Time Travel with R

May 1st - Sean Lopp

RStudio®

Past, Present, and Future

- Code that used to run no longer runs, even though the code has not changed.
- Typing `install.packages` in your environment doesn't do anything, or doesn't do the right thing
- You are afraid to upgrade or install a new package, because it might break your code or someone else's.

RStudio®

Environment

noun

The software your code depends on, including: R, R packages, system dependencies, and the operating system.

Game Plan

- **Strategies** for creating reproducible environments
- **Use cases** for reproducible environments
- **Tools** to implement a strategy for a use case

RStudio®

Game Plan

- **Strategies** for creating reproducible environments
- **Use cases** for reproducible environments
- **Tools** to implement a strategy for a use case

I want _____ with _____ using _____
(use case) (strategy) (tools)

RStudio®

Game Plan

- **Strategies** for creating reproducible environments
- **Use cases** for reproducible environments
- **Tools** to implement a strategy for a use case

I want _____ with _____ using _____
(use case) (strategy) (tools)

I want to bake a cake with Mary Berry's recipe using an oven.

RStudio

Game Plan

- **Strategies** for creating reproducible environments
- **Use cases** for reproducible environments
- **Tools** to implement a strategy for a use case

I want _____ with _____ using _____

(use case) (strategy) (tools)

I want to bake a cake with Mary Berry's recipe using an oven.

RStudio

Game Plan - Specifics

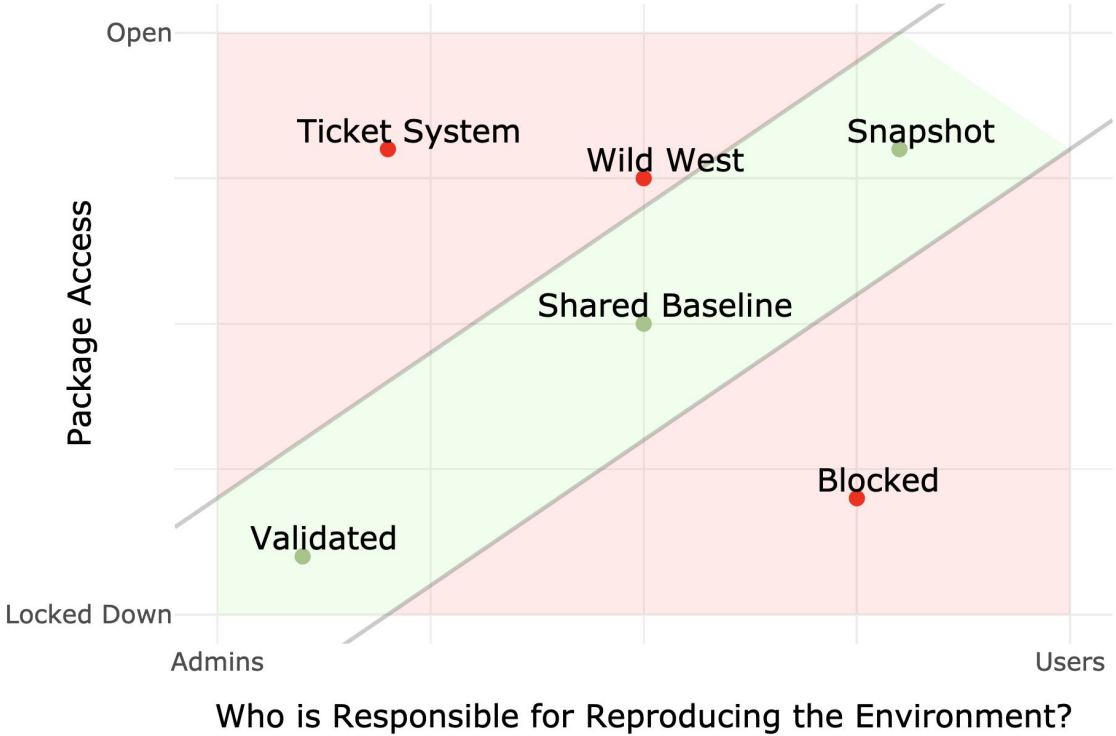
- Strategy Map

- ***Collaborating on a team*** with a **Shared Baseline** using a **Frozen Repository**
- ***Safely upgrading packages*** with **Snapshot and Restore** using **renv**
- ***Using approved packages*** with **Validation** using **Docker**

