Supreme Mind MCP Server

Specification of the Model Context Protocol server for Supreme Mind Expert Agents
Supreme Mind AI, Inc. · Last updated: 5 November 2026 · Specification version 1.1.0 · Server version 1.0.0

Introduction

Supreme Mind builds expert witness AI agents for US litigation, starting with securities and antitrust cases. The Supreme Mind MCP server lets an AI assistant, such as Harvey, Claude or ChatGPT, call the same Expert Agents that firms use in the Supreme Mind app: create a matter, request an expert brief (the opposing expert's likely opinion, cross-examination weaknesses, Rule 702 attack surface and an estimated settlement range, each point cited to the source and run on the expert's own data sources), and run a cross-examination practice session.

This document is intended for developers and security reviewers. It describes exactly:

- **Endpoint and MCP version**: where the server lives and which parts of the MCP specification it supports.
- **Authentication**: how the server authenticates clients and users.
- **Scopes and token security**: how access is limited and how long credentials live.
- **Tool calls**: a complete catalog of the 9 tools the server offers, and what is coming next.
- **Tool descriptions, call arguments and return results**: a reference page for each tool, with the name, datatype and restrictions of every argument.
- **Data handling**: tenant isolation, logging and data use.

Most of this is also described dynamically in the protocol itself (tools/list returns each tool's JSON Schema for its input and output), but it is collected here as a durable reference. Any tool not listed in this document does not exist on the server.

Endpoint

URL	`https://mcp.suprememind.ai/mcp`
Transport	Streamable HTTP over HTTPS (TLS), JSON-RPC 2.0. POST only: each request returns `application/json`; a notification returns 202 with no body. GET and DELETE return 405 (no event stream, no session).
Server name	`supreme-mind` (title: Supreme Mind)
Server version	1.0.0
Protocol revisions	`2025-11-25` (preferred), `2025-06-18`, `2025-03-26`. The server answers initialize with the client's requested revision when supported, otherwise the latest.

Rate limit	60 requests per minute per connector grant; excess requests receive 429 with `Retry-After`.

Authentication

Supreme Mind handles confidential litigation material, so every tool call is made on behalf of a known person at a known firm. All requests must carry `Authorization: Bearer <token>`. Credentials in the query string are refused. A request without valid credentials receives 401 with a `WWW-Authenticate` header that points to the protected resource metadata (RFC 9728), which is how an MCP client discovers how to obtain a token.

For a connector such as Harvey, the server uses **OAuth 2.1** run by Supreme Mind's own authorization server:

- **Grant:** authorization code with **mandatory PKCE, S256 only** (RFC 7636). No implicit, password, client credentials or plain PKCE.
- **Client identification:** Dynamic Client Registration (RFC 7591) at the registration endpoint, or a Client ID Metadata Document (an https client_id URL). Redirect URIs are matched exactly.
- **Consent:** the user signs in to Supreme Mind and approves the connector and its scopes on a consent screen that shows the client's stated name and redirect URI.
- **Audience binding (RFC 8707):** every token is bound to one firm, one user, one client, its granted scopes and one audience, the MCP server. A token issued for the MCP server is refused by every other Supreme Mind endpoint.
- **Ongoing checks on every call:** the token must be unexpired and unrevoked, the firm active with API access enabled, and the user still active.

Issuer	`https://app.suprememind.ai`
Authorization server metadata	`https://app.suprememind.ai/.well-known/oauth-authorization-server`
Protected resource metadata	`https://mcp.suprememind.ai/.well-known/oauth-protected-resource`
Authorization endpoint	`https://app.suprememind.ai/oauth/authorize`
Token endpoint	`https://app.suprememind.ai/oauth/token` (client auth: none, client_secret_basic or client_secret_post)
Registration endpoint	`https://app.suprememind.ai/oauth/register`
Revocation endpoint	`https://app.suprememind.ai/oauth/revoke` (RFC 7009)

For server-to-server integrations a firm may instead use a Supreme Mind API key (prefix `sm_live_`, or `sm_test_` for test mode, which returns fixed, labelled samples and is never charged). API keys are issued by the firm's administrator in the Supreme Mind app.

