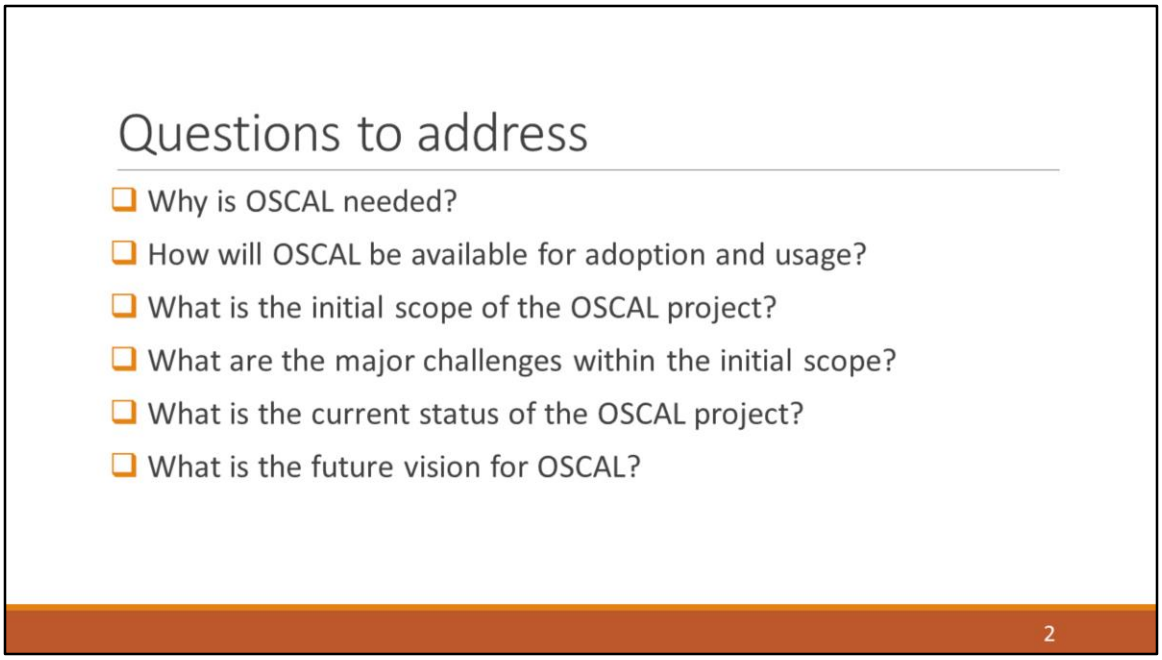
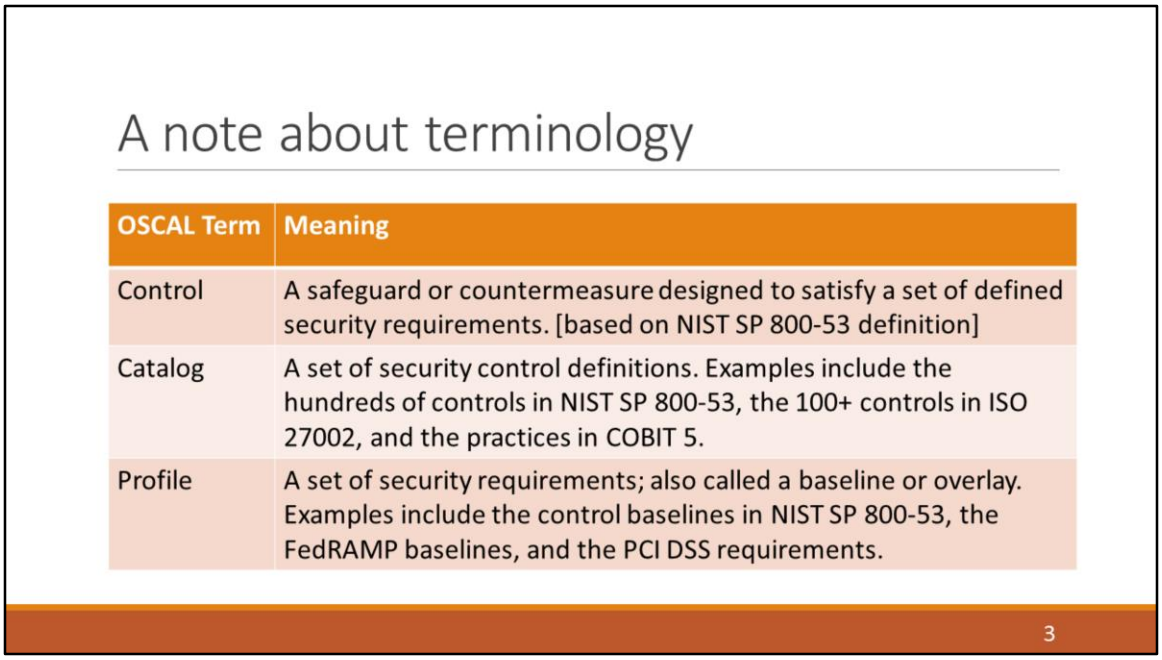
Overview of the Open Security Controls Assessment Language (OSCAL)

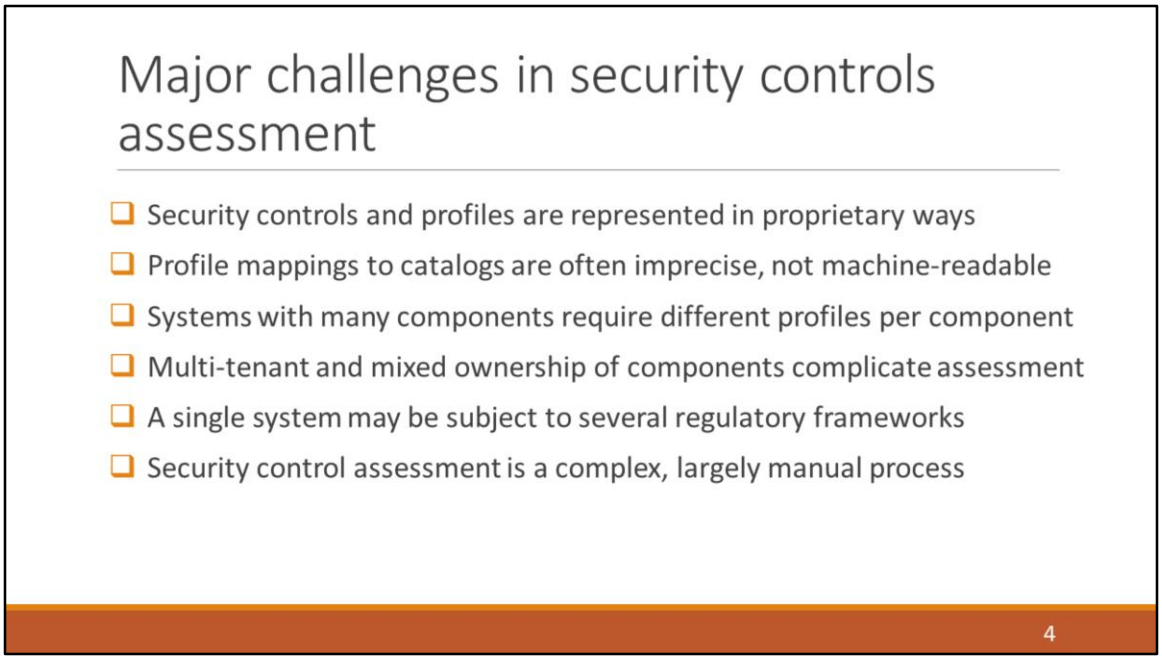
AUGUST 10, 2017



Today I will address a handful of questions, talking about why we're doing OSCAL, who will benefit from the work we're doing, what we're currently working on, the challenges we're currently wrangling with, and where we plan to go with the OSCAL project.



We're using a handful of terms in the presentation. *Control* refers to any safeguard or countermeasure that's designed to satisfy a set of security and potentially privacy requirements. While this is based on the 800-53 definition, when we talk about control, we're talking about a similar kind of requirement from a control catalog. When we talk about a *control catalog*, we're talking about a list of security control definitions. For our work so far, we've been dealing with three control catalogs: NIST SP 800-53, ISO 27001/2, and COBIT 5. Another point of clarification is when we talk about the concept of a profile. Examples are included in NIST SP 800-53, FedRAMP, and PCI DSS. The profile concept is often referred to as a baseline or overlay. We're calling these a profile for OSCAL. A *profile* is basically selecting a set of security requirements from one or more control catalogs.



