

Mike Sanderson on NIST CSWP 53 ipd (OSCAL Roadmap)
m@mikesand.com
January 13, 2026

Comments Abstract / Executive Summary:

The comments' central point is to make a distinction between conceptual data models and logical data models. OSCAL models are both, but this distinction is not clear in the draft. I think the OSCAL conceptual model — Catalogues and Profiles, and Profile Resolution — is excellent and I think it will scale widely. These comments add the distinction at several key places, as well as bringing the concept of the three "layers" (Control, Implementation, Assessment) to the fore, as they are extremely useful to understand OSCAL conceptually. By understanding OSCAL models as conceptual models, users can better understand which models or which layers would apply to their role in the cybersecurity ecosystem and where they should begin to look more closely.

Note regarding the OSCAL logical data models:

Regarding the logical data models, the driving of the adoption of logical data models as a format is a highly technical process requiring widespread detailed technical implementation, and, while it is central to my software development experience from beginning to end, that process is outside the scope of these comments. Having seen how companies that want to push adoption of formats proceed, widespread adoption of the logical data models will require an enormous push with specification, examples, sample tooling, direct developer relations and other steps that should not be underestimated. Again, however, the conceptual model of Catalogues, Profiles, and Components is in fact quite innovative and I think it will scale widely.

About me:

I am an application software developer and since 2013 have worked on iPhone apps, working on projects with high security requirements, including as a vendor for a launch partner for Apple Wallet NFC in 2015, on a secure messaging app in widespread use in high-security settings, handling user health data, and independent security research. I am based in New York City and have delivered multiple presentations to professional development groups since 2016. My comments come from the software development perspective, including extensive experience with API design and use. In the OSCAL ecosystem, this would be the perspective of 1) the need for automation of secure software development for applications, and b) development of applications as tools to work with OSCAL.

Comments based on:

As part of the comments, I reviewed the OSCAL model reference <https://pages.nist.gov/OSCAL-Reference/models/>, the OSCAL github repository <https://github.com/usnistgov/OSCAL>, and the OSCAL content repository <https://github.com/usnistgov/oscal-content>, as well as other concepts related to automation of security.

