Open Security Controls Assessment Language (OSCAL) Leveraged Authorizations and Customer Responsibilities

Brian J. Ruf, CISSP, CCSP, PMP
National Institute of Standards and Technology
Version 6a
October 2, 2020
(Revised April 14, 2023)

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!**

  Today's Focus - Wrap Up

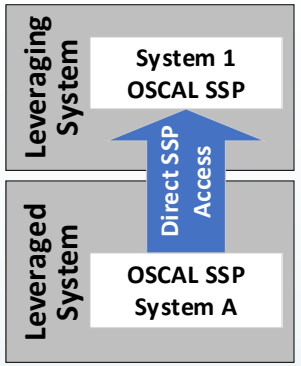
- **Scenario 2**: OSCAL SSP / No Access
  - The leveraged system is using an OSCAL SSP; however, the leveraging system is not permitted access it.
  - An OSCAL CRM is used.
  - **Typical FedRAMP Scenario**

  Late Oct/Early Nov

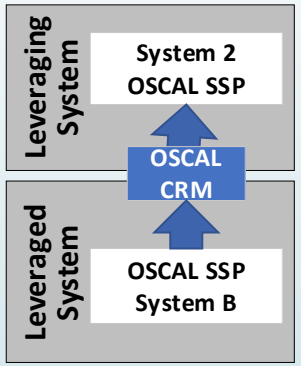
- **Scenario 3**: Legacy SSP
  - A leveraged system is still using a legacy SSP.
  - A legacy Customer Responsibility Matrix (CRM) is used.
  - **Transition scenario for an imperfect world**

  Nov/Dec

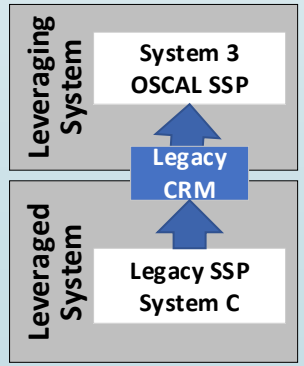
Scenario 1



Scenario 2



Scenario 3



OSCAL Incorporation

- **SSP Syntax Updated**
  - OSCAL Repo PR #762 Pending
- **Leveraged Authorization Sample Files Available**
  - OSCAL Content Repo PR #26 Pending
  - Two sample OSCAL SSPs were drafted in parallel to this presentation:
    - Leveraged System
    - Leveraging System

OSCAL SSP Syntax Changes

- **Added the following assemblies into the `by-component` assembly**
  - `export`; which includes the following assemblies:
    - `provided`
    - `responsibility`
  - `inherited`
  - `satisfied`
- **Each assembly includes:**
  - `description`
  - `prop`
  - `link`
  - `responsible-role`
  - `remarks`
- **Added three additional `name` value options to the `component/prop` field**
  - `implementation-point`, `leveraged-authorization-uuid`, `inherited-uuid`
- **Removed `description` from `implemented-requirement` and `statement`**
  - Ensures all control response statements occur within a `by-component` assembly

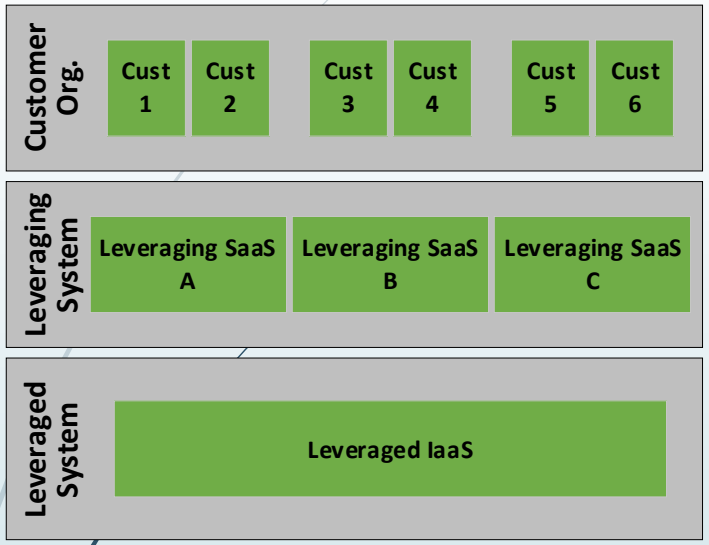
What is a Leveraged Authorization?

- **A leveraged authorization (LA) exists where:**
  - one or more leverag**ing** systems relies on a leverag**ed** system for operation in a stacked hierarchy; and
  - any leveraging system is authorized separately from the leveraged system.
  - External services and interconnections are not regarded as leveraged authorizations.

(*Examples on next slide*)

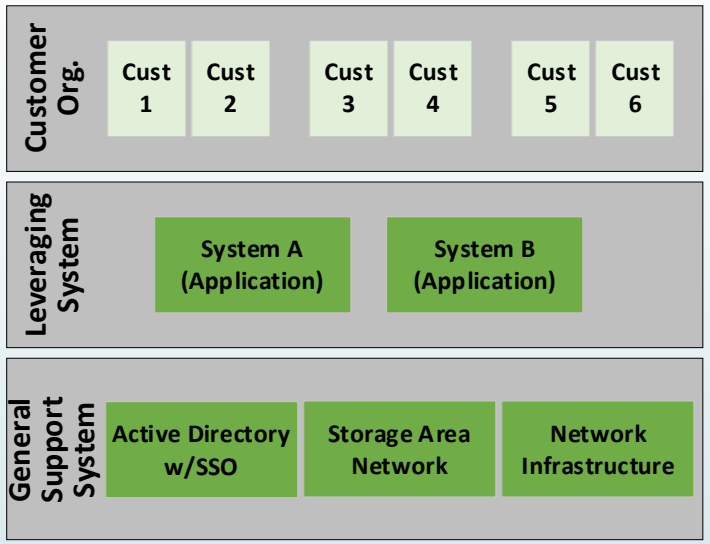
What is a Leveraged Authorization? (continued)

Yes



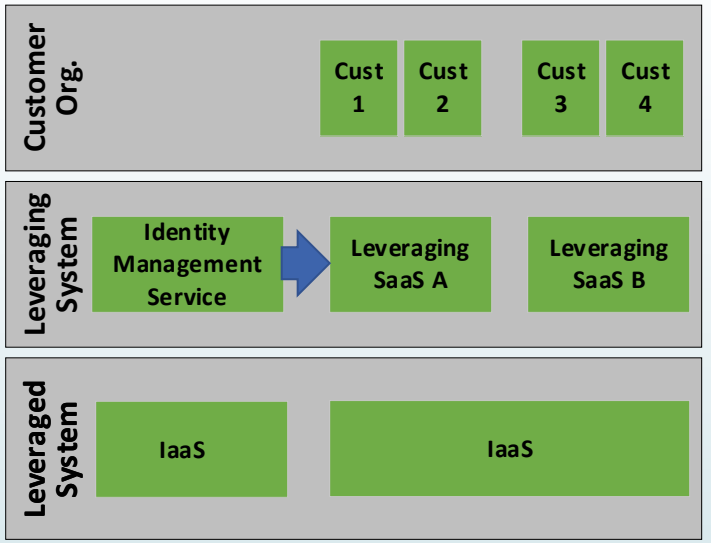
Cloud (SaaS on IaaS)

Yes



Data Center (System on GSS)

No



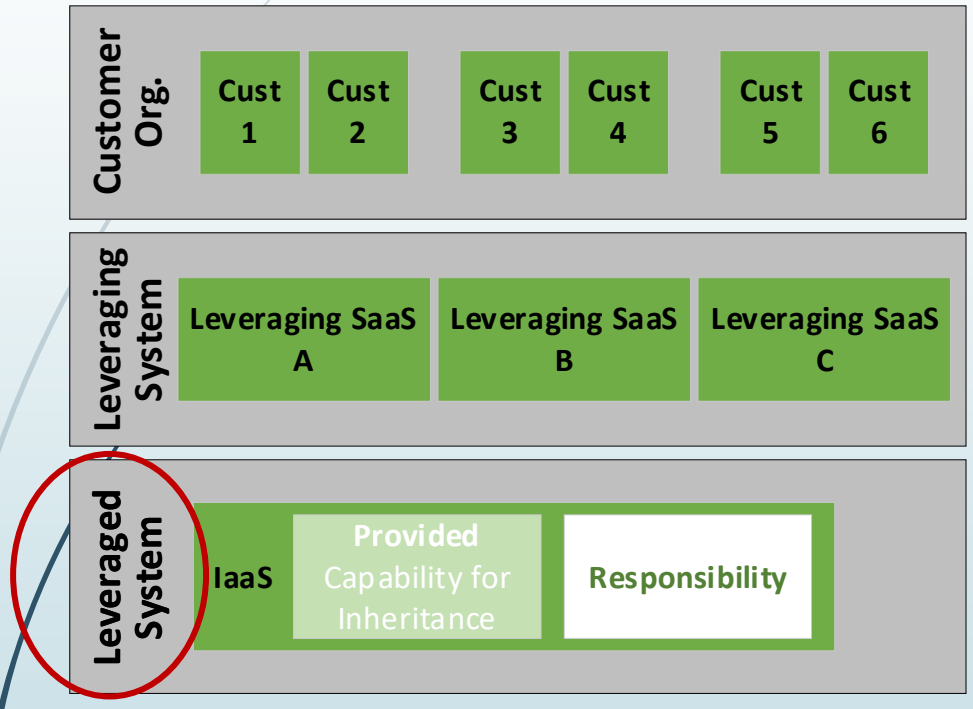
External Service or Interconnection

- **Cloud**: Several SaaS systems running on a separately authorized IaaS.
- **Data Center**: Several systems relying on a separately authorized storage array or other general support system (GSS)

- 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.

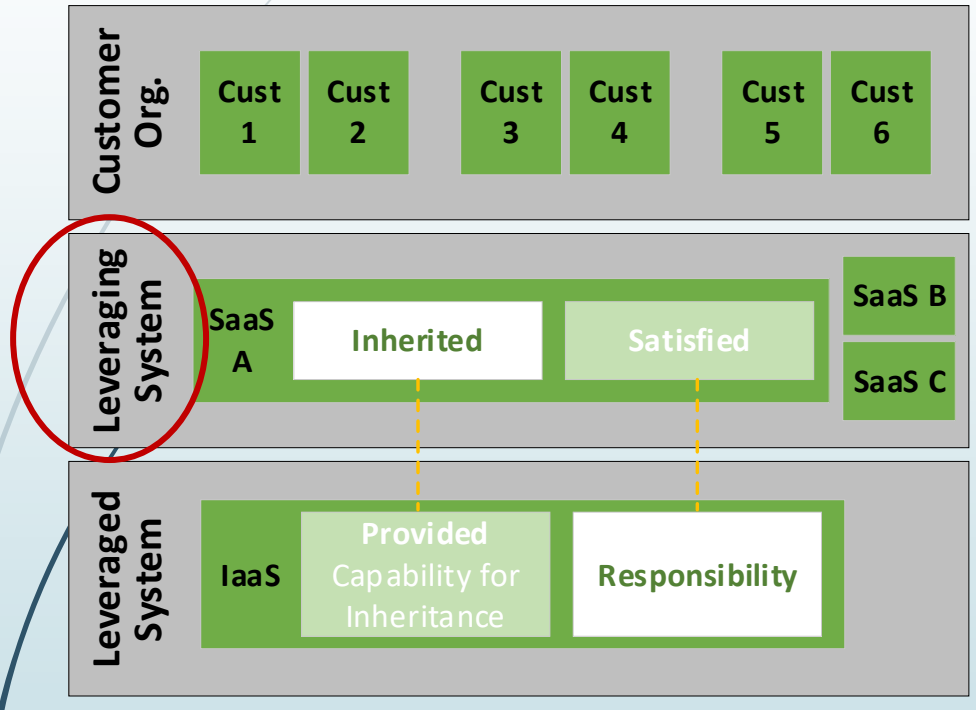
Control Documentation (System View)



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

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

Control Documentation (System View)



- **Leveraging System:**
  - The leverag**ing** system's SSP should:
    - identify what **is** inherited from a leverag**ed** 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.

