OlaPy, un outil pour l'analyse de données métier

Business analytics with OlaPy

Paris Open Source Summit - 11 Dec. 2019

Stéfane Fermigier

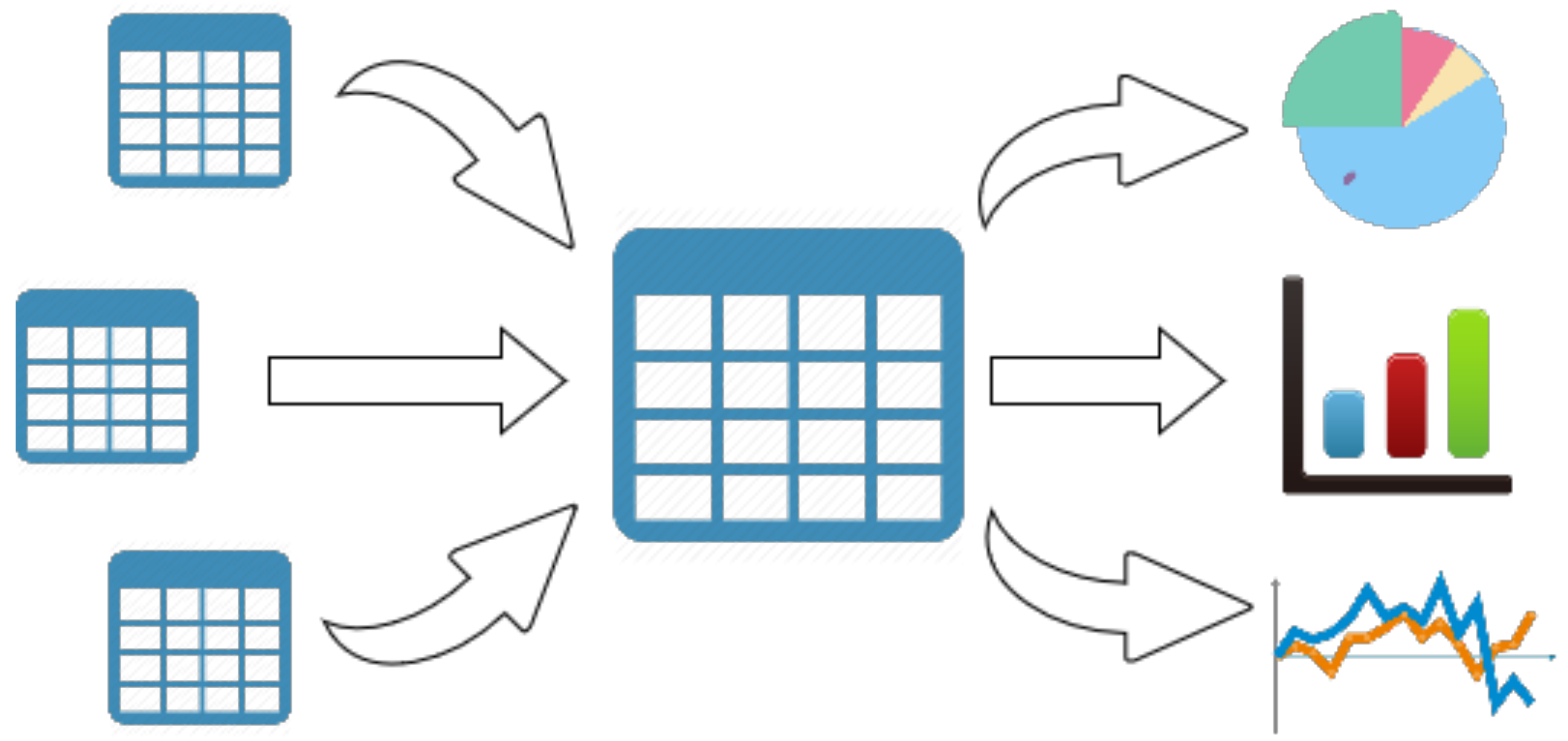
Founder & CEO, Abilian - Enterprise Social Software

abilian

CONNECTED, WE WORK!

Olapy in brief

- Developed since 2016 by Abilian
- In-memory data processing using Pandas
- Aggregated data browsing
- MDX support
- XMLA interface (-> Excel)
- Multiple back-ends (CSV, SQL)
- Simple web front-end and in-browser app



abilian

Before we start / motivations

abilian

Who am I ?

