Attribute-Based Access Control...non solo ruoli in Azure access control!!

Saturday ChNova community

Sabato 30 settembre 2023

Massimo Bonanni

SPONSORS


altitudo

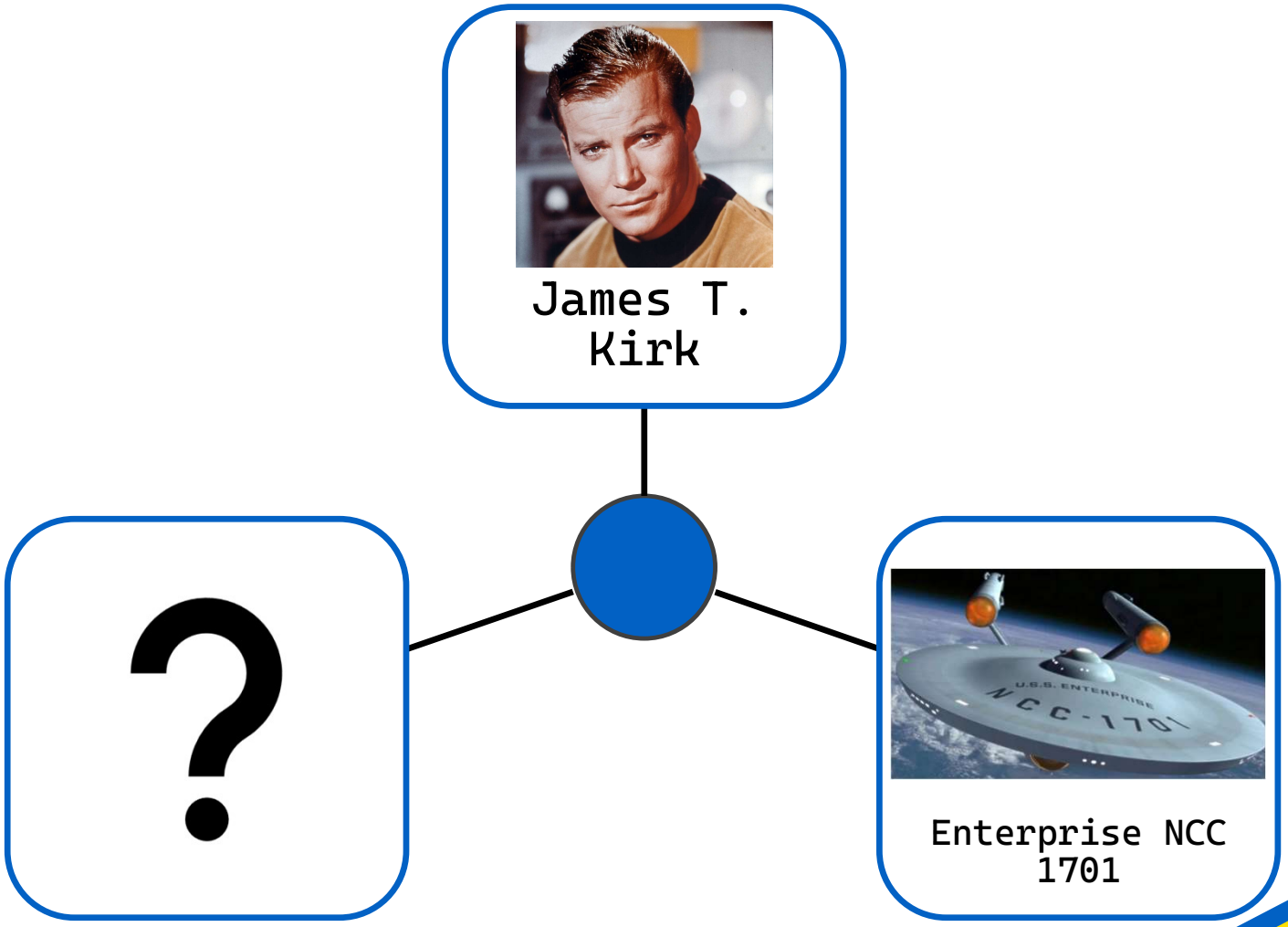

beanTech
IT moves your business


fluentis
the ERP ready to go live!


