Today's presentation is the OSCAL Futurist: Musing on What is Possible and What is Needed by Greg Elin founder and CEO of GovReady.

Greg, are you ready.

Yeah, let me go into presentation mode.

How are we looking does, it look good it? Looks great.

Great so this is the OSCAL futurist presentation and first a brief announcement.

We announced yesterday that GovReady is now a part of the RegScale team, so GovReady is now, has been acquired by RegScale and I'm excited that I'm going to be the OSCAL Evangelist and Senior Principal OSCAL Engineer.

So I'm going to get to spend even more time working on OSCAL and compliances code.

So just want to share that with everybody.

So, you know, we we've spent a lot of time looking at implementation, deep in the woods, figuring things out with OSCAL, and what I wanted to do was pull back a little bit.

And one of the reasons is that OSCAL, the way that I've always thought about it and when I talk with David Waltermire and Michaela, OSCAL is a first necessary step in order to automate and accelerate compliance, it's not the destination.

So I wanted to start by just setting the baseline, what are some of the project goals for OSCAL, remind us of those.

So one of course is to decrease paperwork significantly, make it less of a burden.

Another is to improve system security assessments and I think when we're talking about improving the efficiency of when we do an assessment, that we can do them more timely with greater accuracy.

We want to enable continuous assessment, this is something that I think has always been in everyone's mind.

How do we begin to do assessment at the same speed that we can now do software development and deployment?

There are a couple of key design principles, the inoperative inoperable data formats so that we have machine readable formats that can be used by different tools, and the idea that when we're building this it's relevant now and it enables a better future.

So these are some of the goals that NIST has been working towards and what I want to do is so, rather than worry too much about specifics of implementing OSCAL right now, where can we, where can we go with it, where is it taking us, how do we get there, what kind of challenges do we have, etc.

So, the OSCAL futurist makes the following kind of easy predictions.

One, that continuous ATO becomes mandatory, especially in government.

Two, security tools start speaking OSCAL-ish.

Three, integrated GRCs solve the sensitive systems of record dilemma.

Four, assessors treated as tech savvy.

Five, community drives adoption.

Six, SBOMs drive component creation and sharing, very excited about that.

And seven, DSLs emerge for compliance content.

So I'm going to go through these one by one and then we'll answer

questions at the end.

So, continuous ATO becomes mandatory and this is actually I think one of the media sections that I have so I'm going to spend more time on on this prediction.

So, probably many people in the room are familiar with these books, the Phoenix Project that came out in January 2013, and kind of really established the idea of DevOps and what the potential there was infrastructure's code etc.

Followed up in November of 2019, with the Unicorn Project, kind of talking about those DevOps ideas in the context of startups and and new projects, and what I wanted to share with everyone is as of September 2022, a new book in this series is out, Investments Unlimited.

It's only about 150 pages, it's written in that same friendly narrative style but what Investment Unlimited focuses on is audit compliance.

Basically this book is, talks about the fact that in rushing to do DevOps and build all these pipelines, we kind of forgot about compliance.

And so how do you integrate compliance into the pipeline and, in this case, it's a study of a bank that's under investigation by the SEC for their compliance processes.

So essentially, to kind of do a spoiler on the book, the book it looks at the the CICD (continuous integration and continuous deployment) pipeline, the DevSecOps pipeline, which is in gray and this book talks about a few things that really need to be added to it.

So, in order to do compliance in the pipeline, to collect evidence, to gather evidence continuously as we're building the system, one, we need to add scans as part of our build process.

So, as we're building assets, make sure those assets are scanned.

When we're assembling the packages and gathering up the evidence, we have to add in digital signatures.

This kind of goes back to the software supply chain and it's also the idea that, okay I built this package but what exactly built it, how do I know I can trust it, and can we digital sign it so when we put those packages into our artifact repo they can be trusted packages.

But what we're also going to start putting into a repository, what we're also going to create is an artifact, is going to be the audit evidence.