Leveraged System: Customer Responsibility

- uuid values assigned within leverag**ed** system's SSP begin with "11111111".
- uuid values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraged System (Customer responsibility to address gap - associate with the "This System" component)

```
<control-implementation>
   <implemented-requirement control-id="ac-1" uuid="11111111-0000-4000-9009-001000000000" />
   <implemented-requirement control-id="ac-2" uuid="11111111-0000-4000-9009-002000000000">
      <annotation name="implementation-status" ns="https://fedramp.gov/ns/oscal" value="implemented" />
      <set-parameter param-id="ac-2_prm_1"><value>[SAMPLE]privileged, non-privileged</value></set-parameter>

      <statement statement-id="ac-2_stmt.a" uuid="11111111-0000-4000-9009-002001000000">

         <by-component component-uuid="11111111-0000-4000-9001-000000000001" uuid="assigned-uid-value">
            <description>
               <p>Description of how AC-2, part a is satisfied within this system.</p>
            </description>

            <export>
               <description>
                  <p>Overall description of what is exportable to leveraging systems for AC-2, part a.</p>
               </description>

               <responsibility uuid="11111111-0000-4000-9009-002001001001">
                  <description>
                     <p>Leveraging system's responsibilities in satisfaction of AC-2, part a,
                        where there is no inheritance.</p>
                  </description>
                  <responsible-role role-id="customer" />
               </responsibility>

            </export>
         </by-component>

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

Within the by-component assembly, use export/responsibility to define a customer responsibility.

Leveraging System Component Definitions

- uuid values assigned within leverag**ed** system's SSP begin with "11111111".
- uuid values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraging System (component from leveraged system)

```
<system-implementation>
   <leveraged-authorization uuid="22222222-0000-4000-9000-300000000001">
      <title>CSP IaaS [Leveraged System]</title>
      <link href="./oscal_csp-example_ssp.xml" rel="OSCAL-SSP-XML" />
      <party-uuid>22222222-0000-4000-9000-100000000002</party-uuid>
      <date-authorized>2018-01-01</date-authorized>
   </leveraged-authorization>
   <!-- user -->

      <component uuid="22222222-0000-4000-9001-000000000001" component-type="this-system">
         <title>THIS SYSTEM (SaaS)</title>
         <description>
            <p>This Leveraging SaaS.</p>
            <p>The entire system as depicted in the system authorization boundary</p>
         </description>
         <prop name="implementation-point">system</prop>
         <status state="operational"/>
      </component>

      <component uuid="22222222-0000-4000-9001-000000000002" component-type="system">
         <title>LEVERAGED SYSTEM (IaaS)</title>
         <description>
            <p>Brief description of the leveraged system.</p>
         </description>
         <prop name="implementation-point">external</prop>
         <prop name="leveraged-authorization-uuid">22222222-0000-4000-9000-300000000001</prop>
         <prop name="inherited-uuid"              >11111111-0000-4000-9001-000000000001</prop>
         <status state="operational"/>
      </component>

      <component uuid="22222222-0000-4000-9001-000000000003" component-type="appliance"><!--cut--></component>

</system-implementation>
```

Component represents the leveraged system.

Leveraging System Component Definitions

- `uuid` values assigned within leverag**ed** system's SSP begin with "11111111".
- `uuid` values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraging System (component from leveraged system)

```
<system-implementation>
   <leveraged-authorization uuid="22222222-0000-4000-9000-300000000001">
      <title>CSP IaaS [Leveraged System]</title>
      <link href="./oscal_csp-example_ssp.xml" rel="OSCAL-SSP-XML" />
      <party-uuid>22222222-0000-4000-9000-100000000002</party-uuid>
      <date-authorized>2018-01-01</date-authorized>
   </leveraged-authorization>
   <!-- user -->

      <component uuid="22222222-0000-4000-9001-000000000001" component-type="this-system">
         <title><b>THIS SYSTEM (SaaS)</b></title>
         <description>
            <p>This Leveraging SaaS.</p>
            <p>The entire system as depicted in the system authorization boundary</p>
         </description>
         <prop name="implementation-point">system</prop>
         <status state="operational"/>
      </component>

      <component uuid="22222222-0000-4000-9001-000000000002" component-type="system">
         <title><b>LEVERAGED SYSTEM (IaaS)</b></title>
         <description>
            <p>Brief description of the leveraged system.</p>
         </description>
         <prop name="implementation-point">external</prop>
         <prop name="leveraged-authorization-uuid">22222222-0000-4000-9000-300000000001</prop>
         <prop name="inherited-uuid"              >11111111-0000-4000-9001-000000000001</prop>
         <status state="operational"/>
      </component>

      <component uuid="22222222-0000-4000-9001-000000000003" component-type="appliance"><!--cut--></component>

</system-implementation>
```

Links to Leveraged Authorization assembly

Leveraging System Component Definitions

- `uuid` values assigned within leverag**ed** system's SSP begin with "11111111".
- `uuid` values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraging System (component from leveraged system)

```
<system-implementation>
   <leveraged-authorization uuid="22222222-0000-4000-9000-300000000001">
      <title>CSP IaaS [Leveraged System]</title>
      <link href="./oscal_csp-example_ssp.xml" rel="OSCAL-SSP-XML" />
      <party-uuid>22222222-0000-4000-9000-100000000002</party-uuid>
      <date-authorized>2018-01-01</date-authorized>
   </leveraged-authorization>
   <!-- user -->

      <component uuid="22222222-0000-4000-9001-000000000001" component-type="this-system">
         <title><b>THIS SYSTEM (SaaS)</b></title>
         <description>
            <p>This Leveraging SaaS.</p>
            <p>The entire system as depicted in the system authorization boundary</p>
         </description>
         <prop name="implementation-point">system</prop>
         <status state="operational"/>
      </component>

      <component uuid="22222222-0000-4000-9001-000000000002" component-type="system">
         <title><b>LEVERAGED SYSTEM (IaaS)</b></title>
         <description>
            <p>Brief description of the leveraged system.</p>
         </description>
         <prop name="implementation-point">external</prop>
         <prop name="leveraged-authorization-uuid">22222222-0000-4000-9000-300000000001</prop>
         <prop name="inherited-uuid"              >11111111-0000-4000-9001-000000000001</prop>
         <status state="operational"/>
      </component>

      <component uuid="22222222-0000-4000-9001-000000000003" component-type="appliance"><!--cut--></component>

</system-implementation>
```

Original component UUID from leveraged system. Establishes traceability back to Leveraged SSP.

Leveraging System Statement Response

- `uuid` values assigned within leverag**ed** system's SSP begin with "11111111".
- `uuid` values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraging System (consumer responsibility - addressing a gap)

```
<control-implementation>
   <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-accc-d3e534932efb" />
   <implemented-requirement control-id="ac-2" uuid="uuid-value">
      <!-- details cut -->
      <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

         <by-component component-uuid="22222222-0000-4000-9001-000000000001" uuid="uuid-value">
         </by-component>

         <by-component component-uuid="22222222-0000-4000-9001-000000000002" uuid="uuid-value">
            <description>
               <p>Describe how this internal virtual appliance satisfies AC-2, Part a.</p>
            </description>

            <satisfied uuid="uuid-value" responsibility-uuid="11111111-0000-4000-9009-002001001001">
               <description>
                  <p>Description that directly addresses how the consumer responsibility was satisfied.</p>
               </description>
               <responsible-role>role-id</responsible-role>
            </satisfied>

         </by-component>

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

The leveraging system owner deploys a virtual appliance to address the consumer responsibility defined in the leveraged system's SSP for AC-2, part a.

This is the UUID of the responsibility statement in the leveraged system's SSP

Leveraged System Providing Inheritance

- `uuid` values assigned within leverag**ed** system's SSP begin with "11111111".
- `uuid` values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraged System (Inheritance - associate with each component providing an inheritable capability)

```
<control-implementation>
   <implemented-requirement control-id="ac-1" uuid="11111111-0000-4000-9009-001000000000" />
   <implemented-requirement control-id="ac-2" uuid="11111111-0000-4000-9009-002000000000">
      <annotation name="implementation-status" ns="https://fedramp.gov/ns/oscal" value="implemented" />
      <set-parameter param-id="ac-2_prm_1"><value>[SAMPLE]privileged, non-privileged</value></set-parameter>
      <statement statement-id="ac-2_stmt.a" uuid="uuid-value">
         <by-component component-uuid="11111111-0000-4000-9001-000000000001" uuid="uuid-value"><!--cut--></by-component>
         <by-component component-uuid="11111111-0000-4000-9001-000000000002" uuid="uuid-value">
            <description>
               <p>Description of how the application component satisfies AC-2, part a.</p>
            </description>
            <export>
               <description>
                  <p>Overall description of what is exportable to leveraging systems for AC-2, part a.</p>
               </description>

               <provided uuid="11111111-0000-4000-9009-002001002001">
                  <description>
                     <p>Consumer-appropriate description of what may be inherited.</p>
                     <p>In the context of the application component in satisfaction of AC-2, part a.</p>
                  </description>
                  <responsible-role role-id="poc-for-customers" />
               </provided>

               <responsibility uuid="11111111-0000-4000-9009-002001002002" provided-uuid="11111111-0000-4000-9009-002001002001">
                  <description>
                     <p>Leveraging system's responsibilities with respect to inheriting this capability.</p>
                     <p>In the context of the application component in satisfaction of AC-2, part a.</p>
                  </description>
                  <responsible-role role-id="customer" />
               </responsibility>
            </export>
         </by-component>
      </statement>
   </implemented-requirement>
```

Use "export/provided" to describe what may be inherited using consumer-appropriate language.

