



# Open Security Controls Assessment Language (OSCAL) Workshop Logistics

David Waltermire

National Institute of Standards and Technology

# Welcome

- ▶ This is the first of a series of OSCAL workshops
- ▶ Logistics
  - ▶ Nearest Exits
  - ▶ Thank you for silencing your cell phones
  - ▶ Breaks
  - ▶ Lunch @noon in Building 101

# Goals for this Workshop

- ▶ Our goals for this workshop are:
  - ▶ To connect with the OSCAL community
  - ▶ To share information on the progress made to date
  - ▶ Educate users, content creators, and tool vendors about the current OSCAL models
  - ▶ Discuss next steps for OSCAL

# Agenda

Time	Topic
8:00 am	Registration and Networking
9:00 am	Welcome, Introduction to Workshop
9:10 am	<b>What is OSCAL and Who Needs It</b> Michaela Iorga, OSCAL co-lead NIST
10:00 am	<b>OSCAL and FedRAMP</b> Ashley Mahan, Director, FedRAMP, GSA
10:30 am	Break
10:45 am	<b>Tools That Understand OSCAL</b> Andrew Weiss, GitHub and Wendell Piez, OSCAL team member, NIST
11:20 am	<b>OSCAL Roadmap</b> David Waltermire, SCAP Lead and OSCAL co-lead, NIST
Noon	Lunch
1:30 pm	<b>OSCAL Catalog and Profile Models</b> Brian Ruf, OSCAL team member, Noblis
2:45 pm	Break
3:00 pm	<b>OSCAL Component and System Security Plan Models</b> David Waltermire, SCAP lead and OSCAL co-lead, NIST
4:15 pm	<b>Discussion: Tomorrow is Today – The Need for Automation</b> Moderator: David Waltermire, SCAP lead and OSCAL co-lead, NIST
4:50 pm	Closing Remarks
5:00 pm	Adjourn

# What is OSCAL and Who Needs It

**Michaela Iorga**

Senior Security Technical Lead and OSCAL co-lead  
National Institute of Standards and Technology (NIST)

# Presentation Agenda

- Why OSCAL is needed
- OSCAL in a nutshell
- Risk Management Framework and OSCAL
- OSCAL Guiding Principals
- Overview of OSCAL Layers and Models
- Project focus
- Show and tell



# Why do we care ... and why you should too!

## Today's challenges:

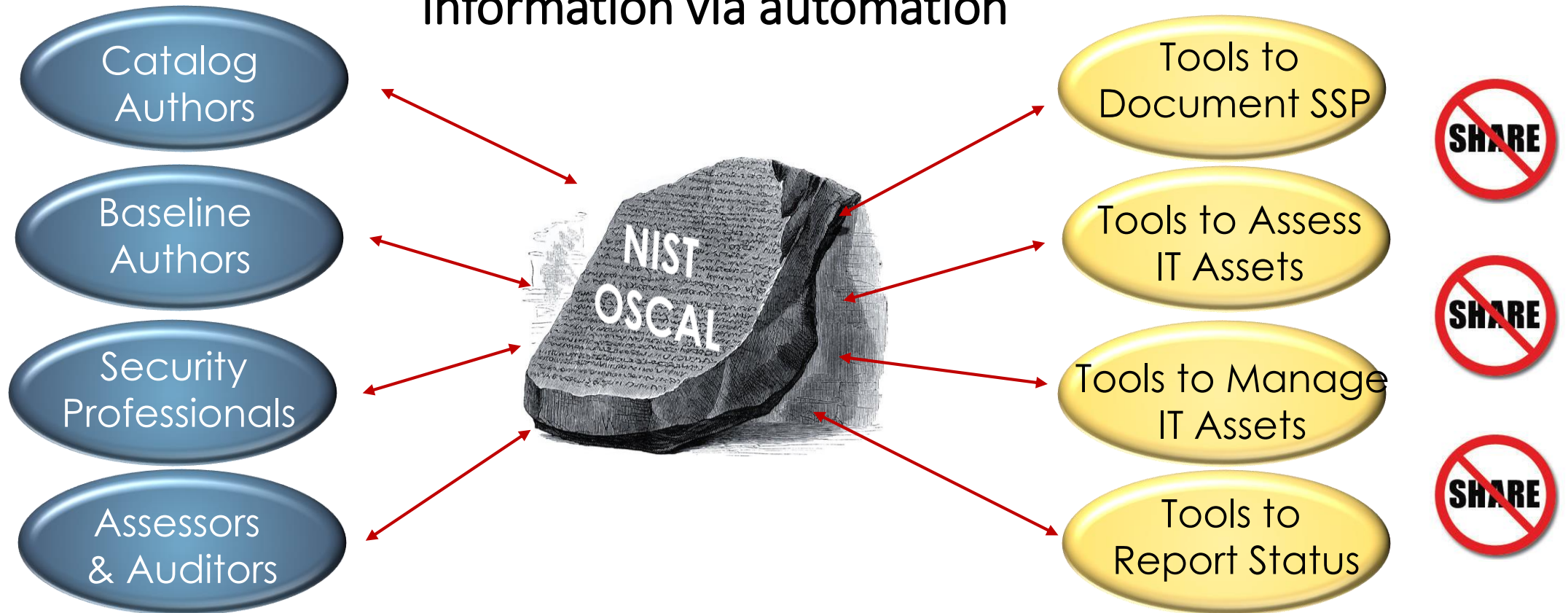
- Information technology is complex
- Security vulnerabilities are everywhere
- Regulatory frameworks are burdensome
- Risk management is hard
- Documentation is out of date

### TABLE OF CONTENTS

1.	INFORMATION SYSTEM NAME/TITLE .....	1
2.	INFORMATION SYSTEM CATEGORIZATION .....	1
2.1.	Information Types .....	1
2.2.	Security Objectives Categorization (FIPS 199) .....	3
2.3.	Digital Identity Determination .....	3
3.	INFORMATION SYSTEM OWNER .....	4
4.	AUTHORIZING OFFICIALS .....	4
5.	OTHER DESIGNATED CONTACTS .....	4
6.	ASSIGNMENT OF SECURITY RESPONSIBILITY .....	5
7.	INFORMATION SYSTEM OPERATIONAL STATUS .....	6
8.	INFORMATION SYSTEM TYPE .....	7
8.1.	Cloud Service Models .....	7
8.2.	Cloud Deployment Models .....	8
8.3.	Leveraged Authorizations .....	8
9.	GENERAL SYSTEM DESCRIPTION .....	9
9.1.	System Function or Purpose .....	9
9.2.	Information System Components and Boundaries .....	9
9.3.	Types of Users .....	10
9.4.	Network Architecture .....	11
10.	SYSTEM ENVIRONMENT AND INVENTORY .....	12
10.1.	Data Flow .....	12
10.2.	Ports, Protocols and Services .....	14
11.	SYSTEM INTERCONNECTIONS .....	15
12.	LAWS, REGULATIONS, STANDARDS AND GUIDANCE .....	17
12.1.	Applicable Laws and Regulations .....	17
12.2.	Applicable Standards and Guidance .....	17
13.	MINIMUM SECURITY CONTROLS .....	18
13.1.	Access Control (AC) .....	25
	AC-1 Access Control Policy and Procedures Requirements (H) .....	25
	AC-2 Account Management (H) .....	26
	AC-2 (1) Control Enhancement (M) (H) .....	27
	AC-2 (2) Control Enhancement (H) .....	28
	AC-2 (3) Control Enhancement (H) .....	29
	AC-2 (4) Control Enhancement (H) .....	30
	AC-2 (5) Control Enhancement (H) .....	31
	AC-2 (7) Control Enhancement (H) .....	31
	AC-2 (9) Control Enhancement (H) .....	32
	AC-2 (10) Control Enhancement (M) (H) .....	33
	AC-2 (11) Control Enhancement (H) .....	34
	AC-2 (12) Control Enhancement (H) .....	35

# What is OSCAL?

OSCAL is like a Rosetta Stone that enables tools and organizations to exchange information via automation



OSCAL sets the foundation for automation and interoperability



# NIST's Goals for OSCAL

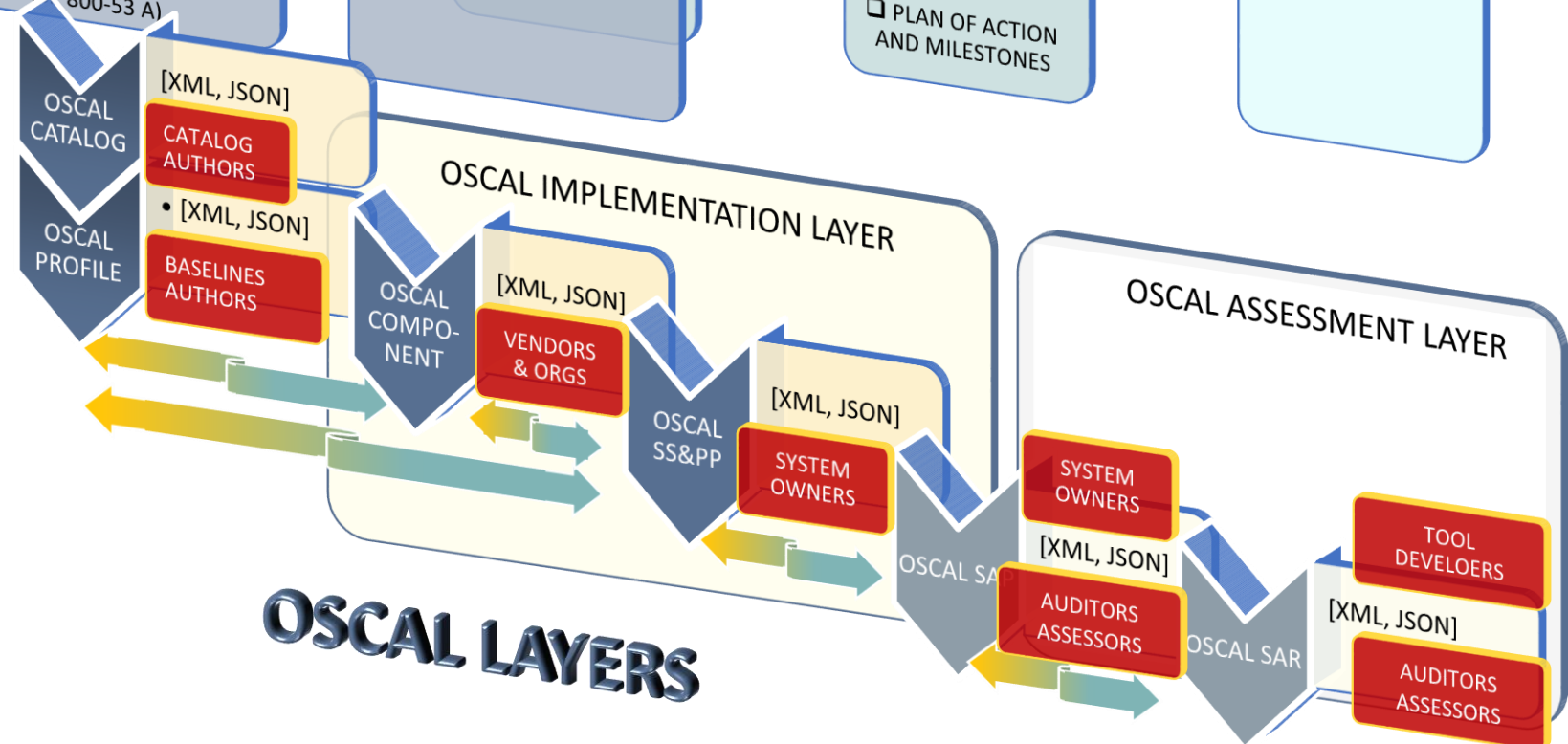
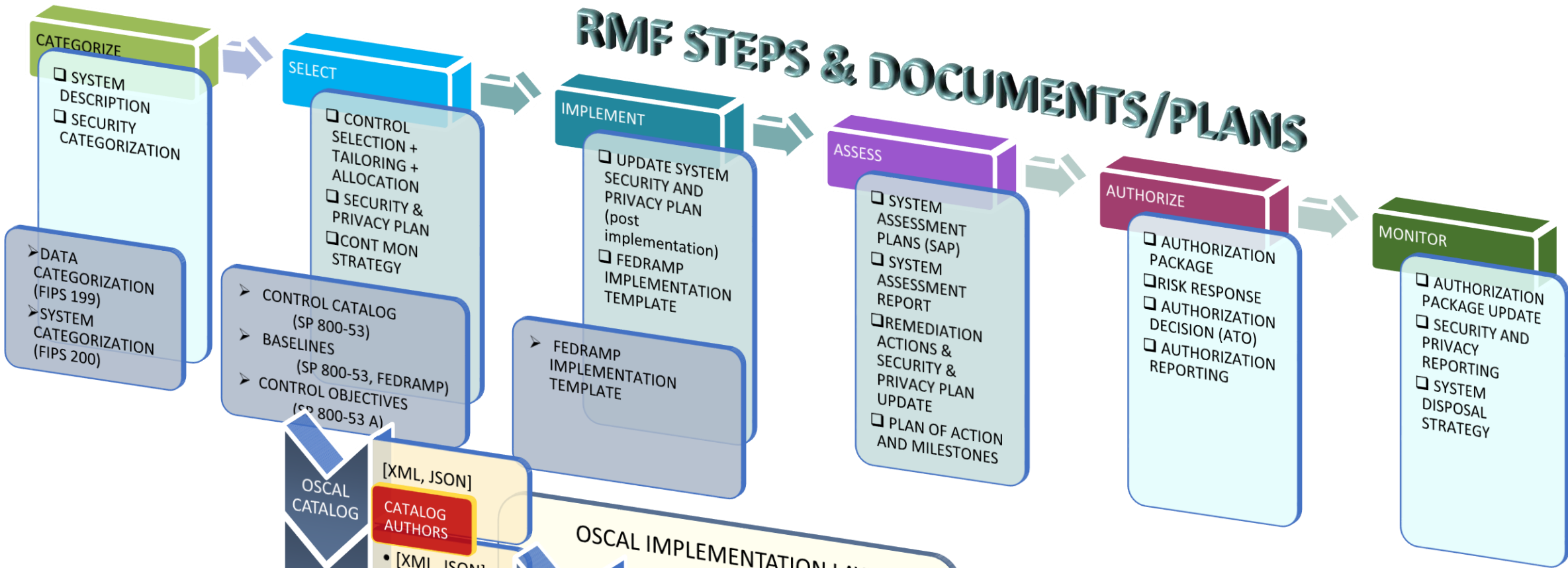
- **Provide** a common/single machine-readable *language*, expressed in XML and JSON, for:
  - ❑ multiple compliance and risk management frameworks (e.g. SP 800-53, ISO/IEC 27001 & 2, COBIT 5)
  - ❑ software and service providers to express implementation guidance against security controls
  - ❑ sharing how security controls are implemented (System Security Plans [SSPs])
  - ❑ sharing security assessment plans (System Assessment Plans [SAPs] )
  - ❑ sharing security assessment results/reports (System Assessment Results [SARs])
- **Enable** automated traceability from selection of security controls through implementation and assessment

# NIST's Targeted Outcome for OSCAL

- Widely available OSCAL-formatted content and OSCAL-enabled tools
  - ❑ A large decrease in assessment-related labor
  - ❑ The ability to assess a system's security much more often, ideally continuously
  - ❑ The ability to assess a system's compliance with several sets of requirements simultaneously
  - ❑ The consistent performance of assessments, regardless of system type



# RMF STEPS & DOCUMENTS/PLANS



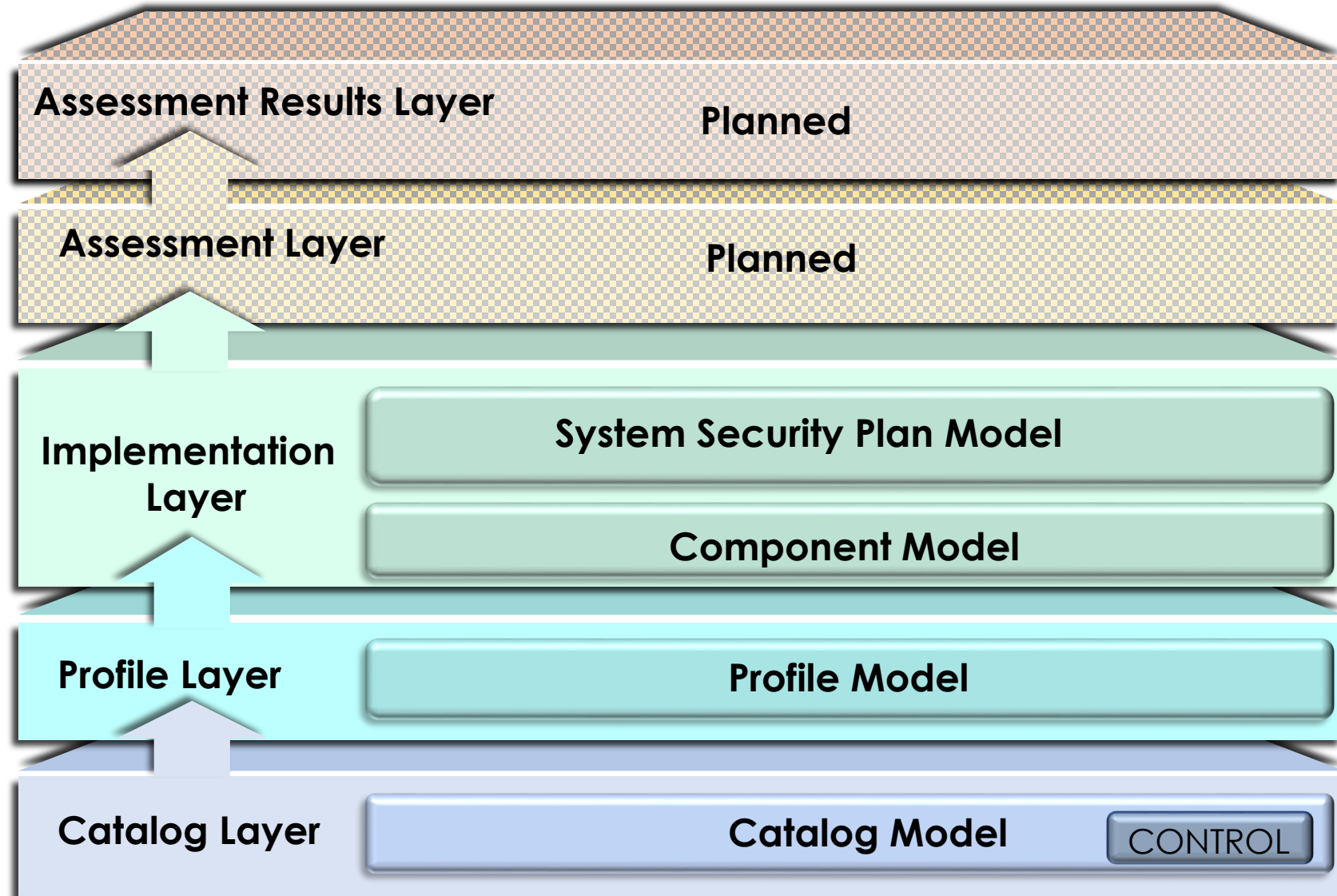
# OSCAL Guiding Principles

- Produce a set of extensible formats through a community-focused effort that supports a **broad range of control-based risk management processes**.
- Support control-based risk assessment based on data collected using a **continuous monitoring capabilities**.
- Ensure security controls, implementation, and verification processes have **full traceability and inherit** at the baseline (software and service provider) and system interconnection levels

## OSCAL Guiding Principles (cont)

- Standardize the expression of artifacts driving crowd-sourced development and improvement across profile and implementation layers
- Support multiple, interoperable, and lossless machine-readable formats including XML and JSON
- Provide a common means to identify and version shared resources
- Align OSCAL models with current, practical information, and support advanced structures that provide for greater automation and verification. **This principle provides a path for early adoption and ongoing evolution around how OSCAL will be used.**

# OSCAL Layers and Models



## Examples of control catalogs that were studied:

- ❑ NIST SP 800-53 + NIST 800-53 A
- ❑ ISO/IEC 27002&1
- ❑ COBIT 5

# OSCAL Control

NIST SP 800-53 Rev4

**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
----	----------	----------	-----------

**9.1.1 Access control policy**

Control

An access control policy should be established, documented and reviewed based on business and information security requirements.

Implementation guidance

Asset owners should determine appropriate access control rules, access rights and restrictions for specific user roles towards their assets, with the amount of detail and the strictness of the controls reflecting the associated information security risks.

Access controls are both logical and physical (see [Clause 11](#)) and these should be considered together. Users and service providers should be given a clear statement of the business requirements to be met by access controls.

The policy should take account of the following:

- a) security requirements of business applications;
- b) policies for information dissemination and authorization, e.g. the need-to-know principle and information security levels and classification of information (see [8.2](#));
- c) consistency between the access rights and information classification policies of systems and networks;
- d) relevant legislation and any contractual obligations regarding limitation of access to data or services (see [18.1](#));
- e) management of access rights in a distributed and networked environment which recognizes all types of connections available;
- f) segregation of access control roles, e.g. access request, access authorization, access administration;
- g) requirements for formal authorization of access requests (see [9.2.1](#) and [9.2.2](#));
- h) requirements for periodic review of access rights (see [9.2.5](#));
- i) removal of access rights (see [9.2.6](#));
- j) archiving of records of all significant events concerning the use and management of user identities and secret authentication information;
- k) roles with privileged access (see [9.2.3](#)).

Other information

Care should be taken when specifying access control rules to consider:

- a) establishing rules based on the premise “Everything is generally forbidden unless expressly permitted” rather than the weaker rule “Everything is generally permitted unless expressly forbidden”;
- b) changes in information labels (see [8.2.2](#)) that are initiated automatically by information processing facilities and those initiated at the discretion of a user;
- c) changes in user permissions that are initiated automatically by the information system and those initiated by an administrator;
- d) rules which require specific approval before enactment and those which do not.

Access control rules should be supported by formal procedures (see [9.2](#), [9.3](#), [9.4](#)) and defined responsibilities (see [6.1.1](#), [9.3](#)).

Role based access control is an approach used successfully by many organisations to link access rights with business roles.

