Progressive FINERP meeting Overview January 24, 2023

Tue, Jan 24, 2023 1:00PM • 53:56

**SUMMARY KEYWORDS**

terraform, diagram, progressive, rsm, aws, vpc, cloud, mule soft, workday, applications, infrastructure, flow, developers, services, release, john, mule, environment, project, traffic

What our program started off like so we engaged with Deloitte back in 2018 and started deploying both Coupa and workday. And across the top here we have here 2019 20 and 21. And we're a little longer after that, because things like COVID and some changes we made to the program. Beyond points here were kind of the cloud updates. So we have several releases that we already have under the belt for the solutions. We went live in release one you've heard some folks on the call already mentioned that release one on or about fall, the date are going to change here in a second. We went live with Coupa in support of sourcing invoicing, spend analytics, things like that. When we went live with Workday, but not the workday, you think workday is a very popular brand for human resources. We're among some of their biggest customers now for workday financials. So we went live with Accounts Payable fixed assets, banking settlement and some project work and General Ledger in our first release. And then things change. So this is the animated version of that where the Navy boxes are the work we've been doing with Deloitte in the scope of things we've called FinERP. Probably everybody can see that if the font small don't worry about it. This is just context setting. So we did release one did Coupa and workday and general ledger. Then we started some other relationships and some other ideas. Took a small pause here and then 2021 delayed release two, which is the scope of this engagement for agent bonus payments and claims payments. So those two features we shorthand as release two or FinERP. Part two other releases occurred release three for some other changes. I'll just build out this slide because there's things we've altered since then. That's our main focus here. So is trying to set up the context that we have a large program that's been going on multiple years. That's got cornerstones a two major software as a service vendors and Coupa and workday. That's that and then in this slide deck. The skip around here because not all of this is relevant for everybody. Here's how we were thinking about the architecture and when we put these systems together. So left to right is the flow of data. Usually, we have our on prem source systems, some for the policy area, which is a mixture of mainframe and non mainframe things are building our claims areas other things incorporate all those things land data in our sub ledger. And then in the cloud where we've hosted stuff on s3 buckets. We pick up those files from s3 and we process them with tools available to us like the mule soft platform, and then with mule soft and those other things we go around.

*In 2018, the program engaged with Deloitte and began deploying both Coupa and Workday software solutions. The program has undergone changes, including updates and delays, due to factors such as COVID. The program has several releases under its belt, with the first release including Coupa for sourcing, invoicing, spend analytics, and Workday for Accounts Payable, fixed assets, banking settlement, and General Ledger. The program has also added relationships and ideas, and a small pause was taken before the delayed release 2, which included agent bonus payments and claims payments. Release 3 and other changes have also occurred. The program has a large scope and has been ongoing for multiple years, with Coupa and Workday as its cornerstones. The slide deck also includes information on the program's architecture and data flow, starting with on-prem source systems and ending with cloud-hosted data on s3 buckets, processed with tools like the MuleSoft platform.*

There's a hyphenated box here around the series of assets that are part of the workday platform. And then here's Coupa. As we originally conceived it, we thought we're going to take data out of Workday and Coupa and put it in PD W, which is a Microsoft parallel data warehouse appliance. We have since updated architecture to use snowflake. That's just for context that's outside of the scope of this engagement. But we also have data in the cloud. And then all this stuff produces reports both internal and external. Because corporate finance and then here's a description of some of those things. So this is just shrunk a little. Here's kind of the slide it started us to where we are today. We had an architecture previously that was on mule softs, hosted platform as a service called cloud up here in the upper left hand part of the diagram. We're going to have a path here and if this was a whiteboard, I'd be drawing all this for you. So instead, this is a numbered flow of how things go. The current path we expected was 23579 10, seven and then 13. And we learned later that that wasn't a good idea in studying the cloud hub features and instead we went out to do 2357 910-775-3414 Because of data loss prevention and inspections. Let me walk you through that. claims in the upper right. makes a call that Ping Pong is around our networks, routers, firewalls. And the other on on premise data loss prevention. Thanks. Those calls then make their way to an AWS cabinet that in wherever we keep it, Ashburn, Virginia but we've got multiples, you'll hear about that here in a second and on its path from five then it went to the application code that was running on platform and the service on mule hub. Cloud hub on mule soft department. That code would do a verification step where I would ping back seven 910. And if I'm still getting this, right, we had an AWS private to progressive database where we would validate a request a bank, us a 10 whatever you wanted to validate, come back and if you passed that test, we would didn't say MuleSoft go ahead and make that payment and go down this hyphenated path line 13 to the workday cloud, where you just went from okay to pay, validate a transaction and told workday please make this payment. This was a headache for us because we had cloud to cloud traffic and at the time, April, almost two years now, the cloud security team did not have cloud based security technology. So they wanted us to hairpin everything back in our data center. So using Platform as a Service was a dead end. And the path we wanted to do was instead to come all the way back. Go through the progressive equipment here and number three, go down path for where we would have inspected it and send it out to 14 and workday. That path was inspectable because it was it put progressive equipment in between the mule soft and workday transactions. However, cloud hub turned out to be an offering we could no longer use so that had to get re hosted. And to tell you more about that you'll hear from the team here in a minute. That was a very fast and high level review of where we started and the course corrections we made, which explains the complexity of the architecture. You're about to see a positive for questions

