Enterprise-Ready MCP

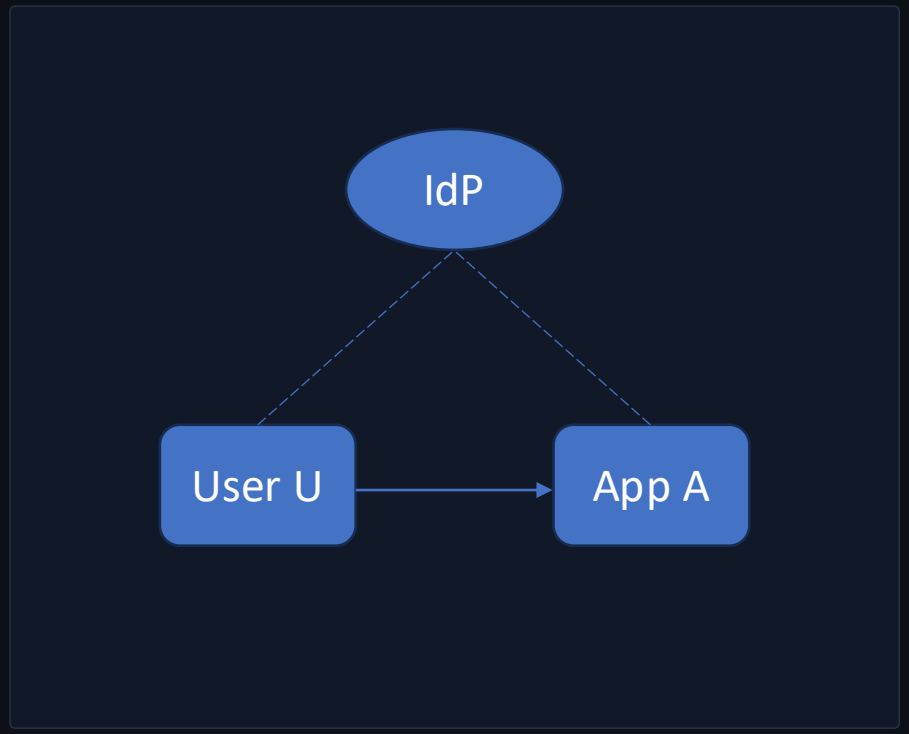
Security Patterns & the "4-Legged" Identity Challenge

