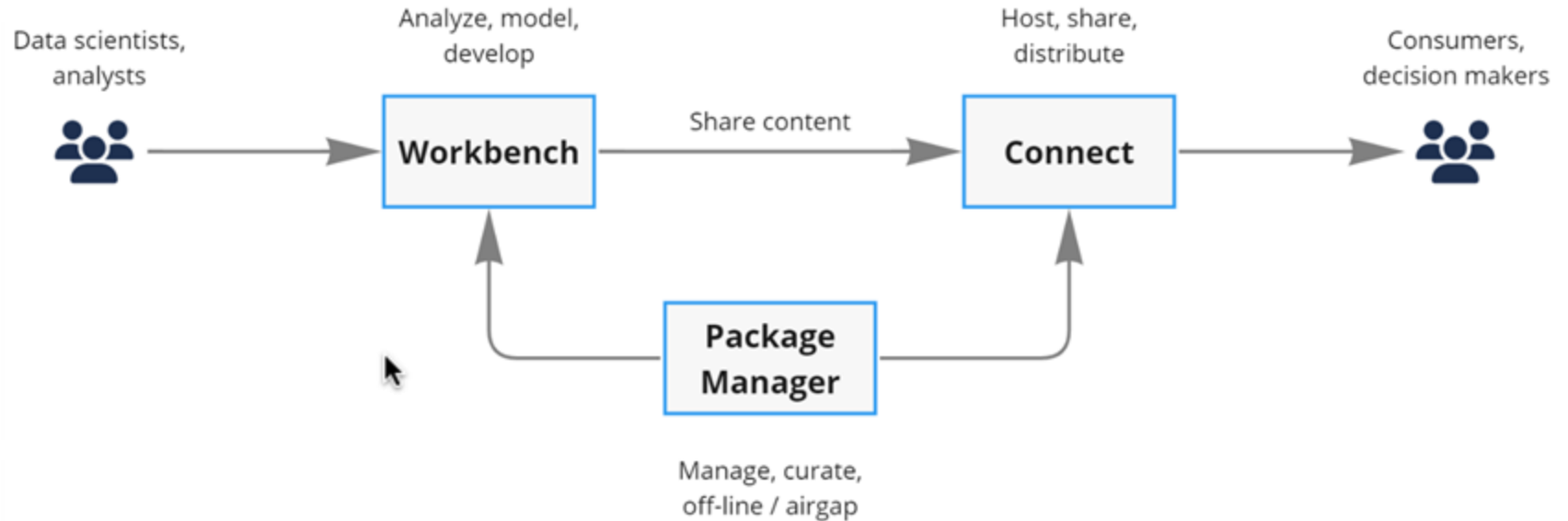


Posit Team: Metadata Driven Risk Assessment & Surfacing Validated R Packages

Michael Mayer & Aaron Jacobs, Posit PBC

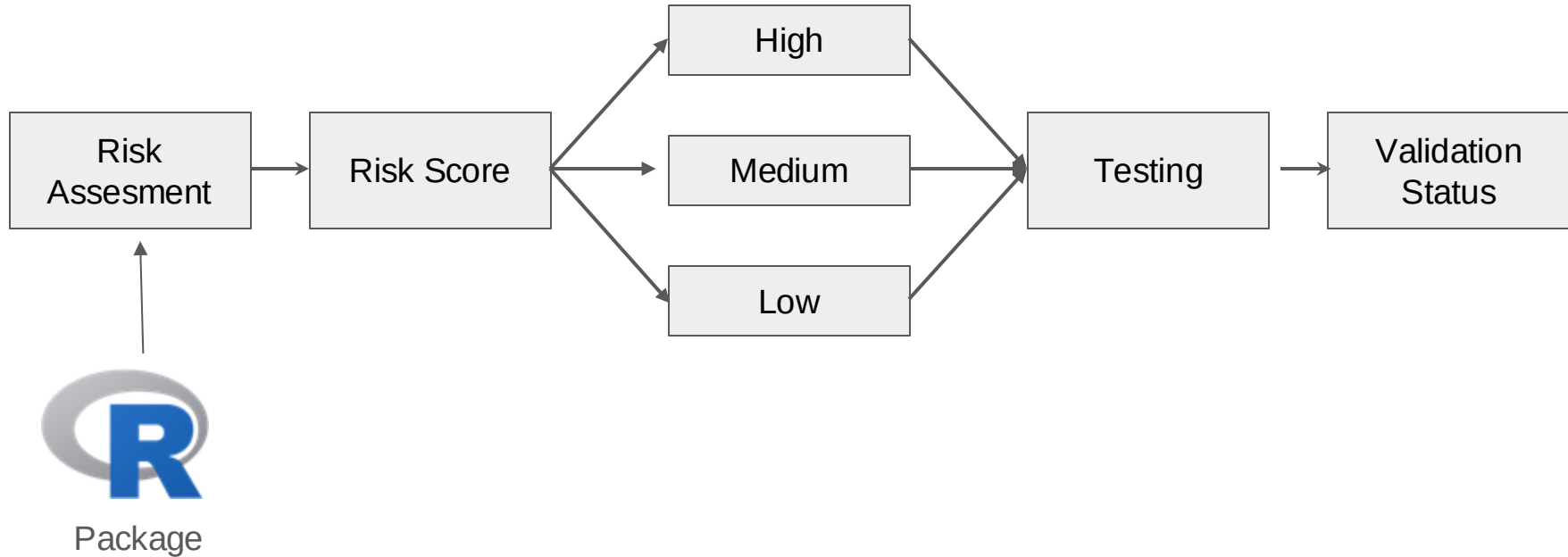
What is Posit Team ?



