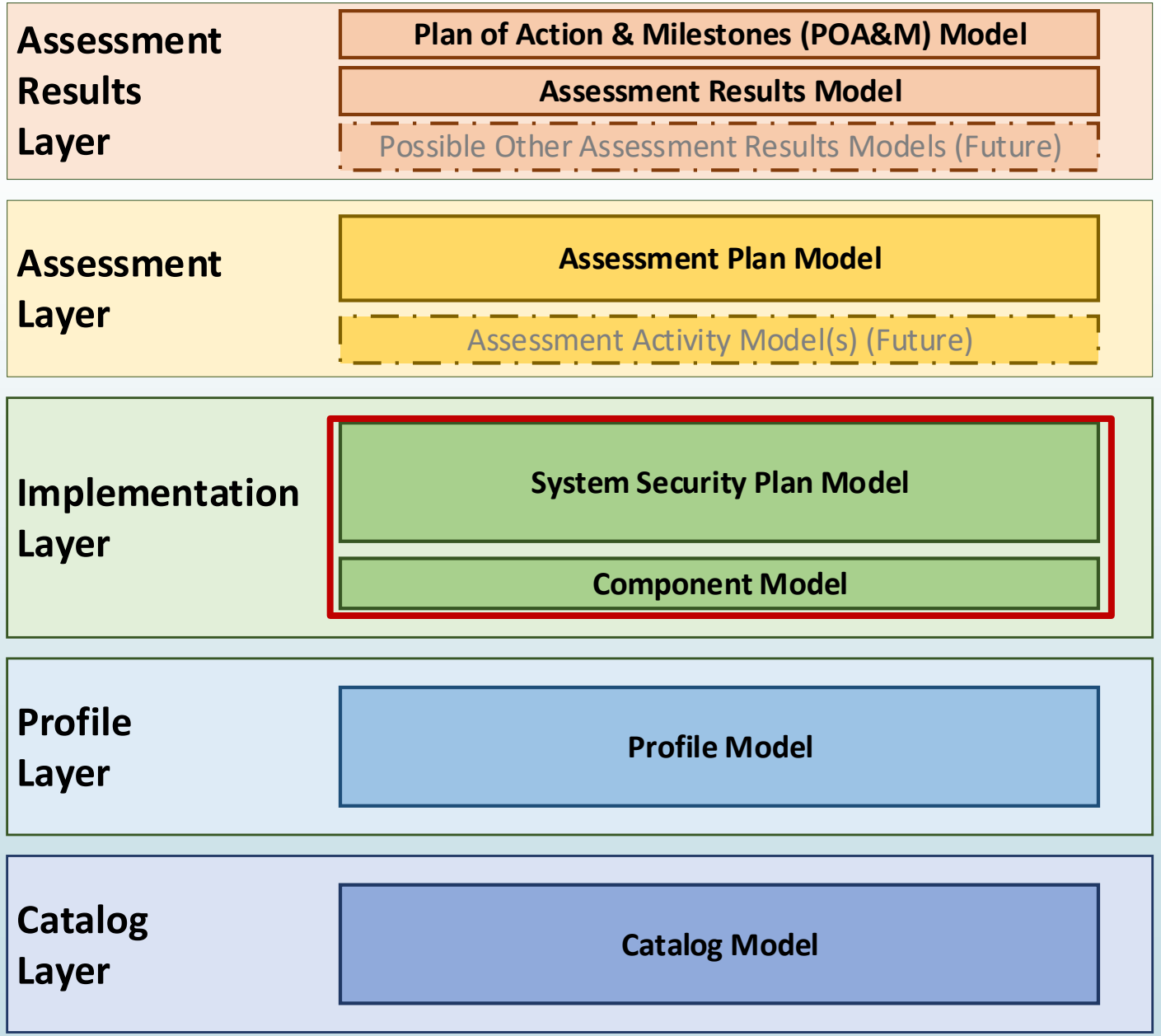
Open Security Controls Assessment Language (OSCAL) Leveraged Authorizations

