structuring your workspace: DS & DE/MLE perspectives

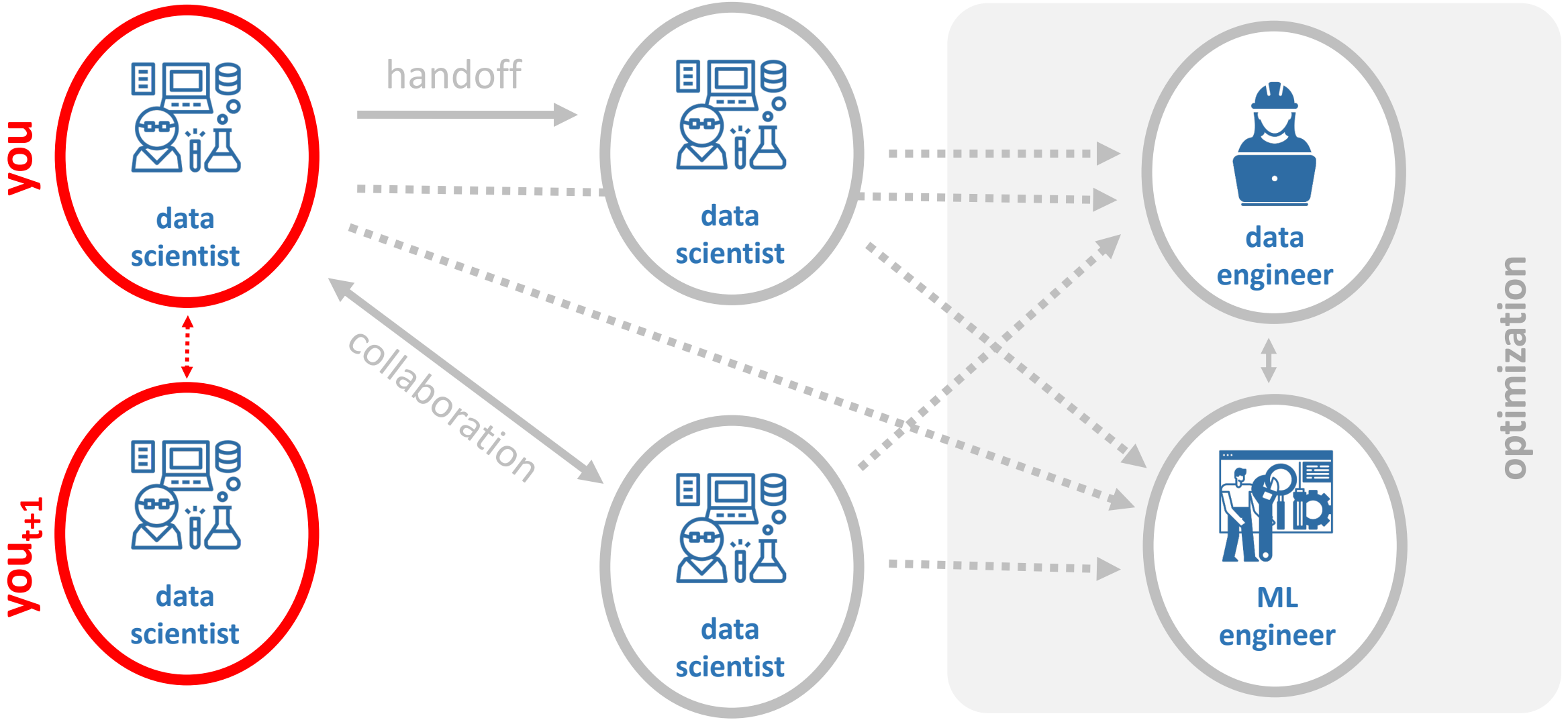
Marco Morales
marco.morales@columbia.edu

Nana Yaw Essuman
ne2388@columbia.edu

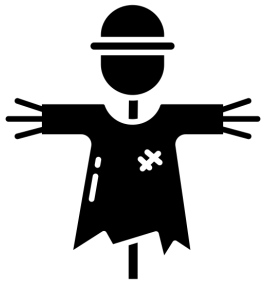
GR5069: Applied Data Science
for Social Scientists

