Game, Set, Match Transforming Live Sports with AI-Driven Commentary

Dunith Danushka (Redpanda Data)
Mark Needham (ClickHouse)

Dunith Danushka
DevRel @ Redpanda

Mark Needham
Product @ ClickHouse

US Open: Djokovic beats Medvedev to win 24th Grand Slam - radio & text

10 September 2023

21:28 10 Sep 2023

Djokovic breaks serve

*Djokovic 2-0 Medvedev

Oh wow.

That is a huge miss on the second serve from Daniil Medvedev and that double fault opens the door for Novak Djokovic at 0-30.

And now the pressure really is on. A stunning forehand from Djokovic fizzes down the line.

Three break points... Only one needed! A miss from Medvedev. A nightmare start.

58 69 Share

21:30 10 Sep 2023

*Djokovic 2-0 Medvedev

This is the response Medvedev needed!

A superb angled forehand takes him to 0-30 as he chases an instant reply.

But Djokovic races in to put away a volley on his way to restoring parity.

58 35 Share

21:28 10 Sep 2023

Jonathan Jurejko
BBC Sport at Flushing Meadows

Hold the line. Djokovic is dialled in.

Stunning start from the 23-time Grand Slam champion, who is already playing patiently and precisely.

78 63 Share

21:28 10 Sep 2023

'Scrappy from Medvedev'

*Djokovic 2-0 Medvedev

Russell Fuller
Tennis correspondent on BBC Radio 5 Sports Extra

What a scrappy start from Medvedev.

49 65 Share

0:38 11 Sep 2023

Game, set and match - Djokovic

Djokovic 6-3 7-6 (7-5) 6-3 Medvedev

All over!

A historic triumph for Novak Djokovic!

A 24th Grand Slam title, a fourth US Open. Djokovic raises his arms aloft before kneeling on the court and letting the emotion out.

842 166 Share

0:36 11 Sep 2023

Championship point Djokovic

*Djokovic 6-3 7-6 (7-5) 5-3 Medvedev

But Djokovic will have his first championship point!

543 203 Share

0:34 11 Sep 2023

*Djokovic 6-3 7-6 (7-5) 5-3 Medvedev

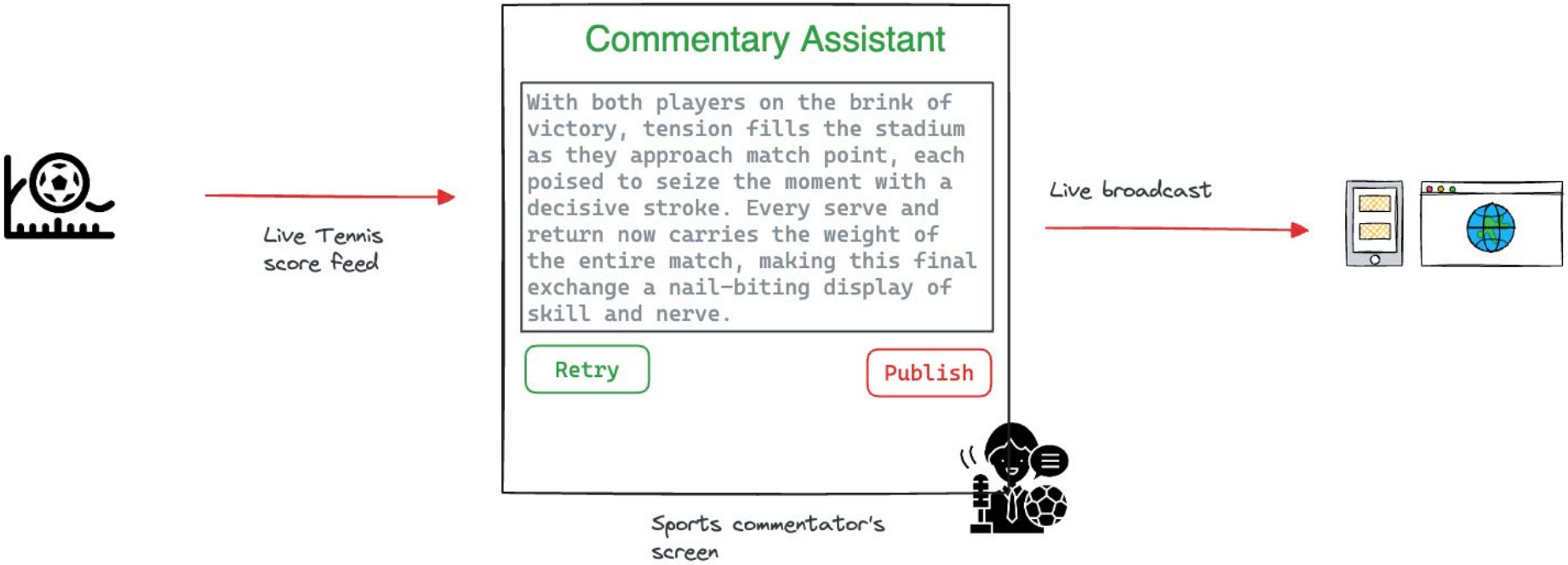
Oooh. A double fault.