SCOPES

Access is limited by the OAuth scopes the client requests and the user approves. An authenticated client can always list tools; each tool call then checks its own scope, so a client with only `read` can see but not call write tools. A request that names no known scope is refused; unknown scopes are dropped.

Scope	Purpose	Tools
`read`	Read expert classes, the firm's briefs and jobs, and its usage.	list_expert_classes, get_expert_class, get_brief, get_job, get_usage
`briefs`	Create matters and request briefs (charged to the firm on delivery).	create_matter, create_brief
`cross_exam`	Run cross-examination practice sessions (charged to the firm at the start of each session).	start_cross_exam, cross_exam_turn

TOKEN SECURITY

All tokens are opaque random strings, stored by Supreme Mind only as a SHA-256 hash. Refresh tokens rotate: each is single use, and presenting a used refresh token is treated as theft and revokes the whole token family. A firm administrator can see and revoke every connector grant in the Supreme Mind app; revocation, or switching off the firm's API access, takes effect within 60 seconds.

Component	Time to live	Notes
Authorization code	5 minutes	Single use; bound to the PKCE challenge and redirect URI.
Access token (`smo_at_`)	1 hour	Bound to firm, user, client, scopes and the MCP audience.
Refresh token (`smo_rt_`)	30 days	Single use, rotated on every refresh.

MCP features

The server publishes **tools** only (`capabilities.tools` , with `listChanged: false`). It does not publish resources or prompts and does not send server-initiated requests (no sampling or elicitation). It returns `instructions` at initialize describing the intended flow: list expert classes, create a matter, request a brief (passing groundings when the facts name a ticker, event dates or periods), poll for it, and show each citation's verification status. Every tool declares an input schema, an output schema and annotations (readOnlyHint, destructiveHint, idempotentHint, openWorldHint), and no tool is destructive: nothing can be deleted through MCP.

Data handling and security

- **Tenant isolation:** the firm is taken from the credential and only from it. Every tool reads and writes through the same firm-scoped operations as the Supreme Mind REST API. An id belonging to another firm returns the same error as one that does not exist.
- **Audit log:** every tool call is logged with the tool name, the connector, the user, the resource id and the result status. Arguments and matter content are not written to the audit log.
- **Public data sources:** a brief's data sources read public data only: market prices, SEC EDGAR, FRED series and the CourtListener judicial record. Only the identifiers they need are sent to those providers (ticker, dates, series ids and the expert's method); case facts and matter content are not.
- **Data use:** client data is never used to train models and is not retained by the AI model provider. Model inference runs in the United States.

- **Protected material:** matters are walled per firm and access is logged; material under a protective order, including attorneys'-eyes-only, is supported under a signed agreement.
- **Output limits:** briefs are attorney work product and consulting analysis, not evidence and not a testifying expert's opinion. Settlement ranges are estimates for decision support, not a prediction of any outcome. Each citation carries its verification status, and figures and calculations are labelled as not checked the way quotations are.

Tool catalog

All 9 tools available to an authenticated client. Each is described in full in the reference pages that follow.

Tool	Summary	Scope	Effect
`list_expert_classes`	List the live expert classes a brief can be written for.	read	Read-only, free
`get_expert_class`	Get one expert class with its methods, typical questions and common attacks.	read	Read-only, free
`create_matter`	Create a matter (a case) for the user's firm, required before a brief.	briefs	Writes, free (briefs on it are charged)
`create_brief`	Request an expert brief on a matter for one expert class.	briefs	Writes, charged on delivery
`get_brief`	Get a brief's status and, once it has succeeded, its content.	read	Read-only, free
`get_usage`	Get the firm's credit balance and the last 30 days of use.	read	Read-only, free
`get_job`	Get the status and result of a background job.	read	Read-only, free
`start_cross_exam`	Start a cross-examination practice session with a witness modelled on an expert class.	cross_exam	Writes, charged at start
`cross_exam_turn`	Put one question to the witness in a session and get the answer.	cross_exam	Writes, free inside a paid session

Pricing: a brief costs 1 credit ($50), charged only on delivery; a failed brief is not charged; the high profile can cost 2 credits. A cross-examination session costs 0.2 credits, charged at start. All other tools are free. Every paid tool states its price in its description, so the model sees it before it acts.