*The program originally intended to take data from the Workday and Coupa platforms and put it into PDW, a Microsoft parallel data warehouse appliance. However, the architecture has since been updated to use Snowflake. The program produces both internal and external reports for corporate finance. The program's previous architecture was hosted on MuleSoft's platform as a service called Cloud Hub, but due to issues with data loss prevention and inspections, the architecture had to be changed. The current path for data flow is 23579 10, 7 and then 13, but this was later found to not be a good idea and the path was changed to 2357 910-775-3414. This new path includes calls that Ping Pong around networks, routers, firewalls, and on-premise data loss prevention, which then make their way to an AWS cabinet in Ashburn, Virginia. The code running on the MuleSoft department would do a verification step and if the request passed, it would tell MuleSoft to proceed with the payment and go down a hyphenated path to the Workday cloud. However, this caused headaches as it involved cloud-to-cloud traffic and the cloud security team did not have cloud-based security technology. The solution was to re-host the system and put progressive equipment in between the MuleSoft and Workday transactions, so the path would be inspectable. But Cloud Hub turned out to be an offering that could no longer be used.*

Well, what so from a heart just from a 10,000 foot view, our claims system is looking to make payments to customers, vendors, etc. And there are they want we're using workday to kind of on the backend and do that work for us. A lot of the complexity comes in that we pay a lot of different people and we pay some of those vendors quite often. We want to be able to collect that data and do the appropriate tax reporting for those vendors, as well as paying individuals that we may only pay once or twice ever. So there's a lot that goes in to the integrations in order to try to reuse a lot of that supplier data and to allow us to utilize different payment methods on the back end. So from from a functional perspective, it's pretty simple. You know, they send us the data they want to pay. So who do you want to pay? How much do you want to pay? What's the accounting information that goes along with that? invoice and then we use mule soft and AWS tools on the back end to provide the integrations between our on premise services. So whether it be claims or agency, we mule soft makes use of the database SQS s3, you know, the various tools you might see from AWS and then it is sending that information along to workday. workday does all kinds of integrations with some of the banks. We pull some of that information out we print it we send it so the printing is all done on premise. So we're pulling that information out of Workday and we're sending it back to our on premise services to print documents. There's also services allow them to make adjustments you know whether it be canceled those payments, cancel the invoices, updating statuses updating and counting that kind of stuff. And then to keep our on premise systems like claims up to date as to what's going on with those transactions. There's all kinds of events that we send back to them through SNS node notifications, so invoice failures. You know what happened with whether we cancelled an invoice or a payment, whether documents were printed, etc. So there's, you know, from our perspective, there is some what we consider to be highly sensitive data within these these transactions. So whether it's taxpayer IDs, names, addresses of our customers and claimants, so all of that is kind of flowing through these integrations in AWS.

*The claims system is looking to make payments to customers and vendors, and is using Workday as the backend to handle this. The complexity comes from the variety of people that are paid and the need to collect data for tax reporting for vendors, as well as the need to pay individuals that may only be paid once or twice. The system uses Mulesoft and AWS tools to integrate with on-premise services such as claims or agency. Mulesoft makes use of databases, SQS, s3 and various other AWS tools to send information to Workday. Workday then integrates with banks and some of that information is pulled out and printed on-premise. The system also allows for adjustments such as canceling payments, updating statuses and keeping on-premise systems like claims up-to-date with the transactions. There are also events sent back to the on-premise systems through SNS node notifications, and the data being processed is considered highly sensitive, including taxpayer IDs, names, and addresses of customers and claimants.*

1. *What security measures are in place to protect sensitive information, such as taxpayer IDs, names, and addresses of customers and claimants, during the data integration process?*
2. *How are on-premise systems, such as claims, protected against external threats and attacks?*
3. *What access controls are in place for the various tools and systems used in the integration process?*
4. *Are there any dependencies on third-party systems, such as Workday and AWS, and if so, what security measures are in place to protect against vulnerabilities in these systems?*
5. *What visibility and monitoring capabilities are in place to detect and respond to security incidents?*
6. *Is there an incident response and disaster recovery plan in place in case of security breaches or system failures?*
7. *How is the integration process tested and validated to ensure it is free of errors and vulnerabilities?*
8. *What training and awareness programs are in place for employees and contractors handling sensitive data?*
9. *Are there proper access controls in place for the on-premise printing services?*
10. *What security controls are in place to protect the system from external threats such as DDoS attacks, malware, or phishing?*

Angelo to kind of go through a network diagram that he produced.