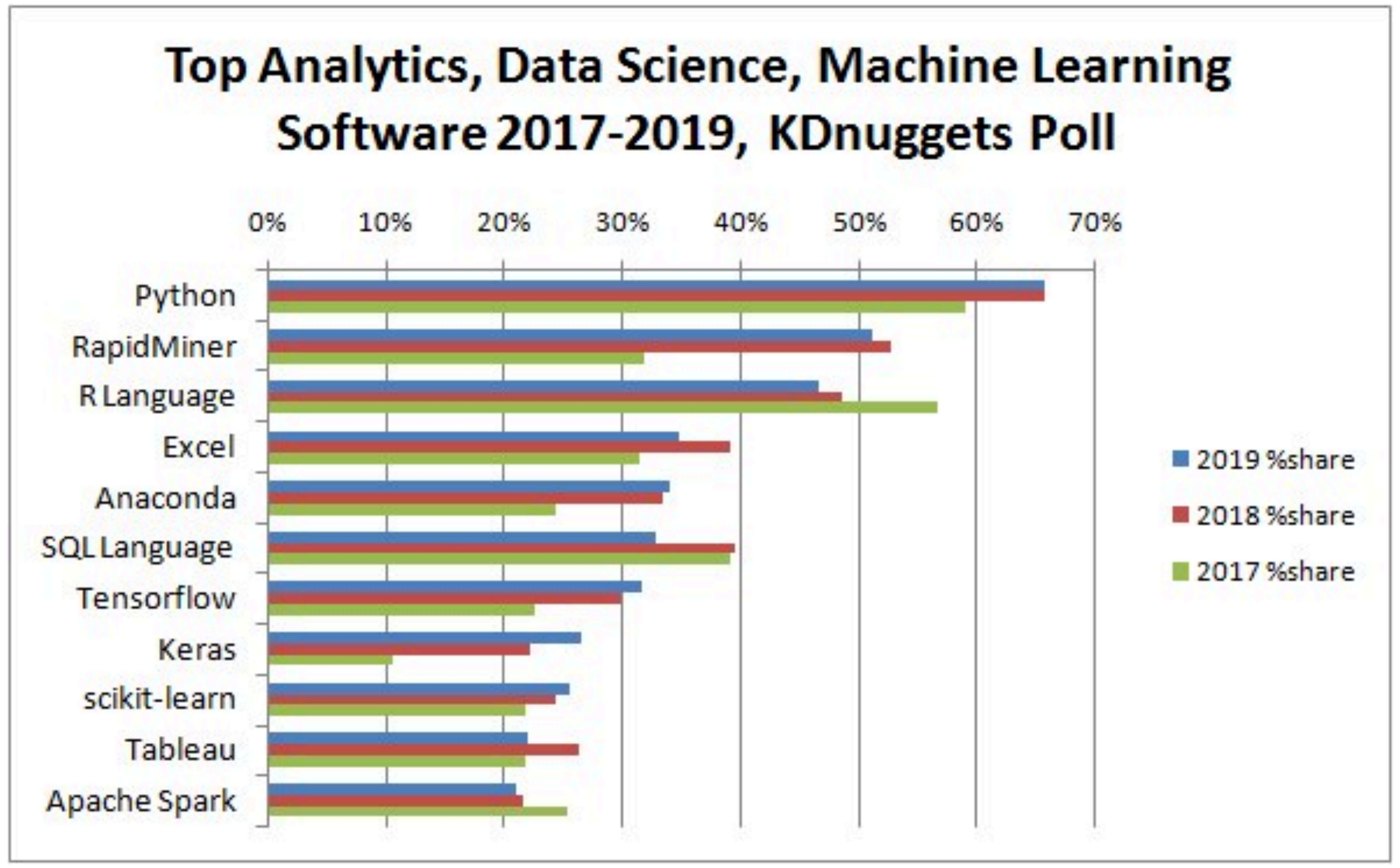
- Stefane Fermigier, Python developer since 1996
- Founder of Abilian SAS
  - Python shop, developing business application (collaboration, CRM, workflow…)
  - R&D activity (Wendelin -> Olapy)
- Organizer of the PyData Paris / PyParis conference (2014-2018)

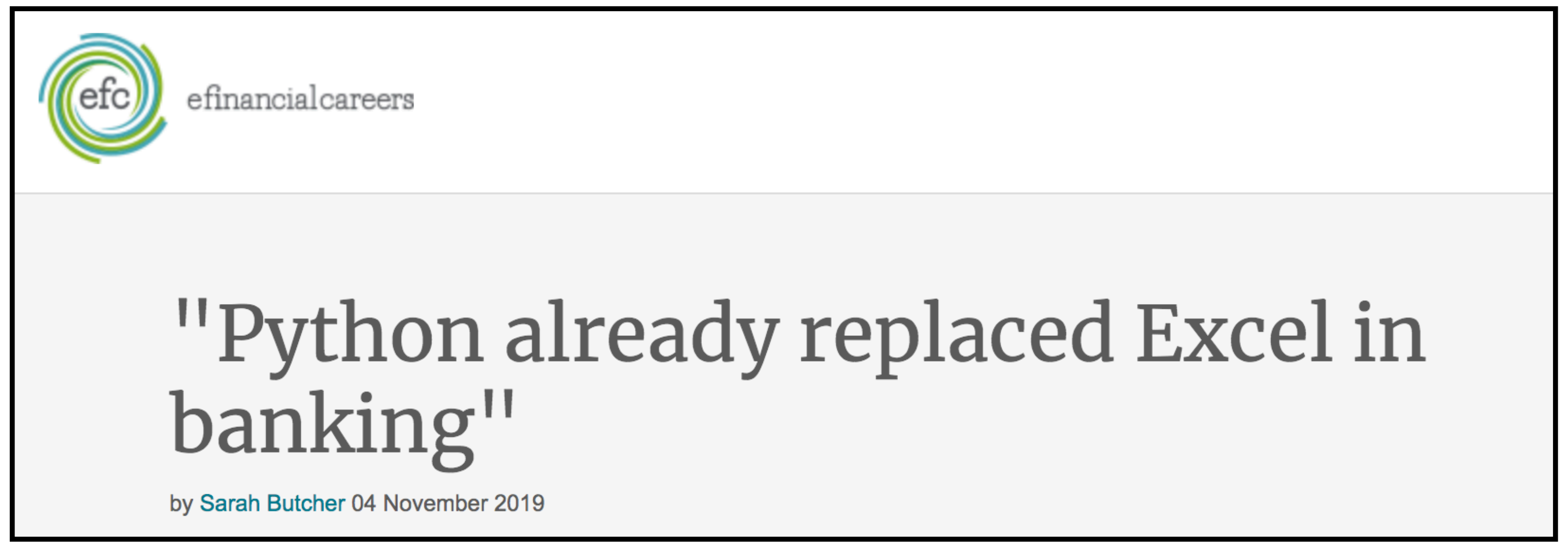
WENDELIN

abilian

Why use Python for business data analysis ?

