Ministry of Higher Education and Scientific Research
Djilali BOUNAAMA University - Khemis Miliana(UDBKM)
Faculty of Science and Technology
Department of Mathematics and Computer Science

Chapter 3

Data Science Tools

AIBD-M1-UEM112 : Introduction to Data Science

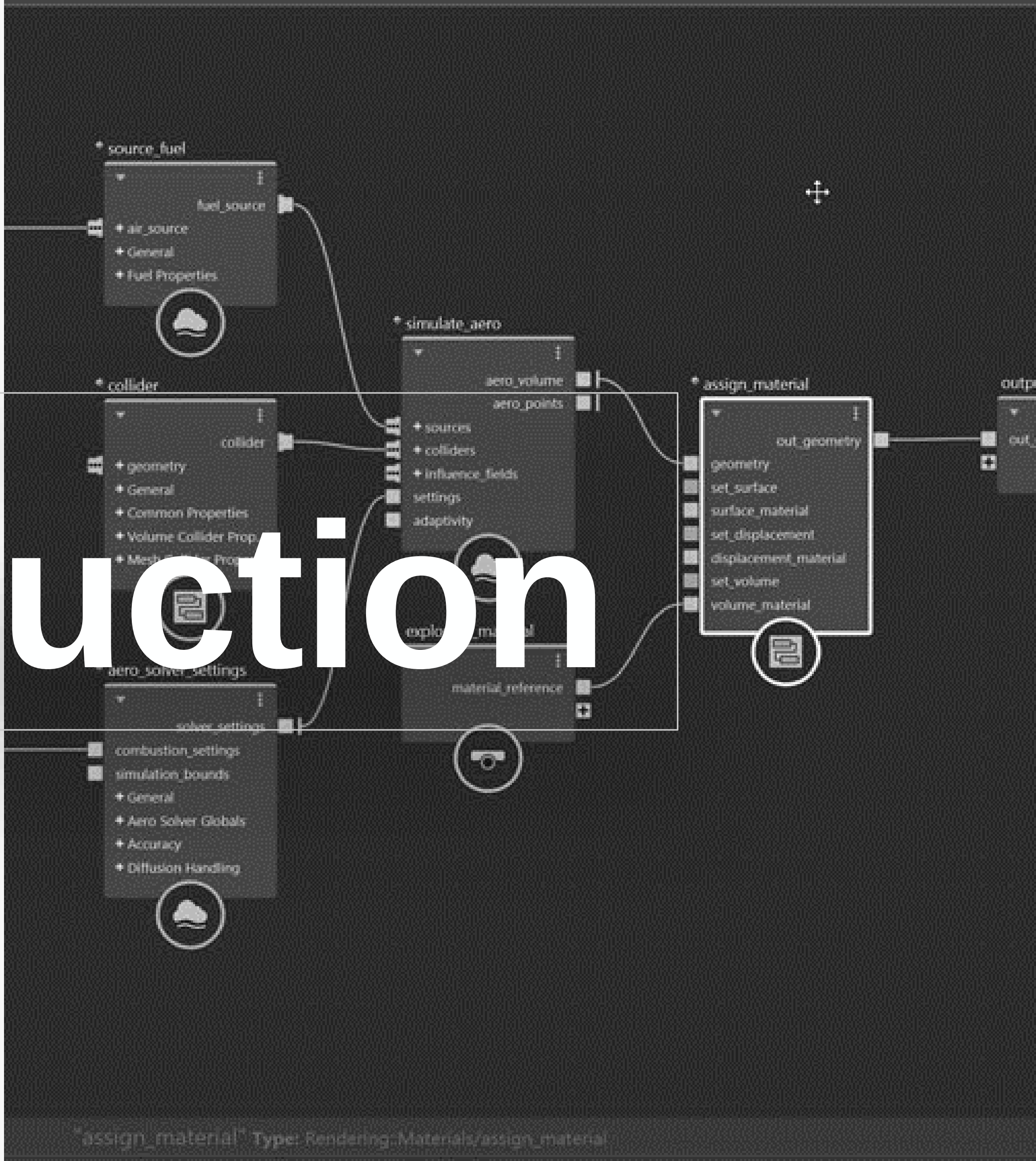
Noureddine AZZOUZA

n.azzouza@univ-dbkm.dz

Course Topics

1. Introduction
2. Data Science Systems
3. Data Science languages
4. Data Science Tools
   - 4.1 Collect & Store
   - 4.2 Transform
   - 4.3 Model
   - 4.3 Visualize & other

Introduction



"assign_material" Type: Rendering::Materials/assign_material

Introduction

- ✓ Data Science is a new field. The subject such as Statistics, Mathematics, Artificial Intelligence, Machine Learning and Data Mining became an integral part of Data Science.
- ✓ The open-source tools were rejected by International Business Machines Corporation (IBM), Microsoft (MS), Systems Applications and Products (SAP), and Oracle.
- ✓ But open-source tools are essential for all bigger, smaller companies and academic institutions nowadays.
- ✓ various tools of Data Science.
- ✓ benefits, challenges and applications of the Data Science tools for researchers/user to decide which tools are better for their need.

Data Science Systems

Linux

- ✓ ***Linux***, is a free, open-source operating system.
- ✓ All most all of distributed software are offered with Linux distributions.

Basics

- ✓ Login to a Linux server
- ✓ Navigation with basic Linux shell commands
- ✓ File Manipulation and transformation
- ✓ Use an editor within a Linux server.

Login to a Linux server

- ✓ ***Windows user***, you need an SSH client
  - ✓ Windows 10+: Windows Terminal
  - ✓ Other Windows systems **PuTTY** or **Xshell**
- ✓ ***Mac or Linux user***, use the **system Terminal** to login to the server
  - ✓ **ssh -p 26506 teacher01@hz-t3.matpool.com**
- ✓ where 26506 is your remote server's SSH port, teacher01 is your remote server's username, hz-t3.matpool.com is your remote server's address.

Run a remote Jupyter notebook using ssh tunne

- ✓ Login to a remote server
  - ▪ **ssh -p 26506 teacher01@hz-t3.matpool.com**
- ✓ Start a **Jupyter notebook** on the server
  - ▪ **jupyter notebook --port 8888**
- ✓ Also copy the string : http://localhost:8888/?token=a373718460df0437c443eeadb7250e7793d6524f80f129cc printed on the terminal.
- ✓ Start a terminal on your local machine and run:
  - ▪ **ssh -p 26506 -L 8008:127.0.0.1:8888 teacher01@hz-t3.matpool.com**
- ✓ where 8008:127.0.0.1:8888 means forwarding your remote server's 8888 prot to your local machine's port 8008.
- ✓ Open your bowser and connect to the remote Jupyter Notebook server locally with the link (replace 8888 with 8008 from your saved string)
- ✓ http://localhost:8008/?token=a373718460df0437c443eeadb7250e7793d6524f80f129cc

Navigation with basic linux shell commands

- ✓ echo $HOME
- ✓ Whoami
- ✓ Pwd
- ✓ ls
- ✓ ls –htla
- ✓ cat /etc/passwd

File manipulation

- ✓ touch lifeng.txt # create a file
- ✓ ls -htla lifeng.txt
- ✓ mkdir hello_linux
- ✓ ls -htla .
- ✓ cd hello_linux # change directory to hello_linux
- ✓ rm hello.py
- ✓ cd .. # swtich to parent directory
- ✓ mv hello_linux goodbye_linux
- ✓ rm -rf goodbye_linux

Use an editor within a linux server

- ✓ **nano** : simple to use
- ✓ **vim** : take a little time to learn
- ✓ **emacs** : steady learning curve

Need help of a command?

- ✓ We take mkdir as an example
- ✓ **man mkdir** : shows the standard manual of Linux command
- ✓ **mkdir --help :** prints short help for the command

Linux pipelines

- ✓ **cat** – Concatenate files
- ✓ **sort** – Sort lines of text
- ✓ **uniq** – Report or omit repeated lines
- ✓ **grep** – Print lines matching a pattern
- ✓ **wc** – Print newline, word, and byte counts for each file
- ✓ **head** – Output the first part of a file
- ✓ **tail** – Output the last part of a file
- ✓ **tee** – Read from standard input and write to standard output and files

Upload and download data

- ✓ The simplest way is to use a graphical tool such as **FileZilla**.

- ✓ When you are more comfortable with commandline tools, you could try **scp** or **rsync**

Data Science

Languages

Recommended languages in Data Science!

- ✓ There is a dizzying amount of choice when it comes to programming languages. Each has it's own strengths and weaknesses and there is no one right answer to the question of which one you should learn first.
- ✓ The answer to that question depends largely on your needs, the problems you are trying to solve, and who you are solving them for.
- ✓ Python, R, and SQL are the languages that we recommend you consider first and foremost.

Data Science Languages

- ✓ There are so many others that have their own strengths and features. Scala, Java, C++, and Julia are some of the most popular.
- ✓ Javascript, PHP, Go, Ruby, and Visual Basic all have their own unique use cases as well.

Python

- ✓ Python is a powerhouse language. It is by far the most popular programming language for data science.
- ✓ According to the 2019 Kaggle Data Science and Machine Learning Survey, 75% of the over 10,000 respondents from around the world reported that they use Python on a regular basis.
- ✓ When asked which language an aspiring data scientist should learn first, most data scientists say Python.
- ✓ Python is great for you because it uses clear, readable syntax.
- ✓ You can do many of the things you are used to doing in other programming languages but with Python you can do it with less code.

Python

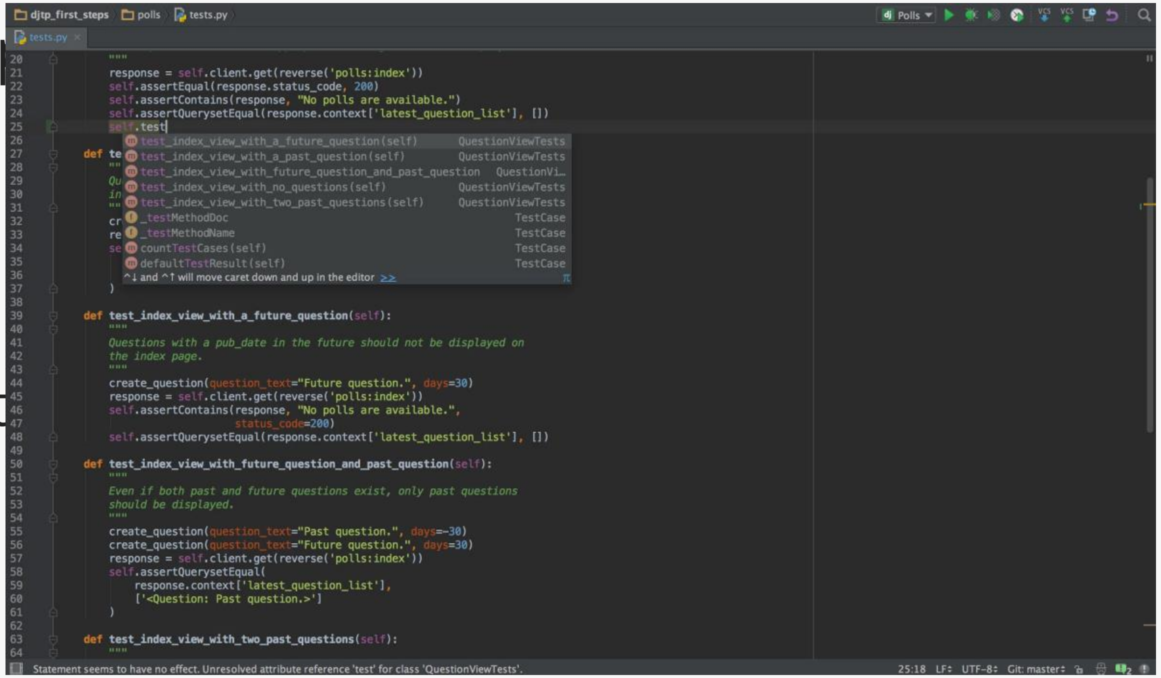
- ✓ If you want to learn to program, it’s also a great starter language because of the huge global community and wealth of documentation.
- ✓ Python is useful for many situations, including data science, AI and machine learning, web development, and IoT devices like the Raspberry Pi.
- ✓ Large organizations that use Python heavily include IBM, Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify, and Reddit.
- ✓ Python is a powerful general-purpose programming language that can do a lot of things. It is widely supported by a global community and shepherded by the Python Software Foundation.

