

AgenticCommerce Enterprise API User Guide

Version 1.1.0 January 2026

CONFIDENTIAL - FOR ENTERPRISE CUSTOMERS ONLY

Table of Contents

- 1. [Introduction](#)
 - 2. [Enterprise Features Overview](#)
 - 3. [Policy Engine](#)
 - [Policies](#)
 - [Policy Rules](#)
 - [Agent Assignments](#)
 - [Policy Evaluation](#)
 - 4. [Spending Controls](#)
 - 5. [Approval Workflows](#)
 - 6. [Audit & Compliance](#)
 - 7. [X402 Enterprise Integration](#)
 - 8. [API Reference](#)
 - 9. [Best Practices](#)
 - 10. [Support](#)
-

1. Introduction

AgenticCommerce Enterprise extends the standard platform with advanced governance, compliance, and control features designed for organizations that need:

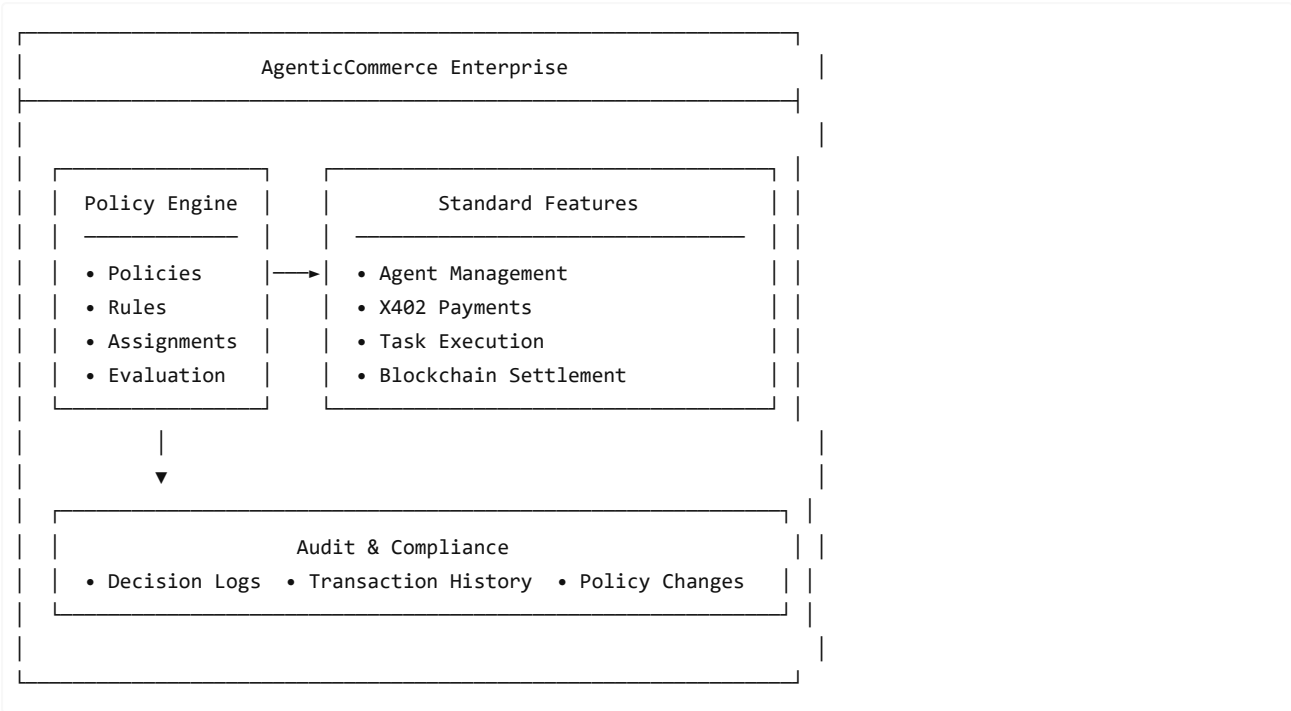
- **Spending Controls:** Set limits on agent transactions
- **Policy Engine:** Define and enforce business rules
- **Approval Workflows:** Require human approval for high-value transactions
- **Audit Trails:** Complete transaction and decision logging
- **Multi-Tenant Support:** Isolated environments per organization

Enterprise vs Standard

| Feature | Standard | Enterprise |
|--------------------|-------------|-----------------|
| AI Agents | Yes | Yes |
| X402 Payments | Yes | Yes |
| Policy Engine | No | Yes |
| Spending Limits | No | Yes |
| Approval Workflows | No | Yes |
| Audit Logging | Basic | Advanced |
| Multi-Tenant | No | Yes |
| SLA | Best Effort | 99.9% |

2. Enterprise Features Overview

Architecture



3. Policy Engine

The Policy Engine is the core enterprise feature that enables fine-grained control over agent behavior and spending.

