DronaHQ for building apps on top of InfluxDB 3.0

Your Hosts

Anais Dotis-Georgiou

Developer Advocate, InfluxData

Shibam Dhar

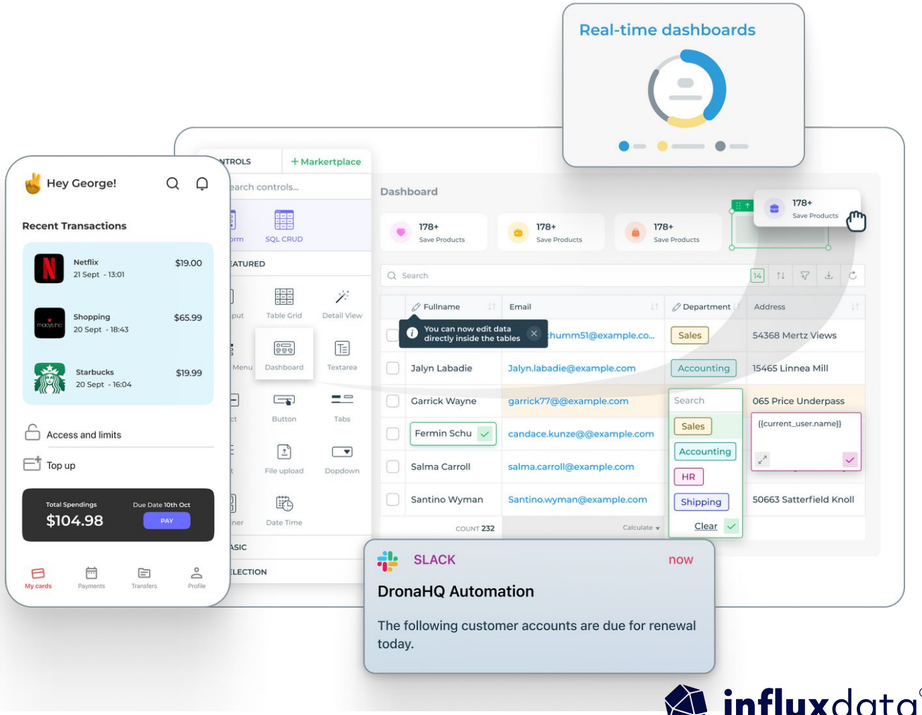
Developer Advocate, DronaHQ

DronaHQ | influxdata®

Agenda

1. Introductions
2. Demo
3. Q&A

influxdb™



influxdata®

Why do I need a Time Series Database?

The age of instrumentation

Sensors
in the physical world
(e.g. IoT)

Instrumentation
of the virtual world
(e.g. DevOps)

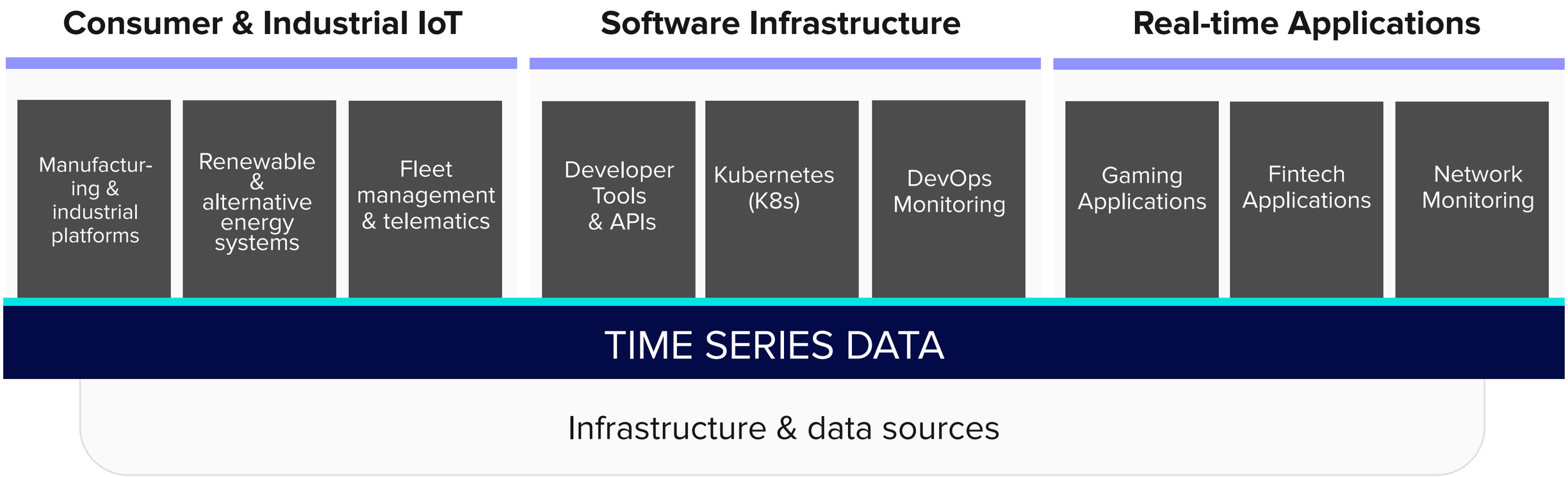
influxdata®

Characteristics of the data

- Time-stamped
- Generated in regular (metric) and irregular (event) time periods
- Huge volumes
- Real time and time sensitive
- Example: traces & logs

