

oscal-compass

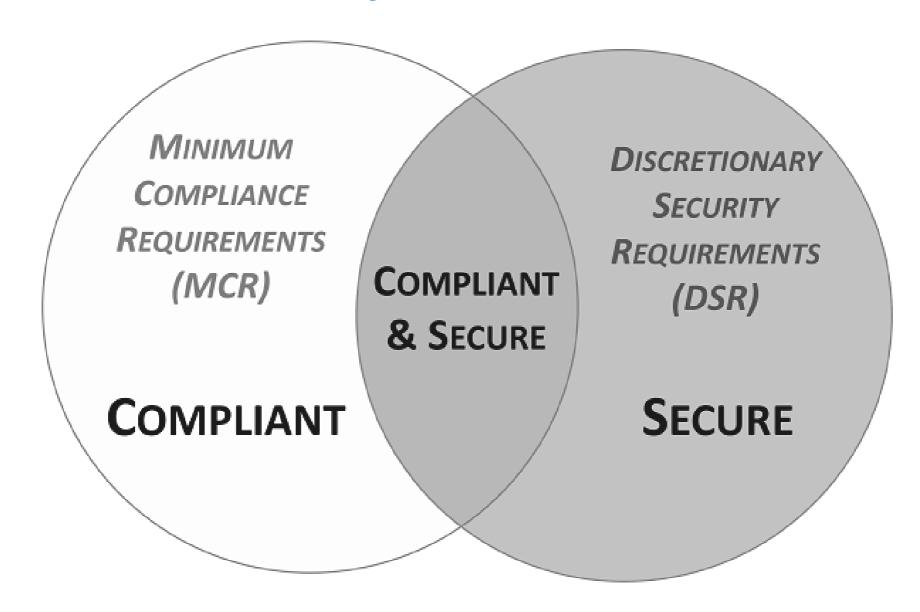
OSCAL-COMPASS Open Security Control Assessment Language Compliance Automated Standard Solution

Vikas Agarwal, Lou DeGenaro, Manjiree Gadgil, Alejandro Leiva, Jenn Power, Anca Sailer, Takumi Yanagawa

Agenda

- Compliance v/s Security
- Compliance artifacts
- Personas in compliance governance and lifecycle
- OSCAL-COMPASS projects overview
 - Compliance Trestle
 - Compliance Agile Authoring
 - Compliance to Policy

Compliance v/s Security



Compliance Artifacts and their Representation as code

Regulatory compliance and Org Policy controls are implemented as rules (technical,

operational, financial, data, or AI) and tested via rule engines or checks based on evidence Example regulatory standards and certifications: Regulatory Compliance and Org **Policies** ISM & E8 (AU) Regulation Specific: MINIMIZE NEGATIVE IMPACT SYSTEM AND Controls OF AI COMMUNICATION e.g., NIST AI RMF MG-2.2-005: PROTECTION (SC): **NIST 800-53 CONTROLS** Filter output for harmful or NIST SC-7: Boundary biased content, Protection **MODEL SERVING and NETWORK** Technology Specific: **MONITORING SECURITY** Ensure that model output has **Technical Control** Ensure Cloud Object Storage is **Implementations** bias score of less than a enabled on private endpoints threshold of 20% w/ associated Ensure Kubernetes ingress is enabled through Cloud Ensure that model output has policy RULES harmful content score of less **Internet Services** procedures than a threshold of 10% Ensure VPC LBaaS is enabled through Cloud Internet Ensure that model output has Services Technology Specific: If config.endpoint ≠ param return FAIL; If model_out.bias <= param1</pre> test **CHECKS** scripts else return PASS; return PASS; Technology Specific: "service_instance_id": "xxxxxxxx-6c4f-4a62else return FAIL;

