Forward-Looking Statements

This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at www.investors.splunk.com or the SEC's website at www.sec.gov. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. We undertake no obligation either to develop the features or functionalities described, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release.

splunk> .conf23

Clara-fication: Going From Woe to Whoa for Access Controls

PLA1169B

Clara Merriman

Manager, Splunk Security Center of Excellence
Splunk

Man Pham

Splunk Engineer, Splunk Security Center of Excellence
Splunk

splunk> .conf23

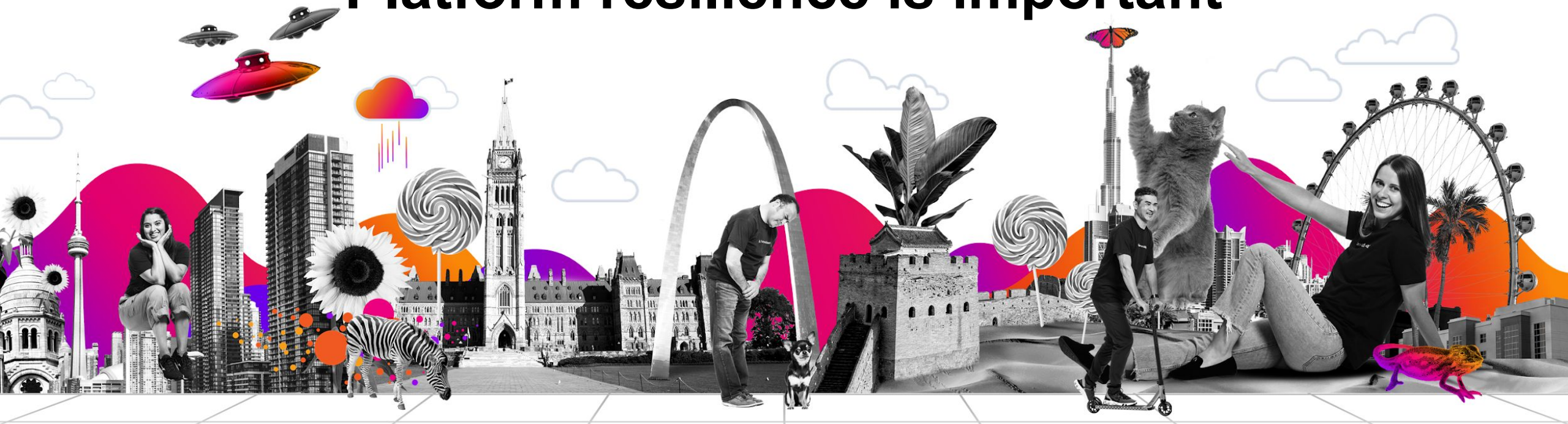
Okay, but why?

Securing data is important!

splunk> .conf23

Okay, but why?

Platform resilience is important



Okay, but why?

Enforcing stateful app permissions is important!

Clara Merriman

Manager, Splunk Security CoE | Splunk

Man Pham

Splunk Engineer, Splunk Security CoE | Splunk

Journey

This is how we do it

- What is RBAC
- How It Started
- How It’s Going
- Considerations

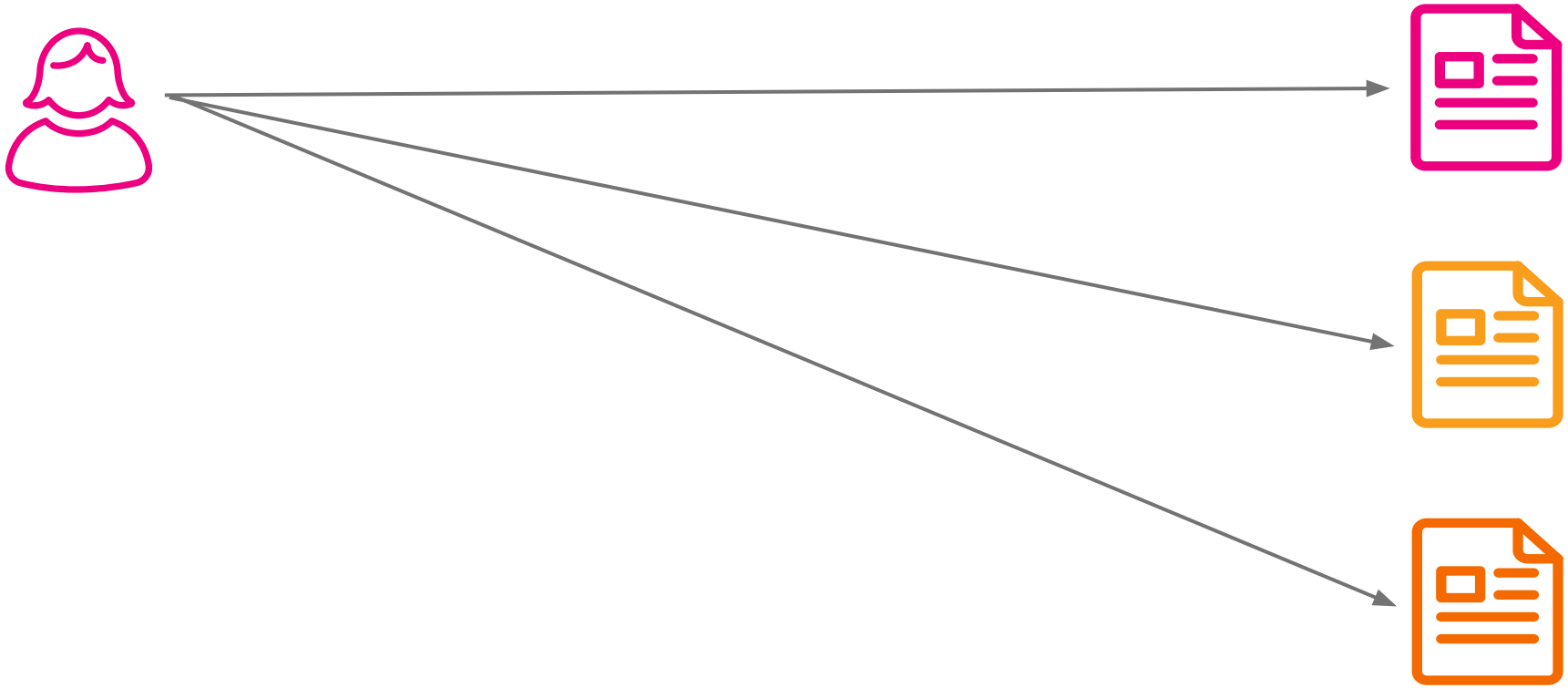
What is RBAC?