https://environments.rstudio.com/

RStudio

Strategy Map

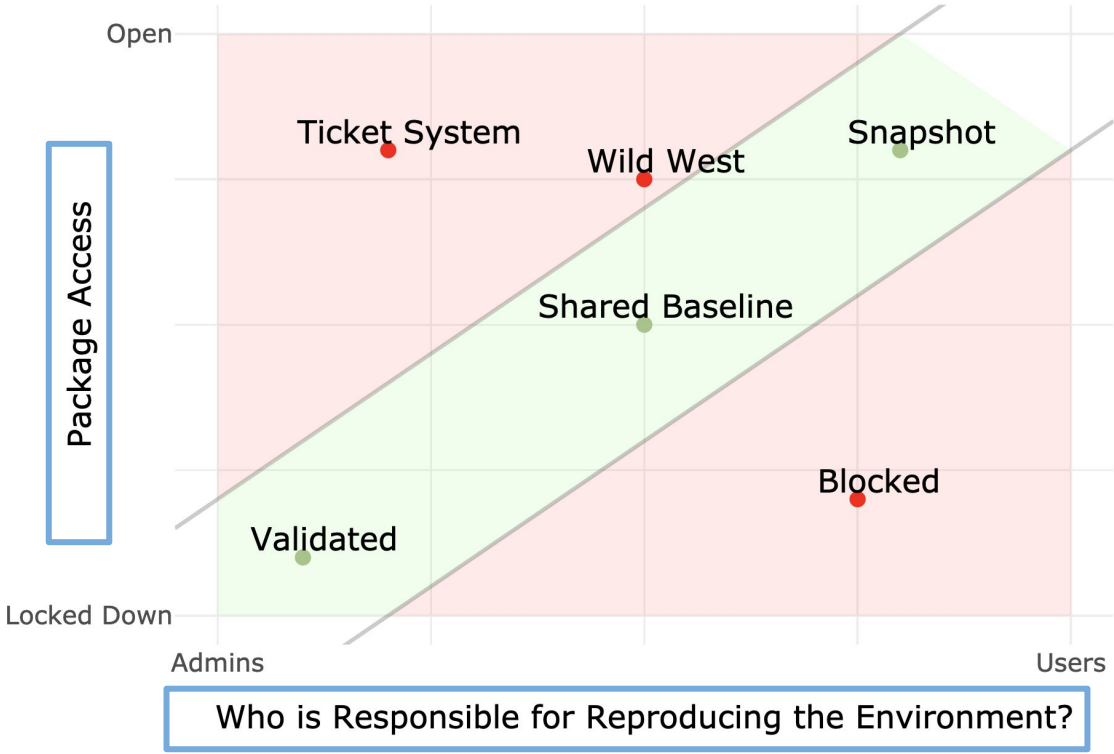


RStudio

Strategy Map

1. Who is responsible?

2. Are there restrictions? (e.g. Licensing, approval, test coverage)

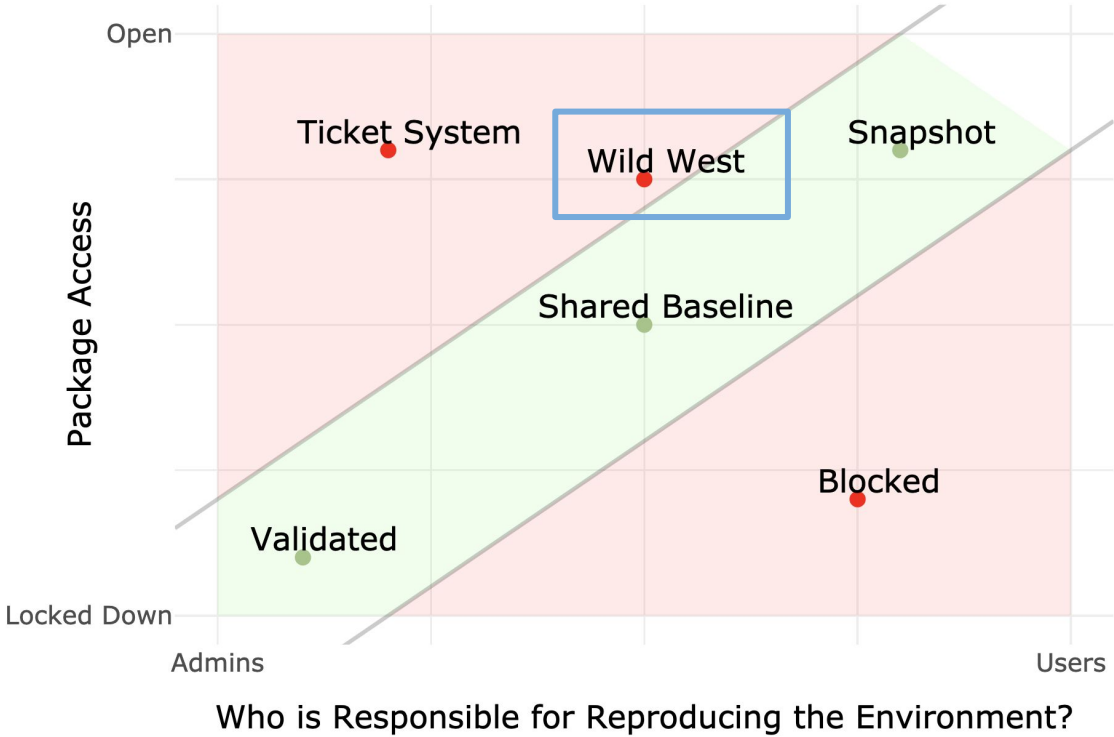


RStudio

Strategy Map

Wild West

- Where we learn
- Open access
- No one is responsible

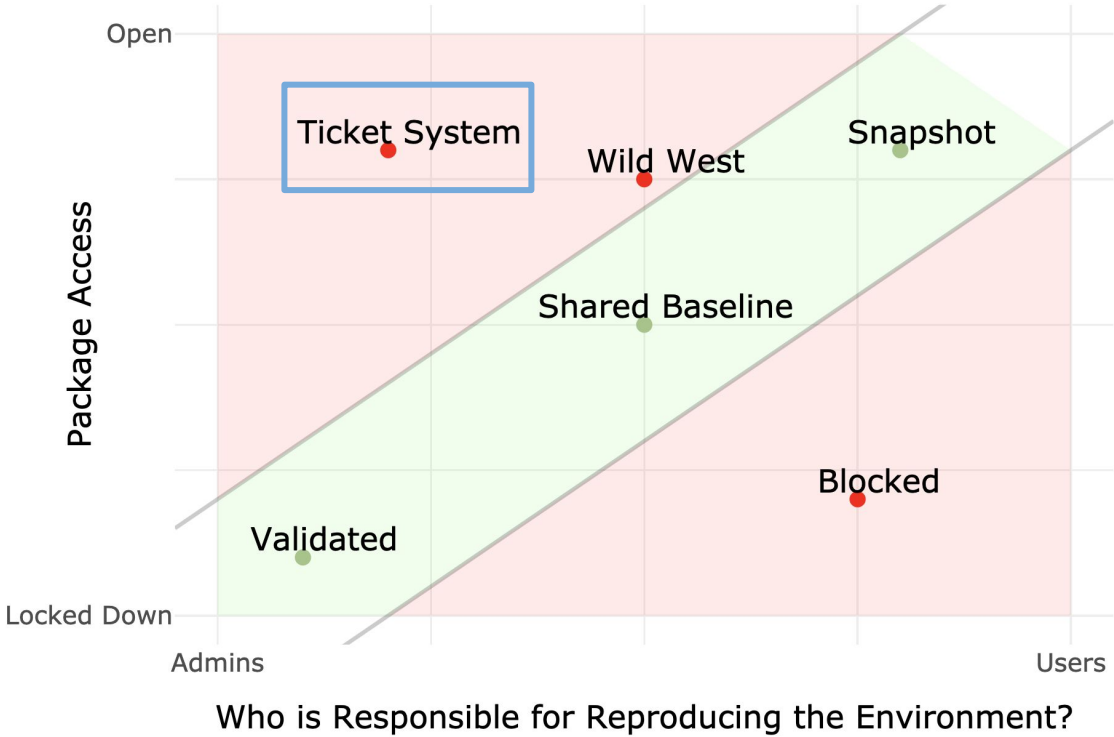


RStudio

Strategy Map

Ticket System

- Admins are mechanically responsible
- Still open access - just slow
- Upgrades often break things

