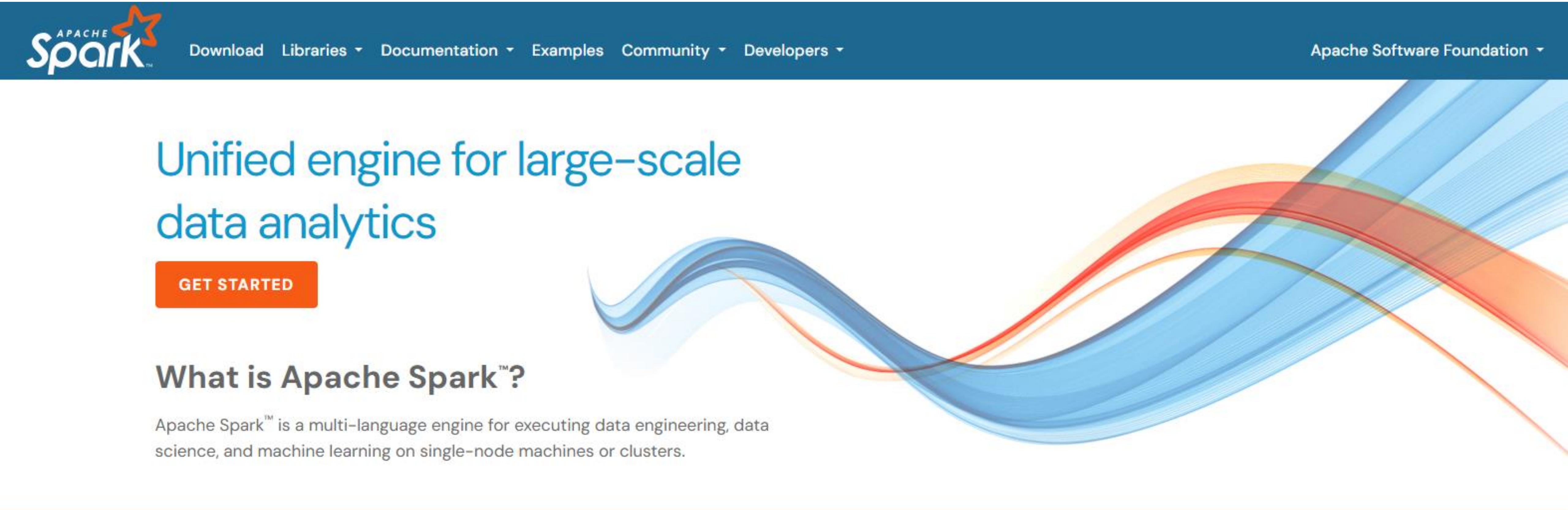
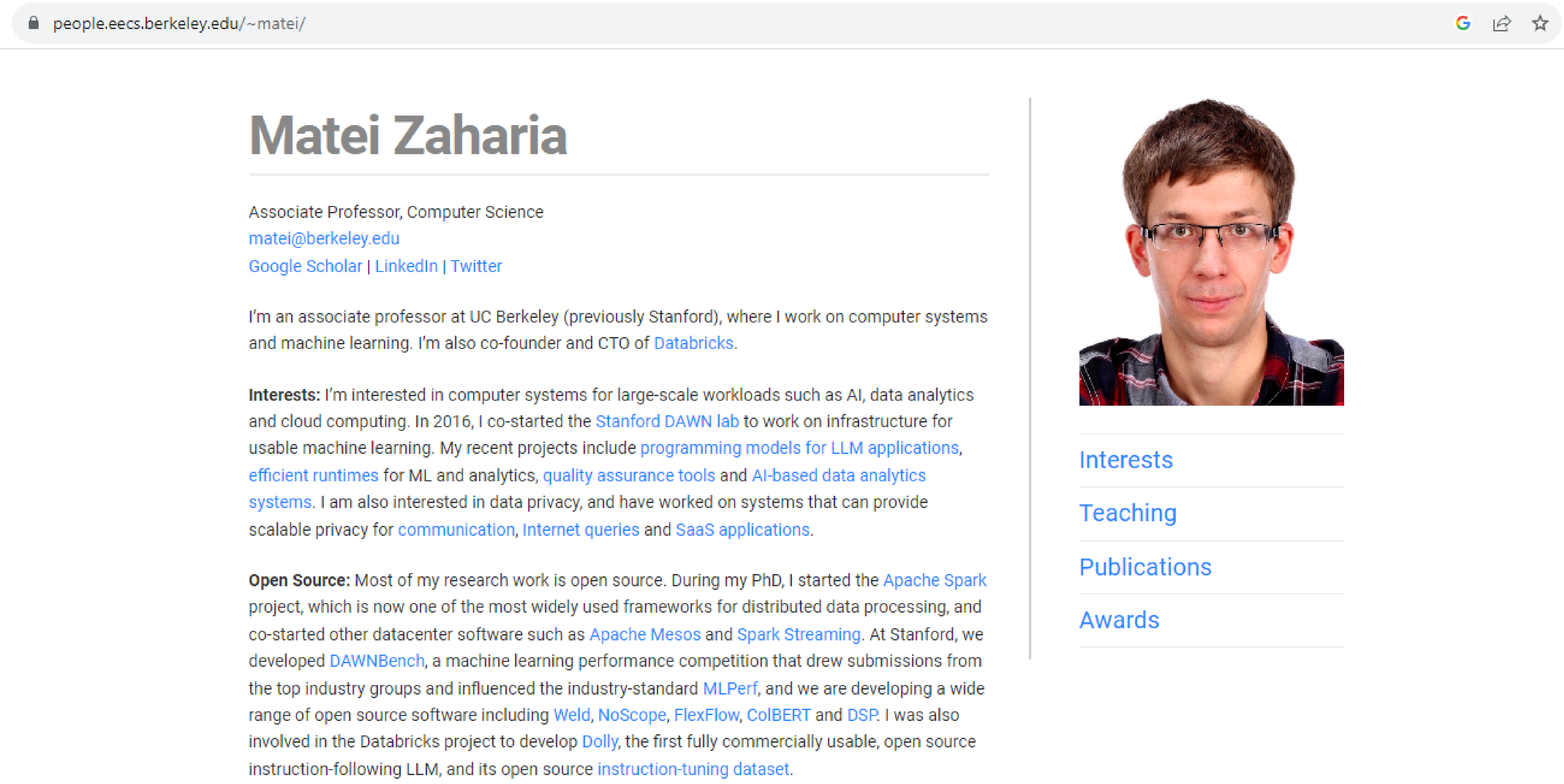


Unified engine for large-scale data analytics

What is Apache Spark™?

Apache Spark™ is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters.