Contains Another Backward Redirect

splunk> .conf23

General RBAC Design

Regularly Brings Abrupt Chaos

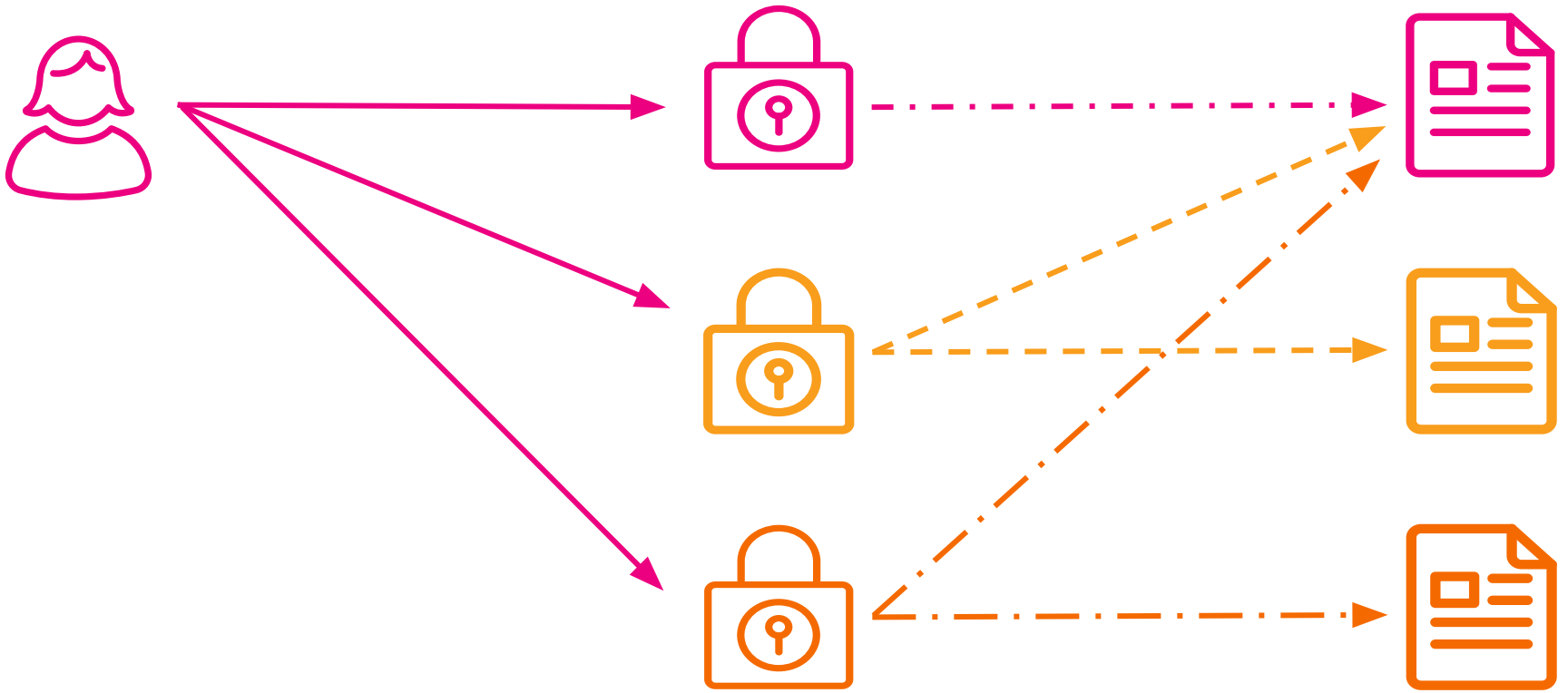


User

Capabilities

General RBAC Design

Rarely Behaves As Created



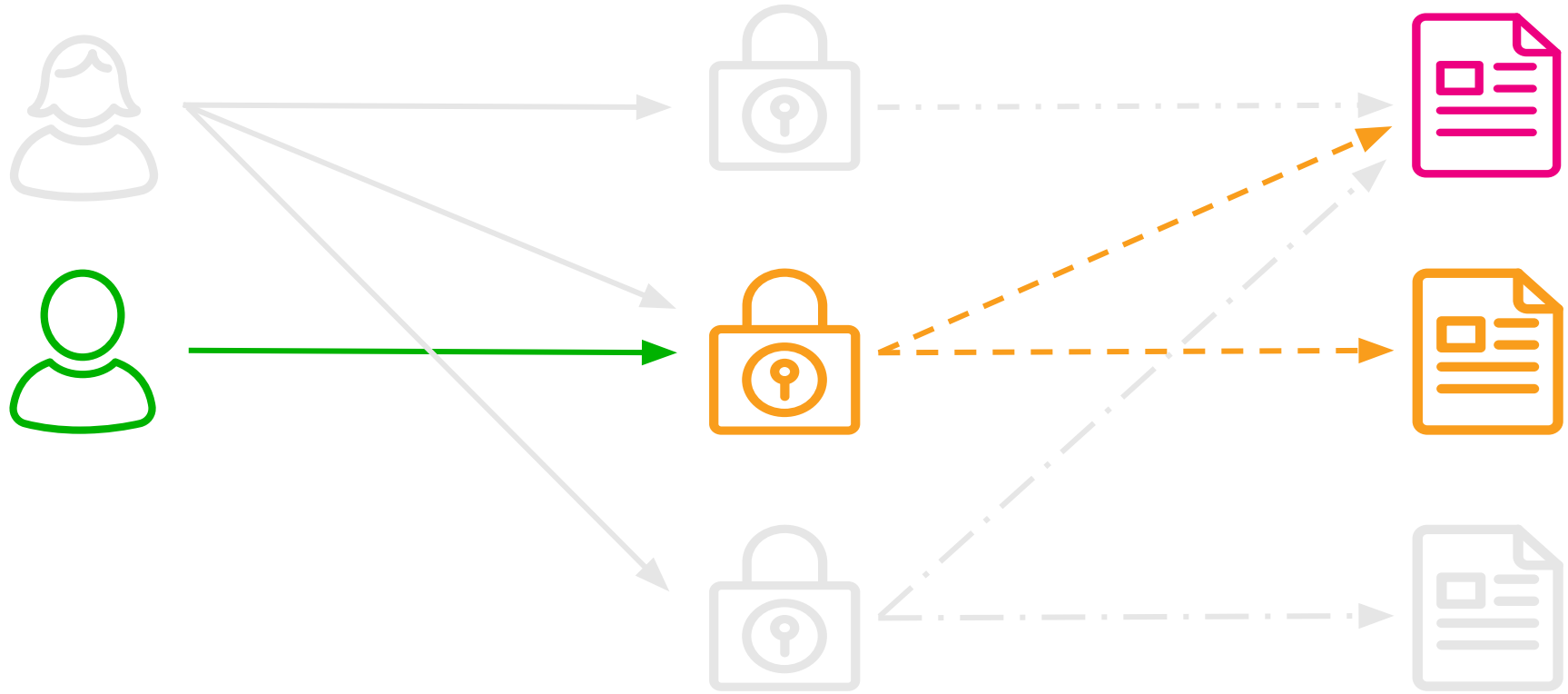
User

Role

Capabilities

General RBAC Design

Rules Beyond All Comprehension



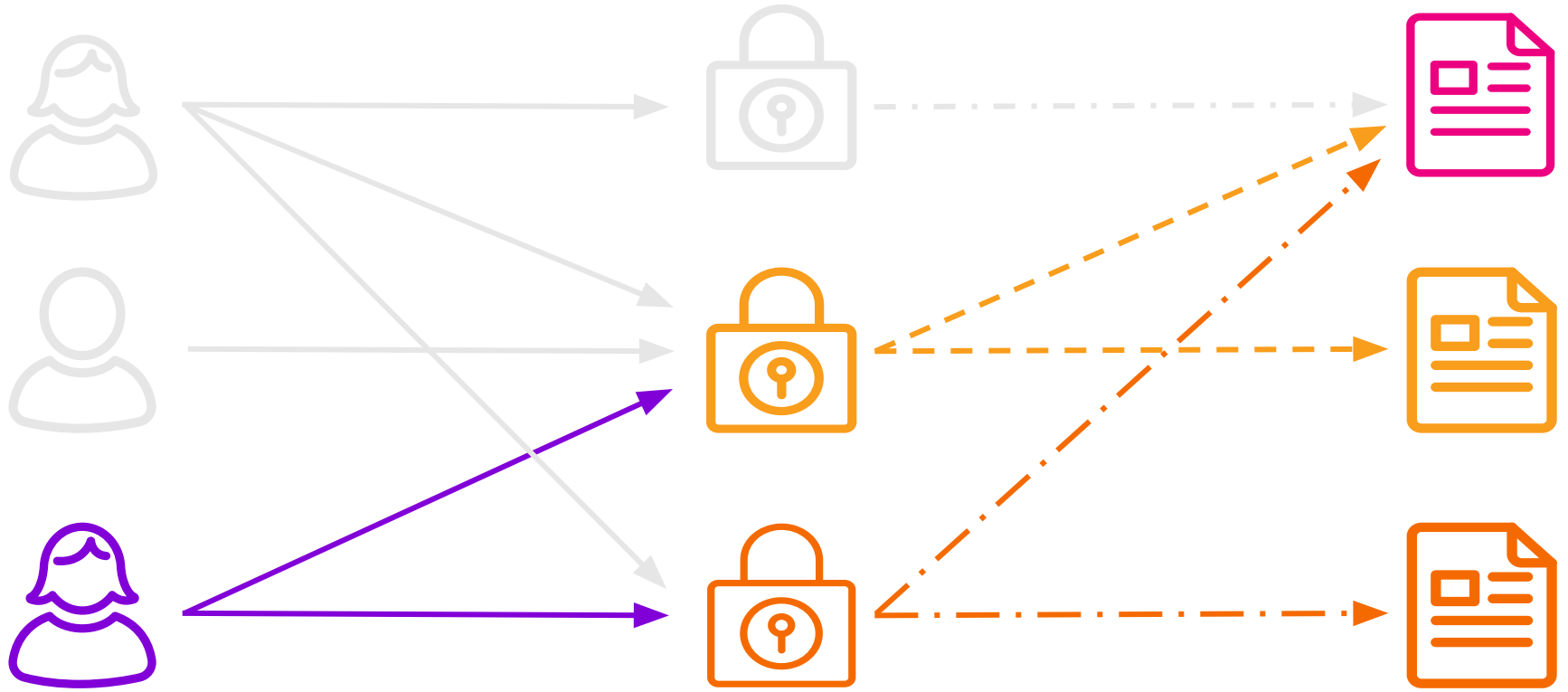
User

Role

Capabilities

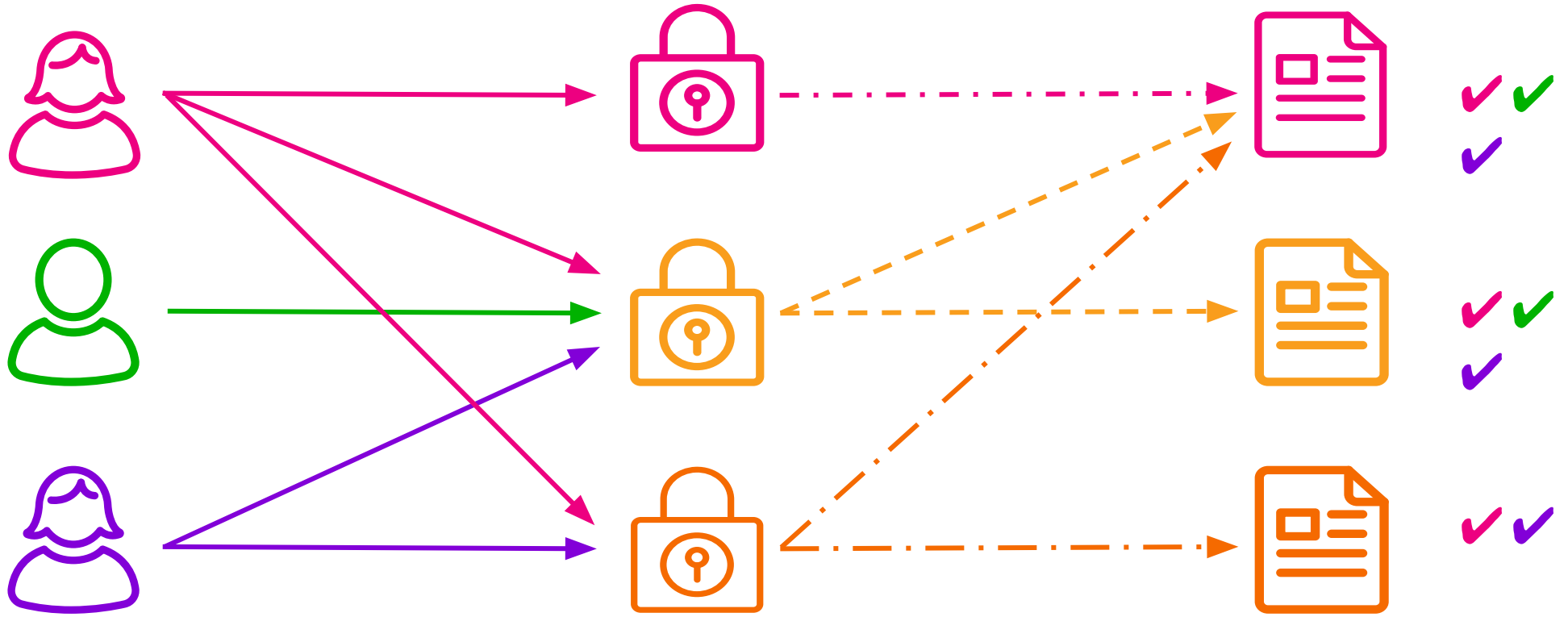
General RBAC Design

Reacting Bravely Amid Crisis



General RBAC Design

Reset Brain Aneurysm Count



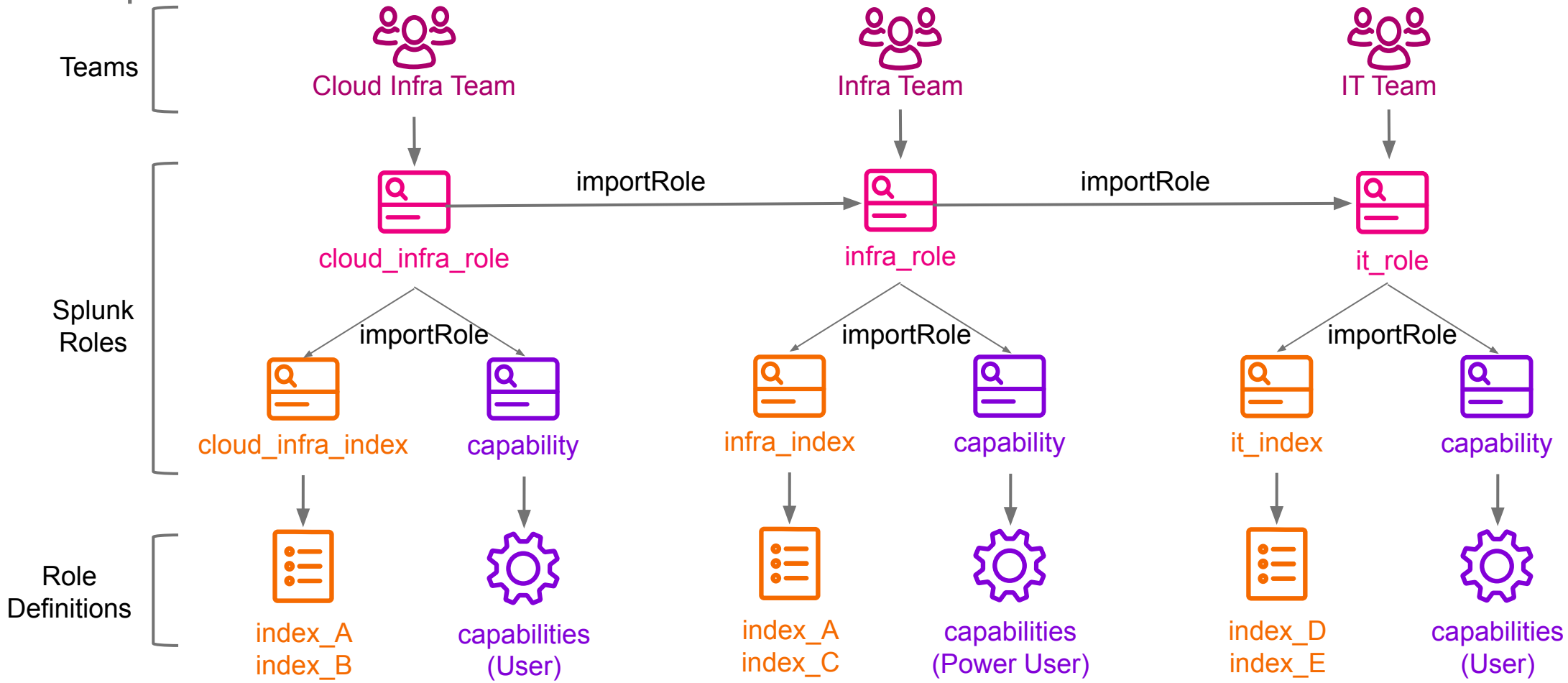
How It Started

Method 1

