oscal-compass

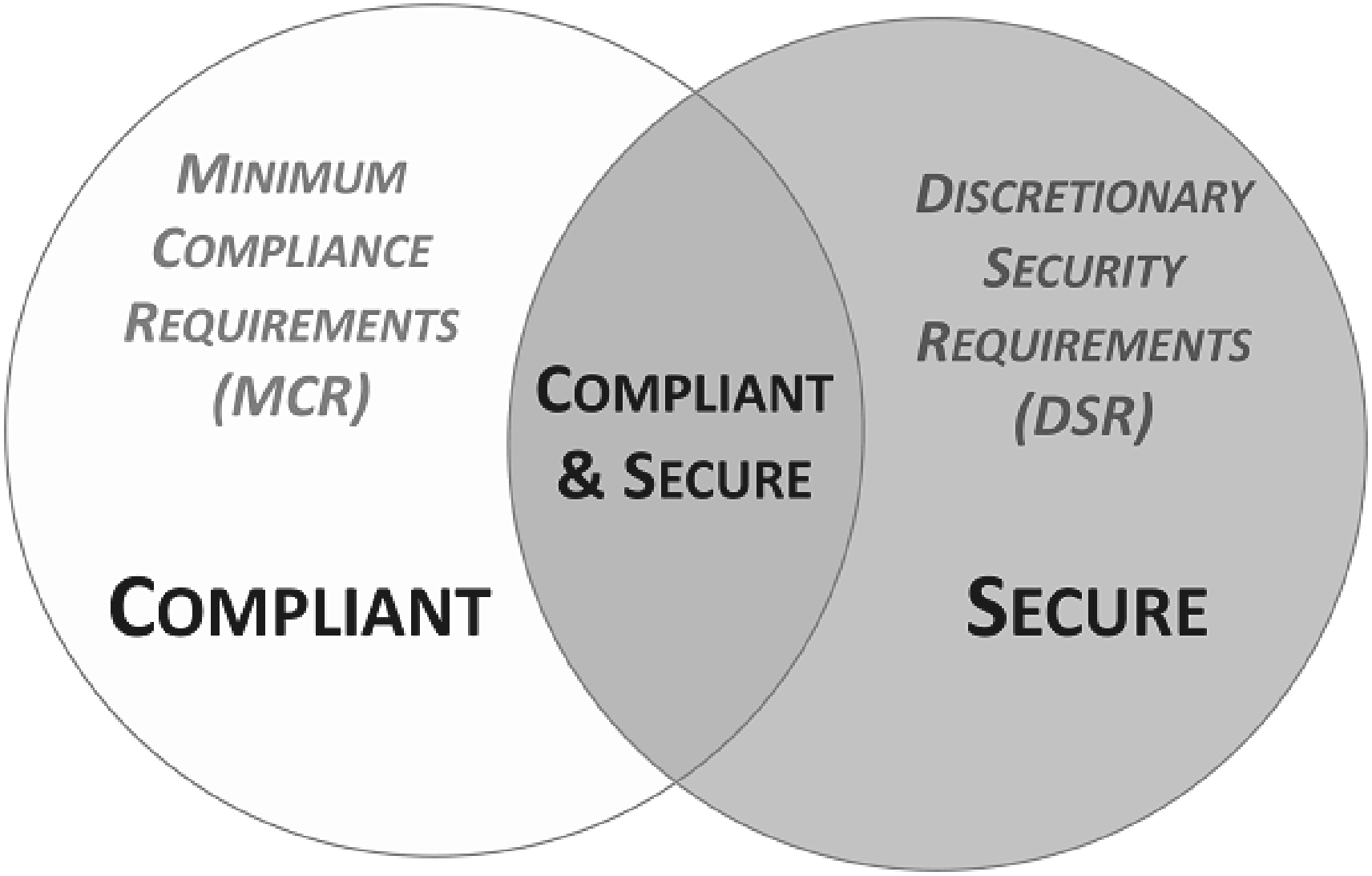
# OSCAL-COMPASS
# Open Security Control Assessment Language Compliance Automated Standard Solution