- Spark can handle data sizes ranging from gigabytes to petabytes. The largest known cluster of Spark has over 8000 nodes 🤔.
- Spark was initially started by Matei Zaharia at UC Berkeley's AMPLab in 2009, and open sourced in 2010 under a BSD license.



people.eecs.berkeley.edu/~matei/

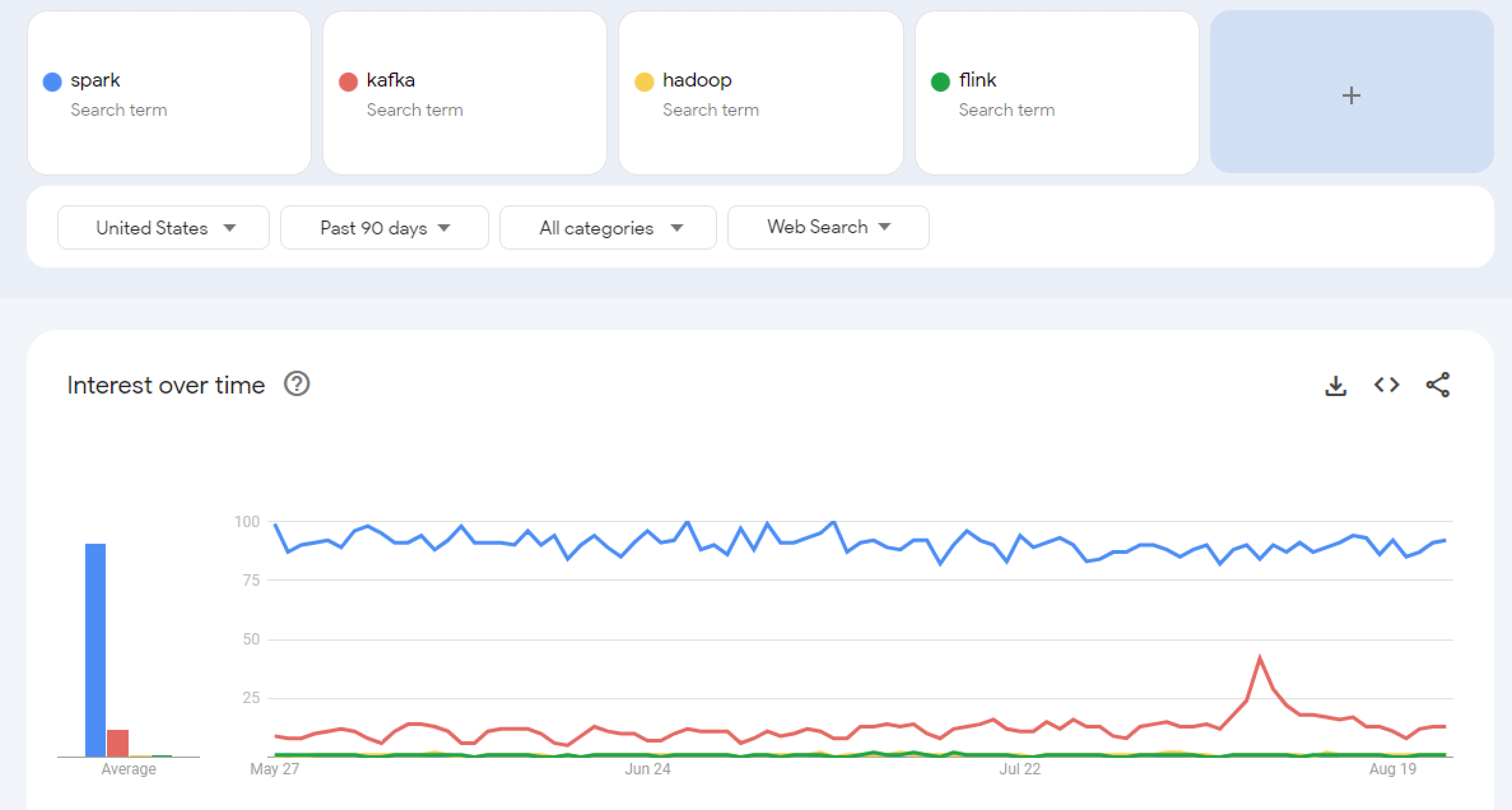
Matei Zaharia

Associate Professor, Computer Science
matei@berkeley.edu
Google Scholar | LinkedIn | Twitter

I'm an associate professor at UC Berkeley (previously Stanford), where I work on computer systems and machine learning. I'm also co-founder and CTO of Databricks.

Interests: I'm interested in computer systems for large-scale workloads such as AI, data analytics and cloud computing. In 2016, I co-started the Stanford DAWN lab to work on infrastructure for usable machine learning. My recent projects include programming models for LLM applications, efficient runtimes for ML and analytics, quality assurance tools and AI-based data analytics systems. I am also interested in data privacy, and have worked on systems that can provide scalable privacy for communication, Internet queries and SaaS applications.

Open Source: Most of my research work is open source. During my PhD, I started the Apache Spark project, which is now one of the most widely used frameworks for distributed data processing, and co-started other datacenter software such as Apache Mesos and Spark Streaming. At Stanford, we developed DAWNBench, a machine learning performance competition that drew submissions from the top industry groups and influenced the industry-standard MLPerf, and we are developing a wide range of open source software including Weld, NoScope, FlexFlow, ColBERT and DSP. I was also involved in the Databricks project to develop Dolly, the first fully commercially usable, open source instruction-following LLM, and its open source instruction-tuning dataset.



● spark
Search term

● kafka
Search term

● hadoop
Search term

● flink
Search term

+

United States ▾ Past 90 days ▾ All categories ▾ Web Search ▾

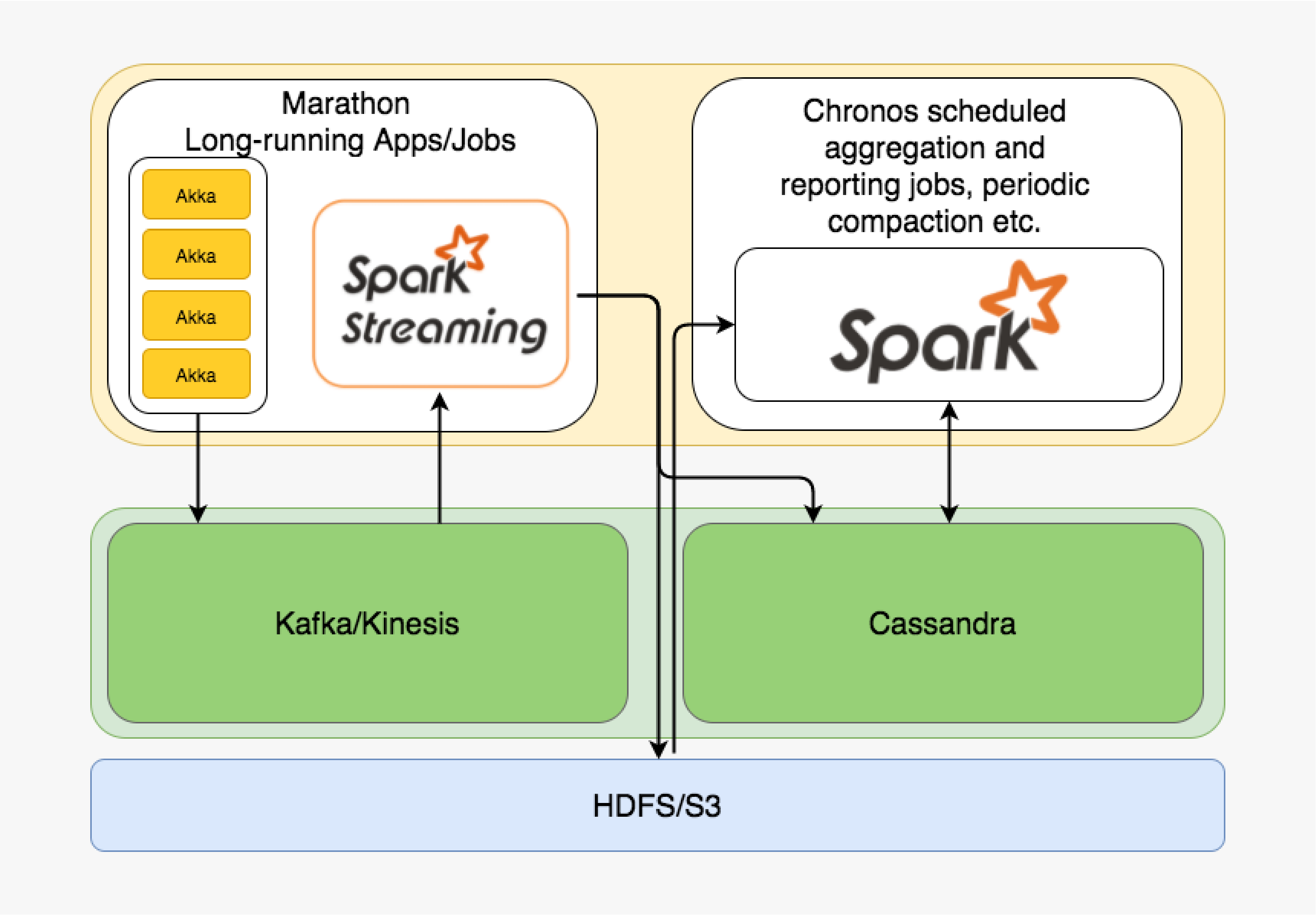
Interest over time ?