The first cut is the deepest

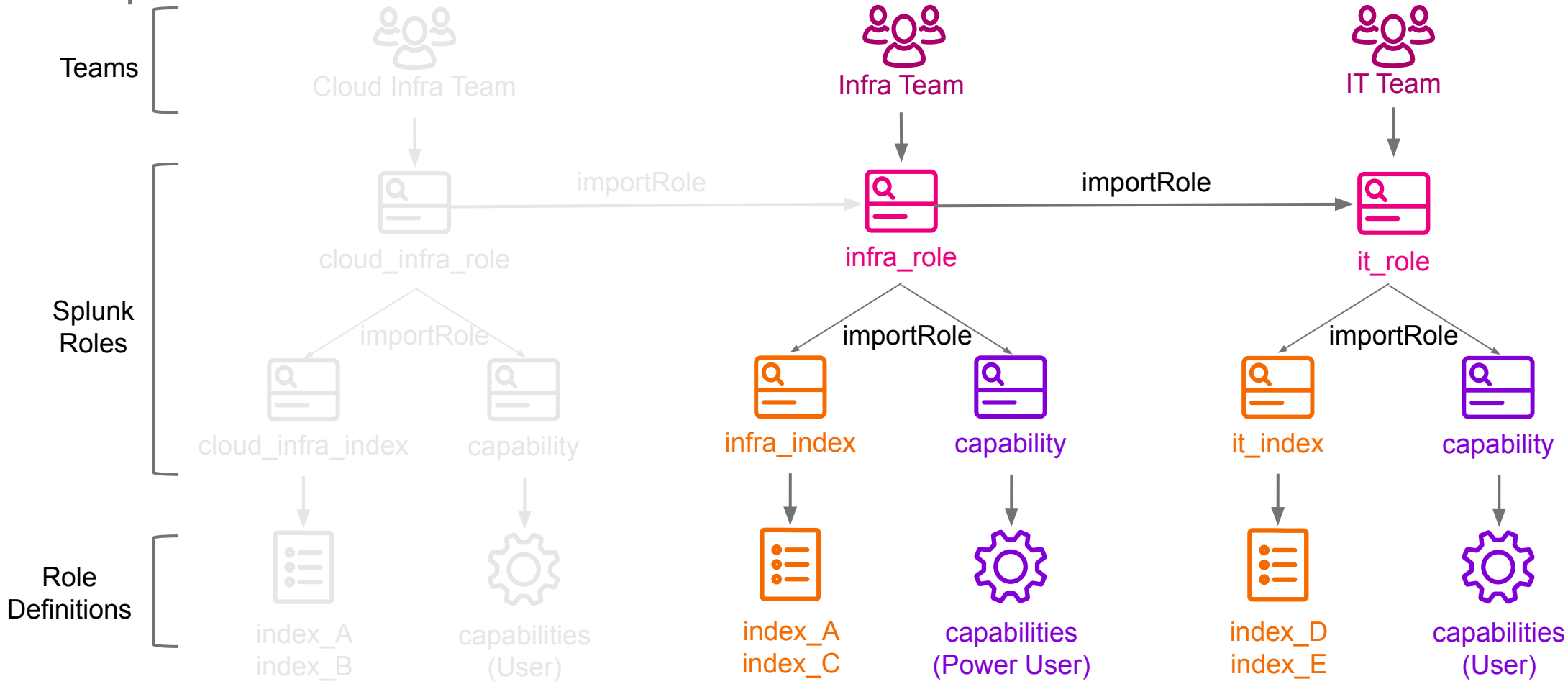
What We Started With

Once upon a time…



What We Started With

Once upon a time…



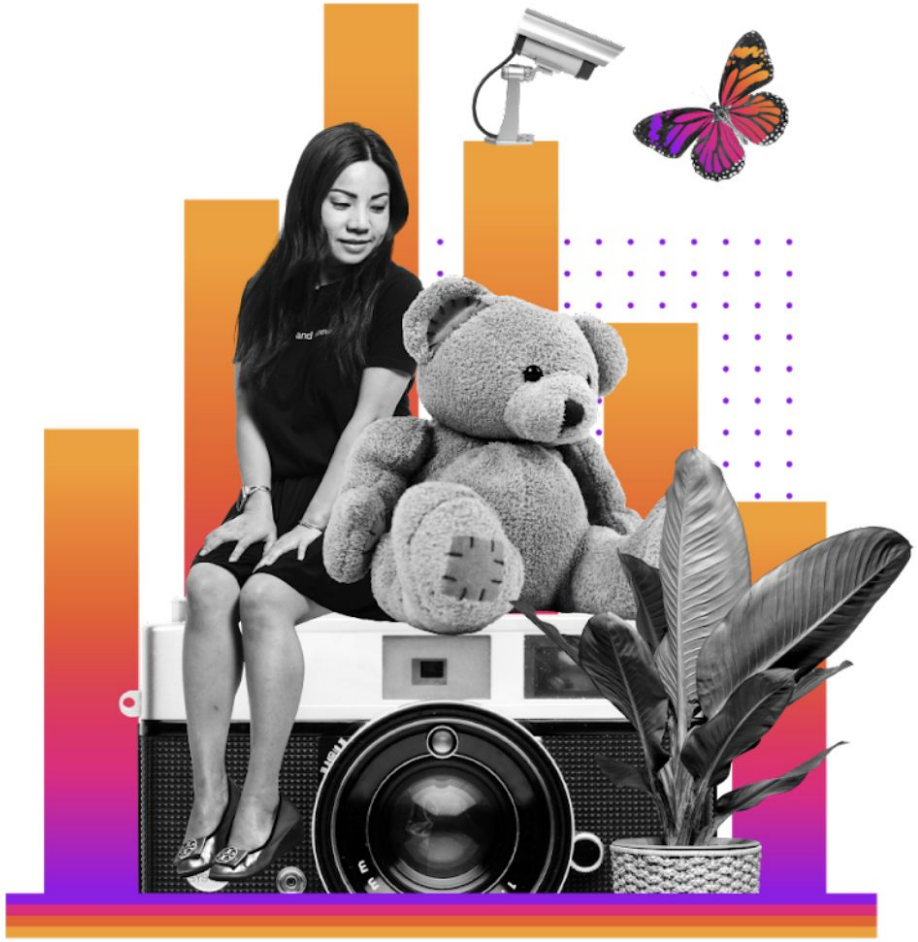
splunk> .conf23

What We Started With

Once upon a time…

Challenges With Team Based RBAC

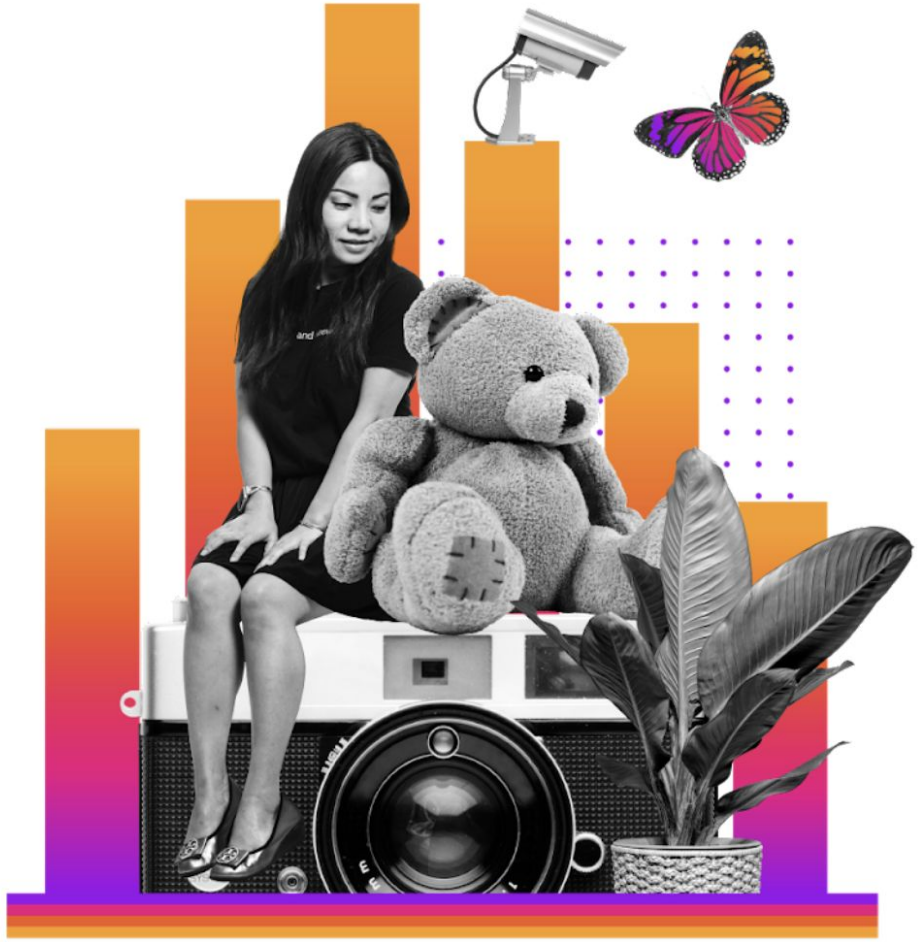
…And then a scary monster attacked…



- Team names can sometimes change
- Nested role inheritance
- Multiple authorize.conf applications
- SAML mapping through UI
- Slow deployment time
- Incredibly permissive
- Application permissions were assigned randomly
- Splunk admins responsible for access approvals