Python for data science

- ✓ It has a large, standard library that provides tools suited to many different tasks, including but not limited to databases, automation, web scraping, text processing, image processing, machine learning, and data analytics.
- ✓ For data science, you can use Python's scientific computing libraries such as Pandas, NumPy, SciPy, and Matplotlib.
- ✓ For artificial intelligence, it has TensorFlow, PyTorch, Keras, and Scikit-learn.
- ✓ Python can also be used for Natural Language Processing (NLP) using the Natural Language Toolkit (NLTK)

Python IDE : PyCharm

- ➢ PyCharm is a widely used Python IDE created by JetBrains
- ➢ This IDE is suitable for professional developers and facilitates the development of large Python projects
  - ✓ Support for JavaScript, CSS, and Ty
  - ✓ Smart code navigation
  - ✓ Quick and safe code refactoring
  - ✓ Support features like accessing dat
  - ✓ Its a great Python IDE for Windows



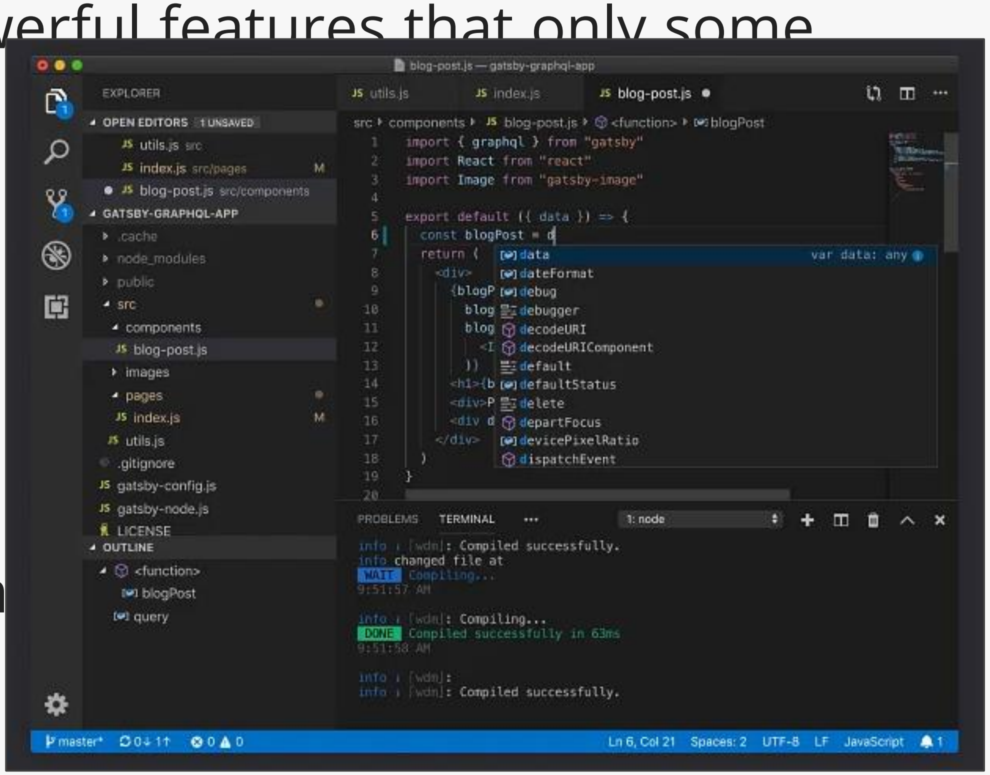
```
        """
        response = self.client.get(reverse('polls:index'))
        self.assertEqual(response.status_code, 200)
        self.assertContains(response, "No polls are available.")
        self.assertQuerysetEqual(response.context['latest_question_list'], [])
        self.test

    def te
        """
        Qu
        in
        """
        cr
        re
        se

        )

    def test_index_view_with_a_future_question(self):
        """
        Questions with a pub_date in the future should not be displayed on
        the index page.
        """
        create_question(question_text="Future question.", days=30)
        response = self.client.get(reverse('polls:index'))
        self.assertContains(response, "No polls are available.",
                            status_code=200)
        self.assertQuerysetEqual(response.context['latest_question_list'], [])

    def test_index_view_with_future_question_and_past_question(self):
        """
        Even if both past and future questions exist, only past questions
        should be displayed.
        """
        create_question(question_text="Past question.", days=-30)
        create_question(question_text="Future question.", days=30)
        response = self.client.get(reverse('polls:index'))
        self.assertQuerysetEqual(
            response.context['latest_question_list'],
            ['<Question: Past question.>']
        )

    def test_index_view_with_two_past_questions(self):
        """
```

```
m test_index_view_with_a_future_question(self)        QuestionViewTests
m test_index_view_with_a_past_question(self)          QuestionViewTests
m test_index_view_with_future_question_and_past_question  QuestionVi…
m test_index_view_with_no_questions(self)             QuestionViewTests
m test_index_view_with_two_past_questions(self)       QuestionViewTests
f _testMethodDoc                                              TestCase
f _testMethodName                                             TestCase
m countTestCases(self)                                        TestCase
m defaultTestResult(self)                                     TestCase
^↓ and ^↑ will move caret down and up in the editor >>              π
```

Statement seems to have no effect. Unresolved attribute reference 'test' for class 'QuestionViewTests'.

Python IDE : Visual Studio Code

- ➢ VS Code is an open-source (and free) IDE created by Microsoft. It finds great use in Python development
- ➢ VS Code is lightweight and comes with powerful features that only some of the paid IDEs offer
  - ✓ One of the best smart code completion
  - ✓ Git integration
  - ✓ Code debugging within the editor
  - ✓ It provides an extension to add addition



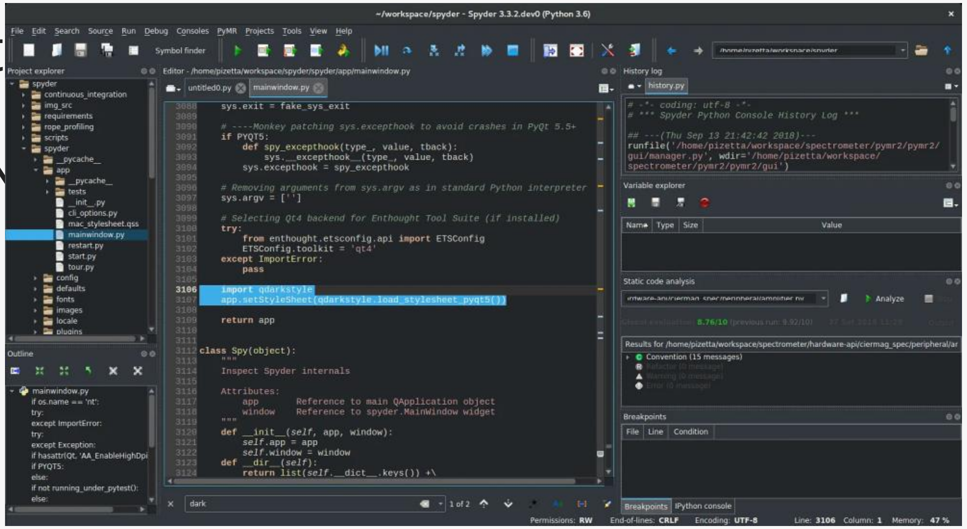
```
import { graphql } from "gatsby"
import React from "react"
import Image from "gatsby-image"

export default ({ data }) => {
  const blogPost = d
  return (
    <div>
      {blogP
        blog
        blog
          <I
        })
      <h1>{b
      <div>P
      <div d
    </div>
  )
}
```

```
info ｉ [wdm]: Compiled successfully.
info changed file at
 WAIT  Compiling...
9:51:57 AM

info ｉ [wdm]: Compiling...
 DONE  Compiled successfully in 63ms
9:51:58 AM

info ｉ [wdm]:
info ｉ [wdm]: Compiled successfully.
```

Data Science Languages

Python IDE : Spyder

- Spyder is an open-source IDE most commonly used for scientific deve
- Spyder comes with Anaconda distribution, which is popular for data science and machine learning
  - ✓ Support for automatic code completion and splitting
  - ✓ Supports plotting different types of chart
  - ✓ Integration of data science libraries like N
  - ✓ Its a great Python IDE for Windows


~/workspace/spyder - Spyder 3.3.2.dev0 (Python 3.6)
File Edit Search Source Run Debug Consoles PyMR Projects Tools View Help
Symbol finder
Project explorer
Editor - /home/pizetta/workspace/spyder/spyder/app/mainwindow.py
untitled0.py
mainwindow.py
History log
history.py
Variable explorer
Name Type Size Value
Static code analysis
Analyze
Outline
Breakpoints
File Line Condition
Breakpoints IPython console
dark
1 of 2
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 3106 Column: 1 Memory: 47 %

```
    sys.exit = fake_sys_exit

    # ----Monkey patching sys.excepthook to avoid crashes in PyQt 5.5+
    if PYQT5:
        def spy_excepthook(type_, value, tback):
            sys.__excepthook__(type_, value, tback)
        sys.excepthook = spy_excepthook

    # Removing arguments from sys.argv as in standard Python interpreter
    sys.argv = ['']

    # Selecting Qt4 backend for Enthought Tool Suite (if installed)
    try:
        from enthought.etsconfig.api import ETSConfig
        ETSConfig.toolkit = 'qt4'
    except ImportError:
        pass

    import qdarkstyle
    app.setStyleSheet(qdarkstyle.load_stylesheet_pyqt5())

    return app


class Spy(object):
    """
    Inspect Spyder internals

    Attributes:
        app       Reference to main QApplication object
        window    Reference to spyder.MainWindow widget
    """
    def __init__(self, app, window):
        self.app = app
        self.window = window
    def __dir__(self):
        return list(self.__dict__.keys()) +\
```

```
# -*- coding: utf-8 -*-
# *** Spyder Python Console History Log ***

## ---(Thu Sep 13 21:42:42 2018)---
runfile('/home/pizetta/workspace/spectrometer/pymr2/pymr2/gui/manager.py', wdir='/home/pizetta/workspace/spectrometer/pymr2/pymr2/gui')
```

R

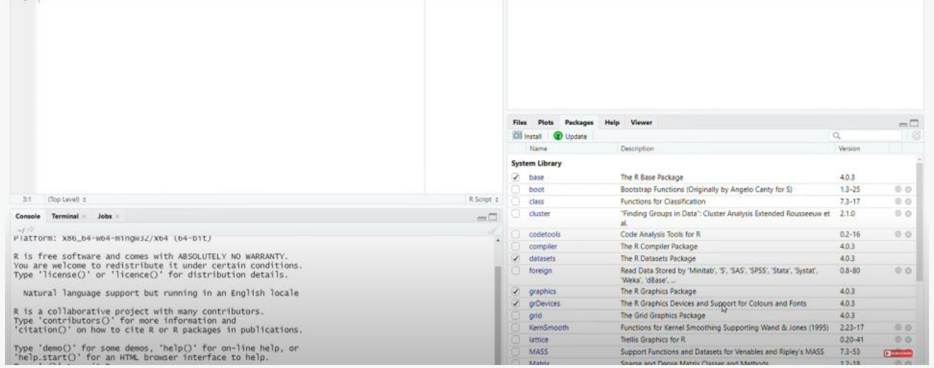
- ✓ R is a programming language and an analytics tool that was developed in 1993 by Robert Gentleman and Ross Ihaka at the University of Auckland
- ✓ It is extensively used by Software Programmers, Statisticians, Data Scientists, and Data Miners. It is one of the most popular Data analytics tools used in Data Analytics and Business Analytics.
- ✓ It has numerous applications in domains like healthcare, academics, consulting, finance, media, and many more.
- ✓ Its vast applicability in Statistics, Data Visualization, and Machine Learning have given rise to the demand for certified trained professionals in R.

R best features

- ✓ It is a free and open-source programming language
- ✓ It has cross-platform interoperability on Windows, Linux, and Mac.
- ✓ It uses an interpreter instead of a compiler (makes development of code easier
- ✓ It effectively associates different databases from Microsoft Excel, as well as, Microsoft Access, MySQL, SQLite, Oracle, etc.
- ✓ It is a flexible language that bridges the gap between Software Development and Data Analysis.
- ✓ It provides a wide variety of packages with a diversity of codes, functions, and features tailored for data analysis, statistical modeling, visualization, Machine Learning, and importing and manipulating data.

Data Science Languages

Data Science Languages



```
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
```

	Name	Description	Version
	System Library		
☑	base	The R Base Package	4.0.3
□	boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-25
□	class	Functions for Classification	7.3-17
□	cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.1.0
□	codetools	Code Analysis Tools for R	0.2-16
□	compiler	The R Compiler Package	4.0.3
☑	datasets	The R Datasets Package	4.0.3
□	foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...	0.8-80
☑	graphics	The R Graphics Package	4.0.3
☑	grDevices	The R Graphics Devices and Support for Colours and Fonts	4.0.3
□	grid	The Grid Graphics Package	4.0.3
□	KernSmooth	Functions for Kernel Smoothing Supporting Wand & Jones (1995)	2.23-17
□	lattice	Trellis Graphics for R	0.20-41
□	MASS	Support Functions and Datasets for Venables and Ripley's MASS	7.3-53
□	Matrix	Sparse and Dense Matrix Classes and Methods	1.2-18

R IDE : R Studio

- ✓ R Studio is an integrated development environment(IDE) for R.
- ✓ A syntax-highlighting editor that supports direct code execution and where you keep a record of your work;
- ✓ a console for typing R commands; a workspace and history tab that shows the list of R objects you created during your R session and shows the history of all previous commands; and a Plots, Files, Packages, and Help tab for showing files in your working directory; history of plots you have created and allow for exporting plots to PDF or images files; external R packages available on your local computer and help on R resources, RStudio support, packages, and more.