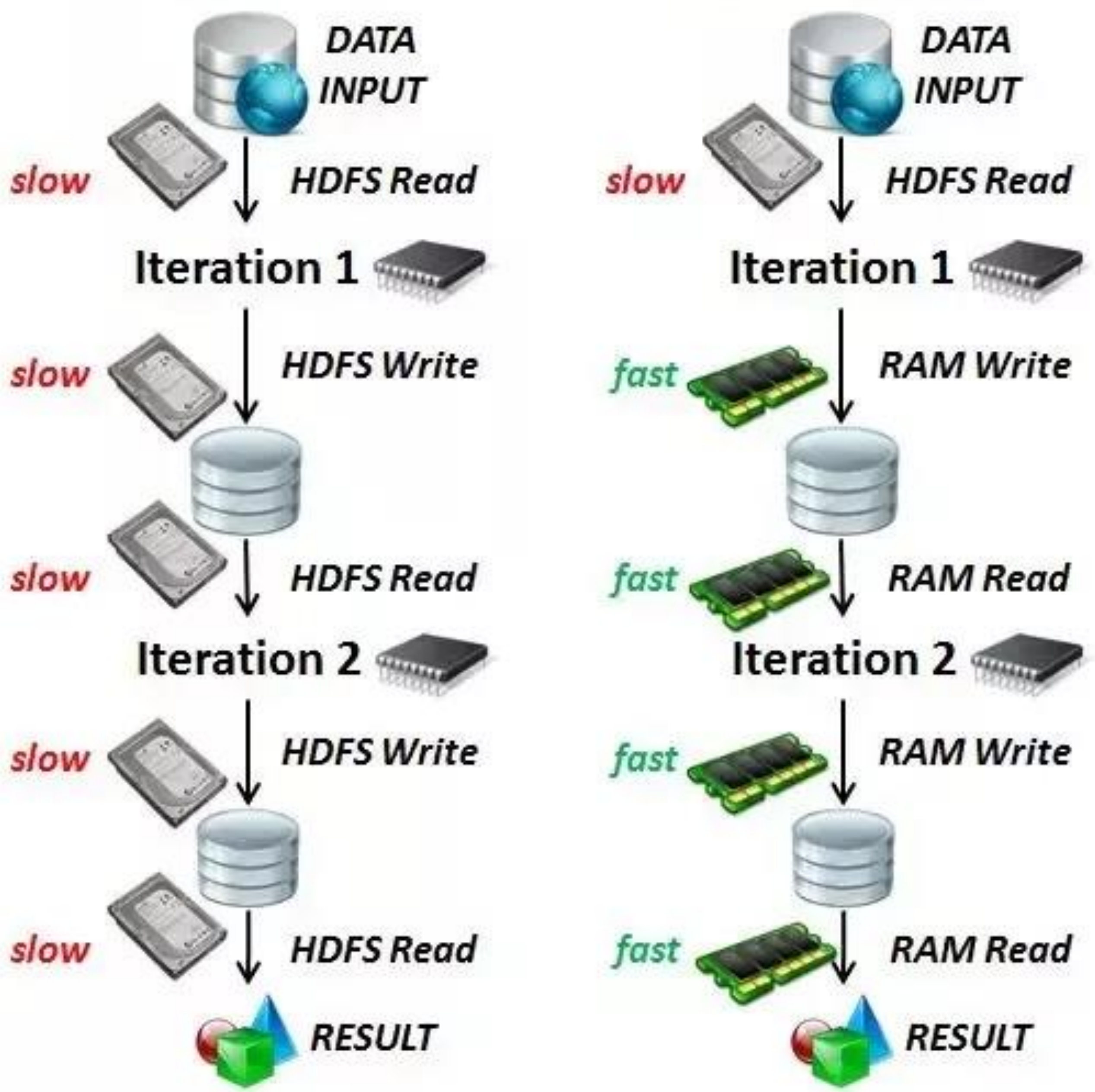
In Hadoop we have

- Yarn (Cluster management)
- HDFS (File system)

But in Spark we do not have any of them or similar pieces. In Spark, sometimes, for example, we work with Casandra or S3 of Amazon (Simple Storage Service).

One awesome cluster management tool for Spark is Kubernetes which we will talk about it in the next Sessions. Just wait! Note that spark can also work with YARN, Mesos or Standalone.