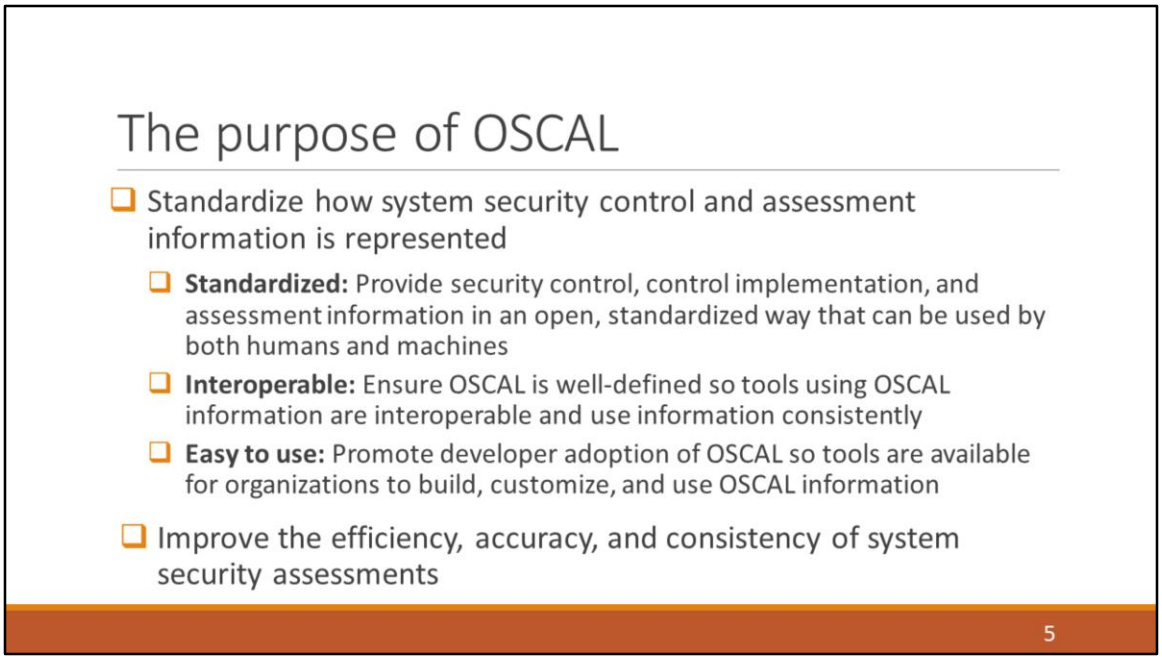
In OSCAL we're trying to deal with a number of challenges around security controls and security controls assessment. The core challenge, and one of the primary reasons why we're creating OSCAL, is that concepts like security controls and profiles are represented today largely in proprietary ways. In many cases they are written in prose documents that are imprecise, lead to differences in interpretation, and are not machine-readable, meaning that the prose instructions require someone to do data entry into a tool in order for the tool to use the information.

We're also struggling with information systems that have many different components, and some components require the use of different profiles per component, which is commonly the case with cloud environments. Also, the cloud environments can be multitenant or have mixed ownership of components. We need to be able to assess the security of these systems against a number of requirements, owners, etc.—to do that simultaneously and provide these views to stakeholders.

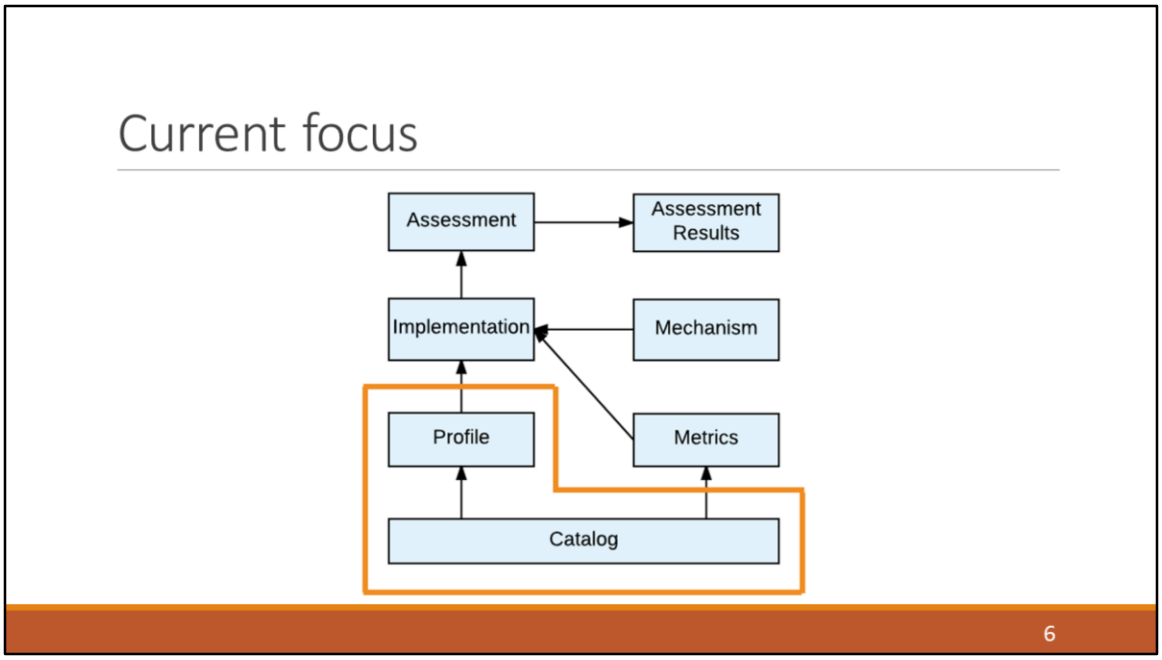
On top of that we have situations where a single system needs to support multiple regulatory frameworks. For example, the VA is a federal agency (FISMA and NIST Cybersecurity Framework requirements) and a healthcare institution (HIPAA requirements) that has credit card transactions (PCI DSS). There is no shortage of

requirements for some organizations that have multiple regulatory frameworks.

Assessing all these security controls is extremely complex. Because of that complexity, it's largely a manual process today. With OSCAL we're trying to change that.



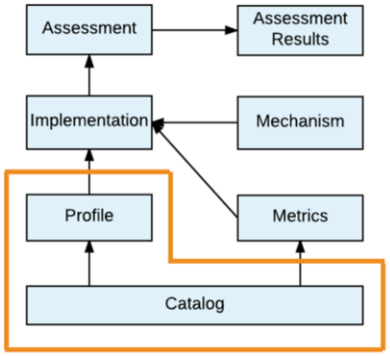
OSCAL is an attempt to standardize how security controls are represented, how you would represent a control implementation for a given system, and how that information is best used and reports are generated in a standardized way that can be used by both humans and machines. That means we want formats that can be generated by machines for communicating with other machines, but can also easily be reformatted so humans can read the information. By standardizing the representation of this information, we can make OSCAL information interoperable by having a well-defined specification with information that's going to be used, imported, and used interoperably for security control assessments. We are trying to keep OSCAL as simple as possible and provide a lot of automation for tools to use.



The current focus of our work is the catalog and profile layers.

Descriptions of current components

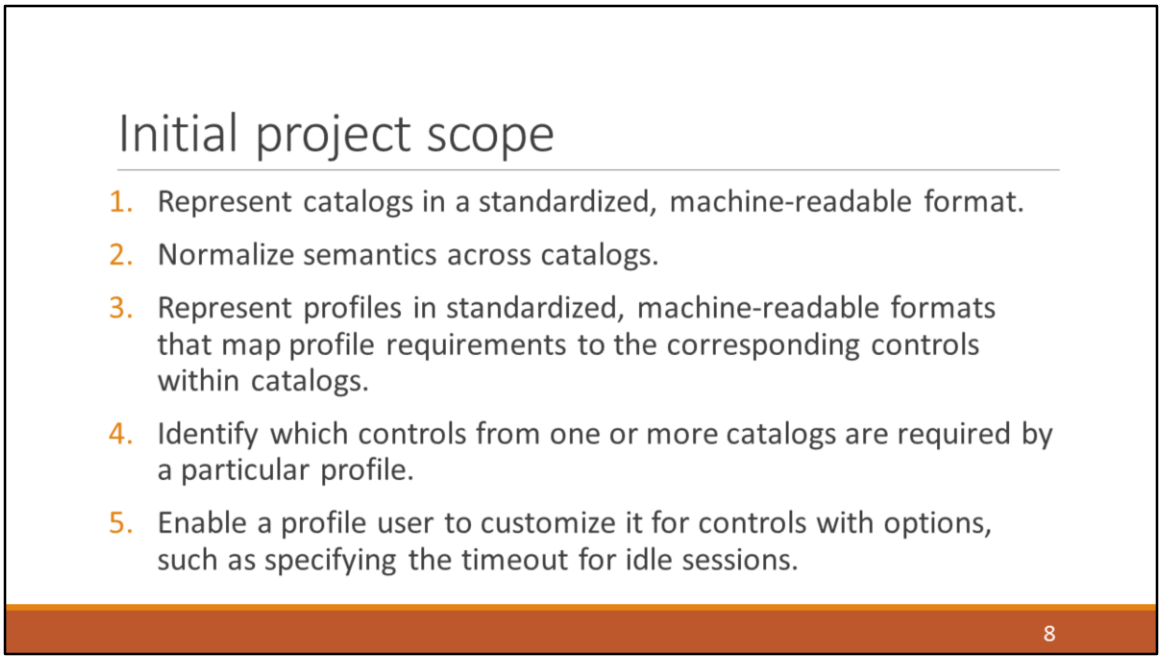
- **Catalog:** Defines a set of security controls (e.g., NIST SP 800-53 Appendix F); may also define objectives and methods for assessing the controls (e.g., NIST SP 800-53A)
- **Profile:** Defines a set of security requirements, where meeting each requirement necessitates implementing one or more security controls