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

# Agenda

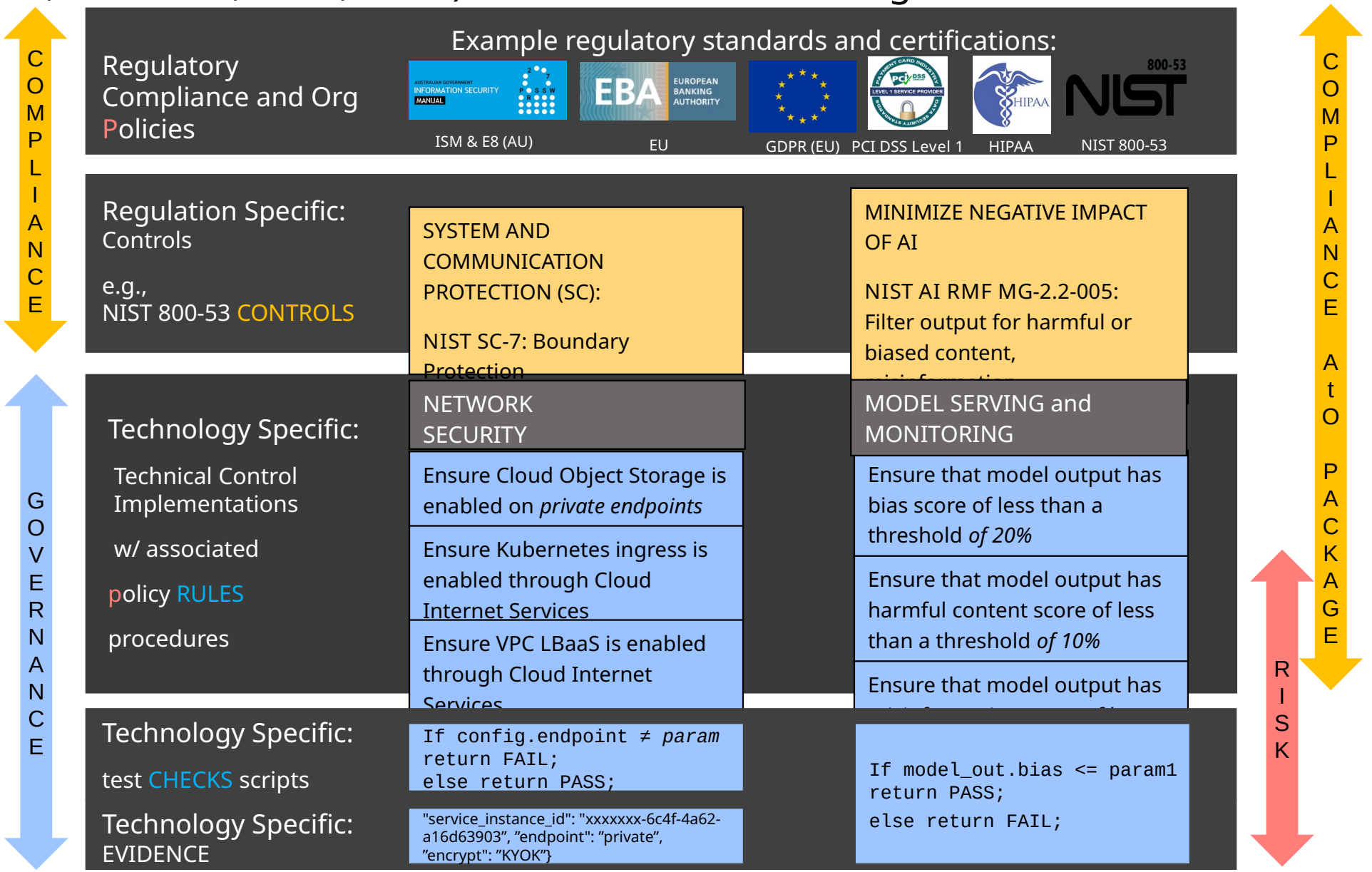
- Compliance v/s Security
- Compliance artifacts
- Personas in compliance governance and lifecycle
- OSCAL-COMPASS projects overview
  - Compliance Trestle
  - Compliance Agile Authoring
  - Compliance to Policy

