Koalas: pandas on Apache Spark

Niall Turbitt

Data Scientist @ Databricks

databricks

About

Niall Turbitt

Data Scientist @ Databricks

- Professional Services and Training

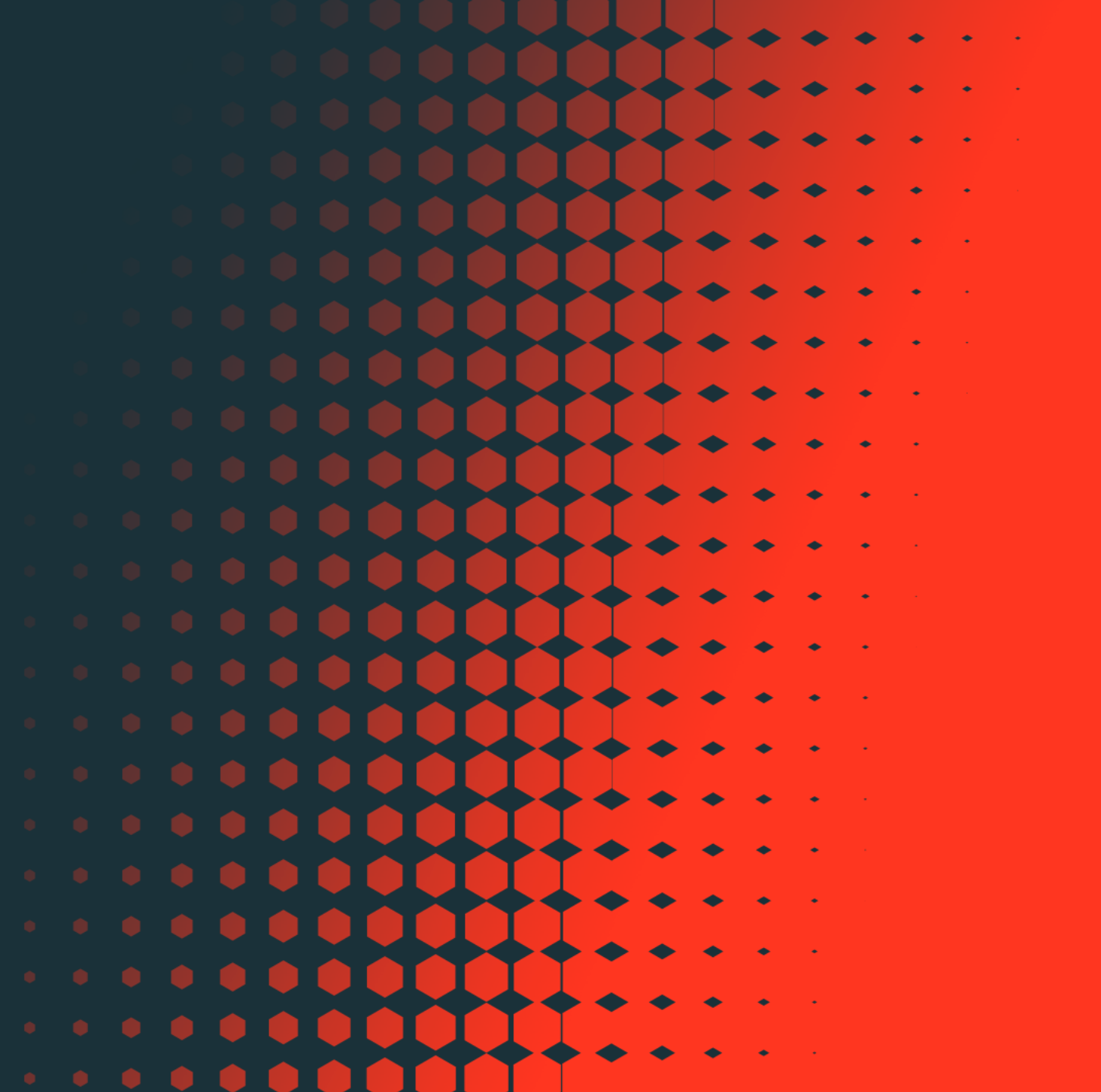
- Previously:
  - e-Commerce
  - Supply Chain and Logistics
  - Recommender Systems and Personalization

