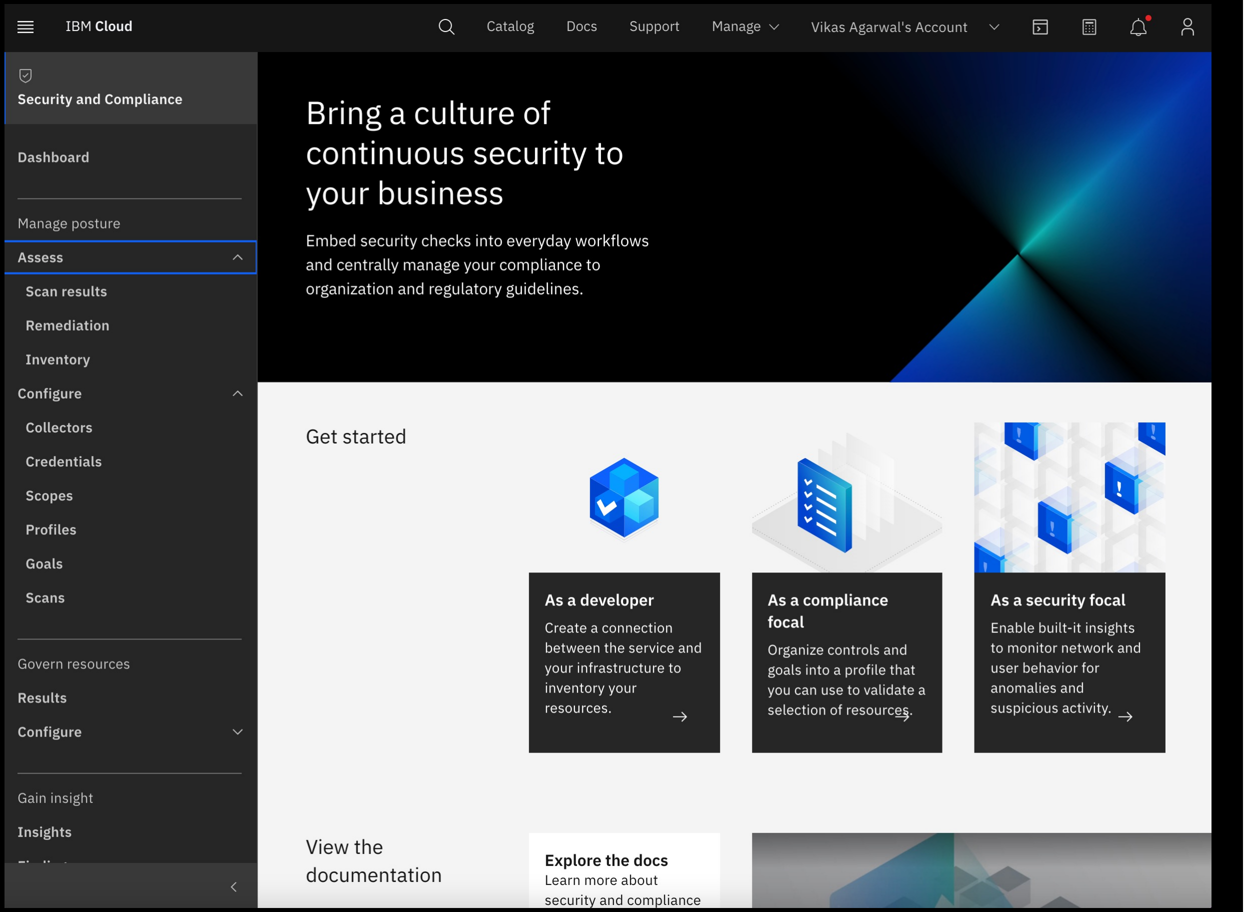
Exchange Protocol for Third Party Tool Integrations via OSCAL

Vikas Agarwal, Lou Degenaro, Anca Sailer
IBM Research

IBM Security and Compliance Center (SCC)

- Centrally manage compliance to organizational and regulatory requirements
  - Posture management
    - Automate security and compliance postures
    - Monitor for and prove compliance
  - Configuration Governance
    - Define configuration rules and templates to standardize provisioning and configuration of resources
    - Set guardrails to prevent unsecure configuration of resources
  - Security Insights
    - Detect vulnerabilities and threats
    - continuously monitor and analyze IBM Cloud resources in real-time for potential risk



Bring a culture of continuous security to your business

Embed security checks into everyday workflows and centrally manage your compliance to organization and regulatory guidelines.

Get started

As a developer
Create a connection between the service and your infrastructure to inventory your resources. →

As a compliance focal
Organize controls and goals into a profile that you can use to validate a selection of resources. →

As a security focal
Enable built-it insights to monitor network and user behavior for anomalies and suspicious activity. →

