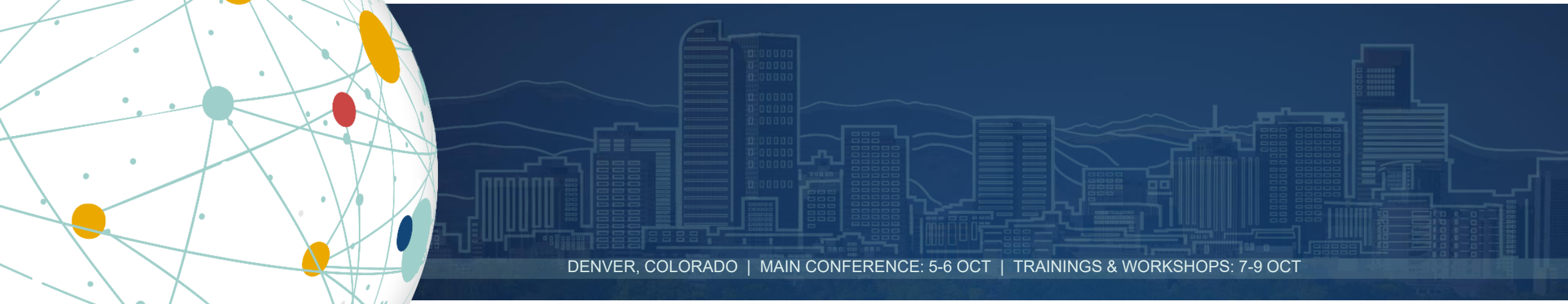
2026 US Interchange

THE FUTURE IS CONNECTED: STANDARDS AND AI POWERING DIGITAL TRANSFORMATION





CDISC Open Rules: Progress, Performance, and Foundations for the Future

Tomás Sabat Stöfsel, CEO & Co-Founder, Verisian

Agenda

1. Why CDISC Open Rules Matter
2. CDISC Open Rules v1.0 – Progress So Far
3. Engine Refactor
4. CDISC CORE vs Pinnacle 21
5. Verisian platform integration
6. Improved Rule Authoring Process