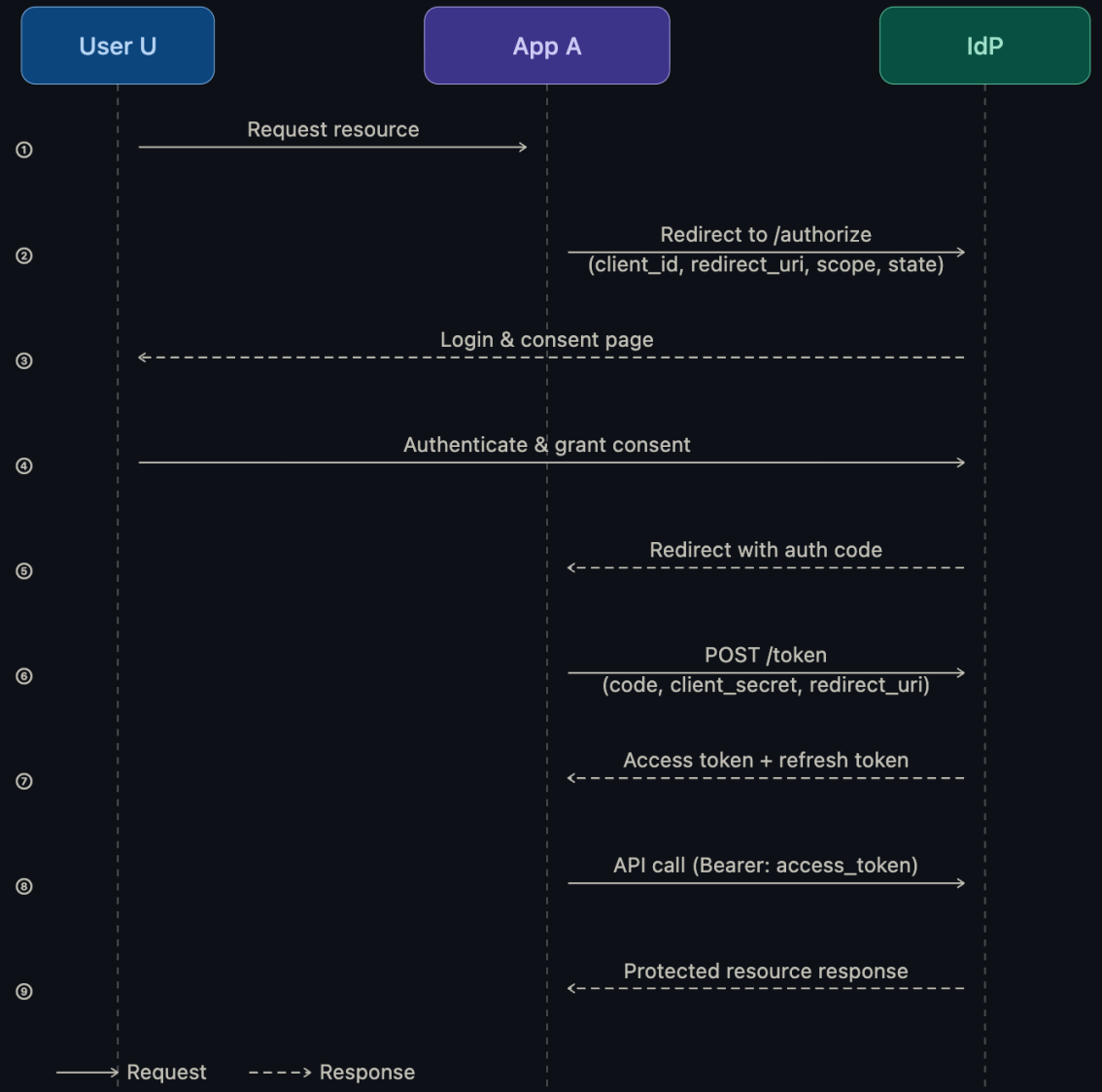
in Modern Agentic Systems

Paulina Xu
https://www.agenticfabriq.com

The Core Problem

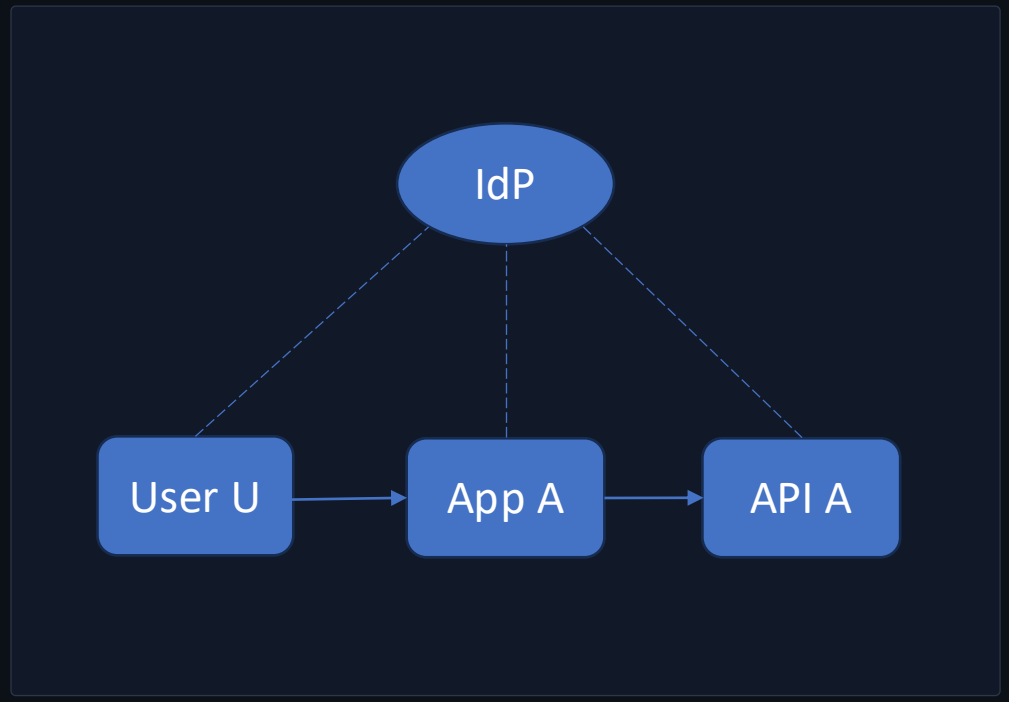
Why Standard OAuth Breaks in Agentic Systems



