rstudio::conf
from RStudio

Best Practices for working with Databases

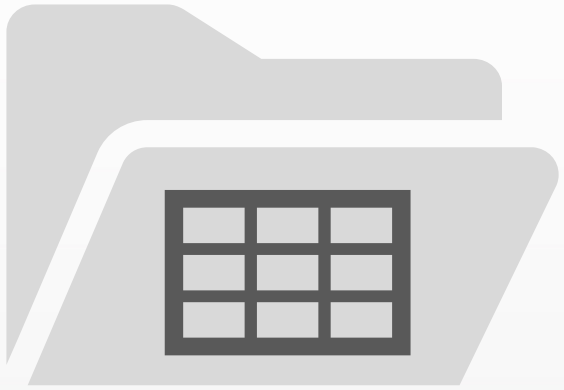
Typical DS project

Import Wrangle Learn Share

Remote Data Sources

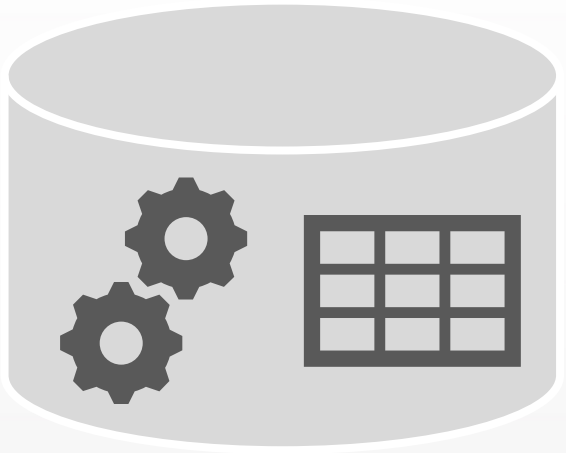