Brian J. Ruf, CISSP, CCSP, PMP
National Institute of Standards and Technology

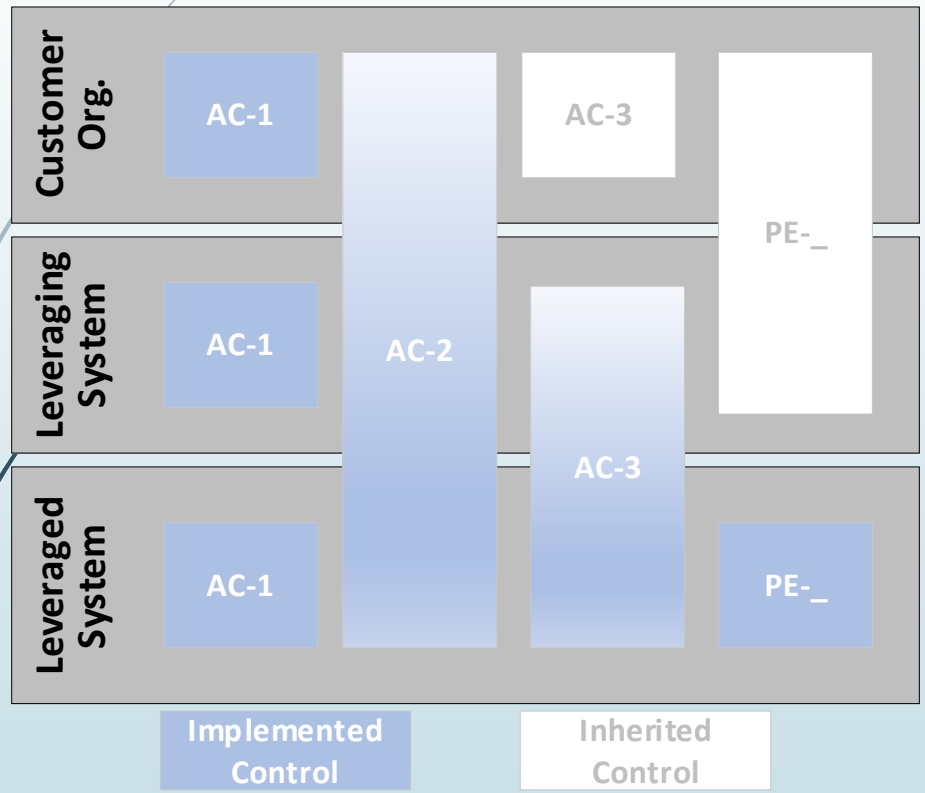
Overview

Leveraged Authorizations:

- Primarily the SSP Model
- Also the Component Model in some instances



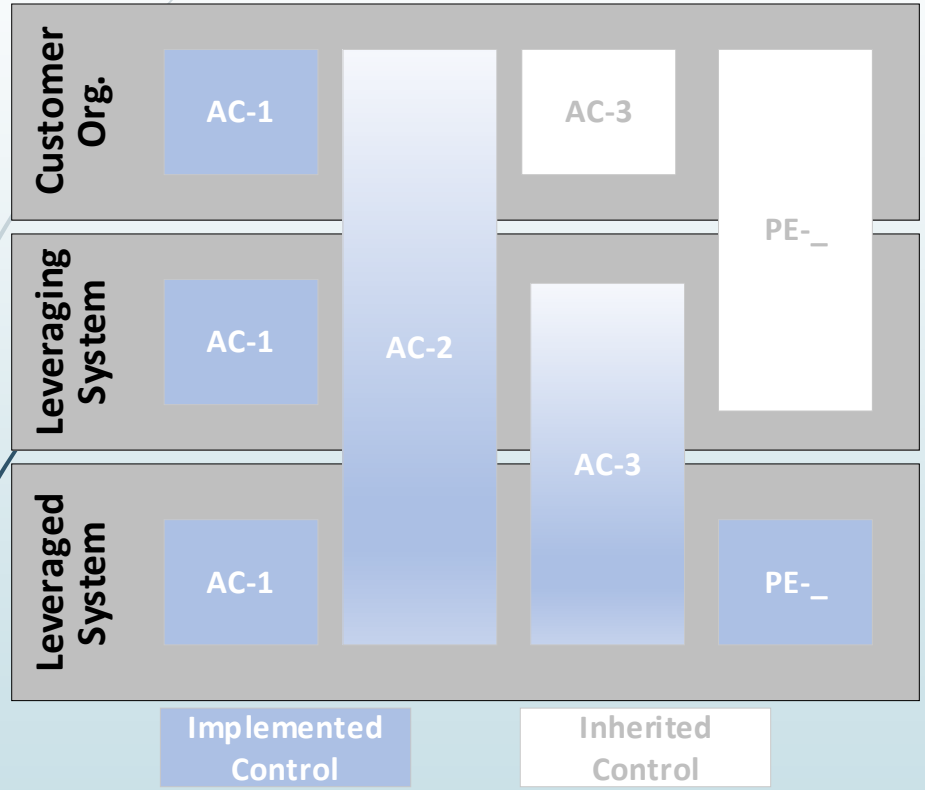
What is a Leveraged Authorization (LA)?