SQL

- ✓ SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.
- ✓ SQL is the standard language for Relational Database System.
- ✓ All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.
- ✓ Also, they are using different dialects, such as − MS SQL Server using T-SQL, Oracle using PL/SQL, MS Access version of SQL is called JET SQL ... etc.

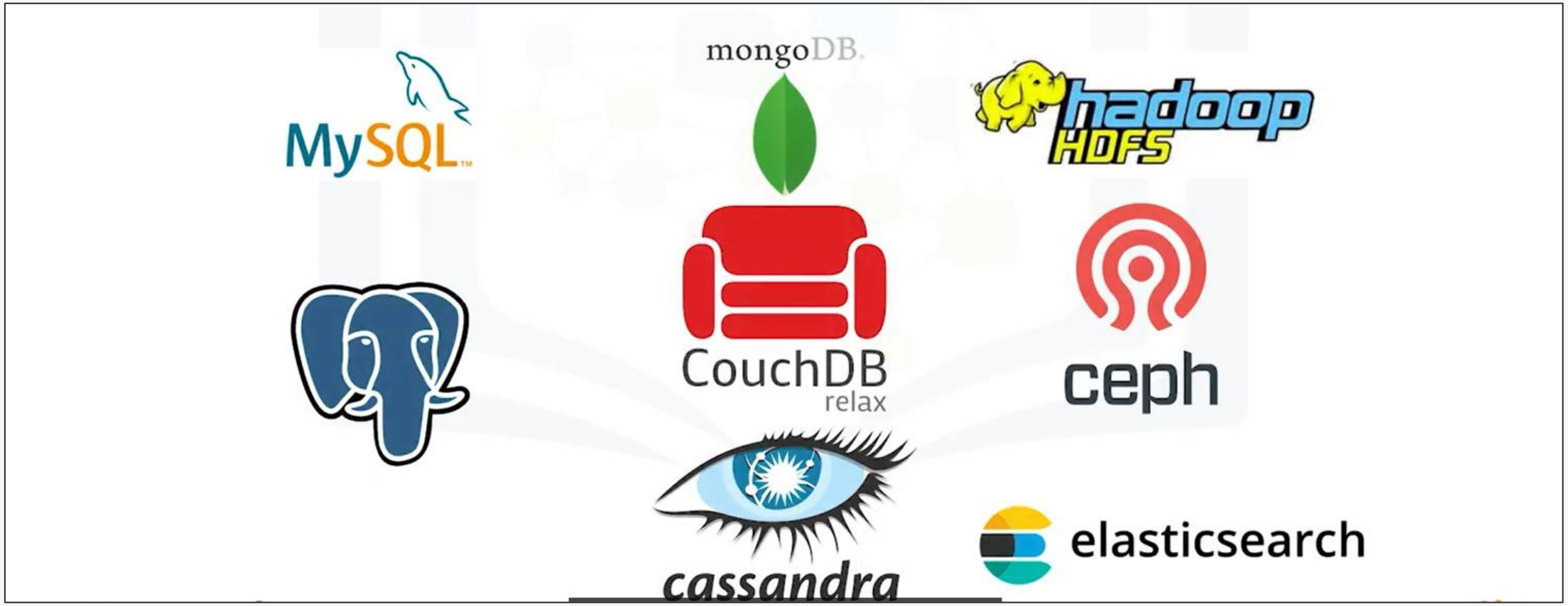
SQL advantages

- ✓ Knowing SQL will help you get jobs as a business and data analyst and is a must in data engineering and data science.
- ✓ When performing operations with SQL the data is accessed directly (without any need to copy it beforehand).
- ✓ This can considerably speed up workflow executions.
- ✓ SQL is the interpreter between you and the database.
- ✓ SQL is a ANSI standard, which means if you learn SQL and use it with one database you will be able to easily apply your SQL knowledge with many other databases.

Data Management

Tools

Data Management Tools



MySQL / MongoDB

- ✓ SQL is used for storing data in normalised form. These tools are dealing with large datasets and normalise the unstructured data in sequence and categorical. These tools have a different type of keys.
- ✓ MySQL used RDBMS. MySQL is a relational database and client-server system in which memory allocation system is thread-based and provide the compatibility of multiple commands
- ✓ MongoDB is schema is a schema-less database and dynamically load balance the queries. It is also using aggregation tools for performing data processing in the pipeline.

NoSQL

- ✓ These are databases used for dealing with structured as well as unstructured data.
- ✓ Because of its high speed and flexibility, NoSQL is the most useful tool in DS.
- ✓ No-SQL deals with large datasets like picture video machine to machine communications and convert them in a structured form.
- ✓ No SQL database easily maintains the unstructured, structured, and semi-structured data.

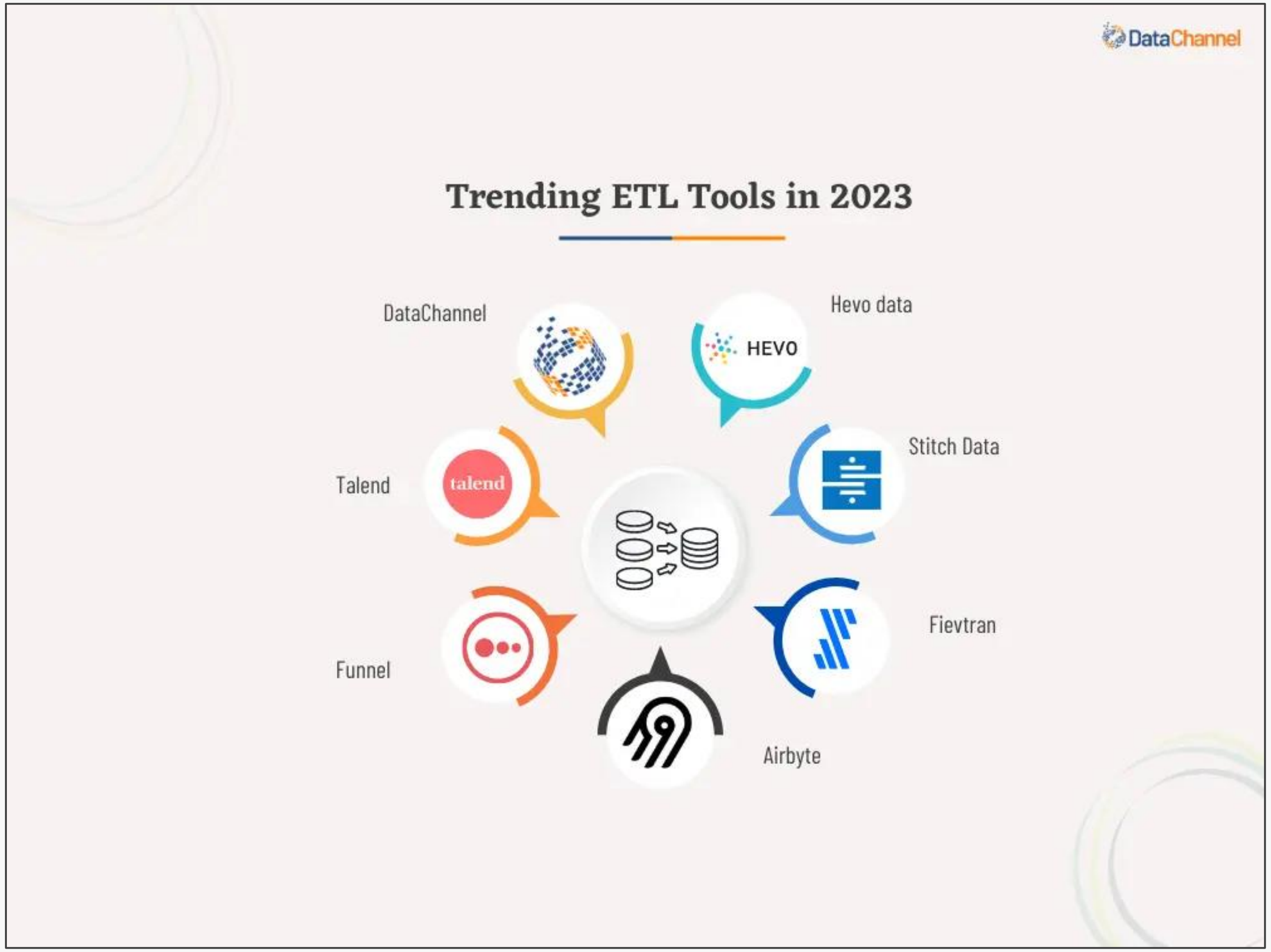
Hadoop

- ✓ Hadoop is open-source software framework that was used for storing and processing the big data. It was developed in 2005 to support for Nutch search engine project. Hadoop is written in java.
- ✓ It provides different frameworks that are used for processing big data.
- ✓ Successfully live massive knowledge in thousands of Hadoop collections
- ✓ It uses the Hadoop Distributed filing system (HDFS) to store knowledge that distributes massive amounts of knowledge across multiple nodes for a distributed, compatible pc.
- ✓ It provides practicality to different processing modules, like Hadoop MapReduce, Hadoop YARN, and so on.

Extract, Transform, Load

Tools

Data ETL Tools



Data ETL Tools

ETL Definition

- ETL is a common approach to integrating data and organizing data stacks. A typical ETL process comprises the following stages:

✓ **Extracting** data from sources

✓ **Transforming** data into data models

✓ **Loading** data into data warehouses

ETL : Extraction

- Extraction is the first phase of “extract, transform, load.” Data is collected from one or more data sources. It is then held in temporary storage, where the next two steps are executed.

- During extraction, validation rules are applied. This tests whether data meets the requirements of its destination. Data that fails validation is rejected and doesn’t continue to the next step.

ETL : Transform

- In the transformation phase, data is processed to make its values and structure conform consistently with its intended use case. The goal of transformation is to make all data fit within a uniform schema before it moves on to the last step.

- Typical transformations include aggregators, data masking, expression, joiner, filter, lookup, rank, router, union, XML, Normalizer, H2R, R2H and web service. This helps to normalize, standardize and filter data. It also makes the data fit for consumption for analytics, business functions and other downstream activities.

ETL : Load

- Finally, the load phase moves the transformed data into a permanent target system. This could be a target database, data warehouse, data store, data hub or data lake — on-premises or in the cloud. Once all the data has been loaded, the process is complete.

- Many organizations regularly perform this process to keep their data warehouse updated.

Traditional ETL vs. Cloud ETL

- **Traditional ETL** : Traditional or legacy ETL is designed for data located and managed completely on-premises by an experienced in-house IT team. Their job is to create and manage in-house data pipelines and databases.
  - As a process, it generally relies on batch processing sessions that allow data to be moved in scheduled batches.
  - Real-time analysis can be hard to achieve.
  - To extract the necessary data analytics, IT teams often create complicated, labor-intensive customizations and exact quality control.
  - ETL systems can't easily handle spikes in large data volumes. That often forces organizations to choose between detailed data or fast performance.