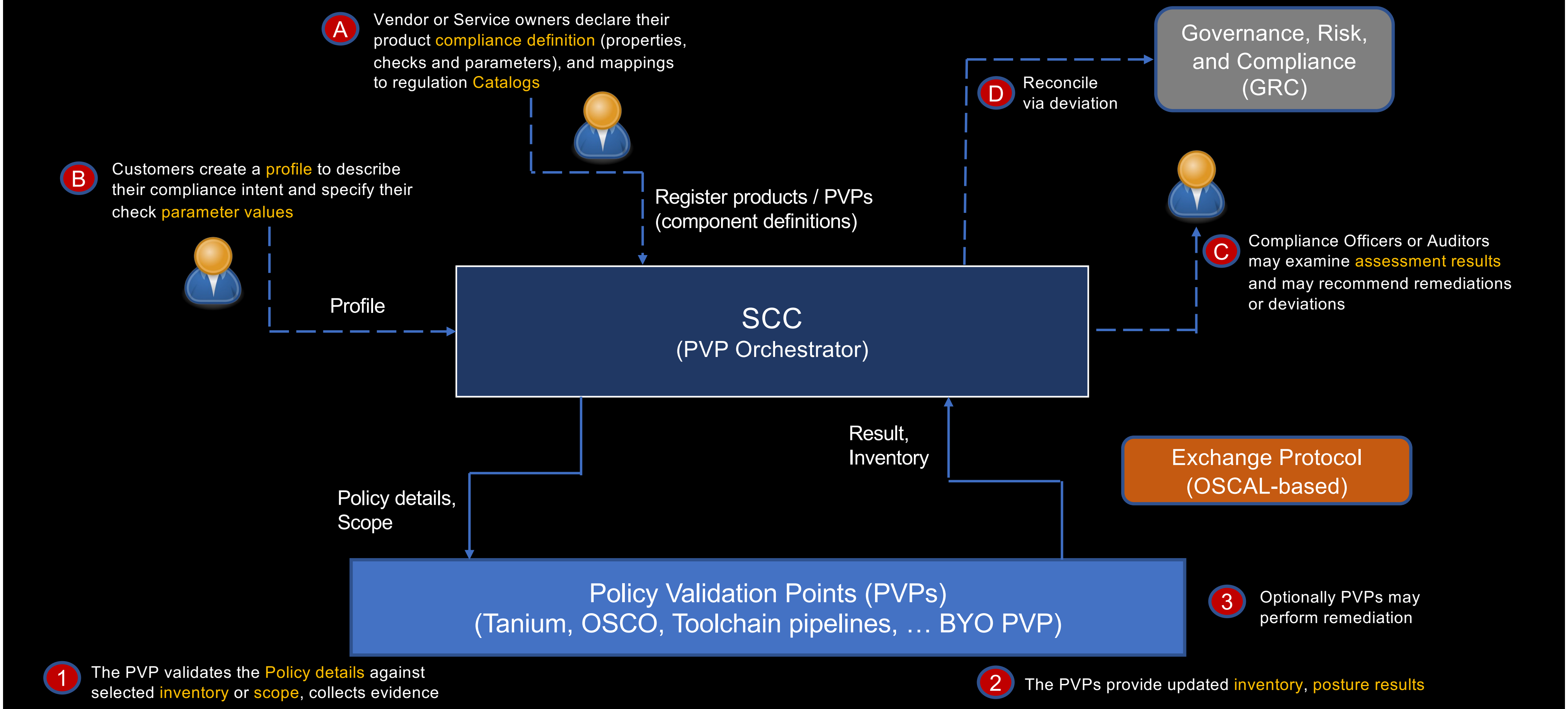
View the documentation

Explore the docs
Learn more about security and compliance

Motivation

- SCC acts as the Policy Validation Point (PVP)
  - Validates status against controls / rules
  - Shows that compliance status against a profile (across all inventory) or individual inventory items (across all controls / rules)
- Problem
  - What if a client uses external validation tools?
    - E.g., Tanium, OpenShift Compliance operator (OSCO), etc.
  - How can a client view the validation results of those tools?
    - Use tools specific UI to view validation status of specific resources
  - No single place for compliance officer to view validation results for their whole environment from multiple validation tools
- What is needed?
  - Transform SCC to a PVP orchestrator
    - a single place to see the results from multiple validation solutions
    - to get a complete view of the security posture of the infrastructure

Exchange Protocol Flow



A Vendor or Service owners declare their product compliance definition (properties, checks and parameters), and mappings to regulation Catalogs

Register products / PVPs (component definitions)

B Customers create a profile to describe their compliance intent and specify their check parameter values

Profile

SCC (PVP Orchestrator)

D Reconcile via deviation

Governance, Risk, and Compliance (GRC)

C Compliance Officers or Auditors may examine assessment results and may recommend remediations or deviations

Policy details, Scope

Result, Inventory

Exchange Protocol (OSCAL-based)

Policy Validation Points (PVPs) (Tanium, OSCO, Toolchain pipelines, … BYO PVP)

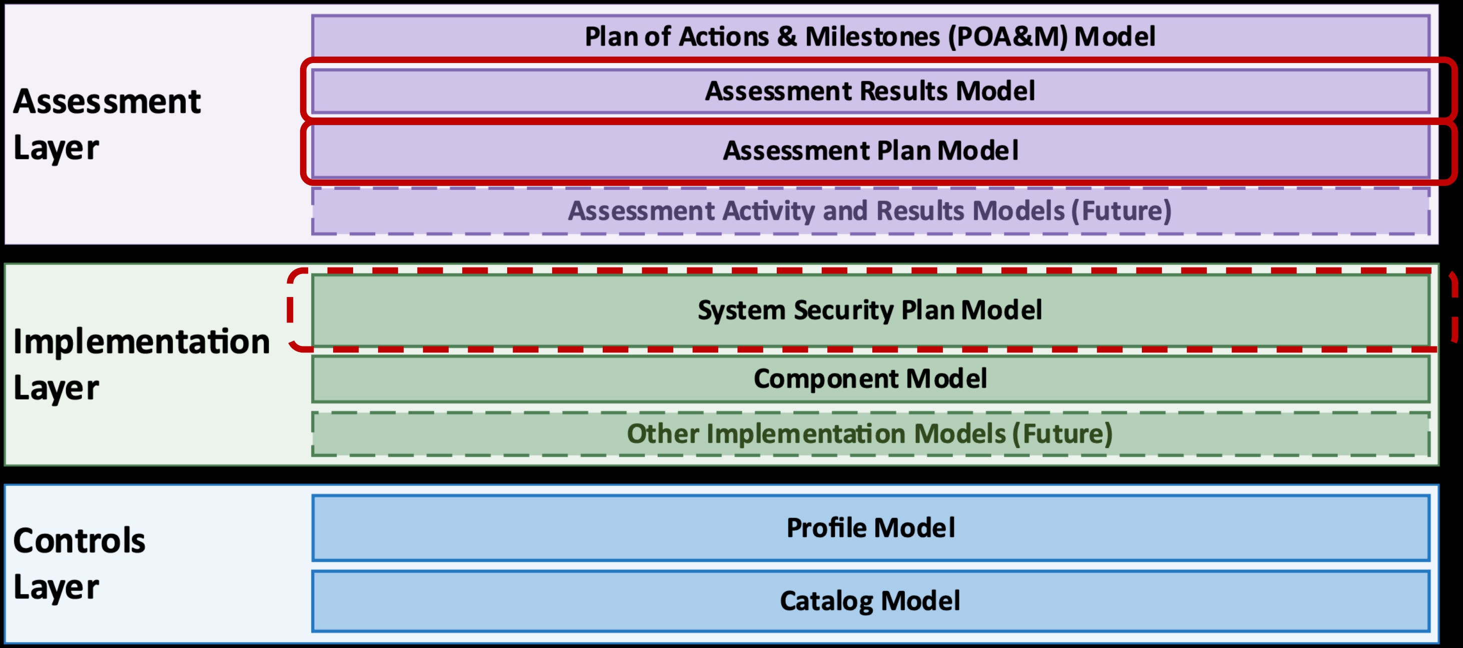
3 Optionally PVPs may perform remediation

