MCP Dev Summit Bengaluru

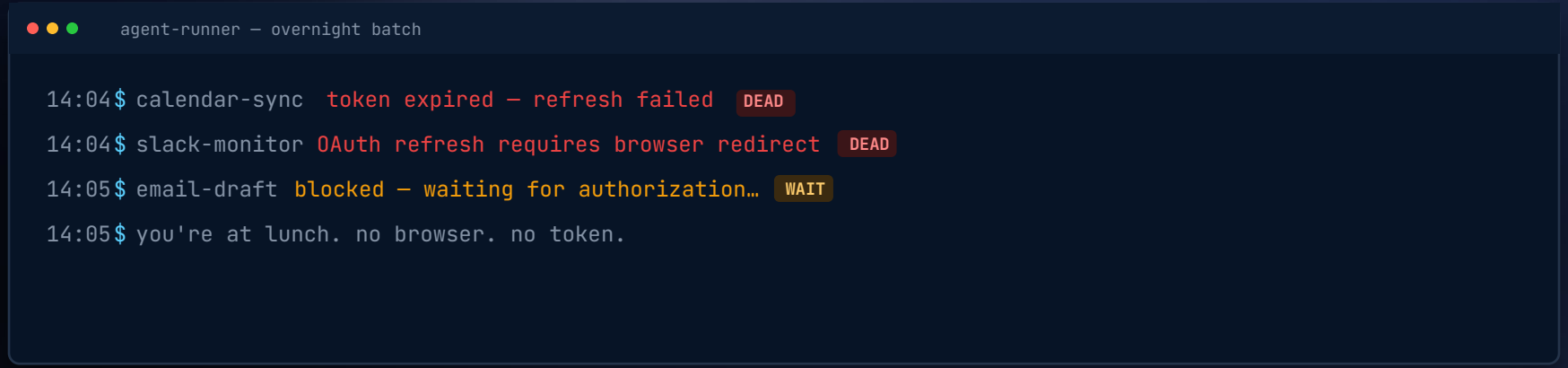
Out-of-Band, Just-in-time Auth for MCP

Decoupled Identity & CIBA for Ambient Agents

Ayesha Dissanayaka
Associate Director / Architect

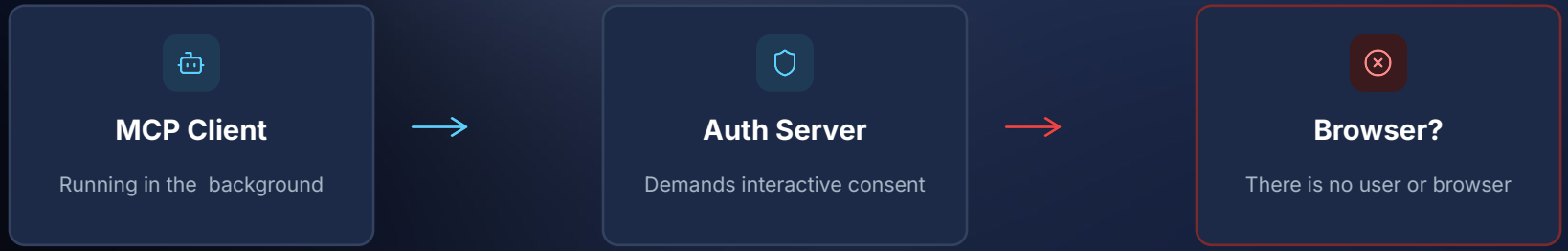
WSO2

It's 2 PM, you're at lunch. Your agent just stopped working.



Your agent monitors the calendar, checks Slack, and drafts emails.
It works beautifully — until nobody's there to click 'Continue'.

Standard Flows Assume a Human at a Browser — Right Now



- No interactive UI at authorization time.
- The user exists and is reachable — just not at the client device
- Authorization must bind to a user

OAuth 2.1 in the Current Spec

Works Well For

- Desktop / IDE plugins
- Web-based MCP clients
- Interactive CLI tools

Breaks Down For

- Headless background agents
- IoT & edge devices
- Sub-agent in a multi-agent system

The Tension

- Short tokens → more interruptions
- Long tokens → weaker security
- No middle ground today

Long-Lived API Keys: The "All-or-Nothing" Trust Model

What Teams Do

1. Generate a personal access token
2. Grant the broadest scopes "to be safe"
3. Set expiry to 1 year (or never)
4. Paste it into an env variable

The Risk

- Massively over-privileged
- Long-lived — a standing liability
- No per-action audit trail
- Violates principle of least privilege

What if the agent could tap you on the shoulder?