Challenges With Team Based RBAC

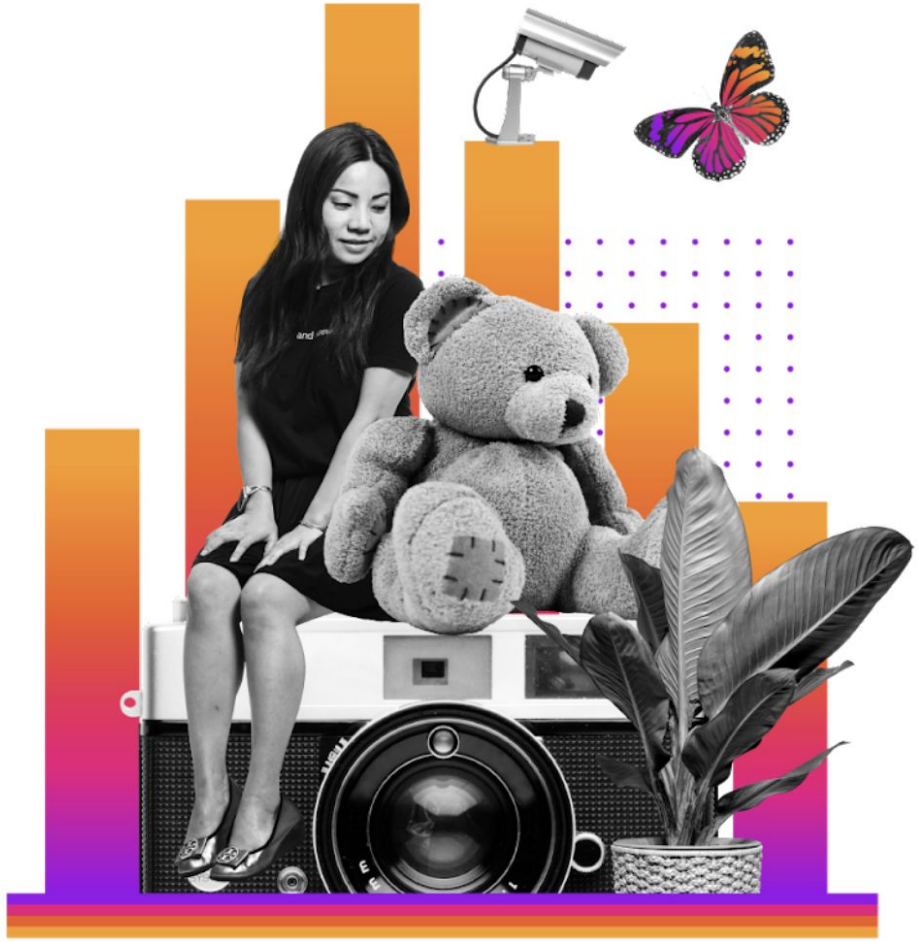
…And then a scary monster attacked…



- Team names can sometimes change
- Nested role inheritance
- Multiple authorize.conf applications
- SAML mapping through UI
- Slow deployment time
- Incredibly permissive
- Application permissions were assigned randomly
- Splunk admins responsible for access approvals

Challenges With Team Based RBAC

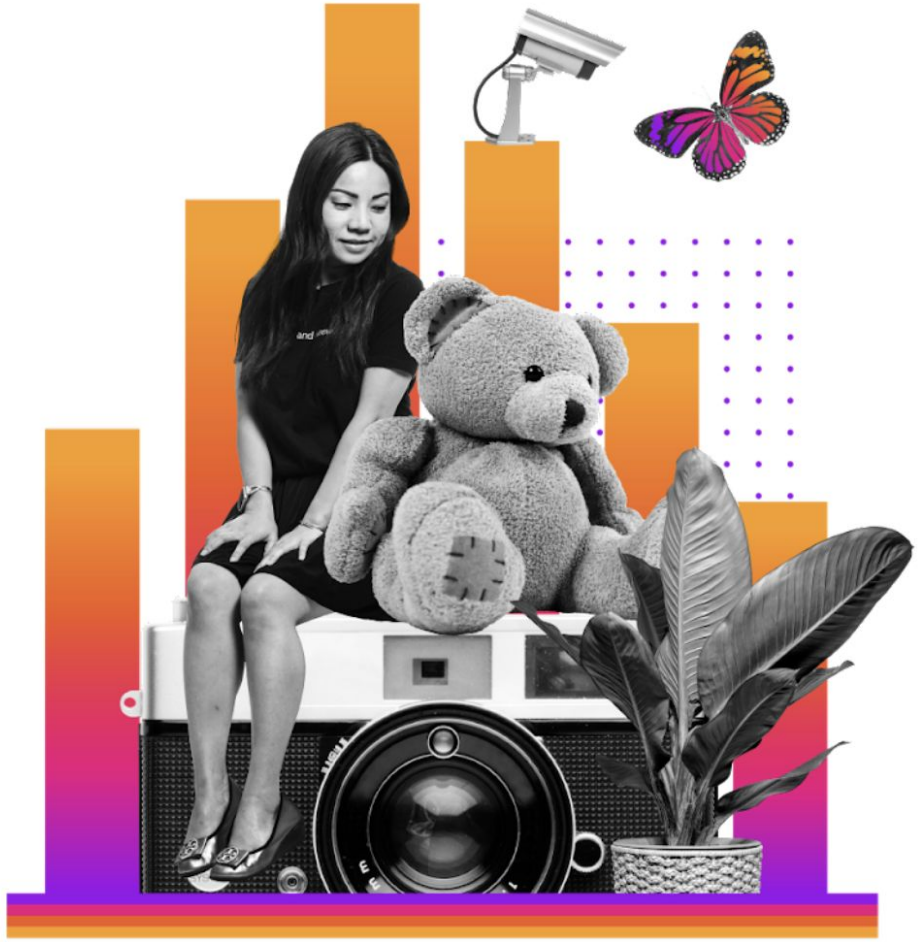
…And then a scary monster attacked…



- Team names can sometimes change
- Nested role inheritance
- Multiple authorize.conf applications
- SAML mapping through UI
- Slow deployment time
- Incredibly permissive
- Application permissions were assigned randomly
- Splunk admins responsible for access approvals

Challenges With Team Based RBAC

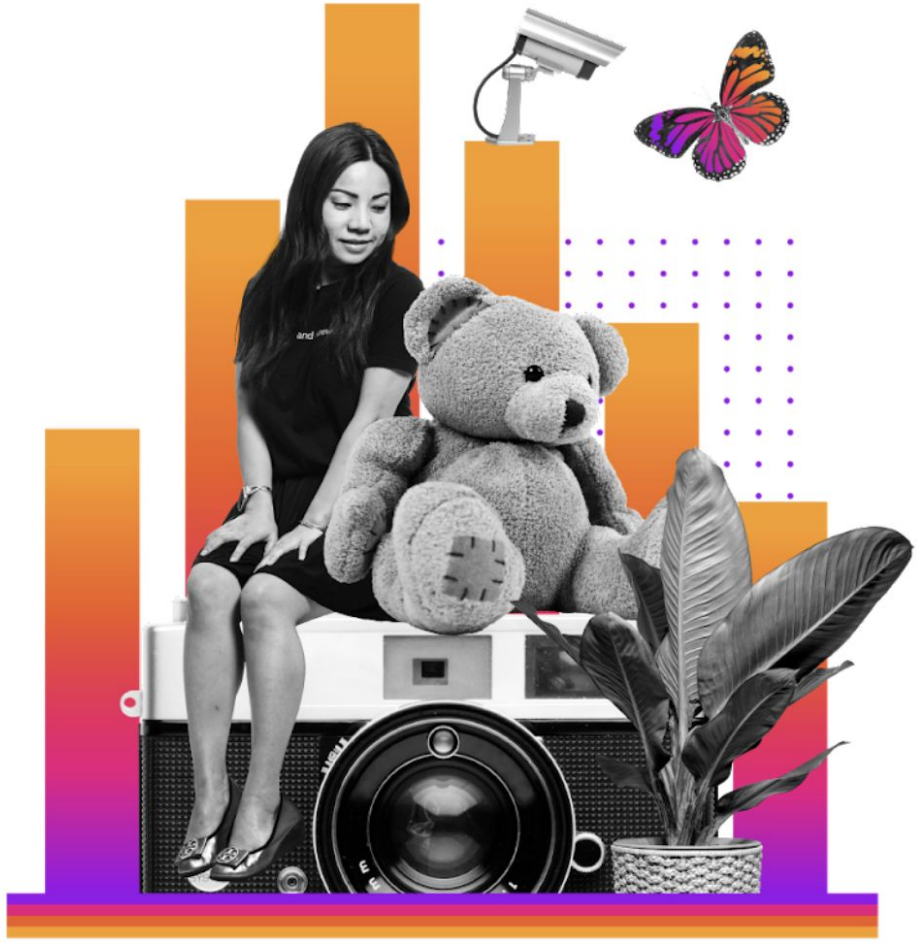
…And then a scary monster attacked…



- Team names can sometimes change
- Nested role inheritance
- Multiple authorize.conf applications
- SAML mapping through UI
- Slow deployment time
- Incredibly permissive
- Application permissions were assigned randomly
- Splunk admins responsible for access approvals

Challenges With Team Based RBAC

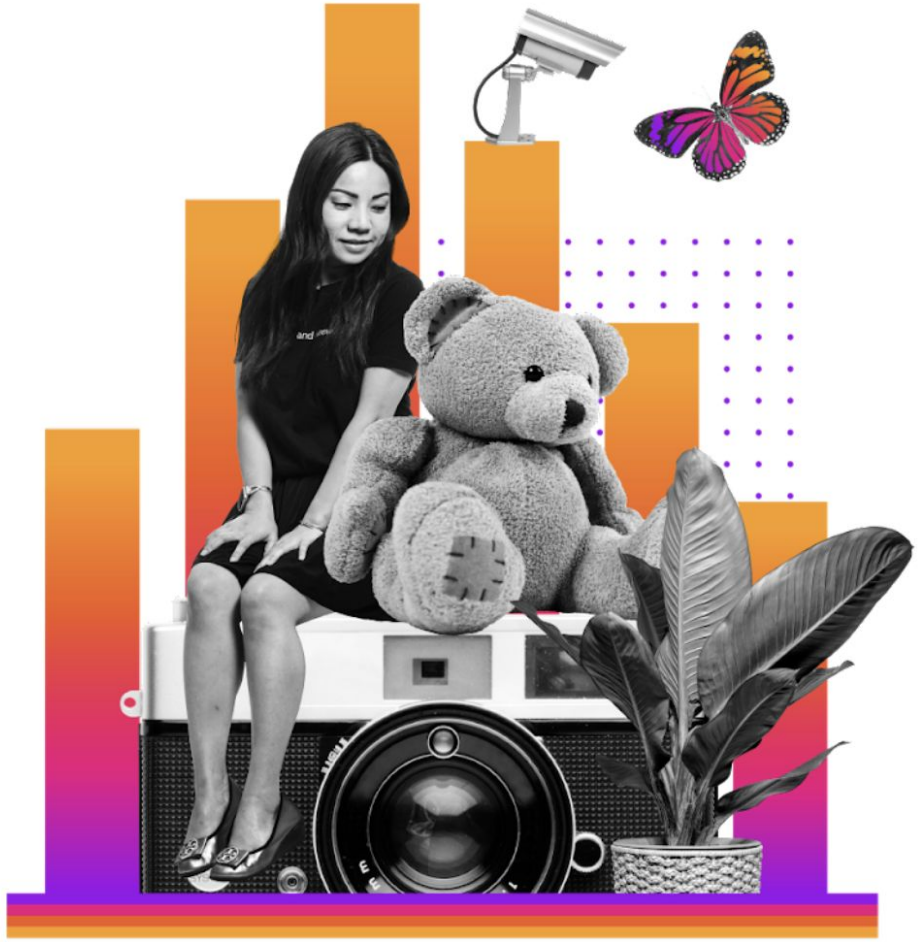
…And then a scary monster attacked…



- Team names can sometimes change
- Nested role inheritance
- Multiple authorize.conf applications
- SAML mapping through UI
- Slow deployment time
- Incredibly permissive
- Application permissions were assigned randomly
- Splunk admins responsible for access approvals

