COBALT PRODUCT BRIEF

Cobalt is a full-featured identity platform designed for modern enterprise datacenter and multi-cloud deployments. It provides a seamless, DevOps-friendly architecture for directory, provisioning, authentication, and authorization services that spans your on-premises, private cloud and public cloud infrastructures. Cobalt's API-driven approach simplifies management, increases agility, and improves overall security and compliance.

Architected for Hybrid and Multi-cloud Deployment

Today's enterprise IT architecture differs substantially from that of even five years ago. Enterprise applications run both on-premises and in the public cloud, and hyperconverged appliances have completely blurred the line between "on-prem" and "in the cloud". We've architected Cobalt to work as an integrated platform across the private/public cloud boundary, allowing you to deploy and consistently manage identity services wherever your applications need them.

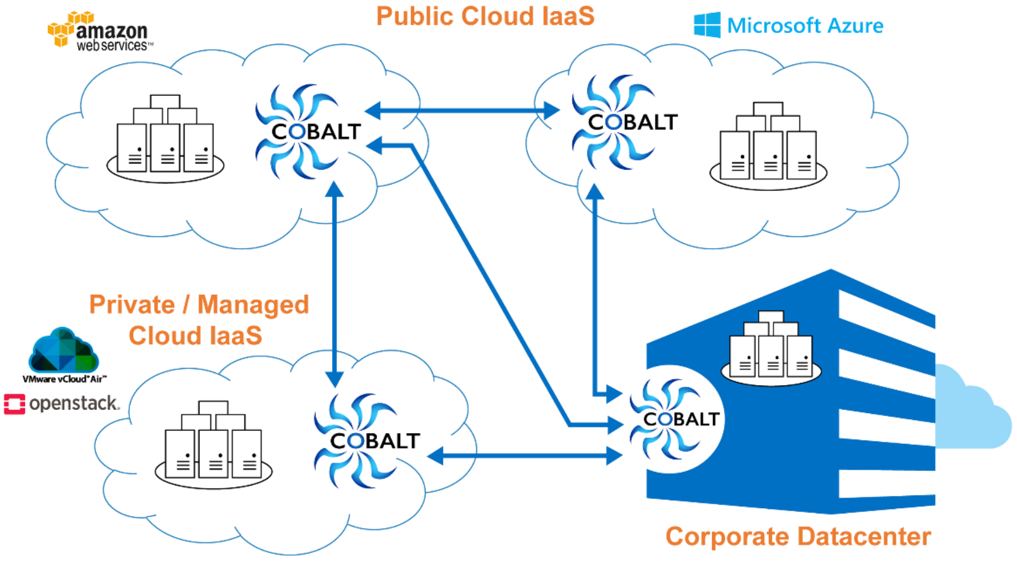

Figure 1 Cobalt multi-cloud deployment

We provide the Cobalt platform as a Docker container making its deployment fast, reliable, and automatable. By leveraging container technology, we've eliminated most of the problems associated with host system configuration and installation. One command and Cobalt is up and running!

Cobalt's microservice architecture means that you can start and configure individual identity services anywhere in the Cobalt cluster with a single API call. If you need a new directory service with authentication in Microsoft Azure, a single API call will start and configure the service within seconds. If you need a new synchronization service to provision and synchronize identity data from a Cobalt directory to a new application, a single API call will do it. You control all aspects of Cobalt service definition and configuration using simple web APIs.

