A gentle introduction to Stream Processing

Nicolas Fränkel

Me, myself and I

- 18 years in technical roles:
  - Developer, team lead, architect, …
- Developer Advocate

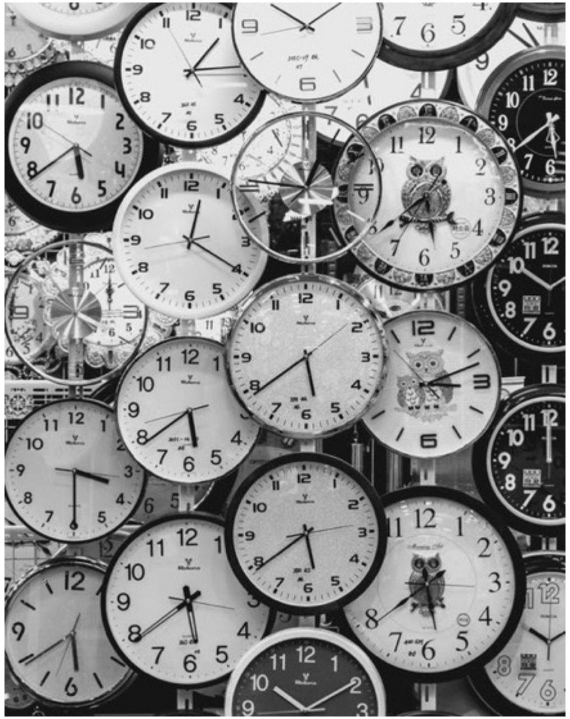
Hazelcast

HAZELCAST IMDG is an **operational, in-memory**, distributed computing platform that manages data using in-memory storage and performs parallel execution for breakthrough application speed and scale.

HAZELCAST JET is the ultra fast, application embeddable, 3rd generation stream processing engine for low latency batch and stream processing.

Schedule

- Why streaming?
- Streaming approaches
- Hazelcast Jet
- Open Data
- General Transit Feed Specification
- The demo



In a time before our time…

Data was neatly stored in SQL databases

The need for Extract Transform Load

- Analytics
  - Supermarket sales in the last hour?
- Reporting
  - Banking account annual closing

Writes vs. reads

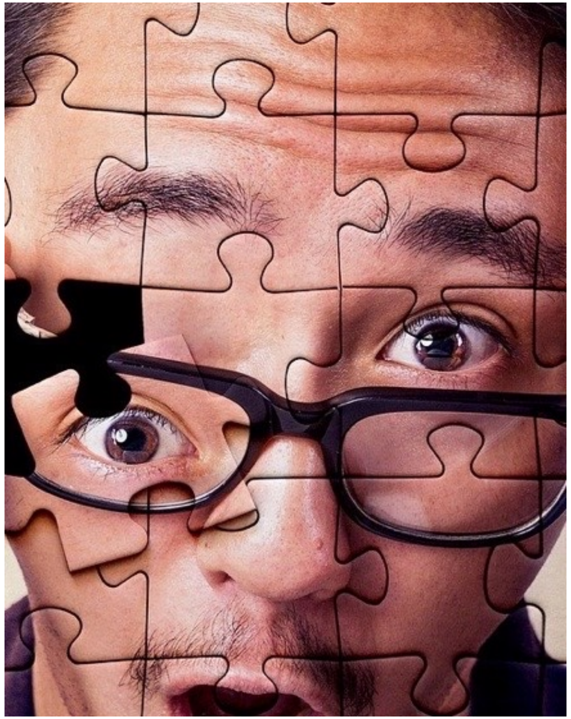
- Normalized vs. denormalized
- Correct vs. fast