Apache Hadoop

DATA INPUT

slow HDFS Read

Iteration 1

slow HDFS Write

slow HDFS Read

Iteration 2

slow HDFS Write

slow HDFS Read

RESULT

Apache Spark

DATA INPUT

slow HDFS Read

Iteration 1

fast RAM Write

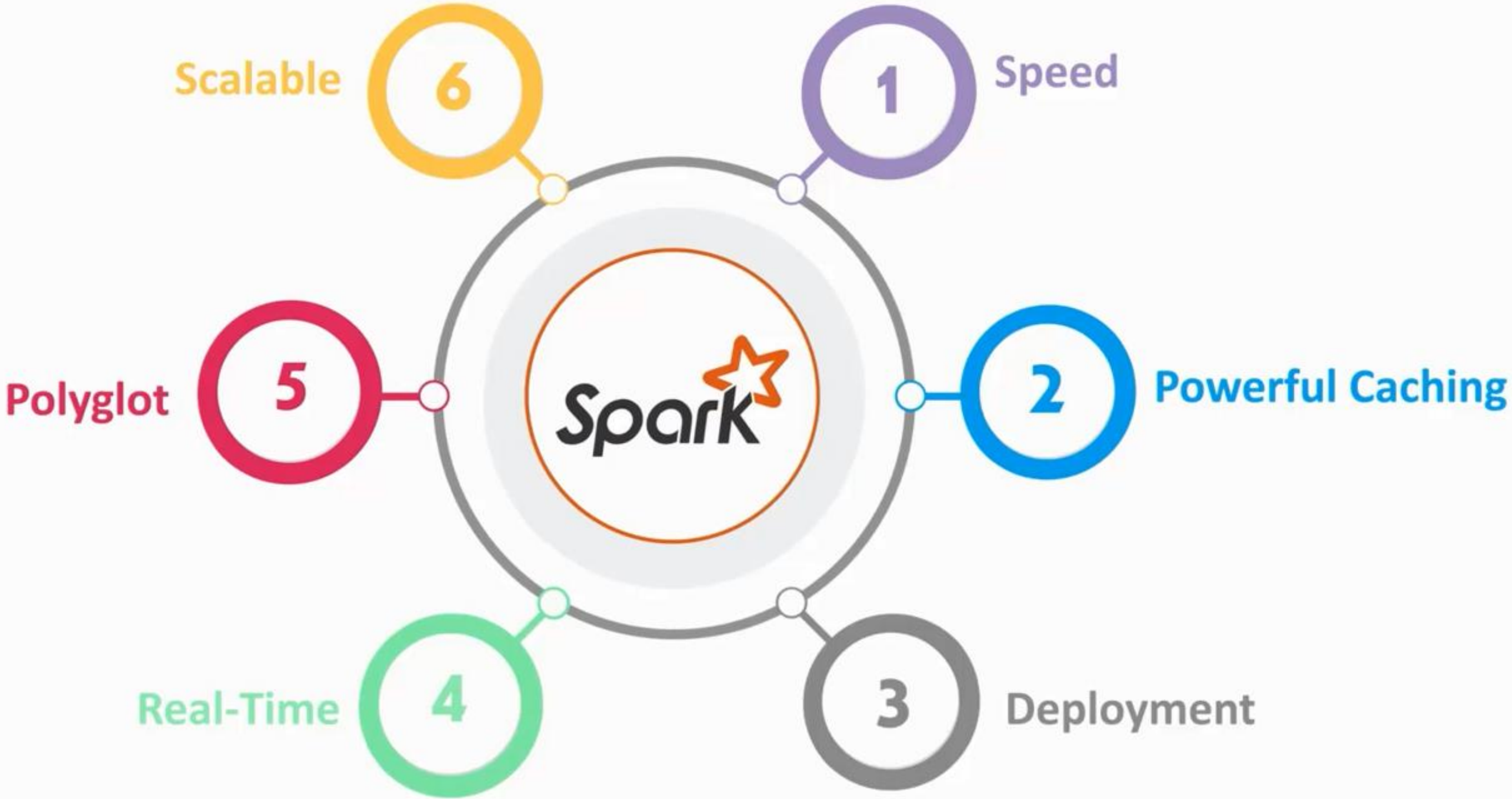
fast RAM Read

Iteration 2

fast RAM Write

fast RAM Read

RESULT



1 Speed

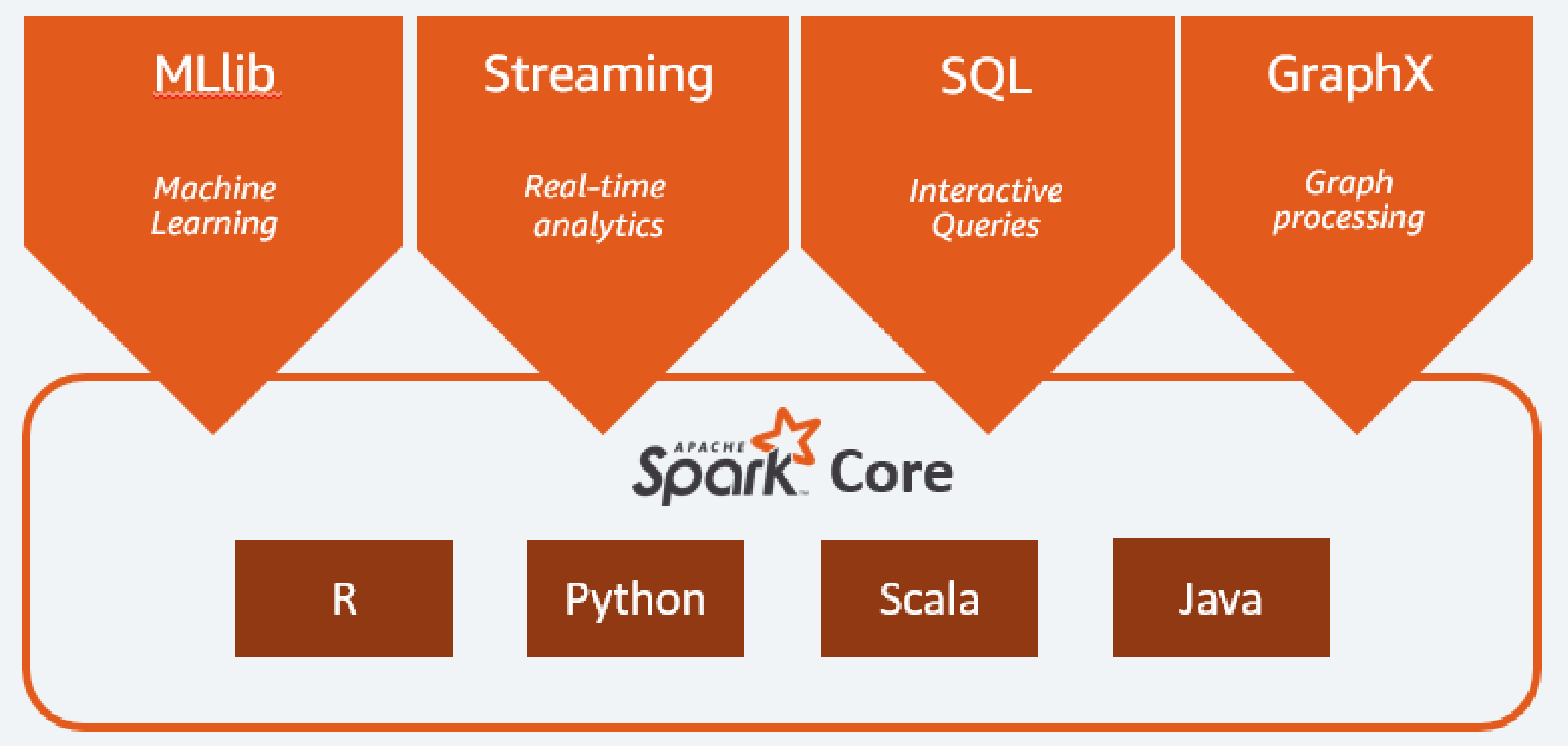
2 Powerful Caching

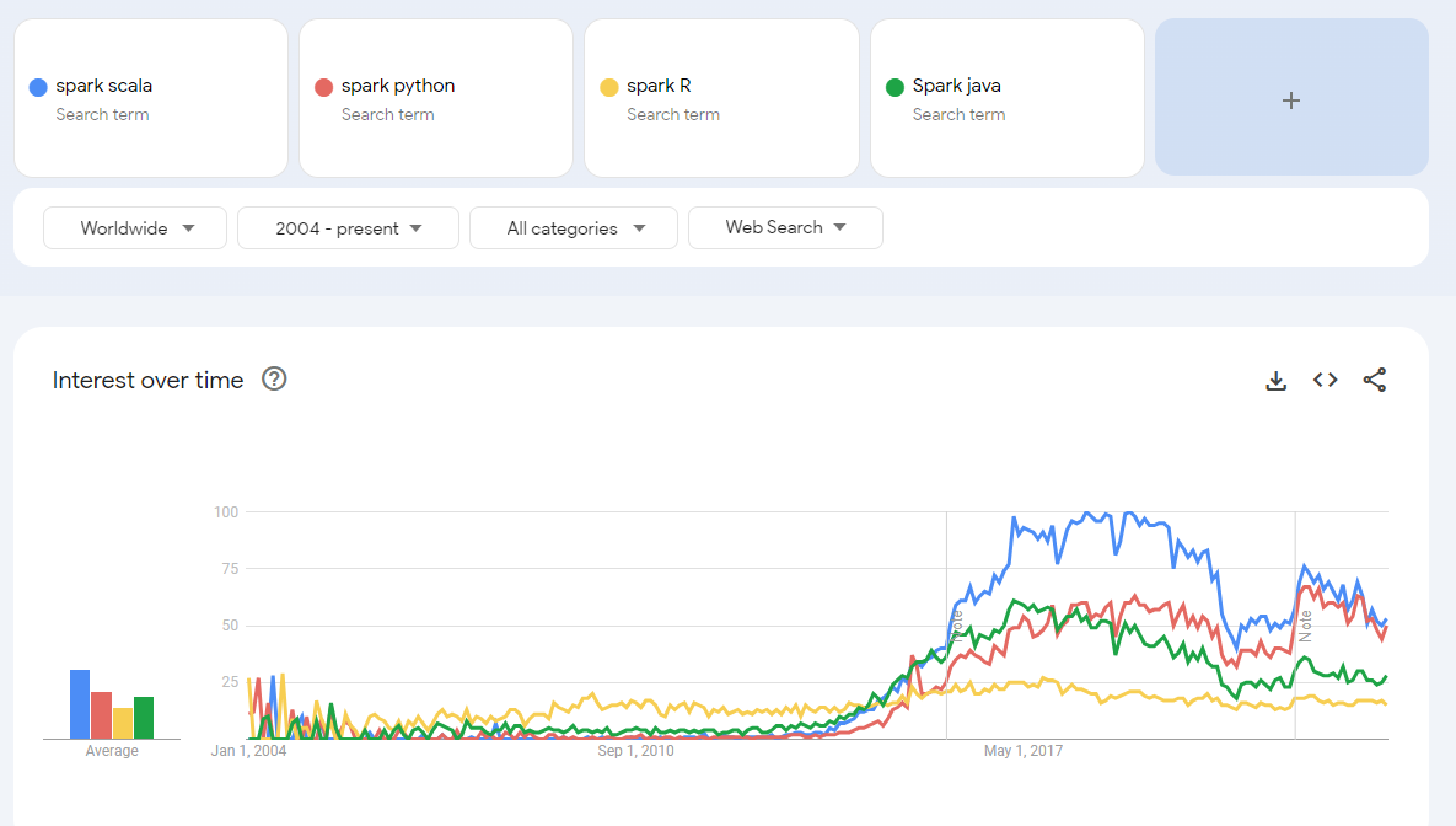
3 Deployment

4 Real-Time

5 Polyglot

6 Scalable


MLlib
Machine Learning
Streaming
Real-time analytics
SQL
Interactive Queries
GraphX
Graph processing
APACHE Spark Core
R
Python
Scala
Java