Reference

Each tool has its own page with: name and title, short summary, full description, required scope, call arguments with datatypes and restrictions, and return results.

list_expert_classes

List the live expert classes a brief can be written for.

Title: List expert classes Scope: `read` Read-only, free

FULL DESCRIPTION

Returns every expert class currently live: the kind of expert, the practice area, which side usually retains it, and its method. The id of a class is the value to pass as expert_class to create_brief and start_cross_exam.

The list is the same for every caller and contains no client data.

Use limit shown to the client: Attorney work product and consulting analysis only. Not evidence and not a testifying expert's opinion.

SCOPES

Requires the `read` scope. A call without it is refused with an error naming the missing scope.

CALL ARGUMENTS

This tool takes no arguments. Supplying any argument returns an error.

RETURN RESULTS

The result carries the object below as `structuredContent`, plus two text blocks: a short summary for the model and the same object serialized as JSON. Errors are returned as tool results with `isError: true` and a message the model can act on.

Field	Type	Description
`object`	string	Always "list".
`data`	array of ExpertClass	The live expert classes (see ExpertClass below).
`has_more`	boolean	Always false: the full list is returned in one call.

EXPERTCLASS OBJECT

Field	Type	Description
`id`	string	The expert class id.
`object`	string	Always "expert_class".
`name`	string	Display name, for example "Defense event-study econometrician".
`practice`	string or null	Practice area, for example securities or antitrust.
`side`	string or null	The side that usually retains this expert.
`summary`	string or null	One-paragraph description.
`uses`	array of string	Typical uses of this expert in a case.

get_expert_class

Get one expert class with its methods, typical questions and common attacks.

Title: Get an expert class Scope: `read` Read-only, free

FULL DESCRIPTION

Returns the full description of one live expert class: the methods the expert typically uses, the questions such an expert is typically asked, and the attacks commonly made on that kind of opinion.

An id that does not name a live expert class returns an error that tells the client to call list_expert_classes.

Use limit shown to the client: Attorney work product and consulting analysis only. Not evidence and not a testifying expert's opinion.

SCOPES

Requires the `read` scope. A call without it is refused with an error naming the missing scope.

CALL ARGUMENTS

Argument	Type	Required	Description and restrictions
`id`	string	Yes	The expert class id from list_expert_classes. Maximum 120 characters.

Unknown arguments are refused. All strings are trimmed for emptiness checks.

RETURN RESULTS

The result carries the object below as `structuredContent`, plus two text blocks: a short summary for the model and the same object serialized as JSON. Errors are returned as tool results with `isError: true` and a message the model can act on.

Field	Type	Description
`(ExpertClass fields)`	object	All fields of ExpertClass (see list_expert_classes).
`methods`	array of object	The methods the expert typically uses, each with a name and description.
`typical_questions`	array	Questions this kind of expert is typically asked.
`common_attacks`	array	Attacks commonly made on this kind of opinion.

create_matter

Create a matter (a case) for the user's firm, required before a brief.

Title: Create a matter Scope: `briefs` Writes, free (briefs on it are charged)

FULL DESCRIPTION

Creates one matter in the firm of the authenticated user. A matter holds the facts of a case; briefs, report analyses and cross-examination sessions are always attached to a matter.

The client sends the case name and exactly one of text (a plain-text summary of the facts) or case_context (the facts as a structured object). Sending both, or neither, returns an error.

The matter is visible only inside the user's firm, under the firm's existing access rules. Creating a matter is free. This tool cannot read, change or delete existing matters; deleting a matter is deliberately not available through MCP.

