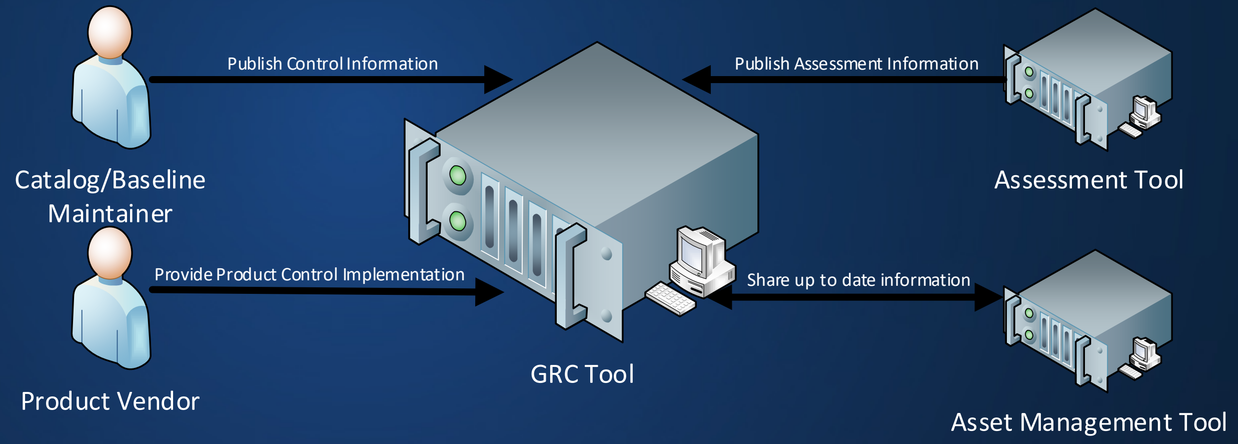
OSCAL Catalog, Profile, and Implementation Layers

Catalog, Profile, Component Definition, and System Security Plan Models

National Institute of
Standards and Technology
U.S. Department of Commerce

What is OSCAL?

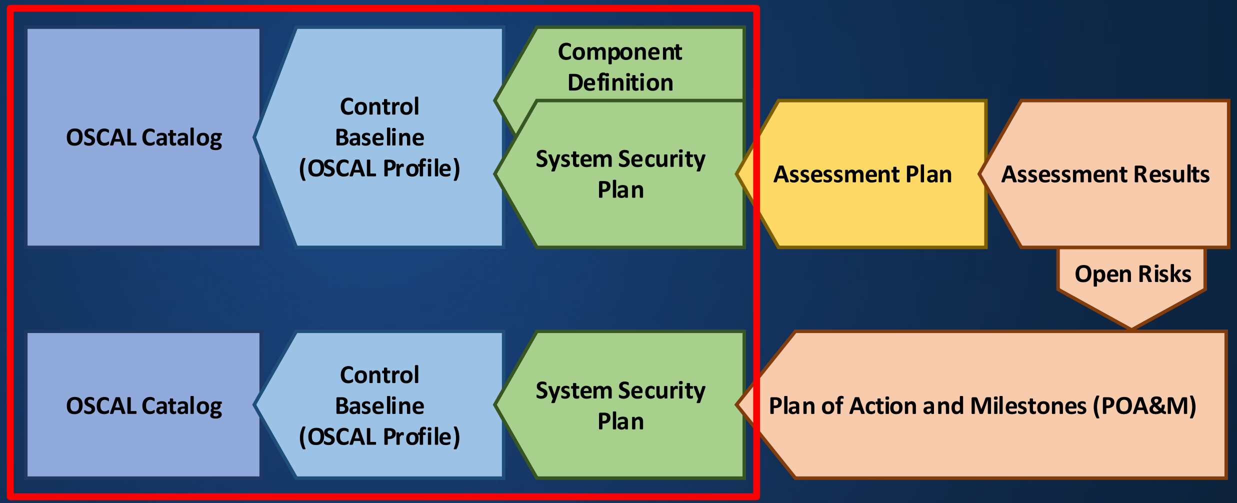

Publish Control Information
Publish Assessment Information
Catalog/Baseline Maintainer
Assessment Tool
Provide Product Control Implementation
Share up to date information
Product Vendor
GRC Tool
Asset Management Tool

OSCAL is not a tool, but it enables tools to share data

OSCAL provides standardized data formats for exchanging control, control implementation, and control assessment information between tools

- ➢ Catalog and baseline information can be easily imported into a tool
- ➢ Product and system control implementation information can be shared
- ➢ Assessors can generate assessment results to share
- ➢ Assessment tools can produce data to import into other tools

The OSCAL Models



OSCAL provides 7 models:

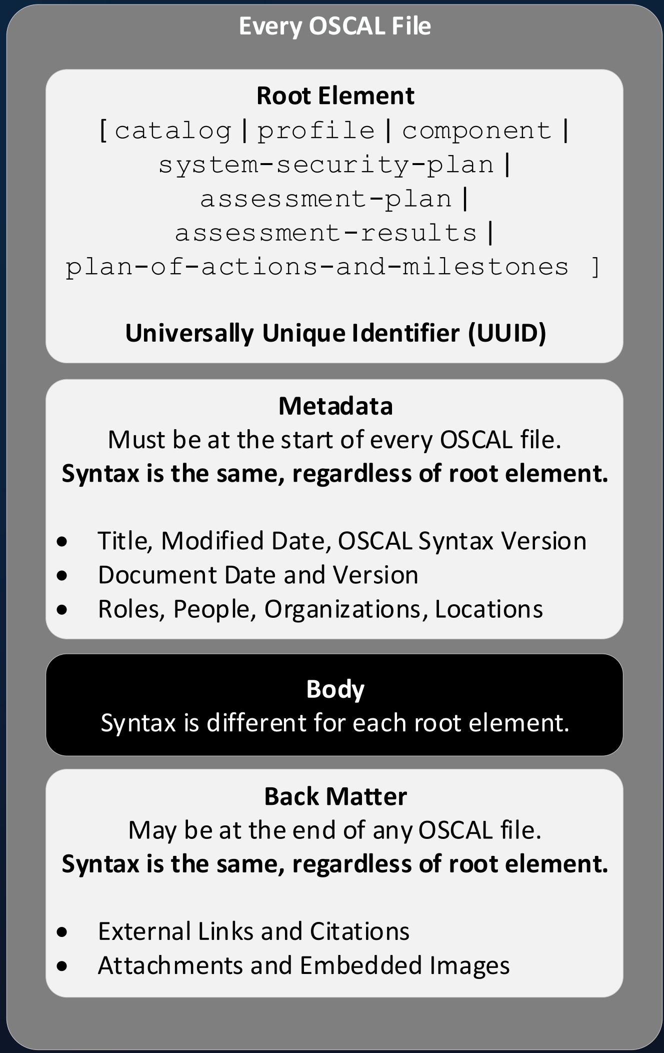
- Offered in XML-, JSON-, and YAML-based formats
- Supports a control-based risk management approach to system security
- Each model build on the models to the left (in the diagram above)

The OSCAL models provide for:

- Improved accuracy and document quality
- Reduced labor costs
- Easy machine-to-machine exchange
- Leverageable, standardized identifiers providing the foundation for assessment automation

Common OSCAL Structure

- **Root Element:** Indicates the model of the data
- **Root UUID:** A RFC 4122 Version 4 Universally Unique Identifier (UUID) that identifies the specific document instance. Changed when the document is modified.
- **Metadata:** Information about the document (i.e., title, last-modified timestamp, OSCAL version). Also used to define roles, parties (people, teams and organizations), and locations referenced in the document.
- **Model-specific Body:** The body is specific to each model.
- **Back Matter:** Used to link to and attach resources, which may contain citations. Used to associate graphics, supporting documentation, etc. with the OSCAL document. A reference entry here can be referenced from within the body of an OSCAL document.



Every OSCAL File

Root Element

`[ catalog | profile | component | system-security-plan | assessment-plan | assessment-results | plan-of-actions-and-milestones ]`

Universally Unique Identifier (UUID)

Metadata

Must be at the start of every OSCAL file.

Syntax is the same, regardless of root element.

- Title, Modified Date, OSCAL Syntax Version
- Document Date and Version
- Roles, People, Organizations, Locations

Body

Syntax is different for each root element.

Back Matter

May be at the end of any OSCAL file.

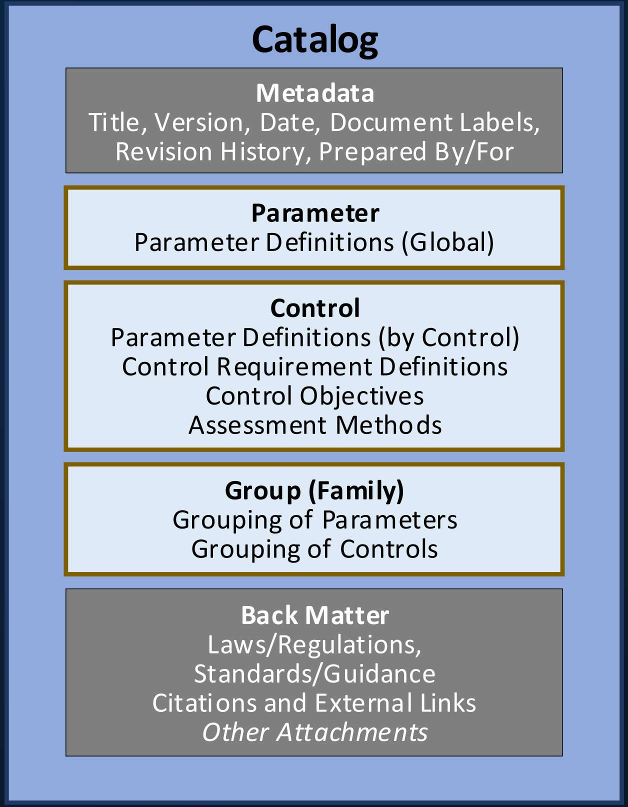
Syntax is the same, regardless of root element.