a16d63903", "endpoint": "private",

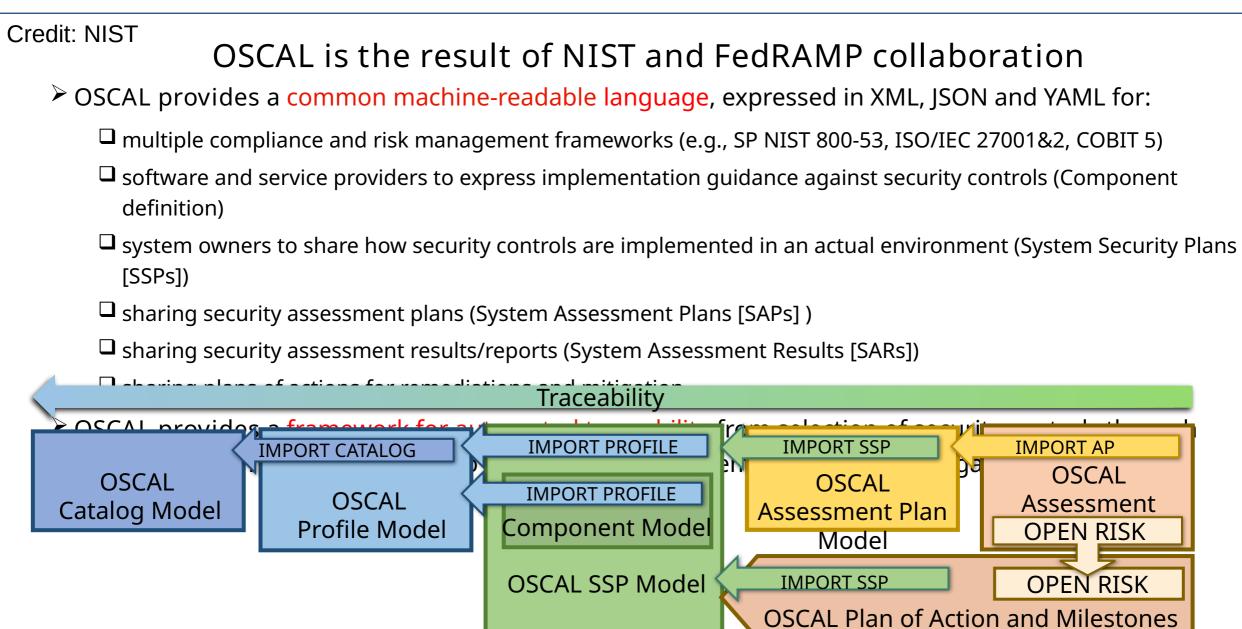
"encrypt": "KYOK"}

EVIDENCE

Compliance Governance, Lifecycle, Personas Regulators define: System Owners select services & products, Compliance Officers, Auditors, deploy apps, are responsible to ensure their - regulations, standards, laws, catalogs System Operators examine the policy PCI, SOC2, NIST 800-53, FedRAMP, HIPAA compliance, define policy assessment plans and assessment results and apply CIS benchmarks: IBM Cloud, OCP, Kube set compliance monitoring scans remediations or deviations - crosswalks mappings between regulations' controls Compliance Team select and tailor control profiles (aka baselines) to describe the Governance Team compliance intent of their regulated evaluate the controls organizations and environments performance, precision, and coverage Catalogs, Crosswalks Organization Profiles, **Environment & Apps** Results Composite Profiles Compliance Scope & Scans Organization Crosswalks Governance, Risk, Agile Compliance GitHut Authoring (GRC) Center **Key Features Key Features** Component Definitions, **OSCAL** artifact Policies, Risk, POAM **POAM Reference** Compliance2Policy (C2P), Exchange C2P ations, Exceptions GitOns workflow (Products, Svcs, Processes) **Key Features** EP3 EP0 CISO-CTO Process Providers or Products & Business Owners and Risk Managers OSCAL based API set context, assess, respond and monitor Services Providers define policies & procedures & restle SDK based normalization plug-Consume POAM, and declare the controls implemented by risk using the Plan of Actions and their components (sw. hw. processes) through Milestones mapping to [policy profile desired state & Results & policy rules (technical, operational, financial) Inventory update inventory scope] Component Definitions policy Validation - Enforcement Points (PVPs - PEPs) w/ declarative & imperative (Policy Validation or policies **Enforcement Points**) CEL, OPA Gatekeeper, Kyverno, Ansible, Terraform OPA, ArgoCD Ansible, Auditree, Erictree, OPA OSCO Assessment Plans DAY 1: CICD, apps, data, AI pipeline DAY 2: IaaS/PaaS/SaaS, apps, data, FS ops, AI (Kubernetes Policies) GitHub Policy Validators or Control Assessors declare the policy checks assessing in their policy validation or enforcement engines the policy rules declared by the Controls *Providers or Owners: use the* aws Evidence Evidence aryaka pre-defined evidence data model IBM Cloud Actual State Data Model or BM Ouantum **Templates Spec**



What is OSCAL?