1 The PVP validates the Policy details against selected inventory or scope, collects evidence

2 The PVPs provide updated inventory, posture results

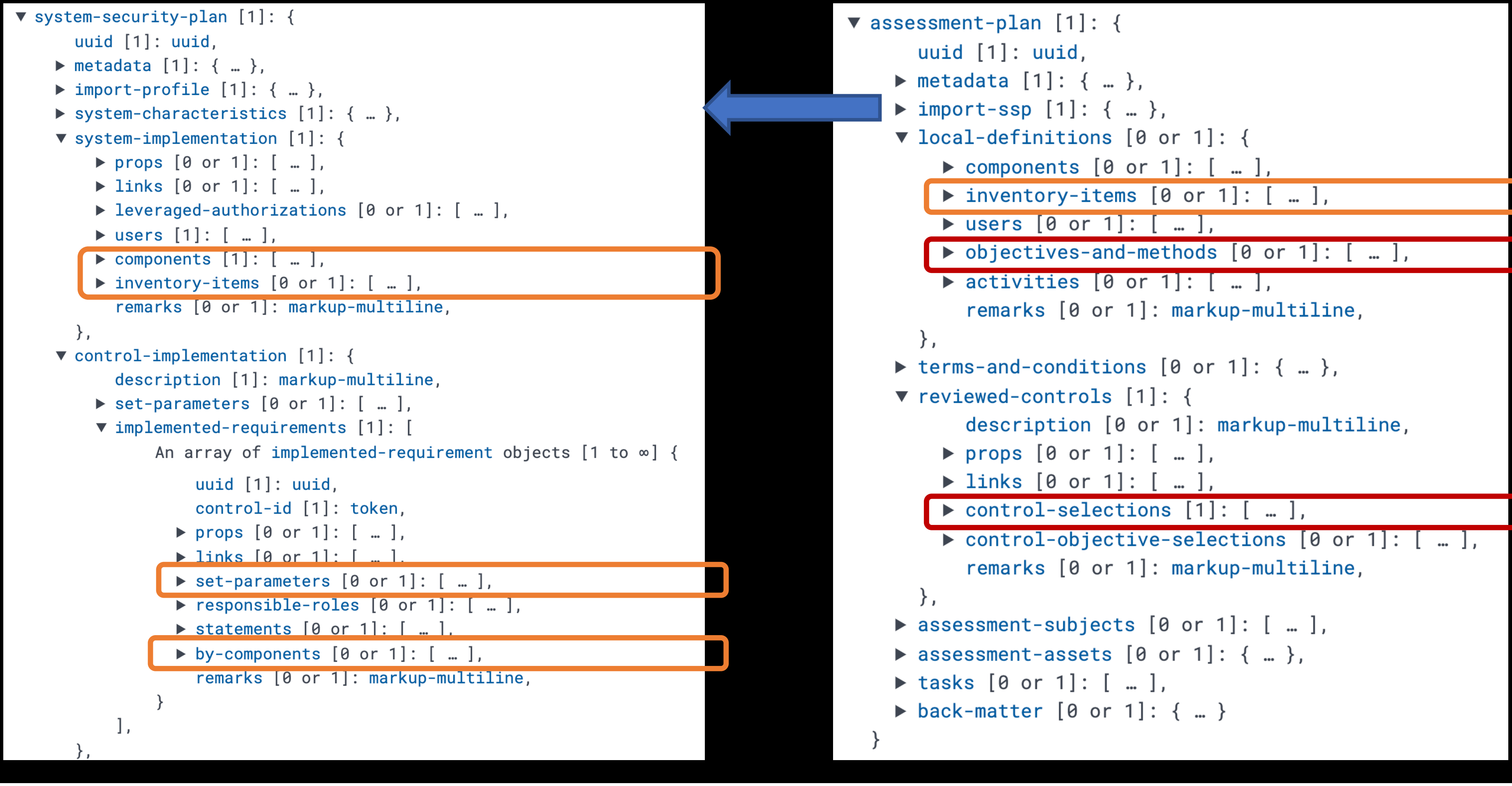
OSCAL: Open Security Controls Assessment Language

A common language for automating compliance
Standardized, data-centric framework for documenting and assessing its security controls


Assessment Layer
Plan of Actions & Milestones (POA&M) Model
Assessment Results Model
Assessment Plan Model
Assessment Activity and Results Models (Future)
Implementation Layer
System Security Plan Model
Component Model
Other Implementation Models (Future)
Controls Layer
Profile Model
Catalog Model

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

Use of OSCAL models



```
▼ system-security-plan [1]: {
    uuid [1]: uuid,
  ▶ metadata [1]: { … },
  ▶ import-profile [1]: { … },
  ▶ system-characteristics [1]: { … },
  ▼ system-implementation [1]: {
      ▶ props [0 or 1]: [ … ],
      ▶ links [0 or 1]: [ … ],
      ▶ leveraged-authorizations [0 or 1]: [ … ],
      ▶ users [1]: [ … ],
      ▶ components [1]: [ … ],
      ▶ inventory-items [0 or 1]: [ … ],
        remarks [0 or 1]: markup-multiline,
    },
  ▼ control-implementation [1]: {
        description [1]: markup-multiline,
      ▶ set-parameters [0 or 1]: [ … ],
      ▼ implemented-requirements [1]: [
            An array of implemented-requirement objects [1 to ∞] {
                uuid [1]: uuid,
                control-id [1]: token,
              ▶ props [0 or 1]: [ … ],
              ▶ links [0 or 1]: [ … ],
              ▶ set-parameters [0 or 1]: [ … ],
              ▶ responsible-roles [0 or 1]: [ … ],
              ▶ statements [0 or 1]: [ … ],
              ▶ by-components [0 or 1]: [ … ],
                remarks [0 or 1]: markup-multiline,
            }
        ],
    },
```

```
▼ assessment-plan [1]: {
    uuid [1]: uuid,
  ▶ metadata [1]: { … },
  ▶ import-ssp [1]: { … },
  ▼ local-definitions [0 or 1]: {
      ▶ components [0 or 1]: [ … ],
      ▶ inventory-items [0 or 1]: [ … ],
      ▶ users [0 or 1]: [ … ],
      ▶ objectives-and-methods [0 or 1]: [ … ],
      ▶ activities [0 or 1]: [ … ],
        remarks [0 or 1]: markup-multiline,
    },
  ▶ terms-and-conditions [0 or 1]: { … },
  ▼ reviewed-controls [1]: {
        description [0 or 1]: markup-multiline,
      ▶ props [0 or 1]: [ … ],
      ▶ links [0 or 1]: [ … ],
      ▶ control-selections [1]: [ … ],
      ▶ control-objective-selections [0 or 1]: [ … ],
        remarks [0 or 1]: markup-multiline,
    },
  ▶ assessment-subjects [0 or 1]: [ … ],
  ▶ assessment-assets [0 or 1]: { … },
  ▶ tasks [0 or 1]: [ … ],
  ▶ back-matter [0 or 1]: { … }
}
```

Policy Retrieval API

- Allow PVPs to pull the policy details from SCC in a standardized way
  - Policy – rules to be validated and their parameter values
- Information registered in SCC
  - Predefined Profiles
    - E,g, FS Cloud Profile with NIST controls, IBM Best Practices Profile, etc.
  - Product/Service control implementation details
    - How the profile controls (e.g., NIST, CIS-benchmarks) are implemented
      - Mapping from Profile controls to SCC rules
  - PVP checks (optional)
    - Mapping from rules to PVP checks
  - Scope
    - Specific resource groups
    - Applicable Profiles
- PVPs
  - Fetch policy details for given resource groups

Policy response from SCC to PVPs

```
{
  ”profile" : {
    "uuid" : "e7b3a8dc-3699-409c-a33b-cea71a0864d9",
    "metadata" : {
      "title" : ” ROKS OCP4 Tailored Profile",
      ”version" : ”1.0.0",
    }
    "props": [  {
          "name": "resource_group",
          "ns": "http://ibm.github.io/compliance-trestle/schemas/oscal/ibm-cloud",
          "value": "RG_1",
          "class": "scc_resource”,
          “remarks”: “link-1”
       },
       {
          "name": ”scope",
          "ns": "http://ibm.github.io/compliance-trestle/schemas/oscal/ibm-cloud",
          "value": ”Scope_1",
          "class": "scc_scope”,
          “remarks”: “link-1”
       } …]
    },
```

```
"imports" : [
      {
        "href" : "",
        "include-controls" : [
          {
            "with-ids" : [
              "xccdf_org.ssgproject.content_rule_etcd_unique_ca",
              "xccdf_org.ssgproject.content_rule_file_groupowner_kubelet_conf",
              "xccdf_org.ssgproject.content_rule_file_groupowner_worker_ca",
              "xccdf_org.ssgproject.content_rule_file_owner_cni_conf",
              …
            ],
    modify" : {
        "set-parameters" : [
            {
               "param-id":
"kubelet_eviction_thresholds_set_soft_memory_available",
               "values": [ "500Mi” ]
            }
        ]
    }
}
```