Leveraged System Providing Inheritance

- `uuid` values assigned within leverag**ed** system's SSP begin with "11111111".
- `uuid` values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraged System (Inheritance - associate with each component providing an inheritable capability)

```
<control-implementation>
   <implemented-requirement control-id="ac-1" uuid="11111111-0000-4000-9009-001000000000" />
   <implemented-requirement control-id="ac-2" uuid="11111111-0000-4000-9009-002000000000">
      <annotation name="implementation-status" ns="https://fedramp.gov/ns/oscal" value="implemented" />
      <set-parameter param-id="ac-2_prm_1"><value>[SAMPLE]privileged, non-privileged</value></set-parameter>
      <statement statement-id="ac-2_stmt.a" uuid="uuid-value">
         <by-component component-uuid="11111111-0000-4000-9001-000000000001" uuid="uuid-value"><!--cut--></by-component>
         <by-component component-uuid="11111111-0000-4000-9001-000000000002" uuid="uuid-value">
            <description>
               <p>Description of how the application component satisfies AC-2, part a.</p>
            </description>
            <export>
               <description>
                  <p>Overall description of what is exportable to leveraging systems for AC-2, part a.</p>
               </description>

               <provided uuid="11111111-0000-4000-9009-002001002001">
                  <description>
                     <p>Consumer-appropriate description of what may be inherited.</p>
                     <p>In the context of the application component in satisfaction of AC-2, part a.</p>
                  </description>
                  <responsible-role role-id="poc-for-customers" />
               </provided>

               <responsibility uuid="11111111-0000-4000-9009-002001002002" provided-uuid="11111111-0000-4000-9009-002001002001">
                  <description>
                     <p>Leveraging system's responsibilities with respect to inheriting this capability.</p>
                     <p>In the context of the application component in satisfaction of AC-2, part a.</p>
                  </description>
                  <responsible-role role-id="customer" />
               </responsibility>
            </export>
         </by-component>
      </statement>
   </implemented-requirement>
```

If there is a consumer responsibility associated with this inheritance, define it within the same export assembly and link it using the provided-uuid flag.

Leveraging System Component Definitions

- `uuid` values assigned within leverag**ed** system's SSP begin with "11111111".
- `uuid` values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraging System (component from leveraged system)

```
<system-implementation>
   <leveraged-authorization uuid="22222222-0000-4000-9000-300000000001">
      <title>CSP IaaS [Leveraged System]</title>
      <link href="./oscal_csp-example_ssp.xml" rel="OSCAL-SSP-XML" />
      <party-uuid>22222222-0000-4000-9000-100000000002</party-uuid>
      <date-authorized>2018-01-01</date-authorized>
   </leveraged-authorization>
   <!-- user -->

      <component uuid="22222222-0000-4000-9001-000000000001" component-type="this-system"><!--cut--></component>

      <component uuid="22222222-0000-4000-9001-000000000002" component-type="system"><!--cut--></component>

      <component uuid="22222222-0000-4000-9001-000000000003" component-type="appliance"><!--cut--></component>

      <component uuid="22222222-0000-4000-9001-000000000004" component-type="application">
         <title><b>Application from Leveraged System</b></title>

         <description>
            <p>Description of the application within the leveraged system.</p>
         </description>

         <prop name="implementation-point">external</prop>
         <prop name="leveraged-authorization-uuid">22222222-0000-4000-9000-300000000001</prop>
         <prop name="inherited-uuid"              >11111111-0000-4000-9001-000000000002</prop>
         <status state="operational"/>
      </component>

</system-implementation>
```

Component represents the application, which provides inheritable capabilities from the leveraged system.

Leveraging System Receiving Inheritance

- `uuid` values assigned within leverag**ed** system's SSP begin with "11111111".
- `uuid` values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraging System (inheriting the capability of a component)

The leveraging system owner elects to inherit the leveraged system's application in satisfaction of AC-2, Part a.

```
<control-implementation>
   <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-accc-d3e534932efb" />
   <implemented-requirement control-id="ac-2" uuid="uuid-value">
      <!-- details cut -->
      <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

         <by-component uuid="uuid-value" component-uuid="22222222-0000-4000-9001-000000000003">
            <description>
               <p><i>duplicated/tailored description of what was inherited,
                     and description of what was configured.</i></p>
               <p>Consumer-appropriate description of what may be inherited.</p>
               <p>In the context of the application component in satisfaction of AC-2, part a.</p>
            </description>

            <inherited provided-uuid="11111111-0000-4000-9009-002001002001">
               <description>
                  <p>Optional description.</p>
                  <p><i>Possibly a duplicated description of what was inherited.</i></p>
                  <p>Consumer-appropriate description of what may be inherited.</p>
                  <p>In the context of the application component in satisfaction of AC-2, part a.</p>
               </description>
            </inherited>

            <satisfied responsibility-uuid="11111111-0000-4000-9009-002001002002">
               <description>
                  <p>Description of how the responsibility was satisfied.</p>
               </description>
            </satisfied>

         </by-component>

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

Leveraging System Receiving Inheritance

- `uuid` values assigned within leverag**ed** system's SSP begin with "11111111".
- `uuid` values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraging System (inheriting the capability of a component)

```
<control-implementation>
   <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-accc-d3e534932efb" />
   <implemented-requirement control-id="ac-2" uuid="uuid-value">
      <!-- details cut -->
      <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

         <by-component uuid="uuid-value" component-uuid="22222222-0000-4000-9001-000000000003">
            <description>
               <p><i>duplicated/tailored description of what was inherited,
                     and description of what was configured.</i></p>
               <p>Consumer-appropriate description of what may be inherited.</p>
               <p>In the context of the application component in satisfaction of AC-2, part a.</p>
            </description>

            <inherited provided-uuid="11111111-0000-4000-9009-002001002001">
               <description>
                  <p>Optional description.</p>
                  <p><i>Possibly a duplicated description of what was inherited.</i></p>
                  <p>Consumer-appropriate description of what may be inherited.</p>
                  <p>In the context of the application component in satisfaction of AC-2, part a.</p>
               </description>
            </inherited>

            <satisfied responsibility-uuid="11111111-0000-4000-9009-002001002002">
               <description>
                  <p>Description of how the responsibility was satisfied.</p>
               </description>
            </satisfied>

         </by-component>

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

Within the statement assembly, the by-component assembly references the component in the leveraging system's SSP that represents the leveraged system's application.

Leveraging System Receiving Inheritance

- `uuid` values assigned within leverag**ed** system's SSP begin with "11111111".
- `uuid` values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraging System (inheriting the capability of a component)

```
<control-implementation>
   <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-accc-d3e534932efb" />
   <implemented-requirement control-id="ac-2" uuid="uuid-value">
      <!-- details cut -->
      <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

         <by-component uuid="uuid-value" component-uuid="22222222-0000-4000-9001-000000000003">
            <description>
               <p><i>duplicated/tailored description of what was inherited,
                     and description of what was configured.</i></p>
               <p>Consumer-appropriate description of what may be inherited.</p>
               <p>In the context of the application component in satisfaction of AC-2, part a.</p>
            </description>

            <inherited provided-uuid="11111111-0000-4000-9009-002001002001">
               <description>
                  <p>Optional description.</p>
                  <p><i>Possibly a duplicated description of what was inherited.</i></p>
                  <p>Consumer-appropriate description of what may be inherited.</p>
                  <p>In the context of the application component in satisfaction of AC-2, part a.</p>
               </description>
            </inherited>

            <satisfied responsibility-uuid="11111111-0000-4000-9009-002001002002">
               <description>
                  <p>Description of how the responsibility was satisfied.</p>
               </description>
            </satisfied>

         </by-component>

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

For inheritance, the description from the leveraged system's "provided" statement may simply be duplicated into the leveraging system's description.

It may also be tailored or completely replaced if appropriate.

Leveraging System Receiving Inheritance

- `uuid` values assigned within leverag**ed** system's SSP begin with "11111111".
- `uuid` values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraging System (inheriting the capability of a component)

```
<control-implementation>
   <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-accc-d3e534932efb" />
   <implemented-requirement control-id="ac-2" uuid="uuid-value">
      <!-- details cut -->
      <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

         <by-component uuid="uuid-value" component-uuid="22222222-0000-4000-9001-000000000003">
            <description>
               <p><i>duplicated/tailored description of what was inherited,
                     and description of what was configured.</i></p>
               <p>Consumer-appropriate description of what may be inherited.</p>
               <p>In the context of the application component in satisfaction of AC-2, part a.</p>
            </description>

            <inherited provided-uuid="11111111-0000-4000-9009-002001002001">
               <description>
                  <p>Optional description.</p>
                  <p><i>Possibly a duplicated description of what was inherited.</i></p>
                  <p>Consumer-appropriate description of what may be inherited.</p>
                  <p>In the context of the application component in satisfaction of AC-2, part a.</p>
               </description>
            </inherited>

            <satisfied responsibility-uuid="11111111-0000-4000-9009-002001002002">
               <description>
                  <p>Description of how the responsibility was satisfied.</p>
               </description>
            </satisfied>

         </by-component>

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

The "inherited" field links this to the original "provided" statement in the leveraged system's SSP.

Leveraging System Receiving Inheritance

- `uuid` values assigned within leverag**ed** system's SSP begin with "11111111".
- `uuid` values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraging System (inheriting the capability of a component)

```
<control-implementation>
   <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-accc-d3e534932efb" />
   <implemented-requirement control-id="ac-2" uuid="uuid-value">
      <!-- details cut -->
      <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

         <by-component uuid="uuid-value" component-uuid="22222222-0000-4000-9001-000000000003">
            <description>
               <p><i>duplicated/tailored description of what was inherited,
                     and description of what was configured.</i></p>
               <p>Consumer-appropriate description of what may be inherited.</p>
               <p>In the context of the application component in satisfaction of AC-2, part a.</p>
            </description>

            <inherited uuid="uuid-value" provided-uuid="11111111-0000-4000-9009-002001002001">
               <description>
                  <p>Optional description.</p>
                  <p><i>Possibly a duplicated description of what was inherited.</i></p>
                  <p>Consumer-appropriate description of what may be inherited.</p>
                  <p>In the context of the application component in satisfaction of AC-2, part a.</p>
               </description>
            </inherited>

            <satisfied uuid="uuid-value" responsibility-uuid="11111111-0000-4000-9009-002001002002">
               <description>
                  <p>Description of how the responsibility was satisfied.</p>
               </description>
            </satisfied>

         </by-component>

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

The original description from the "provided" statement should be duplicated here and should not be adjusted.

Leveraging System Receiving Inheritance

- `uuid` values assigned within leverag**ed** system's SSP begin with "11111111".
- `uuid` values assigned within leverag**ing** system's SSP begin with "22222222".

Leveraging System (inheriting the capability of a component)

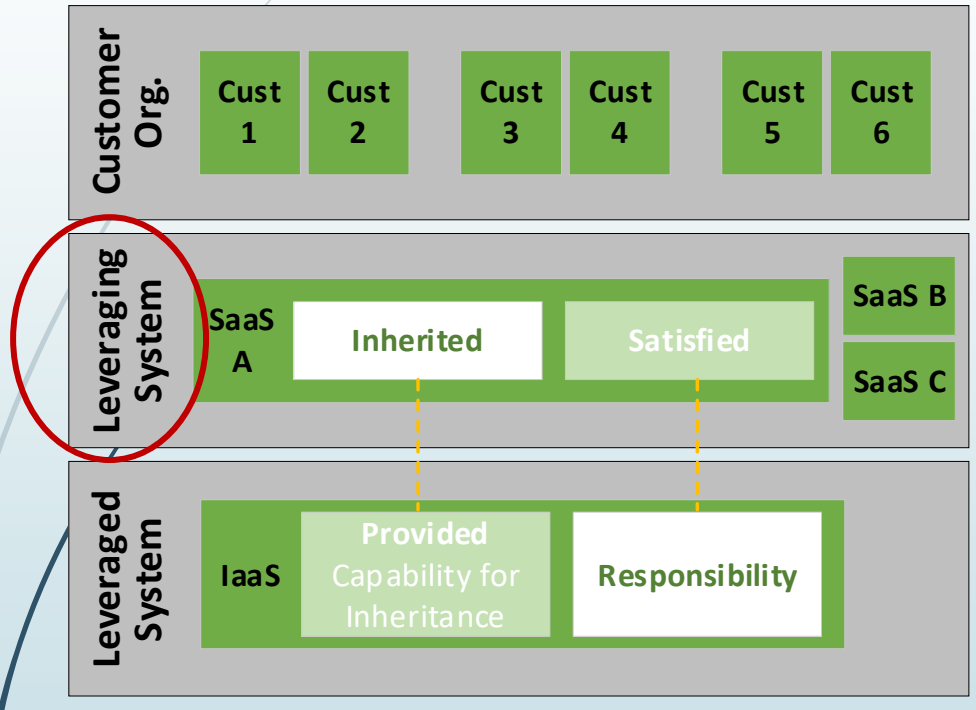
```
<control-implementation>
   <implemented-requirement control-id="ac-1" uuid="eee8697a-bc39-45aa-accc-d3e534932efb" />
   <implemented-requirement control-id="ac-2" uuid="uuid-value">
      <!-- details cut -->
      <statement statement-id="ac-2_stmt.a" uuid="uuid-value">