OSCAL, Trestle, Agile Authoring, Compliance-to-Policy

https://pages.nist.gov/OSCAL/

https://github.com/oscal-compass

https://github.com/oscal-compass/compliance-trestle

https://oscal-compass.github.io/compliance-trestle/



OSCAL is a NIST framework & language for managing compliance artifacts as code end-to-end

From selection of security controls through implementation and assessment

To plans of actions for remediations and mitigation



TRESTLE is an opinionated implementation of the OSCAL standard

Allows editing and manipulation of OSCAL documents while making sure the schemas are enforced

Provides an SDK



AGILE AUTHORING is a collaborative platform enabling various compliance personas to orchestrate their individual aspects of the compliance artifacts via an interface of their choice

Trestle-based GitOps automated workflow Ensures artifacts consistency and traceability



COMPLIANCE_TO_POLICY is a GitOps extension as a pluggable bridge to normalize the policy administration in the policy validation tools

Bridge between compliance-as-code and policy-as-code

Trestle: An open-source OSCAL SDK



Trestle is an ensemble of tools that enables the creation, validation, and governance of documentation artifacts for compliance needs.

- Git repository as a single source of truth for managing compliance artifacts, change history and approvals.
- JSON format for representing OSCAL data and Python as the programming language for easy scripting and enforcing the schema.
- Command line interface instead of GUI to expose its functionality for easy integration with CI/CD tools.
- Markdown format for human users for easy reading and editing of structured documents with seamless conversion to OSCAL JSON and vice-versa.

Trestle Architecture



Applications

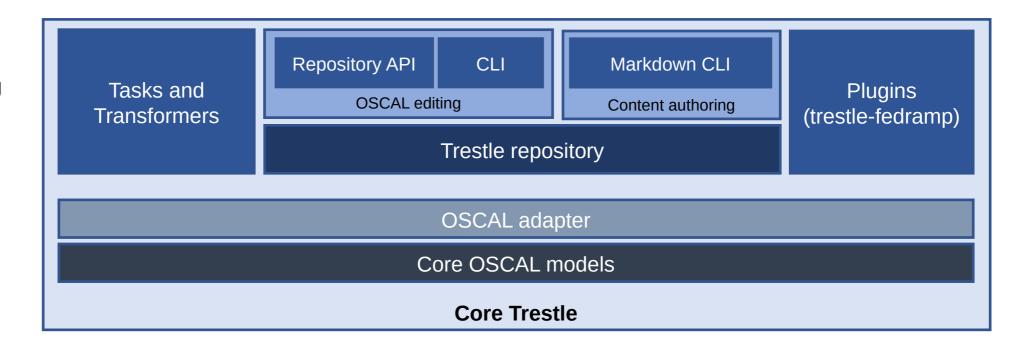
Governed regulatory control content authoring and approval workflows

Specialized Cloud / FedRAMP SSP Workflows

Format conversions to/from OSCAL (e.g., spreadsheet, word doc, native artifacts)

