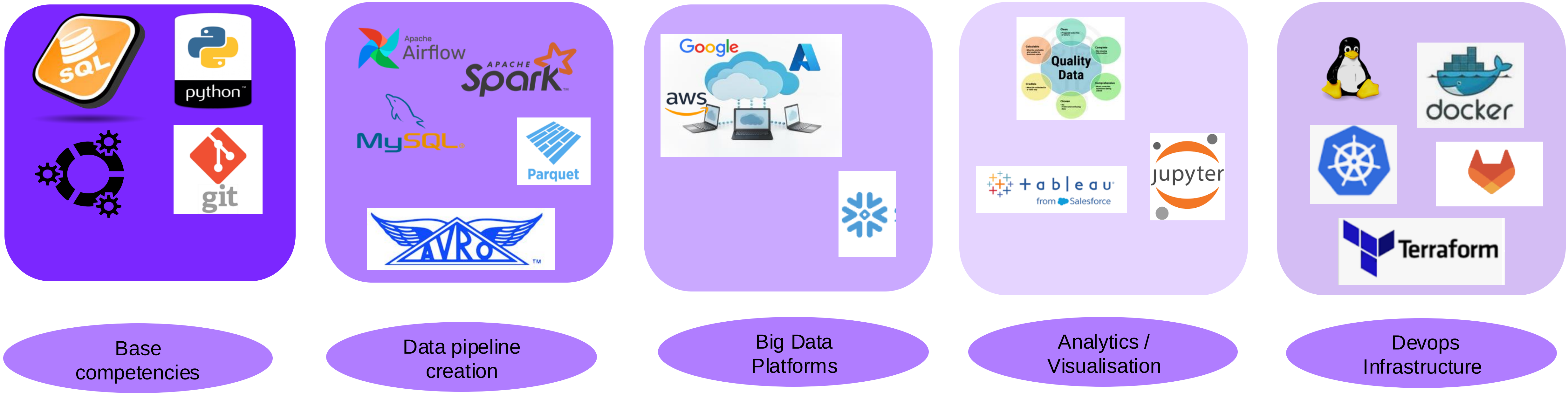
NETAPP DATAOPS TOOL KIT

for data management

Dr. Didier Gava
EMEA & LATAM Senior Solution architect
September 2024

“Data Scientist”/”Data Engineer” Journey



Base competencies

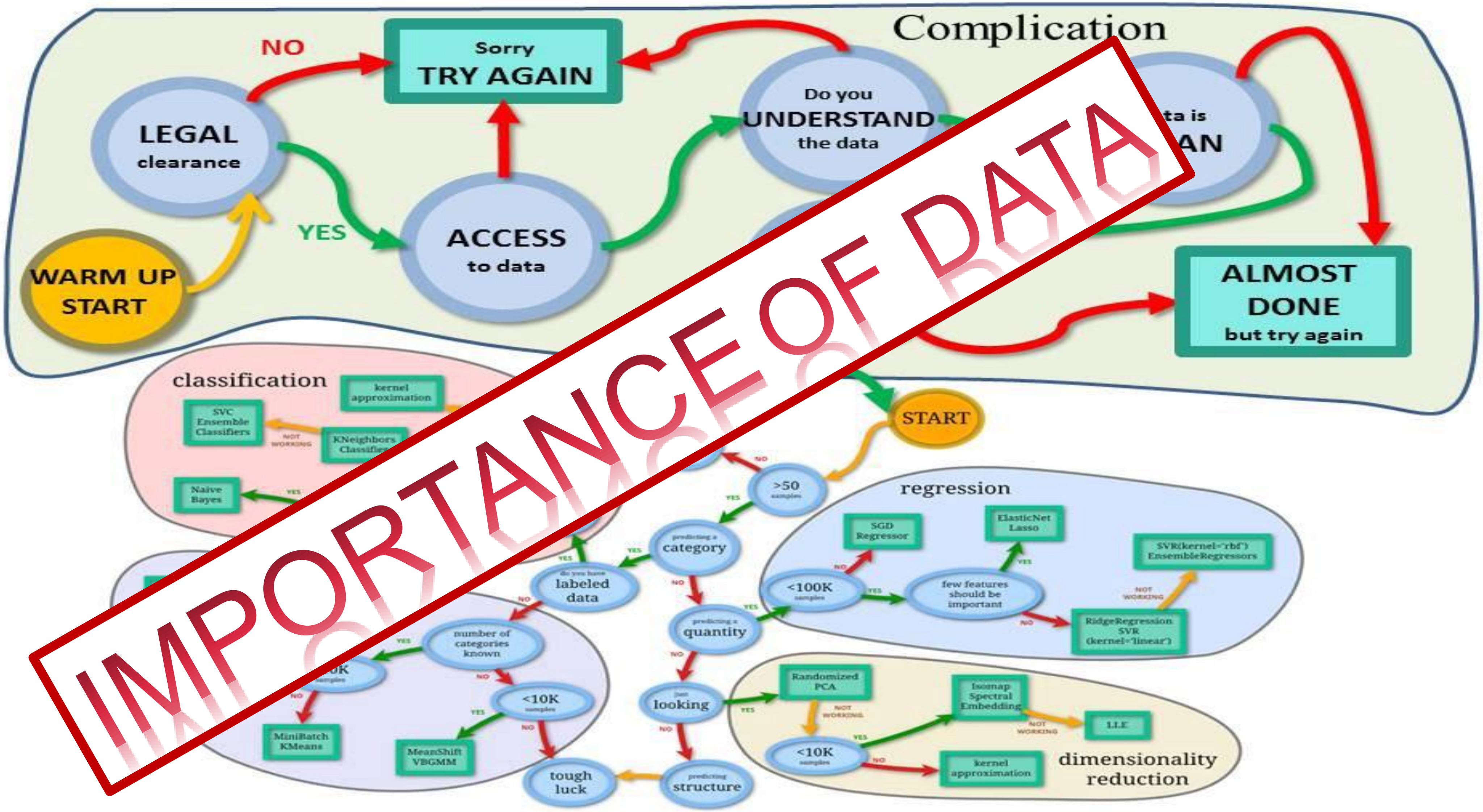
Data pipeline creation

Big Data Platforms

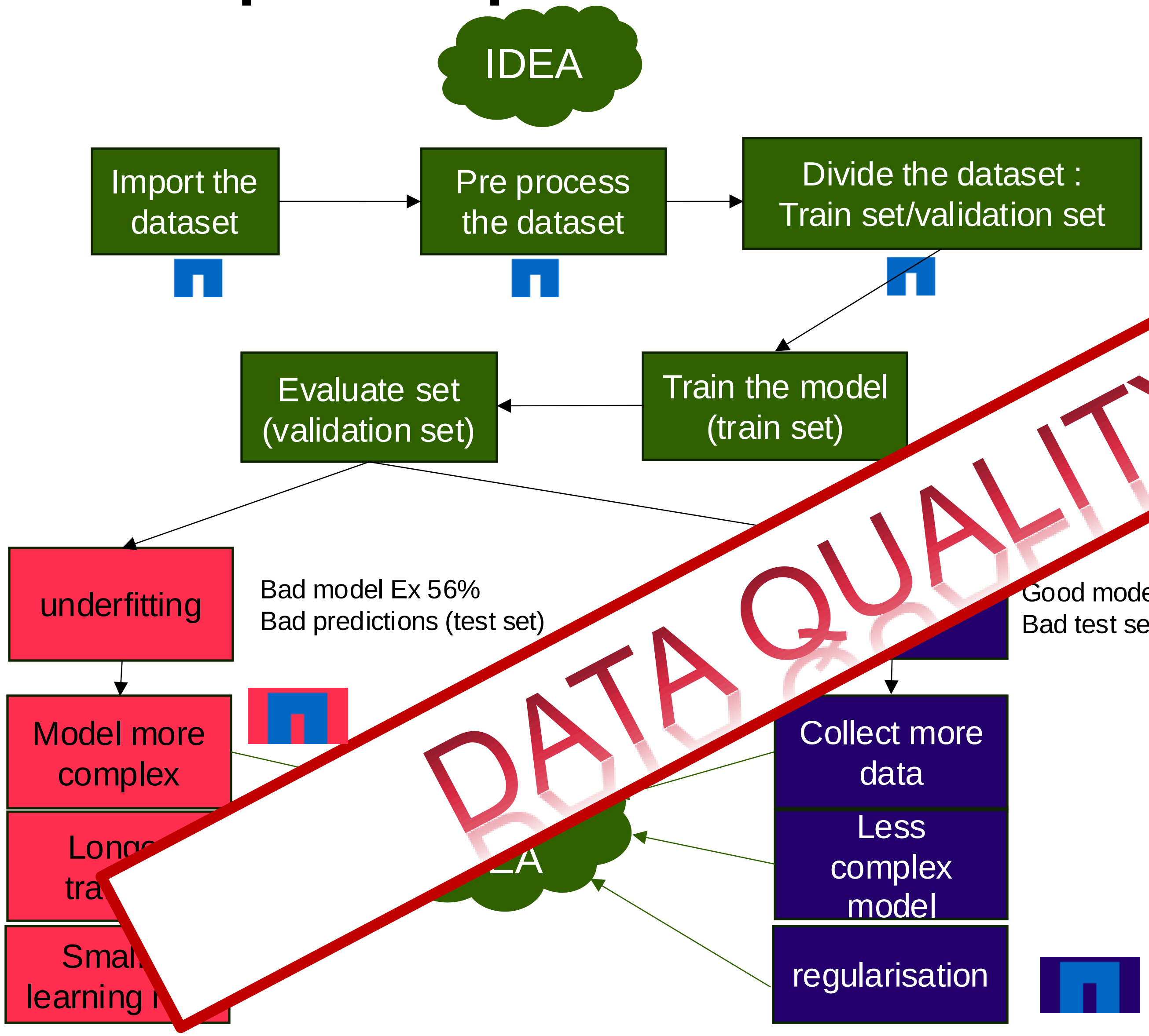
Analytics / Visualisation

Devops Infrastructure

Data Scientist Data