Medvedev reaches 30-30. The excitement bubbles in the crowd. They want more.

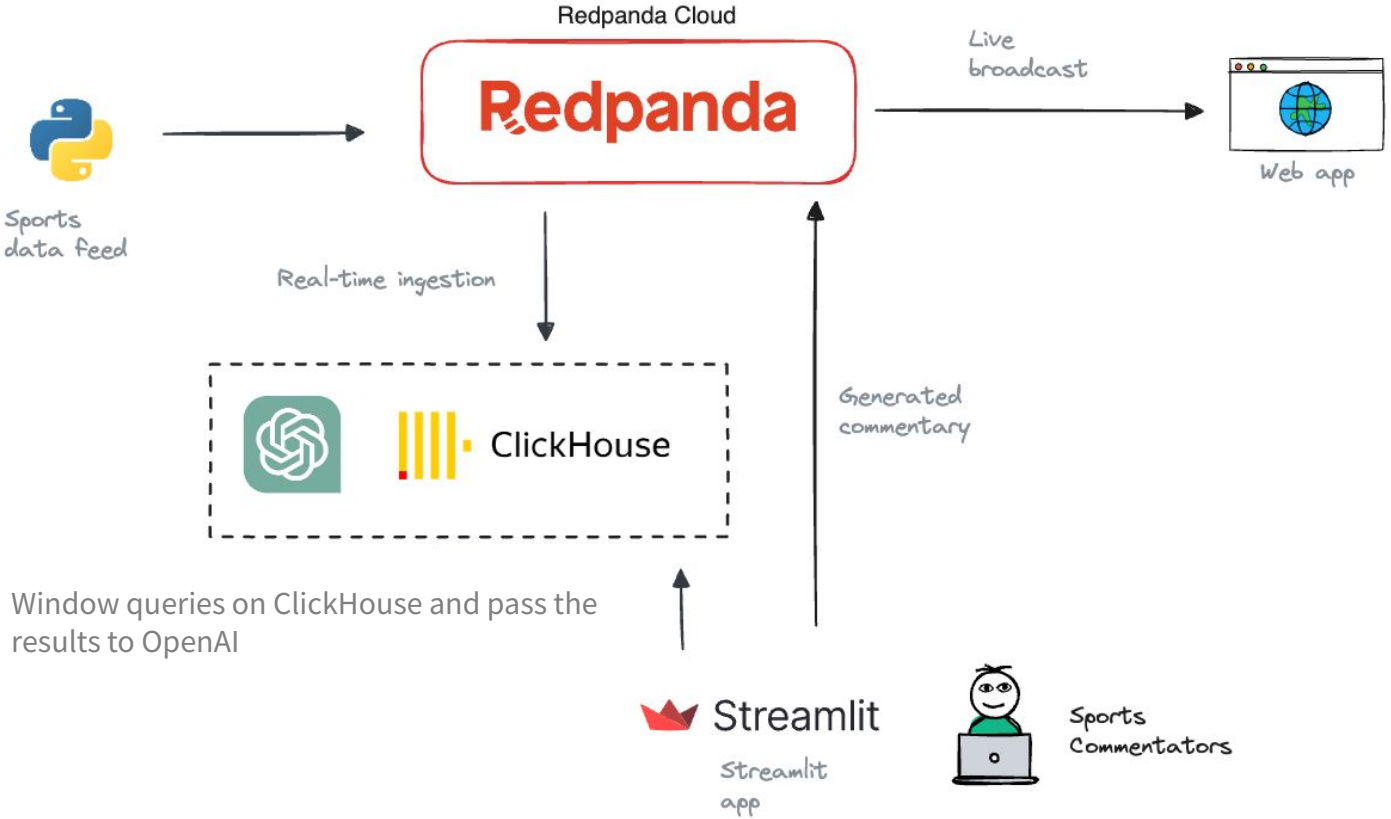
152 60 Share

Can we create an AI Copilot to help live text writers?

What are we going to build?