What SQL implies

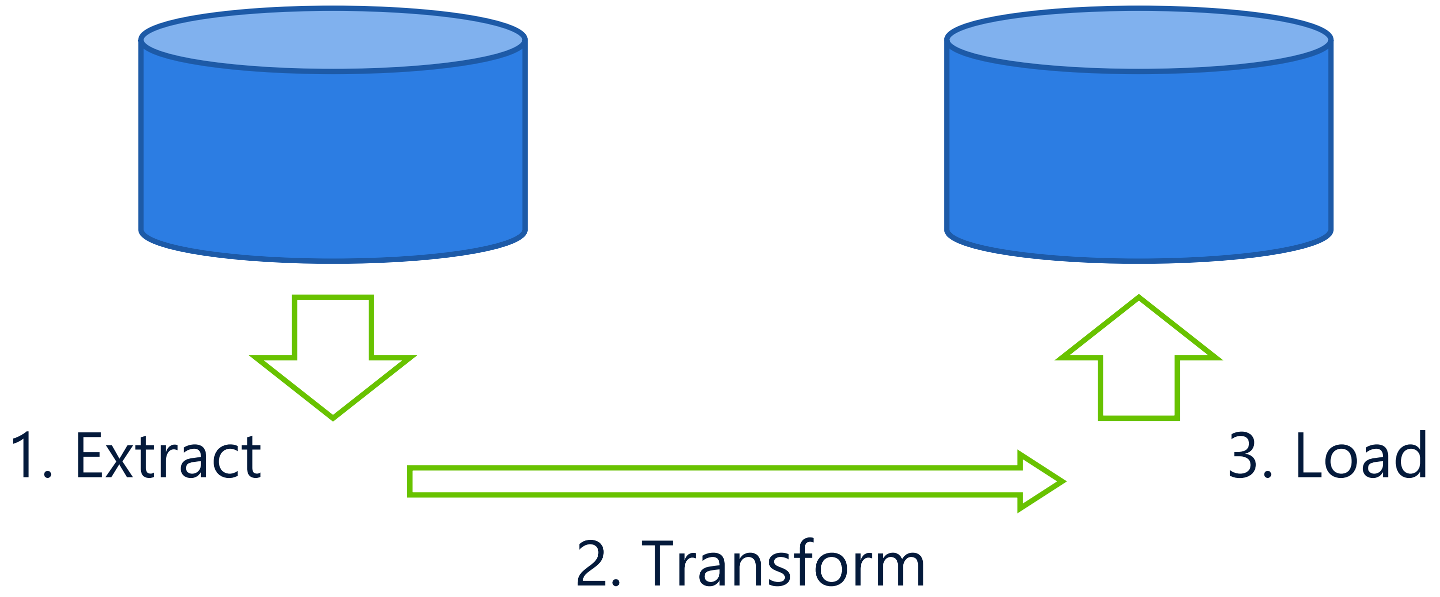
- Normal forms
- Joins
- Constraints

The need for ETL

- Different actors
- With different needs
- Using the same database?



The batch model



Batches are everywhere!

Properties of batches

- Scheduled at regular intervals
  - Daily
  - Weekly
  - Monthly
  - Yearly
  - etc.
- Run in a specific amount of time

Oops

- When the execution time overlaps the next execution schedule
- When the space taken by the data exceeds the storage capacity
- When the batch fails mid-execution
- etc.

Chunking!

- Keep a cursor
  - And only manage “chunks” of data
- What about new data coming in?

Big data!

- Parallelize everything
  - Map - Reduce
  - Hadoop
- NoSQL
  - Schema on Read vs. Schema on Write

Event

“In programming and software design, an event is **an action or occurrence** recognized by software, often originating asynchronously from the external environment, that may be handled by the software. Computer events can be generated or triggered by the system, by the user, or in other ways.”

-- Wikipedia

Make everything event-based!