[stesi]
Powered by Innovation

RBAC

RBAC in Star Trek (Original Series)



RBAC in Star Trek (Original Series)

James T. Kirk

Who

What

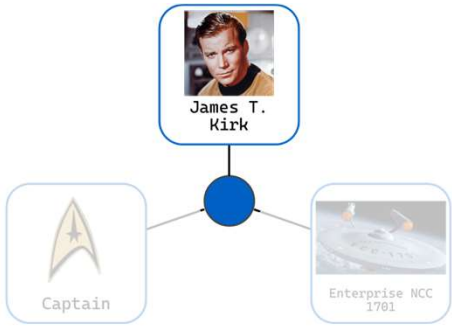
Where

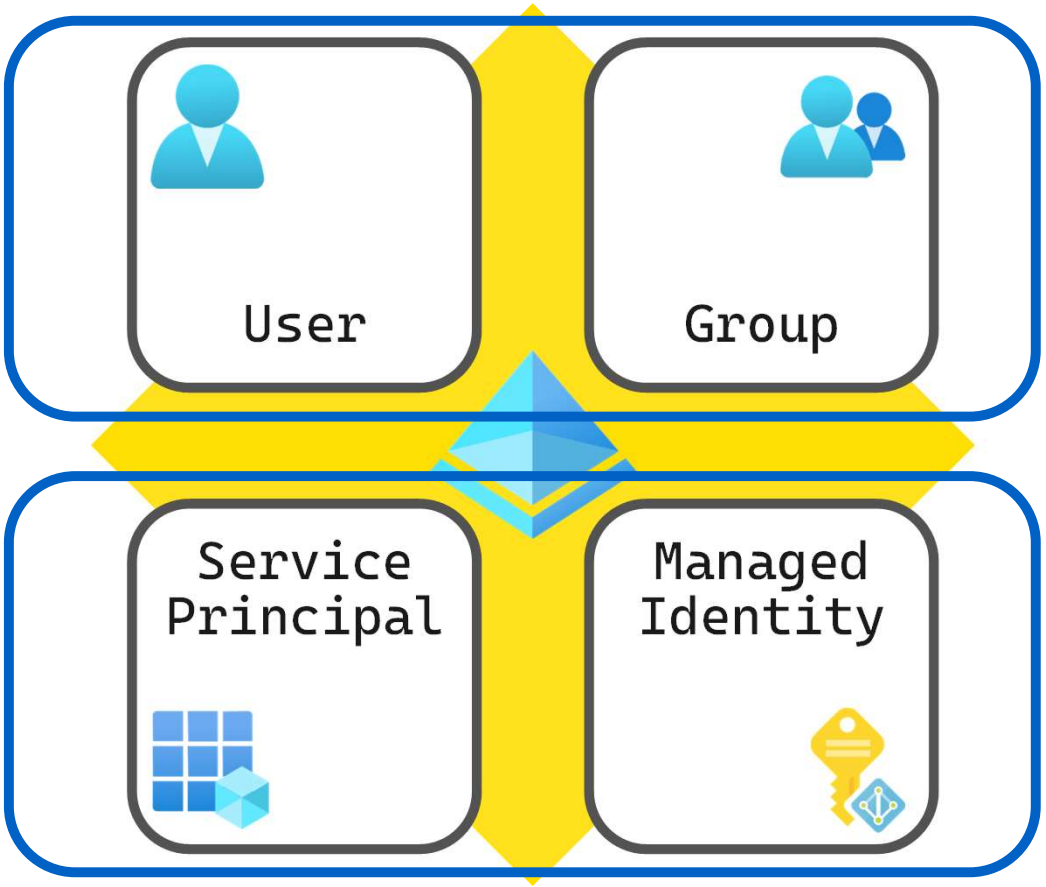
Captain

Enterprise NCC 1701

Azure Role-Based Access Control (RBAC) is an authorization system built on Azure Resource Manager that provides fine-grained access management of Azure resources.