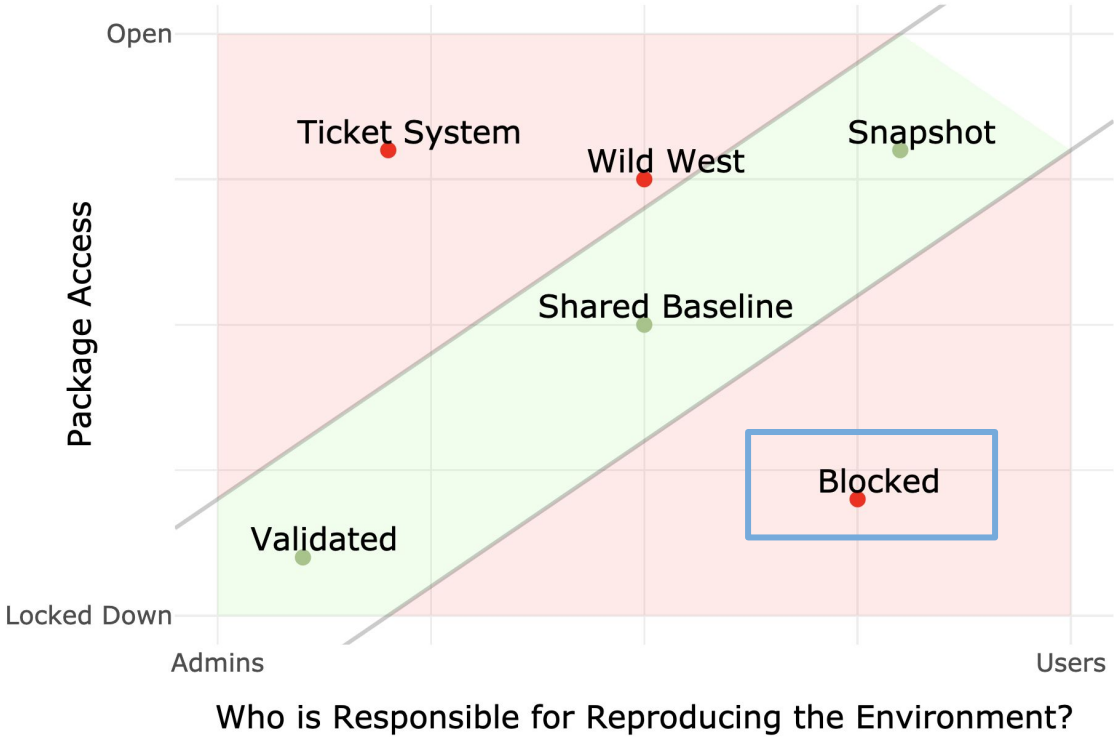


RStudio

Strategy Map

Blocked

- Environment is locked down
- No affordance for packages
- Backdoor behavior

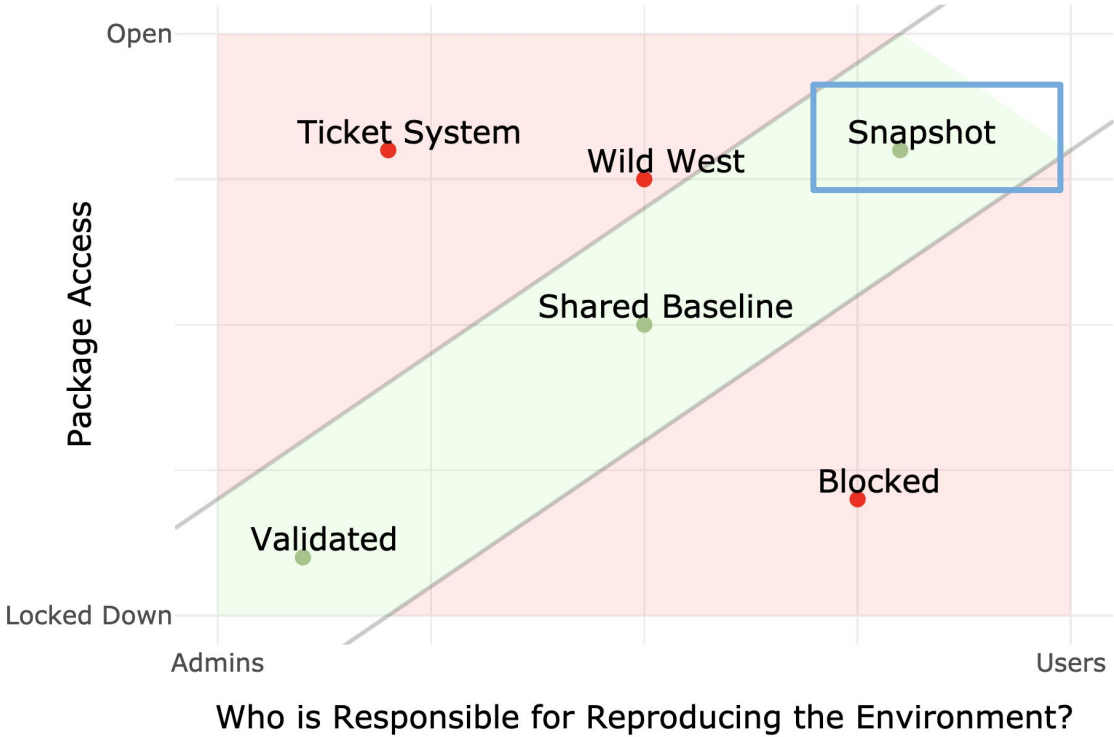


RStudio

Strategy Map

Snapshot

- Users record what they are doing

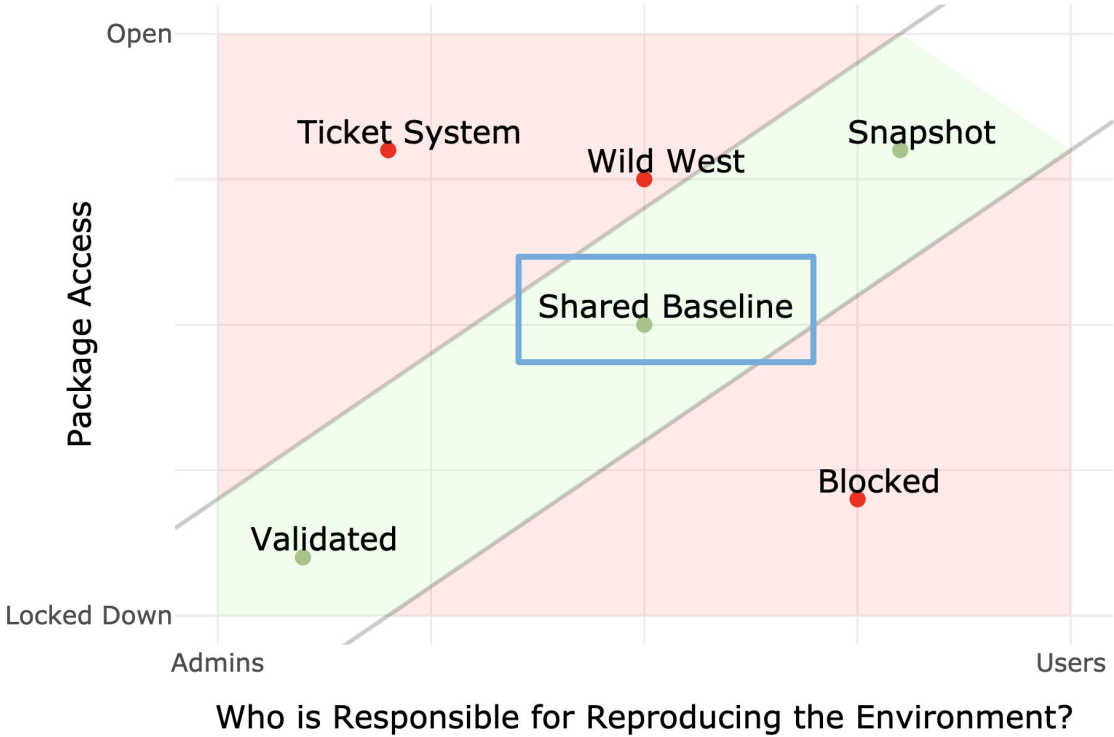


RStudio

Strategy Map

Shared Baseline

- Admins create stable baseline environments
- Immediate package access is traded for consistency and stability