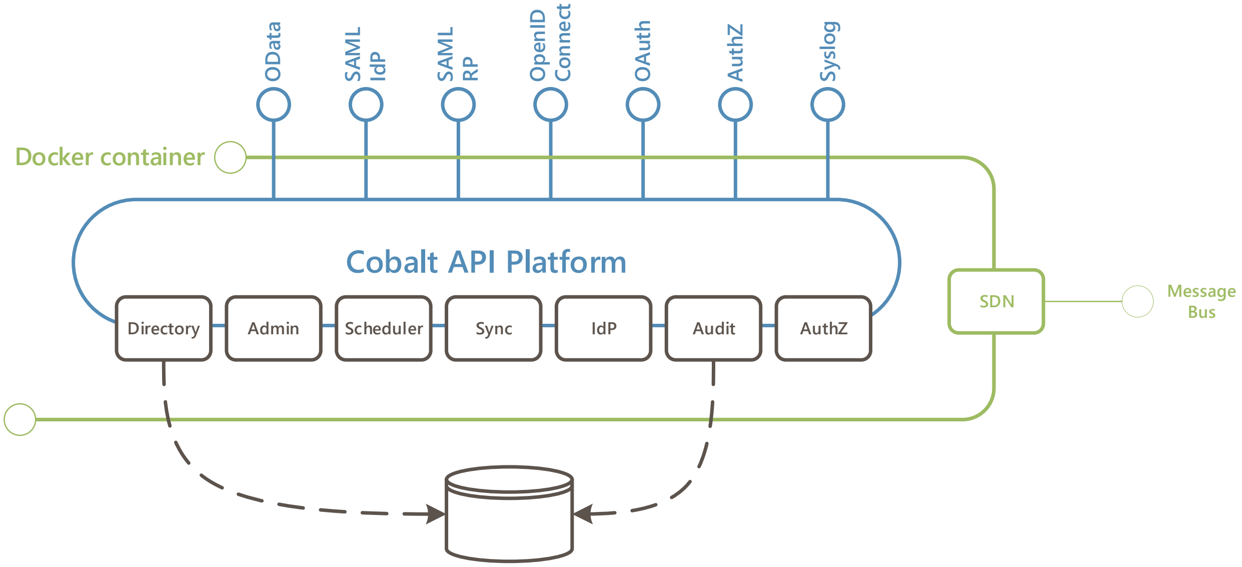

Figure 2 Cobalt microservice architecture

Cobalt microservices work together to provide internal load balancing and failover capabilities. You can easily configure an individual service such as an authentication service to run multiple instances on different hosts so that the load is shared between them. Should one host fail, service requests will be automatically routed to the remaining running services.

Cobalt runs on a software defined network (SDN) that creates a virtualized network between the containers running Cobalt. The SDN works with Docker to define its own private IP address space as well as DNS name space, so you don't have to manage the DNS names or IP addresses of the various Cobalt microservices.

Cobalt Directory Services

The directory is the core of any identity system. It provides the entities, attributes, credentials, and relationships upon which the identity system relies to make its authentication and authorization decisions. The directory also acts as a catalog of users, devices, and services and enables people and applications to search for and locate them.

Cobalt's directory service is multi-tenanted, meaning you can create multiple directory services each with its own distinct identity services. For example, you can have one directory for enterprise users, one for customers, and another one for contractors and partners. Each directory is distinct, with its own schema, authentication, and access control. Each directory requires a single API to create and start.

In keeping with modern cloud architecture, the Cobalt directory is fully replicated with built-in load balancing and failover so that your applications get the best possible performance even in the face of host or networking failures. Spinning up additional replicas of a given tenant is simply a matter of a single API call.

Cobalt exposes an OData-based API to applications to provide access to identity data. The OData standard defines a data model-driven web API that is simple for applications to use and provides fast, flexible queries to the underlying data, including the ability to navigate across relationships between entities in the directory. The OData API is a great fit for modern web and mobile applications and doesn't suffer from the same sorts of firewall and connection restrictions that earlier directory protocols (e.g. LDAP) do.

The Cobalt directory is based on the proven and secure ViewDS Directory Server, and inherits its sophisticated partial and approximate matching capabilities, including the ability to search phonetically, resolve acronyms, abbreviations, and misspellings, and even search for Mandarin text using phonetically equivalent Pinyin spellings. All of these capabilities ensure that users can rapidly find the directory data they are searching for, whether users, resources, or services.

Application requirements and design change rapidly, and one of the most common annoyances developers express about using directories is the pain of extending the schema to support those changes. Schema changes typically involve shutting down and restarting the directory service, and once committed, schema changes are often irrevocable. This leads to additional administrative processes to vet proposed schema changes, which leads to further delays. The Cobalt directory schema is fully extensible at runtime using a web API, and you can undo changes so long as they don't invalidate data that already exists in the directory. This gives you the agility necessary to keep pace with changing requirements in the identity system.

