Open Security Controls Assessment Language (OSCAL)

Lunch with the OSCAL Developers

David Waltermire

National Institute of Standards and Technology

Teleconference Overview

- Ground Rules
- OSCAL Status Summary (5 minutes)
- Issues Needing Help from the Community
- Question and Answer / Discussion
  - Submitted questions will be discussed
  - The floor will be open for new questions and live discussion

OSCAL Lunch with the Developers

Purpose:

- Facilitate an open, ongoing dialog with the OSCAL developer and user communities to promote increased use of the OSCAL models

Goals:

- Provide up-to-date status of the OSCAL project development activities
- Answer questions about implementing and using the OSCAL models, and around development of OSCAL model-based content
- Review development priorities and adjust priorities based on community input
- Help the OSCAL community identify development needs

Ground Rules

- Keep the discussion respectful
    - Using welcoming and inclusive language
    - Being respectful of differing viewpoints and experiences
    - Gracefully accepting constructive criticism
    - Focusing on what is best for the community
    - Wait for one speaker to finish before speaking - one speaker at a time
- Speak from your own experience instead of generalizing ("I" instead of "they," "we," and "you").
- Do not be afraid to respectfully challenge one another by asking questions -- focus on ideas.
- The goal is not to always to agree -- it is to gain a deeper understanding.
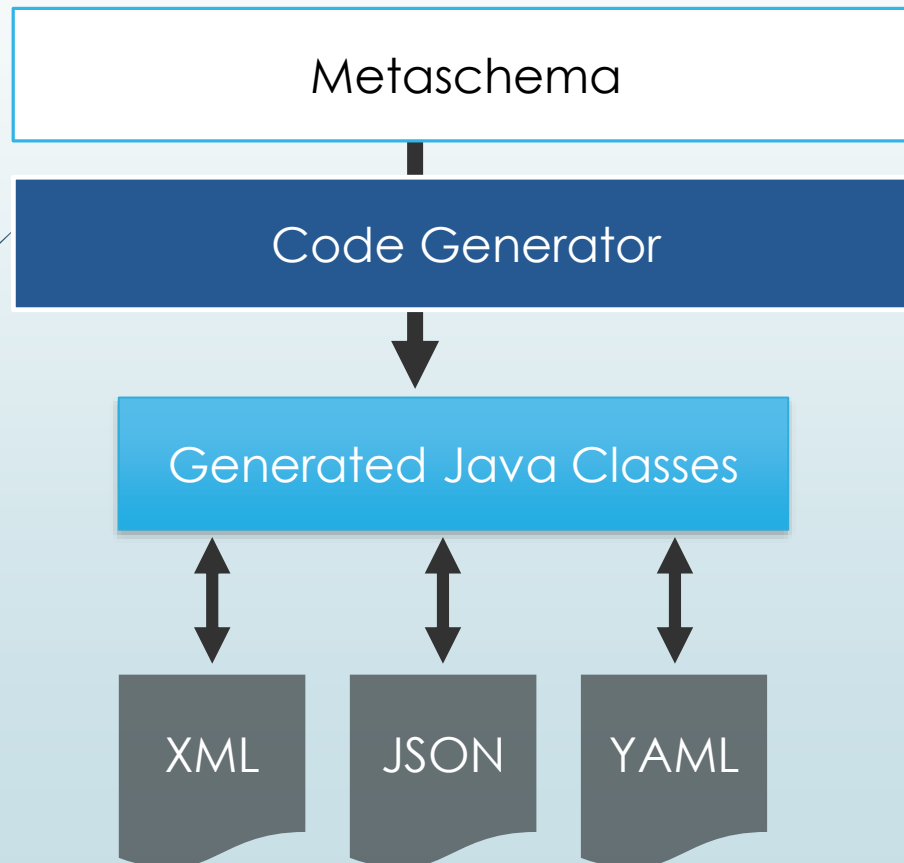
OSCAL Version 1 Milestones

Milestone	Focus	Sprints	Status	Date
Milestone 1	Catalog and Profile Models	**1** to **21**	**Completed**	**6/15/2019**
Milestone 2	System Security Plan (SSP) Model	**6** to **23**	**Completed**	**10/1/2019**
Milestone 3	Component Definition Model	**6** to **~28**	**In Progress**	**~May 2020**
Full Release	Development of a web-based specification	**24** to **~33**	**In Progress**	August 2020
Ongoing Maintenance	Minor and bugfix releases as needed	Additional Sprints	Planned	Ongoing

Current Sprint: 28 (https://github.com/usnistgov/OSCAL/projects/27)

Review of Current/Completed Work

On Github: https://github.com/usnistgov/OSCAL

Other Development Efforts:
Java Code Generation

Metaschema

Code Generator

Generated Java Classes

XML   JSON   YAML

- A tool that generates Java classes and serializers/deserializers based on a Metaschema definitions

- Generated code can read/write valid XML, JSON, and YAML content based on Metaschema generated XML and JSON schema

- Reading and writing XML, JSON and YAML now working

- Working on a Maven plugin to auto generate code

- Will be used to create an OSCAL Java library

https://github.com/david-waltermire-nist/metaschema/tree/java-metaschema/toolchains/java

Other Development Efforts

Developing XSL templates that generate:

- SP 800-53 style control listing in HTML and PDF
- An SSP based on a SP 800-18 template in HTML and PDF

Developing OSCAL Command Line Tools

- Validation and conversion of OSCAL content
- Comparison of OSCAL content (e.g., SP 800-53 rev4 vs rev5)

Help Needed

Please review pull requests and comment on issues you are interested in.

