ZERO TRUST

Modeling a Zero Trust Initiative

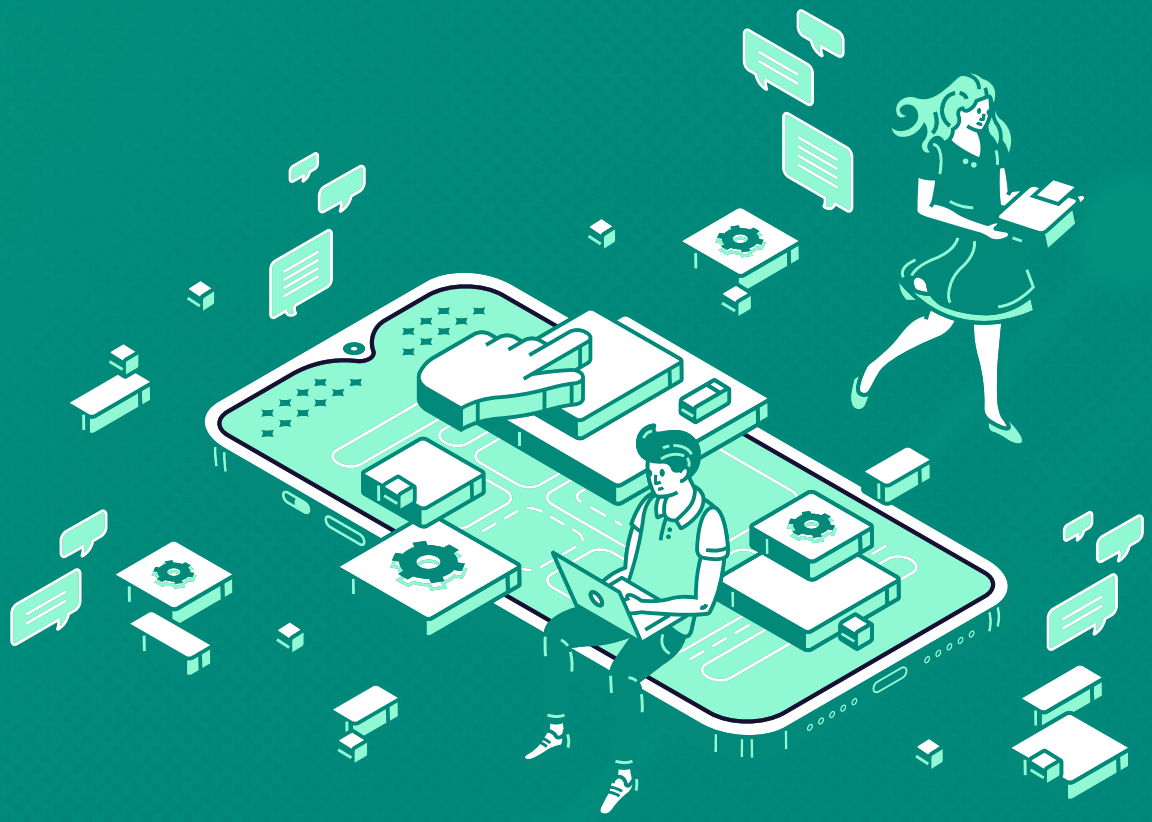
An unbiased guide for IT architects who are starting or supporting a Zero Trust initiative.

Andy Ruth, Managing Director
Jaylene Ruth, Director of Operations
Andrew Wilt, Chief Creative Officer

SUSTAINABLE **EVOLUTION** | JULY **2023**

Start Here

This eBook is written for IT architects who are starting or supporting a Zero Trust initiative. The guidance is framed on architectural best practice and process used for any major IT initiative. This eBook covers the key areas you'll consider when modeling a Zero Trust initiative.



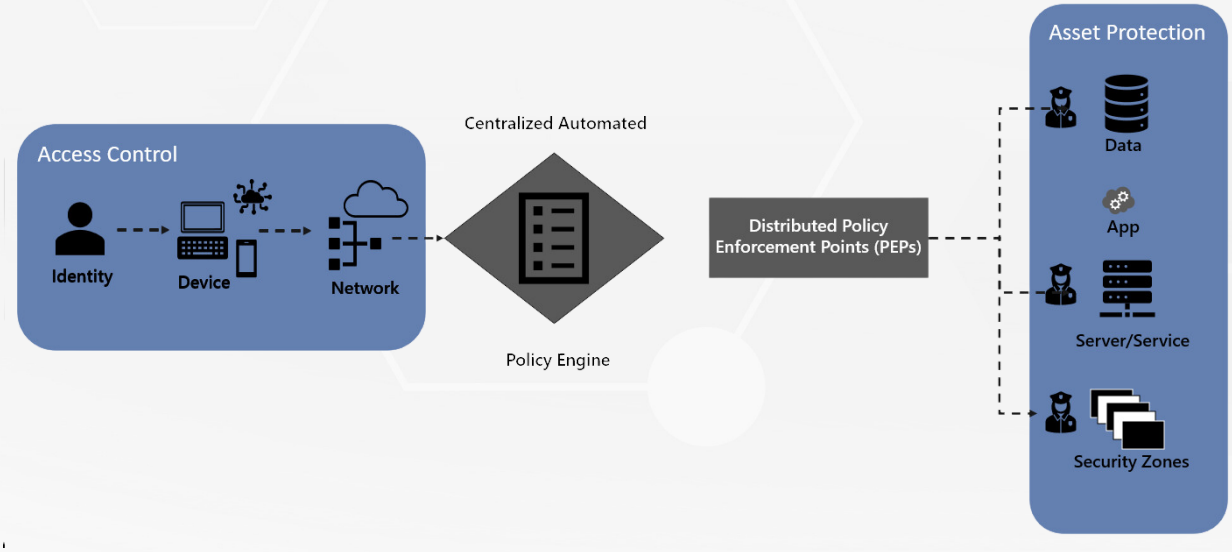
Visit us at sustainableevolution.com

 | **Modeling a Zero Trust Initiative**

TABLE OF CONTENTS

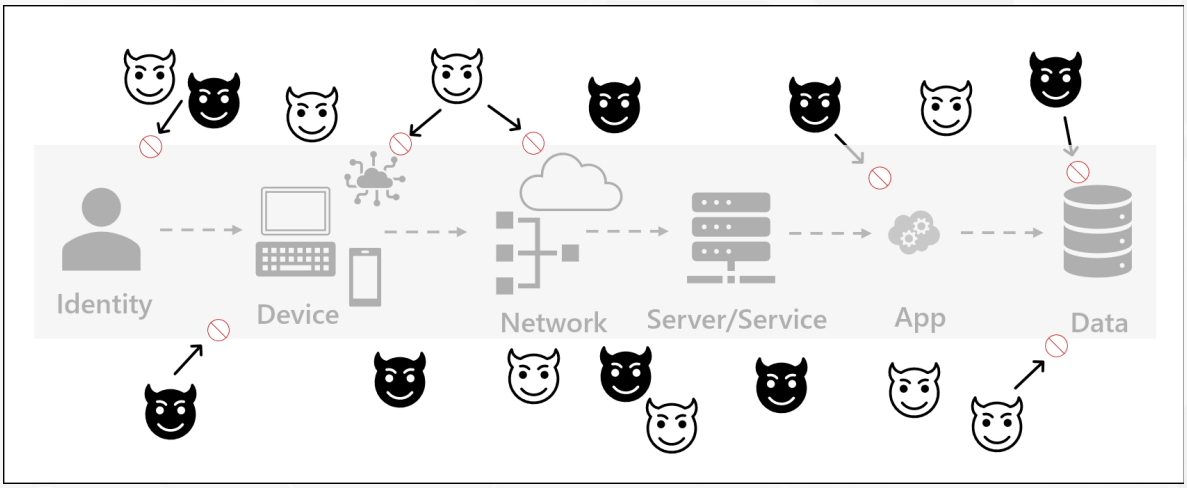
Modeling Goals

A Zero Trust strategy describes the transformation from a network-centric, implicit trust model to a data-centric explicit trust model that is Internet and digital economy ready. A Zero Trust strategy requires all IT assets to be cataloged with a defined business value. The goal is to design access control and asset protection so that automated rules enforcement can be defined and implemented. The asset catalog with business value data enables operations to prioritize detection, response, and recovery efforts:



A large part of the effort is defining states or context for each pillar in the data request/fulfillment journey, set policy based on pillar segmentation, and enforce, monitor, and respond to anomalies to the governance policies through automation.

The strategy must enable the data fulfillment journey to be completed while stopping bad actors from completing or disrupting the journey:

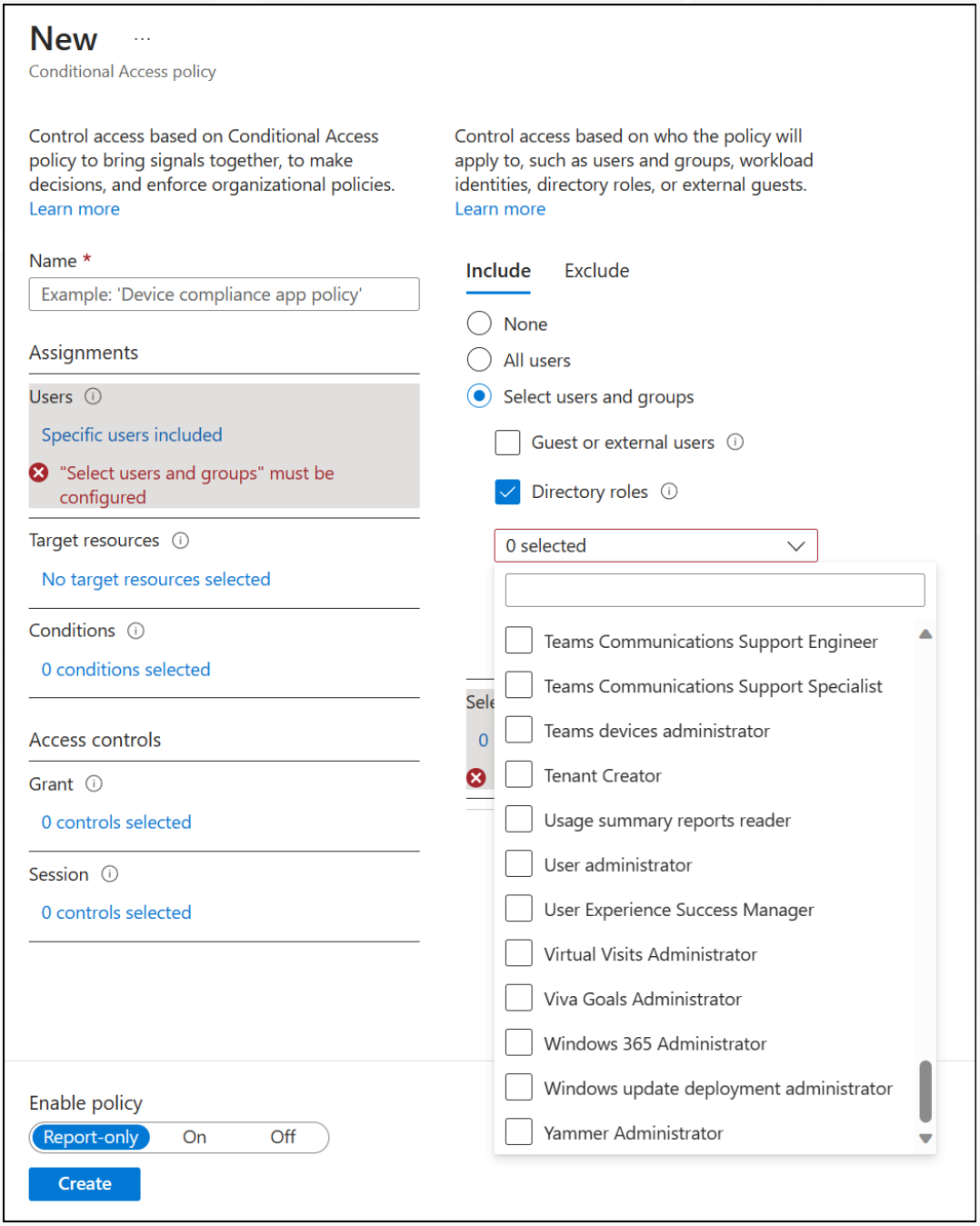


To build and operationalize your model, you must have a complete inventory of all IT assets. The inventory must include a business value for each asset, as well as a risk level if the device is compromised. The inventory doesn't need to be complete prior to modeling but must be complete to fully operationalize your strategy.

For each pillar, the goal is to categorize or group the IT assets in the pillar in a way that is relevant for governance rule enforcement. For example, you might want a governance rule that requires employees to use multifactor authentication to connect to a human resources (HR) system in your environment. And then, if the device they are using is managed by your company they can view and edit more of their employee data than if they are on a personal device. Perhaps on their personal device, they can only see the number of vacation days they have and submit a timecard for recording vacation. If they use an encrypted network link, they can see and do more than with an unencrypted network connection.

You've likely gone through similar segmentation exercises so might be ready to start immediately. However, if you have limited experience

with this type of exercise, consider reverse engineering your segmentation by reviewing security policy tools to get an idea of what type of governance tools support. For example, if your internal productivity environment is primarily Google-based, you can view Context-Aware Access to get an idea of what is possible. If you are using the Microsoft stack, you can review Microsoft Conditional Access documentation to identify what context and controls the tool can be configured to enforce, or you can open the tool in an Azure portal and explore settings that way:



You use the attributes or metadata of the IT assets to segment or group the assets in each pillar and then create a set of policies that are automatically enforced. One point of modeling is to make sure you have segmented each pillar so that you can create impactful governance.