Package Testing/Validation on a GxP system

- Validation Strategy
- Selection of packages
- Validation Plan
- Execution
- Validation Report

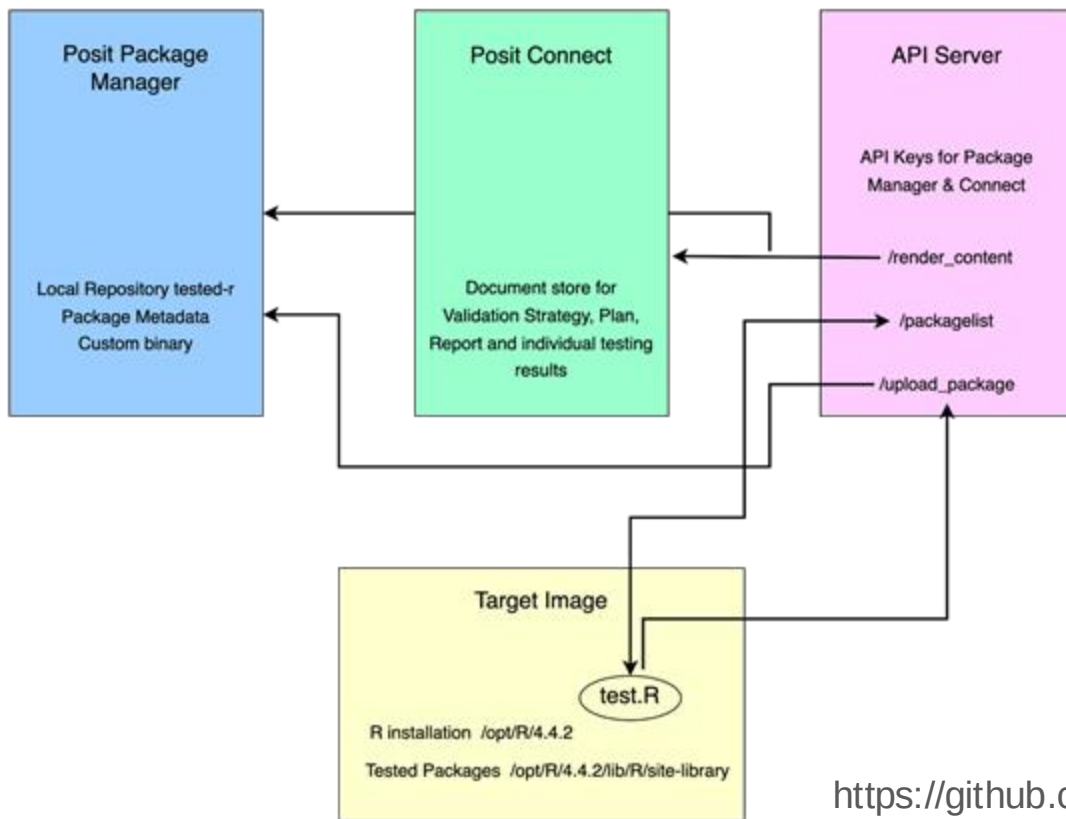
Risk-based validation



Deployment of a new version of R on a GxP/CSV system

- Scope of Work determination
- Impact Analysis and Quality strategy
- Implementation/Validation Plan
- Execution of IQ/OQ/PQ
- Implementation Report
- Closure of activities