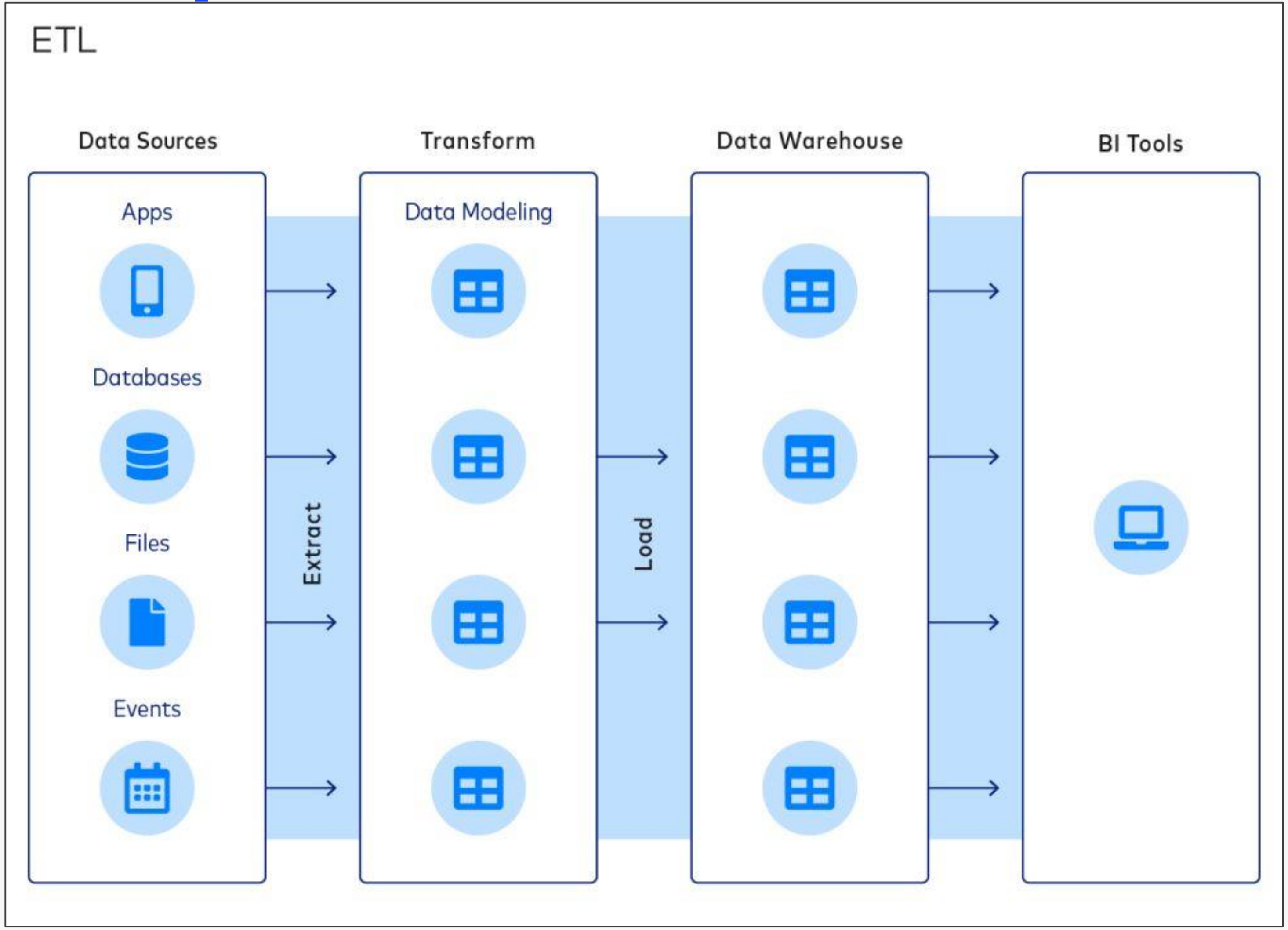
Traditional ETL vs. Cloud ETL

- **Cloud ETL**
  - Cloud, or modern, ETL extracts both structured and unstructured data from any data source type.
  - The data could be in on-premises or cloud data warehouses.
  - It then consolidates and transforms that data. Next, it loads the data into a centralized location where it can be accessed on demand.
  - Cloud ETL is often used to make high-volume data readily available for analysts, engineers and decision makers across a variety of use cases.

Data ETL Tools

Data ETL Tools

ETL Pipeline



Types of ETL pipelines

- **Batch processing pipelines**
  - used for traditional analytics and BI use cases where data is periodically collected, transformed and moved to a cloud data warehouse.
  - Users can quickly deploy high-volume data from siloed sources into a cloud data lake or data warehouse.
  - They can then schedule jobs for processing data with minimal human intervention. With ETL in batch processing, data is collected and stored during an event known as a "batch window."
  - Batches are used to more efficiently manage large amounts of data and repetitive tasks.

Types of ETL pipelines

- **Real-time processing pipelines**
  - Real-time data pipelines enable users to ingest structured and unstructured data from a range of streaming sources. These include IoT, connected devices, social media feeds, sensor data and mobile applications. A high-throughput messaging system ensures the data is captured accurately.
  - Data transformation is performed using a real-time processing engine like Spark streaming. This drives application features like real-time analytics, GPS location tracking, fraud detection, predictive maintenance, targeted marketing campaigns and proactive customer care.

ETL Process

- The ETL paradigm is popular because it allows companies to reduce the size of their data warehouses, which can save on computation, storage, and bandwidth costs.
- However, these cost savings are becoming less important as these constraints disappear.
- As a result, ELT (Extract, Load, Transform) is becoming more popular.
- In the ELT process, data is loaded to a destination after extraction, and transformation is the final step in the process.

ETL Tools Key considerations

- The extent of data integration.
- Level of customizability.
- Cost structure.
- The level of automation provided
- The level of security and compliance
- The performance and reliability of the tool.

Apache Airflow

- Apache Airflow is an open-source platform to programmatically author, schedule, and monitor workflows. The platform features a web-based user interface and a command-line interface for managing and triggering workflows.
- Workflows are defined using directed acyclic graphs (DAGs), which allow for clear visualization and management of tasks and dependencies. Airflow also integrates with other tools commonly used in data engineering and data science, such as Apache Spark and Pandas.
- Companies using Airflow can benefit from its ability to scale and manage complex workflows, as well as its active open-source community and extensive documentation.

Qlik Compose

- Qlik Compose is a data warehousing solution that automatically designs data warehouses and generates ETL code. This tool automates tedious and error-prone ETL development and maintenance. This shortens the lead time of data warehousing projects.
- To do so, Qlik Compose runs the auto-generated code, which loads data from sources and moves them to their data warehouses. Such workflows can be designed and scheduled using the Workflow Designer and Scheduler.
- Qlik Compose also comes with the ability to validate the data and ensure data quality.

Airbyte

- Airbyte is a leading open-source ELT platform. Airbyte offers the largest catalog of data connectors—350 and growing—and has 40,000 data engineers using it as of June 2023.
- Airbyte integrates with dbt for its data transformation and Airflow / Prefect / Dagster for orchestration. It has an easy-to-use user interface and has an API and Terraform Provider available.
- Airbyte differentiates itself by its open-sourceness; it takes 20 minutes to create a new connector with their no-code connector builder, and you can edit any off-the-shelf connector, given you have access to their code. In addition to its open-source version, Airbyte offers both a cloud-hosted (Airbyte Cloud) and a paid self-hosted version (Airbyte Enterprise) for when

Informatica PowerCenter

- Informatica has many products that focus on data integration. However, the Informatica PowerCenter stands out because of its data integration capabilities
- ETL (Extract Transform Load) data collection tool for properties.
- It helps to extract data from various sources, modify it and process it according to the needs of the business and ultimately upload or submit it to the repository.
- Provides distributed processing support, grid computing, adaptive load balancing, dynamic partitioning, and pushdown efficiency.

Data Integration Tools

Data Integration Tools

Apache Airflow
APACHE kafka
Spark SQL
Kubeflow
APACHE nifi
Node-RED

Definition

- ✓ data integration is the process of combining data from different sources into a single, unified view.
- ✓ data integration is the process of moving data between databases — internal, external, or both. Here, databases include production DBs, data warehouses (DWs) as well as third-party tools and systems that generate and store data.
- ✓ all integration tools use the same underlying technology — APIs.

Data Integration Tools

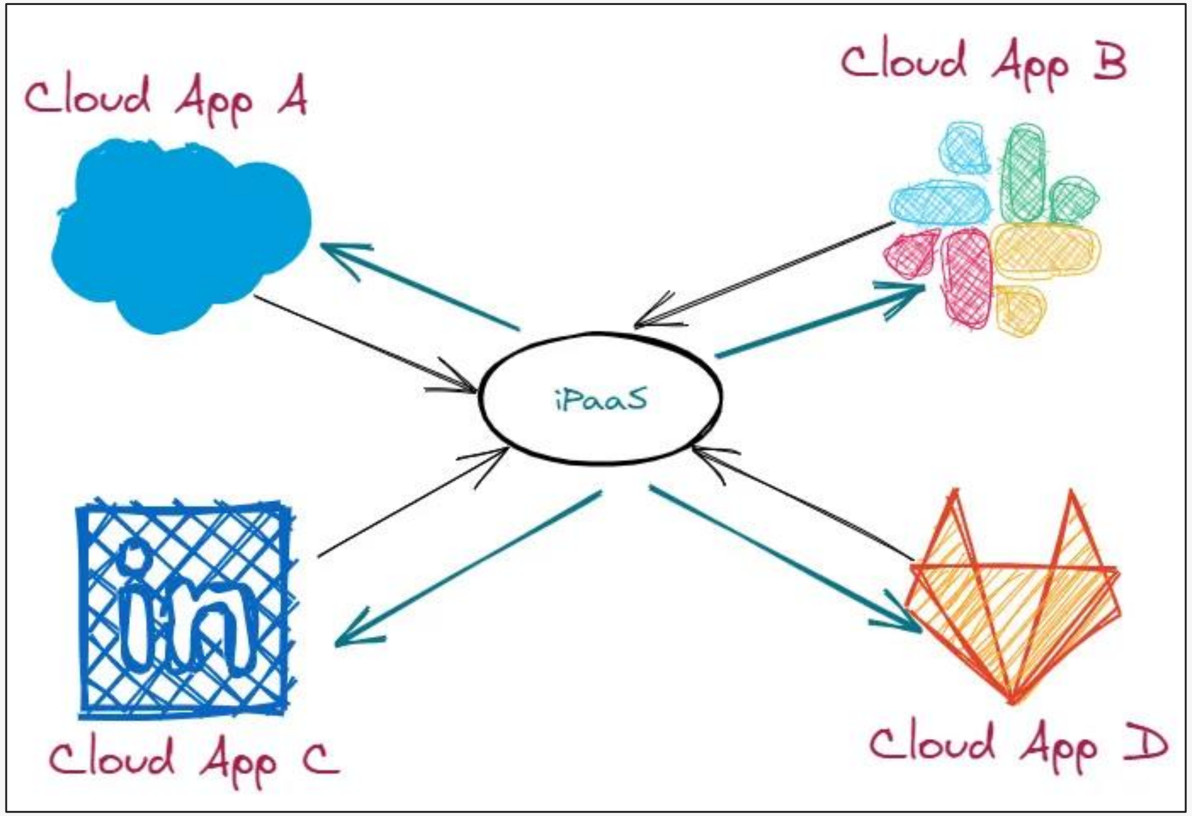
- ✓ iPaaS or Integration Platform as a Service
- ✓ CDP or Customer Data Platform
- ✓ ETL or Extract, Transform and Load
- ✓ ELT or Extract, Load, and Transform
- ✓ Reverse ETL

iPaaS or Integration Platform as a Service

- ✓ iPaaS was allegedly coined in 2008 by Boomi, an enterprise iPaaS vendor. Since then, iPaaS has seen wide adoption and has resulted in a proliferation of companies offering iPaaS solutions in various shapes and packages
- ✓ perform actions based on a trigger. A trigger is essentially an event taking place in system A that is transmitted to the integration platform (via an API call or a Webhook) which then performs one or more predefined actions.
- ✓ iPaaS solutions can also be used to move data between internal systems where events take place.

iPaaS or Integration Platform as a Service

- ✓ they offer a visual interface to build integrations, enabling business teams to take control of their workflow automation needs
- ✓ Amongst the more popular ones today are **Tray** and **Workato** focused on enterprises and **Zapier**, **Integromat**, and **Automate.io** catering to SMBs.

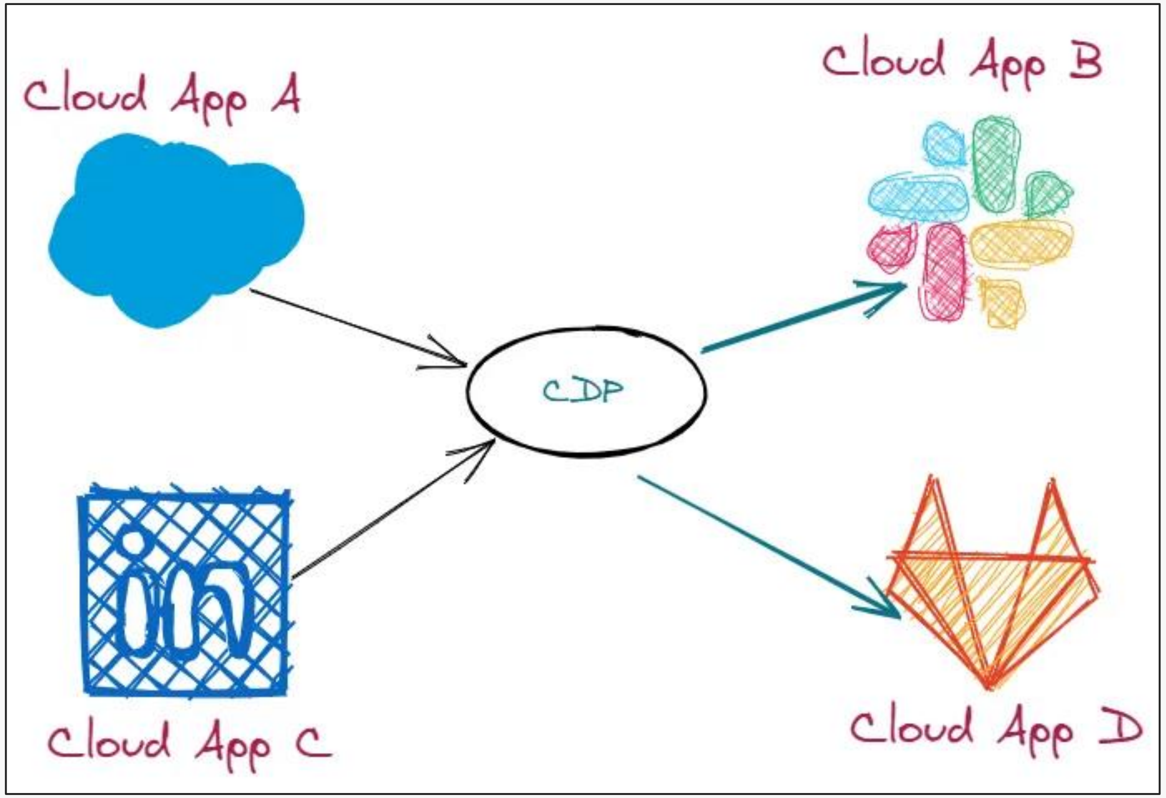


CDP or Customer Data Platform

- ✓ collect and collate customer data from different sources and send that data to different destinations.
- ✓ CDPs also enable data collection via proprietary SDKs and APIs.
- ✓ CDP does a lot more than move data between tools. It enables marketing and growth teams to build segments based on user behaviour and user traits, and sync these segments to third-party tools to deliver personalized experiences

CDP or Customer Data Platform

- ✓ CDP vendors like **Segment**, **mParticle**, **Lytics**, and **Tealium** as well as vertical CDPs like **Amperity**.

