splunk> .conf2017

Splunking with Multiple Personalities

Extending Role Based Access Control to achieve fine grain security of your data

Sabrina Lea | Senior Sales Engineer, Splunk

Shaun C | Splunk Customer

September 2017

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. 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. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Who are we?

Sabrina

- 9 years in Government Cybersecurity and Data Analysis
- Splunk user for the past 5 years
- Second year as Splunk Engineer

Shaun

- 8 years in the security industry for public and private sector
- Splunk user for the past 4 years
- Based in the UK

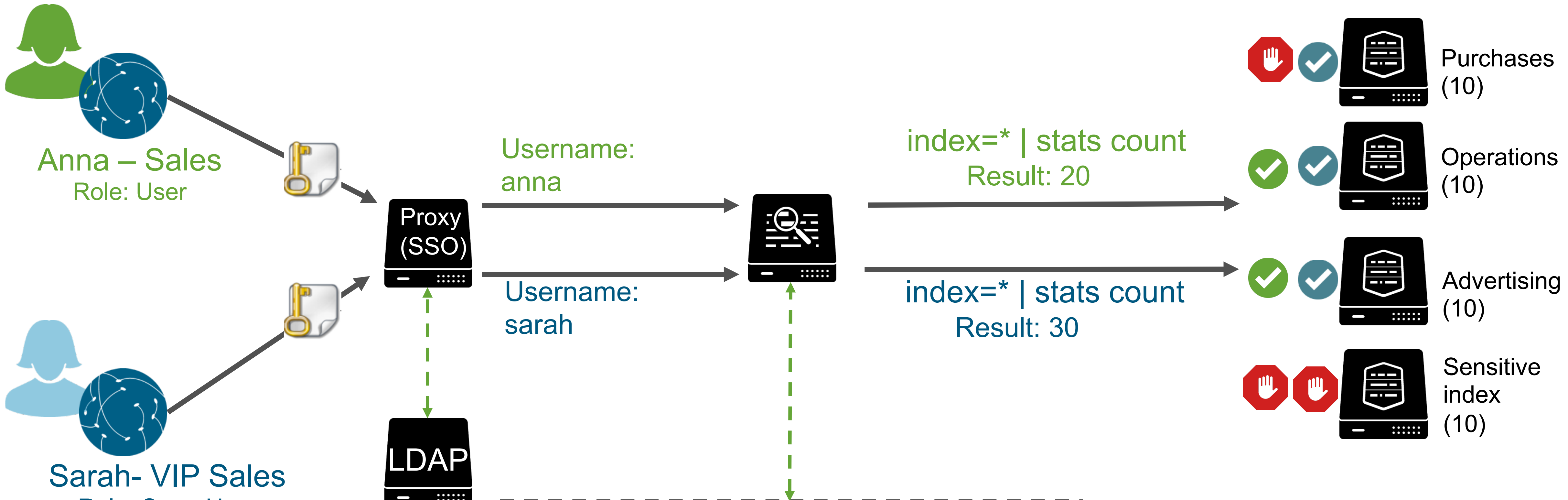
Setting the Scene

What is this presentation about, and when would I use this?

- Role Based Access Control (RBAC) in Splunk:
  - Split data into indexes
  - Restrict user access to specific indexes based on their role
- Two reasons you may need to extend RBAC:
  - Your data needs to be restricted at the event level, not the index level RBAC + Search Filter
  - You need user-level restriction, not role level (and too many users to create a role for each!) ABAC
- Where we see this requirement:
  - Mostly in Government and Finance
  - Other Industry/Requirement? Come talk to me after!