- External Links and Citations
- Attachments and Embedded Images

OSCAL Catalog Model

Represents a collection of security and privacy controls, which may be used as part of a risk management program.

- **Metadata:** Same for each OSCAL model
- **Parameter:** Provides a global policy variable used by one or more control
- **Control:** An individual control in the catalog.
  - May contain control-specific parameters, control requirement statements, control objectives, assessment methods, references
  - Controls can have child controls.
- **Group:** Related controls may be grouped. Parameters related to this group may be defined here.
- **Back Matter:** Same for each OSCAL model



Catalog

Metadata
Title, Version, Date, Document Labels, Revision History, Prepared By/For

Parameter
Parameter Definitions (Global)

Control
Parameter Definitions (by Control)
Control Requirement Definitions
Control Objectives
Assessment Methods

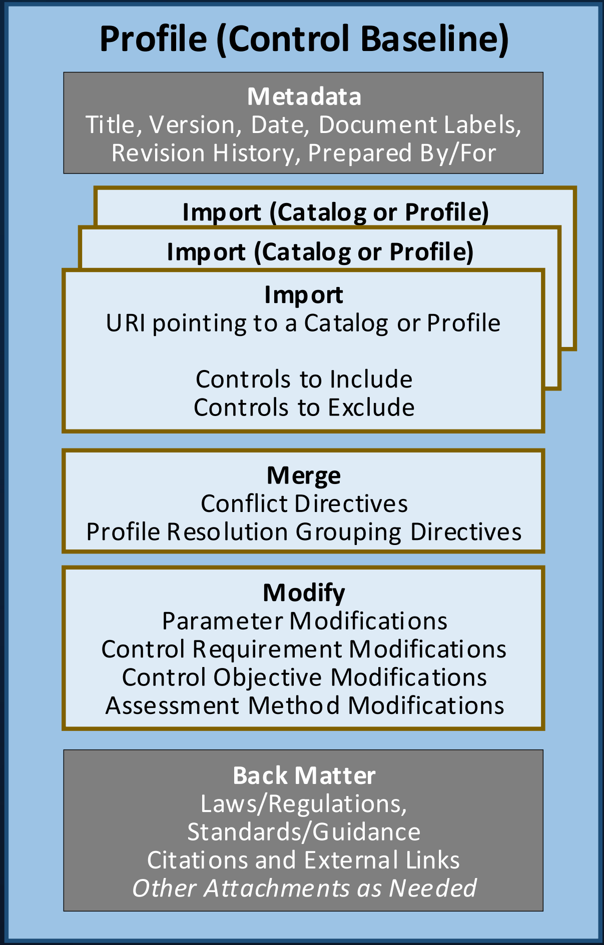
Group (Family)
Grouping of Parameters
Grouping of Controls

Back Matter
Laws/Regulations, Standards/Guidance
Citations and External Links
Other Attachments

OSCAL Profile Model

Used to establish a baseline of controls to be implemented with a system.

- **Metadata:** Same for each OSCAL model
- **Import:** Identifies an OSCAL catalog or other profile to import controls from
  - A control must be imported to be included in a baseline.
  - All parameters and back-matter resources cited by an imported control are also imported.
- **Merge:** Provides directives used to organize controls and to resolve conflicts when the same control is imported multiple times
- **Modify:** Allows tailoring of imported controls, including their parameters, control requirement definitions, references, control objectives, and assessment actions.
- **Back Matter:** Same for each OSCAL model

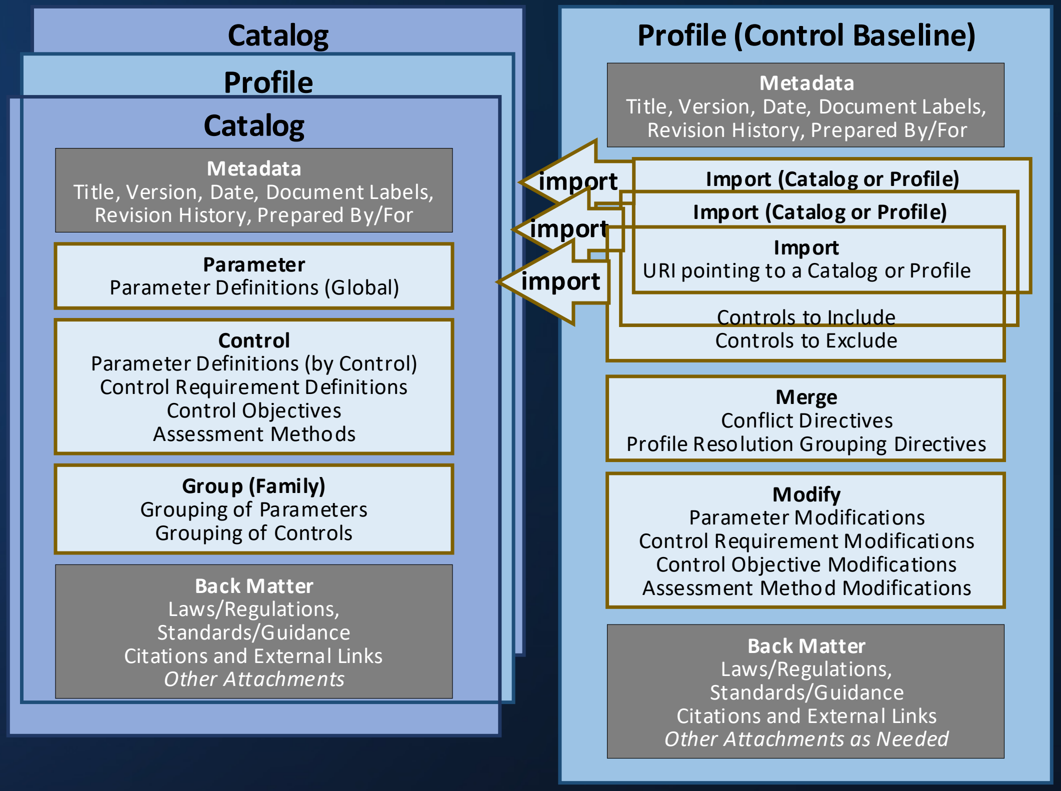

Profile (Control Baseline)
Metadata
Title, Version, Date, Document Labels,
Revision History, Prepared By/For
Import (Catalog or Profile)
Import (Catalog or Profile)
Import
URI pointing to a Catalog or Profile
Controls to Include
Controls to Exclude
Merge
Conflict Directives
Profile Resolution Grouping Directives
Modify
Parameter Modifications
Control Requirement Modifications
Control Objective Modifications
Assessment Method Modifications
Back Matter
Laws/Regulations,
Standards/Guidance
Citations and External Links
Other Attachments as Needed

OSCAL Profile Model - Inheritance

A profile can import controls from:

- A catalog or multiple catalogs
- Another profile or multiple profiles

This allows a baseline to be established by customizing another baseline.

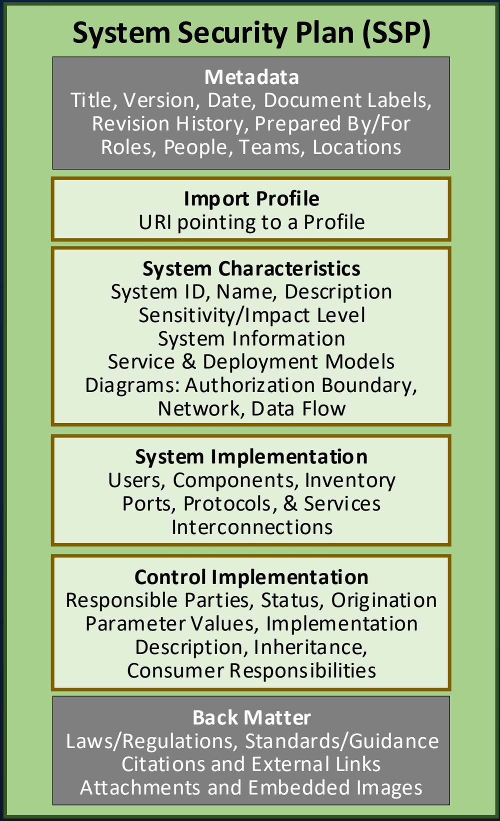


Catalog
Profile
Catalog
Metadata
Title, Version, Date, Document Labels, Revision History, Prepared By/For
Parameter
Parameter Definitions (Global)
Control
Parameter Definitions (by Control)
Control Requirement Definitions
Control Objectives
Assessment Methods
Group (Family)
Grouping of Parameters
Grouping of Controls
Back Matter
Laws/Regulations, Standards/Guidance
Citations and External Links
Other Attachments
import
import
import
Profile (Control Baseline)
Metadata
Title, Version, Date, Document Labels, Revision History, Prepared By/For
Import (Catalog or Profile)
Import (Catalog or Profile)
Import
URI pointing to a Catalog or Profile
Controls to Include
Controls to Exclude
Merge
Conflict Directives
Profile Resolution Grouping Directives
Modify
Parameter Modifications
Control Requirement Modifications
Control Objective Modifications
Assessment Method Modifications
Back Matter
Laws/Regulations, Standards/Guidance
Citations and External Links
Other Attachments as Needed

OSCAL System Security Plan Model

Used to document how controls are implemented for an information system and each component part of an information system.