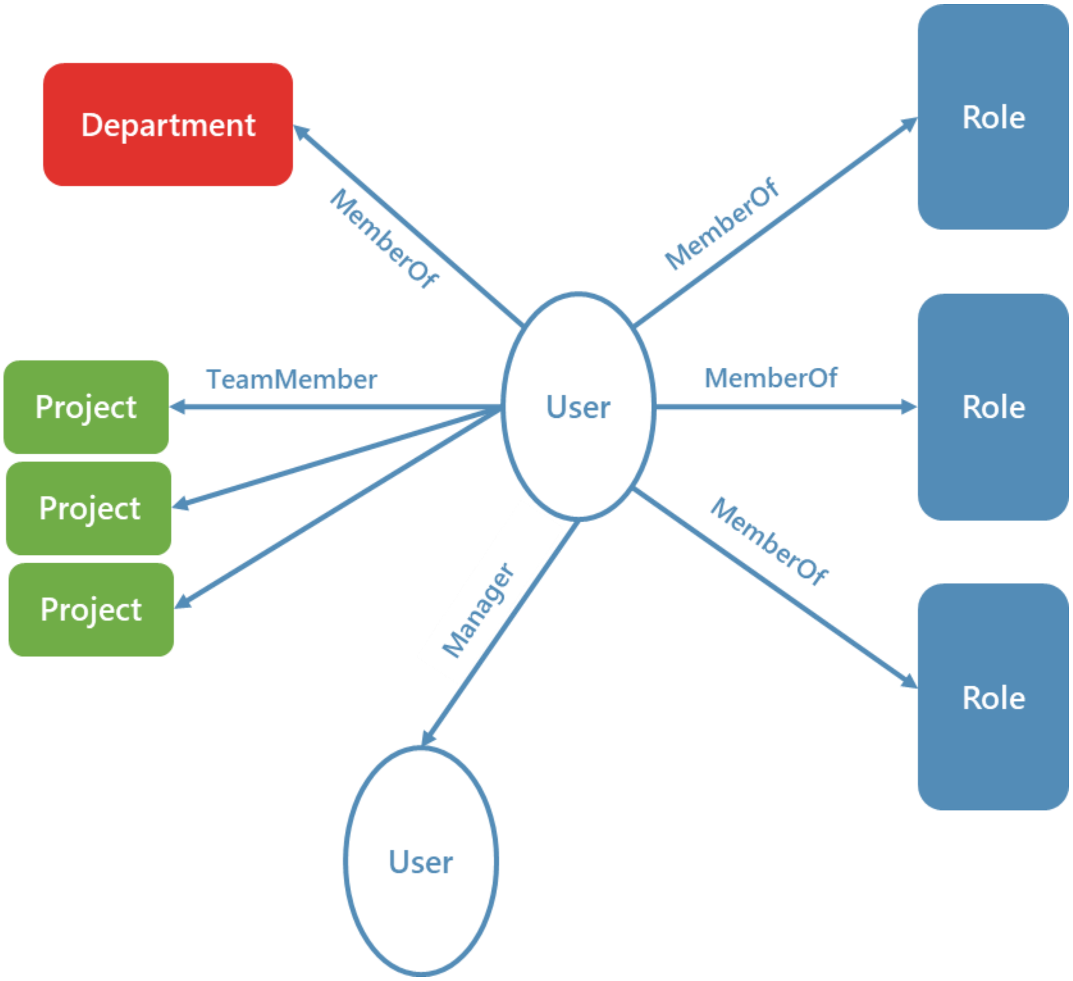

Figure 3 Cobalt directory service

The second biggest directory-related annoyance for both developers and IT staff is the difficulty of managing access rights within the directory. Conventional directories use some combination of access control lists and group memberships to define what users can do in the directory. This doesn't fit well with corporate or regulatory security requirements, and rapidly becomes difficult to manage. Cobalt uses a policy-based approach to access control, where a simple set of policy statement defines what users are allowed to do, and to what. You can, for instance, write a policy that says all users can manage their own phone number in the directory, but only HR managers can modify the users' department or manager properties. In addition to this attribute-based access control approach, Cobalt concurrently supports role-based access control policies, allowing you to assign access rights simply by assigning a role to a user. And as with everything else in Cobalt, you manage policies and roles with simple web API calls.