Challenges With Team Based RBAC

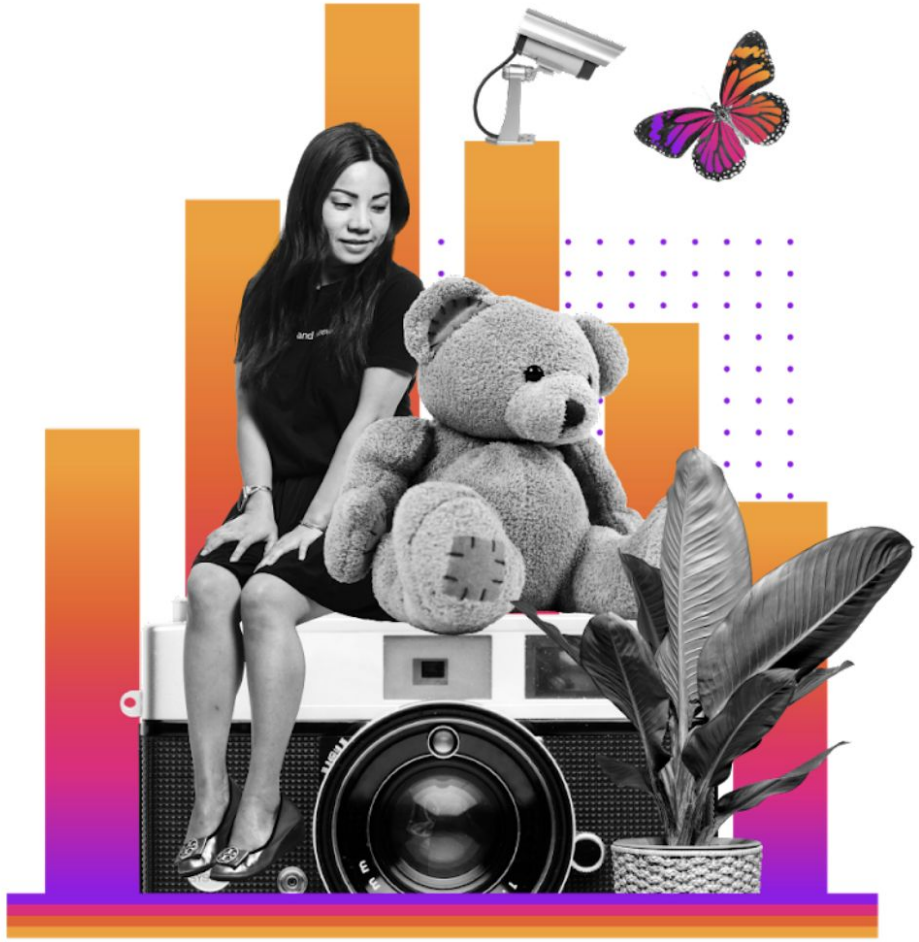
…And then a scary monster attacked…



- Team names can sometimes change
- Nested role inheritance
- Multiple authorize.conf applications
- SAML mapping through UI
- Slow deployment time
- Incredibly permissive
- Application permissions were assigned randomly
- Splunk admins responsible for access approvals

Challenges With Team Based RBAC

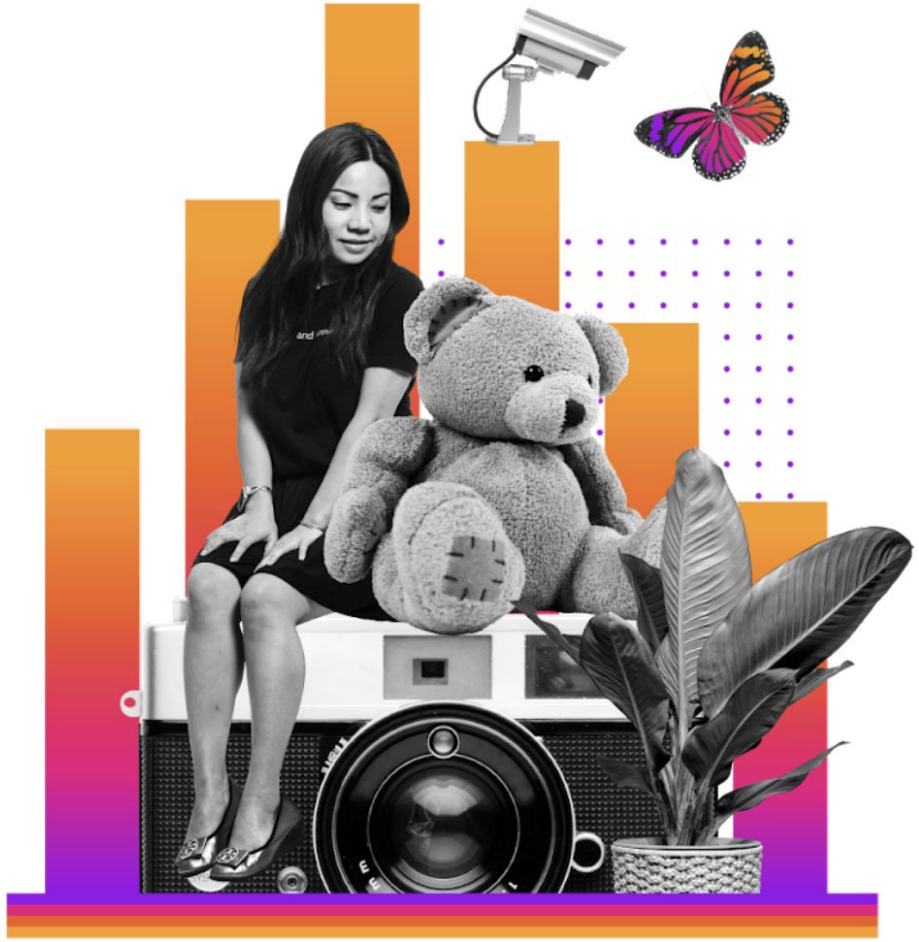
…And then a scary monster attacked…



- Team names can sometimes change
- Nested role inheritance
- Multiple authorize.conf applications
- SAML mapping through UI
- Slow deployment time
- Incredibly permissive
- Application permissions were assigned randomly
- Splunk admins responsible for access approvals

Challenges With Team Based RBAC

…And then a scary monster attacked…



- Team names can sometimes change
- Nested role inheritance
- Multiple authorize.conf applications
- SAML mapping through UI
- Slow deployment time
- Incredibly permissive
- Application permissions were assigned randomly
- Splunk admins responsible for access approvals

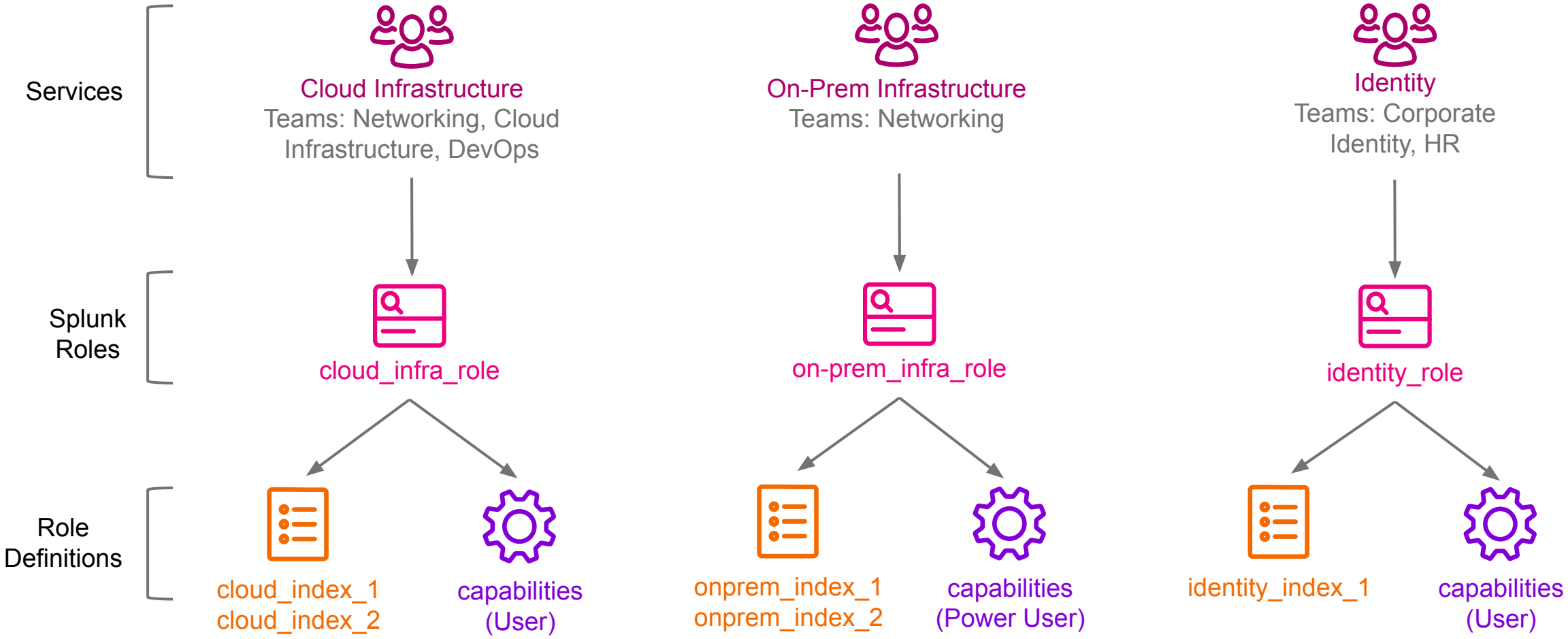
How It Started

Method 2

Let’s try a service based model!

What We Tried

…So the heroes decided to try again…



Challenges With Service Based RBAC

…But the monsters were too strong!...