Who – Security Principal





Interactive Identities

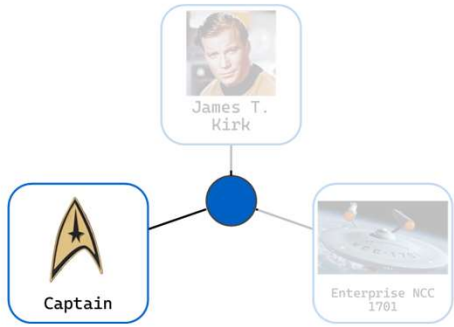
Applicative Identities

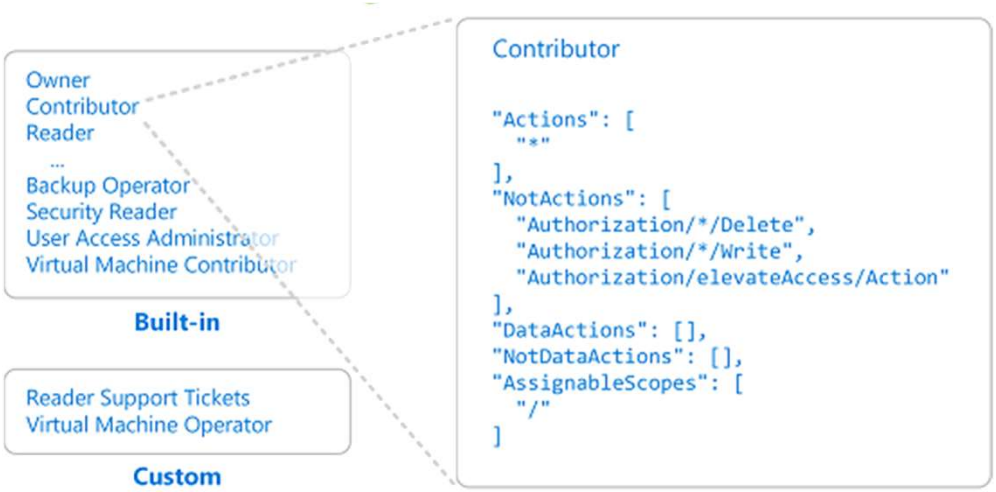
What – Role Definition

A role definition is a collection of permissions, a lists the **operations** that can be performed.

Roles can be high-level, like **owner**, or specific, like **virtual machine reader**.

Azure includes several **built-in roles** that you can use but you can create your **custom role**.



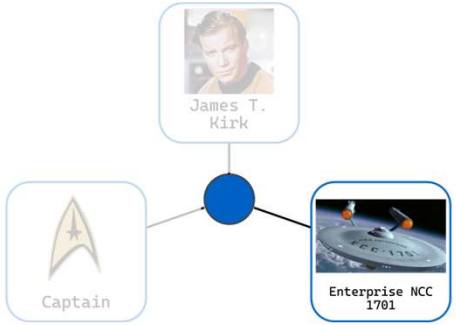


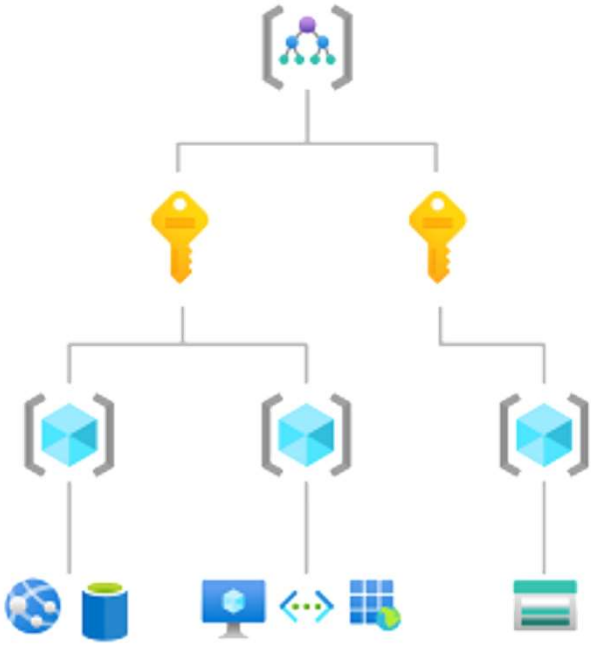
Where – Scope

Scope is the set of resources that the access applies to.

You can specify a scope at multiple levels:

- ✓ Management group
- ✓ Subscription
- ✓ Resource Group
- ✓ Resource (sub-resource)





RBAC in Star Trek (Original Series)

James T. Kirk

Identity

Role

Scope

Captain

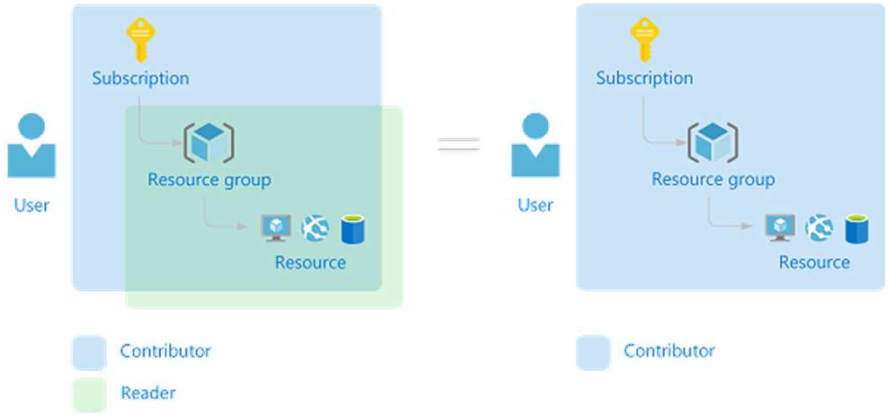
Enterprise NCC 1701

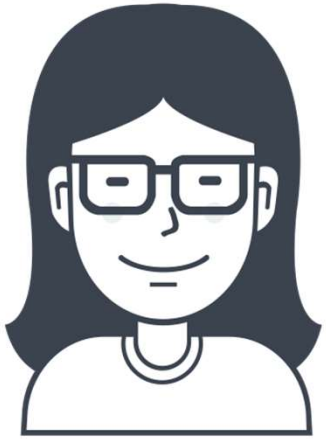
Multiple role assignments

Azure RBAC is an **additive model**.