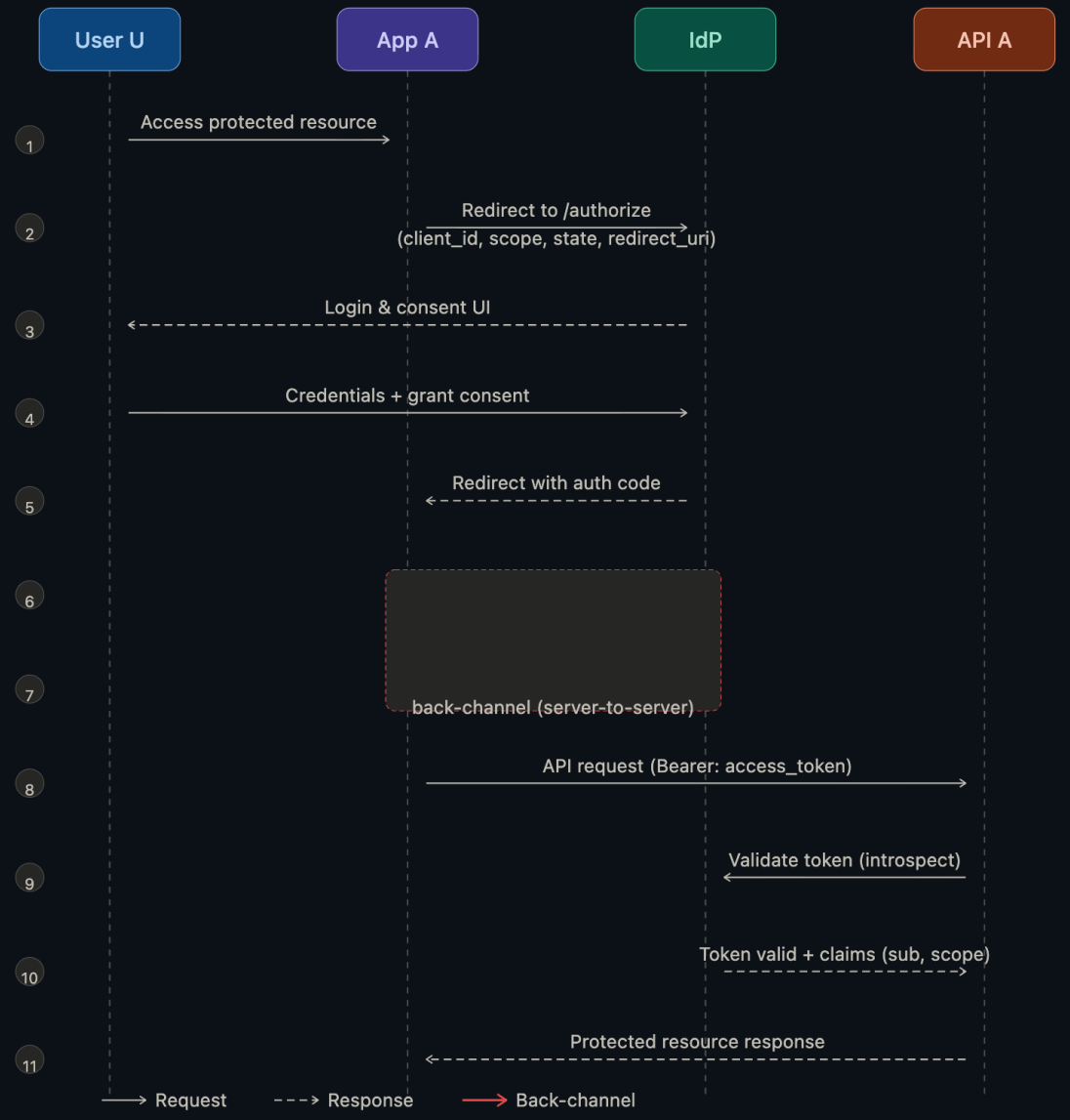


The Core Problem

Why Standard OAuth Breaks in Agentic Systems



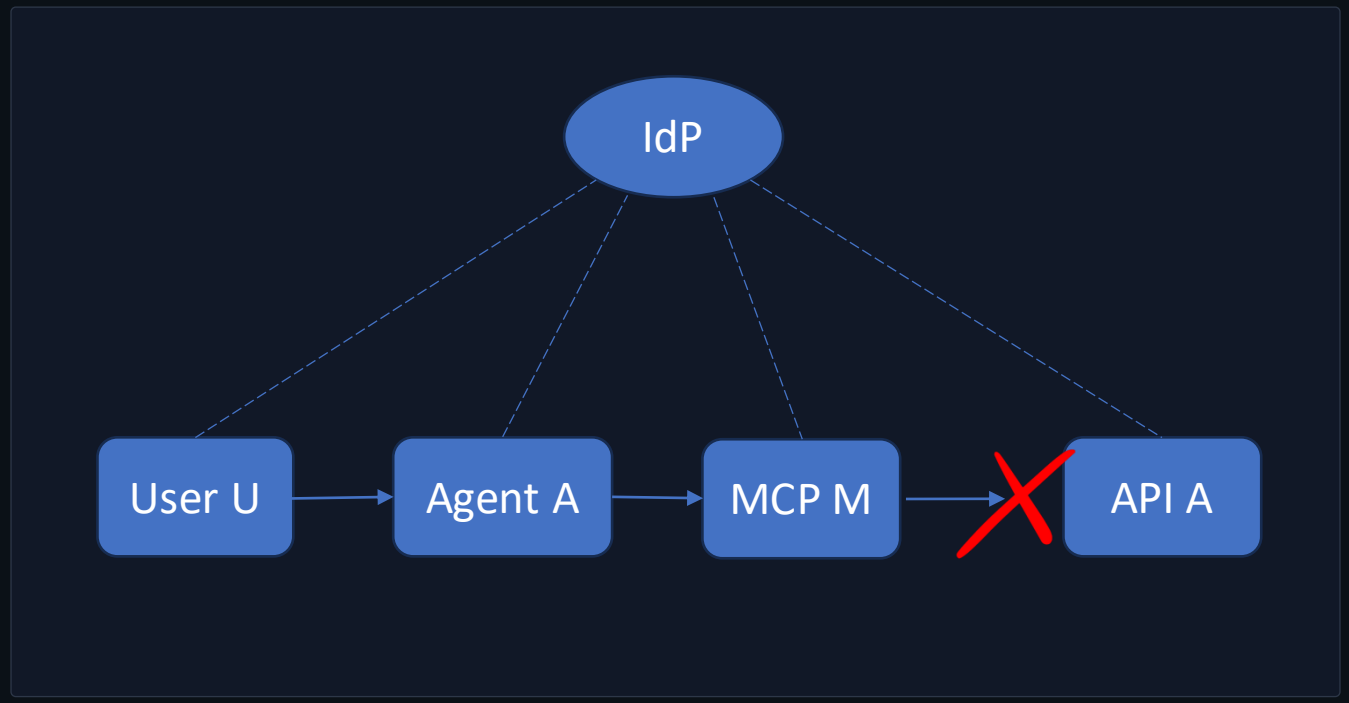
3-legged OAuth flow



The Core Problem

Why Standard OAuth Breaks in Agentic Systems

4 - legged



The 4-Legged OAuth with MCP

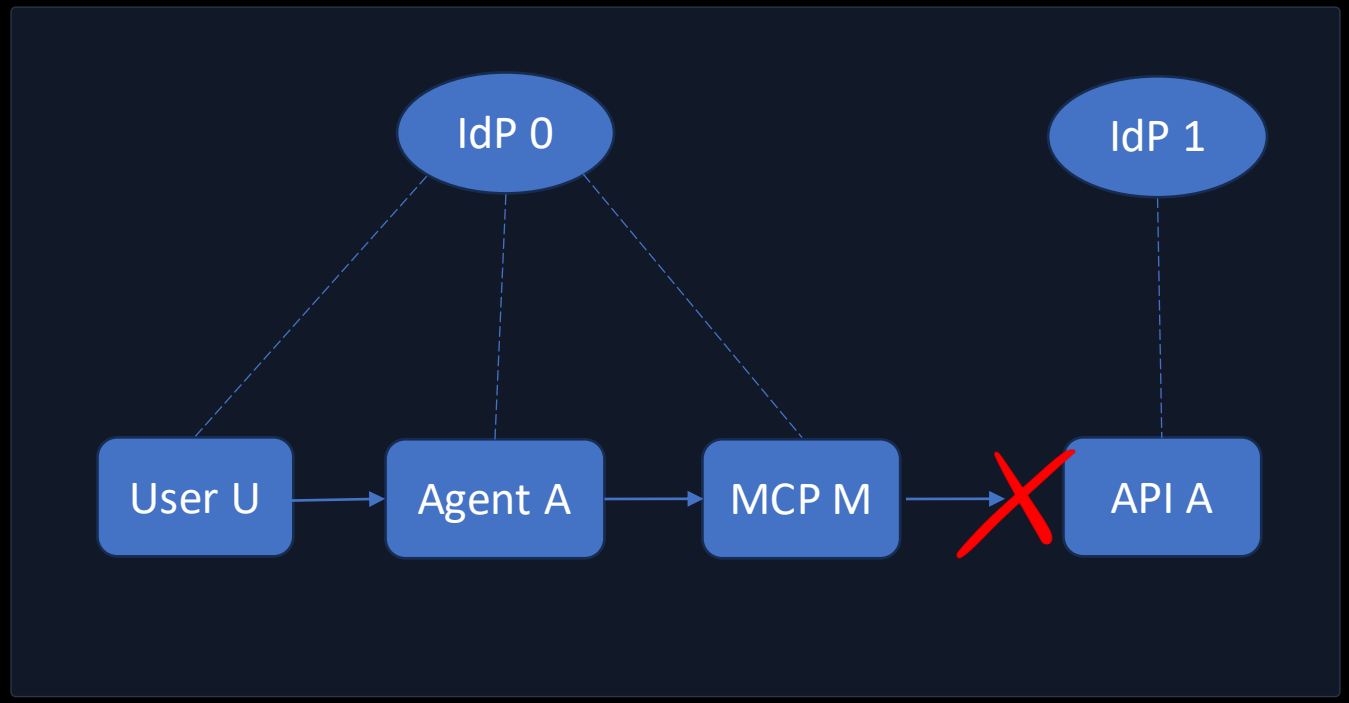
- Tokens target a specific audience — Token for A can't be used with B
- Who does B see — the User or the Agent? Identity is lost mid-chain
- 4-legged OAuth (U→A→M→A) is not natively supported
- TokenExchange RFC8693 was proposed

Result: Identity lost mid-chain

The Core Problem

Why Standard OAuth Breaks in Agentic Systems

4+ - legged



OAuth + Federation

- Can be more complicated for Agent to Agent communication
  - The chain will grow longer beyond 4 legs
  - Will have multiple IdP involved

How do we preserve identity coherently across heterogeneous systems, over multiple hops, potentially across time?

Solving 4-Legged Auth

Choose Based on Your IdP Capabilities & Delegation Requirements

1 Non-Delegating (Client Creds)

Low Complexity

- Simple, widely supported
- User identity hidden from downstream APIs

Use when: Downstream doesn't require user identity

2 Token Exchange (RFC 8693)

Medium Complexity

- Preserves user identity end-to-end
- Limited IdP support

Use when: User identify must follow all the way to MCP Server

3 Pre-injected Token

Medium Complexity

- Works with any IdP
- Agent must know all downstream APIs upfront

Use when: Target IdP doesn't support Token Exchange

4 Out-of-Band Login

Medium Complexity

- Full user consent for each service
- Requires extra browser tab and session coordination

Use when: Interactive user flows

RFC 8693 Token Exchange

Token Exchange Request (POST /token)

```
grant_type = urn:ietf:params:oauth:grant-type:token-exchange
subject_token = <user's current access token>
subject_token_type = urn:ietf:...:access_token
requested_token_type = urn:ietf:...:access_token
audience = resource-app (or third-app)
client_id + client_secret = <requester client creds>
```

IdP Support Landscape

Keycloak 26.4	✓ Full
Microsoft Entra ID	~ Not Supported
Okta Claims support	~ Partial

Four RFCs that complete the OAuth story

Each RFC fills a gap in how clients, servers, and APIs discover and trust each other — all now required by MCP.

RFC 9728

Resource Metadata

APIs describe themselves — completing the discovery trio.

RFC 8414

AS Metadata

Auth servers advertise their endpoints and capabilities.