- Upfront work to deploy automation
- Splunk admins responsible for access approvals
- Defining service groups was nondeterministic
- Still incredibly permissive
- Applications were always read and write

Challenges With Service Based RBAC

…But the monsters were too strong!...

- Upfront work to deploy automation
- Splunk admins responsible for access approvals
- Defining service groups was nondeterministic
- Still incredibly permissive
- Applications were always read and write

Challenges With Service Based RBAC

…But the monsters were too strong!...

- Upfront work to deploy automation
- Splunk admins responsible for access approvals
- Defining service groups was nondeterministic
- Still incredibly permissive
- Applications were always read and write

Challenges With Service Based RBAC

…But the monsters were too strong!...

- Upfront work to deploy automation
- Splunk admins responsible for access approvals
- Defining service groups was nondeterministic
- Still incredibly permissive
- Applications were always read and write

Challenges With Service Based RBAC

…But the monsters were too strong!...

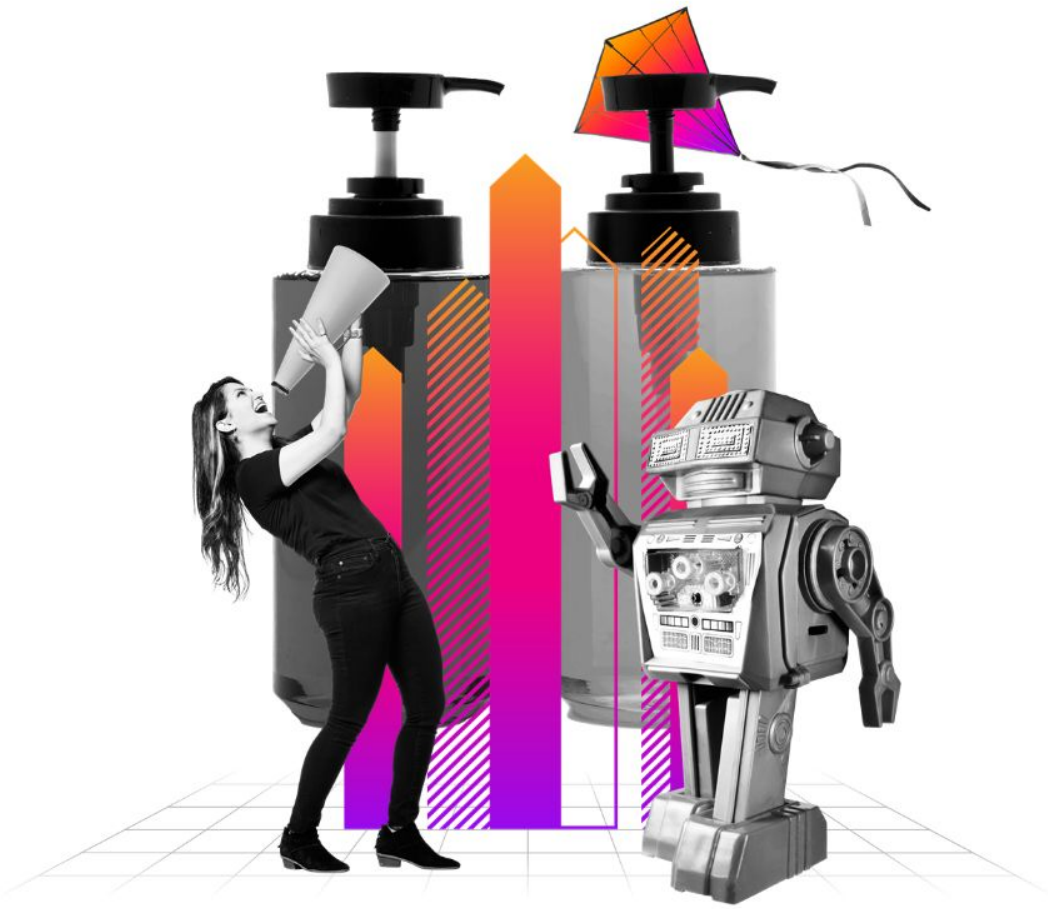
- Upfront work to deploy automation
- Splunk admins responsible for access approvals
- Defining service groups was nondeterministic
- Still incredibly permissive
- Applications were always read and write

How It's Going

Winner winner!

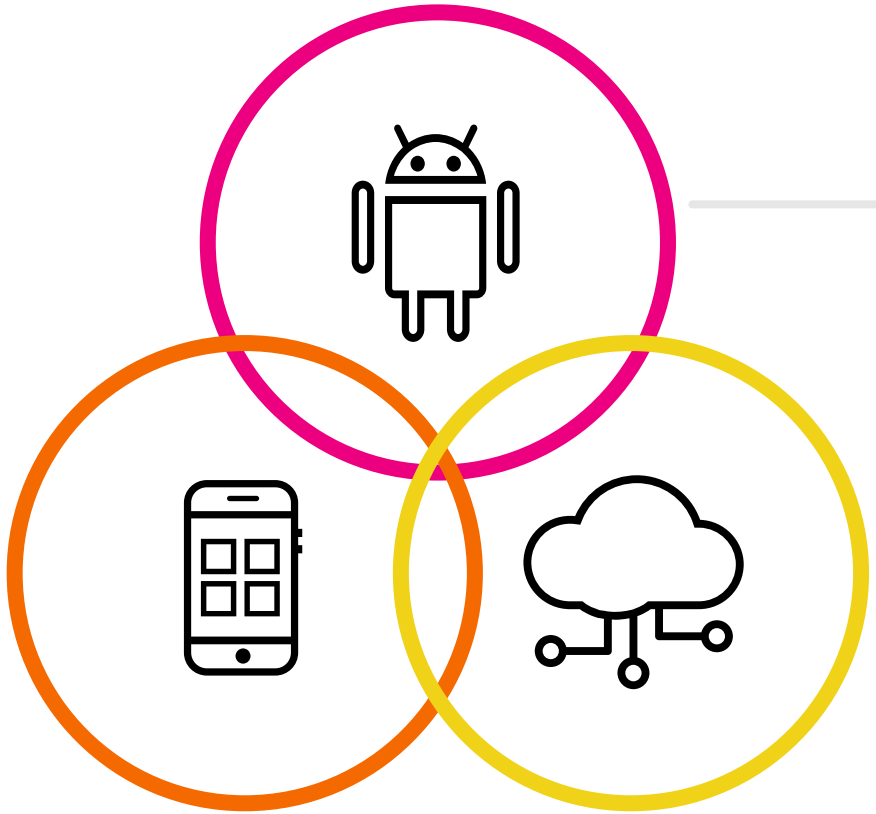
Deterministic model in the making!

We love RBAC so much we redid it…twice!



