R in Pharma 2021

Data-as-a-Product:

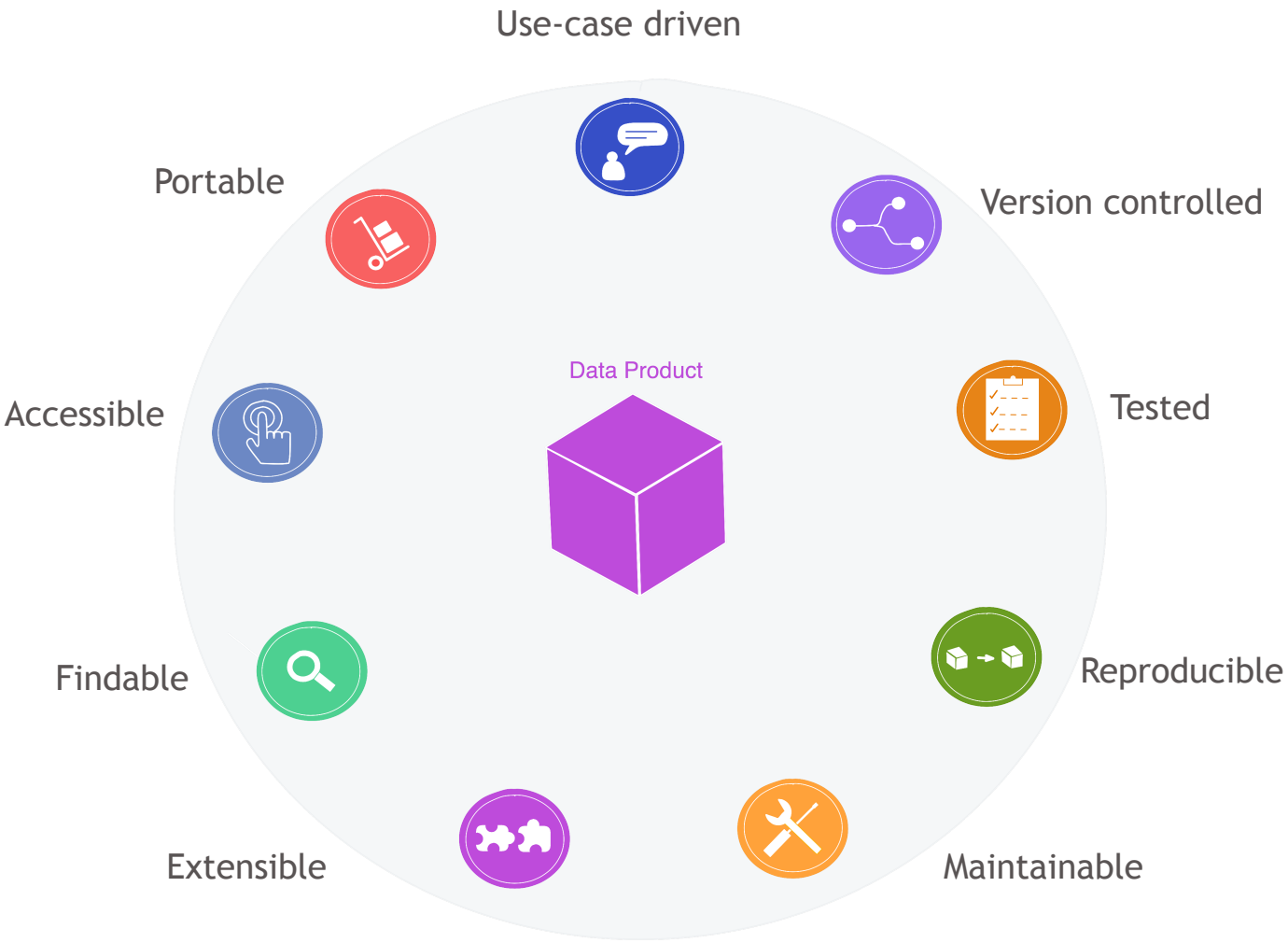
A data science framework for data collaborations

4 Nov 2021

Afshin Mashadi-Hossein, Julie Rytlewski, Garth McGrath

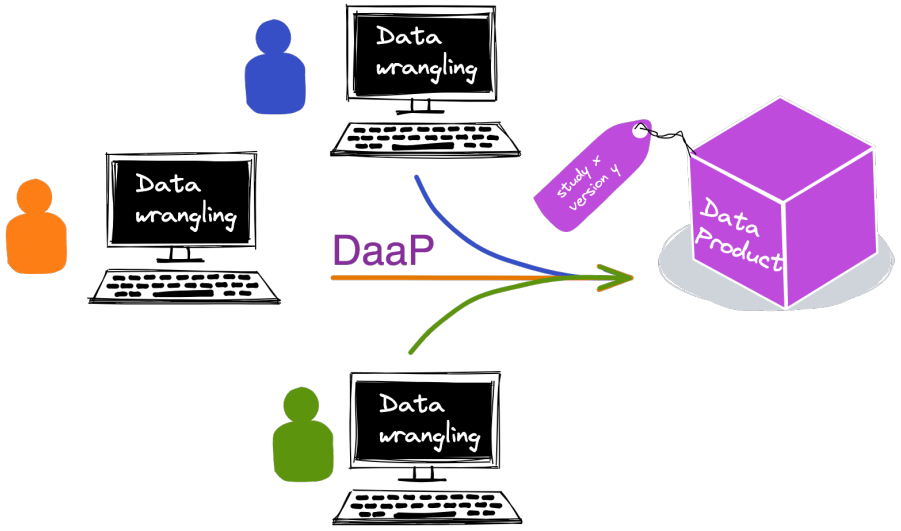
Bristol Myers Squibb™

Data Product



Data-as-a-Product

Data-as-a-Product (DaaP) is a framework to make it practical for data science teams to pool together their “data wrangling” efforts towards collaboratively building data products



Motivation ➔ Addressing the 80:20 rule of data science

How

Heroic data wrangling

"Free up" data science team by offshoring data building

Can science be missed?