Two of the frequent principles directing the access control policy are:

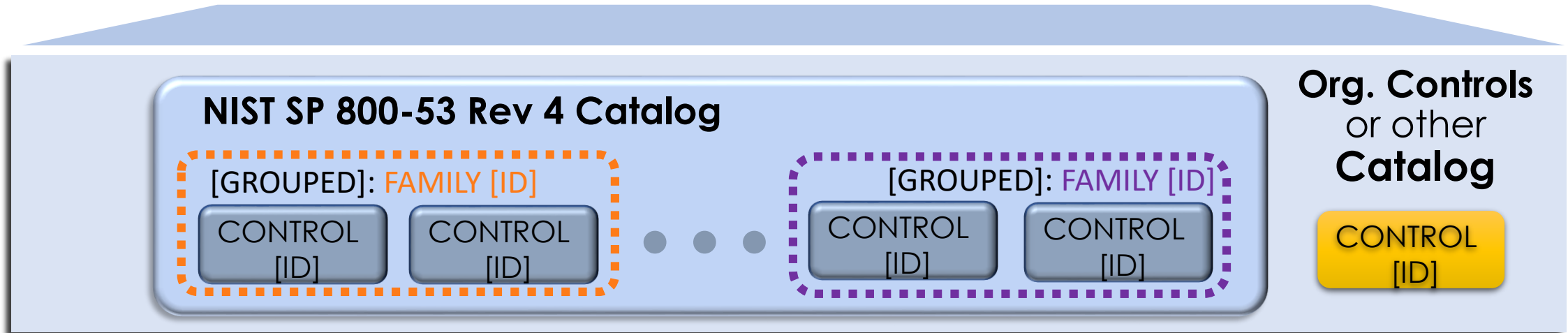
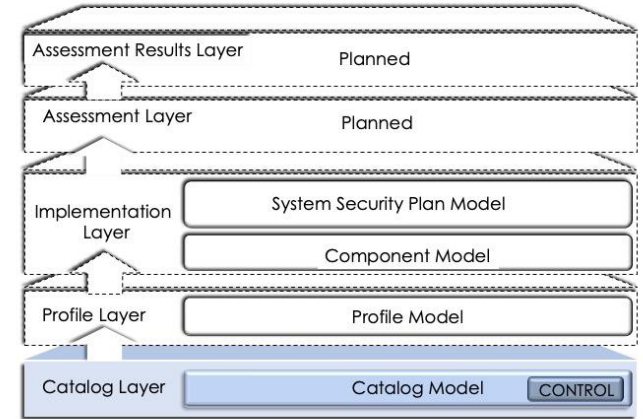
- a) Need-to-know: you are only granted access to the information you need to perform your tasks (different tasks/roles mean different need-to-know and hence different access profile);
- b) Need-to-use: you are only granted access to the information processing facilities (IT equipment, applications, procedures, rooms) you need to perform your task/job/role.

# OSCAL Control

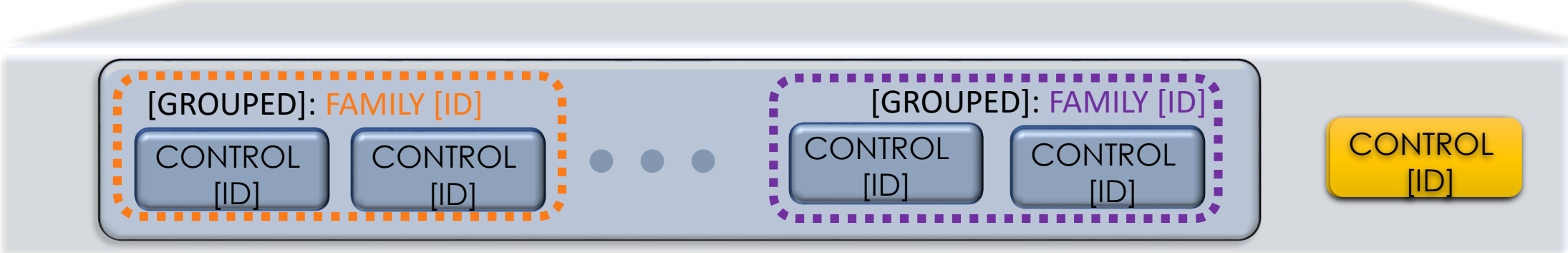
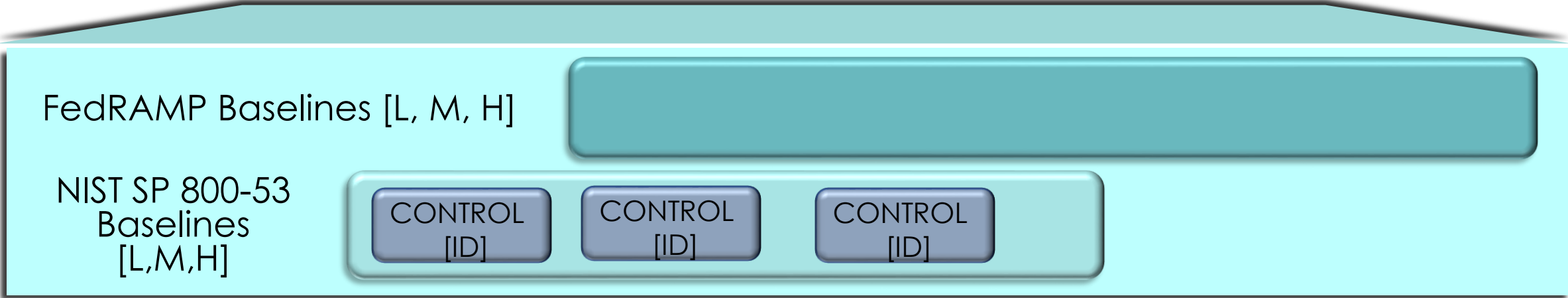
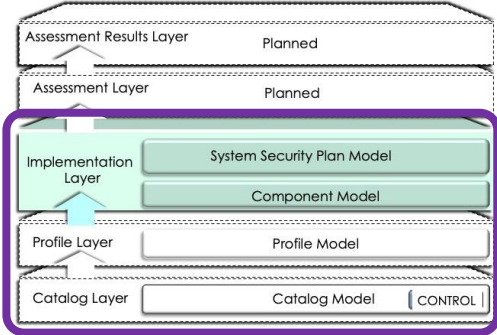
Title	IDs
Parameters	IDs
Properties (name/value)	IDs
Parts (complex & nested): Describe: <ul style="list-style-type: none"><li><input type="checkbox"/> Statements</li><li><input type="checkbox"/> Enhancements</li><li><input type="checkbox"/> Guidance</li><li><input type="checkbox"/> Assessment objective</li><li><input type="checkbox"/> Assessment activity</li></ul>	IDs
	IDs
	IDs
Nested Controls -> [parent & child]	IDs

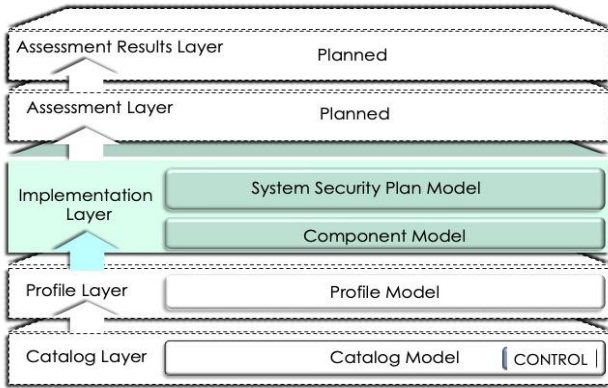


# OSCAL Catalog Layer

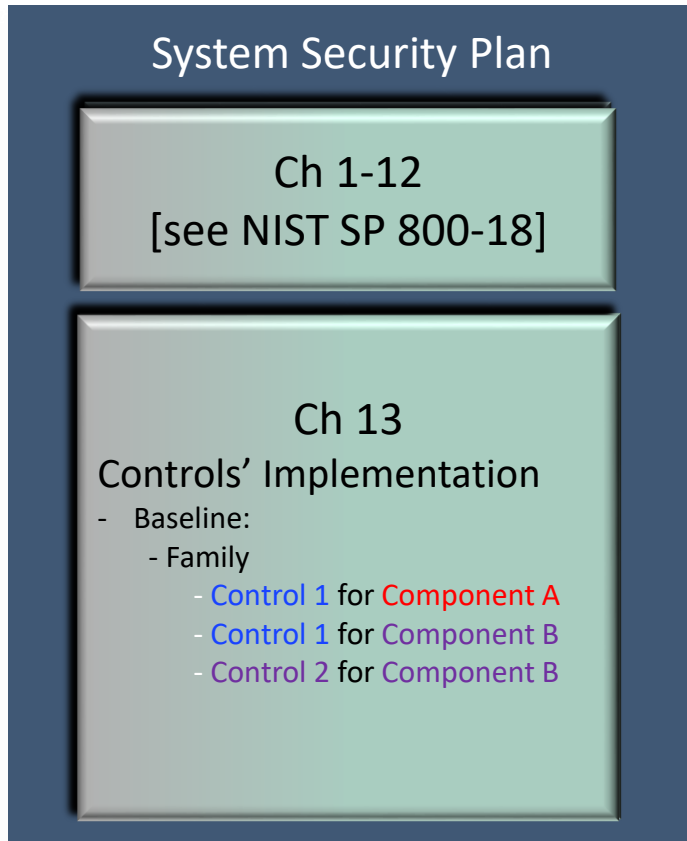


# OSCAL Profile Layer





\* From the top down

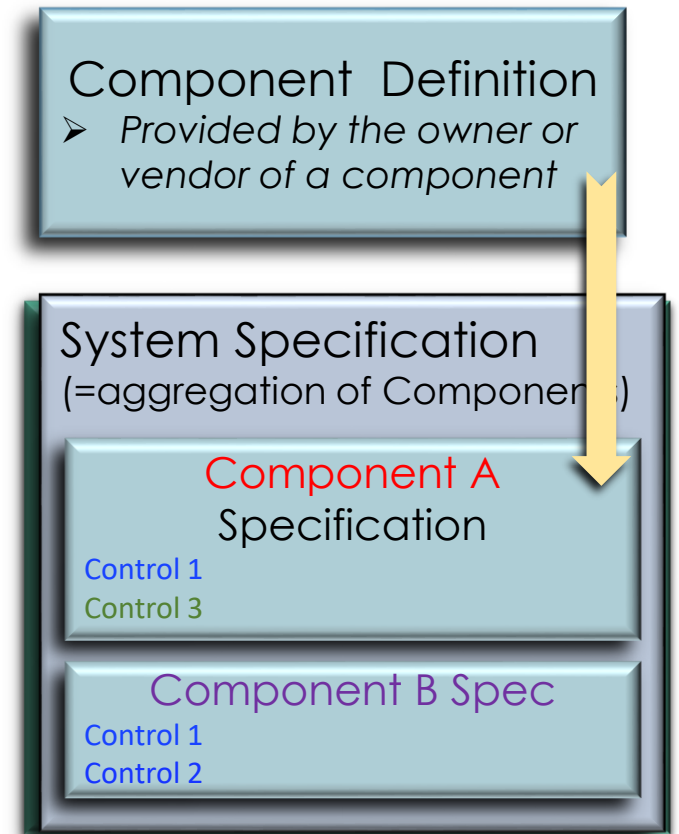


# OSCAL Implementation Layer

## - Component and SSP -



\*From the bottom up



# PARTNERS:

## ➤ Government:

- FedRAMP (GSA)
- NSA

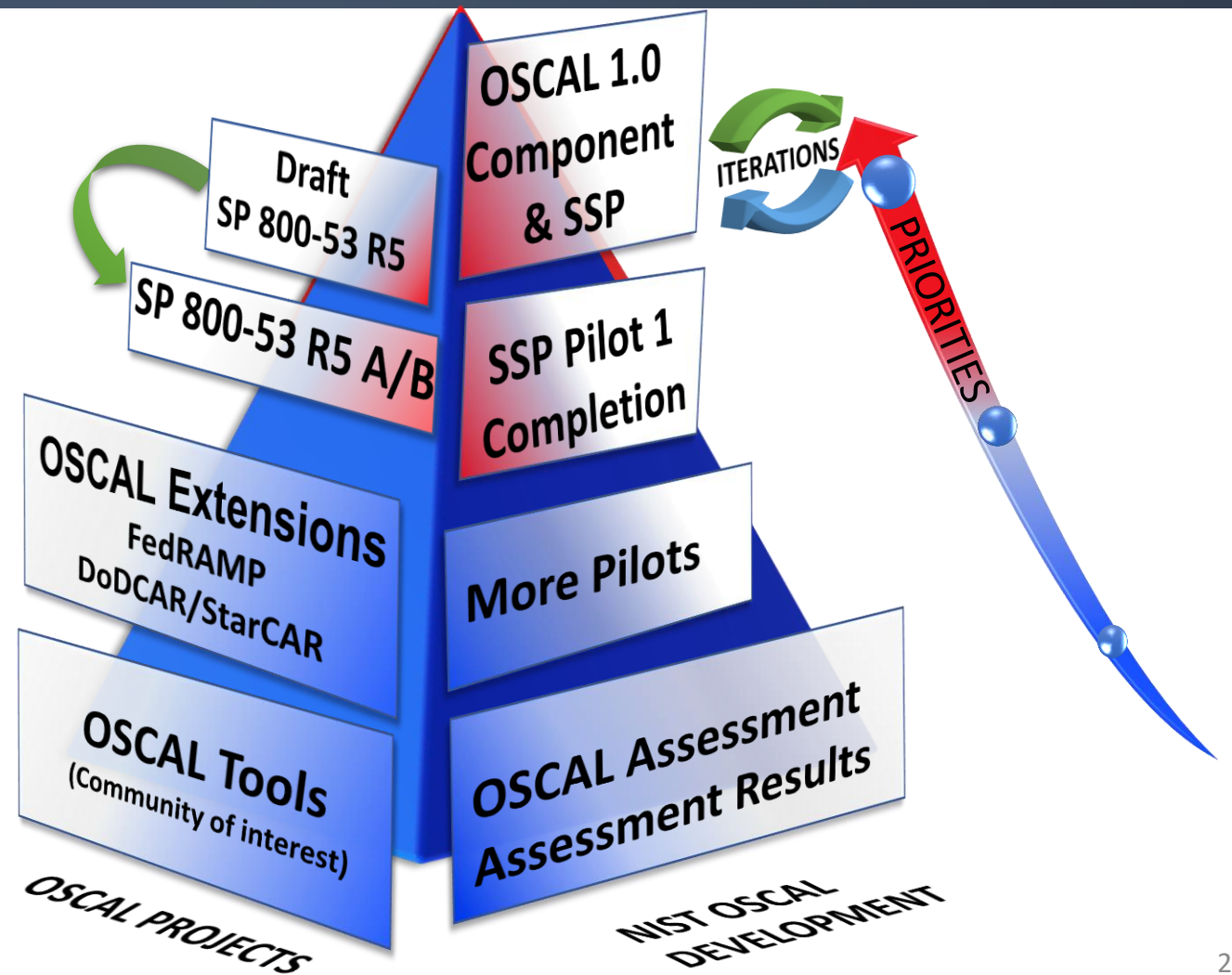
## ➤ Industry:

- Noblis
- GitHub
- Docker
- C2Labs
- Amazon (AWS)
- RedHat

## ➤ Many following, including:

- Telos
- InfoBeyond
- Aponia
- Mitre
- CSA

# CURRENT PROJECT FOCUS:



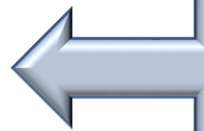
# Show and Tell

## ➤ NIST CATALOG & PROFILE(s):

### ❑ [OSCAL/content/nist.gov/SP800-53/rev4/](https://github.com/usnistgov/OSCAL/tree/master/content/nist.gov/SP800-53/rev4/)

<https://github.com/usnistgov/OSCAL/tree/master/content/nist.gov/SP800-53/rev4>

- ❑ [json](#)
- ❑ [xml](#)
- ❑ [yaml](#)



- ❑ NIST SP-800-53 rev4 HIGH-baseline profile
- ❑ NIST SP-800-53 rev4 MODERATE-baseline profile
- ❑ NIST SP-800-53 rev4 LOW-baseline profile
- ❑ NIST SP-800-53 rev4 control catalog

### ❑ [OSCAL/content/fedramp.gov/](https://github.com/usnistgov/OSCAL/tree/master/content/fedramp.gov/)

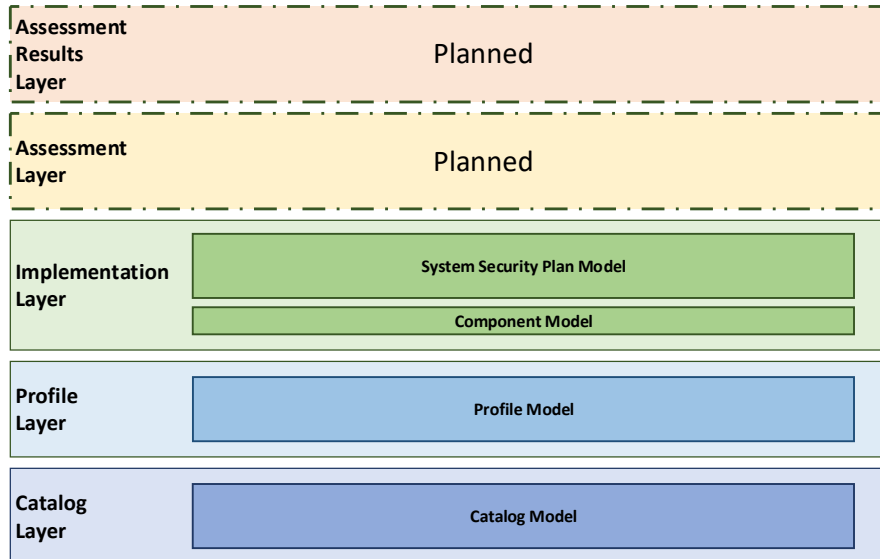
<https://github.com/usnistgov/OSCAL/tree/master/content/fedramp.gov>

- ❑ [json](#)
- ❑ [xml](#)
- ❑ [yaml](#)



- ❑ FedRAMP **RESOLVED** rev4 HIGH-baseline profile
- ❑ FedRAMP **RESOLVED** rev4 MODERATE-baseline profile
- ❑ FedRAMP **RESOLVED** LOW-baseline profile
- ❑ FedRAMP HIGH-baseline profile
- ❑ FedRAMP MODERATE-baseline profile
- ❑ FedRAMP LOW-baseline profile
- ❑ FedRAMP control catalog

# Thank you



The **OSCAL project** is developed openly on GitHub.com.

## Repository:

<https://github.com/usnistgov/OSCAL>

**Project Website:** <https://www.nist.gov/oscal>

## How to Contribute:

<https://pages.nist.gov/OSCAL/contribute/>

We welcome contributions to this project.

Contact us via email at: [oscal@nist.gov](mailto:oscal@nist.gov)

## Questions?



# OSCAL and FedRAMP

**Ashley Mahan**

Director, Federal Risk and Authorization Management Program (FedRAMP) and Secure Cloud Portfolio

General Services Administration (GSA)

24

# Break

On your own

**Note: An escort is needed beyond building 215**



# Agenda

Time	Topic
8:00 am	Registration and Networking
9:00 am	Welcome, Introduction to Workshop
9:10 am	<b>What is OSCAL and Who Needs It</b> Michaela Iorga, OSCAL co-lead NIST
10:00 am	<b>OSCAL and FedRAMP</b> Ashley Mahan, Director, FedRAMP, GSA
10:30 am	Break
10:45 am	<b>Tools That Understand OSCAL</b> Andrew Weiss, GitHub and Wendell Piez, OSCAL team member, NIST
11:20 am	<b>OSCAL Roadmap</b> David Waltermire, SCAP Lead and OSCAL co-lead, NIST
Noon	Lunch
1:30 pm	<b>OSCAL Catalog and Profile Models</b> Brian Ruf, OSCAL team member, Noblis
2:45 pm	Break
3:00 pm	<b>OSCAL Component and System Security Plan Models</b> David Waltermire, SCAP lead and OSCAL co-lead, NIST
4:15 pm	<b>Discussion: Tomorrow is Today – The Need for Automation</b> Moderator: David Waltermire, SCAP lead and OSCAL co-lead, NIST
4:50 pm	Closing Remarks
5:00 pm	Adjourn

# Tools That Understand OSCAL Today and Tomorrow

Andrew Weiss

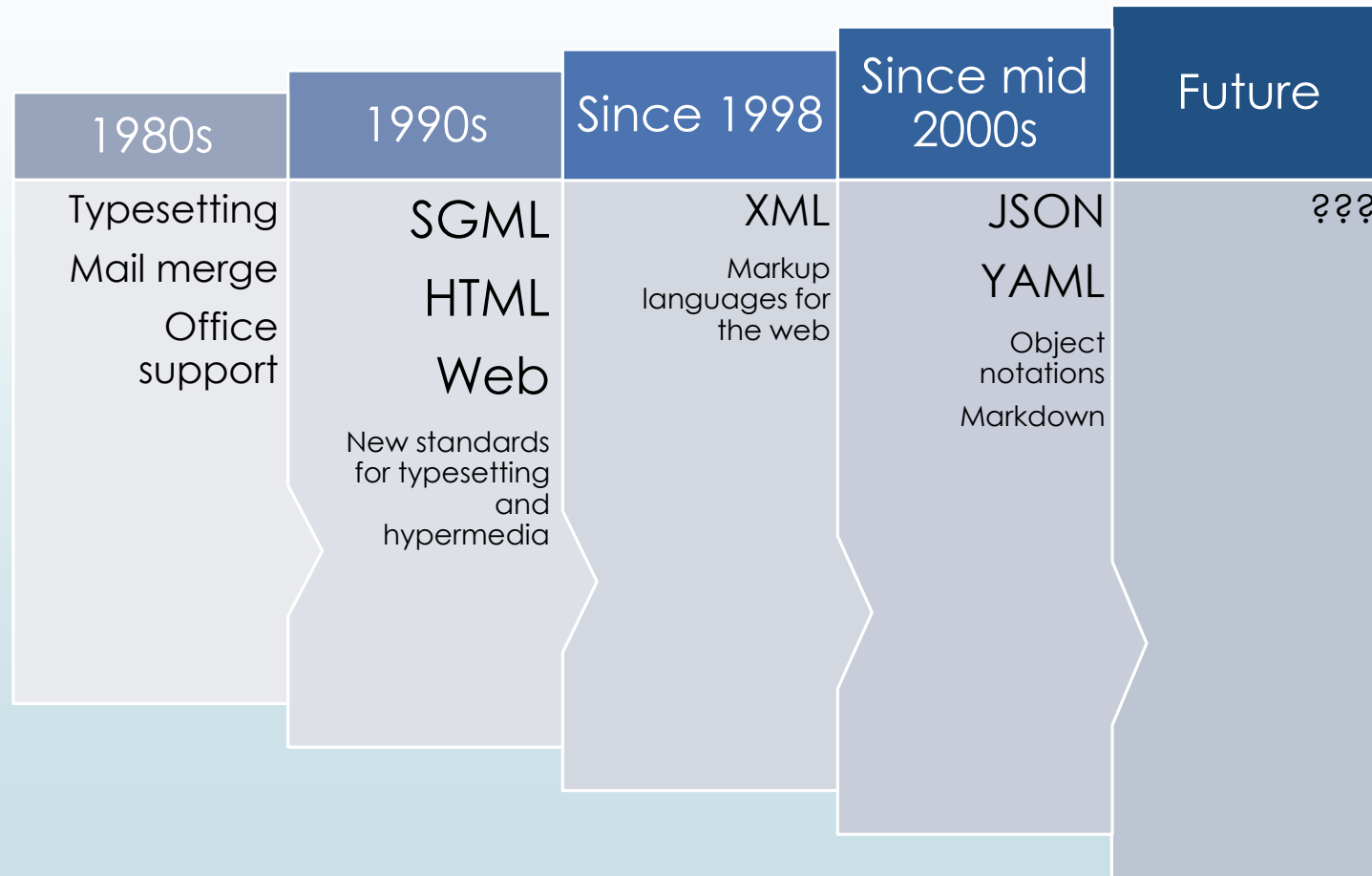
GitHub

Wendell Piez

OSCAL Team Member

National Institute of Standards and Technology (NIST)

# Technologies emerge from practice



# Standards are only half the story

The other half is how we get things done

- Many workflows are still document based
- PDF, MS Word / Excel / Office
- Legacy systems of all kinds
- There are functional gaps but they are unevenly felt
  - Tools, methods and processes are good enough for many things
  - Only not so good for some things
- Many reasons for this inertia

# Three models of software development

## Commercial / proprietary

- Capability is licensed as IP
- Viable and proven business model
- “Black box” – clean division of responsibilities between developer/provider and user/customer
- Division of responsibility helps indemnify parties

## Open source

- Easier to acquire, adapt and run
- More transparent, hence (potentially) more secure(able)
- Easier maintenance? That depends
- Sustainable? How do we support it long-term?