Results API

- Allow PVPs to push validation results to SCC in a standardized way
- Result information sent
  - Inventory items
    - resources that were validated
  - Observations
    - Rules that are checked for each inventory item and their status
  - Scope
    - Group to which the inventory items belong
  - Profile
    - SCC profile which is being validated

PVP Assessment-Result

(by profile and scope)

```
{ "results": [
    {
     "uuid": "e3c3236b-8812-4eda-a0a8-8e8f4600d87e",
    "description": "OpenShift Compliance Operator Scan Results",
     "start": "2021-06-14T18:35:04.000+00:00",
     "end": "2021-06-14T18:35:04.000+00:00",
     "props": [  {
                 "name": ”profile",
                 "ns": " http://ibm.github.io/compliance-trestle/.../oscal/ibm-cloud ",
                 "value": " ROKS OCP4 Tailored Profile ",
                 "class": "scc_predefined_profile”  },
                 {
                 "name": ”scope",
                 "ns": " http://ibm.github.io/compliance-trestle/.../oscal/ibm-cloud ",
                 "value": ”Scope_1",
                 "class": "scc_scope”  } ],
     "local-definitions": {
      "components": {
        "37e808bc-39c2-4313-b880-4b8b81841c0c": {
          "type": "Service",
          "description": "Red Hat OSCO for ocp4” } }  },
      "inventory-items": [
        {
          "uuid": "bf35822f-ee26-4a5f-96e3-106dc357f1da",
          "description": "inventory",
          "props": [
            {
             "name": "target",
             "ns": "http://ibm.github.io/compliance-trestle/schemas/oscal/ar/osco",
             "value": "kube-c18ler8d06m877hrn7jg-roks8-default-31.iks.ibm",
             "class": "scc_inventory_item_id" },

            ] … }
```

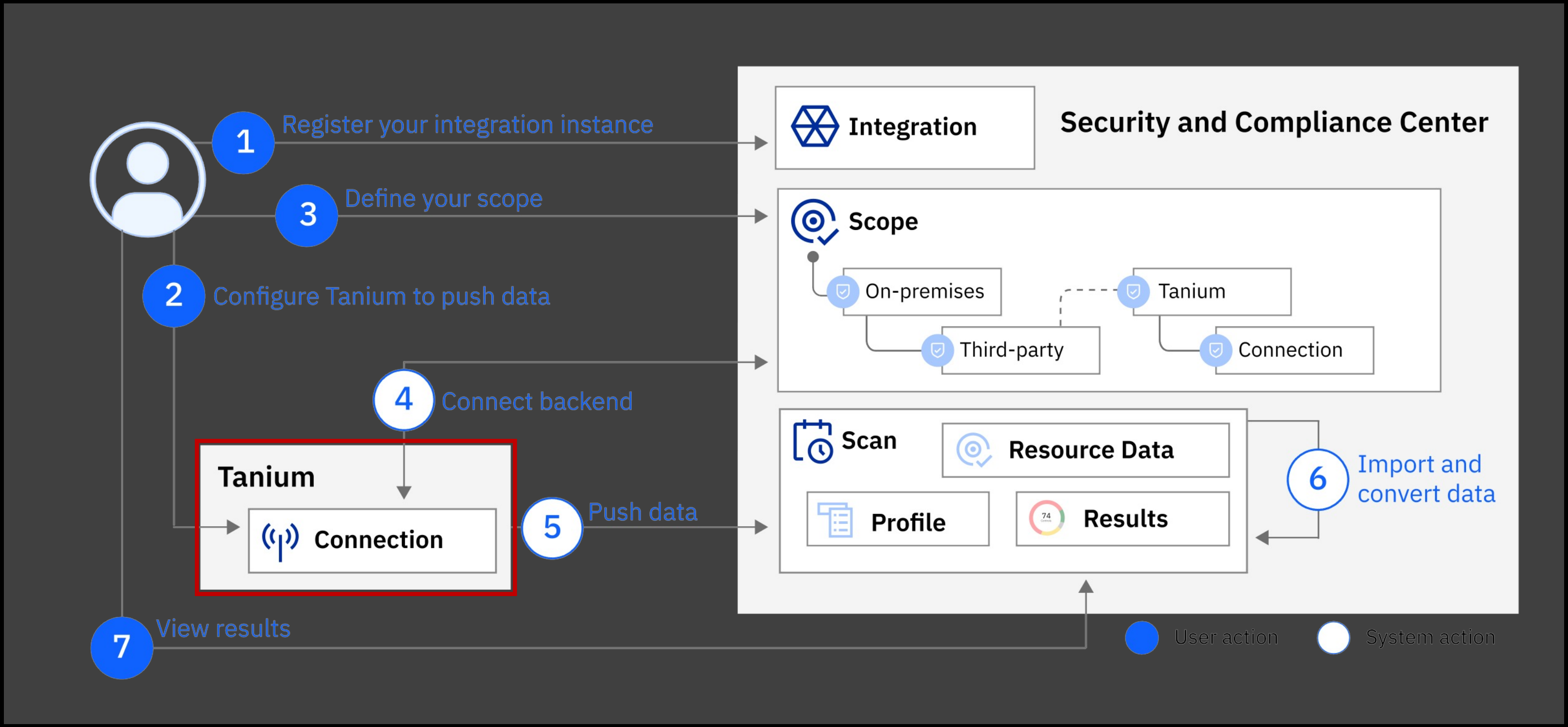
```
"observations": [
      {
        "uuid": "61ed11f7-62ba-4743-9de2-7f60c800ff72",
        "description": "xccdf_org.ssgproject.content_rule_file_integrity_exists",
        "props": [
          {
            "name": "idref",
            "ns": "http://ibm.github.io/compliance-trestle/schemas/oscal/ar/osco",
            "value": "xccdf_org.ssgproject.content_rule_file_integrity_exists",
            "class": "scc_goal_name_id”  },
          {
            "name": "version",
            "ns": "http://ibm.github.io/compliance-trestle/schemas/oscal/ar/osco",
            "value": "0.1.57",
            "class": ”scc_goal_version”  },
          {
            "name": "result",
            "ns": "http://ibm.github.io/compliance-trestle/schemas/oscal/ar/osco",
            "value": "notselected",
            "class": "scc_result”  },
          {
            "name": "time",
            "ns": "http://ibm.github.io/compliance-trestle/schemas/oscal/ar/osco",
            "value": "2021-06-14T01:00:29+00:00",
            "class": "scc_timestamp”   },
           …
        ],
        "subjects": [
          {
            "uuid-ref": "bf35822f-ee26-4a5f-96e3-106dc357f1da",
            "type": "inventory-item"
          }
        ],
        "collected": "2021-06-14T18:35:04.000+00:00"
      },
```

Example: Tanium as PVP

https://docs.tanium.com/client/client/overview.html
The Tanium Client is a service installed on endpoint computers that discovers and reports data from those endpoints.