RBAC in 3-Dimensions

Bringing it all together



Capabilities

Roles for different permissions

Applications

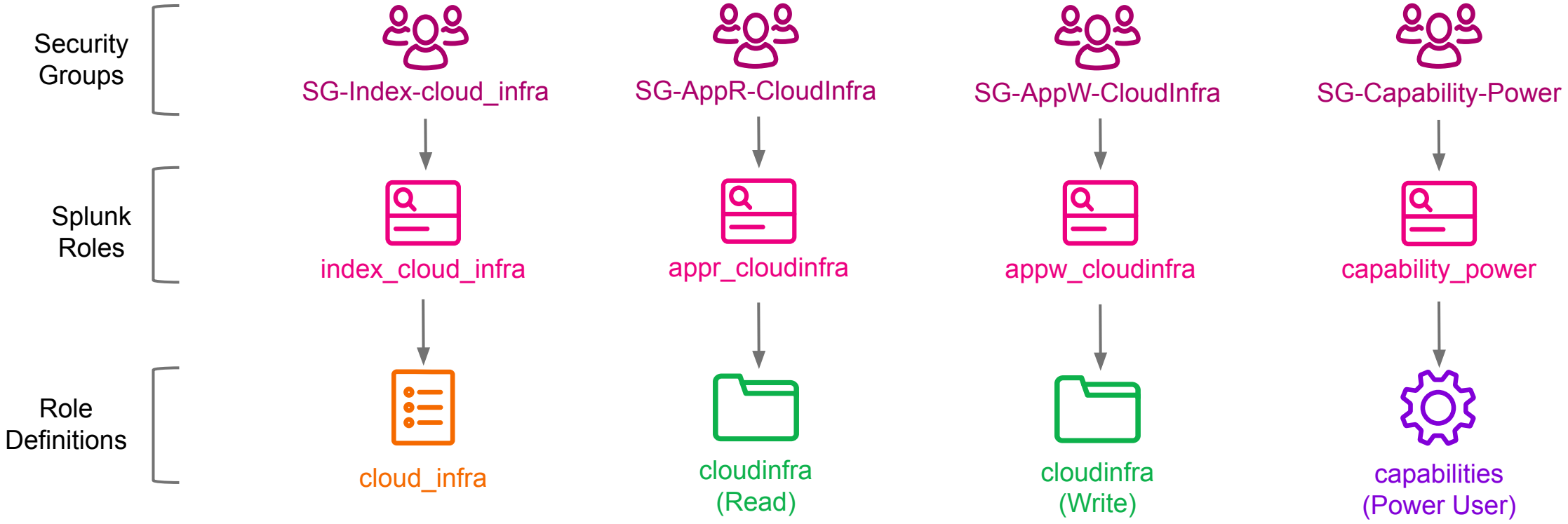
Read and write roles

Indexes

Each index has a role

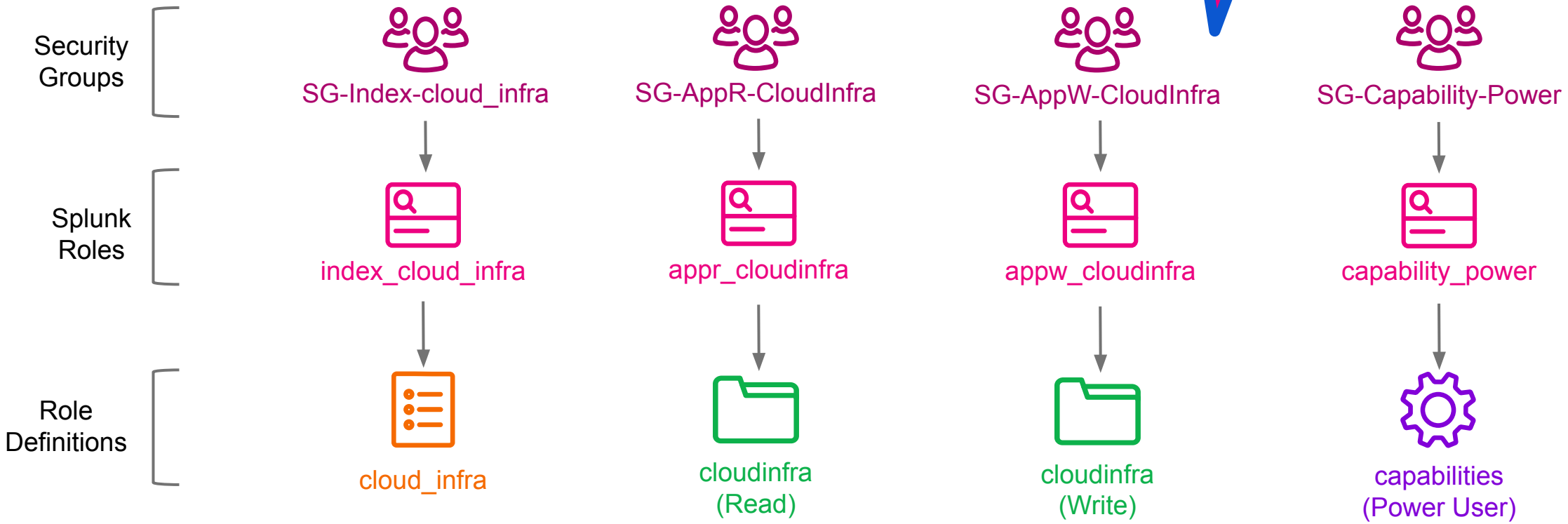
What We Ended Up Doing

…They suited up with new armor and weaponry…



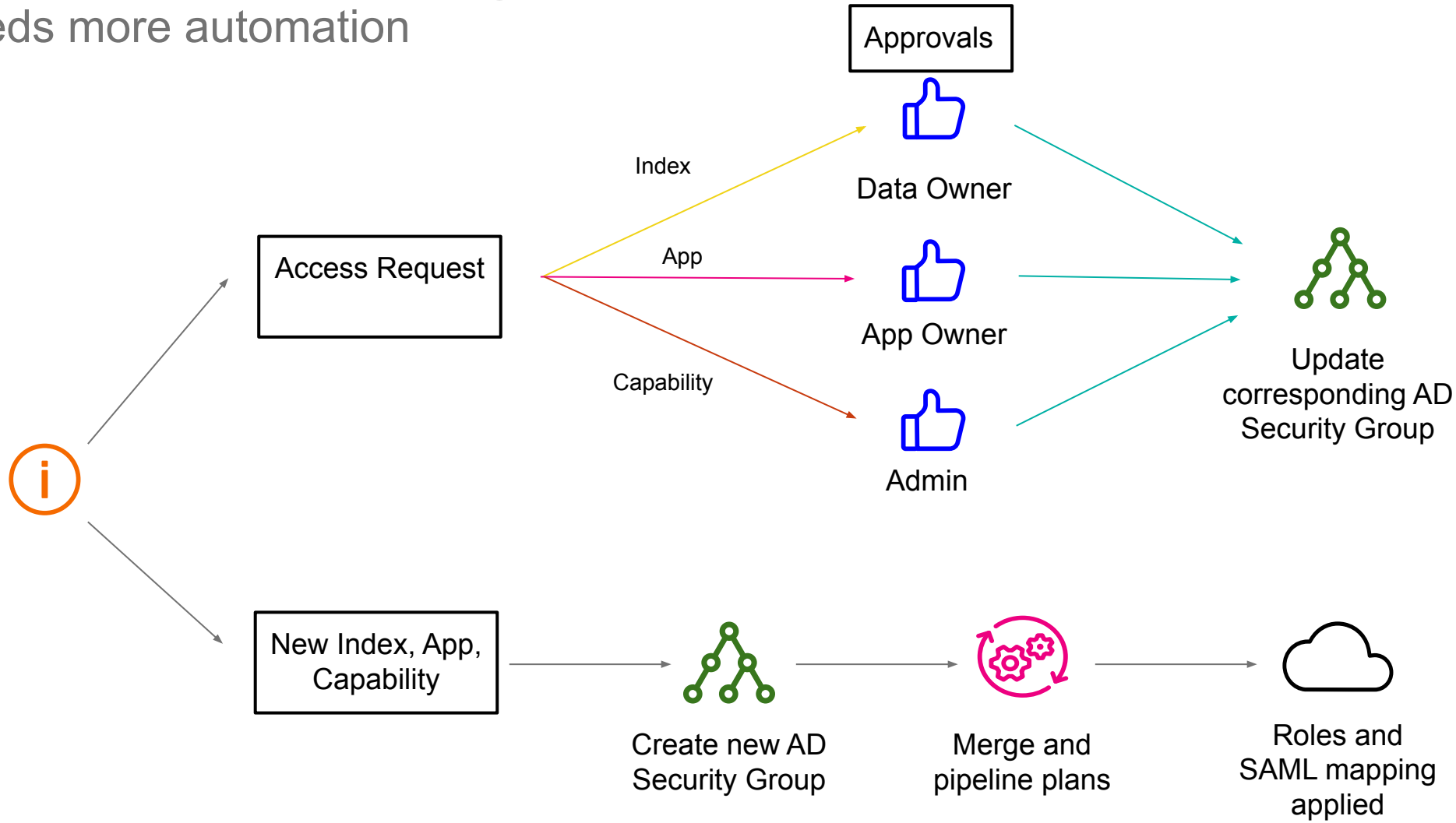
What We Ended Up Doing

…They suited up with new armor and weaponry…



Workflow Diagram

Needs more automation



authorize.yml 27.60 KiB

Open in Web IDE | Lock | Replace | Del

```
1  anchors:
4    role_default_capabilities:
5      capabilities: &role_default_capabilities
6        delete_messages: true
7        edit_own_objects: true
8        edit_search_schedule_window: true
9        export_results_is_visible: true
10       get_metadata: true
11       get_typeahead: true
12       input_file: true
13       list_accelerate_search: true
14       list_all_objects: true
15       list_inputs: true
16       list_metrics_catalog: true
17       list_tokens_own: true
18       pattern_detect: true
19       rest_apps_view: true
20       rest_properties_get: true
21       # capabilities can't really be disabled via REST API, so the [default] are explicitly enabled to avoid confusion in the
22       run_collect: true
23       run_mcollect: true
24       schedule_rtsearch: true
25       schedule_search: true
26       search: true
27   role_default_limits: &role_default_limits
28     # We need to have a default limit defined as an anchor due to the [default] stanza overriding roles that do not have them
29     cumulativeRTSrchJobsQuota: 1
30     cumulativeSrchJobsQuota: 50
31     rtSrchJobsQuota: 1
32     srchJobsQuota: 1
```

```
886   ##########################
887   ## RBAC Capability Roles ##
888   ##########################
889   # Every service role has the base user capabilities, so these are additive
890   - name: capability_admin
891     importRoles: [admin, sc_admin]
892     <<: *role_default_limits
893   - name: capability_phantom
894     capabilities:
895       phantom_read: true
896       phantom_write: true
897       <<: *role_default_capabilities
898     <<: *role_default_limits
899   - name: capability_power
900     # This has custom srchJobsQuota and cumulativeSrchJobsQuota, so isn't using role_default_svc_limits anchor, because it's taking precedence
901     capabilities: &role_capability_power_capabilities
902       metric_alerts: true
903       edit_log_alert_event: true
904       run_msearch: true
905       embed_report: true
906       output_file: true
907       run_sendalert: true
908       run_custom_command: true
909       upload_lookup_files: true
910       <<: *role_default_capabilities
911     cumulativeSrchJobsQuota: 100
912     srchJobsQuota: 5
913     cumulativeRTSrchJobsQuota: 1
914     rtSrchJobsQuota: 1
915   - name: capability_user
916     capabilities:
917       <<: *role_default_capabilities
923     <<: *role_default_limits
924   #capability_vo_user does not need capabilities, the role will be used for read/write on alert actions for VO
925   - name: capability_vo_user
926     capabilities:
927       <<: *role_default_capabilities
928     <<: *role_default_limits
```

splunk> .conf23

```
886   ##########################
887   ## RBAC Capability Roles ##
888   ##########################
889   # Every service role has the base user capabilities, so these are additive
890   - name: capability_admin
891     importRoles: [admin, sc_admin]
892     <<: *role_default_limits
893   - name: capability_phantom
894     capabilities:
895       phantom_read: true
896       phantom_write: true
897       <<: *role_default_capabilities
898     <<: *role_default_limits
899   - name: capability_power
900     # This has custom srchJobsQuota and cumulativeSrchJobsQuota, so isn't using role_default_svc_limits anchor, because it's taking precedence
901     capabilities: &role_capability_power_capabilities
902       metric_alerts: true
903       edit_log_alert_event: true
904       run_msearch: true
905       embed_report: true
906       output_file: true
907       run_sendalert: true
908       run_custom_command: true
909       upload_lookup_files: true
910       <<: *role_default_capabilities
911     cumulativeSrchJobsQuota: 100
912     srchJobsQuota: 5
913     cumulativeRTSrchJobsQuota: 1
914     rtSrchJobsQuota: 1
915   - name: capability_user
916     capabilities:
917       <<: *role_default_capabilities
923     <<: *role_default_limits
924   #capability_vo_user does not need capabilities, the role will be used for read/write on alert actions for VO
925   - name: capability_vo_user
926     capabilities:
927       <<: *role_default_capabilities
928     <<: *role_default_limits
```

```
886   ##########################
887   ## RBAC Capability Roles ##
888   ##########################
889   # Every service role has the base user capabilities, so these are additive
890   - name: capability_admin
891     importRoles: [admin, sc_admin]
892     <<: *role_default_limits
893   - name: capability_phantom
894     capabilities:
895       phantom_read: true
896       phantom_write: true
897       <<: *role_default_capabilities
898     <<: *role_default_limits
899   - name: capability_power
900     # This has custom srchJobsQuota and cumulativeSrchJobsQuota, so isn't using role_default_svc_limits anchor, because it's taking precedence
901     capabilities: &role_capability_power_capabilities
902       metric_alerts: true
903       edit_log_alert_event: true
904       run_msearch: true
905       embed_report: true
906       output_file: true
907       run_sendalert: true
908       run_custom_command: true
909       upload_lookup_files: true
910       <<: *role_default_capabilities
911     cumulativeSrchJobsQuota: 100
912     srchJobsQuota: 5
913     cumulativeRTSrchJobsQuota: 1
914     rtSrchJobsQuota: 1
915   - name: capability_user
916     capabilities:
917       <<: *role_default_capabilities
923     <<: *role_default_limits
924   #capability_vo_user does not need capabilities, the role will be used for read/write on alert actions for VO
925   - name: capability_vo_user
926     capabilities:
927       <<: *role_default_capabilities
928     <<: *role_default_limits
```

app-acls.yml 50.20 KiB

Open in Web IDE | Lock | Replace | Delete

```yaml
anchors:
  tag_action_acl: &tag_action_acl
    name: action
    values: ["acl"]

apps:
  - id: ws_Facilities
    name: Facilities Workspace
    tags:
      - *tag_action_acl
    acl:
      read: [zzz_access_appr_ws_facilities]
      write: [zzz_access_appw_ws_facilities]
      sharing: app
  - id: ws_GSCC
    name: GSCC Workspace
    tags:
      - *tag_action_acl
    acl:
      read: [zzz_access_appr_ws_gscc]
      write: [zzz_access_appw_ws_gscc]
      sharing: app
  - id: ws_Global_Sharing
    name: Global Sharing Workspace
    tags:
      - *tag_action_acl
    acl:
      read: ["*"]
      write: [admin]
      sharing: app
  - id: ws_ITAM
    name: IT Asset Management Workspace
    tags:
      - *tag_action_acl
    acl:
      read: ["*"]
      write: [zzz_access_appw_ws_itam]
      sharing: app
```

authentication.yml 29.42 KiB

Open in Web IDE | Lock | Replace | Delete

```
#################
## SAML GROUPS ##
#################
saml_groups:
  ############################
  ## CAPABILITY SAML GROUPS ##
  ############################
  - name: SG-Capability-Admin
    roles: [capability_admin]
  - name: SG-Capability-Can_Delete
    roles: [can_delete]
  - name: SG-Capability-Phantom
    roles: [capability_phantom,phantom]
  - name: SG-Capability-Power
    roles: [capability_power]
  - name: SG-Capability-VO_User
    roles: [capability_vo_user]

  ######################
  ## INDEX SAML GROUPS ##
  ######################
  - { name: SG-Index-appinspect, roles: [zzz_access_index_appinspect] }
  - { name: SG-Index-apps, roles: [zzz_access_index_apps] }
```