RFC 7591

Dynamic Client Reg.

Clients register themselves at runtime — no manual setup.

RFC 8693

Token Exchange

Trade one token for another — scoped, delegated, or impersonated.

MCP mandates all four. RFC 9728 is required for MCP servers · RFC 8414 for AS discovery · RFC 7591 for dynamic agent registration · RFC 8693 for agent delegation chains.

How MCP uses all four RFCs

MCP SENARIO		RFC MECHANISM
Agent discovers a new MCP server for the first time	→	RFC 9728 resource metadata at /.well-known/oauth-protected-resource
Agent needs to know where to get tokens	→	RFC 8414 AS metadata from authorization_servers in resource doc
Agent has no pre-registered client_id	→	RFC 7591 POST to registration_endpoint with agent metadata + JWKS
MCP server calls an upstream API on user's behalf	→	RFC 8693 token exchange — scoped token with act delegation claim
Downstream API rejects token — insufficient scope	→	RFC 9728 403 with scope in WWW-Authenticate → re-request with wider scope
Orchestrator spawns a sub-agent with less privilege	→	RFC 8693 may_act claim restricts who can further exchange the token

The full zero-to-token journey for an MCP agent: **9728 → 8414 → 7591 → OAuth flow → (8693 if delegating)**

The cheat sheet

RFC 7591 · Dynamic Client Registration

When?

Agent encounters a new AS and needs a **client_id** without manual provisioning.

POST /register → client_id
Key: software_statement, jwks_uri

RFC 8414 · AS Metadata

When?

Bootstrap discovery of any AS — where to authorize, get tokens, fetch keys.

GET /.well-known/oauth-authorization-server
Key: issuer, token_endpoint, jwks_uri

RFC 8693 · Token Exchange

When?

A service needs to call another service, carrying user context with controlled privilege.

grant_type=token-exchange
Key: act claim, may_act claim, audience

RFC 9728 · Protected Resource Metadata

When?

Client hits an API and needs to know which AS to use and what scopes to request.

GET /.well-known/oauth-protected-resource
Key: authorization_servers, scopes_supported

Enterprise Security Principles

Least Privilege

- Agents act **on behalf of users**, never with broad service-level credentials
- Use **short-lived, tightly scoped tokens** with 3-legged OAuth where possible
- **Revalidate identity and scope at every hop** – gateway and each MCP server

Encryption at Rest

- Encrypt data **in transit and at rest** across agents, MCP servers, and APIs
- Use **envelope encryption**: transient DEKs, KEKs stored securely in Vault
- Limit blast radius with **separate keys, secure key handling, and rotation**

Enterprise Security Principles

Audit & Observability

- Log every tool call with full context: **user, agent, tool, inputs, outcome**
- Scan outbound responses with **DLP** to catch PII, secrets, and sensitive data
- Maintain **end-to-end tracing** across user → agent → MCP → API

High Availability

- Treat agent systems as **infrastructure**, not just app features
- Design for **scaling, failover, and resilience** across the full chain
- Use **rolling updates, autoscaling, and circuit breakers** to contain failures

Enterprise Security Principles

Auth Hardening

- Enforce **PKCE on all flows** to prevent token interception
- **Validate JWTs at multiple layers** (gateway + MCP server)
- Strictly enforce **audience (aud) + mTLS** between services

Zero Secrets Exposure

- **No plaintext secrets** – all managed via Vault / KMS
- Use **just-in-time, time-limited access** with full audit logs
- **No hardcoding or commits** – rotate secrets frequently

Key Takeaways

What You Should Take Away Today

01 **4-Legged Identity is the Default**

Not an edge case – every agentic system becomes multi-hop.

02 **Oauth is Incomplete for Agents**

You need token exchange (RFC 8693) for correct delegation.

03 **Broker Patterns Are Required at Scale**

Centralize token exchange, caching, and policy enforcement.

04 **Security Must Be Layered**

Identity + validation + encryption + observability + secrets

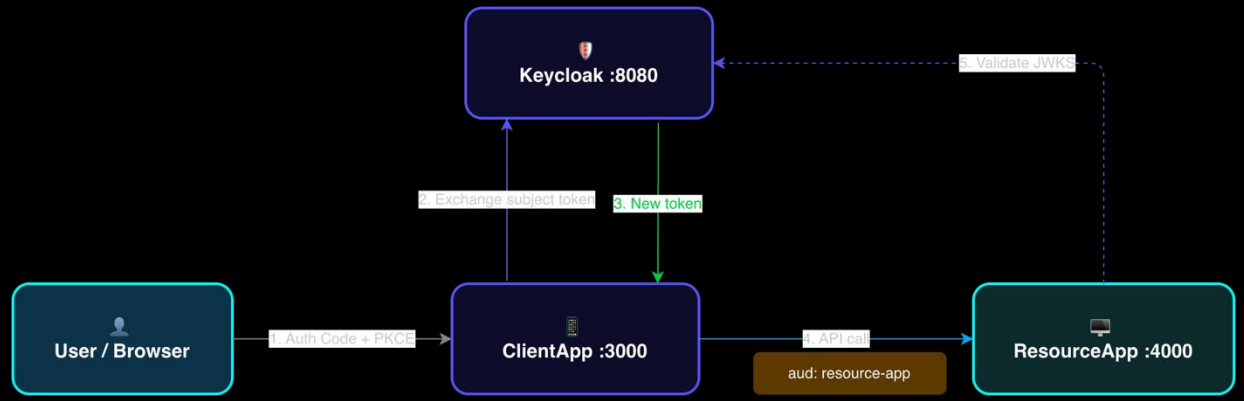
Hands-on Labs

Start with Lab 1

Extend to Lab 2

- Available on GitHub
- Questions welcome!
- Keycloak · OpenBao · PostgreSQL

Lab 1: Standard Token Exchange



ClientApp · OAuth Token Exchange Lab

Walk through the Authorization Code + PKCE login, exchange the user token for a ResourceApp audience token, and call the protected API with full visibility into each step.