Spring 2026
Columbia University

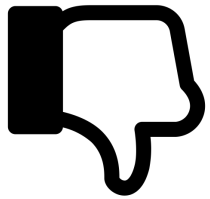
workflow collaboration in Data Science



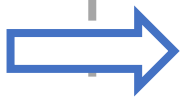
recap: iteration to build Data Products

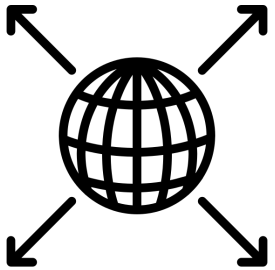


start small
(MVP)

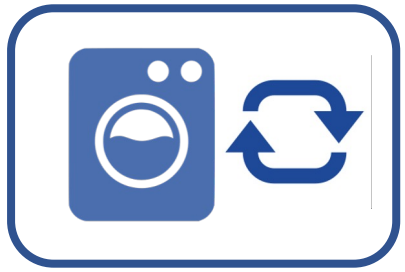


fail fast



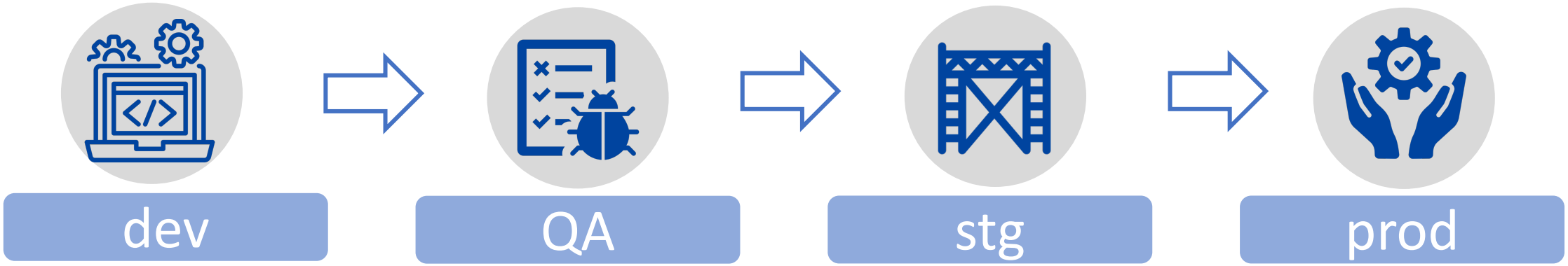


scale up



iterate

working environments to build Data Products



development

quality assurance / testing

staging

production

operational concepts in Data Science

portability

anyone should be able to **pick up where you left off** from **any machine**

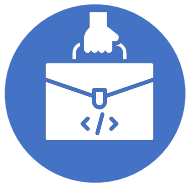
replicability

anyone should be able to arrive at your **same results**

scalability

your prototype should also work for **larger data sets** and/or be on the path of **automation**