## Home built / DIY

- Enables close fit of solutions to requirements
- Can leverage both the other models
- But: an organization responsible for X must now support dev/ops activity as well – sometimes a challenge

# OSCAL Approach

- ▶ Our models are defined independently of their expression
- ▶ Like a mold for making cups from glass or ceramic
- ▶ This means we are agnostic regarding “best” technical implementation
  - ▶ Support multiple approaches
    - ▶ Today, XML and/or JSON (and YAML)
    - ▶ Java, XSLT/XQuery, Python, Ruby, Rust, you name it
    - ▶ Tomorrow, the best technologies available
  - ▶ Support conversion between alternative representations of the same information
    - ▶ System A prefers XML
    - ▶ System B prefers JSON
    - ▶ Commodity XML->JSON converter means it doesn't matter
  - ▶ Handle different aspects of the problem with different tools
- ▶ We can provide demos, testing and reference implementations



# Needs for commodity tools

- ▶ Validate
  - ▶ Is my OSCAL actually OSCAL?
- ▶ Convert among alternative representations
- ▶ Viewing / publishing – “human readable”
  - ▶ In applications
  - ▶ In print / PDF
- ▶ Basic OSCAL functionality
  - ▶ E.g., represent a catalog as tailored by a profile
  - ▶ Editing, packaging ...
- ▶ *We have prototypes*

# Commercial opportunities

## Tools and services

- ▶ Integration with capable systems
  - ▶ OSCAL import / export
- ▶ Analytics
  - ▶ OSCAL crunching
  - ▶ Gap analysis
- ▶ High performance or scale requirements
- ▶ Data acquisition
  - ▶ Convert Word/Excel data into OSCAL
- ▶ Integration with related/complementary technology stacks
  - ▶ SCAP (scanners and security tools)
  - ▶ NISO STS, DITA (product documentation)



# Multiple Implementations = A Good Thing

not many OSCALs but many ways to use OSCAL

- ▶ A robust standard implies different systems can do different things
  - ▶ And still exchange data
  - ▶ Across and within organizational boundaries
- ▶ To do different things we welcome a variety of tools
- ▶ We aim to encourage new applications by
  - ▶ Providing support for the basics
    - ▶ Jump start anyone
    - ▶ Provide a baseline to validate functionality
  - ▶ Offering early wins and ROI
  - ▶ Creating demand for more of the same
    - ▶ Advanced functionality
    - ▶ Customization
    - ▶ Specialized services
  - ▶ Helping ensure network effects

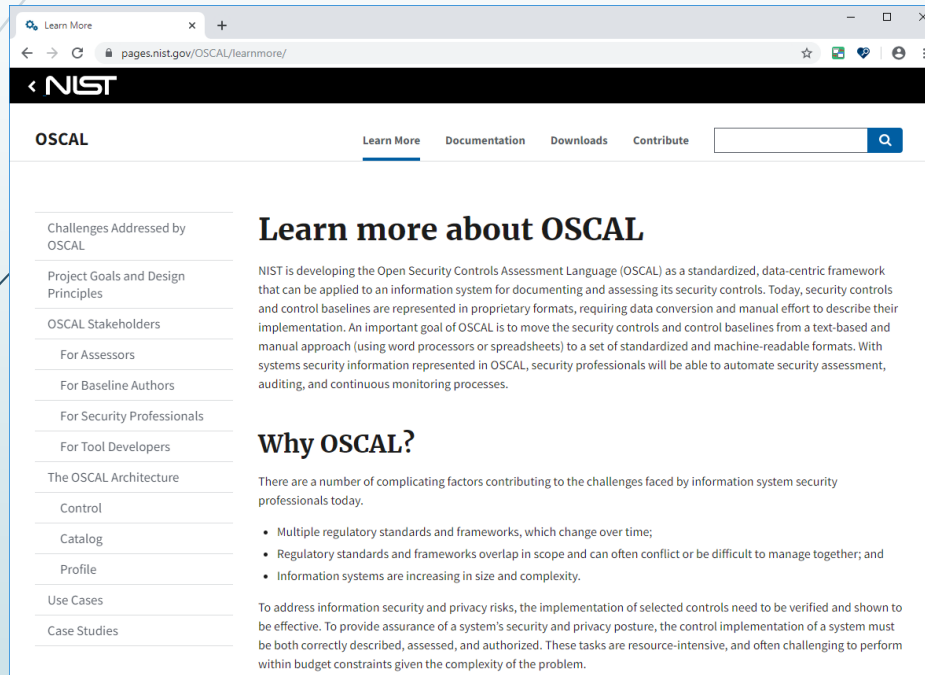
# How to Catalyze a Market



- Share methods and means
- Develop consensus and common understanding

- Break complex problems into simpler pieces
- Provide actual solutions to do more for less

# Resources for OSCAL Developers



**How else can OSCAL help builders?**

The **OSCAL project** is developed openly on GitHub.com

**Repository:** <https://github.com/usnistgov/OSCAL>

**Project Website:** <https://www.nist.gov/oscal>

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

**Chat with Us on Gitter:** <https://gitter.im/usnistgov-OSCAL/Lobby>

**We welcome contributions to this project.**

**Feedback? Questions?**

# Lunch with the Devs

NIST OSCAL team is hosting a bi-weekly, one-hour teleconference starting on Thursday, December 5th 2019 @ noon EST.

## Goals:

- ▶ To increase communication with the OSCAL community.
- ▶ Keep the community up-to-date on what we are working on.
- ▶ Discuss issues that we need community feedback on or help with.
- ▶ Answer questions, and discuss new proposed features or identified defects.

**More Info:** <https://pages.nist.gov/OSCAL/contribute/devlunch/>

# Hackathon

- ▶ We are hosting a hackathon on November 6<sup>th</sup> and 7<sup>th</sup>
- ▶ Bring an OSCAL-related project you would like to work on
  - ▶ You do not have to share code with other participants, although open source is welcome
  - ▶ Or pick one of our ideas
- ▶ The NIST OSCAL team will be in the room to help and answer any questions

## Project Ideas:

- ▶ Build HTML or PDF converters for Catalog, Profile, or SSP content
- ▶ Build a report generator that uses OSCAL data
- ▶ Build OSCAL import / export capabilities into an existing tool
- ▶ Work on an open issue: <https://git.io/Je2zO>

# Hackathon

## Agenda

Time	Topic
<b>Day 1 – November 6<sup>th</sup></b>	
<b>8:30am</b>	Registration
<b>9:00am</b>	Welcome and Logistics
<b>9:20am</b>	Form teams and start work
<b>Noon</b>	Lunch
<b>1:00pm</b>	Resume work
<b>5:00pm</b>	Adjourn for the day

Time	Topic
<b>Day 2 – November 7<sup>th</sup></b>	
<b>8:30am</b>	Registration
<b>9:00am</b>	Resume Work
<b>Noon</b>	Lunch
<b>1:00pm</b>	Resume work
<b>4:00pm</b>	Team Presentations(optional)
<b>5:00pm</b>	Adjourn

# OSCAL Roadmap

Future directions of the Open Security Controls  
Assessment Language

David Waltermire

SCAP Lead and OSCAL Co-Lead

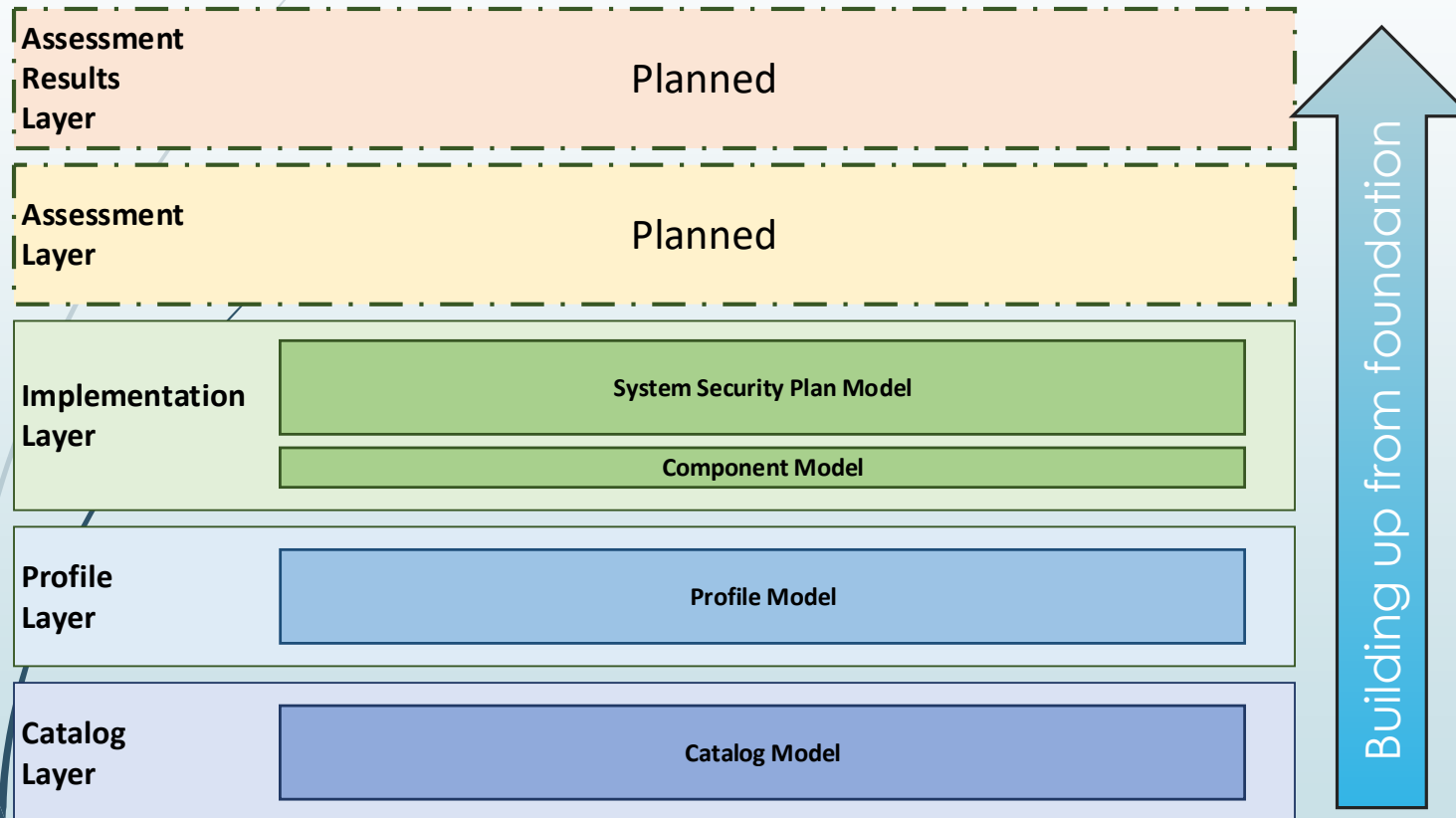
National Institute of Standards and Technology (NIST)

# OSCAL Development

- OSCAL is being designed and developed over a series of development epics
  - Using an incremental, agile approach
  - Each epic consists of a series of sprints
  - focused on reaching a defined milestone.
- This approach allows the project team to provide increased value over time at an accelerated pace, by focusing on an 80% solution (Minimally Viable Product (MVP)) that can be implemented in 20% of the time.



# OSCAL Layers & Models



## OSCAL is architected in layers

- ▶ The lowest layer is foundational
- ▶ Each higher layer builds on layer(s) below it
- ▶ OSCAL development is following this bottom up approach
  - ▶ Allows lower layers to be used, while higher layers are developed
  - ▶ Lower layers can be enhanced based on high-layer information needs
  - ▶ Ensures that data provided in lower layers can be used to meet the information needs in higher layers

# OSCAL Version 1 Milestones

Milestone	Focus	Sprints	Status	Date
<b>Milestone 1</b>	Catalog and Profile Models	<b>1 to 21</b>	<b>Completed</b>	<b>6/15/2019</b>
<b>Milestone 2</b>	System Security Plan (SSP) Model	<b>6 to 23</b>	<b>Completed</b>	<b>10/1/2019</b>
<b>Milestone 3</b>	Component Definition Model	<b>6 to ~29</b>	<b>In Progress</b>	<b>March 2020</b>
<b>Full Release</b>	Development of a web-based specification	<b>24 to ~33</b>	<b>In Progress</b>	July 2020
<b>Ongoing Maintenance</b>	Minor and bugfix releases as needed	Additional Sprints	Planned	Ongoing

# OSCAL 1.0.0 Milestone 1

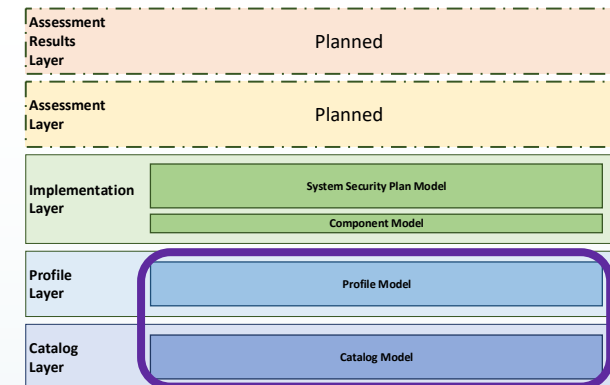
**Focus:** Development of Catalog and Profile Models

**Status:** Completed on 6/15/2019

**Sprints:** 1 through 21

## Challenges Addressed:

- ▶ Developing a catalog model to represent different control catalog representations (e.g., SP 800-53, ISO/IEC 27002, and COBIT).
  - ▶ Statement-level identification supporting granular downstream implementation documentation.
  - ▶ Support for control parameters that can be tailored in baselines and implementations.
- ▶ Unified profile model that works for all control baselines.
  - ▶ Control and parameter customization.
  - ▶ Traceability of baselines back to control catalogs supporting granular implementation documentation.



# OSCAL 1.0.0 Milestone 2

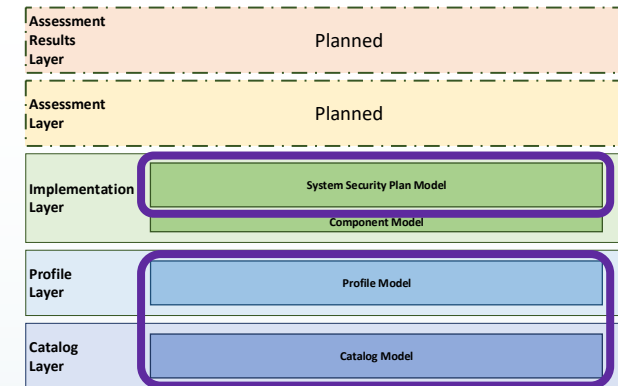
**Focus:** Development of the System Security Plan (SSP) Model

**Status:** Completed on 10/1/2019

**Sprints:** 6 through 23

## Challenges Addressed:

- Alignment with SP 800-18 rev 1 and rev 2 (under development).
- Providing a rich model for documenting SSPs, with flexibility to support datacenter, cloud, and enterprise scenarios.
- Support for flat/system-level documentation of control implementations (current practice).
- Support for granular, component-based documentation of control implementations (evolving practice).



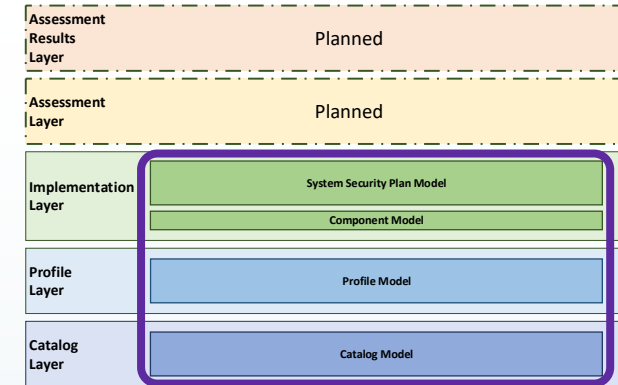
# OSCAL 1.0.0 Milestone 3

**Focus:** Development of the Component Definition Model

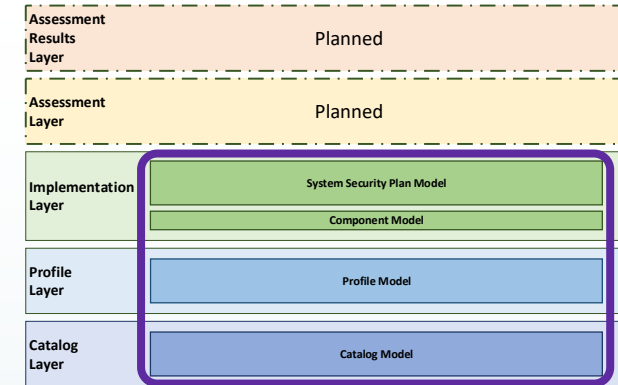
**Status:** Under development (March 2019)     **Sprints:** 6 through ~29

## Challenges to Address:

- Provide a component definition model that can describe controls implemented in asset types including: **hardware, software, services, policies, procedures, plans**, or anything else that *enables a system to satisfy a control*.
- Identify the controls satisfied by the asset.
  - For some asset types, this may include defining multiple options, which satisfy controls differently (e.g., software configurations).
  - Integrate control parameters, and align with configurable options.
- Align with SSP model to allow system implementers to import asset information into a SSP document.



# OSCAL 1.0.0 Full Release



**Focus:** Finalization of OSCAL 1.0.0 and development of the OSCAL v1 Specification

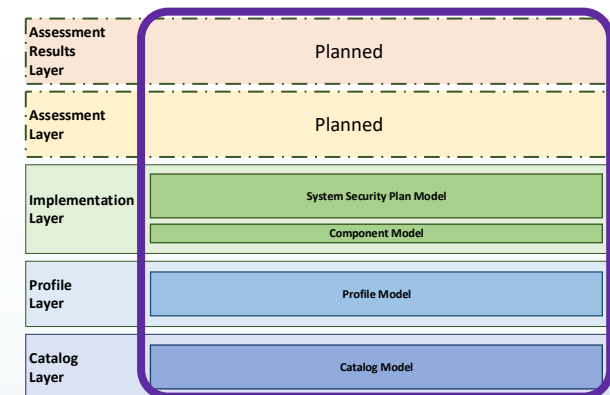
**Status:** Under development (July 2019) **Sprints:** 24 through ~33

## Challenges to Address:

- Create an initial web-based, formal specification for the OSCAL models.
- Fully document use of the OSCAL Catalog, Profile, Component, and SSP models.
  - Describe the semantics, allowed values, and recommended use of each data element.
  - Document processing rules for OSCAL content (e.g., Profile Resolution).
- Incorporate feedback from the OSCAL community based on early implementation and use.

# Beyond OSCAL 1.0.0

- OSCAL v2 will focus on the Assessment and Assessment Results layers.
- Support for assessment plans, auditing, and assessment/audit results.
  - Allow for the definition of assessment workflows (e.g., business impact assessment, privacy assessment, etc.)
  - Attachment of assessment artifacts
- Focus on automated collection of human- and asset-related information.
  - Use of the Open Checklist Interactive Language (OCIL) for human-oriented data collection.
  - Integration with asset, configuration management, and orchestration frameworks (e.g., Puppet, Chef, Ansible, Kubernetes) and the Security Content Automation Protocol (SCAP) for assessment data collection.



- We will continue to maintain OSCAL 1.0.0 as we work on OSCAL v2
  - Allows implementation of OSCAL v1, while we work on OSCAL v2.
  - We can make backward-compatibility breaking changes in v2 without affecting v1 implementations.
  - New features added in v2 can be back ported as optional features of v1

# Discussion Questions

- ▶ Is this roadmap clear?
- ▶ Do you find this incremental approach to be useful?
  - ▶ Does this help or hurt adoption of OSCAL?
  - ▶ At what Milestone will you consider implementation of OSCAL?
- ▶ What layers are most important to you? Why?

The **OSCAL project** is developed openly on GitHub.com.

**Repository:** <https://github.com/usnistgov/OSCAL>

**Project Website:** <https://www.nist.gov/oscal>

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

**We welcome contributions to this project.**

## Questions?

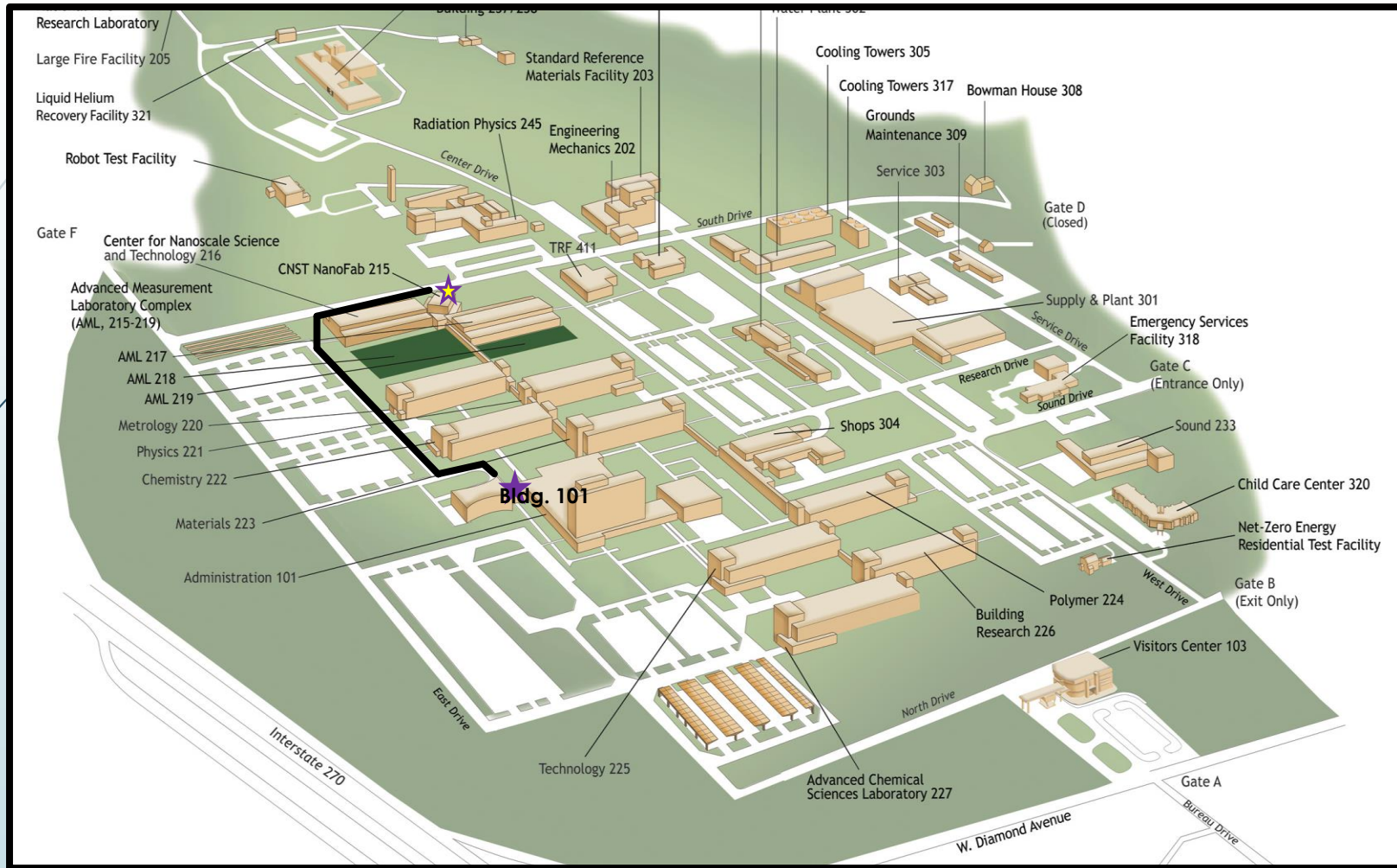


## Lunch

On your own, Cafeteria in Building 101

Will resume the workshop at **1:30pm**

# Lunch – On your own or NIST cafeteria



To reach the NIST cafeteria, you will need to walk outside to building 101.