New DTS

Dev. strict rules
Codify them in pipelines

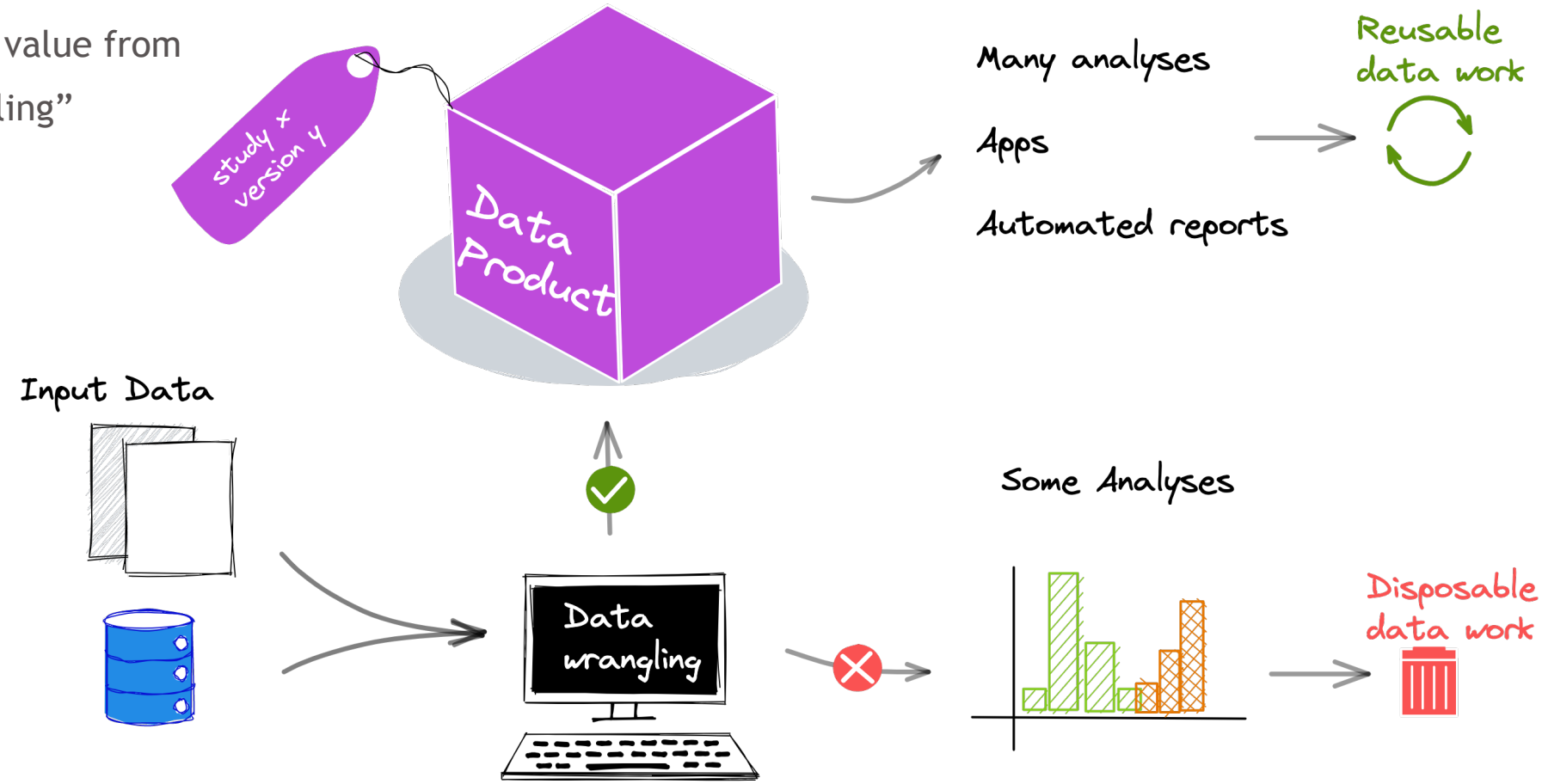
Derive more value from collaborative data processing
- Make collab. easy
- Add a dash of automated standards
- Put in place process framework for collab.

Switch data platform

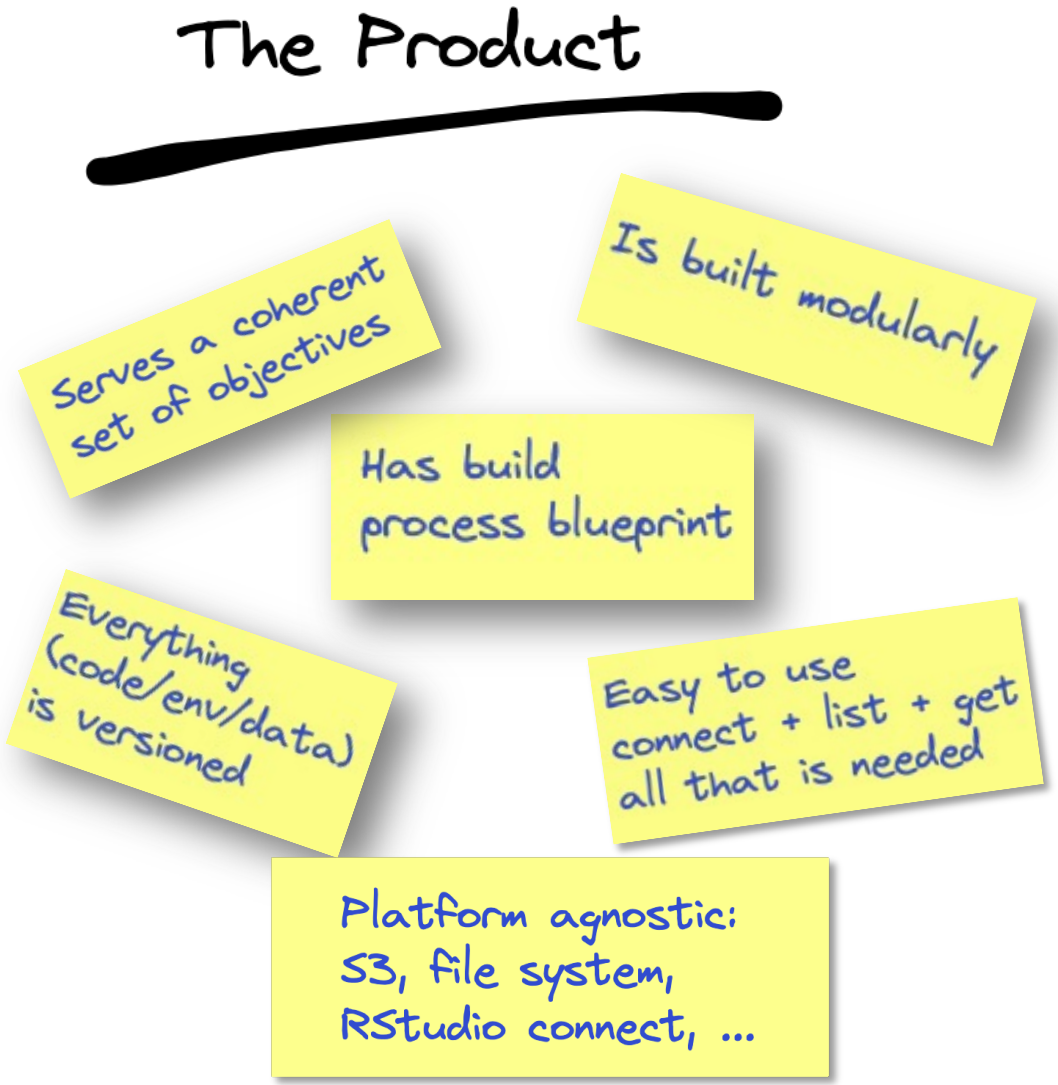
Motivation ➔ Addressing the 80:20 rule of data science