Flat Files

Only Data

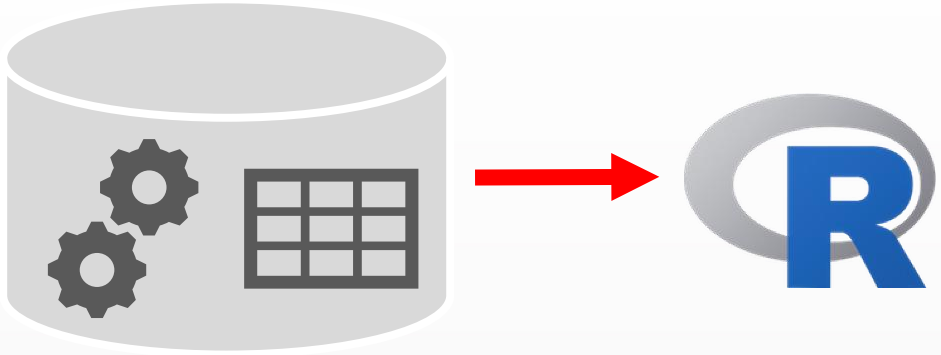


Remote Sources

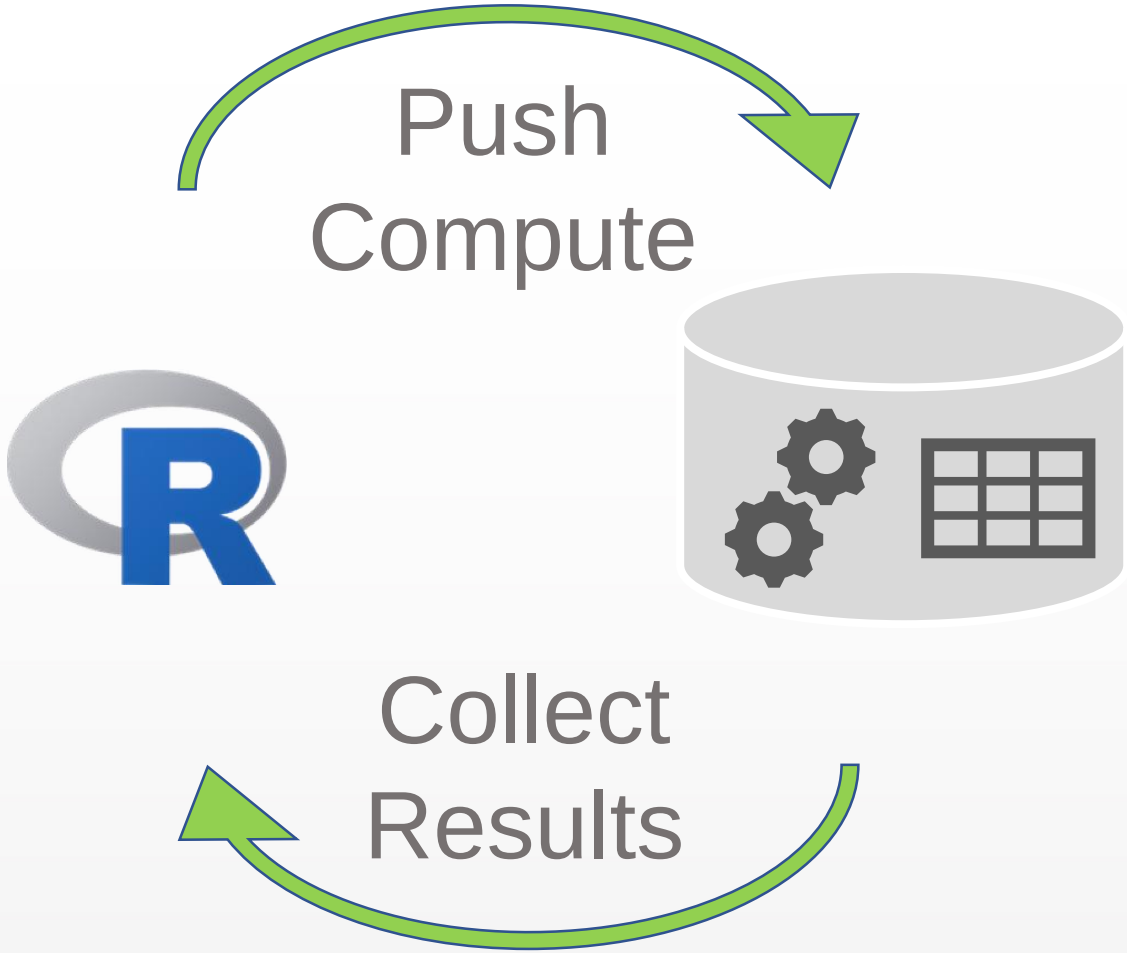
Data & Compute engine



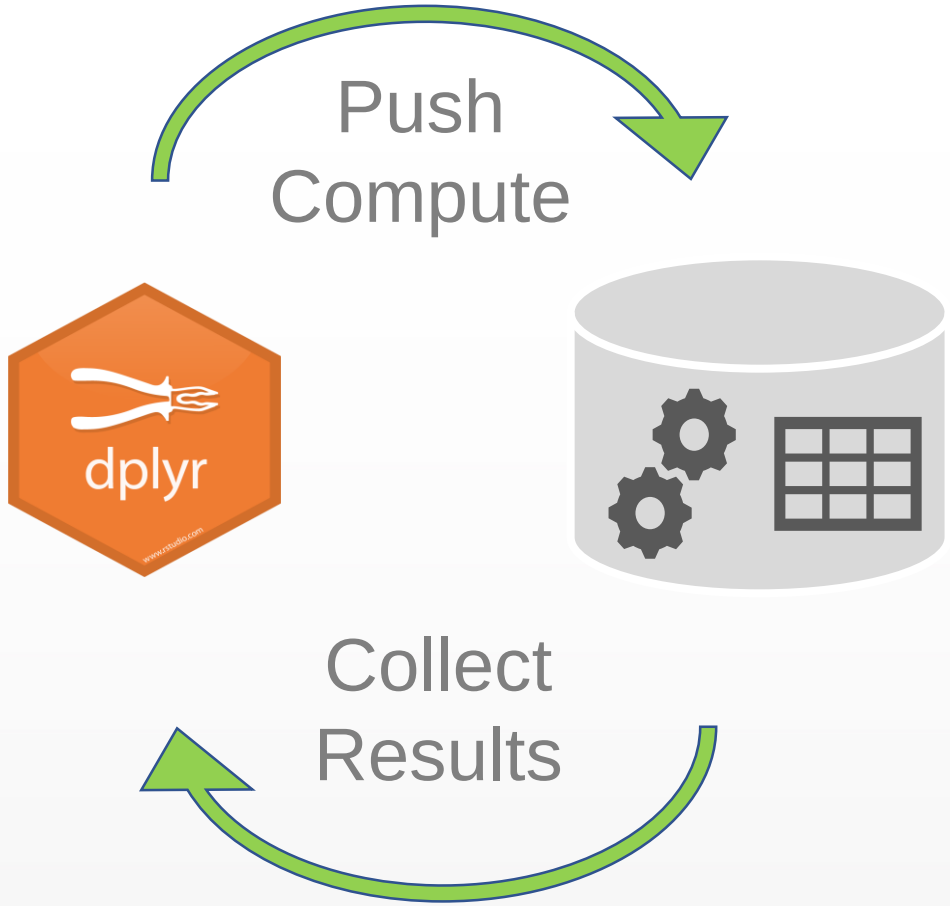
