

TECHNICAL DATASHEET
FOR ARCHITECTS & IT TEAMS

AURAPLAYER MCP SERVER: DEPLOYMENT ARCHITECTURE, PREREQUISITES & OPTIONS

CONNECTING AI AGENTS TO ORACLE EBS, YOUR WAY

A technical reference for architects and IT teams evaluating how to connect AI agents to Oracle EBS through the AuraPlayer MCP server — covering both the local (stdio) and enterprise (Streamable HTTP) deployment models, authentication, and rollout.

**Governed, process-layer MCP access to Oracle EBS
for the AI tools your enterprise already uses**

The Oracle MCP Landscape: Where AuraPlayer Fits

There's more than one way to skin a cat, and the same is true for connecting AI to Oracle. Most approaches connect an AI agent straight into the database, which is fast to stand up but puts table-level read/write access one prompt away, with no guarantee that Oracle's own validations, approvals, and workflows are respected along the way. AuraPlayer takes a different route: it runs predefined, secured, and governed services that automate Oracle E-Business Suite processes at the application layer, so every trigger, validation, and approval your team already built keeps doing its job.

Worth noting: AI clients such as Claude, Microsoft Copilot, ChatGPT, and Gemini aren't MCP servers at all. They're the interface that discovers and calls whichever MCP server below fits the task, so any of them can be pointed at any of the four options.

Comparing the Four Ways to Connect AI to Oracle EBS

	Oracle SQLcl MCP	Oracle Database Tools MCP	Oracle ORDS Streaming MCP	AuraPlayer MCP
Layer & what it exposes	Database layer. Local, stdio-based CLI server; credentials and connections stay on the developer's machine	Database layer. Oracle-managed, cloud-hosted MCP server for OCI databases	Database layer. REST/HTTP MCP server exposing the Oracle Database over standard HTTPS	Business process layer. Predefined, governed Oracle EBS workflows, not tables
Key benefit	Nothing to deploy beyond SQLcl; ideal for local SQL generation, schema exploration, AWR analysis	Centrally managed, enterprise-scale MCP access to OCI databases; no per-machine install	Modern, standards-based DB connectivity for SaaS AI clients, no custom integration	Existing Forms validations, approvals, and audit trails run exactly as they do today. Nothing bypassed
Key risk	Runs under the developer's own DB privileges; blast radius depends entirely on that account's grants	Still direct database access; governance depends on OCI IAM policy, not application-layer rules	Still a direct database connection; validations above the DB aren't enforced	The target process must be recorded and published as a service first
Best for	Individual developers and DBAs working locally against a dev or test database	Cloud teams centralizing MCP access and governance across OCI databases	Giving AI clients broad, secure query access to database content	Letting AI safely execute EBS transactions: approvals, receipts, orders, expenses
Example ask	"Explain this AWR report"	"Compare schemas across these two OCI databases"	"List all overdue invoices over \$100k"	"Approve every invoice that matches our policy"

The take-away for architects

- **These aren't competing options, they're different responsibilities.** The three Oracle database MCP servers answer questions. AuraPlayer executes business processes.
- **The strongest architectures combine both:** query insight from the database layer, then governed action through AuraPlayer, orchestrated by whichever AI client your team already uses.
- **Ask the right question first.** Not "which MCP server is better," but "what am I asking AI to do?" The answer decides the layer.

Introducing AuraPlayer: Two Deployment Models. One MCP Server.

The AuraPlayer MCP server publishes your recorded Oracle EBS workflows as tools that AI agents can discover and call. It runs in two deployment modes. Both drive the same AuraPlayer Server (ServiceManager) and the same Oracle EBS instance. The difference is where the MCP server lives, how clients reach it, and how identity is handled.

OPTION 1

LOCAL DEPLOYMENT

stdio transport. A lightweight MCP bridge installed on each workstation. Runs under each user's own credentials. Ideal for pilots and desktop clients.

OPTION 2

ENTERPRISE DEPLOYMENT

Streamable HTTP transport. One shared, stateless service for every user and every AI client. Central auth, audit, and control. Nothing installed on client machines.

Use the comparison below to pick a starting point. The pages that follow detail the architecture, prerequisites, and assumptions for each option.