On your phone — asking for exactly the permission it needs, right when it needs it.

Client-Initiated Backchannel Authentication

{ CIBA }

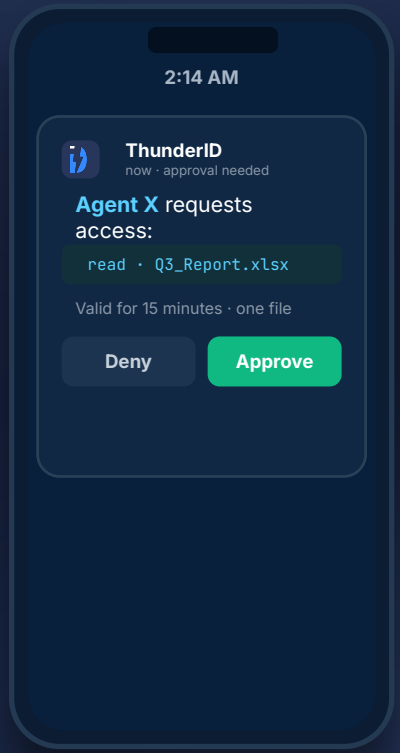
An OpenID Connect extension that lets a backend initiate an auth request, delivered to the user on a separate trusted device.

THE KEY INSIGHT

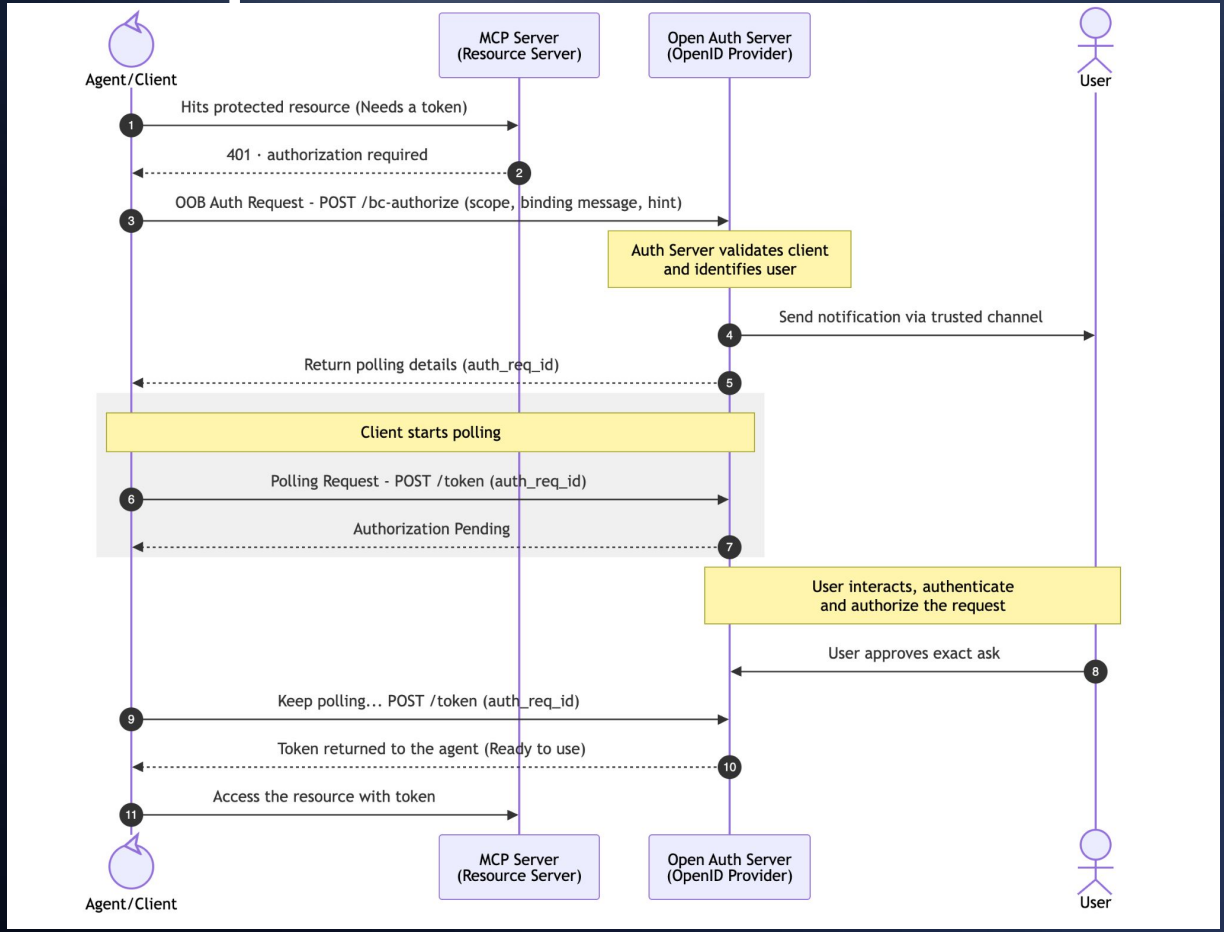
The device **consuming** the resource ≠ the device **approving** access.