- Why not? :)
- Python is one of the leading languages for data science / data processing, and also a leading language for web & business apps
- As a Python shop, we'd like to leverage this leadership in data processing tools to build exploration / reporting features in our business applications using a familiar language




efinancialcareers

"Python already replaced Excel in banking"

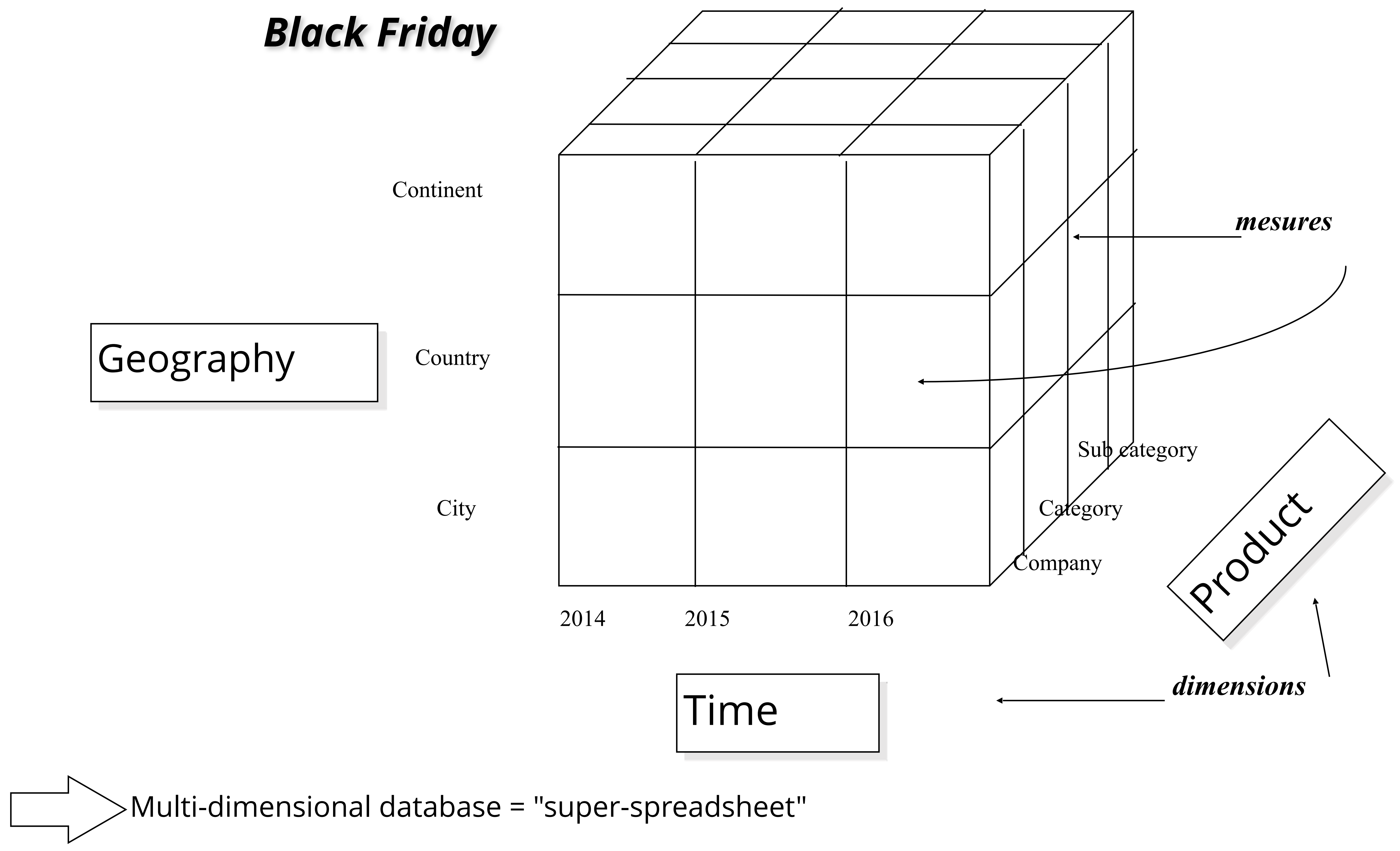
by Sarah Butcher 04 November 2019

Concepts and architecture

abilian

On-Line Analytical Processing (OLAP) & Multidimensional Databases

- A multidimensional DB is an hypercube
- Axes are called user-defined **dimensions**
- Cells contain **measures** calculated from more or less complex formulas
- **Operators** on the cube are **algebraic** (return a cube) and can thus be combined