Benefits

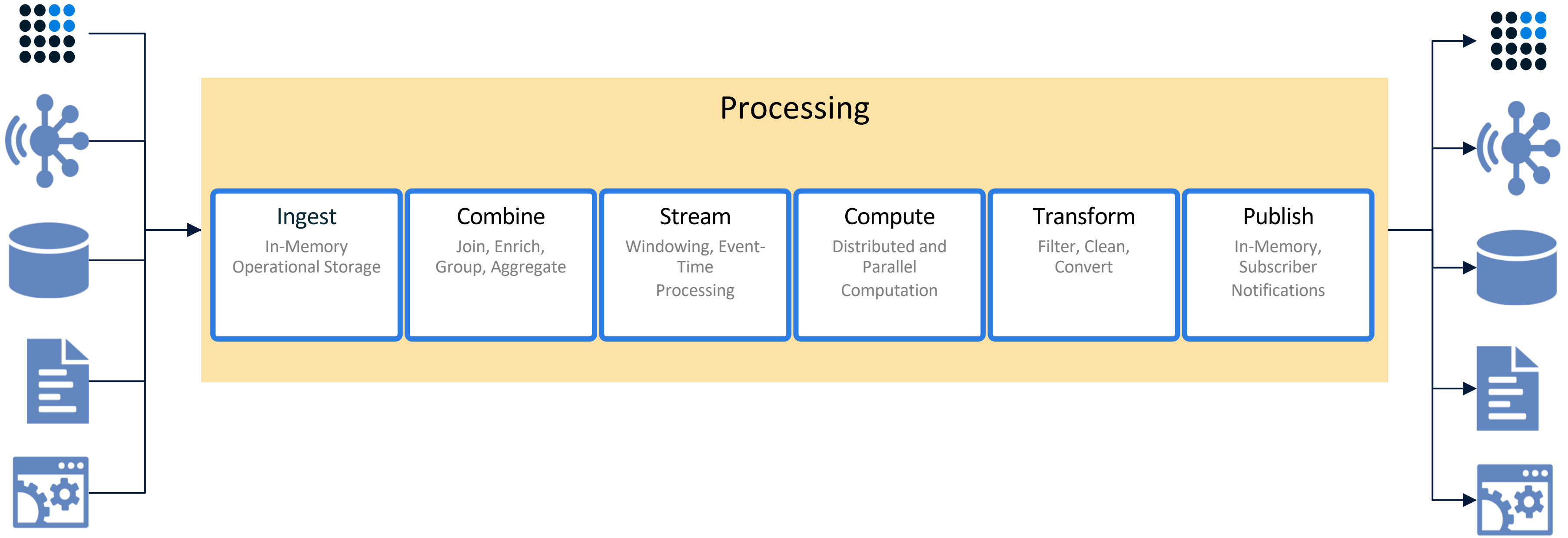
- Memory-friendly
- Easily processed
- Pull vs. push
  - Very close to real-time
  - Keeps derived data in-sync

From finite datasets to infinite

Stateful streams

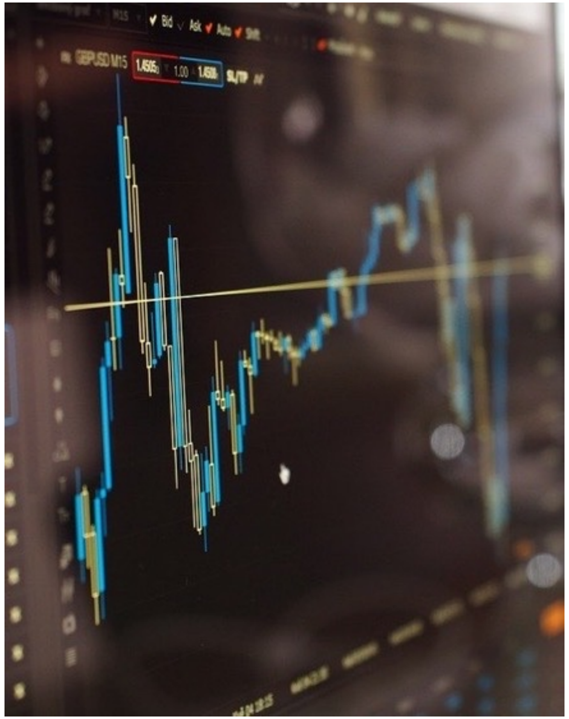
- Aggregation
- Windowing

Streaming is “smart” ETL


Processing
Ingest
In-Memory Operational Storage
Combine
Join, Enrich, Group, Aggregate
Stream
Windowing, Event-Time Processing
Compute
Distributed and Parallel Computation
Transform
Filter, Clean, Convert
Publish
In-Memory, Subscriber Notifications