operational concepts in Data Science



portability

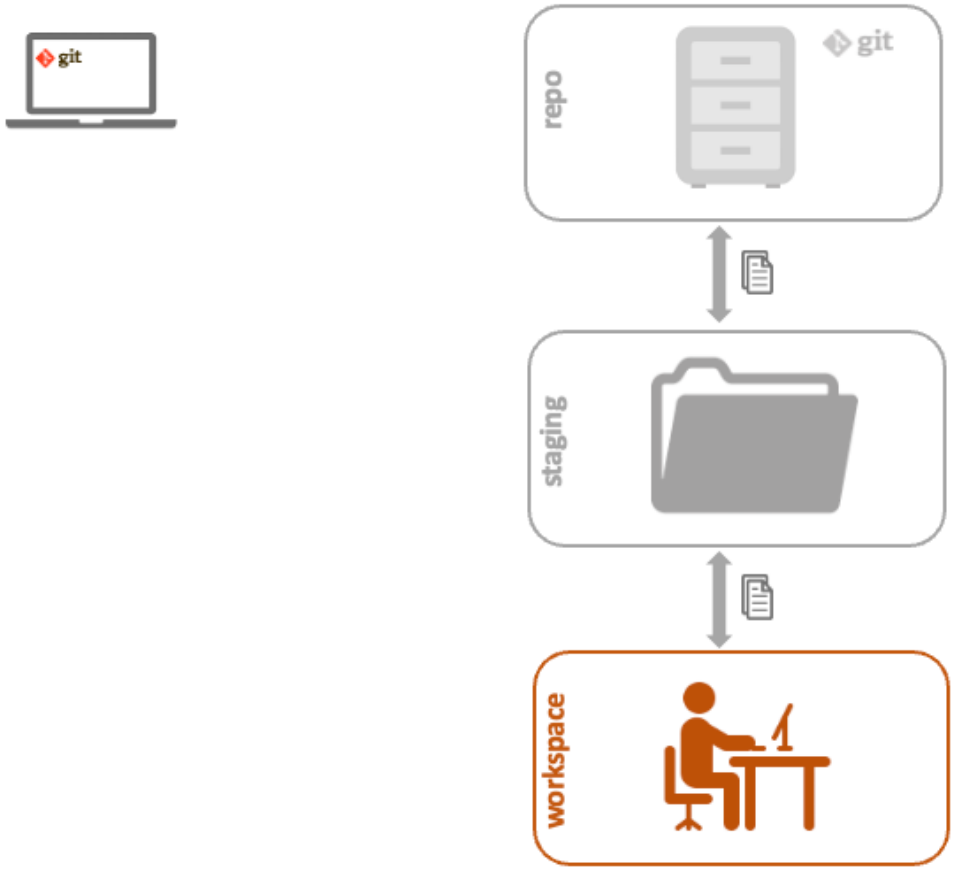
replicability

scalability

	portability	replicability	scalability
what	• flexible references • structured and documented code • replicate original environment	• documentation: data, software, hardware, environments • commented code • no manual processes	• high quality code • flexible functions • modularized code
why	• seamless handoff • frictionless transitions across environments	• seamless examination, review or validation • cordial troubleshooting • harmonious optimization	• simplified review and validation • reduce time optimizing, automating and deploying

our focus :

structuring your (local) **workspace**



a **thin layer** to structure your (local) workspace

```
workspace/
|
| -- src/                  <- code
|
| -- data/                 <- inputs
|
| -- reports/              <- outputs
|
| -- references/           <- data dictionaries, explanatory materials
|
| -- README.md
| -- .gitignore
| -- ToDo                  <- bug fixes, future enhancements
| -- LabNotebook           <- chronological records of project
```

a few principles to work by

principle 1: code in a common location

```
workspace/
|
| -- src/
|    | -- data/                  <- code to read raw data
|    | -- features/              <- code to transform data
|    | -- models/                <- code to model data
|    | -- visualizations/        <- code to create visualizations
|    | -- functions/             <- scripts to centralize functions
|    | -- config/                <- configuration files
|
| -- data/
|
| -- reports/
|
| -- references/
|
| -- README.md
| -- .gitignore
| -- ToDo
| -- LabNotebook
```

principle 1: code in a common location

- use `src` to organize your **source code**
  - **source code generates outputs**; it's not generated
- use **one script per purpose**
- use **version control to "update"** your scripts
- use code to document **"manual" changes**
- call **additional scripts** as needed
- if too many functions, keep a **script with functions**

principle 2: input **raw data** is immutable

```
workspace/
|
| -- src/
|
| -- data/
|    | -- raw/                    <- original, immutable data dump
|    | -- external/               <- data from third party sources
|    | -- interim/                <- intermediate transformed data (for validation)
|    | -- processed/              <- final processed data
|
| -- reports/
|
| -- references/
|
| -- README.md
| -- .gitignore
| -- ToDo
| -- LabNotebook
```

principle 2: input **raw data** is immutable

- **ALWAYS** keep your **raw data** as **immutable**
- keep **external data** separate and immutable
- (if/when needed) keep **interim data for validation**
- **processed data is ALWAYS replaceable!**
- all data should be linked to a script in `src`
- **document** origin of **raw & external data**