Speaker



Tomás Sabat Stöfsel

CEO & Co-Founder

Verisian

Why CDISC Open Rules Matter

… and why they matter to you …

CDISC Conformance Rules and Regulatory Requirements

TODAY

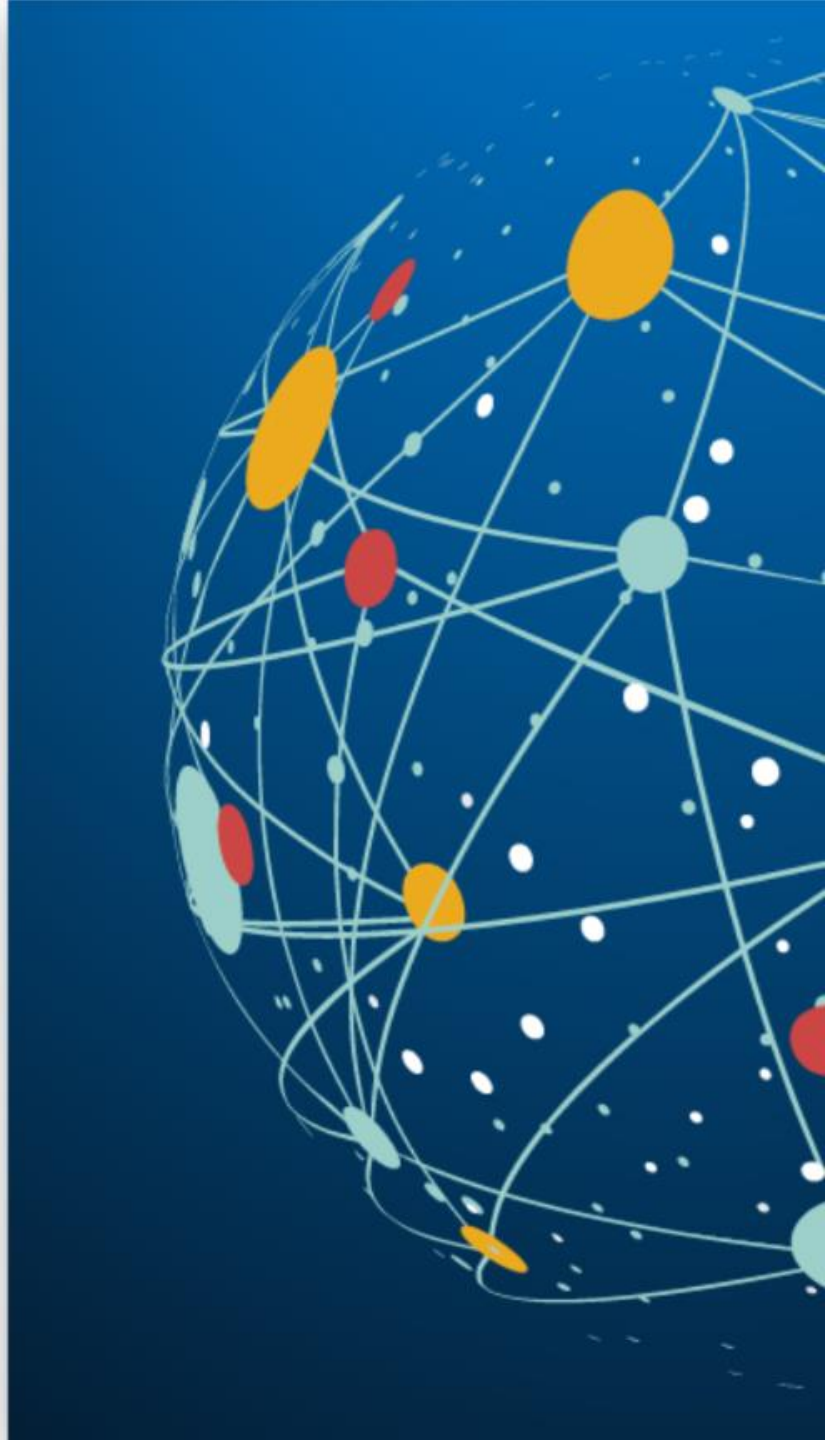
- Various source formats and structures
- Specifications that are not machine-executable
- Variety in interpretations of rule purpose
- Individual implementations to executable code

Lack of consistency in data for regulatory submissions

GOAL

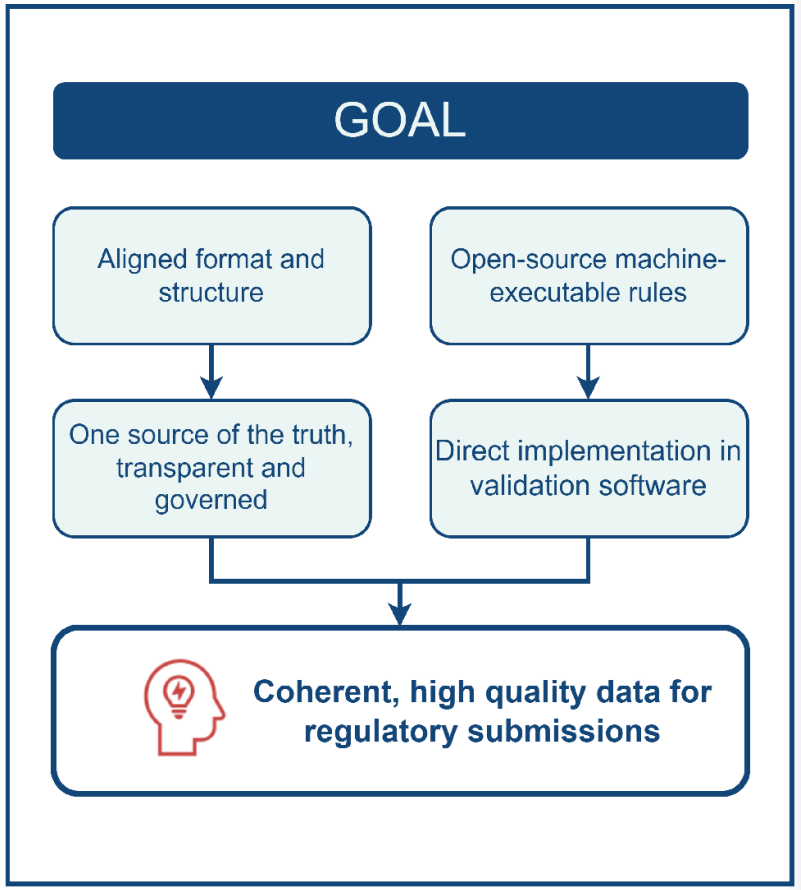
- Aligned format and structure
- Open-source machine-executable rules
- One source of the truth, transparent and governed
- Direct implementation in validation software