Development steps in ML



IDEA

Import the dataset

Pre process the dataset

Divide the dataset : Train set/validation set

Train the model (train set)

Evaluate set (validation set)

underfitting

Bad model Ex 56%
Bad predictions (test set)

Model more complex

Long
tra

Smal
learning r

EA

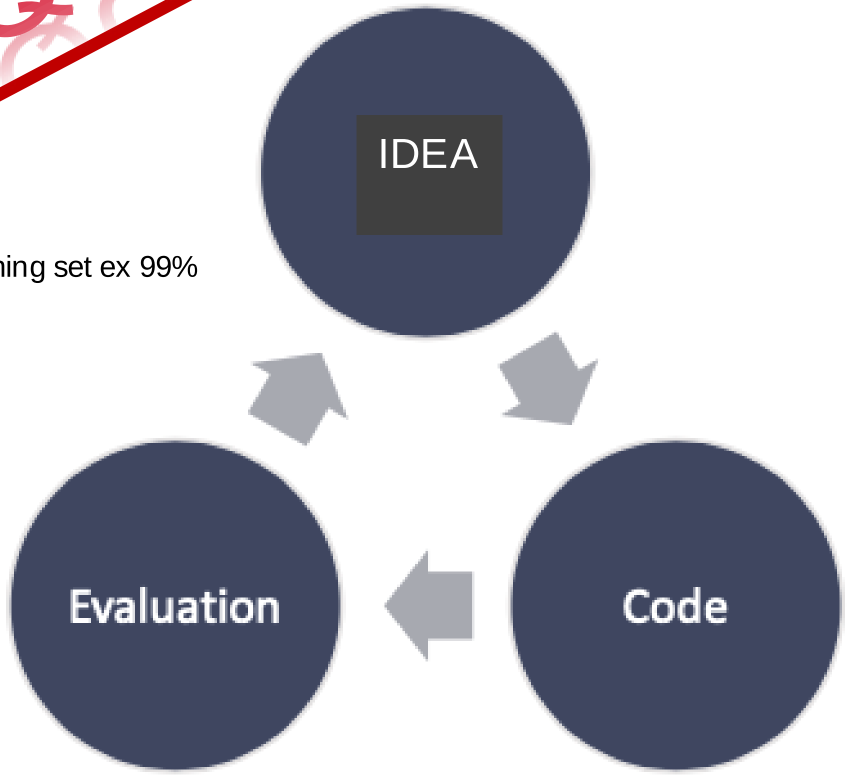
Good model on the training set ex 99%
Bad test set predictions