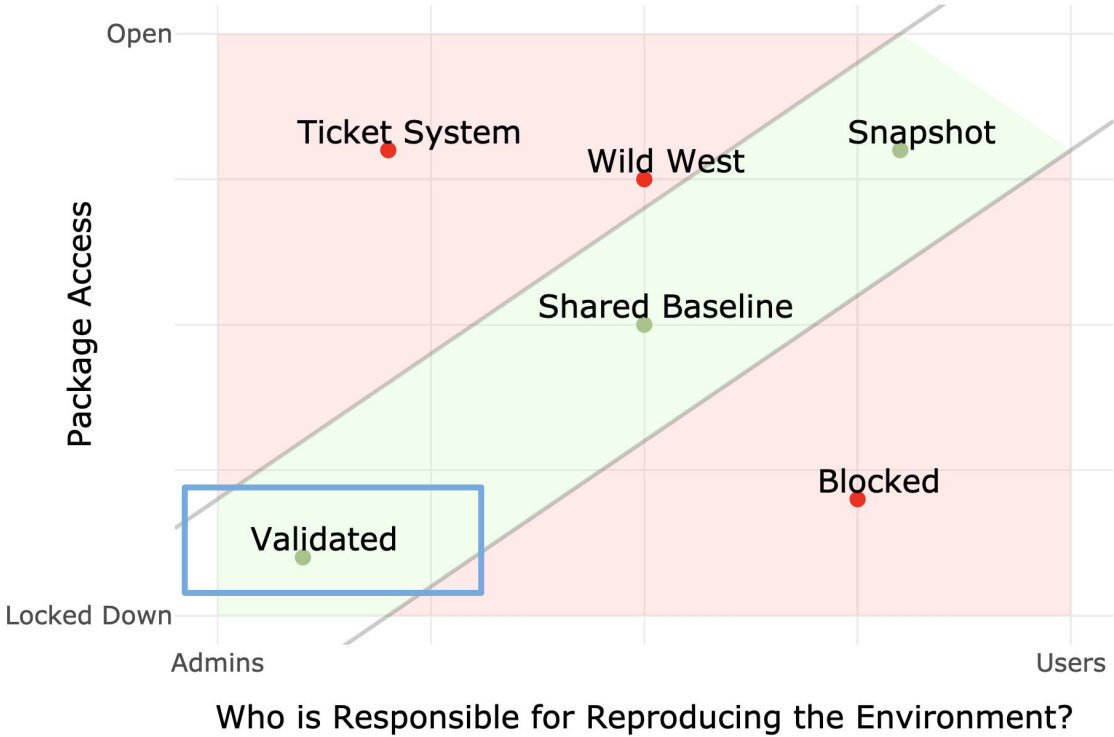


RStudio

Strategy Map

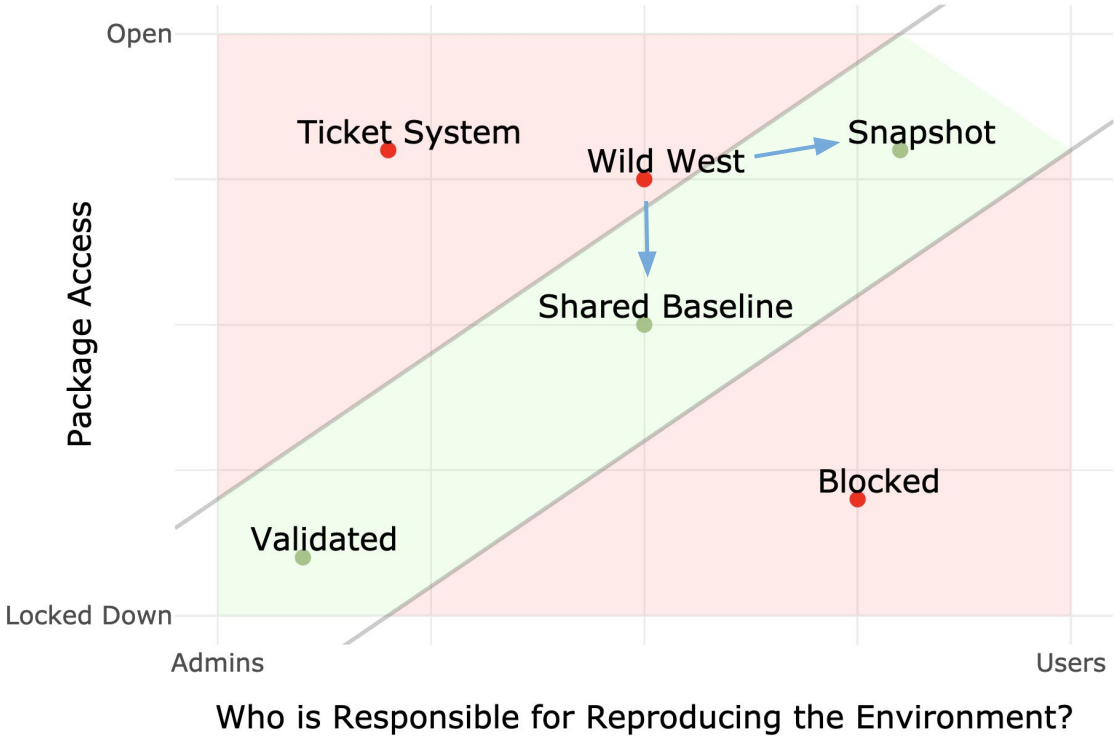
Validated

- Admins control **and test** the environment with approved package subsets



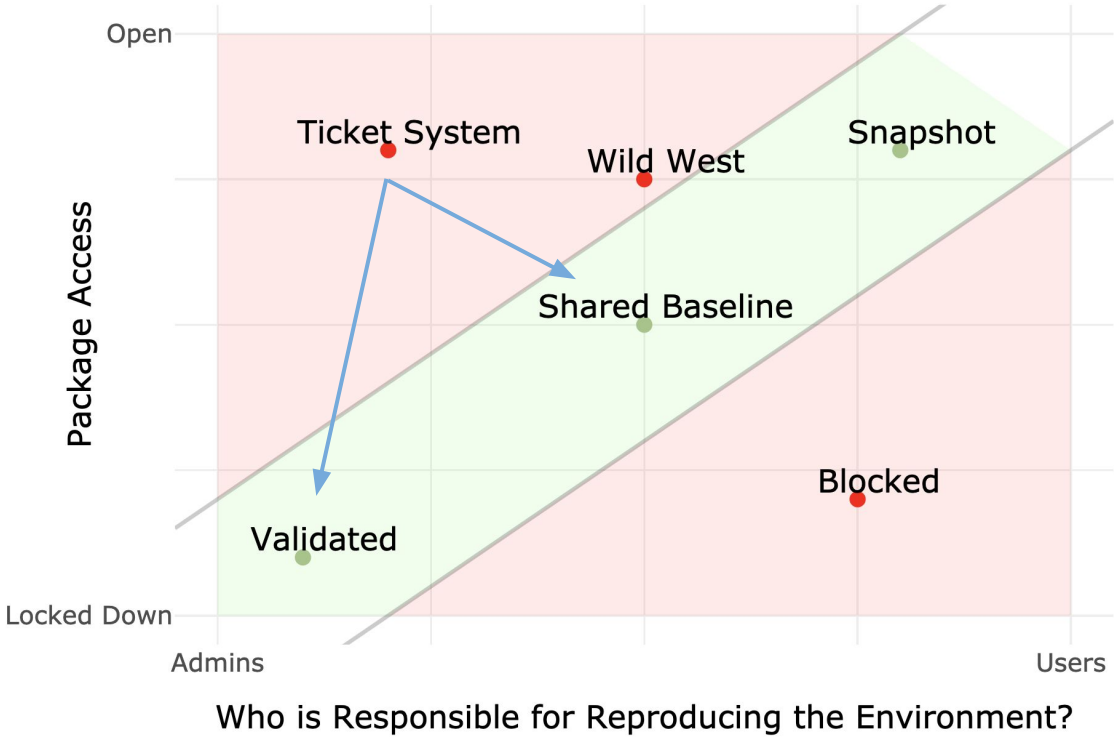
RStudio

Strategy Map



RStudio

Strategy Map

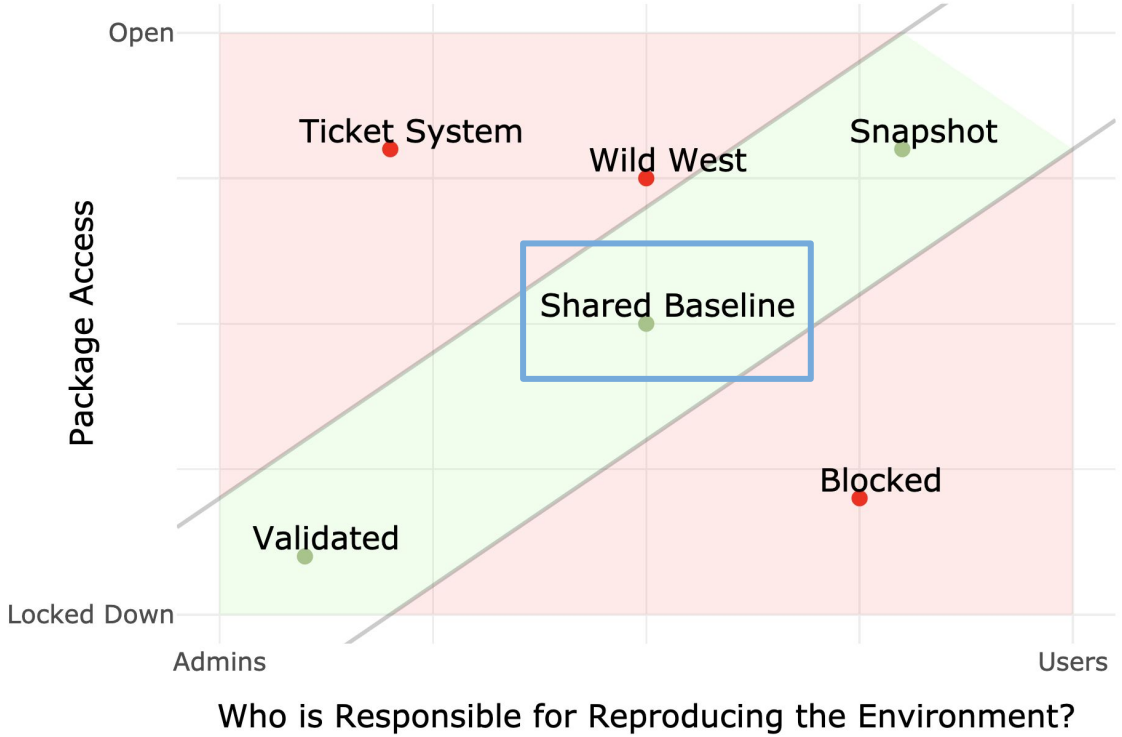


RStudio

Collaborating on a team with a **Shared Baseline** using a **Frozen Repository**

Goal: *Share code with others and they can easily run it*