Standard Splunk Access Control: RBAC

How to restrict access at the index level



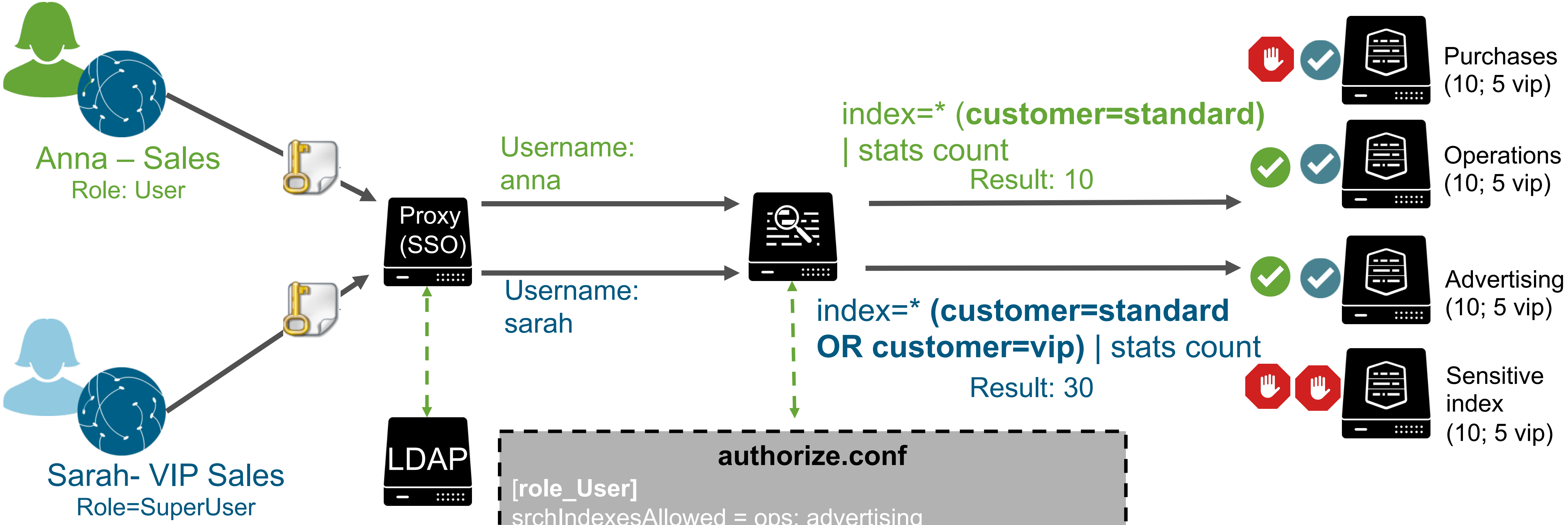
authorize.conf

```
[role_User]
srchIndexesAllowed = ops; advertising
srchIndexesDefault = ops; advertising

[role_SuperUser]
srchIndexesAllowed = purchases; ops; advertising
srchIndexesDefault = purchases; ops; advertising
```

Extension: RBAC with Search Filtering

How to restrict access at the event level AND the index level



Anna – Sales
Role: User

Sarah- VIP Sales
Role=SuperUser

Proxy (SSO)

LDAP

Username: anna

Username: sarah

index=* **(customer=standard)** | stats count
Result: 10

index=* **(customer=standard OR customer=vip)** | stats count
Result: 30

Purchases (10; 5 vip)

Operations (10; 5 vip)

Advertising (10; 5 vip)

Sensitive index (10; 5 vip)

