EDC Fleet Coordination

Overview and Architecture

Jim Marino
jmarino@metaformsystems.com

Fleet Coordination Requirement

As an organization, I want to ensure consistency and transparency for all data-sharing operations across multiple EDC deployments.

- **Consistency:** Enforce common policies, access control, and usage control across applications and partner integrations.

- **Transparency:** Observe, track, and administer data-sharing contracts and activities across applications and partner integrations.

Managing Multiple EDC Deployments

EDC as a Service

Enables a hosting company to operate EDCs for multiple organizations

Involves creating provisioning and deployment control infrastructure.

Management Domains

Enables a single organization to deploy multiple EDCs that are independently operated and managed

Creates a unified catalog from multiple EDC deployments in the same organization

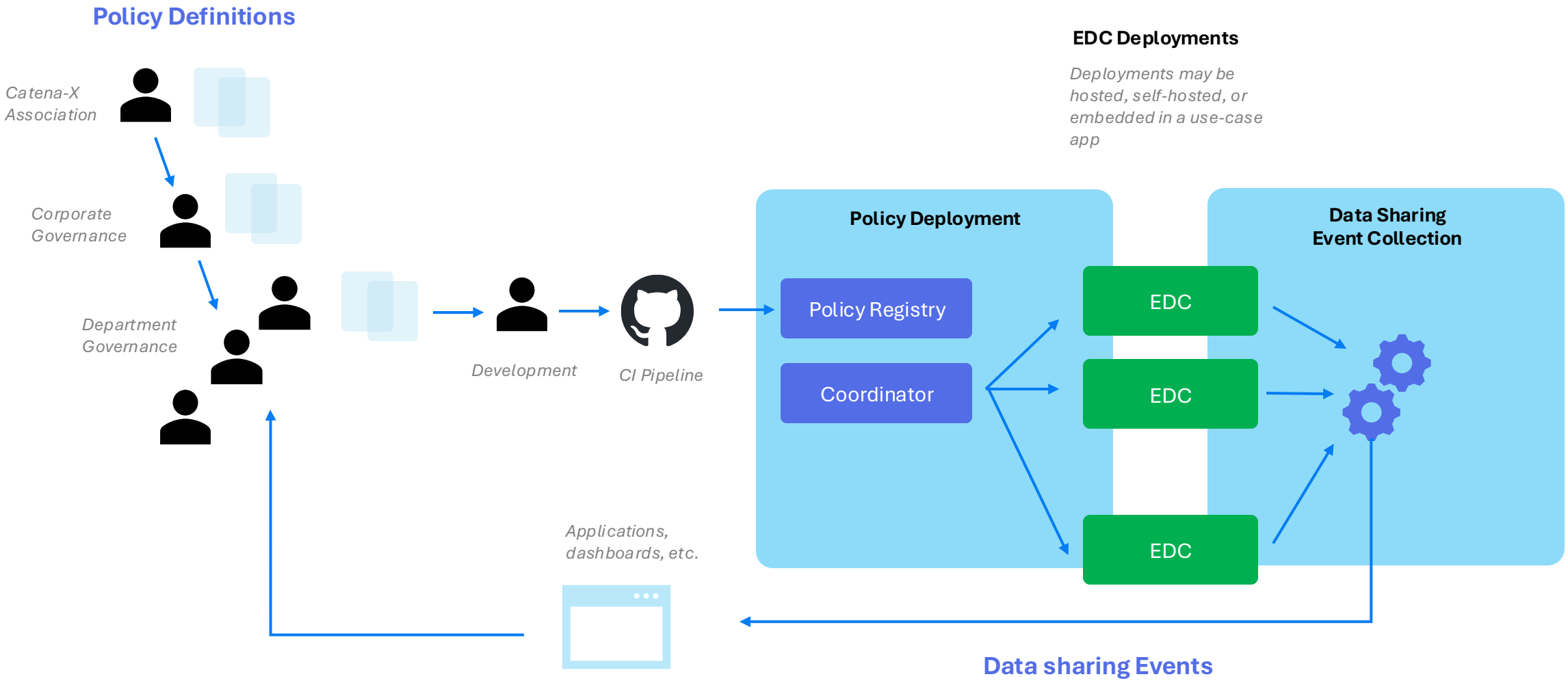
Fleet Coordination

Enables a single organization to ensure consistency across multiple EDC deployments

Creates a unified way to deploy policies and other artifacts across management domains