Provisioning and Synchronization Services

Directories almost never exist in isolation. They usually get their information from some authoritative source, like an HR system or enterprise directory, and then that directory data often ends up in application directories and databases. Ensuring the right systems get the right identity data at the right time is always problematic, and keeping it up to date is just as difficult. Most organizations rely on a jury-rigged combination of manual data entry and custom scripts to try to keep everything up to date.

Cobalt includes a provisioning and synchronization system that can automatically import and synchronize directory data from external sources such as a cloud-based or on-premises HR system, and can provision, deprovision, and synchronize user data in external applications either on-premises or in the cloud. It's sophisticated change tracking mechanism ensures that synchronization occurs quickly and efficiently, ensuring that critical identity changes end up in the right systems at the right time.

Very often you need to transform or synthesize identity attributes as you move identity data from one system to another. For instance, in some target applications, the format of a username might be restricted to a fixed number of uppercase characters, and you need to convert the Cobalt username to something the application can use. You can effect these sorts of transformations using the Javascript engine built into the Cobalt synchronization system.

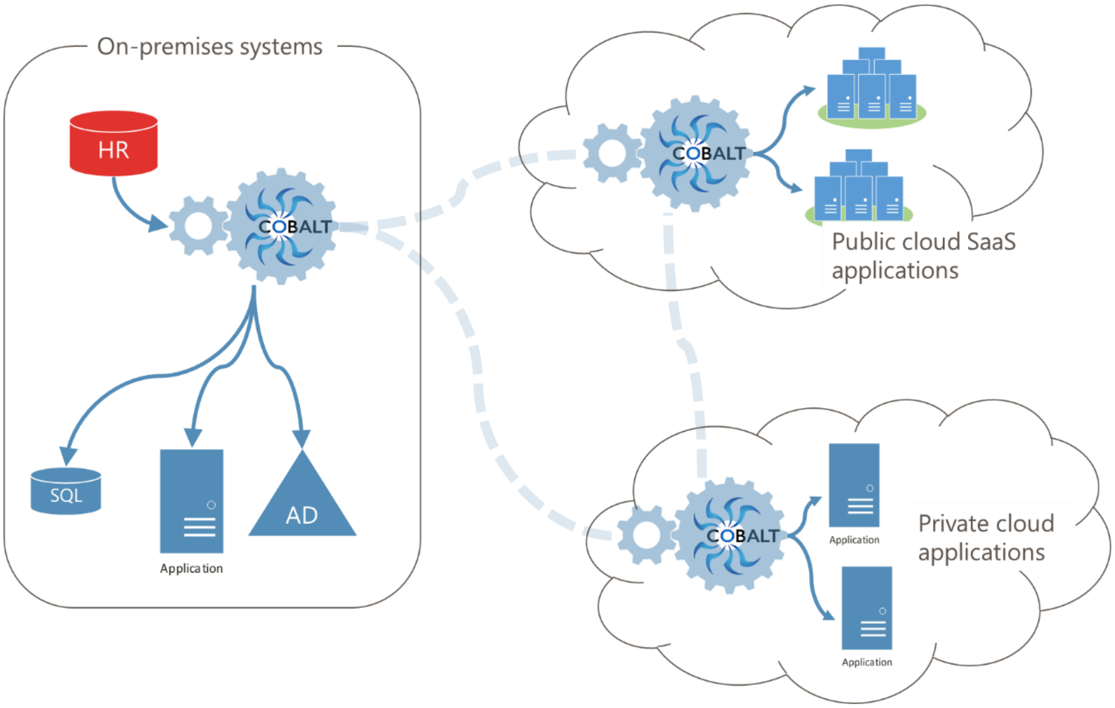

Figure 4 Cobalt provisioning system

The Cobalt synchronization system includes built-in connectors for many data storage technologies, including SQL databases, LDAP directories and structured text files such as comma-delimited CSV files. You can also easily build custom connectors using the included connector toolkit.

Authentication Services

After the directory, the ability to authenticate users and other subjects is probably the most critical function an identity platform provides. Cobalt provides several authentication mechanisms and lets you choose which approach (or approaches) are best for your particular scenario. Cobalt supports basic username and password authentication, either using web basic authentication, interactively through a customizable login form or through a web API. To control the password lifecycle, Cobalt provides a flexible password policy mechanism that lets you define different policies for different users, defining the lexical requirements of the password itself and how often it needs to be changed. This lets you easily define simple password requirements for non-critical accounts while enforcing more stringent rules for passwords belonging to sensitive or critical accounts.

