Hexa + IDQL

Hexa Policy Orchestration and the Identity Query Language (IDQL)

July 2022

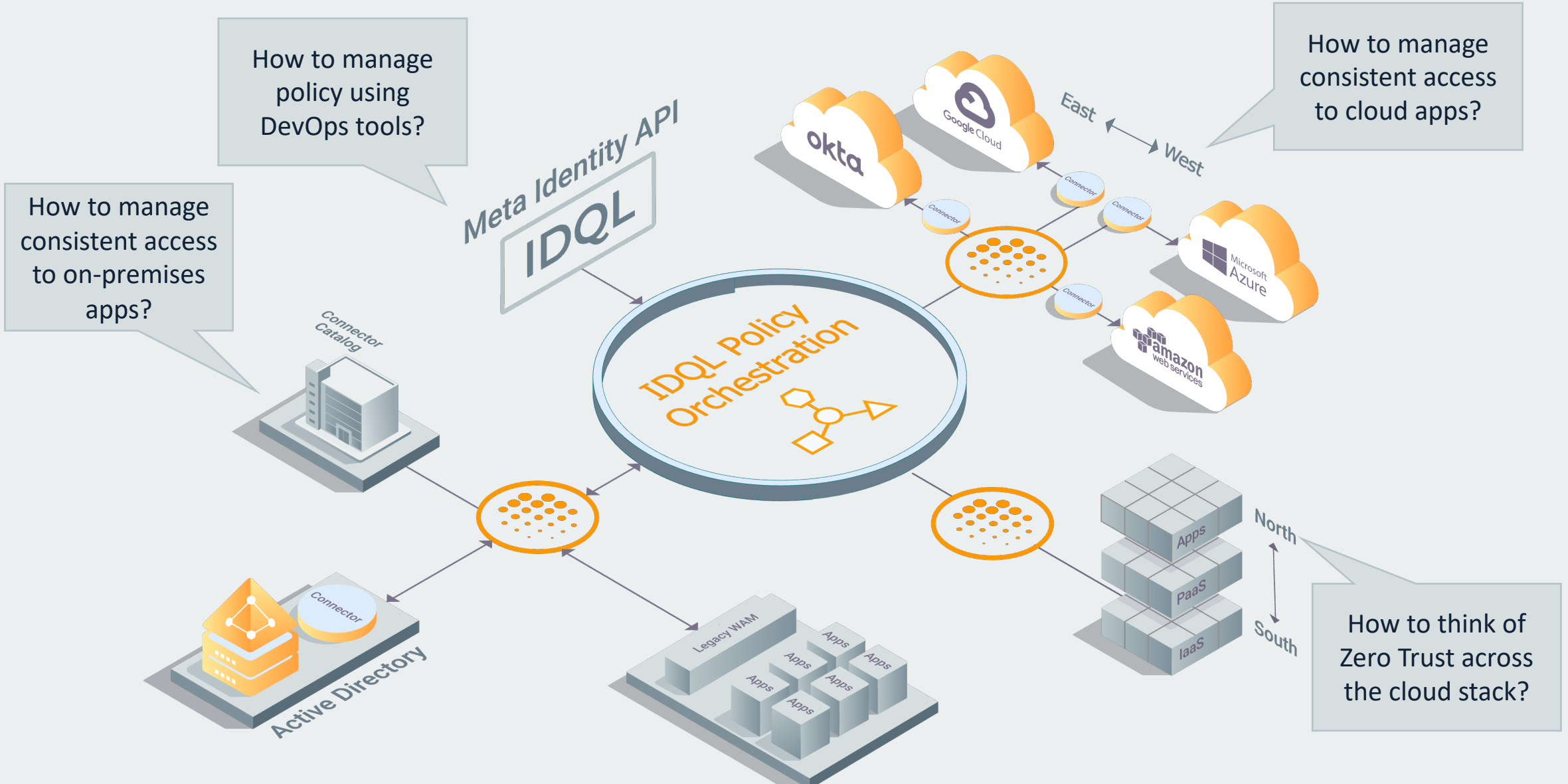
Agenda

- Introductions
- IDQL/Hexa overview
- Demonstration
- TOC questions/issues
- Open Q&A

Hexa and IdentityQL (IDQL)