MDX: a query language for business analytics

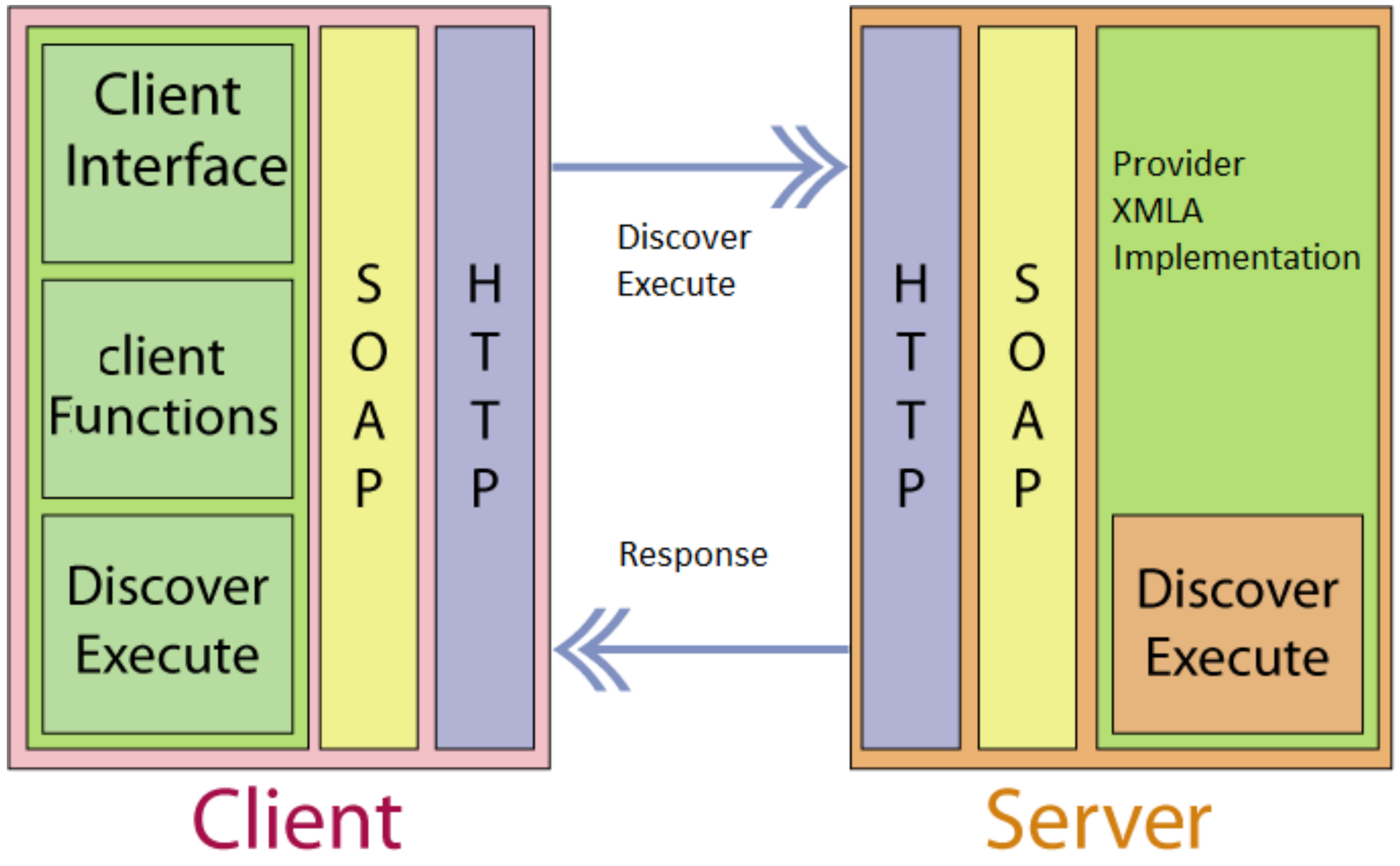
- MDX = *Multi Dimensional Expressions*
- SQL extension for querying a multi-dimensional database
- Example:

```
SELECT
  [Geography].[Geo].[Country] ON ROWS,
  [Time].[Calendar].[Year].[2010] ON COLUMNS
FROM sales
WHERE [Measures].[Count]
```