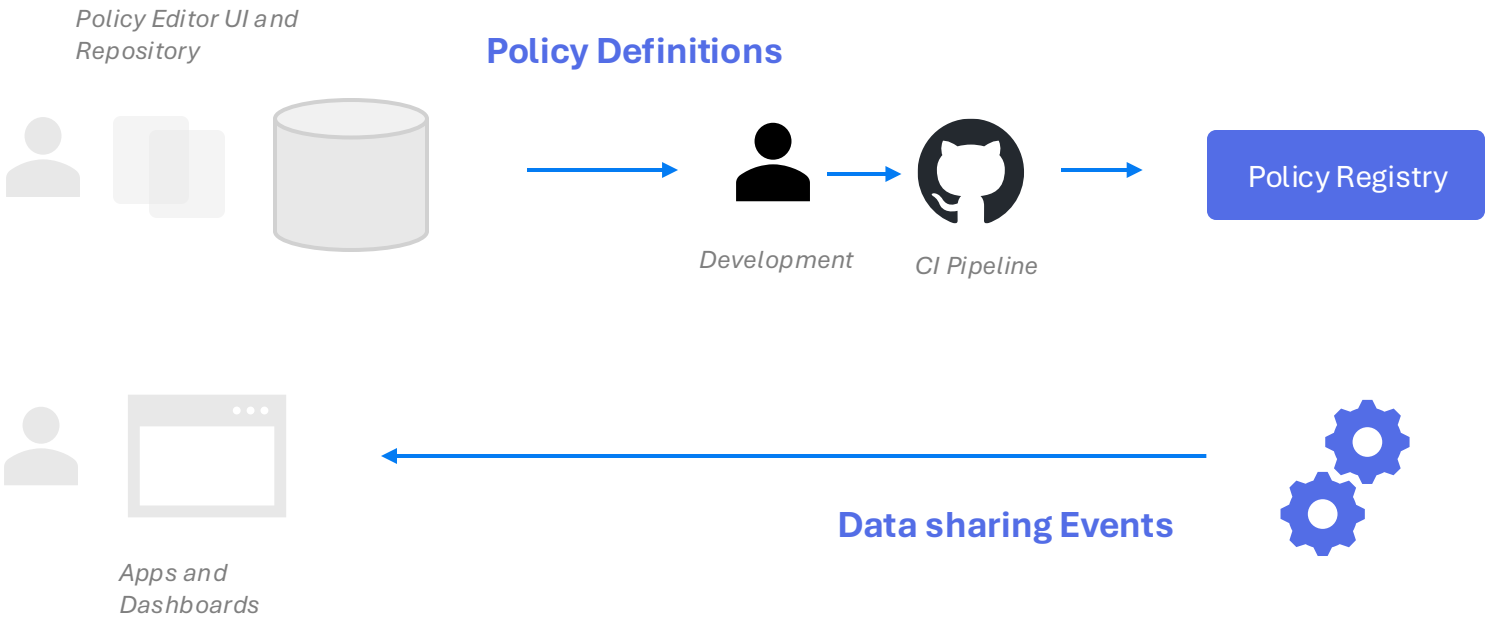
Creates a unified way to receive and process events related to data sharing activities

Fleet Coordination Use Case



Fleet Coordination Deployment Focus

- The Policy Registry is not an “editor”
- Raw event feeds, no dashboard



Consistency

Enforce common policies, access control, and usage control across applications and partner integrations

Consistency Technical Problem Statement

How do we coordinate data-sharing resources across disparate applications and EDC runtimes in an organization?

- Resources include
  - Access control policies
  - Usage control policies
  - Contract definitions
  - Verifiable credentials

- EDC runtimes may be
  - Directly deployed by the organization
  - Deployed as part of a proprietary application
  - Hosted by a service provider