Policies

A **Policy** is a container for rules that govern agent behavior. Policies can be assigned to specific agents or applied organization-wide.

Create a Policy

```
curl -X POST https://localhost:7098/api/policies \
-H "Content-Type: application/json" \
-d '{
  "name": "Standard Agent Policy",
  "description": "Default spending limits for standard agents",
  "organizationId": "org_abc123",
  "isActive": true
}'
```

Response:

```
{
  "id": "policy_xyz789",
  "name": "Standard Agent Policy",
  "description": "Default spending limits for standard agents",
  "organizationId": "org_abc123",
  "isActive": true,
  "createdAt": "2026-01-18T10:00:00Z",
  "updatedAt": "2026-01-18T10:00:00Z",
  "rules": []
}
```

List Policies

```
curl https://localhost:7098/api/policies
```

Get Policy Details

```
curl https://localhost:7098/api/policies/{policyId}
```

Update Policy

```
curl -X PUT https://localhost:7098/api/policies/{policyId} \
-H "Content-Type: application/json" \
-d '{
  "name": "Updated Policy Name",
  "isActive": false
}'
```

Delete Policy

```
curl -X DELETE https://localhost:7098/api/policies/{policyId}
```

Policy Rules

Rules define the specific constraints within a policy. Multiple rules can be combined in a single policy.

Rule Types

| Rule Type | Description |
|---------------------|--|
| SpendingLimit | Maximum transaction and daily spend limits |
| ResourceRestriction | Allowed/blocked API endpoints |
| TimeWindow | Allowed transaction hours |
| ApprovalRequired | Transactions requiring human approval |
| AgentRestriction | Specific agent permissions |

Add a Spending Limit Rule

```
curl -X POST https://localhost:7098/api/policies/{policyId}/rules \
-H "Content-Type: application/json" \
-d '{
  "ruleType": "SpendingLimit",
  "name": "Daily Spending Cap",
  "parameters": {
    "maxAmountPerTransaction": 1.0,
    "maxDailySpend": 10.0,
    "currency": "USDC"
  }
}'
```

Add a Resource Restriction Rule

```
curl -X POST https://localhost:7098/api/policies/{policyId}/rules \
-H "Content-Type: application/json" \
-d '{
  "ruleType": "ResourceRestriction",
  "name": "Allowed APIs",
  "parameters": {
    "allowedResources": [
      "/api/x402-example/*",
      "/api/data/*"
    ],
    "blockedResources": [
      "/api/admin/*"
    ]
  }
}'
```

Add an Approval Required Rule

```
curl -X POST https://localhost:7098/api/policies/{policyId}/rules \
-H "Content-Type: application/json" \
-d '{
  "ruleType": "ApprovalRequired",
  "name": "High Value Approval",
  "parameters": {
    "thresholdAmount": 5.0,
    "approvers": ["admin@company.com"],
    "expirationMinutes": 60
  }
}'
```

Get Policy Rules

```
curl https://localhost:7098/api/policies/{policyId}/rules
```

Agent Assignments

Policies must be assigned to agents to take effect.

Assign Policy to Agent

```
curl -X POST https://localhost:7098/api/policies/{policyId}/assign/{agentId}
```

Response:

```
{
  "policyId": "policy_xyz789",
  "agentId": "agent_abc123",
  "assignedAt": "2026-01-18T10:00:00Z",
  "assignedBy": "admin"
}
```

Remove Policy from Agent

```
curl -X DELETE https://localhost:7098/api/policies/{policyId}/assign/{agentId}
```

List Agent Policies

```
curl https://localhost:7098/api/agents/{agentId}/policies
```

Policy Evaluation

Before any transaction, the Policy Engine evaluates all applicable rules.

Manual Policy Evaluation

Test policy evaluation without executing a transaction:

```
curl -X POST https://localhost:7098/api/policies/evaluate \
-H "Content-Type: application/json" \
-d '{
  "agentId": "agent_abc123",
  "resource": "/api/x402-example/simple",
  "amount": 0.01,
  "network": "arc-testnet"
}'
```

Response (Approved):

```
{
  "decision": "Approved",
  "policyId": "policy_xyz789",
  "evaluatedRules": [
    {
      "ruleType": "SpendingLimit",
      "result": "Pass",
      "details": "Amount $0.01 within limit $1.00"
    }
  ],
  "evaluatedAt": "2026-01-18T10:00:00Z"
}
```

Response (Denied):

```
{
  "decision": "Denied",
  "policyId": "policy_xyz789",
  "reason": "Daily spending limit exceeded",
  "evaluatedRules": [
    {
      "ruleType": "SpendingLimit",
      "result": "Fail",
      "details": "Daily spend $10.50 exceeds limit $10.00"
    }
  ],
}
```

```
"evaluatedAt": "2026-01-18T10:00:00Z"
}
```

4. Spending Controls

Overview

Spending controls help organizations manage agent budgets and prevent unauthorized transactions.