# Compliance v/s Security

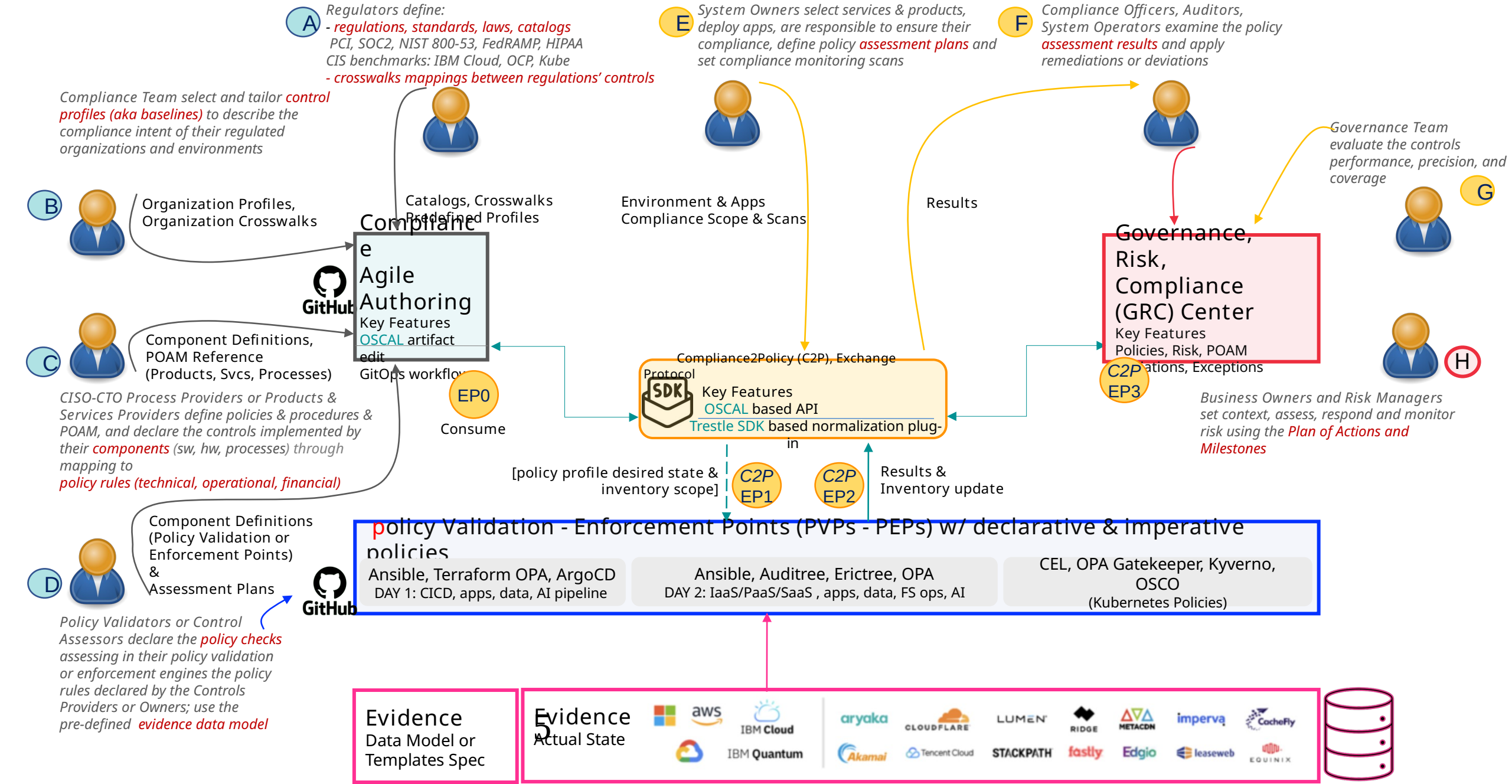


# Compliance Artifacts and their Representation as code

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



# Compliance Governance, Lifecycle, Personas

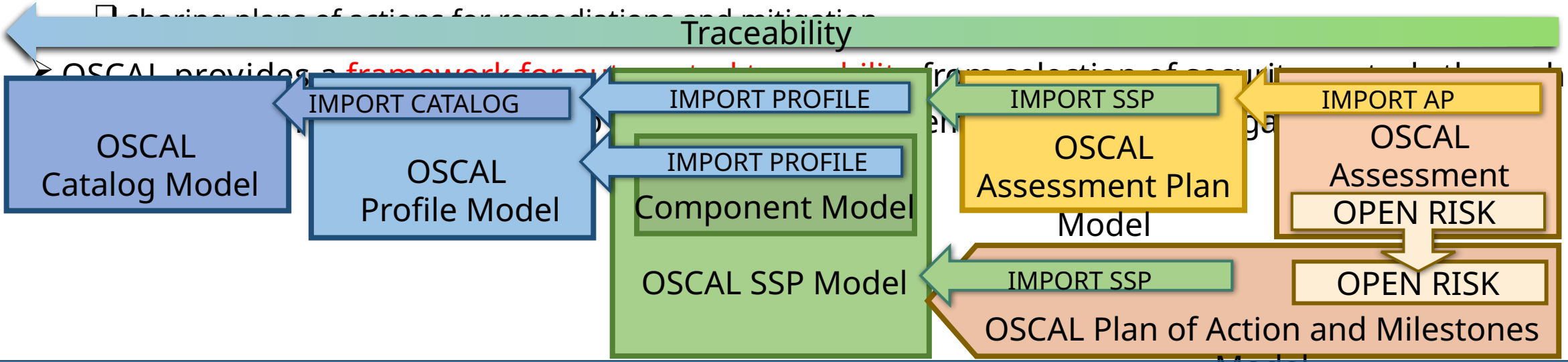


# What is OSCAL?

Credit: NIST

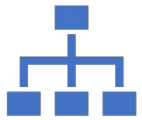
## OSCAL is the result of NIST and FedRAMP collaboration

- OSCAL provides a common machine-readable language, expressed in XML, JSON and YAML for:
  - multiple compliance and risk management frameworks (e.g., SP NIST 800-53, ISO/IEC 27001&2, COBIT 5)