● spark scala
Search term

● spark python
Search term

● spark R
Search term

● Spark java
Search term

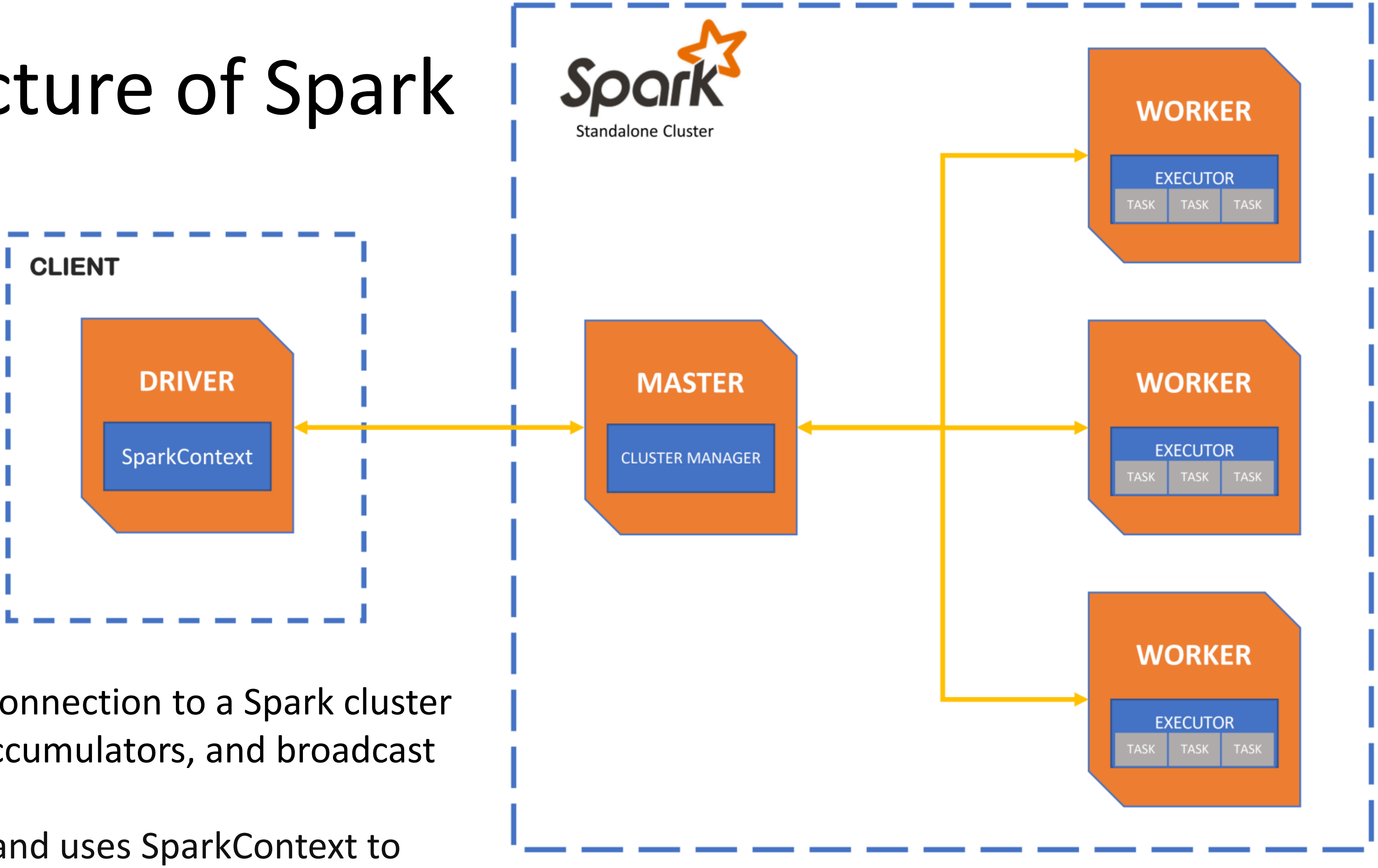
+

Worldwide ▾ | 2004 - present ▾ | All categories ▾ | Web Search ▾

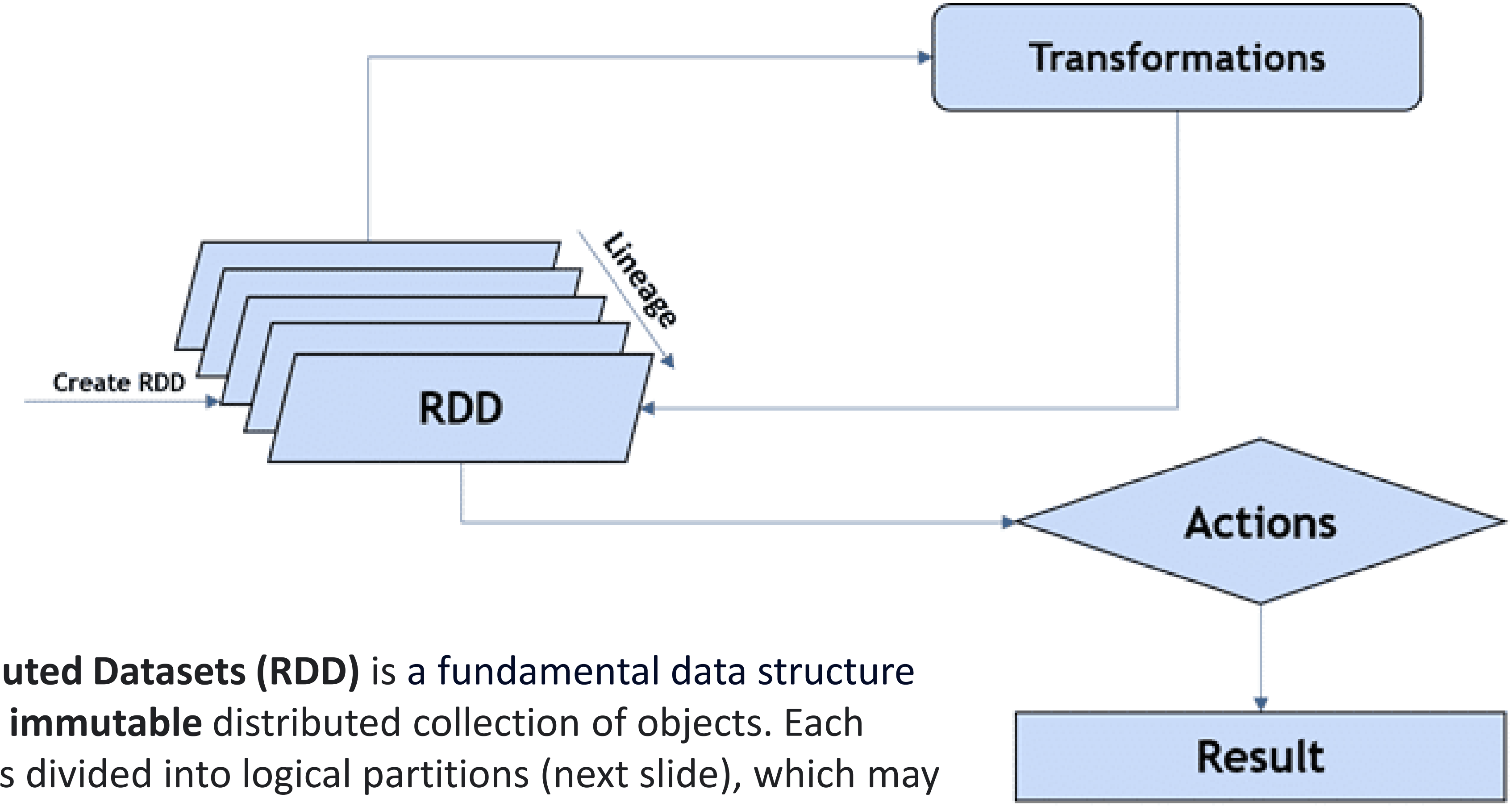
Interest over time

Some statistics shows that usage of Python in the world of Spark is 70%

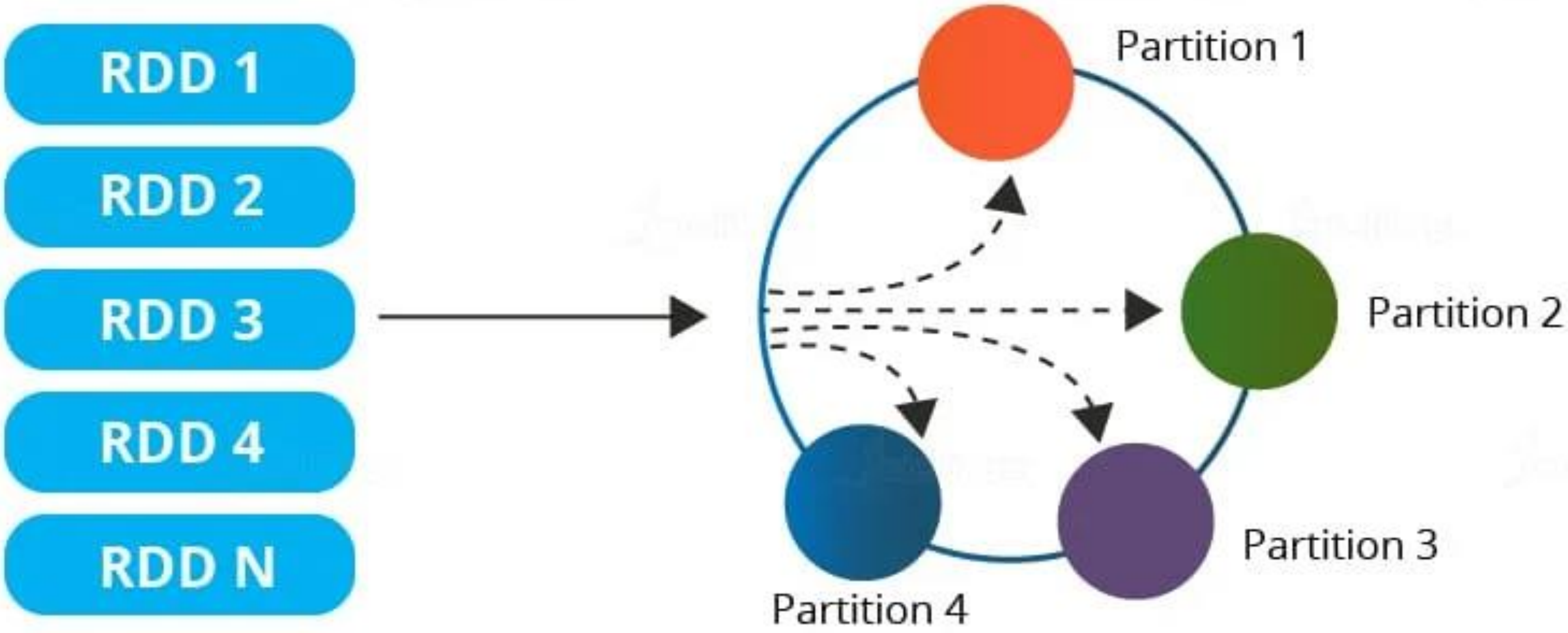
General Architecture of Spark

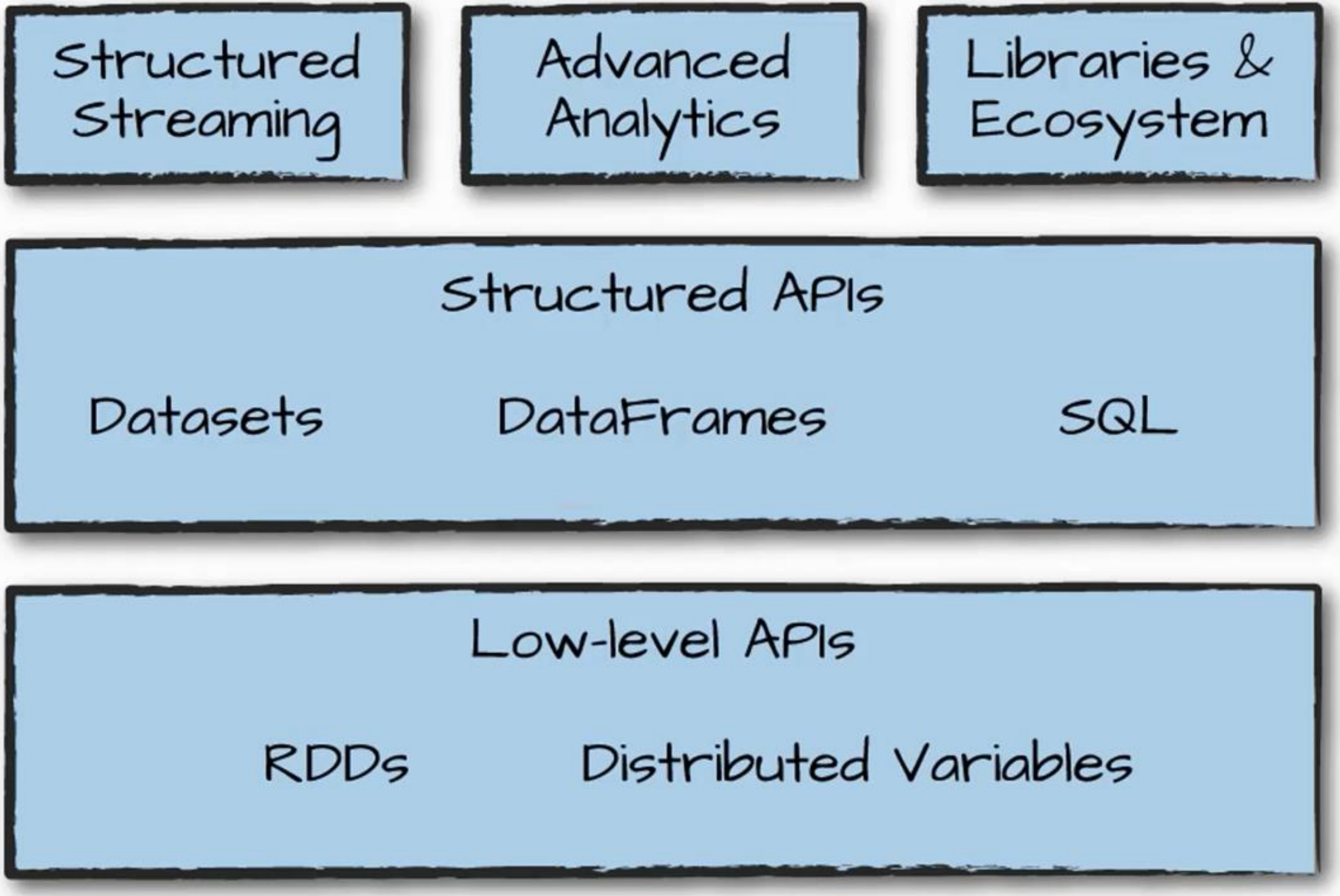


- **The SparkContext** represents the connection to a Spark cluster and can be used to create RDDs, accumulators, and broadcast variables on that cluster.