Collect more data

Less complex model

regularisation

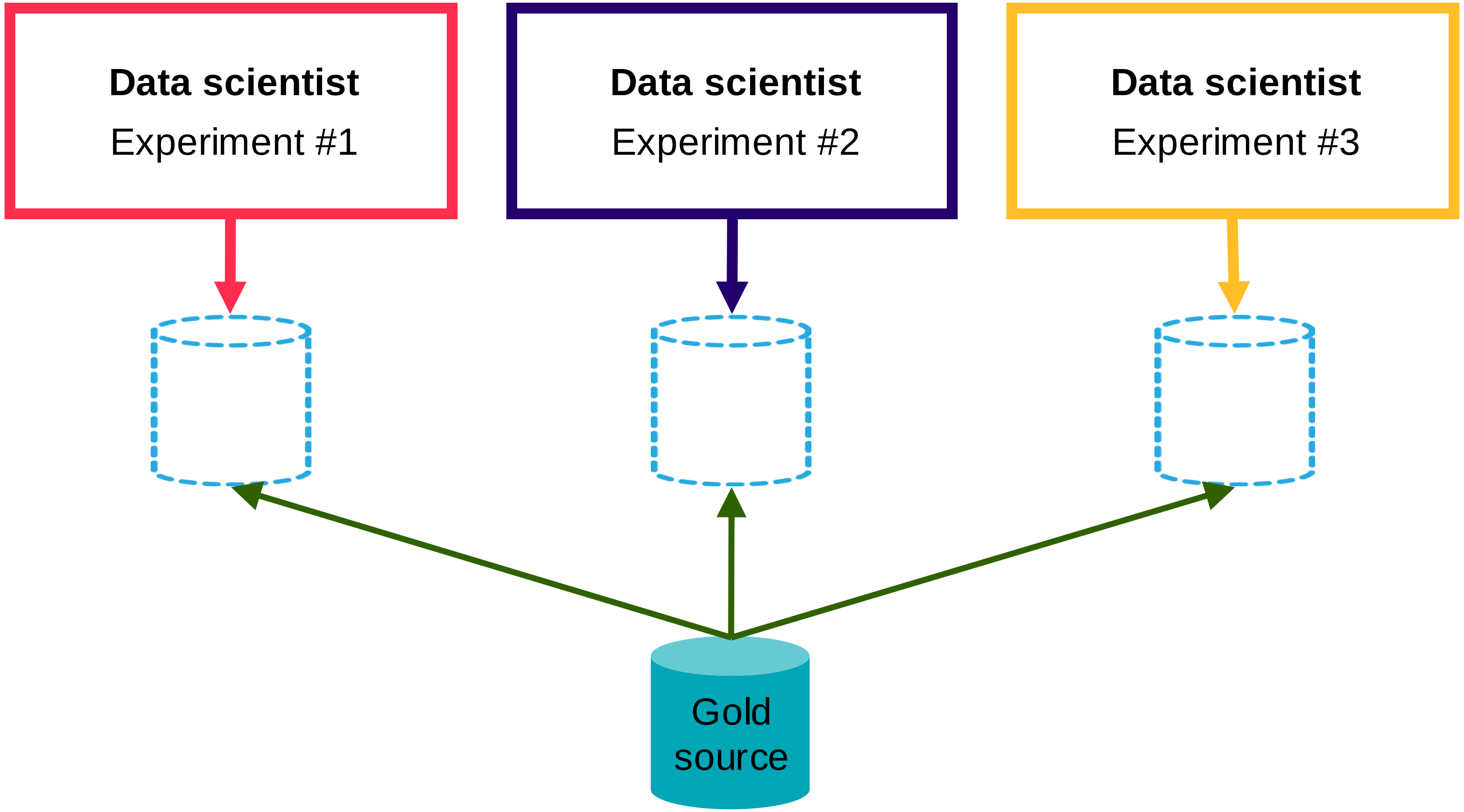
- Developing a machine lea
done the first time.
- We often sta a simple **and** quick m en we analyze if we ha as and we try a new **idea** blems encountered, etc.