abilian

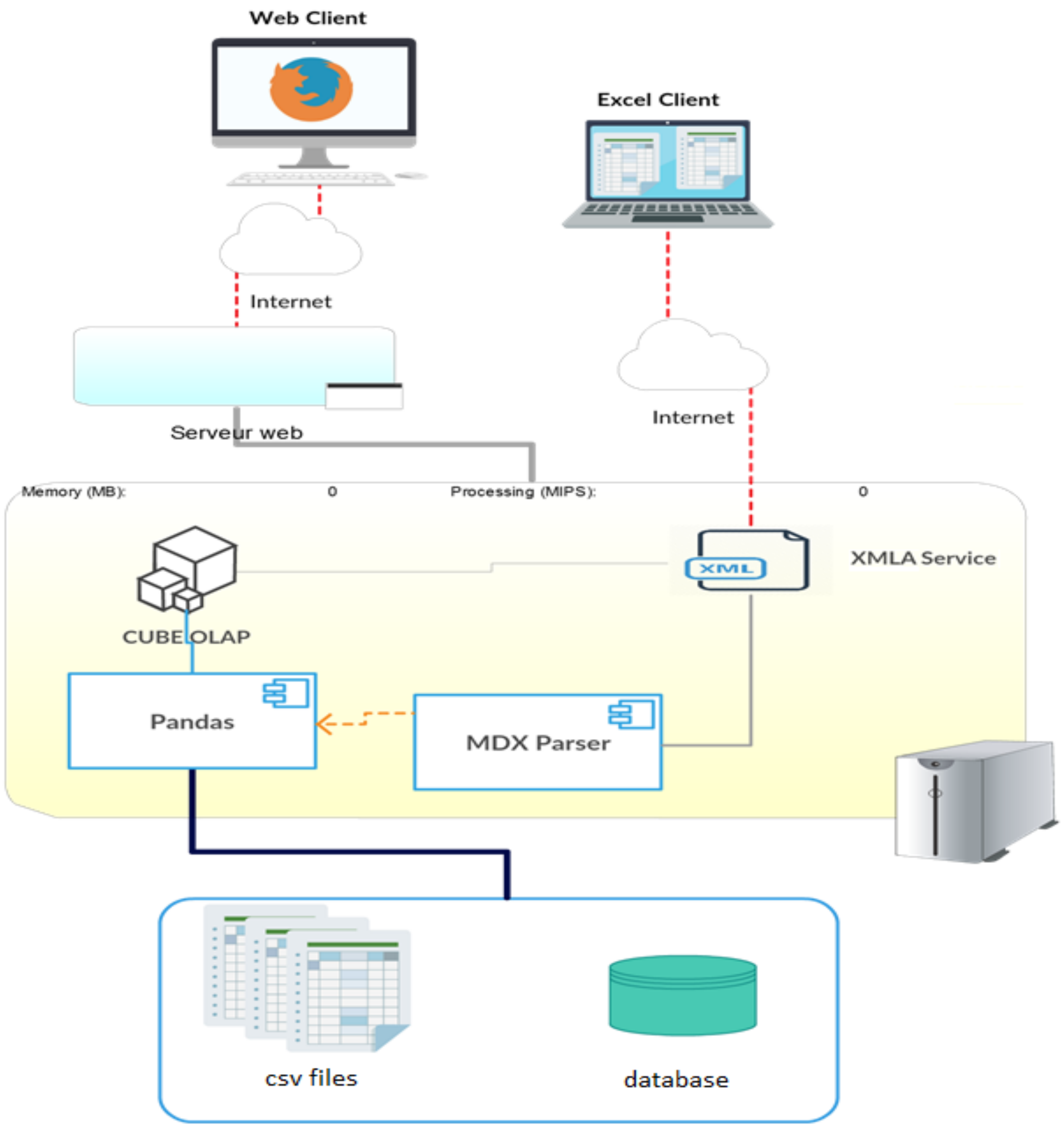
XMLA - Extensible Markup Language for Analysis

- Data Access Protocol
- Supports exchange of analytical data between clients and servers
    - Available on any device or platform
    - Using any programming language
- SOAP with just 2 methods
    - Discover
    - Execute