At-a-glance comparison

Aspect	Local (stdio)	Enterprise (Streamable HTTP)
Where it runs	On every user workstation, launched by the MCP client	One shared service next to the AuraPlayer Server (VM, container, or K8s)
Transport	MCP stdio (JSON-RPC over stdin/stdout)	MCP Streamable HTTP: a single <code>/mcp</code> endpoint, responses may stream as SSE
Install footprint	Per-machine install, configure, and update (.mcpb bundle or exe)	Zero client install. Clients are pointed at a URL
User identity	Built in. Each install carries that user's AuraPlayer/EBS credentials	OAuth 2.1 via your corporate IdP, with per-user impersonation into EBS
Network reach	Every workstation needs direct access to the AuraPlayer Server	Only the MCP endpoint is published. The AuraPlayer Server stays internal
Client support	Desktop MCP clients that can launch local processes (e.g. Claude Desktop). Web clients cannot use it	Any MCP client supporting Streamable HTTP: claude.ai, ChatGPT, Copilot Studio, Claude Code, IDEs, custom agents
Central control	None. No single place to audit, rotate credentials, or enforce policy	Upgrades, auditing, service filtering, and credential policy handled in one place
Best for	Individual pilots, demos, small technical teams	Production rollouts, SaaS assistant connectors, governed enterprise access

What stays the same in both models

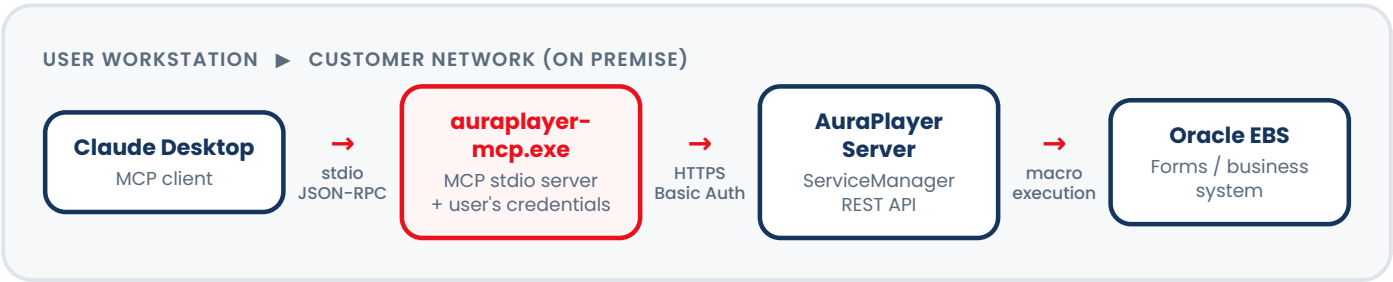
- **The business layer.** Tool calls execute recorded macros through the ServiceManager REST API, so Oracle Forms triggers and validations always run.
- **Tool discovery.** Services labeled "MCP" in ServiceManager are published automatically as MCP tool definitions.
- **Oracle EBS is never exposed.** In both models, EBS is reached only through the AuraPlayer Server inside your network.

Option 1: Local Deployment (stdio)

The MCP server is a small Python bridge, packaged as a standalone Windows executable and installed on every client machine. The MCP client, for example Claude Desktop, launches it as a child process and communicates over stdin/stdout using the MCP stdio transport.

Each installation is configured with that user's AuraPlayer/EBS credentials through environment variables. These are entered in the Claude Desktop extension UI and stored in Windows Credential Manager.

Architecture



Components

Component	Location	Role
MCP client (Claude Desktop, etc.)	User workstation	Hosts the conversation, decides which tools to call
auraplayer-mcp stdio server	User workstation	Translates MCP <code>tools/list</code> / <code>tools/call</code> into AuraPlayer REST calls
AuraPlayer Server (ServiceManager)	Customer network	Publishes recorded macros as services and executes them
Oracle EBS	Customer network	The business system the macros drive