authorize.conf

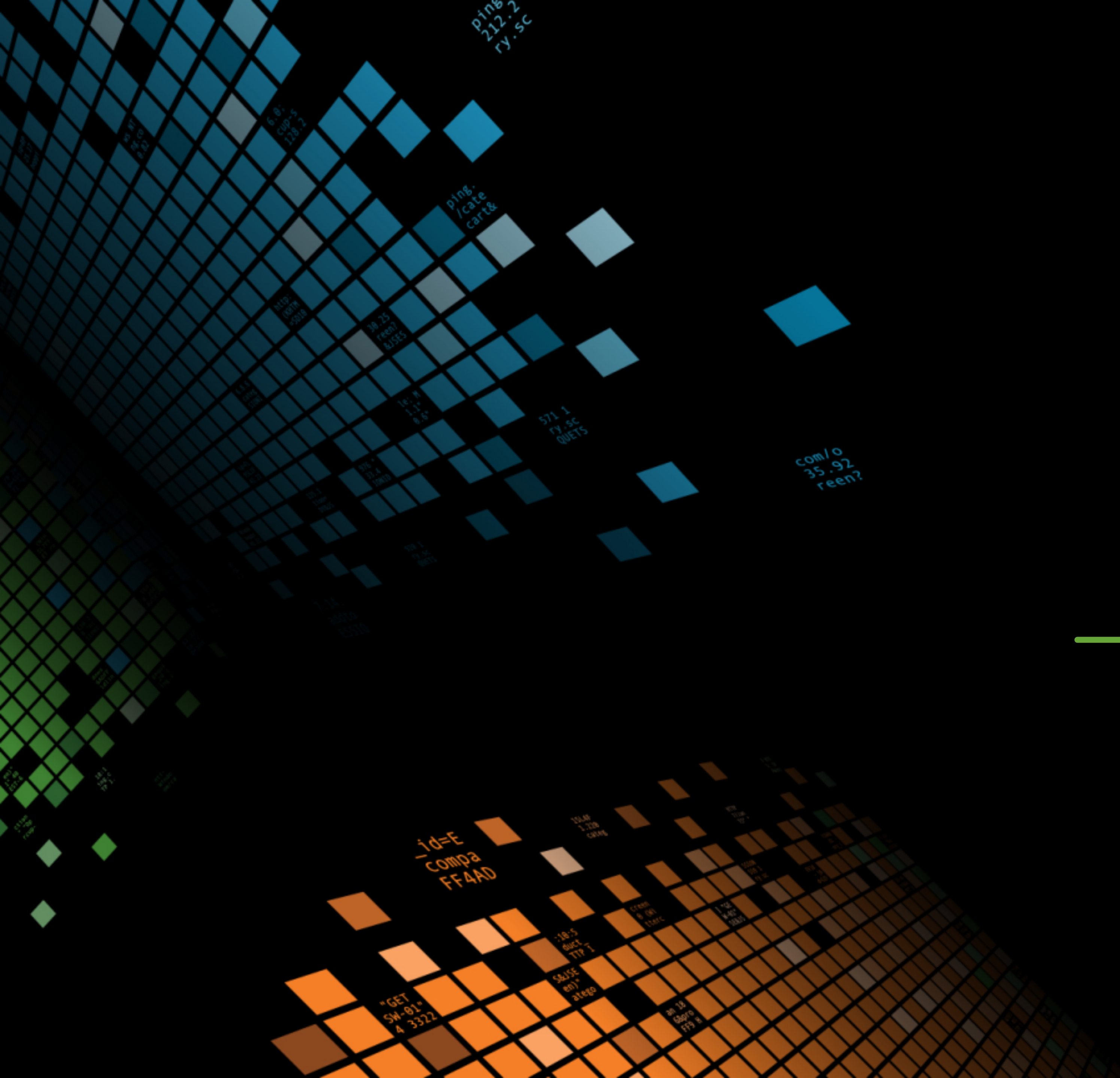
```
[role_User]
srchIndexesAllowed = ops; advertising
srchIndexesDefault = ops; advertising
srchFilter = (customer=standard)

[role_SuperUser]
srchIndexesAllowed= purchases; ops; advertising
srchIndexesDefault = purchases; ops; advertising
srchFilter = (customer=standard OR customer=vip)
```

RBAC + Search Filtering: Limitations

Things to remember if you try to implement this technique

- Requires all data is tagged with the fields you plan to use as filters
- Search filters can create a performance hit (hint: use indexed extractions and “::”)
  - Although sometimes they can improve performance
- Search filters are not applied to accelerated data models!
- Shared Reports typically run as owner, with search filtering they must run as user



Demo

RBAC + Search Filter

Introduction to ABAC

What is Attribute Based Access Control?