- **Metadata:** Same for each OSCAL model
- **Import Profile:** Identifies the applicable control baseline for the system as an OSCAL profile.
- **System Characteristics:** Represents attributes of the system, such as its name, description, models, and information processed.
- **System Implementation:** Represents relevant information about the system's deployment, including user roles, interconnections, services, and system inventory.
- **Control Implementation:** Describes how each control in the baseline is implemented within the system.
- **Back Matter:** Same for each OSCAL model



System Security Plan (SSP)

Metadata
Title, Version, Date, Document Labels,
Revision History, Prepared By/For
Roles, People, Teams, Locations

Import Profile
URI pointing to a Profile

System Characteristics
System ID, Name, Description
Sensitivity/Impact Level
System Information
Service & Deployment Models
Diagrams: Authorization Boundary,
Network, Data Flow

System Implementation
Users, Components, Inventory
Ports, Protocols, & Services
Interconnections

Control Implementation
Responsible Parties, Status, Origination
Parameter Values, Implementation
Description, Inheritance,
Consumer Responsibilities

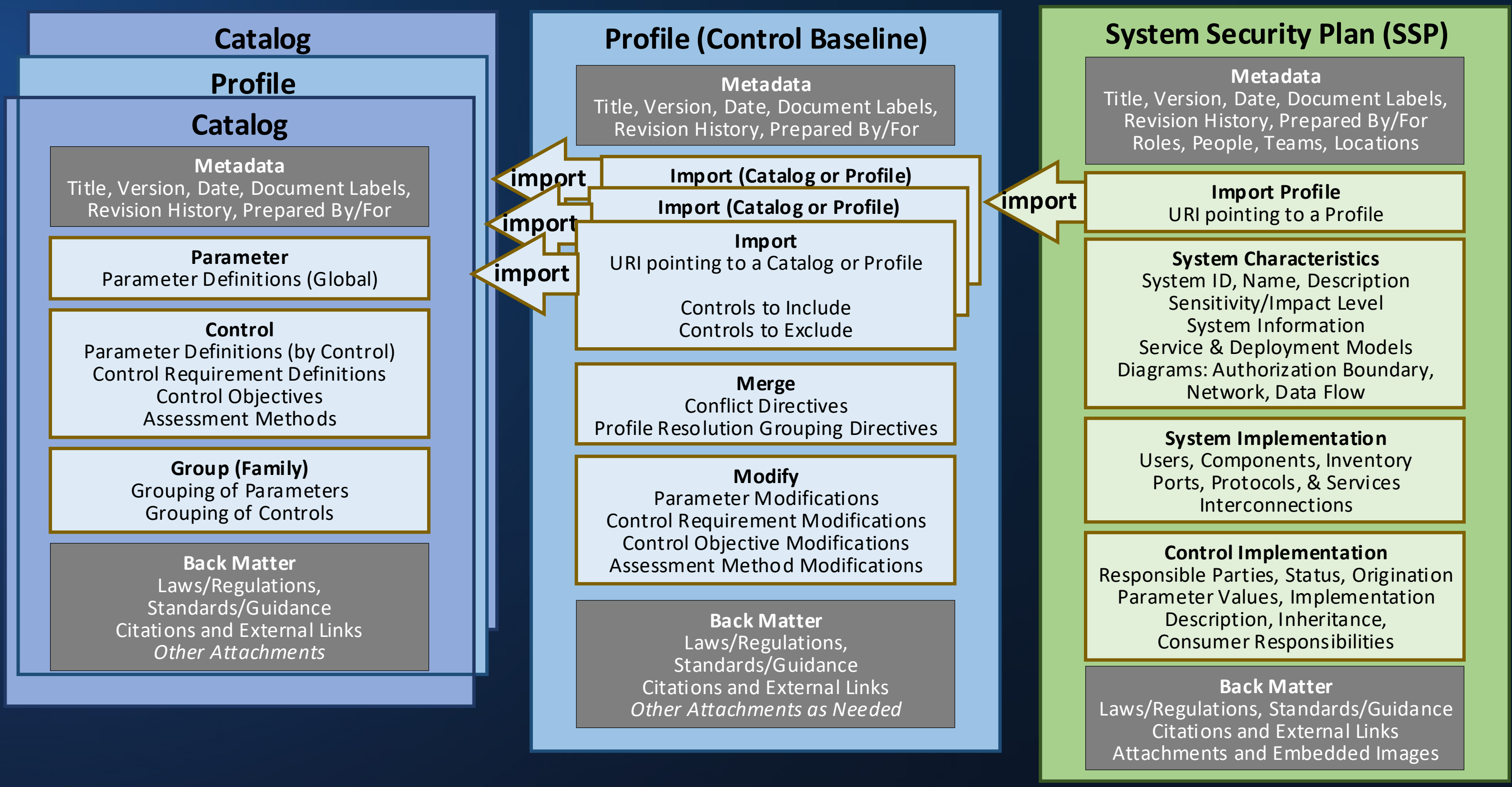
Back Matter
Laws/Regulations, Standards/Guidance
Citations and External Links
Attachments and Embedded Images

OSCAL System Security Plan Model - Inheritance

A system security plan has a single baseline for the system.

- **The baseline is established by an OSCAL Profile**
- **The controls are inherited from the catalog(s) imported by the Profile and any Profile(s) it imports**

This allows a baseline to be reused by multiple systems and for organizations to create custom baselines.



Catalog

Profile

Catalog

Metadata
Title, Version, Date, Document Labels, Revision History, Prepared By/For

Parameter
Parameter Definitions (Global)

Control
Parameter Definitions (by Control)
Control Requirement Definitions
Control Objectives
Assessment Methods

Group (Family)
Grouping of Parameters
Grouping of Controls

Back Matter
Laws/Regulations, Standards/Guidance
Citations and External Links
Other Attachments

import

import

import

Profile (Control Baseline)

Metadata
Title, Version, Date, Document Labels, Revision History, Prepared By/For

Import (Catalog or Profile)

Import (Catalog or Profile)

Import
URI pointing to a Catalog or Profile

Controls to Include
Controls to Exclude

Merge
Conflict Directives
Profile Resolution Grouping Directives

Modify
Parameter Modifications
Control Requirement Modifications
Control Objective Modifications
Assessment Method Modifications

Back Matter
Laws/Regulations, Standards/Guidance
Citations and External Links
Other Attachments as Needed

import

System Security Plan (SSP)

Metadata
Title, Version, Date, Document Labels, Revision History, Prepared By/For
Roles, People, Teams, Locations

Import Profile
URI pointing to a Profile

System Characteristics
System ID, Name, Description
Sensitivity/Impact Level
System Information
Service & Deployment Models
Diagrams: Authorization Boundary, Network, Data Flow

System Implementation
Users, Components, Inventory
Ports, Protocols, & Services
Interconnections

Control Implementation
Responsible Parties, Status, Origination
Parameter Values, Implementation Description, Inheritance,
Consumer Responsibilities

Back Matter
Laws/Regulations, Standards/Guidance
Citations and External Links
Attachments and Embedded Images

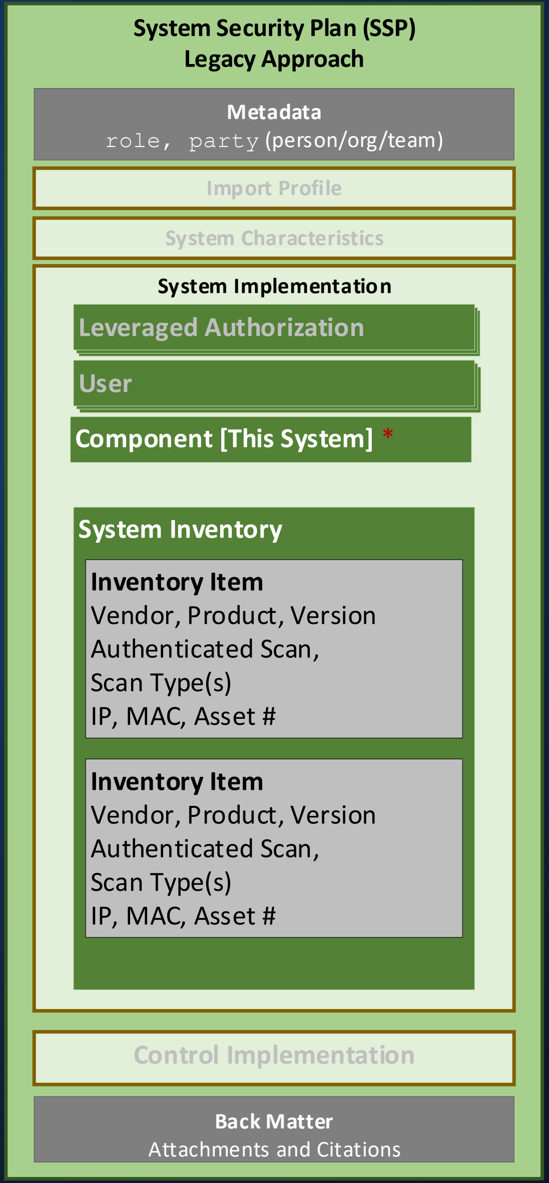
OSCAL System Security Plan Model - Inventory

The assets that compose a system are defined by the "system implementation".

The system inventory can be compositional.

- **Components are used to describe individual system parts**
- **Components are associated with individual inventory items**

This allows the parts of a system to be individually identified.