influxdata®

Time series in every application



influxdata®

Rise of time series as a category

RELATIONAL

- Orders
- Customers
- Records

PostgreSQL

DOCUMENT

- High throughput
- Large document

MongoDB

SEARCH

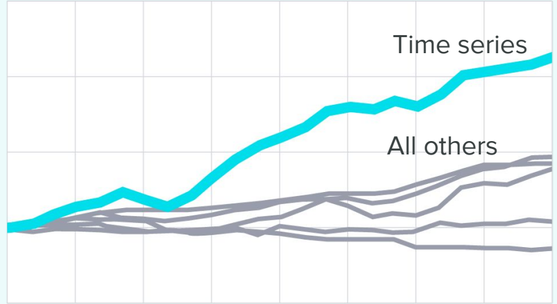
- Distributed search
- Logs
- Geo

elastic

TIME SERIES

- Events, metrics, time stamped
- for IoT, analytics, cloud native

Time series is fastest growing data category by far

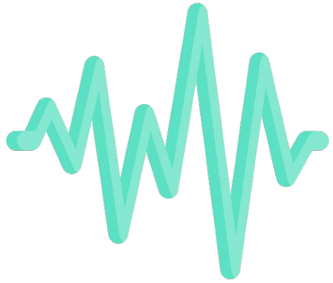


influxdata®

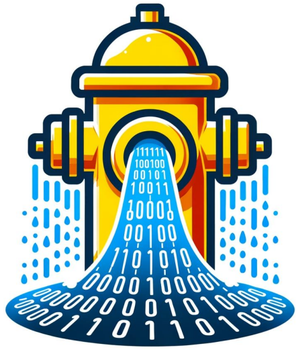
source: DB Engines

influxdata®

The characteristics of a TSDB

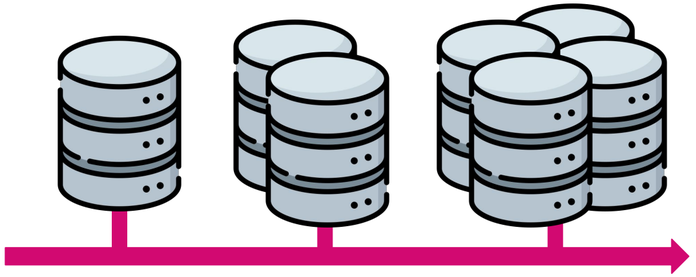


Time Series Data



High write throughput

Efficient Queries Over Time Ranges



Scalability and Performance

influxdata®

Large and Growing Customer Base

1900+

Customers

IOT

ThermoFisher SCIENTIFIC
Schneider Electric
TESLA
ptc
Lam RESEARCH
nest
equinor

OTHER

SAP
CISCO
salesforce
IBM
coupa
BNP PARIBAS
Expedia
Bethesda
Capital One
COMCAST
DISCOVER
SIEMENS

influxdata

The InfluxDB Platform

InfluxDB is 3 things

1

POWERFUL

API & Toolset

for real-time apps

2

HIGH PERFORMANCE

Time Series Engine

for real-time data workloads

3

MASSIVE

Community & Ecosystem

of cloud & open source developers

influxdata®

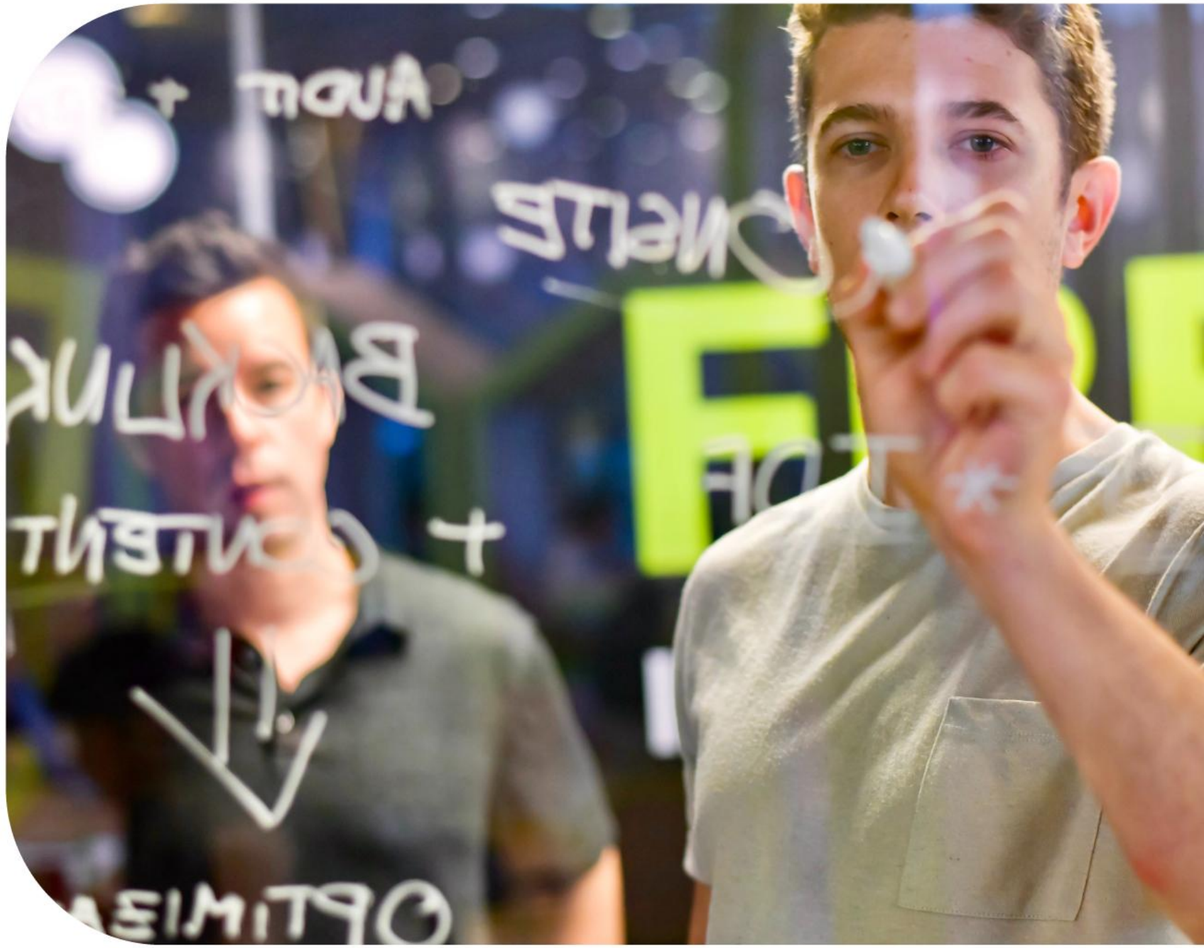
influxdb

Enables organizations to make cost–disruptive decisions on high volumes of time-sensitive data