- **The Spark driver** program creates and uses SparkContext to connect to the cluster manager to submit Spark jobs, and know what resource manager to communicate to. It is the heart of the Spark application.



Resilient Distributed Datasets (RDD) is a fundamental data structure of Spark. It is an **immutable** distributed collection of objects. Each dataset in RDD is divided into logical partitions (next slide), which may be computed on different nodes of the cluster.


RDD 1
RDD 2
RDD 3
RDD 4
RDD N
Partition 1
Partition 2
Partition 3
Partition 4



Structured Streaming

Advanced Analytics

Libraries & Ecosystem

Structured APIs

Datasets

DataFrames

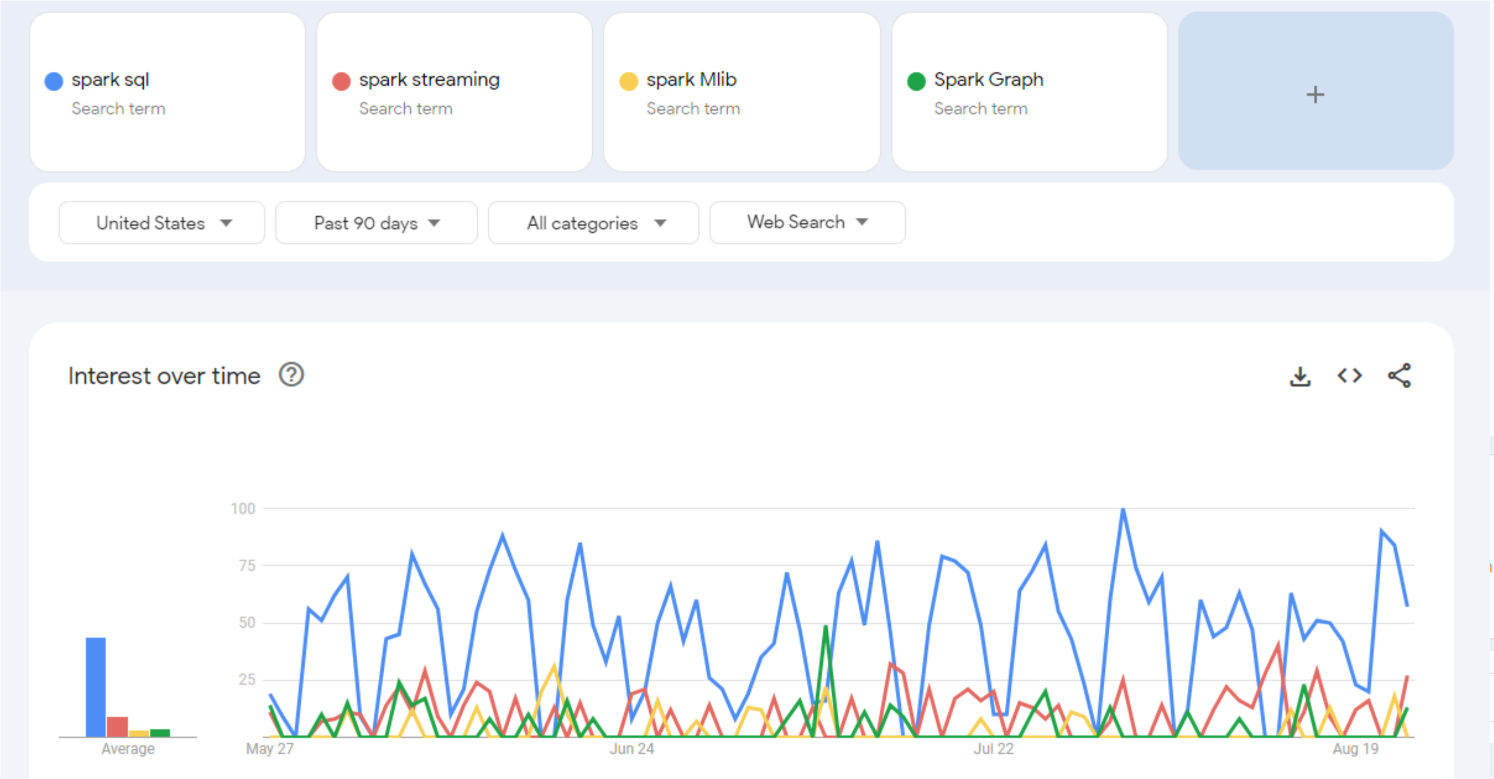
SQL

Low-level APIs

RDDs

Distributed Variables

The most applicable tool



spark sql
Search term

spark streaming
Search term

spark Mlib
Search term

Spark Graph
Search term

United States ▾ | Past 90 days ▾ | All categories ▾ | Web Search ▾

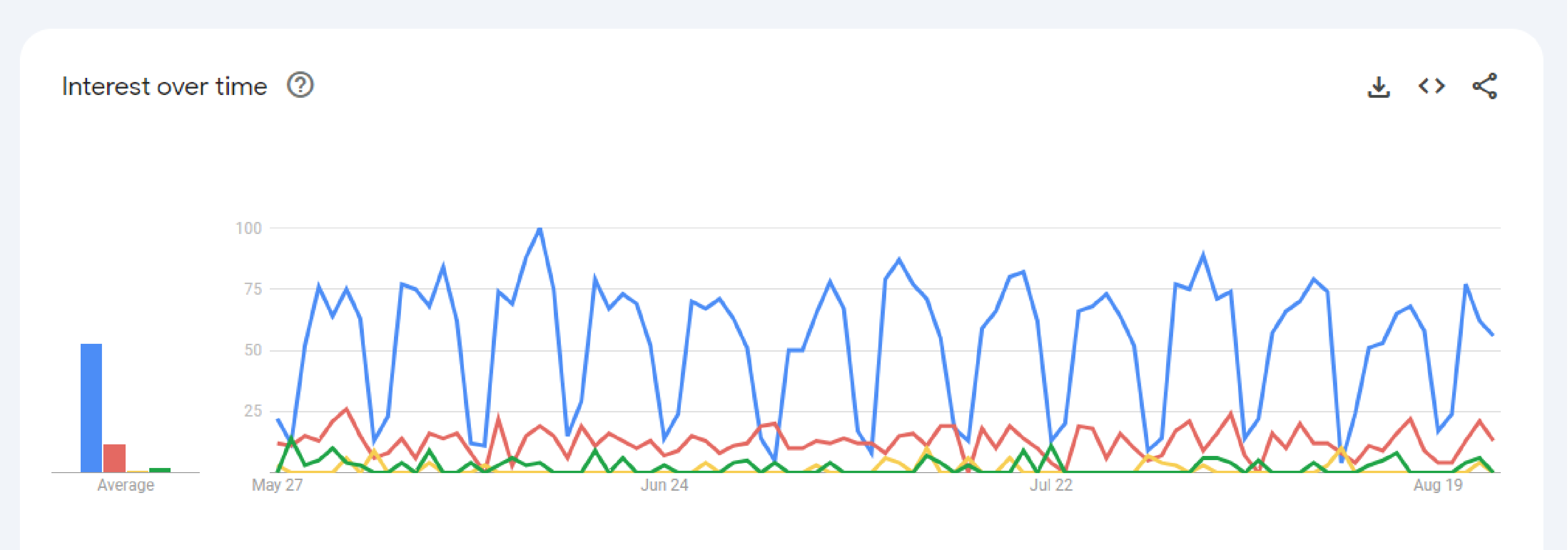
Interest over time

Spark SQL is a very important and most used module that is used for structured data processing. Spark SQL allows you to query structured data using either SQL or DataFrame API.