- **A leveraged authorization exists where:**
  - a leverag**ed** system passes responsibility for control satisfaction to one or more leverag**ing** systems (Customer Responsibility);

    and/or
  - a leverag**ing** system inherits security control satisfaction from a separately leverag**ed** system. (Inherited Control)
- Common examples:
  - **Cloud**: Several SaaS systems running on a separately authorized IaaS.
  - **Legacy**: Several systems relying on a separately authorized storage array or other general support system (GSS)

Control Satisfaction: Responsibilities and Inheritance



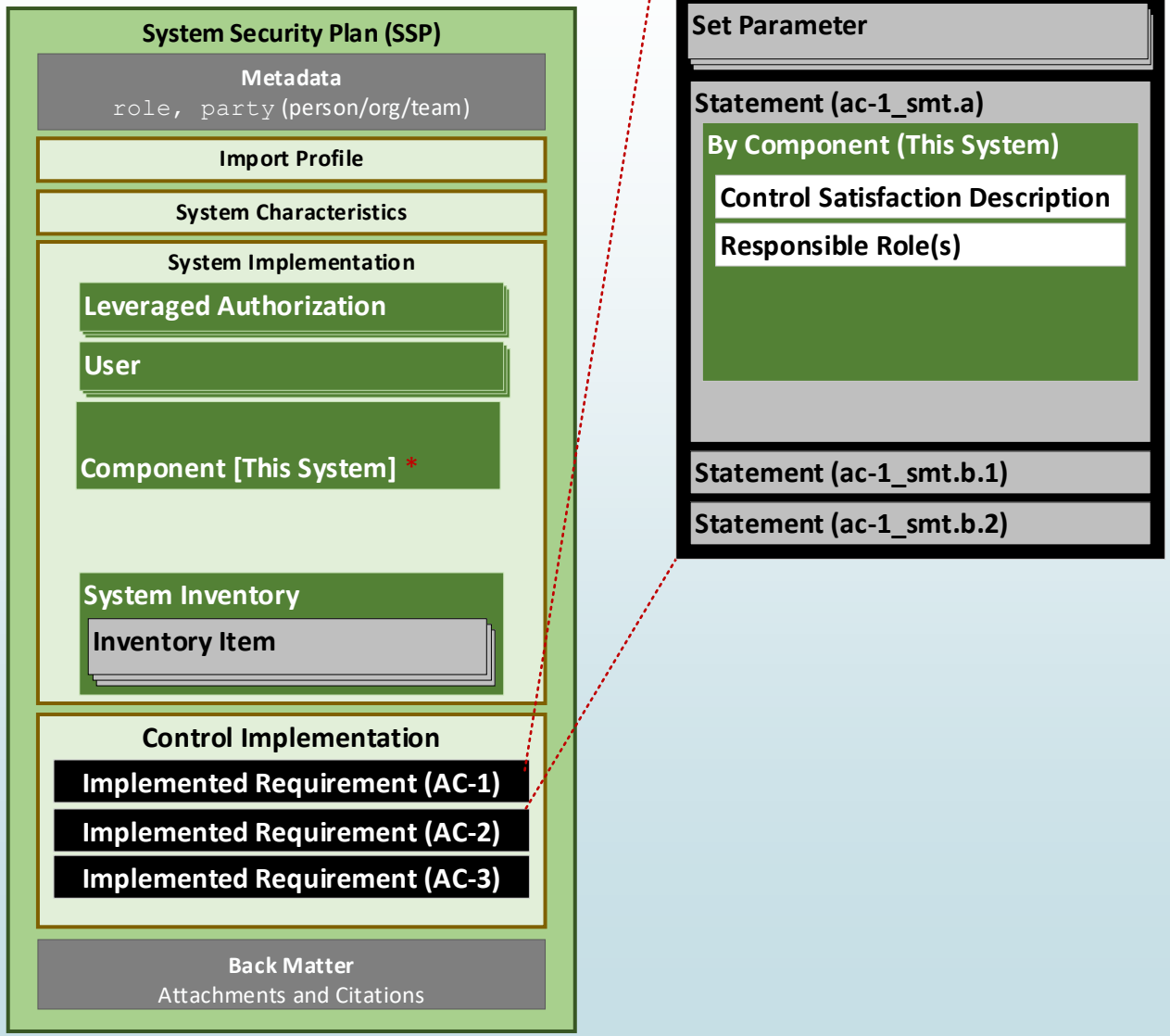
- **In fully satisfying a given control:**
  - Some controls must be satisfied independently by each system
    - Example: FedRAMP does not allow policies to be inherited. Each system owner must satisfy policy requirements independently.
  - Some controls are only fully satisfied if individual each system does their part.
    - Example: Logical access control must be implemented on all components in "the stack".
  - Some controls are fully satisfied at a lower level, thus fully inherited higher in the stack.
    - Example: Usually an IaaS takes care of all physical controls. Each SaaS has no ability to implement physical controls and fully inherits those controls from the IaaS.