# Agenda

Time	Topic
8:00 am	Registration and Networking
9:00 am	Welcome, Introduction to Workshop
9:10 am	<b>What is OSCAL and Who Needs It</b> Michaela Iorga, OSCAL co-lead NIST
10:00 am	<b>OSCAL and FedRAMP</b> Ashley Mahan, Director, FedRAMP, GSA
10:30 am	Break
10:45 am	<b>Tools That Understand OSCAL</b> Andrew Weiss, GitHub and Wendell Piez, OSCAL team member, NIST
11:20 am	<b>OSCAL Roadmap</b> David Waltermire, SCAP Lead and OSCAL co-lead, NIST
Noon	Lunch
1:30 pm	<b>OSCAL Catalog and Profile Models</b> Brian Ruf, OSCAL team member, Noblis
2:45 pm	Break
3:00 pm	<b>OSCAL Component and System Security Plan Models</b> David Waltermire, SCAP lead and OSCAL co-lead, NIST
4:15 pm	<b>Discussion: Tomorrow is Today – The Need for Automation</b> Moderator: David Waltermire, SCAP lead and OSCAL co-lead, NIST
4:50 pm	Closing Remarks
5:00 pm	Adjourn

# OSCAL Catalog and Profile Models

Brian Ruf

OSCAL Team Member

Noblis

# Presentation Agenda

## OSCAL Overview

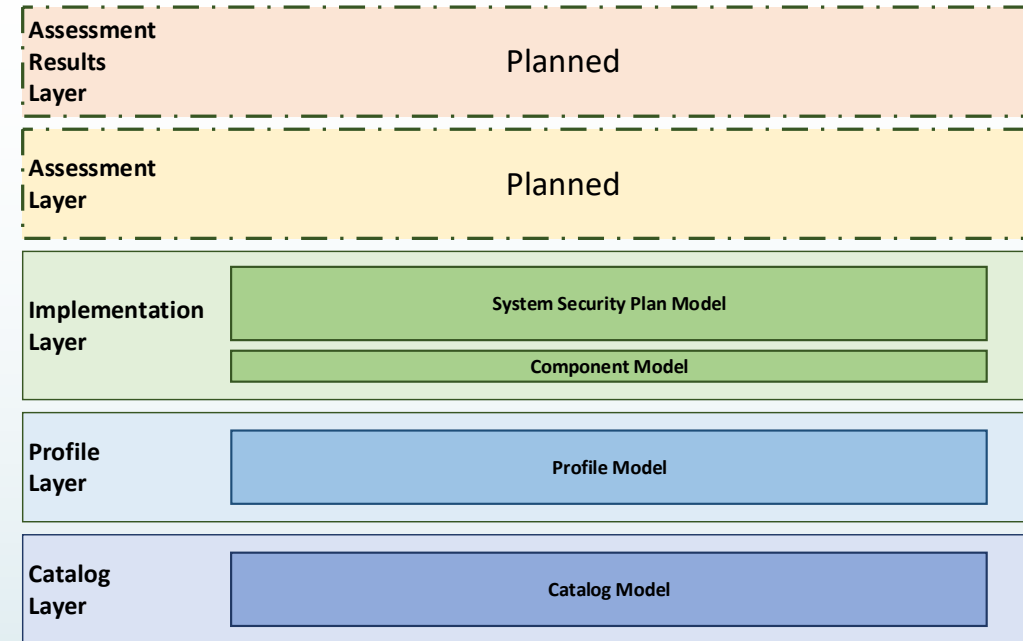
- Layers and Models
- Keeping In Sync
- Anatomy of an OSCAL File

## CATALOG MODEL

- Goals
- Scope
- Authors & Consumers

## PROFILE MODEL

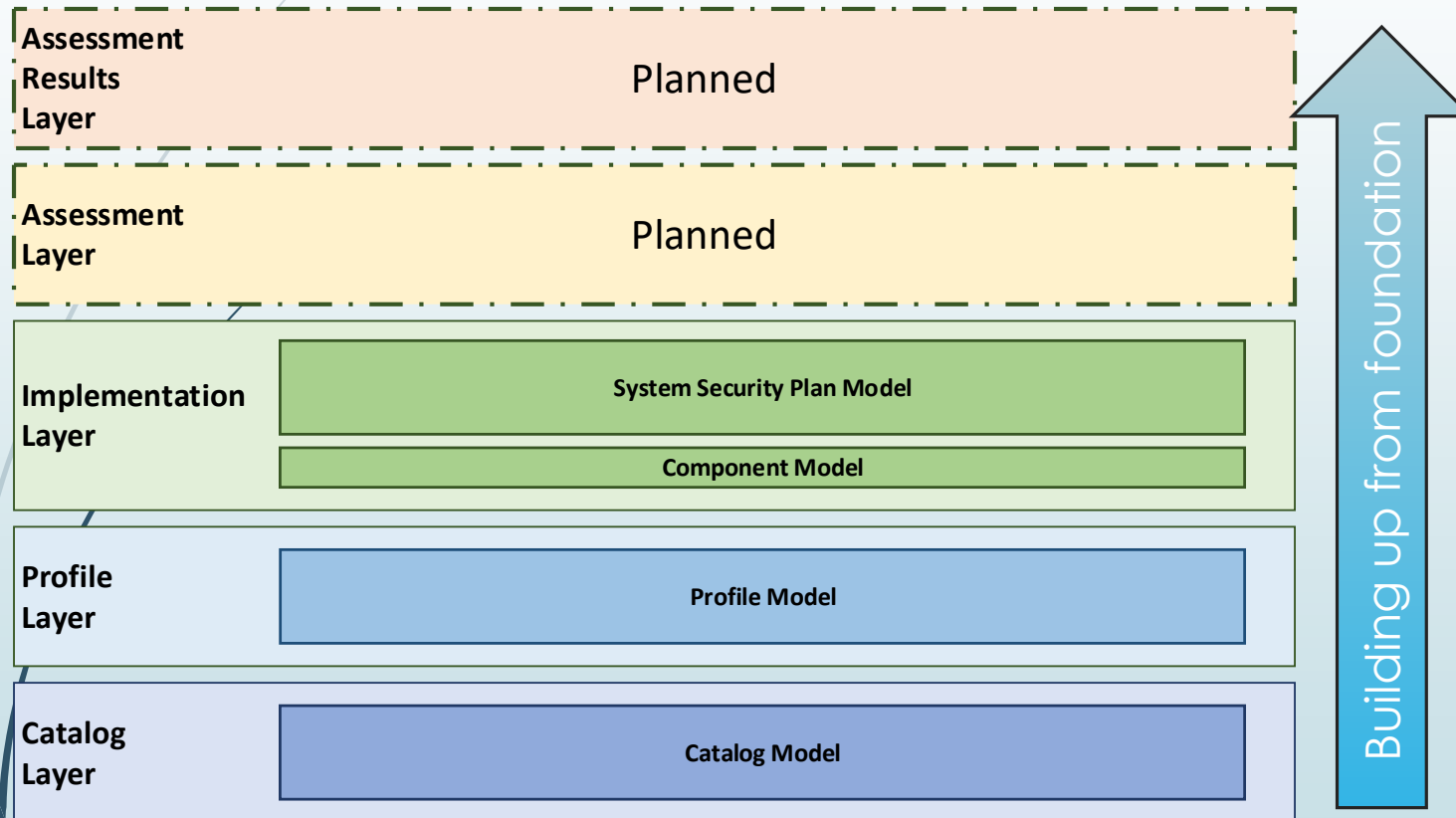
- Goals
- Scope
- Authors & Consumers



## WORKING WITH CATALOGS AND PROFILES

- Case Studies / Examples
- Resources
- How You Can Help
- Questions?

# OSCAL Layers & Models



## OSCAL is architected in layers

- The lowest layer is foundational
- Each higher layer builds on layer(s) below it

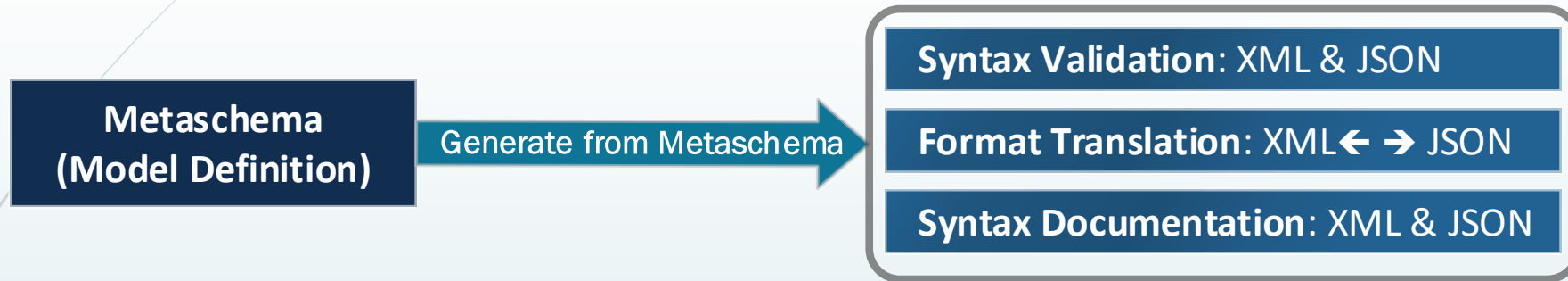
- We will discuss this from the bottom up:

- Catalog
- Profile

- Implementation: Component and System Security Plan

- Looking Forward: Assessment and Assessment Results

# OSCAL Syntax Stays In Sync



## For Each Model:

- Syntax is defined in one place for all supported formats (Metaschema)
  - Today: XML and JSON
  - Future: YAML
- For each supported format, the CI/CD pipeline generates:
  - Validation Schema (XML, JSON)
  - Converters (XML -> JSON, JSON -> XML)
  - Documentation (XML, JSON)
- **This ensures all formats align and interoperate as expected**

# OSCAL Syntax Validation



## XML Syntax Validation `oscal_[mode]_schema.xsd`

- ▶ OSCAL XML Schema Files:
  - ▶ <https://github.com/usnistgov/OSCAL/tree/master/xml/schema>
- ▶ XML Schema Definition Language (XSD) 1.1
  - ▶ Standard: <https://www.w3.org/TR/xmlschema11-1/>



## JSON Syntax Validation `oscal_[mode]_schema.json`

- ▶ OSCAL JSON Schema Files:
  - ▶ <https://github.com/usnistgov/OSCAL/tree/master/json/schema>
- ▶ JSON Schema, draft 07
  - ▶ Standard: <https://json-schema.org/>



# OSCAL Format Translation



**Format Translation:** XML → JSON  
`oscal_[model]_xml-to-json-converter.xsl`

- XML to **JSON**: <https://github.com/usnistgov/OSCAL/tree/master/json/convert>



**Format Translation:** JSON → XML  
`oscal_[model]_json-to-xml-converter.xsl`

- JSON to **XML**: <https://github.com/usnistgov/OSCAL/tree/master/xml/convert>

**Translation (in either direction) requires support for:**

- Extensible Stylesheet Language Transformation (XSLT) 3.0
  - Standard: <https://www.w3.org/TR/2017/REC-xslt-30-20170608/>
- XML Path Language (XPath) 3.1
  - Standard: <https://www.w3.org/TR/xpath-31/>

# Anatomy of an OSCAL File

## Every OSCAL File

### Root Element:

[ `catalog` | `profile` | `component` | `system-security-plan` ]  
Universally Unique Identifier (UUID)

### Metadata (Required)

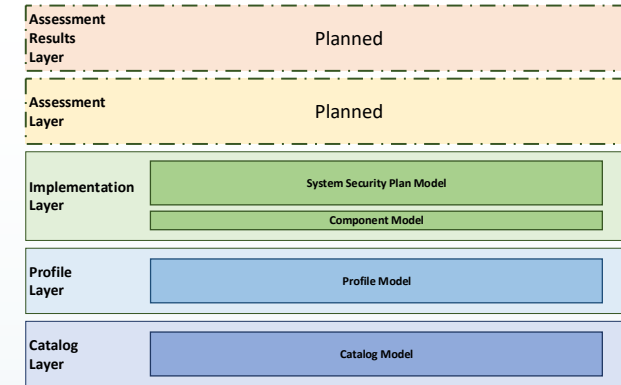
At the start of every OSCAL file, regardless of model.  
**Syntax is the same in every model.**

### Body

**Syntax differs by model.**

### Back Matter (Optional)

At the end of every OSCAL file, regardless of model/layer.  
**Syntax is the same in every model.**



## Every OSCAL File has:

- A root element at the top
  - Automation tools should use this to determine which model is in use
- The root element contains a universally unique identifier (UUID)
  - Identifies the revision of the document
  - This should be updated every time the content or structure changes
- A **metadata** section immediately following the root element & UUID
- A model-specific **body** in the middle
- A **back matter** section at the end

# Metadata

## XML Example

OSCAL File

Metadata

Body

Back Matter

- **Document Properties**
  - Title, Version, Publication Date
  - Last Modified Date, Document ID
  - Keywords
- **Actors and Roles**
  - Party/Org, Party/Person
  - Roles and Role Assignments
- **OSCAL Administrivia**
  - OSCAL (Syntax) Version
  - Alternate and Canonical Information

Red underline denotes a mandatory field

```
<catalog xmlns="http://csrc.nist.gov/ns/oscal/1.0"
          id="uuid-47fdefdb-dc1a-4040-9f27-b517a16b06d2">
  <metadata>
    <title>Document Title</title>
    <last-modified>2019-11-05T14:00:00.000-04:00</last-modified>
    <version>1.0</version>
    <oscal-version>1.0.0-milestone1</oscal-version>

    <role id="creator">
      <title>Document creator</title>
    </role>

    <party id="afla">
      <org>
        <org-name>A Four Letter Agency (AFLA)</org-name>
      </org>
    </party>

    <responsible-party role-id="creator">
      <party-id>afla</party-id>
    </responsible-party>
  </metadata>
  <!-- [cut] -->
</catalog>
```

# Metadata

## JSON and YAML Examples

OSCAL File

Metadata

Body

Back Matter

```
{
  "catalog": {
    "id": "uuid-47fdefdb-dc1a-4040-9f27-b517a16b06d2",
    "metadata": {
      "title": "Document Title",
      "last-modified": "2019-11-05T14:00:00.000-04:00",
      "version": "1.0",
      "oscal-version": "1.0.0-milestone2",
      "roles": [
        {
          "id": "creator",
          "title": "Document creator"
        }
      ],
      "parties": {
        "id": "afla",
        "org": {
          "org-name": "A Four Letter Agency (AFLA)"
        }
      },
      "responsible-parties": {
        "creator": {
          "party-ids": "afla"
        }
      }
    }
  }
}
```

```
catalog:
  id: "uuid-47fdefdb-dc1a-4040-9f27-b517a16b06d2"
  metadata:
    title: "Document Title"
    last-modified: "2019-11-05T14:00:00.000-04:00"
    version: "1.0"
    oscal-version: "1.0.0-milestone2"
    roles:
      - id: "creator"
        title: "Document creator"
  parties:
    id: "afla"
    org:
      org-name: "A Four Letter Agency (AFLA)"
  responsible-parties:
    creator:
      party-ids: "afla"
```

# Back Matter

## XML Example

OSCAL File

Metadata

Body

Back Matter

- **Back Matter is not required**
- **If present, Back Matter must appear last in the document**
- **Bibliographic References**
  - Citations
- **Resources**
  - Links to external resources
    - A single resource can have several links
  - **and/or** embedded Base64 attachments
  - Referenced within an OSCAL document using a URI fragment **"#id"**

```

<catalog xmlns="http://csrc.nist.gov/ns/oscal/1.0"
  id="uuid-47fdefdb-dc1a-4040-9f27-b517a16b06d2">
  <!-- [cut] -->
  <back-matter>
    <citation id="ref079">
      <target>https://doi.org/10.6028/NIST.SP.800-53r4</target>
      <title>NIST Special Publication 800-53 Revision 4</title>
      <doc-id type="doi">10.6028/NIST.SP.800-53r4</doc-id>
    </citation>
    <resource id="resource-policy-at">
      <rlink media-type="application/pdf"
        href="http://intranet.af1a.gov/AT-Policy.pdf"/>
    </resource>
    <resource id="diagram">
      <desc>Main Diagram</desc>
      <rlink media-type="image/png"
        href="http://intranet.af1a.gov/main-diagram.png"/>
      <base64 filename='main_diagram.png'>
        wo1QTkIRFI--cut--ABJRU5Eq5CYMKC
      </base64>
    </resource>
  </back-matter>
</catalog>

```

# Back Matter

## JSON and YAML Examples

OSCAL File

Metadata

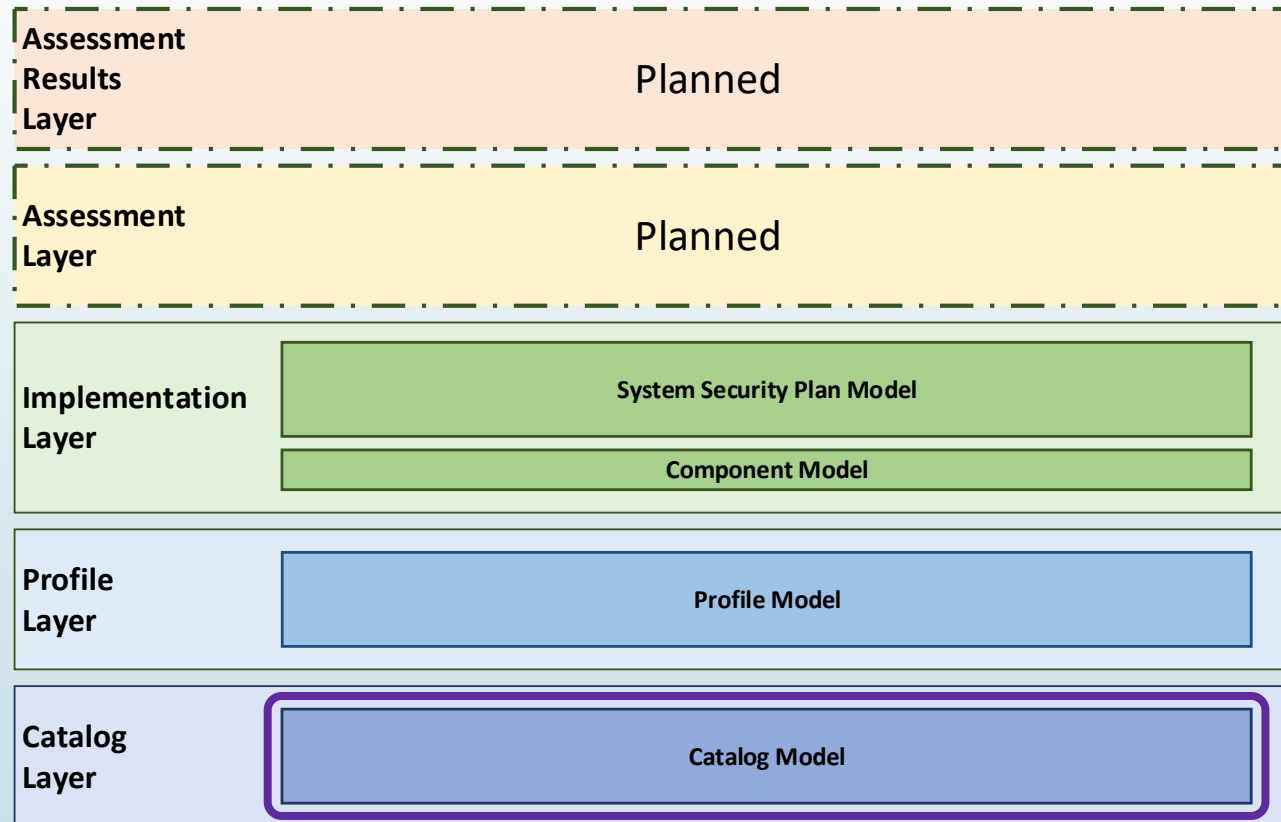
Body

Back Matter

```
{
  "catalog": {
    "id": "uuid-47fdefdb-dc1a-4040-9f27-b517a16b06d2",
    "back-matter": {
      "citations": [
        {
          "id": "ref079",
          "targets": "https://doi.org/10.6028/NIST.SP.800-53r4",
          "title": "NIST Special Publication 800-53 Revision 4",
          "document-ids": {
            "type": "doi",
            "identifier": "10.6028/NIST.SP.800-53r4"
          }
        }
      ],
      "resources": [
        {
          "id": "resource-policy-at",
          "rlinks": {
            "href": "http://intranet.afla.gov/AT-Policy.pdf",
            "media-type": "application/pdf"
          }
        },
        {
          "id": "diagram",
          "desc": "Main Diagram",
          "rlinks": {
            "href": "http://intranet.afla.gov/main-diagram.png",
            "media-type": "image/png"
          },
          "base64": {
            "filename": "main_diagram.png",
            "value": "wo1QTKIRFI--cut--ABJRU5Ewq5CYMKC"
          }
        }
      ]
    }
  }
}
```

```
---
catalog:
  id: "uuid-47fdefdb-dc1a-4040-9f27-b517a16b06d2"
  back-matter:
    citations:
      - id: "ref079"
        targets: "https://doi.org/10.6028/NIST.SP.800-53r4"
        title: "NIST Special Publication 800-53 Revision 4"
        document-ids:
          type: "doi"
          identifier: "10.6028/NIST.SP.800-53r4"
    resources:
      - id: "resource-policy-at"
        rlinks:
          href: "http://intranet.afla.gov/AT-Policy.pdf"
          media-type: "application/pdf"
      - id: "diagram"
        desc: "Main Diagram"
        rlinks:
          href: "http://intranet.afla.gov/main-diagram.png"
          media-type: "image/png"
        base64:
          filename: "main_diagram.png"
          value: "wo1QTKIRFI--cut--ABJRU5Ewq5CYMKC"
```