DATA QUALITY/QUANTITY

Training accurate models

Rapid experimentation is necessary



Data scientist

Experiment #1

Data scientist

Experiment #2

Data scientist

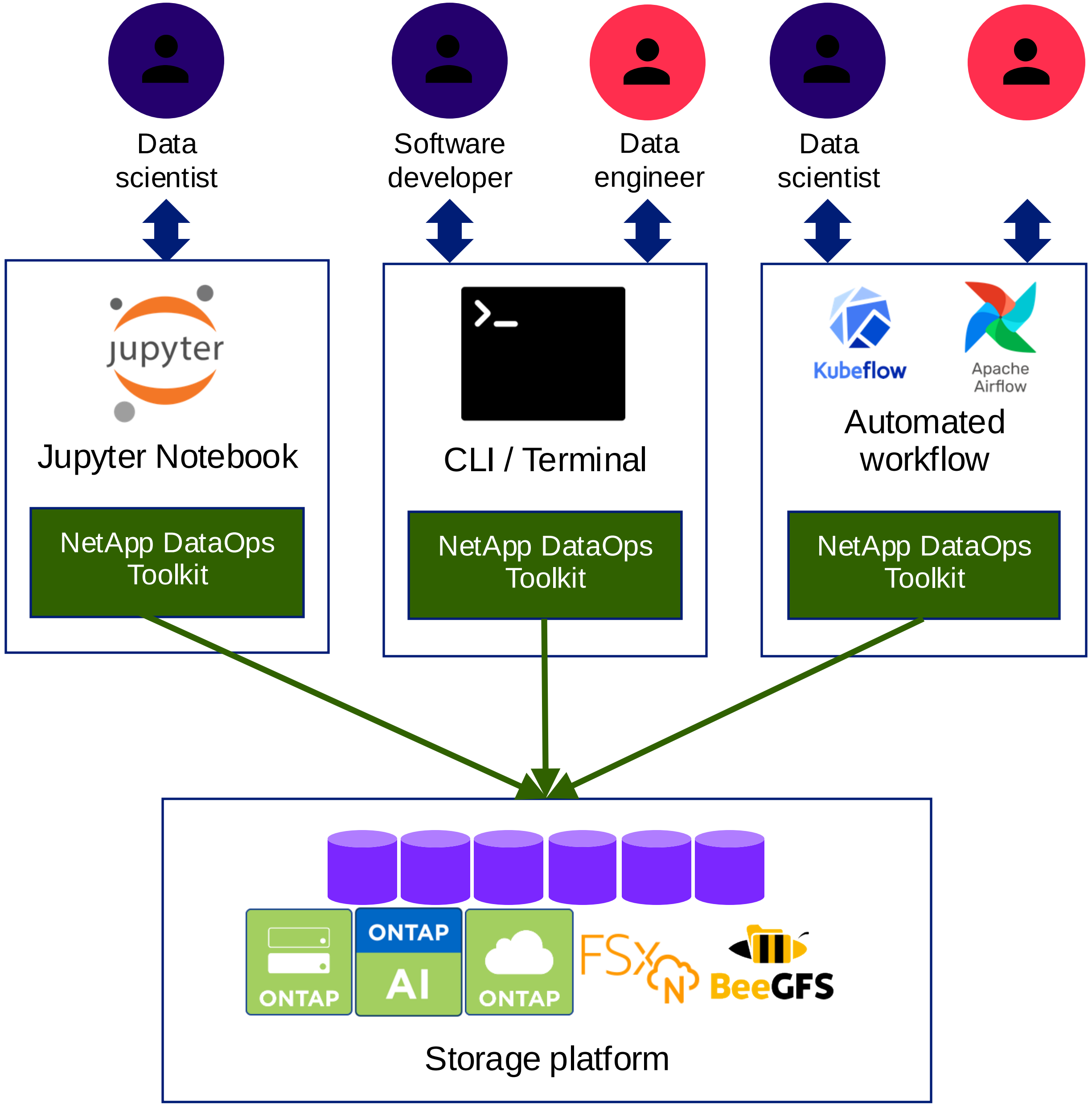
Experiment #3

Gold source

NetApp DataOps Toolkit

Simplifying AI data management

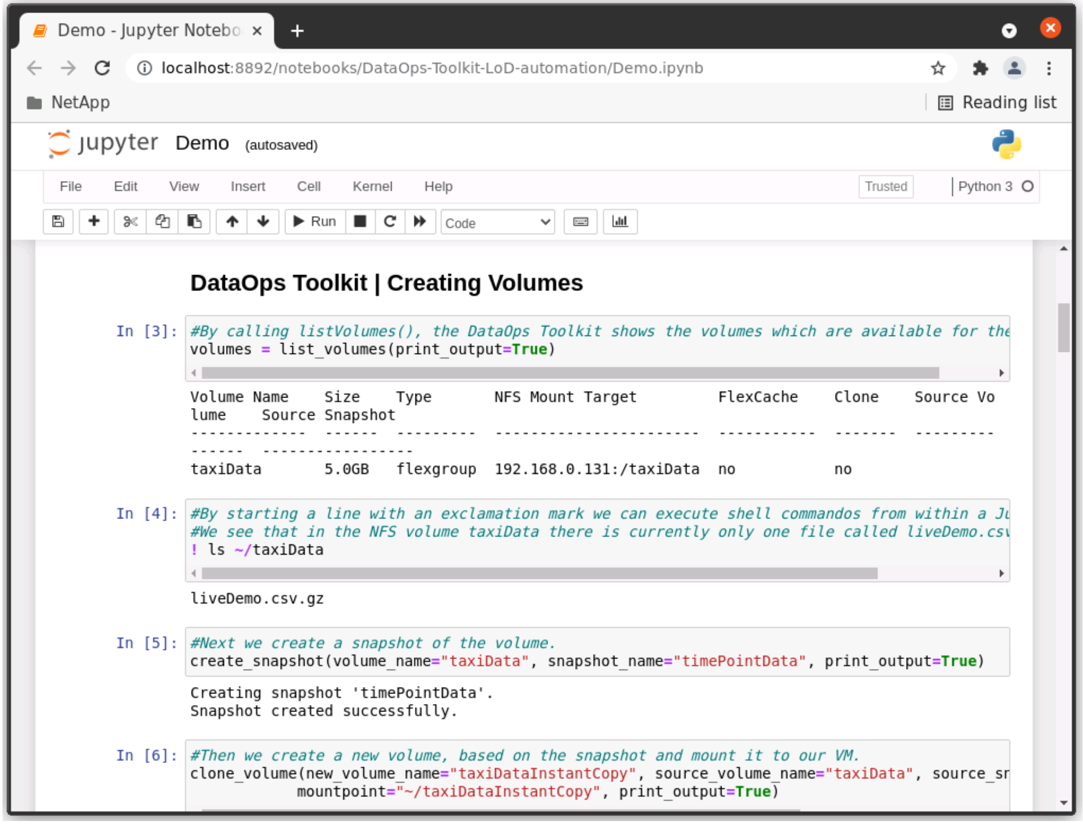
- NetApp’s industry-leading, multitenant data management capabilities
  - Traditional toolkit: Supports ONTAP (AFF, FAS, FSx, Cloud, Select)
  - Kubernetes toolkit: Supports ONTAP (AFF, FAS, FSx, Cloud, Select), Azure NetApp Files (ANF), Cloud Volumes Service (CVS), and BeeGFS (limited)
- Simple, easy-to-use interfaces; designed for data scientists and data engineers
  - CLI utility
  - Importable library of Python functions
- Provides access to advanced features that would normally require help from storage admin
- Key capabilities
  - Rapidly provision a new data volume
  - Near-instantaneously clone a data volume
  - Snapshot a data volume for traceability/versioning
  - Trigger data sync



NetApp DataOps Toolkit (Traditional)

Simplify access to NetApp solutions from Data Science environments

Jupyter Notebook:



DataOps Toolkit | Creating Volumes

```
In [3]: #By calling listVolumes(), the DataOps Toolkit shows the volumes which are available for the
        volumes = list_volumes(print_output=True)

        Volume Name    Size    Type       NFS Mount Target         FlexCache    Clone    Source Vo
        lume     Source Snapshot
        -------------  ------  ---------  -----------------------  -----------  -------  ---------
        ------  ----------------
        taxiData       5.0GB   flexgroup  192.168.0.131:/taxiData  no           no

In [4]: #By starting a line with an exclamation mark we can execute shell commandos from within a Ju
        #We see that in the NFS volume taxiData there is currently only one file called liveDemo.csv
        ! ls ~/taxiData

        liveDemo.csv.gz

In [5]: #Next we create a snapshot of the volume.
        create_snapshot(volume_name="taxiData", snapshot_name="timePointData", print_output=True)

        Creating snapshot 'timePointData'.
        Snapshot created successfully.

In [6]: #Then we create a new volume, based on the snapshot and mount it to our VM.
        clone_volume(new_volume_name="taxiDataInstantCopy", source_volume_name="taxiData", source_sr
                     mountpoint="~/taxiDataInstantCopy", print_output=True)
```

Key Functions:

Cloning of volumes

Deleting volumes

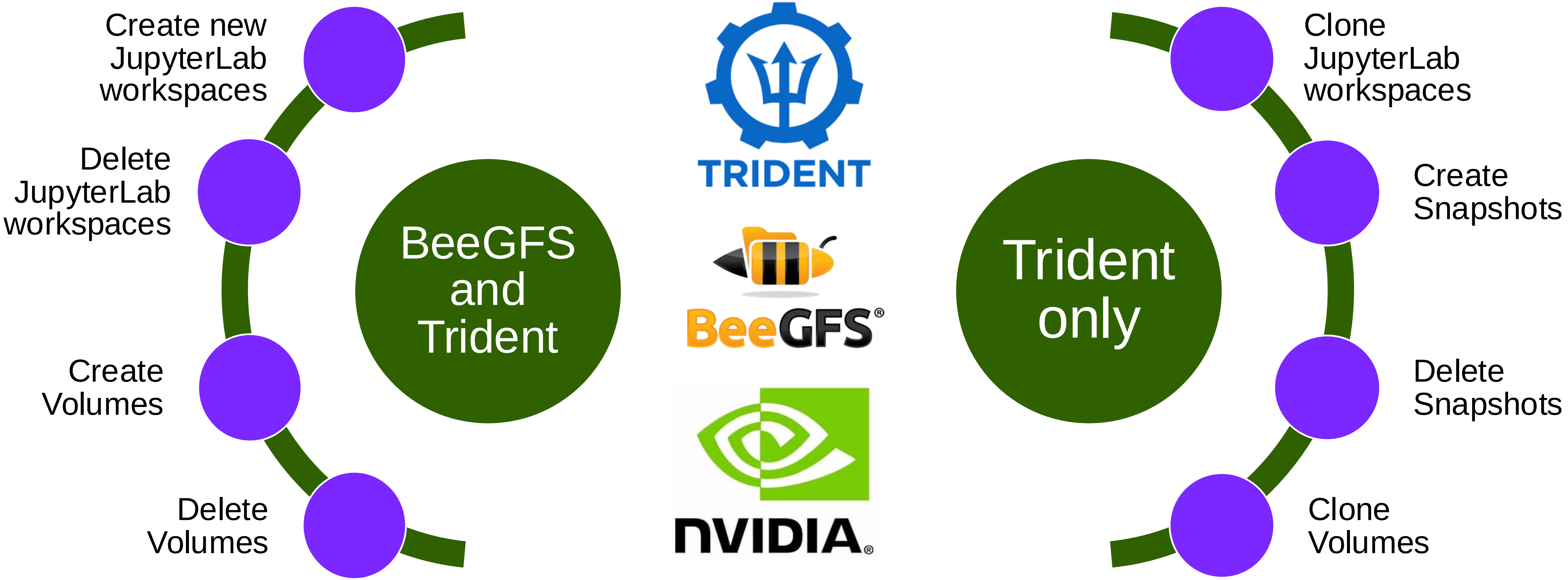
Creating Snapshots

Pull Content from S3

Push files to S3

NetApp DataOps Toolkit (Kubernetes)

Simplified data management in Kubernetes environments



BeeGFS and Trident

Create new JupyterLab workspaces

Delete JupyterLab workspaces

Create Volumes

Delete Volumes

Trident only

Clone JupyterLab workspaces

Create Snapshots

Delete Snapshots

Clone Volumes

Quick and easy installation and config

Get started in seconds

- NetApp DataOps Toolkit for Kubernetes – 1 step:

```
1. pip install netapp-dataops-k8s
```

- NetApp DataOps Toolkit for Traditional Environments – 2 steps

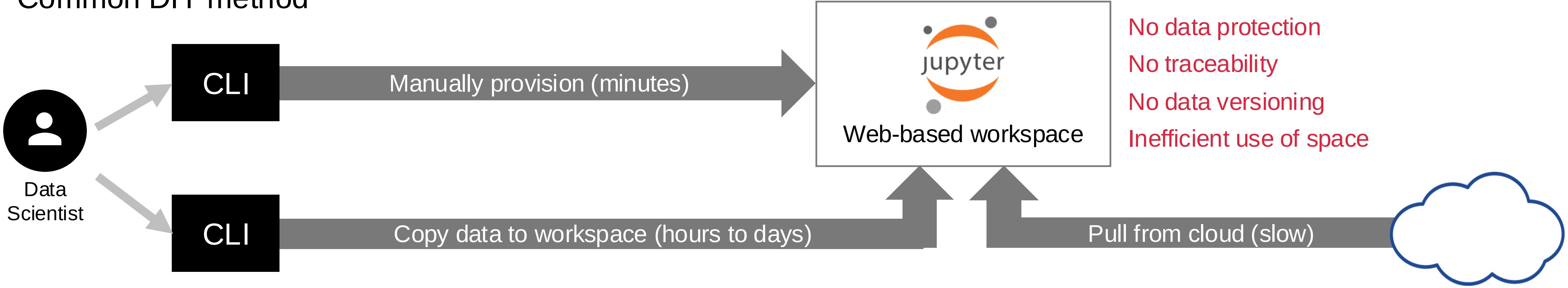
```
1. pip install netapp-dataops-traditional
2. netapp_dataops_cli.py config
```

Get up and running in under…

2 minutes

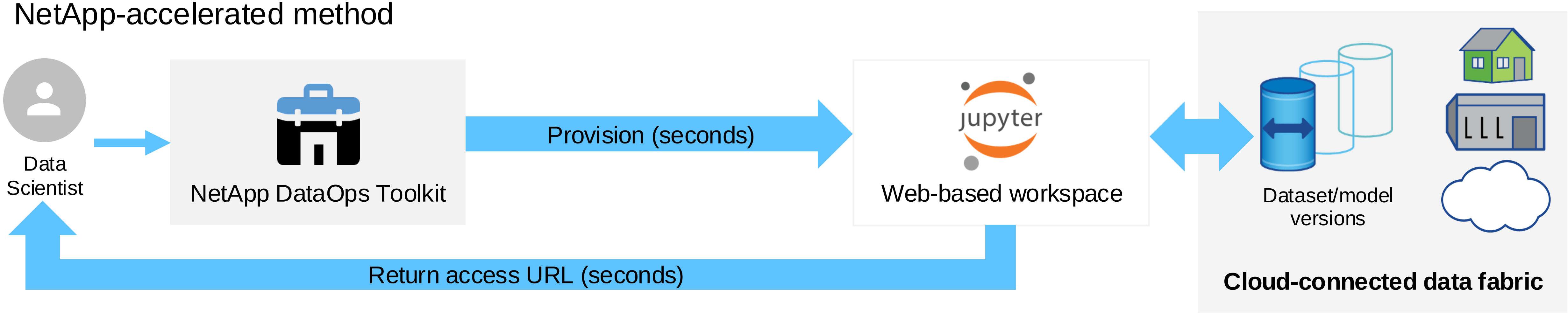
Self-service data science workspace creation