So, we're going to run our scans, we're going to sign the results of those scans, and we're going to capture that and other information, audit information, and that's part and we're going to treat audit evidence as just another artifact that we build and we include, and then we can begin to have risk gates where we look at that audit evidence and see if we did our code review, if we have them, if we have, if we don't have too many, if we've addressed all of our critical vulnerabilities etc., and make sure that we're in, that we're in a good state there.

if you don't want to read the entire book there is a shorter white paper by IT Revolutions which talks about this DevOps automated governance and reference architecture and it's a shorter read but both, but these I think what's powerful about this book coming out is it's going to provide a lot of momentum and something to talk with with executives of how we're going to do automation in the pipeline and OSCAL of course is going to play a role in that.

In fact I've started having some conversations with parties about using OSCAL to collect that evidence in the pipeline and so I wanted to talk a little bit that this is actually creating a bit of a challenge I think for our community at the moment.

In the OSCAL model we kind of build content and build data left to

right starting with the catalogs and the profiles and the system security plan, but when we look at the data and we go to understand its meaning, we refer back right to left.

So in the OSCAL model the evidence that's collected in the pipeline is really put into the system assessment results module object or model rather.

So it goes over there on the right and and I've had a couple conversations with teams that are operating the DevSecOps pipeline and they want to start including evidence and are looking at OSCAL of how to do it and what is apparent is, OSCAL the system assessment results kind of assumes you have all these other pieces in place by the time you get around to adding in the evidence.

But organizationally and even with a lot of the tools, we're still working our way right to left to build out the content.

So we've got a bit of a gap at the moment and I think one of the challenges is, how do we help this the federated teams, the teams that are working on different parts of the pipeline in the process, how do we create some data structures that enable them to begin to collect evidence even if we're still in the process of building out all of the pieces left to right.

So I wanted to share that and I think it's an interesting area for us to think about.

Now while we're thinking about shifting compliance left and everything, the furthest left we can go is actually policy.

Also, kind of realize that we've been focused very much on the tooling on the data structures but within many government agencies and organizations the current policies and the current standard operating procedures only talk about that you will do the risk management framework, they only talk about that you will do system security

plans, and there's very little if anything in those policy and standard operating procedures about doing that in an automated fashion.

There's nothing in the policies about structured data or machine readable formats and it this is also true that we don't have in the various RFPs that go on in the market for development or for DevSecOps, there are no requirements currently in those RFPs for automated processes.

So I had the opportunity to draft some policy language and I wanted to share that with everyone that was attending today, and I'm going to go through these points just really quickly.

And what we have seen, and before I go through these points, when we look at open data, when we look at agile, when we look at SBOMs, what we see is there's often community language for policy, for procurement, already out there that the government could utilize in order to make these things happen.

So I think that this is actually, as we go into the future, we need the policy language that can easily be adopted in order to drive, that can easily be used and included, in order to drive the adoption of automated and accelerated compliance.

So I'm going to walk this through.

It's the agency policy to support the acceleration and measurement of the RMF process through approved digital and automated mechanisms and vendor-independent data schemes so that we can programmatically represent the SSP and SSP documents in a machine readable, NIST recommended formats.

So the idea that we're saying up front that we want to have, that we want to be in the machine on digital age.

Point number two is that we want to generate and update the content of the information system and and make it intended for human consumption with the changes to the information system.

So when there are in changes, when changes happen to the information system, we're making it policy that we are updating our documentation at the same time and we're going to do that through digital means.

And that's point number three is as well is the idea of supporting real-time or near real-time processes, to share information that we find out about vulnerabilities risks and etc., and that we can share that information to need to know parties.

The idea that we're sharing it but we're sharing it to need to know.

Obviously this can't happen right away but we want to make it policy that we're progressively replacing manual processes around compliance and with more, with automated data interchange APIs and other digital tooling.

We want to collect information from the mission areas and agencies to support audit activities.

The purpose of this policy is often when we get around to doing an audit, we don't know who exactly has the information, especially in a large organization where people move around.

So we want to make it policy to emphasize that we're going to be collecting audit information from different lines of business etc.

Number six, part of the power of being in the digital world and having an automated process is that we can measure, we can measure the way

that we do our work and our work gets done.