principle 3: outputs are disposable

```
workspace/
|
| -- src/
|
| -- data/
|
| -- reports/
|    | -- notebooks/              <- documents synthesizing analysis, experiments, etc
|    | -- figures/                <- images generated by code
|
| -- references/
|
| -- README.md
| -- .gitignore
| -- ToDo
| -- LabNotebook
```

principle 3: outputs are disposable

- use whichever document works best for your purpose:
  - Jupyter notebooks, R Markdown, scripts
- use notebooks to keep track of
  - **analyses**
  - **experiments**
  - **visualizations**
  - **tests**
  - **validation**
- **notebooks** can be **updated** and are **subject to change**

principle 4: keep documentation

```
workspace/
|
| -- src/
|
| -- data/
|
| -- reports/
|
| -- references/                  <- data dictionaries, explanatory materials
|
| -- README.md
| -- .gitignore
| -- ToDo                         <- bug fixes, future enhancements
| -- LabNotebook                  <- chronological records of project
```

principle 4: keep documentation

- **data**
  - **origin** and **schema** of your data sources (raw, first-to-third party)
  - data **dictionaries**
  - **other** relevant documentation
- **process** (LabNotebook)
  - what have you **tried** so far (and results)
  - what have you **not tried** but would if you had more time
- **ToDo**
  - **fixes** to current codebase
  - **improvements** to current codebase

principle 5: keep records of your **session**

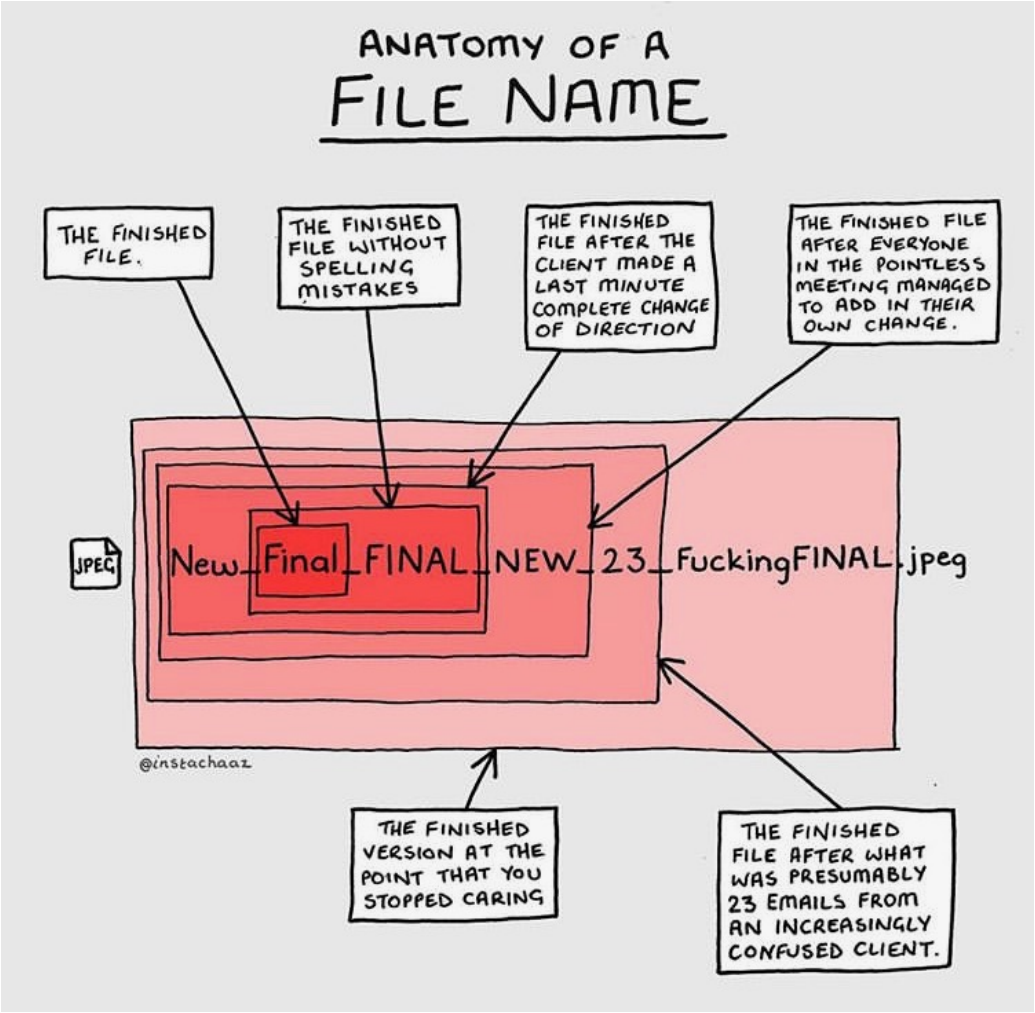
```
session_info.show(dependencies=True)

**Output:**
-----
matplotlib          3.7.2
numpy               1.24.3
pandas              2.0.3
session_info        1.0.0
-----
Python 3.11.5 | packaged by conda-forge | (main, Aug 27 2023, 03:34:09) [GCC 11.4.0]
Linux-5.15.0-1041-aws-x86_64-with-glibc2.35
Session information updated at 2024-02-14 15:30
-----
Dependencies
-----
asttokens           NA
certifi             2023.07.22
charset_normalizer  3.2.0
cycler              0.11.0
dateutil            2.8.2
```