- Hardware and software inventory
- Software configuration
- Local or domain user details
- Installed application or services, startup programs, and running processes
- Existence of Windows registry keys and values
- Windows Management Instrumentation (WMI) data elements
- File system details, including identification of files by hash or contents
- Event log results
- Network configuration settings and state

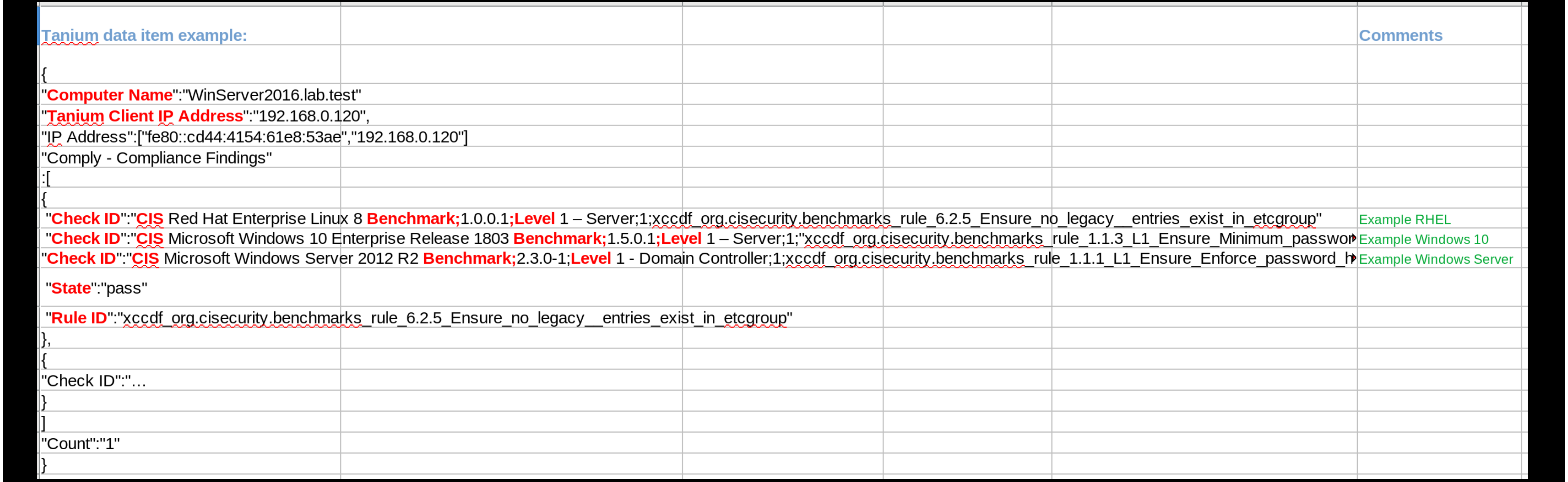
Tanium Integration



https://cloud.ibm.com/docs/security-compliance?topic=security-compliance-setup-tanium&interface=ui

Mapping Tanium data to OSCAL: Native Tanium format

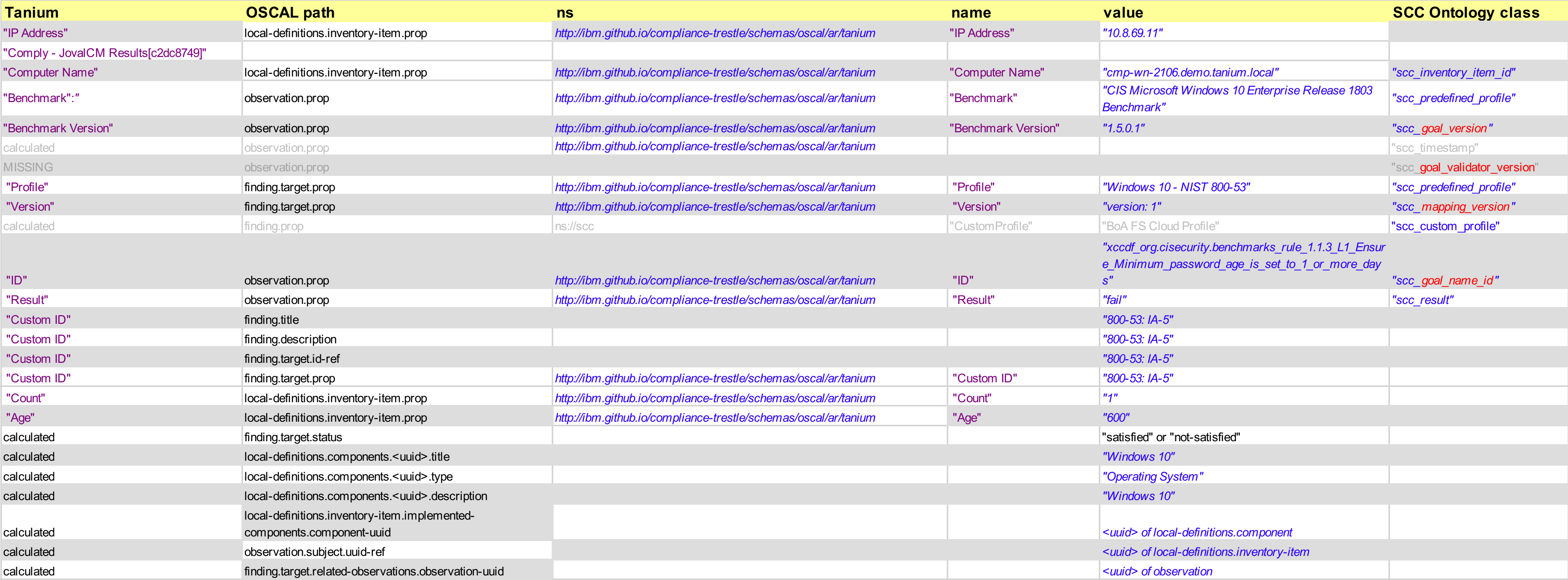
- Native format of different tools need to mapped to OSCAL Results format
- Tanium data item example: one check result in an array of many, to be represented in standardized format as OSCAL System Assessment Results.