         <by-component uuid="uuid-value" component-uuid="22222222-0000-4000-9001-000000000003">
            <description>
               <p><i>duplicated/tailored description of what was inherited,
                     and description of what was configured.</i></p>
               <p>Consumer-appropriate description of what may be inherited.</p>
               <p>In the context of the application component in satisfaction of AC-2, part a.</p>
            </description>

            <inherited provided-uuid="11111111-0000-4000-9009-002001002001">
               <description>
                  <p>Optional description.</p>
                  <p><i>Possibly a duplicated description of what was inherited.</i></p>
                  <p>Consumer-appropriate description of what may be inherited.</p>
                  <p>In the context of the application component in satisfaction of AC-2, part a.</p>
               </description>
            </inherited>

            <satisfied responsibility-uuid="11111111-0000-4000-9009-002001002002">
               <description>
                  <p>Description of how the responsibility was satisfied.</p>
               </description>
            </satisfied>

         </by-component>

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

If a "responsibility" statement was associated with this inherited capability, it is also addressed here with a "satisfied" statement.

The "responsibility-uuid" links to the original "responsibility" statement in the leveraged system's SSP using its original UUID value.

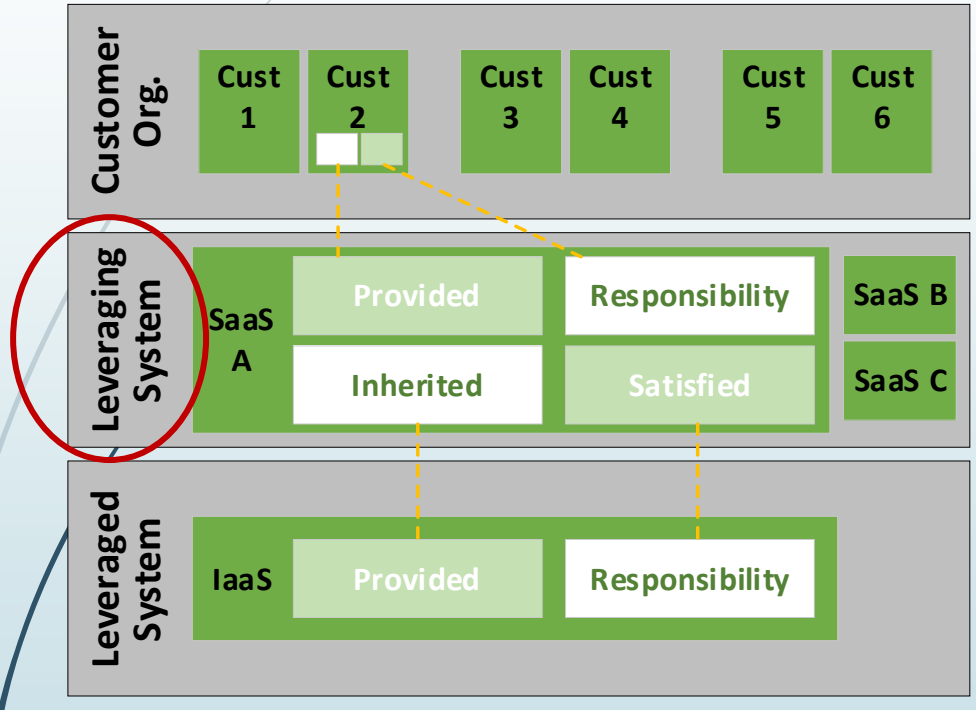
The linkage between "responsibility" and "provided" is maintained in the leveraged system's SSP and is not referenced here.

Handling "provided" and "responsibility"



- **Leveraging System:**
  - **Inheritance of each "provided" capability is at the discretion of the system owner.**
    The leveraging system owner may either:
    - inherit the provided capability; or
    - address the control directly as if no inheritance is provided.
  - **If the leveraging system owner elects to inherit a "provided capability:**
    - the component providing the inherited capability **must** be defined in the leveraging SSP
      - including a property that identifies the original uuid of the component in the leveraged system's SSP;
    - the control part being satisfied by inheritance **must** include a by-component assembly that links to the inherited component; and
    - any "responsibility" associated with the "provided" capability **must** be addressed.
  - A "responsibility" statement linked to a "provided" capability is ignored if the leveraging system owner elects not to inherit the capability
  - Every "responsibility" not linked to a "provided" capability **must** be addressed.

When a Leveraging System is also a Leveraged System

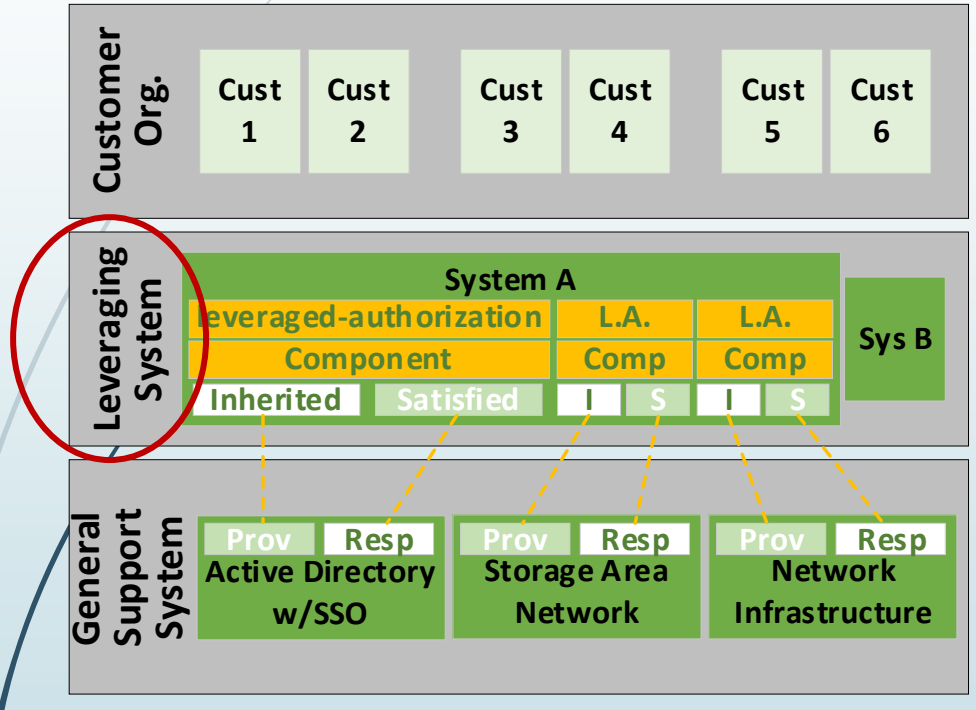


- **Leveraging System:**
  - The leverag**ing** system's SSP should:
    - identify what **is** inherited from a leverag**ed** system
    - identify any addressed responsibilities (as identified by the leveraged system)

In addition to:

- identifying what **may be** inherited by the leverag**ing** system's customers
- any responsibilities the leverag**ing** system's customers must address to fully satisfy a control

When a Leveraging System has more than one Leveraged System



- **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.

Questions?
Thank you!

We want your feedback!

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

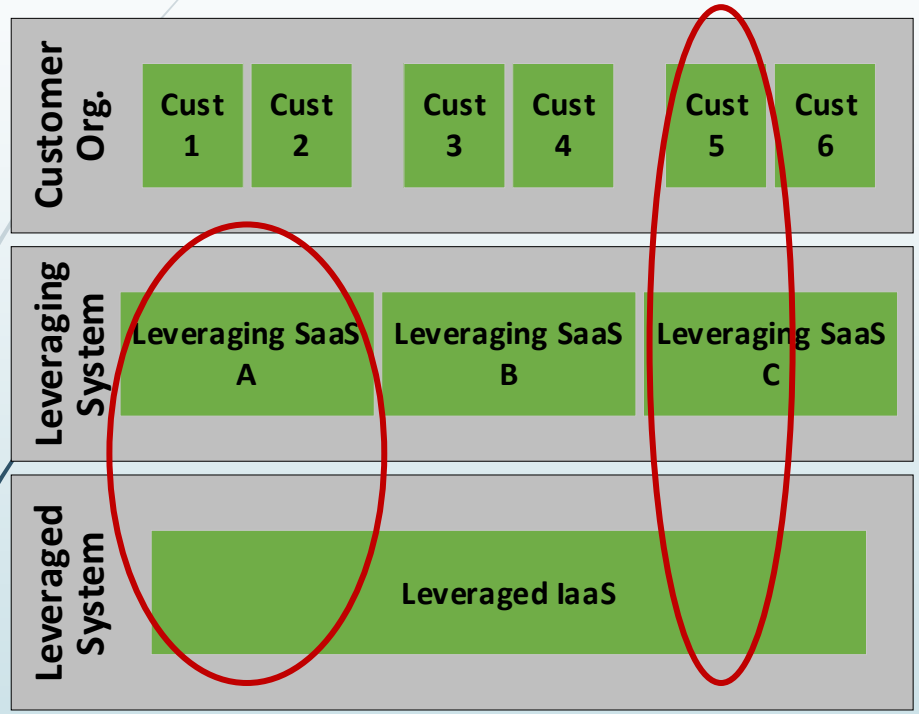
Project Website:
https://www.nist.gov/oscal