Common DIY method

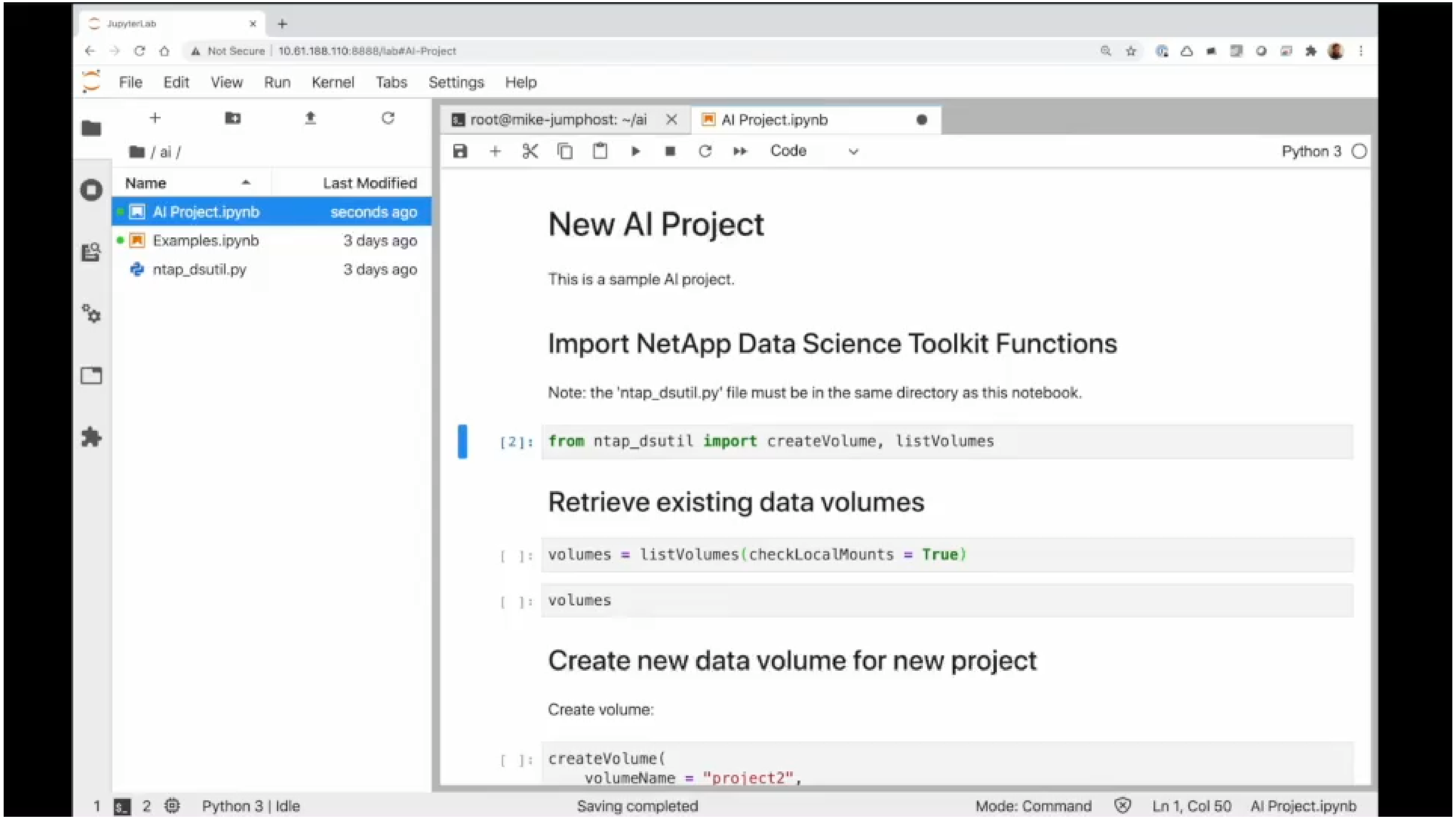


No data protection
No traceability
No data versioning
Inefficient use of space

NetApp-accelerated method



Provision Volumes



Name	Last Modified
AI Project.ipynb	seconds ago
Examples.ipynb	3 days ago
ntap_dsutil.py	3 days ago

New AI Project

This is a sample AI project.

Import NetApp Data Science Toolkit Functions

Note: the 'ntap_dsutil.py' file must be in the same directory as this notebook.

```
[2]: from ntap_dsutil import createVolume, listVolumes
```

Retrieve existing data volumes

```
[ ]: volumes = listVolumes(checkLocalMounts = True)
```

```
[ ]: volumes
```