- Traditionally, access control has been based on the identity of a user requesting execution of a capability to perform an operation (e.g., read) on an object (e.g., a file), either directly, or through predefined attribute types such as roles or groups assigned to that user

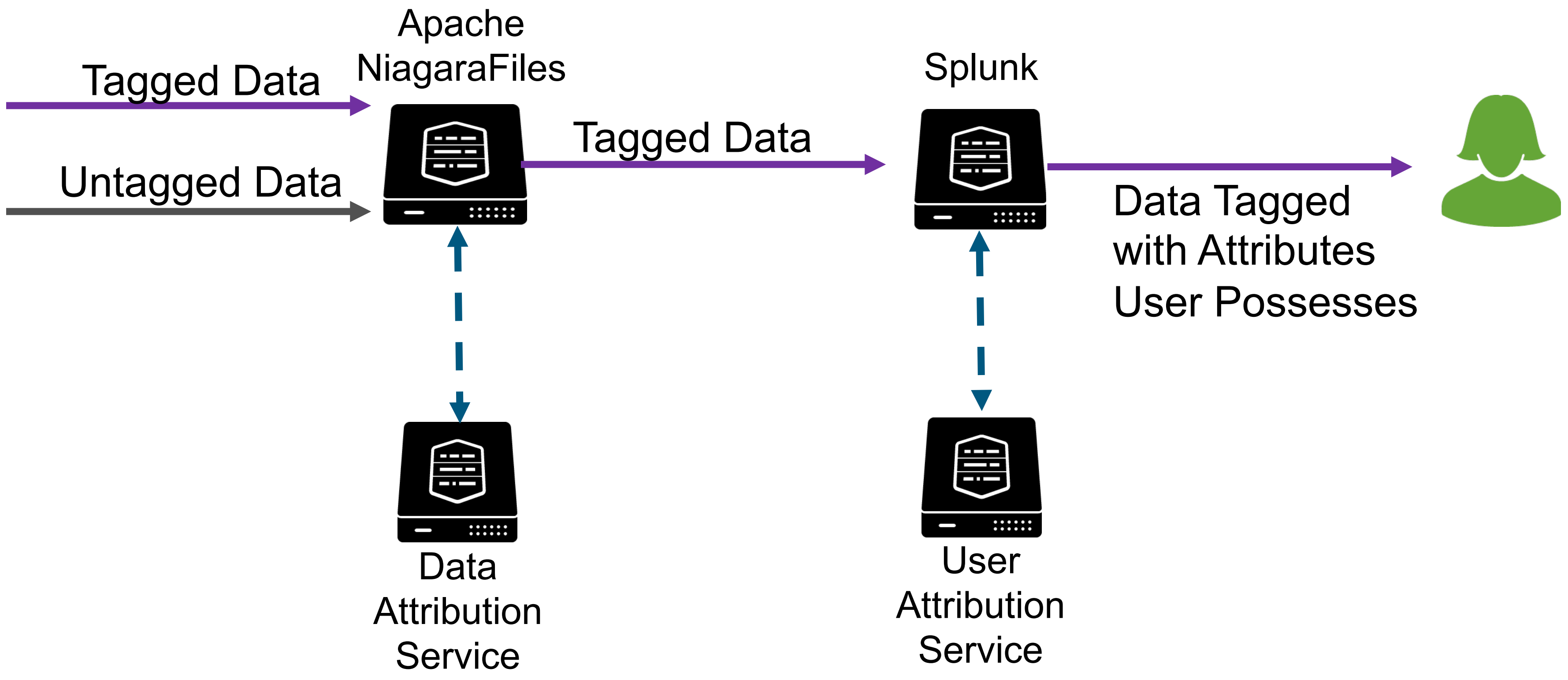
- An alternative is to grant or deny user requests based on arbitrary attributes of the user and arbitrary attributes of the object, and environment conditions that may be globally recognized and more relevant to the policies at hand. This approach is often referred to as ABAC.