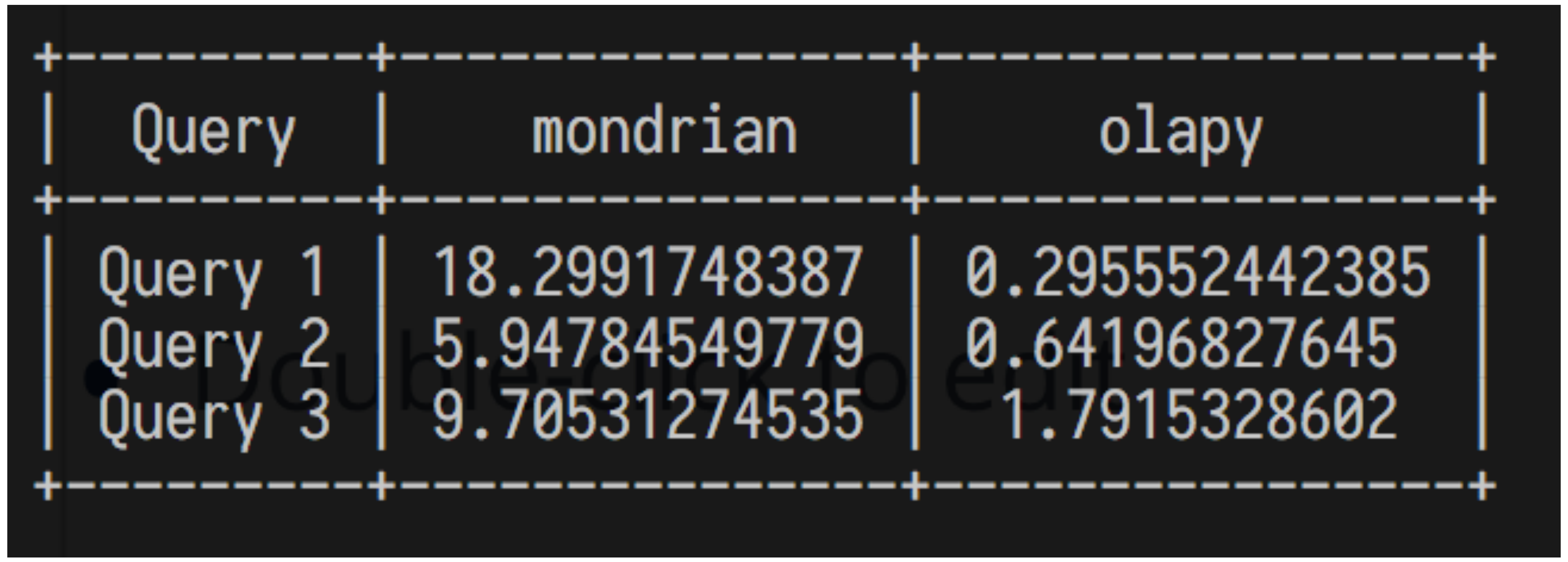


abilian

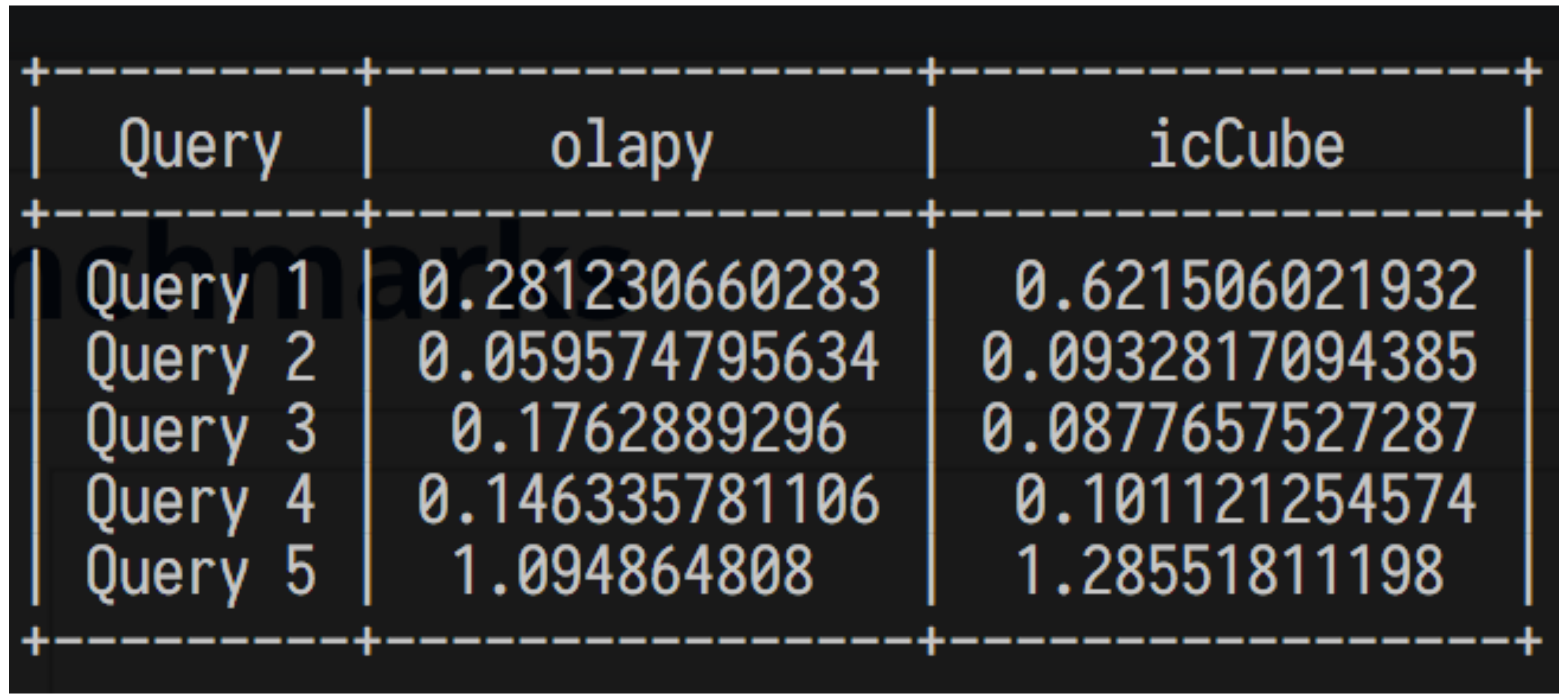
Detailed architecture



Benchmarks (WIP)



```
+----------+---------------+-----------------+
|  Query   |   mondrian    |      olapy      |
+----------+---------------+-----------------+
| Query 1  | 18.2991748387 | 0.295552442385  |
| Query 2  | 5.94784549779 | 0.64196827645   |
| Query 3  | 9.70531274535 |  1.7915328602   |
+----------+---------------+-----------------+
```



```
+---------+----------------+-----------------+
|  Query  |     olapy      |      icCube     |
+---------+----------------+-----------------+
| Query 1 | 0.281230660283 |  0.621506021932 |
| Query 2 | 0.059574795634 | 0.0932817094385 |
| Query 3 |  0.1762889296  | 0.0877657527287 |
| Query 4 | 0.146335781106 |  0.101121254574 |
| Query 5 |  1.094864808   |  1.28551811198  |
+---------+----------------+-----------------+
```

abilian

Use cases & applications

From a spreadsheet software (e.g. Excel)

- Install & run:

```
pip install olapy

olapy runserver
```

- Then, from excel go to:
  - Data/from other sources/
- And on “analyses services”
- Use URL: `http://127.0.0.1:8000/xmla`

```
(olapy) C:\Users\Mouadh\Google Drive\olapy-core\olapy-core>python olapy -c
```

C:\WINDOWS\system32\cmd.exe
olapy-core
cubes
olapy.egg-info
__init__.py
Fichier PY
2,79 Ko
MANIFEST.in
Fichier IN
135 octet(s)
requirements.txt
Document texte
2,14 Ko
tox.ini
Paramètres de configuration
175 octet(s)
24 élément(s)
Courrier entran...
Lecteur Windo...
Nouvel onglet ...
olapy-core
C:\WINDOWS\...
Book1 (version ...
18:39