InfluxDB – Time Series Platform

Empowers developers to build IoT, analytics, & monitoring software

- Designed for time series analysis
- Easy to share, easy to extend
- Open Source (MIT license)
- Easy to get started, powerful to scale



influxdata®

InfluxDB – Time Series Platform

Core focus: Developers and Builders

- Developer happiness
- Time to awesome
- Ease of scale-out & deployment

influxdata®

Reference Architecture

InfluxDB Platform

influxdb™ 3.0

Data Sources

All time-stamped data

Metrics
Sensors
Events
Devices
& more

Data Collection

Multiple data ingest methods

Telegraf
Client libraries
& more

Data Storage & Transformation

Purpose-built time series database

Collect
Store
SQL queries

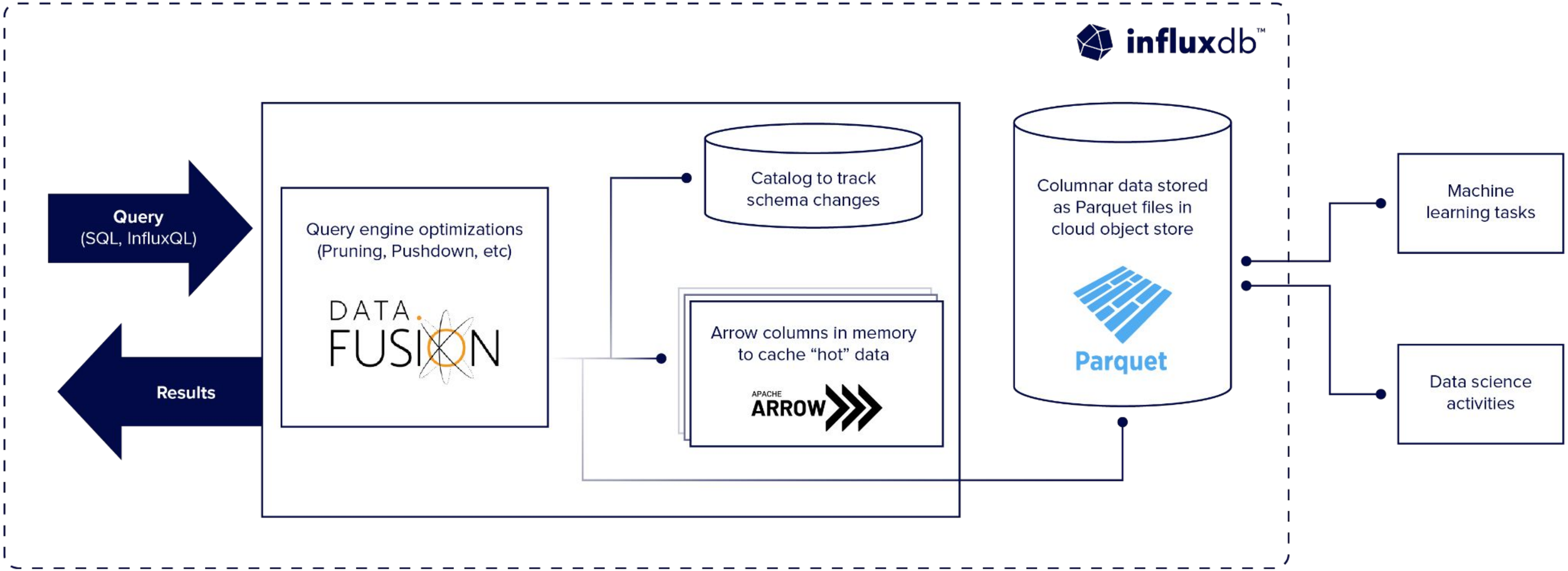
+ INTEGRATIONS

Data Visualization & Analysis

Machine learning
Analytics tools
BI tools
& more

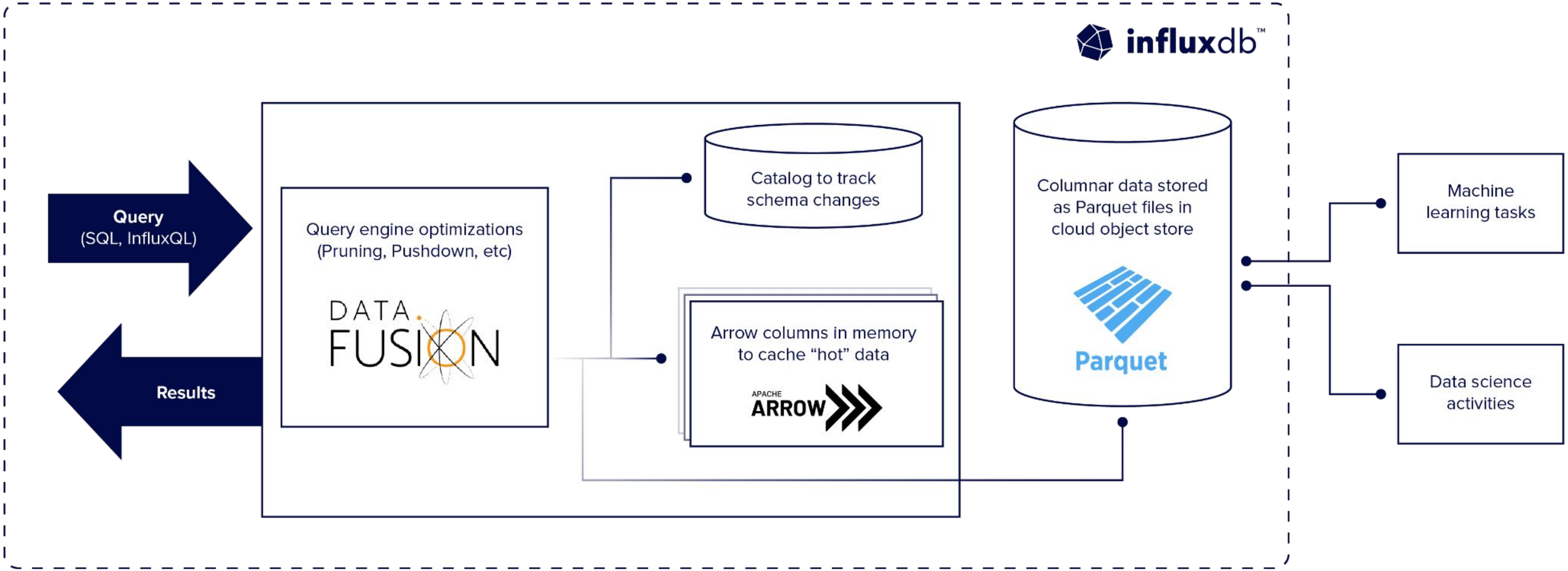
influxdata®

Deep dive Architecture



influxdata®

Deep dive Architecture



influxdata®

Using InfluxDB

Concepts: Data Model

Bucket

- All InfluxDB data is stored in a bucket. A bucket combines the concept of a d
  duration of time that each data point persists).

Measurement

- A name to a group of data at a high level (Table)

Tag set

- A set of key-value pairs to group data at a low level (values are strings)

Field set

- A set of key-value pairs to represent data (values are numerical & strings)

Timestamp

Line Protocol: Simple but powerful

- Writing points to InfluxDB uses Line Protocol, which takes the follow

```