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

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

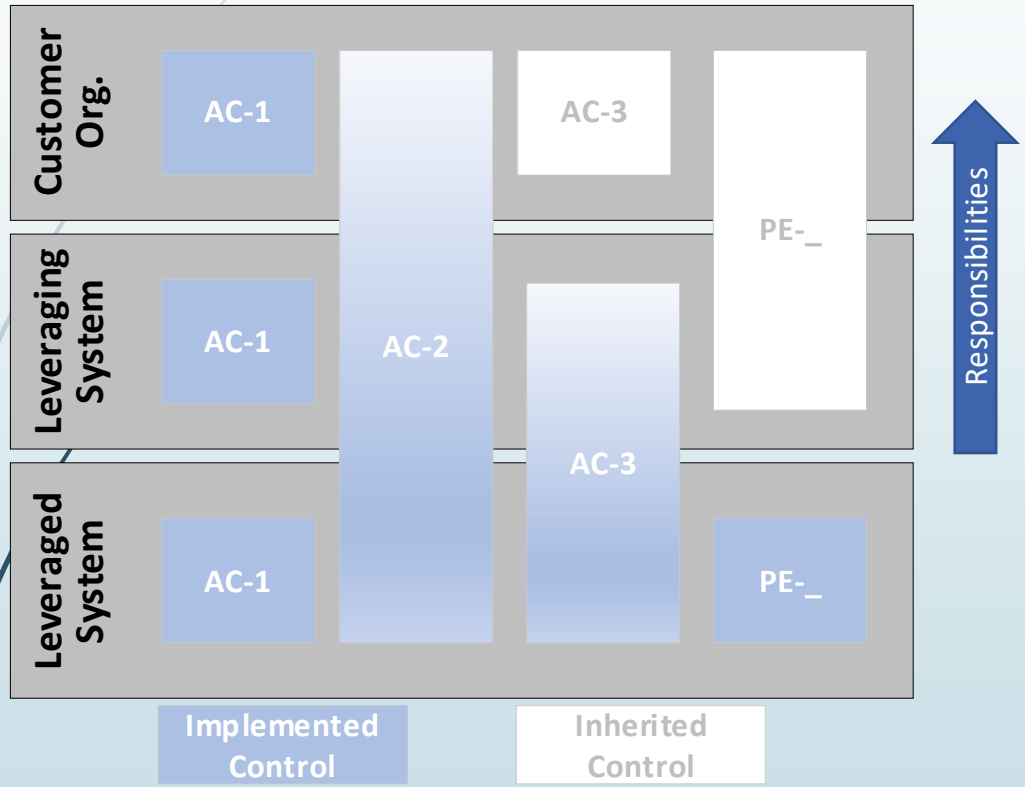
BACKUP SLIDE(S)

Responsibility and Adjudication



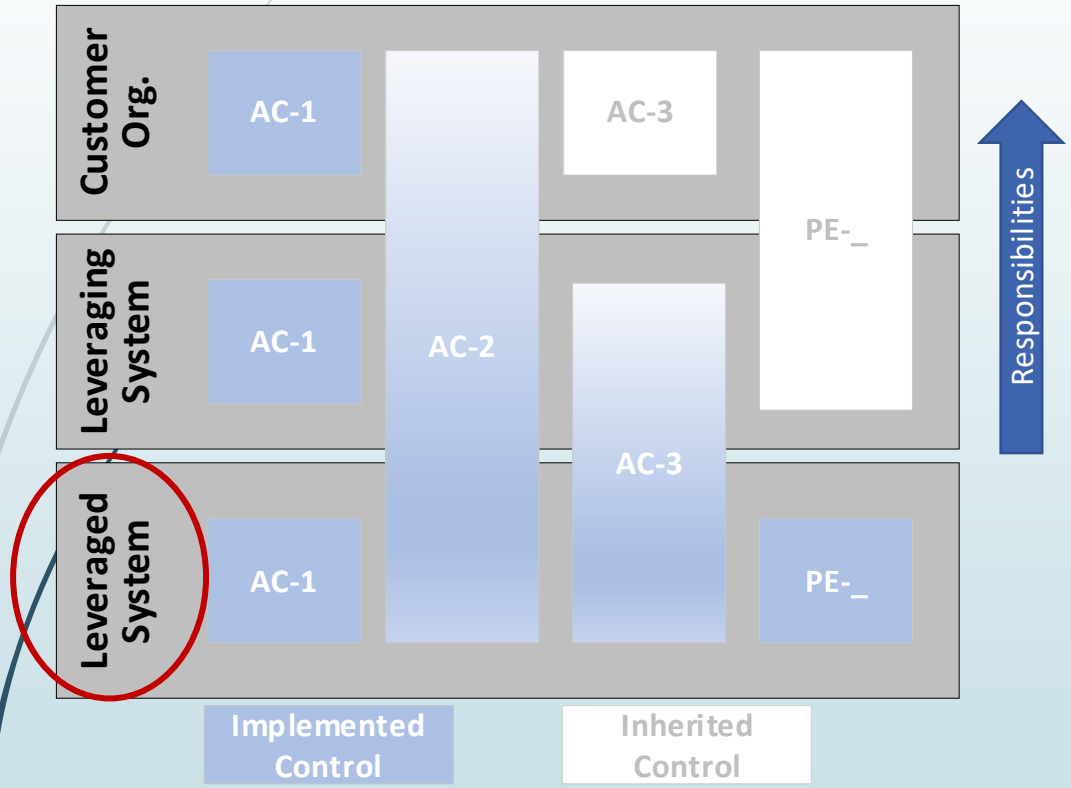
- **An authorizing official (AO) must adjudicate the entire stack - relative to the authorization they are issuing. (Holistic View)**
- **Each system owner is responsible for their system in the stack. (System View)**
- **Examples:**
  - The AO for *Leveraging SaaS A* must adjudicate its authorization in consideration of the controls within both:
    - *Leveraging SaaS A*; and
    - the Leveraged IaaS.
  - The AO for *Cust 5* must adjudicate its authorization in consideration of the controls implemented:
    - by the *Cust 5*;
    - within the *Leveraging SaaS C*; and
    - within the *Leveraged IaaS*.

Full Control Satisfaction (Holistic View)



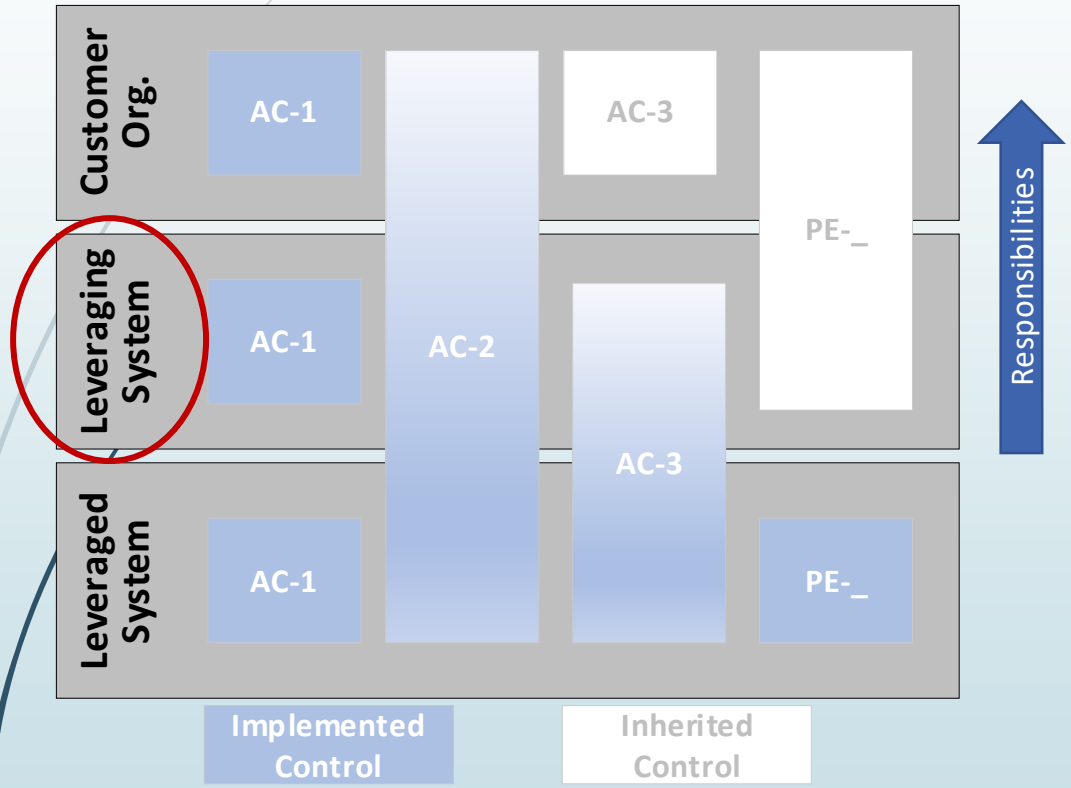
- **Some controls must be satisfied independently by each system**
  - Example: FedRAMP does not allow inheritance for XX-1 controls.
- **Some controls are only fully satisfied if each system does their part.**
  - Example: Logical access control must be implemented on all components in "the stack".
- **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.

Full Control Satisfaction (System View)



- **Leveraged System:**
  - The leveraged system may address each control, through some combination of:
    - implementing the control
      - privately; or
      - available for inheritance;
    - defining a customer responsibility for the control.