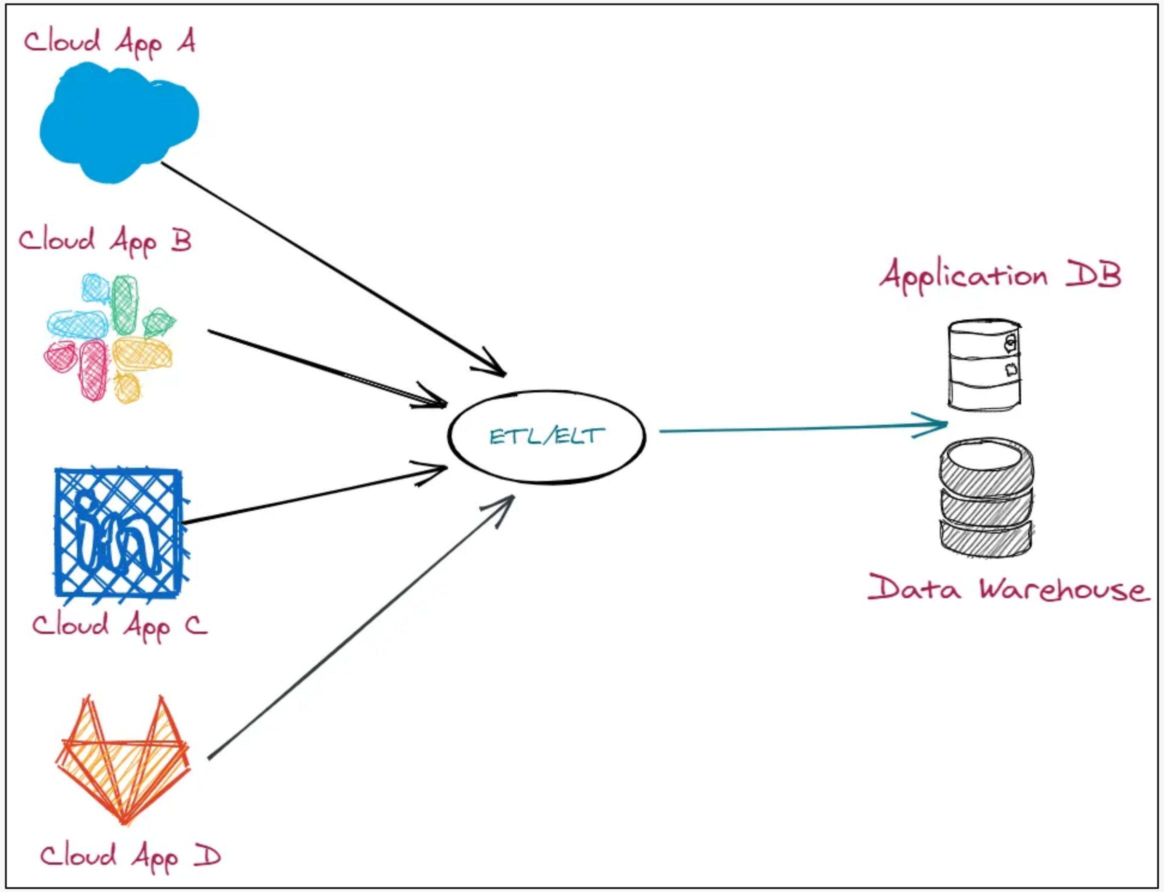


ETL or Extract, Transform and Load

- ✓ ETL is a traditional data integration process date back to the 1970s. However, it was only in the early nineties that Informatica made ETL commonplace in the enterprise.
- ✓ Under the ETL paradigm, data is first extracted from first-party databases and third-party sources (primarily SaaS tools for sales, marketing, and support), transformed to meet the needs of analysts and data scientists, and finally loaded into a Data Warehouse.

ETL or Extract, Transform and Load

- ✓ The transformation is particularly resource-incentive and time-consuming which significantly impacts the time it takes between the extraction and the loading of data

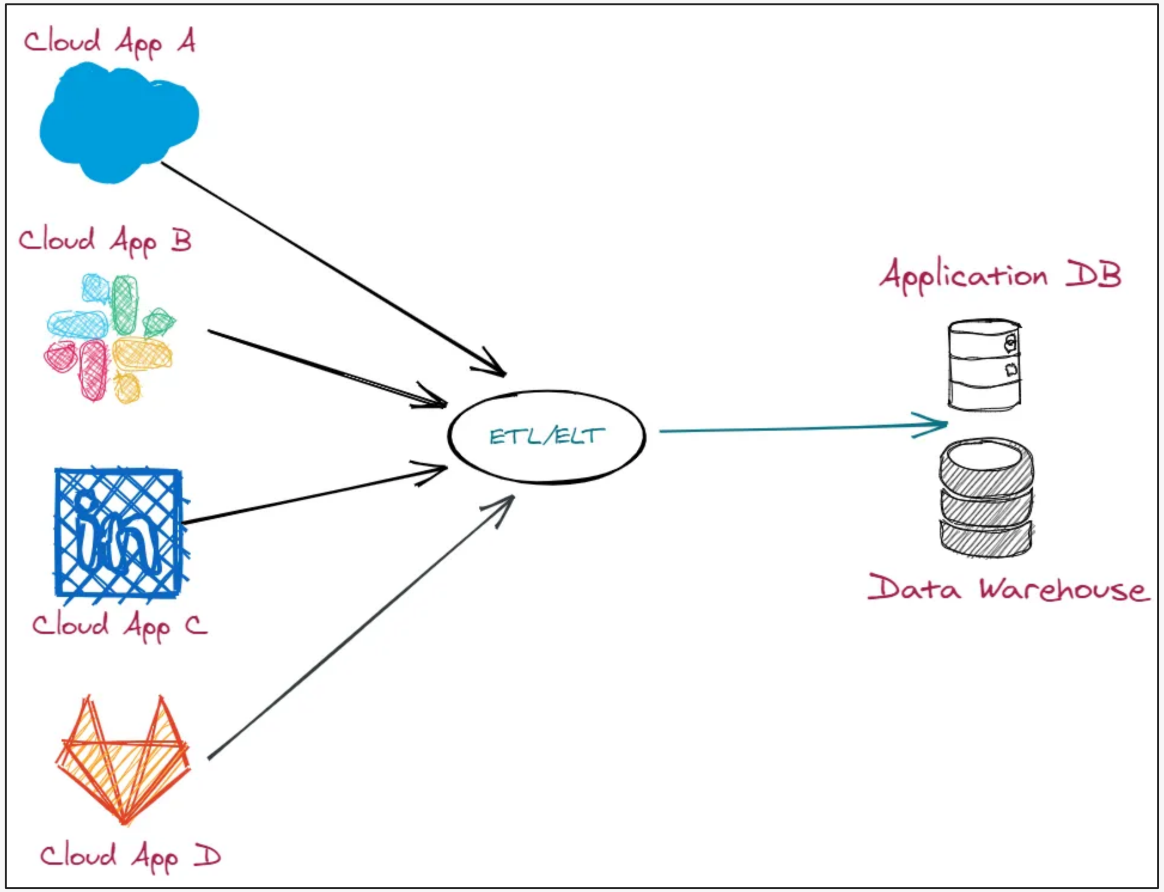


ELT or Extract, Load, and Transform

- ✓ ELT is the modern approach to ETL which is largely being fueled since cloud data warehouses such as **Redshift**, **Snowflake**, and **BigQuery** have become extremely fast and reliable, enabling transformation to take place inside the warehouse itself.
- ✓ data is extracted from source systems and loaded into the warehouse without any transformation taking place. In fact, modern ELT tools don’t even offer in-built transformation capabilities but integrate well with services like dbt that are purpose-built solutions to handle the transformation layer within the data warehouse

ELT or Extract, Load, and Transform

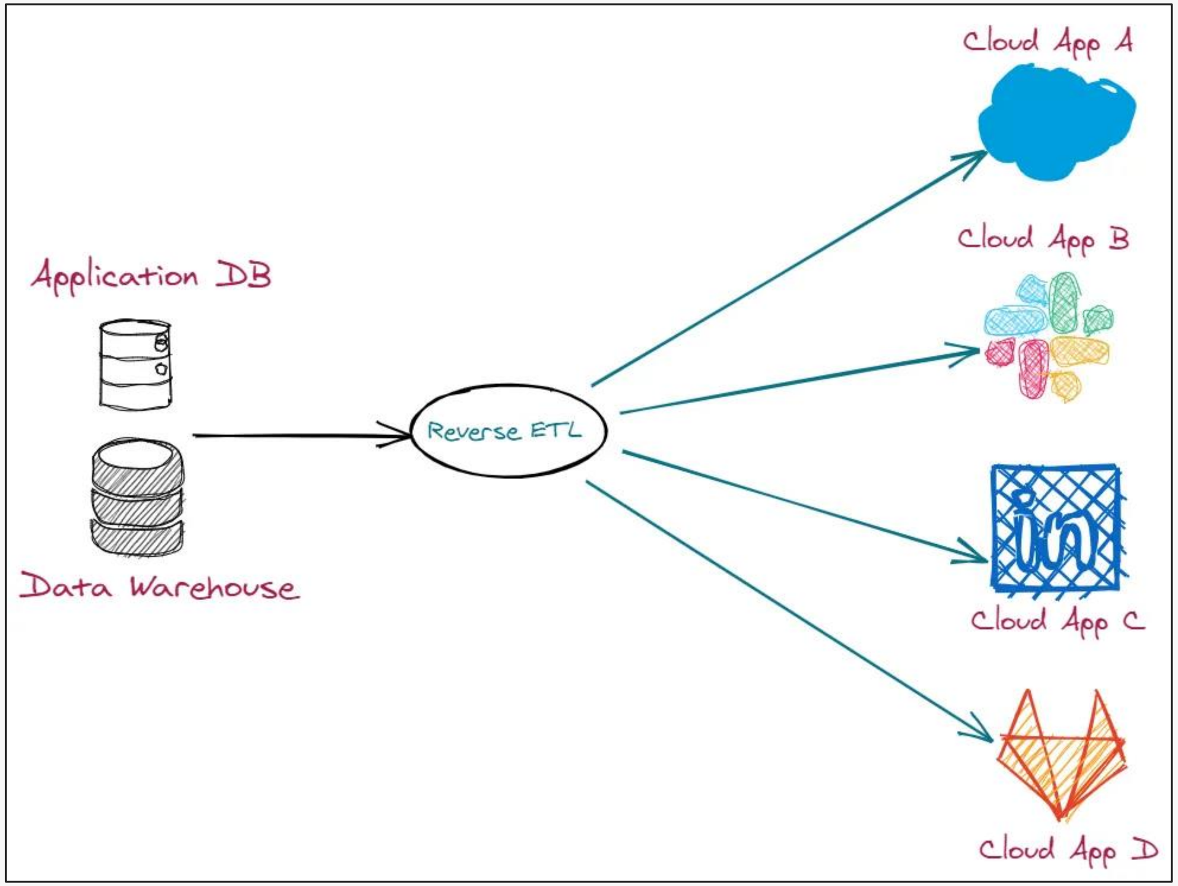
- ✓ ELT is fast, affordable and most importantly, requires no coding, all of which is fueling the shift from ETL to ELT
- ✓ **Fivetran**, **Stitch**, and **Matillion** are companies leading the new ELT paradigm. **Talend**, which is a leading ETL provider, acquired Stitch in 2018 to embrace ELT.