- MS Statistics University College Dublin
- BA Mathematics & Economics Trinity College Dublin

databricks

Outline

- pandas vs Spark
- Why Koalas?
- Koalas under the hood
- Demo
- Koalas Roadmap



Typical Journey of a Data Scientist

- Education (MOOCs, Books, Universities) -> pandas
- Analyze Small Datasets -> pandas
- Analyze Big Datasets -> Apache Spark

- Authored by Wes McKinney in 2008
- The standard tool for data manipulation and analysis in Python
- Deeply integrated into Python data science ecosystem
  - numpy
  - matplotlib
  - scikit-learn
- Can deal with a lot of different situations, including:
  - Basic statistical analysis
  - Handling missing data
  - Time series, categorical variables, strings

- De facto unified analytics engine for large-scale data processing
  - Streaming
  - ETL
  - ML
- Originally created at UC Berkeley by Databricks' founders
- PySpark API for Python; also API support for Scala, R, Java and SQL

A short example

pandas

```
import pandas as pd
df = pd.read_csv("my_data.csv")

df.columns = ['x', 'y', 'z1']

df['x2'] = df.x * df.x
```

PySpark

```
df = (spark.read
  .option("inferSchema", "true")
  .csv("my_data.csv"))

df = df.toDF('x', 'y', 'z1')

df = df.withColumn('x2', df.x * df.x)
```