7

We're working on developing models for how you represent a security control catalog, which is a collection of controls, and how you represent a profile, which is a collection of requirements. We're taking what is traditionally considered a control catalog and also optionally including methods for assessing those controls, something like the information from NIST SP 800-53A. This is a departure from what's been discussed before. This has been done because we've been looking at a number of control catalog formats. Some address assessment information directly, such as COBIT 5, while others have it separately, like 800-53A. Including assessment objectives within the OSCAL catalog model simplifies the OSCAL operational model.

We are also planning a format for profiles. This will allow for selecting security controls using a number of different mechanisms as well as tailoring those controls. A profile can include controls from more than one catalog. So an agency could have a single profile that references controls from several catalogs.

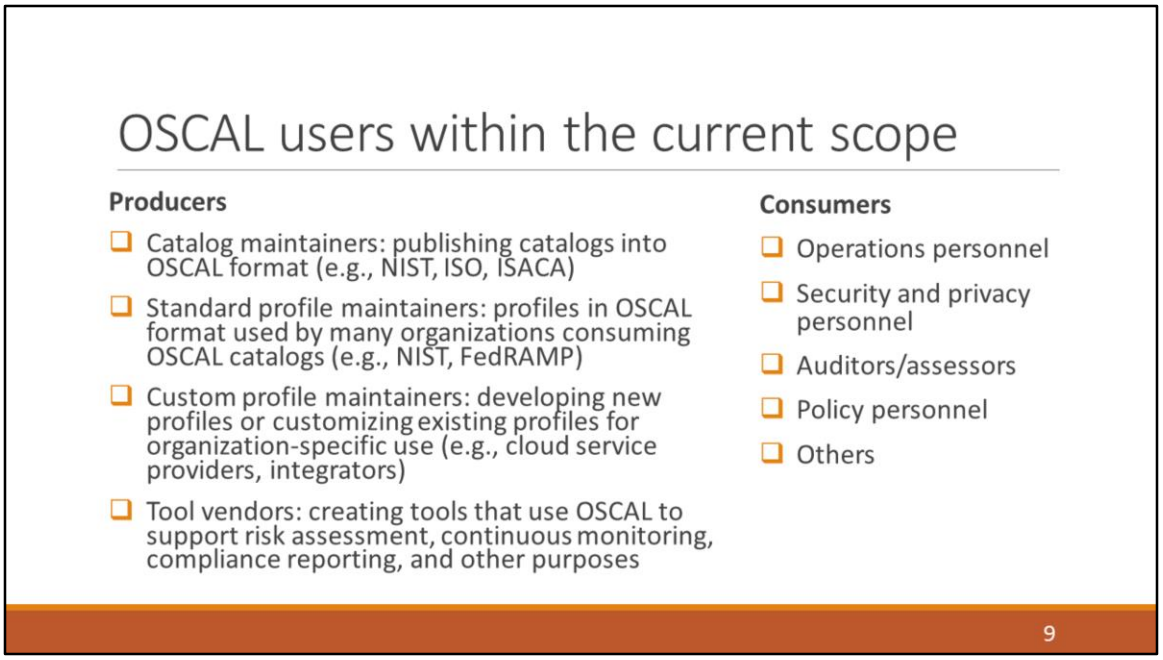


Let's talk about what we're currently doing. We've been primarily focused on representing catalogs in a standardized XML-based format. We're leveraging NIST SP 800-53, ISO 27001/2, and COBIT 5 as control catalogs. We're working to represent each of those catalogs in a common OSCAL format. By doing so, we're normalizing some of the terminology concepts that go across these various catalogs. Each catalog may refer to the same concept that other catalogs have but do so with different words. In OSCAL we use the same XML elements to describe these similar concepts. We feel like this semantic normalization provides significant initial value.

We are also working on a representation of profiles, also in standardized XML, that will allow us to map profile requirements to one or more controls in the underlying control catalog. The idea is that we can build a profile out of multiple control catalogs. We're working on terminology to customize controls through a profile to do things like assign a parameter value, modify a requirement, and similar kinds of things.

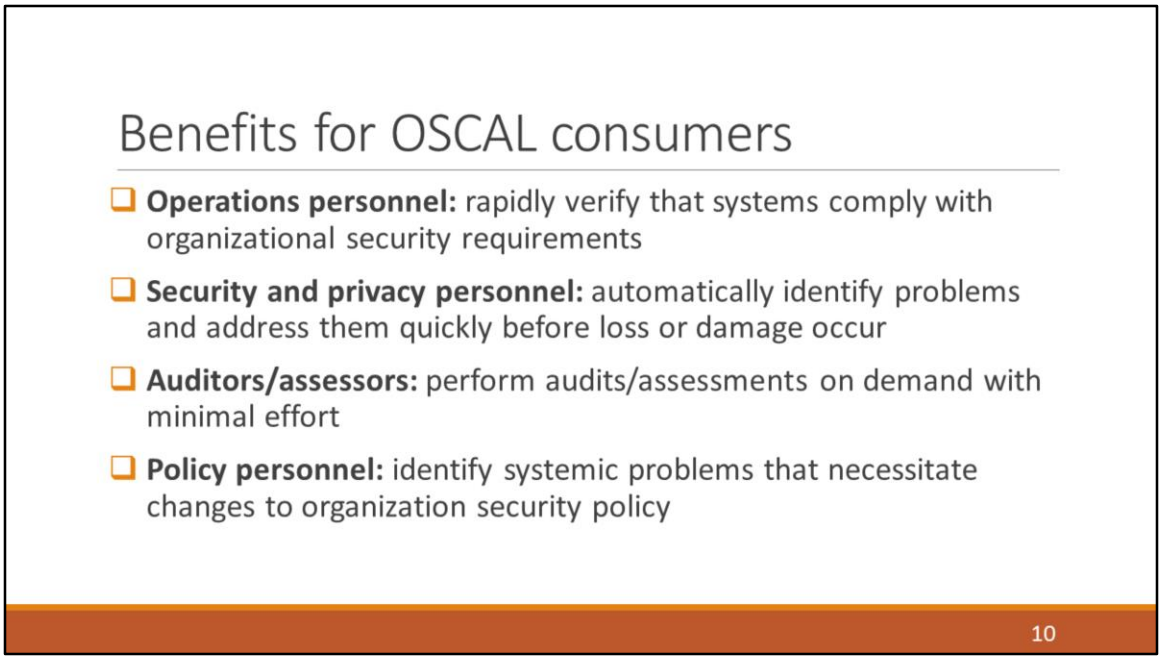
We're specifically focusing on the catalog and profile layers because we feel that these are the foundation of OSCAL. The OSCAL model includes implementation, assessment, metrics, etc., but the foundation is really catalog and profile based. We

can’t move on without having those in place.