Reverse ETL

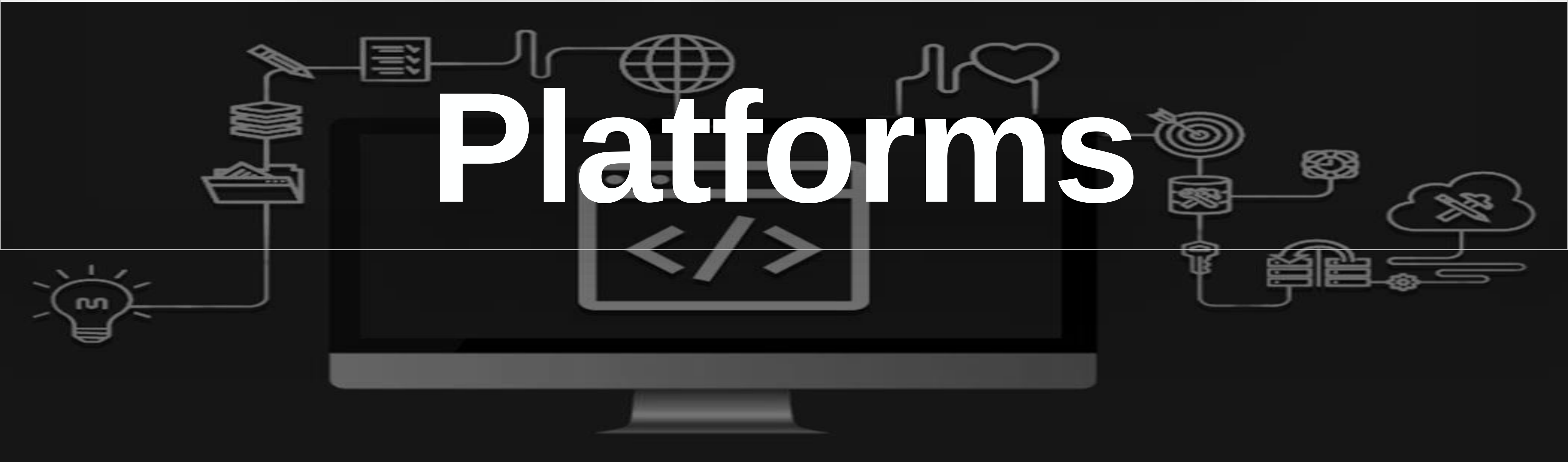
- ✓ the ability to transform data in the warehouse using tools like dbt makes the data warehouse the true source of truth for all types of data
- ✓ A reverse ETL tool like **Hightouch** or **Census** takes care of the following:
  - ➢ Extract data from a warehouse or a database on a regular cadence and load it into sales, marketing, and analytics tools
  - ➢ Trigger a webhook or make an arbitrary API call every time the data changes
  - ➢ Move extracted rows of data to a production database to deliver personalized experiences

Reverse ETL


Cloud App A
Cloud App B
Application DB
Reverse ETL
Data Warehouse
Cloud App C
Cloud App D

Data Science

Platforms



Data Science Platforms

- ✓ A data science platform is software that includes a variety of technologies for machine learning, data science, and other advanced analytics projects.
- ✓ It is important to have a centralized and unified platform so data science teams can collaborate on those projects.
- ✓ A single, integrated platform where a whole team of data scientists works together can lead to better results and, therefore, greater business value.
- ✓ These platforms offer collaborative environments, helping organizations to incorporate data-driven decisions into operational and customer-friendly systems to enhance business outcomes.

Types of Data Science Platforms

- ✓ The data science platform landscape can be overwhelming. There are dozens of products describing themselves using similar language despite addressing different problems for different types of users.
- ✓ We can divide the types of Data Science Platforms into 3 parts.
  - ➢ 1. Automation Tools
  - ➢ 2. Proprietary (Often GUI-driven) Data Science Platforms
  - ➢ 3. Code-first Data Science Platforms

1. Automation Tools

- ✓ These tools help engineers to automate repetitive tasks in data science, including training models, selecting algorithms, and more.
- ✓ These solutions are targeted primarily at non-expert coders or data scientists interested in shortcutting tedious steps and repetitive steps.
- ✓ They help spread data science work by getting non-expert data scientists into the model-building process, offering drag-and-drop interfaces.

2. Proprietary Data Science Platforms

- ✓ Proprietary tools support a lot of use cases, including data science and model building.
- ✓ They provide both drag-and-drop and code interfaces and have a stronghold in big companies and may even offer unique capabilities or algorithms.
- ✓ While these solutions offer a great breadth of functionality, users must leverage proprietary user interfaces or programming languages to express their logic.

3. Code-first Data Science Platforms

- ✓ Code-first Data Science Platforms target data scientists and coders who use statistical programming languages and spend their days in IDEs like Jupyter and Colab, leveraging a mix of open-source and ML packages and tools.
- ✓ These data scientists require the flexibility to use a constantly evolving software and hardware stack to optimize each step of their model lifecycle.
- ✓ These code-first data science platforms orchestrate the necessary infrastructure to accelerate power users' workflows and create a system of record for organizations with hundreds or thousands of models.

Data Science Platforms Features

1. **Integrate multiple data science tools**

   ✓ The most important feature of these platforms is integrating all the tools in one place so that all the work like data cleaning, analysis, modeling, and deployment can be done with ease.

2. **Centralize data resources**

   ✓ Data Science Platforms have a unified location for all work.

3. **Handle very large amounts of structured and unstructured data**

   ✓ They help in the smooth handling of large GBs of data

Data Science Platforms Features

4. Data mining, Data access, gathering, and preparation

- ✓ The platforms provide tools to fasten cleaning and data analysis.

5. No code options

- ✓ Even people with no coding knowledge can work on these platforms with the help of no-code tools

6. GUI Dashboards

- ✓ They have integrated dashboards to help visualize the graphs and results for the clients

Data Science Platforms Features

7. Multiple programming language support

- ✓ Data Science Notebooks come with multiple language support like Python, R, etc

8. Model development and iteration

- ✓ These platforms come with inbuilt tools for model building and training, which does the work in a few lines of code.

Data Science Platforms Features

9. Machine Learning Deep learning

- ✓ It has inbuilt advanced ML and DL libraries like Keras, Pytorch, etc., which makes coding very simple and faster

10. Automated documentation and explainers

- ✓ It comes with automated documentation and code helpers to guide the engineers in the further steps of modeling

Data Science Platforms Features

11. Security

- ✓ Since a lot of people collaborate together, good security services are a must on these platforms.

12. Cloud-based, on-premises, hybrid installations

- ✓ Data Science platforms have cloud-based services infused like google colab for efficient collaboration on cloud without wasting local resources.

Need for a Data Science Platform

1. To Enable Better Teamwork with Data Scientists
2. Help Minimalize Engineering Effort
3. Help to Offload a Number of Low Value Tasks
4. Facilitate Faster Research and Experimentation

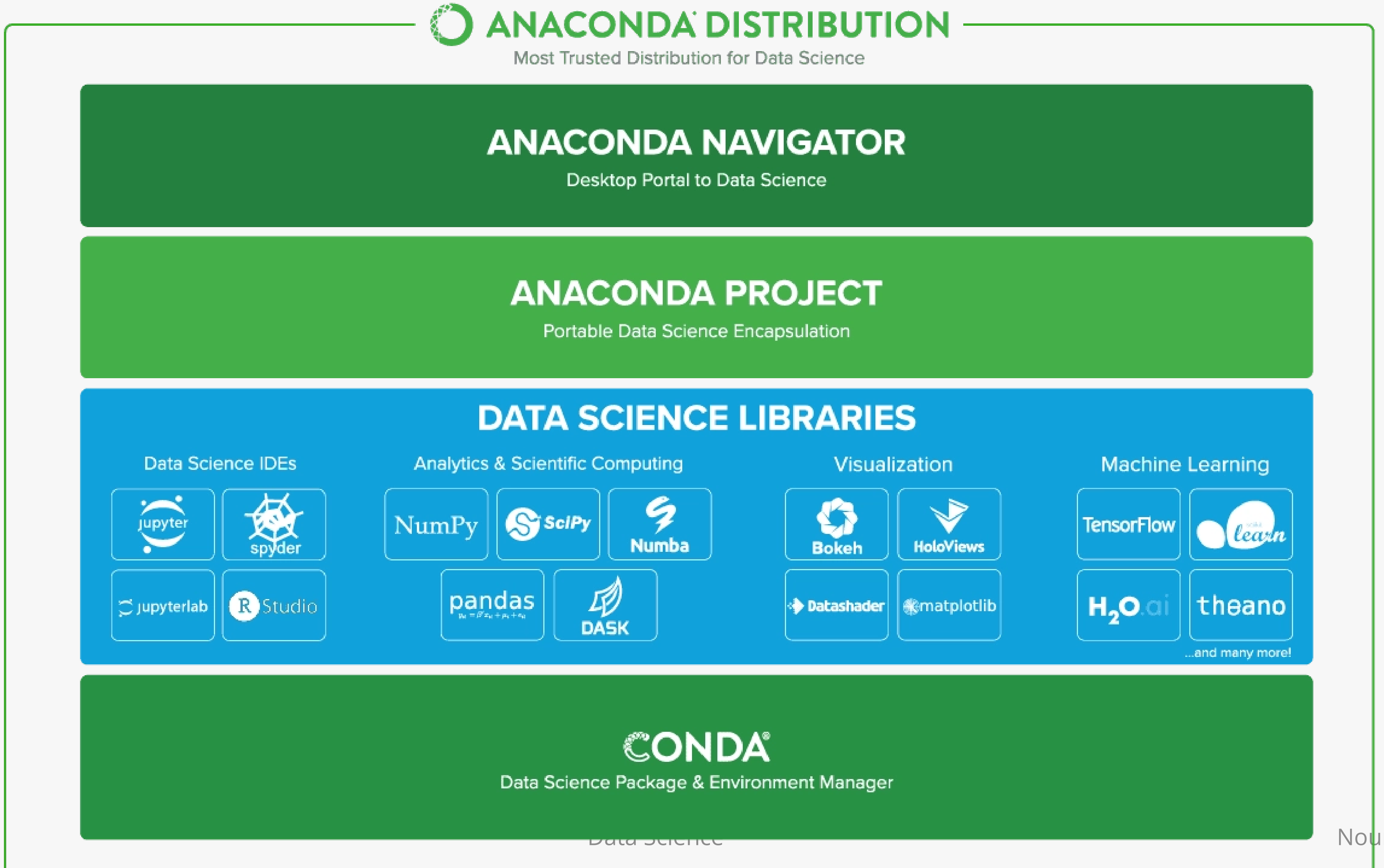
Top Data Science Platforms

1. Anaconda Data Science Platform
2. H2o.ai Platform
3. Data Science on Google cloud platform
4. Data Science on AWS
5. Data science on Microsoft Azure

Anaconda Data Science Platform

- ✓ Anaconda offers the easiest way to perform Python/R DS and ML on a single machine. Navigators can search packages on an anaconda cloud or local repository, install them and update them as required
- ✓ **Features of Anaconda**
  - ➢ It is free and open source with more than 1500 Python/R packages.
  - ➢ It simplifies package management and working with tools and libraries.
  - ➢ It has tools to easily collect data from sources using machine learning and AI.
  - ➢ It creates a simplified environm that is easily manageable for deploying any project.
  - ➢ Build and train ML and DL models with scikit-learn, TensorFlow, Pytorch, etc.

Anaconda Data Science Platform



ANACONDA DISTRIBUTION

Most Trusted Distribution for Data Science

ANACONDA NAVIGATOR

Desktop Portal to Data Science

ANACONDA PROJECT

Portable Data Science Encapsulation

DATA SCIENCE LIBRARIES

Data Science IDEs

Analytics & Scientific Computing

Visualization

Machine Learning

...and many more!