We recommend that you take an iterative approach to modeling the pillars and defining governance for your Zero Trust strategy. The act of cataloging your IT assets for each pillar will provide insight into the governance you want to consider putting in place. For example, as you start cataloging identities, you'll determine you likely want different governance enforced on administrative accounts than on end users or customers. The act of creating security policies for your environment will, in turn, inform the segmentation as the team continues the modeling exercise. For example, as you start modeling your governance, you'll likely want to create different policies for internal users (employees), partners, and temporary workers. The segmentation you want, and metadata needed to enforce the governance will inform the identity modeling effort.

Once you have cataloged all your assets and modeled your segmentation for each of the pillars, you'll have to operationalize the strategy. The operationalizing of the strategy includes enforcing the five functions of the cybersecurity framework. NIST describes the five functions as **Identify**, **Protect**, **Detect**, **Respond**, and **Recover**. If you already use a set of tools to provide security Information and Event Management (SIEM), automated security policy enforcement, endpoint detection and response (EDR), extended detection and response (XDR), and managed detection and response (MDR), you can reverse engineer their feature sets and settings to help determine what policies are possible and what segmentation of pillars is useful.

We define the data request and fulfillment journey as "an **identity** uses a **device** to traverse a **network** to reach **infrastructure** (a service or service) running an **application** that provides access to the **data** requested". Each of the points of the journey represents a pillar that can be segmented and automated rules set in place to manage control. The rest of this eBook covers modeling the six pillars. It does not cover operationalizing the Zero Trust strategy other than when referring to the tools you might reverse engineer to help model.

Modeling an Identity Workstream

Keep it simple. As you model your identity landscape you might be tempted to define every group of identities that require different rules for data access. Don't. There's more than enough complexity in identity already. Some keys to modeling this workstream:

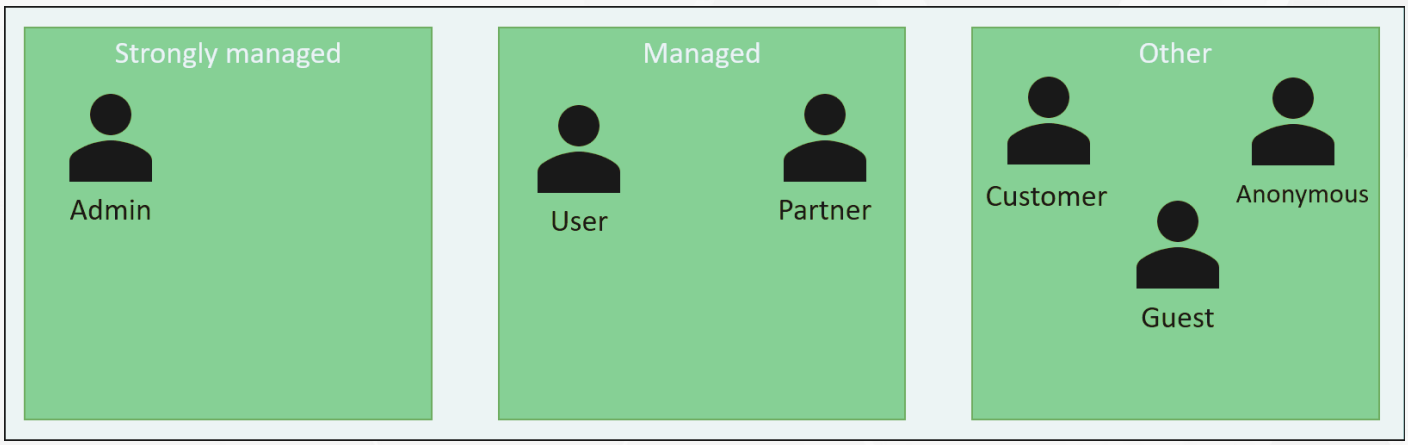
- Determine how many identity and access management (IAM) systems you use and if any consolidation is possible.
- Decide if one (and which one) policy engine can be used across the board, or if IAM consolidation or limited scope of identity types can get you down to one engine.
- Baseline your model with five or six identity groups. Think admins, users, partners, guests, customers, and anonymous.

While you might be tempted to start with the inventory tasks in the bullet list and go top-to-bottom, we recommend you go from the last bullet and work up. By modeling your identity groupings, it might be easier to think in a greenfield manner to determine what target state you'd like to achieve. This might make it easier to make keep/kill decisions for whole IAM solutions as some may fall into the legacy category and not be needed anymore.

To start your model, consider creating three boxes:

- Strongly managed identities
- Managed identities
- Other identities

Place the six identities listed above into the three boxes. Use this as a starting point to decide if you need additional segmentation for identity definition. The results could look like this:



Create a starter set of questions you have as you look at the model:

- From an authentication perspective, are all privileged identities treated the same or do we separate global or service admins from those that have administrative privilege for a collaboration portal or print queue? Do we remove administrative rights if inactive for x number of days or months?
- Do we need to separate partners and vendors?
- When we say guest are we thinking about old guest identities that were used to provide open access to a server or are we talking about accounts that provide access without federation between two organizations?
- If we have multiple tenants in our org will that need to be considered and will it change this model?
- What words will you use for labels? Admin? Privileged, Elevated? Customer? Consumer?

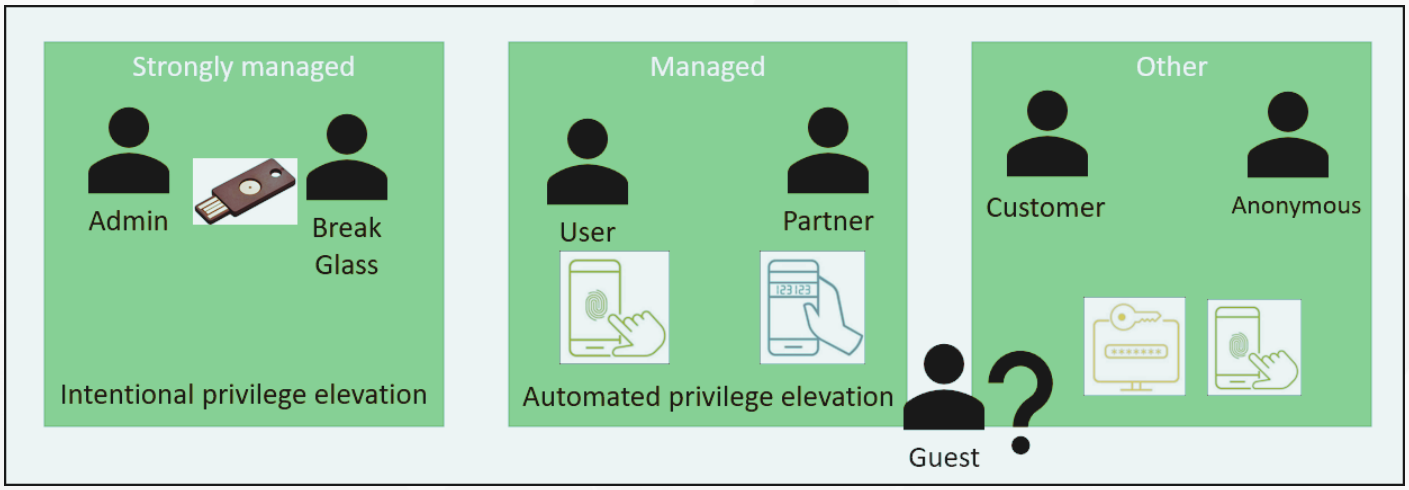
Create a starter set of questions about authentication methods:

- What methods can we use and not use? Authenticator app? Email with PIN? Text or phone call back? Username/password? Passwordless? Anonymous access? Biometrics? Gemalto synced PIN? Yubi key? Yubi with biometrics?
- How long should authentication be good for? Should we and can we set up accounts so that administrative privilege can only be given by account elevation? How long should elevations be good for? Is there an idle period where the elevation is nullified?
- Do we want to use/require Privileged Access Workstations? Do we want to use/require managed devices? Are we good with unmanaged devices?
- What about our break-glass accounts? Will they be managed differently?

The idea behind these first models and questions is to provide a starting point for a modeling exercise. The modeling exercise will likely be broken into three or four exercises, and each exercise might take more than one meeting to complete. The outcome should be a model that you can show to members of your architect network and to your stakeholders.

To start the modeling exercises, gather an argument of architects to have the discussions required to create a model that will work in your organization. Agree to a set of principles up-front. For example, simplicity over elegance, limited end-user friction, and limited loss of productivity. The first exercise can be used to decide if three big boxes (strongly managed, managed, other) are enough and labeled appropriately, and if there are enough or too many identity groupings, and are the correct ones, and placed in the correct box. This is typically a whiteboard session, and the result is a better-refined model.

The second part of the modeling exercise can be used to determine what authentication methods are available for each of the three big boxes, and then assign them to the identity groups. This is also a whiteboard session that results in a more refined model and the addition of authentication mechanisms.



A potential third portion of a modeling exercise might be to start a decision matrix with a high-level list of costs, training required, development/delivery needs, political considerations, and operational considerations:

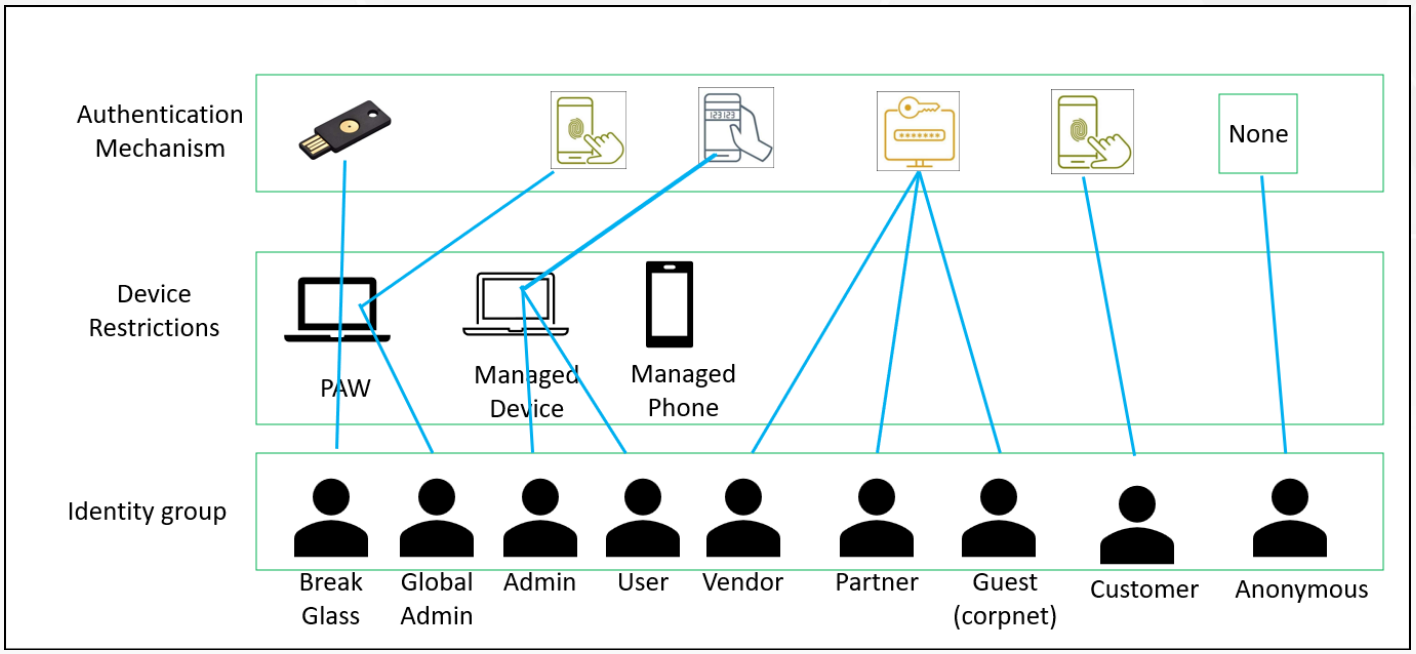
	A	B	C	D	E	F	G	H
1	Consideration Area	Success Probability <50% 50-75% >75%	Cost Low/Medium/High	Technical Hurdle Low/Medium/High	Political Hurdle Low/Medium/High	Operational Hurdle Low/Medium/High	Training Hurdle Low/Medium/High	End User Hurdle Low/Medium/High
2	Elevation for usage of privilege							
3	Authenticator app							
4	Gemalto device							
5	Yubi key (no bio)							
6	Yubi key (bio)							
7	PAW for privilege							
8	Biometrics							
9	Admin							
10	User							

The spreadsheet shows a small sample of the entries you would have. The complete spreadsheet would have all the identity groups and all the mechanisms or methods that would be part of the initiative.