Spending Limit Configuration

```
{
  "ruleType": "SpendingLimit",
  "parameters": {
    "maxAmountPerTransaction": 1.0,
    "maxDailySpend": 10.0,
    "maxWeeklySpend": 50.0,
    "maxMonthlySpend": 200.0,
    "currency": "USDC",
    "resetTime": "00:00:00 UTC"
  }
}
```

Get Agent Spending Summary

```
curl https://localhost:7098/api/policies/spending/{agentId}
```

Response:

```
{
  "agentId": "agent_abc123",
  "currentBalance": 8.50,
  "spending": {
    "today": 1.50,
    "thisWeek": 5.00,
    "thisMonth": 15.00,
    "allTime": 150.00
  },
  "limits": {
    "perTransaction": 1.0,
    "daily": 10.0,
    "weekly": 50.0,
    "monthly": 200.0
  },
  "remaining": {
    "daily": 8.50,
    "weekly": 45.00,
    "monthly": 185.00
  }
}
```

5. Approval Workflows

For high-value or sensitive transactions, you can require human approval.

Configure Approval Workflow

```
{
  "ruleType": "ApprovalRequired",
  "parameters": {
    "thresholdAmount": 5.0,
    "approvers": [
      "finance@company.com",
      "manager@company.com"
    ],
    "requiredApprovals": 1,
    "expirationMinutes": 60,
    "notificationChannels": ["email", "slack"]
  }
}
```

Pending Approvals

When a transaction requires approval, it enters a pending state:

```
curl https://localhost:7098/api/policies/approvals/pending
```

Response:

```
[
  {
    "id": "approval_123",
    "agentId": "agent_abc123",
    "resource": "/api/premium-data",
    "amount": 10.0,
    "requestedAt": "2026-01-18T10:00:00Z",
    "expiresAt": "2026-01-18T11:00:00Z",
    "status": "Pending",
    "approvers": ["finance@company.com"]
  }
]
```

Approve Transaction

```
curl -X POST https://localhost:7098/api/policies/approvals/{approvalId}/approve \
-H "Content-Type: application/json" \
-d '{
  "approver": "finance@company.com",
  "notes": "Approved for Q1 research project"
}'
```

Reject Transaction

```
curl -X POST https://localhost:7098/api/policies/approvals/{approvalId}/reject \
-H "Content-Type: application/json" \
-d '{
  "approver": "finance@company.com",
  "reason": "Over budget for this quarter"
}'
```

6. Audit & Compliance

Decision Logs

Every policy evaluation is logged for compliance:

```
curl https://localhost:7098/api/policies/decisions?agentId={agentId}&limit=100
```

Response:

```
[
  {
    "id": "decision_456",
    "agentId": "agent_abc123",
    "policyId": "policy_xyz789",
    "resource": "/api/x402-example/simple",
    "amount": 0.01,
    "decision": "Approved",
    "evaluatedRules": [...],
    "timestamp": "2026-01-18T10:00:00Z",
    "transactionId": "tx_abc123"
  }
]
```

Export Audit Report

```
curl https://localhost:7098/api/policies/audit/export \
-H "Accept: application/json" \
-d '{
  "startDate": "2026-01-01",
  "endDate": "2026-01-31",
  "format": "json"
}'
```

Compliance Dashboard Data

```
curl https://localhost:7098/api/policies/compliance/summary
```

Response:

```
{
  "period": "2026-01",
  "totalTransactions": 1500,
  "approvedTransactions": 1450,
```



```

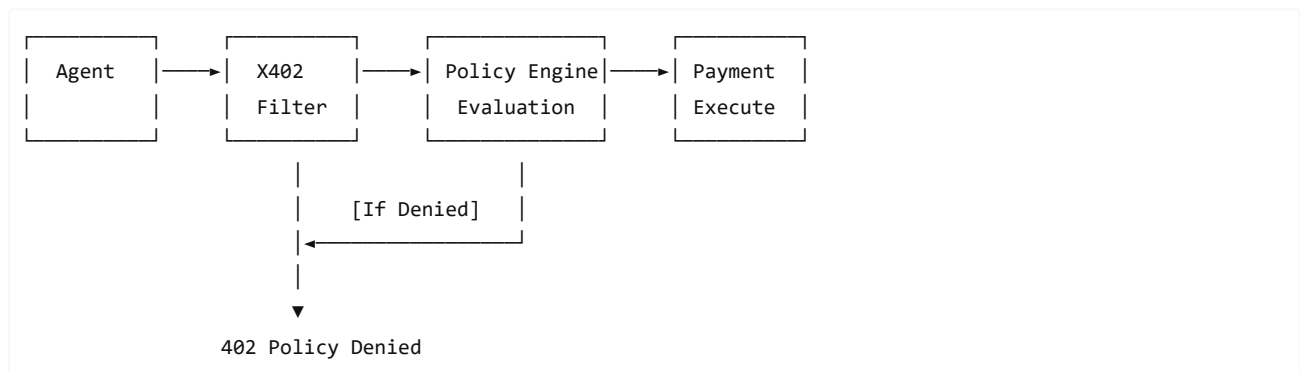
"deniedTransactions": 45,
"pendingApprovals": 5,
"totalSpend": 125.50,
"topAgents": [
  { "agentId": "agent_abc123", "spend": 50.00, "transactions": 500 }
],
"policyViolations": [
  { "type": "SpendingLimit", "count": 30 },
  { "type": "ResourceRestriction", "count": 15 }
]
}

