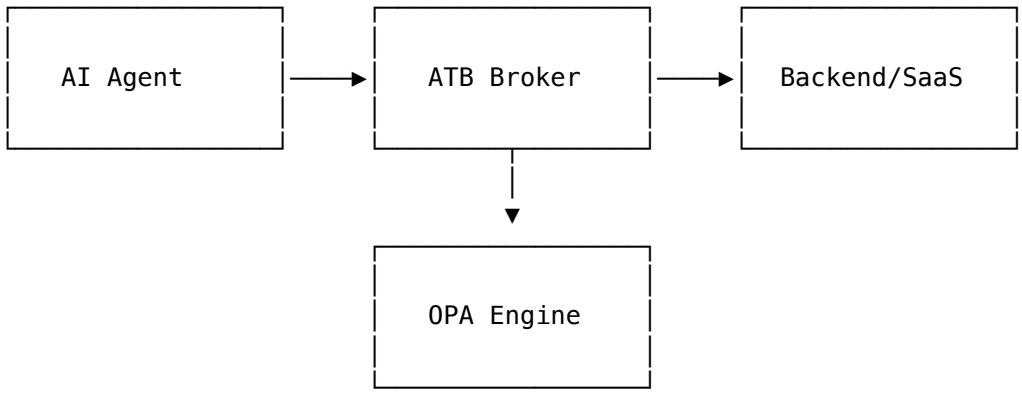
ATB Architecture Guide

This document describes the architecture of the Agent Trust Broker (ATB) system.

Overview

ATB is an enterprise security enforcement layer that validates AI agent actions before they execute on backend systems. It implements a **Proof-of-Authorization (PoA)** framework with risk-tiered governance.



Components

1. ATB Broker (`atb-gateway-go`)

The broker is the core gateway that:

- **Terminates mTLS** connections from AI agents
- **Extracts SPIFFE IDs** from client certificates
- **Validates PoA tokens** (RS256 JWT mandates)
- **Queries OPA** for policy decisions
- **Proxies authorized requests** to upstream backends
- **Emits audit events** for compliance

Key Files:

- cmd/broker/main.go - Broker entry point
- broker/ - Broker package

2. AgentAuth Service (`atb-gateway-go`)

Issues PoA tokens to authorized agents:

- Validates agent identity via mTLS/SPIFFE
- Mints short-lived PoA JWTs with action scope
- Enforces platform-specific constraints
- Supports risk-tier approval requirements

Key Files:

- cmd/agentauth/main.go - AgentAuth entry point

3. OPA Policy Engine (`opa/policy/`)

Enforces risk-tiered authorization rules:

- **LOW risk**: Auto-approved (health checks, read-only)
- **MEDIUM risk**: Requires single human approval
- **HIGH risk**: Requires dual control (2 approvers)

Key Files:

- poa.rego - Main policy
- poa_risk_tier_test.rego - Risk tier tests
- poa_leg_test.rego - Legal basis tests

4. SPIRE Integration (`spire/`)

Provides workload identity:

- Issues SPIFFE IDs to workloads
- Supports JWT-SVIDs and X.509-SVIDs
- Enables zero-trust mTLS between services

Key Files:

- spire/server/server.conf - SPIRE server config
- spire/agent/agent.conf - SPIRE agent config

Request Flow

```
1. Agent → Broker: mTLS with SPIFFE cert + PoA token
   │
   ├─ Extract SPIFFE ID from client cert
   ├─ Validate PoA JWT signature and claims
   │
2. Broker → OPA: Policy decision request
   │
   ├─ Input: { claim, method, path, spiffe_id }
   ├─ Output: { allow, risk_tier, reasons }
   │
3. If allowed:
   │  Broker → Upstream: Proxy request
   │  Broker → Audit: Log success event
   │
4. If denied:
   │  Broker → Agent: 403 + denial reasons
   │  Broker → Audit: Log denial event
```

Proof-of-Authorization (PoA) Token

The PoA token is a signed JWT mandate that authorizes a specific action:

```
{
  "sub": "spiffe://example.org/ns/default/sa/agent/connector",
  "act": "sap.vendor.change",
  "con": {
    "max_amount": 10000,
    "dual_control": true
  },
  "leg": {
    "basis": "contract",
```

```
    "jurisdiction": "US",
    "accountable_party": {
      "type": "human",
      "id": "alice@example.com"
    },
    "approval": {
      "approver": "bob@example.com",
      "timestamp": "2026-01-11T10:00:00Z"
    }
  },
  "iat": 1704067200,
  "exp": 1704067500,
  "jti": "unique-request-id"
}
```

Token Claims

Claim	Description
`sub`	SPIFFE ID of the agent
`act`	Action being authorized (e.g., `sap.vendor.change`)
`con`	Constraints (limits, conditions)
`leg`	Legal basis with accountability chain
`iat`	Issued at timestamp
`exp`	Expiration timestamp
`jti`	Unique token identifier

Risk Tiers

ATB enforces three risk tiers based on the action being performed:

Tier	Authorization	Examples
LOW	Auto-approved	Health checks, status reads, safe paths
MEDIUM	Single approval	CRM updates, data modifications
HIGH	Dual control	Financial transactions, bulk exports, privileged ops