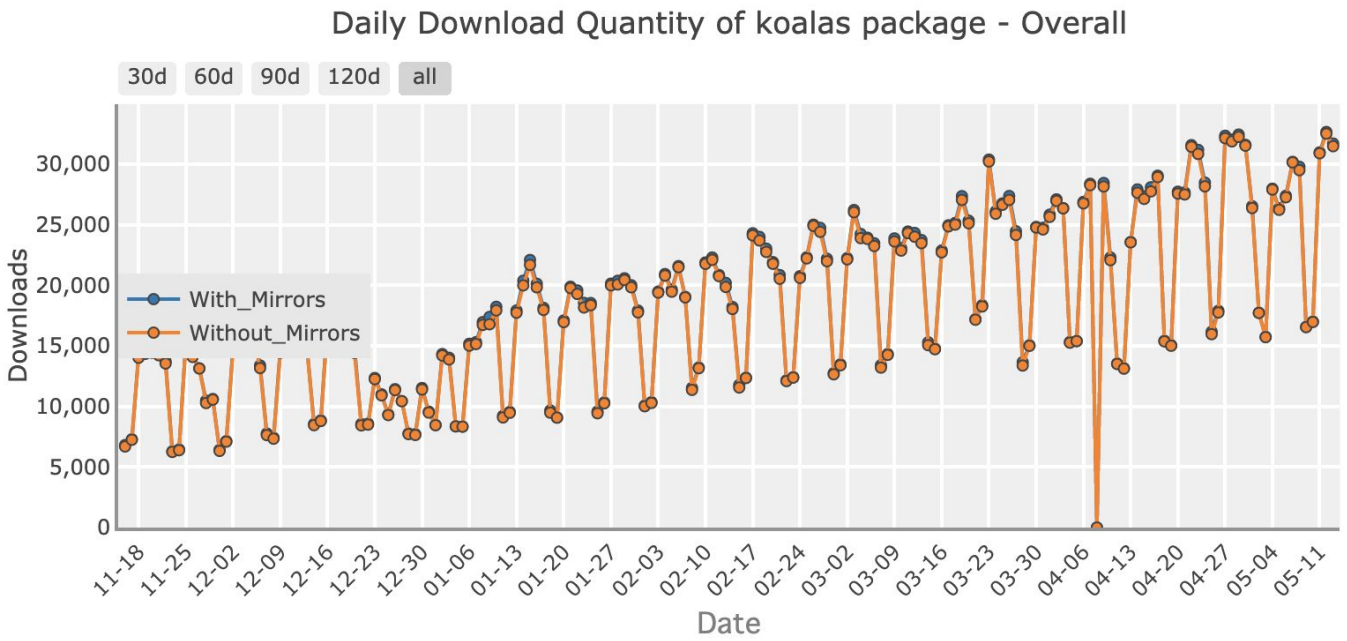
Koalas

- Announced April 2019
- Pure Python Library
- Aims at providing the pandas API on top of Apache Spark
  - Unifies the two ecosystems with a familiar API
  - Seamless transition between small and large data

Koalas Growth

- > 30,000 daily downloads
- > 2,000 GitHub stars

Downloads last day: 31,476
Downloads last week: 188,030
Downloads last month: 777,975



A short example

pandas

```
import pandas as pd
df = pd.read_csv("my_data.csv")

df.columns = ['x', 'y', 'z1']

df['x2'] = df.x * df.x
```

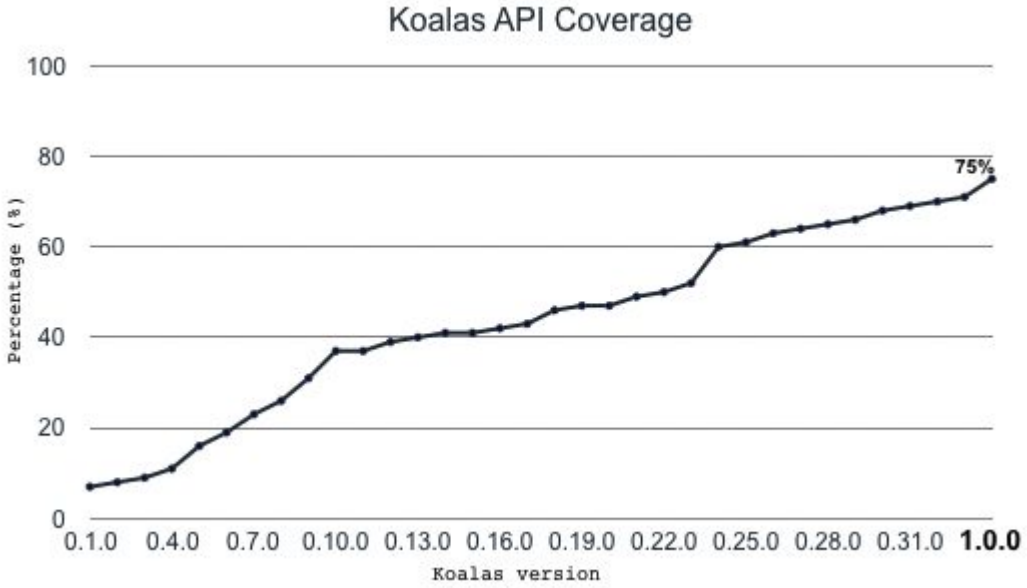
Koalas

```
import databricks.koalas as ks
df = ks.read_csv("my_data.csv")

df.columns = ['x', 'y', 'z1']

df['x2'] = df.x * df.x
```

Current Status

- Bi-weekly releases, very active community with daily changes

- Most common pandas functions have been implemented in Koalas:
  - Series : 70%
  - DataFrame : 74%
  - Index : 56%
  - MultiIndex : 51%
  - DataFrameGroupBy : 67%
  - SeriesGroupBy : 69%
  - Plotting: 80%