Wrangle inside the DB



Extract
Data

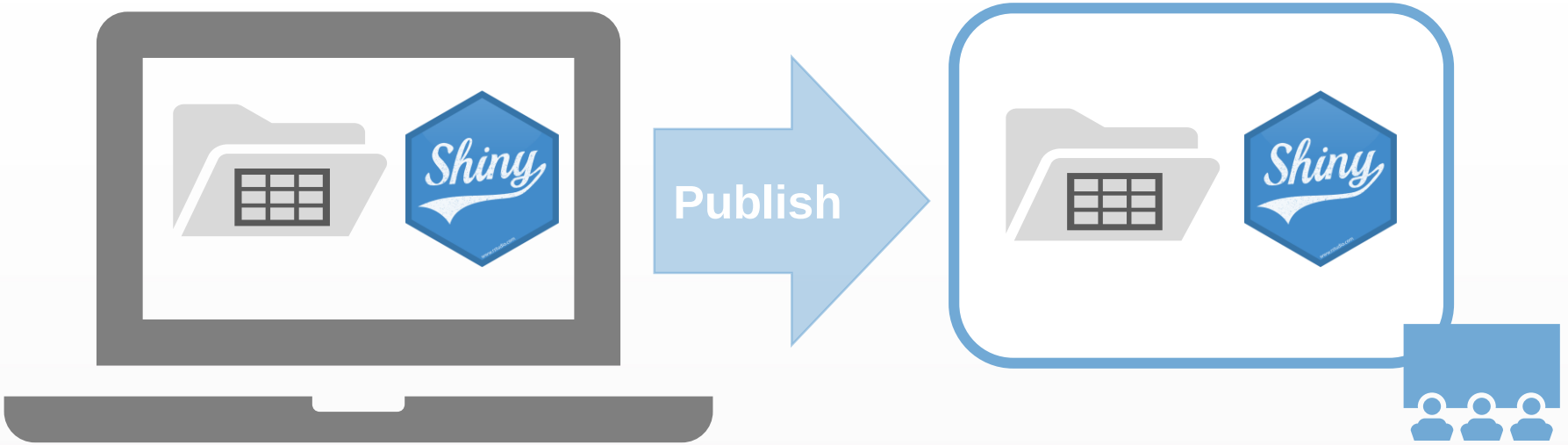


Advantages

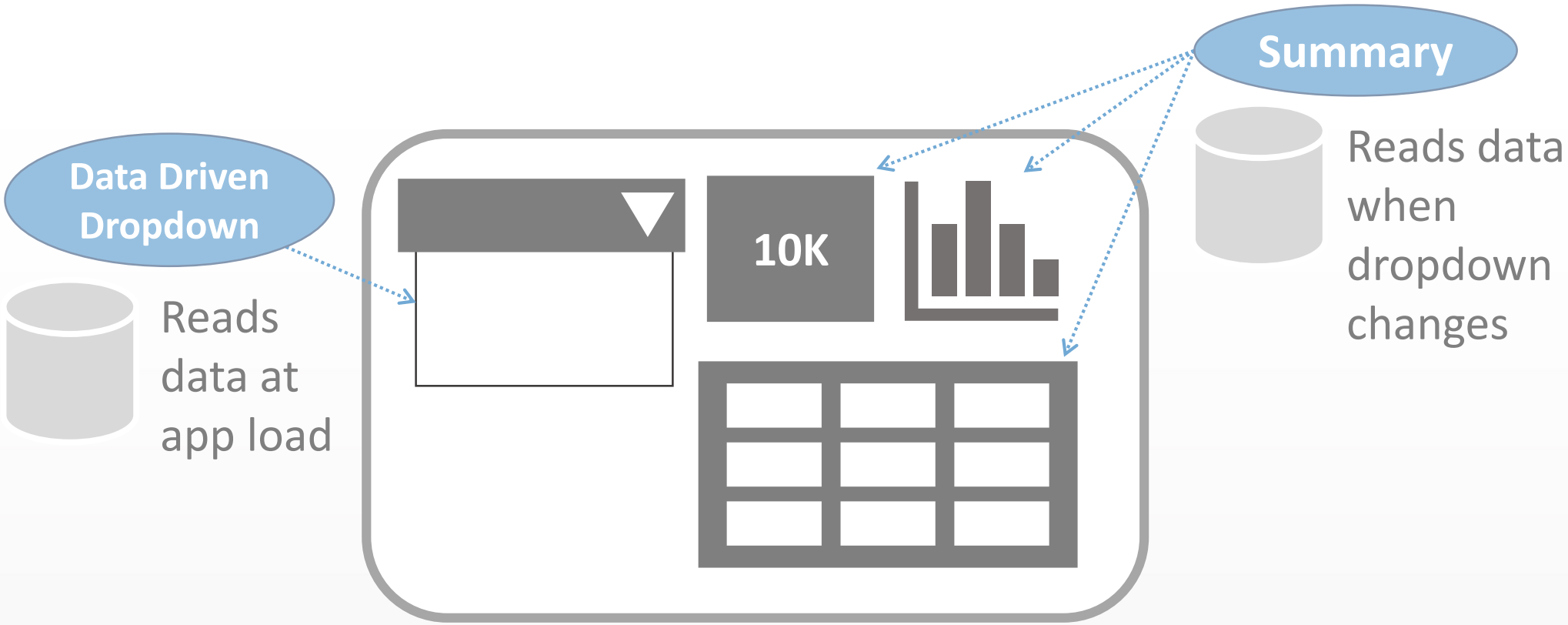


1. dplyr translates to SQL
2. Take advantage of piped code
3. All your code is in R!