# Catalog Model



# Catalog Model

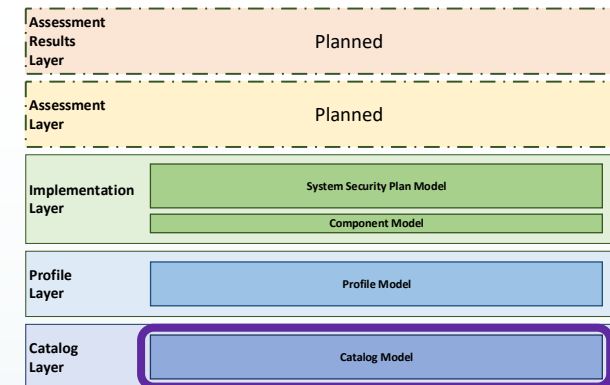
## Purpose and Goals

### Purpose

- Model control definitions from any cybersecurity framework
- Model control assessment objectives and activities

### Goals

- Provide a structure for expressing control detail at level of granularity that ensures machine-readable processing in every phase of the cybersecurity lifecycle
- Provide a foundation for referencing modeled control information across all other OSCAL models



### Examples of control catalogs that can be expressed in OSCAL:

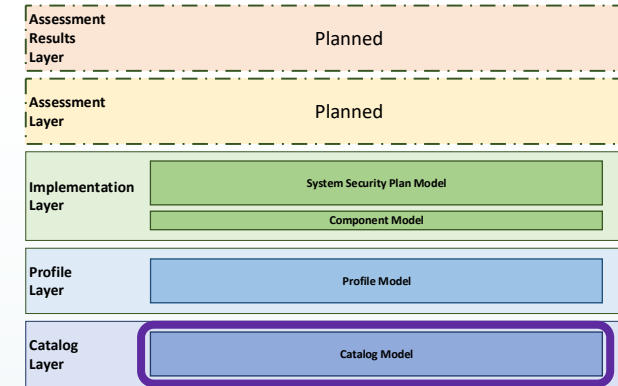
- NIST SP 800-53 & 53A
- ISO/IEC 27002
- COBIT 5



# Catalog Model

## Features

- ▶ Standardizes the semantics of control information
  - ▶ by normalizing the representation of control definitions
  - ▶ regardless of their originating governance frameworks
- ▶ Provides appropriate granularity to support automation across the cybersecurity lifecycle:
  - ▶ Provides identifiers down to the sub-statement level
  - ▶ Provides for in-line parameters
  - ▶ Represents control enhancements or child controls
  - ▶ Combines control definitions with assessment objectives and methods
  - ▶ Provides for grouping of controls into families
  - ▶ Allows for arbitrary relationships between controls



# Catalog Model

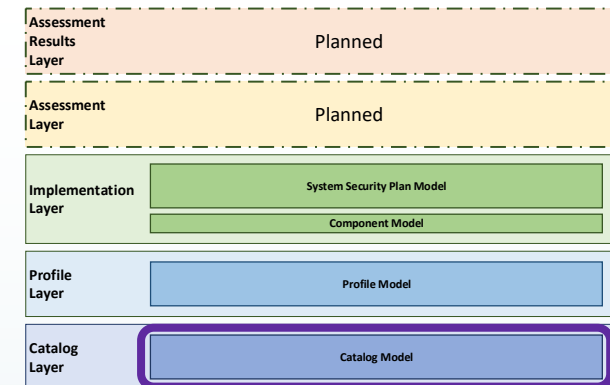
## Authors and Consumers

### Authors

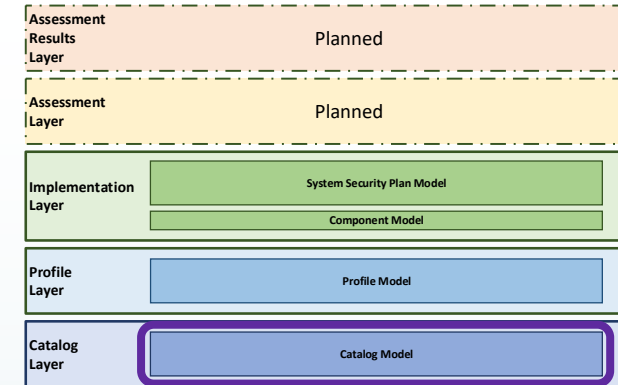
- ▶ Authors of cybersecurity governance frameworks, standards, practices, or guidelines.
- ▶ Organizations wishing to define organization-specific requirements, such as commercial organizations or government agencies.

### Consumers

- ▶ Anyone who must align with a framework such as NIST, ISO, or COBIT
- ▶ Anyone who must align with organization-defined controls.
- ▶ Authors or consumers of any higher layer in the OSCAL stack, such as:
  - ▶ baseline authors
  - ▶ component authors
  - ▶ SSP authors
  - ▶ assessors
  - ▶ authorizing officials



# Anatomy of a Catalog



## Groups

- Allow for grouping of controls
- Optional ID and class assignments
- For 800-53, used to express families

## Controls

- Can be directly under the root or grouped
- Must contain an ID
- Optional class
- Controls may be nested (control within a control)

**For 800-53, control enhancements are nested within the parent control**

```
<catalog xmlns="http://csrc.nist.gov/ns/oscal/1.0"
  id="uuid-47fdefdb-dc1a-4040-9f27-b517a16b06d2">
  <metadata />

  <group class="family" id="ac">
    <title>Access Control</title>
    <control class="SP800-53" id="ac-1">
      <!-- [control details omitted] -->
    </control>

    <control class="SP800-53" id="ac-2">
      <!-- [control details omitted] -->

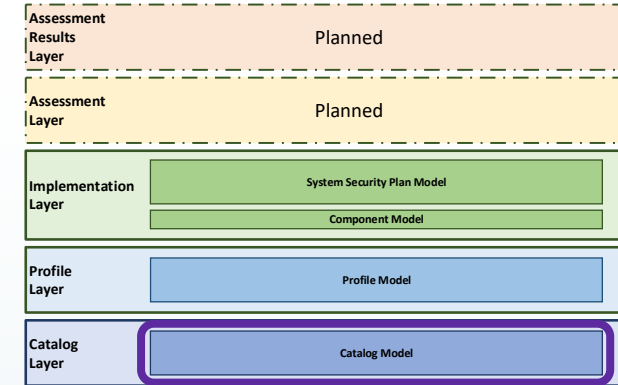
      <control class="SP800-53-enhancement" id="ac-2.1">
        <!-- [enhancement details omitted] -->
      </control>
    </control>
  </group>

  <control class="SP800-53" id="xx-1">
    <!-- [enhancement details omitted] -->
  </control>
</catalog>
```

# Anatomy of a Catalog Controls

## Controls Consist Of:

- Title (required)
- Parameters (optional)
- Properties (optional)
  - Simple name/value data intended for a single piece of information
- Parts (optional)
  - Parts contain more complex, nested information
  - Used to describe **Statements, Guidance, Assessment Objectives,** and **Assessment Activity** details
- Nested Controls (optional)



```
<catalog xmlns="http://csrc.nist.gov/ns/oscal/1.0"
  id="uuid-47fdefdb-dc1a-4040-9f27-b517a16b06d2">
  <metadata />
  <control class="SP800-53" id="ac-2">
    <!-- [control details omitted] -->
    <control class="SP800-53-enhancement" id="ac-2.1">
      <!-- [control details omitted] -->
    </control>
    <control class="SP800-53-enhancement" id="ac-2.2">
      <title>Removal of Temporary/Emergency Accounts</title>
      <param id="ac-2.2_prm_1"><!-- cut --></param>
      <param id="ac-2.2_prm_2"><!-- cut --></param>
      <prop name="label">AC-2(2)</prop>
      <prop name="sort-id">ac-02.02</prop>
      <part id="ac-2.2_smt" name="statement"><!-- cut --></part>
      <part id="ac-2.2_gdn" name="guidance"><!-- cut --></part>
      <part id="ac-2.2_obj" name="objective"><!-- cut --></part>
      <part name="assessment"><!-- cut --></part>
      <part name="assessment"><!-- cut --></part>
      <part name="assessment"><!-- cut --></part>
    </control>
  </control>
</catalog>
```

# Anatomy of a Catalog Statements

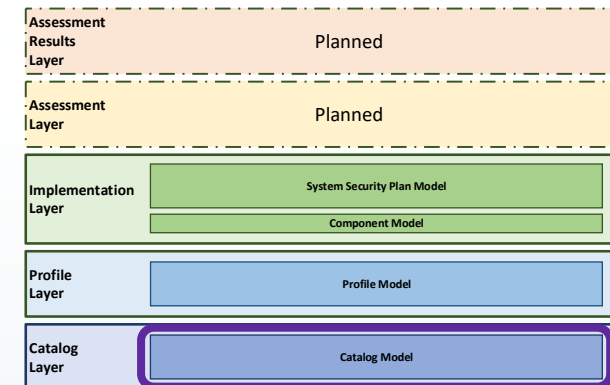
## Statements:

- Broken down into individual pieces
- Each piece can be referenced individually because it has its own ID
- Parameters are referenced by insert statements, with a parameter ID

### 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



```
<control class="SP800-53" id="ac-1">
  <title>Access Control Policy and Procedures</title>
  <param id="ac-1_prm_1"> ... </param>
  <param id="ac-1_prm_2"> ... </param>
  <param id="ac-1_prm_3"> ... </param>

  <part id="ac-1_smt" name="statement">
    <p>The organization:</p>

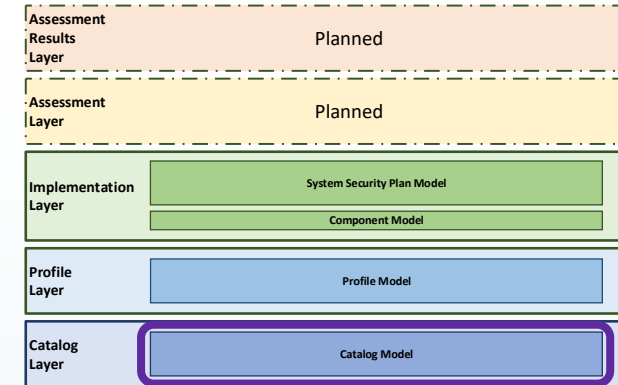
    <part id="ac-1_smt.a" name="item">
      <prop name="label">a.</prop>
      <p>Develops, documents, and disseminates to
        <insert param-id="ac-1_prm_1" />:</p>

      <part id="ac-1_smt.a.1" name="item">
        <prop name="label">1.</prop>
        <p>Procedures to facilitate ... and</p>

        <part id="ac-1_smt.a" name="item">
          <p>An access control policy that ... and</p>
        </part>
      </part>
    </part>
  </part>
</control>
```

# Anatomy of a Catalog Parameters

- Any control statement can reference any parameter
- Two parameter types:
  - Assignment
  - Select
- Parameters may be nested
  - In 800-53, Rev 4 there are a few cases of an Assignment parameter within a Select parameter
- Can have constraints expressed



```
<control class="SP800-53" id="ac-1">
  <title>Access Control Policy and Procedures</title>

  <param id="ac-1_prm_1">
    <label>organization-defined personnel or roles</label>
  </param>

  <param id="ac-1_prm_2">
    <label>organization-defined frequency</label>
    <constraint>at least annually</constraint>
  </param>

  <param id="ac-1_prm_3"> ... </param>

  <part id="ac-1_smt" name="statement">
    <p>The organization:</p>
    <part id="ac-1_smt.a" name="item">
      <prop name="label">a.</prop>
      <p>Develops, documents, and disseminates
        to <insert param-id="ac-1_prm_1" />:</p>
    </part>
  </part>
</control>
```

## AC-1 ACCESS CONTROL POLICY AND PROCEDURES

Control: The organization:

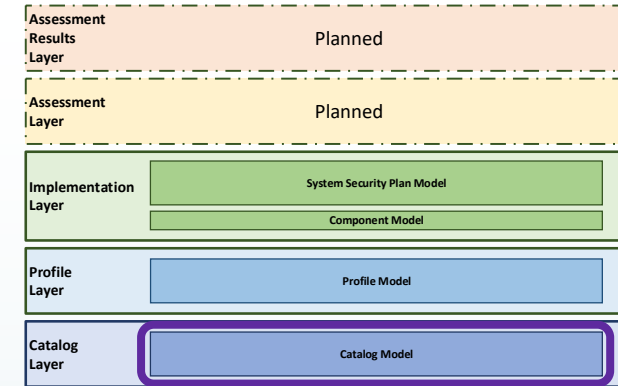
- a. Develops, documents, and disseminates to [*Assignment: organization-defined personnel or roles*]:

# Anatomy of a Catalog

## Other Notes

**Guidance, Objectives, and Assessment Activities** are expressed similar to statements.

For 800-53A, Rev 4, there are separate assessment parts for **Test, Examine, and Interview**.



```
<control class="SP800-53-enhancement" id="ac-2.2">
  <title>Removal of Temporary/Emergency Accounts</title>

  <param id="ac-2.2_prm_1"> ... </param>
  <param id="ac-2.2_prm_2"> ... </param>

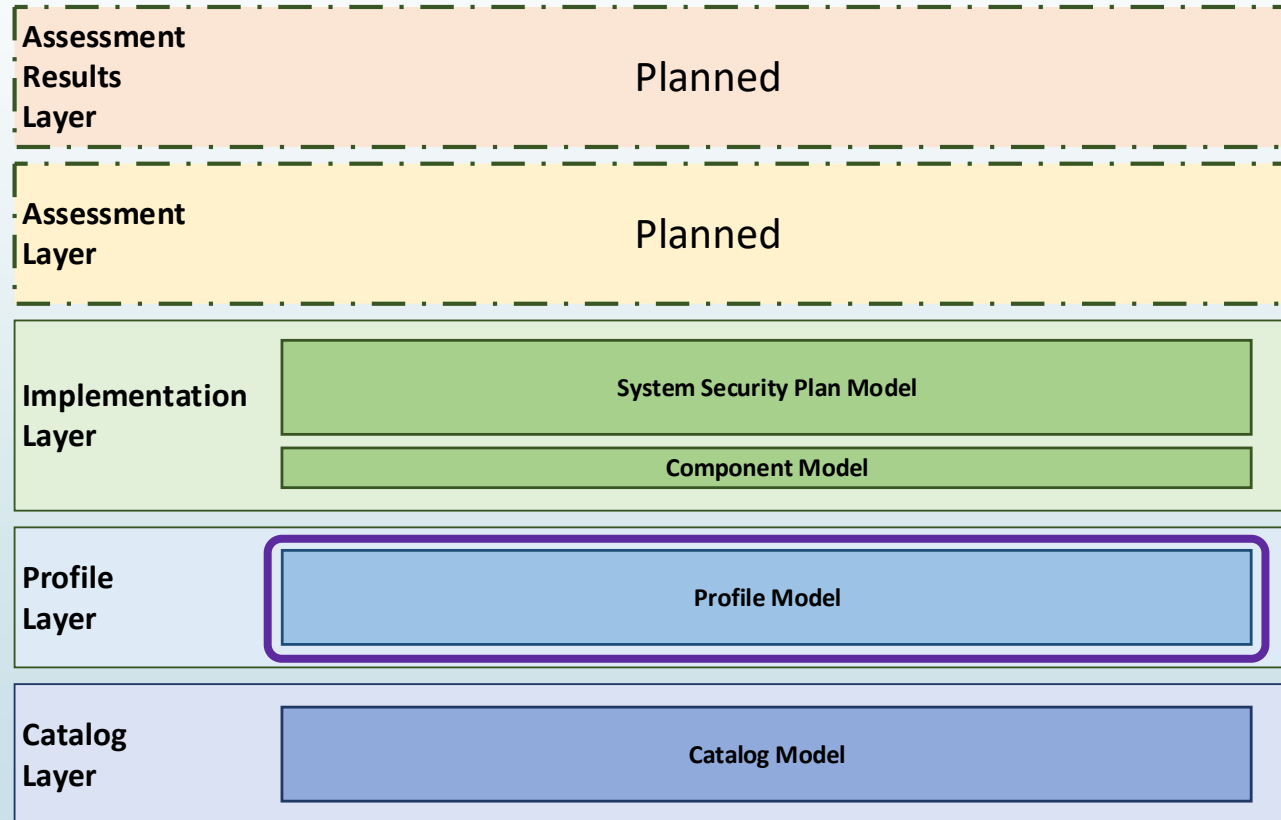
  <prop name="label">AC-2(2)</prop>
  <prop name="sort-id">ac-02.02</prop>

  <part id="ac-2.2_smt" name="statement"> ... </part>
  <part id="ac-2.2_gdn" name="guidance"> ... </part>
  <part id="ac-2.2_obj" name="objective"> ... </part>

  <part name="assessment"> ... </part>
  <part name="assessment"> ... </part>
  <part name="assessment">
    <prop name="method">TEST</prop>

    <part name="objects">
      <p>Automated mechanisms implementing information system auditing</p>
    </part>
  </part>
</control>
```

# Profile Model





# Profile Model

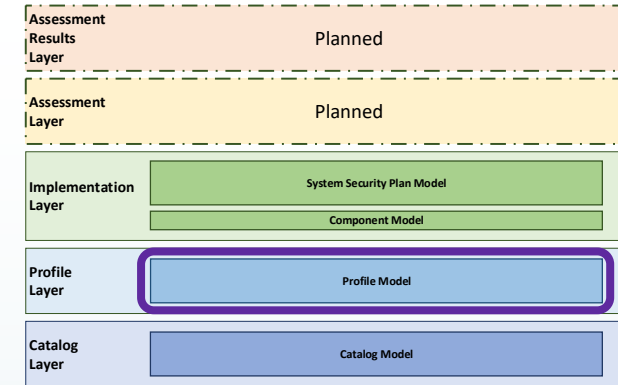
## Purpose and Goals

### Purpose

- Model control baselines/overlays – which are a customized subset of controls selected from one or more catalogs for a given purpose or security posture

### Goals

- Identify the controls to include from each relevant catalog
  - Possibly also from other profiles (baselines)
- Modify control parameters, statements, or assessment objectives/activities as needed
- Ensure transparency of control selection and customization, such that all changes can be traced back to its origination



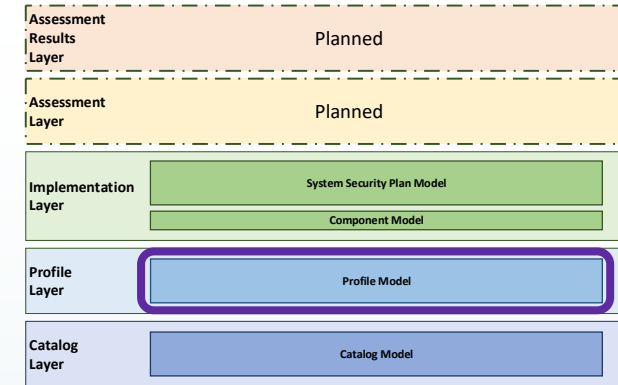
### Examples of control baselines that can be expressed in OSCAL:

- NIST High, Moderate, and Low Baselines
- FedRAMP High, Moderate, Low and Tailored Baselines
- Agency-specific Baselines
- Combined ISO/IEC-27002/NIST Moderate Baseline

# Profile Model

## Features

- ▶ Selection of controls
- ▶ Customization of controls, including:
  - ▶ Parameters
  - ▶ Statements
  - ▶ Assessment Objectives and Activities
- ▶ Traceability of controls through all tailoring, back to originating catalog



# Profile Model

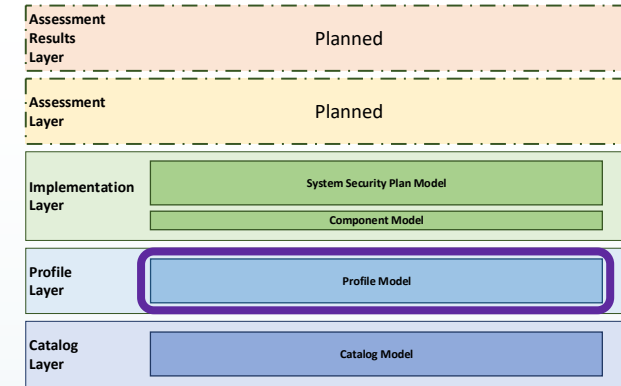
## Authors and Consumers

### Authors

- Control baseline/overlay authors, such as NIST or FedRAMP
- Organizations wishing to define organization-specific baselines/overlays such as commercial organizations or government agencies

### Consumers

- System owners who must align with a control baseline/overlay
- Authors of downstream profiles that reference existing profiles
- Authors or consumers of any higher layer in the OSCAL stack, such as:
  - component authors
  - SSP authors
  - assessors
  - authorizing officials



# Anatomy of a Profile

## Import

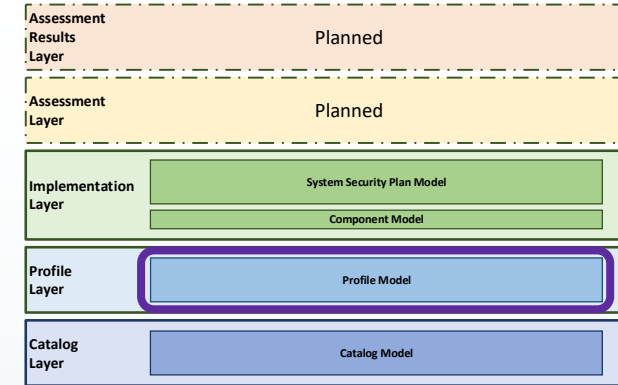
- Identifies a source a catalog or profile
- Identifies the controls to import from that catalog or profile
- One import per catalog or profile

## Merge

- Directives as to how a profile should “resolve” (merge) and organize the information it imports

## Modify

- Directives that add, change, or remove portions of an imported control



```
<profile xmlns="http://csrc.nist.gov/ns/oscal/1.0"
  id="uuid-47fdefdb-dc1a-4040-9f27-b517a16b06d2">
  <metadata /><!-- contents omitted -->

  <import href="NIST_SP-800-53_rev4_catalog.xml">
    <include>
      <call control-id="ac-2" />
      <call control-id="ac-2.1" />
    </include>
  </import>

  <merge /><!-- contents omitted -->

  <modify /><!-- contents omitted -->
</profile>
```

# Anatomy of a Profile

## Modify

### Modify (modify)

#### Parameters (set)

- Add a label, constraint, or value

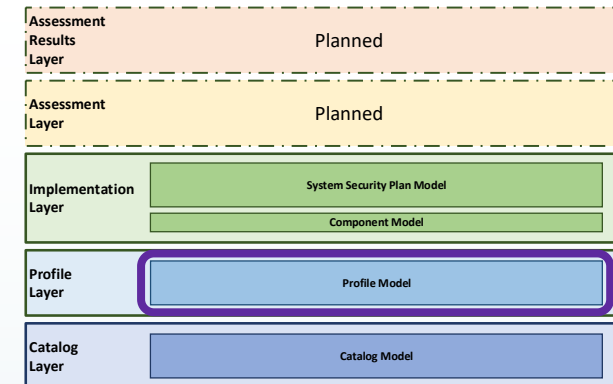
#### Remove Portions of a Control (remove)