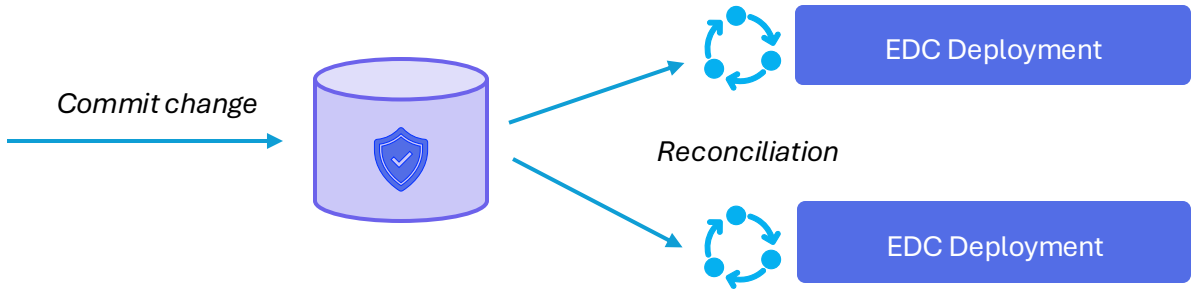
Fleet Coordination

- Disparate EDC runtimes form a *fleet*
  - Individually operated deployments
  - Spans multiple *management domains*
- Based on the CNCF xRegistry Specification
  - https://github.com/xregistry/spec
  - For an overview see the presentation by SAP [1]
- Useful for single EDC deployments where top-down organizational control is not required
  - Allows artifacts such as policies to be placed under source control and deployed as part of a CI pipeline

[1] https://www.youtube.com/watch?v=msqjUa11t50&ab_channel=CNCF%5BCloudNativeComputingFoundation%5D

Reconciliation

- Fleet coordination is based on ***reconciliation***
  - Resources are declared in a registry
  - It's the job of individual deployments to update based on registry changes periodically
- Reconciliation is a pull model
  - Traditional approaches "push" changes to a target through a management API.
  - Changes are made to a registry without the need to target specific deployments



Fleet Coordinator

- EDC extensions that can be deployed along with the EDC control plane
  - Deployed per management domain
  - Periodically pulls from configured registries via HTTP(S)
- Proprietary distributions can use the EDC extensions or support the registry interface directly
  - Applications
  - Hosted EDC solutions

Transparency

Observe, track, and administer data-sharing contracts and activities across applications and partner integrations

Transparency Technical Problem Statement

How do we observe data-sharing activity events across a fleet of EDCs?

- Events include
  - Contract negotiations and agreements
  - Transfer processes
  - Policy violations

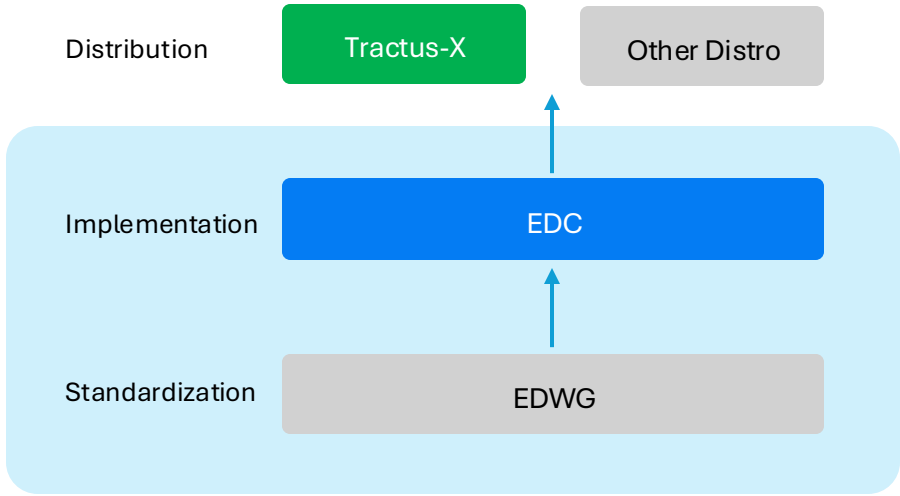
- EDC runtimes may be
  - Directly deployed by the organization
  - Deployed as part of a proprietary application
  - Hosted by a service provider

