Rules: June 2022 Edition

National Institute of Standards and Technology
U.S. Department of Commerce

ITL/CSD/OSCAL Team

Model Meeting 20220610

What's changed?

- We engaged the community around framing and design
  - Explanations of key terms, concepts, and context
  - Itemize personas and how they will interact with rules
  - Itemize use cases for how those personas will use rules

What's changed?

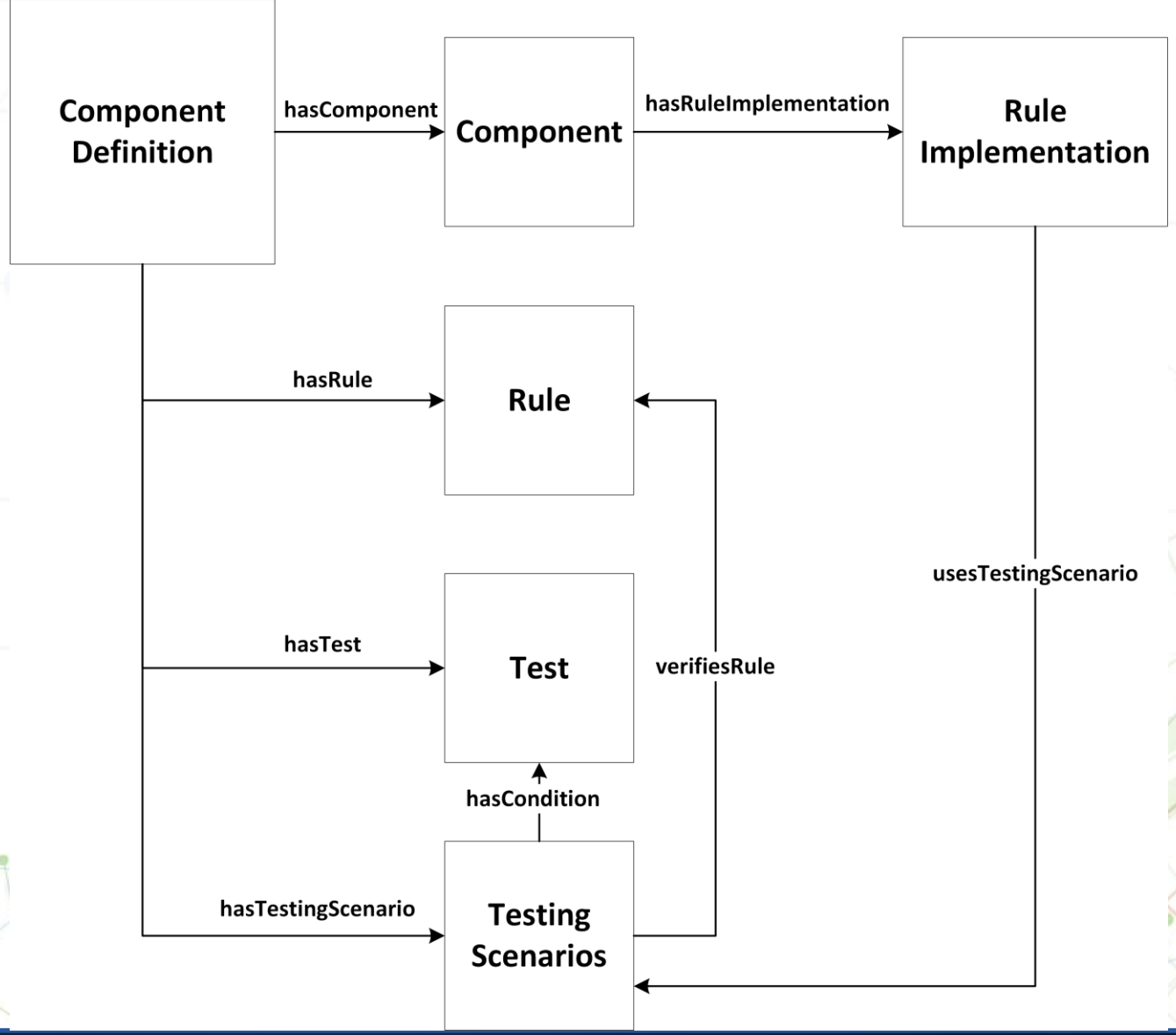
- Rolled community feedback into model updates:
  - Rules, test, and testing-scenarios at the top level
    - Pre-condition and conditions for tests in testing-scenarios
    - Conditions allow logical chaining: and, or, not
  - Independent rule, test, and testing-scenario definitions
    - One definition, reference elsewhere, less copy-paste
    - Binding in rule-implementation inside a component in a component-definition

What's new? What do we review now?

- **Now: Focus on the relationship between the key objects.**
- Later: focus on the internal structure of the key objects.

What are the key elements?

- **rule**: a desired trait of an information system
- **test**: method for checking an aspect of a trait
- **testing-scenario**: a set of **condition**s consisting of one or more **test**s to determine if a trait is present
- **rule-implementation**: in a component definition, bind a testing-scenario to a specific component and implemented requirement


Component Definition
hasComponent
Component
hasRuleImplementation
Rule Implementation
hasRule
Rule
hasTest
Test
verifiesRule
usesTestingScenario
hasCondition
hasTestingScenario
Testing Scenarios

“Our organization needs data to be encrypted at rest.”

- Rule 1: Red Hat servers in the data center must encrypt server filesystem in approved configuration.
- Test 1: During development, staff use Infrastructure-as-Code (IaC) tools to build Red Hat Linux images, checking LUKS (encryption management) are configured in Ansible and Packer code.
- Test 2: During operations, staff create a server with this RedHat Linux image and run OpenSCAP baseline to check LUKS settings in a Continuous Integration (CI) pipeline.
- Testing Scenario 1: Test 1 and Test 2 must be checked.
  - Operator: and
  - Condition
    - Pre-condition 1: A specified location for the IaC tools contains valid files.
    - Pre-condition 2: Static analysis tool is installed and configured on the developer workstation
    - Test 1
  - Condition
    - Pre-condition 1: a valid CI environment is defined.
    - Pre-condition 2: a valid CI pipeline file is defined.
    - Test 2
- Component: this component describes servers that are part of the information system.
  - Rule-implementation: Testing Scenario 1 implements requirements for NIST SP 800-53 Rev 5 control SC-28.

“Our organization needs data to be encrypted at rest.”

- Rule 1: Red Hat servers in the data center must encrypt server filesystem in approved configuration.
- Test 1: During development, staff use Infrastructure-as-Code (IaC) tools to build Red Hat Linux images, checking AWS EBS volumes and KMS key settings are configured in Ansible and Packer code.
- Test 2: During operations, staff create a server with this AWS AMI and check EBS encryption and KMS key settings in a Continuous Integration (CI) pipeline with the AWS CLI.
- Testing Scenario 1: Test 1 and Test 2 must be checked.
  - Operator: and
  - Condition
    - Pre-condition 1: A specified location for the IaC tools contains valid files.
    - Pre-condition 2: Static analysis tool is installed and configured on the developer workstation
    - Test 1
  - Condition
    - Pre-condition 1: a valid CI environment is defined.
    - Pre-condition 2: a valid CI pipeline file is defined.
    - Test 2
- Component: this component describes servers that are part of the information system.
  - Rule-implementation: Testing Scenario 1 implements requirements for NIST SP 800-53 Rev 5 control SC-28.

Feedback

GitHub: github.com/usnistgov/OSCAL/issues/1058

Gitter: gitter.im/usnistgov-OSCAL/Lobby

Mailing list: oscal-dev@list.nist.gov