1) Login | 2) Exchange | 3) Call ResourceApp | 4) Via ExchangeApp | Logout | Revoke consent

Step 1 · Login with Keycloak

Kick off Authorization Code + PKCE with a single click. Use the tabs above to advance through each stage.

Launch Login

Step 1 Complete · Tokens received

These tokens were issued to ClientApp and are stored in-session for the downstream exchange.

ID Token

JWT

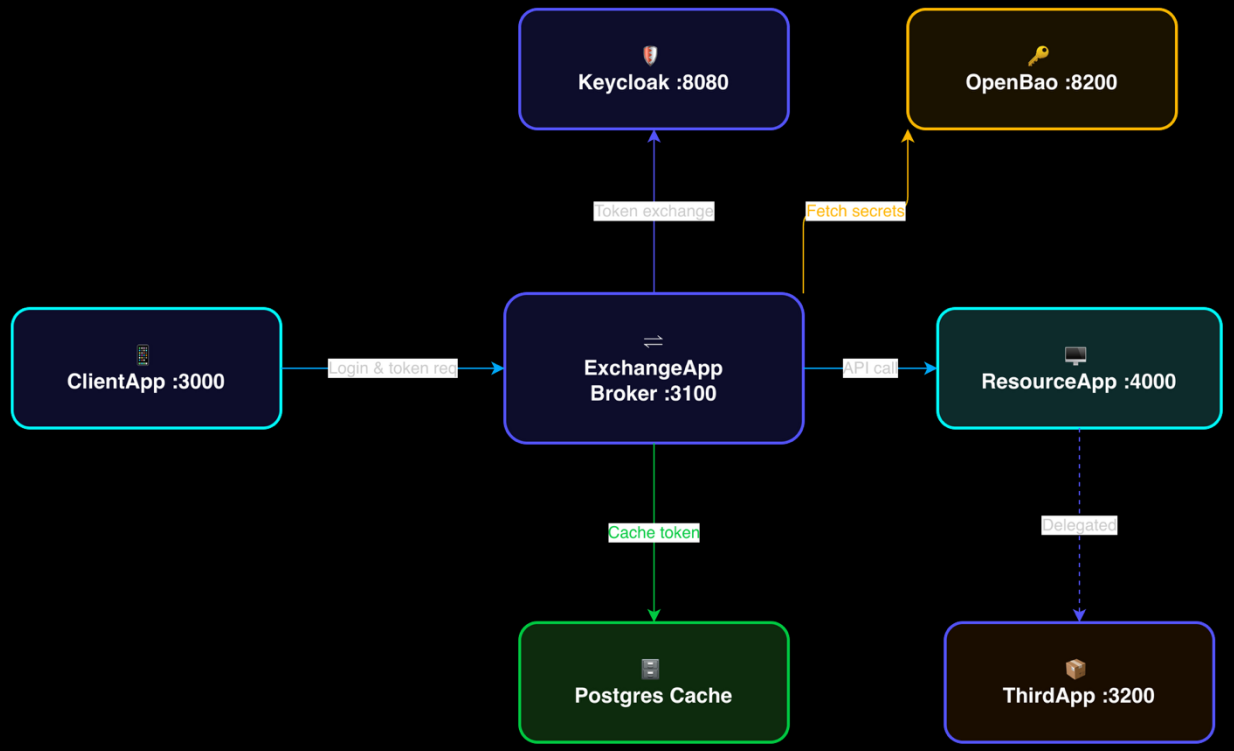
eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICIzVkJkX21uT2daaWVSYlRvWUxZSC0wRFZTdE5NbEcwZWQ5cWxkWXU4RGRJIn0.eyJleHAiOjE3NzUwOTU0ODcsImlhdCI6MTc3NTA5NTE4NywiYXV0aF90aWlIjoxNzc1MDk1MTg3LCJqdGkiOiJlYTBkYjE4MS1jMmMxLTQ4ZDItMGRkNy1mMTA3MDA2OWJjNWEiLCJpc3MiOiJodHRwOi8vbG9jYWxob3N0OjgwODAvcmVhbG1zL29iby1sYWIiLCJhdWQiOiJjbGllbnQtYXBwIiwic3ViIjoiNTVlYmM2ZGEtYjM2NS00ZGU2LTg2OWQtYTY5NGFiY2JjNTg4IiwidHlwIjoiSUQiLCJhenAiOiJjbGllbnQtYXBwIiwic2lkIjoiMTA2YmJjODEtODY1NC1hYTcwLWJmYTktZmY4OGY0ZmRjMzRjIiwiYXRfaGFzaCI6InpUdG5hQzRfOWJDbHY4NWhINExHZ1EiLCJhY3IiOiIxIiwiZW1haWxfdmVyaWZpZWQiOnRydWUsIm5hbWUiOiJhbGljZSB3b25kZXIiLCJwcmVmZXJyZWRfdXNlcm5hbWUiOiJhbGljZSIsImdpdmVuX25hbWUiOiJhbGljZSIsImZhbWlseV9uYW1lIjoid29uZGVyIiwiZW1haWwiOiJhbGljZUBmYWJyaXEuY29tIn0.Rxlvn4QDYZ0QEtEup-WpOM_EZ2DiMy4S5i13SuzBn52FyCh_S-CdX1wD4RsbcXGRrX-tbjmNeJ8O-6j-66oMKZO0zOOmi7MA8pjOtNMk-7Wb46mWBca6Xt7TPqI7-IaT-87EuElKfcWNPDofqxbHL3bIsFZpdD5VhaG55nVBkRFodLpFHhbAjqL0Yg4ACcwwD01RiJwA5FChjcG8aIIr1Yd48uGwSKQU06EFfFnBA67JKV08mfYRKH6a-jlaSAEq3MGkqyQp1Z2b0ekBensOhKSFpQkL5tRgy2hdMYHhadpS-hBkQun640BcDn7OT3dK0hy225FXm31yV4aew4IcQg