The flow of events



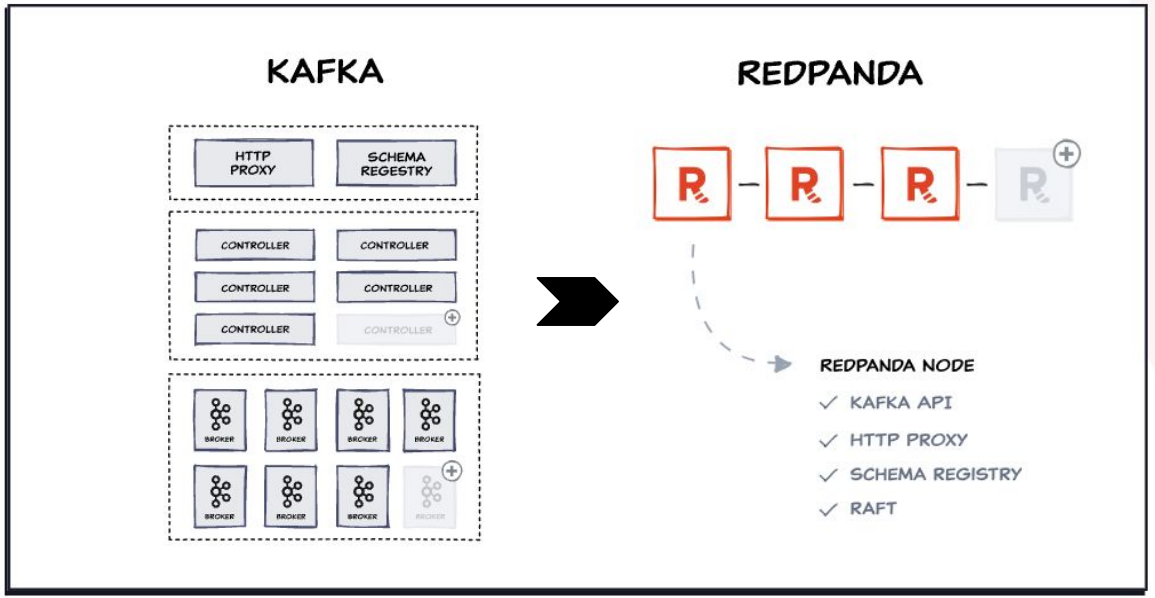
What is Redpanda?

Redpanda is a Kafka API compatible streaming data platform

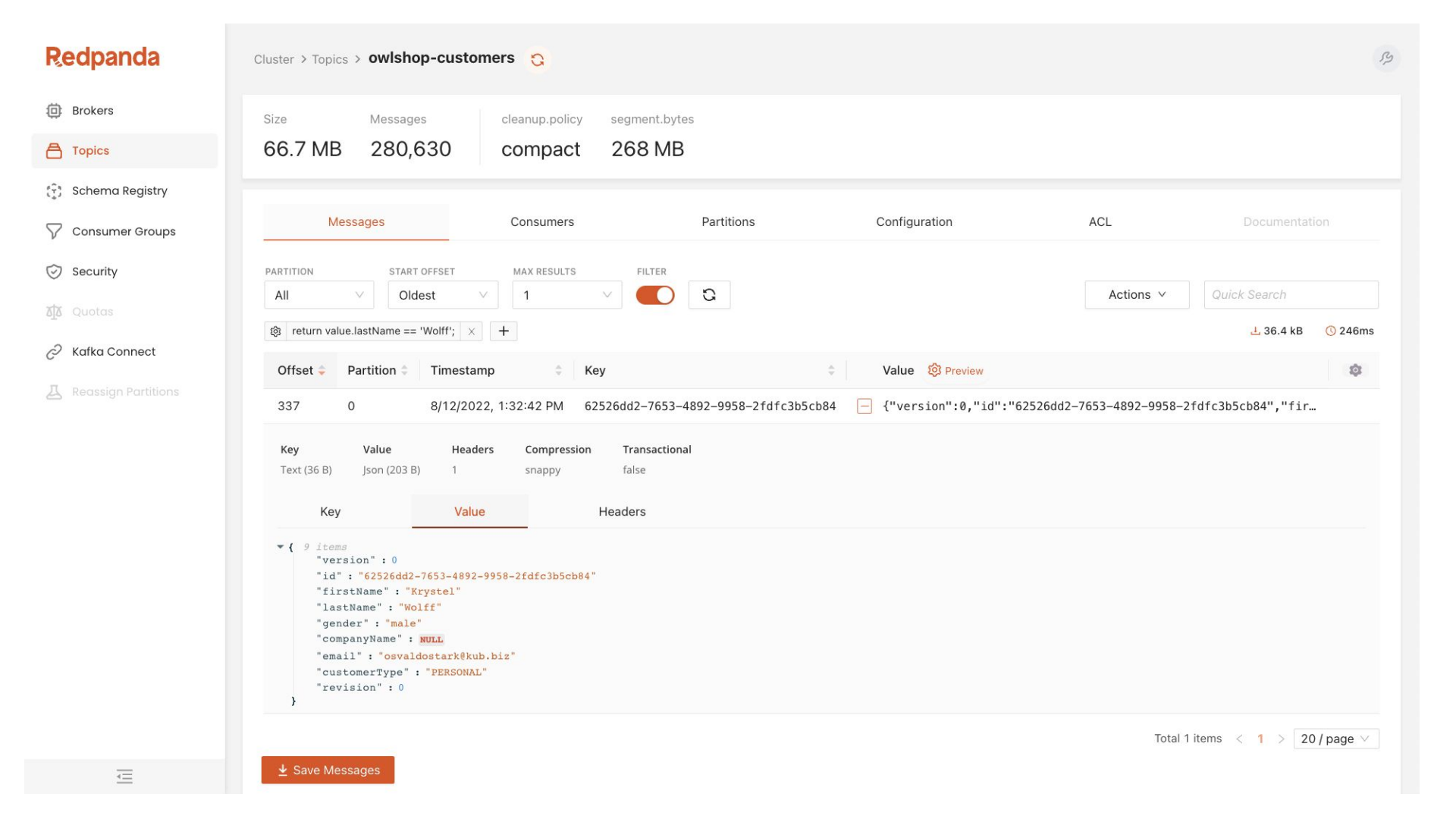
- Not a Kafka fork!
- Kafka rewritten in C++
- Identical read/write interfaces as Kafka
- Designed for modern hardware

