# Automated Governance

## Modular Assessments for Feedback Loops

Brandt Keller - OSS Maintainer

April 2024

# Overview

- What is Automated Governance?
- The Problem
- Modular Assessments for Software Compliance
- Lula Introduction
- Demo
- Summary
- Where next?
- Call to Action

# Automated Governance

“An automated process for tracking governance throughout the deployment pipeline” [1]

- Treat Governance as a required quality gate in the deployment to production (CI/CD).

“Security, compliance, and audit needs are met 100% for every commit. With Automated Governance, there are no workarounds or shortcuts that inadvertently lead to risk and vulnerabilities.”[2]

[1] DevOps Automated Governance Reference Architecture - 2019 https://itrevolution.com/product/devops-automated-governance-reference-architecture/
[2] What is Automated Governance - 2022 https://itrevolution.com/articles/what-is-automated-governance/

# The Problem (Why)

- Speed is the ultimate discriminator
  - Even systems delivered via DevOps wait 6+ months in compliance mapping, response statements and approval

- Competency resides with the code, not after
  - Code owners/maintainers can and should be accountable to compliance attestation

- Drift is real
  - SREs change systems at will after point in time compliance checks and systems no longer accurately reflect risk posture

# Modular Assessments (How)

- Individual Application Development
  - Include processes to establish mapping of controls to a specific “component” of a future system - owned by the application expertise. These are Source of Truth and will be aggregated to systems.
- Modular Assessments
  - Ability to Assess required controls that apply or are provided by a given application such as to reconcile updates or dependencies.
- Evaluation
  - Established thresholds can be used to provide different required Compliance based on scope (application, namespace, global, etc)

# Lula

- Free and Open Source Project
  - https://github.com/defenseunicorns/lula
  - Apache 2 license - No data or vendor lock
- Runtime validation of control satisfaction using OSCAL + automation
- OSCAL-native = Data freedom and no lock-in
- Validation format - Data to Collect + Adherence Required = Proof

# DEMO

Open
Security
Controls
Assessment
Language

# Summary

- Component driven assessments for quick feedback loops
  - Close proximity to the Code
- Validation Proofs provide clear structure
  - What is being measured + Adherence to Policy
- OSCAL Native
  - Data Freedom and portability
- Extensible
  - Future Data collection and Policy Provider enhancements

# Next Steps for Lula

- OSCAL Artifact Template Generation
  - Component Definition
  - Assessment Plan
  - System Security Plan
  - Plan of Actions and Milestones
- Reporting
  - Gap Analysis of system state vs Standard (catalog)
- Data Collection
  - Native interfaces to handle collecting data from Cloud infrastructure, Vulnerabilities, and other key data sources

# Call to Action

- Intent to donate this project to an Open Source foundation
  - Governance
- Help the project grow
  - Guidance, Development, Testing
- Bring your use cases
  - Enhance depth of accessibility to unique circumstances
  - Provide insight through an issue that highlights what you would like to accomplish