I'm going to start off with the lab, sort of environment that's a little bit simpler than the rest of the environments. Dave Q H, stress production or pretty much copies of each other. So I'm just gonna talk about one of those environments in context. The difference between the lab environment is really it's really just kind of there for development purposes only for developers to to be able to access information and perform the development activities. So the one difference between lab and the rest of the environments is where the mule soft code actually executes. And when you're when you're busy developing these mule soft flows, etc. You actually run the whole new runtime locally on the developer's laptop. It is in an Eclipse based IDE. You develop all of these flows, and then when you test them, you actually run a copy of the mule runtime service on on the developer's workstation, I'm sorry, I don't know why that hadn't changed, right. So in all of the diagrams, you will see this comment boxes in the middle, which is really the connectivity infrastructure, and we've got progressive applications. We've got the corporate ID developers in the bottom right hand corner they all connect through dual redundant links. And John alluded to the set the main path kind of flows through some progressive infrastructure, which at the time of this diagram was in bankies, there have been some updates to the infrastructure your you may sort of your references to Metro and some updates to the data center infrastructure, and ultimately, it goes through a direct connect link to Ashburn, as John said, that's the kind of the primary loop the primary route that is followed and then there's a secondary which flows through bank of waste and connects to Equinix. In Dallas as a backup. Okay, so this really just gives us the redundant connectivity between the progressive network and what is in AWS. So from a development perspective, the traffic originates, as I said, from the developer's workstation. It flows through this infrastructure and they need to access to different types of AWS services. Some of those AWS services are actually inside of VPC. So those are like our Aurora, Postgres sequel RDS instance. We had spoken in the design, whether we were going to be using redshift eventually. We decided not to. And then some of the lambda functions are actually deployed within the VPC as well, just in terms of the services that they need to access and for security purposes. Then we also use Brian and John also needed to this reuse some of the other common sort of global AWS services such as s3 AWS glue SQS SNS. So those services actually sit outside of the VPC part of stock collected through this direct connect infrastructure back to the progressive network. So the neck network parts are pretty much the same. So if I move on to the production environment, so the difference in the production environment is we have the mule soft applications now hosted in a Kubernetes infrastructure. Which were hosted in an AWS account. Going back to the introduction that John gave about the inability to investigate this traffic and know what traffic was actually able to leave that cloud hub VPC that was one of the main concerns of security as well as as well as the inspection. We had no control over the egress. We could not shut down the egress in that environment. It had permanent connectivity to the internet. So the eventual solution that we've come up with is we're hosting the mule runtime in a Kubernetes infrastructure. hosted in AWS inside a progressive account. The Kubernetes cluster is actually managed by the Cloud team and the fin ERP release to project is actually just a tenant in the shared infrastructure. We have our projects spun up in this Kubernetes environment and we deploy our applications into this Kubernetes environment. Was there a question here, sorry.

*The speaker is describing the different environments that are used in the organization's system, specifically focusing on the lab environment and the production environment. The lab environment is used for development purposes only, where developers can access information and perform development activities. The main difference between the lab environment and the other environments is where the mule soft code is executed. In the lab environment, the mule soft code is executed locally on the developer's laptop using an Eclipse-based IDE. In the other environments, the mule soft code is executed in a Kubernetes infrastructure hosted in AWS inside a progressive account. The traffic originates from the developer's workstation, flows through the connectivity infrastructure, and the developers need access to different types of AWS services such as Aurora, Postgres sequel RDS instance, and lambda functions. Some of these services are inside a VPC for security purposes, while others sit outside of the VPC. The speaker also mentions that there have been updates to the infrastructure, but the overall network architecture remains the same. The production environment is different because it uses a Kubernetes infrastructure to host the mule runtime, which is managed by the Cloud team. This solution was chosen to address security concerns around the inability to investigate traffic and control egress in the previous environment.*

1. *In the lab environment, the mule soft code is executed locally on developer's laptops, which could create a security vulnerability if a developer's laptop is compromised or if a developer has access to sensitive data without proper security clearance.*
2. *The system depends on third-party systems such as AWS, which could introduce security vulnerabilities if those systems are not properly configured or maintained.*
3. *The system is using Kubernetes infrastructure which is managed by the Cloud team, if any misconfigurations or vulnerabilities in the Kubernetes infrastructure, it could lead to a security breach*
4. *The system is storing sensitive data in the AWS services, and there is a lack of proper access controls in place for these services, which could make them vulnerable to unauthorized access.*
5. *The system might not have proper monitoring and incident response plan in place to detect and respond to security incidents in a timely manner.*
6. *The system's security is dependent on the security of the developer's laptops, which may not be as secure as a dedicated server.*
7. *The system is using direct connect links to AWS, which could lead to potential security vulnerabilities if the links are not properly secured.*
8. *The system's security is dependent on the security of the progressive network, which may not be as secure as a dedicated server.*
9. *The system relies on printing services on-premise that might not*

I was going to just observe we're the only tenant in this entire thing, right.

I believe that is called the case. John. We were certainly the first ones that made use of a web of OpenShift hosted on AWS versus on premises. There are other mule soft applications running on OpenShift hosted in the data centers. In the progressive data centers. We were the first project that actually hosted the Kubernetes OpenShift environment in an AWS account.

As far as I know, we are still the only ones in that environment, but it was set up to be a reusable pattern. So that other groups within progressive could also leverage it.

Yeah, thanks for that.

And then, going into a slightly more detailed diagram, it is very much the same. We were just showing some of the additional cloud connectivity the Direct Connect gateway and the transit gateway. Now. When we set up this entire environment, we had a lot of conversations with the cloud and networking teams are everything should be configured. To be honest, it was so long ago and I don't remember the exact reason but we had to make a decision that one of these V PCs could be connected to the transit gateway, the other one could not. It had to do with the number of connections that we were utilizing and how many were still available. So ultimately, what we had decided upon at the time was that because the OpenShift environment is potentially a shared environment that could be utilized by other projects. It made more sense for that environment to be connected to the transit gateway. The services that are in the funder of specific VPC are of a more limited nature. They connect directly back to the direct connect services are interconnected through the gateway through peering connections and network configurations at the Direct Connect level. If you need more information about this connectivity here, we'll have to put you in touch with the cloud networking folks. For them to explain what the reason was that we didn't connect both of these V PCs to the transit gateway, but there were some routing limitations and concerns at the time. So this was the ultimate sort of setup. So once again, whether the traffic originates now from a developer's laptop or from one of the pier applications the claims direction or the the agent system, they flow through the infrastructure, they hit a direct connect as a routing logic based on the addressing logically then if the if the traffic is destined towards SQS, SNS s3, etc. It flows down this path into the AWS environment. These applications yet don't ever directly talk to our Aurora database. Our developers and support staff may need to access their database so they would follow the network path to get down to this Aurora database that sits inside the VPC. And then all of the requests that come from the times and agency systems hit me also have the application that is running in this OpenShift Kubernetes cluster, from their cluster then the traffic flows back into the Direct Connect environment where there is a firewall device that can inspect the traffic, ensure the DLP controls that the network guys wanted to the are holding the traffic back onto the network. And then it drops back to either these other AWS services or it connects to the Aurora database that is in the VPC.