How It Works — Overall Flow

1. **Client hits a protected resource**
   It needs a token it doesn't yet hold.
2. **Client calls the OP backchannel**
   `POST /bc-authorize` with scope & a binding message.
3. **Auth server validates and notifies the user**
   A notification through a trusted channel. A URL for Client to poll.
4. **User approves the exact ask**
   They authenticate, and authorize exact grant.
5. **Token is returned to the agent**
   Narrowly scoped, short-lived, ready to use.



Full CIBA Sequence in MCP



Tightening CIBA for MCP

> CIBA has many options, we need to be specific to make it real for MCP

Proposing as an Optional extension.

- ✓ **Optional & opt-in**
- ✓ **Confidential clients only**
- ✓ **Graceful fallback to auth-code**
- ✓ **Restrictive use of options**

Trust Bootstrap

One-time setup — not per-session

Client Discover CIBA

Discovery rides the existing metadata documents.

Register MCP Client

Client obtains credentials from the auth server (client_id + client_secret).

Link Auth Device/ Notification Mechanism

User registers their email, mobile, phone / trusted device and links it to their identity.

Just-in-Time, Out-of-Band Auth Request

Client initiated Backchannel Auth Request

```
POST /bc-authorize

  scope           = drive.file.read:Q3Report.xlsx
  id_token_hint   = eyJhbGciOiJIUzI1NiIs...
  binding_message = "MCP agent needs Q3 report"
  requested_expiry= 900  // 15 min
```

- Resource-specific scope, not broad access
- Human-readable, **mandatory** binding message
- User identity via **id_token_hint**
- Time-bound: expires in 15 minutes

Auth Server validation and response

Successful Authentication Request Acknowledgement

```
HTTP/1.1 200 OK


{
  "auth_req_id": "1c266114-a1be-4252-8ad1-04986c5b9ac1"
  "expires_in": 120,
  "interval": 2
}
```

Authenticate the client first

client_id + secret at minimum.

`auth_req_id`

The handle the client polls with.

`expires_in`

How long the request stays valid — seconds.

interval

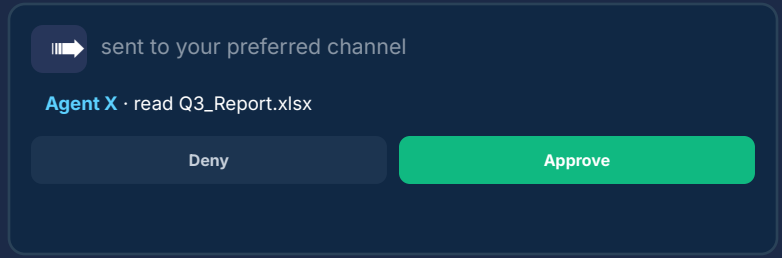
Minimum seconds between polls — respect it.