- Source: NIST Special Publication 800-162
  Guide to Attribute Based Access Control (ABAC) Definition and Considerations
  http://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-162.pdf

Enterprise Data Header (EDH)

One requirement for developing an ABAC solution in Splunk

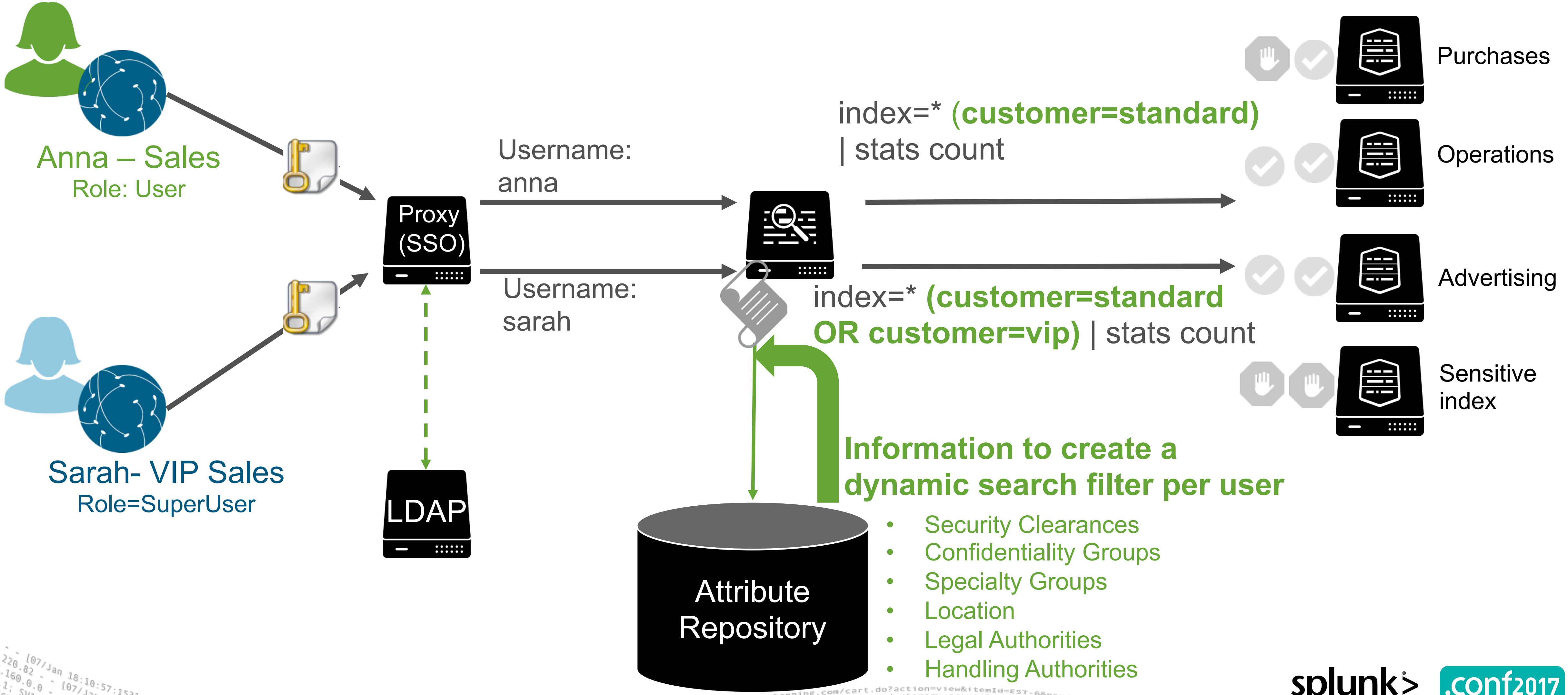
- Created by the Office of the Director of the National Intelligence (ODNI)