As you start filling out the matrix, we recommend you spend a minimal amount of time researching and use the team's experience to provide best-guess answers. For example:

- The team likely has experience with implementing and enforcing using elevation techniques when using administrative rights. Success probability is high and training hurdle low since you are working with technical staff. The pollical hurdles are likely low since the leadership doesn't really care how their technical staff goes about their day-to-day tasks. This might indicate a high probability of success and minimal hurdles.
- For Yubi key (bio), each unit costs money, has to be distributed, and collected upon termination, will be forgotten at home when traveling, and requires you to carry a thing that may or may not work on every device. This might indicate a low probability of success and significant hurdles.

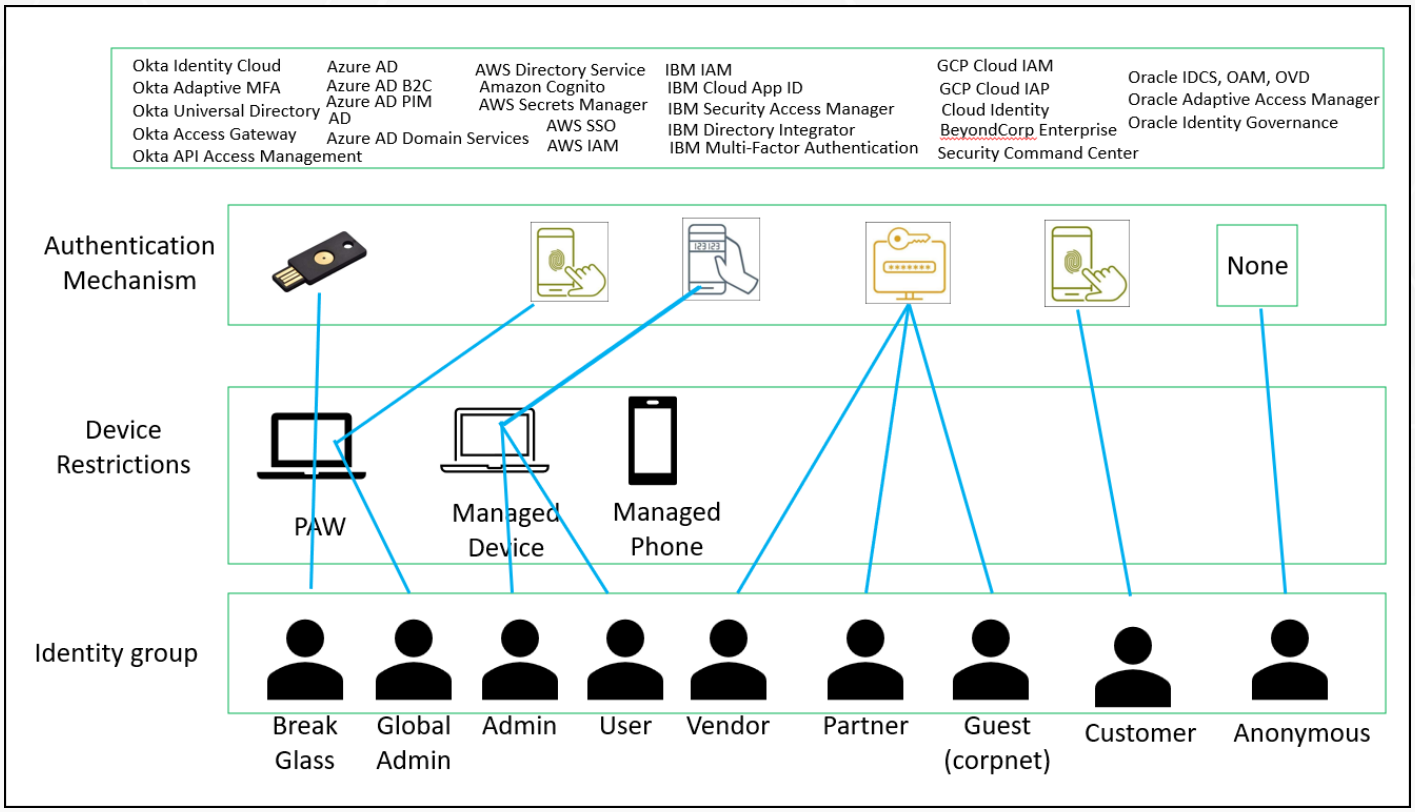
The result of this portion of the modeling effort is a spreadsheet that can be used as a heat map or decision matrix to capture how you made decisions for your recommendation and what the measurement of each is. With each consideration area being separated, you have individual results for each that can later be combined. For example, you might use Yubi (bio) keys for the Admin identity group and based on that scope, the cost comes down and the success probability goes up. The result might be similar to this, but yours would be quite a bit more complex and pretty.



Where there is a dependency on another pillar, such as the device pillar, try to capture the information you have. In this example, we show the Break Glass and Global Admin identities require a PAW for authentication. The device types listed don't need to be complete as devices might be part of this workstream and might be a separate workstream.

We recommend combining identities and devices (endpoints) into one workstream as there is such a strong dependency between them and the team that operates the identity likely has heavy overlap with the team that operates device endpoints.

The final portion of the modeling exercise is identifying each of the IAM mechanisms and authorities in use. You'll want to determine if there is any chance of deprecating one or more or consolidating the tools used. You'll also want to verify that the authentication method you want to use for the identity group is supported by the IAM platform with authority over the identity.



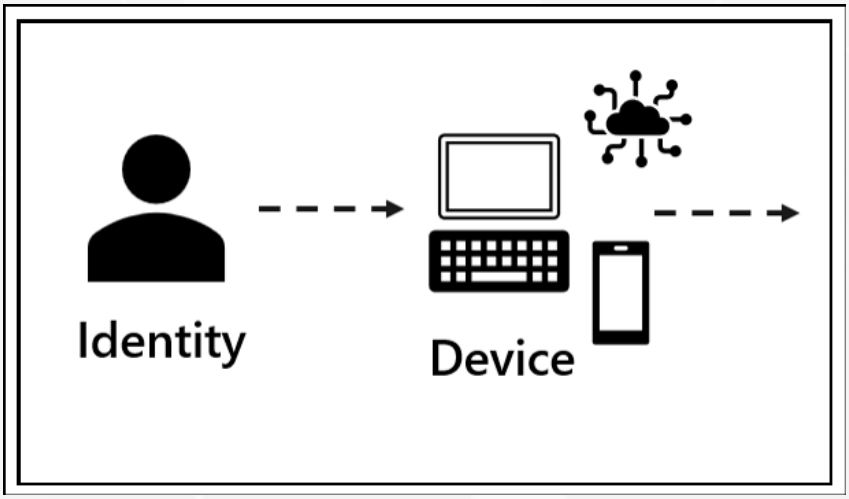
Where there is a dependency on another pillar, such as the device pillar, try to capture the information you have. In this example, we show the Break Glass and Global Admin identities require a PAW for authentication. The device types listed don't need to be complete as devices might be part of this workstream and might be a separate workstream.

We recommend combining identities and devices (endpoints) into one workstream as there is such a strong dependency between them and the team that operates the identity likely has heavy overlap with the team that operates device endpoints.

The final portion of the modeling exercise is identifying each of the IAM mechanisms and authorities in use. You'll want to determine if there is any chance of deprecating one or more or consolidating the tools used. You'll also want to verify that the authentication method you want to use for the identity group is supported by the IAM platform with authority over the identity.

Modeling an Endpoint Workstream

Keep it simple. That'll be the mantra for each workstream you model. There is a ton of complexity so start simple and build up. Start with the essentials for the ideal target and add in as you discover while modeling. As discoveries are made, determine if it fits what you have or if you'll need to adjust your target state.



As mentioned in Modeling an Identity Workstream for Zero Trust Initiatives, you might combine the Identity workstream and the Endpoint workstream. If the teams and people work together already, it makes sense to combine the two workstreams. If not, it might make more sense to keep the workstreams separate initially as the modeling effort might be new to some participants and you need them to be focused and feel comfortable speaking up. With people or teams they don't know, they might be less collaborative. If you keep them separate, we recommend starting with the Identity workstream and then modeling endpoints.

Once both are modeled, bring the two groups together to present their models. The presentation will help build rapport and give the opportunity for all to identify anything they might have missed.

As you model your endpoint landscape, you'll attempt to define every group/type of endpoint in use that might have unique automation, governance/policy, operations, and incident response requirements. Some keys to modeling this workstream:

- Use the identity model as a starting point for this model, update the identity use cases based on discoveries while creating this model, and then combine the two.
- Determine how many control tools are used to manage devices and whether they are automated and effective enough for onboarding, operation, and incident response. In other words, can the operations team and the security operations center (SOC) onboard new devices and successfully identify, protect, detect, respond, and recover from incidents? This will help with weighing the benefit versus the operational costs of your tools portfolio, and whether you have the trained talent to take advantage of the tools.
- Decide if one (and which one) policy engine can be used across the board for governance enforcement and if the operations and SOC team have a clear line-of-sight into potential incidents.
- Baseline your model with five or six endpoint groups. Think laptops, tablets, phones, one-way endpoints (sensors), two-way endpoints (controlled sensors), wired endpoints, and wireless endpoints.

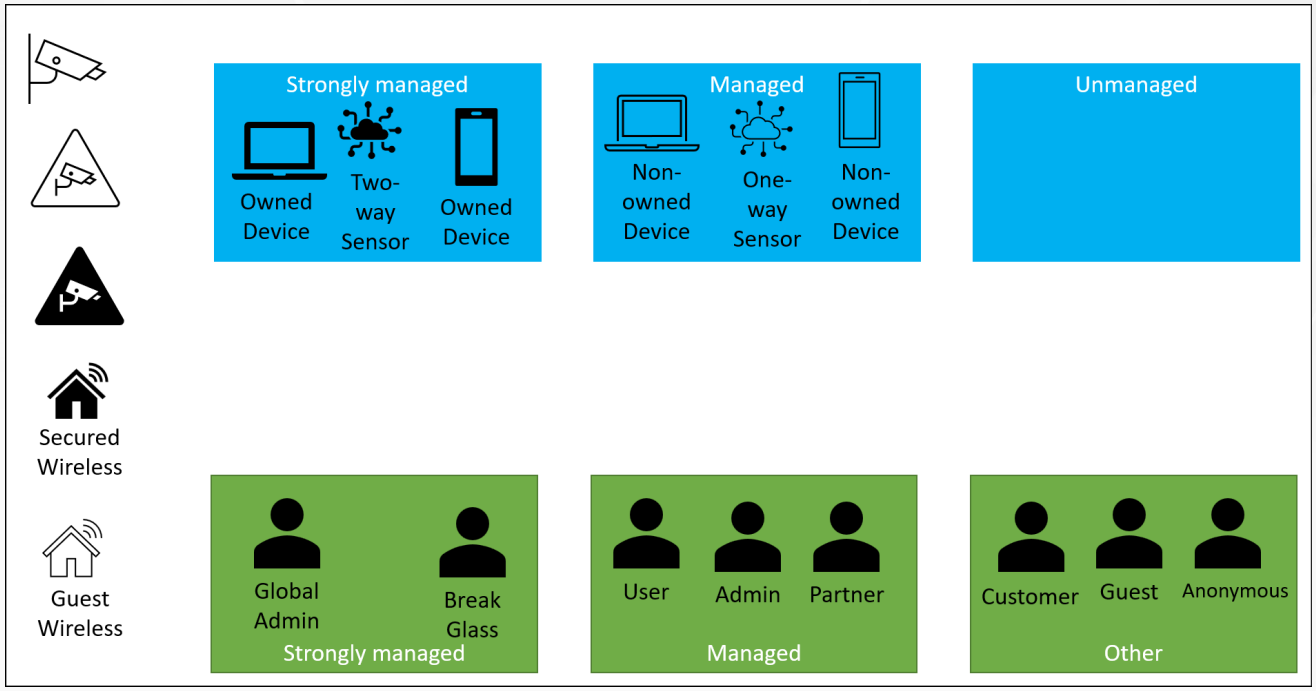
We recommend that you model the endpoint groups first and make sure you cover the use cases to support the identity/endpoint scenarios for your environment. By modeling first, it might be easier to think in a greenfield manner to determine what target state you'd like to achieve. This might also minimize your bias towards a specific set of tools before you know what tools are needed for a capability.

To start your modeling, consider creating three boxes:

- Strongly managed devices – *Typically company-owned endpoints*
- Managed devices – *Typically Individual or third-party-owned endpoints*
- Unmanaged devices – *Typically anonymous and customer-owned endpoints*

To build rapport and spur ideas with the participants, consider leaving some obvious groups of identities or endpoints off and adding additional icons that are not placed. The missing information will enable people to provide input immediately and will build a level of trust between the participants. If you don't, you risk not establishing the level of engagement and focus the effort requires. Having additional icons might spur ideas that get participants thinking.

Since identities use endpoints to interact with the environment, the two workstreams are tightly integrated, though the teams that focus on devices might be different from the people that focus on identities. We recommend that if the workstreams are separate, you combine portions of the identity modeling into your starting model, similar to this:



From the enablement side of the device security and operation coin, the key differences are the level of authentication and access given to the identity using the endpoint and the ownership of the endpoint. From the control and operational side of the coin, it's the amount of control and monitoring available. For example:

- **Strongly managed** - In Microsoft's Azure AD and AD worlds, an endpoint can be joined to the directory service and policies from the directory service used for part of the control solution. That type of endpoint is fully managed by the IT team in the organization. You can limit the software installed and basically **brick** an endpoint if it is lost or the associated identity is terminated. This includes removing all the restore points from before the system was joined and removing the local accounts and administrative privileges from the machine. This is typically for company-owned endpoints.
- **Managed** - For bring-your-own-device (BYOD), partner, and vendor scenarios, the endpoints are still managed, just not as strongly. For example, I might use monitoring tools to validate that the endpoint's operating system is patched, but not verify that all the apps installed on the endpoint are machines and on the internal approved application list.
- **Unmanaged** – For unmanaged devices, your organization has no control or monitoring capability on the device. Some telemetry is available about form factor, operating system, and time and amount of time accessed, but nothing for the security control and monitoring side.