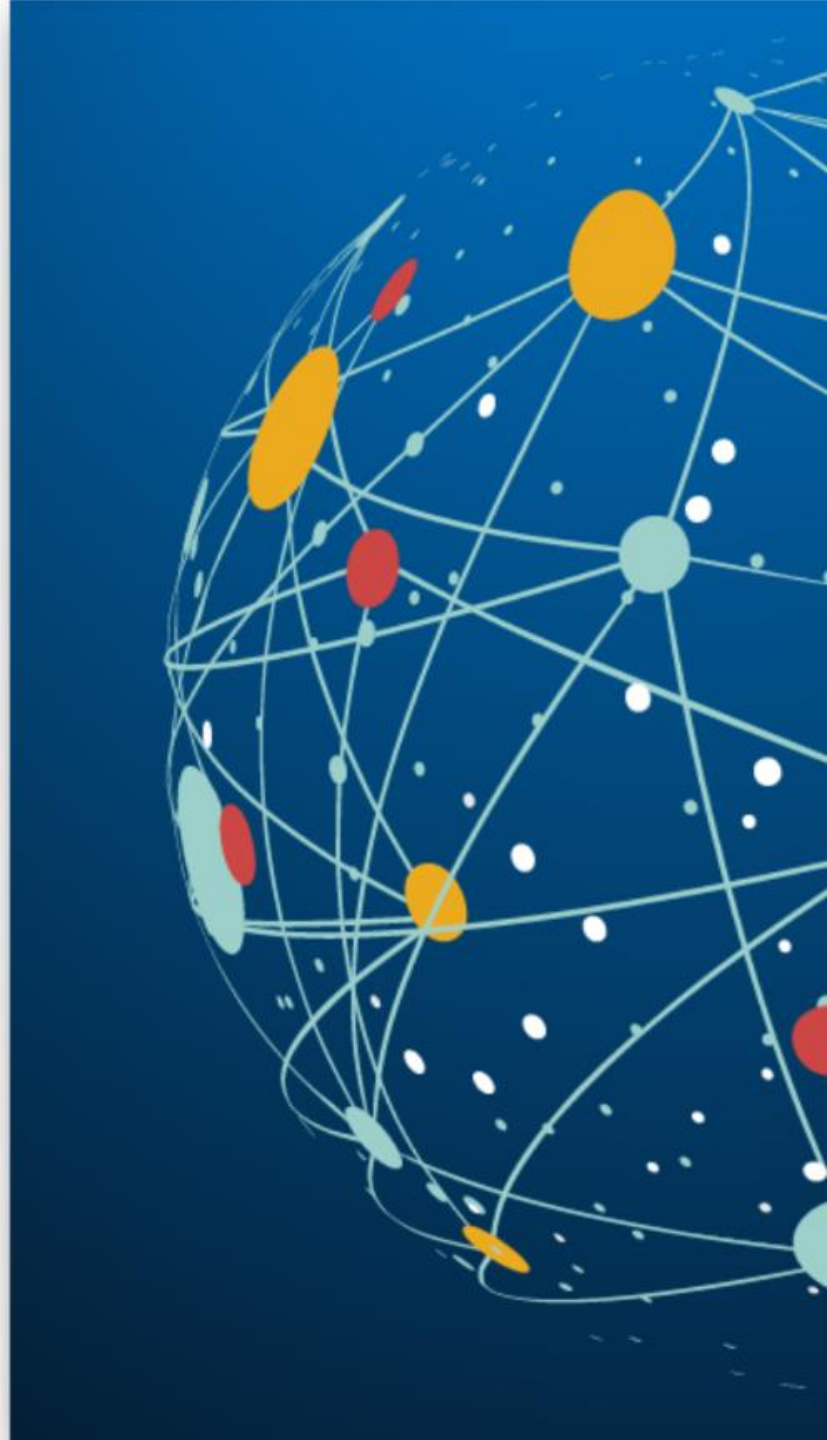
Coherent, high quality data for regulatory submissions