Data-Sharing Activity Events

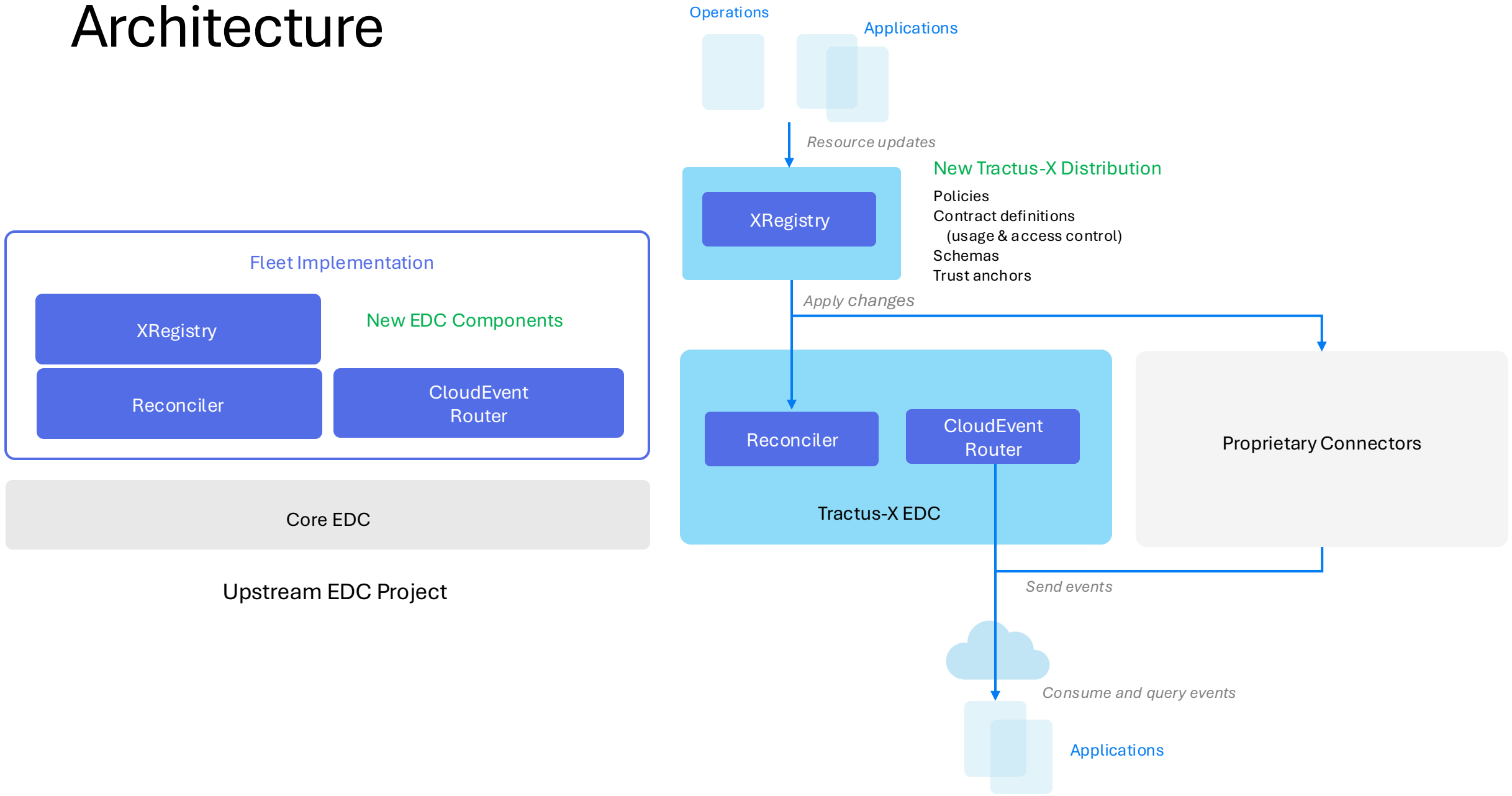
- Create EDC extensions to emit events in a standardized format
  - Extends the existing low-level EDC eventing system
- Activity Events
  - Have business meaning
  - Are wire protocol agnostic
    - Work with Kafka, MQTT, RabbitMQ, AMQP, cloud event systems, etc.
  - Can be routed/relayed across diverse infrastructures (cloud, on-premise, hybrid)
  - Can be queried and stored for future use
- Different than telemetry
  - Telemetry records system events
  - Errors, latency, throughput, etc.
- Based on CNCF Cloud Events specification
  - https://cloudevents.io/
  - Companion specification to xRegistry