These classifications work well for devices that humans use but might not work well for devices used purely in automation scenarios. For example, you might have wired or wireless sensors for refrigeration units on in-store freezers or on delivery trucks. Or fixed and unfixed security cameras or sensors and controls on the windmills of a wind farm. Or any number of apps that run under their own identity and those that run under the identity of a human user. If this occurs, consider creating a box labeled Sensors (or similar) if a pattern emerges, or TBD (to be determined) if there are a variety of displaced endpoints.

Schedule one or more modeling sessions that bring an argument of architects together with experts from the IT Operations and SOC team. The term argument of architects might sound harsh or demeaning, but it is not and is a necessary part of practicing architecture. The architects and other experts collaborate on defining multiple approaches to land on the best practice or best approach, and to land on an ultimate, or at least single source of truth to move forward with. Create a starter set of questions to use as you look at and refine the model:

- Do we need to separate productivity apps into laptop/PC, tablet, and smartphone groups or can they be combined?
- Do we need the productivity devices/endpoints represented in the strongly managed, managed, and unmanaged buckets?
- Do we have edge cases, such as privileged access workstations (PAWs) that we need to include?
- What IoT scenarios do we need to support?
- Is there a difference between wired and wireless endpoint requirements and constraints?
- Do we need to collect data on one-way versus two-way endpoints (sensors and controlled sensors)?
- Should business value and risk factors be specific to endpoints or combined with identities?
- Can we use a low/medium/high designator for cataloging risk and value? If so, are identities and endpoints combined or separate?
- Do we need to separate partners and vendors? What if they are both a vendor and a partner? Or some other combination of roles?
- Will we need anything for unmanaged devices? If so, what? Will we need incident response? Support teams? Certain governance for when unmanaged devices can be used?
- If we have multiple tenants in our org will that need to be considered and will it change this model? Or will it just change how we perform tasks? OR have no impact?
- What words will you use for labels? Managed? Strongly managed? Unmanaged? Joined? Anonymous? Customer? Mobile? Sensor? Human-paired? Automation-paired? Having a definitive glossary that everyone uses is critical.

- Can onboarding be automated? Should any use cases be manual?
- Can day-to-day operations (like patching) be automated?
- Can most incident responses be automated? What can be preemptively determined and managed with automation? Are there lists of what can be or currently is automated?
- Do we have gaps we need to fill?
- What are the readiness requirements that need training to get staff ready for operationalizing this initiative?
- Is there any end-use or delivery team training needed?

The idea behind the initial models and questions is to provide a starting point for a larger modeling exercise. The first modeling exercise will likely be most effective if you start with the entire group participating in a whiteboard session.

For this first session, have four whiteboards, one for each bucket (grouping) on your starting model, and one for orphans. Make sure and have multiple printouts of the list of starting questions, poster-sized, and up on the walls.

- Have all participants use sticky notes to identify the different endpoints being used. Have them think through operations and incident management to determine all the scenarios they react to.
- After a refreshment and networking break, review all the sticky notes to gain agreement for the bucket each falls in. Don't remove the duplicates, rather put them in the same area – there's value in knowing how many times a specific type/scenario of device pops up.
- Determine if additional buckets are needed, and then determine if groups of devices are close enough from an Ops/SOC perspective that they can be grouped together.

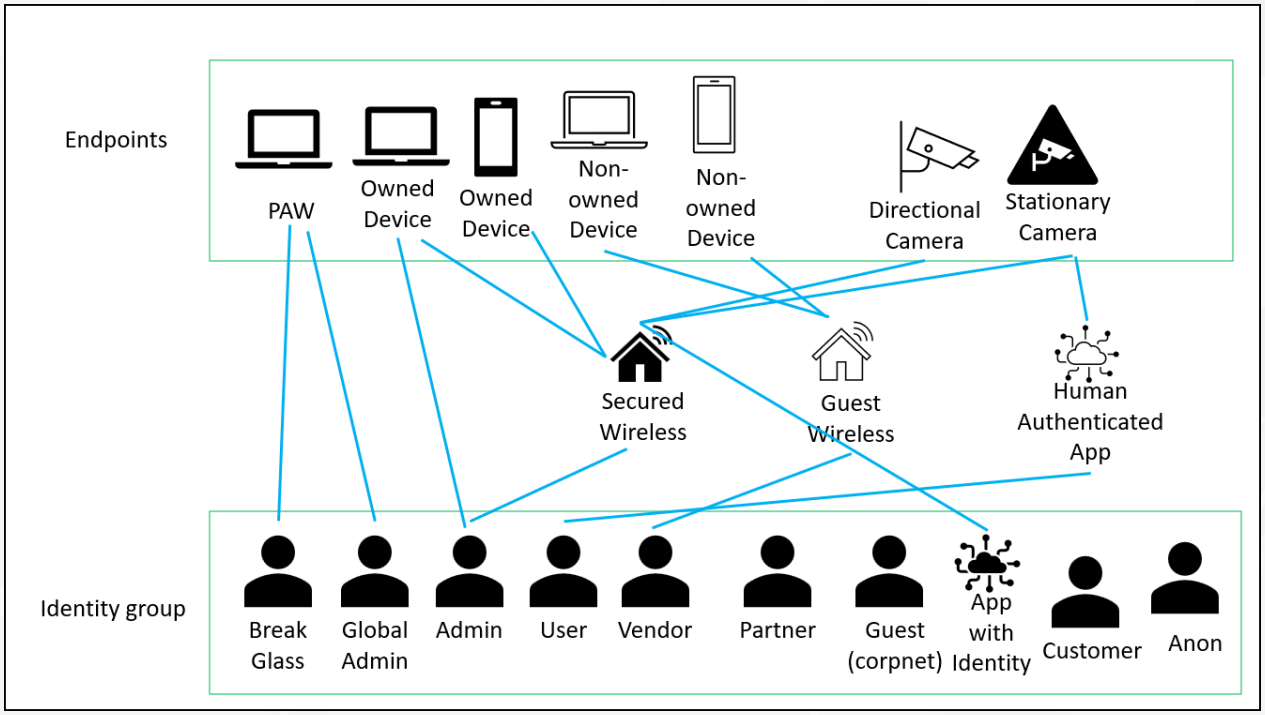
Be sure and take pictures of the result, and if available, have a designated scribe capture notes.

Next, remove all the stickies and re-label the whiteboards to low, medium, high, and other for business value. As a group, place the stickies in the proper place. This should go a bit faster because people understand what you are trying to accomplish, are building a rapport, and you've grouped some of the stickies together.

Finally, remove all the stickies and re-label the whiteboards for risk assessment. You might use low, medium, and high risk. Or perhaps it might make more sense to categorize by critical devices, sensitive data devices, highly vulnerable devices, end-user devices, and non-owned devices. Whatever you decide on labels, go through the same exercise of placing the stickies up on each of the boards.

For each of these steps, take pictures and have a designated scribe take notes. The results can be used in several ways. You can present the results to the leadership as proof of forward momentum and a tool to help elicit buy-in. It might also be used as a great evaluation tool to determine if the participants' gut feel and experience are enablers or barriers/blind spots that require attention. You can show and share with participants to build team and community.

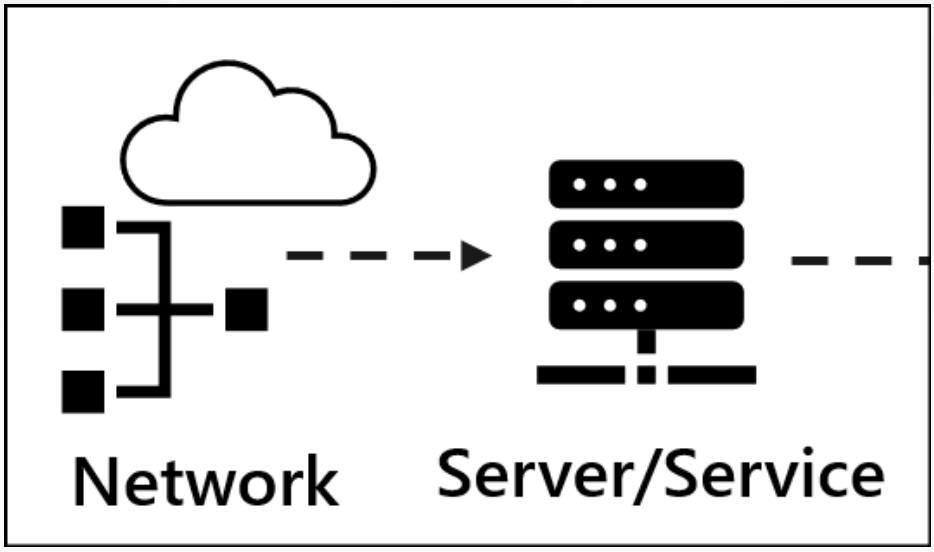
Your goal is to create a model like this, as well as several pictures from whiteboard exercises and documented lists of endpoints, value/risk metrics, and their integration with other pillars of the data request journey. Yours will be more complete and better than this, but we hope this at least gives you an idea.



In your resulting model, you might group all the identities together in one box and all the endpoints together in one box. However, it might make sense to keep the groupings separated into the buckets (groupings) you had in your initial model. When modeling is complete, there will be six pillars, governance and automation represented, and any number of other elements. Models will become complex, and you'll need to consider the best way to capture the complexity while making the information digestible for the people that will apply fingers to keyboards. But, most importantly, you have identities and endpoints modeled. There's still data, apps, networks, and infrastructure. But one bite at a time if you want to eat an elephant.

Modeling a Network Workstream

If there is a pillar that is already using a Zero Trust strategy, it is networking. Networking has been at the center of IT security since the early 2000s and has evolved from the simple firewall for the network perimeter paradigm. Perimeterless computing is a foundational principle for Zero Trust and de-perimeterisation was coined in the early 2000s. Network Access Control (NAC) has been in use since the early 2000s as well. Software-Defined Networking (SDN) has been around since about 2010. For those of you who have depth in networking, these are known terms. For you architects whose focus is on other areas, here's a brief definition: NAC is a set of technologies and protocols that are used to control access to a network; SDN, on the other hand, is a network architecture that separates the control plane from the data plane. This provides more centralized control and management of network resources.



While there are more tools that help to align the networking pillar to Zero Trust, designing to be perimeterless and incorporating NAC and SDN gets you most of the way there. That said, there are more tools that are very effective, but these are the key areas. For most organizations, the initiatives and projects will focus on making sure all the network definitions are captured and defined along with a business value and business impact upon failure for each network asset.

Some keys to getting started:

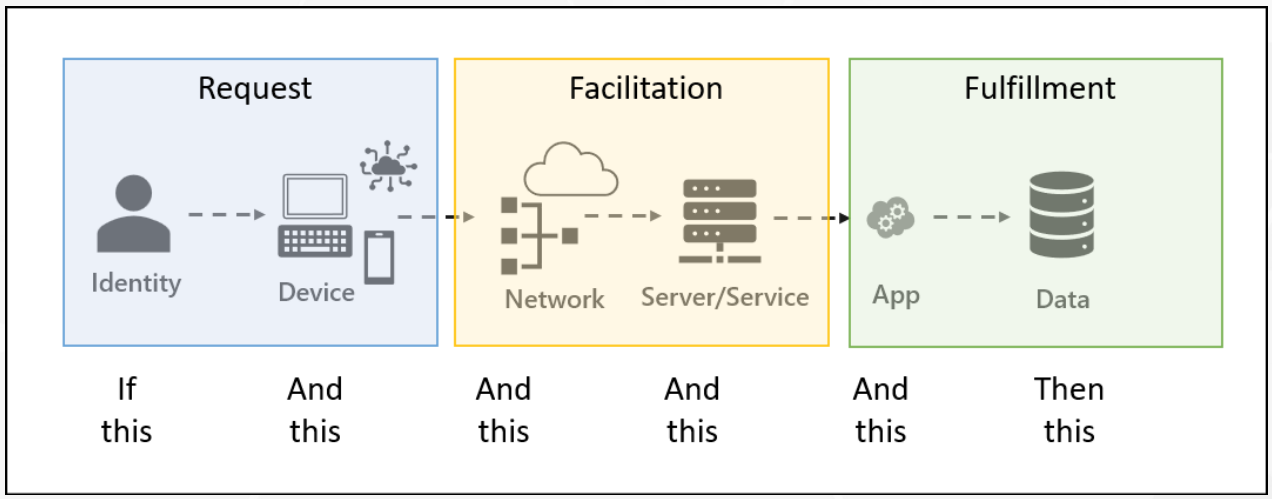
- If your organization does not have a high degree of competence with perimeterless design, NAC, and SDN, perform a gap analysis and hire or train existing employees to fill the gap. Any older school employees that have been resistive will quickly come up to speed as they see how easy (with their background) NAC and SDN are.
- Collect baseline information on the use and maturity of microsegmentation, virtual private networks (VPNs), and encryption both inside and across the firewalled perimeters.
- Collect baseline information for the amount and automation level of network logging, monitoring, and analysis, as well as the ability to monitor and react in real time.
- Determine the level of maturity using machine learning tools and techniques for threat protection and filtering.
- Determine the level and adoption of virtual desktop usage for end users that work remotely.