CDISC Conformance Rules and Regulatory Requirements







Rule Authoring Status for v1.0

Standard	# Rules	% Written	% Verified
SDTMIG v3.2, 3.3, and 3.4	473	100%	97%
FDA Business Rules (SDTMIG)	186	100%	95%

Rule Authoring Status for ADaMIG

Standard	# Rules	% Written
ADaMIG v1.0, 1.1, 1.2 and 1.3	802	79%

- *Started authoring ADaM rules. These completed rules are unverified but CDISC verification will start shortly after SDTMIG completion and v1.0 release*

Rule Authoring Status for ADaMIG

Standard	# Rules	% Written
ADaMIG v1.0, 1.1, 1.2 and 1.3	802	79%

79% written in 2 weeks!

- *Started authoring ADaM rules. These completed rules are unverified but CDISC verification will start shortly after SDTMIG completion and v1.0 release*

Verisian’s Engine Refactoring

Verisian’s Engine Refactoring

- Complete internal refactor of the engine architecture to utilise SQL

Verisian's Engine Refactoring

- Complete internal refactor of the engine architecture to utilise SQL
- Minimal changes to existing CDISC rule logic, operations, or operators

Verisian's Engine Refactoring

- Complete internal refactor of the engine architecture to utilise SQL
- Minimal changes to existing CDISC rule logic, operations, or operators
- No additional setup or installation required*

Verisian's Engine Refactoring

- Complete internal refactor of the engine architecture to utilise SQL
- Minimal changes to existing CDISC rule logic, operations, or operators
- No additional setup or installation required*

*With pgserver the engine runs locally without any performance drop

Engine Refactoring – Performance Improvements

Old Engine	Refactored Engine

cdisc®

Engine Refactoring – Performance Improvements

Old Engine	Refactored Engine
Took 30+ minutes to upload data > 10GB	

Engine Refactoring – Performance Improvements

Old Engine	Refactored Engine
Took 30+ minutes to upload data > 10GB	Uploads GB data in under a minute

Engine Refactoring – Performance Improvements

Old Engine	Refactored Engine
Took 30+ minutes to upload data > 10GB	Uploads GB data in under a minute
Failed to run datasets > 1 GB	

Engine Refactoring – Performance Improvements

Old Engine	Refactored Engine
Took 30+ minutes to upload data > 10GB	Uploads GB data in under a minute
Failed to run datasets > 1 GB	Successfully runs datasets > 15GB

Engine Refactoring – Performance Improvements

Old Engine	Refactored Engine
Took 30+ minutes to upload data > 10GB	Uploads GB data in under a minute
Failed to run datasets > 1 GB	Successfully runs datasets > 15GB
Memory-based implementation	

Engine Refactoring – Performance Improvements

Old Engine	Refactored Engine
Took 30+ minutes to upload data > 10GB	Uploads GB data in under a minute
Failed to run datasets > 1 GB	Successfully runs datasets > 15GB
Memory-based implementation	SQL-based implementation with near unlimited scaling

cdisc®

Engine Refactoring – Performance Improvements

Old Engine	Refactored Engine
Took 30+ minutes to upload data > 10GB	Uploads GB data in under a minute
Failed to run datasets > 1 GB	Successfully runs datasets > 15GB
Memory-based implementation	SQL-based implementation with near unlimited scaling
No alternative options	