CONDA

Data Science Package & Environment Manager

Anaconda Data Science Platform

Pros	Cons
• Easily manageable for deploying any project • Good community support • Manage libraries, dependencies, and environments with Conda • Build and train ML and deep learning models with inbuilt libraries	• It can be a bit bulky sometimes, slowing down and lagging while you are working on your code, especially when you are on a low-end system. • Lots of packages and environments can complicate simple stuff sometimes. • Gets slow when working on heavy Deep Learning Algorithms

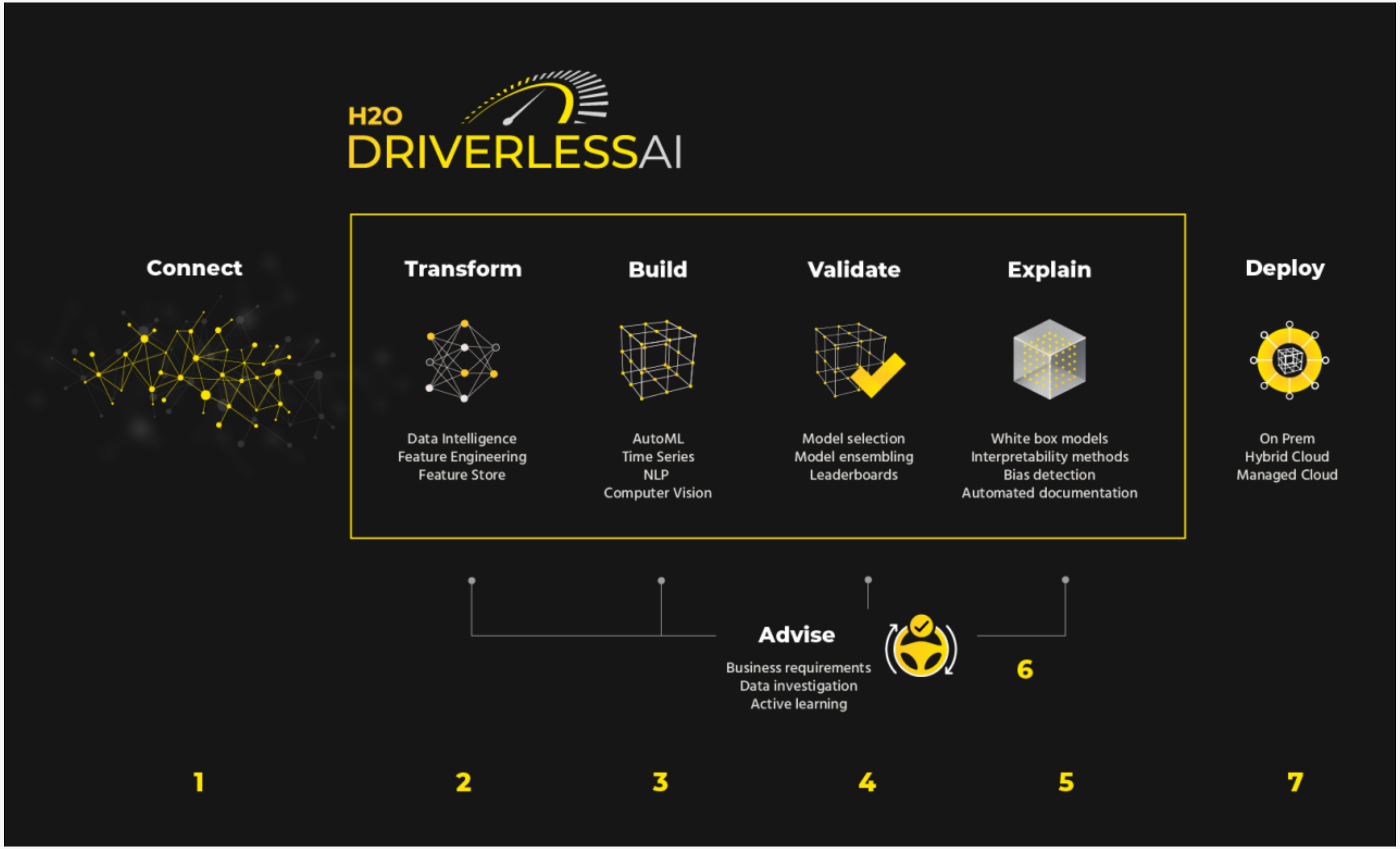
H2o.ai Platform

- ✓ H2O.ai is an Open-source and freely distributed platform. H2O is popular among novice and expert data scientists. H2O.ai Machine learning suite.

- ✓ **Features of H2O.ai**

  - ➢ It works across a variety of data sources, including HDFS, Amazon S3, and more. It can be deployed everywhere in different clouds
  - ➢ Driverless AI is optimized to take advantage of GPU acceleration to achieve up to 40X speedups for automatic machine learning.
  - ➢ Feature engineering is the secret weapon that advanced data scientists use to extract the most accurate results from algorithms, and it employs a library of algorithms and feature transformations to automatically engineer new, high-value features for a given dataset.
  - ➢ It provides an AutoDoc for each experiment, relieving the user from the time-consuming task of documenting and summarizing their workflow used when building ML models.
  - ➢ Driverless AI provides robust interpretability of ML models to explain modeling results in a

H2o.ai Platform



H2O DRIVERLESSAI

1 Connect

2 Transform
Data Intelligence
Feature Engineering
Feature Store

3 Build
AutoML
Time Series
NLP
Computer Vision

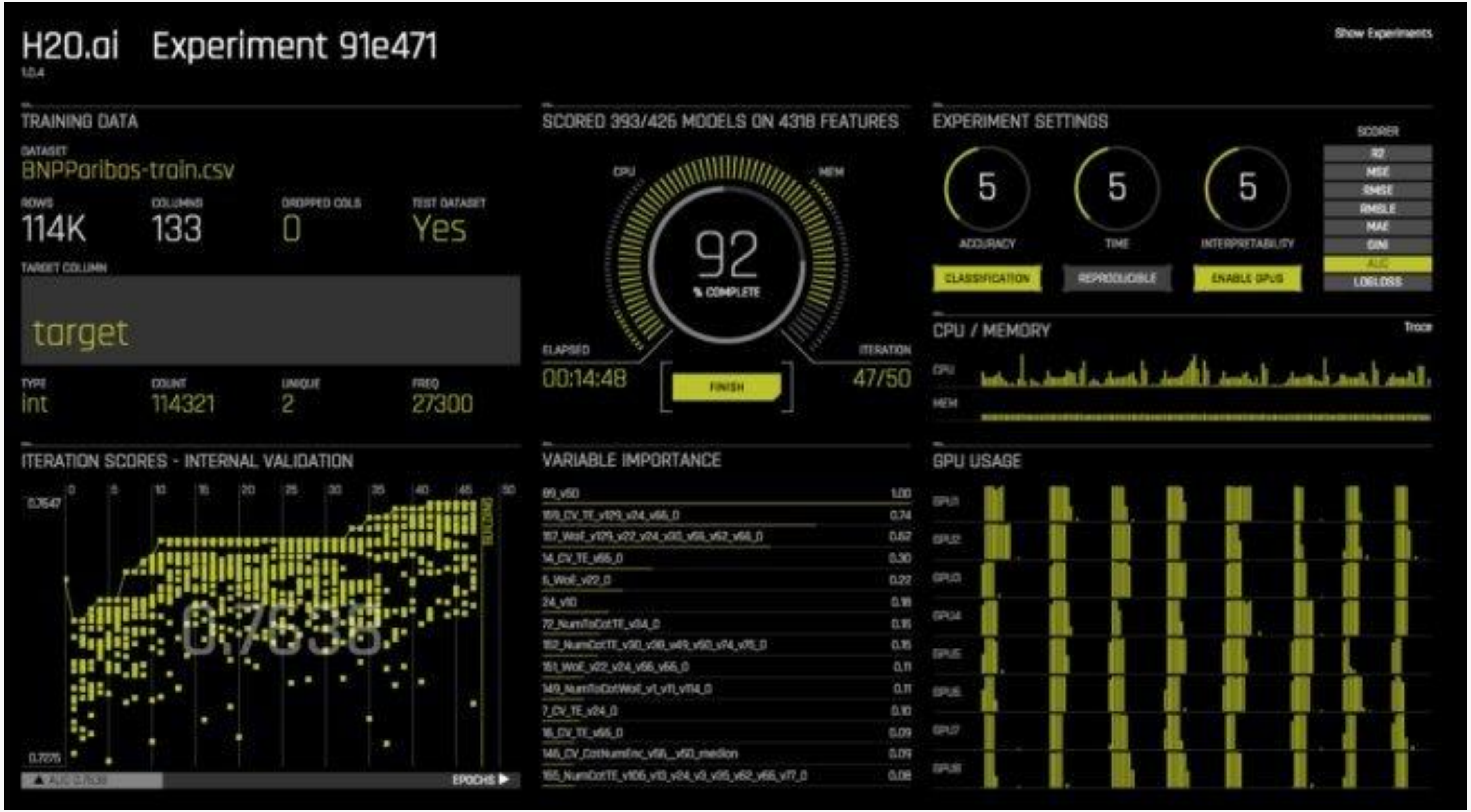
4 Validate
Model selection
Model ensembling
Leaderboards

5 Explain
White box models
Interpretability methods
Bias detection
Automated documentation

6 Advise
Business requirements
Data investigation
Active learning

7 Deploy
On Prem
Hybrid Cloud
Managed Cloud

H2O.ai Platform



H2O.ai Experiment 91e471
1.0.4

TRAINING DATA
DATASET BNPParibas-train.csv
ROWS 114K · COLUMNS 133 · DROPPED COLS 0 · TEST DATASET Yes
TARGET COLUMN target
TYPE int · COUNT 114321 · UNIQUE 2 · FREQ 27300

SCORED 393/426 MODELS ON 4318 FEATURES
92 % COMPLETE
ELAPSED 00:14:48 · ITERATION 47/50

EXPERIMENT SETTINGS
ACCURACY 5 · TIME 5 · INTERPRETABILITY 5
CLASSIFICATION · REPRODUCIBLE · ENABLE GPUS
SCORER AUC

VARIABLE IMPORTANCE

89_v50	1.00
159_CV_TE_v129_v24_v66_0	0.74
157_WoE_v129_v22_v24_v30_v56_v62_v66_0	0.62
14_CV_TE_v66_0	0.30
6_WoE_v22_0	0.22
24_v10	0.18
72_NumToCatTE_v94_0	0.15
152_NumCatTE_v30_v38_v49_v50_v74_v75_0	0.15
151_WoE_v22_v24_v56_v66_0	0.11
149_NumToCatWoE_v1_v11_v114_0	0.11
7_CV_TE_v24_0	0.10
16_CV_TE_v56_0	0.09
146_CV_CatNumEnc_v56__v50_median	0.09
165_NumCatTE_v106_v13_v24_v3_v56_v62_v66_v17_0	0.08

H2O.ai Platform

Pros	Cons
• With the system's powerful GPU acceleration support, H2O Driverless AI is a quick performing automation platform that provides. • Employing H2O Driverless AI service would allow the automation of a big chunk of workflows, which would mean reduced expenses for the company and speed up the process of the work. • This platform has features referred to as interpretability tools that give users the ability to acquire, model in English, and debug reason codes. • It is a user-friendly automation platform compared to many of the other solutions in the market.	• It is not very scalable compared to other platforms. • Lack of proper documentation • H2O.AI can take up lots of memory.

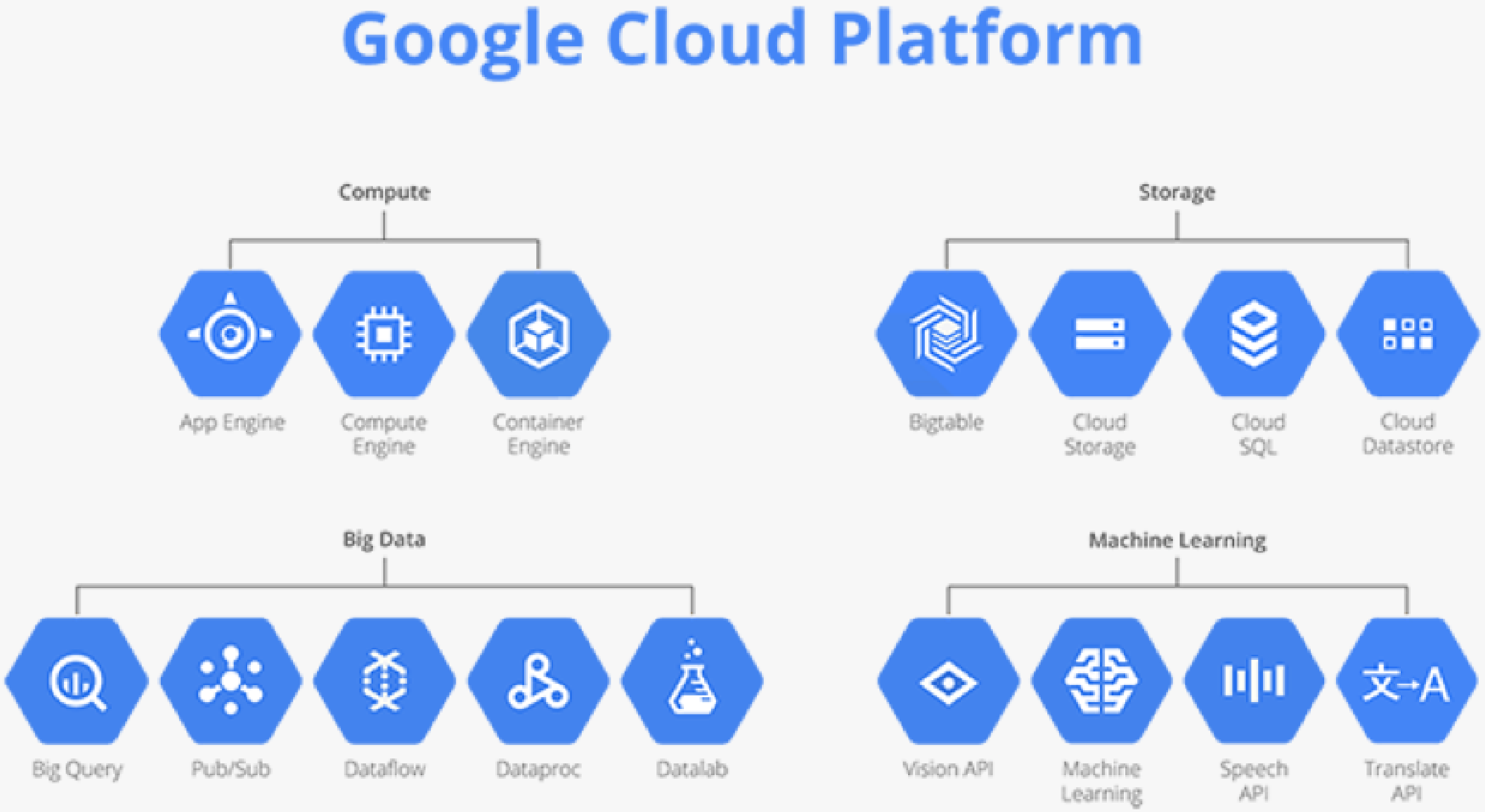
Data Science on Google Cloud Platform

- ✓ Google Cloud Platform is one of the best data science learning platforms. From data engineering to ML engineering, TensorFlow to PyTorch, GPUs to TPUs, data science on Google Cloud helps your run faster, smarter, and at planet scale.