*The diagram being discussed is a setup for a cloud environment with multiple Virtual Private Clouds (VPCs) and connectivity to various services, such as Amazon Web Services (AWS) and Aurora databases. One VPC is connected to a Direct Connect gateway and a transit gateway, while the other VPC is not. The decision to connect one VPC to the transit gateway and not the other was made due to limitations and concerns around routing and the number of connections being utilized. Traffic originates from various sources, such as developer laptops and other applications, and flows through the infrastructure to hit a Direct Connect. Routing logic is then applied based on addressing, with traffic destined for services like SQS, SNS, and S3 flowing into the AWS environment. However, the applications do not directly talk to the Aurora database and developers and support staff must follow a specific network path to access it. Additionally, traffic from the OpenShift Kubernetes cluster flows back into the Direct Connect environment, where it is inspected by a firewall device for Data Loss Prevention (DLP) controls and then either sent to other AWS services or connected to the Aurora database in the VPC.*

1. ***Limited connectivity to the transit gateway****:* 
   * *By connecting only one VPC to the transit gateway and not the other, the VPC that is not connected may have less visibility and control over the traffic flowing through the gateway. This could potentially create a security vulnerability if the VPC that is not connected has sensitive data or systems that need to be protected.*
2. ***Lack of direct access control to Aurora database****:* 
   * *The setup described states that the applications do not directly talk to the Aurora database, which means that developers and support staff must access it through a specific network path. This could create a potential vulnerability if the network path is not properly secured or if there is not enough access control in place for the database.*
3. ***Limited DLP controls****:* 
   * *The firewall device that inspects traffic for DLP controls is only in place for traffic flowing back into the Direct Connect environment. This means that traffic flowing from the OpenShift Kubernetes cluster and into the AWS environment may not be inspected for DLP controls. This could create a potential vulnerability for sensitive data that may be transmitted through this path.*
4. ***Limited visibility on the traffic****:* 
   * *The setup mentions that the traffic flows through the infrastructure and hit a Direct Connect, but it does not mention if there is any monitoring or visibility in place for the traffic. This means that it may be difficult to detect any malicious traffic or anomalies.*
5. ***Limited control over the routing****:* 
   * *The setup mentions that routing logic is applied based on addressing, but it does not mention if there is any control in place over the routing. This means that there may be potential vulnerabilities if the routing logic is not properly configured or if there is not enough visibility into the routing.*

*Questions:*

* 1. *Can you provide more details about the routing logic that is applied based on addressing in the setup? Are there any specific routing rules or configurations in place?*
  2. *Are there any access controls in place for the Aurora database? How are developers and support staff able to access the database, and what level of access do they have?*
  3. *How is traffic flowing into and out of the VPCs that are not connected to the transit gateway controlled and secured?*
  4. *Are there any monitoring or visibility tools in place for the traffic flowing through the infrastructure? How is malicious or anomalous traffic detected?*
  5. *What are the specific reasons why one VPC was connected to the transit gateway and the other was not? Are there any limitations or concerns that still exist with this setup?*
  6. *What type of data is transmitted through the OpenShift Kubernetes cluster, and what type of DLP controls are in place for this traffic?*
  7. *How are the Direct Connect gateway and the transit gateway configured and connected to the VPCs and other services? Are there any specific configurations or protocols in place for these connections?*
  8. *Are there any backups or disaster recovery plans in place for the Aurora database and other critical systems?*
  9. *Are there any compliance or regulatory requirements that need to be met with this setup? How are these requirements being addressed?*
  10. *Are there any other systems or services that are connected to this setup that were not mentioned in the description? Can you provide more details about these systems and how they are connected to the overall environment?*

So then, for the progressive folks who are on the call that are responsible for infrastructure as code they do not have control of most of the purple elements is that is the progressive cloud networking and security teams is that correct?

That is correct. So this superstructures generally, I wouldn't say that don't have access, but if any changes need to be made to this I do consult with the Cloud team as well before making any changes yet but our TerraForm activities are primarily sort of related to this bottom left hand corner.

Thank you. Yep, that was point of clarification for me as well. So thanks.

A lot. So that was the overview of the connectivity and another meeting bacterial.

So I guess I could take over here a little bit, but I'll put it into RSM what you're hearing here in the diagram you just saw and in the overview stuff I was doing is to follow the transactions from claims and agency through these diagrams, so that we can confirm all the best practices and all the security is executed in a way that we can protect progressive from risk.

Yeah, that is helpful on our side, John, so I don't know who was sharing diagram, but if the person that we're sharing it can reopen it again, because I think I'm trying to understand the different I guess building blocks in here right and who controls what

Okay, Angela? could pick your mind.

So on the right side of this diagram, you have the developers tools right and appreciate Angelo, and on the top side of this diagram, we have the systems right like the PR applications, basically and then on the lower side, we have, I guess, the developers themselves. Let me start on the development on the development side. I think someone mentioned TerraForm and the type of restrictions that someone using TerraForm would would have. So can you guide us through that whole process of the development side? I mean, what are the developers able to do with TerraForm? And at what point it stops, I guess and on which account because that will be again, flowing just from a developer's point of view that will be one of the view that I would like to get and then the second one would be on the system side, really flowing the other way around, right? Like the connectivity from that upper level in your diagram, all the way to the cloud, and then we can drill down into the account with just trying to understand what happens on this right side of the diagram and then flow a little bit better. Again, I It's probably something that everybody in this call knows really well. For us. Everything is new. So that's why we're asking these questions, right?