Full Control Satisfaction (System View)

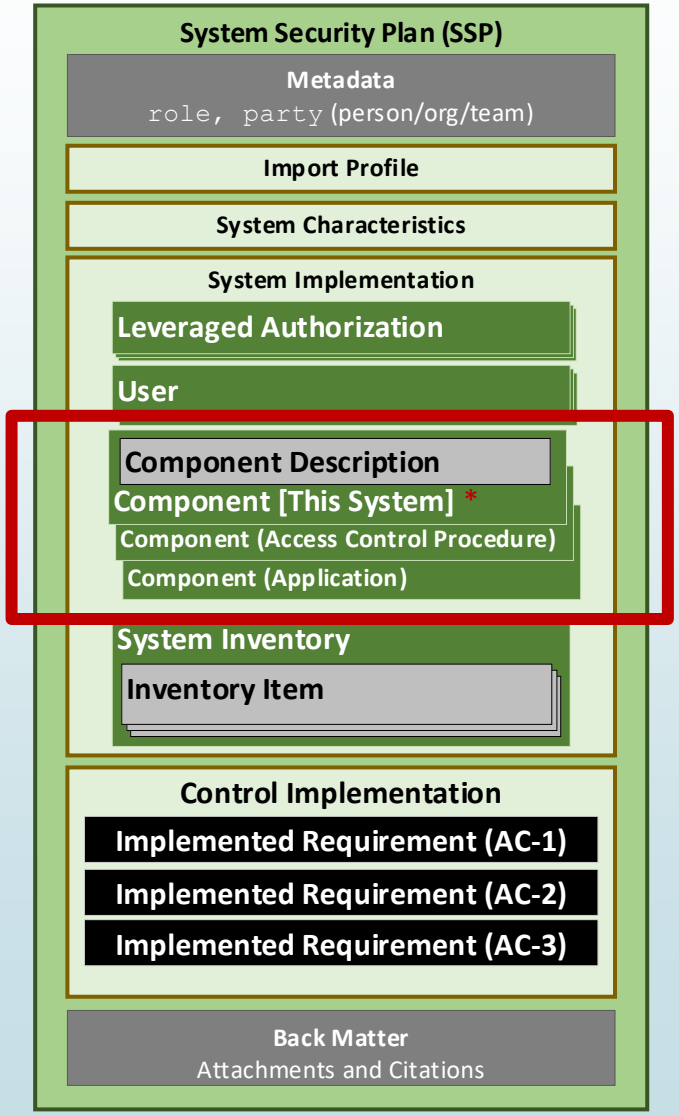


- **Leveraging System:**
  - The leveraging system may address each control, through some combination of:
    - **identifying an inherited control from a leveraged system;**
    - implementing the control; or
    - defining a customer responsibility for the control

Responding to Controls in the SSP: Define Components

- Each control response is broken down to the individual components involved.
- Enables a more robust response to controls

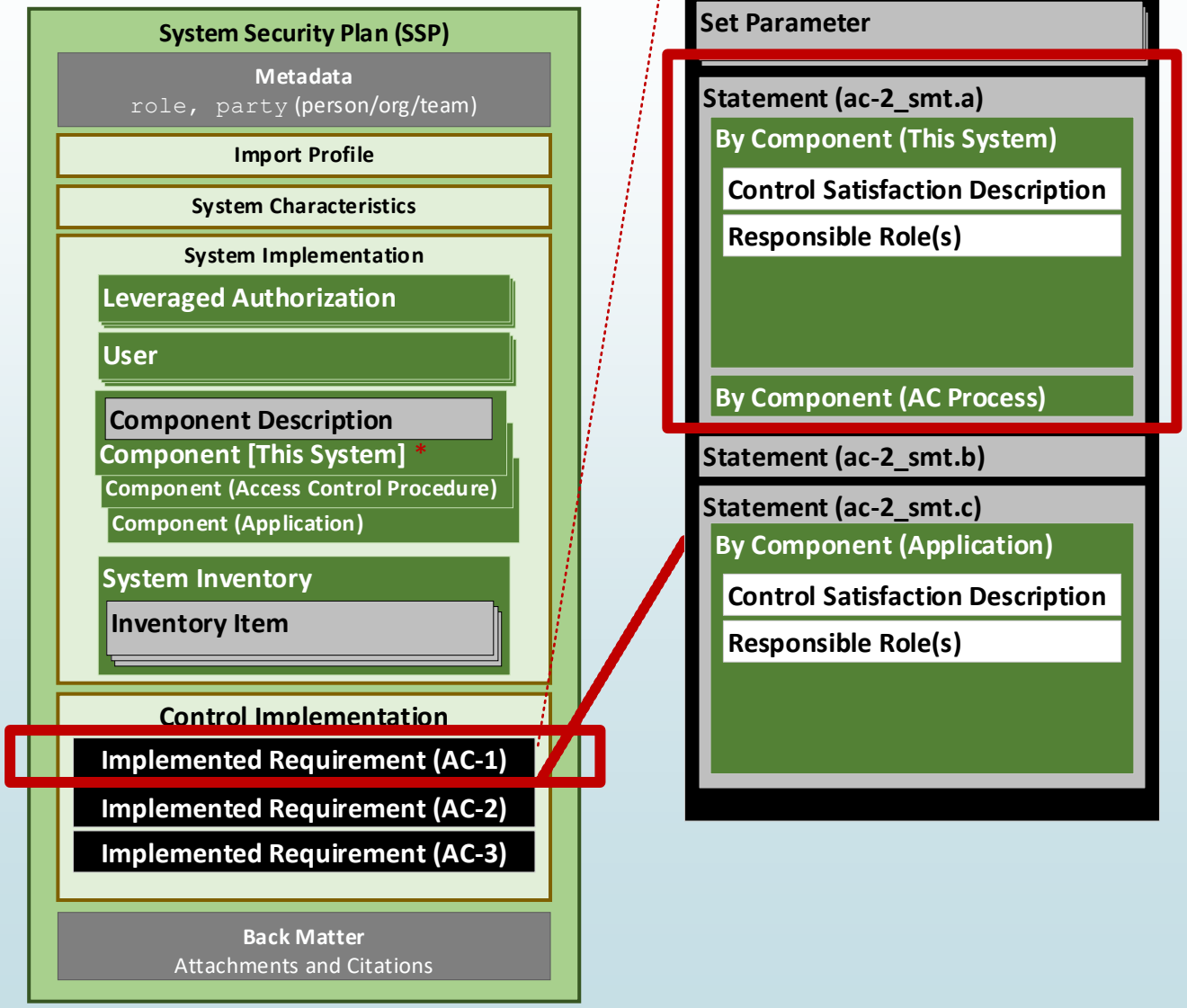
- Example: The access control implementation that satisfies *AC-2, part a* is described separately for:
  - This System
  - The Access Control Procedure
  - A shared Application



- Components are defined in the `system-implementation` assembly.
- One `component` assembly for each component.
- There must always be a "This System" component defined.
- Other components are defined as appropriate.
- SSP authors have flexibility in how granular they define components.

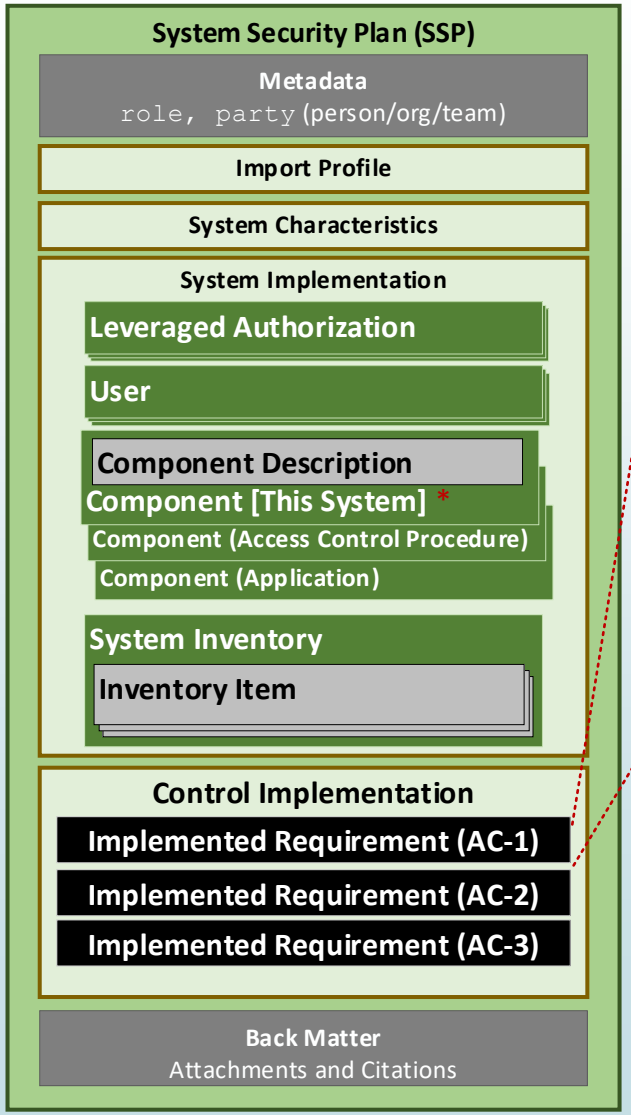
Responding to Controls in the SSP: Respond By Component

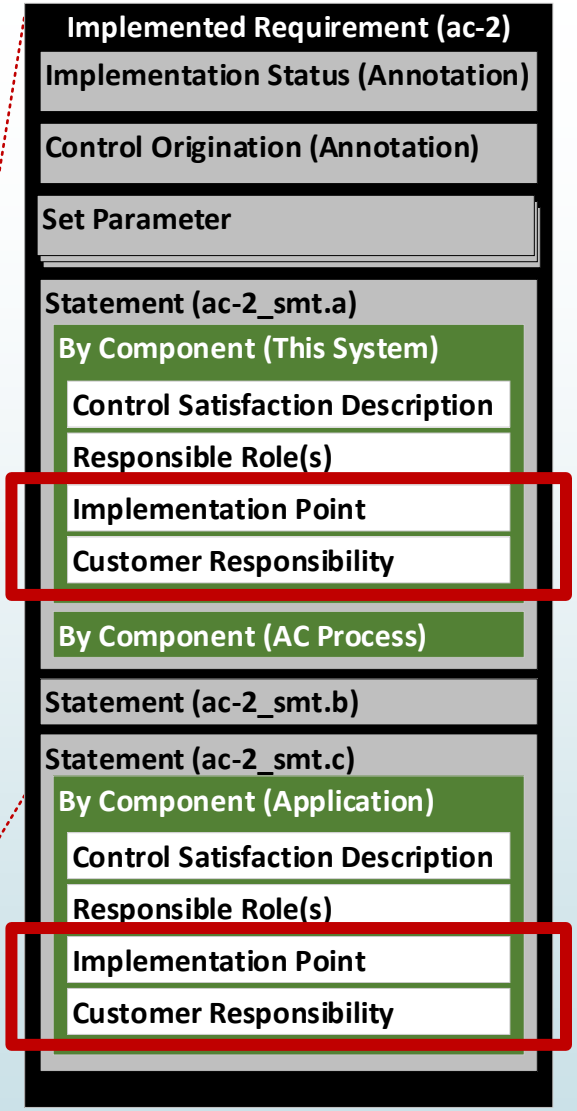
- For each control there is an `implemented-requirement` assembly.
- Within each `implemented-requirement` assembly, there are one or more `statement` assemblies.
- Each `statement` assembly has one or more `by-component` assemblies. Each references a component involved with control satisfaction.
- Control satisfaction responses are provided in the `description` field within each `by-component` assembly.
- NOTE: 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 applicable `by-component` assembly using an `annotation`.
- If the customer has a responsibility within the application, there should be a `by-component` assembly in the `statement` assembly, which identifies the application and includes the customer responsibility `annotation`.
- If a customer responsibility statement does not fit any specific component, place it in the "This System" component.