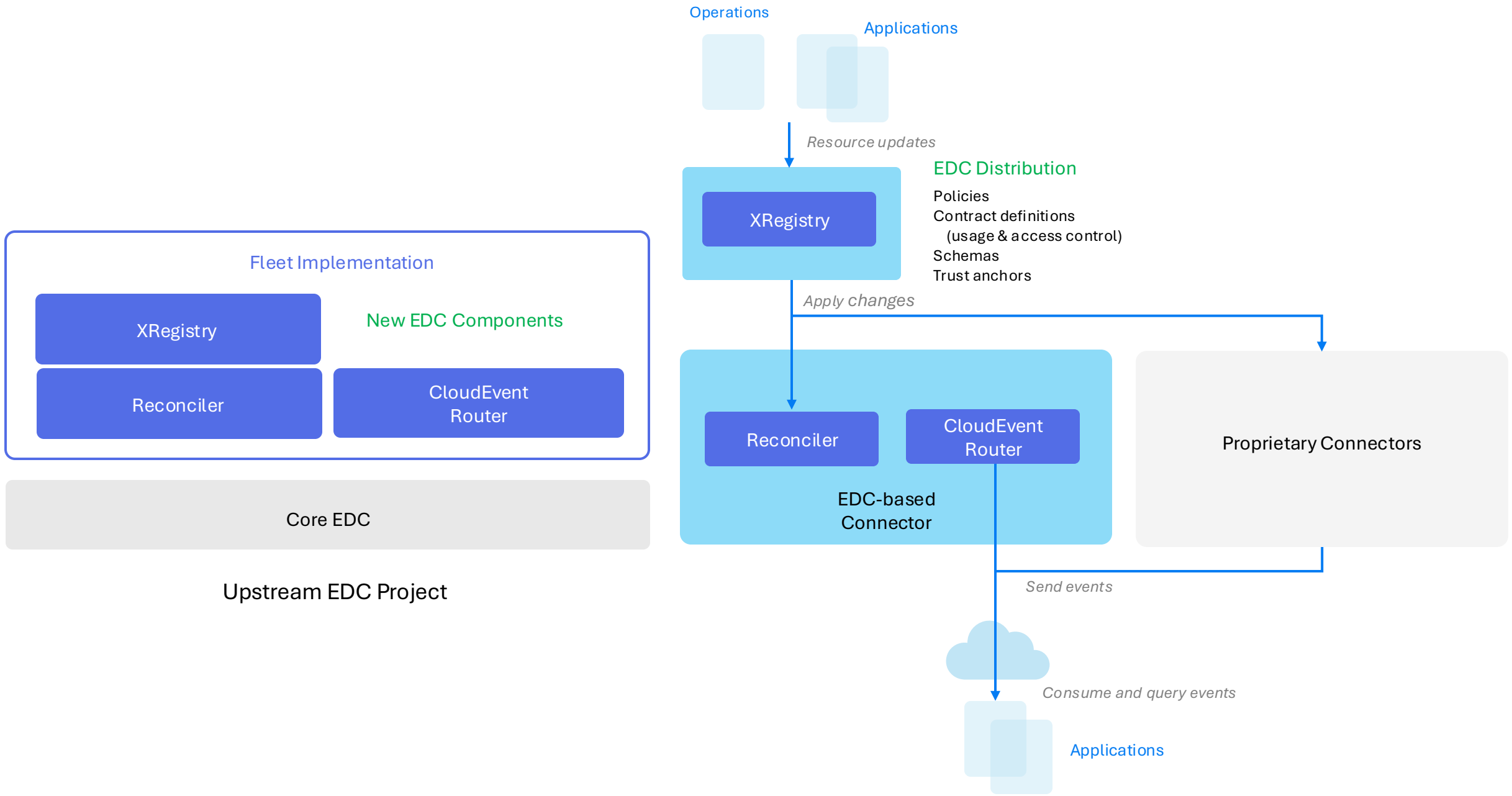
Work Streams

- EDWG
  - Standardization of registry artifacts as extensions to the xRegistry specification
  - TCK to verify compliance
  - Optional standard – not required for baseline interoperability (DSP and DCP)
- EDC project
  - Define and implement Fleet coordination as extensions
  - Maintained in an EDC technology repository
- Tractus-X, proprietary EDC distributions
  - Optionally incorporate upstream EDC modules