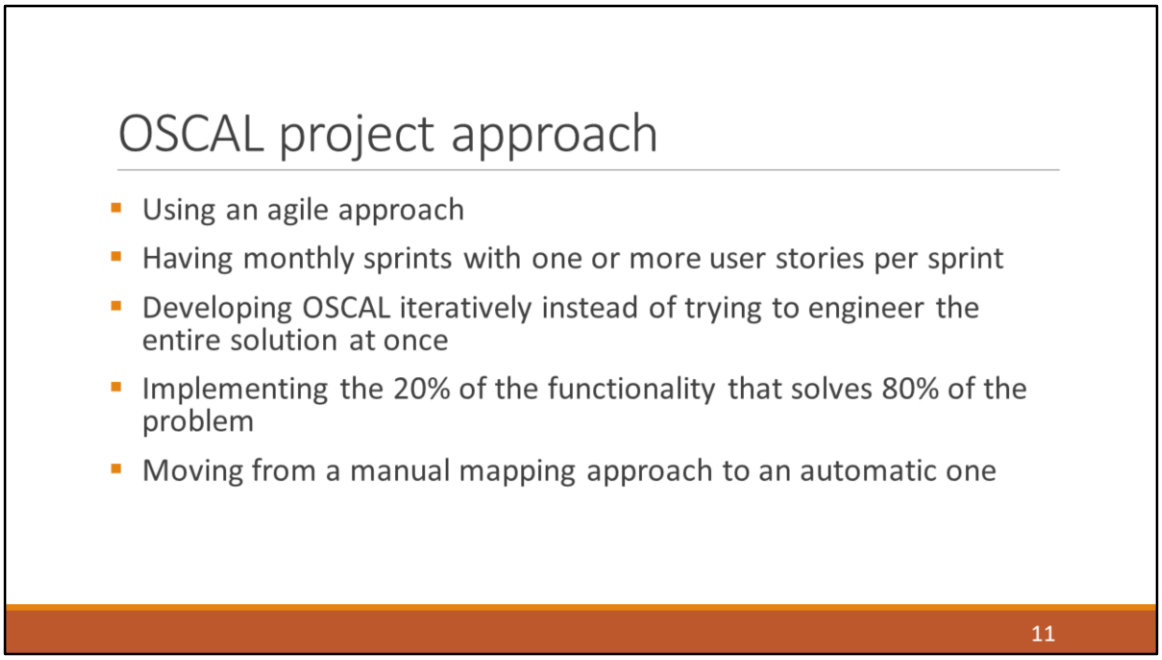


This speaks to who the users are within our current scope. We believe the work we're doing is immediately useful to them.

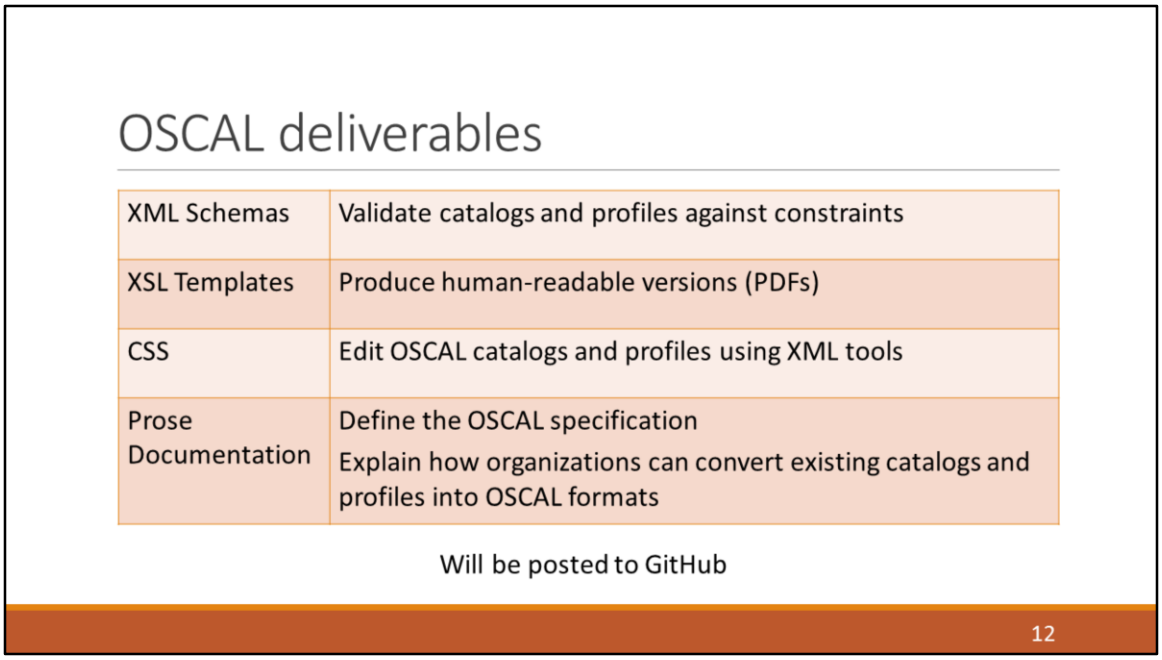
There are two different groups of stakeholders. There are *producers*, who are producing catalogs and profiles as well as tools using the OSCAL format. There are probably other types of producers who would also use this information, but this is our initial list. We are going to start reaching out to some of the catalog maintainers. The next slide discusses the other group of stakeholders, the consumers.



This talks to the benefits for OSCAL consumers. By providing a standardized catalog and profile model, operations personnel would be able to access that information and put it to use to understand if their systems comply with organization security requirements. For example, if NIST SP 800-53 revision 5 is in OSCAL format, agencies could use it to assess their compliance. A standardized catalog and profile model would similarly help security and privacy personnel, auditors, and assessors identify problems. For example, a profile with parameter values could be used to point out issues. Policy personnel can also benefit from OSCAL, such as using OSCAL when creating customized profiles for an organization.



The approach we're taking with OSCAL is an agile approach. We're adopting the philosophy of implementing the 20% of the functionality that solves 80% of the problem. We're trying to focus on the core capabilities that are needed to provide the greatest amount of benefit. Because we're working on a small set of capabilities, that allows us to make very fast progress. We're building the features that we believe solve the biggest problems, so we're providing the most value. This new strategy has been a game changer for the OSCAL project.



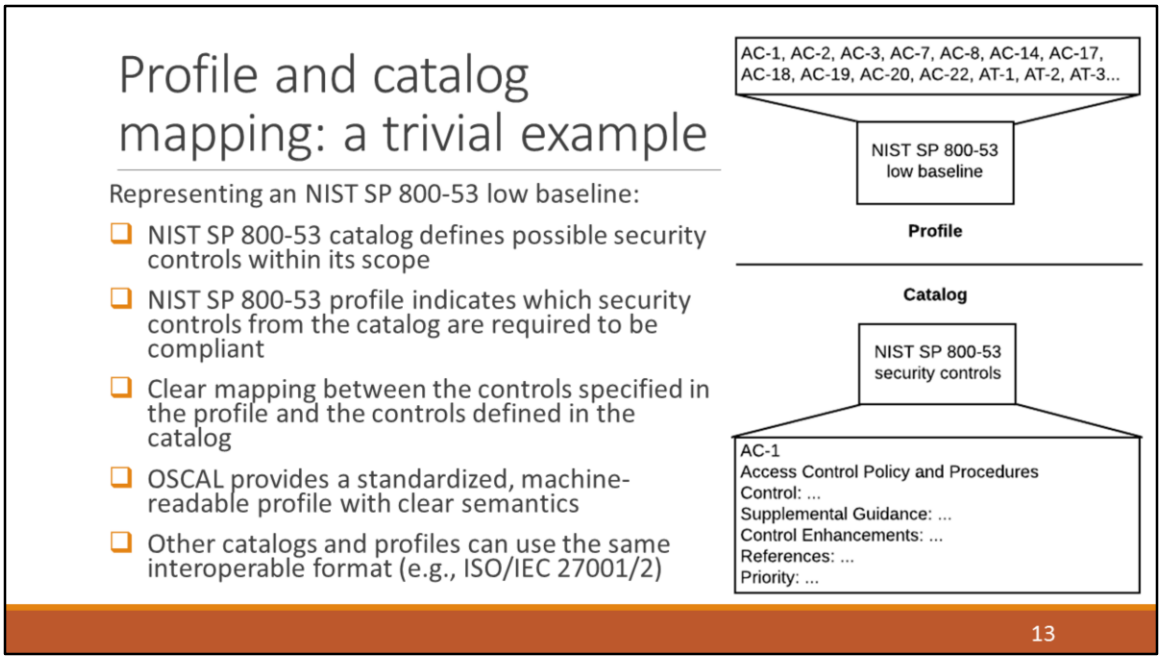
What we're producing as part of our monthly sprints are things like updated XML schemas for the catalog and profile models that will allow an organization to create a catalog or profile to validate that information. We're also producing Schematron definitions, which are kind of an extension of the XML schemas that provide more validation capabilities.

We're providing XSL templates that allow production of human-readable versions out of OSCAL XML. It will be easy to convert an OSCAL XML catalog into a human-readable form. For right now we are supporting HTML, and we are working on other formats to produce PDFs, etc. We want to be able to leverage a similar kind of approach with our schemas so we can produce specification documents directly out of the schemas we're producing.

We're producing CSS—people who are developing catalogs and profiles using XML tools can use CSS for data entry. This makes the interface much more usable.

We are also working on a prose OSCAL specification. It will define the catalog and profile models; capture the operational model of how to use OSCAL, and explain how you can convert catalog and profile information into OSCAL formats.

We are posting all of this information to GitHub. Some of this information is already there. We are working to get everything updated with the most recent development information. We are also working on updating the documentation to explain what's on the repository, how to use it, how the repository is organized, etc. This is much of the focus of our current sprint.