  - software and service providers to express implementation guidance against security controls (Component definition)
  - system owners to share how security controls are implemented in an actual environment (System Security Plans [SSPs])
  - sharing security assessment plans (System Assessment Plans [SAPs] )
  - sharing security assessment results/reports (System Assessment Results [SARs])
  - sharing plans of actions for remediations and mitigation
- OSCAL provides a framework for automated traceability from selection of security controls through



# 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

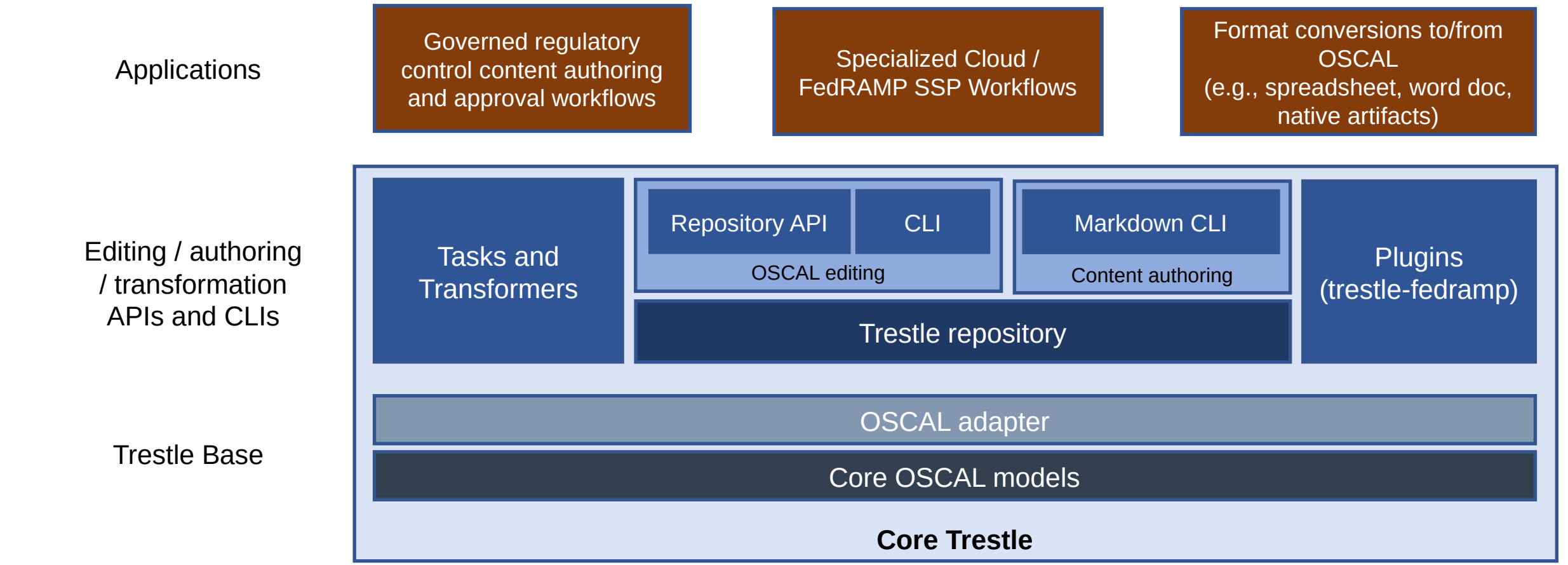
Bridge between compliance-as-code and policy-as-code