System Security Plan (SSP)
Legacy Approach

Metadata
`role, party` (person/org/team)

Import Profile

System Characteristics

System Implementation

Leveraged Authorization

User

Component [This System] *

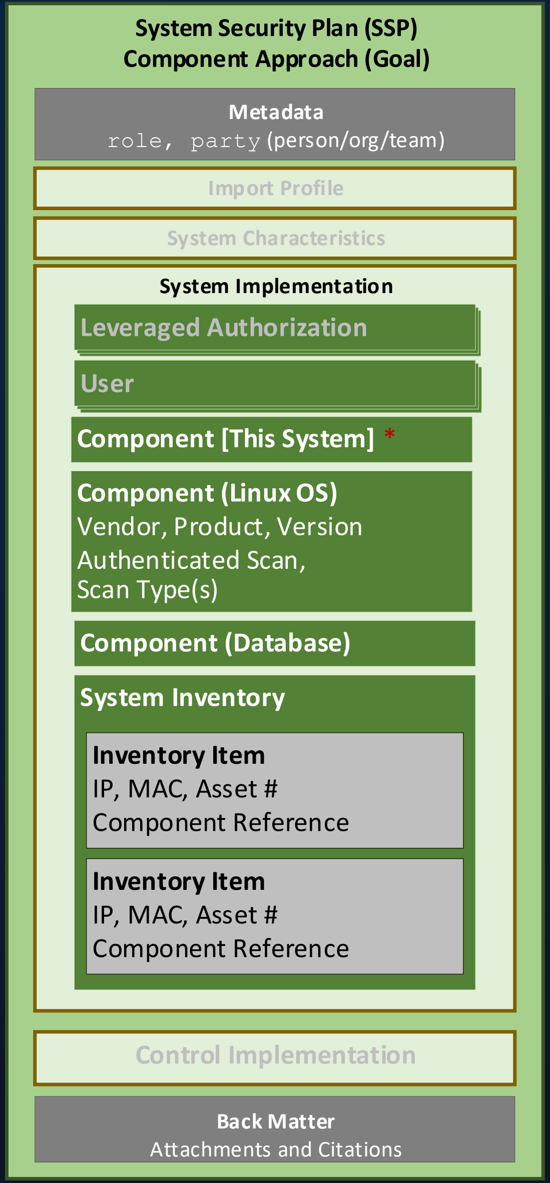
System Inventory

Inventory Item
Vendor, Product, Version
Authenticated Scan,
Scan Type(s)
IP, MAC, Asset #

Inventory Item
Vendor, Product, Version
Authenticated Scan,
Scan Type(s)
IP, MAC, Asset #

Control Implementation

Back Matter
Attachments and Citations



System Security Plan (SSP)
Component Approach (Goal)

Metadata
`role, party` (person/org/team)

Import Profile

System Characteristics

System Implementation

Leveraged Authorization

User

Component [This System] *

Component (Linux OS)
Vendor, Product, Version
Authenticated Scan,
Scan Type(s)

Component (Database)

System Inventory

Inventory Item
IP, MAC, Asset #
Component Reference

Inventory Item
IP, MAC, Asset #
Component Reference

Control Implementation

Back Matter
Attachments and Citations

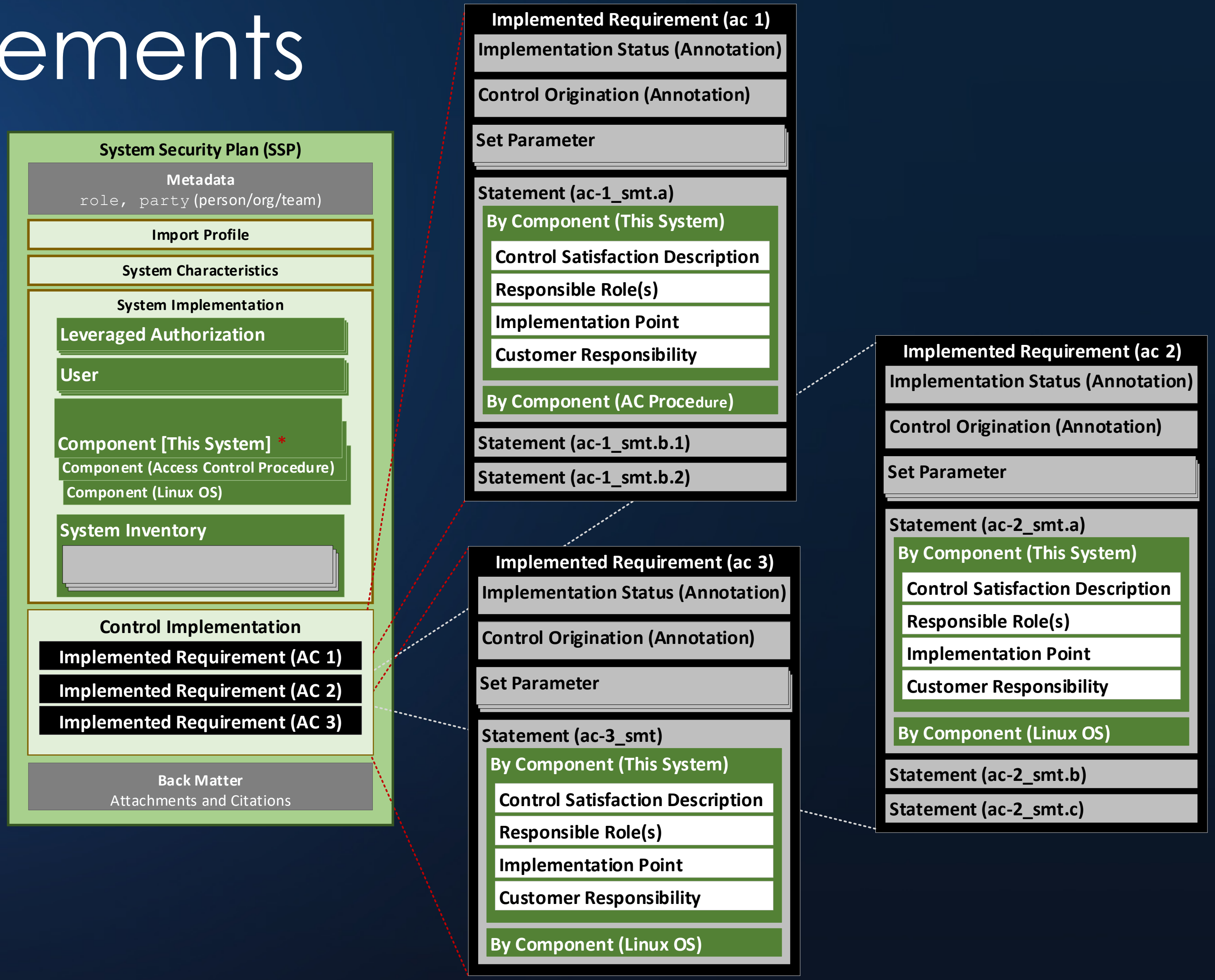
OSCAL System Security Plan Model – Control Statements

Control statements are used to document a control’s implementation in the system.

Statements can be made for:

- ➢ The entire system using “This System”
- ➢ A specific component

This allows a fine-grained definition of the system implementation supporting greater automation and rigor.



System Security Plan (SSP)

Metadata
`role, party` (person/org/team)

Import Profile

System Characteristics

System Implementation

Leveraged Authorization

User

Component [This System] *

Component (Access Control Procedure)

Component (Linux OS)

System Inventory

Control Implementation

Implemented Requirement (AC 1)

Implemented Requirement (AC 2)

Implemented Requirement (AC 3)

Back Matter
Attachments and Citations

Implemented Requirement (ac 1)

- Implementation Status (Annotation)
- Control Origination (Annotation)
- Set Parameter
- Statement (ac-1_smt.a)
  - By Component (This System)
    - Control Satisfaction Description
    - Responsible Role(s)
    - Implementation Point
    - Customer Responsibility
  - By Component (AC Procedure)
- Statement (ac-1_smt.b.1)
- Statement (ac-1_smt.b.2)

Implemented Requirement (ac 2)

- Implementation Status (Annotation)
- Control Origination (Annotation)
- Set Parameter
- Statement (ac-2_smt.a)
  - By Component (This System)
    - Control Satisfaction Description
    - Responsible Role(s)
    - Implementation Point
    - Customer Responsibility
  - By Component (Linux OS)
- Statement (ac-2_smt.b)
- Statement (ac-2_smt.c)

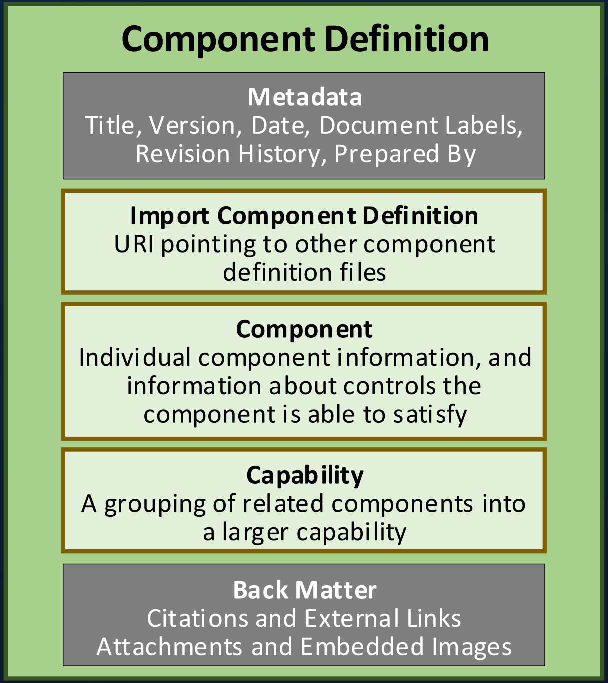
Implemented Requirement (ac 3)

- Implementation Status (Annotation)
- Control Origination (Annotation)
- Set Parameter
- Statement (ac-3_smt)
  - By Component (This System)
    - Control Satisfaction Description
    - Responsible Role(s)
    - Implementation Point
    - Customer Responsibility
  - By Component (Linux OS)

OSCAL Component Definition Model

Used to document how controls are implemented for a given software, hardware, service, policy, process, procedure, or validation (i.e. FIPS 140-2).