Responding to Controls in the SSP: Two Approaches

- **Component Approach** (Preferred)
  - Each control response is broken down to the individual components involved.
  - Enables a more robust response to controls
  - Example: The access control implementation that satisfies *AC-2, part a* is described separately for the firewall, the router, the platform, the web server, etc.
- **System Approach** (Legacy)
  - Enables initial conversion of a document-based SSP to OSCAL with minimum re-organization of control responses.
  - Except for leveraged authorization content, each control response is tied to a single component: "This System"
  - Example: A legacy SSP has a single space for *AC-2, part a*, which has a free-text description the access controls within the system. This single description is associated with "This System" component in an OSCAL SSP.

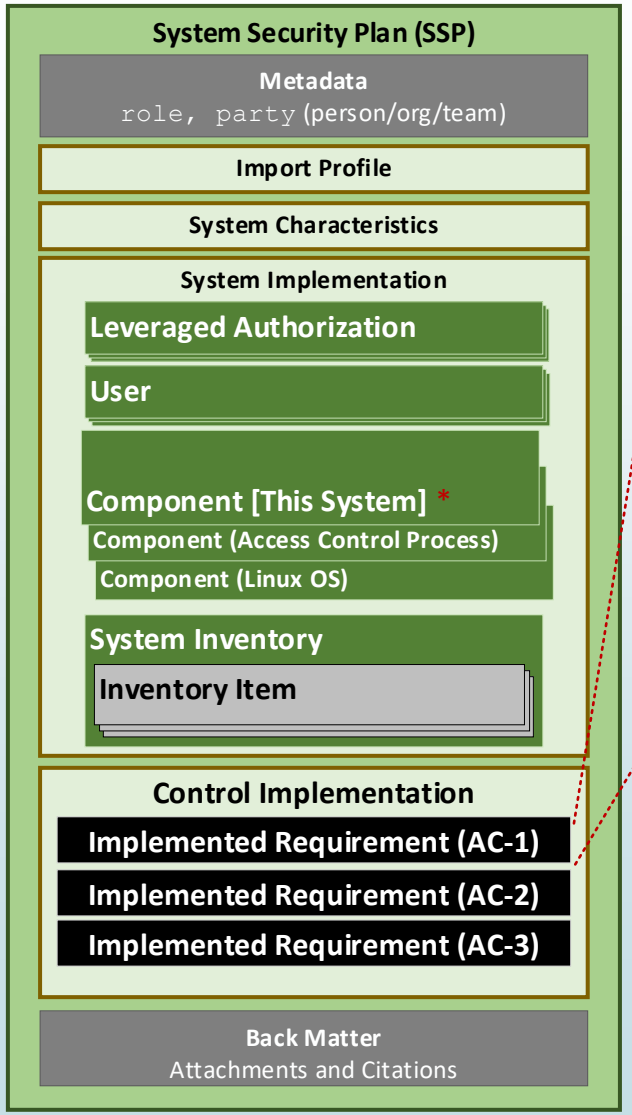
Responding to Controls in the SSP: System Approach