Azure RBAC supports **deny assignments**.

A role assignment defines a set of actions that are **allowed**, while a **deny assignment** defines a set of actions that are **not allowed**.

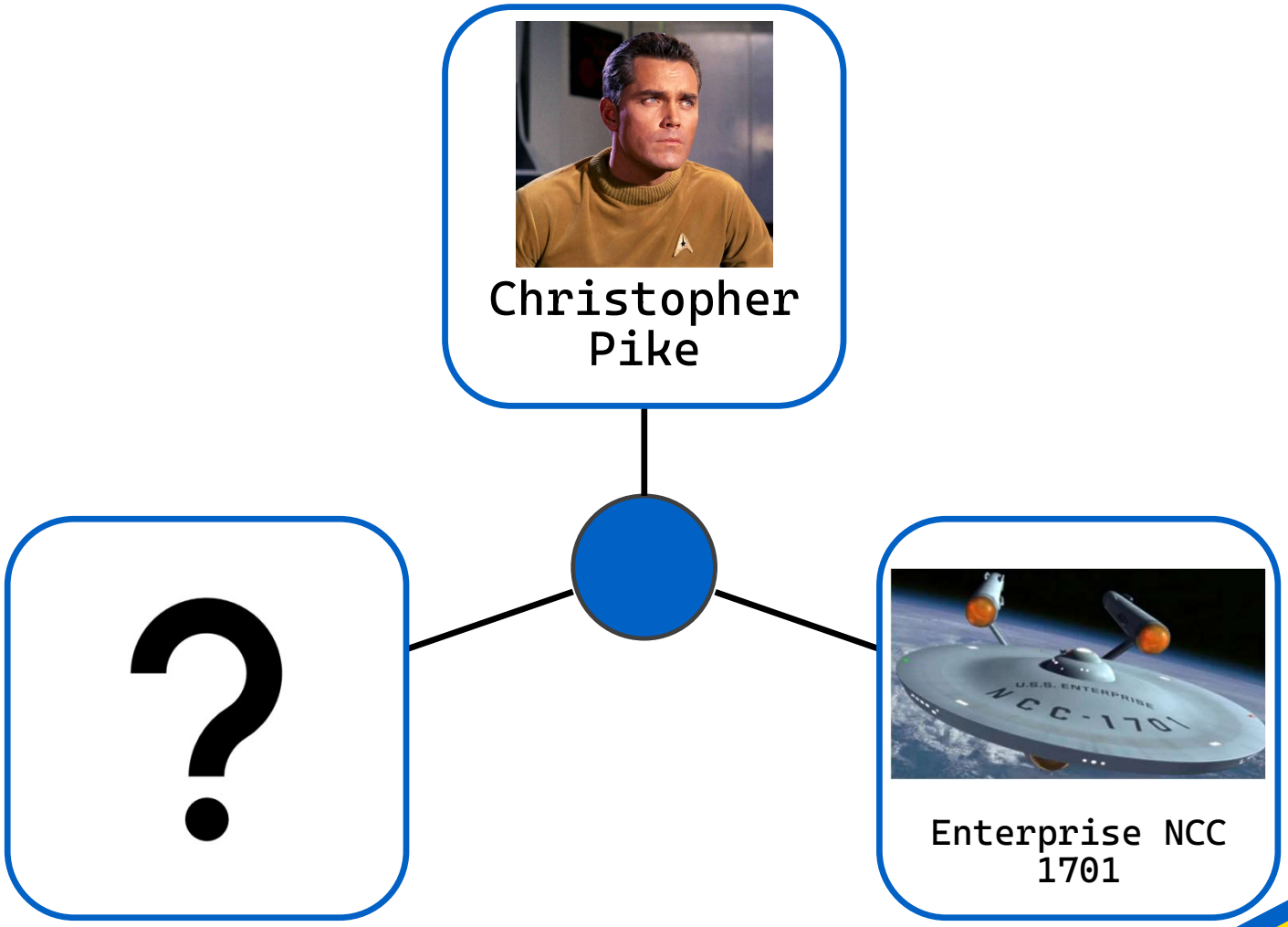




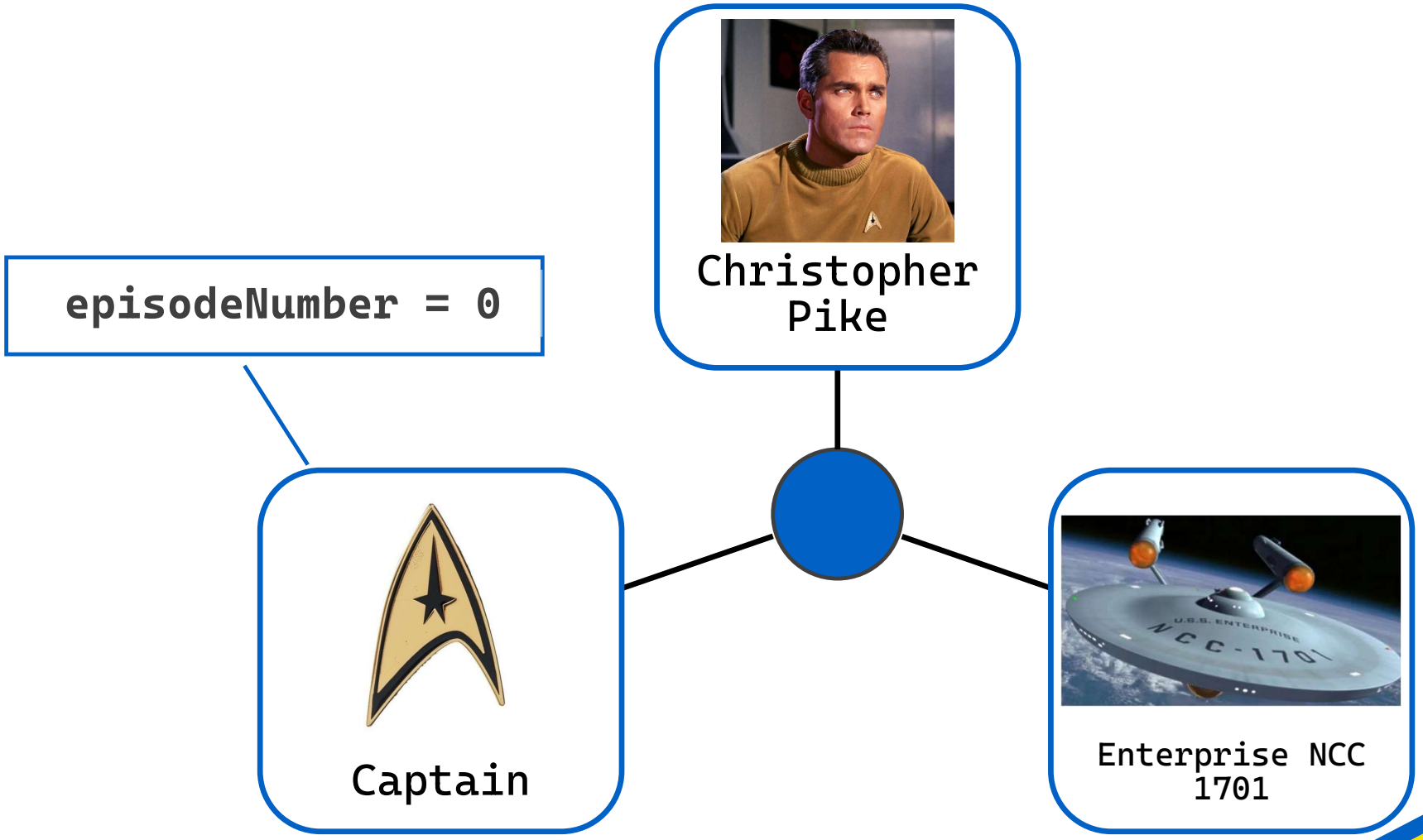
- ✓ **Jane** has been hired to manage the administration of the company, but she can read only the invoices.
- ✓ **Invoices** are stored in a **Storage Account**.
- ✓ Within the same storage, in the same container, there are also **reports** and **receipts**.
- ✓ Jane will have to manage documents using **custom software** or through the **Storage Explorer**

ABAC

ABAC in Star Trek (Original Series)