- https://www.dni.gov/index.php/who-we-are/organizations/ic-cio/ic-cio-related-menus/ic-cio-related-links/ic-technical-specifications/enterprise-data-header
- Apache NiFi

Implementing ABAC

Similar to RBAC + Search Filter: but it is user-based and more flexible



Anna – Sales
Role: User

Sarah- VIP Sales
Role=SuperUser

index=* **(customer=standard)** | stats count

index=* **(customer=standard OR customer=vip)** | stats count

Information to create a dynamic search filter per user

- Security Clearances
- Confidentiality Groups
- Specialty Groups
- Location
- Legal Authorities
- Handling Authorities

Using Splunk Scripted Authentication

The most dynamic way to implement ABAC in Splunk at this time

- **Three easy steps:**

1. **Create the authentication script**
   Samples of the authentication script:
   $SPLUNK_HOME$/share/splunk/authScriptSamples/
2. **Test the script**
   Instructions for command line testing:
   http://docs.splunk.com/Documentation/Splunk/latest/Security/Createtheauthenticationscript#Test_the_script
3. **Enable the script in authentication.conf**
   Instructions for enabling the script:
   http://docs.splunk.com/Documentation/Splunk/latest/Security/Editauthenticationconf

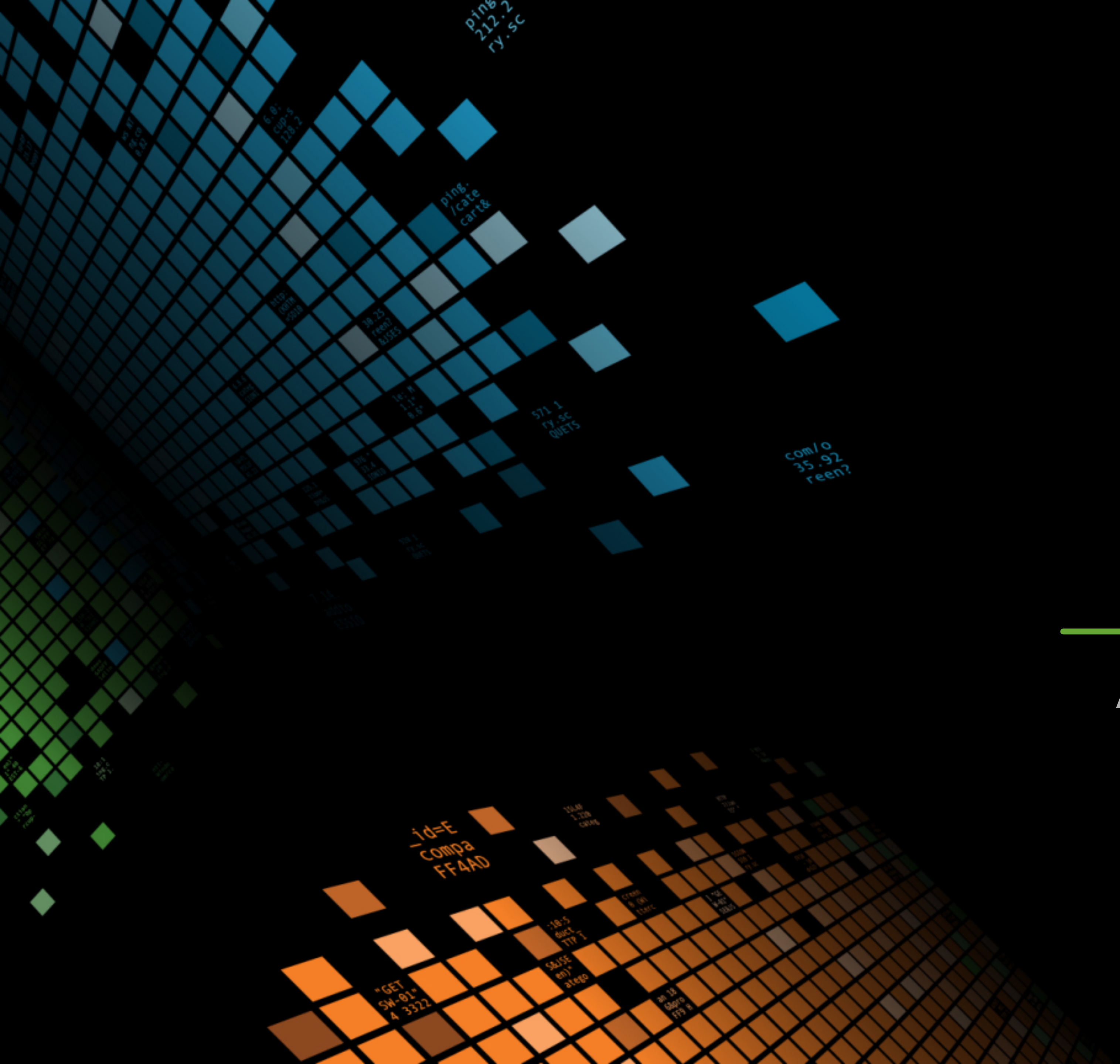
For more information: http://docs.splunk.com/Documentation/Splunk/latest/Security/Createtheauthenticationscript