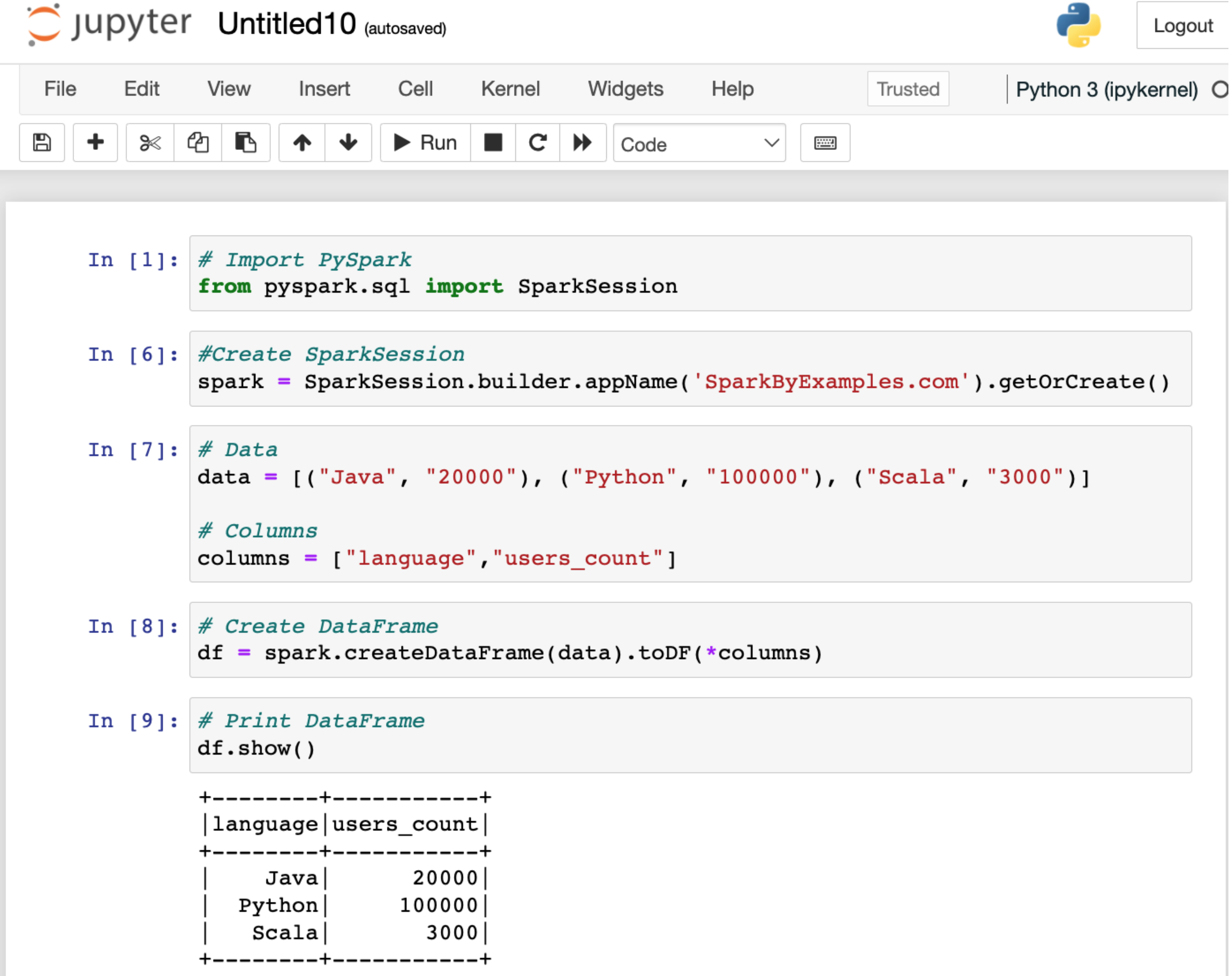
Mlib
term

Spark Graph
Search term

Web Search ▾



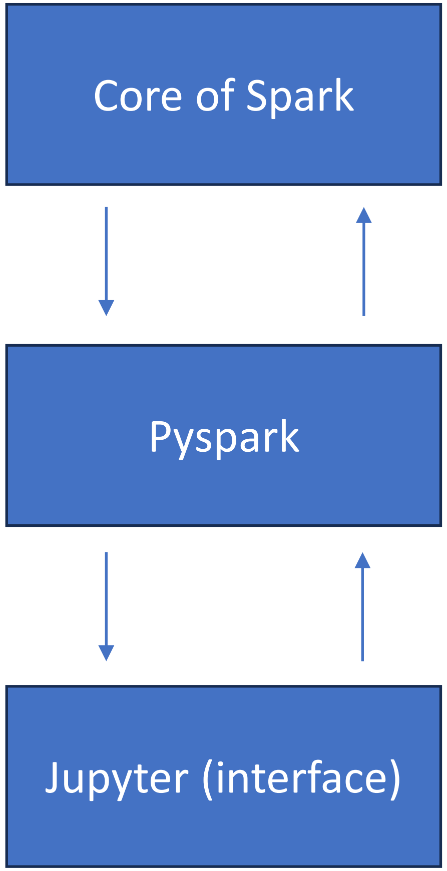
Interest over time



```
In [1]: # Import PySpark
        from pyspark.sql import SparkSession

In [6]: #Create SparkSession
        spark = SparkSession.builder.appName('SparkByExamples.com').getOrCreate()

In [7]: # Data
        data = [("Java", "20000"), ("Python", "100000"), ("Scala", "3000")]

        # Columns
        columns = ["language","users_count"]

In [8]: # Create DataFrame
        df = spark.createDataFrame(data).toDF(*columns)

In [9]: # Print DataFrame
        df.show()

        +--------+-----------+
        |language|users_count|
        +--------+-----------+
        |    Java|      20000|
        |  Python|     100000|
        |   Scala|       3000|
        +--------+-----------+
```

We will work with

- Pyspark
- Jupyter



PyCharm

The Python IDE for data science and web development

Make development more productive and enjoyable

Full-fledged Professional or Free Community

SparkSession vs. SparkContext

- اسپارک‌کانتکست مسئول مدیریت کلاستر اسپارک و هماهنگی تسک‌های آن است.
- اسپارک‌سشن روی SC ساخته می‌شود و یک API سطح بالا و کاربرپسند برای کار با دیتاهای با ساختار می‌باشد (با سایر دیتاها نیز کار می‌کند اما با ساختاریافته‌ها بهتر عمل می‌نماید).

نکته مهم: در اپ‌های مدرن از SS استفاده کنید. از طرفی SC را وقتی بکار بگیرید که می‌خواهید در سطح پائین‌تری کار کنیم (بعنوان مثال با RDDها).