Toy example



<https://github.com/sol-eng/package-testing>

Process outline

- Specify R version and list of R packages to be installed and validated
- Install desired R version into desired server/image
- Specify list of R packages and use script (test.R) to
 - Install package
 - Calculate risk metric
 - Test package based on the risk category
 - Upload binary package to PPM as “internal” binary into custom repository for reproducibility reasons
 - Write Quarto document for each package highlighting IQ/OQ/PQ results and upload to Posit connect
 - Add custom metadata to Package manager (Risk score and category, Test report, Validation Strategy/Plan/Report,

Package Manager Repository structure



Example output for custom metadata

posit Package Manager

Repository: tested-r

Packages in the tested-r repository

SETUP

R packages in tested-r

19 of 19

Tested and assessed R packages

cli	3.6.3
Helpers for Developing Command Line Interfaces	
clustermq	0.9.5
Evaluate Function Calls on HPC Schedulers (LSF, SGE, SLURM, PBS/Torque)	
crayon	1.5.3
Colored Terminal Output	
globals	0.16.3
Identify Global Objects in R Expressions	
glue	1.8.0
Interpreted String Literals	
hms	1.1.3
Pretty Time of Day	
lifecycle	1.0.4
Manage the Life Cycle of your Package Functions	
magrittr	2.0.3
A Forward-Pipe Operator for R	
narray	0.5.1
Subset- And Name-Aware Array Utility Functions	
pkgconfig	2.0.3
Private Configuration for R Packages	
prettyunits	1.2.0
Pretty, Human Readable Formatting of Quantities	

clustermq

Evaluate Function Calls on HPC Schedulers (LSF, SGE, SLURM, PBS/Torque)

Overview

Readme

Evaluate arbitrary function calls using workers on HPC schedulers in single line of code. All processing is done on the network without accessing the file system. Remote schedulers are supported via SSH.

INSTALL CODE

Copy

```
install.packages('clustermq')
```

VERSION
0.9.5

METADATA

6 METADATA

risk_category	High
risk_score	0.98833479417112
test_url	http://localhost:3939/content/1c5ccf43-177c-4b6d-959c-ee12b265584a/
validation_plan	http://localhost:3939/content/17232791-efd1-4914-baed-fc32b3538dff/
validation_report	http://localhost:3939/content/ac4d9fcc-ch5f-4dec-8fc1-35ab9c6caead/
validation_strategy	http://localhost:3939/content/0d8b5e42-703f-4db7-8a41-df9eb2154af2/

ABOUT

<https://mschubert.github.io/clustermq/>

7 IMPORTS

1 DEPENDS

11 SUGGESTS

1 LINKING TO

PUBLISHED

Aug 19, 2024 9:10 AM UTC

Testing Report for clustermq 0.9.5

AUTHOR

Michael Mayer

Package description

field	value
Package	clustermq
Title	Evaluate Function Calls on HPC Schedulers (LSF, SGE, SLURM, PBS/Torque)
Version	0.9.5
Authors@R	<code>c(person('Michael', 'Schubert', email='mschu.dev@gmail.com', role = c('aut', 'cre', 'cph'), comment = c(ORCID='0000-0002-6862-5221')), person('ZeroMQ authors', role = c('aut', 'cph'), comment = "source files in 'src/libzmq' and 'src/cppzmq'"))</code>
Maintainer	Michael Schubert <mschu.dev@gmail.com>
Description	Evaluate arbitrary function calls using workers on HPC schedulers in single line of code. All processing is done on the network without accessing the file system. Remote schedulers are supported via SSH.

Conclusion and Outlook

- Metadata feature in Package Manager can be very helpful to highlight risk scores and category to end users
- Process described can be further expanded
- Decisions on how to calculate risk score and what metrics to choose and the resulting testing strategy is out of scope
- Further extensions to this metadata feature are in the works that include
 - Automatic ingestion of new package sources
 - Selection of packages based on metadata
 -