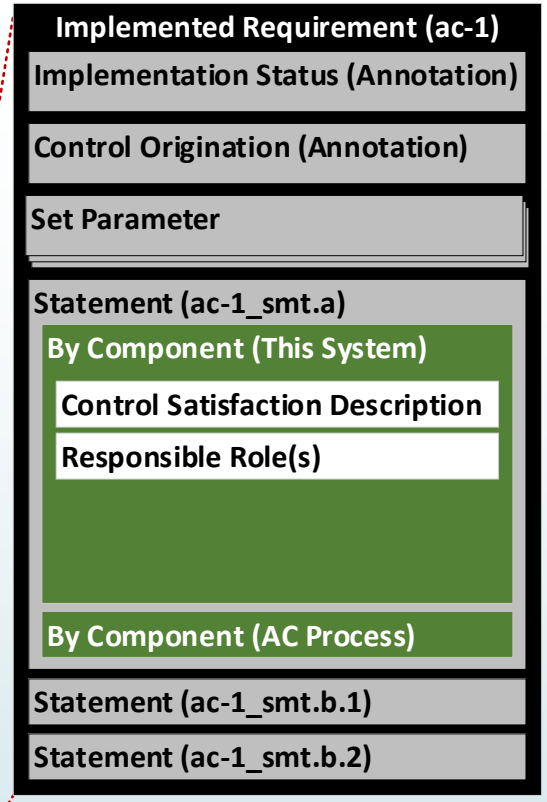
- A single component is defined in the *System Implementation* assembly.
  - This represents "This System"
- For each control:
  - There is an *Implemented Requirement* assembly.
  - Within the implemented requirement assembly, there is one or more *Statement* assemblies. One for each required response point.
  - Each statement has exactly one *By Component* assembly, which references "This System"
  - The entire control satisfaction description is entered in this By Component assembly exactly as it appeared in the legacy SSP.



Responding to Controls in the SSP: Component Approach

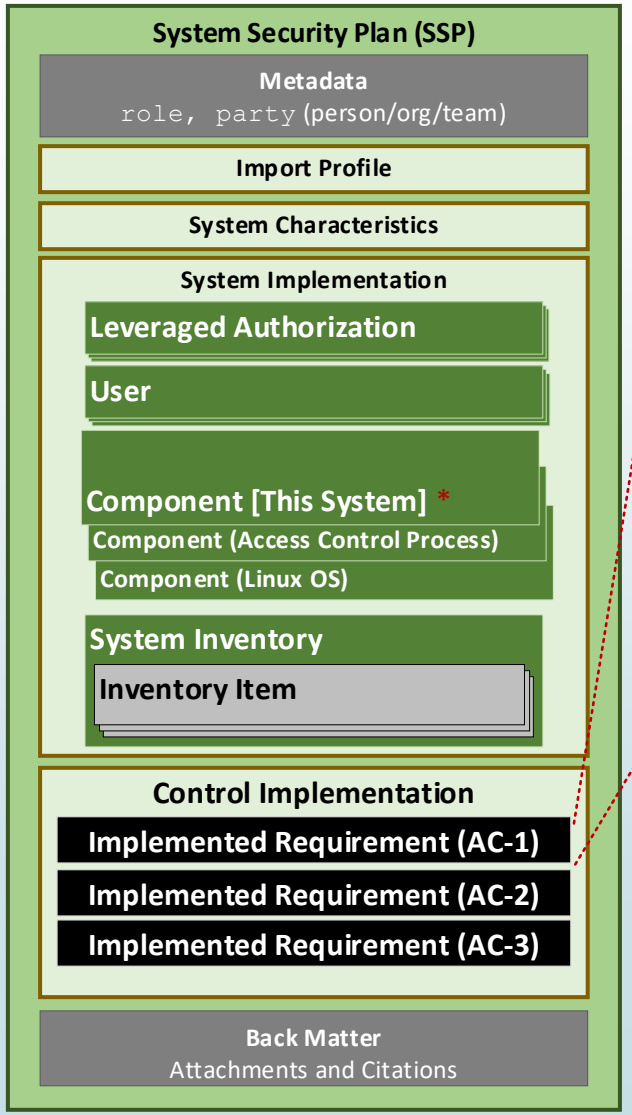
- Multiple components are defined in the *System Implementation* assembly.
  - There must still be a component for "This System"
- For each control:
  - There is an *Implemented Requirement* assembly.
  - Within the implemented requirement assembly, there is one or more *Statement* assemblies. One for each required response point.
  - Each statement has one or more *By Component* assemblies. Each references a component involved with control satisfaction.
  - Control satisfaction descriptions are provided within each by-component assembly.
  - Use the "This System" component for any control satisfaction explanation that does not fit cleanly with a more specific component, or to describe how the components work together.