<measurement>[,<tag-key>=<tag-value>] [<field-key>=<field-value>]
[unix-nano-timestamp]
```

Measurement	Tag Set	Field Set	Timestamp
cpu_load	,hostname=server02,us_west=az	temp=24.5,volts=7	1234567890000000

Reference: https://docs.influxdata.com/influxdb/cloud/reference/syntax/line-protocol/

Schema Considerations

Benefits of the InfluxDB 3.0 Schema

- Eliminates cardinality restraints

Schema Restrictions

- No duplicate columns
- 200 column limit

Design for performance

- Avoid Wide Schemas
- Avoid Sparse Schemas
- Homogeneous

Design for query simplicity

- Keep simple
- Avoid Special characters

Writing Data

```
from influxdb_client_3 import InfluxDBClient3, Point
import datetime
```

Library Import

```
host = “eu-central-1-1.aws.cloud2.influxdata.com”
org="6a841c0c08328fb1"
token = “”
database = “database”

client = InfluxDBClient3(
    token=token,
    host=host,
    org=org)
```

Initialization

```
data = Point().tag().field().field().time()
client.write(data)
```

Write

Querying Data

```
from influxdb_client_3 import InfluxDBClient3, Point
import datetime
```

Library Import

```
host = “eu-central-1-1.aws.cloud2.influxdata.com”
org="6a841c0c08328fb1"
token = “”
database = “database”

client = InfluxDBClient3(
    token=token,
    host=host,
    org=org)
```

Initialization

```
data = Point().tag().field().field().time()
client.write(data)
```

Write

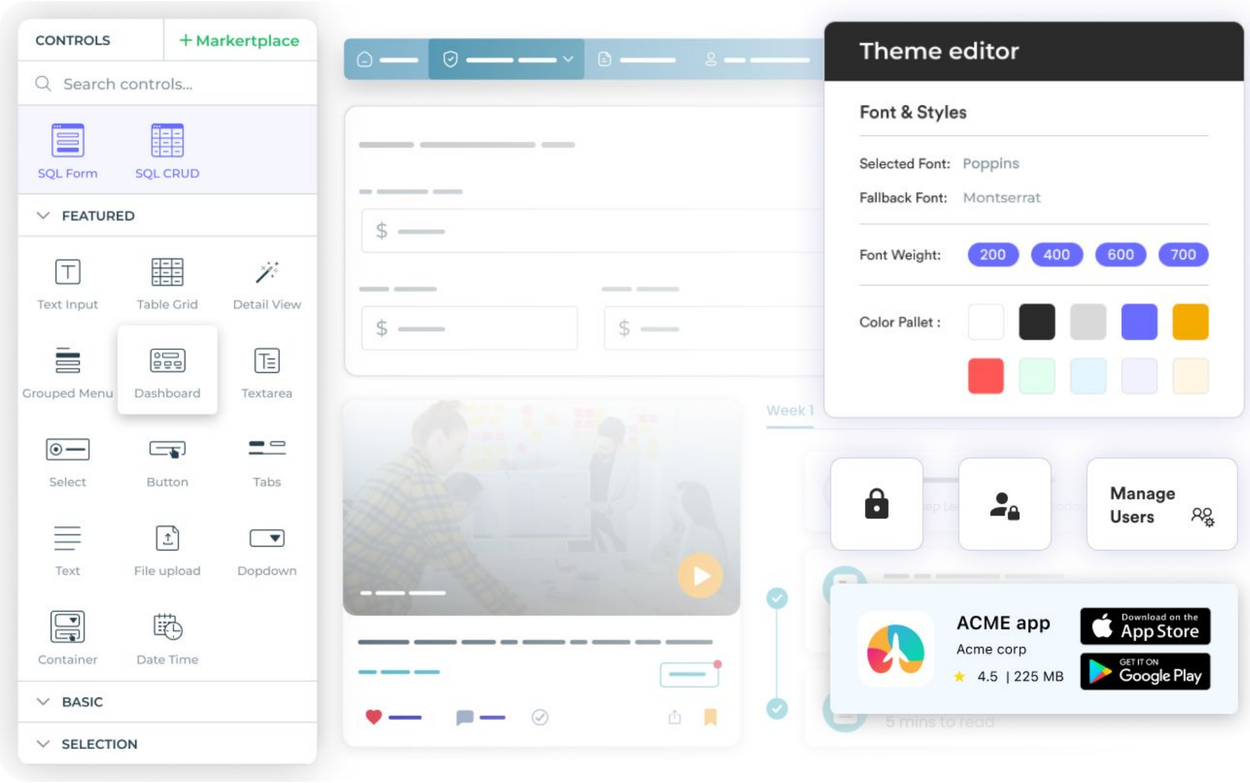
```
sql = '''SELECT * FROM table'''
table = client.query(query=sql, language='sql', mode='all')
print(table)
```