Beyond usernames and passwords, Cobalt supports social login via Google, LinkedIn, and Twitter, and you can restrict which type of authentication service can be used by which applications, ensuring that users don't access sensitive applications using less trusted authentication providers. You can also configure step-up authentication so that users are required to reauthenticate using a stronger credential such as a PKI certificate or smartcard when accessing sensitive application functions.

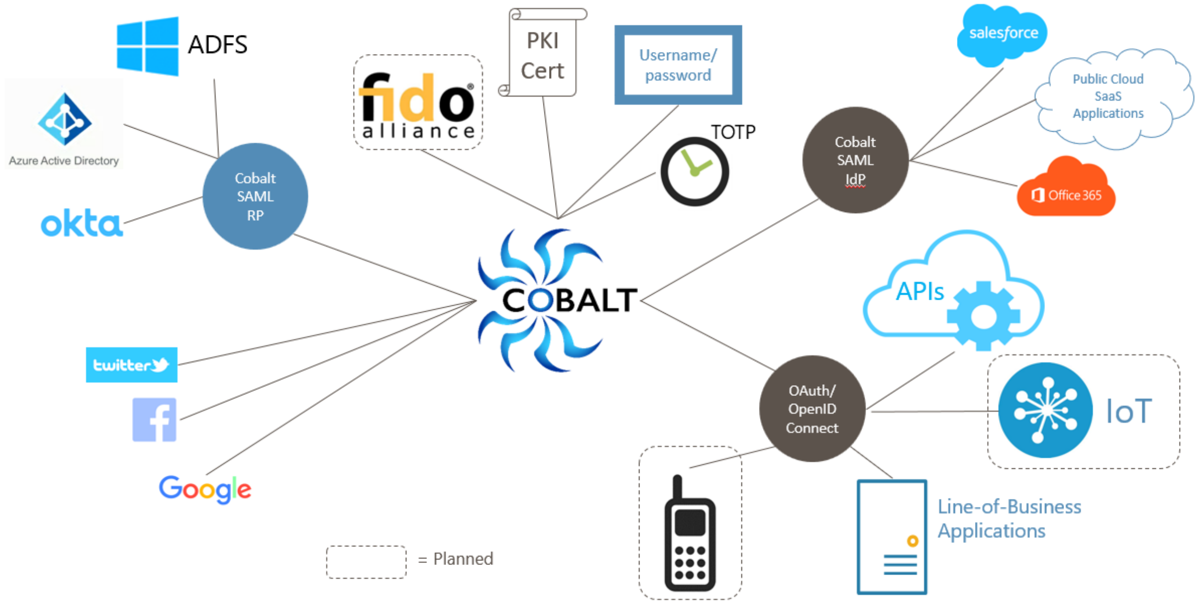

Figure 5 Cobalt authentication and SSO

Perhaps the most valuable capability an authentication system can provide is single sign-on, i.e. the ability to login once and access multiple applications without having to reauthenticate for each one. Cobalt provides authentication and single sign-on through the standard OpenID Connect protocol, and also provides single sign-on to cloud and on-premises applications by acting as a SAML 2.0 Identity Provider (IdP). Because all application registrations and trust relationships are managed using web API calls, you can instantly add new applications to Cobalt for single sign-on purposes. In addition, Cobalt provides specialized application entitlement policies that determine which users are allowed to authenticate to which applications. These policies are based on the same attribute and role information used within Cobalt to provide fine-grained authorization in the directory.

If you already have an authentication provider such as Active Directory, Azure Active Directory, or Okta that you prefer to continue to use, Cobalt can take advantage of that authentication service by acting as a SAML Relying Party and continuing to provide all the other identity services including single sign-on for OpenID Connect and SAML applications.

Authorization Services

The primary purpose of identity and access management (IAM) is not to store user information in directories, or to authenticate users and devices. The primary purpose of IAM is to control access to resources. Yet many IAM systems fall short in this area by leaving access control either entirely up to the application, or at best by providing rudimentary support for access control decisions through group memberships. Cobalt provides a policy-based approach to authorization that supports both attribute-based access control (ABAC) and role-based access control (RBAC) models in a way that is easy for administrators to manage and for developers to leverage in their applications.