Spark vs pandas - Key Differences

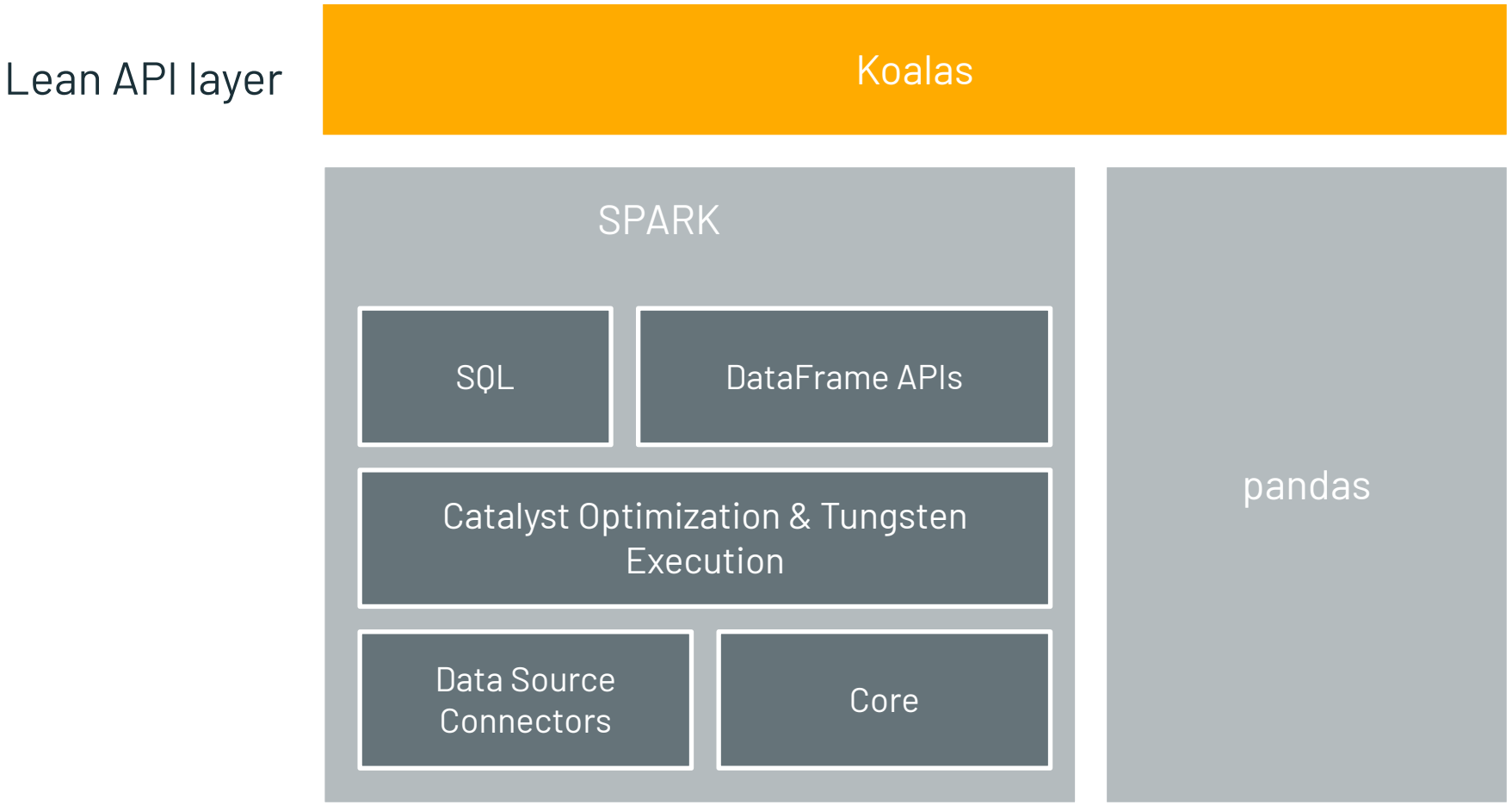
Spark

- DataFrame is immutable
- Lazy Evaluation
- Distributed
- Does not maintain row order
- Performance working at scale

pandas

- DataFrame is mutable
- Eager execution
- Single-machine
- Maintains row order
- Restricted to single machine