Standardizing Access Policy Across The Cloud and Across The Stack

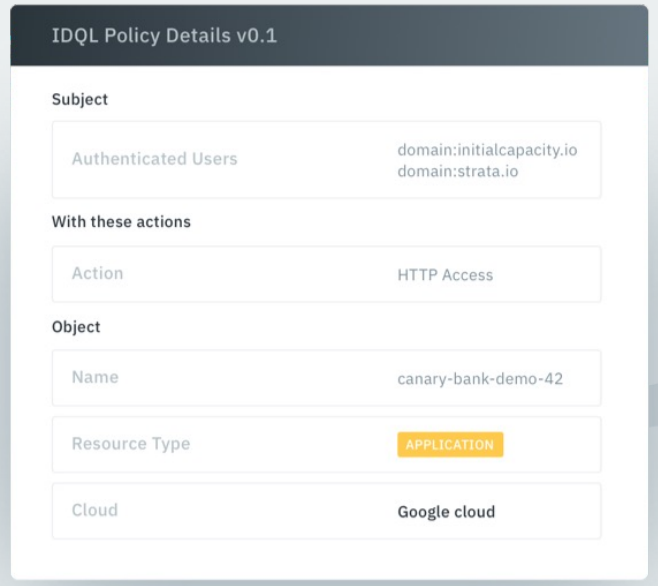
Identity and Access Policy is Fragmented



IDQL and Hexa – Open Source Policy Orchestration

Hexa

IDQL
Universal Declarative Policy



IDQL Policy

Hexa
Policy Orchestrator

Policy Discovery

Policy Translation

Policy Orchestration

Native policy

Native policy

Across Clouds

Across Stack

Custom

Hexa Provider Interface Architecture

Hexa

IDQL JSON

```
{
  "policies": [
    {
      "meta": {
        "version": "0.1",
        "date": ""
      },
      "subject": {
        "type": "",
        "provider": "",
        "role": ""
      },
      "actions": [
        {
          "name": "",
          "actionUri": ""
        }
      ],
      "object": {
        "assetId": "",
        "pathSpec": ""
      }
      "condition": {
        "rule": "",
        "action": ""
      },
    }
  ]
}
```

REST API

Hexa Policy Orchestrator

Hexa REST API

Manage Provider Integrations
Manage Resources (app, etc)

Hexa Provider Interface

for each provider {

Discover(ProviderInfo)
GetPolicyInfo(ProviderInfo)
SetPolicyInfo(ProviderInfo)

}

Providers

Azure provider pkg

Provider implementation
Translates Native ↔ IDQL

REST API

Azure

Azure REST API
Applications, platform, data

AWS provider pkg

Provider implementation
Translates Native ↔ IDQL

REST API

AWS

AWS REST API
Applications, platform, data

GCP provider pkg

Provider implementation
Translate Native ↔ IDQL

REST API

GCP

GCP REST API
Applications, platform, data

Any provider pkg

Provider implementation
Translate Native ↔ IDQL

REST API

Any

Any REST API
Applications, platform, data

Hexa Policy Orchestration

- **Open Source**

  The Hexa Policy Orchestration Gateway will be freely available open source – Apache licensed.

- **Zero Overhead**

  Policy orchestration is done through admin processes, not runtime.

- **Simplified compliance**

  Easy to read access and user reports generated from across clouds.

- **East <-> West policy**

  Policy is unified across AWS, Azure and GCP (and any other cloud platform).

- **Enterprise-ready**

  Hexa works with your containers, Kubernetes, OPA, and CI/CD tools and processes.

- **Seamless API integration**

  Native integration with cloud and identity APIs insulates you from changing cloud APIs.

- **North <-> South policy**

  Policy is also managed across the stack, across apps, data, platform and networking.

- **Simplified security**

  IDQL is a clear, declarative policy that is simple to understand and enforce.

- **Natively integrated with other CNCF tools**

  Orchestrate OPA policies with your other access management systems, orchestrate K8s policies and run Hexa on Kubernetes.

Benefits of Policy Orchestration with Hexa and IDQL

- **Agentless and proxyless Policy Orchestration**

  Implement in minutes without changes to your infrastructure.

- **Distributed policy management**

  Orchestrate access policy securely through APIs with no change required to target systems.