DECODED

{
  "header": {
    "alg": "RS256",
    "typ": "JWT",
    "kid": "3VBd_mnOgZieRbToYLYH-0DVStNMlG0ed9qldYu8DdI"
  },
  "payload": {
    "exp": 1775095487,
    "iat": 1775095187,
    "auth_time": 1775095187,
    "jti": "ea0db181-c2c1-48d2-0dd7-f1070069bc5a",
    "iss": "http://localhost:8080/realms/obo-lab",
    "aud": "client-app",
    "sub": "55ebc6da-b336-4de6-869d-a694abcbc588",
    "typ": "ID",
    "azp": "client-app",
    "sid": "106bbc81-8654-aa70-bfa9-ff88f4fdc34c",
    "at_hash": "zTtnaC4_9bClv85hH4LGgQ",
    "acr": "1",
    "email_verified": true,
    "name": "alice wonder",
    "preferred_username": "alice",
    "given_name": "alice",
    "family_name": "wonder",
    "email": "alice@fabriq.com"
  },
  "signature": "Rxlvn4QDYZ0QEtEup-WpOM_EZ2DiMy4S5i13SuzBn52FyCh_S-CdX1wD4RsbcXGRrX-tbjmNeJ8O-6j-66oMKZO0zOOmi7MA8pjOtNMk-7Wb46mWBca6Xt7TPqI7-IaT-87EuElKfcWNPDofqxbHL3bIsFZpdD5VhaG55nVBkRFodLpFHhbAjqL0Yg4ACcwwD01RiJwA5FChjcG8aIIr1Yd48uGwSKQU06EFfFnBA67JKV08mfYRKH6a-jlaSAEq3MGkqyQp1Z2b0ekBensOhKSFpQkL5tRgy2hdMYHhadpS-hBkQun640BcDn7OT3dK0hy225FXm31yV4aew4IcQg"
}

Lab 2: Brokered Exchange Architecture

What You Should Take Away Today



ExchangeApp - OAuth Token Exchange Lab

Logout

Acts as the middle tier for token exchange and caching. Login with user credentials, exchange tokens for ResourceApp or ThirdApp, proxy downstream calls, and observe Postgres-backed caches and audit trails in real time.