So, we want to make it a policy that we're actually going to start to track the work, how long it takes for us to find audit information so that, or to do anything basically so that we can identify bottleneck.

The last three, I think that these are actually a pretty important, aggressively again the idea that we can't do this right away but we do want to associate progressively, information system data in accordance with risk and organized approved labels and categories to facilitate zero trust capabilities.

The idea here, as we have to start tagging our data and we have to do that in a way where as we move to the zero trust architectures, it's easy for us to apply this, apply the trust properties to the data.

Number eight, associate information assets and their workloads to specific information systems and assessment boundaries when such assets and workloads are created or modified in order to immediately identify the responsible party operating the asset and authorized to make the modifications.

The purpose here is, we can we've got sections inside of OSCAL where we've got the inventory of the system and all the assets.

In theory that's awesome but it only will work if we are actively, if we are actively associating those assets when they're created with the systems, the business, and the business owners who are accountable for them.

It's not enough for us to simply associate a workload with an AWS account because there may be many parties that are associated with that account.

So, we want to make it policy to address the idea of tagging the assets when they're created.

And in a similar fashion, we want to create a mechanism that if an asset starts to be created or modified and it's not associated with anything, then we can immediately stop that asset from being created.

So, we never have assets running that aren't tagged in such a way that we can then subsequently associate them with uuids and other content in OSCAL.

So as I said, that was kind of the longest presentation but I do think it has some really useful information to drive, to make continuous ATO and what I mean continuous, continuous in real time ATO, doable and mandatory.

So, prediction number two, security tools start speaking OSCAL-ish.

What I mean by that is, when you look in OSCAL, we've got a lot of small units of information that refer to responsible parties, users, assessment subjects, relative relevant evidence, and these are just a few but each of these represent entities in the real world that are long-lived and right now they're buried, we have a lot of these buried inside of our large OSCAL objects and I think we really need and we can expect security tools to actually want to talk to exchange information about these objects in small hunks.

I keep thinking about how useful it would be if I had a registry of responsible parties and I could easily go get the response, I could go look up a responsible party outside of OSCAL and I could begin to manage certain information.

And so I do think that we have many security tools that are only going to need specific information.

Also, APIs and command line interfaces are fundamentally designed to update just bits of information and so I think that there's a question for NIST to think about and for us to think about as a community is, how do we exchange these small hunks, what kind of wrapper might we need in order to pass around a responsible party or send some update information to update information on a responsible party etc., and then allow all the different tools talking to use that.

And then of course, how do we manage these small hunks of information, where do they live, can they live inside of our repositories, do they live inside of databases, and how do we match them to a variety of things and in the same vein, we also need to talk about, at some point, uuids and props.

And what I mean by that is, are uuids only inside of the OSCAL objects or can they be found in a registry or elsewhere, what makes sense? Similarly, how do we make the props more reusable and dry across different systems.

In parentheses here I've also mentioned durations and signatures as we think to the future and the idea that the many durations that we have in the organizational defined parameters, there's a real opportunity there to create a variety of mappings and standards for the durations and similarly signatures.

Digital encrypted signatures are clearly very important with the SBOMs, they're clearly very important in collecting the evidence, which we want to do for the for the SAP.

And I have to admit, signatures have always felt like a very cumbersome, complicated process, and I think about how easy it is to now, through let's encrypt, to do SSL certificates and the idea of how to and there's actually a project called six door and the idea of making signatures much easy for parties to generate in order that we get the benefit of the digital signatures.

So, number three, prediction number three, integrated GRCs solve the

sensitive system of records dilemma.

What is the systems, the sensitive systems of record dilemma? I am glad that you asked.

So, what we have encountered again and again is that everyone imagines they're going to set up a GRC or another enterprise system of record and that system of record is going to centralize all this information that different users need to use.

And what they're thinking about is the network effects that we get time and time again, whether it's through aggregated search, or social media networks, or the web itself, that all this information is available.

However, in this case, when it comes to cyber security and compliance, we're actually centralizing sensitive information.

And so in reality what happens, as we centralize more information, your trusted users have access to that central information and those fewer trusted users with access to that aggregated information start spending more of their time sharing that information with a larger number of parties without access.