This is a simple conceptual example of what we're trying to do with catalogs and profiles. This example represents the NIST SP 800-53 low baseline. Basically, in the low baseline, which is a profile in OSCAL nomenclature, it's effectively selecting controls represented in NIST SP 800-53. Using OSCAL formats for these makes the mappings between the control catalog and the profile explicit and machine readable. OSCAL formats will allow profiles to be generated using the same format regardless of the underlying catalogs that are being used, like ISO 27001/2 and COBIT 5.

Prose vs. OSCAL (Machine-Readable)

AC-1 ACCESS CONTROL POLICY AND PROCEDURES

Control: The organization:

a. Develops, documents, and disseminates to [*Assignment: organization-defined personnel or roles*]:

1. An access control policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

2. Procedures to facilitate the implementation of the access control policy and associated access controls; and

b. Reviews and updates the current:

1. Access control policy [*Assignment: organization-defined frequency*]; and

2. Access control procedures [*Assignment: organization-defined frequency*].

Supplemental Guidance: This control addresses the establishment of policy and procedures for the effective implementation of selected security controls and control enhancements in the AC family. Policy and procedures reflect applicable federal laws, Executive Orders, directives, regulations, policies, standards, and guidance. Security program policies and procedures at the organization level may make the need for system-specific policies and procedures unnecessary. The policy can be included as part of the general information security policy for organizations or conversely, can be represented by multiple policies reflecting the complex nature of certain organizations. The procedures can be established for the security program in general and for particular information systems, if needed. The organizational risk management strategy is a key factor in establishing policy and procedures. Related control: PM-9.

Control Enhancements: None.

References: NIST Special Publications 800-12, 800-100.

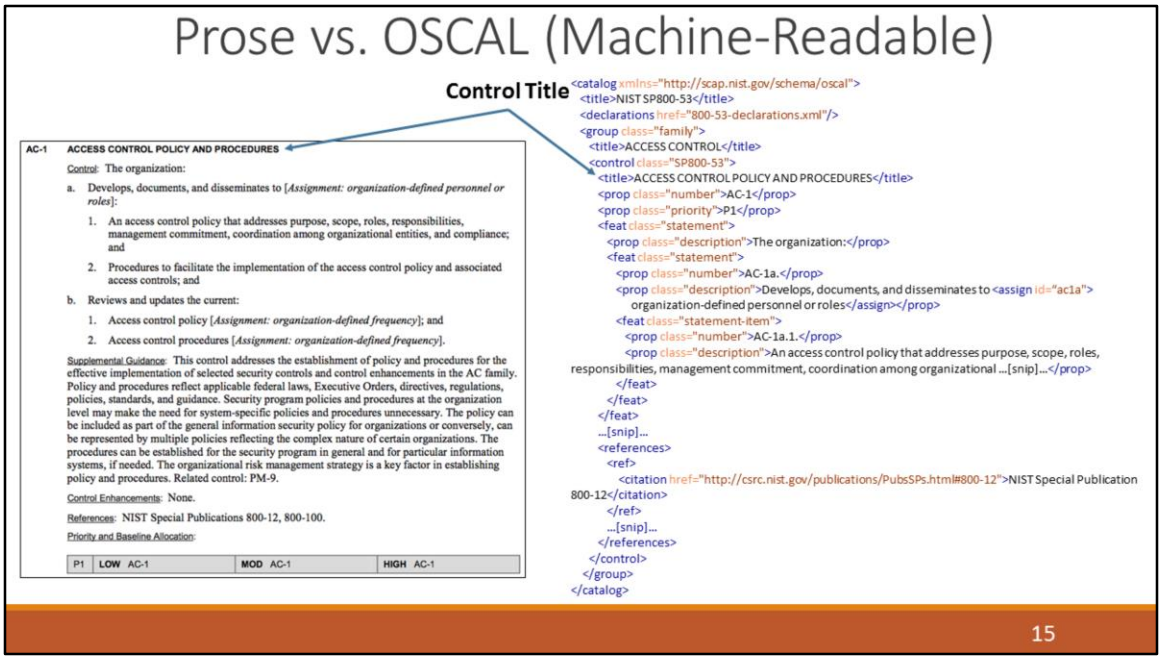
Priority and Baseline Allocation:

P1	LOW AC-1	MOD AC-1	HIGH AC-1

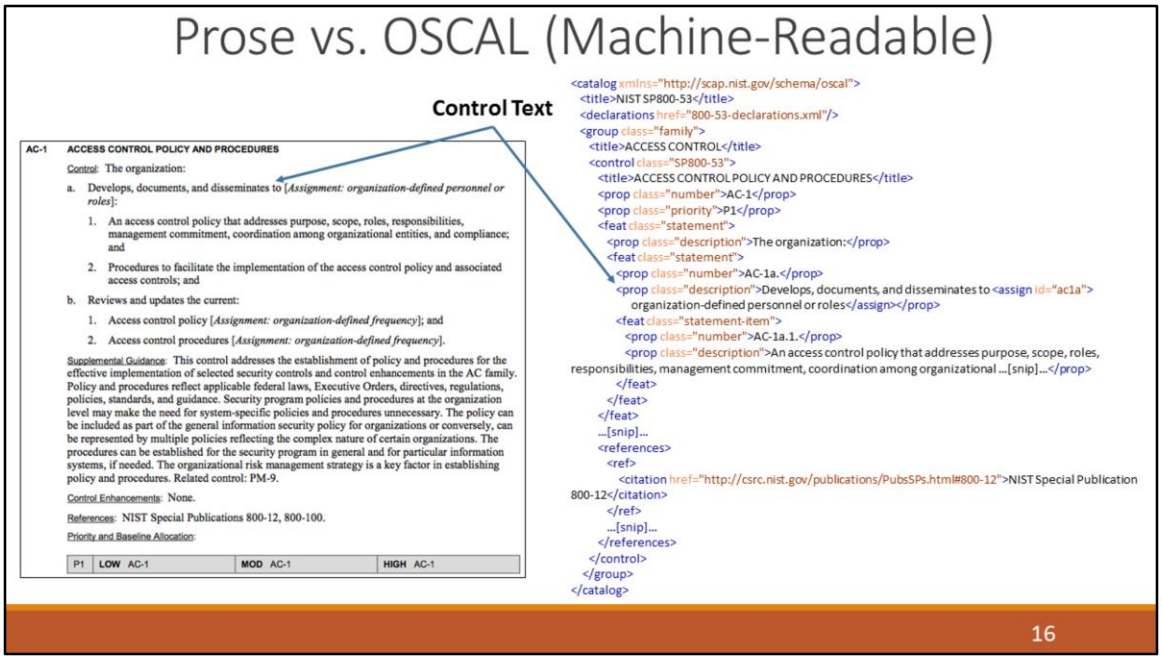
```
<catalog xmlns="http://scap.nist.gov/schema/oscal">
  <title>NIST SP800-53</title>
  <declarations href="800-53-declarations.xml"/>
  <group class="family">
    <title>ACCESS CONTROL</title>
    <control class="SP800-53">
      <title>ACCESS CONTROL POLICY AND PROCEDURES</title>
      <prop class="number">AC-1</prop>
      <prop class="priority">P1</prop>
      <feat class="statement">
        <prop class="description">The organization:</prop>
        <feat class="statement">
          <prop class="number">AC-1a.</prop>
          <prop class="description">Develops, documents, and disseminates to [Assignment: organization-
defined personnel or roles]:</prop>
          <feat class="statement-item">
            <prop class="number">AC-1a.1.</prop>
            <prop class="description">An access control policy that addresses purpose, scope, roles,
responsibilities, management commitment, coordination among organizational ...[snip]...</prop>
          </feat>
        </feat>
      </feat>
      ...[snip]...
      <references>
        <ref>
          <citation href="http://csrc.nist.gov/publications/PubsSPs.html#800-12">NIST Special Publication
800-12</citation>
        </ref>
        ...[snip]...
      </references>
    </control>
  </group>
</catalog>
```