Not a problem. So in terms of TerraForm I think I'll hand it over to chuck and Peter who are really are the two developers who do all the TerraForm work. So just to sort of clarify the we have restricted, I guess, is the word I would use or limited the number of developers who actually do the TerraForm or to just a small set of individuals who focus on AWS infrastructure. The rest of the developers who do meal soft development, etc, doesn't actually touch any of the TerraForm code. They just issue requests to chuck and all Pina for infrastructure requirements for their code. So Chuck computer you guys can expand a little bit more how it starts from your sort of desktop here until the infrastructure is eventually eventually provisioned.

And I just want to add that, you know, Chuck and Peter are are provisioned to make certain TerraForm changes that are specific to the project. But then you've heard mentioned in the Cloud team, the call team has further capabilities to do TerraForm for things that are more enterprise grade infrastructure.

Okay, that makes sense.

Thanks. So this is Chuck as far as what we do with the TerraForm. It will be any change in any infrastructure that's created within our AWS account is other than in lab is encrypted with TerraForm. So developers have the ability to create AWS objects in the lab. Account. Any higher environment, all infrastructure changes are scripted in TerraForm. That TerraForm infrastructures code is housed in a GitHub repository. We have change control processes in place for that. Any pull requests to our stress environment or prod environment goes through different levels of approval processes, and those pull requests before that code can get elevated. As part of that, change control process. We go through code review and have code owner sign off which would be a QA person, as well as additional approver sign off, which would be your developers, infrastructures, code, peer developers on the development team, as far as what we can do with that TerraForm we can create just about any AWS object with the TerraForm but that is strictly limited to items within our account. I'll give you an example of one thing we don't have access to do is we can create a VPC within our account whether this VPC has connection to our direct connect access link, in that that requires connectivity on both sides, both within our AWS account plus the Direct Connect link itself. The Cloud team would create that link to our VPC. And when it comes, another thing we have is with our V PCs, in the in the traffic going back between AWS and on prem, and everything, obviously goes to firewall. So all that firewall setup is handled by another security team within progressive.

Okay, and you were explaining on the GitHub site, the the approval process that you have, and you mentioned that it was through a peer review. And someone else in your

idea we have code owners, which would be quality control. So every every item is QA before it gets elevated the higher environment. So we'll do our development Dev. It'll get tested in dev and QA. And then we can go through the development, l&d. process to get the prod.

Just a point of order here are like a dialect thing. I don't know why but progressive people our whole culture. We don't deploy code to production. We elevate it because it's holy. I don't know why hire environments. You don't even hear it anymore. So we say elevated just when we're talking about code deployment.

Okay. And then, I guess the elevation of code here that can you explain who's I guess involved in that? approval and, and review process? Is it people from your team, are there anybody else that looks at this code from the Cloud team as well?

This is this is people from the project team.

Okay, that makes sense. And just for the RSM folks on the call. So each of these environments Dev, QA, stress, prod, are separate individual AWS accounts. Yeah.

Okay, do they do they live on in an organization or they're managed by control tower? How does that work? I mean, you're saying separate AWS accounts, but I mean, are there separate completely or do they live in AWS organizations of some sort?

That's probably a question for the security slash cloud teams. I don't think anybody on this call has got that level of visibility. into how they manage them provisioned. I would assume it's an organization but I would want one of those teams to confirm that.

Yeah. Because what we have an organization ID we can use and that will give access anybody with AWS progressive account to access your account on okay. So, that's the other piece of all our TerraForm they call it the core piece. And so our Cloud team has the core tip that they set up and they have a lot of templates to utilize to set things up out of the gate. So that gives them that authority and then it's all tied back to one organization. Okay.

Then your, your other question was around the source applications if we're done with the developers or the TerraForm if we've handled all your questions,

well, so I'm just looking at the diagram real quick. I guess trying to identify where I'm missing anything. I mean, I'm sure that I'm gonna have follow up questions but I think for for the time being, I'm okay on the TerraForm side on this, someone else on the RSM team has additional questions from a good common TerraForm perspective.

*The diagram being discussed shows the different components of a development environment, including the developers tools, systems, and the developers themselves. The discussion focuses on the development side of the environment, specifically the use of TerraForm for managing infrastructure changes within the AWS account. There are a limited number of developers who are specifically trained and authorized to work with TerraForm, and the rest of the developers issue requests to these individuals for infrastructure requirements. The TerraForm code is stored in a GitHub repository and there are change control processes in place for pull requests to the production environment. These pull requests go through a detailed review process, which includes code review, QA sign-off, and additional approval from other developers who have the role of code owners. The developers are able to create most of the AWS objects within the account, but some actions like creating a VPC with connection to the Direct Connect access link, require the involvement of the Cloud team, that has further capabilities to do TerraForm for things that are more enterprise-grade infrastructure. Additionally, all traffic going back between AWS and on-prem goes to a firewall which is handled by another security team within the organization to ensure the security of the data.*