- **Metadata:** Same for each OSCAL model
- **Import:** Other component definitions from another resource, from which related information is referenced.
- **Component:** A defined component that can be part of an implemented system.
- **Capability:** A grouping of multiple components or capabilities.
- **Back Matter:** Same for each OSCAL model



Component Definition

Metadata
Title, Version, Date, Document Labels, Revision History, Prepared By

Import Component Definition
URI pointing to other component definition files

Component
Individual component information, and information about controls the component is able to satisfy

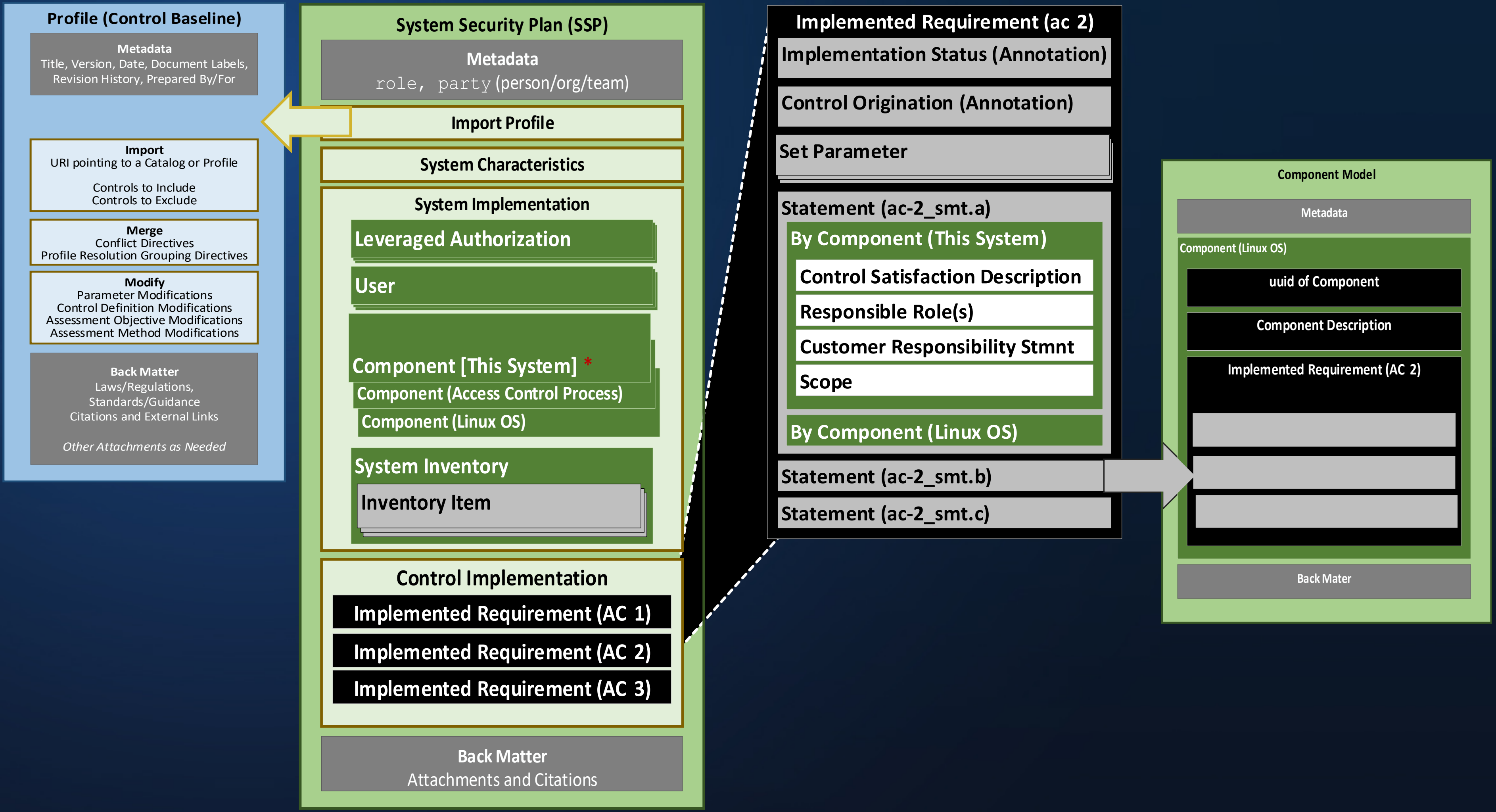
Capability
A grouping of related components into a larger capability

Back Matter
Citations and External Links
Attachments and Embedded Images

OSCAL Component Definition Model – Using with a System Security Plan

Components from a Component Definition can make documenting a system easier.

Implementation statements in the SSP can be populated from the Component Definition



Profile (Control Baseline)

Metadata
Title, Version, Date, Document Labels, Revision History, Prepared By/For

Import
URI pointing to a Catalog or Profile

Controls to Include
Controls to Exclude

Merge
Conflict Directives
Profile Resolution Grouping Directives

Modify
Parameter Modifications
Control Definition Modifications
Assessment Objective Modifications
Assessment Method Modifications

Back Matter
Laws/Regulations,
Standards/Guidance
Citations and External Links

Other Attachments as Needed

System Security Plan (SSP)

Metadata
`role`, `party` (person/org/team)

Import Profile

System Characteristics

System Implementation

Leveraged Authorization

User

Component [This System] *

Component (Access Control Process)

Component (Linux OS)

System Inventory

Inventory Item

Control Implementation

Implemented Requirement (AC 1)

Implemented Requirement (AC 2)

Implemented Requirement (AC 3)

Back Matter
Attachments and Citations

Implemented Requirement (ac 2)

Implementation Status (Annotation)

Control Origination (Annotation)

Set Parameter

Statement (ac-2_smt.a)

By Component (This System)

Control Satisfaction Description

Responsible Role(s)

Customer Responsibility Stmnt

Scope

By Component (Linux OS)

Statement (ac-2_smt.b)

Statement (ac-2_smt.c)

Component Model

Metadata

Component (Linux OS)

uuid of Component

Component Description

Implemented Requirement (AC 2)

Back Mater

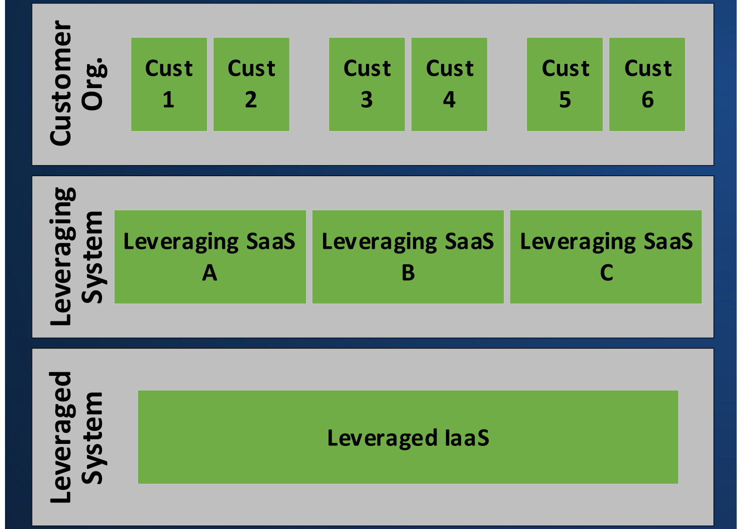
* Every SSP, must have a component representing the whole system.

OSCAL Support for Leveraging Existing ATOs

National Institute of
Standards and Technology
U.S. Department of Commerce

Examples of a Leveraged Authorization?

Yes



Customer Org.: Cust 1, Cust 2, Cust 3, Cust 4, Cust 5, Cust 6

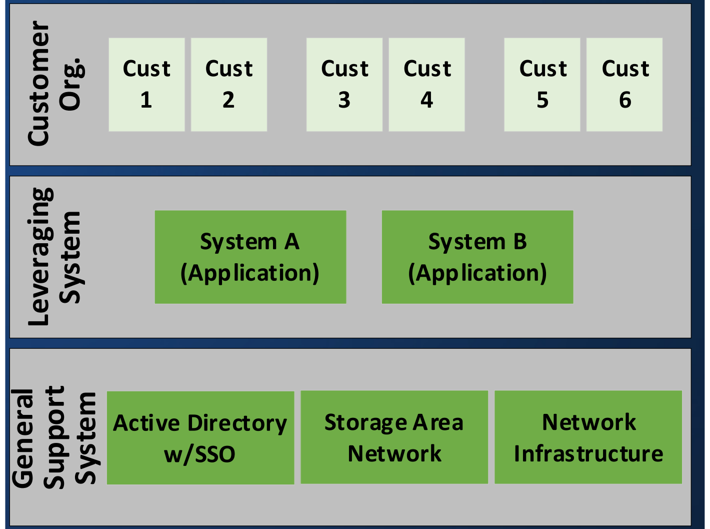
Leveraging System: Leveraging SaaS A, Leveraging SaaS B, Leveraging SaaS C

Leveraged System: Leveraged IaaS

Cloud (SaaS on IaaS)

Cloud: Several SaaS systems running on a separately authorized IaaS.