Engine Refactoring – Performance Improvements

Old Engine	Refactored Engine
Took 30+ minutes to upload data > 10GB	Uploads GB data in under a minute
Failed to run datasets > 1 GB	Successfully runs datasets > 15GB
Memory-based implementation	SQL-based implementation with near unlimited scaling
No alternative options	Embedded alternative with equal capabilities; no additional software installation required

Engine Refactoring – Performance Improvements

Old Engine	Refactored Engine
Took 30+ minutes to upload data > 10GB	Uploads GB data in under a minute
Failed to run datasets > 1 GB	Successfully runs datasets > 15GB
Memory-based implementation	SQL-based implementation with near unlimited scaling
No alternative options	Embedded alternative with equal capabilities; no additional software installation required

The refactored engine removes dataset size and scalability limitations while improving accuracy and error reporting

cdisc®

Further development to the Engine is on the way…

- Faster data ingestion
- Logic for execution of several large previously non-executable rules
- Up to 12x faster validation run-time

Comparison to Pinnacle 21

- P21 runs rules on outdated SDTMIG 3.4 on SDTM Model 1.7:
  - Inconsistent standard application
  - Incorrect flagging of issues

Comparison to Pinnacle 21

- P21 runs rules on outdated SDTMIG 3.4 on SDTM Model 1.7:
  - Inconsistent standard application
  - Incorrect flagging of issues
- CORE validates all latest rules on the latest SDTM Model 2.0:
  - Direct publisher to true validation mapping

Comparison to Pinnacle 21

- P21 ignores null-flavour mechanisms:
  - Disregards nuanced data
  - Singular rules with no context

Comparison to Pinnacle 21

- P21 ignores null-flavour mechanisms:
  - Disregards nuanced data
  - Singular rules with no context
- CORE does:
  - Wider rule context - multiple null rules working in tandem
  - Mutual exclusivity pairs between null-flavours and what they mean
  - Implements specific CDISC guidance

Comparison to Pinnacle 21

- How are we different to P21?

Comparison to Pinnacle 21

- How are we different to P21?
  - All CORE rules are CDISC-authenticated and approved