1. ***Limited access to Terraform***
   * *By limiting the number of developers who can access Terraform, the organization may not have enough resources to manage and make changes to the infrastructure in a timely manner. Additionally, if the developers who have access to Terraform lack the necessary security knowledge, this could lead to misconfigurations or other security issues.*
2. ***Limited control over infrastructure changes***
   * *The setup describes a change control process for Terraform changes, but it does not mention if there are any controls in place to prevent unauthorized changes or misconfigurations. This could create a potential vulnerability if the change control process is not properly enforced.*
3. ***Limited control over VPC and Direct Connect***
   * *The setup mentions that the developers are not able to create VPC with connection to the Direct Connect access link, this could create a dependency on the Cloud team and delay the provisioning of resources. Additionally, this could create a potential vulnerability if the Cloud team is not properly trained or does not have enough resources to manage the VPC and Direct Connect access link*
4. ***Limited visibility into the traffic***
   * *The setup mentions that all traffic going back between AWS and on-prem goes to a firewall, but it does not mention if there is any monitoring or visibility in place for the traffic. This means that it may be difficult to detect any malicious traffic or anomalies.*
5. ***Limited control over firewall****:* 
   * *The setup mentions that the firewall is handled by another security team within the organization, but it does not mention if there are any controls in place to ensure the security of the firewall or if there is enough visibility into the firewall. This could create a potential vulnerability if the firewall is not properly configured or managed.*

***Questions***

* 1. *Can you provide more details about the change control process for TerraForm changes? Are there any specific controls in place to prevent unauthorized changes or misconfigurations?*
  2. *Are there any additional controls in place to ensure the security of the TerraForm code stored in the GitHub repository?*
  3. *How are the developers who are authorized to work with TerraForm selected and trained? Are there any specific qualifications or certifications required?*
  4. *Are there any limitations or restrictions on what the developers can create or manage with TerraForm? Are there any particular AWS objects or services that are off-limits?*
  5. *How is the VPC and Direct Connect access link created, and who is responsible for managing it? Are there any limitations or restrictions on what can be done with the VPC and Direct Connect access link?*
  6. *How is the firewall managed and configured? Is there any monitoring or visibility in place for the traffic?*
  7. *Are there any disaster recovery or backup plans in place for the infrastructure managed with TerraForm?*
  8. *Are there any compliance or regulatory requirements that need to be met with this setup? How are these requirements being addressed?*
  9. *Are there any additional security measures in place to protect the infrastructure managed with TerraForm, such as encryption or access controls?*
  10. *Are there any other systems or services that are connected to this setup that were not mentioned in the description? Can you provide more details about these systems and how they are connected to the overall environment?*

Now, we don't have questions on that. side, but we will need some explanation or documentation on how their VPC and the networking is set up just so we have a better understanding on the traffic flowing in that specific way. So I'm not sure who we who we can reach out for that once all the informations.

Well, so like I was, this is John again, going left to right in the diagram there stuff on prem that we won't touch right. And then there's stuff that after it's processed here on the outbound calls, leave this environment and make it to workday. financials were contractually forbidden from poking at their software as a service. So what we're really trying to do here is highlight on this diagram, and I think every box and every line is an interesting point of inspection. I just don't know where the RSM team thinks it's interesting. So what I was going to propose, but I'm open to suggestions based on the RSM methodology. Do you want a PDF of this page that we send to you that you mark up and then send it back and then you could index this with numbers or letters or something and say for this box are for this arrow, we would like to see x or we would like to see y or please describe the documentation that does such and such. And then we could use these a diagram as an index for a punch list until we've inspected all the things that merit inspection.

Yeah, so that was one of the questions that I was asking the team while we were in this call in some of these conditions had been provided to us so we can do I mean, something similar to what you just suggested, John, because obviously, I mean, having that many people in the call and asking different questions across I mean, might be beneficial for a niche, an initial conversation, but I think we're gonna have to dissect each one of these, I guess blocks and try to identify who are the people that we need to bring in to the conversation to ask specific questions and drill down. So I would totally welcome that. That approach, right? I mean, just do providing some of these recommendations in PDF format if that is what feels comfortable on the progressive side. And then on our side, we can basically start dissecting which are the specific components of the solution that we want to focus on and then you can tell us you know what, this is the team that you need to be talking to, based on the questions that you have and the specific side of diagram that you knew selected or the boxes that you've selected. With that said, though, I mean, I would welcome the explanation on the system side and that that flow that I think Angela was going to go through just to have a general high level overview of the solution from that perspective. And then I think from that before, we're probably going to take the rest of the 15 minutes that we have here. And after that, we can take the PDF information and other information that progressive can provide to us and then we'll we'll do exactly what I just described, right? I mean, we'll start putting together questions per different components in the diagram and reach out to you and others and Scott in the team just to identify who's the best person to talk to if we want to drill down into his specific components of the of the solution. So I guess Angelo going back to you, if you don't mind, if we can go to the opposite side of the diagram and maybe just rattle down.