Poll while the User Consents

Client — polling

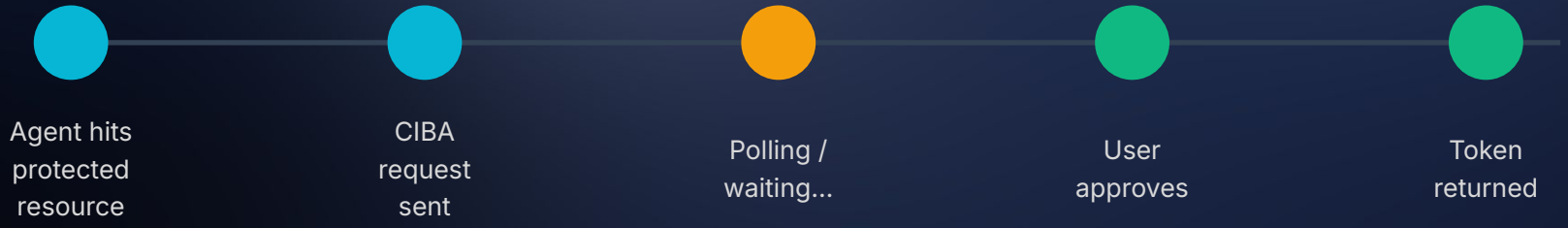
```
$ POST /token
   grant_type = urn:openid:params:grant-type:ciba
   auth_req_id = 1c266114-a1be-4252-8ad1-04986c5b9ac1

→ authorization_pending

$ POST /token  (auth_req_id)

→ authorization_pending

$ POST /token  (auth_req_id)

→ access_token ✓
```

User — consent



Two clocks, one flow. The agent **polls** on its schedule; the user approves on theirs.
Token delivery over **push** and **ping** are out.

Handling Async in MCP's Lifecycle



Key Patterns

Timeouts	60s default; configurable per resource sensitivity
Retry with backoff	Exponential backoff on polling; max 3 attempts
Graceful degradation	Queue the task, notify user, resume when authorized

Edge Cases, Threats & Mitigations

Spam & Consent Fatigue	Batch related requests
Account enumeration	Opt-in for id_token_hint
Binding Message Tampering	Signed binding messages;
Device Compromise	Revocation flows & MFA
The "Sleepy User"	Queue requests; degrade gracefully

Where to apply? Examples?

`login_hint`

CI/CD · Deploying environments

A finished CI build spawns a deployment agent on staging's MCP server. Each preview environment costs money, so the server needs the release manager's sign-off first. The agent CIBAs the manager — pulled from the project's release roster. One tap, and it provisions the env and notifies the QA channel.

BINDING MESSAGE

"Provision staging preview for PR #347 (est. $12/hr). Approve?"

`id_token_hint`

Hospitality · Suite upgrade

A guest wants a suite at check-in — none free — so the booking system stores their session id_token. At 2 PM a checkout frees one and an upgrade agent hits the hotel's MCP server. Charging the $75 difference needs the guest's consent. They confirm, and the agent update the booking and issues the new room key.

BINDING MESSAGE

"Suite 1204 now available — upgrade for $75/night. Confirm?"

Trusted, confidential clients with a known roster may use `login_hint`; the open MCP profile requires `id_token_hint`.

Let's see it in action

A headless background attempt for a flight upgrade — and your phone lights up.

One tap, scoped token, job done.

```
$ ./agent run --background   ·   awaiting approval…
```

Powered by ThunderID

thunderid.dev

Where this goes from here

CIBA in the MCP spec

Should backchannel authorization be a first-class part of the protocol?

Multi-hop delegation chains

Composing CIBA + JWT Auth Grant + Token Exchange for agent-to-agent trust.

Adaptive consent UX

Smart batching and trust scoring so humans approve less, safely.

Three things to carry home

01

Decoupled Auth

CIBA bridges headless MCP clients and the humans who control them.

02

JIT Delegations

Trade all-or-nothing API keys for granular, session-based permissions.

03

Human-in-the-Loop

Handle async out-of-band callbacks gracefully inside the MCP lifecycle.

Your agent doesn’t have to stall when you are not looking.

It should knock on your door and ask.

Thank You!

MCP Dev Summit Bengaluru

WSO2