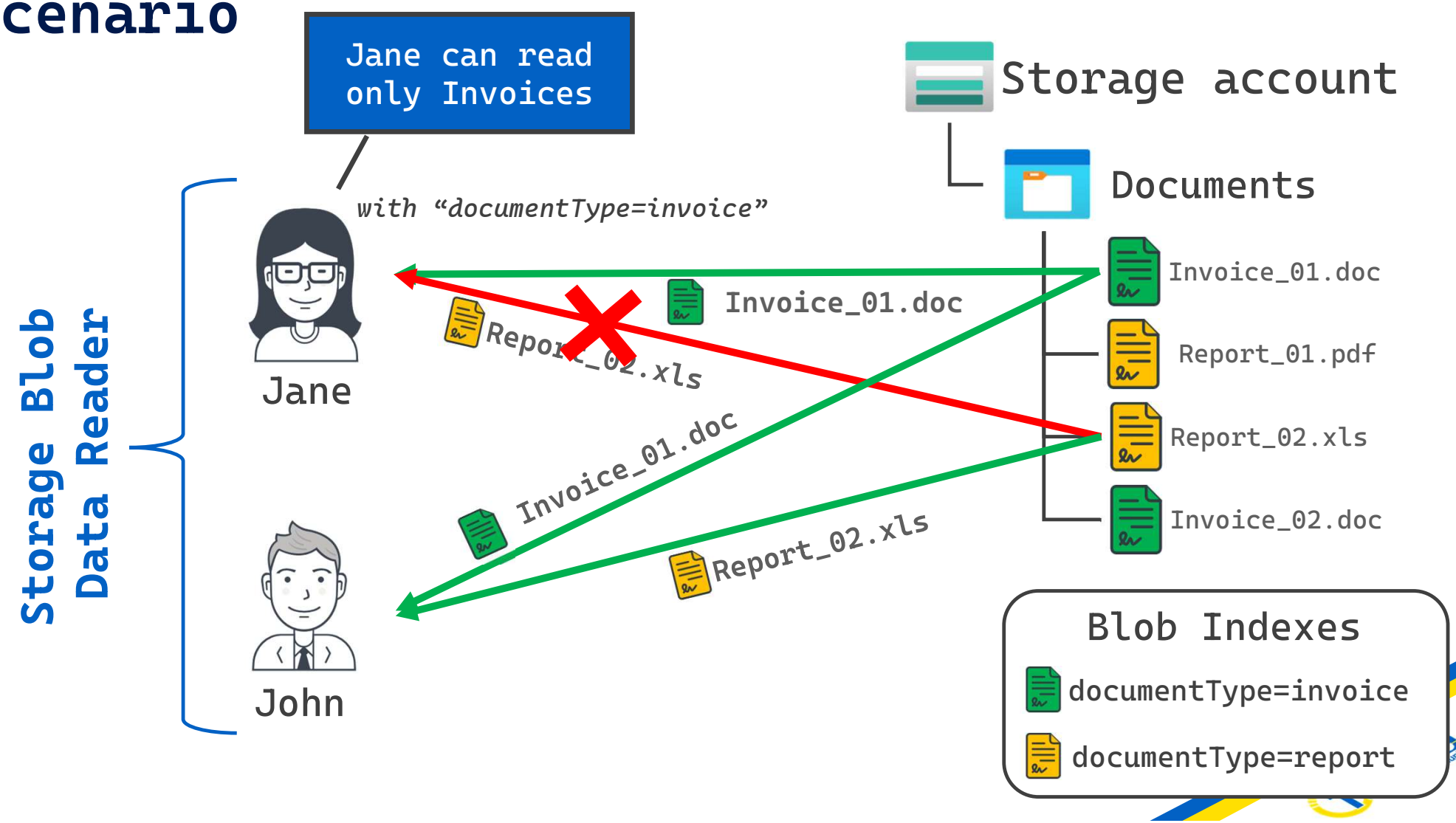


ABAC in Star Trek (Original Series)



Azure Attribute-Based Access Control (ABAC) builds on Azure RBAC by adding role assignment conditions based on principal, resource and request attributes.

Scenario


Jane can read only Invoices
Storage account
Documents
with "documentType=invoice"
Storage Blob Data Reader
Jane
John
Invoice_01.doc
Report_02.xls
Invoice_01.doc
Report_02.xls
Invoice_01.doc
Report_01.pdf
Report_02.xls
Invoice_02.doc
Blob Indexes
documentType=invoice
documentType=report

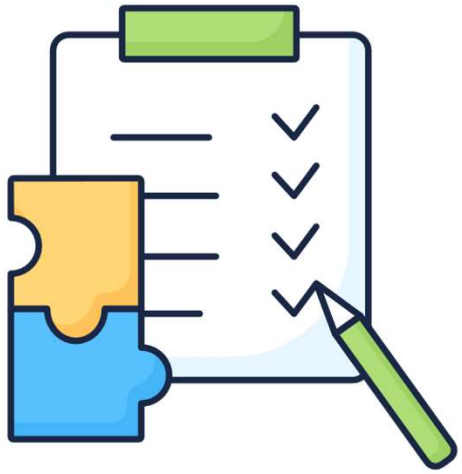
Conditions

You can configure conditions on role assignments for **DataActions** to achieve these goals.

You can add conditions to **built-in roles** or **custom roles**.