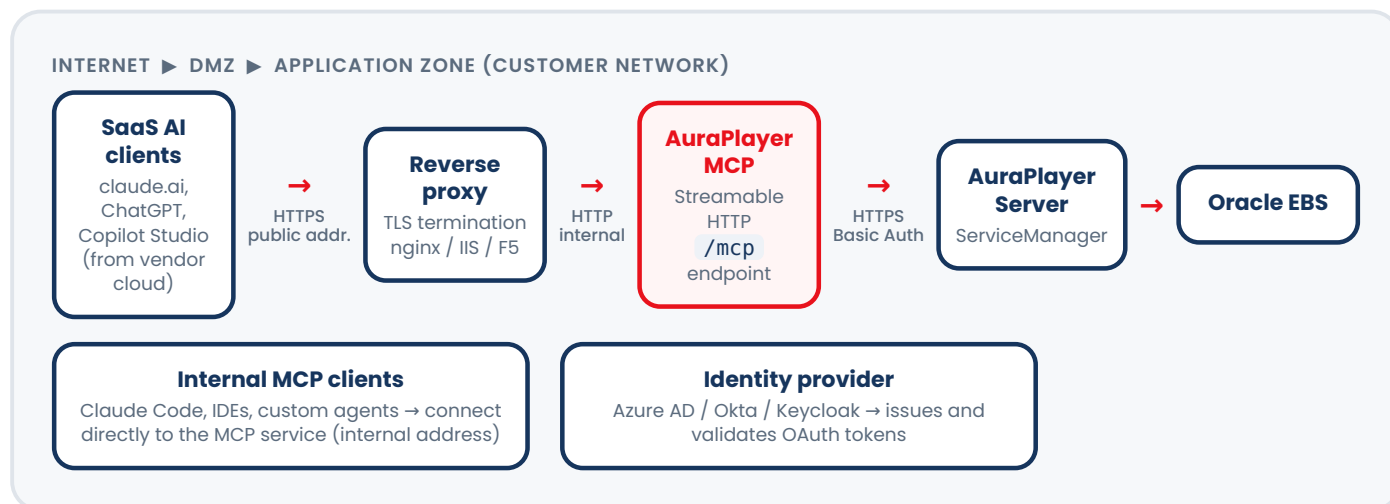
Properties and limitations

- **Per-machine install.** The .mcpb bundle or exe must be installed, configured, and updated on every workstation.
- **Per-user identity comes for free.** Each install carries that user's credentials, so macros run under the correct EBS identity.
- **Network reach.** Every workstation needs direct network access to the AuraPlayer Server.
- **Client support.** Limited to MCP clients that can launch local stdio servers (desktop applications). Web clients such as claude.ai cannot use it.
- **No central control.** No single place to audit usage, rotate credentials, enforce policy, or roll out new versions.

Option 2: Enterprise Deployment (Streamable HTTP)

In the enterprise deployment, the MCP server runs as a single shared service next to the AuraPlayer Server, exposing the MCP Streamable HTTP transport. There is one `/mcp` endpoint. Requests are HTTP POSTs, and responses may stream as Server-Sent Events. Nothing is installed on client machines. Clients are simply pointed at a URL.

Architecture



Key points

- **Flexible placement.** The MCP service is typically co-located with the AuraPlayer Server, but it is a stateless HTTP service whose only requirement is HTTP(S) access to the ServiceManager REST API. It can run on the same machine, a VM, or a container. Co-location keeps latency low and avoids another firewall path.
- **The AuraPlayer Server is never exposed.** Only the MCP `/mcp` endpoint is published, and only as far as needed.
- **Ports.** The MCP service listens on a configurable port (default 8000, plain HTTP, distinct from ServiceManager's default 8080). Clients never see that port. They connect to the reverse proxy on standard 443, which forwards to the MCP port. The only network paths needed are proxy → MCP port, and MCP host → ServiceManager port.
- **One deployment serves everyone.** All users and all client types through one endpoint, with upgrades, auditing, service filtering, and credential policy handled in one place.

Service endpoints

Endpoint	Purpose
POST <code>/mcp</code>	The MCP Streamable HTTP endpoint used by all clients
GET <code>/health</code>	Unauthenticated health check for monitoring and load balancers
GET <code>/.well-known/oauth-protected-resource</code>	OAuth resource metadata for connector discovery (OAuth mode only)

The service is stateless, so it can be restarted or load-balanced freely. Listen address and port, authentication mode, API keys, identity provider settings, and the impersonation user field are all set through configuration.

About AuraPlayer

AuraPlayer's patented technology automatically generates REST APIs, mobile applications, RPA automations, and MCP tools from existing back-office applications, helping you begin your AI transformation while leveraging existing investments.

AuraPlayer's MCP server is built specifically for Oracle Forms and E-Business Suite. It exposes governed business workflows at the process layer, so every AI-initiated action runs through your existing Oracle validations, triggers, and security. No replatforming. No direct database access. Just safe, auditable AI automation on the system you already run.

Let us join your Oracle AI journey.

LinkedIn: @auraplayer

X: @auraplayer

YouTube: @AuraplayerAdmin

Email: info@auraplayer.com

Web: www.auraplayer.com

Schedule a call