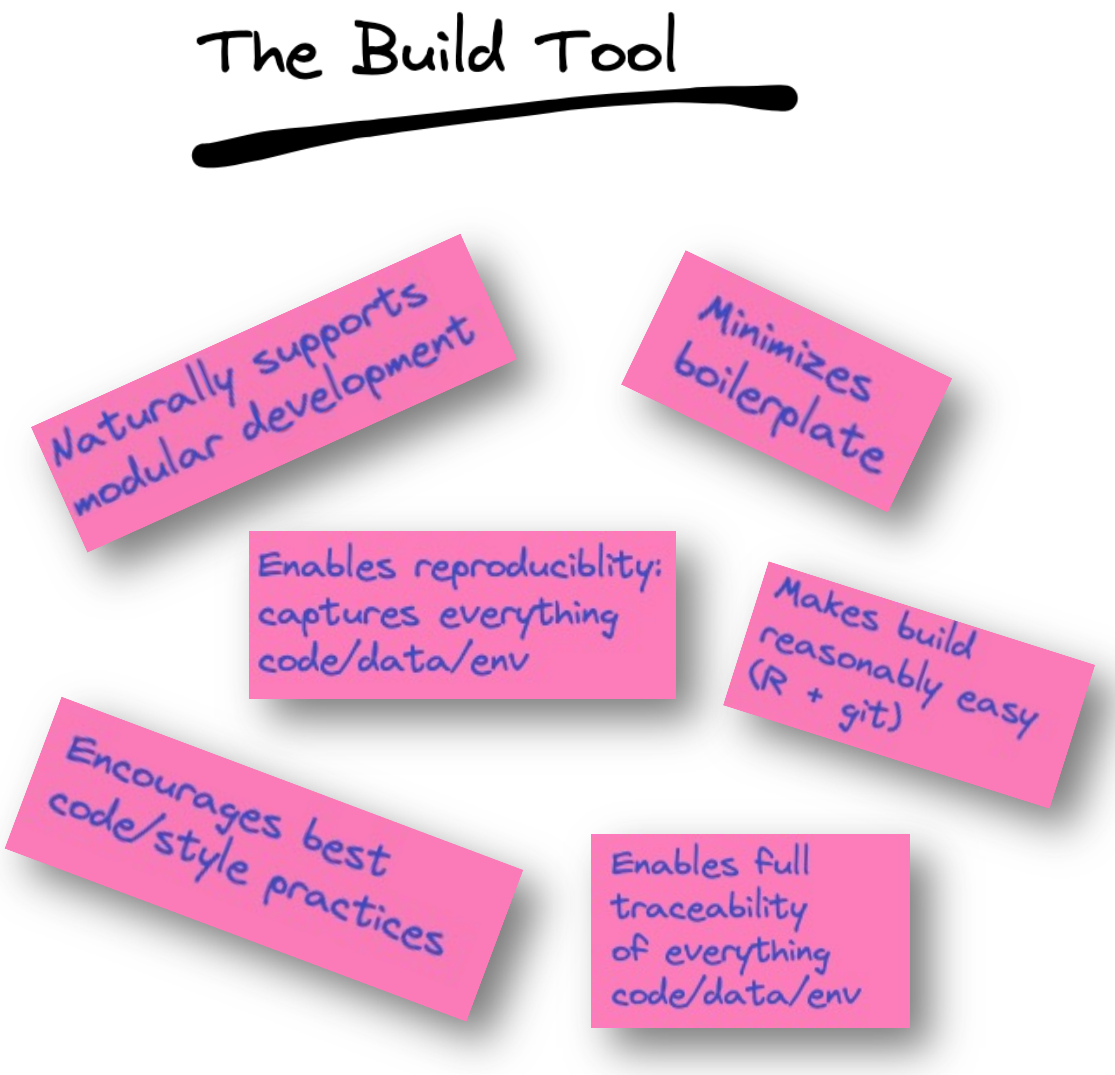
How

Derive more value from "data wrangling"