```

7. X402 Enterprise Integration

The Policy Engine integrates with X402 payments to enforce policies at the payment layer.

X402 + Policy Flow



Policy Evaluation in X402

When an agent makes a paid API call:

1. Agent calls `Http.GetWithAutoPay(endpoint, budget)`
2. Server returns 402 Payment Required
3. Agent signs EIP-3009 authorization
4. Agent sends X-PAYMENT header
5. **Policy Engine evaluates the payment**
6. If approved: Payment settles, 200 response
7. If denied: 402 with policy denial reason

Policy Denial Response

```

{
  "x402Version": 2,
  "error": "PolicyDenied",
  "reason": "Daily spending limit exceeded",
  "policyId": "policy_xyz789",
  "agentId": "agent_abc123",
  "evaluatedAt": "2026-01-18T10:00:00Z"
}

```

8. API Reference

Policy Endpoints

| Method | Endpoint | Description |
|--------|-------------------------------------|----------------------|
| GET | /api/policies | List all policies |
| POST | /api/policies | Create a policy |
| GET | /api/policies/{id} | Get policy details |
| PUT | /api/policies/{id} | Update a policy |
| DELETE | /api/policies/{id} | Delete a policy |
| GET | /api/policies/{id}/rules | List policy rules |
| POST | /api/policies/{id}/rules | Add a rule |
| DELETE | /api/policies/{id}/rules/{ruleId} | Remove a rule |
| POST | /api/policies/{id}/assign/{agentId} | Assign to agent |
| DELETE | /api/policies/{id}/assign/{agentId} | Remove from agent |
| POST | /api/policies/evaluate | Evaluate policy |
| GET | /api/policies/spending/{agentId} | Get spending summary |
| GET | /api/policies/decisions | Get decision logs |

Request Headers

| Header | Description |
|-------------------|---------------------------------|
| Content-Type | application/json |
| X-Organization-Id | Your organization identifier |
| X-API-Key | Enterprise API key (if enabled) |

Rate Limits (Enterprise)

| Endpoint | Limit |
|-------------------|-------------|
| Policy management | 100/minute |
| Policy evaluation | 1000/minute |
| Audit queries | 50/minute |

9. Best Practices

Policy Design

- Start Restrictive:** Begin with strict limits, then relax as needed
- Layered Policies:** Use organization-wide defaults + agent-specific overrides
- Test First:** Use the `/evaluate` endpoint before assigning policies
- Monitor Denials:** High denial rates may indicate limits are too strict

Spending Controls

1. **Set Multiple Limits:** Per-transaction, daily, weekly, monthly
2. **Buffer Amounts:** Leave headroom for legitimate burst usage
3. **Review Regularly:** Adjust limits based on actual usage patterns

Approval Workflows

1. **Multiple Approvers:** Require at least 2 approvers for high-value transactions
2. **Short Expiration:** Set approvals to expire in 1-4 hours
3. **Notification Channels:** Enable email + Slack for faster response

Audit & Compliance

1. **Regular Exports:** Schedule weekly/monthly audit exports
 2. **Retention Policy:** Keep decision logs for at least 1 year
 3. **Anomaly Alerts:** Set up alerts for unusual spending patterns
-

10. Support

Enterprise Support Channels

| Channel | Response Time | Hours |
|-----------|---------------|----------------|
| Email | 4 hours | 24/7 |
| Phone | 1 hour | Business hours |
| Slack | 30 minutes | Business hours |
| Emergency | 15 minutes | 24/7 |

Contact

- **Enterprise Support:** enterprise@agenticcommerce.com
- **Account Manager:** Your dedicated contact
- **Status Page:** <https://status.agenticcommerce.com>
- **Documentation:** <https://docs.agenticcommerce.com/enterprise>

SLA

Enterprise customers are covered by our 99.9% uptime SLA. See your contract for details.

CONFIDENTIAL

This document contains proprietary information about AgenticCommerce Enterprise features. Distribution is limited to authorized enterprise customers and partners.