Yes



Customer Org.: Cust 1, Cust 2, Cust 3, Cust 4, Cust 5, Cust 6

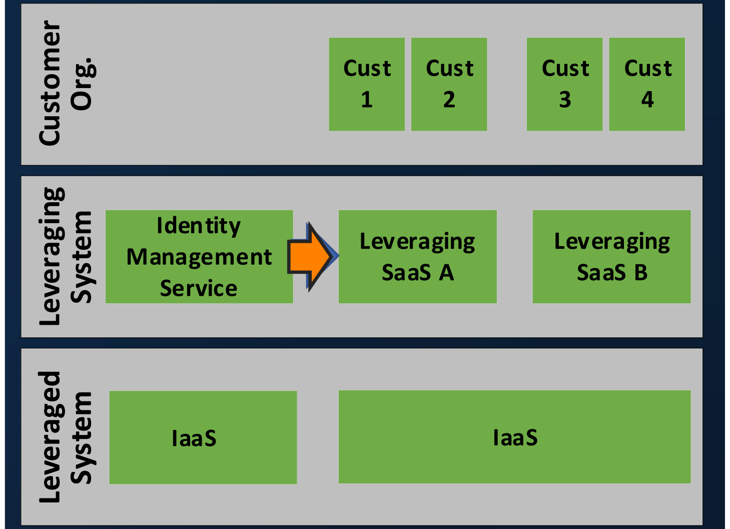
Leveraging System: System A (Application), System B (Application)

General Support System: Active Directory w/SSO, Storage Area Network, Network Infrastructure

Data Center (System on GSS)

Data Center: Several systems relying on a separately authorized storage array or other general support system (GSS)

No



Customer Org.: Cust 1, Cust 2, Cust 3, Cust 4

Leveraging System: Identity Management Service → Leveraging SaaS A, Leveraging SaaS B

Leveraged System: IaaS, IaaS

External Service or Interconnection

Interconnections or External Services are not leveraged authorizations

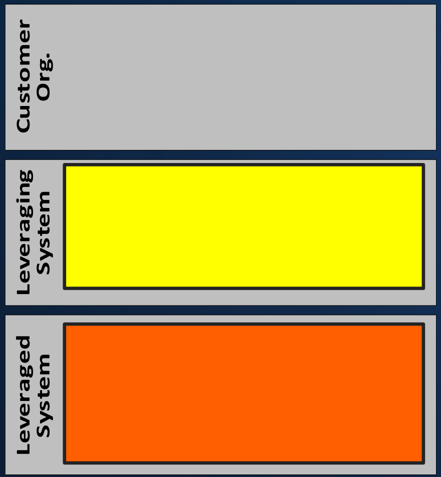
- ➢ Even if they have an authorization
- ➢ SaaS A handles the Identity Management Service as a system component

OSCAL supports this, just not as a L.A.

What is a Leveraged Authorization?

A leveraged authorization exists where:

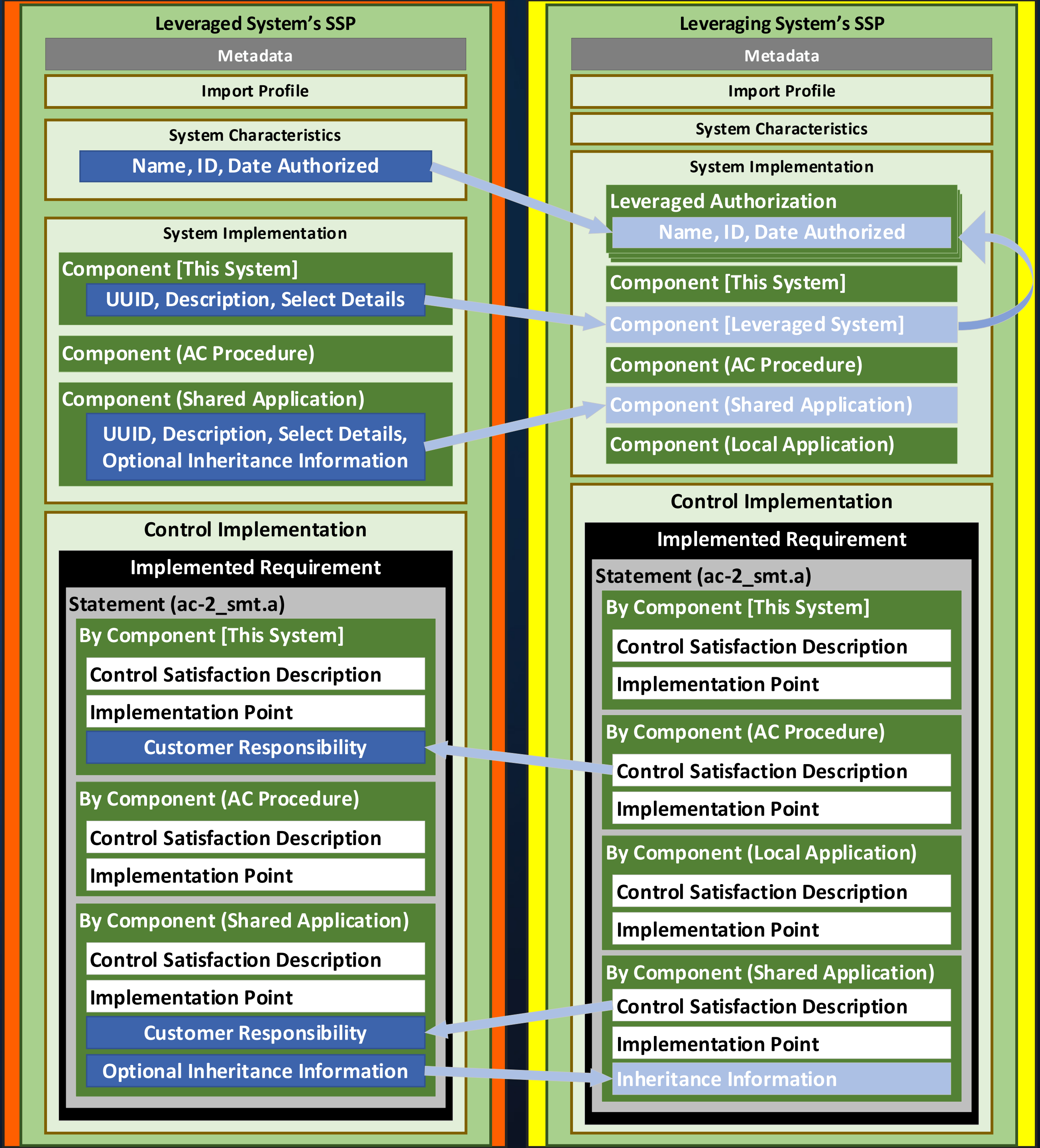
- one or more leveraging systems rely on a leveraged system for operation in a stacked hierarchy; and
- any leveraging system is authorized separately from the leveraged system.



Systems Operating in a Stacked Hierarchy

NOTE:

External services and interconnections are not regarded as leveraged authorizations.



Leveraged System's SSP

- Metadata
- Import Profile
- System Characteristics
  - Name, ID, Date Authorized
- System Implementation
  - Component [This System]
    - UUID, Description, Select Details
  - Component (AC Procedure)
  - Component (Shared Application)
    - UUID, Description, Select Details, Optional Inheritance Information
- Control Implementation
  - Implemented Requirement
    - Statement (ac-2_smt.a)
      - By Component [This System]
        - Control Satisfaction Description
        - Implementation Point
        - Customer Responsibility
      - By Component (AC Procedure)
        - Control Satisfaction Description
        - Implementation Point
      - By Component (Shared Application)
        - Control Satisfaction Description
        - Implementation Point
        - Customer Responsibility
        - Optional Inheritance Information

Leveraging System's SSP

- Metadata
- Import Profile
- System Characteristics
- System Implementation
  - Leveraged Authorization
    - Name, ID, Date Authorized
  - Component [This System]
  - Component [Leveraged System]
  - Component (AC Procedure)
  - Component (Shared Application)
  - Component (Local Application)
- Control Implementation
  - Implemented Requirement
    - Statement (ac-2_smt.a)
      - By Component [This System]
        - Control Satisfaction Description
        - Implementation Point
      - By Component (AC Procedure)
        - Control Satisfaction Description
        - Implementation Point
      - By Component (Local Application)
        - Control Satisfaction Description
        - Implementation Point
      - By Component (Shared Application)
        - Control Satisfaction Description
        - Implementation Point
        - Inheritance Information

Leveraging ATOs - Three Scenarios

Scenario 1: OSCAL SSP / With Access

The leveraged system is using an OSCAL SSP; and the leveraging system is permitted to access it.

No CRM/SSRM is needed.

Preferred approach!

Completed

Scenario 2: OSCAL SSP / No Access

The leveraged system is using an OSCAL SSP; however,

the leveraging system is not permitted to access it.

An OSCAL CRM/SSRM will be used.