Analytics and Decision Making

- Real-time dashboards
- Stats
- Predictions
  - Push stream through ML model
- Complex-Event-Processing



Persistent event-storage systems

- Apache Kafka
- Apache Pulsar

Apache Kafka

- Distributed
- On-disk storage
- Messages sent and read from a topic
- Consumer can keep track of the offset

Some in-memory stream processing engines

- On-premise
  - Apache Flink
  - Hazelcast Jet
- Cloud-based
  - Amazon Kinesis
  - Google Dataflow
- Apache Beam
  - Abstraction over some of the above

Hazelcast Jet

- Apache 2 Open Source
- Leverages Hazelcast IMDG
- Unified batch/streaming API
- (Hazelcast Jet Enterprise)

Pipeline

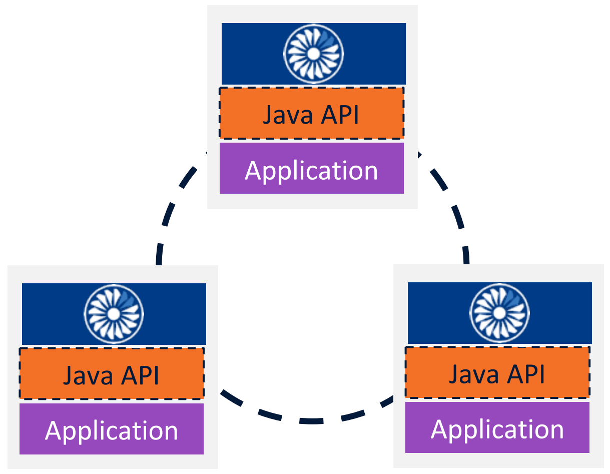
- Declarative code that defines and links sources, transforms, and sinks
- Platform-specific SDK
- Client submits pipeline to the SPE

Job

- Running instance of pipeline in SPE
- SPE executes the pipeline
  - Code execution
  - Data routing
  - Flow control

Deployment modes

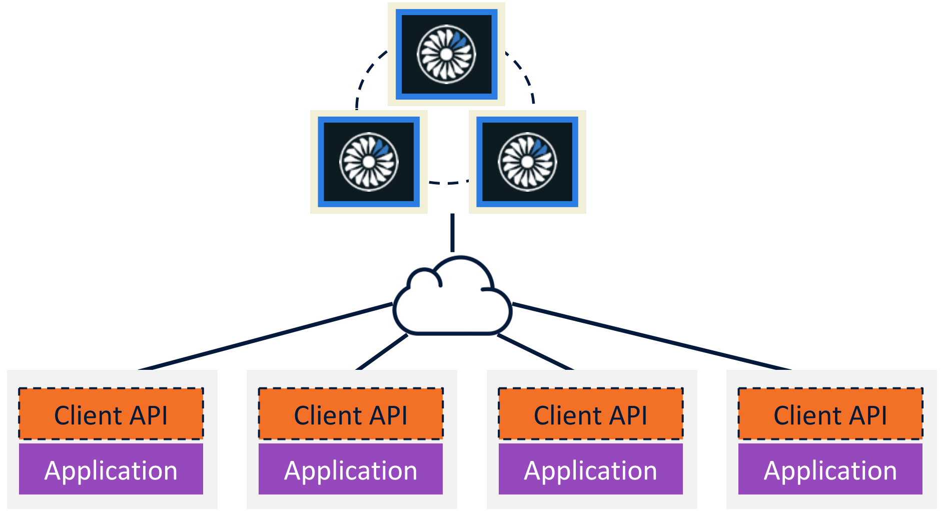
Embedded



```
// Create new cluster member

JetInstance jet = Jet.newJetInstance();
```