This, as I said, these kind of little blocks are just the plumbing. They're just a network that it flows over from a data flow perspective. claims and the agency system, make outbound HTTP REST API calls to a mule soft API. Okay, so all of that traffic is initiated by making edits, HTTP calls to a new API, primarily speaking, alright, that's, that's where the kind of whole process initiates when they need a payment to be processed on their behalf. Within our mule soft sort of environment. Now we take those messages and we store them into queues for reliability for asynchronous processing. So from your it kind of goes into a bunch of SQS queues. We then have other mule applications that listen to those queues, pick them up for processing Brian mentioned earlier, those of your applications might do some validations. John also mentioned so they will reach out into this Aurora Postgres SQL database for some information that will store some auditing information that may store some information in s3 Samuel is very chatty, with AWS services, I can put it there and there's a lot of interaction between the mule soft logic and other AWS services including storage s3 queues, SNS topics, the Aurora database once everything is validated, everything is in order from an yourself application. The data will flow back to the progressive network, leave through the outbound connectivity and get posted into workday. As John said, workday is its own kind of cloud. We consume it as a service. It's a black box, we don't see what happens behind those API's. But we interact with the out of the box API's. Right. Brian mentioned that there are some integrations in between workday, banks and other such providers right so their, their processes happen on an asynchronous basis could be overnight events happen eventually, responses come back from the bank. And certain events happen in most cases, workday will generate an extract file that will be blown that will be put in an s3 bucket, which yourself will then pick up and process so whether it is confirmations of payments, rejections of payments reconciliations and researches a bunch of different processes that run but largely speaking from workday, we get a bunch of extracts in s3. The one exception to that is when we need to bring documents. workday also then makes an HTTP API call into one of our mule soft API's with all of the data that's needed to print a document whether it's a draft or a check. In that instance, I worked on makes an HTTP call to mule soft, we save that data in the Aurora database and then interact with the print services that are actually on premises in the progressive network.

*The text describes a system for processing payments using a network of "little blocks" or plumbing. This network allows for outbound HTTP REST API calls to be made to a MuleSoft API in order to initiate the payment process. These calls are then stored in SQS queues for reliability and asynchronous processing. Other Mule applications listen to these queues and pick up the messages for processing, where they may perform validations and reach out to an Aurora Postgres SQL database for information. The MuleSoft logic also interacts with various AWS services, including storage in S3, queues, SNS topics, and the Aurora database. Once everything has been validated, the data flows back to the progressive network and is posted into Workday, a cloud-based service that functions as a black box. Workday may also interact with banks and other providers, and generate extract files that are placed in an S3 bucket for processing. The exception to this is when documents need to be printed, in which case Workday makes an HTTP API call to a MuleSoft API, the data is saved in the Aurora database, and the print services on the progressive network are used.*

*Issues:*

1. *Unauthorized access to the SQS queues and Mule applications.*
2. *Injection attacks on the Aurora Postgres SQL database.*
3. *Insecure data storage in S3 buckets.*
4. *Unauthorized access to the print services on the progressive network.*
5. *Lack of encryption for data in transit and at rest.*
6. *Lack of proper authentication and access controls for the REST API.*
7. *Lack of monitoring and logging for the system.*
8. *Unsecured interaction with external providers, such as banks.*
9. *Lack of proper incident response and disaster recovery plans.*
10. *Lack of regular security testing and vulnerability assessments.*

*Unauthorized access to the SQS queues and Mule applications:*

*If proper access controls are not in place, an attacker could potentially gain access to the SQS queues and Mule applications, potentially accessing sensitive data or altering the flow of payment processing.*

*A lack of monitoring and logging in this area could make it difficult to detect unauthorized access and respond to any security incidents.*

*If encryption is not used, an attacker could potentially intercept and read sensitive data being stored or processed by the SQS queues and Mule applications.*

*Injection attacks on the Aurora Postgres SQL database:*

*If the database is not properly configured, an attacker could potentially exploit SQL injection vulnerabilities to gain unauthorized access to the data stored in the database.*

*This could allow an attacker to exfiltrate sensitive data or make changes to the data stored in the database, potentially impacting the payment processing system.*

*A lack of monitoring and logging in this area could make it difficult to detect an injection attack and respond to any security incidents.*

*Insecure data storage in S3 buckets:*

*If proper security controls are not in place for the S3 buckets, an attacker could potentially gain unauthorized access to the data stored in the buckets.*

*This could include sensitive data such as payment information or documents that need to be printed.*

*A lack of encryption for data stored in the S3 buckets could make it vulnerable to data breaches or data exfiltration.*

*Unauthorized access to the print services on the progressive network:*

*If proper access controls are not in place, an attacker could potentially gain access to the print services and potentially intercept or alter sensitive documents.*

*A lack of monitoring and logging in this area could make it difficult to detect unauthorized access and respond to any security incidents.*

*An attacker could take advantage of this to steal sensitive information, disrupt the printing process, or even launch a social engineering attack using the printed documents.*

*Lack of encryption for data in transit and at rest:*

*If data is not properly encrypted, an attacker could potentially intercept and read sensitive data as it is being transmitted or stored.*

*This could include payment information, personal identifiable information, and other sensitive data.*

*A lack of encryption could also make it difficult to comply with data protection regulations and standards.*

*Lack of proper authentication and access controls for the REST API:*

*If proper authentication and access controls are not in place, an attacker could potentially gain unauthorized access to the REST API and potentially disrupt the payment processing system.*

*A lack of monitoring and logging in this area could make it difficult to detect unauthorized access and respond to any security incidents.*

*An attacker could take advantage of this to steal sensitive information, disrupt the payment process, or launch a DDoS attack.*

*Lack of monitoring and logging for the system:*

*Without proper monitoring and logging, it could be difficult to detect and respond to security incidents, such as unauthorized access or data breaches.*

*It could also make it difficult to comply with regulatory requirements for security and data protection.*

*A lack of logging could make it hard for incident response teams to understand the scope and nature of a security incident, and to identify the root cause.*

*Unsecured interaction with external providers, such as banks: -*

*If proper security controls are not in place for interactions with external providers, an attacker could potentially intercept or alter sensitive information being exchanged. –*

*This could include payment information, personal identifiable information, and other sensitive data. - An attacker could also potentially disrupt the payment process by intercepting or altering communication between the system and external providers.*