Simple to deploy, use and manage

- Single binary
- Kafka-compatible APIs
- Easy Day 2 Ops
- Dev-friendly interface



© 2023 REDPANDA DATA


Redpanda
Brokers
Topics
Schema Registry
Consumer Groups
Security
Quotas
Kafka Connect
Reassign Partitions
Cluster > Topics > owlshop-customers
Size
66.7 MB
Messages
280,630
cleanup.policy
compact
segment.bytes
268 MB
Messages
Consumers
Partitions
Configuration
ACL
Documentation
PARTITION
All
START OFFSET
Oldest
MAX RESULTS
1
FILTER
Actions
Quick Search
return value.lastName == 'Wolff';
36.4 kB
246ms
Offset
Partition
Timestamp
Key
Value
Preview
337
0
8/12/2022, 1:32:42 PM
62526dd2-7653-4892-9958-2fdfc3b5cb84
{"version":0,"id":"62526dd2-7653-4892-9958-2fdfc3b5cb84","fir…
Key
Text (36 B)
Value
Json (203 B)
Headers
1
Compression
snappy
Transactional
false
Key
Value
Headers
{ 9 items
"version" : 0
"id" : "62526dd2-7653-4892-9958-2fdfc3b5cb84"
"firstName" : "Krystel"
"lastName" : "Wolff"
"gender" : "male"
"companyName" : NULL
"email" : "osvaldostark@kub.biz"
"customerType" : "PERSONAL"
"revision" : 0
}
Total 1 items
1
20 / page
Save Messages

rpk

Redpanda's command line interface (CLI) utility.

Check health of cluster

```
rpk cluster health
```

Create a topic

```
rpk topic create my-topic -p 5
```

List topics

```
rpk topic list
```

Describe a topic

```
rpk topic describe
```

Redpanda Demo

What is ClickHouse?

What is ClickHouse?

Open Source

- Developed since 2009
- OSS 2016
- 34k+ Github stars
- 1k+ contributors
- 500+ releases

Column-Oriented

- Files per column
- Vectorized query execution
- Optimised for aggregations
- Sorting and indexing
- Background merges

Distributed

- Replication
- Sharding
- Multi-master
- Cross-region

OLAP Database

- Analytics use cases
- Aggregations
- Visualizations
- Mostly immutable data

Row Oriented vs Column Oriented

Row Oriented

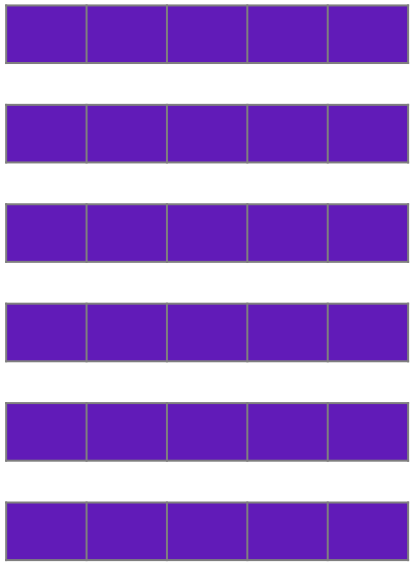
location	ts	temperature	wind_speed	humidity
Aberystwyth	2022-01-01 00:00:00	14	21	79
Blackpool	2022-01-01 00:20:00	13	9	82