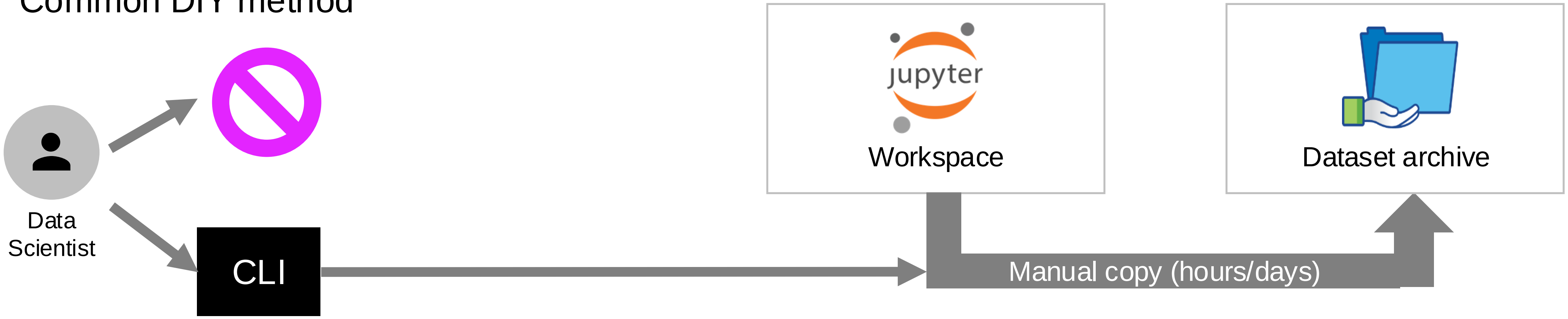
Create new data volume for new project

Create volume:

```
[ ]: createVolume(
         volumeName = "project2",
```

Dataset-to-model traceability

Common DIY method



Data Scientist

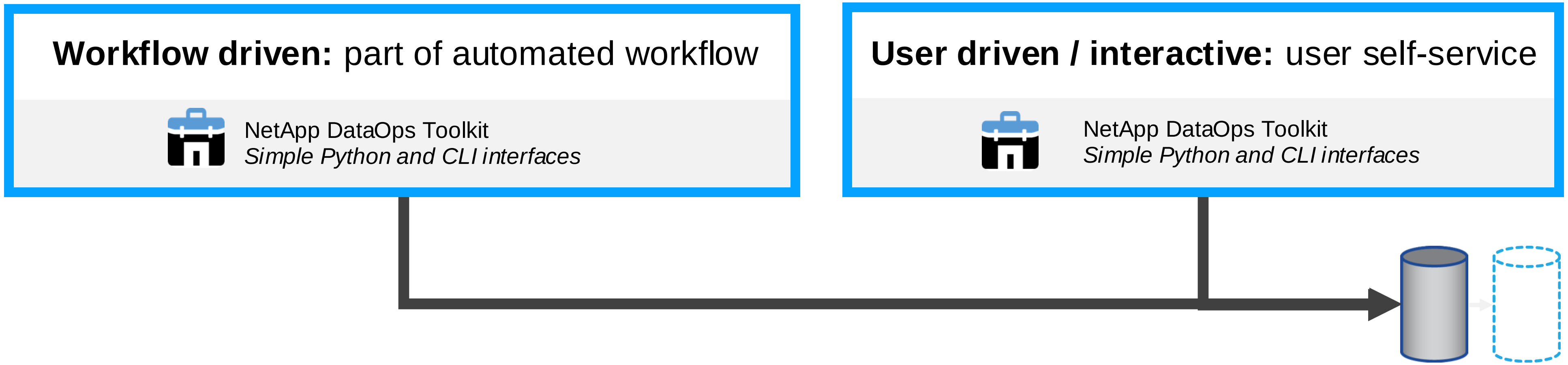
CLI

Workspace

Dataset archive

Manual copy (hours/days)

NetApp-accelerated method



Workflow driven: part of automated workflow

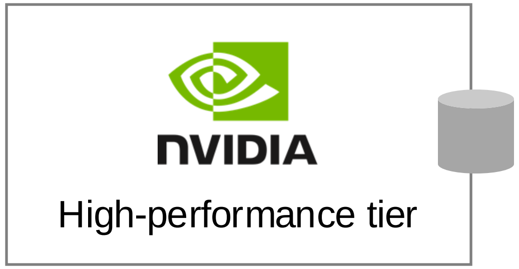
NetApp DataOps Toolkit
Simple Python and CLI interfaces