Query

DronaHQ | influxdata®

About DronaHQ

DronaHQ is a low-code developer toolset, to build internal tools, custom portals, AI-powered apps, and custom visualizations & dashboards at 10X speed.



influxdata®

DronaHQ | influxdata®

- 16+ years of Enterprise Product
- Million+ users supported
- 100+ enterprise customers
- Global audience

> Where we've really enjoyed success with DronaHQ is that we can turn things around, now, **from an idea to a real thing in five weeks**, which for us is wonderful.

Andrew Scott
Global Solutions Owner
Mondelēz International

Gartner 4.2/5.0

Capterra 4.6/5.0

G2 CROWD 4.4/5.0

Trusted by

Mondelēz International, Nestlé, wipro, LTI Let's Solve, Pfizer, Cipla, IDFC FIRST Bank, purplle, McKinsey & Company, ICICI PRUDENTIAL, SANY, BAJAJ FINSERV, dolly, Boston Scientific

 |

influxdata®

DronaHQ | influxdata®

How does it help?

Visual development

Ready UI controls, action blocks

Ready data connectors

50+ APIs + Databases connectors

Customization & flexibility

Flexible to add your custom code, JS,

Deploy across devices

Web, mobile, URL, embed, public apps

Iterate & Maintain

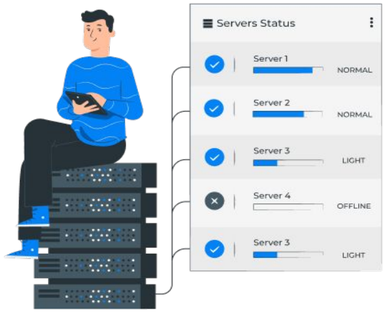
Easy to maintain and upgrade

influxdata®

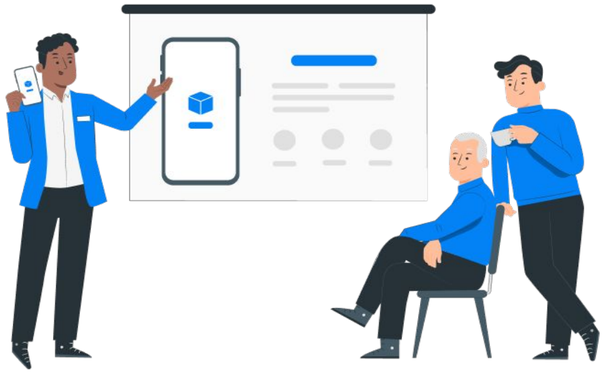
DronaHQ | influxdata®

Who can use DronaHQ?