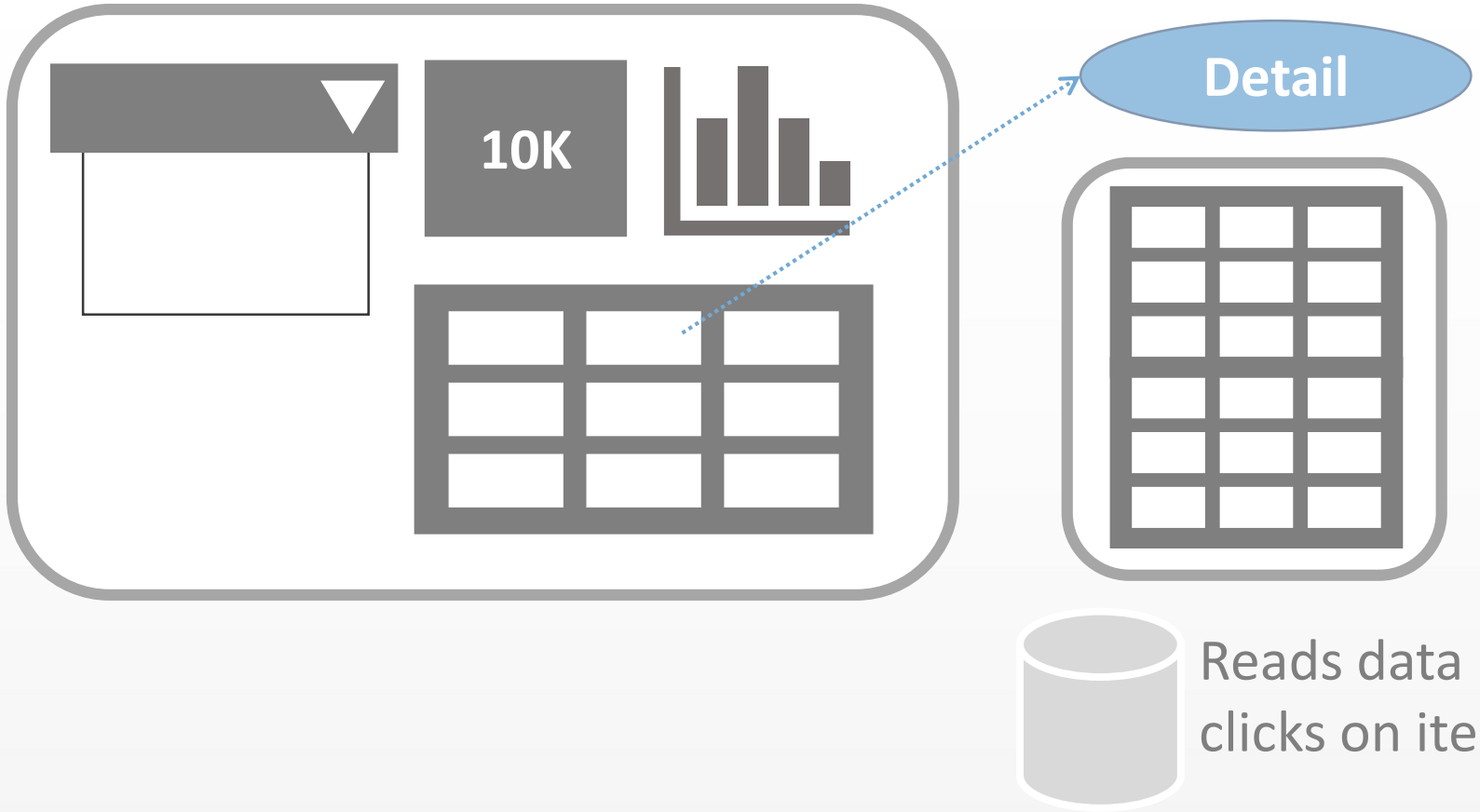
Normal Shiny app



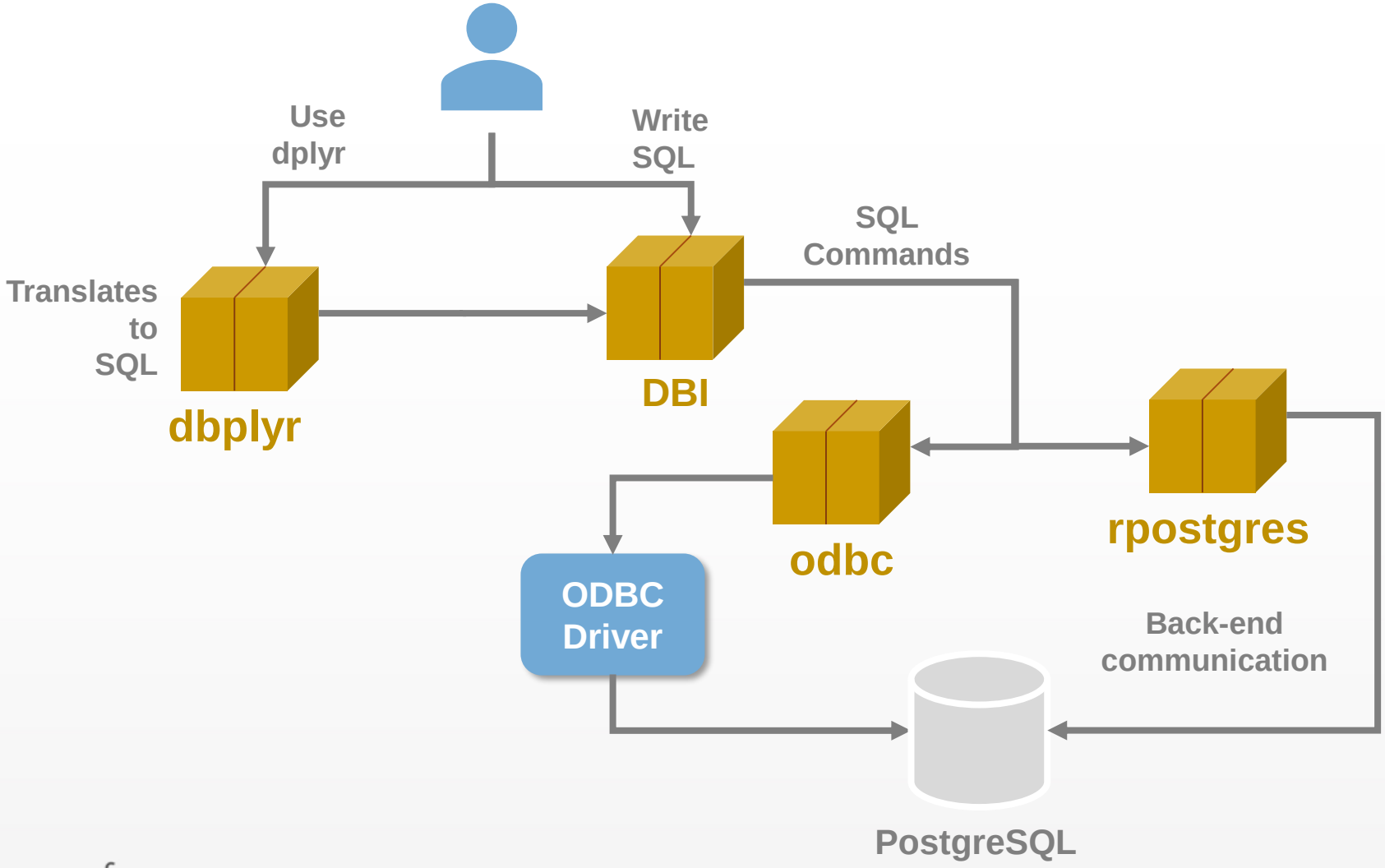
Database + Shiny Dashboard



Database + Shiny Dashboard



Architecture



DS project using DBs

Access Wrangle Learn Share

In the database

Options to Push Compute

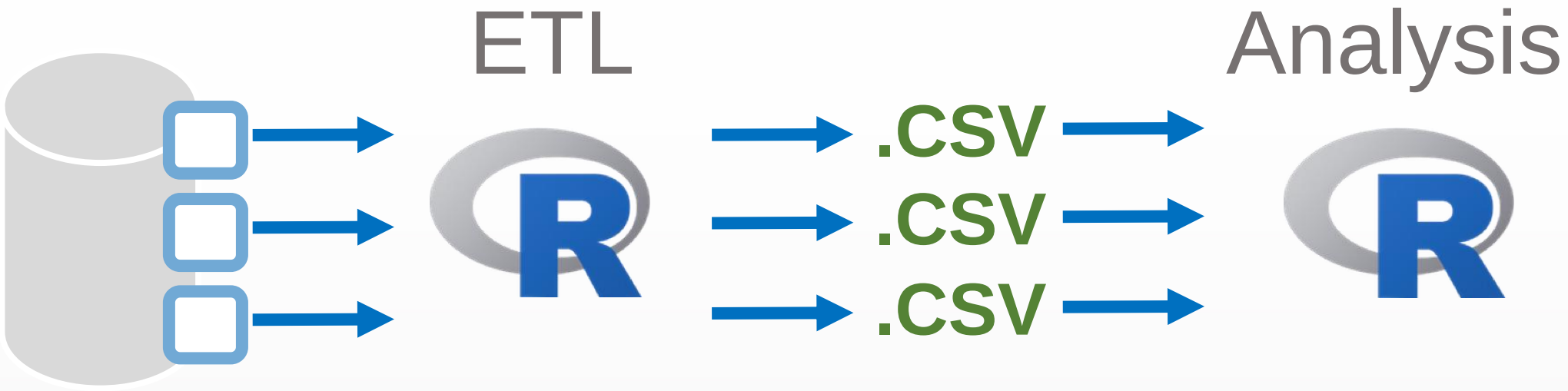
Write SQL statements

```
SELECT "name",
COUNT(*) AS "n"