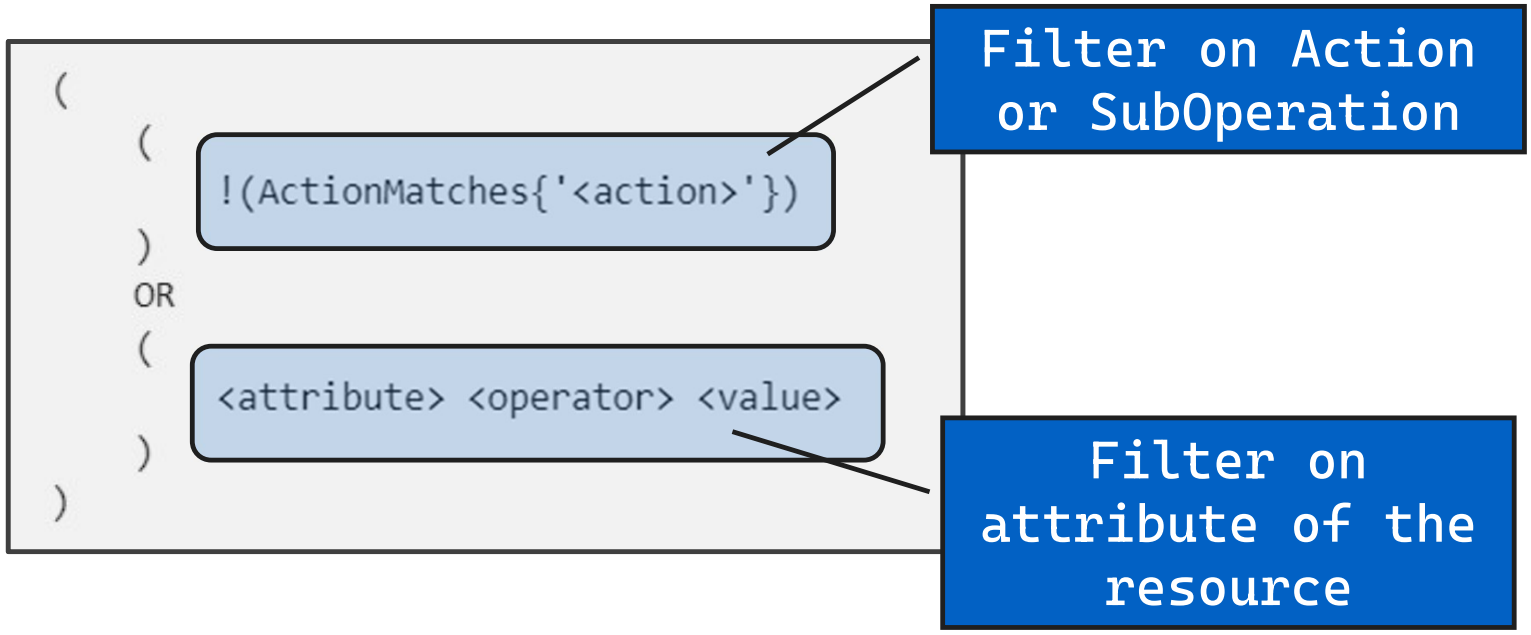
The built-in roles on which you can use role-assignment conditions include:

- Storage Blob Data Reader
- Storage Blob Data Contributor
- Storage Blob Data Owner



Condition format and syntax

A condition is an additional check that you can optionally add to your role assignment to provide more fine-grained access control.



Condition sample

Storage Blob Data Contributor	Allows for read, write and delete access to Azure Storage blob containers and data
Storage Blob Data Owner	Allows for full access to Azure Storage blob containers and data, including assigning POSIX access control.
Storage Blob Data Reader	Allows for read access to Azure Storage blob containers and data
Storage Blob Delegator	Allows for generation of a user delegation key which can be used to sign SAS tokens