Developers



Database Engineers

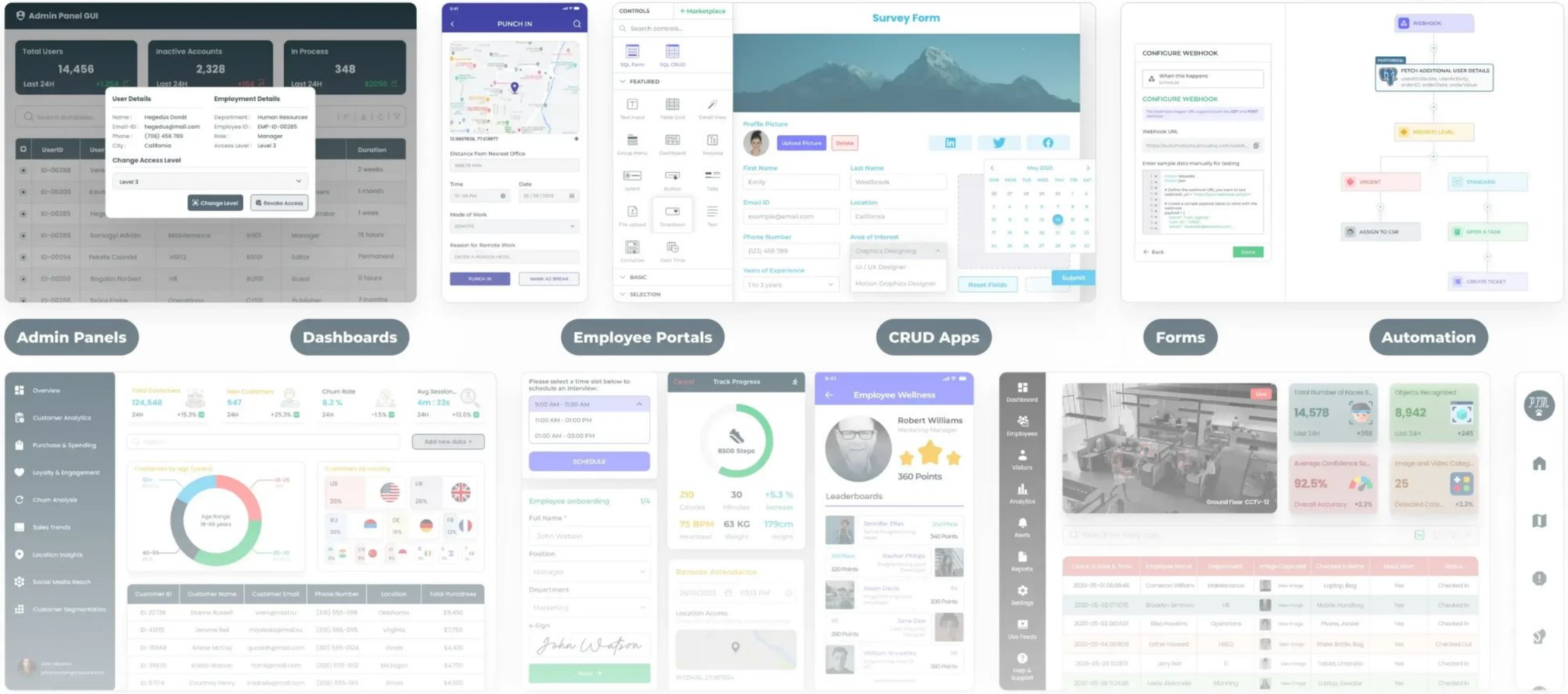


Product Managers

influxdata®

DronaHQ | influxdata®

What can you build?



influxdata®

DronaHQ | influxdata®

Code vs Connector

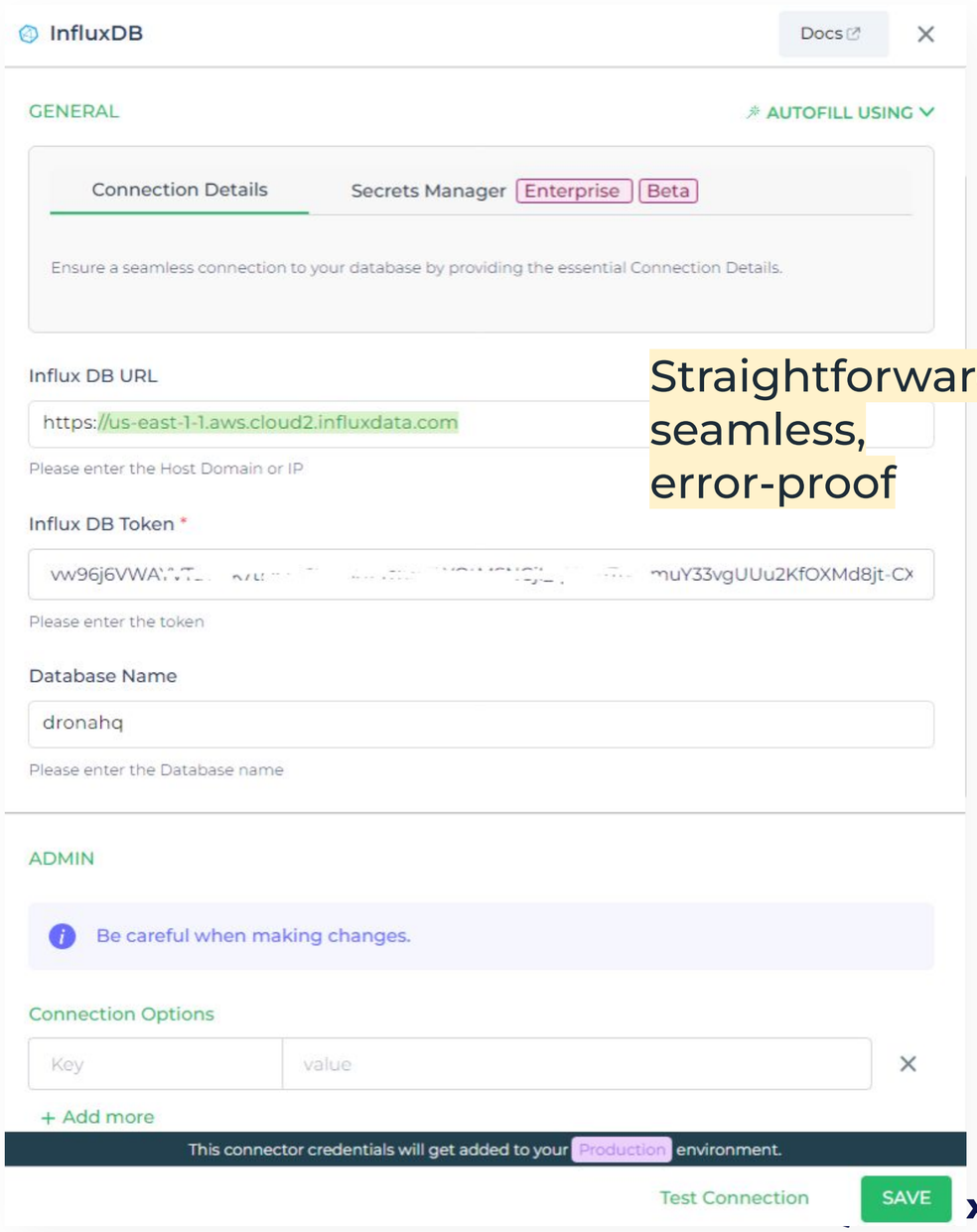
How does it help?

```python
import requests

def fetch_data_from_api(api_url):
    try:
        response = requests.get(api_url)
        if response.status_code == 200:
            return response.json()
        else:
            print(f"Error: {response.status_code} - {response.reason}")
            return None
    except requests.exceptions.RequestException as e:
        print(f"Request Error: {e}")
        return None

# Example usage
api_url = "https://api.example.com/data"
data = fetch_data_from_api(api_url)
if data:
    print("Received data from API:", data)
else:
    print("Failed to fetch data from API.")
```