- **Universal Access Policy**

  Manage access policy that works across disparate systems to help enable a Zero Trust Architecture.

- **Policy-as-Code**

  Bring identity and access policy into code for large-scale automation.

- **Declarative Policy**

  Understand who has access to which apps and data at a glance.

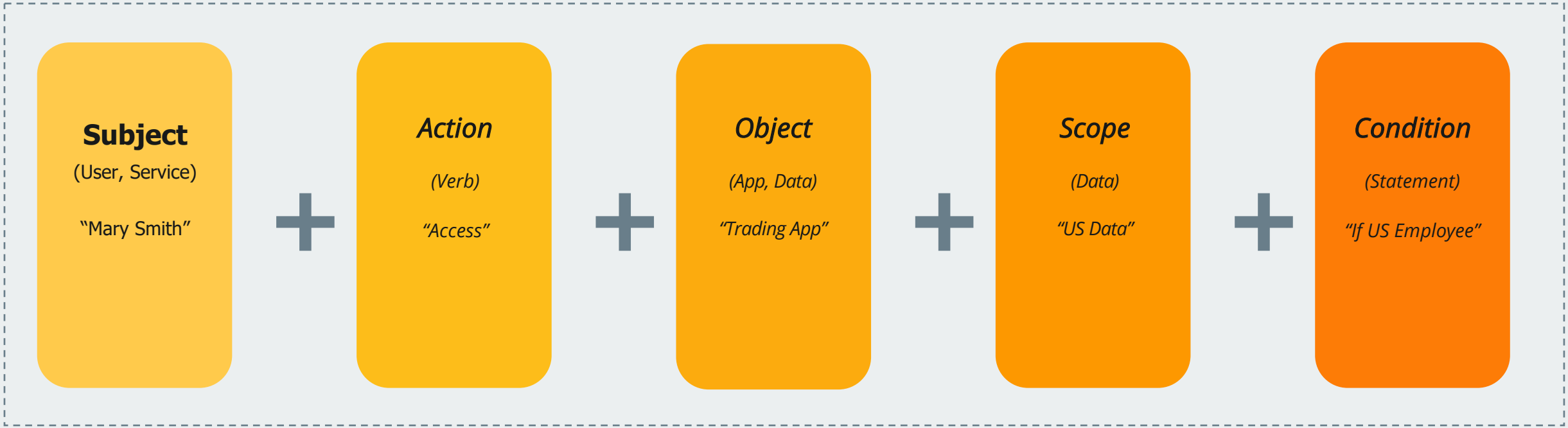
- **Break lock-in**

  Break lock-in and enjoy portability and vendor choice

IDQL – Simple, Declarative Access Policies

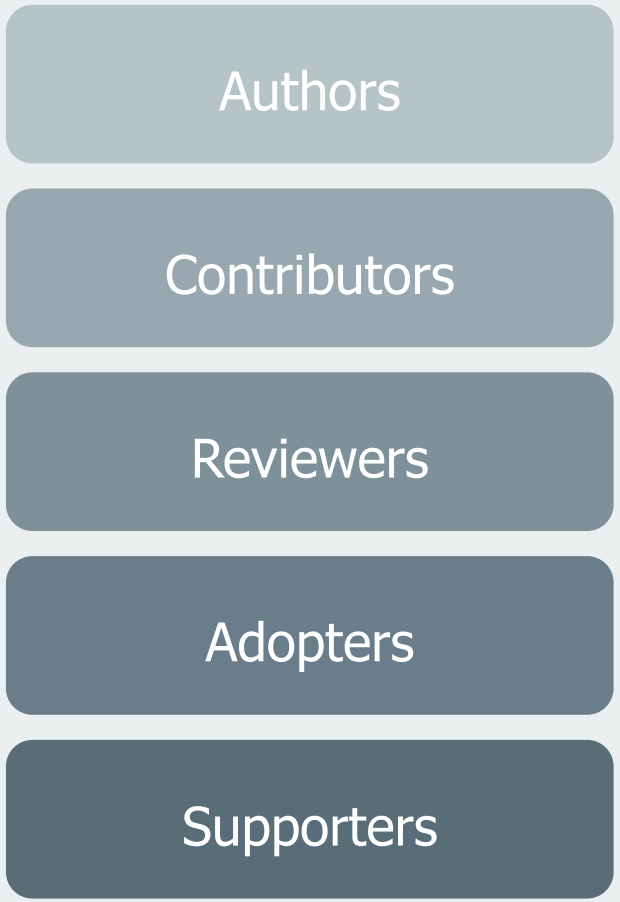
Hexa

Context



Simple, **declarative** syntax is easy to read and understand by people.
Clearer understanding of policies reduces risk