As we aggregate more sensitive information, we're actually pulling that sensitive information for more and more parties that are in different lines of business, thinking we're all going to put it in the same database.

But then the organization has this resistance to let everybody have access to it and a really common example here is, if I've got a POAM there's a very good chance that that POAM is going to have to be looked at by a developer on a contractor team and the developer on the contractor team or this developer subcontractor on the team is not going to have access to the Enterprise GRC tool.

And so the trusted users spend their time still moving content around in spreadsheets etc.

So what's the solution and the solution is the is the integrated GRCs.

So with the integrated GRCs we extend the data management back to multiple lines of business in order to get the work done.

The reality is, in our large organizations not everybody works the same way, not everybody is even on the same network, there's all sorts of things going on, so what we need to do is have GRCs that are integrated with each other and that are optimized or that are, when they're put in place, they're customized for different lines of business but they can exchange information back and forth and automate that and control richer and and provide better role-based access control.

How does everything speak with each other, of course through OSCAL-ish.

Okay, prediction number four, assessors treat it as tech savvy.

I think we need to, it's time that we as developers recognize that our assessors are techies too.

Every information worker these days has a smartphone, after the pandemic everybody uses videos, everybody knows how to use very many applications, we are dealing with everyone in the workforce these days is really, especially if they're young, are part of the digital generation.

Our assessors and our ISSOs are overwhelmed just like, just like everyone else's, and they are looking for better tools to do their work.

I was at the ISACA GRC conference and I was struck by how generic many of the presentations talked about the technology specifics and how hungry the auditors were to get into greater detail.

And so I really think that and maybe it was a mistake that I was making that I was kind of pigeonholing many assessors and auditors as people that were using spreadsheets and that they knew the compliance frameworks but they didn't really know the technology.

And as I spoke with more people at that conference and I took a step back, it became apparent that assessors are quite tech savvy these days and we as tool makers need to think of them as very technical users who just like us are overwhelmed.

And it's not that we need to give them better tools rather other than often try to think about how we overly simplify things, The right tool and the power tool is often the most simplest one to use when it's

well fit.

Prediction number five, community drives adoption.

I think the big point here is that it's time for the community to step forward and take a much more active role in creating documentation, and examples, and preparing information.

We need to make the community's engagement and the community's production of content and material much larger than NIST.

I think we've often, NIST has done an amazing job moving the OSCAL format forward, organizing workshops, but it's time for the community.

We need a lot more documentation, we need a lot more videos, in order to gain the adoption from all the different parties involved in compliance and it's up to us in the community to set that up and do it and not wait around for NIST.

Number six, SBOMs drive component creation and sharing.

So, probably everyone on the call is aware that in September OMB came out with a memo that piled on to the executive order that came out in 2021, and basically saying that agencies needed to implement SBOMs.

I pulled out a couple of the key bullet points that one, agencies are required to obtain self attestation from software producers before using software.

This is a big step forward and two, agencies may obtain from software producers artifacts that demonstrate conformance to secure software development practices.

And what we're talking about here in the primary is the SBOM, what the software bill of materials.

But I think it's the OSCAL community and, as when we were developing GovReady, we always thought that the community would get tremendous reuse from the component model, that if we were associating control implementation statements with individual security components that were used to build the system, it would become much easier to build the system security plan.

You identify the components, you build the system security plan, the system security plans have the control implementation statements, it's easy to assemble the draft control implementation statements and be able to reuse that.

But it's always been a challenge for people, it has proven a challenge for people to create the components and without the components we

don't get that reuse, we don't get that acceleration.

But now that it'll be a requirement to produce software bill of materials, I think what's going to happen is, we're going to start with the simple software bill of materials listing what other software is compose but those software bill of materials are very quickly going to evolve into the components themselves.

And so we're going to get software bill of materials which are essentially components in disguise and that this is going to be excellent once people are using some of the new data formats for representing their their SBOMs.

It's going to be very easy for them to take the next step and generate from that SBOM, generate and manage their OSCAL components.

And now suddenly, as the tooling begins to form in the community to share the s-bombs, the sharing of the components will ride on top of that as well.

