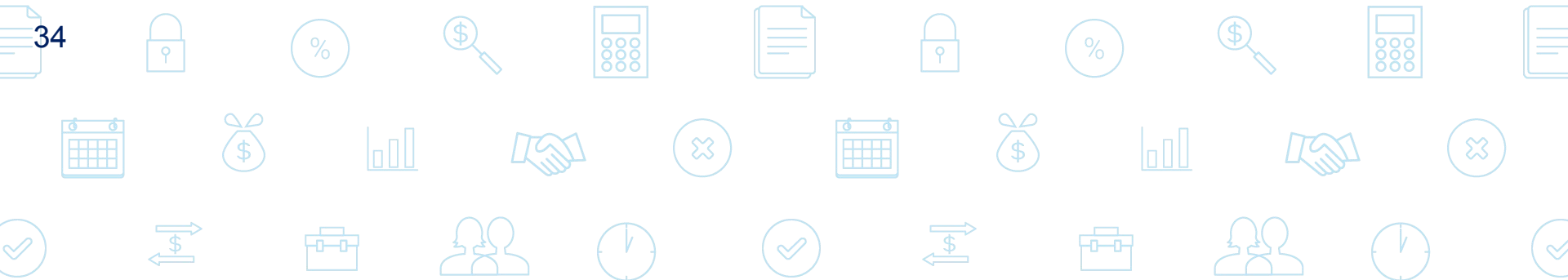
Como?



Accumulation modes

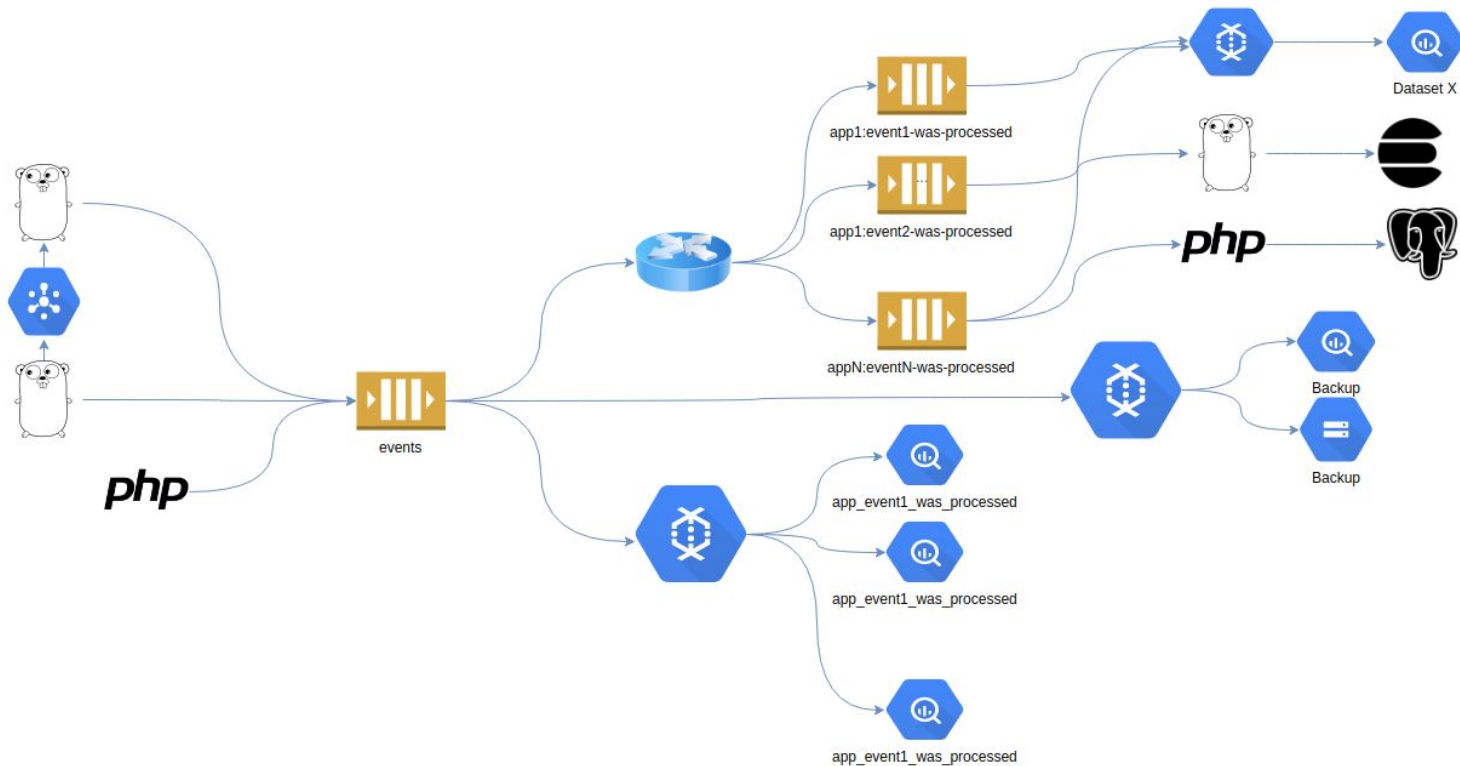
- ▶ Descarte
 - resultados futuros não dependem de dados passados
- ▶ Acúmulo
 - resultados futuros dependem de dados passados
- ▶ Acúmulo + Retração
 - resultados futuros dependem de dados passados
 - resultados passados dependem de dados futuros





Como usamos

Kafka+Dataflow



FAQ

- ▶ Bibliografia
 - Akidau, 2015
 - Designing Data-Intensive Applications
- ▶ Códigos: github.com/arquivei
- ▶ Contatos:
 - @leonardobarbie (twitter)
 - leonardo.miguel@arquivei.com.br