Column Oriented

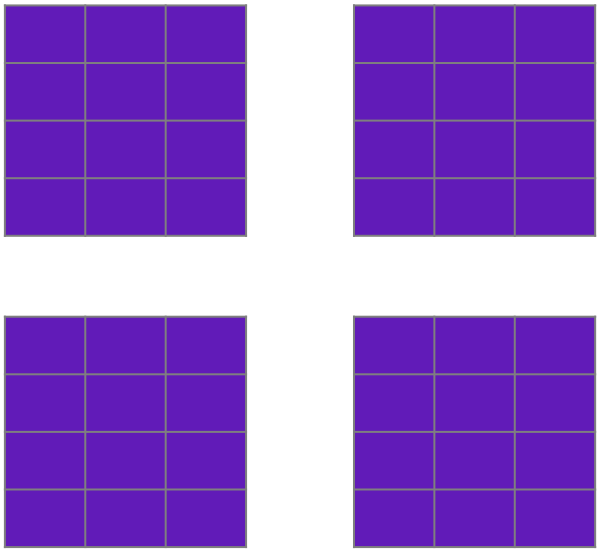
location	ts	temperature	wind_speed	humidity
Aberystwyth	2022-01-01 00:00:00	14	21	79
Blackpool	2022-01-01 00:20:00	13	9	82

Vectorised Query Execution

Process rows sequentially



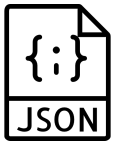
Process chunks of values




Parquet


protobuf
Protocol Buffers


APACHE
ARROW


{:}
JSON


AVRO


CSV


NumPy


ClickHouse


RabbitMQ


kafka


RedPanda


mongoDB.


MySQL


redis


PostgreSQL


ICEBERG


Apache
hudi


DELTA LAKE


http://

Amazon S3


Microsoft Azure
Blob Storage
10
01

Flavours of ClickHouse

ClickHouse Demo

What is Streamlit?

What is Streamlit?

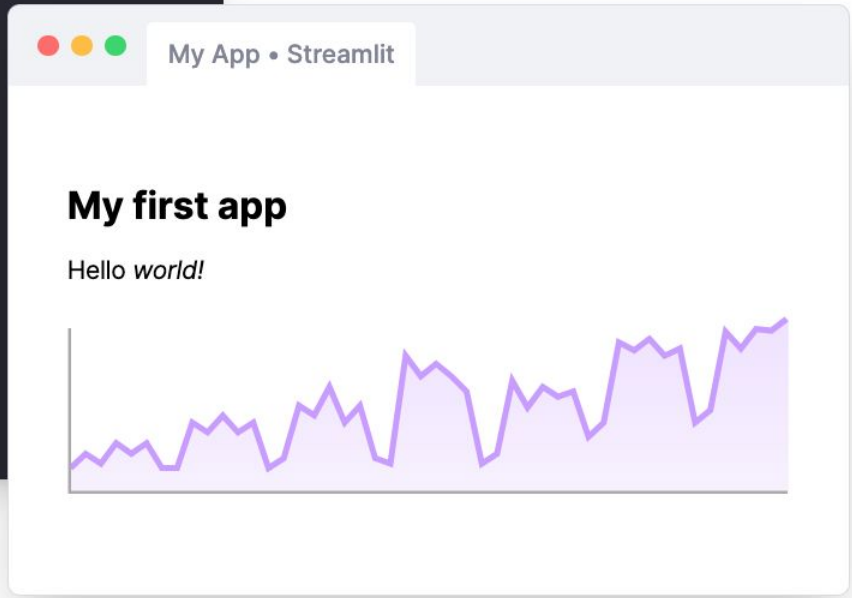
Streamlit turns data scripts into shareable web apps in minutes.
All in pure Python. No front-end experience required.

