

ORACLE

MySQL Contributor Summit

MCP Access for MySQL Server

Miroslav Rajcic

MySQL

Oracle

May 21, 2026.

Safe harbor statement

The following is intended to outline our general product direction. It is intended for Information purposes only and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Core Proposal

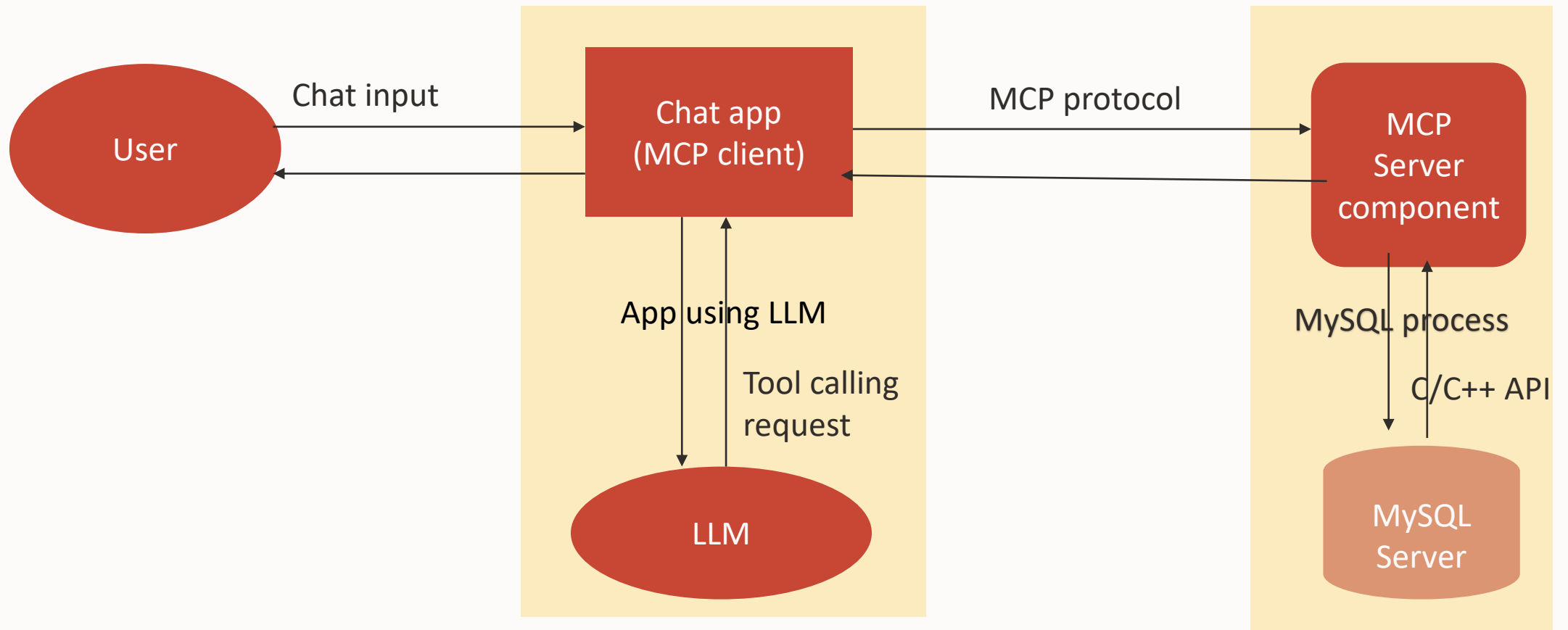
- Provide MCP (Model Context Protocol) server for MySQL
- Enables AI tools to safely discover schemas, inspect metadata, run safe read-only queries, and retrieve selected server diagnostics
- MySQL-native, likely implemented in-process as a MySQL component
- Later evolve to allow write access as well