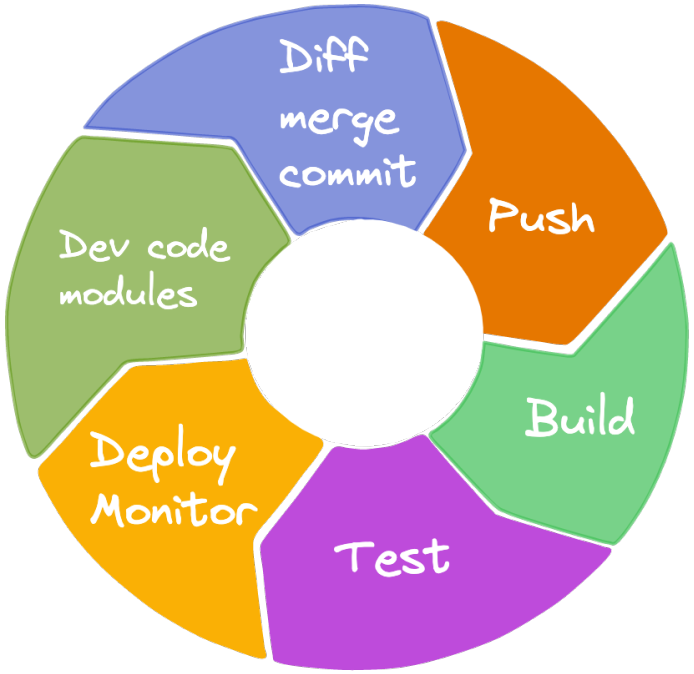
What to aim for





What to aim for

If it was just code

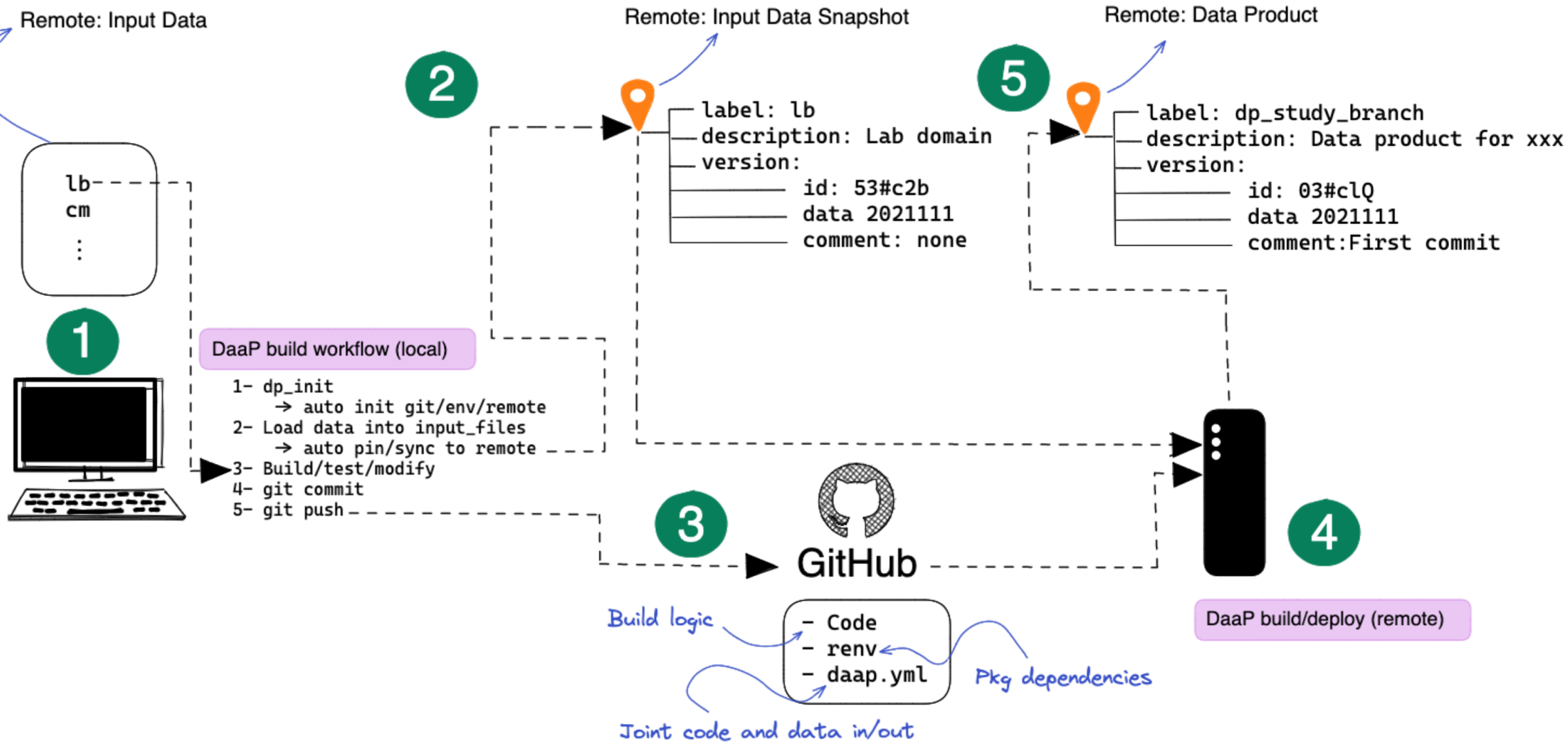


Inspiration ➔ capture data as code

The Build Tool

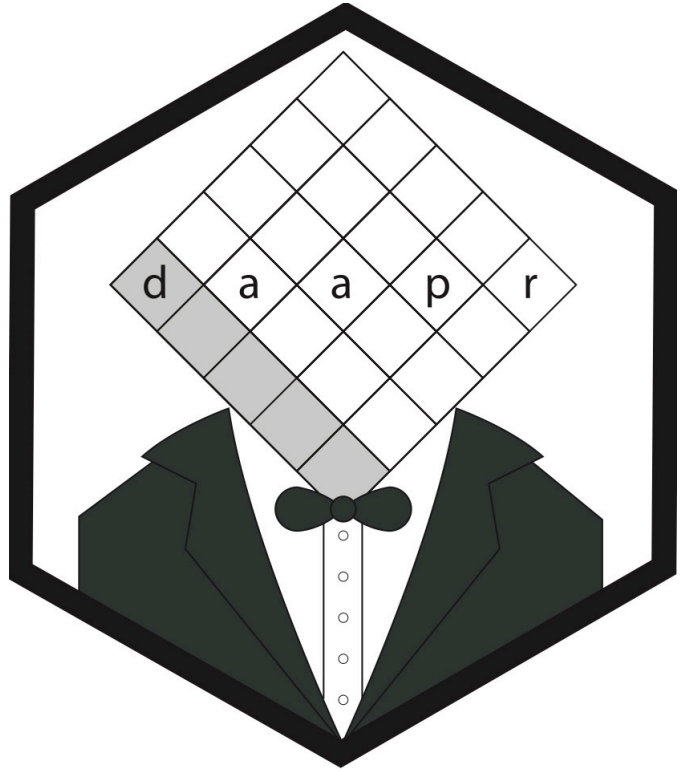
- Naturally supports modular development
- Minimizes boilerplate
- Enables reproduciblity: captures everything code/data/env
- Makes build reasonably easy (R + git)
- Encourages best code/style practices
- Enables full traceability of everything code/data/env

Concept: “Building” with data as code



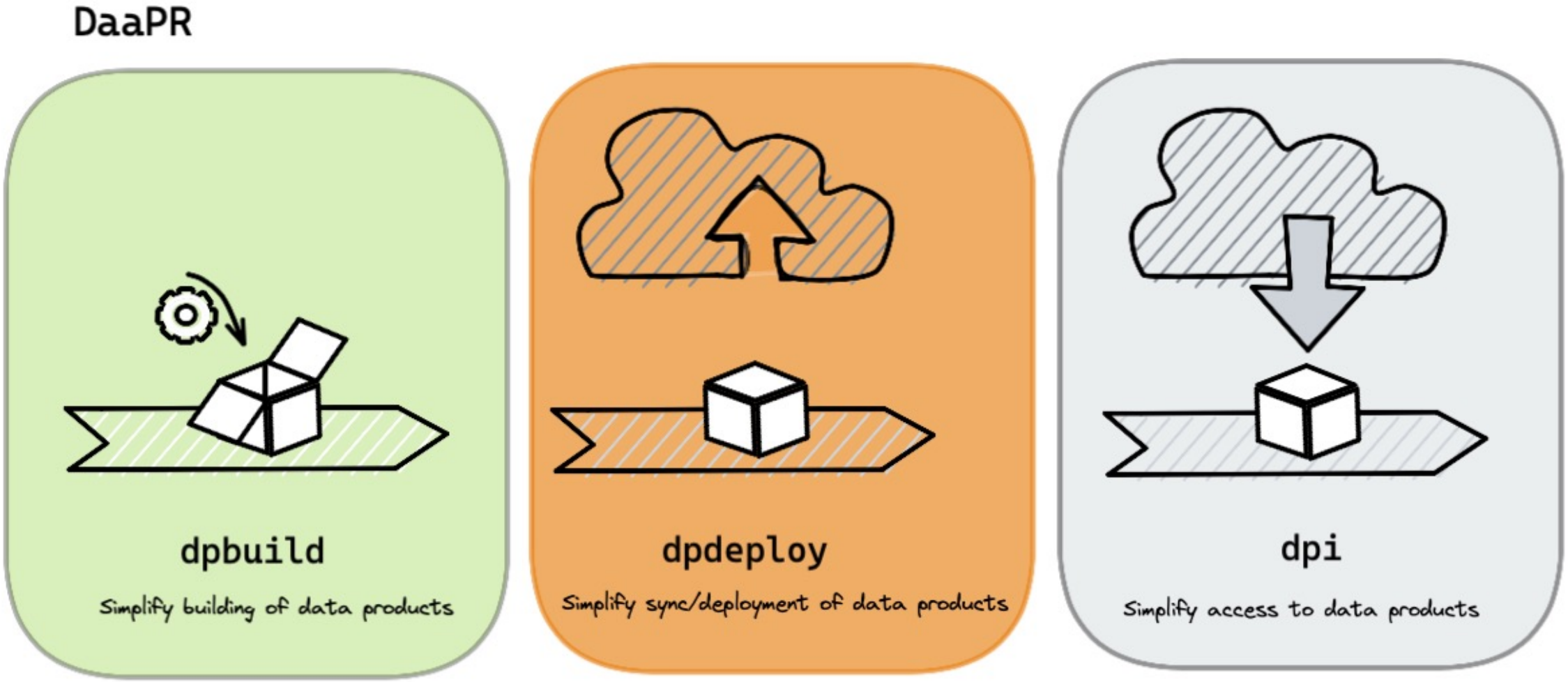
DaaPR:

R-based implementation of DaaP framework

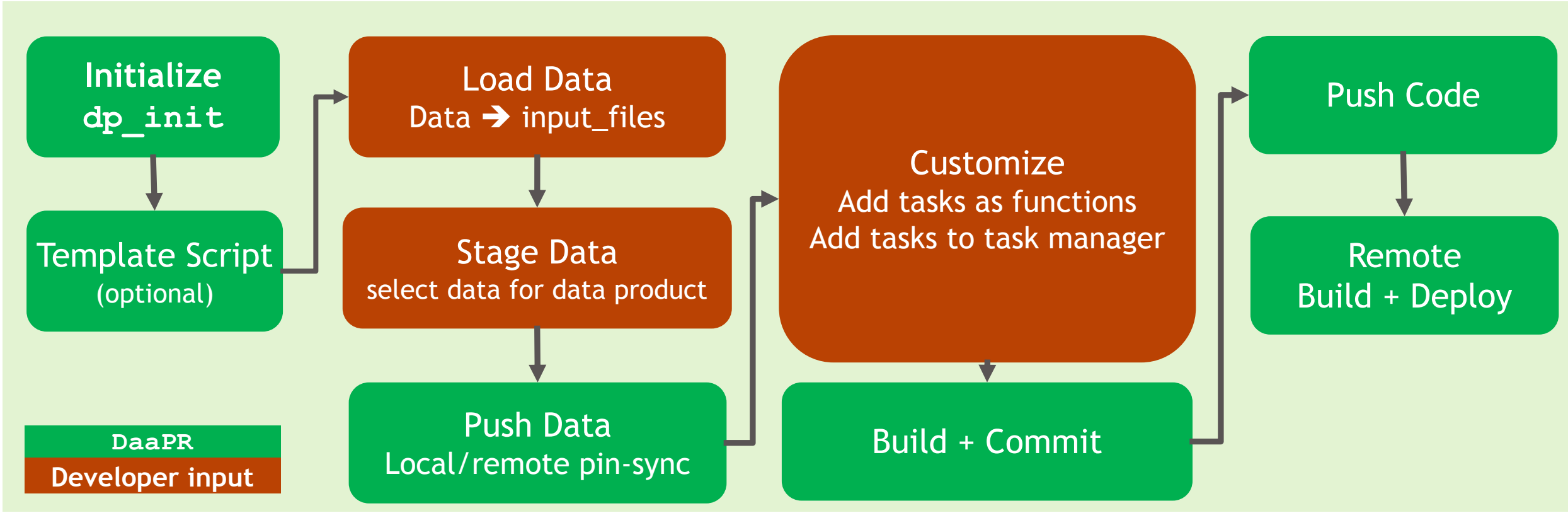


https://github.com/amashadihossein/daapr

DaaPR: Family of packages to implement DaaP framework



DaaPR workflow: putting it all together



Setup → Data → Code → Deploy

Set up

`dp_init`

↓

Sets up project folder

↓

Runs `git init`

↓

Runs `renv init`

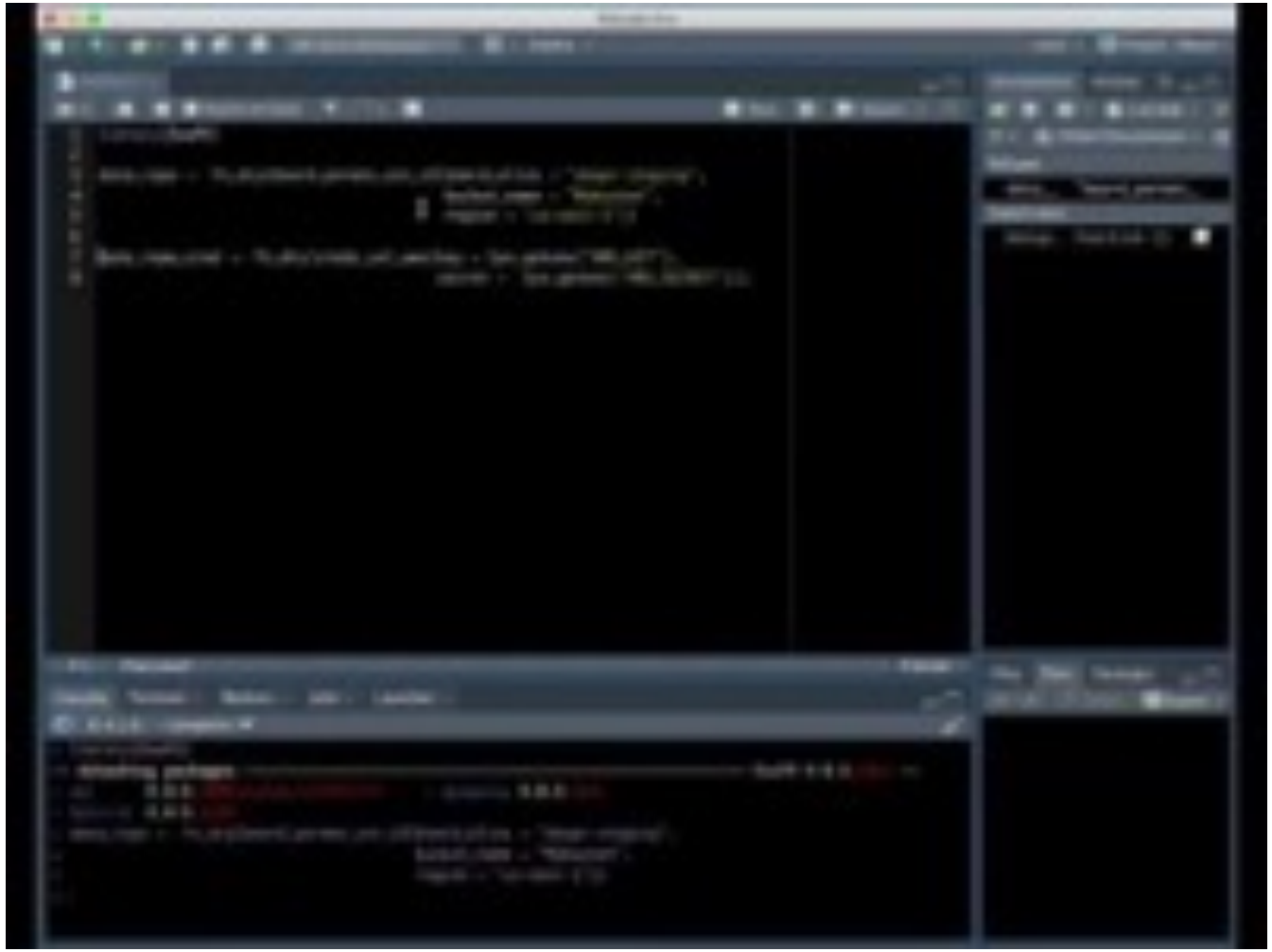
↓

Adds remote for code + data

↓

Records project metadata

Video Link



Onboard Data

Load data into `input_folder`

↓