Streamlit

Streamlit Hello World

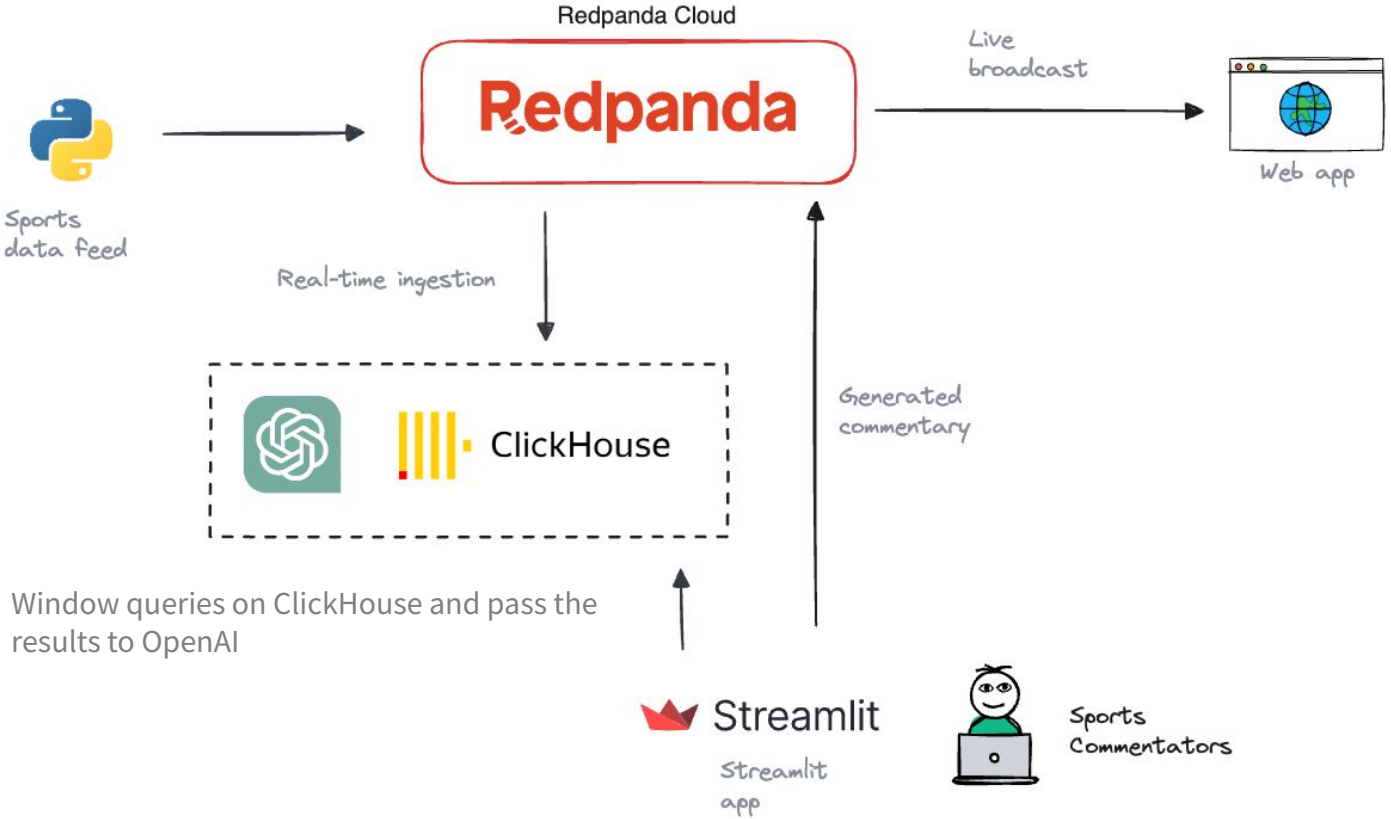
MyApp.py

```
import streamlit as st
import pandas as pd

st.write("""
# My first app
Hello *world!*
""")

df = pd.read_csv("my_data.csv")
st.line_chart(df)
```