Write custom codes for the APIs and connectors



Straightforward, seamless, error-proof

xdata®

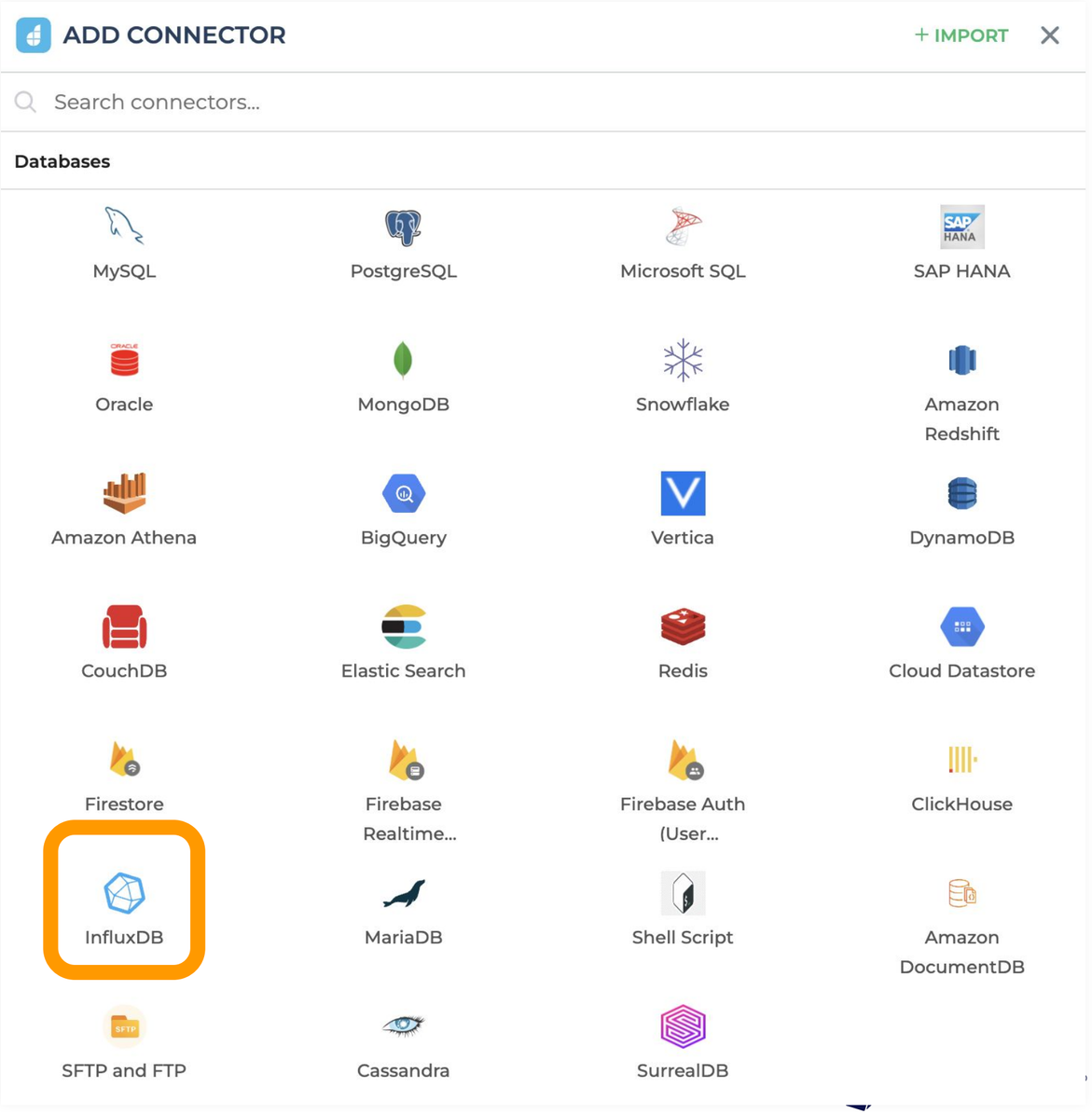
DronaHQ | influxdata®

What we will demonstrate

1. Integration

2. Frontend

3. Data Binding



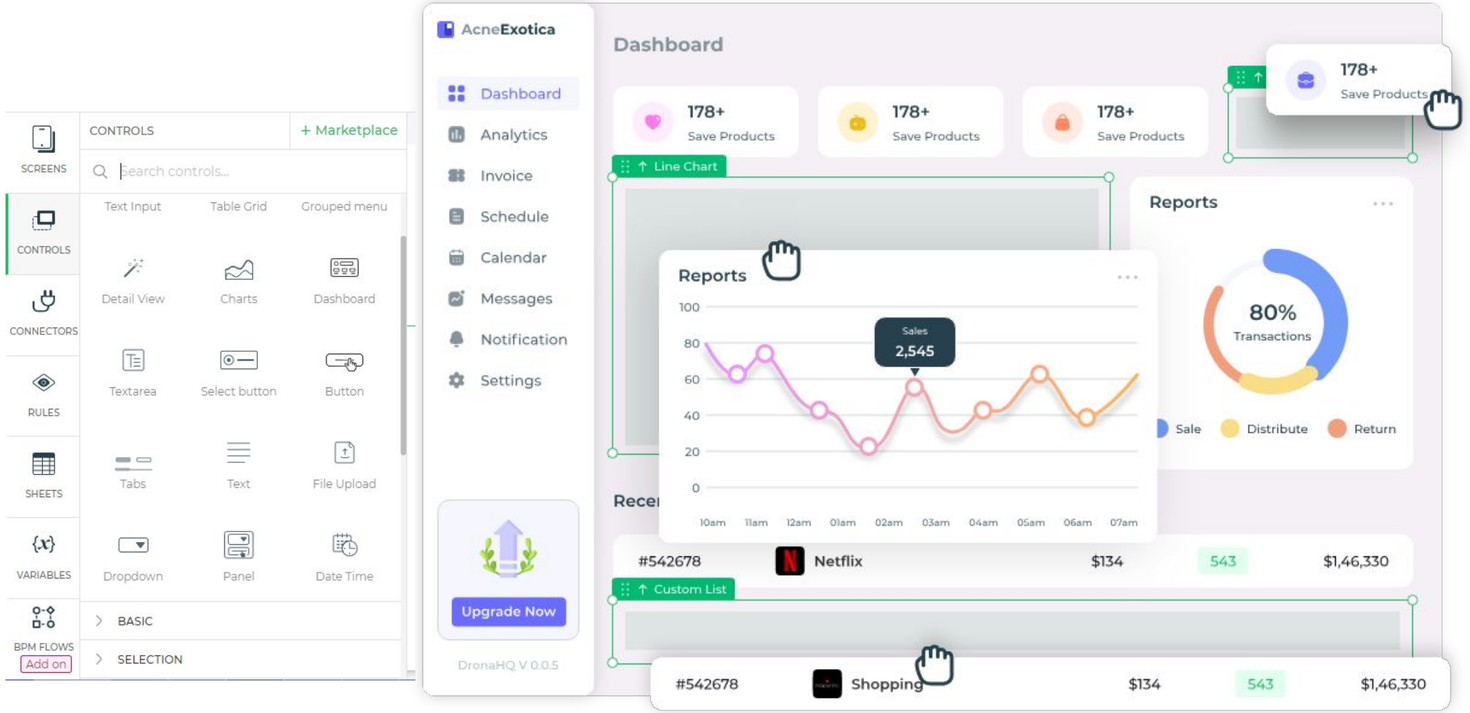
DronaHQ | influxdata®

What we will demonstrate

1. Integration

2. Frontend