Comparison to Pinnacle 21

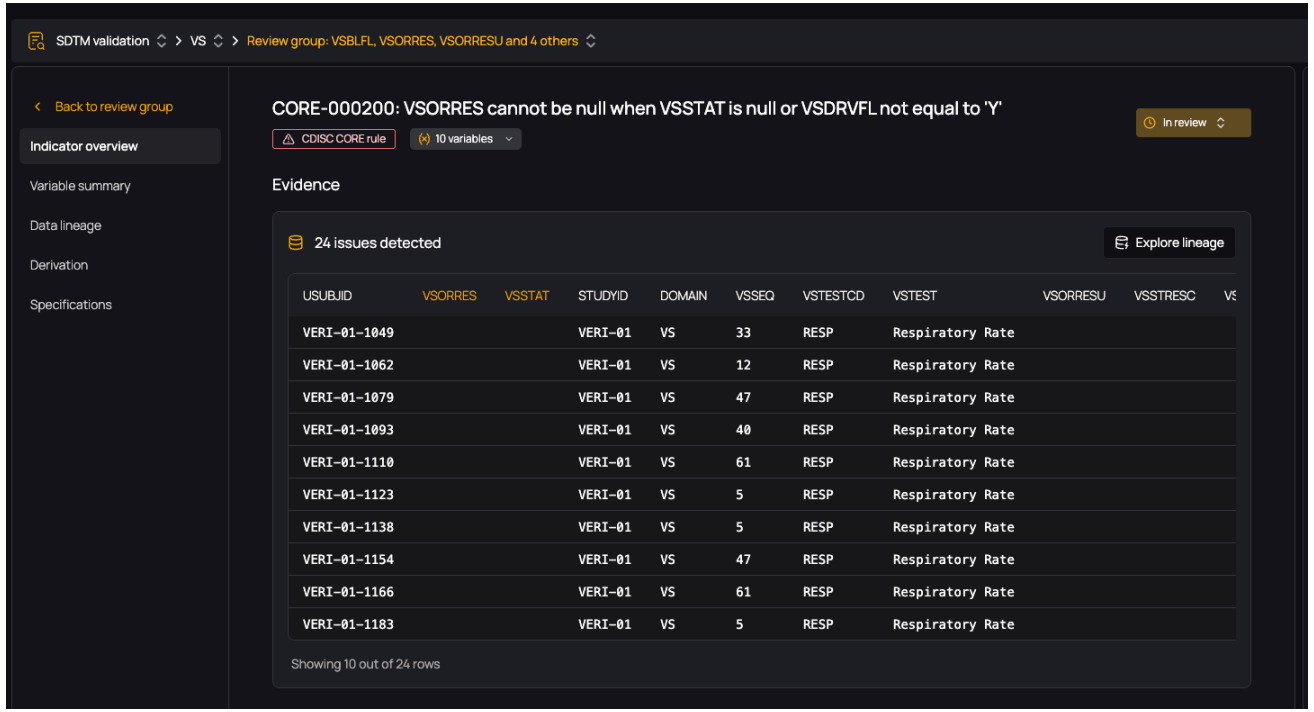
- How are we different to P21?
  - All CORE rules are CDISC-authenticated and approved
  - Many rules where P21 deviates from CDISC guidance

Comparison to Pinnacle 21

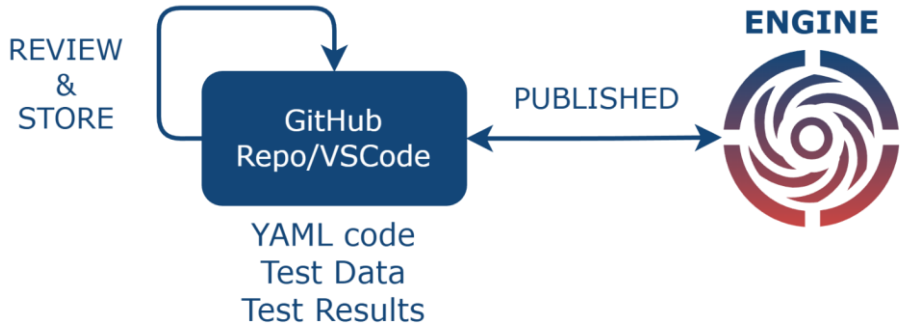
- How are we different to P21?
  - All CORE rules are CDISC-authenticated and approved
  - Many rules where P21 deviates from CDISC guidance
  - CORE integrates with wider Verisian system

Verisian Platform Integration

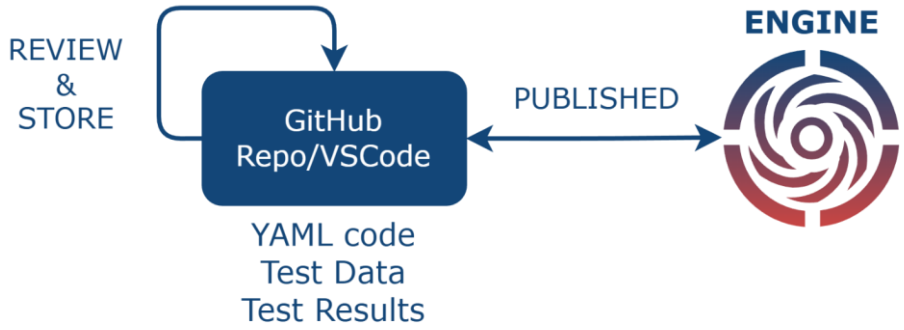
- Full integration with the Verisian suite:
  - See all your issues and only your issues
  - Browse issue by variable and dataset
  - Downloadable excel report
  - Links to rest of data validation suite tools