Next is modeling your network environment. All the data collection and modeling for each pillar are meant to support the goal of following and automating a Zero Trust strategy. As a reminder, we've stated that we believe:

- The goal of IT is to deliver the right data to the right identity in the right location at the right time.
- The journey of a request for data is for an identity to use a device to traverse a network to reach infrastructure running an application that provides access to data.

- The goal of IT security employing a Zero Trust strategy is to ensure that only the right identity can accomplish the goal and can always accomplish the goal.

To make it easier to model, you can break the journey into the request side and fulfillment side, or even break out the network and infrastructure as the transport or facilitation mechanism. This might make it easier to determine how you define your segmentation and perimeters. A mental model you can use to help stay focused could look like this:



The goal is to segment each workload/pillar so you can provision different governance rules in a sensible manner that considers the state of each workload or pillar. In previous blogs, we covered segmenting identities and devices. As you model your network, you're not just creating a network topology map, you're trying to distinguish the network states you would want different rules for.

Network state is combined with the other states that support the data request journey to determine what access to data is provided. For example, on an encrypted network, you might be willing to deliver company-sensitive data to an identity using a company-managed device. However, on an unencrypted network or unmanaged device, you might not fulfill the data request. Another example might be accessing a

human resources (HR) system to submit vacation days. While you might not display the employee identification number, title, email address, or phone number, you might display the number of days of vacation available and enable the employee to submit a vacation request.

Readiness from both a competency and a willingness perspective is critical. People like to do what they've always done and to rely on what they know so there might be resistance to introducing or asking about new capabilities and approaches. People also are proud of their expertise and might feel challenged if you ask if they have competence and experience with cloud-based networking technologies. Perhaps approaching the competence assessment from a technology-in-use angle might be the most proactive.

We recommend that you start collecting competence insight as part of the modeling exercise. For example, as you collect information about segmentation you can tell the team of experts that you have a goal of limiting exposure and blast radius with network segmentation. Ask them if or how recently the current state was assessed against your proposed scenario. Ask them to review the current state and make suggestions for changes to be more effective.

Another approach you can take is to leverage the exercise to inventory the as-is state. For example, if you are primarily using the Azure stack, you can use their Zero Trust security posture assessment to create a list of networking capabilities Microsoft has. Then create a spreadsheet and ask the experts to fill out the spreadsheet. The spreadsheet might look like this:

	A	B	C	D	E	F
1	**Product**	**IT capability**	**Currently using?**	**Needed for our environment?**	**Justification (what will it solve for us)**	**Cost model**
2	**Azure DDOS Protection Service**					
3	**Azure Firewall**					
4	**Azure Web Application Firewall**					
5	**Azure VPN Gateway**					
6	**Azure Virtual Desktop**					
7	**Azure AD Application Proxy**					
8	**Azure Bastion**					
9	**Azure Front Door**					
10	Azure Firewall Premium TLS inspection					
11						

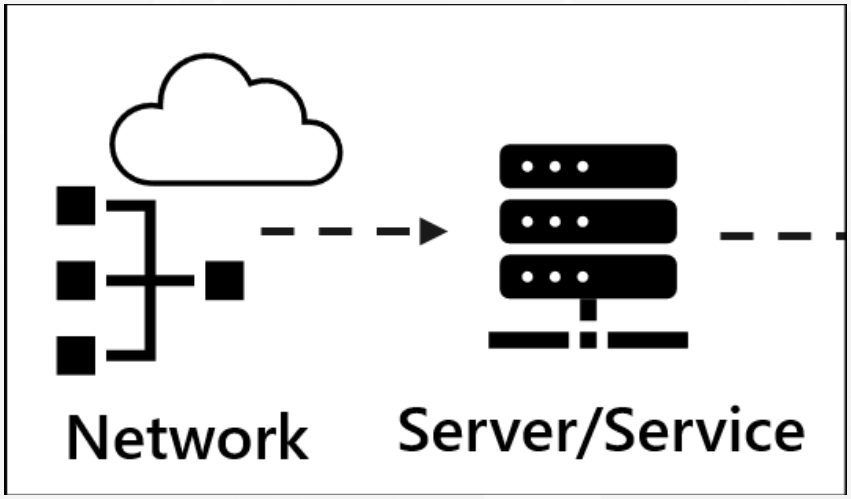
You can populate the spreadsheet with the stacks you have in use or any combination of products you are using or are interested in. This accomplishes several goals including:

- You get the team invested.
- You build expertise internally.
- You discover new capabilities that might be available.
- You have some level of business cost/value and potentially the start of a decision matrix for choosing a stack or product.
- You have the start of gap analysis in your network security posture.

The goal of this modeling exercise is primarily to gain enough insight into your network to create data access rules for an automated policy enforcement engine. However, the result is so much more. You validate and evaluate your current state, technologies in use, and level of competency with current products that are available. One bite at a time.

Modeling an Infrastructure Workstream

We would love to say, “Don’t worry about infrastructure, it is all good.” But we can’t. And it isn’t. The good news is that the network and infrastructure are likely better maintained from a security perspective than most other areas. The attacks on infrastructure have shifted from brute force to more sophisticated approaches, seem to be trending towards cyber espionage as the driver, and healthcare and education as favorite targets.

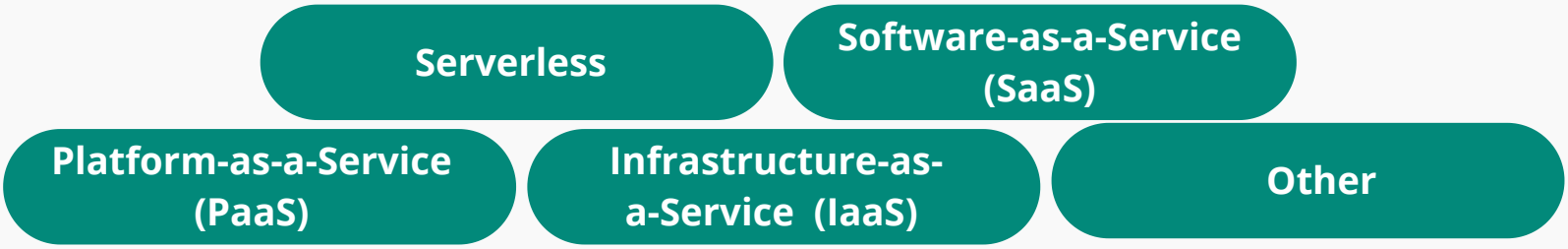


The landscape has also shifted with the introduction of Infrastructure as Code (IaC), containers, serverless, and microservices. The shift in infrastructure strategy specific to Zero Trust might be hard to draw, but we absolutely need to know what assets we have, what their business value is, and what the impact on the operations of the business is if they are compromised. We need to know what automated rules and abilities are needed to stand up new instances, operate infrastructure, and, as end-user or clients, gain access to data using applications that run on the infrastructure.

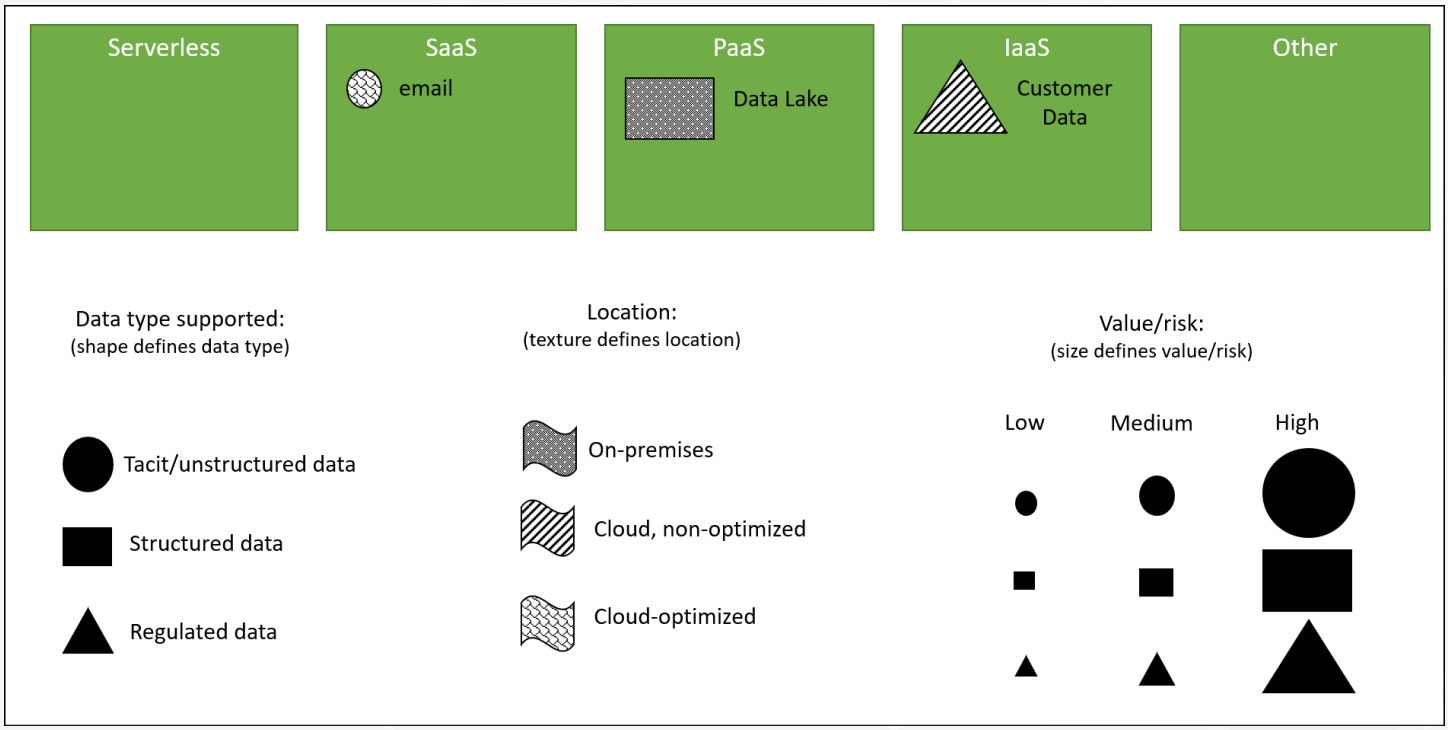
Server patching is still needed but might change as operating platforms move to product delivery teams. Whether specifically tied to Zero Trust or not, Zero Trust does provide an imperative to push infrastructure initiatives through and define ownership (and responsibility) for the use and operations of infrastructure. Hopefully, modeling for this workstream is limited and the bulk of the work is collecting information on what exists and what is needed. Some keys to getting started:

- Ensure you have a complete inventory of infrastructure assets, some form of business impact, and value provided (and at risk) for each asset.
- Ensure the use of IaC in the application development process is defined and followed and that the images are patched and maintained.
- Ensure infrastructure provisioning is automated, and that you can verify host configuration and supply chain provenance and integrity for all technology components.
- Ensure incident response processes are defined for all instances of infrastructure, including virtual machine (VM) usage, app development, front-end, middleware, back-end servers, and productivity platforms.
- Ensure the infrastructure follows a versioning scheme, that the live version has no administrative privilege available (is effectively read-only), and patching is primarily achieved through versioning.

As you start your modeling process, consider including all the technological assets that might not be represented in any of the other pillars. You can decide later who ultimately has ownership and responsibility for technology, but better to be aware of a class of technology that is not assigned correctly than to have one forgotten. To start your model, consider creating five boxes:



You might find you need to separate by on-premises and off-premises or that using a back-end/middleware/front-end/edge/other structure makes more sense. You might also find you need to create buckets for containers, VMs, or edge servers. As the team of architects and experts who help with this effort starts, make sure they do not feel constrained by your starting model. As part of the process, encourage participants to play with the structure and adjust as needed. The starting point could look like this:



With luck, you've already got a solid naming convention for infrastructure and a good, detailed mapping of each infrastructure asset with other key information. And hopefully, you have a solid (and as automated as prudent) approach to patching. This exercise might be a review of existing data or a restructuring of the data to support the overall Zero Trust modeling process.

A good next step is to determine the level of automation and tools in use for defining and maintaining infrastructure. Again, hopefully, you're already steps ahead of this.

If not, using IaC can:

- Speed time to production or market because the environment is codified, and documents and human error are greatly reduced.
- Limit configuration drift and increase consistency by ensuring the configuration between development, test, and deployment environments remain consistent and are in a known state.
- Faster, efficient development as the developer team enjoys having a near one-click environment hydration process. This removes the environment as a variable for inconsistency as development work begins. This also removes the delay or churn that can occur as environment creation is not dependent on a limited number of people that might be on vacation, out sick, or moving to a different team or company.
- Lower cost of development from the efficiencies listed above. The cost reduction and improved stability can be used to justify any initiative to shift to IaC.

With IaC, there is a balance or tradeoff that comes down to the ability to modify and control the environment that people have. The more control people have, the less consistent and efficient the environment. Conversely, the less control people have the more efficient and consistent.