***Questions***

* 1. *Can you provide details on the access controls and authentication methods in place for the SQS queues and Mule applications?*
  2. *How is the Aurora Postgres SQL database configured to protect against injection attacks?*
  3. *What security controls are in place for data stored in S3 buckets, including encryption and access controls?*
  4. *How are the print services on the progressive network protected against unauthorized access?*
  5. *Can you provide details on the encryption methods used for data in transit and at rest within the system?*
  6. *How is access controlled for the REST API and what authentication methods are used?*
  7. *Can you provide details on the monitoring and logging in place for the system?*
  8. *How are interactions with external providers, such as banks, secured to protect sensitive information?*
  9. *Can you provide details on the incident response and disaster recovery plans in place for the system?*
  10. *How often are security testing and vulnerability assessments conducted on the system and what methodologies are used?*

Okay,

then there's a bunch of daily sort of reconciliation jobs that once again, largely run through s3. So either within yourself or within view, we've got a bunch of jobs that extract data to produce reports that are stored in s3, which client and or agency then ultimately connect to perform some of the reconciliations. On a per transaction basis, we send the individual transaction events back on the simple notification queues that they subscribe on the support notification topics. That they subscribe to and receive the events. So there's kind of two different sets of responses that they get from us. One is per transaction, event messages, and then they get aggregate kind of daily balancing reports that they're also access. So these applications sent primarily HTTP requests. They receive responses back through SNS topics, which says that they subscribe to and they also receive data back in specific s3 bucket which is more of a bulk ledger.

So many, I mean, you don't have this answer, but I mean, is there any documentation of all the API's I just mentioned, or flow? I mean, you mentioned that some of the applications do API calls REST API calls to to initiate the mule soft

swagger interface and swagger definitions for all of the API's. That the more sort of functional requirements are all in the project documentation in a do as your DevOps where all of the stories were built. While we were developing the project to say this is the required functionality. But each and every API that is deployed has gotten a Swagger definition of all of the resources that it exposes.

So that will be also helpful for us to inspect before we reach the next session, right because understanding the API's and how they are designed and used would be important for us to really have an intelligent conversation here right. And then the third Go ahead, sir. I think John, you're going to jump in.

Oh, yeah, I was just wanted to pop off mute before the call finishes your lose folks. For both in progressive in the RSM teams here, just to set some understanding. One is we're talking about some pretty sensitive things for a payment system. So under no circumstances should we be sending stuff back and forth using public Internet email because God knows where those attachments will go. So we expect and haven't ironed out entirely how we're going to do secure document sharing. However, for the progressive folks on the call in order to accelerate the RSM team's engagement, we did not onboard them in the classic professional services sense this is expected to be a short engagement. So they don't have a badge. They don't have a progressive email. They're not in our intranet. We didn't give them a virtual machine. They're continuing to be guests on teams calls like this. So let's say we wanted to share this document as a PDF. We're probably going to only name one or two people who will securely share information because we want to scrub some of these things too. And on our side, send info from us to RSM with a secure document shared way. We talked about that and a warm up call last week before we launched here with Carrie and some other folks. I'm not sure we've completely ironed that out yet. There's alternate ways to do that. We've had vendor posted SharePoint sites before. Nancy North campers InfoSec generalist who joined our call a little bit later, I want to follow up with her in a few minutes after this call before we start sharing, but I like the direction of the conversation. And we just started talking about eto and some of the project documentation. Please know that the RSM team can't snoop around our radio tenant so if we have to provide them specific things, we're going to have to do it in a kind of more formal career sort of way. That makes sense to everybody. That's kind of a constraint here with the engagement so if I miss something, I'm sorry, David, again, for either David or carry on a bit in the RSM team. Have we, for this engagement, decided how we're going to do that documentary so we can get you the things you were about to ask for.

Yep. So I know that we did send out the are helped us team sent out the wasn't my train of thought this the site request for the team Doc site. We sent that to Brian and Scott. Do you need me to resend those

memos Brian got to you otherwise

I apologize. I don't recall receiving that.

Okay, and may have hit your spam folder. But let me go ahead and reshare it after this call. I apologize. Brian, my service desk told me you guys were fine.

Please include John Krycek on anything

Okay. Yep. Well,

thank you. I was just gonna say I haven't seen anything but then again, you may have heard my spam. I've haven't seen it

yet. Okay. I will reshare that right after this call. Thank you.

Do you happen to know or other RSM engagement with progressive those other teams before us are ready to establish a document sharing methodology besides the one you talked about her?

Know that all those other teams are just using our SharePoint or team DocuSign as they call it?

Well, I think that's the document sharing with established I guess, yeah, from our perspective every single time that we have shared information with the permanency team, it has been by using this method right as I mentioned before, we can provide other methods if there's any concerns on the methods that we have used previously. We

Potential flaws

1. Insecure data storage in S3 buckets
2. Lack of encryption for data in transit and at rest
3. Lack of proper access controls for SNS topics
4. Lack of proper authentication for accessing the reconciliation reports
5. Lack of monitoring and logging for the system
6. Unsecured document sharing

Questions:

1. Can you provide details on the security controls in place for data stored in S3 buckets, including encryption and access controls?
2. Can you provide details on the encryption methods used for data in transit and at rest within the system?
3. How are access controls implemented for the SNS topics and what authentication methods are used?
4. How is access controlled for the reconciliation reports, and what authentication methods are used?
5. Can you provide details on the monitoring and logging in place for the system?
6. How is sensitive information shared securely with external parties, such as the RSM team?