splunk> .conf2017

Creating your Script

Four methods will be called by Splunk so they must be in your script

- **userLogin (username, password)**
  - Method is called once at user sign-on
  - If using SSO method should always return “false”– or have script interface with LDAP, Radius, etc
- **getUserInfo (username, password)**
  - Method is called repeatedly during authorized user session (can cache)
  - Should return this user’s full name and role
- **getUsers (username, password)**
  - Method is called intermittently when Splunkd is running (can cache)
  - Should return ALL users full name and role
- **getSearchFilter (username, password)**
  - Method is called when user searches
  - Should return your search filter, dynamically built for that specific user



Demo

ABAC using Scripted Authentication

getSearchFilter: Specify a Range

- …and (customerTier<=2)

Paula - Sales

Access : tier2customers

customerTier=1

customerTier=2

customerTier=3

getSearchFilter: Implement Boolean “OR”

- …and (label=customer OR label=vipCustomer)

Sarah - VIP Sales
Access : customers, vipcustomers

Label=customer
Label=customer
Label=vipCustomer
Label=specialCustomer

getSearchFilter: Implement Boolean “AND”

- …and (label=group1 OR label=group2 OR label=group1_group2)

Jane
Access : group1, group2

Label=group1
Label=group1_group2
Label=group2
Label=group3

getSearchFilter: Combine for your use case

Back to the Enterprise Data Header Requirement!