Whether you are collecting information on existing IaC capability or using the Zero Trust initiative to provide IaC capability, you must make sure you determine whether your approach is using mutable or immutable infrastructure, a declarative or imperative approach, and what tools and templates are used for environment definition. There are dependencies you must consider as you design the environment.

A mutable environment can be modified after its original provisioning. An immutable environment can't be modified. We recommend an immutable environment, however, some considerations in making your decision include:

- **A mutable environment provides more flexibility** for the development team to react to customer feedback, respond to emergent security issues, and make changes to their environment.
- **A mutable environment that can and is modified might limit the ability to maintain consistency** between deployments or within versions and can hamper infrastructure tracking capability.
- **An immutable environment "hardens" an IaC environment** as it cannot be changed so the alignment with governance, reduction of drift, and consistency between environments and versions is maintained.
- **An immutable environment requires one or more experts** to create and test the definition. This might introduce a bottleneck to requests for changes to a given environment and might have a negative impact on productivity and a team's feeling of independence.

A **declarative approach** is considered a functional approach where you define the end state and the IaC software and tools take care of the details to get there. This is like using a serverless workflow – you define the workflow and a serverless, unknown thing completes the tasks in the workflow and provides the answer.

An **imperative approach** is considered a procedural approach. An expert defines each step in the environment definition. This is more like a PaaS or SaaS solution – you define it all and have full control over everything.

Note: If you've got to discuss this with leadership to get an endorsement, don't use the PaaS/SaaS/Serverless as an example. Rather use a GPS versus personal/navigator discussion as example. For example, with GPS, it takes an expert to set up the GPS and device, but it is easy to use as you just plug the destination in once it is working. The tradeoff with GPS is you get the directions the GPS gives you with little or no understanding or control of how the GPS derived the route. With the personal/navigator approach, you use personal experience to determine the route. Works great but you might need to call for help if there are temporary obstacles, you want a more efficient or more scenic route, or are going to a new destination you've not been to.

Some considerations for choosing a declarative or imperative environment:

- **A declarative approach requires an expert** to set up, and the experts typically specialize in a specific product or tool.
- **A declarative approach is easier to use once set up**, but you have no internal expertise for maintaining or updating it.
- **An imperative approach requires you to build the automation scripts yourself**, but the tools and guidance make it easy enough to accomplish and you grow organizational expertise.
- **An imperative approach might not scale well** as it requires internal resources to set up the scripts which might lead to a bottleneck.

As with every decision, there typically are no right or wrong decisions, just tradeoffs and living with the decisions. The last area of consideration or piece of information to gather for IaC is what tools and templates are in use (which hopefully they are). The tools and templates are built for specific combinations of mutable/immutable, and declarative/imperative decisions. You must decide if the tool is key and the decisions follow the tool, or if the decisions drive tool selection. The following is a chart you can find in an IBM blog post titled Infrastructure as Code: Chef, Ansible, Puppet, or Terraform.

	Chef	Puppet	Ansible	SaltStack	Terraform
Cloud		All	All	All	All
Type	Config Mgmt	Config Mgmt	Config Mgmt	Config Mgmt	Orchestration
Infrastructure	Mutable	Mutable	Mutable	Mutable	Immutable
Language	Procedural	Declarative	Procedural	Declarative	Declarative
Architecture	Client/Server	Client/Server	Client only	Client only	Client only
Orchestration					
Lifecycle (state) management	No	No	No	No	Yes
VM provisioning	Partial	Partial	Partial	Partial	Yes
Networking	Partial	Partial	Partial	Partial	Yes
Storage Management	Partial	Partial	Partial	Partial	Yes
Configuration					
Packaging	Yes	Yes	Yes	Yes	Partial [1]
Templating	Yes	Yes	Yes	Yes	Partial [1]
Service provisioning	Yes	Yes	Yes	Yes	Yes

[1] Using CloudInit

If your product development teams are using agile or DevSecOps-styled approaches for product development and delivery, then they hopefully are using some automation for version control and configuration management. If so, this might influence the approach and tools used for infrastructure. For example, Terraform works across all cloud providers, provides orchestration capability, and has coverage for the entire lifecycle. It might be a good choice if your product teams have already selected various tools for their product provisioning and management. But Terraform is immutable and declarative so might change your choices for infrastructure.

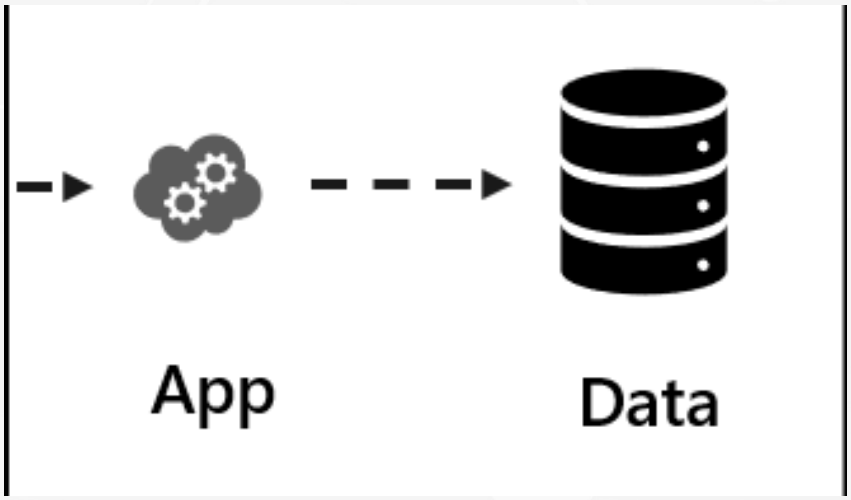
For operating your IT environment, the Security, Information, and Event Management (SIEM) system must be a good fit for the infrastructure. Once you have a complete inventory of your infrastructure, we recommend you complete an architectural-level evaluation of your SIEM to ensure good alignment. At a minimum, we suggest you evaluate IBM QRadar, Splunk Enterprise Security, LogRhythm, McAfee Enterprise Security Manager (Trellix), Elastic SIEM, and Microsoft Sentinel. The evaluation should include the cost of setup and three years of operations, evaluation of organizational competence and available training for each, and the features of each against your IT landscape.

As you evaluate your SIEM environment, consider evaluating your Extended Detection and Response (XDR) capability and performing a similar architectural evaluation. You might consider this part of your SIEM solution or treat it separately and it might be operated by a separate group. XDR also might not fit well into any pillar evaluation so could be overlooked if not captured here.

Zero Trust requires identification and valuation of all information technology (IT) assets, automated enforcement of governance, and automated detection, response, and remediation to threats and attacks. The number of releases and updates to the operating environment can easily be in the thousands daily, and the number of attacks can easily reach ten or twenty million each day. Knowing what you have, what it is worth to the business, and a robust set of automation tools are essential to achieving a Zero Trust security posture. How you model and structure your infrastructure should support these Zero Trust goals.

Modeling an Application Workstream

Keeping it simple even to start is not quite as easy for applications. But you can break it down, have some quick wins, and buy time to work through the more complex challenges. For quick wins, start a proof-of-concept (POC) effort focused on new app development. Set up a secure DevOps toolset and repo and require two-token authentication for data access or access to other servers.



Most delivery teams are independent and don't want outsiders interfering with their processes or tools. If you drive the change, realize it is easy to set up, but it is a bit harder to gain adoption - sharpen your human dynamics skills up front. Another obvious first step is discovering if (and how relevant) of an app catalog you have. Most places have a catalog and attestation process in place. While not complete, it will help you start discovering the scope of the effort. Some objectives you might consider:

- Require all new or modified apps to use multifactor authentication (MFA).
- Configure an Adaptive Access Control (AAC) solution to enforce governance and enable access from anywhere using a secured remote access mechanism.
- Create a microsegmentation strategy and design as part of solution and app design.
- Require apps to enforce least privilege.
- As part of the app attestation process, require security testing, pen testing, and code review against new and top attack vectors.

Some keys to modeling this workstream:

- Applications and data are tightly integrated. Make sure members of each team are collaborating and treat any siloed / team thinking infections that have formed.
- Drive the Zero Trust principle of verify explicitly by requiring two-token access to the data layer and using OAuth or similar for integration. Incorporate this guidance into organizational principles and architectural review board processes.
- Leverage any application cataloging and attestation capability to shorten discovery of applications in the landscape. Automate the tooling if it isn't already.
- Focus on front-end and middleware tiers initially and if you must start with just one, start with middleware.

If the automated app release pipeline doesn't enforce Zero Trust best practices, use a POC-style approach to securing the pipeline. Make sure the delivery teams are aware of what good Zero Trust practice in software development is. Success will require participants take some training and potentially work with tools they are not familiar with, but the effort will get teams aligned and help you to stop growing the problem of more apps that are not aligned with a Zero Trust strategy. It'll also help you develop good practices in your organization. Training and adoption will take time to grow to a tipping point, and then to reach the entire development community.

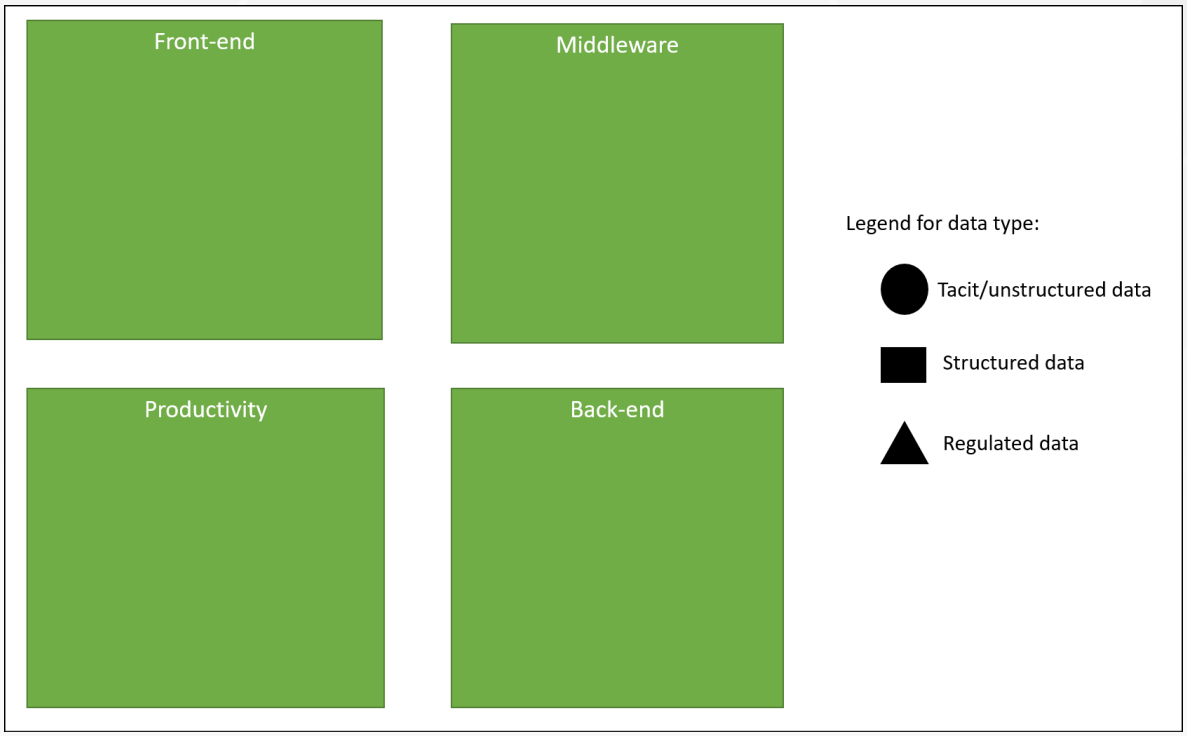
To start your modeling of the application environment, consider creating four boxes:

- Back-end
- Middleware
- Front-end
- Productivity

This should align with technological roles across the organization and make it easier when assigning business value and risk metrics to your information set. Create a legend with three data types defined:

- Tacit/unstructured data
- Structured data
- Regulated data

As you start categorizing your apps and you have an application management system that assigns a unique identifier for each app you can use that, or to make it more readable by people, you can use the people-friendly name of the app. The blank canvas could look like this:



Create a starter set of questions for the team tasked with verifying or creating the list of applications. The goal of the exercise is to have a list of applications with the type of data each app consumes or creates, the authentication methods supported, as well as a business value and impact rating. The questions are to ensure the team feels empowered to question everything as they build the list. Examples of questions might be:

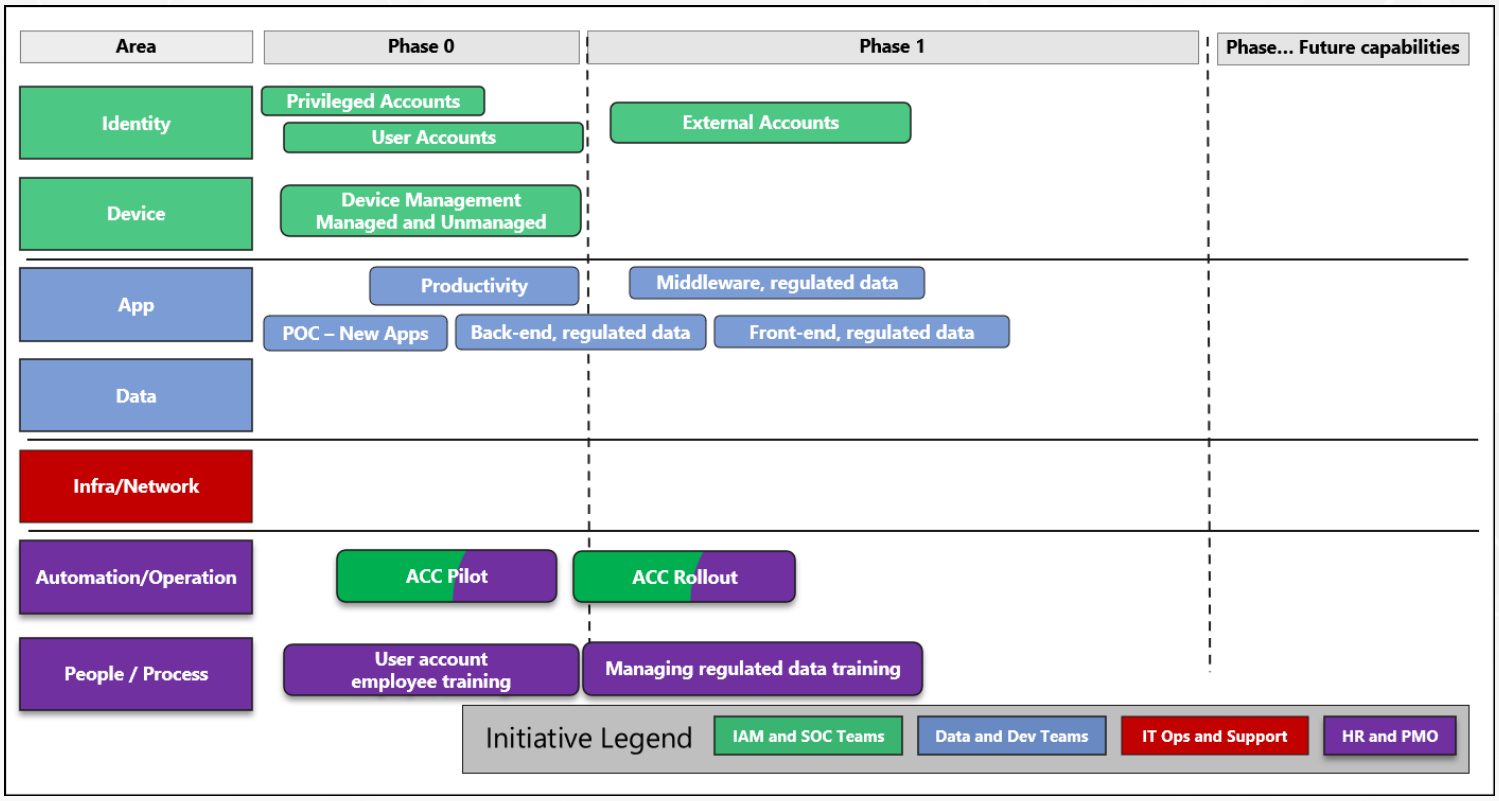
- From an authentication perspective, are all privileged identities treated the same or do we separate global or service admins from those that have administrative privilege for a collaboration portal or print queue? Do we remove administrative rights if inactive for x number of days or months?
- Are authentication methods limited for some applications or solutions?
- Do we need to separate partners and vendors?
- When we say guest are we thinking about old guest identities that were used to provide open access to a server or are we talking about accounts that provide access without federation between two organizations?
- If we have multiple tenants in our org will that need to be considered and will it change this model?
- What words will you use for labels? Admin? Privileged, Elevated? Customer? Consumer?

Create a starter set of questions about authentication methods. While the goal is to capture the authentication method information, the other questions will get the team to consider the other areas and perhaps notate their thoughts in the list:

- What methods can we use and not use? Authenticator app? Email with PIN? Text or phone call back? Username/password? Passwordless? Anonymous access? Biometrics? Gemalto synced PIN? Yubi key? Yubi with biometrics?
- How long should authentication be good for? Should we and can we set up accounts so that administrative privilege can only be given by account elevation? How long should elevations be good for? Is there an idle period where the elevation is nullified?
- Do we want to use/require Privileged Access Workstations? Do we want to use/require managed devices? Are we good with unmanaged devices?

The idea behind this modeling exercise is to create a comprehensive list of applications that are categorized in some way that makes sense in your organization. The list also needs to be actionable, so it must provide enough insight to start updating projects for one or more application categories.

With a good mix of luck and strong leadership skills, the POC and the application categorization should be completed at about the same time. Additionally, the Identity workstream MFA initiative and the Device workstream managed devices initiative will be at points where MFA and ACC for applications are viable. And, of course, not all applications will have dependencies. For example, productivity apps might be able to be configured for MFA and ACC with limited dependency on anything else. A starting roadmap for an apps workstream might look like this:



We recommend the initiatives you prioritize first are the ones you can complete relatively quickly that present limited risk to the organization if the project is not a complete success. For example, employee productivity apps likely can use MFA by enabling the capability. This type of project won't do much for giving developers experience but will move the initiative

forward and help you to identify blockers to launch and to user adoption and can be completed while the POC is being completed.

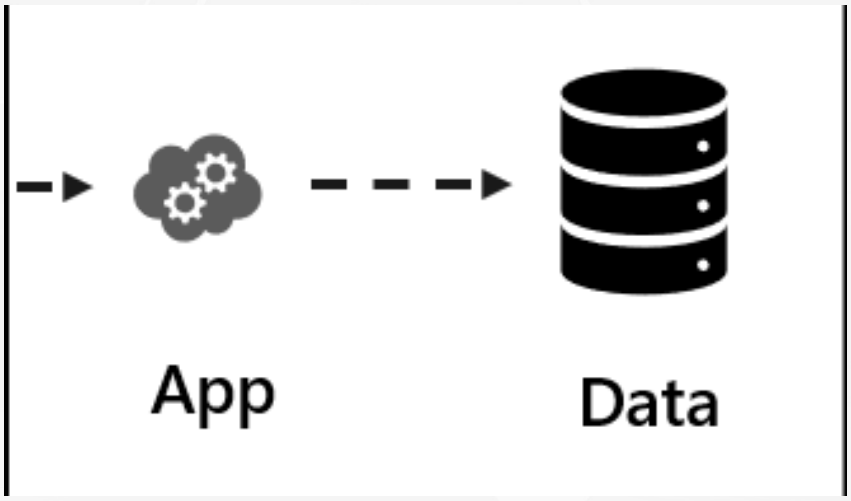
Using the approach of learning and practicing before working on the “crown jewels” will help with the prioritization and order of projects:

By combining the learning curve and value at risk reduction over time together, you can create a compelling message that sets expectations for leadership and have some measures they can monitor to show investment impact.

How do you eat an elephant? One bite at a time.

Modeling a Data Workstream

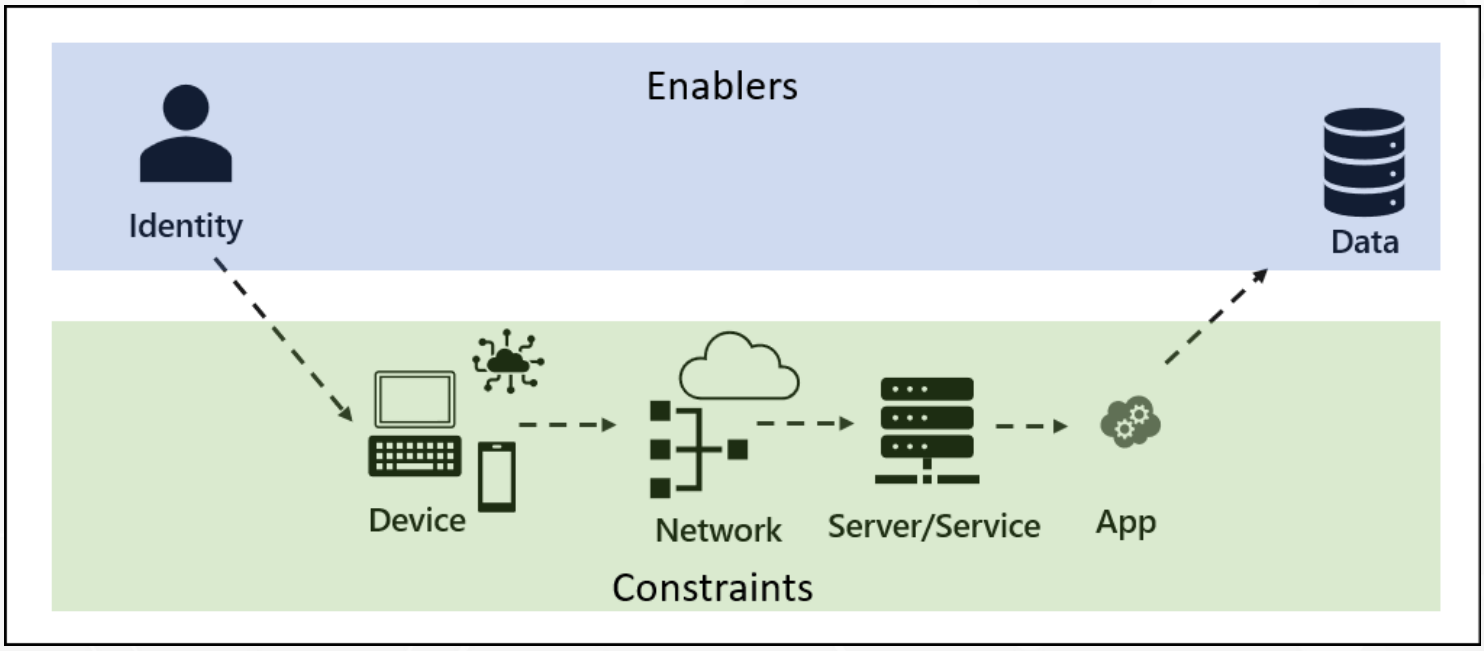
The goal when modeling the data environment for a Zero Trust initiative is to have the information available to decide what data should be available when, where, and by whom. That requires you to know what data you have, its value to the business, and the risk level if lost. The information is used to inform an automated rules engine that enforces governance based on the state of the data request journey. It is not to define or modify a data model.



Hopefully, you already have this information catalogued. From a digital asset perspective, most companies think of their data as their crown jewels so the data pillar might be the most important pillar.

One challenge with data is that applications supply data access. Many applications are not written to support modern authentication mechanisms and don't handle the protocols needed to integrate with contemporary data environments so the applications might not support a Zero Trust data model.

Hopefully, you're already experimenting with current mechanisms for your microservice environment. But, if not, as with any elephant, you eat it one bite at a time. An approach that might make modeling easier is to consider this mental model to start:



The goal of a Zero Trust strategy is to get the right data to the right identity (and only the right identity) by enforcing the governance model. The gatekeeper to the data should always be through an application. The device, network, and server/service states function as constraints to what data is accessible. Some keys to modeling this workstream:

- Get principles in place to drive your goals for data access being aligned with your zero trust goals.
- Specifically, make sure there is a principle that direct connection to data sources must be through a programmatic interface and direct connections are cause for immediate termination.
- Make sure there is another principle that states access control requires two certificates, such as JWT (JSON web token) for user context and API (Application Programming Interface) auth-n via SPN (service principal names).
- Make sure your data classification strategy supports your Zero Trust objectives.

- Verify data models are complete and up to date.
- Decide the level of automation in place to control unstructured (tacit) data.

Two potential goals for your modeling exercise could be to:

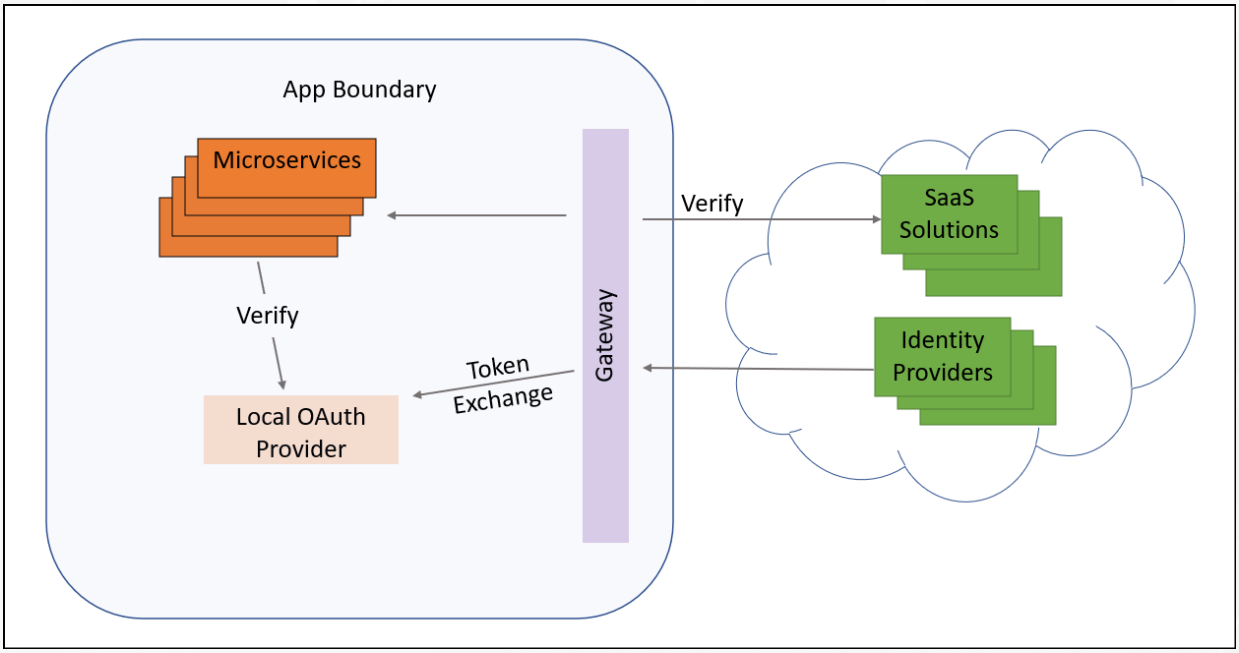
- define an experiment to create an application development pattern that uses current authentication methods. A common framework is OAuth 2.0 covered in RFC6749. Marcus Nilsson has an interesting article that provides some basic insights.
- catalog all data assets in a manner that captures their business value, both from the data's value to business operation and from the potential risk if lost. The catalog is used to create governance that's enforced automatically.