Use limit shown to the client: Attorney work product and consulting analysis only. Not evidence and not a testifying expert's opinion.

SCOPES

Requires the `briefs` scope. A call without it is refused with an error naming the missing scope.

CALL ARGUMENTS

Argument	Type	Required	Description and restrictions
`name`	string	Yes	The case name or caption. 1 to 300 characters.
`text`	string	One of text or case_context	A plain-text summary of the case facts. Not empty.
`case_context`	object	One of text or case_context	The facts as a structured object, for example court, parties, class period and allegations.

Unknown arguments are refused. All strings are trimmed for emptiness checks.

RETURN RESULTS

The result carries the object below as `structuredContent`, plus two text blocks: a short summary for the model and the same object serialized as JSON. Errors are returned as tool results with `isError: true` and a message the model can act on.

Field	Type	Description
`id`	string	The new matter id, used as matter_id in later calls.
`object`	string	Always "matter".
`livemode`	boolean	false when created with a test key.
`name`	string or null	The case name.
`status`	string or null	The matter status.

Field	Type	Description
`case_context`	object	The stored facts.
`created_at`	string or null	ISO 8601 timestamp.

create_brief

Request an expert brief on a matter for one expert class.

Title: Request a brief Scope: `briefs` Writes, charged on delivery

FULL DESCRIPTION

Queues an expert brief on one matter for one expert class. The brief is written in the background and takes several minutes. The call returns at once with the brief id and status queued; the client then calls get_brief with that id about once a minute until the status is succeeded or failed.

Groundings: the brief runs on the expert class's own data sources, computed on the server for this brief and cut at groundings.as_of: a market-model event study on the issuer's stock and SEC EDGAR (filings and public float, valued at the class period) for securities experts, a cost pass-through study for antitrust experts, and, for every expert class, the judicial record on the method (prior Rule 702 and Daubert rulings, from CourtListener). The client passes groundings when the case facts name a ticker, event dates or a class or conspiracy period. When groundings is omitted, the server runs the sources the matter itself supports, as of today. A source that cannot run (for example, no ticker is known) is skipped and the brief still runs; get_brief reports which sources ran and why any were skipped. A renamed ticker is mapped to the issuer's current symbol (for example FB to META) and the mapping is reported. Data sources add up to about two minutes and cost nothing extra.

A succeeded brief contains the expert's likely opinion, cross-examination weaknesses, methodology challenges (the Rule 702 attack surface) and an estimated settlement range, with every citation and its verification status.

Price: 1 credit ($50), charged only when the brief is delivered; a brief that fails is not charged. The high profile can cost 2 credits. Test keys are never charged.

A request the firm cannot pay for, or that exceeds a matter plan's weekly fair-use limit, is refused with an error that names the reason. The matter must belong to the user's firm; any other matter id returns the same error as one that does not exist.

Use limit shown to the client: Attorney work product and consulting analysis only. Not evidence and not a testifying expert's opinion.

SCOPES

Requires the `briefs` scope. A call without it is refused with an error naming the missing scope.

CALL ARGUMENTS

Argument	Type	Required	Description and restrictions
`matter_id`	string	Yes	The matter id from create_matter. Maximum 60 characters.
`expert_class`	string	Yes	The expert class id from list_expert_classes. Maximum 120 characters.
`side`	string (enum)	No	plaintiff, defense or either. Refused when it contradicts the expert class.

Argument	Type	Required	Description and restrictions
`profile`	string (enum)	No	standard (default, or the firm's own setting) or high. high uses a stronger model and can cost 2 credits.
`groundings`	object (Groundings)	No	The expert's data sources for this brief. See the Groundings object below. Unknown fields are refused.

Unknown arguments are refused. All strings are trimmed for emptiness checks.

RETURN RESULTS

The result carries the object below as `structuredContent`, plus two text blocks: a short summary for the model and the same object serialized as JSON. Errors are returned as tool results with `isError: true` and a message the model can act on.