14

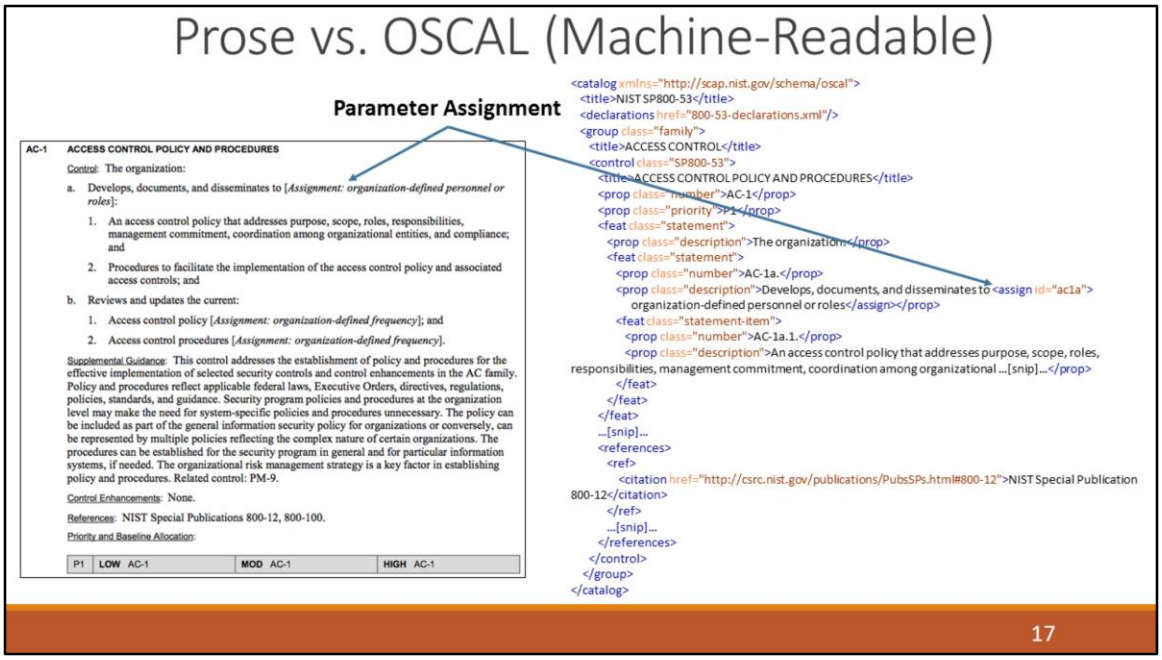
This slide and the following few slides provide a quick view of what OSCAL looks like in XML (right side) and how the OSCAL XML represents the information that is written in the prose version of the control catalog (left side).



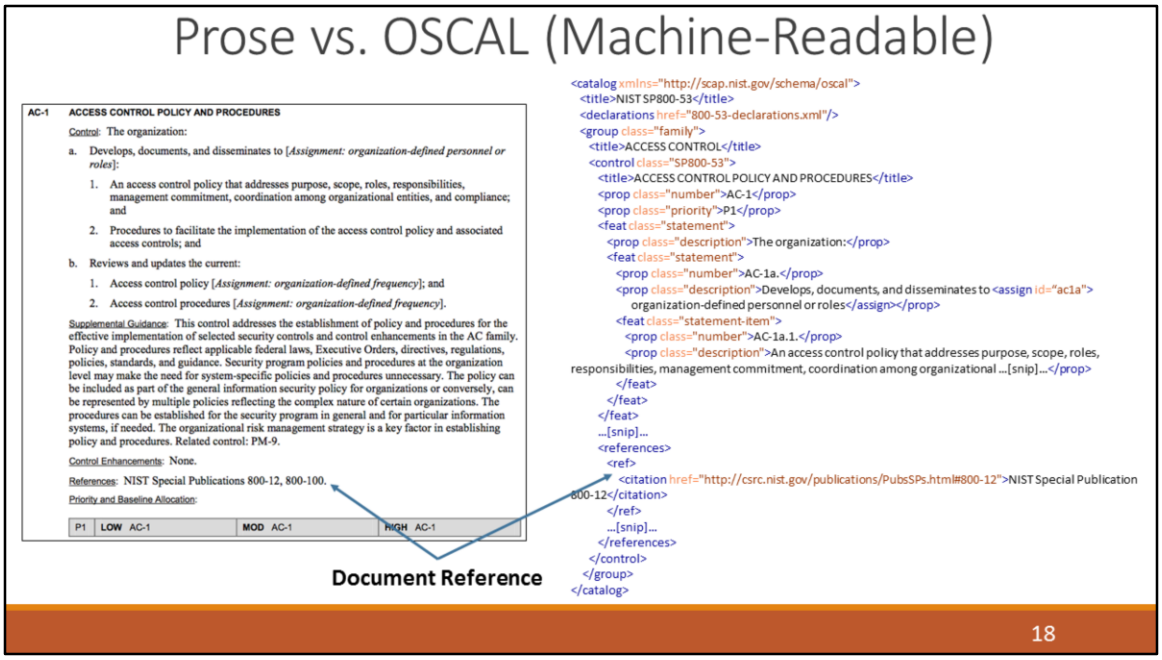
This example shows the Control Title element. The control is AC-1 in NIST SP 800-53. The Control Title is “Access Control Policy and Procedures”. On the right hand side you can see the title represented in OSCAL through the <title> element.



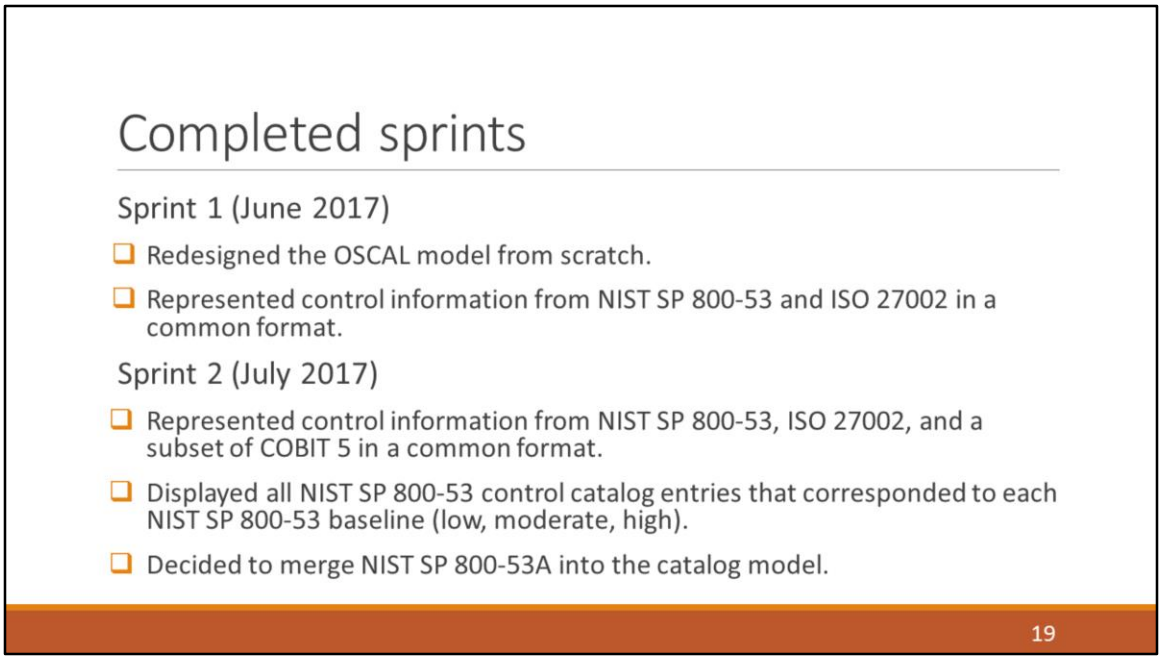
Here's an example for Control Text. On the left side, we have control text that says, "Develops, documents, and disseminates to [*Assignment: organization-defined personnel or roles*]." On the right side the description is defined through the description class within the <prop> element.



This example is for the Parameter Assignment. This defines the parameter that was embedded within the Control Text on the previous slide, "[*Assignment: organization-defined personnel or roles*]." The right side shows the ID defined for this parameter.