Building IDQL With The Open Community



- **Authors**: Edit the policy, API specifications, and Hexa source code
- **Contributors**: Provide thought leadership and design, use cases, design and environmental requirements
- **Reviewers**: Design reviews, code reviews
- **Adopters**: Use Hexa and IDQL in Pilot and Production
- **Supporters**: Members of the Hexa and IDQL Community

Demonstration

TOC questions / issues

“What is the target audience for IDQL/Hexa”

Hexa

- The challenge of consistently and uniformly managing access policy is one that impacts developers, administrators, DevSecOps, information security and so on
- IDQL is designed to be translated/mapped to the bespoke access policy format of target platforms/systems
- Hexa is the open source software that connects to target systems and translates to/from IDQL
- The combination of IDQL and Hexa simplifies managing disparate systems and platforms – instead of having to manage each of them with proprietary tools and specialized skills

“Integration with other projects is curious 1/2”

Hexa

- We have spent a lot of time/effort to build and include a working demonstration environment that new users/contributors can access to get started quickly with IDQL/Hexa
- The project includes
  - Hexa orchestrator: **the main component**
- The following are OPTIONAL and may eventually be moved to another repo
  - Hexa admin: a sample administrative interface for viewing/changing IDQL policies
  - Hexa demo: a simple demonstration application that uses OPA for its internal access control
  - OPA server: an instance of OPA is included for the demonstration environment

“Integration with other projects is curious 2/2”

Hexa

- IDQL/Hexa utilizes a number of CNCF projects and other open source components to automate our CI/CD pipeline, which automates our testing scenarios and supports the demonstration components.
- The following components that are mentioned in the repo are not deliverables or dependencies – we have also removed them from the main README to help clarify intentions
  - Kubernetes: K8s is used to run the orchestrator in a demo environment
  - Contour: is used as the ingress service on k8s
  - Cert Manager: used to set up TLS certs for communication between Hexa components on k8s
  - Harbor: this is the container registry used for CI/CD
  - Concourse: is used for the Hexa CI/CD pipeline
  - Open Policy Agent: the demo application uses OPA for access decisions. OPA is also one of the target environments that integrates with IDQL/Hexa

“Use of HAWK”

- HAWK is used for securing API requests to the orchestrator. The main driver for using HAWK early is to avoid needing a full OAuth server for those just getting started
  - This is a stepping stone on the path to full OAuth implementation

- In the current implementation HAWK verifies that the request URI used by the client matches the configured request URI.

- Eventually the orchestrator shall be expanded to support common API authentication methods such as JWT tokens, PATs, and sender constrained tokens such as OAuth DPOP

“What is the relationship between IDQL and Hexa?”

Hexa

“Does the project include both or one of IDQL and Hexa?”

- IDQL is the policy format and Hexa is the software that translates target policies to/from IDQL, and uses orchestration to publish changed policies to the target systems
- We separated IDQL and Hexa into separate repos so that they could be managed/maintained independently – similar to SPIFFE and SPIRE

- The combination of IDQL and Hexa represent the sandbox submission

“Issues don’t have descriptions, only titles”

Hexa

- Going forward, we will add more information in issue descriptions to inform potential future contributors and also to improve project documentation

There was a reference to “email received about Contour”