Field	Type	Description
`id`	string	The brief id.
`object`	string	Always "brief".
`livemode`	boolean	false with a test key.
`status`	string (enum)	queued, running, succeeded or failed.
`matter_id`	string or null	The matter.
`expert_class`	string or null	The expert class.
`created_at`	string or null	ISO 8601 timestamp.
`completed_at`	string or null	ISO 8601 timestamp.
`charged_credits`	number	Credits charged for this brief.
`result`	object	When succeeded: sections (likely_opinion, cross_ex, methodology_challenges, settlement_range, and where available decision and ruling_prediction) and citations (each with section, citation text, url, verification status, provider, checked_at and supporting quotes with quote_verified).
`groundings`	object	When the brief ran data sources: as_of, defaulted (true when the server chose the sources from the matter), sources (each with id, status, and where relevant detail, ticker_note for a mapped ticker, and the reason a source was skipped) and a one-line summary.
`provenance`	object	States what was checked (quotations and citations) and what was not (figures and analysis), and the use limit.
`sample`	object	Present only for test keys: marks the brief as a fixed sample.
`error`	object	When failed: type and message.

GROUNDINGS OBJECT

Field	Type	Description
`as_of`	string (date)	Required when groundings is sent. The last day of information the brief may use, YYYY-MM-DD; normally today. Every source is cut at this date.
`ticker`	string	The issuer's stock ticker, for example FB. 1 to 40 characters: letters, digits and . _ ^ -. Runs the event study (with event_dates) and SEC EDGAR.
`event_dates`	array of string (date)	The disclosure or price-drop dates the case turns on, YYYY-MM-DD. Up to 40.
`class_period`	object {start, end}	The class period the complaint pleads, each date YYYY-MM-DD.
`market_ticker`	string	Optional market index for the event study; ^GSPC (S&P 500) by default. Same character rules as ticker.
`sector_ticker`	string	Optional sector index or ETF to compare the event-date moves against, for example XLK.
`price_series`	string	Antitrust: the FRED or price-index series id for the product's price.
`cost_series`	string	Antitrust: the FRED series id for the main input cost.
`conspiracy_period`	object {start, end}	Antitrust: the alleged conspiracy period; start is required.
`benchmark_period`	object {start, end}	Antitrust: an optional clean benchmark period outside the conspiracy.

get_brief

Get a brief's status and, once it has succeeded, its content.

Title: Get a brief Scope: `read` Read-only, free

FULL DESCRIPTION

Returns the status of a brief and, once it has succeeded, its sections, the data sources that ran (and why any were skipped), every citation with its verification status, and the provenance statement. Clients poll this about once a minute after create_brief.

Every citation carries a verification status (for example verified or unconfirmed). The server instructs clients to show that status beside each citation and never to present an unverified citation as verified.

Only briefs of the user's firm can be read. Any other id returns the same error as an id that does not exist.

Use limit shown to the client: Attorney work product and consulting analysis only. Not evidence and not a testifying expert's opinion.

SCOPES

Requires the `read` scope. A call without it is refused with an error naming the missing scope.

CALL ARGUMENTS

Argument	Type	Required	Description and restrictions
`id`	string	Yes	The brief id returned by create_brief. Maximum 60 characters.

Unknown arguments are refused. All strings are trimmed for emptiness checks.

RETURN RESULTS

The result carries the object below as `structuredContent`, plus two text blocks: a short summary for the model and the same object serialized as JSON. Errors are returned as tool results with `isError: true` and a message the model can act on.

Field	Type	Description
`id`	string	The brief id.
`object`	string	Always "brief".
`livemode`	boolean	false with a test key.
`status`	string (enum)	queued, running, succeeded or failed.
`matter_id`	string or null	The matter.
`expert_class`	string or null	The expert class.
`created_at`	string or null	ISO 8601 timestamp.