Editing / authoring / transformation APIs and CLIs

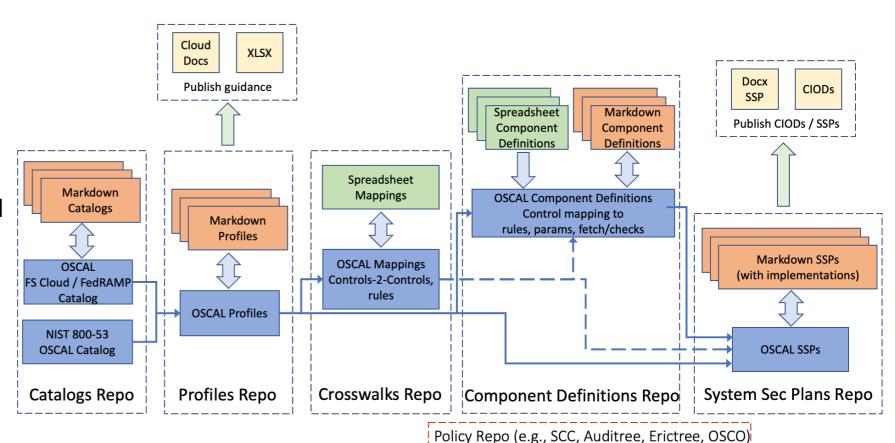
Trestle Base



Agile Authoring: Collaborative Authoring Platfor



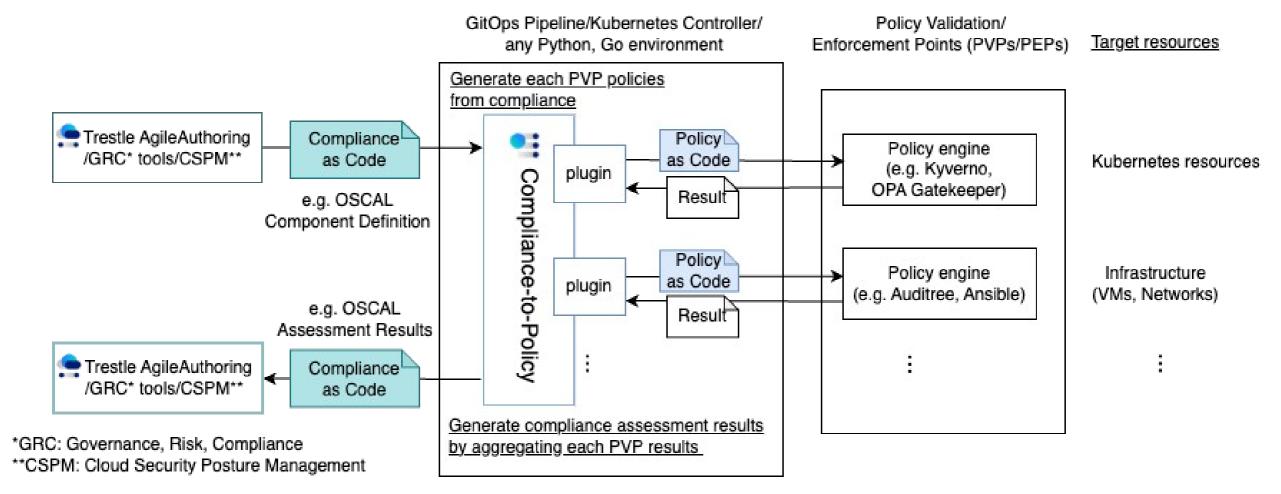
- Human friendly authoring /editing of compliance content
- Structured and auditable workflow
- Trigger automatic validation, updates, and deployments
- Collaborative editing and review process through code review and approval process
- Automatic semantic release management



Compliance-to-Policy (C2P) and plugin architecture



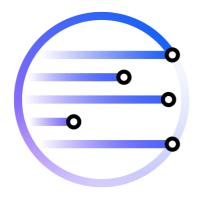
Flexibility in choice of policy engines and compliance framework Community-driven plugin extension



OSCAL Compass Community

Where To Start

- Our community <u>README.md</u>
- Our biweekly <u>community calls</u>



Decision Making

We strive for a consensus-based approach to encourage open discussion and collaboration on most project decisions.

We use a voting based approach when necessary or if consensus cannot be reached or in special circumstances.

Leadership

We have an Oversight Committee made up of maintainers across the projects and project representatives. Learn more at <u>GOVERNANCE.md</u>.

Contribution

We welcome contributions from everyone! Whether you're a seasoned developer or just starting out, we value your input. Learn more at CONTRIBUTING.md.

Keep up with Compass and Trestle

- Community calls
 - OSCAL Compass community calls -<u>https://docs.google.com/document/d/1XTYM7xnWlIqd-8Nn5-qtgvgk8kH3NSmYle5yZvaS7qs/edit#heading=h.6pq38r2red0n</u>
- Github organization
 - oscal-compass https://github.com/oscal-compass
- Blogs
 - Personas and Roles
 - Trestle SDK
 - Artifacts and Personas
 - <u>Topologies of Compliance Policy Administration Centers</u>
 - A Lack of Network Boundaries Invites a Lack of Compliance
 - Compliance to Policy for Multiple Kubernetes Clusters