Architecture



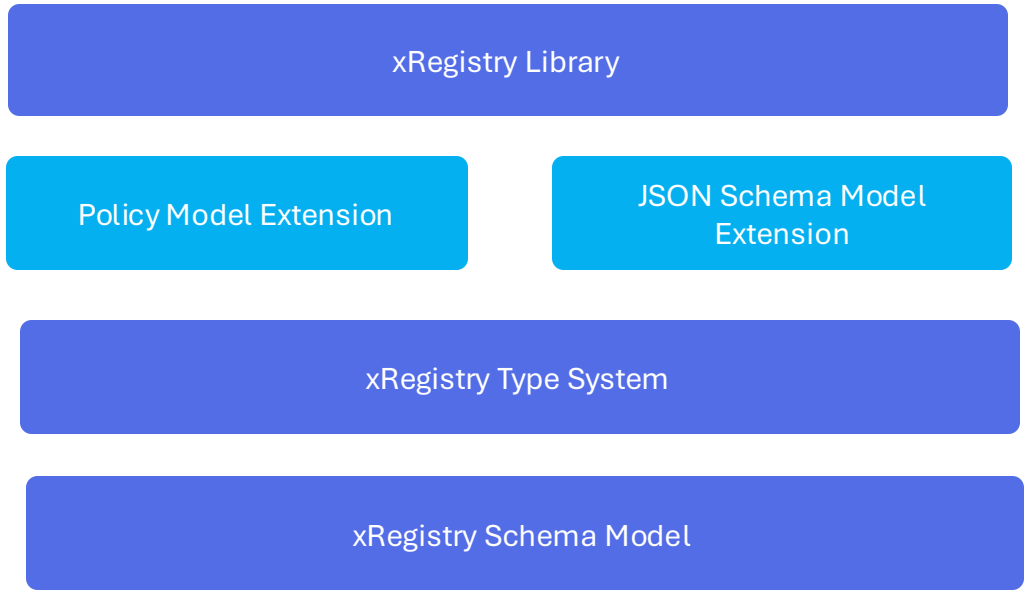

Operations
Applications
Resource updates
EDC Distribution
XRegistry
Policies
Contract definitions
(usage & access control)
Schemas
Trust anchors
Apply changes
Fleet Implementation
XRegistry
New EDC Components
Reconciler
CloudEvent Router
Core EDC
Reconciler
CloudEvent Router
EDC-based Connector
Proprietary Connectors
Send events
Consume and query events
Applications
Upstream EDC Project

Proof-of-Concept

- Produce working code that verifies use of xRegistry and reconciliation
  - Not throw-away - Code will be reused in the production system
- xRegistry library
  - Parsing, manipulating, and writing registry types
- xRegistry Server
  - Initial server built using the EDC module system
- EDC Reconciliation extension
  - Initial system
- CloudEvents not addressed
  - Relatively simple
  - Focus on more difficult parts

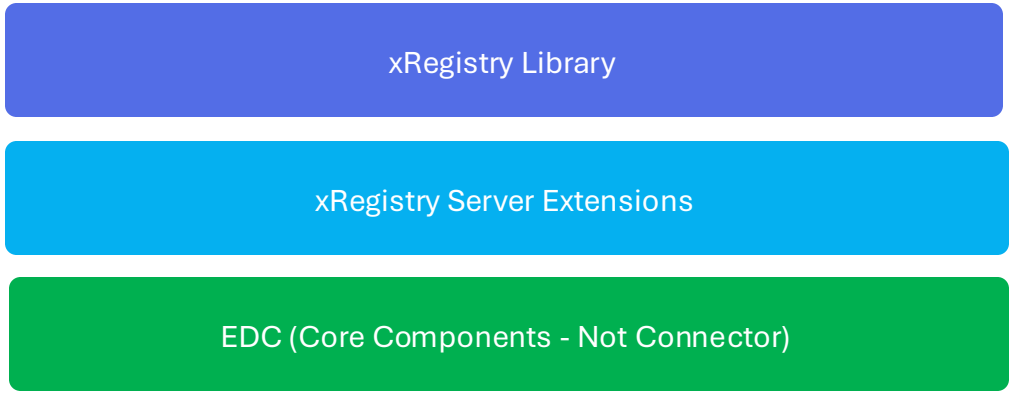
xRegistry Library and associated modules

- Schema model
  - Defines xRegistry resources
- xRegistry type system
  - Provides a strongly typed interface over the xRegistry model
- Policy and Schema extensions
  - Defines EDC policies and policy schemas
- xRegistry Library
  - Utilities
  - Model validators and resolvers
- Used in multiple contexts
  - xRegistry Server
  - Validation tooling
  - EDC Reconciliation



xRegistry Service

- Built on EDC core components – not connector
- Uses the xRegistry Library
- Fully extensible
  - Add custom resource types (e.g. schema)



EDC Reconciliation

- Reconciliation Manager
  - Periodically runs *Resource Reconcilers* responsible for updating EDC artifacts
- PolicyResourceReconciler
  - Reconciles EDC policy definitions
  - Implementation not complete, just demonstrate how it works