#	Organization	Commentor	Comment's Type (General [G], Technical [T], Editorial [E])	Line #	Section	Comment(Include rationale for comment and proposed change)	Suggested change
1	(Independent)	Mike Sanderson	E	78-79	2. The OSCAL Ecosystem	OSCAL includes both parts of the software data model: The Conceptual Model and the Logical Model. (The document refers to "information model" below, which should be equated to conceptual models. OSCAL provides conceptual models backed by specific logical models or schemas to ensure interoperability. This section should clarify this up front. This comment also applies to section 3.1 (164-165). Regardless as it is key to understanding what OSCAL is, conceptual models should be introduced in this introduction.	"OSCAL provides complete data models: each OSCAL Model is a conceptual models backed by a logical model (or schemas) to ensure interoperability. The conceptual models are grouped into three layers: Control, Implementation, and Assessment, with layers here being essentially a term-of-art for OSCAL conceptual models. The OSCAL conceptual models are designed to be broad enough to incorporate information that needed to represent security concepts and processes; that examples are provided of how the conceptual models relate and interoperate to each other; and that different models may be of interest to each participant in the security ecosystem.
2	(Independent)	Mike Sanderson	E	122-123	2. The OSCAL Ecosystem	A minor point, since "not be possible" makes it sound like OSCAL is a binary format that is not human-readable at all, which is much too strong. Since the advent of YML, there are current automated tools that ingest YML files with the expectation that they are simple, top-level files that are hand-created or hand-edited.	"Without editorial tools, creating and maintaining content will be a cumbersome manual process of working with YML, JSON, or XML files."
3	(Independent)	Mike Sanderson	E	164-166	3.1. OSCAL's Layers	It is unfortunate hand-waving to say "information model, which is "bound" to multiple serialization formats..." This is technically incorrect and hand-waves the entirety of the data structure and key-value. The logical data model is the core of what OSCAL is, how different vendors/companies/system can interact. It is also the hard part — getting the data into a standard data model is arduous mapping.	"The information model is represented in logical data models that include fields and structures that can represent details relevant to the use of the model for the anticipated uses."
4	(Independent)	Mike Sanderson	E	164-166	3.1. OSCAL's Layers	I have never encountered the phrase "information models." The preferred term is "conceptual data model" — https://en.wikipedia.org/wiki/Data_model#cite_note-6 citing American National Standards Institute. 1975. ANSI/X3/SPARC Study Group on Data Base Management Systems; Interim Report. FDT (Bulletin of ACM SIGMOD) — and at the least that should be mentioned. See also comment for 78-79: The conceptual models are grouped into three "layers", Control, Implementation, and Assessment (with layers here being essentially a term-of-art for OSCAL conceptual models); that the OSCAL conceptual models are designed to be broad enough to incorporate information that needed to represent security concepts and processes; that examples are provided of how the conceptual models relate and interoperate to each other; and that different models may be of interest to each participant in the security ecosystem.	Include recap of conceptual models / logical models distinction. It would be outstanding to include the layers diagram at https://pages.nist.gov/OSCAL/learn/concepts/layer/ is not included in section 1, include it here.
5	(Independent)	Mike Sanderson	E	189-192	3.1. OSCAL's Layers	Community members should not be making pull requests for the schemas. Community members should be making pull requests for the Content - Both OSCAL Content Artifacts and OSCAL Content Operationalized, as described in Figure 1 and lines 80-91	Community members are encouraged to participate by forking the <i>oscal-content</i> repository, as well as listings of tools and editorial examples, making improvements, and submitting pull requests to contribute to the ongoing development of OSCAL.
6	(Independent)	Mike Sanderson	E	203	3.2. The OSCAL Advantage: Standardized Efficiency and Interoperability	Unfortunately, many companies see vendor lock-in as their goal—just like machine parts in the 19th century, whose relation to NIST should need no elaboration. It could be framed, at the risk of being glib, as "opportunities for big players to rip off new innovative players easily"	Cut "reduced vendor lock-in"—maybe "increased opportunities for innovative vendors" and "allows incorporation of new features into existing deployed or existing-vendor tooling"
7	(Independent)	Mike Sanderson	E	284- 285 (referencing 270-378)	5.2. Elevating Assessment	While a machine-readable format may allow advanced applications, it will also allow immediate, non-speculative traditional automation	Adoption of OSCAL's machine-readable format will immediately allow for efficiencies using procedural, human-engineered automation. It also is essential to allow the next-generation of tooling to benefit security.
8	(Independent)	Mike Sanderson	E	388-addition	6.1. Join the OSCAL Community	Probably needs a more concrete connection — starting with the Layers, as different layers will be of interest to different participants.	This can include looking at what layer of the OSCAL conceptual models is most relevant for your use case, and identifying how your existing concepts map to OSCAL data model's fields. You can also look at existing NIST cybersecurity concepts that have been translated into OSCAL format, and see how that could inform your applications to OSCAL formats in relation to them.
9	(Independent)	Mike Sanderson	E	388-389	6.1. Join the OSCAL Community	Regarding the logical data models, the driving of the adoption of logical data models as a format is a highly technical process requiring widespread detailed technical implementation, and, while it is central to my software development experience from beginning to end, that process is outside the scope of these comments. Having seen how companies that want to push adoption of formats proceed, widespread adoption of the logical data models will require an enormous push with specification, examples, sample tooling, direct developer relations and other steps that should not be underestimated.	API designers and developers are encouraged to look at how their data fields map to fields and concepts on OSCAL data models, and if adoption of OSCAL models will provide an opportunity to include a standardized terms that can increase internal efficiency and connect tasks to the larger cybersecurity infrastructure.