

Building an Open-Source-First Statistical Computing Environment

A Cloud-Native Approach with R, Python, and Kubernetes

Rafael Pereira, Appsilon | **Amanda Lopuski**, Solid Biosciences

PhUSE US Connect 2026, AS13



The Changing Landscape

Legacy SCE Challenges

- Built around SAS, limited flexibility
- Monolithic, hard to update
- Manual validation processes
- Disconnected from modern tooling

Modern Expectations

- ✓ R and Python as primary languages
- ✓ Cloud-native, scalable infrastructure
- ✓ Git-based workflows and IaC
- ✓ AI/ML readiness

R drove open-source adoption. Python brings data engineering standards. Together, they create the foundation for AI. Companies adopting modern practices are adopting AI significantly faster.

The Open-Source-First Opportunity

Solid Biosciences chose to build their clinical development infrastructure with R and Python from day one



Design Flexibility

Build around modern workflows without legacy constraints



AI Enablement

Faster open-source adoption = faster AI adoption



Rich Tool Ecosystem

Pharmaverse, modern IDEs, cloud-native services



Security & Control

Better visibility, configurable security options

Organizations that modernize today will be positioned to leverage AI for clinical programming, automated QC, and advanced analytics — capabilities rapidly moving from experimental to essential.

Containers: The Foundation for Reproducibility



"Write once, run anywhere"

Containers encapsulate the exact computational environment needed to reproduce any analysis



Platform Engineering

- Simplified deployment & scaling
- Isolated workloads
- Defined resource limits
- Enhanced security boundaries



Biometrics

- Dev environment = Prod environment
- Reproducible analyses
- No environment drift
- Consistent package versions

Kubernetes: The Modern Operating System

Container orchestration for enterprise-grade clinical computing



Auto-scaling

Scale up during submission crunch, scale down for cost savings

Self-healing

Failed workloads restart automatically

Declarative Config

Version-controlled infrastructure manifests

AI/ML Ready

Standard platform for modern workloads

Creates a natural bridge to modern data engineering tools — Databricks, Snowflake — within an open ecosystem

Git + IaC + CI/CD

The Automation Trifecta for Compliance



Git

- Immutable change record
- Attributed & timestamped
- Full audit history



Infrastructure as
Code

- Environment in version control
- Automated provisioning
- Consistent environments



CI/CD Pipelines

- Fast, clear deployments
- Auto-generated documentation
- Compliance as a byproduct

The Combined Effect

Speed: Changes that took weeks are deployed in hours with full traceability

Clarity: Every deployment generates its own audit trail and documentation

Compliance: Inspection-ready at all times — compliance emerges from the process, not as an afterthought

R Package Validation

Risk-Based Methodology Aligned with 21 CFR Part 11



R Validation Hub Methodology

Moving away from treating each R package as a one-off validation project



Automated Assessment

Packages assessed against predefined organizational rules



Dependency Analysis

Flags risks in package supply chains



Historical Tracking

Monitor changes affecting validation status



Role-Based Workflows

Structured review and approval process

Result: Reduced validation burden while maintaining rigor where it matters most

Implementation Outcomes

1

Environment

Unified workflows — explore in R,
submit in R

2

Zones

Validated + flexible, same tools,
different controls

∞

Modular

Update components without full
revalidation

Additional Benefits

- ✓ Automated compliance — audit trails generated through Git workflows
- ✓ Scalability — handles concurrent users, scales during peak submission periods
- ✓ Future-ready — AI/ML workloads supported on same infrastructure

Guidance for Modernization

1

Start with goals, not technology

Design around purpose and organizational needs

2

Invest in modularity

Flexibility today prevents disruption tomorrow

3

Support open source deliberately

Modern teams expect R, Python, and version control

4

Balance control and innovation

Isolate validated work without limiting experimentation

5

Partner where needed

External support accelerates success and reduces risk

Key Takeaways

- ✓ Open-source-first is viable for clinical analytics
- ✓ Containers + Kubernetes provide reproducibility at scale
- ✓ Git + IaC transform compliance from overhead to built-in
- ✓ Faster open-source adoption = faster AI adoption
- ✓ The SCE is the foundation for modern drug development

"A great SCE feels invisible when things work and indispensable when you need it. It accelerates your work instead of adding friction."

Thank You

Questions?

Rafael Pereira

Platform Unit Lead, Appsilon

rafael.pereira@appsilon.com

Amanda Lopuski Davis

Stat Programming Director - Biometrics, Solid Biosciences

alopuski@solidbio.com