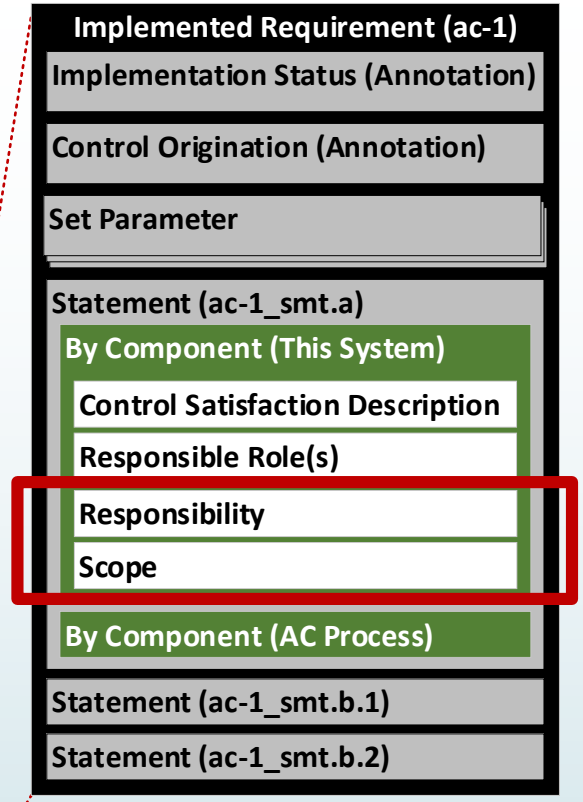




Correct Placement of Customer Responsibility Statements

- Customer responsibility statements are placed within a *By Component* assembly
- Ideally, they are placed in each *By Component* assembly for every component where a customer responsibility must be stated.
- If a customer responsibility statement does not fit any specific component, place it in the "This System" component.





Looking at the OSCAL (Customer Responsibilities)

Leveraged System

```
<control-implementation>
   <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-accc-d3e534932efb" />
   <implemented-requirement control-id="ac-2" uuid="7a36cf53-156d-4d1f-9a8b-433f61cc57b7">
      <annotation name="implementation-status" ns="https://fedramp.gov/ns/oscal" value="implemented" />
      <responsible-role role-id="admin-unix"/>
      <responsible-role role-id="program-director"/>
      <set-parameter param-id="ac-2_prm_1"><value>[SAMPLE]privileged, non-privileged</value></set-parameter>
      <set-parameter param-id="ac-2_prm_2" />
      <set-parameter param-id="ac-2_prm_3" />
      <set-parameter param-id="ac-2_prm_4" />
      <statement statement-id="ac-2_stmt.a" uuid="24a85abb-25ad-4686-850c-5c0e8ab69a0c">

         <by-component component-uuid="uuid-of-component-this-system" uuid="8a72663c-28c7-41c2-8739-f1ee2d5761ac">
            <description>
               <p>For the portion of the control satisfied by this system or its owning organization, describe
                  how the control is satisfied.</p>
            </description>
            <annotation name="responsibility" value="customer">
               <remarks>
                  <p>General customer responsibility description.</p>
               </remarks>
            </annotation>
         </by-component>

         <by-component component-uuid="uuid-of-component-application" uuid="8a72663c-28c7-41c2-8739-f1ee2d5761ac">
            <description>
               <p>For the portion of the control satisfied application, describe how the control is satisfied.</p>
            </description>
            <annotation name="responsibility" value="customer">
               <remarks>
                  <p>Describe the customer's responsibility within the application to satisfy this AC-2, part a.</p>
               </remarks>
            </annotation>
         </by-component>

      </statement>
   </implemented-requirement>
```

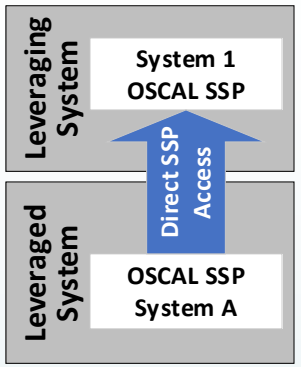
Three Scenarios

- **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 is needed.
  - **Preferred approach!**

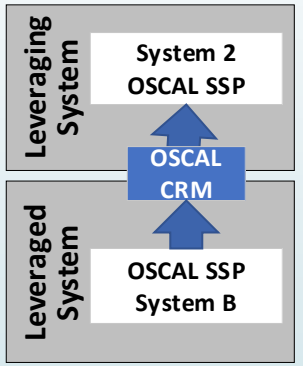
- **Scenario 2**: OSCAL SSP / No Access
  - The leveraged system is using an OSCAL SSP; however, the leveraging system is not permitted access it.
  - An OSCAL CRM is used.