Tanium data item example:	Comments
{	
"**Computer Name**":"WinServer2016.lab.test"	
"**Tanium Client IP Address**":"192.168.0.120",	
"IP Address":["fe80::cd44:4154:61e8:53ae","192.168.0.120"]	
"Comply - Compliance Findings"	
:[	
{	
"**Check ID**":"**CIS** Red Hat Enterprise Linux 8 **Benchmark;**1.0.0.1**;Level** 1 – Server;1;xccdf_org.cisecurity.benchmarks_rule_6.2.5_Ensure_no_legacy__entries_exist_in_etcgroup"	Example RHEL
"**Check ID**":"**CIS** Microsoft Windows 10 Enterprise Release 1803 **Benchmark;**1.5.0.1**;Level** 1 – Server;1;"xccdf_org.cisecurity.benchmarks_rule_1.1.3_L1_Ensure_Minimum_passwor	Example Windows 10
"**Check ID**":"**CIS** Microsoft Windows Server 2012 R2 **Benchmark;**2.3.0-1;**Level** 1 - Domain Controller;1;xccdf_org.cisecurity.benchmarks_rule_1.1.1_L1_Ensure_Enforce_password_h	Example Windows Server
"**State**":"pass"	
"**Rule ID**":"xccdf_org.cisecurity.benchmarks_rule_6.2.5_Ensure_no_legacy__entries_exist_in_etcgroup"	
},	
{	
"Check ID":"…	
}	
]	
"Count":"1"	
}	

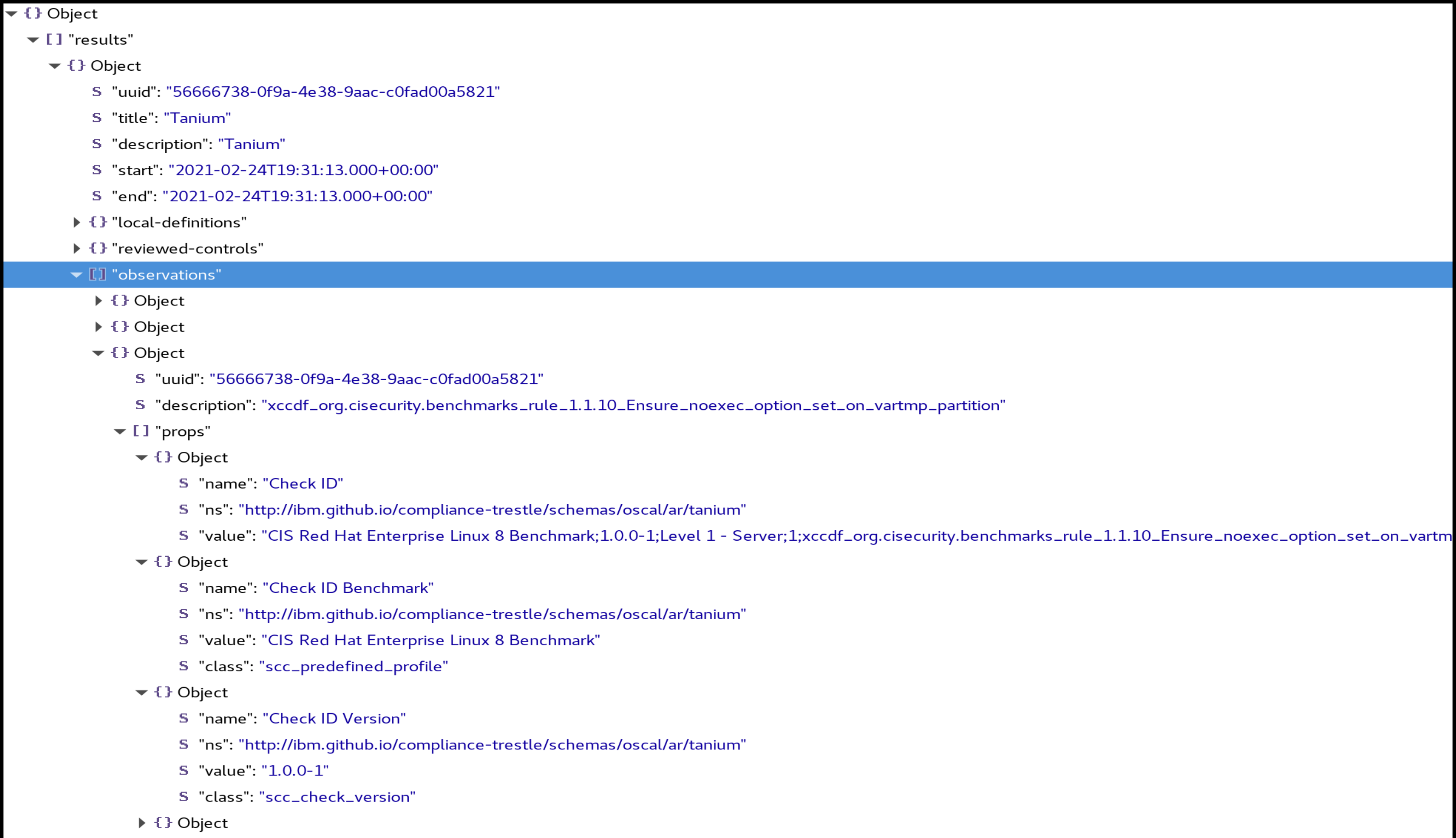
Mapping Tanium data to OSCAL:

- Specific property “classes” for SCC
  - Helps SCC identify the required property that it needs to store in SCC DB
  - Used as reference document for coding Tanium-to-OSCAL trestle based transformer



Tanium	OSCAL path	ns	name	value	SCC Ontology class
"IP Address"	local-definitions.inventory-item.prop	http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium	"IP Address"	"10.8.69.11"	
"Comply - JovalCM Results[c2dc8749]"					
"Computer Name"	local-definitions.inventory-item.prop	http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium	"Computer Name"	"cmp-wn-2106.demo.tanium.local"	"scc_inventory_item_id"
"Benchmark":"	observation.prop	http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium	"Benchmark"	"CIS Microsoft Windows 10 Enterprise Release 1803 Benchmark"	"scc_predefined_profile"
"Benchmark Version"	observation.prop	http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium	"Benchmark Version"	"1.5.0.1"	"scc_goal_version"
calculated	observation.prop	http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium			"scc_timestamp"
MISSING	observation.prop				"scc_goal_validator_version"
"Profile"	finding.target.prop	http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium	"Profile"	"Windows 10 - NIST 800-53"	"scc_predefined_profile"
"Version"	finding.target.prop	http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium	"Version"	"version: 1"	"scc_mapping_version"
calculated	finding.prop	ns://scc	"CustomProfile"	"BoA FS Cloud Profile"	"scc_custom_profile"
"ID"	observation.prop	http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium	"ID"	"xccdf_org.cisecurity.benchmarks_rule_1.1.3_L1_Ensure_Minimum_password_age_is_set_to_1_or_more_days"	"scc_goal_name_id"
"Result"	observation.prop	http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium	"Result"	"fail"	"scc_result"
"Custom ID"	finding.title			"800-53: IA-5"	
"Custom ID"	finding.description			"800-53: IA-5"	
"Custom ID"	finding.target.id-ref			"800-53: IA-5"	
"Custom ID"	finding.target.prop	http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium	"Custom ID"	"800-53: IA-5"	
"Count"	local-definitions.inventory-item.prop	http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium	"Count"	"1"	
"Age"	local-definitions.inventory-item.prop	http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium	"Age"	"600"	
calculated	finding.target.status			"satisfied" or "not-satisfied"	
calculated	local-definitions.components.<uuid>.title			"Windows 10"	
calculated	local-definitions.components.<uuid>.type			"Operating System"	
calculated	local-definitions.components.<uuid>.description			"Windows 10"	
calculated	local-definitions.inventory-item.implemented-components.component-uuid			<uuid> of local-definitions.component	
calculated	observation.subject.uuid-ref			<uuid> of local-definitions.inventory-item	
calculated	finding.target.related-observations.observation-uuid			<uuid> of observation	

