Whitepaper: A non-proprietary hybrid integration layer that provides automation, data integration and transformation required for Attribute Based Access Control (ABAC) in support of ICAM modernization

Introduction

The current manual implementation of the more than 42 United States Air Force (AF) Identity Credential Access Management (ICAM) systems presents several problems around security, operational efficiency, accuracy, and sustainment and contribute to undermining the effectiveness of the warfighter as well as the civilian employees and contractors that support them. Of great benefit would be the automation of the processes related to granting and changing access to systems as Airman, civilian employees, veterans and contractors are onboarded, change jobs, change locations, and depart the AF or contracts end.

Many of these problems have been identified and are well understood which is why the AF in partnership with the Defense Information Systems Agency (DISA) are working to modernize ICAM across the AF as well as across the Department of Defense (DoD).

There are three primary components of AF ICAM systems: Identity management, credential management, and access management. This whitepaper relates specifically to the application of the PrivOps Matrix hybrid integration layer to support policy-based automation of the access management component of ICAM, a goal for both AF and DISA teams charged with ICAM modernization. Also known as Attribute Based Access Control (ABAC), policy-based access management automation solves several significant problems for the AF:

- Delays in providing access to needed systems and information reduce the ability for airmen, civilian employees, and contractors to do their jobs effectively because of an error prone, manual system. This problem is exacerbated by the fact that personnel frequently change roles.
- Security risks increase when system users continue to have access to systems they no longer need to access.
- Operational inefficiencies exist due to an error prone, manual processes and redundant systems for access management.

One of the biggest challenges of implementing ABAC automation for the AF is the complexity introduced by the fact that data from so many systems must be integrated, scrubbed, and transformed to be used. For example, the AF's Forms and Account Management Service (FAMS) system contains contract data that includes which contractors are assigned to which projects. That data must be extracted and combined with data from the Defense Enrollment Eligibility Reporting System (DEERS), the Military Personnel Data System (MPDS) and other authoritative data sources, validated and scrubbed for errors/conflicts, and mapped to roles that guarantee personnel have access to only the systems needed to do their jobs, and in a way that efficiently supports lifecycle management as personnel are onboarded, offboarded and change projects and roles.