principle 5: keep records of your **session**

- always keep a record of your **session**'s:
  - programming **language** and **version**
  - **libraries** / **packages** (versions, dependencies)
  - **OS**
  - **hardware**
- you'll need this info to **scale** or **hand off**:
  - `pip freeze > requirements.txt`

principle 6: use informative **script names**



principle 6: use informative **script names**

- provide detailed information about **purpose** of script / notebook in file name
  - use `clean_updated_data.py` instead of `untitled1.py`
- less is not more in this case
  - use `download_and_transform_dialy_data_updates.py` instead of `etl.py`
- use _ as needed to **separate words**
  - use `download_and_transform_dialy_data_updates.py` instead of `donwloadandtransfrormdailydataupdates.py`

- **good file name**: informative + easy to read + appropriate length

principle 7: always keep **portability** in mind

- use **relative paths** in your scripts `"data\raw\file.csv"` as opposed to `"C:\users\data\raw\file.csv”`
- or leverage your language’s tools:

```
from pathlib import Path
# Creating a path
my_path = Path("data") / "raw" / "file.csv"
```

principle 8: follow **workplace** ***idioms***

- make sure to understand all ***idioms*** at your workplace
  - **best practices**
  - **conventions**
  - **styles**
  - **ad hoc rules**
- **specifics** may be unique to your workplace, general principles apply universally