User driven / interactive: user self-service

NetApp DataOps Toolkit
Simple Python and CLI interfaces

Cold data tiering

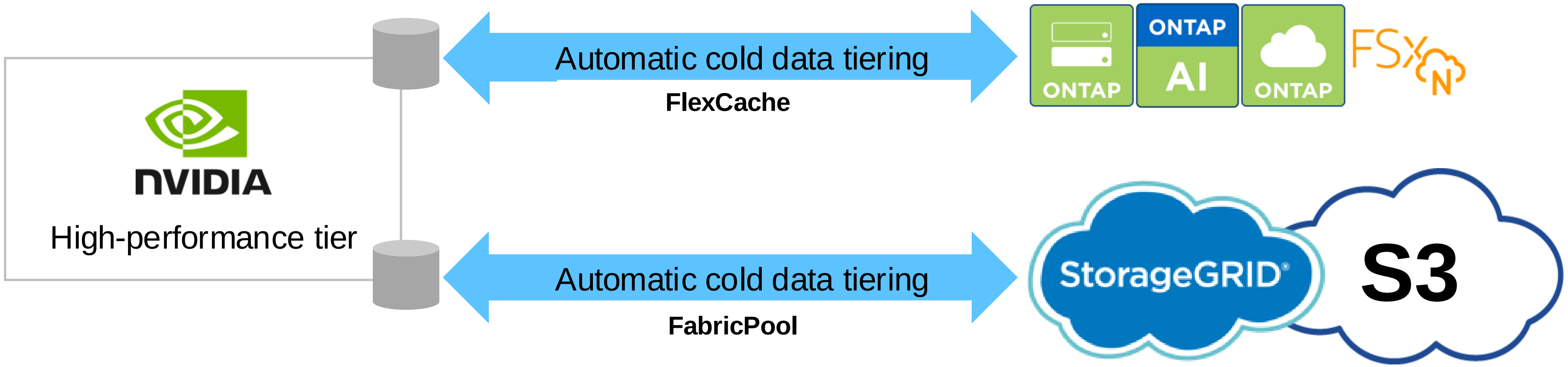
Without NetApp



High-performance tier

Cold data consumes valuable storage space in high-performance tier

With NetApp FlexCache and FabricPool



High-performance tier

Automatic cold data tiering

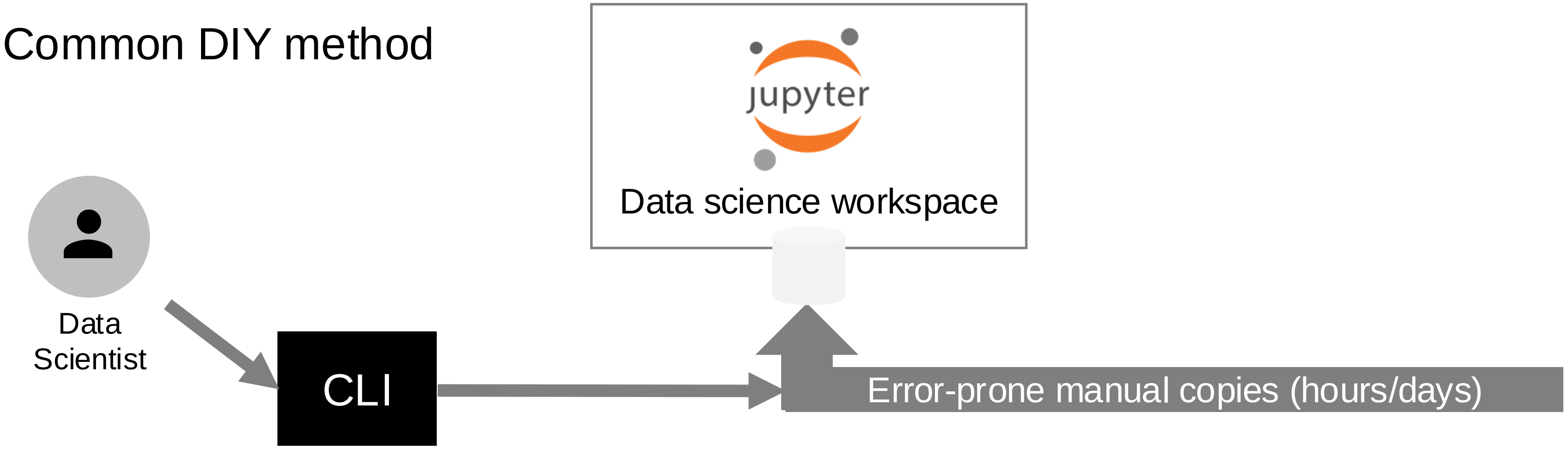
FlexCache

Automatic cold data tiering

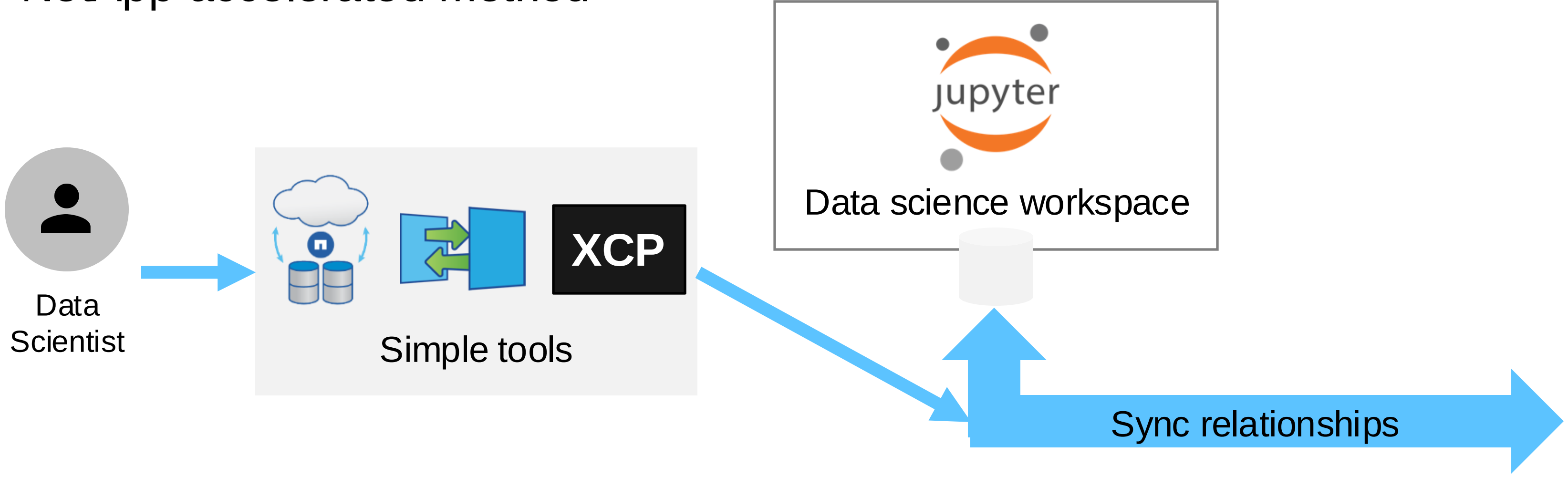
FabricPool

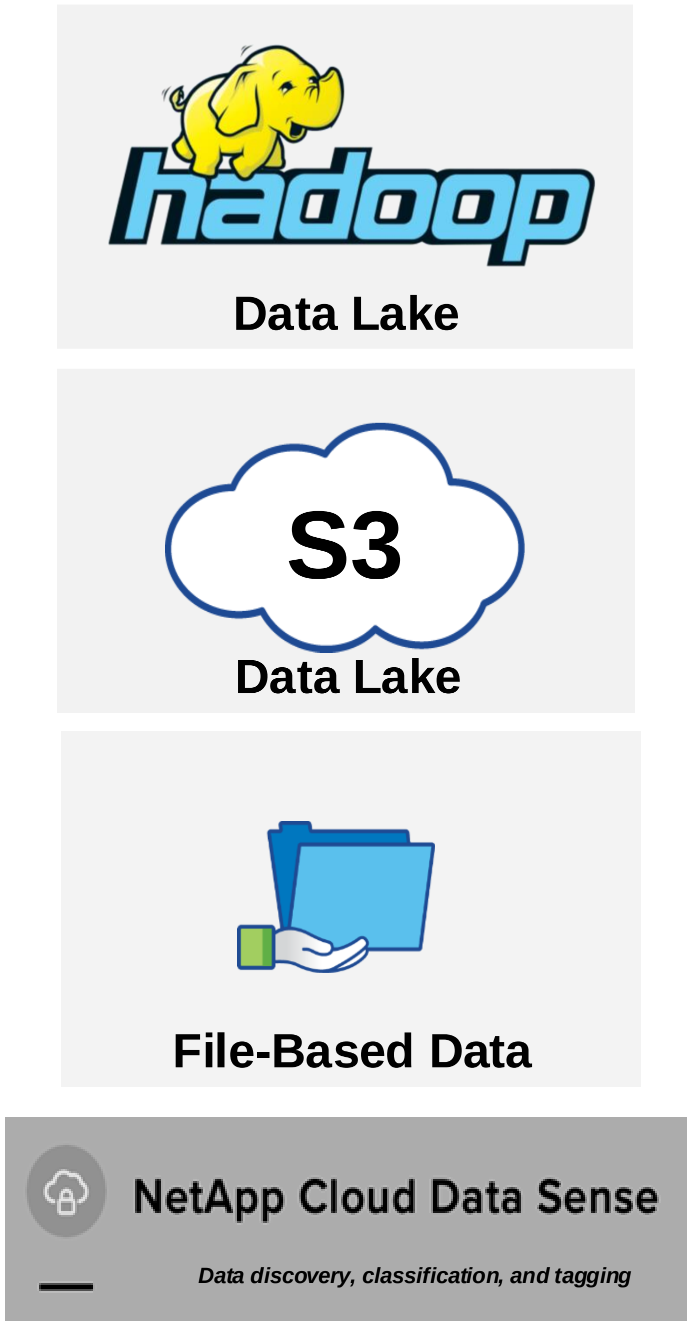
Data movement and sync

Common DIY method



NetApp-accelerated method





KEY TAKEAWAYS

01

Increase of Speed by creating clones within seconds and mounting volumes within Jupyter notebook

02

Access to Data easily as they are ”located” in servers

03

Traceability of Experiments by creating clones and/ or snapshots of working environments

THANK YOU