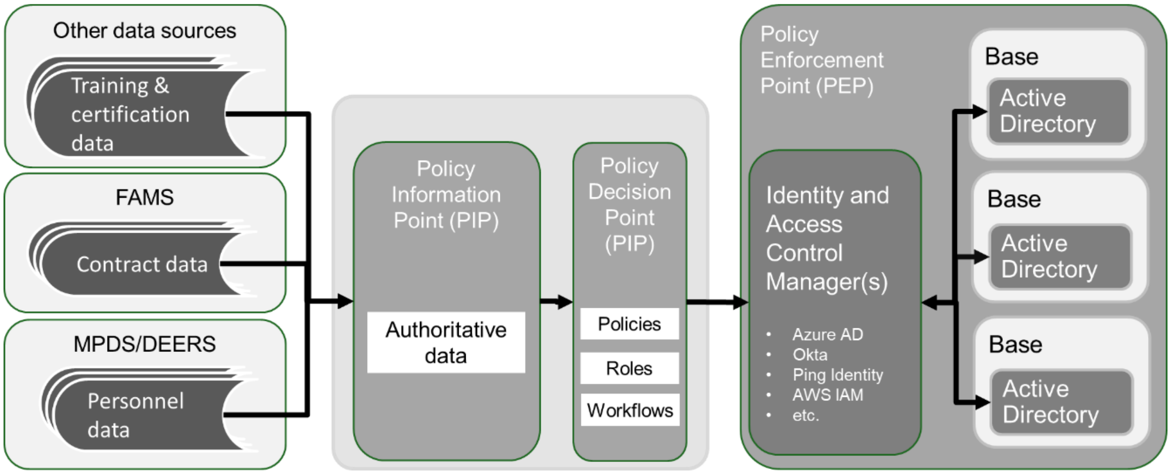


FIGURE 1: AN ABAC AUTOMATION SOLUTION

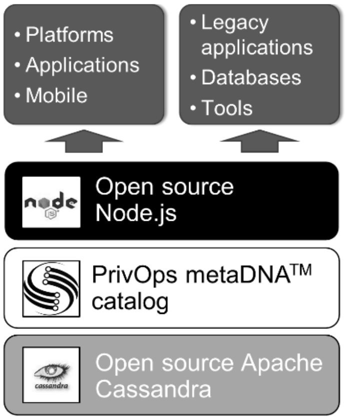
ABAC Solution Overview

There are three primary components of an ABAC architecture: The Policy Enforcement Point (PEP) which is the point where authorization to an application (or data) is permitted or denied, the Policy Decision Point (PDP) which is a decision engine that evaluates incoming events (access requests, role changes, security breaches, etc.) against access policies, and the Policy Information Point (PIP) which is the data integration pipeline required to connect sources of attributes (or claims) data to map to access policies. While PEP is out of scope for the proposed feasibility study, the PDP policy decision engine and the PIP data integration pipeline are in scope; there are significant implementation challenges for each. For the PDP policy decision engine, there has to be a way to automate the application of policies to create context based roles, mapping those context based roles to user attributes and a way to simplify the creation and change management of policies For PIP, data integration must scale across multiple heterogeneous data sets containing attributes in a way that minimizes rework if the underlying systems change and the attributes must be automatically validated and mapped to a central common set of attributes.

Technology Overview

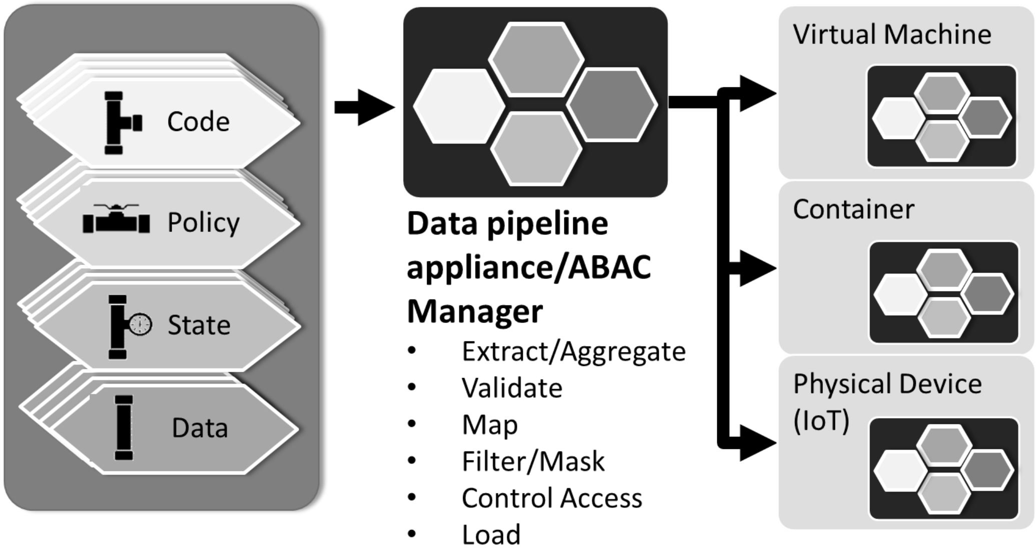
Traditional IT technology procurement works against modularity. Simply put, it is in the financial interest of large Commercial Off-the-Shelf (COTS) vendors and some technology consultants to create proprietary data structures and interfaces that make it difficult to be replaced when contracts end. For that reason, any attempt at automating data intensive processes like ABAC should require open source as a central element of its architecture. We designed the PrivOps Matrix^{TM} as an open platform for making the process of rapidly creating and deploying data intensive process automations, i.e. data pipelines, scalable with a hot pluggable architecture that makes it possible to switch out COTS and custom software interchangeable components as soon as better ones are available.