Other clients

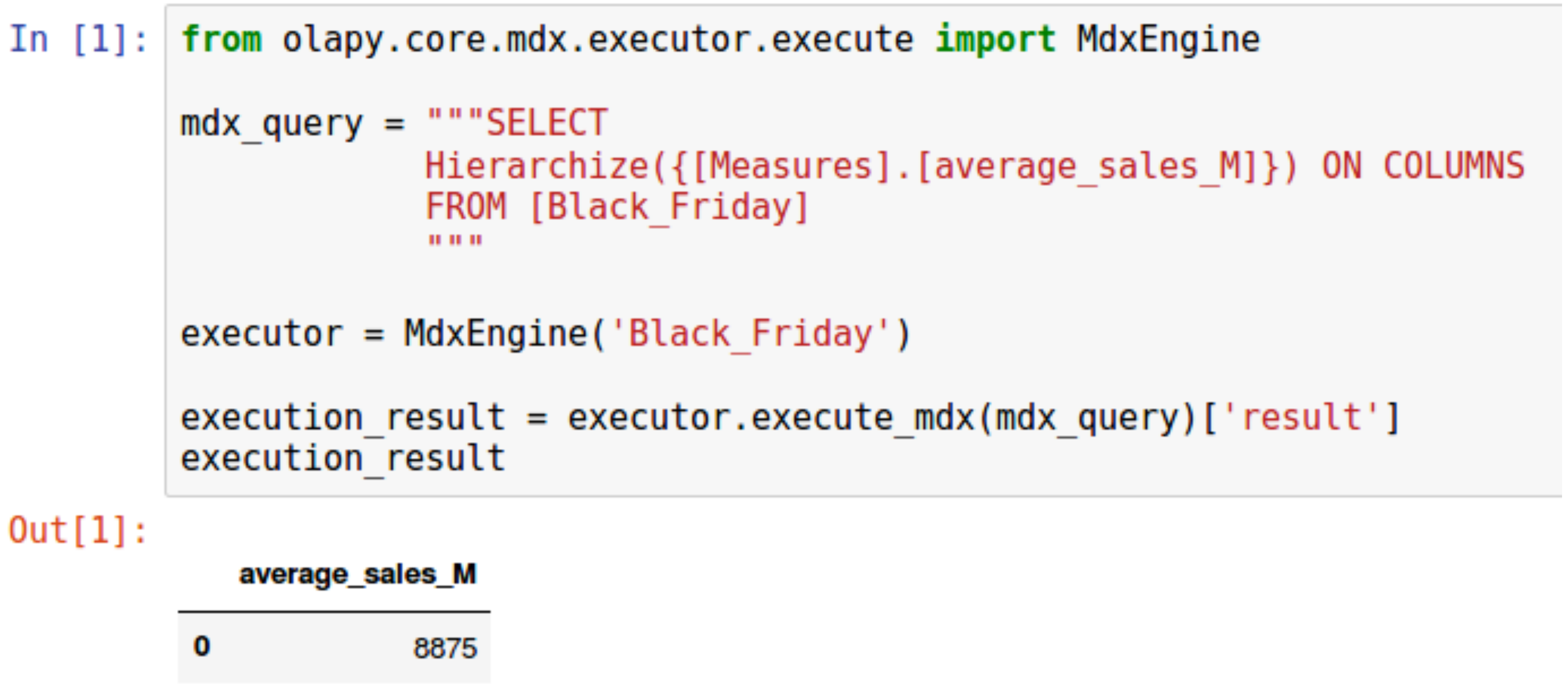
- xmla.js : JavaScript client
  - Ongoing work to be able to call OlaPy (or any other XMLA server) from browser-based spreadsheet software, such as OnlyOffice, Jexcel, Sheetjs, etc.
- olap4j: Java client
  - Used (among others) by the PalOOca plugin for LibreOffice
- Clients also for Python, .NET, Perl, Ruby, etc.

Web application (POC)

- Flask-based Web application (other framework will be supported)
- GUI-based MDX query editor
- GUI-based data explore / aggregator
- Graphical widgets
- Support for dashboarding

HOME
Dashboards 0
Cubes 0
Query Builder 0
+Add Cube

As a Python library - using Jupyter (or not)



```
In [1]: from olapy.core.mdx.executor.execute import MdxEngine

        mdx_query = """SELECT
                    Hierarchize({[Measures].[average_sales_M]}) ON COLUMNS
                    FROM [Black_Friday]
                    """

        executor = MdxEngine('Black_Friday')

        execution_result = executor.execute_mdx(mdx_query)['result']
        execution_result

Out[1]:
      average_sales_M
   0             8875
```

abilian

Notebook in the browser - using Pyiodide

- **Pyodide** brings the Python runtime to the browser via **WebAssembly**, along with the Python scientific stack including NumPy, Pandas, Matplotlib, parts of SciPy, and NetworkX. The packages directory lists over 35 packages which are currently available.
- Pyodide provides transparent conversion of objects between Javascript and Python. When used inside a browser, Python has full access to the Web APIs.
- While closely related to the iodide project, a tool for literate scientific computing and communication for the web, Pyodide goes beyond running in a notebook environment. To maximize the flexibility of the modern web, Pyodide may be used standalone in any context where you want to run Python inside a web browser.