- ✓ **Features of GCP**

  - ➢ An automated environment with web-based tools. Therefore, no human intervention is required to access the resources.
  - ➢ The resources and the information can be accessed from anywhere.
  - ➢ Google has its own network that enables users to have more control over GCP functions for smooth performance and increased efficiency over the network.
  - ➢ Users are getting a more scalable platform over the private network and have more scalability.
  - ➢ There is a high number of security professionals working at Google to give high security to its customers.

Data Science on Google Cloud Platform



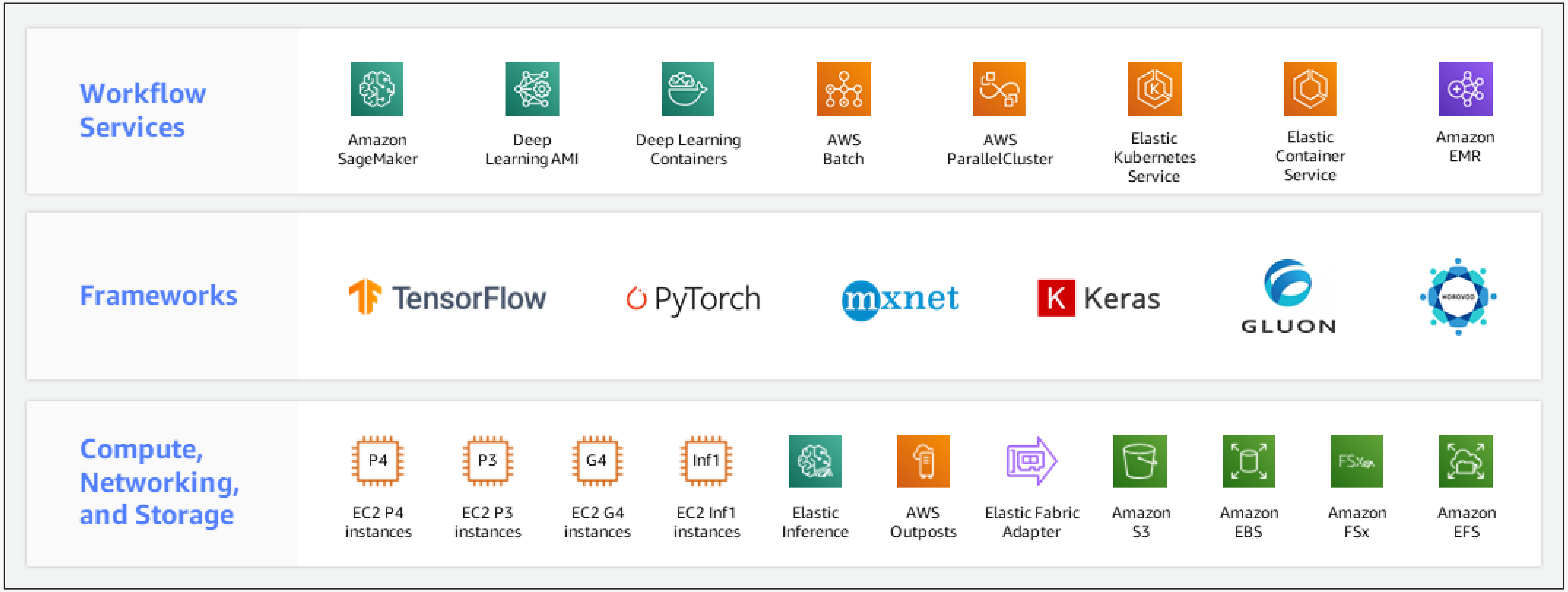
Google Cloud Platform

Pros	Cons
• The availability of more resources whenever required. • The easy-to-pay feature enables users to pay only for consumed services. • Google enables users to get Google Cloud hosting at the cheapest rates. The hosting plans are not only cheaper than other hosting platforms but also offer better features than others. GCP provides a pay-as-you-go option to the users where users can pay separately only for the services and resources they want to use. • Once the account is configured on GCP, it can be accessed from anywhere. That means that the user can use GCP across different devices from different places. • It is possible because Google provides web-based applications that allow users to have complete access to GCP.	• GCP has relatively few global data centers across the World compared to other cloud services. • There are very few customization options available in GCP products such as BigQuery, Spanner, and Datastore. • GCP Application Engine is restricted only to languages like Java, Python, PHP, and Google Go only. • GCP's support is not the strongest when it involves handling customer issues plus the support fees are quite expensive.

Data Science on AWS

- ✓ Amazon Web Services (AWS) provides a dizzying array of cloud services, from the well-known Elastic Compute Cloud (EC2) and Simple Storage Service (S3) to platform as a service (PaaS) offering covering almost every aspect of modern computing.
- ✓ It specifically provides a mature big data architecture with services covering the entire data processing pipeline : from ingestion through treatment and pre-processing, ETL, querying, and analysis to visualization and dashboarding.
- ✓ It lets you manage big data seamlessly and effortlessly without having to set up complex infrastructure or deploy software solutions like Spark, which makes it one of the best and most used platforms globally

Data Science on AWS



Workflow Services

Amazon SageMaker
Deep Learning AMI
Deep Learning Containers
AWS Batch
AWS ParallelCluster
Elastic Kubernetes Service
Elastic Container Service
Amazon EMR

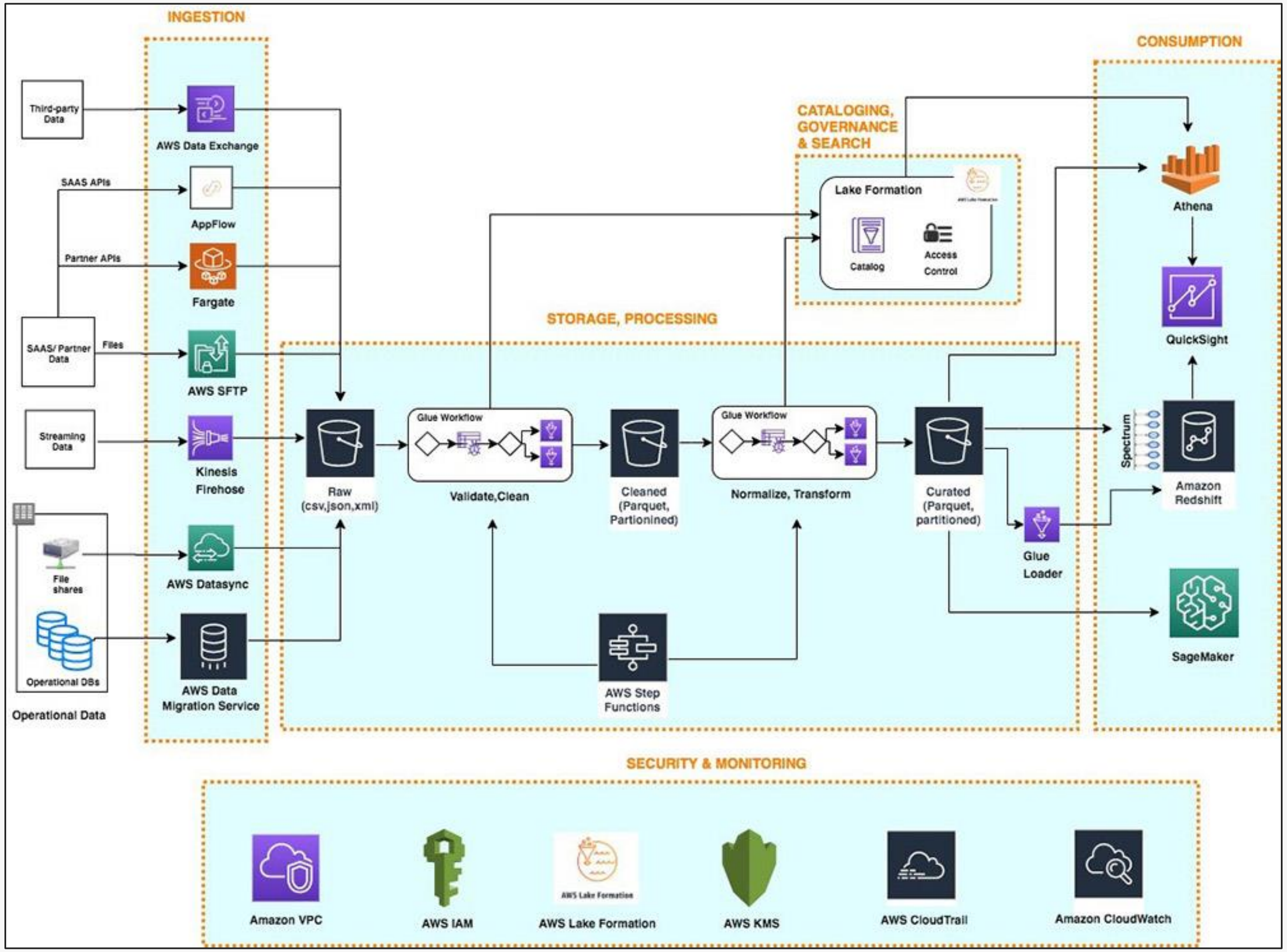
Frameworks

TensorFlow
PyTorch
mxnet
Keras
GLUON
HOROVOD

Compute, Networking, and Storage

EC2 P4 instances
EC2 P3 instances
EC2 G4 instances
EC2 Inf1 instances
Elastic Inference
AWS Outposts
Elastic Fabric Adapter
Amazon S3
Amazon EBS
Amazon FSx
Amazon EFS

Data Science on AWS



Data Science on AWS

✓ Features of AWS

- ➢ Flexibility is one of the most popular key features of AWS. The flexibility of AWS is a great asset for organizations to deliver the product with updated technology in time and overall enhance productivity. Scalability in AWS has the ability to scale the computing resources up or down when demand increases or decreases respectively.
- ➢ AWS provides a scalable cloud-computing platform that provides customers with end-to-end security and end-to-end privacy.
- ➢ AWS incorporates security into its services and also maintains confidentiality, integrity, and availability of your data which is of the utmost importance.

Amazon Web Services Platform

Pros	Cons
• A very user-friendly interface that provides access to a wide number of applications and services. • Expanded into over 70 more services. This includes database, software, mobile, analytics, and networking. • Huge, unlimited bandwidth for highly trafficked websites • Another major benefit of AWS is its flexibility, with basically no limit to how much you can use	• AWS has quite complicated billing, which can be confusing for beginners • Another downgrade is Amazon's EC2 has limits like limiting resources by region. So, where you are located, or your region can determine just how many resources you will have access to • Limit spending on resources for new users • Common Cloud Computing Problems like backup protection, risk of data leakage, privacy issues, security, downtime, and limited control.

Ministry of Higher Education and Scientific Research
Djilali BOUNAAMA University - Khemis Miliana(UDBKM)
Faculty of Science and Technology
Department of Mathematics and Computer Science

Chapter 3

Data Science Tools

AIBD-M1-UEM112 : Introduction to Data Science

Noureddine AZZOUZA

n.azzouza@univ-dbkm.dz