Discussion

Goals:

- Establish a robust, engaged OSCAL development community. Enable the community to assist other community members.

- Promote adoption of the OSCAL data formats in GRC and assessment tooling.

What barriers can we break down to promote these goals?

Are there other things we should be focusing on?

Open Floor

What would you like to discuss?

What questions do you have?

Thank you

Next Lunch with Devs:

April 9, 2020

12:00 Noon EST (5:00 PM UTC)

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

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

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

Contact Us: oscal@nist.gov

Backup Slides

OSCAL Layers & Models

Layer	Model
Assessment Results Layer	Planned
Assessment Layer	Planned
Implementation Layer	System Security Plan Model / Component Model
Profile Layer	Profile Model
Catalog Layer	Catalog Model

Building up from foundation
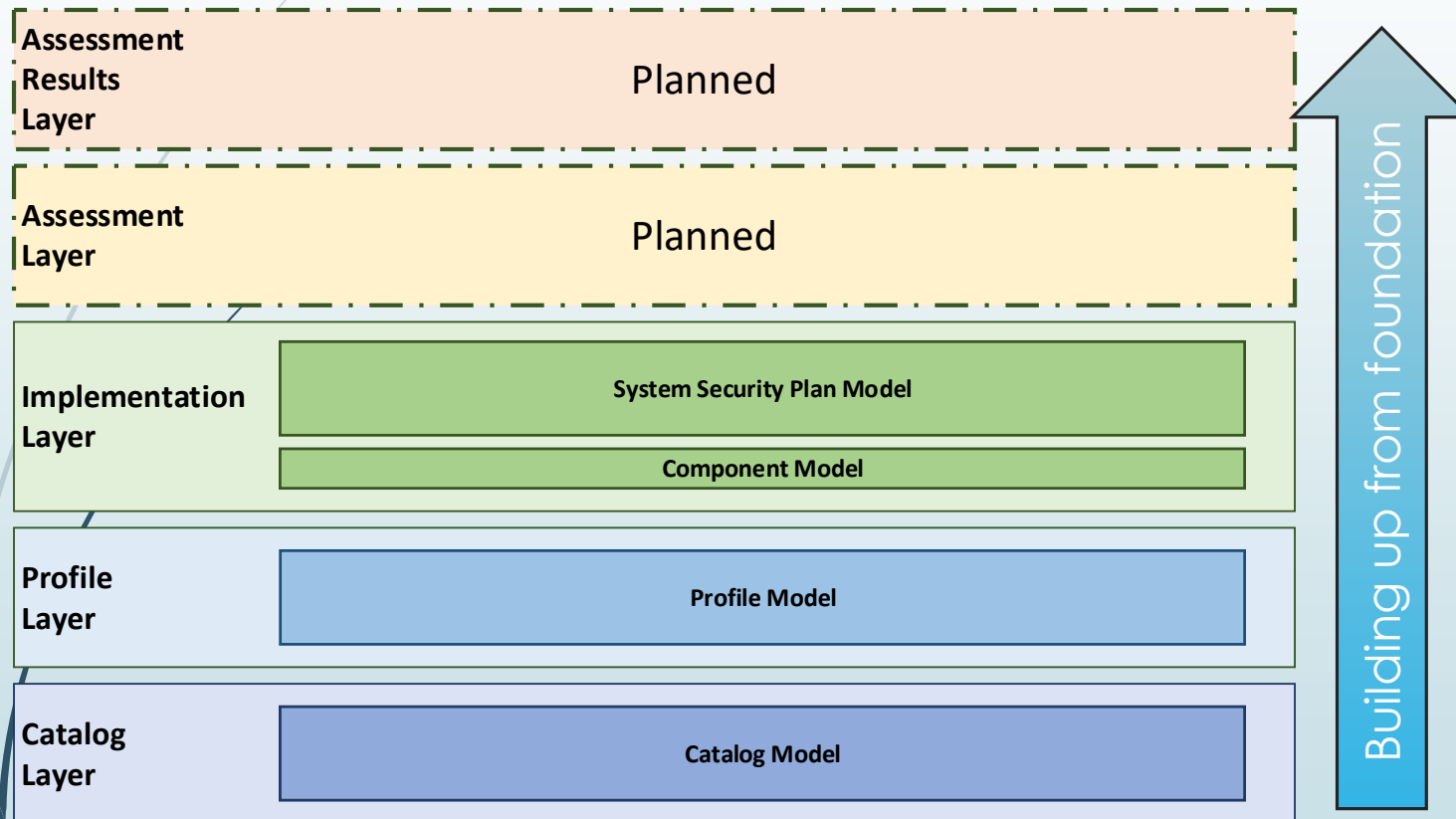
OSCAL is architected in layers

- 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