- **Scenario 3**: Legacy SSP
  - A leveraged system is still using a legacy SSP.
  - A legacy Customer Responsibility Matrix (CRM) is used.

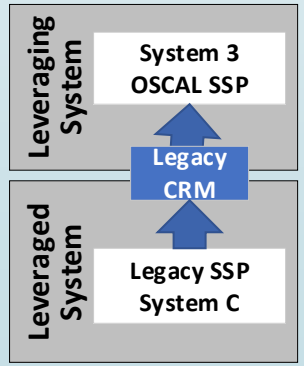
Scenario 1



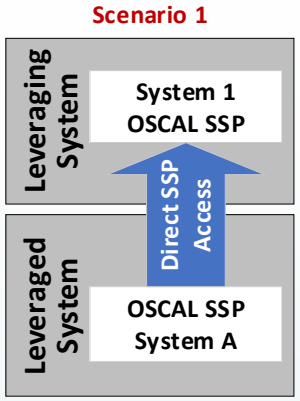
Scenario 2



Scenario 3

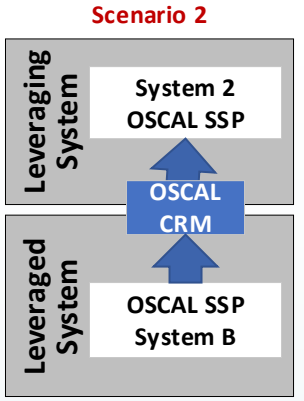


Scenario 1: OSCAL SSP With Access



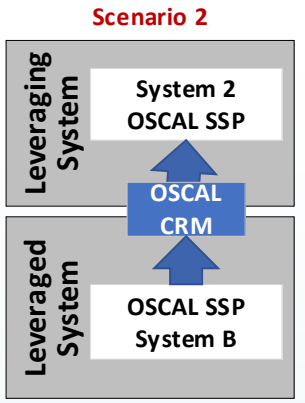
- Preferred scenario
- The SSP of the leveraging system can "see" the leveraged system's SSP
- Tools can identify which statements in the leveraged system's SSP have a customer responsibility
- Tools can further identify the leveraged system's components associated with these customer responsibility statements.
- The leveraging system's ISSO must determine if fulfillment of their customer responsibility involves the component from the leveraged system, or a new component that must be supplied by the leveraging system's organization.

Scenario 2: OSCAL SSP - No Access



- The SSP of the leveraging system is not permitted to "see" the full leveraged system's SSP.
- The leveraged system's owner, creates an OSCAL customer responsibility matrix (CRM), using the OSCAL Component model.
- Every component in the leveraged system's SSP, with a customer responsibility annotation is created in the OSCAL CRM with only basic information, such as the component title and general description.
  - The exact level of detail is a situation-specific decision.
  - The original Component UUID value from the leveraged system's SSP must be duplicated.
  - Every control, which cites that component AND associates it with a customer responsibility statement is cited in the control-implementation assembly within the component.
  - The entire "responsibility" annotation is duplicated from the SSP model by-component entry to the Component model statement-id assembly.
- The leveraging system's ISSO must determine if fulfillment of their customer responsibility involves the component from the leveraged system, or a new component that must be supplied by the leveraging system's organization.
  - If the leveraged system's component is used, the leveraging system's SSP must import the component detail from the CRM into the leveraging system's SSP.
  - The original UUID must be maintained.
  - The leveraging system's SSP must ensure they fully satisfy every customer responsibility statement in the CRM, which requires at least one entry within the cited statement.

Scenario 2: OSCAL SSP - No Access



System Security Plan (SSP)

Metadata
role, party (person/org/team)

Import Profile

System Characteristics

System Implementation

Leveraged Authorization

User

Component Description
Component [This System] *
Component (Access Control Process)
Component (Application)

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 (AC Process)

Statement (ac-2_smt.b)

Statement (ac-2_smt.c)

By Component (Application)

Control Satisfaction Description

Responsible Role(s)

Customer Responsibility Stmnt

Scope

Customer Responsibility Matrix (CRM)
Component Model

Metadata

Component

Component [This System] *

uuid of Component

Component Description

Implemented Requirement (AC-2)