# 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.

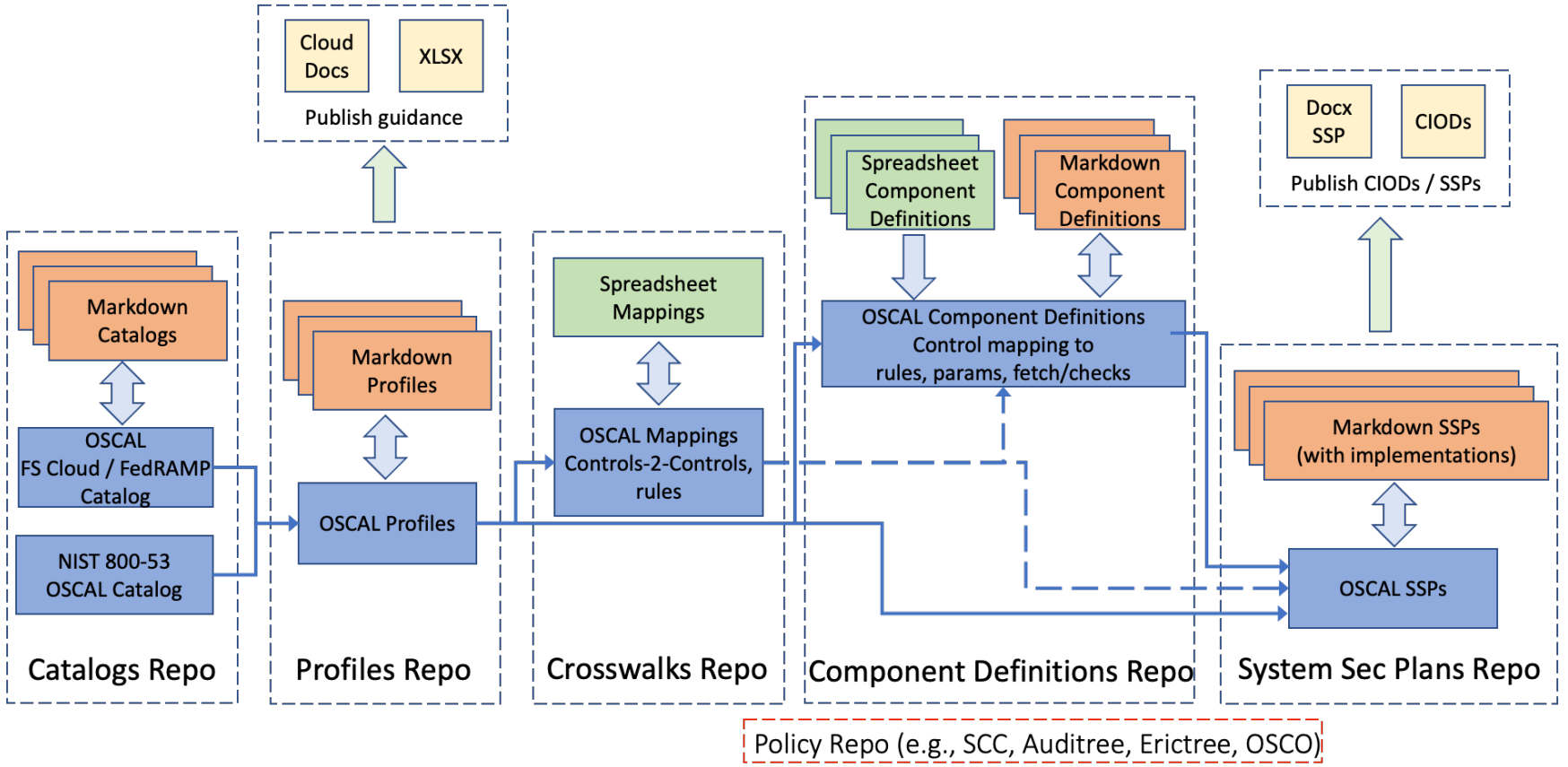
- Git repository as a single source of truth for managing compliance artifacts, change history and approvals.
- JSON format for representing OSCAL data and Python as the programming language for easy scripting and enforcing the schema.
- Command line interface instead of GUI to expose its functionality 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