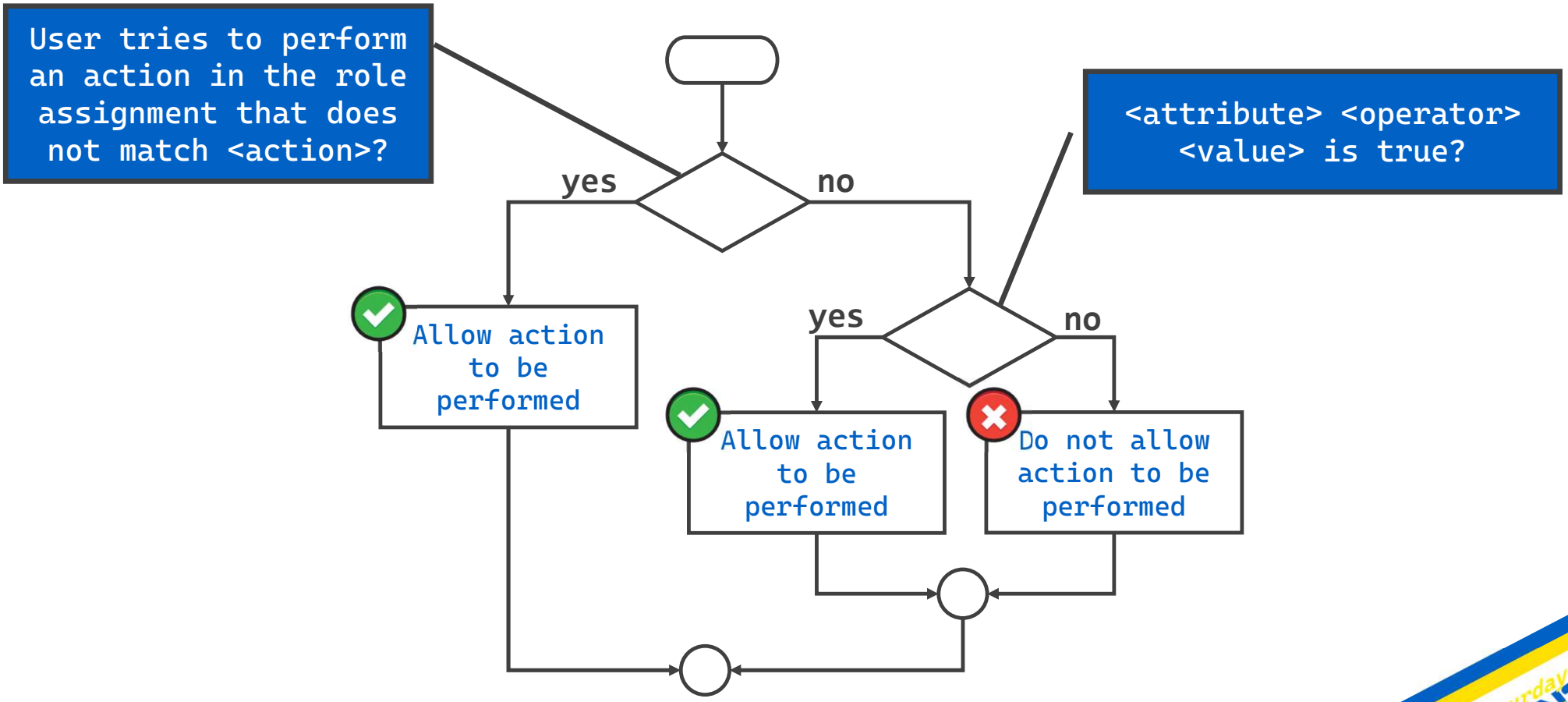
```
(
 (
  !(ActionMatches{'Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read'})
 )
 OR
 (
  @Resource[Microsoft.Storage/storageAccounts/blobServices/containers:name] StringEquals 'documents'
 )
)
```

The action requested by the user is not "*reading a blob*"

The name of the container the user want to access to is "*documents*"

The identity can execute all the operation except the **read** operation or read the blobs in the **documents** container.

How a condition is evaluated



Why use ABAC?

More fine-grained access control

You can write conditions to filter down RBAC permissions for more fine-grained access control.

Reduce the number of role assignments

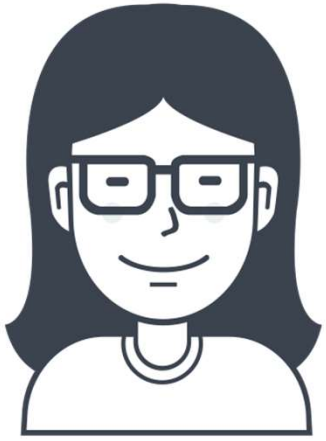
There are scenarios that would require a lot of role assignments, and all those role assignments would have to be managed.

In these scenarios, you could potentially add conditions to use significantly fewer role assignments.

Use attributes with business meaning

Conditions allow you to use attributes that have specific business meaning to you in access control.

Some examples of attributes are project name, software development stage, and classification levels.



- ✓ **Jane** has been hired to manage the administration of the company, but she can read only the invoices.
- ✓ **Invoices** are stored in a **Storage Account**.
- ✓ Within the same storage, in the same container, there are also **reports** and **receipts**.
- ✓ Jane will have to manage documents using **custom software** or through the **Storage Explorer**

Takeaways

RBAC vs ABAC

RBAC and ABAC are not competing technologies for access control.

Give access with RBAC and refine it with ABAC.

ABAC pros

Flexibility

Agility

Granularity

ABAC cons

Complexity

Only Blob Storage, Data Lake Gen2 and Storage Queue

GA for Standard Storage, preview for Premium Storage

Thank you for your attention!!!

Massimo Bonanni

Technical Trainer @ Microsoft

massimo.bonanni@microsoft.com

aka.ms/maxlinkedin

Thanks!

Sabato 30 settembre 2023

Marco Parenzan