Set promised environment vars

↓

`dpinput_map` Map input folder content

↓

Stage: select to sync

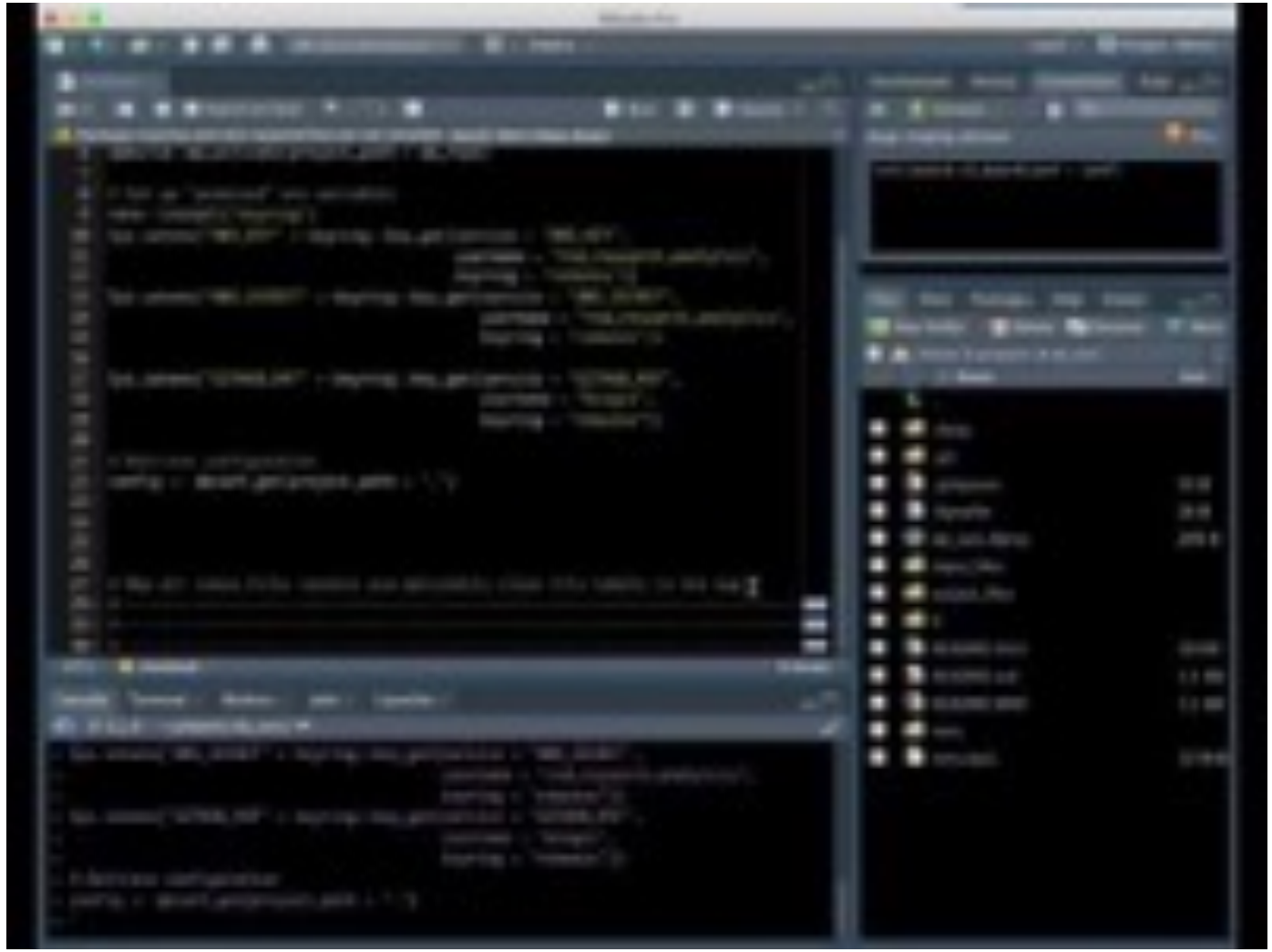
↓

`dpinput_sync` Sync input data to remote

↓

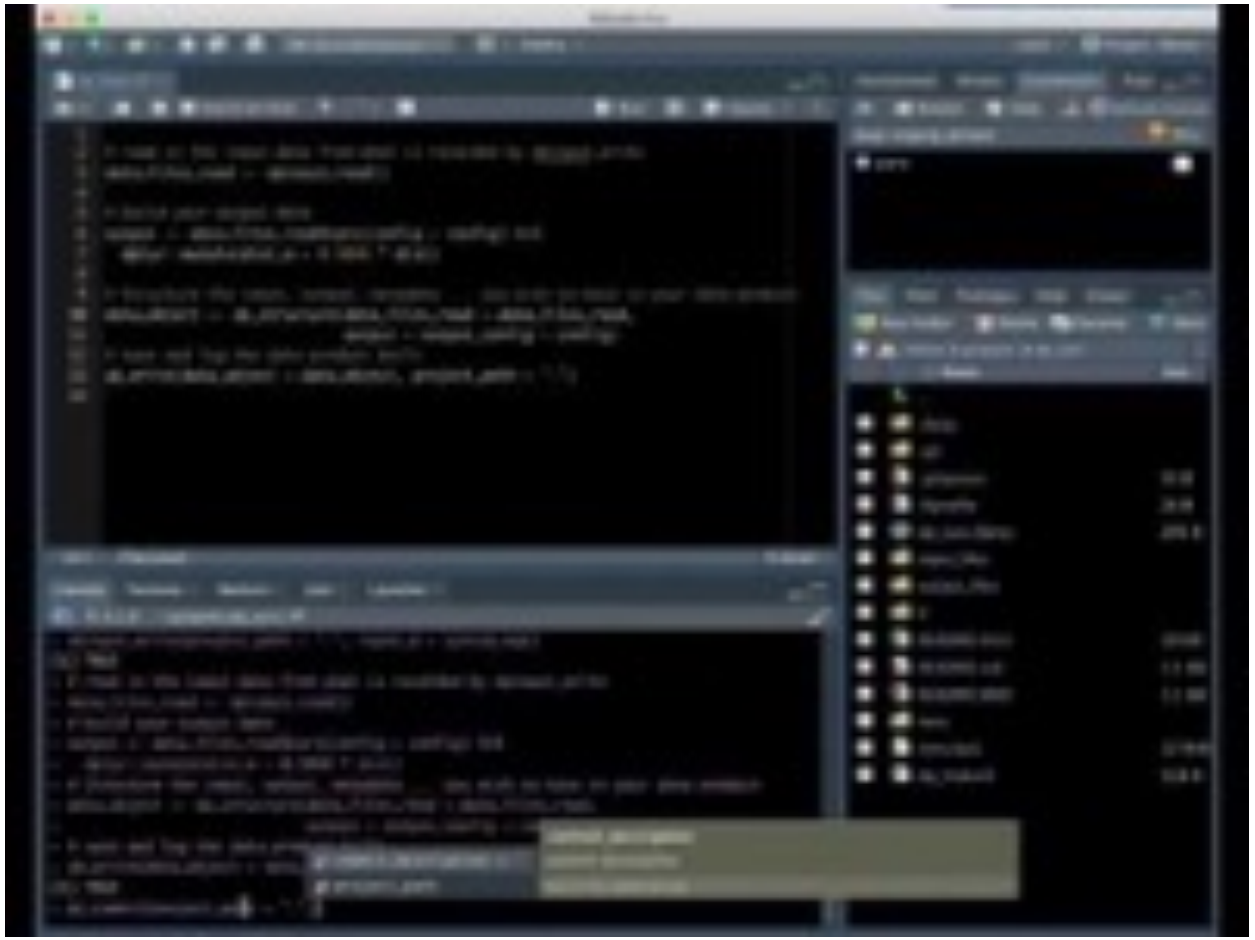
`dpinput_write` Write input: save sync records

Video Link



Build and push

Video Link



`dpinput_read`	Read synced input
	↓
	Add “wrangling” logic
	↓
`dp_structure`	Assemble data object
	↓
`dp_write`	Write data product: Log data product
	↓
`dp_commit`	Commit
	↓
`dp_push`	Push