3. Data Binding



influxdata®

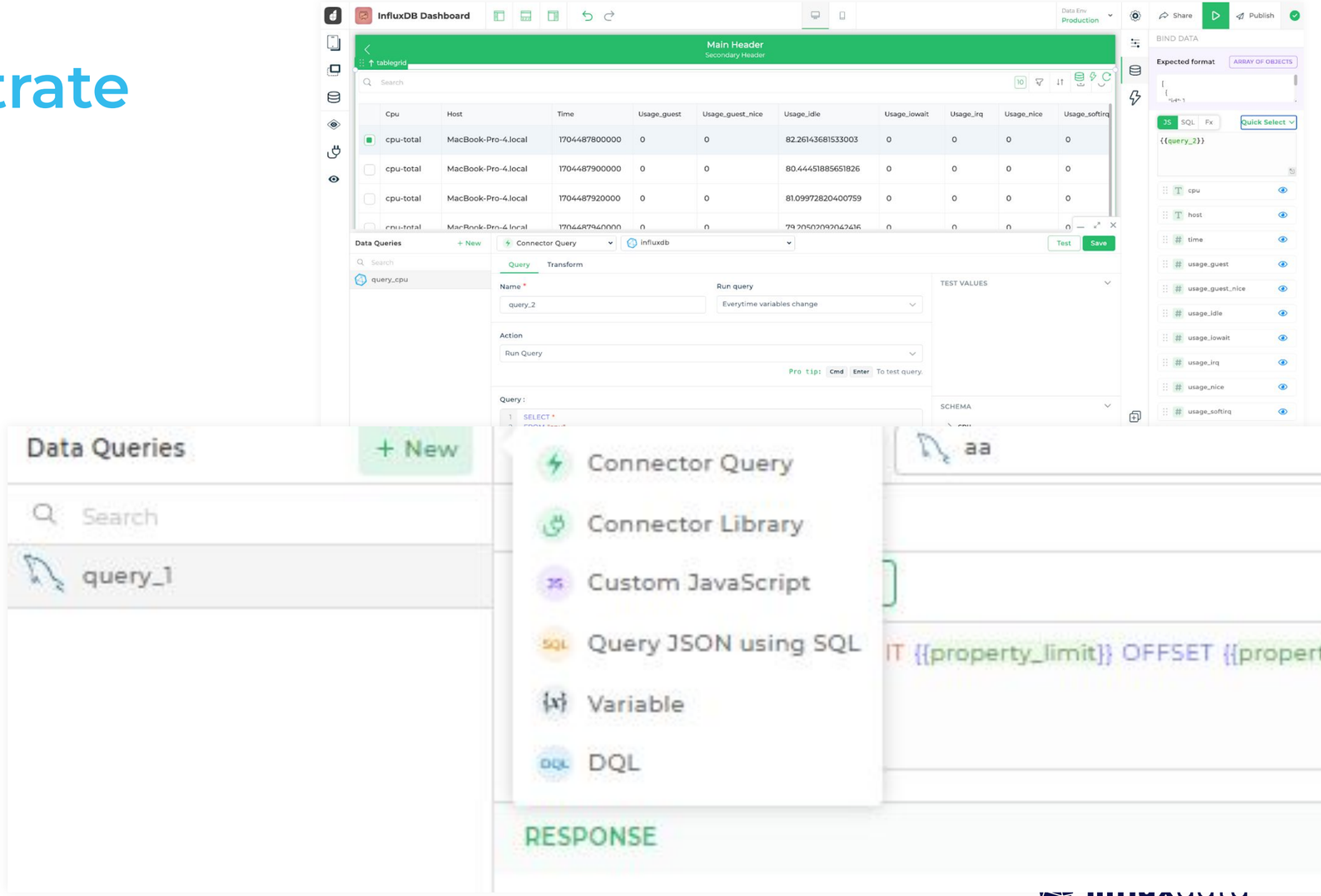
DronaHQ | influxdata®

What we will demonstrate

1. Integration

2. Frontend

3. Data Binding



On to the demo...

 |

More resources

1. Get started with 30-day free trial: https://www.dronahq.com/signup/
2. DronaHQ documentation: https://docs.dronahq.com/getting-started/introduction/