your (local) **workspace** in a nutshell…

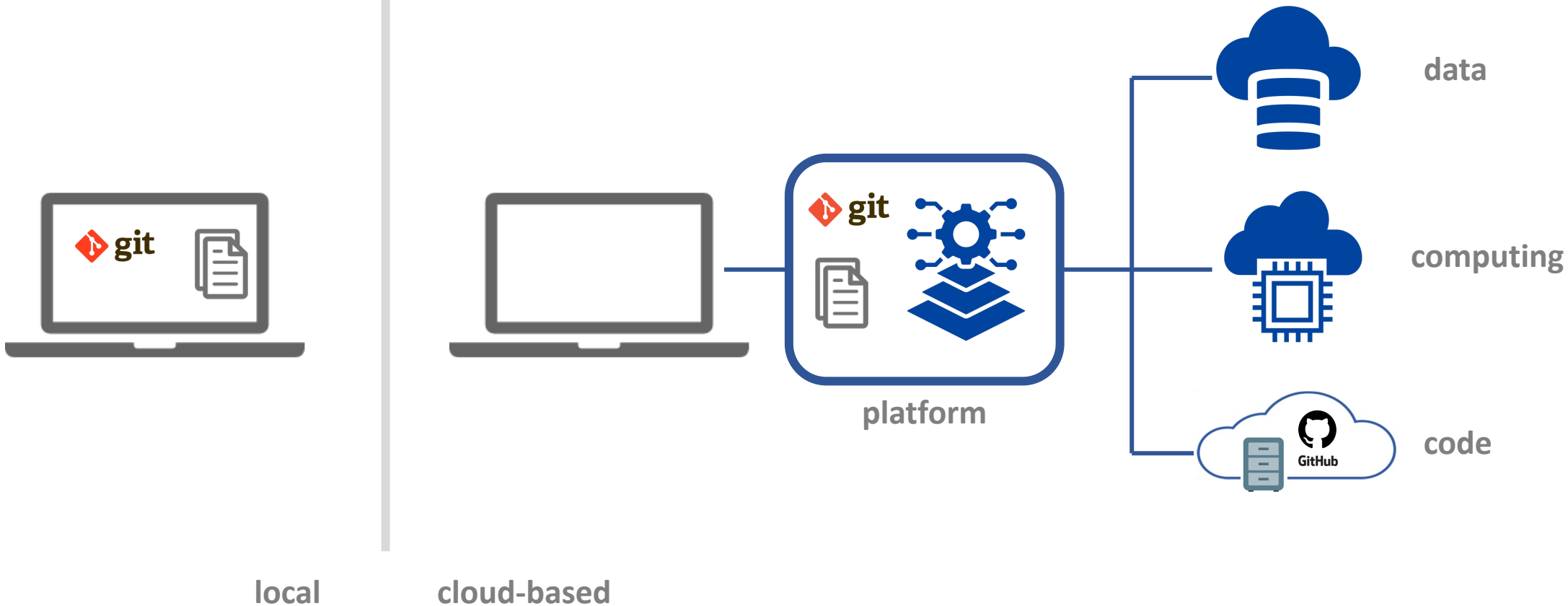
```
workspace/
|
| -- src/                    <- code
|    | -- data/              <- code to read raw data
|    | -- features/          <- code to transform data
|    | -- models/            <- code to model data
|    | -- visualizations/    <- code to create visualizations
|    | -- functions/         <- scripts to centralize functions
|    | -- config/            <- configuration files
|
| -- data/                   <- inputs
|    | -- raw/               <- original, immutable data dump
|    | -- external/          <- data from third party sources
|    | -- interim/           <- intermediate transformed data (for validation)
|    | -- processed/         <- final processed data
|
| -- reports/                <- outputs
|    | -- notebooks/         <- documents synthesizing analysis, experiments, etc
|    | -- figures/           <- images generated by code
|
| -- references/             <- data dictionaries, explanatory materials
|
| -- README.md               <- high-level project description/overview
| -- .gitignore              <- files to exclude from version control
| -- ToDo                    <- bug fixes, future enhancements
| -- LabNotebook             <- chronological records of project
```

but not all **workplaces** are **local**…

your workspace in real life

your workspace

also your workspace



with a little help from our GitHub friends…

what gets pushed to GitHub

```
workspace/
|
| -- src/
|    | -- data/                  <- code to read raw data
|    | -- features/              <- code to transform data
|    | -- models/                <- code to model data
|    | -- visualizations/        <- code to create visualizations
|    | -- functions/             <- scripts to centralize functions
|    | -- config/                <- configuration files
|
| -- README.md
| -- .gitignore
```

what gets `pushed` to GitHub

- your `README.md` file should provide a **general overview** of your project
  - short (executive) **summary**
  - **structure** (navigation)
  - **instructions**
  - **limitations**
- yes, ALWAYS keep a meaningful `.gitignore` in your repo

what gets `pushed` to GitHub

- `data` is **NEVER** pushed to GitHub!!!!!!
- `{secret keys}` are **NEVER** pushed to GitHub!!!!!!
- reports could live in GitHub (depends)
- references are transferred to GitHub **wiki**
- TODO is transferred to GitHub **issues** / **projects**

why is a workspace **structure** needed?

- structure **removes friction** for
  1. any **first-time user** of your codebase – **collaborators**, **handoffs**
  2. the **you of the future** (if/when you return to a project)
- creates a **shared understanding** of where everything is – and where to look for anything!
- backbone of **portability**, **reproducibility**, and **scalability!**

structuring your workspace: DS & DE/MLE perspectives

Marco Morales
marco.morales@columbia.edu

Nana Yaw Essuman
ne2388@columbia.edu

GR5069: Applied Data Science
for Social Scientists

Spring 2026
Columbia University