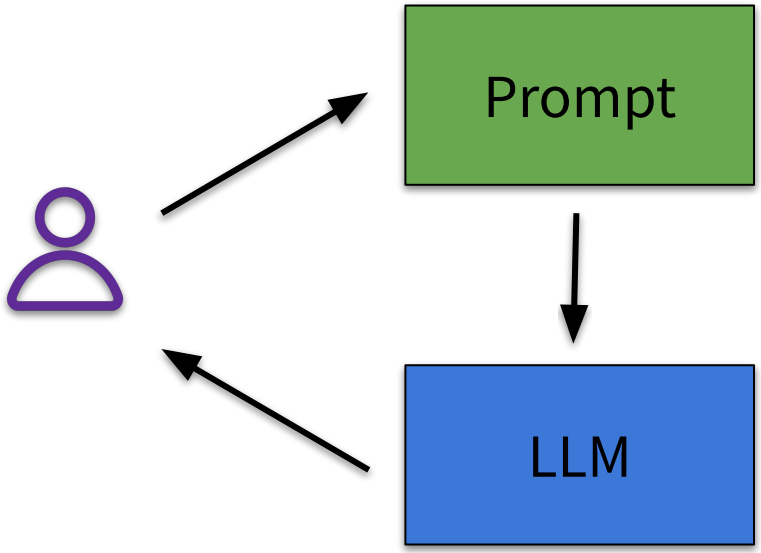
Building the AI Copilot

The flow of events

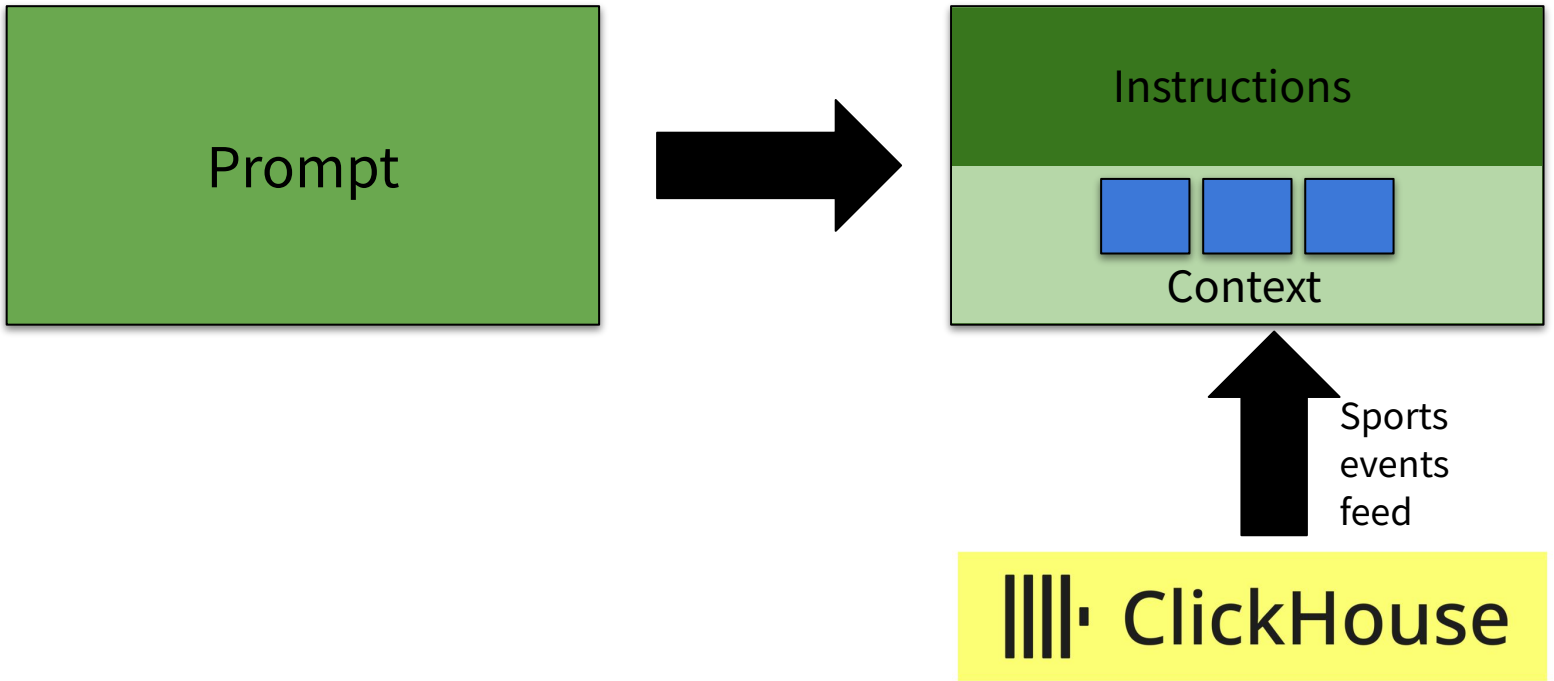


Generated Commentary Demo

How do Large Language Models work?



Ingesting context into the prompt



Sports events feed


```
{
  "match_id": "1402",
  "id": "match1402_set1_game_1_point_1",
  "time": "0:00:00",
  "player1": "A. Zverev",
  "player2": "J. Sinner",
  "previous_sets": [],
  "server": "J. Sinner",
  "set": "1",
  "game": "1",
  "set_score": "0 - 0",
  "point_score": "15 - 0",
  "description": "J. Sinner wins the point with an ace",
  "event_type": "Men's Singles",
  "event_round": "R4",
  "potential_publish_time": "2024-04-19T12:03:46.553367",
  "published": false,
  "publish_time": "2024-04-19T12:03:46.554138"
}
```

LLM Code

```python
from openai import OpenAI

client = OpenAI()

system_message = """
You're the assistant or copilot to a person who's writing live-text updates for a tennis match.
You'll receive a list of events describing what's happened and you need to come up with a one or two-sentence summary based on that data.
Pull out things that you think are interesting including the length of the game if it's very short or long, shots on important points, and so on.
Write in the present tense and in a way that's accessible to your average tennis fan.
"""

def call_llm(events, model="gpt-4", stream=True):
    response = client.chat.completions.create(
        model=model,
        messages=[
            {
            "role": "system",
            "content": system_message
            },
            {
            "role": "user",
            "content": "\n".join([str(item) for item in events])
            }
        ],
        stream=stream
    )
    return response
```

Pulling events from ClickHouse

```
import clickhouse_connect
from generate_commentary import call_llm

client = clickhouse_connect.get_client(host='localhost')

query_response = client.query_df("<query>")
latest_events = [row.to_dict() for idx, row in query_response.iterrows()]
response = call_llm(latest_events, stream=False, model="gpt-3.5-turbo")
```