Client/Server



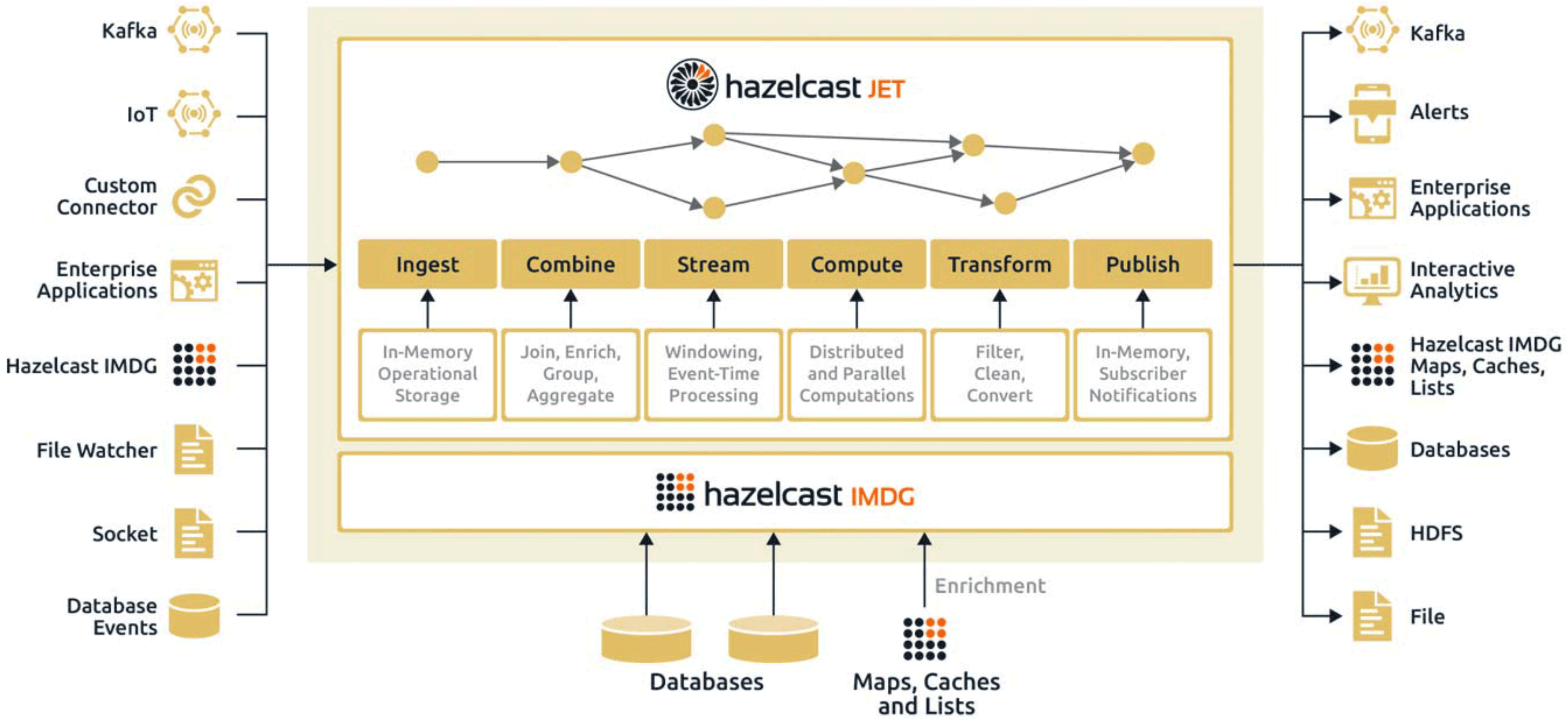
```
// Connect to  running cluster

JetInstance jet = Jet.newJetClient();
```

@nicolas_frankel

hazelcast®

Hazelcast Jet



@nicolas_frankel

hazelcast®

Open Data

« **Open data** is the idea that some data should be freely available to everyone to use and republish as they wish, without restrictions from copyright, patents or other mechanisms of control. »

--https://en.wikipedia.org/wiki/Open_data

Some Open Data initiatives

- France:
  - https://www.data.gouv.fr/fr/
- Switzerland:
  - https://opendata.swiss/en/
- European Union:
  - https://data.europa.eu/euodp/en/data/

Challenges

1. Access
2. Format
3. Standard
4. Data correctness

Access

- Access data interactively through a web-service
- Download a file

Format

In general, Open Data means Open Format

- PDF
- CSV
- XML
- JSON
- etc.