Implied Goal: Everyone has quick access to the same sets of installed packages (library)



RStudio

Collaborating with a team

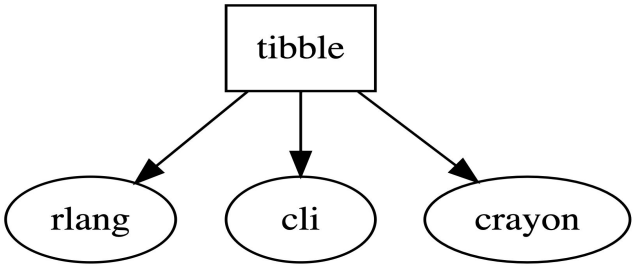
Goal: *Share code with others and they can easily run it*

Implied Goal: Everyone has quick access to the same sets of installed packages (library)

Naive Approach:

1. Setup a shared development environment (e.g. RStudio Server)

2. Admins get requests for packages and install them into a system library

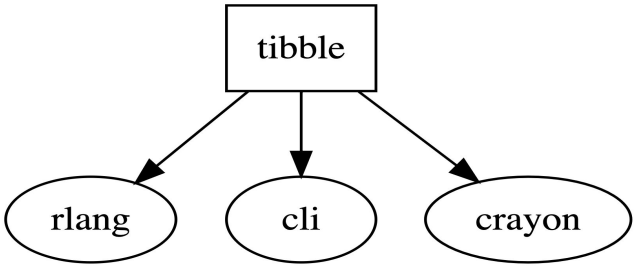


RStudio®

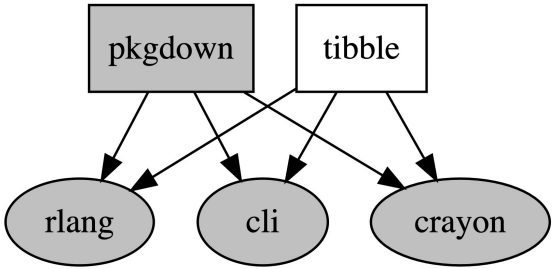
Collaborating with a team

Naive Approach:

1. Setup a shared development environment (e.g. RStudio Server)
2. ~~Admins get requests for packages and install them into a system library~~



January 1st



March 1st

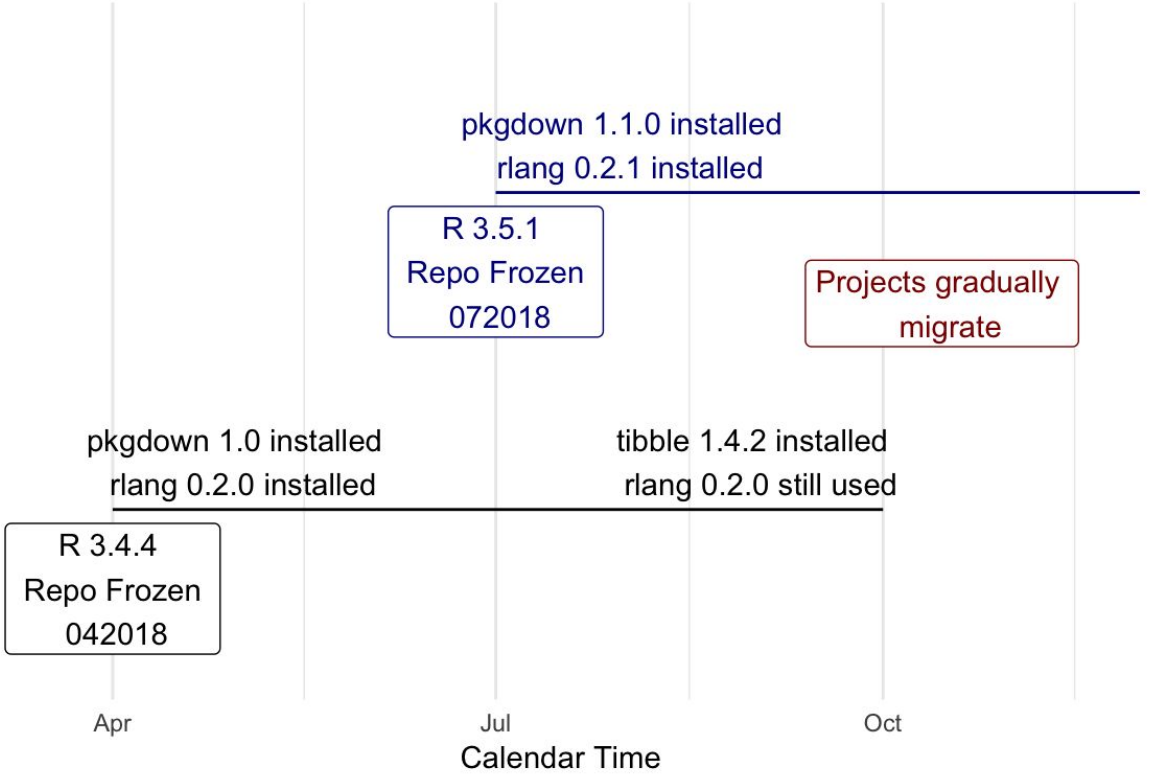
RStudio

Collaborating with a team

Better Approach:

1. Setup a shared development environment (e.g. RStudio Server)

2. Admins point each R version at a frozen repository



Demo

Use Case: Collaborate on a team

Strategy: Shared Baseline

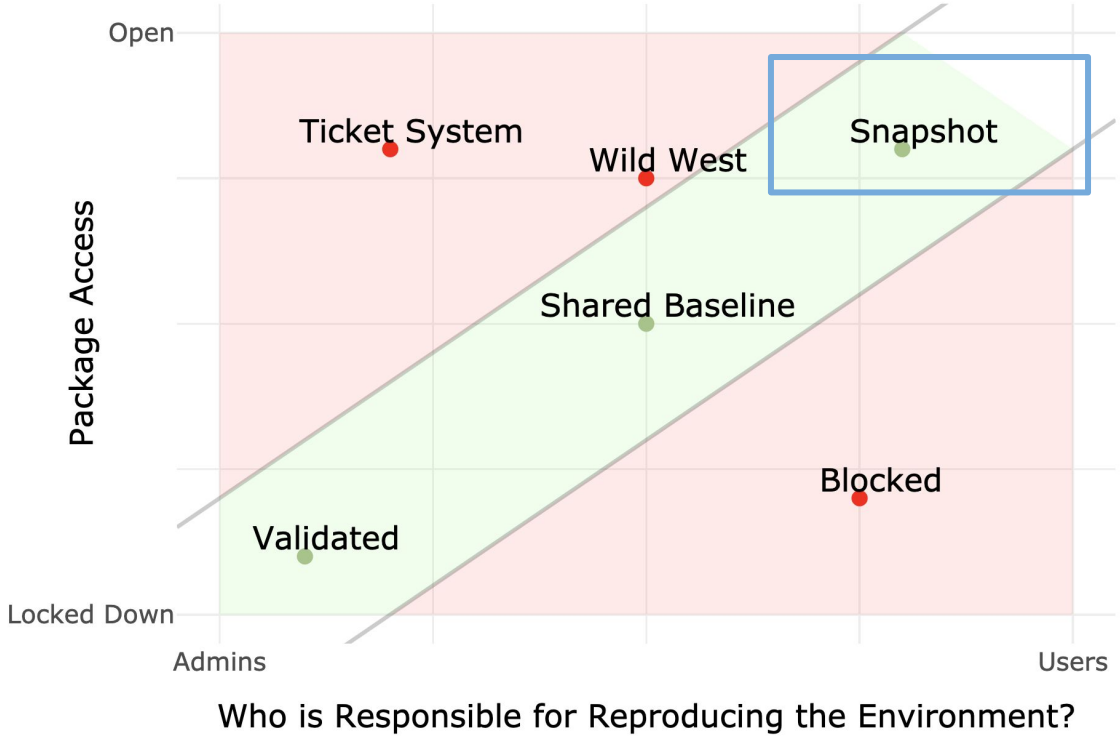
Tools: Frozen Repository (RSPM), Multiple R Versions, Rprofile.site, Site Library

Safely upgrading packages with **Snapshot and Restore** using **renv**

Goal:
Upgrade or add new packages to access exciting new things.

Implied Goals:

- Don't break other things (isolate)
- Safely roll back changes.

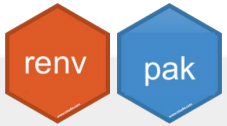


RStudio

Safely upgrading packages

Goal:
Upgrade or add new packages to access exciting new things.

Implied Goals:
- Don't break other things (isolate)
- Safely roll back changes.

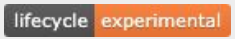



```
# create an isolated per-project library
renv::init()

# snapshot state (and commit lock file)
renv::snapshot()

# optionally upgrade packages
pak::pak_install("ggplot2")

# fall back to older versions
renv::history()
renv::revert("commit123abc")
renv::restore()
```



RStudio

Demo

Use Case: Upgrade a package

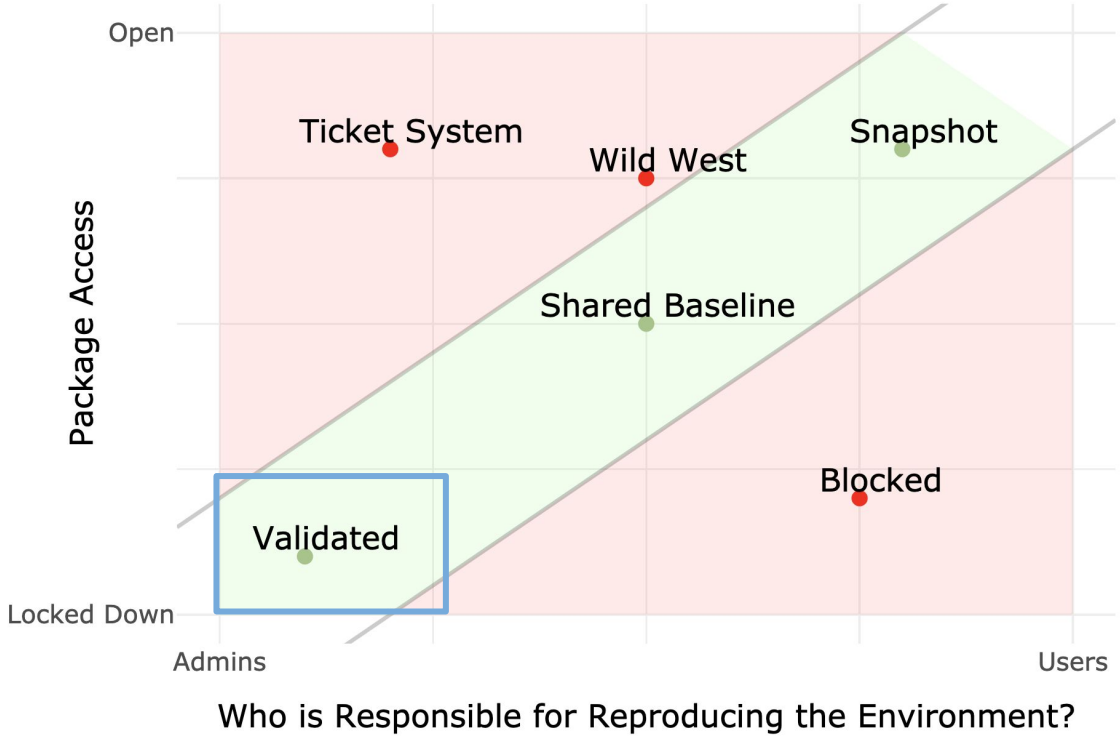
Strategy: Snapshot and Restore

Tools: renv, pak, Internal Repository (RSPM)