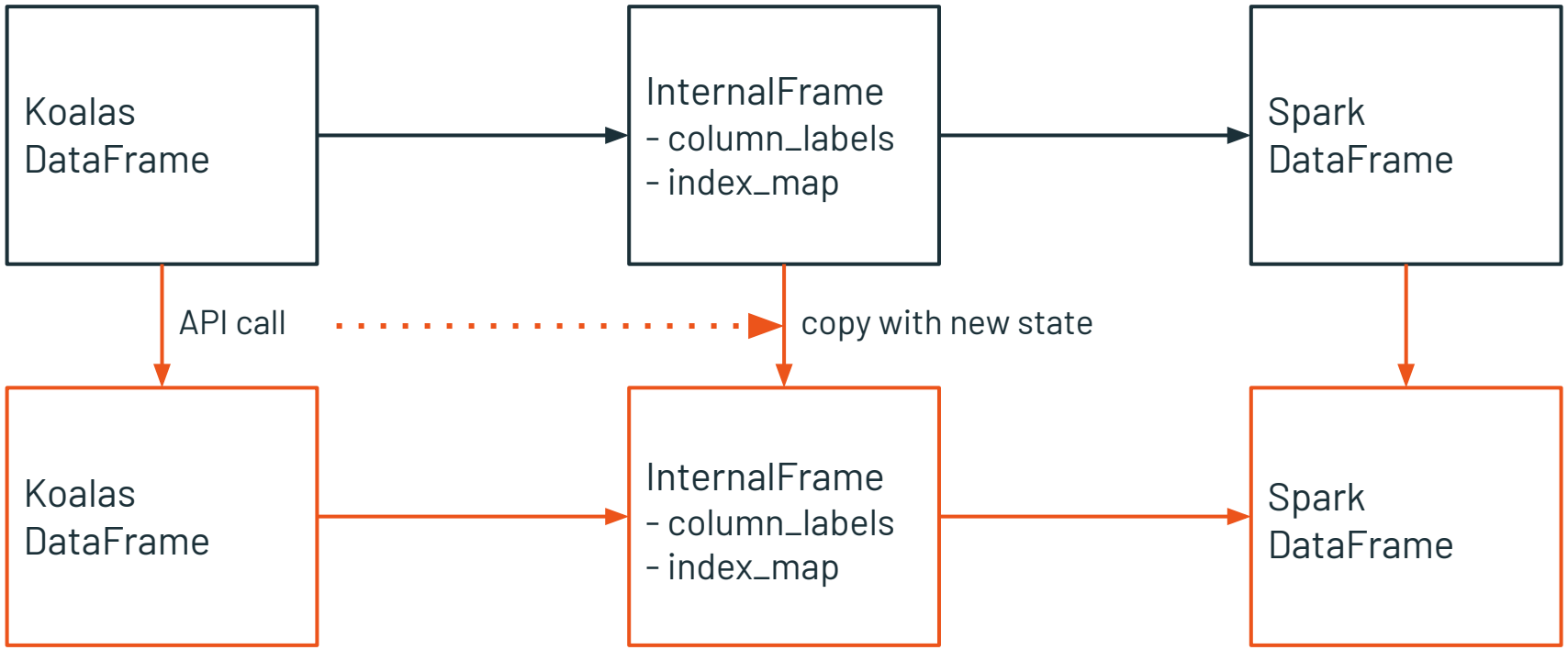
Koalas – Architecture



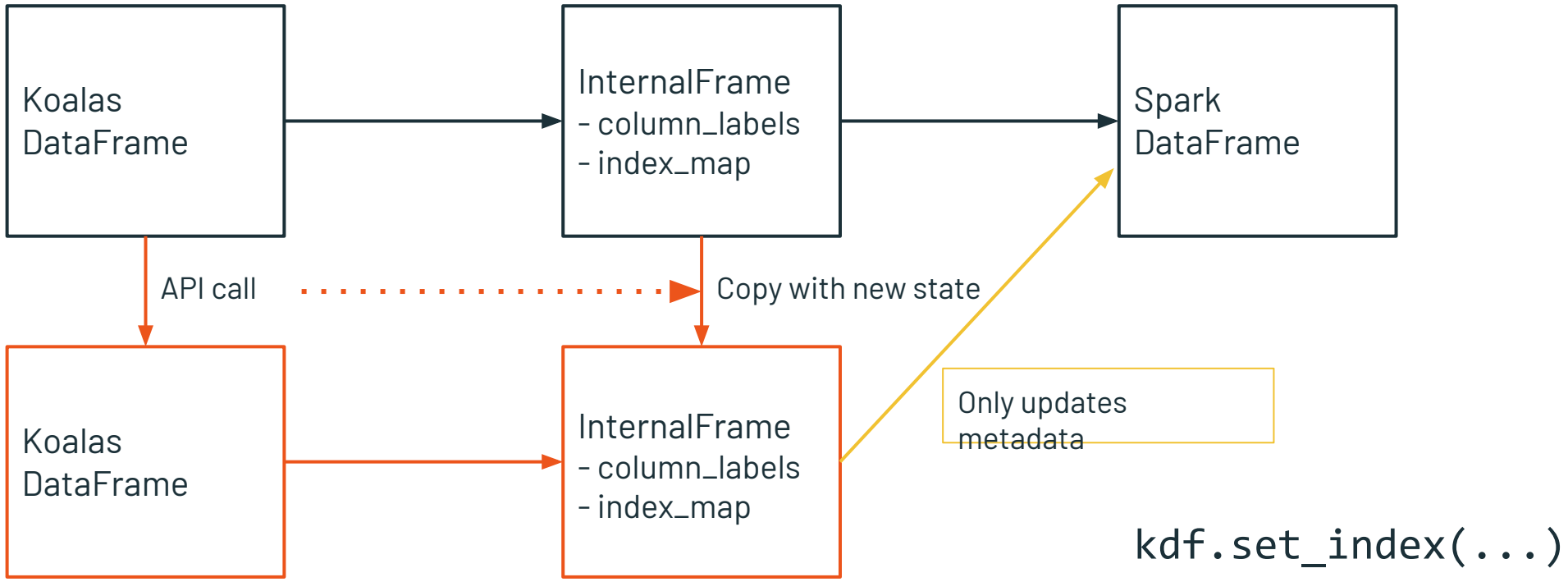
Koalas - Under the hood

- InternalFrame
  - Holds the current Spark DataFrame
  - Internal immutable metadata
  - Manages mappings from Koalas column names to Spark column names
  - Manages mapping from Koalas index names to Spark column names
  - Converts between Spark DataFrame and pandas DataFrame