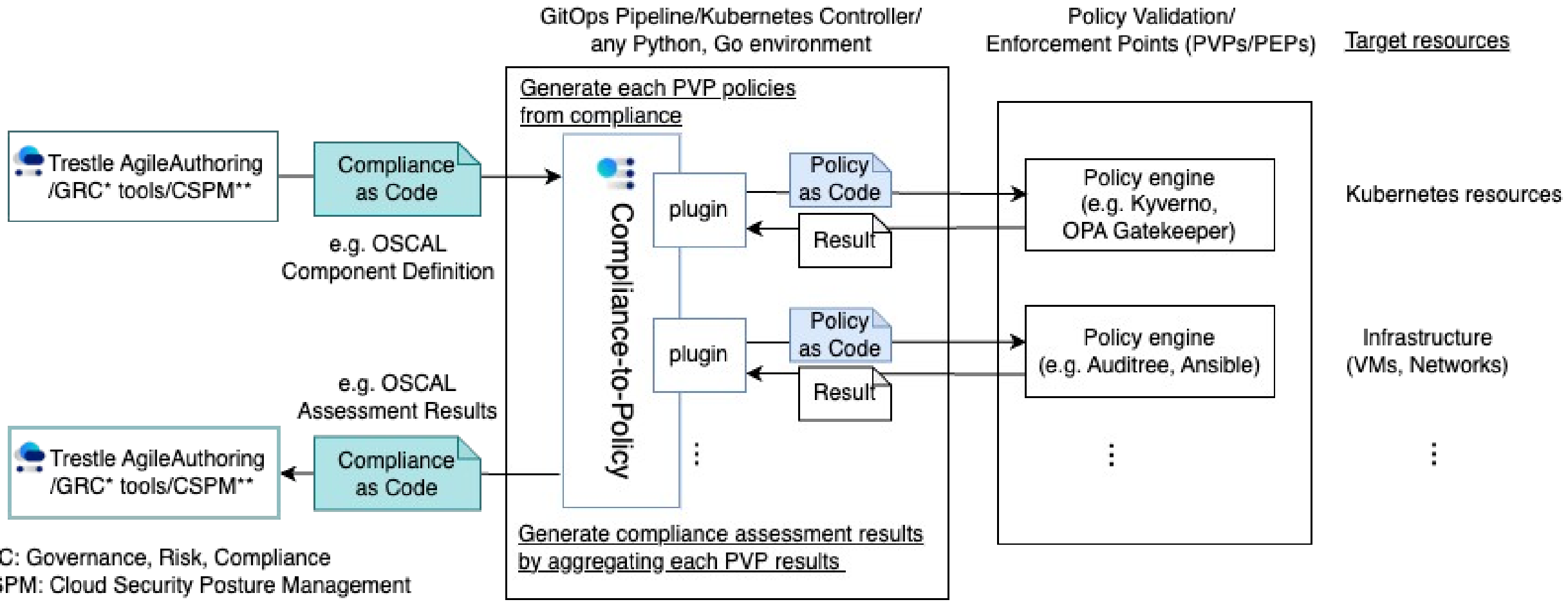
# Agile Authoring: Collaborative Authoring Platfor

- 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

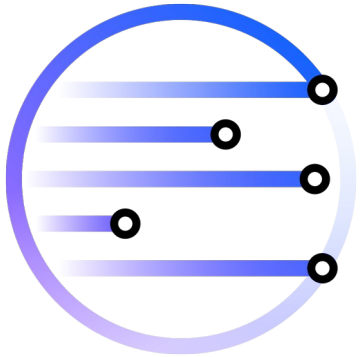
Flexibility in choice of policy engines and compliance framework
Community-driven plugin extension



# OSCAL Compass Community

## Where To Start

- Our community README.md
- Our biweekly community calls



### 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 GOVERNANCE.md.

### 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 - https://docs.google.com/document/d/1XTYM7xnWlIqd-8Nn5-qtgvgk8kH3NSmYle5yZvaS7qs/edit#heading=h.6pq38r2red0n

- Github organization
  - oscal-compass - https://github.com/oscal-compass

- Blogs
  - Personas and Roles
  - Trestle SDK
  - Artifacts and Personas
  - Topologies of Compliance Policy Administration Centers
  - A Lack of Network Boundaries Invites a Lack of Compliance
  - Compliance to Policy for Multiple Kubernetes Clusters