The Cobalt authorization service allows you to define access control policies that both define access rights within Cobalt (for instance to establish access rights in the directory) as well as fine-grained access rights for applications. For example, an administrator can define a single role within Cobalt that grants the ability to change user names within a department and update scores in the bowling team application. Application developers can easily take advantage of the authorization service through a simple web API that allows the application to determine whether a particular operation is allowed or not.

The Cobalt authorization service is based on the XACML authorization model, but uses a simplified policy language for expressing both role definitions as well as attribute-based policies.

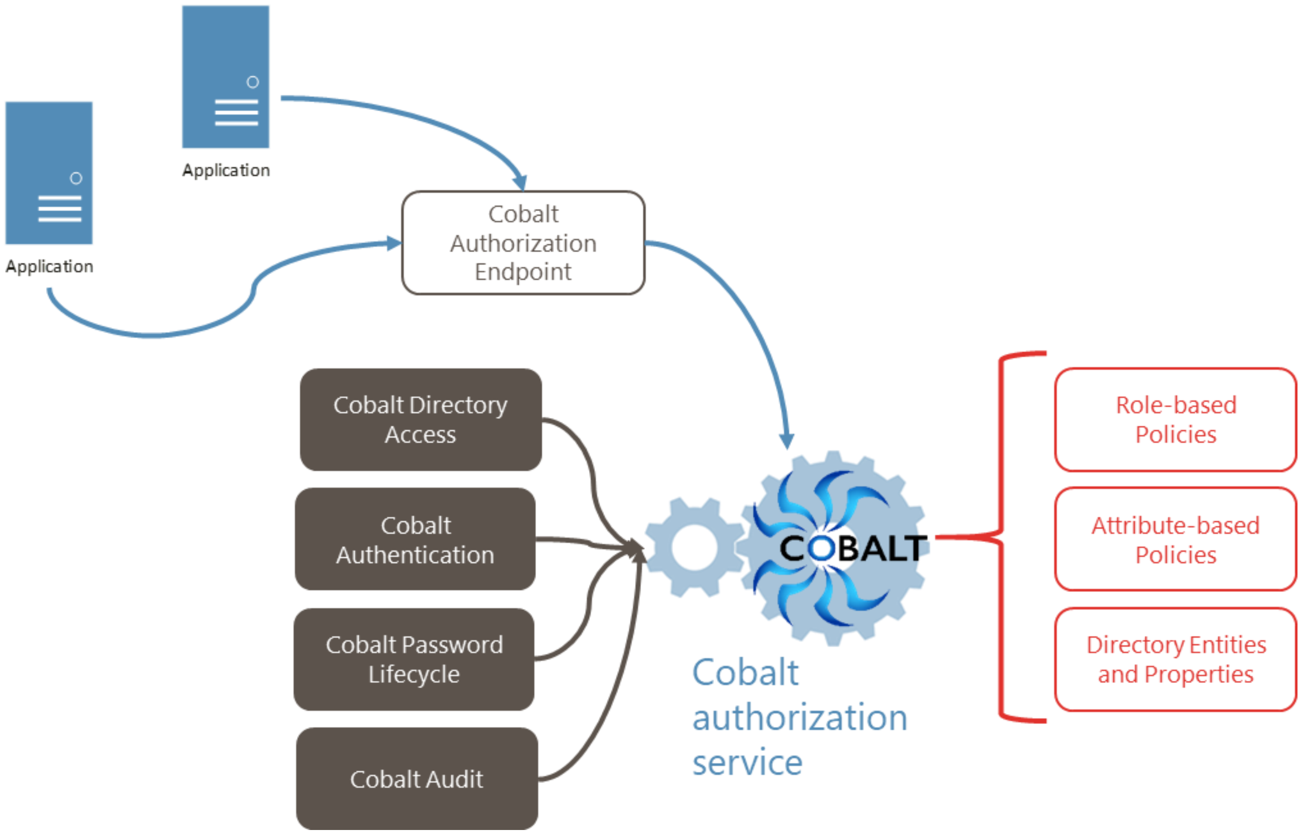

Figure 6 Cobalt authorization service

Audit Services

Because IAM systems manage access to potentially sensitive resources, it is important to not simply audit access to the application or resource, but to audit all the changes to the IAM system itself. Most IAM systems provide no or limited auditing capability when it comes to the system's internal configuration, leaving IT in the dark as to why things have suddenly stopped working the way they were supposed to.