Example OSCAL System Assessment Results (SAR) snippet:

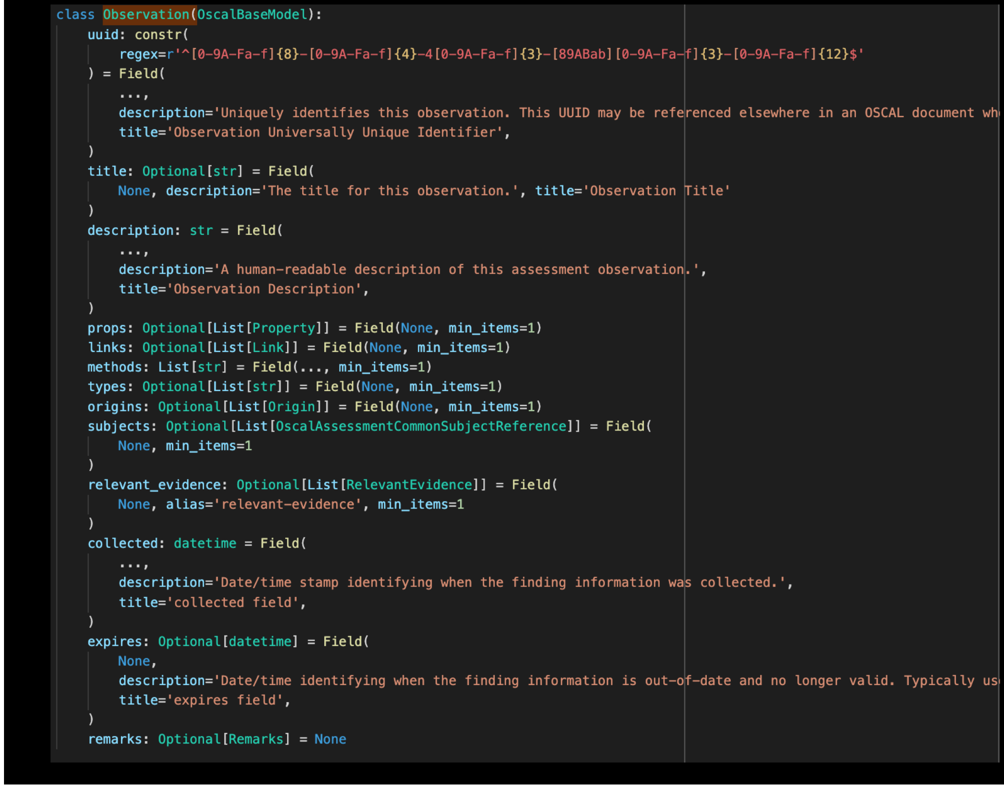


```
▼ {} Object
  ▼ [] "results"
    ▼ {} Object
        S "uuid": "56666738-0f9a-4e38-9aac-c0fad00a5821"
        S "title": "Tanium"
        S "description": "Tanium"
        S "start": "2021-02-24T19:31:13.000+00:00"
        S "end": "2021-02-24T19:31:13.000+00:00"
      ▶ {} "local-definitions"
      ▶ {} "reviewed-controls"
      ▼ [] "observations"
        ▶ {} Object
        ▶ {} Object
        ▼ {} Object
            S "uuid": "56666738-0f9a-4e38-9aac-c0fad00a5821"
            S "description": "xccdf_org.cisecurity.benchmarks_rule_1.1.10_Ensure_noexec_option_set_on_vartmp_partition"
          ▼ [] "props"
            ▼ {} Object
                S "name": "Check ID"
                S "ns": "http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium"
                S "value": "CIS Red Hat Enterprise Linux 8 Benchmark;1.0.0-1;Level 1 - Server;1;xccdf_org.cisecurity.benchmarks_rule_1.1.10_Ensure_noexec_option_set_on_vartm
            ▼ {} Object
                S "name": "Check ID Benchmark"
                S "ns": "http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium"
                S "value": "CIS Red Hat Enterprise Linux 8 Benchmark"
                S "class": "scc_predefined_profile"
            ▼ {} Object
                S "name": "Check ID Version"
                S "ns": "http://ibm.github.io/compliance-trestle/schemas/oscal/ar/tanium"
                S "value": "1.0.0-1"
                S "class": "scc_check_version"
            ▶ {} Object
```

Trestle

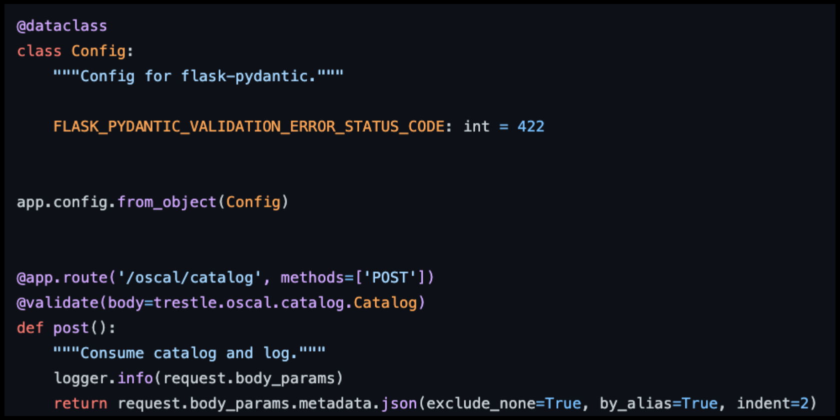
An opinionated, open-source tool to allow editing and automation workflows of NIST OSCAL documents by managing compliance as code

Trestle builds an object model that strongly enforces constraints



```
class Observation(OscalBaseModel):
    uuid: constr(
        regex=r'^[0-9A-Fa-f]{8}-[0-9A-Fa-f]{4}-4[0-9A-Fa-f]{3}-[89ABab][0-9A-Fa-f]{3}-[0-9A-Fa-f]{12}$'
    ) = Field(
        ...,
        description='Uniquely identifies this observation. This UUID may be referenced elsewhere in an OSCAL document wh
        title='Observation Universally Unique Identifier',
    )
    title: Optional[str] = Field(
        None, description='The title for this observation.', title='Observation Title'
    )
    description: str = Field(
        ...,
        description='A human-readable description of this assessment observation.',
        title='Observation Description',
    )
    props: Optional[List[Property]] = Field(None, min_items=1)
    links: Optional[List[Link]] = Field(None, min_items=1)
    methods: List[str] = Field(..., min_items=1)
    types: Optional[List[str]] = Field(None, min_items=1)
    origins: Optional[List[Origin]] = Field(None, min_items=1)
    subjects: Optional[List[OscalAssessmentCommonSubjectReference]] = Field(
        None, min_items=1
    )
    relevant_evidence: Optional[List[RelevantEvidence]] = Field(
        None, alias='relevant-evidence', min_items=1
    )
    collected: datetime = Field(
        ...,
        description='Date/time stamp identifying when the finding information was collected.',
        title='collected field',
    )
    expires: Optional[datetime] = Field(
        None,
        description='Date/time identifying when the finding information is out-of-date and no longer valid. Typically us
        title='expires field',
    )
    remarks: Optional[Remarks] = None
```