For the experiment, we recommend a direct report of the chief data architect work with a development team to run the experiment. The approach might be to create a microservice design where authentication can be outsourced to multiple external identity providers without requiring each microservice to be coupled to each of the external identity providers. The goal of the experiment might be to validate that the principle is worded correctly, your pattern works, and that the impact to feature release velocity isn't excessive. By calling it an experiment you'll reduce friction with the development team as they run experiments all the time and you won't be seen as coming in and telling them what to do. This approach should also help to build a rapport with the development teams as partners.

You also need leadership endorsement. Setting the goals as validation of the approach and verifying velocity of feature delivery isn't impacted might be better received than having a goal of more secure. As soon as you ask leadership for a team to do anything, leadership starts thinking three developers is equal to $1 million dollars (fully burdened) and teams are typically about six people, so you're asking for $2 million (mental cost models they think in might be higher with inflation and change in the workforce over recent years). And they lose six bodies delivering new

features. If you can set the time of the experiment to 90 days (one quarter), then they'll consider it a $500,000 experiment. By stating the goal is to validate the approach and minimize velocity (rather than promising features or saying it'll be more secure) the leadership is more likely to endorse you, especially if you can say the experiment will be on a set of features that are already planned. Then you are simply doing the same thing in a different way and the leadership will still get their features. By setting it to 90 days, they are less likely to doubt you can accomplish the goal. Longer might cost too much and shorter might raise doubt that you can do what you say in a shorter time. And the business rhythm is quarter, half, and annual, so shorter than a quarter is irrelevant to getting the endorsement.

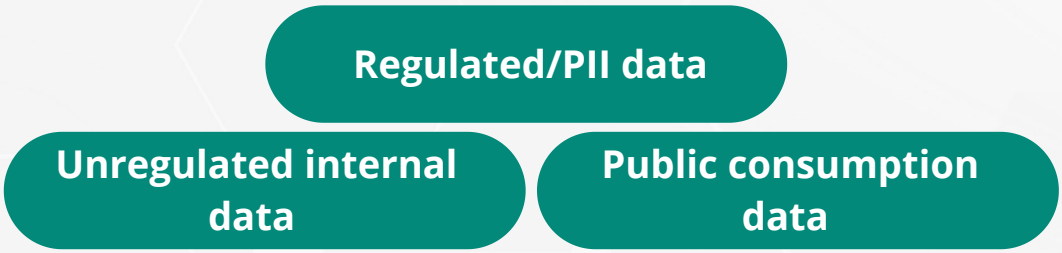
For your experiment, you might start with the designs that Marcus Nilsson uses or one that specifically mirrors internal and external separation of authentication and function, and limits integration requirements for every microservice:



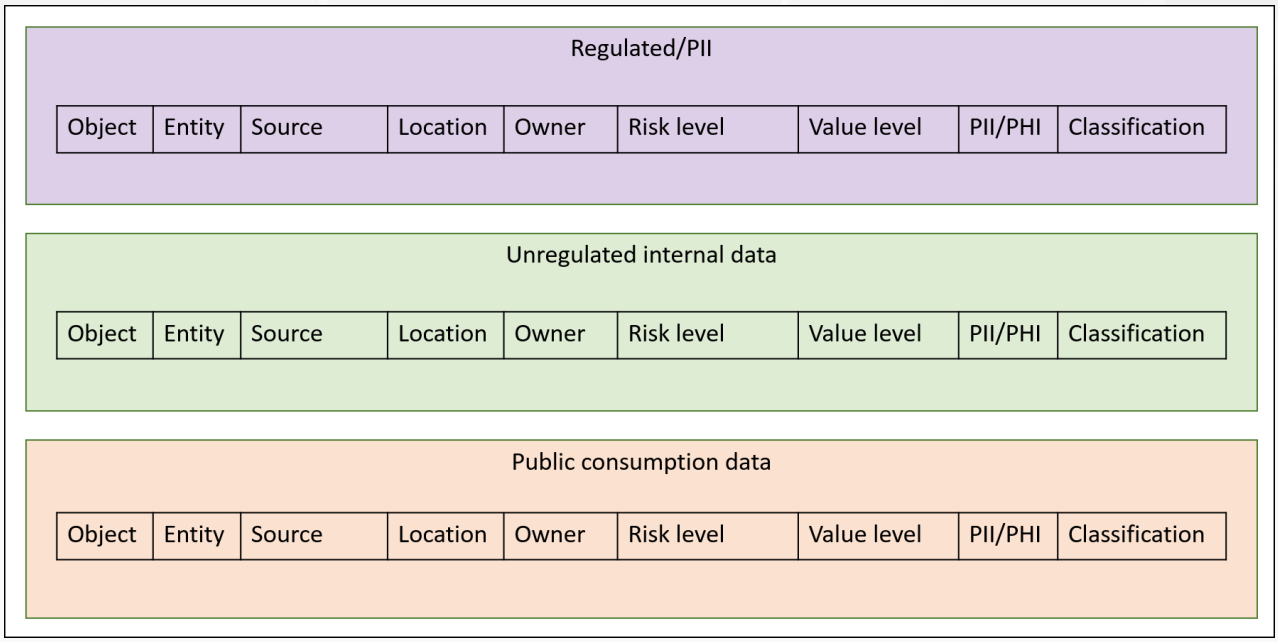
With your experiment, you want to make sure there is a high probability of success, that the reference implementation can be repeatedly used by all the development teams with limited training, and, most importantly, that it provides the Zero Trust alignment without negative business impact. The experiment also starts growing internal competency and provides the insight needed to create training for all developers.

For the data catalog effort, start assessing how mature your data environment is, and how defined and documented the data structure is. Since separate groups are likely to handle the structured data and the unstructured data, you can likely run concurrent efforts to increase velocity towards achieving Zero Trust goals. We recommend that you model the structured and unstructured data separately and that you model the structured data first if you can only model one at a time. Hopefully, you already have a list of authoritative data sources, entities, and objects so this will go quickly. If not, there is no better time than now to collect that data.

To start your structured data model, consider making three boxes:



This effort is primarily meant to find data objects and entities, and their sources. We recommend that, if possible, you categorize your data by object, and capture the entity and authoritative database as metadata. If there are any data sources you do not have metadata for, this exercise is a wonderful opportunity to collect any missing data. Your starting point may look like this:



Organizing data by object will potentially take longer but supplies the level of detail needed to enforce more granular control such as is available when using attribute-based (ABAC) rather than role-based (RBAC) access control. The decision to use ABAC or RBAC is an architectural decision that might differ from solution to solution so having the data collected gives you more flexibility in solution design. What data you collect and how you organize it will, of course, be specific to your environment, but this diagram should give you a starting point for setting up your collection process.

Consider making a checklist of entities before you start. The checklist should be a good reminder for you and your team to use to make sure you have a complete inventory. The checklist might look like this:

Entity	Description	Included?
Customers	Details about the individuals or companies who buy goods or services.	☑
Products/Services	Information about what the company sells, including descriptions, prices, and SKU numbers.	
Employees	Records of people employed by the business, including roles, contact information, and performance data.	
Orders	Information related to sales transactions, including what was bought, when, and by whom.	
Suppliers	Data related to businesses that supply goods or services to the company.	
Invoices	Details about billing and payment information for transactions.	
Inventory	Records of stock for sale, including quantities and locations.	
Financial Accounts	Details of financial transactions, including revenues, costs, and profits.	
Contracts	Information about agreements with customers, suppliers, employees, and others.	
Marketing Campaigns	Data related to promotional efforts, including costs, timelines, and results.	

As team members review the list, having the entities and descriptions should help them think through what objects and entities might be missing. It'll also supply valuable insight to whoever is part of your succession chain.

The tacit or unstructured data is likely managed by the operations and security teams, and the data scientists and data architects might not have much insight. If there are siloes in your environment that separate the teams responsible for the structured and the unstructured data, we recommend you use these exercises to build rapport with both teams and establish a collaboration communications bridge for the two factions.

For your tacit or unstructured data, your goal is still to ensure that the right data is available to the right identity when and where needed, so long as the request falls within the governance boundaries for tacit information. We recommend using a comprehensive suite of tools for unstructured data if your organization has concerns around controlling that data.

You can take two approaches to start your modeling:

- You can choose a tool suite and use the features and options available in the tools to reverse engineer your organization.
- You can build your model from scratch.

For example, you can use the Microsoft stack of tools for productivity - Microsoft Purview and Priva (tools or documentation) can guide your modeling. The tools have a quick start that should show what might be the most relevant objects in your model. The documentation is free and easy enough to skim so you can also use that information to build your model.

If you prefer to build your own model from scratch, we recommend you start by building a table with the common types of data and adding a list of metadata relevant to your organization. Once you have a significant list built, you can organize the data by the productivity tool(s) associated with it. For the common types of tacit data, consider using this list as a starter set:

- Emails
- Social media posts
- Word documents
- PDF files

- PowerPoint presentations
- Audio files
- Video files
- Images
- Website pages

The metadata that might be relevant includes:

- **Key Identifier** - This could be a unique filename or ID, URL, or other unique identifier you discover as you collect information.
- **Source** - Where the data originated, such as the author of a document, the sender of an email, or the creator of a video.
- **Creation Date** - When the data object was created. This might be helpful to help find the most recent version of the document and help with harvesting intellectual property (IP).
- **Last Modified Date** - The most recent date the data object was altered.
- **Location** - Where the data is stored, which could be a file path, a database identifier, or a URL.
- **Format/Type** - The format of the data, such as PDF, MP3, MP4, DOCX, JPEG, etc.
- **Size** - The size of the data object, typically in kilobytes (KB), megabytes (MB), or gigabytes (GB).
- **Access Permissions** - Who can view, edit, or delete the data.
- **Associated Project/Department** - Information about which project or department the data is associated with.
- **Tags/Keywords** - Words or phrases that describe the content of the data object to facilitate search and categorization.
- **Classification** – Classifications vary across organizations. If you don't have a data classification in place, consider starting with public, internal, confidential, highly sensitive.
- **Version** - If the data object has multiple versions, keep track of the version number.

As you model unstructured data, you must start with a clear idea of what your goals are. Just looking at the types of data and the metadata, you can recognize that the information might supply value as IP, introduce risk of exposure of company confidential data, break regulatory and compliance regulations, or could generate insider risks.

The goals you set for unstructured data will help define what you need to collect. However, if the collection of additional data doesn't extend the time or cost of data collection, the information might be useful in later phases of your Zero Trust initiative.

With the data pillar, it is easy to forget the goal is to catalog the data in a manner that governance can be written for access. Use whatever approach you need to remind participants that you are not defining a data model, you are collecting metadata for data types and sources to define governance that can be automated.

Remember, one bite at a time.

Additional Resources

Zero Trust for Architects course

Sustainable Evolution's Zero Trust for Architects course is designed to help you understand how to drive an initiative to gain a Zero Trust security posture for your IT environment. The training is based on the guidance available from government entities, standards groups, and cloud-based products and services companies.

This instructor-guided online course is fourteen classroom hours long. Our typical delivery is six weeks long and includes suggested reading as homework between each session. The course starts with planning a Zero Trust initiative and takes you through the process of planning for the key pillars of a Zero Trust framework.

We share our experience and strategies on how to assemble a team, scope out each pillar, and assess your current security state. We provide links to Zero Trust content for the major cloud service providers.

COURSE OBJECTIVES

By the end of this course, you will be able to:

- Create a set of principles to guide a Zero Trust initiative.
- Assess your current Zero Trust maturity state.
- Use Forrester TEI reports to perform cost-benefit analysis on Zero Trust initiatives.
- Use causality mapping to roadmap Zero Trust initiatives to business objectives.
- Plan Zero Trust solutions to achieve boundary and segment restructuring.
- Determine the skills needed for core and extended teams on Zero Trust initiatives.
- Set a Zero Trust strategy for identities.
- Set a Zero Trust strategy for endpoints.
- Set a Zero Trust strategy for networks.
- Set a Zero Trust strategy for infrastructure.
- Set a Zero Trust strategy for applications.
- Set a Zero Trust strategy for data access and management.
- Automate security policy management.
- Use logged information to perform manual and automated intrusion detection and response.
- Create a change management plan for Zero Trust initiatives.
- Create a communications plan for projects and initiatives related to Zero Trust.

Sign up at sustainableevolution.com/training