Deploy

`dp_deploy`

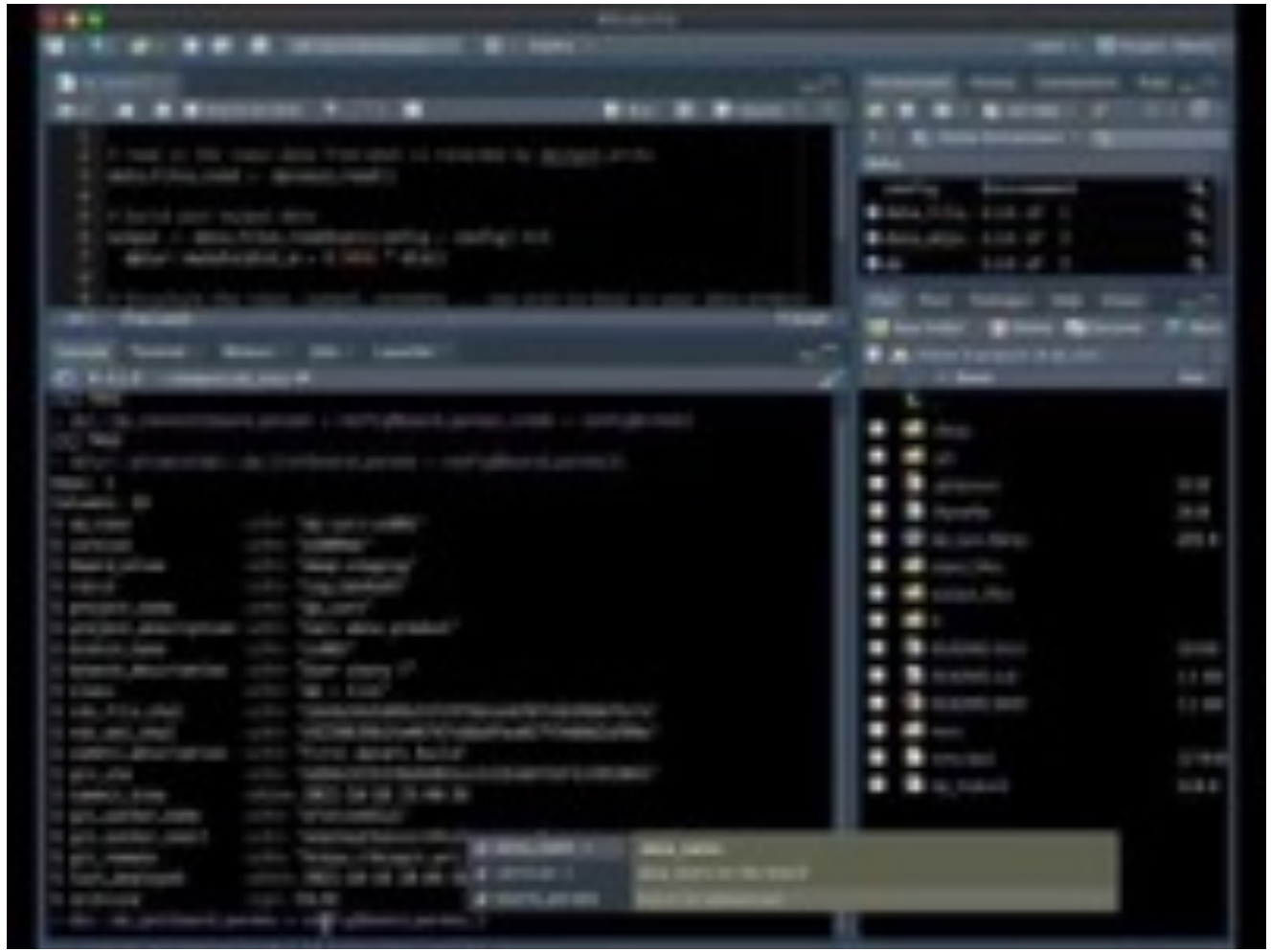
How to access:

`dp_connect` Logs in to remote data

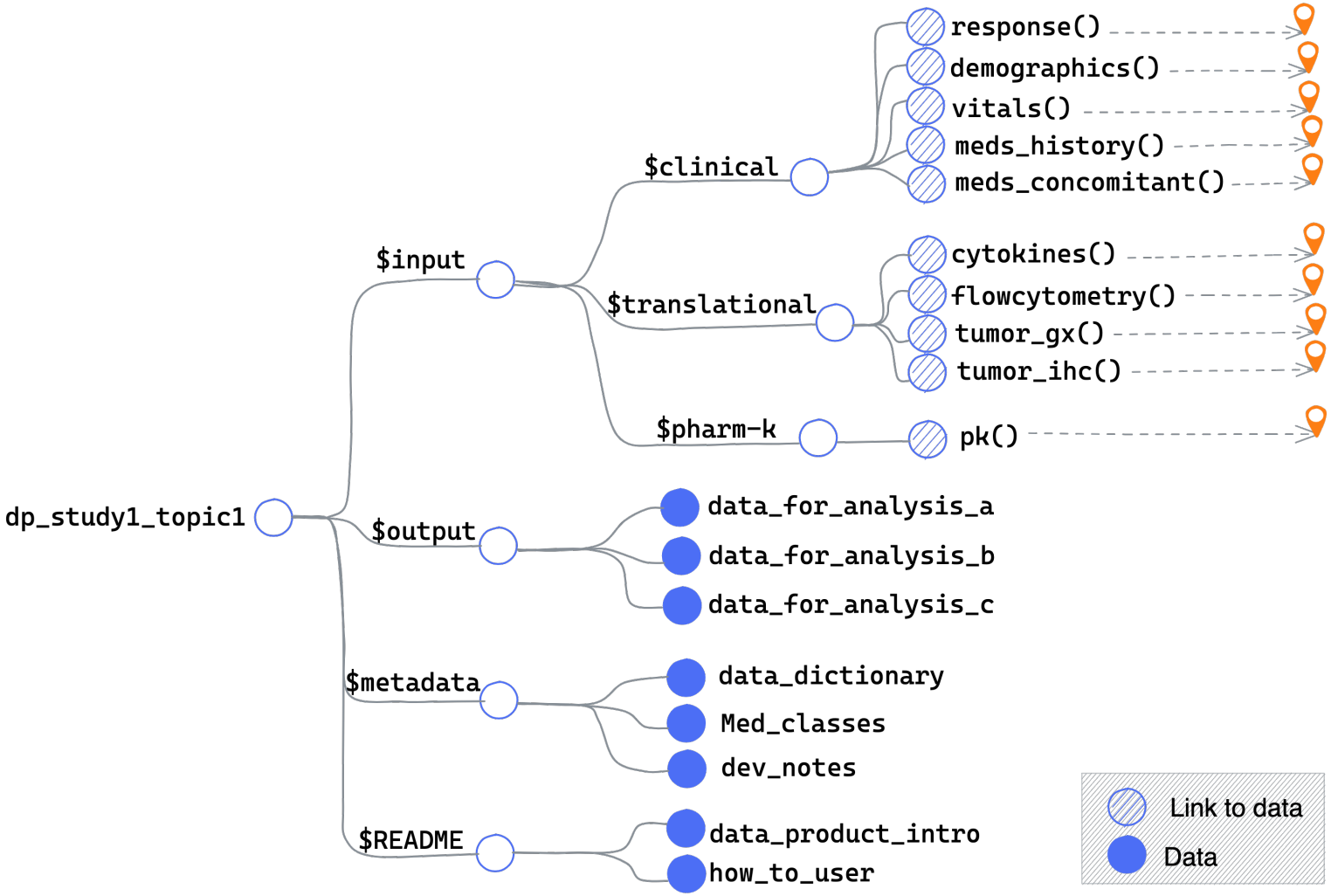
`dp_list` Lists: available data + metadata