FIGURE 2: THE PRIVOPS MATRIX^{TM}



The PrivOps Matrix^{TM} hybrid integration fabric is a modular system for mass producing data pipelines, including the PIP and PDP data pipeline required for ABAC automation. It is built from three components: Apache Cassandra is an open-source distributed database that provides massive scalability and resilience with its distributed architecture; Node.js is an open-source software platform that scales integration and development with thousands of free connectors and modules; and the PrivOps Matrix^{TM} software accelerates and scales the automation, protection, and control of data. Since the Matrix software is the only non-open-source component, we license the source code, making the data fabric an entirely non-proprietary system that serves as the foundation for a best-of-breed, agile approach to data integration platform development.

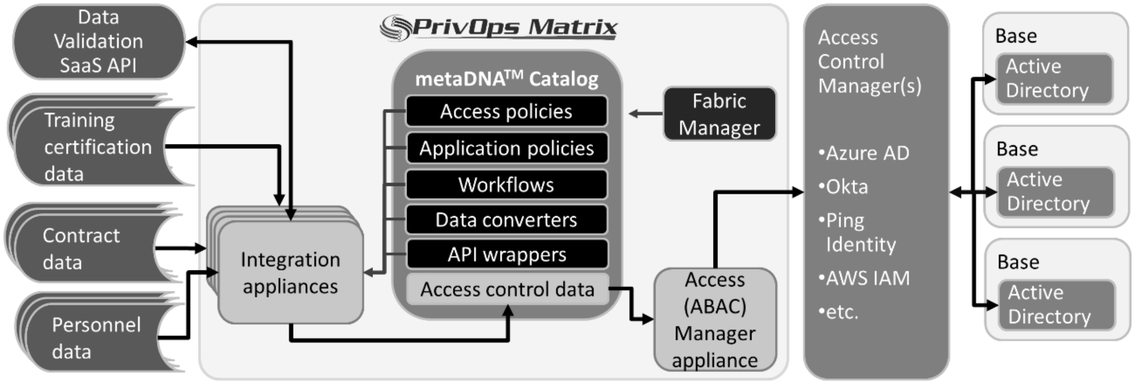
FIGURE 3: USING THE METADNA^{TM} CATALOG TO COMPOSE DATA PIPELINES



The key innovation in the PrivOps Matrix^{TM} is the patented metaDNA^{TM} Catalog. Just as biological structures are built from DNA molecules, digital structures (data pipelines) are built from data objects in the metaDNA^{TM} Catalog. The PrivOps Matrix^{TM} contains the data pipelines themselves as well as the metaDNA^{TM} Catalog and software that manages the Catalog and the data pipelines.

All components are built as microservices or “appliances”. These appliances are then installed on containers, virtual machines, serverless instances and/or physical servers, either directly or using container orchestration tools like Red Hat OpenShift or Pivotal Cloud Foundry. Appliances then connect to the Cassandra cluster which then connects to other clusters in other cloud environments, traditional datacenters, and edge devices in a distributed database architecture that provides for resilience, data transmission, and other capabilities related to scaling and protection (encryption).

FIGURE 4: PROPOSED ARCHITECTURE - ABAC AUTOMATION WITH THE PRIVOPS MATRIX^{TM}



ABAC Automation with the PrivOps Matrix^{TM}

Figure 4 above depicts a high-level view of the ABAC automation data pipeline in the proposed solution. Multiple ABAC Manager microservices, each customized to the source data are constructed from components in the metaDNA^{TM} Catalog. As Figure 5 below shows, at runtime each ABAC manager first pulls policies and software code for mapping and conversion of data. Since it is at runtime, the workflow can pull different API wrappers depending on the cloud environment, making this a multi-cloud solution. Once initialized, the ABAC manager then ingests data from the data source and maps attributes via a regular expression matching engine to the attributes of a common authoritative data set consisting of users, group and roles, maps users to roles, and roles to groups. Access is based on context which includes attributes like location (base assignment) as well as role. Context rules are governed by policies as well. Once created, authoritative access control data is then used to update the access control manager. The solution is cloud native; Figure 6 depicts a physical implementation on CloudONE.