Exchanges & Proxies

Exchange ResourceApp | Exchange ThirdApp | Proxy ResourceApp | Proxy ThirdApp

Output


```
{
  "status": 200,
  "headers": {
    "connection": "keep-alive",
    "content-length": "4423",
    "content-type": "application/json; charset=utf-8",
    "date": "Thu, 02 Apr 2026 02:43:55 GMT",
    "etag": "W/\"1147-Ec8S/CVfOK88CiMWleEk4bHeUyc\"",
    "keep-alive": "timeout=5",
    "x-powered-by": "Express"
  },
  "body": {
    "ok": true,
    "message": "Resource data returned successfully",
    "user": "alice",
    "debug": {
      "receivedAt": "2026-04-02T02:43:55.919Z",
      "requestHeaders": {
        "accept": "*/*",
        "accept-encoding": "gzip, deflate, br",
        "authorization": "Bearer
eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICIzVkJkX21uT2daaWVSYlRvWUxZSC0wRFZTdE5NbEcwZWQ5cWxkWXU4RGRJIn0.eyJleH
AiOjE3NzUwOTgxMzUsImlhdCI6MTc3NTA5NzgzNSwiYXV0aF90aW1lIjoxNzc1MDk3NjM5LCJqdGkiOiJvbnJ0dGU6OGZiODdhYmUtMGE0My0yZmVhLTQy
ZjktOTZjMTBkNTVhNGVhIiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdDo4MDgwL3JlYWxtcy9vYm8tbGFiIiwiYXVkIjoicmVzb3VyY2UtYXBwIiwic3ViIj
oiNTVlYmM2ZGEtYjMzNi00ZGU2LTg2OWQtYTY5NGFiY2JjNTg4IiwidHlwIjoiQmVhcmVyIiwiYXpwIjoiZXhjaGFuZ2UtYXBwIiwic2lkIjoiMTk3Y2Vh
YWQtNDE5NC0yZDhlLWMyODktNjJiMzljZWJjMWQxIiwiYWNyIjoiMSIsImFsbG93ZWQtb3JpZ2lucyI6WyJodHRwOi8vbG9jYWxob3N0OjMxMDAiXSwicm
VhbG1fYWNjZXNzIjp7InJvbGVzIjpbImRlZmF1bHQtcm9sZXMtb2JvLWxhYiIsIm9mZmxpbmVfYWNjZXNzIiwidW1hX2F1dGhvcml6YXRpb24iXX0sInJl
c291cmNlX2FjY2VzcyI6e30sInNjb3BlIjoicHJvZmlsZSBlbWFpbCIsImVtYWlsX3ZlcmlmaWVkIjp0cnVlLCJuYW1lIjoiYWxpY2Ugd29uZGVyIiwicH
JlZmVycmVkX3VzZXJuYW1lIjoiYWxpY2UiLCJnaXZlbl9uYW1lIjoiYWxpY2UiLCJmYW1pbHlfbmFtZSI6IndvbmRlciIsImVtYWlsIjoiYWxpY2VAZmFi
cmlxLmNvbSJ9.2WNE0HzTndKymhhe23CXWdgFyB1nJepDfOmTOAorGSjvI76qUoOV9BMAiiWvqBRmYQYdhr96kjLWYpXEjn80g61NkfNYc-
KiE99Cqbua3NOrMZXDBd_I7pVTVXa17OF5dx2-RrQI5DIt1Q814T6eUsYfg-5JGYfW7x2G6ZFJFsvlJ0qRGiF7xirHeuWwz4QjBOW4vfuN-
fI6klbP2RHmJZ1QPGiy5rbLbAw1TjUSxVxOv-oeyLJ0L319vQZHpuTLu0tg6BIR_2GrVzB-
WltGvJuJ38HyKIsy8paNxRaYlkW2RZKxfUUWClgpsuJS6eOQf-KLIU3QObgOedQ-cfaKnw",
        "user-agent": "node-fetch",
        "host": "localhost:4000",
        "connection": "keep-alive"
      },
      "tokenRaw":
"eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICIzVkJkX21uT2daaWVSYlRvWUxZSC0wRFZTdE5NbEcwZWQ5cWxkWXU4RGRJIn0.eyJle
HAiOjE3NzUwOTgxMzUsImlhdCI6MTc3NTA5NzgzNSwiYXV0aF90aW1lIjoxNzc1MDk3NjM5LCJqdGkiOiJvbnJ0dGU6OGZiODdhYmUtMGE0My0yZmVhLTQ
yZjktOTZjMTBkNTVhNGVhIiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdDo4MDgwL3JlYWxtcy9vYm8tbGFiIiwiYXVkIjoicmVzb3VyY2UtYXBwIiwic3ViI
joiNTVlYmM2ZGEtYjMzNi00ZGU2LTg2OWQtYTY5NGFiY2JjNTg4IiwidHlwIjoiQmVhcmVyIiwiYXpwIjoiZXhjaGFuZ2UtYXBwIiwic2lkIjoiMTk3Y2V
hYWQtNDE5NC0yZDhlLWMyODktNjJiMzljZWJjMWQxIiwiYWNyIjoiMSIsImFsbG93ZWQtb3JpZ2lucyI6WyJodHRwOi8vbG9jYWxob3N0OjMxMDAiXSwic
mVhbG1fYWNjZXNzIjp7InJvbGVzIjpbImRlZmF1bHQtcm9sZXMtb2JvLWxhYiIsIm9mZmxpbmVfYWNjZXNzIiwidW1hX2F1dGhvcml6YXRpb24iXX0sInJ
lc291cmNlX2FjY2VzcyI6e30sInNjb3BlIjoicHJvZmlsZSBlbWFpbCIsImVtYWlsX3ZlcmlmaWVkIjp0cnVlLCJuYW1lIjoiYWxpY2Ugd29uZGVyIiwic
HJlZmVycmVkX3VzZXJuYW1lIjoiYWxpY2UiLCJnaXZlbl9uYW1lIjoiYWxpY2UiLCJmYW1pbHlfbmFtZSI6IndvbmRlciIsImVtYWlsIjoiYWxpY2VAZmF
icmlxLmNvbSJ9.2WNE0HzTndKymhhe23CXWdgFyB1nJepDfOmTOAorGSjvI76qUoOV9BMAiiWvqBRmYQYdhr96kjLWYpXEjn80g61NkfNYc-
KiE99Cqbua3NOrMZXDBd_I7pVTVXa17OF5dx2-RrQI5DIt1Q814T6eUsYfg-5JGYfW7x2G6ZFJFsvlJ0qRGiF7xirHeuWwz4QjBOW4vfuN-
fI6klbP2RHmJZ1QPGiy5rbLbAw1TjUSxVxOv-oeyLJ0L319vQZHpuTLu0tg6BIR_2GrVzB-
WltGvJuJ38HyKIsy8paNxRaYlkW2RZKxfUUWClgpsuJS6eOQf-KLIU3QObgOedQ-cfaKnw",
      "tokenDecodedHeader": {
        "alg": "RS256",
        "typ": "JWT",
        "kid": "3VBd_mnOgZieRbToYLYH-0DVStNMlG0ed9qldYu8DdI"
      },
```

Thank You

Get Hands-on Experience with our labs

https://github.com/agenticfabriq/oauth-lab