Typical FedRAMP Scenario

Post 1.0 Release Candidate

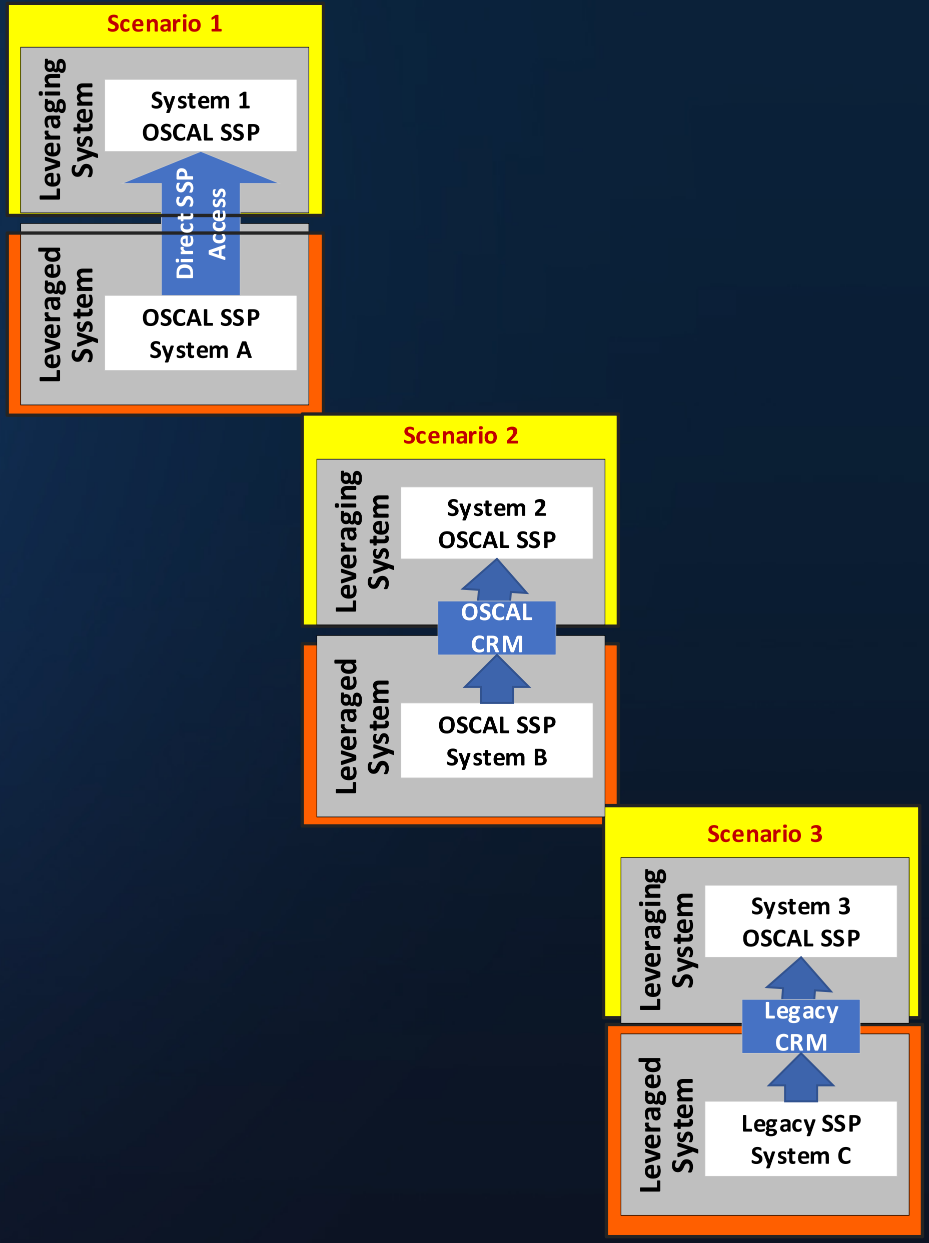
Scenario 3: Legacy SSP

A leveraged system is still using a legacy SSP.

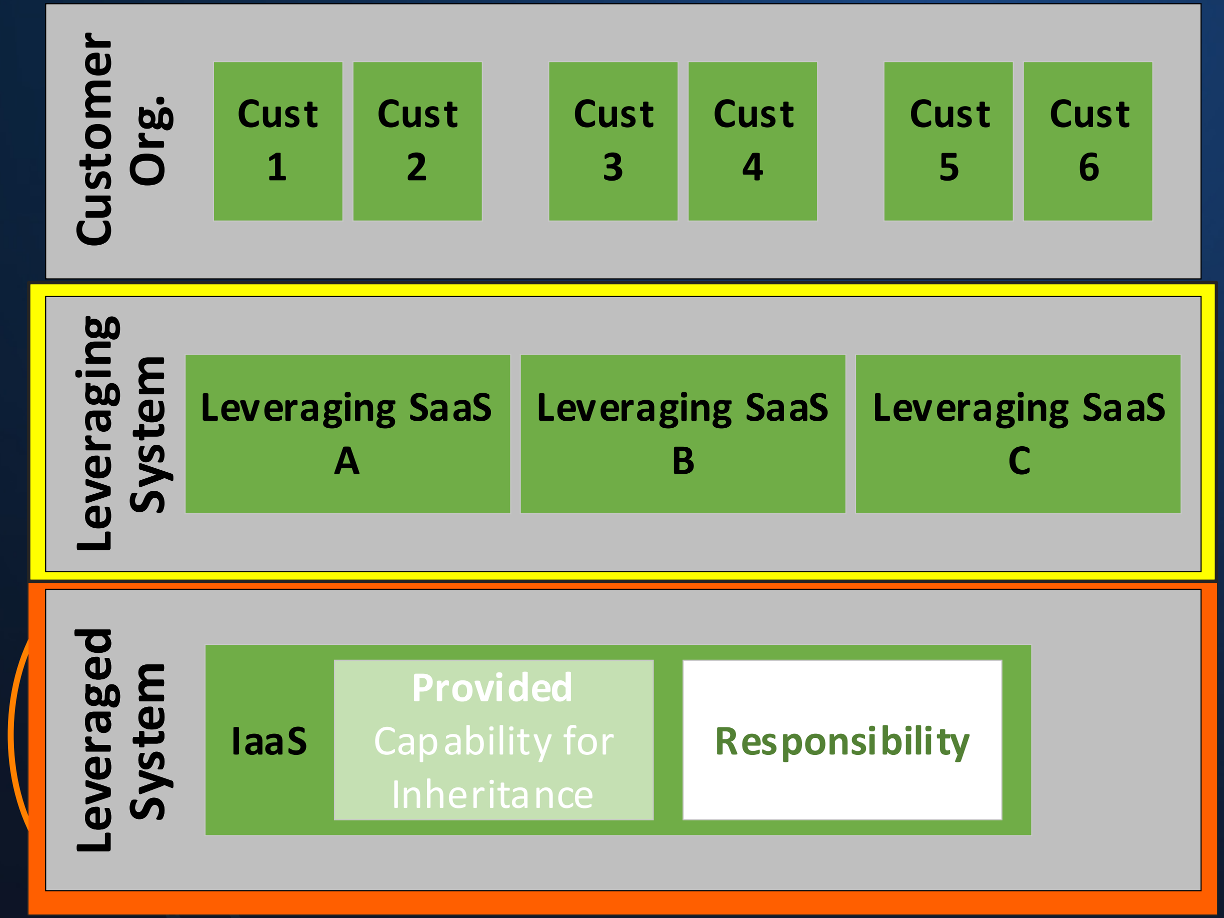
A legacy Customer Responsibility Matrix (CRM) or System Security Responsibility Matrix (SSRM) are used/available.

Transition scenario for an imperfect world

Post 1.0 Release Candidate



Control Documentation (Leveraged System View)

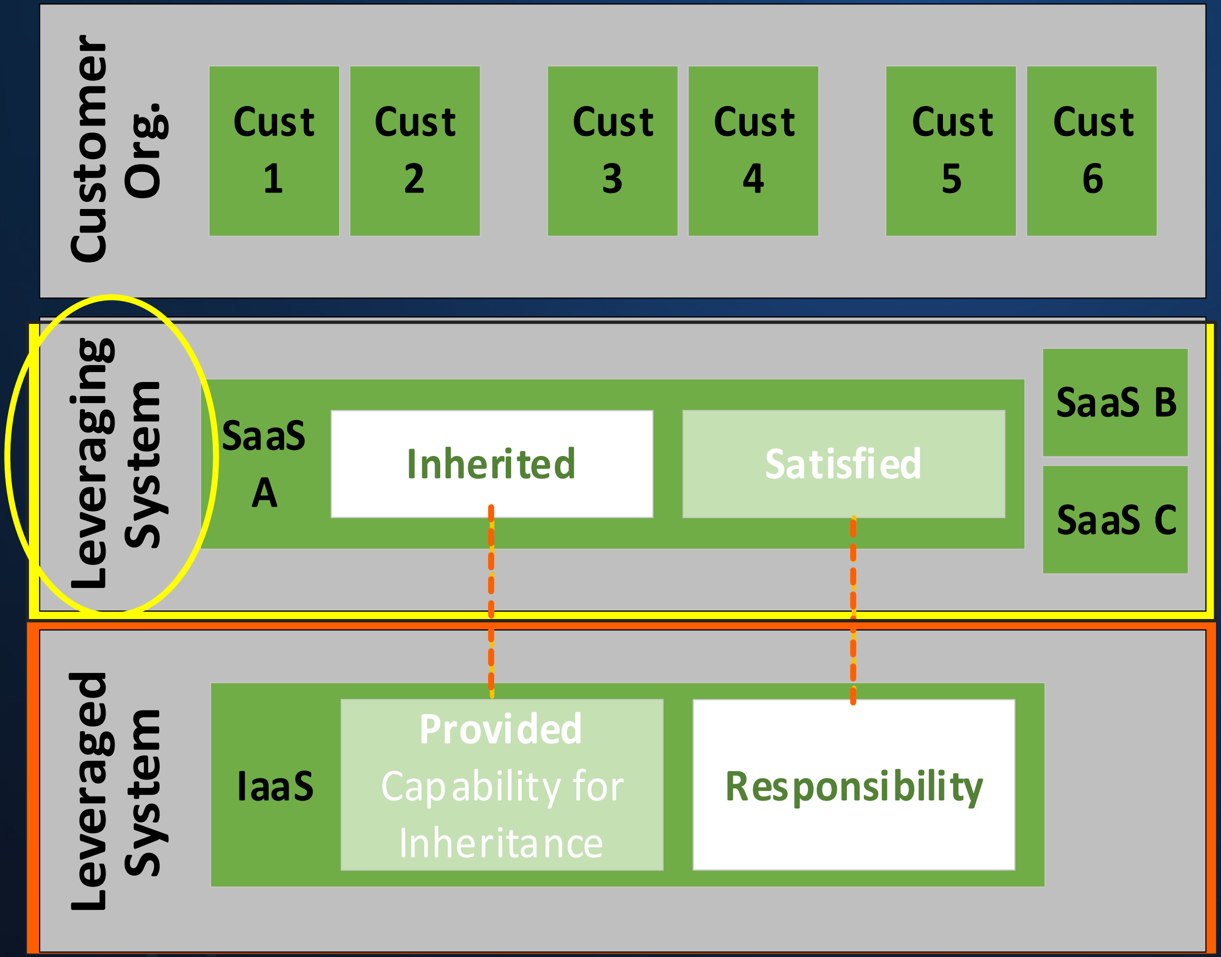


Leveraged System:

- The **leveraged** system's SSP should:
  - identify what **may be** inherited by **leveraging** systems
    - including a consumer-appropriate description of the control inheritance; and
  - Identify any responsibilities that must be addressed by the **leveraging** system to fully satisfy a control …

    … including where:
    - the **leveraging** system must be configured for an inherited capability; or
    - there is a gap in control satisfaction which must be addressed by the **leveraging** system

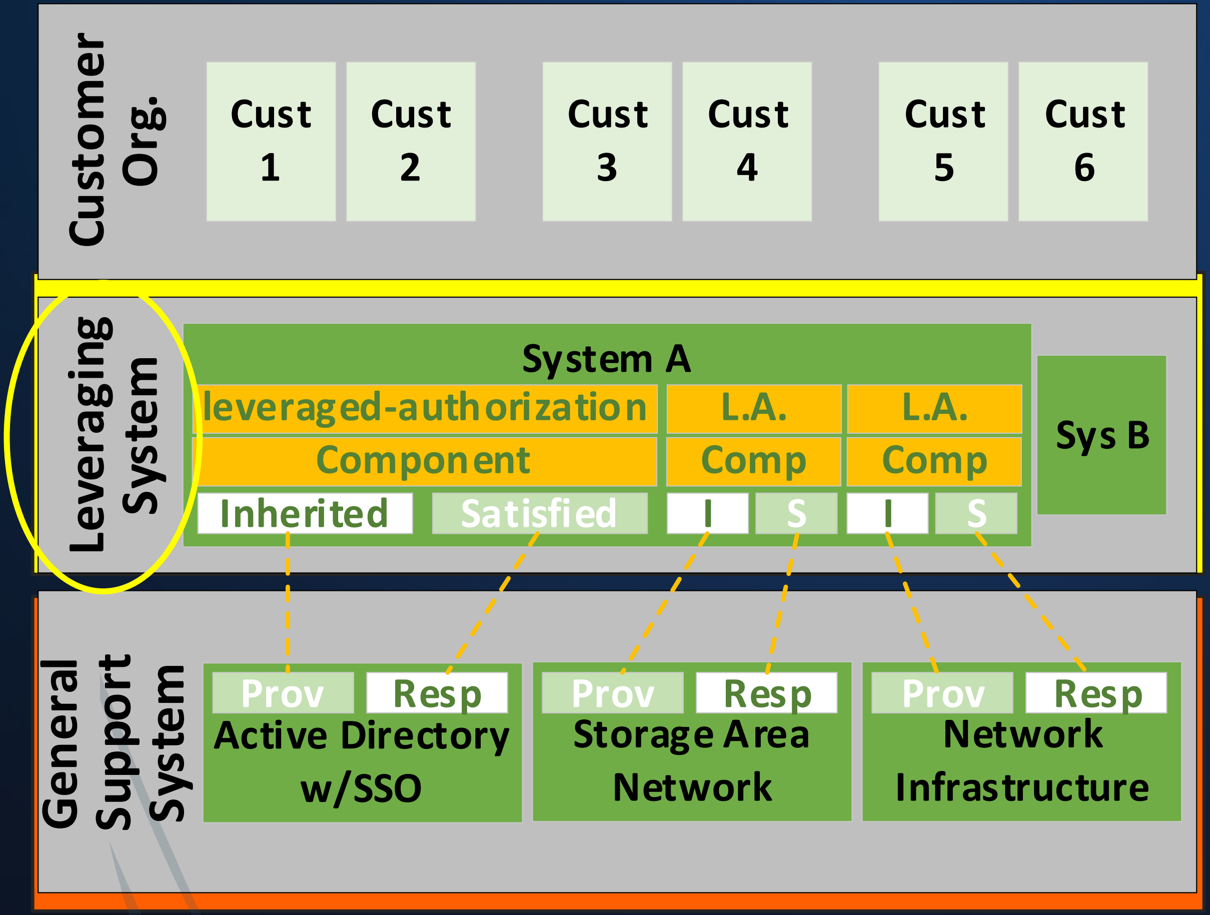
Control Documentation (Leveraging System View)



Leveraging System:

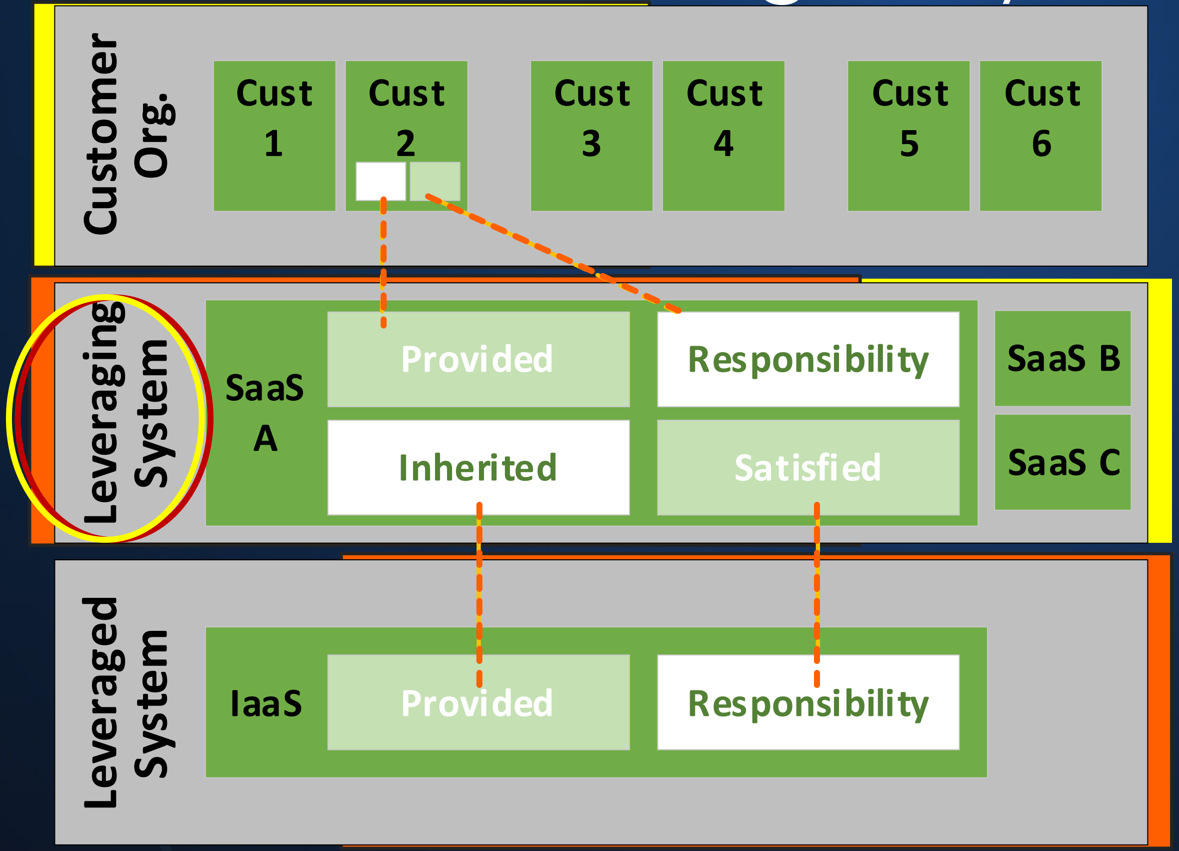
- The leveraging system's SSP should:
  - identify what **is** inherited from a leveraged system; and
  - identify any addressed responsibilities (as communicated by the leveraged system's SSP)
- These are linked from the leveraging system's SSP to the leveraged system's SSP using the UUID value associated with the "provided" and "responsibility" statements.
- Any components associated with these statements from the leveraged system's SSP must also be represented in the leveraging system's SSP.

Leveraging System with multiple Leveraged Systems



- **The same syntax is used**
  - It is simply replicated for each leveraged system
- **The Leveraging System's SSP:**
  - Has a separate "leveraged-authorization" assembly for each leveraged system.
  - Has a separate "component" representing each leveraged system.
  - Has a separate "component" representing the leveraged system components associated with inherited capabilities.

When a Leveraging System is also a Leveraged System



Leveraging System

- The **leveraging** system's SSP should:
  - identify what **is** inherited from a **leveraged** system
  - identify any addressed responsibilities (as identified by the leveraged system)
- In addition to:
  - identifying what **may be** inherited by the **leveraging** system's customers
  - any responsibilities the leverag**ing** system's customers must address to fully satisfy a control

Questions?

Have more questions?

Contact us directly at **oscal@nist.gov**

Join the community conversation at **https://gitter.im/usnistgov-OSCAL/Lobby**

National Institute of Standards and Technology
U.S. Department of Commerce