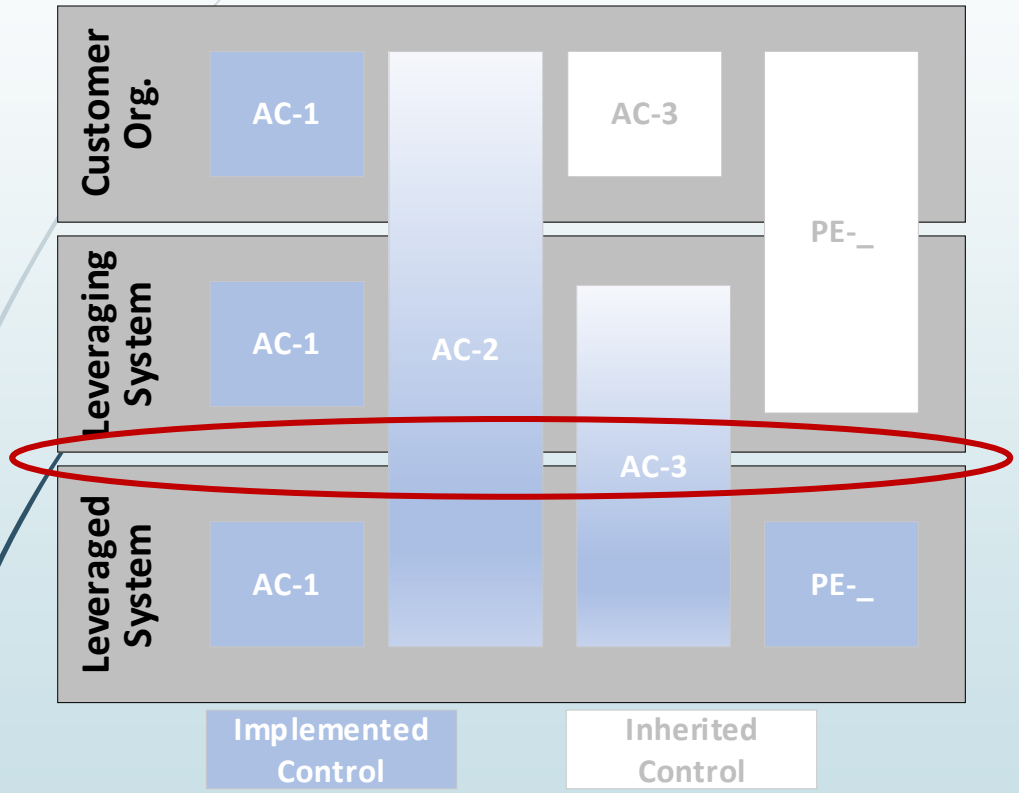


Looking at the OSCAL (Components)

Leveraged System

```
<system-implementation>
   <user />
   <component uuid="11111111-0000-4000-9001-000000000001" component-type="system">
      <title>This System</title>
      <description>
         <p>This Leveraged IaaS.</p>
         <p>The entire system as depicted in the system authorization boundary</p>
      </description>
      <status state="operational"/>
   </component>

   <component uuid="11111111-0000-4000-9001-000000000002" component-type="procedure">
      <title>Access Control Procedure</title>
      <description>
         <p>This is the procedure that governs access to the application.</p>
      </description>
      <link href="#8b9d82a9-dd49-4309-a466-685b0081f28c"/>
      <status state="operational"/>
   </component>

   <component uuid="11111111-0000-4000-9001-000000000003" component-type="software">
      <title>Application</title>
      <description>
         <p>An application within the IaaS, exposed to SaaS customers and their downstream customers.</p>
         <p>This Leveraged IaaS maintains aspects of the application.</p>
         <p>The Leveraging SaaS maintains aspects of their assigned portion of the application.</p>
         <p>The customers of the Leveraging SaaS maintain aspects of their sub-assigned portions of the application.</p>
      </description>
      <status state="operational"/>
      <responsible-role role-id="admin">
         <party-uuid>11111111-0000-4000-9000-100000000001</party-uuid>
      </responsible-role>
   </component>

</system-implementation>
```

Relationship View: SSP Documentation



- **The Leveraged System's SSP:**
  - may provide information about controls that may be inherited by a leveraging system
  - must explicitly identify all customer responsibilities required to fully satisfy a control
    - The number of levels beyond the leveraging system is irrelevant
- **The Leveraging System's SSP:**
  - must identify what is inherited from the leveraged system
  - must address control requirements not explicitly satisfied through inheritance
  - should link customer responsibilities identified by its leveraged system to:
    - control implementation statements
    - customer responsibilities the leveraging system defined for its downstream customers

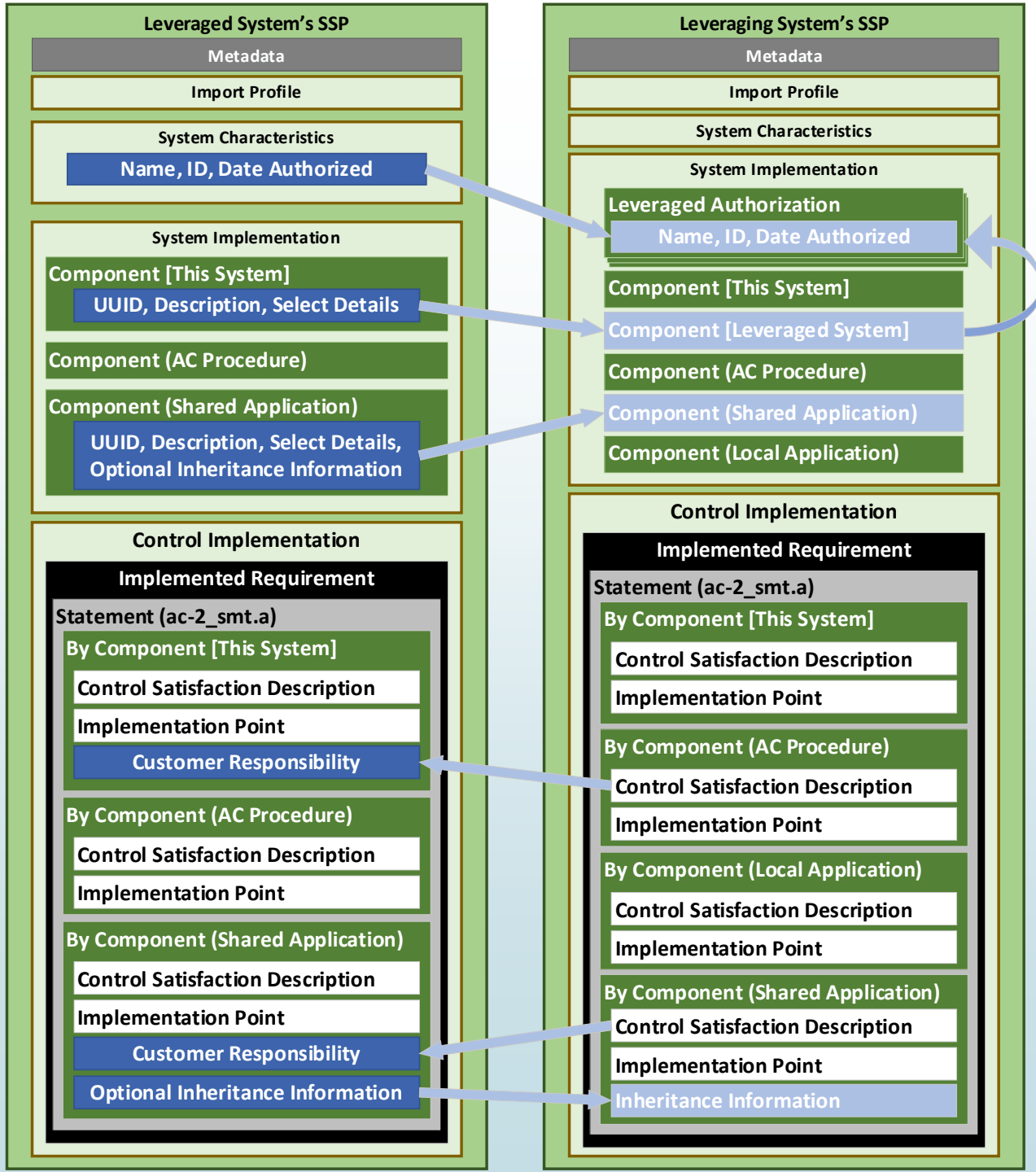
Leveraged System -> Leveraging System Use Cases

- The Leveraged System has an application exposed to the Leveraging System
  - The customer configuration responsibilities are defined within *AC-2, part a*; within a by-component assembly associated with the application
  - An optional inheritance statement is defined within *AC-2, part a*; within a by-component assembly associated with the application. It describes additional aspects of *AC-2, part a* addressed by the application with no customer requirement.
  - The component definition for the application is communicated to the leveraging system
- The Leveraged System has an access control procedure
  - The procedure is only for the leveraged system. The leveraging system requires its own procedure to satisfy AC-2, part a.
  - A customer responsibility statement is made with within *AC-2, part a*; within a by-component assembly associated with "This System" describing the need for the customer to create their own access control procedure.
  - In this instance it does not make sense to include the component representing the leveraged system's access control procedure.

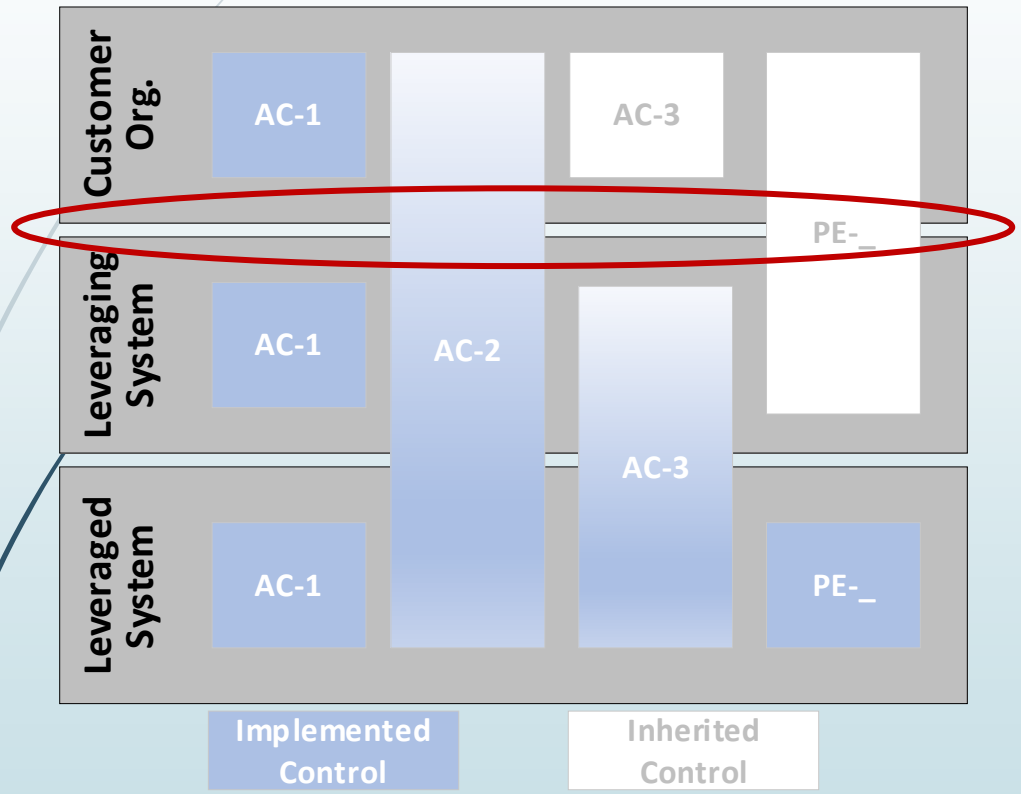
Leveraging System

A leveraging system must communicate the following to customers and AOs:

- Information about the authorizations for both the Leveraging and Leveraged Systems (dates, system IDs, etc.)
- Control Satisfaction Descriptions that satisfy a customer responsibility statement
- Statements about what the leveraging system has inherited from the leveraged system
  - In the component definition; and/or
  - In the by-component response to a specific control/part
- Component information from the leveraged system must be referenced in the leveraging system
- End Consumer (Customer) responsibility statements may also be defined the same way the leveraged system defines them



Relationship Views: Simplify and Modularize

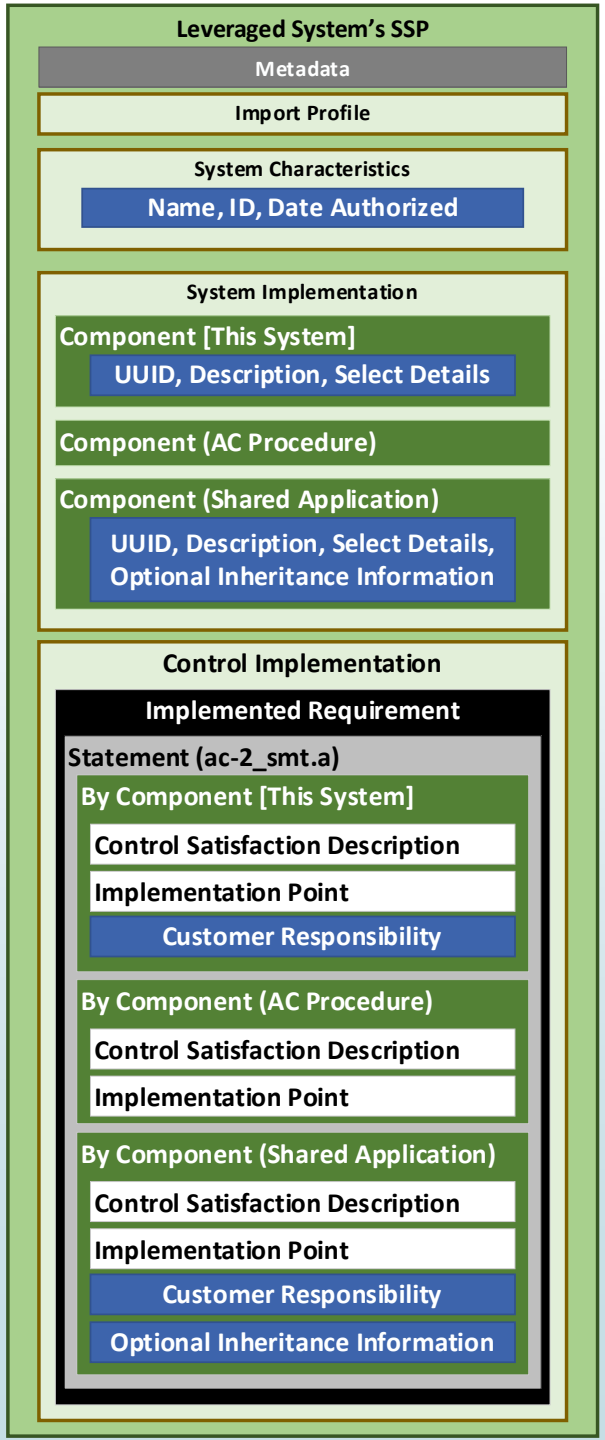


- **For additional layers:**
  - The leveraging system becomes the leveraged system relative to the customer layer
  - In addition to
    - information controls that may be inherited by a leveraging system
    - explicit customer responsibilities required to fully satisfy a control
  - The number of levels beyond the leveraging system is irrelevant.

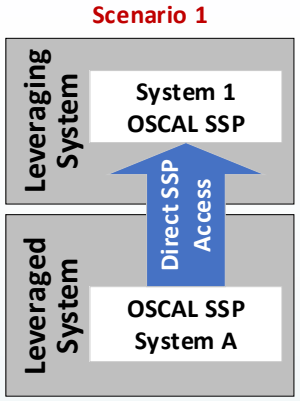
Leveraged System

A leveraged system must communicate the following to a leveraging system:

- Information about the Leveraged System's authorization (date, system ID, etc.)
- Consumer (Customer) responsibility statements
  - In the by-component response to a specific control/part
    - System-wide statements - associated with the by-component statement for "This System"
    - Component-specific statements
- Statements about what the leveraging system could inherited
  - In the component definition; and/or
  - In the by-component response to a specific control/part
- Certain information about any component associated with consumer responsibility or inheritance statements

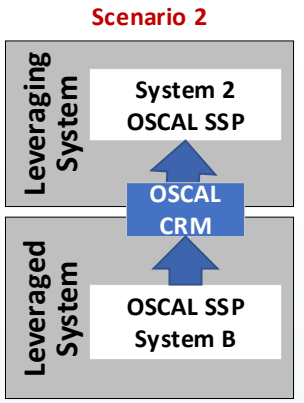


Scenario 1: OSCAL SSP With Access



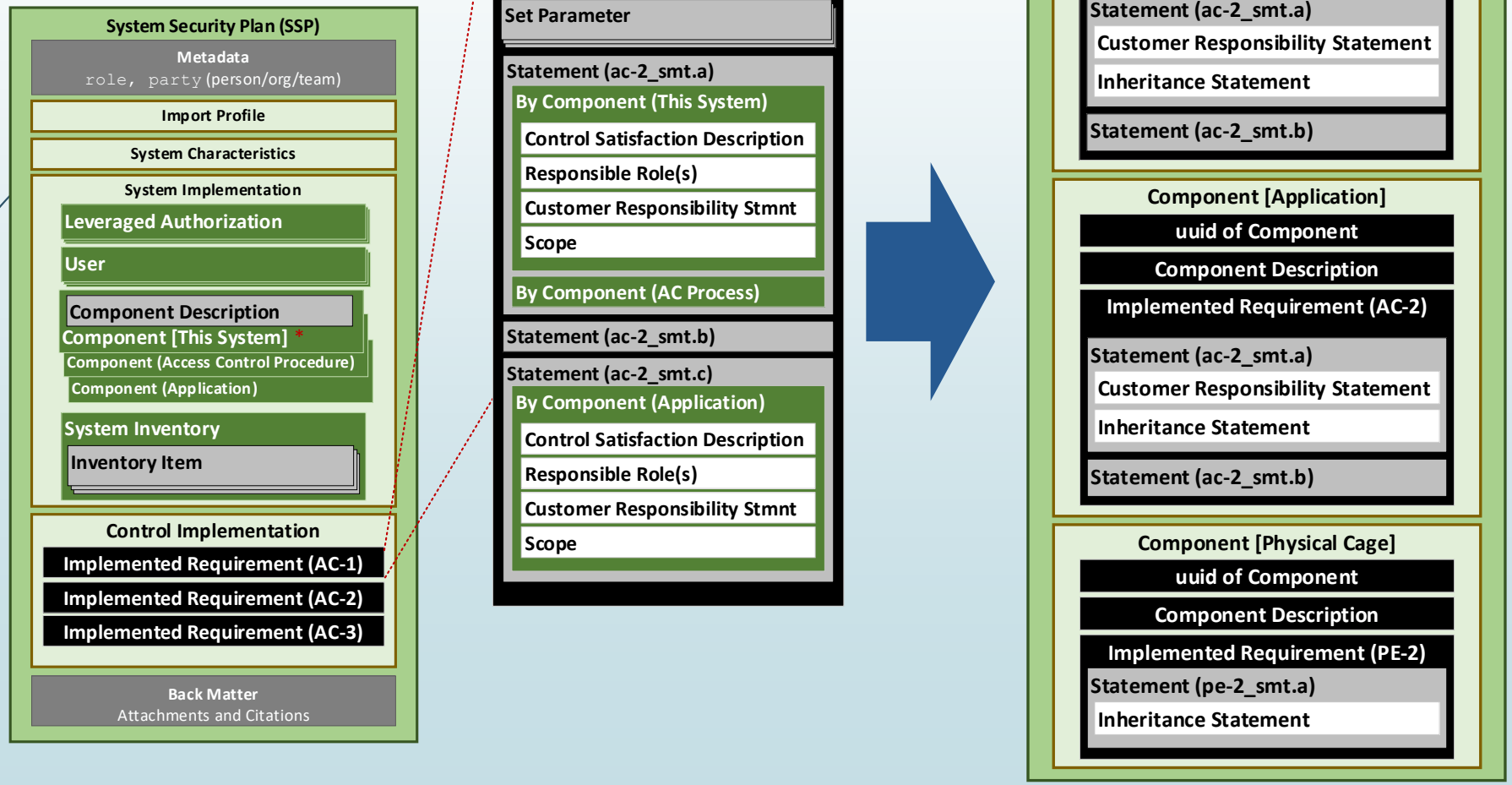
- Preferred scenario
- The SSP of the leveraging system can "see" the leveraged system's SSP
- Tools can identify which statements in the leveraged system's SSP have a customer responsibility
- Tools can further identify the leveraged system's components associated with these customer responsibility 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

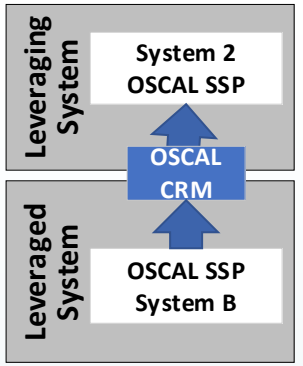


- The SSP of the leveraging system is not permitted to "see" the full leveraged system's SSP.
- The leveraged system's owner, creates an OSCAL customer responsibility matrix (CRM), using the OSCAL Component model.
- Every component in the leveraged system's SSP, with a customer responsibility annotation 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

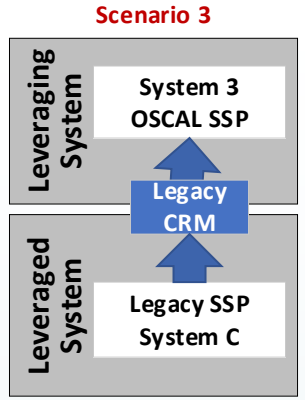


Scenario 2

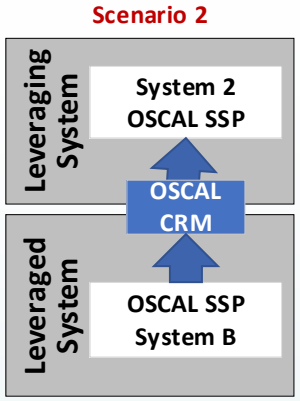


Scenario 3: Legacy SSP or CRM

- 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.