About this presentation

This presentation was originally presented at the2024 Linux Foundation Open Compliance Summit.

Red Hat

OPEN COMPLIANCE SUMMIT

Managing compliance artifacts as code with oscal and `compliance-trestle`

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

Dr. Chris Butler

Senior Principal Chief Architect

chris.butler@redhat.com

Red Hat

Fundamental challenges for organisations in the cloud regulatory environment

How does a *service* owner measure the compliance posture of their system to standards or regulatory framework?

How does a *software project* or service owner surface compliance relevant information to consumers?

How do we maintain developer productivity for security and compliance tasks

Requirements flow down from enterprise onto open source projects to meet and measure these controls

Historically standards such as SCAP have provided low level controls such as STIGs and CIS benchmarks

ComplianceAsCode

Operating systems,
Kubernetes,
Some middleware

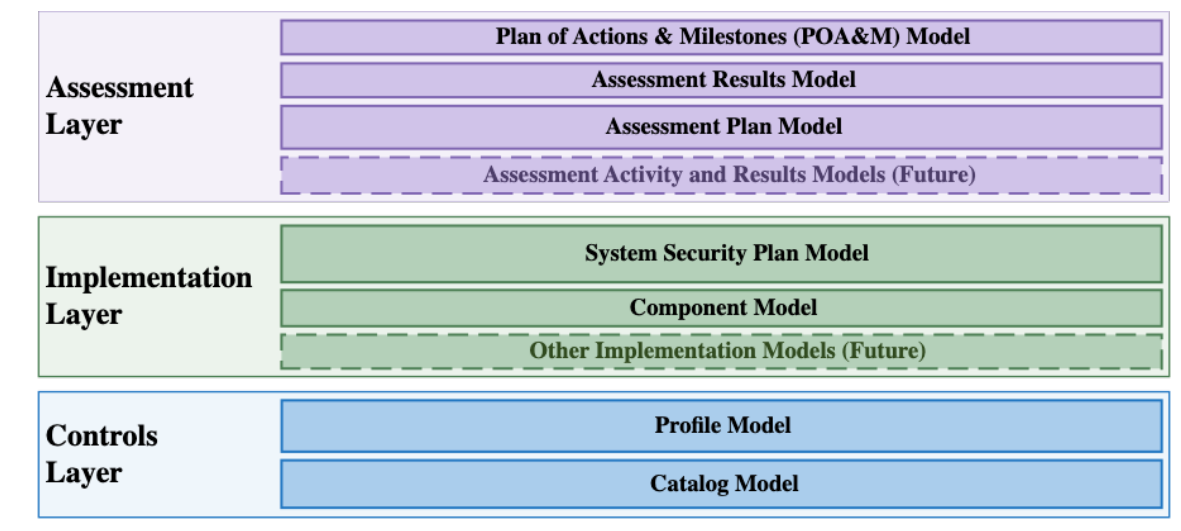
Based on SCAP

?

?

Services, Systems, leveraged authorisations
arbitrary components
OSS software

OSCAL provides a higher level set of data models designed for cloud security assessments