Challenges We're Facing

…and went to face the beast…

- Terraform state issues
- New features require updates to the provider
- API doesn't support all options
- Current automation doesn't separate environments
- Applications need to be removed from configurations when uninstalled

Challenges We're Facing

…and went to face the beast…

- Terraform state issues
- New features require updates to the provider
- API doesn't support all options
- Current automation doesn't separate environments
- Applications need to be removed from configurations when uninstalled

Challenges We're Facing

…and went to face the beast…

- Terraform state issues
- New features require updates to the provider
- API doesn't support all options
- Current automation doesn't separate environments
- Applications need to be removed from configurations when uninstalled

Challenges We're Facing

…and went to face the beast…

- Terraform state issues
- New features require updates to the provider
- API doesn't support all options
- Current automation doesn't separate environments
- Applications need to be removed from configurations when uninstalled

Challenges We're Facing

…and went to face the beast…

- Terraform state issues
- New features require updates to the provider
- API doesn't support all options
- Current automation doesn't separate environments
- Applications need to be removed from configurations when uninstalled

Improvements We'll Make

…And they all lived happily ever after…until next time…

- Automating AD security group creation and management
- Splitting environments
- Skipping uninstalled applications
- Self-service for access requests
- Field filtering
- Data governance and auditing

Improvements We'll Make

…And they all lived happily ever after…until next time…

- Automating AD security group creation and management
- Splitting environments
- Skipping uninstalled applications
- Self-service for access requests
- Field filtering
- Data governance and auditing

Improvements We'll Make

…And they all lived happily ever after…until next time…

- Automating AD security group creation and management
- Splitting environments
- Skipping uninstalled applications
- Self-service for access requests
- Field filtering
- Data governance and auditing

Improvements We'll Make

…And they all lived happily ever after…until next time…

- Automating AD security group creation and management
- Splitting environments
- Skipping uninstalled applications
- Self-service for access requests
- Field filtering
- Data governance and auditing

Improvements We'll Make

…And they all lived happily ever after…until next time…

- Automating AD security group creation and management
- Splitting environments
- Skipping uninstalled applications
- Self-service for access requests
- Field filtering
- Data governance and auditing

Improvements We'll Make

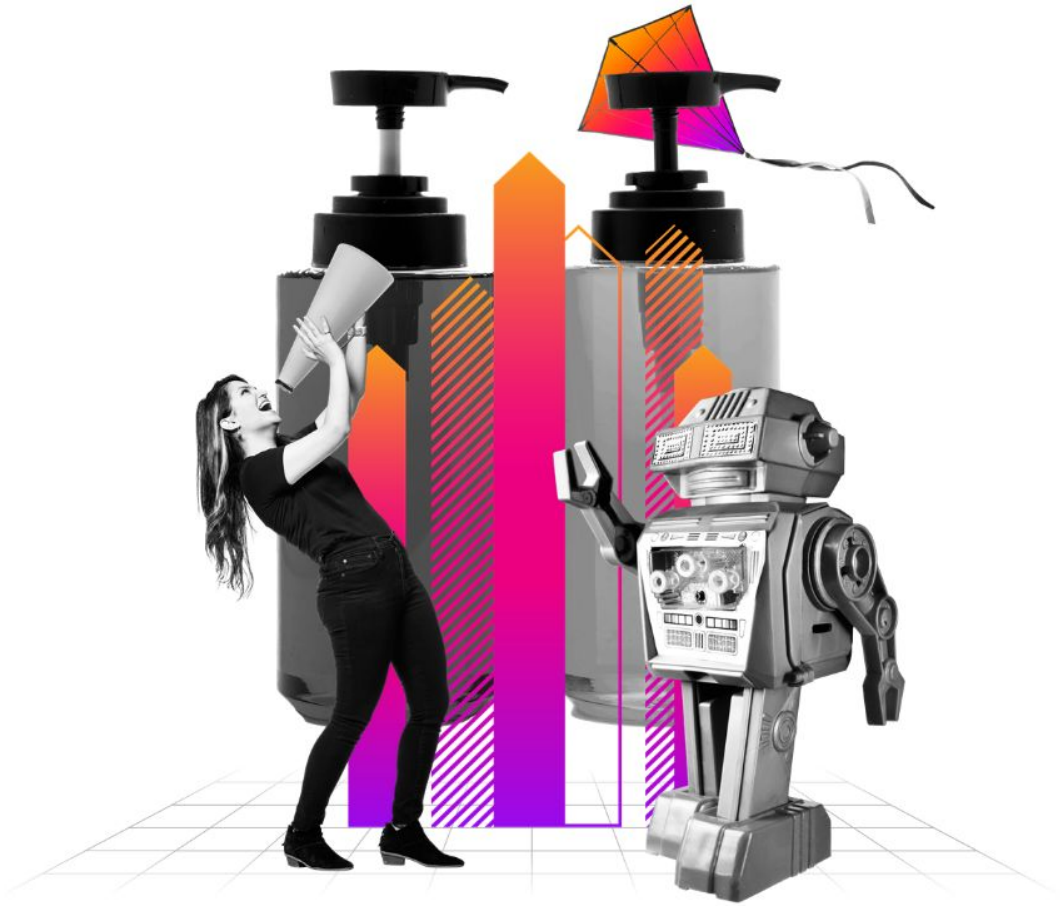
…And they all lived happily ever after…until next time…

- Automating AD security group creation and management
- Splitting environments
- Skipping uninstalled applications
- Self-service for access requests
- Field filtering
- Data governance and auditing

Strategy Changes

Where we were to where we are

Nested Model (Original)	3D Model (Current)
Importing roles	No Inheritance*
More permissive	Least permissive
Hard to troubleshoot	Easily understood
No automation	Automation
Fewer roles	Many roles



Considerations

Too Many Choices

Active Directory

Choice

Open

Automation

SSO

Apps

Granularity

Least Privilege

Design Considerations

Category	Questions to ask
Groups	• Should roles have individuals added to them or groups related to team or function? • How many groups need access to Splunk?

Design Considerations

Category	Questions to ask
Groups	• Should roles have individuals added to them or groups related to team or function? • How many groups need access to Splunk?
Roles	• Consolidated roles based on team or function or roles for each permission? • What job functions and responsibilities are encompassed under each role?

Design Considerations

Category	Questions to ask
Groups	• Should roles have individuals added to them or groups related to team or function? • How many groups need access to Splunk?
Roles	• Consolidated roles based on team or function or roles for each permission? • What job functions and responsibilities are encompassed under each role?
Permissions	• What resource limits should this role have? • What data should this role be allowed to see? • What do we want them to be able to do or not do?

Design Considerations

Category	Questions to ask
Groups	• Should roles have individuals added to them or groups related to team or function? • How many groups need access to Splunk?
Roles	• Consolidated roles based on team or function or roles for each permission? • What job functions and responsibilities are encompassed under each role?
Permissions	• What resource limits should this role have? • What data should this role be allowed to see? • What do we want them to be able to do or not do?
Inheritance	• Do roles already exist that other roles should have? • Is it possible to use anchors or references instead of importRoles?

Design Considerations

Category	Questions to ask
Groups	• Should roles have individuals added to them or groups related to team or function? • How many groups need access to Splunk?
Roles	• Consolidated roles based on team or function or roles for each permission? • What job functions and responsibilities are encompassed under each role?
Permissions	• What resource limits should this role have? • What data should this role be allowed to see? • What do we want them to be able to do or not do?
Inheritance	• Do roles already exist that other roles should have? • Is it possible to use anchors or references instead of importRoles?
Owners	• Will admins own roles and assignments or will stakeholders own their data and applications?

Design Considerations

Category	Questions to ask
Groups	• Should roles have individuals added to them or groups related to team or function? • How many groups need access to Splunk?
Roles	• Consolidated roles based on team or function or roles for each permission? • What job functions and responsibilities are encompassed under each role?
Permissions	• What resource limits should this role have? • What data should this role be allowed to see? • What do we want them to be able to do or not do?
Inheritance	• Do roles already exist that other roles should have? • Is it possible to use anchors or references instead of importRoles?
Owners	• Will admins own roles and assignments or will stakeholders own their data and applications?
Audit	• Is there a process to audit permissions assigned to users to ensure validity?

Design Considerations

Category	Questions to ask
Groups	• Should roles have individuals added to them or groups related to team or function? • How many groups need access to Splunk?
Roles	• Consolidated roles based on team or function or roles for each permission? • What job functions and responsibilities are encompassed under each role?
Permissions	• What resource limits should this role have? • What data should this role be allowed to see? • What do we want them to be able to do or not do?
Inheritance	• Do roles already exist that other roles should have? • Is it possible to use anchors or references instead of importRoles?
Owners	• Will admins own roles and assignments or will stakeholders own their data and applications?
Audit	• Is there a process to audit permissions assigned to users to ensure validity?
Automation	• Do you have the ability to automate RBAC?

Takeaways

- RBAC is a necessary evil
- Design RBAC thoughtfully
- Automate where possible
- Implement self-service

Resources

splunk> .conf23

Helpful Links

- PLA1945B: Factoring in SSO Integration: Setting up SAML in Your Splunk Environment - *Thursday, July 20 10:15 am - 11:00 am*
- Make It So! Unified, Simple Splunk Cloud Platform Administration Using Terraform https://conf.splunk.com/watch/conf-online.html?search=S4U1298C#/
- Splunk REST API Documentation https://docs.splunk.com/Documentation/SplunkCloud/latest/RESTREF/RESTaccess
- Splunk Terraform Documentation https://registry.terraform.io/providers/splunk/splunk/latest
- GitHub Splunk Terraform Provider https://github.com/splunk/terraform-provider-splunkconfig
- Now They See It, Now They Don't: Role Based Access Controls and Data Filtering In Splunk https://conf.splunk.com/watch/conf-online.html?search=TRU1713B#/
- Field Filtering https://docs.splunk.com/Documentation/Splunk/latest/Security/setfieldfiltering#Configuring_role-based_field_filters

splunk> .conf23

Thank You

splunk> .conf23