Standard

- Let's pretend the format is XML
  - Which grammar is used?
- A shared standard is required
  - Congruent to a domain

Data correctness

```
"32.TA.66-43","16:20:00","16:20:00","8504304"
"32.TA.66-44","24:53:00","24:53:00","8500100"
"32.TA.66-44","25:00:00","25:00:00","8500162"
"32.TA.66-44","25:02:00","25:02:00","8500170"
"32.TA.66-45","23:32:00","23:32:00","8500170"
```

A standard for Public Transport

- General Transit Feed Specification (GTFS)
- " […] defines a **common format for public transportation schedules and associated geographic information**. GTFS feeds let public transit agencies publish their transit data and developers write applications that consume that data in an interoperable way."
- Based on two kinds of data:
  - "**Static**" *e.g.* stops
  - **Dynamic** *e.g.* position

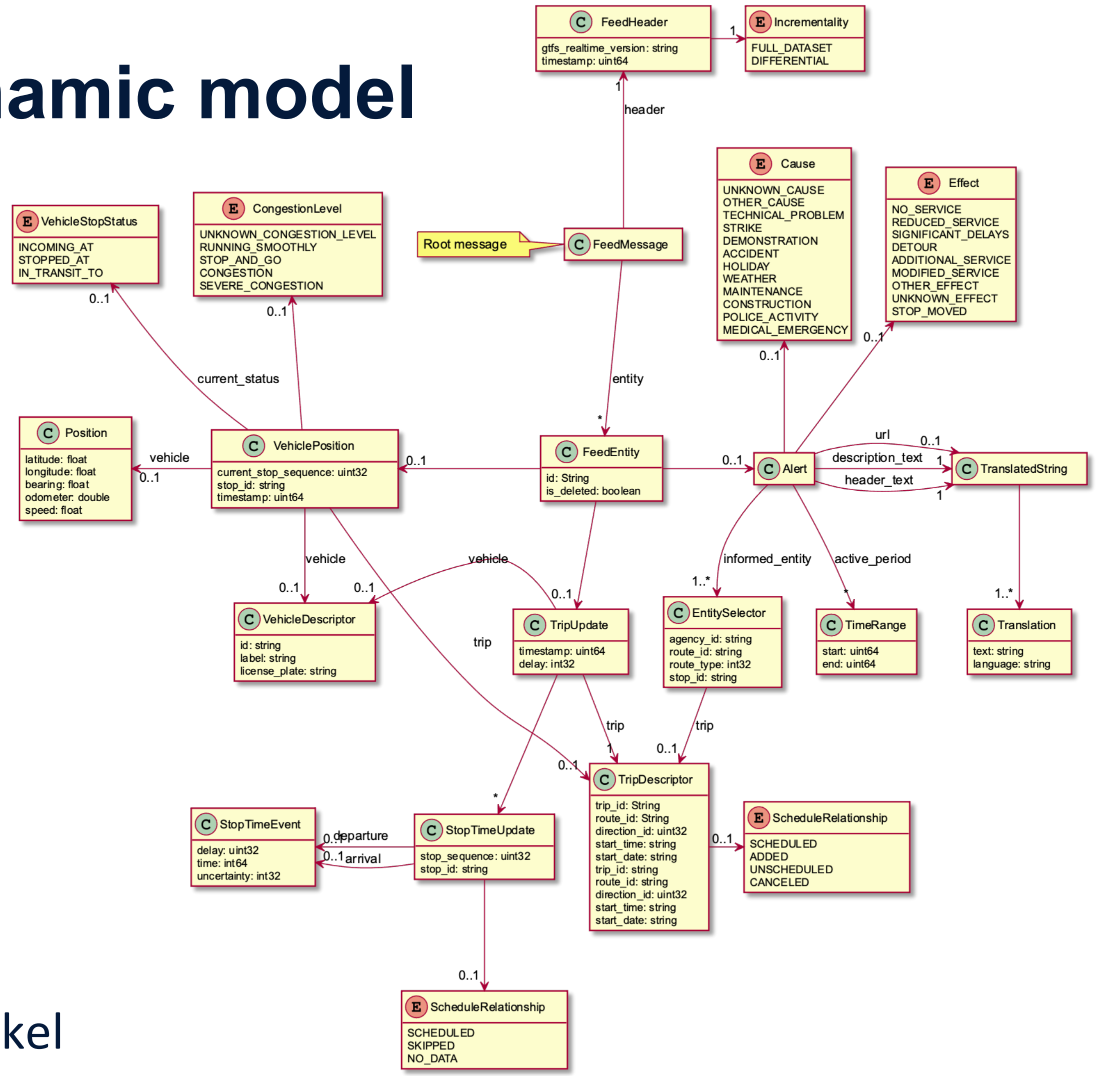
GTFS static model

Filename	Required	Defines
agency.txt	**Required**	Transit agencies with service represented in this dataset.
stops.txt	**Required**	Stops where vehicles pick up or drop off riders. Also defines stations and station entrances.
routes.txt	**Required**	Transit routes. A route is a group of trips that are displayed to riders as a single service.
trips.txt	**Required**	Trips for each route. A trip is a sequence of two or more stops that occur during a specific time period.
stop_times.txt	**Required**	Times that a vehicle arrives at and departs from stops for each trip.
calendar.txt	**Conditionally required**	Service dates specified using a weekly schedule with start and end dates. This file is required unless all dates of service are defined in calendar_dates.txt.
calendar_dates.txt	**Conditionally required**	Exceptions for the services defined in the calendar.txt. If calendar.txt is omitted, then calendar_dates.txt is required and must contain all dates of service.
fare_attributes.txt	Optional	Fare information for a transit agency's routes.

GTFS static model

Filename	Required	Defines
fare_rules.txt	Optional	Rules to apply fares for itineraries.
shapes.txt	Optional	Rules for mapping vehicle travel paths, sometimes referred to as route alignments.
frequencies.txt	Optional	Headway (time between trips) for headway-based service or a compressed representation of fixed-schedule service.
transfers.txt	Optional	Rules for making connections at transfer points between routes.
pathways.txt	Optional	Pathways linking together locations within stations.
levels.txt	Optional	Levels within stations.
feed_info.txt	Optional	Dataset metadata, including publisher, version, and expiration information.
translations.txt	Optional	Translated information of a transit agency.
attributions.txt	Optional	Specifies the attributions that are applied to the dataset.