FIGURE 5: ABAC AUTOMATION WORKFLOW - MAP ATTRIBUTES TO GROUPS AND ROLES, UPDATE ACCESS MANAGER

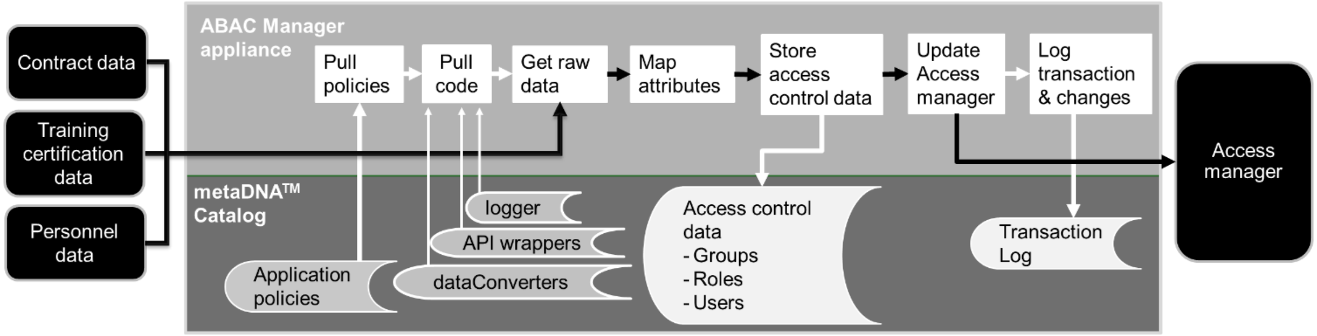
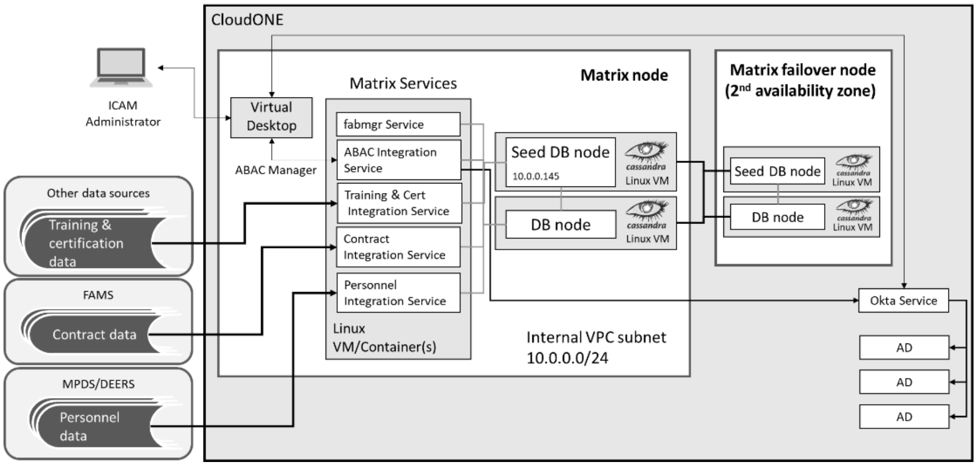


FIGURE 6: ABAC PHYSICAL ARCHITECTURE


CloudONE
ICAM Administrator
Virtual Desktop
ABAC Manager
Matrix Services
Matrix node
fabmgr Service
ABAC Integration Service
Training & Cert Integration Service
Contract Integration Service
Personnel Integration Service
Linux VM/Container(s)
Seed DB node
10.0.0.145
cassandra
Linux VM
DB node
cassandra
Linux VM
Internal VPC subnet 10.0.0.0/24
Matrix failover node (2nd availability zone)
Seed DB node
cassandra
Linux VM
DB node
cassandra
Linux VM
Okta Service
AD
AD
AD
Other data sources
Training & certification data
FAMS
Contract data
MPDS/DEERS
Personnel data