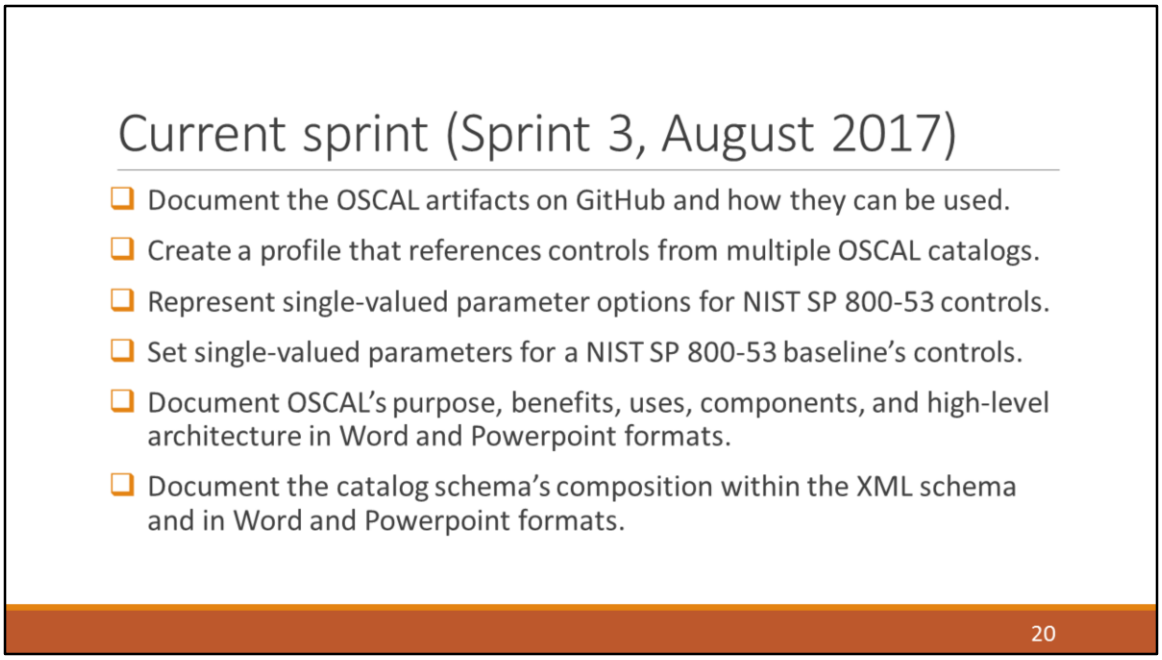


The final example is a Document Reference. Control AC-1 references NIST SPs 800-12 and 800-100, as shown on the left side. On the right side you see the OSCAL-formatted reference for NIST SP 800-12. (The other reference, to NIST SP 800-100, is omitted for brevity.) This is how a reference to another publication is represented. The URL provided in the citation can be resolved to the document. This is an example of how OSCAL can support tooling—you could click on a link or a button that would actually take you to the document being referenced.



We've completed the first two sprints. We completely redesigned the OSCAL model from scratch. When we started Sprint 1, we looked at the work that had been done and we weren't happy with what it was providing. We started trying to evolve it and quickly realized we needed to take a different approach. That led us down the path we're on. We've been able to represent control information from NIST SP 800-53, ISO 27002, and COBIT 5. We started small and became more aggressive. COBIT 5 was originally going to be in Sprint 1, but it's a very different animal from 800-53 and ISO 27002. We didn't want to take on the complexity of COBIT 5 right away, but rather spend some time in developing the initial format.

In Sprint 2 we did the COBIT 5 work. It was good that we considered COBIT 5 because it pushed us in a number of different ways we hadn't been thinking about. We were able to represent all these control catalogs. We were also able to develop a primitive representation of a NIST SP 800-53 baseline (profile) during Sprint 2. Because of the availability of assessment objectives, we decided to merge NIST SP 800-53A into the catalog model. Similar kinds of assessment information are available in ISO 27002 and COBIT 5, so we normalized that information within the catalog model. Assessment objectives are an optional piece of the catalog format; you don't have to provide assessment objectives.

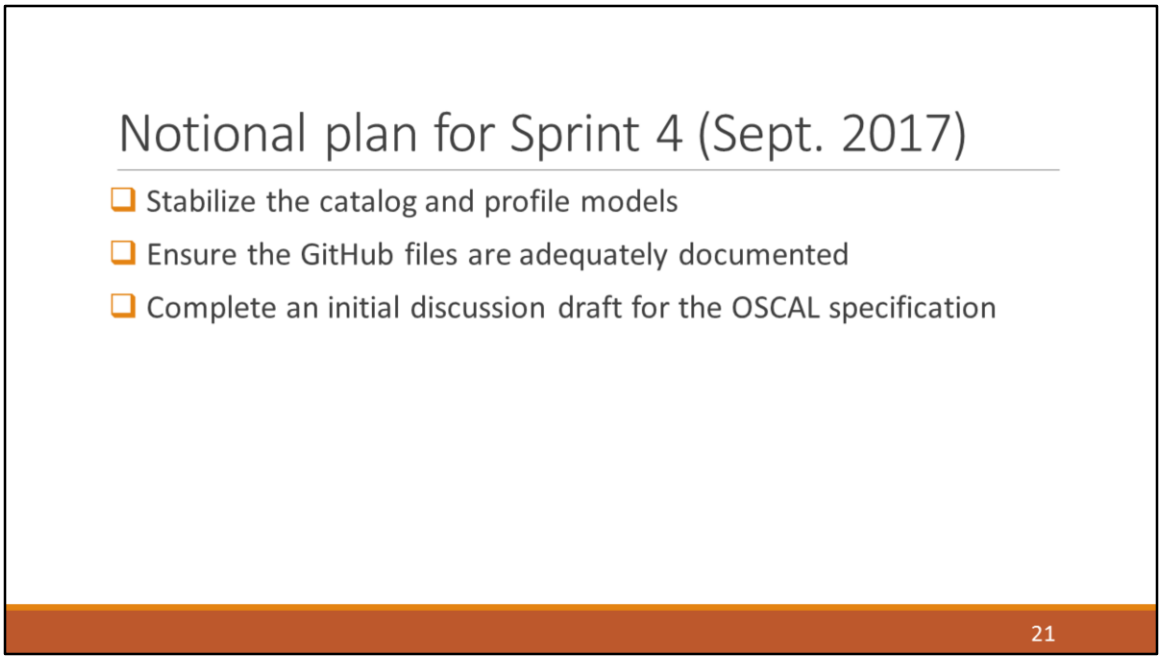


We did a lot of design and engineering work in our first two sprints. The first thing we wanted to do in Sprint 3 is take a step back and start to document more of the engineering work we've done. In this sprint we're documenting the OSCAL artifacts on GitHub. We've been publishing to a development branch on GitHub. We're in the process of merging back into the master branch. We're working on the repository structure, what the artifacts are in the repository, how to use them, etc. to help people digest all this information. The layout of the repository has changed significantly. Having documentation would be helpful for improving the repository's usability.

We've been cleaning up all the documentation. We've also been working on documenting what we're trying to accomplish with OSCAL, what its benefits are, etc. Some of the information in this presentation is a result of work we've been doing in this sprint. We are organizing our educational materials around what our approach is for OSCAL and what we're trying to accomplish. We are also starting to work on documenting the model used for OSCAL. We're creating an XML schema and working on a transformation approach that will enable us to automatically extract documentation out of the schema to build prose documentation for the specification. This will save us time in the long run because it will help us keep our prose

documentation and schemas in sync.

We're also doing some new engineering work in this sprint, shown in the second through fourth bullets. We're supporting referencing controls for multiple OSCAL catalogs in the profile model as part of this sprint. We're working on building out the parameter value support for NIST SP 800-53 controls. The previous XML example with the parameter assignment—this is the work we're doing in this sprint as well. The other thing we're working on is allowing values to be assigned to these parameters in a profile. That is particularly useful for custom profiles.

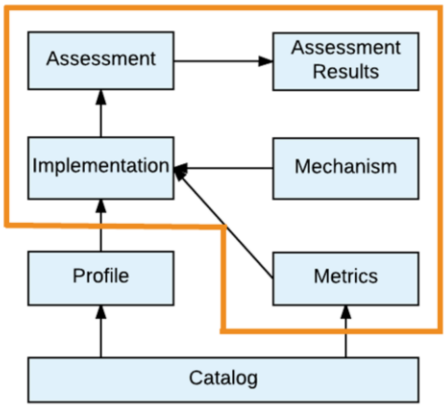


This is a notional plan for our next sprint. We start out each sprint with a set of user stories, and we accomplish a major portion of them. We may encounter challenges and defer some things to the next sprint. We won't know for sure what we'll be planning for Sprint 4 until Sprint 3 has been completed. This slide has some ideas.

We want to get to a point where we can stabilize the catalog and profile models, to lock things down to the point where we're only making minimal changes and can publish a draft of the specification and the schemas. That will facilitate OSCAL review and use. In Sprint 4 we'll be going through the remaining issues we have with the models. People can be reviewing and trying what we've done so far while we start working on the next major problem.