Risk Classification

Actions are classified by prefix in the OPA policy:

```
# Low risk - auto-approved
low_risk_actions := {
    "system.status.read",
    "crm.contact.read"
}

# Medium risk - single approval
medium_risk_actions := {
    "crm.contact.update",
```

```
    "crm.lead.create"
}

# High risk - dual control
high_risk_actions := {
    "sap.payment.execute",
    "salesforce.bulk.export"
}
```

Legal Basis Framework

Every PoA token must include a legal basis (`leg` claim) that establishes:

1. **Basis** - Legal ground for the action (contract, consent, legal_obligation)
2. **Jurisdiction** - Applicable legal jurisdiction
3. **Accountable Party** - Human or system responsible
4. **Approval** - Required approvals based on risk tier

Example with dual control for HIGH risk:

```json
{
  "leg": {
    "basis": "contract",
    "jurisdiction": "US",
    "accountable_party": {
      "type": "human",
      "id": "alice@example.com"
    },
    "dual_control": {
      "approvers": [
        {"id": "bob@example.com", "timestamp": "2026-01-11T10:00:00Z"},
        {"id": "carol@example.com", "timestamp": "2026-01-11T10:05:00Z"}
      ]
    }
  }
}
```

Audit Trail

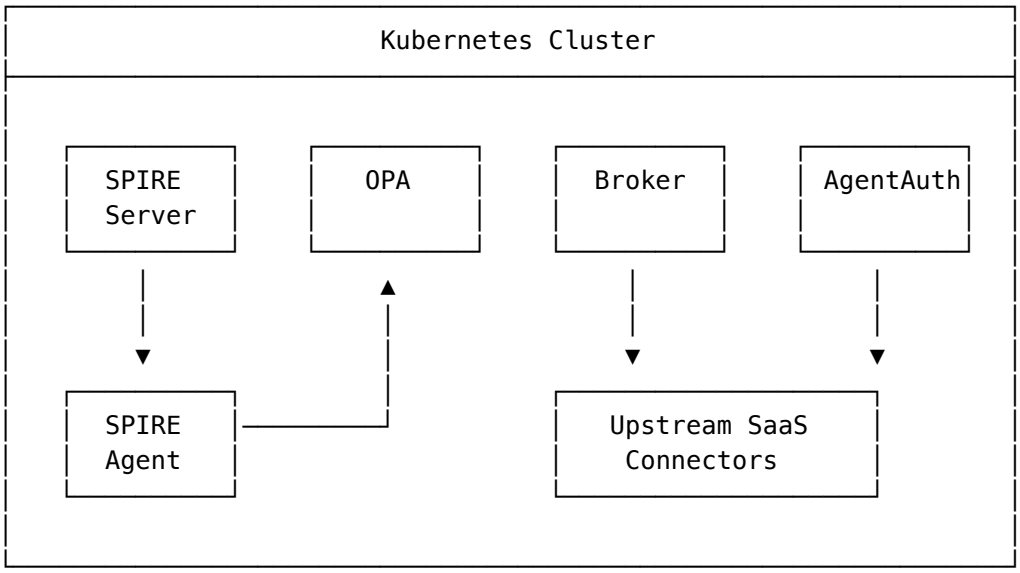
All requests are logged with:

- Timestamp
- Request ID (PoA `jti`)
- Agent identity (SPIFFE ID)
- Action attempted
- Decision (allow/deny)
- Risk tier
- Denial reasons (if any)
- Accountability chain

See audit.md for audit event schema.

Deployment Architecture

Kubernetes Deployment



Helm Chart

The charts/atb/ Helm chart deploys:

- OPA with policy ConfigMap
- ATB Broker with mTLS
- AgentAuth service
- ServiceMonitor for Prometheus
- PrometheusRules for alerting

Configuration

Connector Configuration

config/connectors.example.json defines upstream SaaS connectors:

```
{
  "connectors": [
    {
      "id": "salesforce-prod",
      "type": "salesforce",
      "upstream": "https://na1.salesforce.com",
      "actions": ["crm.contact.*", "crm.lead.*"]
    }
  ]
}
```

Federation Configuration

config/federation.example.json enables trust federation:

```
{
  "trusted_issuers": [
```

```
    {
      "issuer": "https://auth.partner.com",
      "jwks_uri": "https://auth.partner.com/.well-known/jwks.json",
      "spiffe_prefix": "spiffe://partner.com/"
    }
  ]
}
```

Security Model

Zero Trust Principles

1. **Verify explicitly** - All requests require PoA tokens
2. **Least privilege** - Tokens are scoped to specific actions
3. **Assume breach** - mTLS everywhere, short-lived tokens

Defense in Depth

1. **Transport** - mTLS with SPIFFE identity
2. **Authentication** - PoA JWT validation
3. **Authorization** - OPA policy evaluation
4. **Accountability** - Legal basis with human oversight
5. **Audit** - Complete request logging

Related Documentation

- README.md - Project overview and quick start
- Operating Model - Operational guidelines
- Enterprise Actions - Action catalog
- Requirements Compliance - Compliance mapping
- K8s Quickstart - Kubernetes deployment guide