Field	Type	Description
`completed_at`	string or null	ISO 8601 timestamp.
`charged_credits`	number	Credits charged for this brief.
`result`	object	When succeeded: sections (likely_opinion, cross_ex, methodology_challenges, settlement_range, and where available decision and ruling_prediction) and citations (each with section, citation text, url, verification status, provider, checked_at and supporting quotes with quote_verified).
`groundings`	object	When the brief ran data sources: as_of, defaulted (true when the server chose the sources from the matter), sources (each with id, status, and where relevant detail, ticker_note for a mapped ticker, and the reason a source was skipped) and a one-line summary.
`provenance`	object	States what was checked (quotations and citations) and what was not (figures and analysis), and the use limit.
`sample`	object	Present only for test keys: marks the brief as a fixed sample.
`error`	object	When failed: type and message.

get_usage

Get the firm's credit balance and the last 30 days of use.

Title: Get usage Scope: `read` Read-only, free

FULL DESCRIPTION

Returns the credit balance of the user's firm and its use over the last 30 days, broken down by API key and by OAuth connector (such as this integration). Contains no matter content.

Use limit shown to the client: Attorney work product and consulting analysis only. Not evidence and not a testifying expert's opinion.

SCOPES

Requires the `read` scope. A call without it is refused with an error naming the missing scope.

CALL ARGUMENTS

This tool takes no arguments.

RETURN RESULTS

The result carries the object below as `structuredContent`, plus two text blocks: a short summary for the model and the same object serialized as JSON. Errors are returned as tool results with `isError: true` and a message the model can act on.

Field	Type	Description
`object`	string	Always "usage".
`balance`	number	Current credit balance.
`period`	object	The period covered (days, start and end).
`credits_charged`	number	Credits charged in the period.
`credits_refunded`	number	Credits refunded in the period.
`by_key`	array of object	Use per API key.
`by_oauth_client`	array of object	Use per OAuth connector.

get_job

Get the status and result of a background job.

Title: Get a job Scope: `read` Read-only, free

FULL DESCRIPTION

Returns the status of a background job (a brief, or a document extraction or report analysis started in the Supreme Mind app) and, once it has succeeded, its result; for a report analysis, the structured analysis. A failed job states the reason and is not charged.

Only jobs of the user's firm can be read. Any other id returns the same error as an id that does not exist.

Use limit shown to the client: Attorney work product and consulting analysis only. Not evidence and not a testifying expert's opinion.

SCOPES

Requires the `read` scope. A call without it is refused with an error naming the missing scope.

CALL ARGUMENTS

Argument	Type	Required	Description and restrictions
`id`	string	Yes	A job id, or a brief id from create_brief. Maximum 60 characters.

Unknown arguments are refused. All strings are trimmed for emptiness checks.

RETURN RESULTS

The result carries the object below as `structuredContent`, plus two text blocks: a short summary for the model and the same object serialized as JSON. Errors are returned as tool results with `isError: true` and a message the model can act on.

Field	Type	Description
`id`	string	The job id.
`object`	string	Always "job".
`kind`	string (enum)	brief, extract or report.
`livemode`	boolean	false with a test key.
`status`	string (enum)	queued, running, succeeded or failed.
`matter_id`	string or null	The matter.
`created_at`	string or null	ISO 8601 timestamp.
`completed_at`	string or null	ISO 8601 timestamp.
`charged_credits`	number	Credits charged (0 for report analyses).
`result`	object	When succeeded: the job's result, for a report the structured analysis.
`error`	object	When failed: type and message.

Field	Type	Description
`sample`	object	Present only for test keys.

start_cross_exam

Start a cross-examination practice session with a witness modelled on an expert class.

Title: Start a cross-examination Scope: `cross_exam` Writes, charged at start

FULL DESCRIPTION

Starts a cross-examination practice session on a matter, with a witness modelled on the chosen expert class and grounded in the matter's facts. mode oppose examines the other side's expert; prepare rehearses the firm's own expert. difficulty sets how cooperative the witness is.

The client then calls cross_exam_turn with the returned session id, one question at a time.

Price: 0.2 credits per session, charged when the session starts; a session holds up to 25 questions and each answer inside it is free. Test keys are never charged.

Use limit shown to the client: Attorney work product and consulting analysis only. Not evidence and not a testifying expert's opinion.