Object model integrates with flask to provide IO validation with minimal developer effort

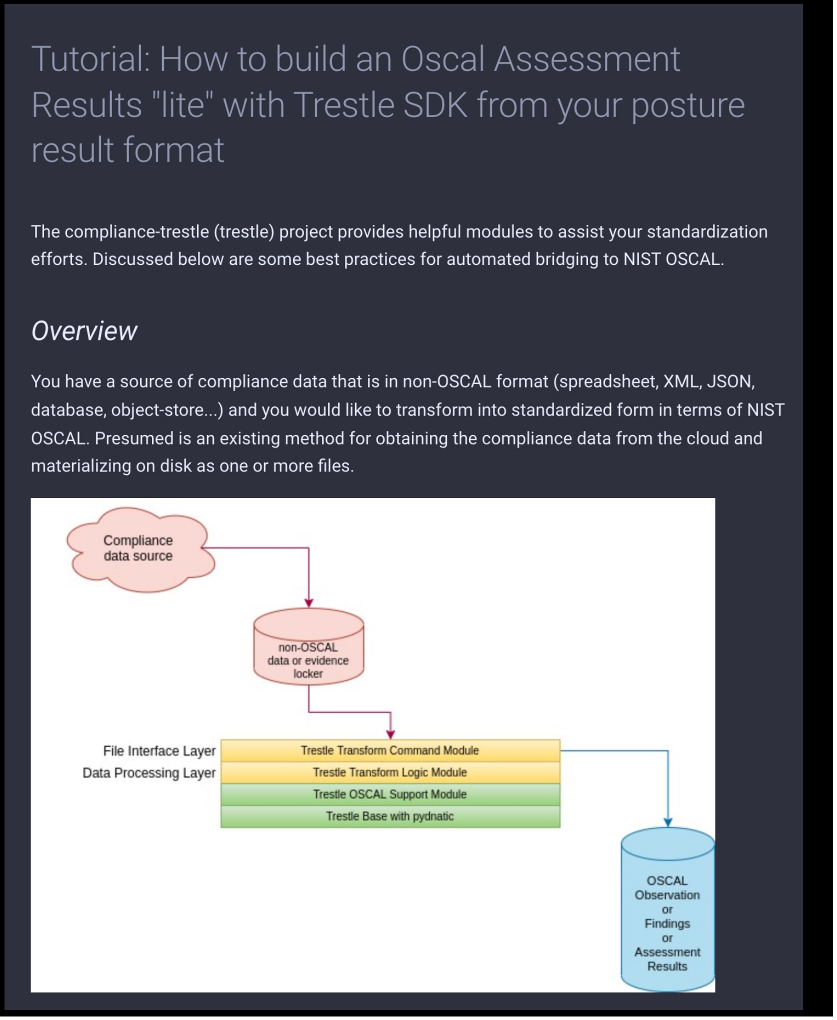


```
@dataclass
class Config:
    """Config for flask-pydantic."""

    FLASK_PYDANTIC_VALIDATION_ERROR_STATUS_CODE: int = 422


app.config.from_object(Config)


@app.route('/oscal/catalog', methods=['POST'])
@validate(body=trestle.oscal.catalog.Catalog)
def post():
    """Consume catalog and log."""
    logger.info(request.body_params)
    return request.body_params.metadata.json(exclude_none=True, by_alias=True, indent=2)
```

Transforming 3rd party content to OSCAL

trestle as an SDK to safely and automatically create OSCAL artifacts

- In order for OSCAL to provide value a set of converters are required from various formats.
- Trestle contains transformers which can be used both as an SDK and a CLI (e.g.: https://ibm.github.io/compliance-trestle/tutorials/task.tanuim-to-oscal/transformation/)
- Trestle conversion SDK is the basis for 3rd party conversions into IBM 'Security and Compliance Centre'
- Trestle OSCAL object model can easily be used to convert content:
  - Excel files: https://github.com/IBM/compliance-trestle-demos/tree/develop/CIS_controls
  - XML content: https://github.com/IBM/compliance-trestle-demos/tree/develop/ISM_catalog_profile



Tutorial: How to build an Oscal Assessment Results "lite" with Trestle SDK from your posture result format

The compliance-trestle (trestle) project provides helpful modules to assist your standardization efforts. Discussed below are some best practices for automated bridging to NIST OSCAL.

Overview

You have a source of compliance data that is in non-OSCAL format (spreadsheet, XML, JSON, database, object-store...) and you would like to transform into standardized form in terms of NIST OSCAL. Presumed is an existing method for obtaining the compliance data from the cloud and materializing on disk as one or more files.

Compliance data source
non-OSCAL data or evidence locker
File Interface Layer
Data Processing Layer
Trestle Transform Command Module
Trestle Transform Logic Module
Trestle OSCAL Support Module
Trestle Base with pydnatic
OSCAL Observation or Findings or Assessment Results

Backup

Vendor products/services compliance definition registration

- Product vendor / service provider needs a standardized way to specify compliance definitions

```
"component-definition": {
   "metadata": {
     "title": "Component definition for OCP4 profiles",
     "props": [
       {
          "name": "profile_name",
          "ns": "http://ibm.github.io/compliance-trestle/schemas/oscal/ibm-
cloud",
          "value": "OCP4 CIS-benchmark v4",
          "class": "scc_profile_name"
       } …]
  "components": {
     "dc417056-3ef7-42c5-b048-47ada2cedb69": {
       "type": "Service",
       "title": ”OSCO",
       “control-implementations”: [
         {
           “uuid”: “bbec83e4-d190-4aa0-86bb-67973e17207c”,
           “source”:
“https://github.com/ComplianceAsCode/content/blob/master/products/ocp4/
profiles/cis-node.profile”,
```

```
          “implemented-requirements”: [
           {
             “uuid”: “4c97bf1d-e1fa-49c9-8750-6de6be7e7ae6",
             “control-id”: “CIS-1.3.1”,
             "description": "Ensure that garbage collection is configured as
appropriate"
               “props”: [
               {
                 "name": "XCCDF_rule",
                 "ns":
"https://github.com/ComplianceAsCode/content/tree/master/ocp4",
                 "value":
"xccdf_org.ssgproject.content_rule_kubelet_eviction_thresholds_set_soft_me
mory_available",
                 "class": "scc_goal_name_id",
                 "remarks": "Ensure that garbage collection is configured as
appropriate"
               } ….]
             “set-parameters” " : [
              {
                 "param-id":
"kubelet_eviction_thresholds_set_soft_memory_available",
                 "values":  [ "500Mi” ],
                 "remarks": "Memory Available for the EvictionSoft threshold to
trigger."
               } …]
```