- Identify what to remove using:

- ID (ref-id)
- name (name-ref)
- class (class-ref)
- element (item-name)

#### Add content to a control (add)

- Add statements, guidance, objectives, or assessment activities



```

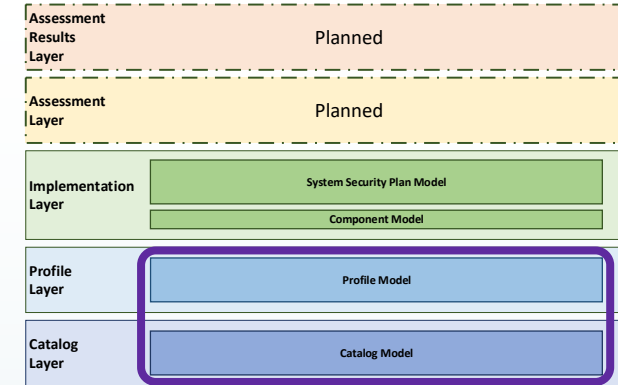
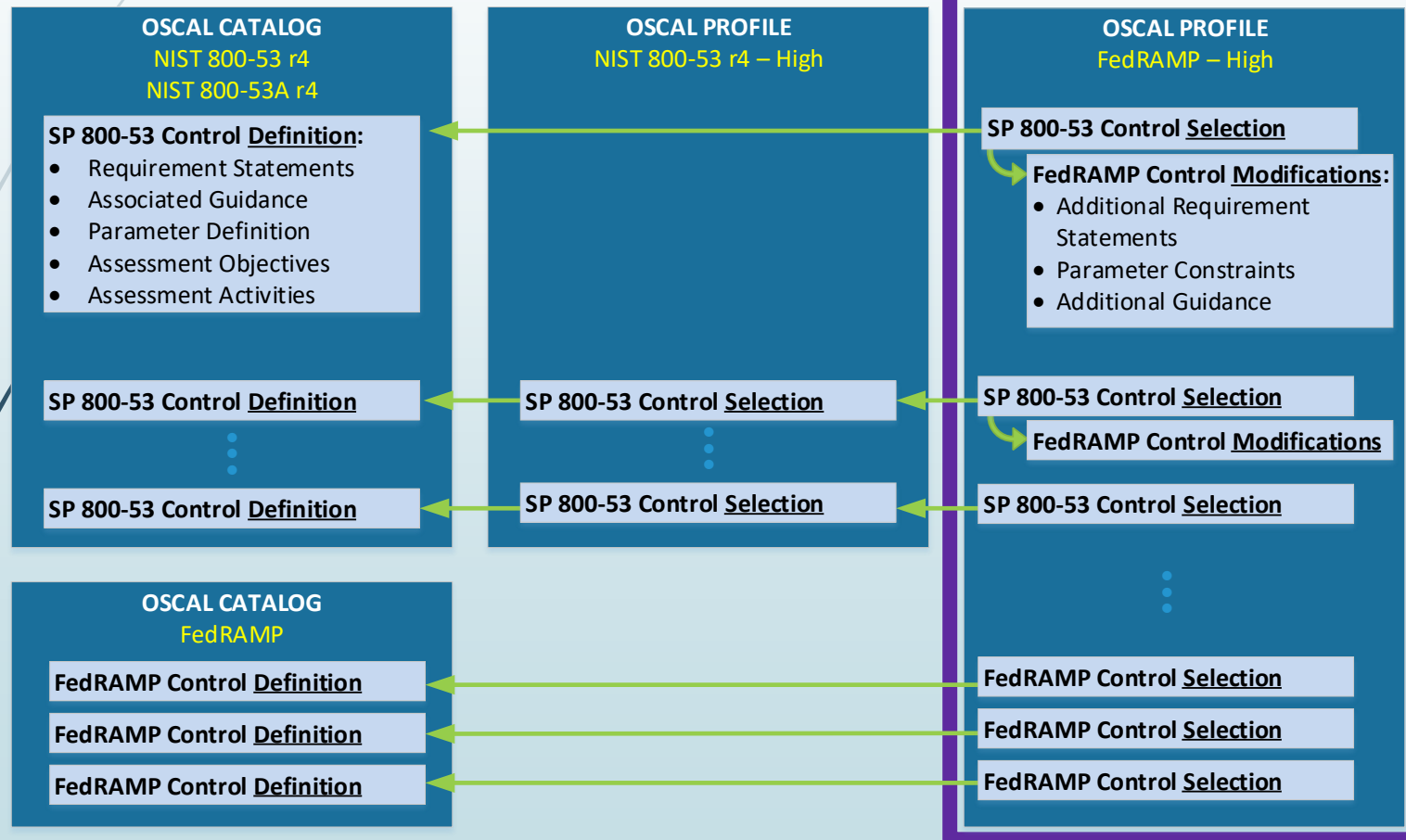
<!-- cut -->
<modify>
  <set param-id="ac-1_prm_3">
    <constraint>at least annually</constraint>
  </set>

  <alter control-id="au-2">
    <remove id-ref="au-2_smt.d" />

    <add position="ending">
      <part id="au-2_add" name="item" ns="my-org">
        <title>Additional Requirement</title>
        <part id="au-2_smt.a.add" name="item">
          <prop name="label">Additional a.</prop>
          <p>My replacement for statement.</p>
        </part>
      </part>
    </add>
  </alter>
</modify>
<!-- cut -->

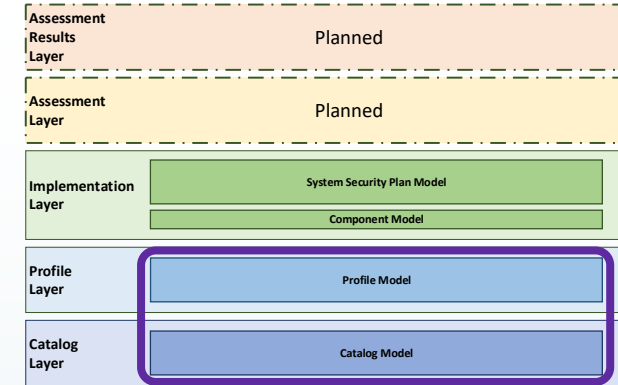
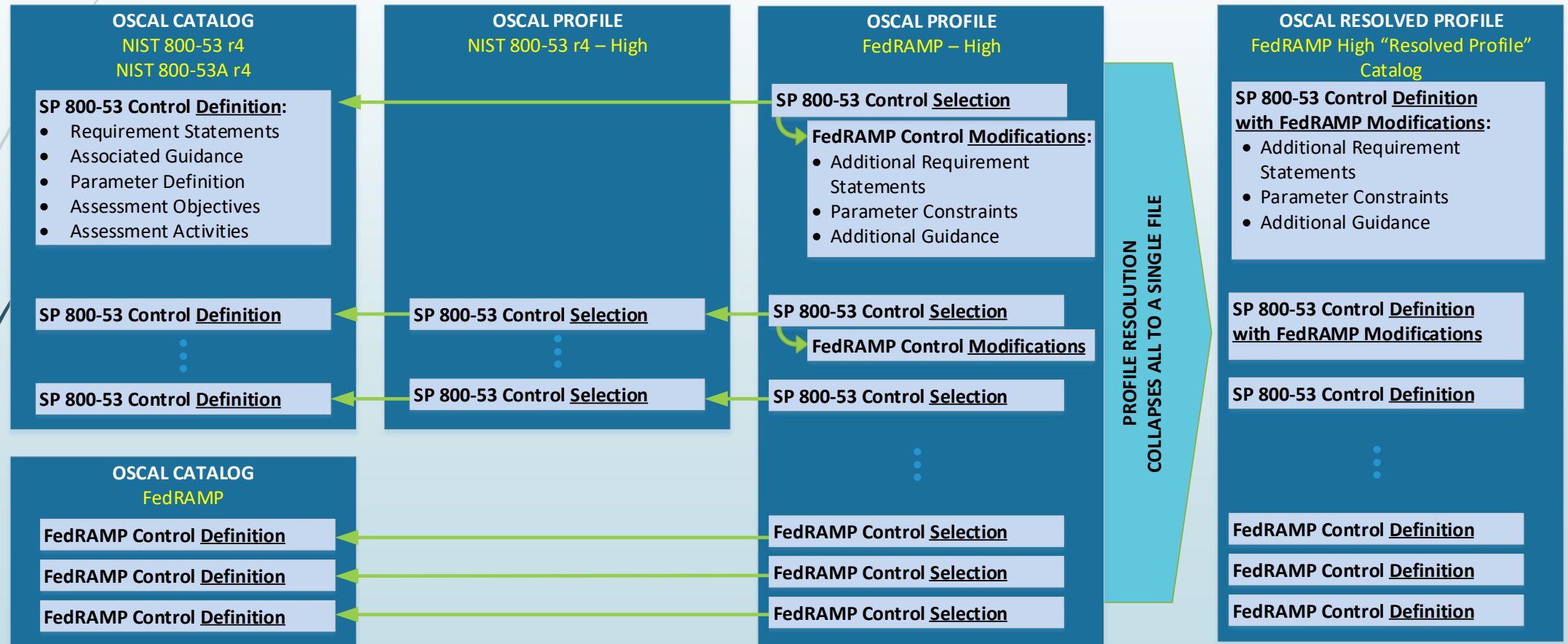
```

# Working With Catalogs and Profiles



# Profile Resolution

Merges all of the content to a single catalog file, as defined by the downstream-profile.



# How You Can Help

## We need your help!

- ▶ **Provide Feedback on the Models**
- ▶ Produce Catalogs:
  - ▶ ISO-27002, COBIT 5
  - ▶ Organizational Catalogs
- ▶ Produce Profiles
  - ▶ Organizational baselines/overlays
- ▶ Produce Tools That Import, Export, and/or Manipulate OSCAL Catalogs and Profiles

## Ways to Submit Feedback

- ▶ Add a comment to an appropriate issue: <https://github.com/usnistgov/OSCAL/issues>
- ▶ Create a new issue
  - if no appropriate issue exists
- ▶ Send us email: [oscal@nist.gov](mailto:oscal@nist.gov)
- ▶ Contribution Guidance: <https://pages.nist.gov/OSCAL/contribute/>



# Resources



## Documentation

Catalog: <https://pages.nist.gov/OSCAL/docs/model/catalog/>

Profile : <https://pages.nist.gov/OSCAL/docs/model/profile/>



## Catalog Examples

NIST SP 800-53: <https://git.io/JegXq>

FedRAMP: <https://git.io/JegXm>



## Profile Examples (Baselines)

NIST SP 800-53 Moderate : <https://git.io/JegXs>

FedRAMP Moderate: <https://git.io/JegXZ>



## Tools

OSCAL Kit: <https://github.com/docker/oscalkit>

OSCAL GUI: <https://github.com/brianrfgsa/OSCAL-GUI>

The **OSCAL project** is developed openly on GitHub.com.

**Repository:** <https://github.com/usnistgov/OSCAL>

**Project Website:** <https://www.nist.gov/oscal>

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

**We welcome contributions to this project.**

# Questions?

# Break

On your own

**Note: An escort is needed beyond building 215**

# Agenda

Time	Topic
8:00 am	Registration and Networking
9:00 am	Welcome, Introduction to Workshop
9:10 am	<b>What is OSCAL and Who Needs It</b> Michaela Iorga, OSCAL co-lead NIST
10:00 am	<b>OSCAL and FedRAMP</b> Ashley Mahan, Director, FedRAMP, GSA
10:30 am	Break
10:45 am	<b>Tools That Understand OSCAL</b> Andrew Weiss, GitHub and Wendell Piez, OSCAL team member, NIST
11:20 am	<b>OSCAL Roadmap</b> David Waltermire, SCAP Lead and OSCAL co-lead, NIST
Noon	Lunch
1:30 pm	<b>OSCAL Catalog and Profile Models</b> Brian Ruf, OSCAL team member, Noblis
2:45 pm	Break
3:00 pm	<b>OSCAL Component and System Security Plan Models</b> David Waltermire, SCAP lead and OSCAL co-lead, NIST
4:15 pm	<b>Discussion: Tomorrow is Today – The Need for Automation</b> Moderator: David Waltermire, SCAP lead and OSCAL co-lead, NIST
4:50 pm	Closing Remarks
5:00 pm	Adjourn

# OSCAL Component and System Security Plan Models

David Waltermire

SCAP Lead and OSCAL Co-Lead

National Institute of Standards and Technology (NIST)

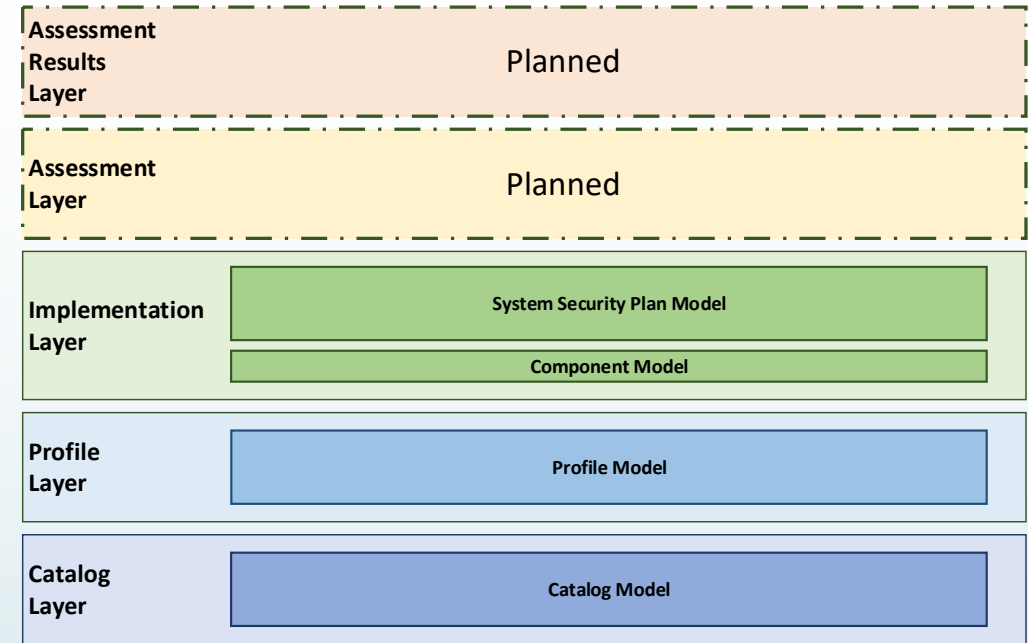
# Presentation Agenda

## Component Model

- Goals
- Scope
- Authors & Consumers

## System Security Plan (SSP) Model

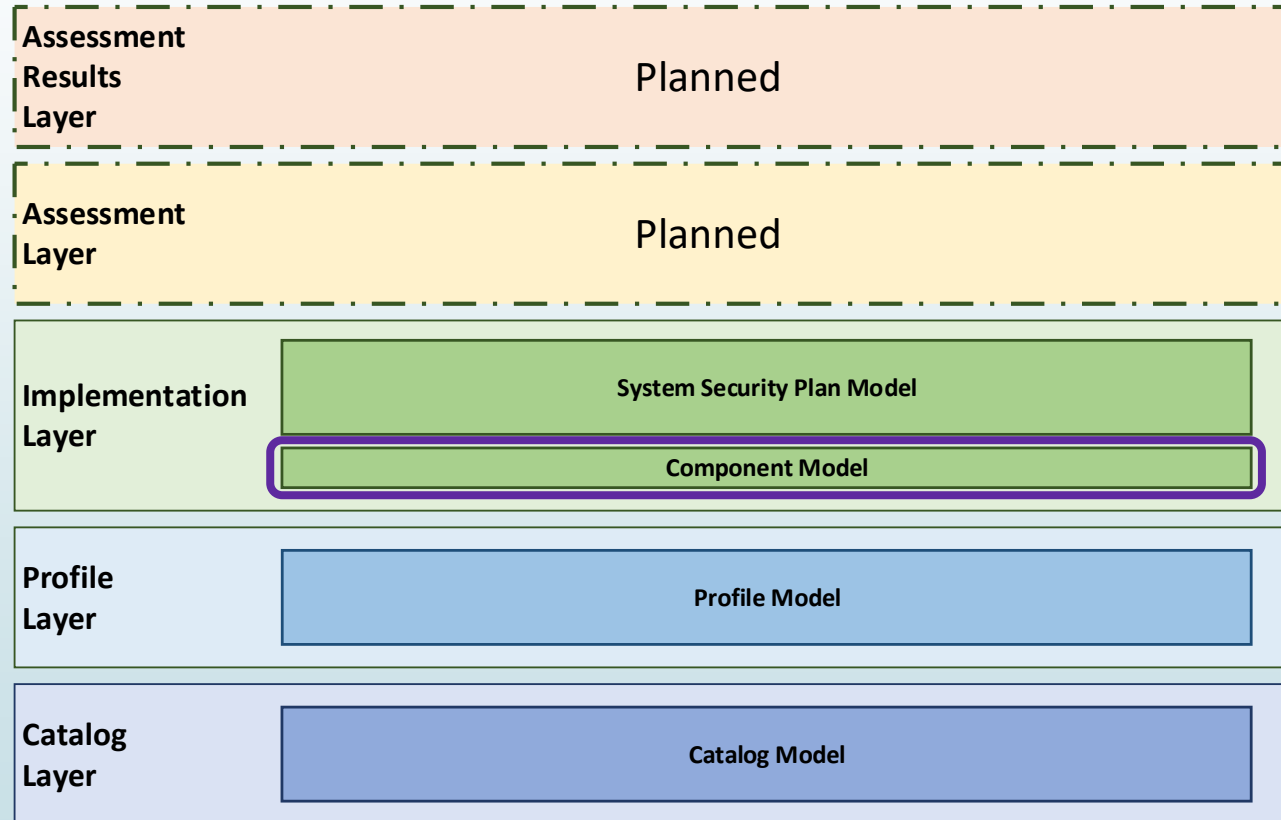
- Goals
- Scope
- Authors & Consumers



## Working with Components and SSPs

- Case Studies / Examples
- Resources
- How You Can Help
- Questions?

# Component Model



# Component Model

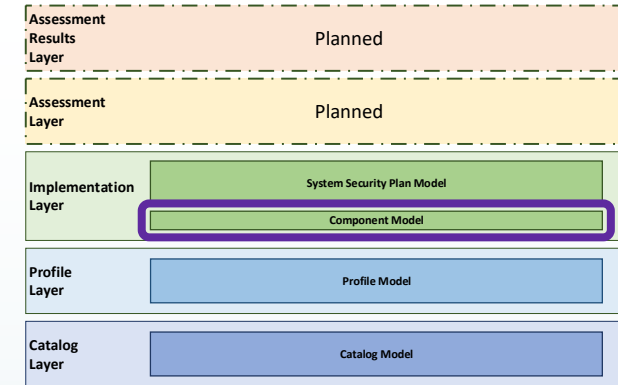
## Purpose and Goals

### Purpose

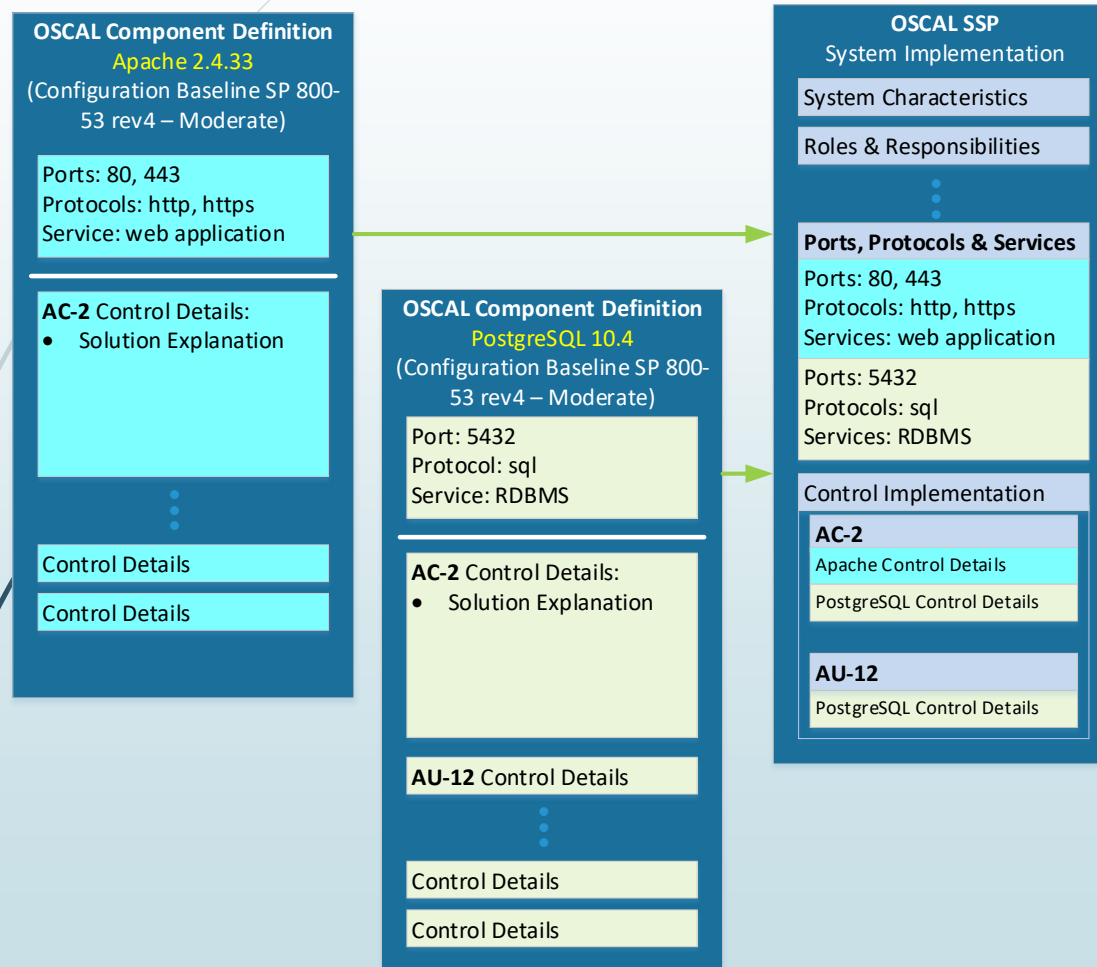
- ▶ Allow maintainers of assets to describe and share how the asset can be implemented to satisfy specific controls.
  - ▶ Asset types include hardware, software, services, policies, procedures, plans, or anything else that enables a system to satisfy a control.
  - ▶ For some asset types, this may include defining multiple options, which satisfy controls differently.

### Goals

- ▶ Provide machine-readable detail about assets.
- ▶ Identify the controls satisfied by the asset, and provide a description that details how each control is satisfied for a given implementation.
- ▶ Enable system implementers to import asset information into a system implementation document, saving effort and reducing error.