Generated text events


```
{
  "match_id": "1316",
  "score": "L. Djere 6 - 4, 6 - 4, 1 - 6, 1 - 6, 3 - 6 N. Djokovic",
  "p1": "L. Djere",
  "p2": "N. Djokovic",
  "title": "Djokovic comeback complete",
  "message": "In a nail-biting face-off, N. Djokovic takes the 5th set 6 - 3 against L. Djere after a series of unforced errors from both players. Djokovic advances to the next round after three hours and forty-five minutes of rigorous gameplay, with the final point won on a Djere backhand error. It's a remarkable display of endurance and skill, proving again why Djokovic is one of the best in the game today.",
  "datetime": "2024-04-22T12:00:05.290848"
}
```

Retrieval queries: Recent points

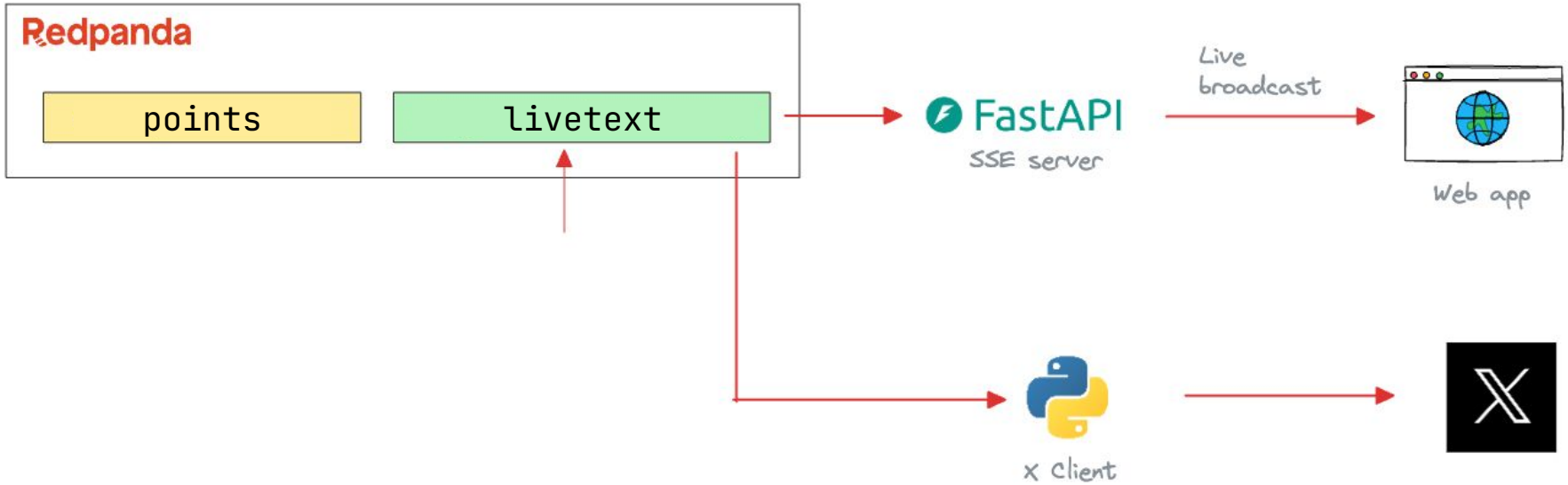
```
SELECT points.* EXCEPT (match_id, id, publish_time)
FROM points
WHERE match_id = $matchId AND (now() - 60) < publish_time
ORDER BY id
```

Retrieval queries: Last completed game

```
SELECT points.* EXCEPT (match_id, id, publish_time)
FROM points
INNER JOIN
(
    SELECT *
    FROM points
    WHERE (point_score = 'FINISH') AND (match_id = $matchId)
    ORDER BY id DESC
    LIMIT 1
) AS latestPoint ON (latestPoint.set = points.set) AND (latestPoint.game = points.game)
WHERE match_id = $matchId
ORDER BY id
```

Serving the live commentary

Multiple consumers on the Redpanda topic



Live Commentary Demo

Future Ideas

How can we extend this work?

- Automatic summaries every <x> seconds
- A Copilot that has access to fine-grained statistics
- Text to SQL so that the writer can ask questions of the data
- Can we use more batch data?
- Store the generated commentary for later analysis/tweaking of the prompt
- Compare the generated commentary with what the analyst chooses to publish

Is it only for sports?

We'll be focusing on sports, but you could use it for any of the following problems:

- Live auctions
- Weather updates
- Local traffic reporting
- Current location of food delivery
- ...

Any use case where you have events that you want to summarise into a more readable format.

Thanks and Questions

github.com/mneedham/devoxx-ai-sports-commentary

www.linkedin.com/in/dunithd
dunith.medium.com

www.linkedin.com/in/markhneedham
youtube.com/@LearnDataWithMark

SCAN ME