Statement (ac-2_smt.a)

Customer Responsibility Stmnt

Component (Application)

uuid of Component

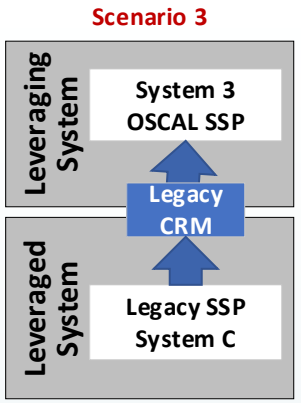
Component Description

Implemented Requirement (AC-2)

Statement (ac-2_smt.c)

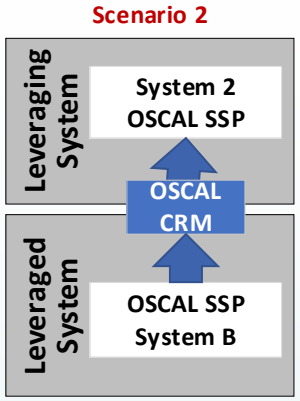
Customer Responsibility Stmnt

Scenario 3: Legacy SSP or CRM



- The leveraged system's SSP is not expressed in OSCAL, or its CRM is not.
- The leveraging system SSP must define an additional component representing the leveraged system itself.
- Every responsibility statement in the leveraged system's legacy SSP/CRM must be addressed by the leveraging system's SSP within the cited control statement.
- If the responsibility is addressed by customer action in the leveraged system, the leveraging system's statement should cite that component. Otherwise, it should cite the appropriate component.

Inheritance in an OSCAL CRM



- The leveraged system's CRM can represent components from the system even if there is no customer responsibility.
- While individual component references are preferred, if the leveraged system's owner or ISSO does not wish to expose individual components, they may still provide a CRM with a "this system" component.
- Whether individual components or simply a "this system" component, the leveraged system's CRM can cite each control satisfied by the component and provide a customer-appropriate description of the satisfaction.
  - For example, FedRAMP requires the leveraging system to only describe "what" is being inherited from a leveraged system in satisfaction of a control but does not require a description of "how" in this case. The CRM can provide a control-statement-specific description of what is being inherited.

Questions?
Thank you!

We want your feedback!

OSCAL Repository:
https://github.com/usnistgov/OSCAL

Project Website:
https://www.nist.gov/oscal

How to Contribute:
https://pages.nist.gov/OSCAL/contribute/

FedRAMP Implementation Guides
https://github.com/gsa/fedramp-automation **(Available in July)**

BACKUP SLIDE(S)

System Approach vs. Component Approach

- The **System Approach** is more consistent with legacy SSP content, where a single description exists for the entire system.

SYSTEM APPROACH: AC-2 What is the solution and how is it implemented?

Part a	System	Describes how *part a* is satisfied by this system or this organization's policies/processes.
	Inherited	Describes what is inherited from the underlying Infrastructure as a Service (IaaS) provider to satisfy *part a*.
	Customer	Describes the customer responsibilities with respect to *part a*.
Part b	System	Describes how *part b* is satisfied by this system or this organization's policies/processes.
	Inherited	Describes what is inherited from the underlying Infrastructure as a Service (IaaS) provider to satisfy *part b*.
	Customer	Describes the customer responsibilities with respect to *part b*.

- The **Component Approach** is preferred. It provides a description for each component contributing to the satisfaction of the control.

COMPONENT APPROACH: AC-2 What is the solution and how is it implemented?

Part a	Platform	Describes how *part a* is satisfied by the platform.
	Web-server	Describes how *part a* is satisfied by the web server
	Process	Describes how *part a* is satisfied by an identified process within this organization.
	Inherited	Describes what is inherited from the underlying Infrastructure as a Service (IaaS) provider to satisfy *part a*.
	Customer	Describes the customer responsibilities with respect to *part a*.
Part b	Platform	Describes how *part b* is satisfied by the platform.
	Web-server	Describes how *part b* is satisfied by the web server
	Process	Describes how *part b* is satisfied by an identified process within this organization.
	Inherited	Describes what is inherited from the underlying Infrastructure as a Service (IaaS) provider to satisfy *part b*.
	Customer	Describes the customer responsibilities with respect to *part b*.

- The System Approach is intended for converting legacy SSP content to OSCAL. Once converted, system owners are encouraged to migrate to the Component Approach. The design allows for a mix of both, enabling an organization to migrate slowly over time.