GSA

GOVTECH SINGAPORE

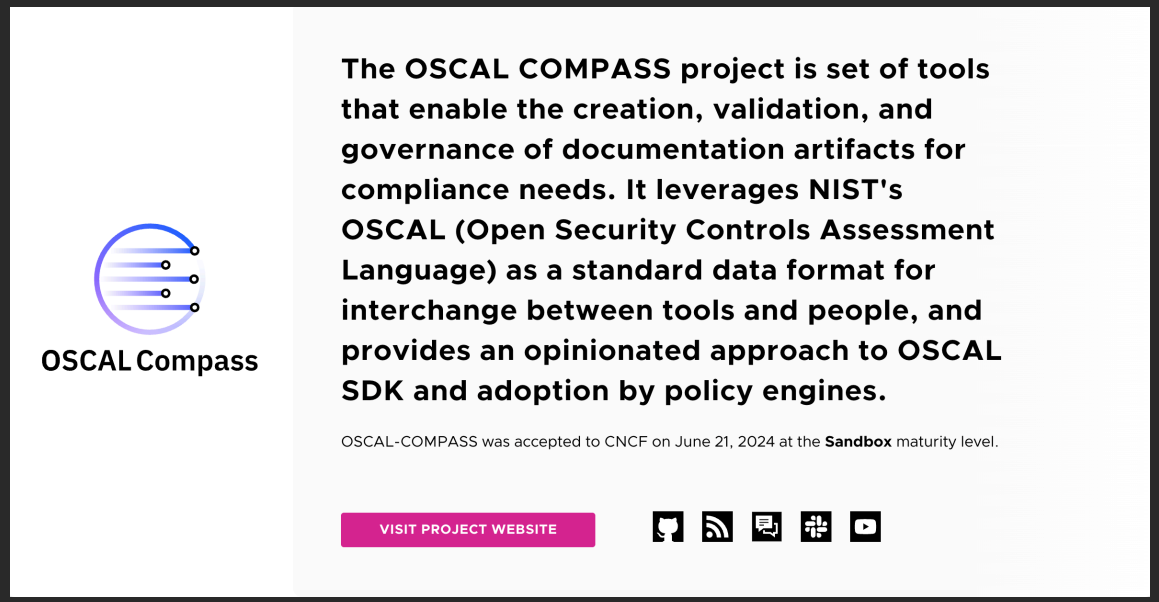
ACSC Australian Cyber Security Centre

CIS Center for Internet Security®

Missing link today: Explicit connectivity to tools executing compliance controls

Red Hat

OSCAL Compass is a *new CNCF* sandbox project for regulatory compliance tooling

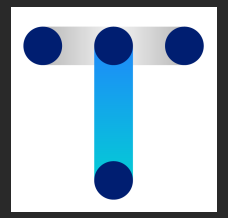


CLOUD NATIVE COMPUTING FOUNDATION

Sub-projects

- `compliance-trestle`
- `compliance-trestle-fedramp`
- `compliance-to-policy`
- `agile-authoring`

Red Hat



`trestle` is a SDK and cli tooling for managing OSCAL artefacts

- Manipulation and generation of OSCAL-artefacts
- Python object model for OSCAL objects
- Markdown based editing
- Library of tasks to import to and export from OSCAL
- Customisation hooks for arbitrary workflows
-

Contributors from

IBM

Meta

GSA 18F

EASY DYNAMICS

aws

Red Hat

Goal is to provide tooling for developers so the source code management system can be used as a single source of truth

```
.
├── .trestle
├── dist
│   ├── catalogs
│   ├── profiles
│   ├── component-definitions
│   ├── system-security-plans
│   ├── assessment-plans
│   ├── assessment-results
│   └── plan-of-action-and-milestones
├── catalogs
├── profiles
├── component-definitions
├── system-security-plans
├── assessment-plans
├── assessment-results
└── plan-of-action-and-milestones
```

Example: Using trestle to edit SSPs as markdown

https://redhatproductsecurity.github.io/trestle-bot/

Red Hat Product Security use: *trestle-bot*

Used for managing Red Hat's SSP and components for FedRAMP high

- Abstract users from Git{Hub|Lab} to simplify experience for non-developers
- Automation of pull requests
- Syncing of with upstream repositories (particularly for low-to-high workflows)
- Pre-baked workflows on top of trestle

https://redhatproductsecurity.github.io/trestle-bot/

Open problems

Standardised integration points

Lower level tool integration (e.g. SCAP) and OSCAL to OSCAL transforms.

Evidence lockers

Assessment / audit requires tamper-evident storage to avoid manual data collection

Verifiable assertions

Leveraged assessments require the ability to build trust across entities.

Geopolitics driving differing regulatory frameworks will remain an enduring issue

Come and join us!

Community calls

Every other Tuesday

starting on April 23, 2024 ·

11:00 – 11:30am ET

CNCF Slack

Help required:

APJ timezone interested users / contributors :-)

chris.butler@redhat.com