`dp_get` Retrieves data by id + version

Video Link



Data Object: An example



Limitations and considerations

- **DaaPR does not natively impose or accommodate relational data models**

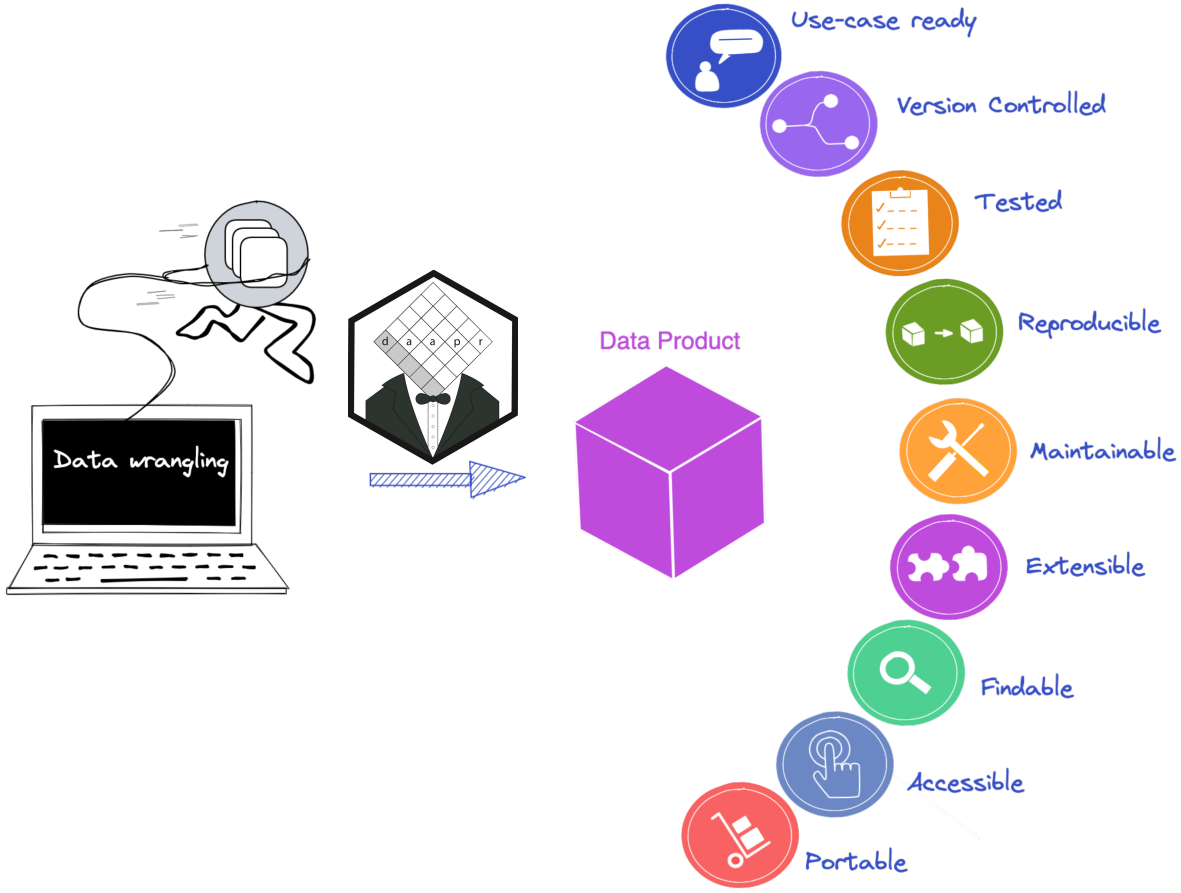
- **DaaPR is not intended for large data tables**
  - DaaPR uses pins under the hood
  - Ex: 10 x 300 MB files works great. 1 x 3GB not ideal

- **DaaPR does not address data governance**
  - Data platforms with governance management complement DaaPR

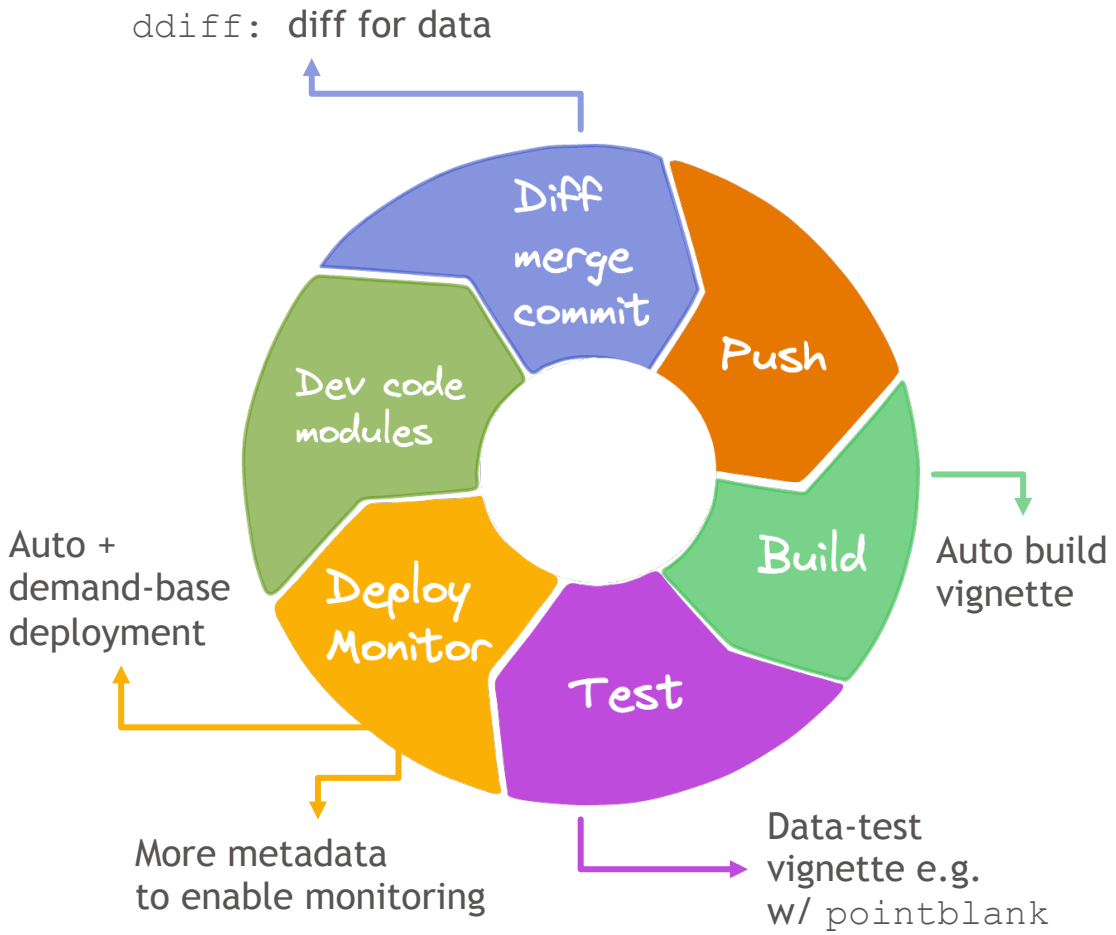
- **DaaPR and the packages it is built upon are very young**
  - `DaaPR` packages are not yet mature
  - Even packages like `pins` and `renv` are still very young

Summary

- DaaP framework can enable data science teams to trade up on data wrangling



Ongoing Developments



Acknowledgements

- R in Pharma organizers and community!
- BMS Informatics and Predictive Sciences:
  - Yue Jiang
  - Ron Hause
  - Mandeep Takhar
  - Michael Penhallegon
  - Cole Sobel
  - Stefan Ponko
  - Aditya Radhakrishnan

… and thank you!