Python
VIEW
plugin
js
py

```
{
  "languageId": "py",
  "displayName": "python",
  "codeMirrorMode": "python",
  "keybinding": "p",
  "url": "/pyodide_dev.js",
  "module": "pyodide",
  "evaluator": "runPython",
  "pluginType": "language"
}
```

```
pyodide.loadPackage('olapy')
```

```
import pandas as pd
import pyodide
from olapy.core.services.xmla_lib import get_response

xmla_request_params = {'cube': 'sales','request_type': 'DISCOVER_PROPERTIES','properties': {},
          'restrictions': {'PropertyName': 'ServerName'},'mdx_query': None}

dataframes = {'Facts' : pd.read_csv(pyodide.open_url("olapy-data/cubes/sales/Facts.csv"),sep=';', encoding='utf8'),
'Product':pd.read_csv(pyodide.open_url("olapy-data/cubes/sales/Product.csv"),sep=';', encoding='utf8'),
'Geography':pd.read_csv(pyodide.open_url("olapy-data/cubes/sales/Geography.csv"),sep=';', encoding='utf8')
}
```

Out-of-core in-memory computing - using Wendelin

"Wendelin is a big data framework designed for industrial applications based on python, NumPy, Scipy and other NumPy based libraries. It uses at its core the NEO distributed transactional NoSQL database to store **petabytes of binary data**. Wendelin combines the performance of scikit-learn machine learning with NEO distributed storage in order to provide **out-of-core processing of large data sets**. Its goal is to bring the best open source, big data engine based on Numpy python technologies and gather a wide community of contributors of new data analytics algorithms."

Roadmap and support

abilian

Roadmap

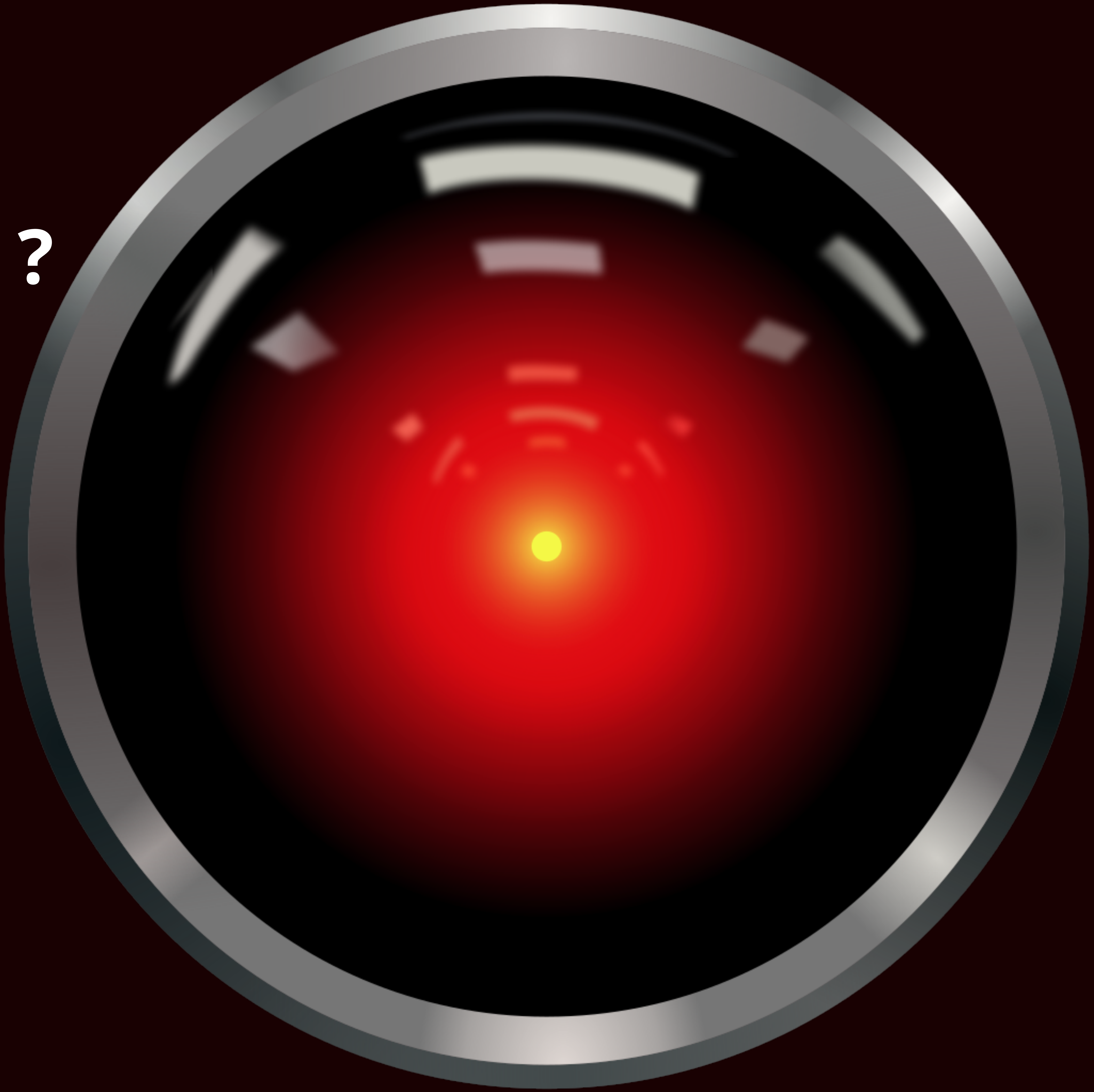
- Version 0.8 will be released before year end
  - Last version to support Python 2.7
- Then (2020):
  - Supported release of Olapy / Pyodide
  - Integration with Web spreadsheets
  - Web app (both standalone and as a component)
  - More use cases

abilian

Support offer

- Starting with release 0.8, we will sell support on Olapy
- Contact us for details :)

Questions ?



abilian