- Not sure what this means

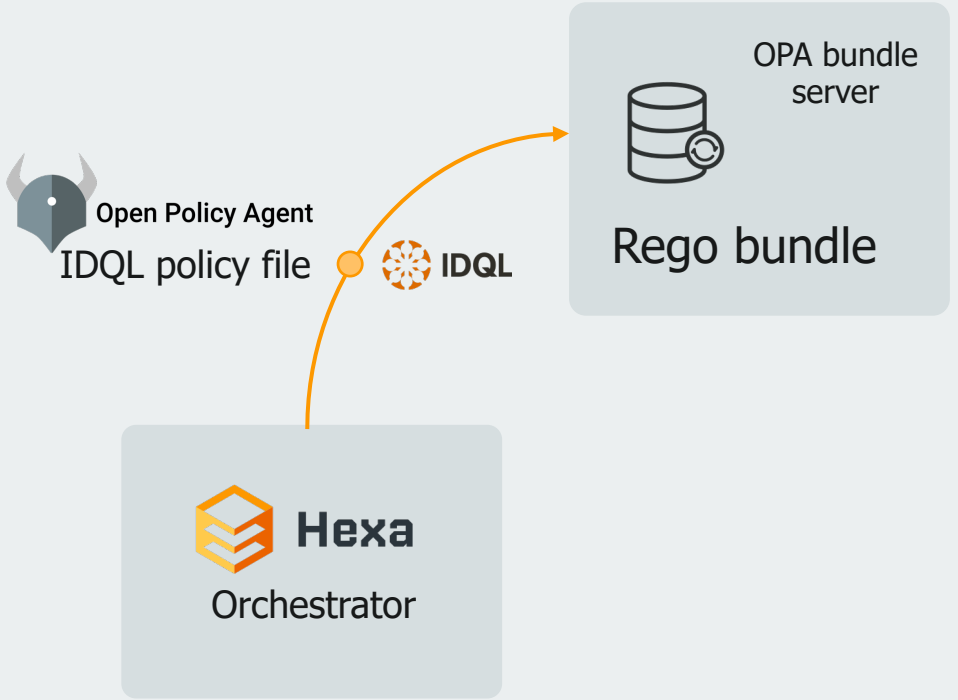
Reference information

- Hexaorchestration.org
- Github.com/hexa-org
- Webinar recording: https://www.brighttalk.com/webcast/19147/545972

Backup

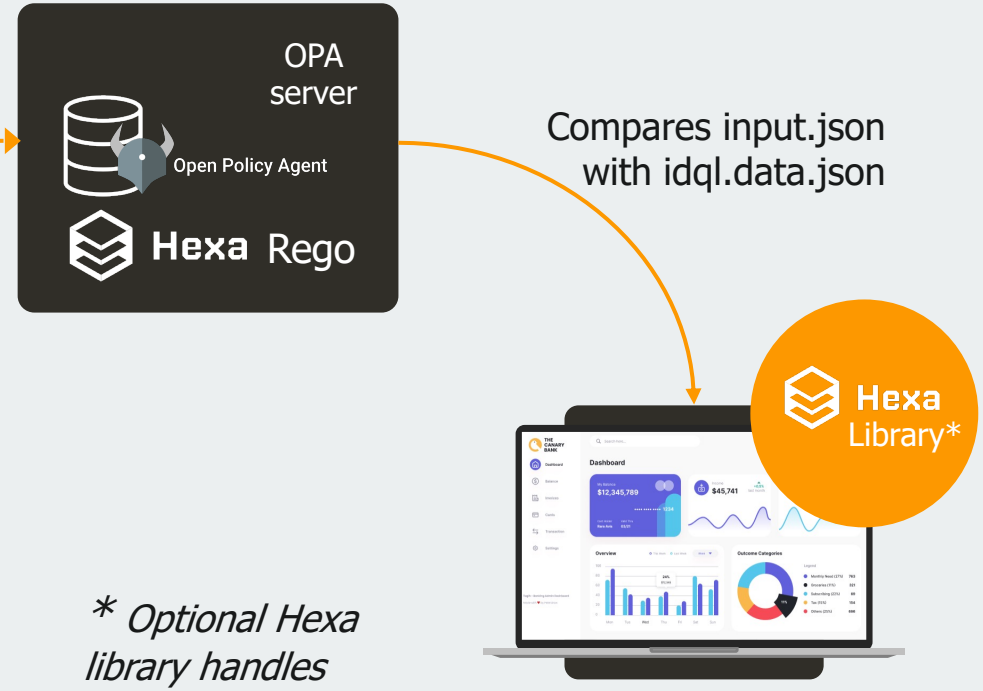
Admin Experience

OPA Bundle Server creates bundle with IDQL as the data element



Runtime experience

OPA server with Hexa Rego module evaluates incoming access request



** Optional Hexa library handles structuring inputs for the developer and sends request to OPA*