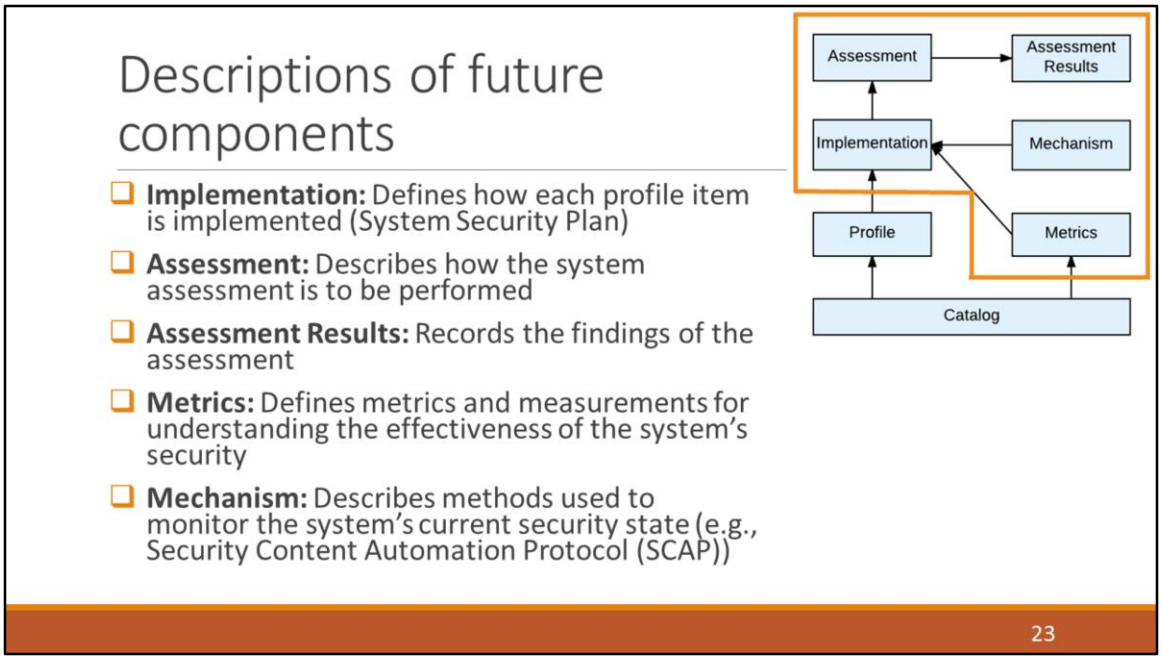
We also plan to finish all the documentation we aren't able to get done during Sprint 3 on GitHub to make sure the repository is adequately documented. It would be useful after this sprint to get tiger team feedback about whether the documentation is useful.

Future focus



22

We will be spending the next few months working on wrapping up the catalog and profile work. Beyond that we want to be able to address higher-level concepts relating to assessment.



This slide describes the OSCAL components we're expecting to work on in the future. Don't focus too much on the definitions here; focus more on the body of work we are planning to do. We recognize that our understanding of these problem spaces is going to evolve over time. We want to work with the tiger team to help us understand them better.

Implementation: We want to develop models as part of OSCAL that will support definitions of implementation. We want a model representing a machine-readable system security plan in OSCAL. We also want to support transforms from that machine-readable form to a human-readable version.

Assessment and Assessment Results: We want to be able to model how assessments are to be performed so that the data can be used to drive an assessment. On the opposite side of the coin, we want to have an OSCAL format for representing assessment results.

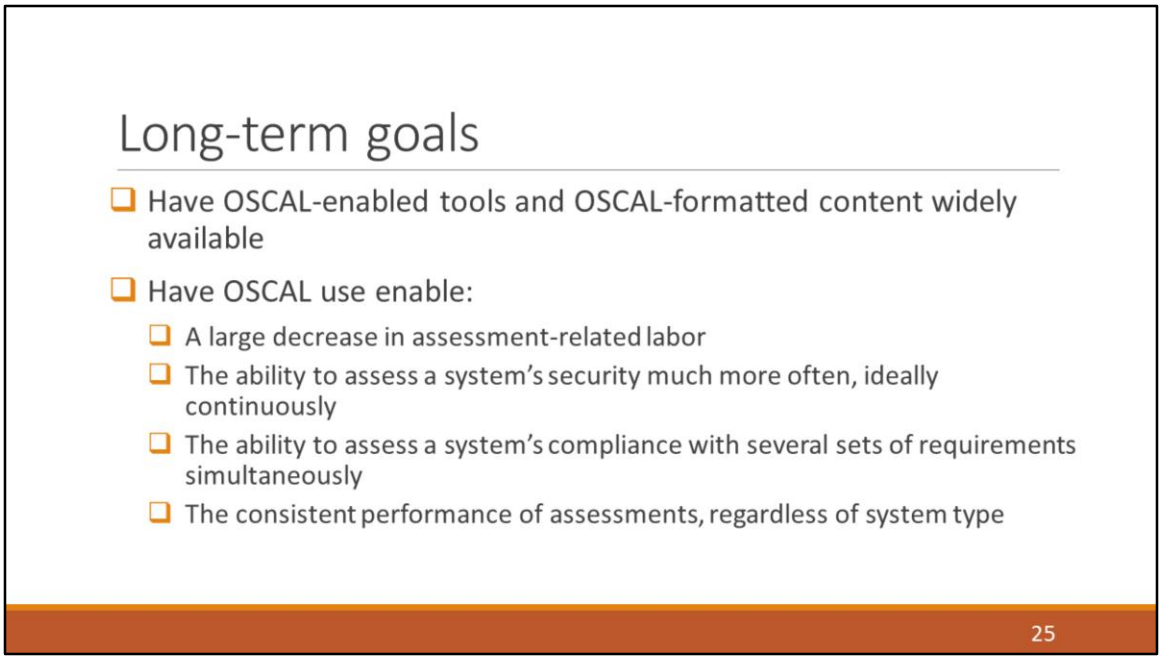
Metrics and Mechanisms: We want to talk about mechanisms to get metrics for implementation and assessment, as well as mechanisms that can be used to do the underlying monitoring. This might be a way of connecting with SCAP, as an example.

Some of these components are easier and clearer to do, while some are harder and more research-focused. This is just a glimpse of what we'd like to move towards.

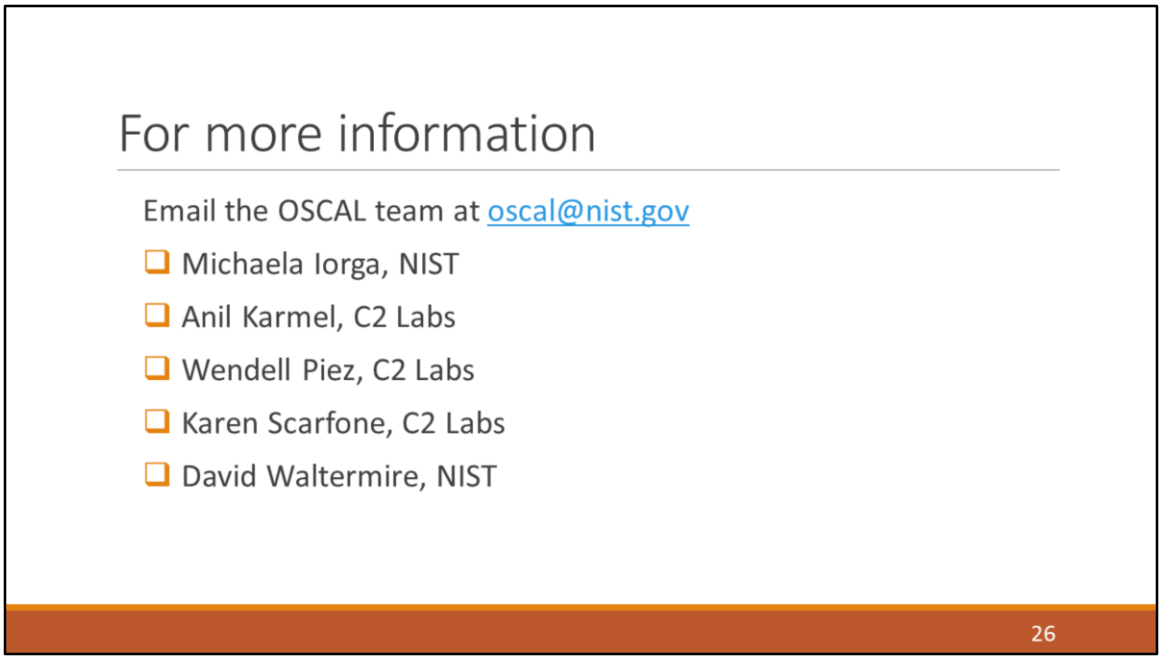
Once we finish the catalog and profile work, we'll probably want to target the implementation component next.

Future OSCAL work

1. Represent the following in standardized, machine-readable formats:
   a. How each control is implemented on a system
   b. Which controls are applicable to a system
   c. How each control on a system can be assessed (procedures) to ensure it's implemented and operating properly
   d. How the control implementations on a system can be measured (metrics)
2. Identify which controls from a profile are implemented on a system.
3. Verify whether each control meets the options (parameters) specified in the profile.



We've spoken a bit to what we want to do in the long term. Mostly what these items are trying to accomplish is to create more automation around the actual assessment process to decrease the amount of labor needed by having the computers do more of the assessment. By automating more, assessments can be done more often, ideally continuously. OSCAL can make it easy to determine if a system is compliant with one or more sets of requirements. OSCAL can also improve the consistency of how assessments are performed, with repeatability regardless of the type of system and the assessor.



We are in the process of setting up an email alias. We will be able to use the email alias and the GitHub repository to provide better methods of communication.