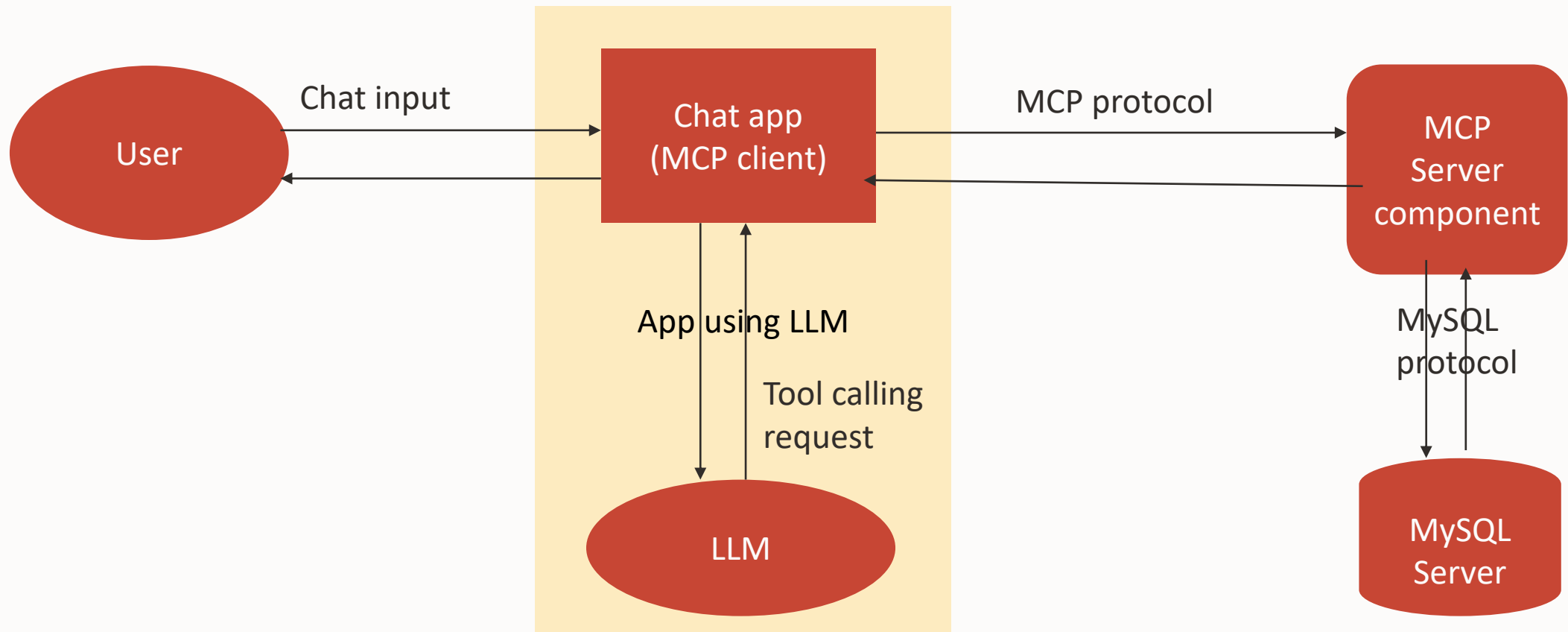
Advantages over existing implementations

- part of the DB server, less failure points in distributed system, no traffic tunnelling needed
- reduced latencies when executing multiple SQL statements per MCP function in-process
- reuse MySQL login infrastructure (users, grants, roles)
- configured within server (system variables) instead of using env. variables
- safer defaults, better privilege enforcement
- auditability, shares log with MySQL server, possible integration with OpenTelemetry (MCP metrics)
- access to internal server diagnostics (expose CPU/MEM/disk usage, uptime, ...)
- CI testing (less likely to break due to some MySQL change)

MCP within MySQL (in-process architecture)



MCP as separate process (out-of-process architecture)



Use cases

- Schema exploration

AI assistants help to understand database structure by exposing schemas, tables, columns, indexes, constraints, and relationships through a standard interface.

- Query assistance

Developers and analysts using AI tools to generate, explain, and execute safe read-only queries against MySQL.

- Operational diagnostics

A MySQL-native MCP interface could expose selected server information useful for DBAs, support engineers, and troubleshooting workflows.

- Developer onboarding and support

MCP access could make MySQL easier to use from IDEs, notebooks, internal AI assistants, and support tools by allowing natural-language interaction with database metadata and selected runtime information.



Architecture choices

- one MCP server per single MySQL server or one per the entire fleet of servers:
 - Handling entire fleet is more powerful but also more vulnerable and more complex to implement
 - Lower latency in MCP per MySQL server
- MCP server as component?, as part of existing MySQL REST Service (MRS)?, both?
- Authentication/authorization via mTLS (server/user certificates), OAUTH or JWT?
- Enable future DB modification queries (run arbitrary SQL queries) ?
 - Is existing GRANTS/ROLES system enough to express all the permission needs



Conclusion

- MCP is a standard, market is exploding with many 3rd party solutions for any imaginable system
- Integration will unlock many new possible use cases (including the ones we may not foresee yet)
- Iterating and evolving over time (first release with read-only MCP methods only)
- Consider implementing this quickly (speed is a key thing for capturing the fast moving AI market)



ORACLE