GTFS dynamic model



@nicolas_frankel

hazelcast®

A data provider

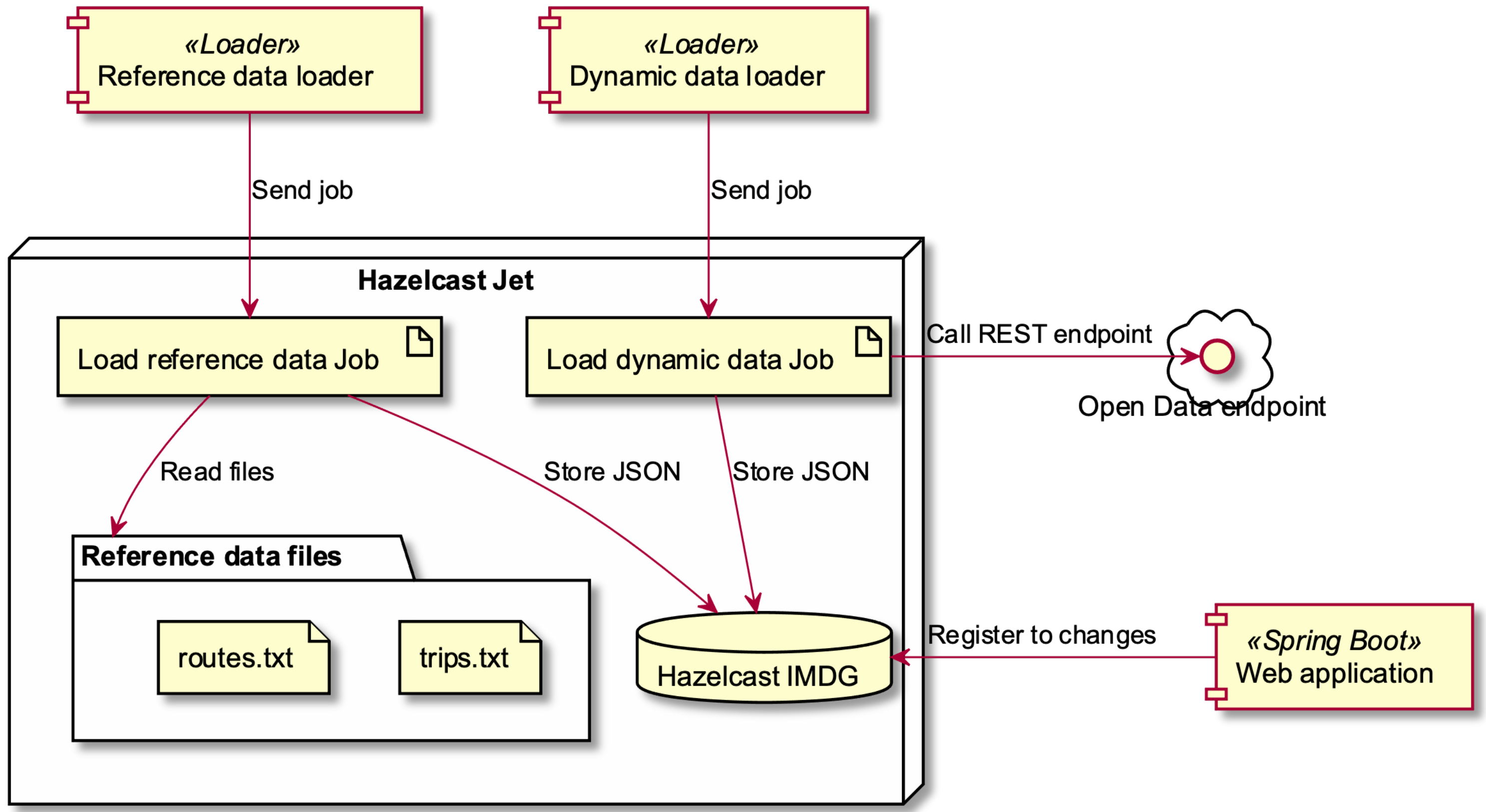
“511 is your phone and web source for Bay Area traffic, transit, carpool, vanpool, and bicycling information. It's FREE and available whenever you need it – 24 hours a day, 7 days a week – from anywhere in the nine-county Bay Area”

-- https://511.org/open-data

The dynamic data pipeline

1. Source: web service
2. **Split** into trip updates
3. Enrich with **trip data**
4. Enrich with **stop times data**
5. Transform hours into **timestamp**
6. Enrich with **location data**
7. Sink: Hazelcast IMDG

Architecture overview



Talk is cheap, show me the code!

Recap

- Streaming has a lot of benefits
- Leverage available Data
  - Open Data has a lot of untapped potential
- But you can get cool stuff done!

Belfort
Héricourt
Audincourt
Brombach
Wehr
Delémont
Blanche-Roche
La Chaux-de-Fonds
Biel
Neuchâtel
Murten
Flamatt
Fribourg
Bulle
Montreux
Langenthal

Thanks a lot!

- https://blog.frankel.ch/
- @nicolas_frankel
- https://jet-start.sh/
- https://bit.ly/jet-train
- https://slack.hazelcast.com/