SCOPES

Requires the `cross_exam` scope. A call without it is refused with an error naming the missing scope.

CALL ARGUMENTS

Argument	Type	Required	Description and restrictions
`matter_id`	string	Yes	The matter id. Maximum 60 characters.
`expert_class`	string	Yes	The expert class id from list_expert_classes. Maximum 120 characters.
`mode`	string (enum)	No	oppose (default) or prepare.
`difficulty`	string (enum)	No	cooperative (default) or hostile.

Unknown arguments are refused. All strings are trimmed for emptiness checks.

RETURN RESULTS

The result carries the object below as `structuredContent`, plus two text blocks: a short summary for the model and the same object serialized as JSON. Errors are returned as tool results with `isError: true` and a message the model can act on.

Field	Type	Description
`id`	string	The session id.
`object`	string	Always "cross_examination".
`livemode`	boolean	false with a test key.
`status`	string	The session status.
`matter_id`	string	The matter.
`expert_class`	string or null	The expert class.

Field	Type	Description
`mode`	string	oppose or prepare.
`difficulty`	string	cooperative or hostile.
`questions_asked`	number	Questions asked so far.
`question_limit`	number	Maximum questions in the session (25).
`charged_credits`	number	Credits charged.
`transcript`	array of object	Questions and answers so far.

cross_exam_turn

Put one question to the witness in a session and get the answer.

Title: Ask the witness a question Scope: `cross_exam` Writes, free inside a paid session

FULL DESCRIPTION

Sends one question to the witness in an existing cross-examination session and returns the answer. Each answer is free inside a session that has been paid for. A session holds up to 25 questions; further questions are refused.

Only sessions of the user's firm can be used. Any other id returns the same error as an id that does not exist.

Use limit shown to the client: Attorney work product and consulting analysis only. Not evidence and not a testifying expert's opinion.

SCOPES

Requires the `cross_exam` scope. A call without it is refused with an error naming the missing scope.

CALL ARGUMENTS

Argument	Type	Required	Description and restrictions
`id`	string	Yes	The session id from start_cross_exam. Maximum 60 characters.
`message`	string	Yes	The question, as it would be put to the witness. 1 to 2,000 characters.

Unknown arguments are refused. All strings are trimmed for emptiness checks.

RETURN RESULTS

The result carries the object below as `structuredContent`, plus two text blocks: a short summary for the model and the same object serialized as JSON. Errors are returned as tool results with `isError: true` and a message the model can act on.

Field	Type	Description
`object`	string	Always "cross_examination_turn".
`cross_examination_id`	string	The session id.
`question_number`	number	The number of this question in the session.
`question`	string	The question as received.
`answer`	string	The witness's answer.
`questions_remaining`	number	Questions left in the session.

Errors

Protocol errors (malformed JSON, unknown method, unknown tool) are returned as JSON-RPC errors. Errors during a tool call, such as a missing argument, an id that does not exist, an unpaid balance or a fair-use limit, are returned as a tool result with `isError: true` and a plain message, plus the structured error object (`type`, `message`, and where relevant `reason`), so the model can correct itself or explain the refusal to the user.

Coming next

Not available on the server today, and not part of this specification:

- **analyze_report**: analysis of an expert report (methodology, opinions, assumptions, data sources and apparent vulnerabilities). It first needs a way to register a document on a matter through MCP. Report analysis is available today in the Supreme Mind app. New tools will be described in a revised specification before they are listed.

Revision history

Date	Version	Change
4 November 2026	1.0.0	First published specification.
5 November 2026	1.1.0	create_brief: new optional groundings argument; when it is omitted, the expert's data sources run on what the matter supports. get_brief: new groundings field reporting which sources ran. analyze_report removed from the MCP server and listed under Coming next. No change to authentication, scopes, pricing or the other tools.

Contact

Richard John Van Der Aa, Founder and CEO, Supreme Mind AI, Inc. · richard@suprememind.ai · suprememind.ai/developers