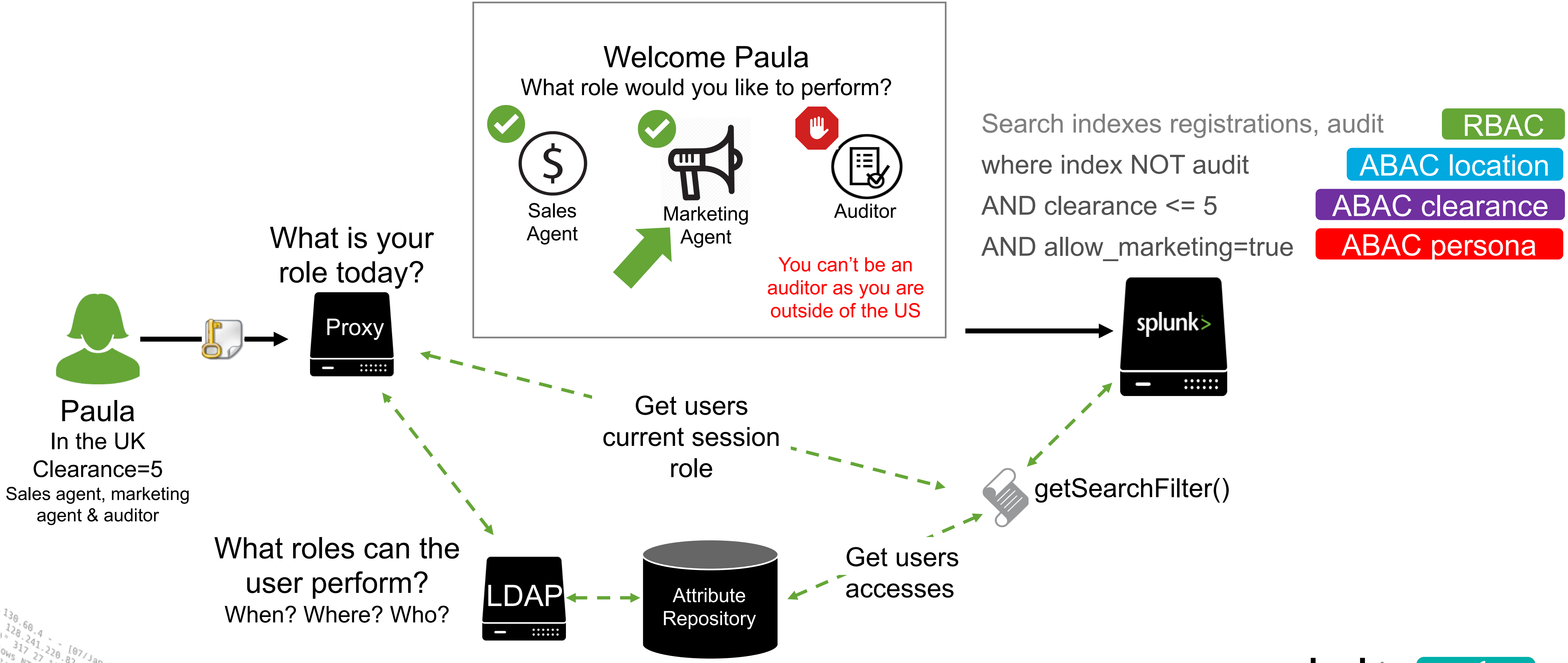
- …and
- securityClearance <= 4 AND
- (confidentiality=public OR confidentiality=confidential) AND
- (label=groupA OR label=groupB OR label=groupA_groupB)

Mandy
Security Clearance: Tier 4
Confidentiality: public, confidential
Specialty Groups: groupA, groupB

securityClearance=1, confidentiality=public, Label=group1
securityClearance=2, confidentiality=confidential, Label=group2
securityClearance=5, confidentiality=public, Label=group1
securityClearance=1, confidentiality=public, Label=groupC

Future Vision and Way Forward

Multiple Personality Searches: Users choose from roles they can and need to use



Welcome Paula

What role would you like to perform?

Sales Agent

Marketing Agent

Auditor

You can't be an auditor as you are outside of the US

Search indexes registrations, audit

where index NOT audit

AND clearance <= 5

AND allow_marketing=true

RBAC

ABAC location

ABAC clearance

ABAC persona

What is your role today?

Proxy

Paula

In the UK

Clearance=5

Sales agent, marketing agent & auditor

Get users current session role

getSearchFilter()

What roles can the user perform?

When? Where? Who?

LDAP

Attribute Repository

Get users accesses

Successes

- Implemented in production at 2 customer organizations
- Working at scale, in distributed/clustered environments
- Works with Splunk Analytics for Hadoop

Limitations

- All of the same limitations from “RBAC + Search Filtering” exist for ABAC!
- Additionally: User based search filters only work for **tstats** in the following versions:
  - 6.4.7+
  - 6.5.4+
- Research Areas:
  - Product change to address these limitations and streamline this from a “band-aid fix” to a solution?
  - This is, in essence, “row level security”– how do we get to “cell level security?”

Don't forget to **rate this session** in the .conf2017 mobile app

splunk> .conf2017