FROM "vwFlights"
GROUP BY "name"
```

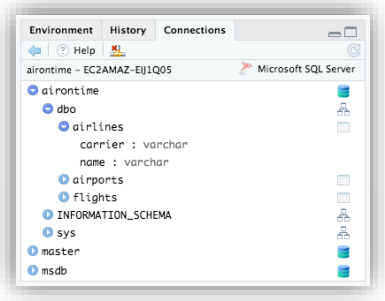
Use dplyr verbs

```
flights %>%
  group_by(name) %>%
  tally()
```

“The Method”



RStudio's approach to Databases

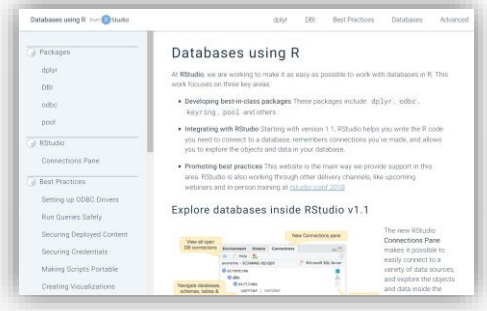


1. RStudio v1.1 Integration

- View databases, schemas, tables, and fields
- Explore data in tables or views
- Remembers connections you've made

2. Utilize best-in-class packages

- dplyr
- odbc
- DBI



3. Promoting best practices

- db.rstudio.com
- Training & presentations
- Blog posts (rviews.rstudio.com)

Translations available in *dbplyr*

1. Microsoft SQL Server
2. Oracle
3. Apache Hive
4. Apache Impala
5. PostgreSQL
6. MS Access
7. MariaDB (MySQL)
8. SQLite
9. Amazon Redshift
10. Teradata
11. Big Query