So I'm quite excited about this development.

Finally DSL's, oh little typo, DSLs emerge for compliance content.

So, how close is OSCAL right now to a domain-specific language? When I tend to think of OSCAL as a database person, I tend to think of OSCAL as a data interchange format.

More specifically, a serialization of various objects representing the system and so it's a serialized data interchange format.

And what I and and for a long time I think it was very easy to look at OSCAL and think okay, if I'm replacing the content in each of these areas, I'm going to be able to assemble my documentation, I'm going to be able to exchange information, and so OSCAL is this nice representation of a variety of entities that we need to know about for our system security and compliance.

But what's important here is that OSCAL is very entity or noun oriented.

All it does is what it primarily focuses on representing the actual distinct elements and details of an existing system.

Here I have an example of Docker compose and I think what's interesting about Docker compose, you can see in the white text near the top, is that Docker compose has verbs.

It has the run verb, it has the copy verb, and because it has the run I can actually run a command that's available to me on the operating

system.

So if we think about Docker compose and how Docker works in generating containers or building images, it builds those images one image layer after another image layer and each image layer that's built in Docker is a transformation from the previous layer and it's a transformation because we've changed some piece of data or we've actually executed a command.

And you can see and, if you look further down, you can see within the services we can specify particular information but when you look at the web service we can actually build and we've got commands, and we also have this sense of environment.

So, when we look at Docker compose as a DSL, it specifies services, it performs actions, it supports variable passing, and there's a runner that builds something else.

What we've always thought about, that our tools would create the OSCAL files, but I think when we, so we've always thought about we're going to have tools that create that output OSCAL that manipulate the OSCAL information.

But I think that there's a step here when we begin to look at Docker in which we're, the reusable asset we have is actually the DSL script or the composition file, which then Docker changes into the asset.

Very similar is Terraform.

So, if we look at Terraform in the comments, you can see again that we're setting some, we're actually providing specific values in order to allow something to happen, in order to do something, and then so I just grabbed some snippets of Terraform.

A lot of configuration information, again configuration information that can be set by parameters and passed around as variables within the declarative statements.

And but we also have conditions and actions, so I do think that there's a the next step in our evolution and it's inevitable is that we begin to look at DSLs and pipelines for actually starting with smaller bits of information and having verbs and other descriptive of language which then generates out our content which is then gathered together into our larger OSCAL objects.

What are and I think before I give a couple of examples, again, if we think about CSS and and the evolution of styling of web pages, how we started with just a few very specific attributes associated with HTML tags and then we move to CSS and then people said, look I'm tired of writing all the CSS so I'm going to have, I think it's SAS, I'm going

to have other tooling which allows me to write what I want my CSS to be and then it will and then we can generate that CSS.

And I think in a similar fashion one of the powers of XSLT is that transformation, is that ability to specify the transformations that we want to make.

So I think it's, of wrap up here, I think that there's, it's very interesting to think about.

Let me give an example, right now I think we tend to think of a person, you know a responsible role or a party, we think of that as, oh Greg exists, Greg is the system owner, or Greg is the responsible party, but what we could do instead is we could think about a DSL which allows us to create a responsible party, to actually produce a Greg.

And we could abstract the and we could have certain and we could begin to think of it in a decorative manner where we say okay, and let me go back to the Terraform example, I want to create a resource which is a responsible party and I have these kind of abstract responsible parties and I want to be able to generate them but then I just want to be able to plug in the specific name of who that responsible party is.

So within every organization I'm going to have a SOC, within every organization I'm going to have a risk owner, I'm going to have certain responsible parties again and again and again.

What if I could specify them out and then pull in from a variable file who those responsible parties were and generate that, and then, if the responsible party ever changed, I don't have to go in and update data.

I just have to change a single parameter and re-run my script to regenerate my parties, my responsible parties, and then regenerate all of the content, the SSPs and other content, that can read that OSCAL-ish description of the individual parties.

And that really wraps up my presentation, I'm eager to do a little bit of question and answer because I have I I'm eager to do some questions and answers and get a little bit of feedback, so thank you very much.

Thank you Greg, yes.