# A quick discussion on components



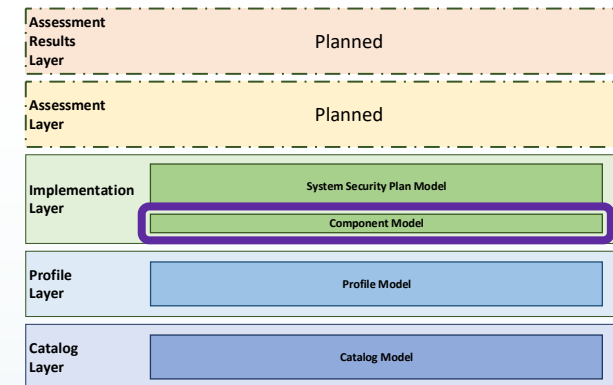
- A *component definition* provides for the sharing of knowledge about a component asset
- Allows an asset provider to share information about
  - how the asset *can be* implemented
  - the controls that are supported by the asset
- Implementors of a system can use this component information to “seed” their system implementation details



# Component Model

## Features

- ▶ Ability to capture details about an asset in machine-readable format such as its name, version, provider, and implementation characteristics
- ▶ Controls implemented by the asset
- ▶ The specific baselines (profiles) that may lead to particular control implementation
- ▶ Required configuration of asset in order to implement those controls relative to a particular baseline (profile)
- ▶ Examples:
  - ▶ Access Control Policy
  - ▶ Incident Response Plan
  - ▶ Account Authorization Process
  - ▶ Network Firewall
  - ▶ Database
  - ▶ Application Server



# Component Model

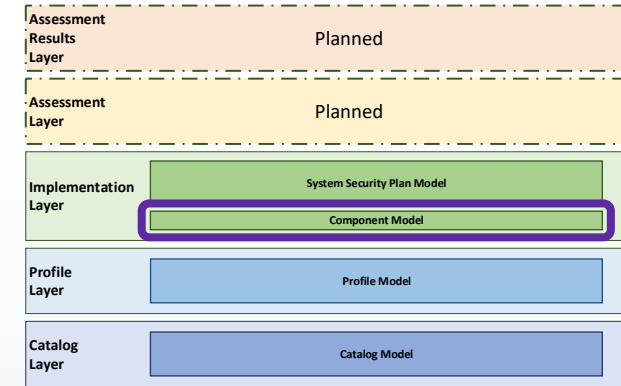
## Authors and Consumers

### Authors

- ▶ Asset maintainers that wish to publish how the given asset implements a set of controls
  - ▶ For **hardware, software, and services** this may include open source, commercial, or custom implementations
  - ▶ For **policies, procedures, plans,** and other documentary artifacts this author may be the artifact maintainer

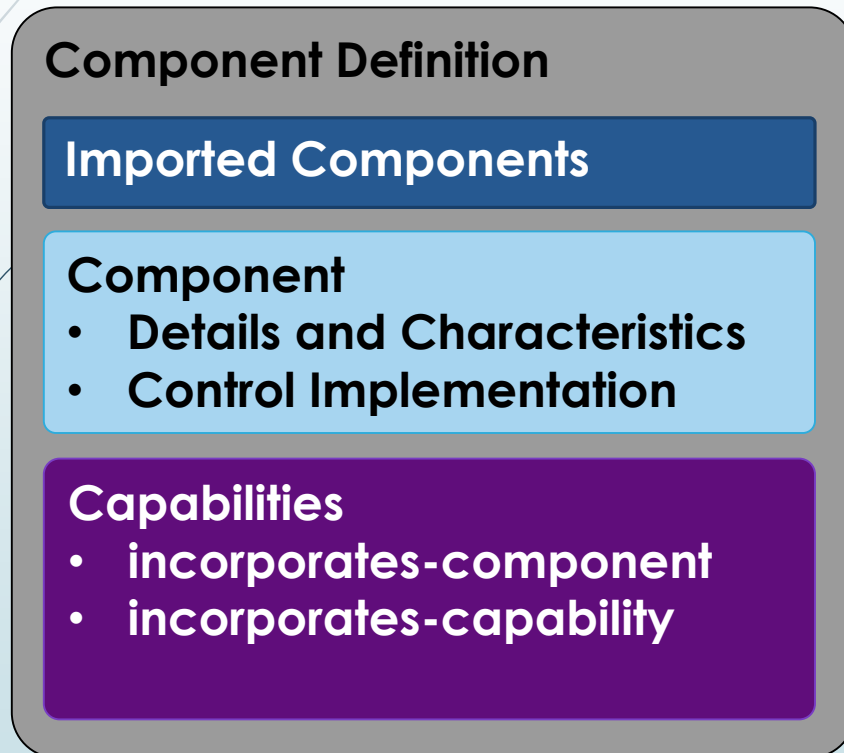
### Consumers

- ▶ Organizations acquiring and implementing the asset
- ▶ Anyone who must document controls
- ▶ Anyone who must audit or assess control implementation compliance
- ▶ Anyone who must adjudicate or render an authorization decision for a system based on control compliance



# Component Definition

## Organization



### A component definition can:

- Import components from other component definitions
- Define one or more components
- Capabilities group components
  - can incorporate other capabilities
  - define how multiple components can be used together as part of a solution

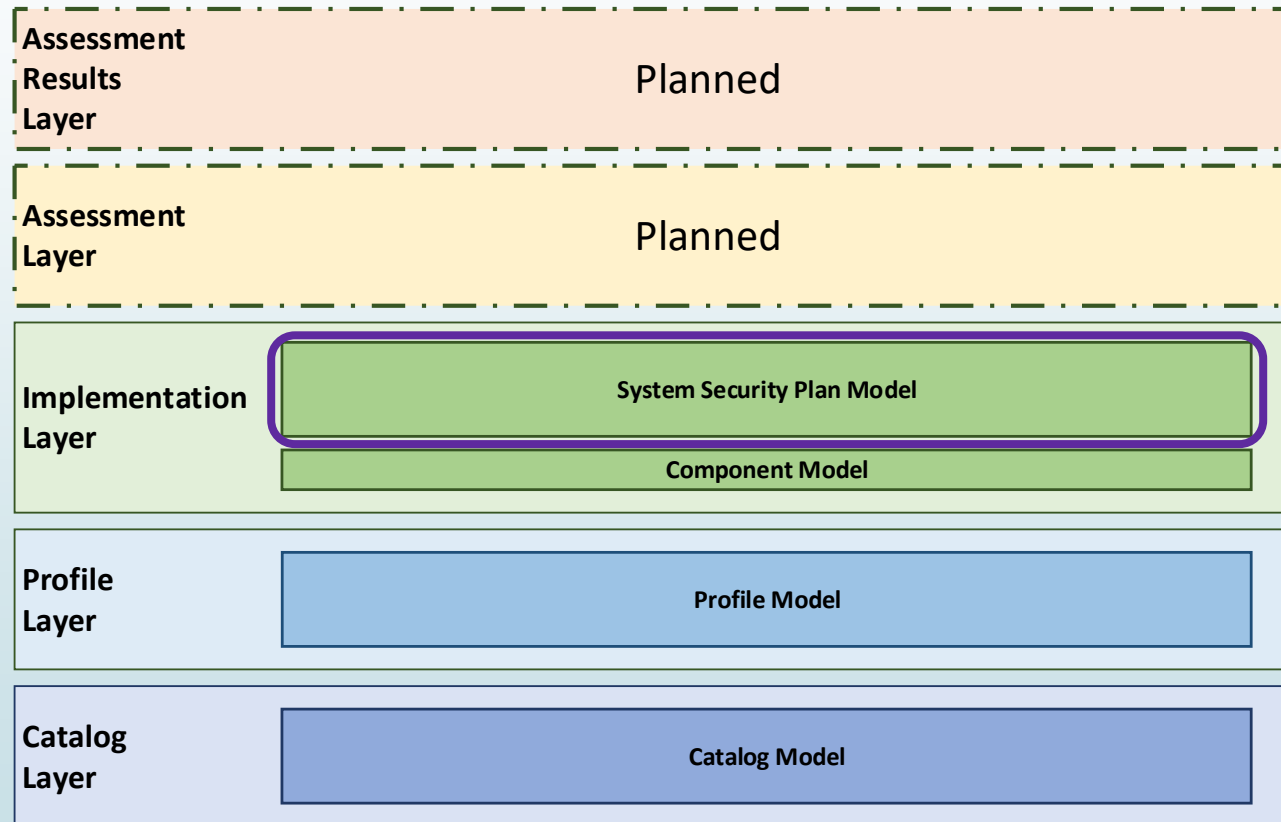
# Component Definition

## Component

### A component can define:

- **name:** The name of the component
- **type:** The component type: hardware, software, services, policies, procedures, plans, etc.
- **properties:** version(s), model, release date, identifiers, other characteristics
- **responsible-parties:** creator, maintainer, publisher, etc.
- **control-implementation:** How the asset can meet a set of control requirements
  - Can be defined at the control- or control statement-levels.
  - Can detail how specific baselines can be implemented
    - Use of parameters to define implementation options and constraints
- Component dependencies
  - component A *requires* component B
  - Component B *implements* component A
- Methods for implementing and assessing the component implementation

# System Security Plan (SSP) Model



# SSP Model

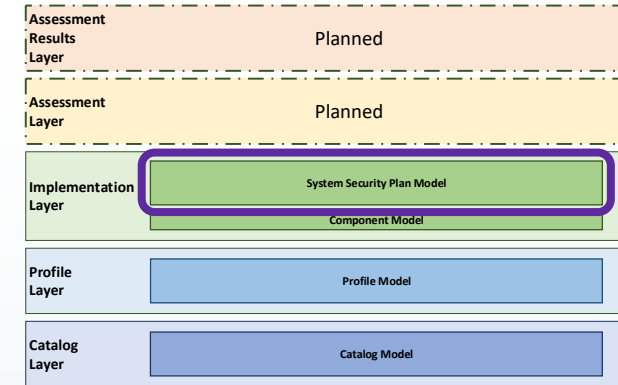
## Purpose and Goals

### Purpose

- Document the present and planned states of control implementation within an information system

### Goals

- Facilitate up-to-date documentation about the current state of the system
- Provide a level of granularity that allows each control implementation to be documented at the component level as well as at the system-wide level
- Support current SSP practices, and future automation based on improved SSP practices, with a migration path
- Enable automated analysis of current system implementation relative to the SSP content for a given system and connected systems
- Enable automated dashboards of cybersecurity status across a portfolio of systems

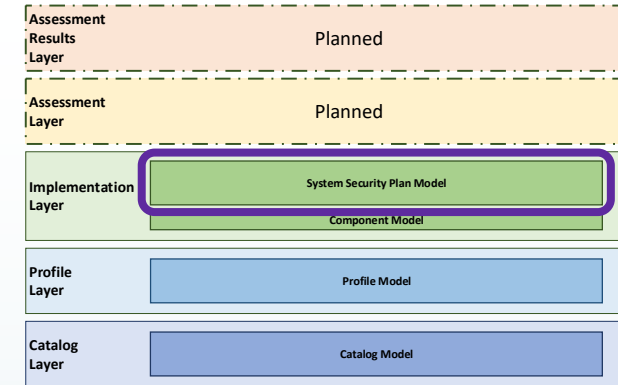


# SSP Model

## Features

### Provides appropriate granularity to support automation across the cybersecurity lifecycle:

- Rich description of the system's characteristics including: handled information types and criticality, system boundary, network architecture, and data-flow information
- Security categorization by overall impact and handled information
- Supports system planning as well as snapshot in time
- System's implementation can be detailed for users, components, services, interconnections, and system inventory
- Standardizes the semantics of control implementation information
  - by normalizing the representation of control implementation descriptions
  - regardless of their originating governance frameworks
  - Uses control identifiers down to the sub-statement level, allowing fine-grained implementation descriptions
  - Allows for parameter values to be defined on a per-control, per-component basis
- Defines the connectedness of SSPs, allowing data across multiple SSPs to be aggregated
  - Interconnected systems
  - Common control providers



# SSP Model

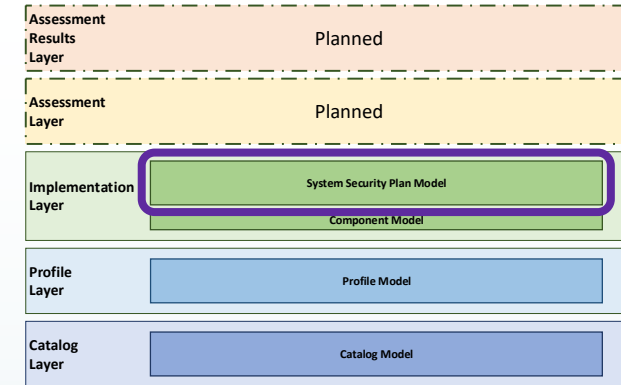
## Authors and Consumers

### Authors

- ▶ Control baseline/overlay authors, such as NIST or FedRAMP
- ▶ Organizations wishing to define organization-specific baselines/overlays such as commercial organizations or government agencies

### Consumers

- ▶ Authors or consumers of any higher layer in the OSCAL stack, such as:
  - ▶ assessors
  - ▶ authorizing officials
- ▶ Auditors who must assess control implementation alignment
- ▶ Adjudicators who must render an authorization recommendation or decision for a system
- ▶ Anyone who must document controls.
- ▶ Anyone who must audit or assess control implementation compliance.

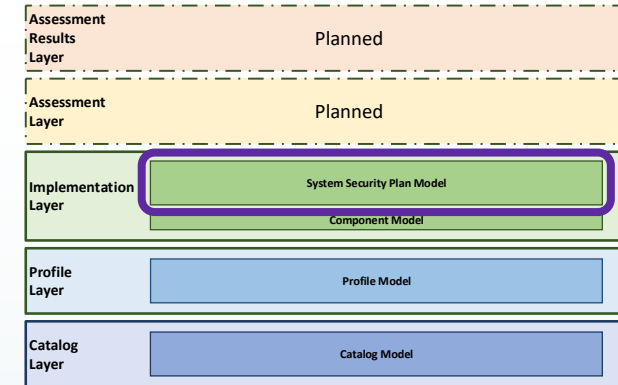




# Anatomy of an SSP

The major sections of an SSP are:

- metadata
- import-profile
- system-characteristics
- system-implementation
- control-implementation
- back-matter (optional)



```

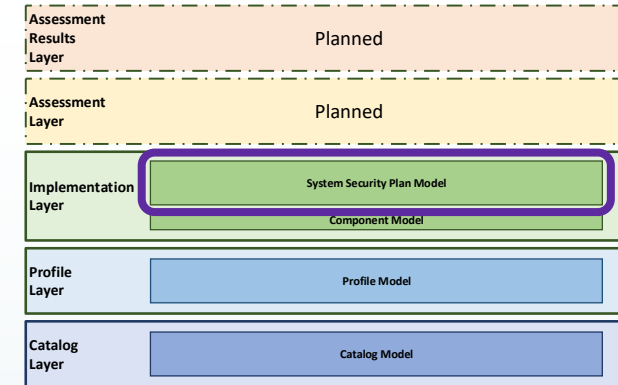
---
system-security-plan:
  id: "uuid-2e825608-980c-4c96-9435-54ce73bb09b6"
  metadata:
    # provides parties and roles used in the SSP
  import-profile:
    # identifies the system baseline, an OSCAL Profile
  system-characteristics:
    # System description, information types, impact levels,
    # system boundary, network architecture
  system-implementation:
    # Users, components, system inventory
  control-implementation:
    # Control implementation details
  back-matter:
    # provides resources that are referenced (e.g., diagrams,
    # policies, procedures, etc.)
  
```

# Anatomy of an SSP

## Metadata

The metadata section in an SSP is used to define:

- ▶ **role** entries: All specialized roles used in the system
- ▶ **party** entries: All organizations and individuals related to the system are defined as parties



```

---
metadata:
  title: "Enterprise Logging and Auditing System Security Plan"
  last-modified: '2019-09-23T18:14:50.591Z'
  version: '1.0'
  oscal-version: '1.0.0-milestone2'
  roles:
    id: 'legal-officer'
    title: "Legal Officer"
  parties:
    - id: 'enterprise-asset-owners'
    - id: 'enterprise-asset-administrators'
    - id: 'legal-department'
    - id: 'it-department'
    - id: 'logging-server-vendor'
  org:
    org-name: "Acme Corp"
  
```

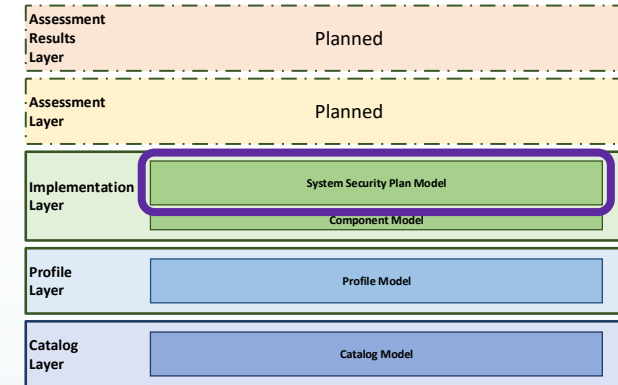
# Anatomy of an SSP

## System Control Baseline

The `import-profile` section defined the system's control baseline

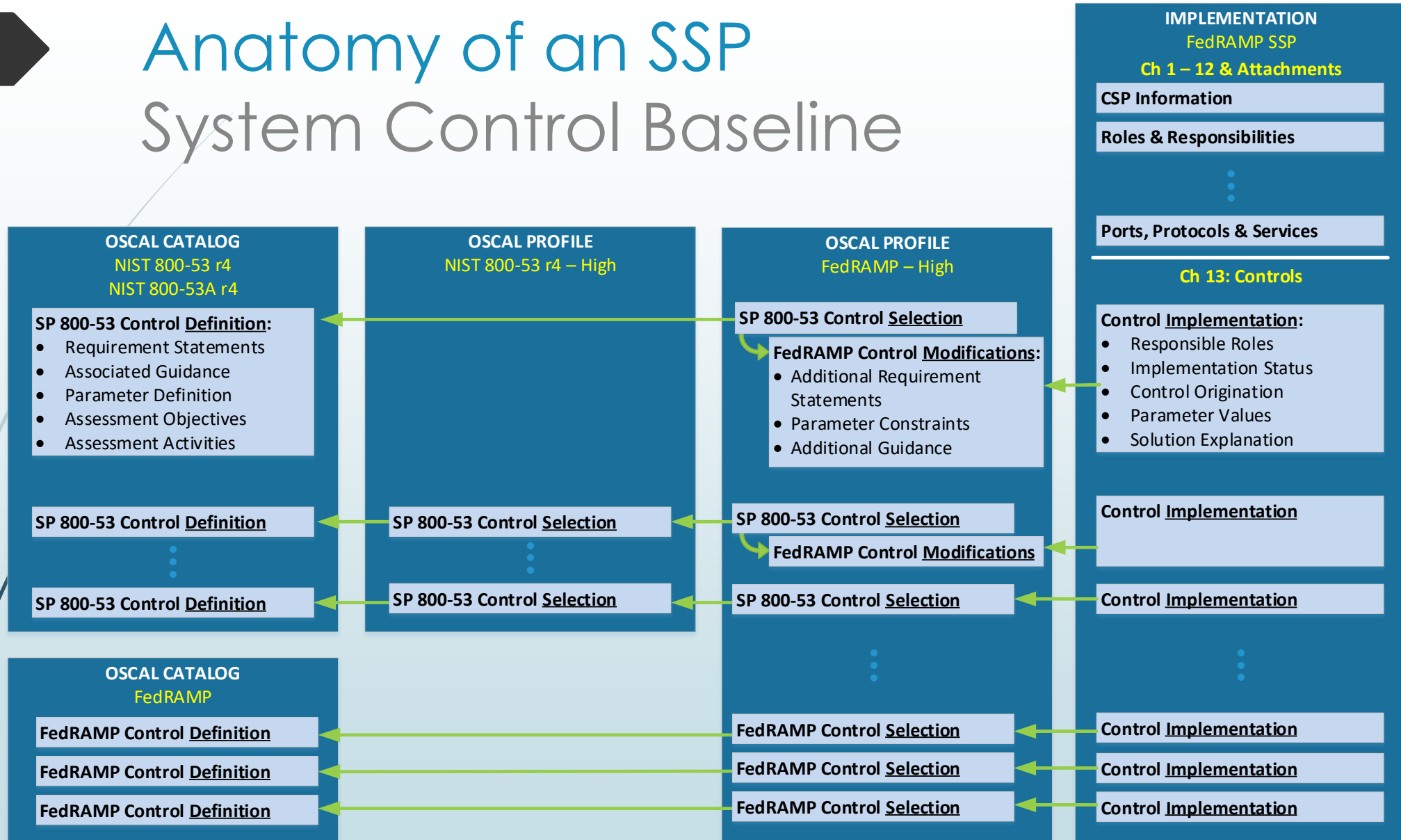
- References an OSCAL profile used as the system baseline

```
---  
# ... snip ...  
import-profile:  
  href: "NIST_SP-800-53_rev4_MODERATE-baseline_profile.json"  
# ... snip ...
```



# Anatomy of an SSP

## System Control Baseline

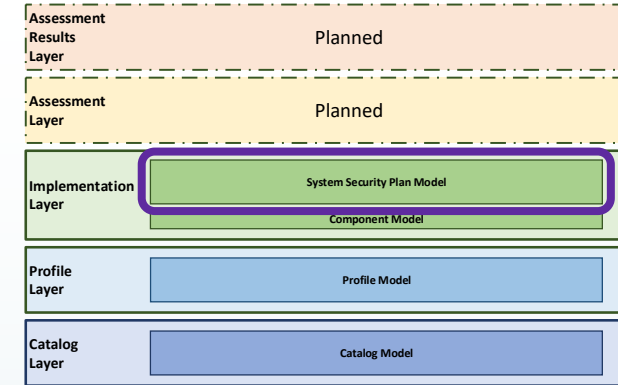


# Anatomy of an SSP

## System Characteristics

The system-characteristics section in an SSP is used to define:

- Descriptive information: the system's **name**, **description**, and **identifiers**, **status**
- Use of **props**, **annotations**, and **links** provide for extensible metadata
- Overall **system sensitivity level**
- Enumeration of the **types** and **criticality** of **information** handled by the system
- System **security objectives**
- System's **authorization boundary** definition
- Other descriptive information (e.g., data flows, network architecture)



```

---
# ... snip ...
system-characteristics:
  system-name: "Enterprise Logging and Auditing System"
  description: ''
  system-ids:
    id: 'd7456980-9277-4dcb-83cf-f8ff0442623b'
    identifier-type: 'https://ietf.org/rfc/rfc4122'
  status:
    state: 'other'
    remarks: "This is an example, and is not intended to be implemented
              as a system"
  annotations:
    - name: "deployment-model"
      value: 'private'
    - name: "service-models"
      value: 'iaas'
  security-sensitivity-level: "moderate"
  system-information:
    information-types:
      # ... snip ...
  security-impact-level:
    security-objective-confidentiality: 'fips-199-moderate'
    security-objective-integrity: 'fips-199-moderate'
    security-objective-availability: 'fips-199-low'
  authorization-boundary:
    description: "The description of the authorization boundary would go
                 here."
# ... snip ...

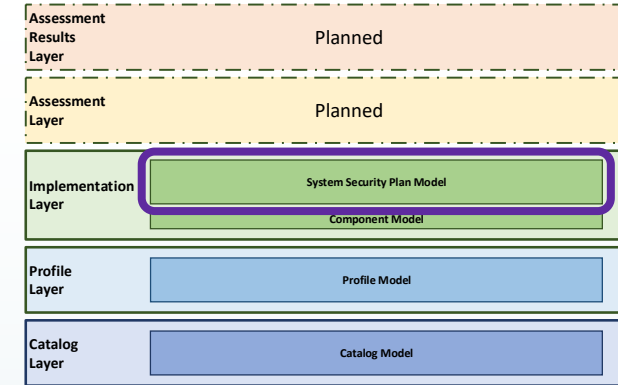
```