Cobalt includes a comprehensive audit system that tracks every change to identity information, policy definitions, and service configuration, including who made the change, when and where the change was made, and what the specific change was including the old value and new value. All the information can be sent to an external log aggregator via the syslog interface or through the file system in Common Event Format (CEF). This greatly simplifies troubleshooting and makes recovery of changed system configuration settings quick and easy.

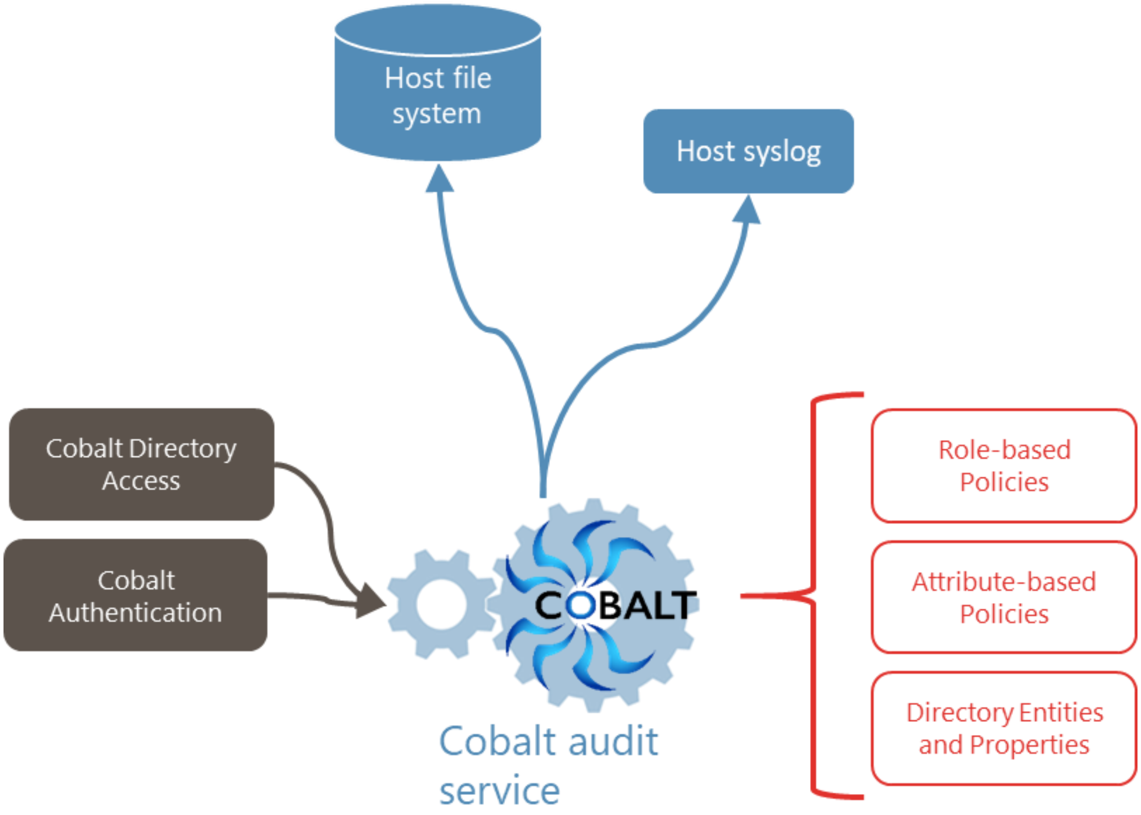

Figure 7 Cobalt audit service

Summary

Cobalt is a modern enterprise identity platform that provides all core identity services, including

- Directory services
- Authentication
- Single sign-on
- Fine-grained authorization
- Role-based access control
- Attribute-based access control
- Identity provisioning and synchronization
- Fine-grained audit

Cobalt uses a microservice architecture that can span your computing infrastructure from on-premises data center, to private cloud and multiple public cloud environments. It's containerized, API-driven approach provides your identity infrastructure superior agility and manageability.