InternalFrame

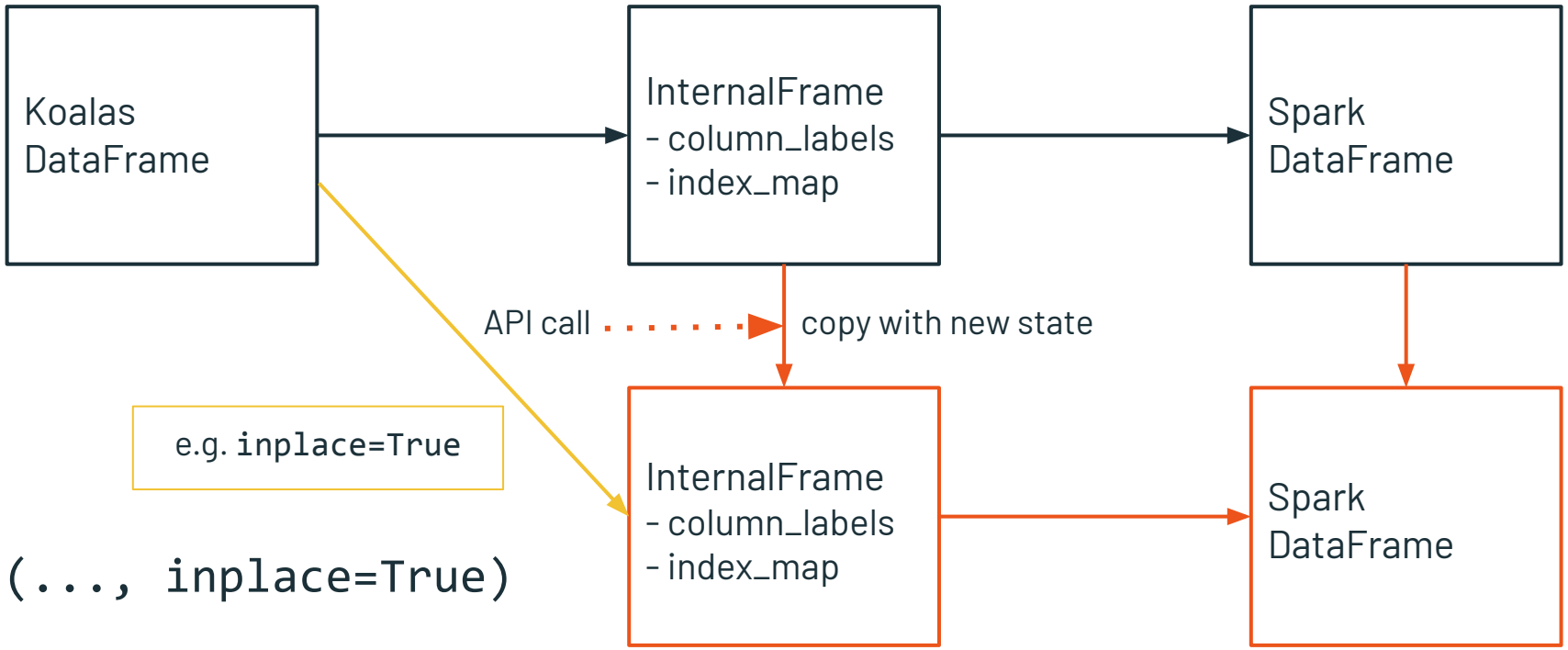


InternalFrame - Metadata update only




```
kdf.set_index(...)
```

InternalFrame




```
kdf.dropna(..., inplace=True)
```

Koalas - Under the hood

- Default Index
  - Koalas manages a group of columns as an index
  - Behaves in same manner as pandas
  - If no index is specified when creating Koalas DataFrame
    - A "default index" is attached automatically
  - Each "default index" has pros and cons

Default Index Type

`sequence`

- Used by default
- Implements sequence that increases one by one
- Uses PySpark Window function without specifying partition
- Can end up with whole partition on single node
- Avoid when data is large

`distributed-sequence`

- Implements sequence that increases one by one
- Uses group-by and group-map in distributed manner
- Recommended if the index *must be* sequential for a large dataset and increasing one by one

`distributed`

- Implements monotonically increasing sequence
- Uses PySpark's *monotonically_increasing_id* in distributed manner
- Values are non-deterministic
- Recommended if the index *does not* have to be a sequence increasing one by one

Configurable by the option `compute.default_index_type`

github.com/niall-turbitt/koalas_demos

Koalas – On the roadmap

- June 19th: Koalas 1.0 released!
  - Supports Spark 3.0

- July/Aug 2020: Release DBR/MLR 7.1 will pre-install Koalas 1.x

Koalas – Getting started

```
pip install koalas

conda install koalas
```

- Docs: https://koalas.readthedocs.io
- Github: https://github.com/databricks/koalas

Resources

- 10 Minutes from pandas to Koalas on Apache Spark
- Koalas: Easy Transition from pandas to Apache Spark
- Reducing Time-To-Insight for Virgin Hyperloop's Data

Thank You!