Using approved packages
with **Validation** using **Docker**

Goal: *Use reproducible, approved packages*

Implied Goal: Access and recreate consistent **subsets** of installed packages

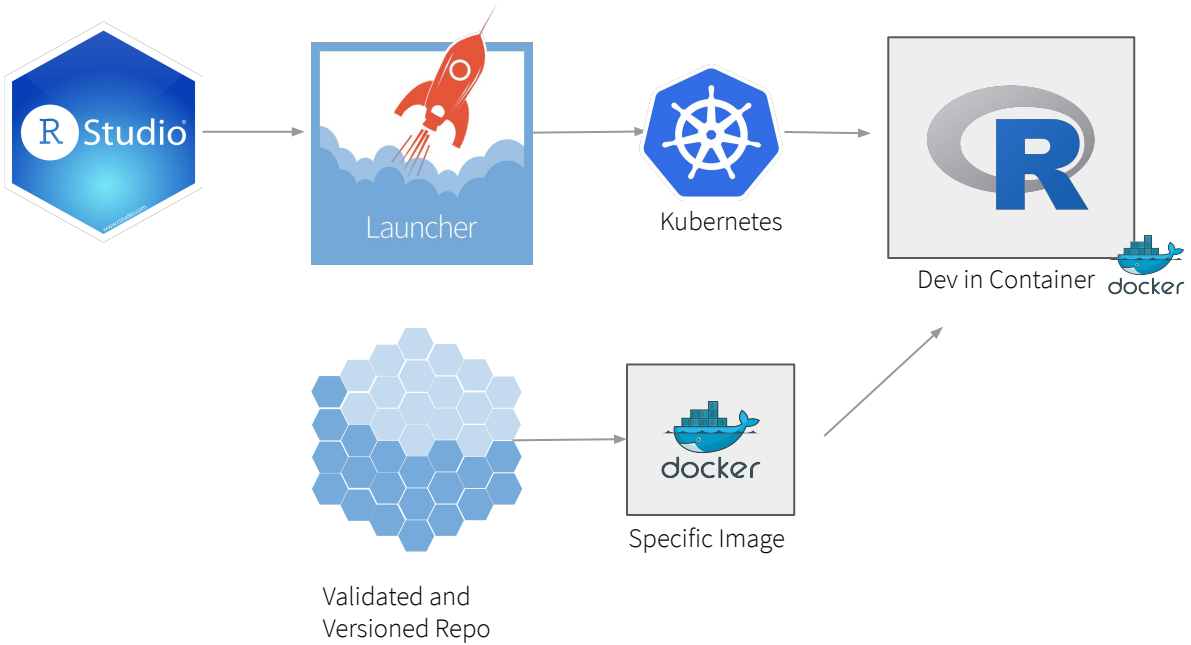


RStudio

Validated environment

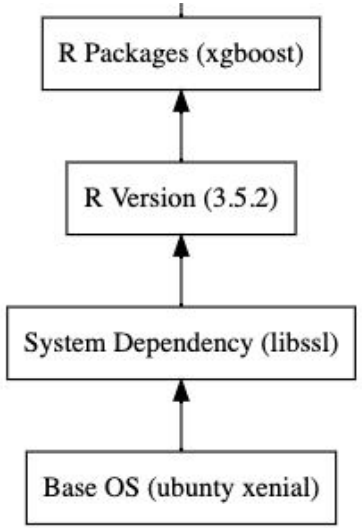
Goal: *Use reproducible, approved packages*

Implied Goal: Access and recreate consistent **subsets** of installed packages



RStudio

Validated environment - Docker Image



Data Science “Layers”

```
FROM ubuntu:bionic

# steps to build a specific version of R
RUN wget ${R-VERSION} \
  .... \
  /opt/R/${R-VERSION}

RUN /opt/R/${R-VERSION}/bin/R -e 'capabilities()'

# steps to install system deps for packages (from RSPM)
RUN  apt-get install -y \
  libssl-dev

# point the container at the validated repo
RUN cat 'options(repos = c(CRAN = "validated-url"))' >   \
      /opt/R/${R-VERSION}/lib/etc/Rprofile.site

# or actually install the validated packages
RUN R -e 'install.packages(c("xgboost"), repos = "validated-url")'
```

Sample Dockerfile

RStudio

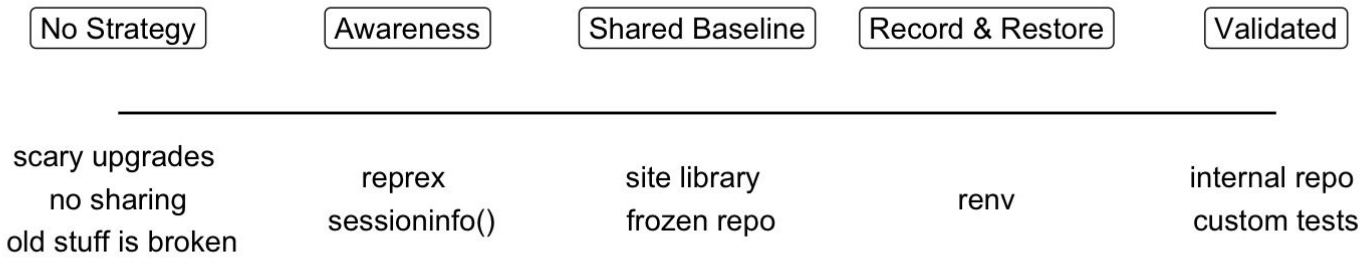
Demo

Use Case: Accessing approved packages

Strategy: Validation

Tools: Docker and a Validated Repository (RSPM)

Recap



Reproducibility isn’t binary. Pick a strategy based on your use cases and implement it with appropriate tools

RStudio®

Questions?

https://environments.rstudio.com