Improved Rule Authoring Process



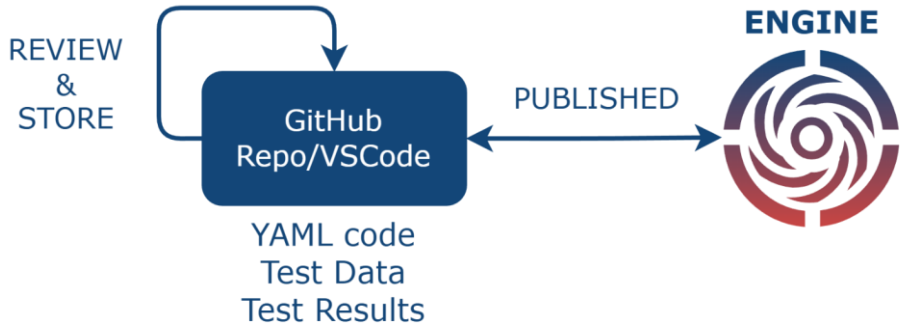
Improved Rule Authoring Process



Centralized and Trackable Workflow

- Git-based rule repository for storing, versioning, and tracking
- Streamlit dashboard for quick analysis and progress visibility

Improved Rule Authoring Process



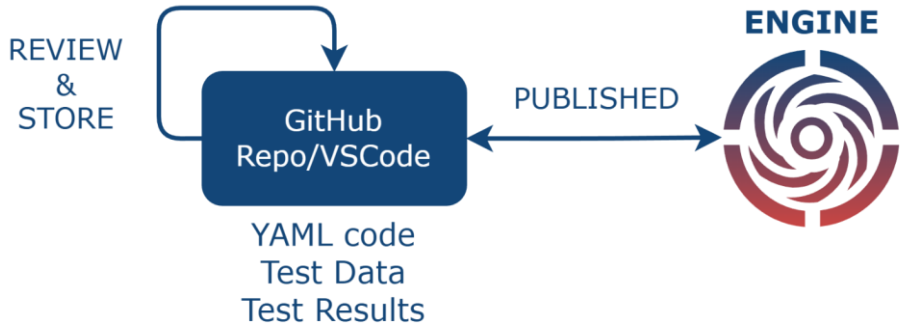
Centralized and Trackable Workflow

- Git-based rule repository for storing, versioning, and tracking
- Streamlit dashboard for quick analysis and progress visibility

Easy and Guided Contribution

- Simple setup scripts and clear contributor guide
- Accessible to all skill levels
- Structured review process

Improved Rule Authoring Process



Centralized and Trackable Workflow

- Git-based rule repository for storing, versioning, and tracking
- Streamlit dashboard for quick analysis and progress visibility

Easy and Guided Contribution

- Simple setup scripts and clear contributor guide
- Accessible to all skill levels
- Structured review process

Integrated Training and Testing

- Built-in training materials
- Automated testing within the repository

cdisc®

Further Rule Authoring Progress

- Faster ADaM rule authoring:
  - ~80% of rules written over the course of 2 weeks
  - To be completed after v1.0

Further Rule Authoring Progress

- Faster ADaM rule authoring:
  - ~80% of rules written over the course of 2 weeks
  - To be completed after v1.0

- Bundling rules based on logic:
  - Much faster authoring time
  - One comprehensive set of test data per bundle
  - Review and maintenance significantly easier

Further Rule Authoring Progress

- Faster ADaM rule authoring:
  - ~80% of rules written over the course of 2 weeks
  - To be completed after v1.0

- Bundling rules based on logic:
  - Much faster authoring time
  - One comprehensive set of test data per bundle
  - Review and maintenance significantly easier

- Rule-creation scripts:
  - Allows contributors to focus solely on writing rule logic
  - Eliminates human error

Thank You!



Questions?

tomas@verisian.com