# Anatomy of an SSP

## System Characteristics: Information Types

The information-types section is used to define:

- The different types of information handled by the system
- For each information type:
  - Reference(s) to information type identification system(s)
  - a description of what the information is used for
  - associated impact levels



```

---
# ... snip ...
system-characteristics:
# ... snip ...
system-information:
  information-types:
    name: "System and Network Monitoring"
    information-type-ids:
      https://doi.org/10.6028/NIST.SP.800-60v2r1:
        id: 'C.3.5.8'
    description: "This system maintains historical logging and auditing information for all client devices connected to this system."
    confidentiality-impact:
      base: 'fips-199-moderate'
    integrity-impact:
      base: 'fips-199-moderate'
    availability-impact:
      base: 'fips-199-low'
# ... snip ...

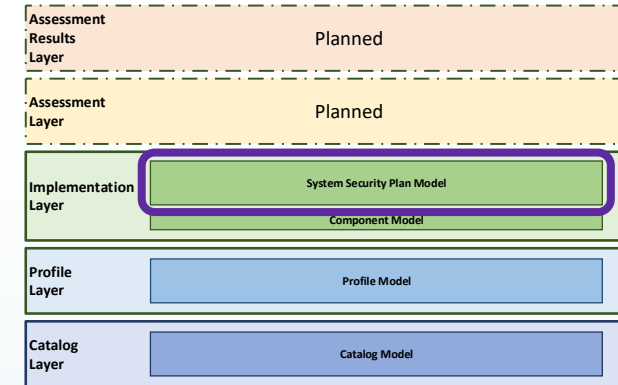
```

# Anatomy of an SSP

## System Implementation

The system-implementation section in an SSP is used to define:

- The system's user groups
- The components used in the system
  - Use of components is an optional feature
- A system inventory consisting of each unique inventory item



```

--- snip ---
system-implementation:
  remarks: "This is a partial implementation that addresses the logging server
  portion of the auditing system."
  # see snip ...
  users:
    # see snip ...
  components:
    component: logging-server:
      name: "Logging Server"
      description: "Provides a means for hosts to publish logged events to a
      central server."
      status:
        state: operational
      component-type: software
      responsible-roles:
        provider:
          party-ids: logging-server-vendor
        asset-owners:
          party-ids: enterprise-asset-owners
        asset-administrators:
          party-ids: enterprise-asset-administrators
    component: logging-policy:
      name: "Enterprise Logging, Monitoring, and Alerting Policy"
      component-type: policy
      description:
        - Requires all components to send logs to the enterprise logging
        solution
        - Requires all components synchronize their time with the appropriate
        enterprise time service, and at what frequency
        - Identifies the events that must be captured
        - Identifies who is responsible/accountable for performing these
        functions
      status:
        state: operational
      properties:
        version: [1,1]
        last-modified-date: 20181015
      responsible-roles:
        maintainer:
          party-ids: legal-department
    system-integration-process:
      name: "System Integration Process"
      component-type: process
      description: "Ensures proper integration into the enterprise as new
      systems are brought into production."
      status:
        state: operational
      properties:
        last-modified-date: 20181015
      responsible-roles:
        maintainer:
          party-ids: it-department
      links:
        href: "#components-policy"
        href: "#component-logging-policy"
      text: "Ensures logs from components in new system are able to
      published to the logging server. Ensures log monitoring capabilities
      recognize new system as authorized."
    inventory-management-process:
      name: "Inventory Management Process"
      component-type: process
      description: "Describes how new components are introduced into the
      system - ensures monitoring teams know about every asset that should
      be producing logs, thus should be monitored."
      status:
        state: operational
      properties:
        last-modified-date: 20181015
      responsible-roles:
        maintainer:
          party-ids: it-department
      links:
        href: "#components-policy"
        href: "#component-logging-policy"
      text: "Ensures that all hosts are known and authorized. Ensures that
      these hosts publish log events to the logging server."
    configuration-management-guidance:
      name: "Configuration Management"
      component-type: guidance
      description: "Describes how to configure a component to ensure its logs
      are transmitted to Splunk in the appropriate format. Also describes
      how to configure time synchronization."
      status:
        state: operational
      properties:
        last-modified-date: 20181015
      responsible-roles:
        maintainer:
          party-ids: it-department
      links:
        href: "#components-policy"
        href: "#component-logging-policy"
      text: "Ensures that all hosts are configured to publish log events to
      the logging server."
  system-inventory:
    # see snip ...
    id: "inventory-logging-server"
    description: "Logging server"
    asset-id: "asset-id-logging-server"
    responsible-roles:
      asset-administrators:
        party-ids: enterprise-asset-administrators
      asset-owners:
        party-ids: enterprise-asset-owners
    implemented-component:
      component: logging-server
      use: "runs-software"
    enforces-policy:
      policy: "enforces-policy"

```

# Anatomy of an SSP

## System Implementation: Users

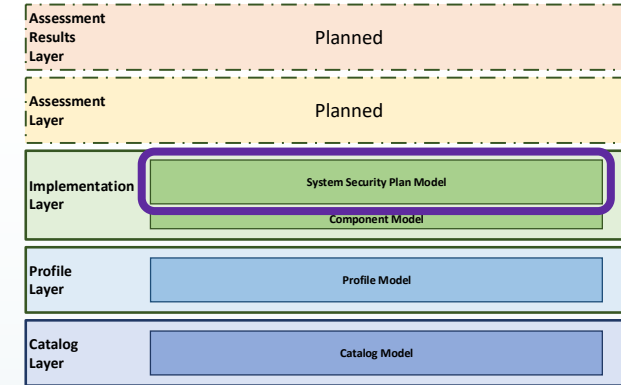
### The users section is used to define:

- Each user group that interacts with the system
- For each user, you can provide:
  - **title** (required)
  - **short-name**
  - **description** (required)
  - Map to **roles** (defined in the metadata (required))
  - **Privileges used** (optional)

```

---
# ... snip ...
system-implementation:
  users:
    # ... snip ...
    user-asset-administrator:
      title: "System Administrator"
      description: "The systems administrators responsible for the day-to-day
        operation of all assets."
      role-ids: 'asset-administrator'
      annotations:
        name: "type"
        value: 'internal'
    user-asset-owner:
      title: "Asset Owner"
      description: "The owners of specific assets in the system that are
        responsible for ensuring the security of these assets."
      role-ids: 'asset-owner'
      annotations:
        name: "type"
        value: 'internal'
    user-legal:
      title: "Legal Department"
      description: "The legal team is responsible for approving all policies
        and procedures used in the system."
      role-ids: 'legal-officer'
      annotations:
        name: "type"
        value: 'internal'
# ... snip ...

```



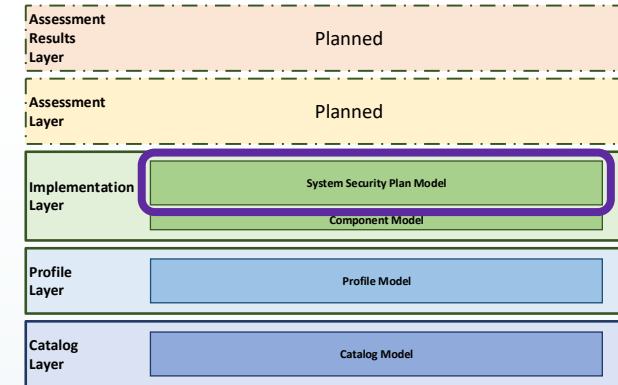


# Anatomy of an SSP

## System Implementation: Components

The components section is used to define:

- ▶ Each component used in the system
- ▶ For each component, you can provide:
  - ▶ **name, description, status**
  - ▶ Responsible roles
    - ▶ Who provides the component?
    - ▶ Who owns the implementation?
    - ▶ Who is responsible for administration of the asset?

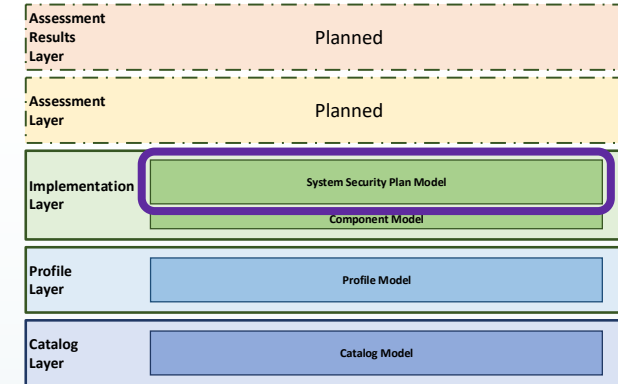


```

---
# ... snip ...
system-implementation:
# ... snip ...
components:
  component-logging-server:
    name: "Logging Server"
    description: "Provides a means for hosts to publish logged
    events to a central server."
    status:
      state: 'operational'
    component-type: 'software'
    responsible-roles:
      provider:
        party-ids: 'logging-server-vendor'
      asset-owner:
        party-ids: 'enterprise-asset-owners'
      asset-administrator:
        party-ids: 'enterprise-asset-administrators'
  
```

# Anatomy of an SSP

## System Implementation: Components



```

---
# ... snip ...
system-implementation:
# ... snip ...
components:
  component-logging-policy:
    name: "Enterprise Logging, Monitoring, and Alerting Policy"
    component-type: 'policy'
    description: |
      - Requires all components to send logs to the enterprise logging solution
      - Requires all components synchronize their time with the appropriate enterprise time service, and at what frequency.
      - Identifies the events that must be captured
      - Identifies who is responsible/accountable for performing these functions
    status:
      state: 'operational'
    properties:
      version: '2.1'
      last-modified-date: '20181015'
    responsible-roles:
      maintainer:
        party-ids: 'legal-department'
  
```

```

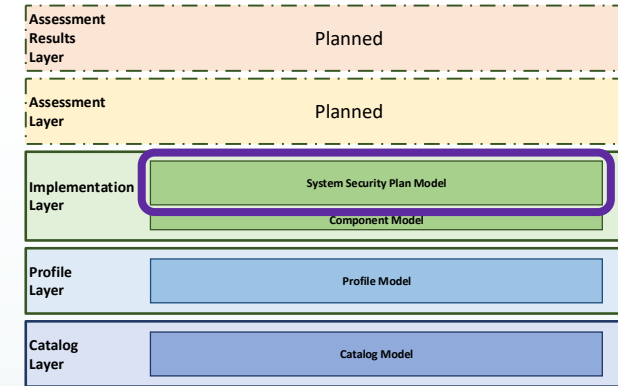
---
# ... snip ...
system-implementation:
# ... snip ...
components:
  system-integration-process:
    name: "System Integration Process"
    component-type: 'process'
    description: "Ensures proper integration into the enterprise as new systems are brought into production."
    status:
      state: 'operational'
    properties:
      last-modified-date: '20181015'
    responsible-roles:
      maintainer:
        party-ids: 'it-department'
    links:
      rel: 'implements-policy'
      href: "#component-logging-policy"
      text: "Ensures logs from components in new system are able to be published to the logging server. Ensures log monitoring capabilities recognize new system as authorized."
  
```

# Anatomy of an SSP

## System Implementation: System Inventory

The system-inventory section is used to define:

- Each inventory-item in the system
- For each inventory item you can provide:
  - **description, asset-id**
  - Map roles to parties
  - Define implemented components



```

---
# ... snip ...
system-implementation:
# ... snip ...
system-inventory:
  inventory-items:
    id: 'inventory-logging-server'
    description: "The logging server."
    asset-id: 'asset-id-logging-server'
    responsible-parties:
      asset-administrator:
        party-ids: 'enterprise-asset-administrators'
      asset-owner:
        party-ids: 'enterprise-asset-owners'
    implemented-components:
      component-logging-server:
        use: 'runs-software'
      component-logging-policy:
        use: 'enforces-policy'
# ... snip ...

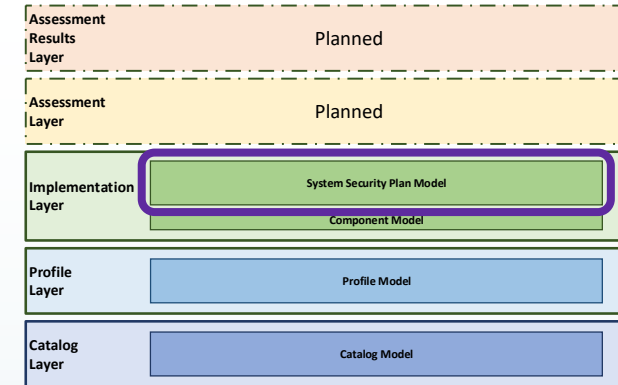
```

# Anatomy of an SSP

## Control Implementation

The control-implementation section is used to define:

- How a given control is implemented in the system?
- Can be documented at the statement-level
- Can be documented per-component
- Parameter values can be defined at the control-, statement-, and component-levels

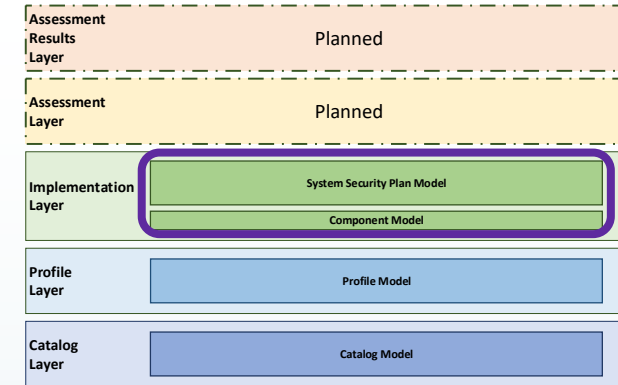


```

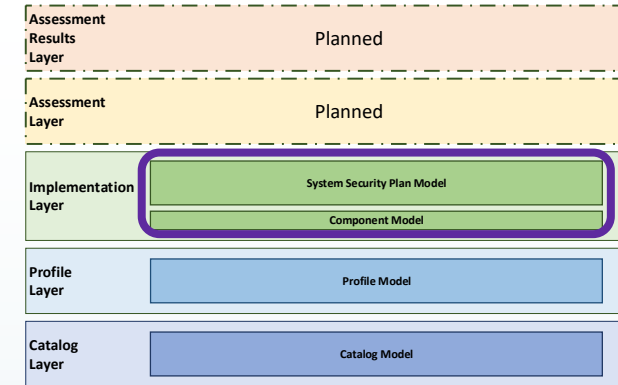
---
# ... snip ...
control-implementation:
  description: "Controls are implemented by a number of
  components used in this system."
  implemented-requirements:
    control-id: 'au-1'
  statements:
    # ... snip ...
    au-1_smt.b.1:
      by-components:
        component-logging-policy:
          set-params:
            au-1_prm_2:
              value: "annually, and other times as necessary in response to
              regulatory or organizational changes"
          description: "The legal department reviews this policy annually,
          and other times as necessary in response to regulatory or
          organizational changes. The legal department updates the policy
          as needed based on these reviews."
    au-1_smt.b.2:
      by-components:
        inventory-management-process:
          set-params:
            au-1_prm_3:
              value: "annually, and other times as necessary in response to
              regulatory or organizational changes"
          description: "The IT department reviews this process annually, and
          other times as necessary in response to regulatory or
          organizational changes. The IT department updates the policy as
          needed based on these reviews."
  
```

# Working With Components and SSPs

- ▶ Use of components in SSPs provide a path for assessing and managing distinct components
  - ▶ Implementation responsibilities can be clearly articulated
  - ▶ Automated collection can be used to gather configuration information for specific components
- ▶ Many SSPs today do not enumerate specific components used in the system
  - ▶ Use of components is an optional feature
  - ▶ OSCAL-based SSPs without components support current SSP practices
  - ▶ Definition of components in OSCAL-based SSPs provide for additional automation based on improved SSP practices, with a migration path
- ▶ Providers of assets who also provide an associated component definition help to support this transition!



# Working With Components and SSPs



- ▶ The “connectedness” of SSPs provides for analysis of the larger connected system.
  - ▶ System interconnections provide details of how the remote system is implemented.
  - ▶ Leveraged authorizations provide details of locally connected systems on which the security of the current system depends.
  - ▶ This information can be used to identify control implementation gaps that pose organizational risk.
- ▶ While some SSP information may be sensitive, some portions might be sharable.
  - ▶ Some portions of an SSP can be redacted to support more sharing between partners.
  - ▶ We want to better understand how information can be partitioned in the OSCAL SSP to support this type of use.

**Discussion topic:** Any recommendations on how to approach this?

# How You Can Help

## We need your help!

- ▶ **Provide Feedback on the Models**
- ▶ Produce Component Files:
  - ▶ Hardware and Software Producers, and Service Providers
  - ▶ Policy/Process/Plan Authors
- ▶ Produce System Security Plans
  - ▶ System Owners
  - ▶ Service Providers
- ▶ Produce Tools That Import, Export, and/or Manipulate OSCAL SSP Content

## Ways to Submit Feedback

- ▶ Add a comment to an appropriate issue: <https://github.com/usnistgov/OSCAL/issues>
- ▶ Create a new issue  
– if no appropriate issue exists
- ▶ Send us email: [oscal@nist.gov](mailto:oscal@nist.gov)
- ▶ Contribution Guidance: <https://pages.nist.gov/OSCAL/contribute/>

# Resources



## Documentation

Component: <https://pages.nist.gov/OSCAL/docs/model/component/>  
SSP: <https://pages.nist.gov/OSCAL/docs/model/ssp/>



## SSP Example

Component-based SSP: <https://git.io/Jegyk>



## Tools

OSCAL Kit: <https://github.com/docker/oscalkit>  
OSCAL GUI: <https://github.com/brianrufgsa/OSCAL-GUI>

The **OSCAL project** is developed openly on GitHub.com

**Repository:** <https://github.com/usnistgov/OSCAL>

**Project Website:** <https://www.nist.gov/oscal>

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

**We welcome contributions to this project.**

# Questions?



# Discussion: Tomorrow is Today The Need for Automation

David Waltermire

SCAP Lead and OSCAL Co-Lead

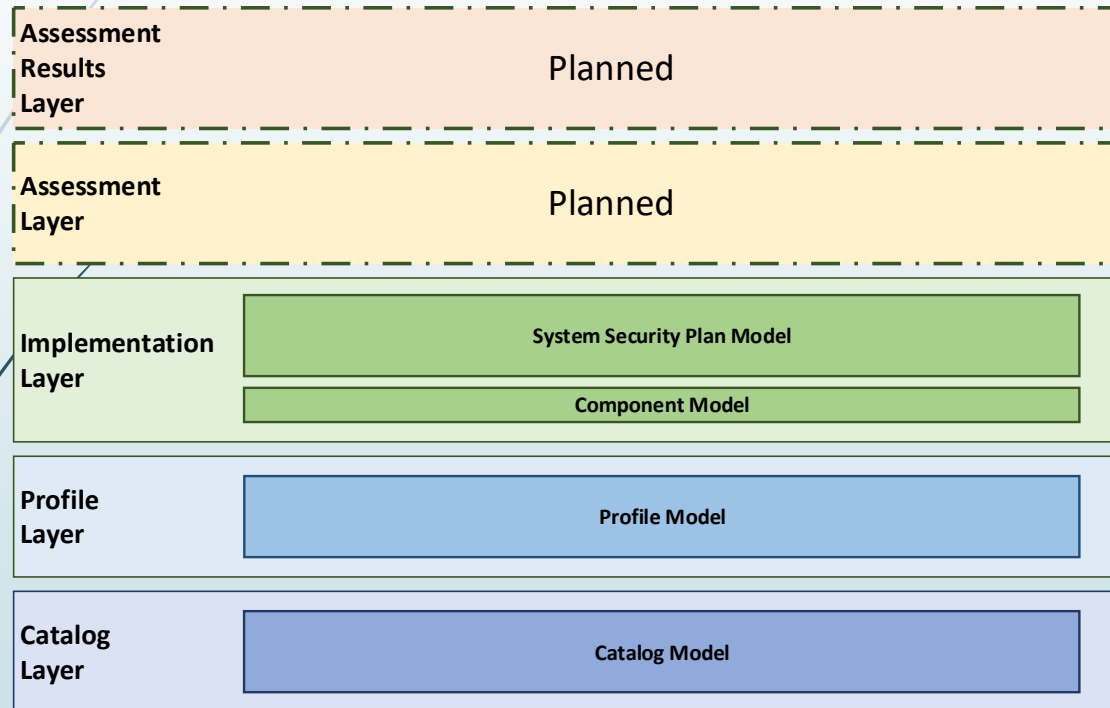
National Institute of Standards and Technology (NIST)

# What we have discussed so far...

- ▶ Overview of OSCAL
- ▶ OSCAL and FedRAMP: A use case for OSCAL
- ▶ Issues to consider related to OSCAL tooling
- ▶ Deep dive into the Catalog, Profile, Component, and SSP models

# Discussion

## OSCAL Architecture



The OSCAL architecture segments the information needed to document control-based **system implementation** and to support control-based **assessment**.

- Does this architectural approach align with how you produce and use this information?
- Do you have any ideas on how we can organize the assessment/assessment results layers?

# Discussion

## OSCAL Adoption

### **OSCAL can be used to:**

- Import and export control, baseline, and control implementation information between tools
  - Support advanced analytics
    - find implementation gaps
    - Provide context to other domains (e.g., asset / information criticality used for patch / vulnerability management)
  - But, current system implementation information is not in OSCAL format
- Do you find the OSCAL models useful as data exchange formats? Why so or not?
  - What use cases beyond control assessment do you see OSCAL data supporting?
  - How can we facilitate a transition from Word, PDF, and spreadsheets (current practice) into OSCAL-based formats?

# Discussion

## Other Topics?

Do you have any other points for discussion?

Any other Questions?

### **OSCAL Repository:**

<https://github.com/usnistgov/OSCAL>

### **Project Website:**

<https://www.nist.gov/oscal>

### **How to Contribute:**

<https://pages.nist.gov/OSCAL/contribute/>

# Closing Remarks

# Thank you

We would like to host another workshop in early April 2020.

Are there any conflicts we should avoid?

**OSCAL Repository:**

<https://github.com/usnistgov/OSCAL>

**Project Website:**

<https://www.nist.gov/oscal>

**How to Contribute:**

<https://pages.nist.gov/OSCAL/contribute/>