Independent Submission
Internet-Draft
Intended status: Informational
Expires: 24 September 2026

C. Hood
Nomotic, Inc.

23 March 2026

AGTP Standard Extended Method Vocabulary
draft-hood-agtp-standard-methods-00

Abstract

The Agent Transfer Protocol (AGTP) defines a core method vocabulary (Tier 1) of twelve intent-based methods covering the most common agent operations. This document defines the Tier 2 Standard Extended Method Vocabulary: methods registered in the IANA AGTP Method Registry that are available for use in any AGTP implementation but are not required for baseline compliance. Methods are organized into six categories reflecting the full operational range of AI agent systems: ACQUIRE, COMPUTE, TRANSACT, INTEGRATE, COMMUNICATE, and ORCHESTRATE. This document also specifies the QUOTE method referenced in the AGTP core specification for pre-flight resource cost estimation. The six-category taxonomy is aligned with the ACTION framework described in [AGENTIC-API].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 24 September 2026.

Copyright Notice

Table of Contents

1. Introduction

1.1. Motivation

The AGTP core specification deliberately limits its Tier 1 method set to twelve methods that represent the universal operations of AI agent systems: data retrieval, content synthesis, resource booking, scheduling, context management, delegation, collaboration, confirmation, escalation, notification, capability discovery, and session suspension. This minimalist core enables broad baseline compliance without requiring implementations to support methods they do not need.

Real-world agent deployments require a richer vocabulary. An agent performing data analysis needs ANALYZE, CLASSIFY, and EVALUATE. An agent connecting enterprise systems needs SYNC, MERGE, and CONNECT. An agent managing complex workflows needs CHAIN, BATCH, and RUN. An agent executing commercial operations needs AUTHORIZE, CANCEL, and PURCHASE. These operations are well-defined, widely needed, and deserve standardized method names, but they are not universal enough to belong in the Tier 1 core.

This document defines the Tier 2 Standard Extended Method Vocabulary: a registered set of methods available to any AGTP implementation. Tier 2 methods are not required for AGTP compliance but *SHOULD* be implemented where their semantics apply.

1.2. Empirical Justification

Agent Transfer Protocol is motivated by measured agent performance gains. Independent benchmarking (4,800 trials across Claude, Grok, and OpenAI-family models) compared pure CRUD endpoints against a mixed catalog containing both CRUD and the semantic verbs defined in this vocabulary.

In mixed-paradigm conditions (Tier 1 CRUD + optional Tier 2 semantic methods), agentic verbs improved exact-match accuracy by 1029 percentage points depending on model family (Claude +29 pp, Grok +18 pp, OpenAI +10 pp; all $p<0.001$). Parameter fidelity and clarification rate also improved.

A description-swap ablation (J1/J2) isolated the mechanism: when CRUD paths received agentic-style descriptions, performance collapsed dramatically on weaker models (Grok -39 pp, OpenAI -43 pp). When agentic paths received CRUD-style descriptions, performance held nearly flat on Claude and degraded far less on other models. This demonstrates that *semantic method names encode intent more resiliently* than REST-style generics, even when documentation quality varies — precisely the scenario Tier 2 methods are designed to address.

Two-stage discovery (as required by AGTP session establishment) incurs a modest cost (1217 pp) but does not erase the net gain. The QUOTE method and Supported-Methods header further reduce unnecessary invocations.

These results validate the tiered design: Tier 1 provides the operational simplicity developers require, while the registered Tier 2 vocabulary delivers measurable agent performance and resilience. Implementations are encouraged to adopt Tier 2 methods where their domain semantics align.

1.3. Taxonomy

Methods are organized into six categories. Each category has a distinct operational character:

ACQUIRE: Retrieve data, resources, or state without modifying it. Agents observe, locate, and extract information.

COMPUTE: Process or transform information to produce a derived result. Agents analyze, classify, summarize, and generate outputs.

TRANSACT: Execute operations that alter system state or complete an external commitment. Agents book, purchase, authorize, and register.

INTEGRATE: Connect, synchronize, or unify data and services across system boundaries. Agents merge, link, sync, and map across silos.

COMMUNICATE: Deliver signals, messages, or structured outputs to recipients. Agents notify, alert, broadcast, and reply.

ORCHESTRATE: Coordinate, sequence, and manage workflows across tasks, agents, and time. Agents chain, batch, route, and schedule.

The category taxonomy aligns with the ACTION framework defined in [AGENTIC-API]. The INTEGRATE category is the key distinction from simpler models: cross-system unification operations are neither transactions (no external commitment) nor computations (no transformation of content into a new analytical form). They are connective operations with distinct failure modes, conflict policies, and reversibility characteristics.

1.4. Method Registration

All methods in this document are registered in the IANA AGTP Method Registry per the registration procedure defined in [AGTP] Section 9.2. Implementations *MUST* list supported Tier 2 methods in the Supported-Methods response header at session establishment. Clients *SHOULD* query this header before invoking Tier 2 methods.

2. Terminology

The key words "*MUST*", "*MUST NOT*", "*REQUIRED*", "*SHALL*", "*SHALL NOT*", "*SHOULD*", "*SHOULD NOT*", "*RECOMMENDED*", "*NOT RECOMMENDED*", "*MAY*", and "*OPTIONAL*" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals.

Tier 1 Method: A core AGTP method defined in [AGTP] Section 7.2, required for baseline AGTP compliance.

Tier 2 Method: A standard extended AGTP method defined in this document, registered in the IANA AGTP Method Registry, available to any implementation but not required for baseline compliance.

Idempotent: A method is idempotent if multiple identical invocations produce the same result as a single invocation. Non-idempotent methods *MUST NOT* be retried without explicit application-layer handling.

3. ACQUIRE Category Methods

ACQUIRE methods retrieve data, resources, or state without modifying it. Agents use ACQUIRE methods to observe, locate, and extract information from internal or external sources. All ACQUIRE methods are idempotent unless noted.

3.1. FETCH

Purpose: Retrieve a specific resource by identifier. Distinguished from QUERY (which expresses an information need without specifying a location) by targeting a known resource at a known address.

Use case: An agent retrieves the current version of a contract document by its document ID before passing it to ANALYZE.

Parameter	Required	Description
resource_id	*MUST*	Identifier or URI of the resource to retrieve
format	*MAY*	Requested response format
version	*MAY*	Specific version of the resource to retrieve
if_modified_since	*MAY*	ISO 8601 timestamp; return only if modified after

Table 1: FETCH Parameters

Response: Resource content with metadata. Idempotent: Yes. Error codes: 404, 403, 408.

3.2. SEARCH

Purpose: Execute a structured search query against a data source and return matching results. Distinguished from QUERY by accepting structured query syntax (field filters, range constraints, sort order) rather than natural language intent.

Use case: An agent locates all purchase orders above $50,000 in pending status, sorted by submission date.

Parameter	Required	Description
query	*MUST*	Structured query expression or filter object
scope	*SHOULD*	Data sources or indices to search
limit	*MAY*	Maximum number of results to return
offset	*MAY*	Pagination offset
sort	*MAY*	Sort field and order

Table 2: SEARCH Parameters

Response: Result set with pagination metadata. Idempotent: Yes. Error codes: 400, 422.

3.3. SCAN

Purpose: Enumerate or iterate over a collection of resources, potentially applying lightweight filters. Distinguished from SEARCH by prioritizing completeness over relevance ranking.

Use case: An agent audits all agents in a governance zone by iterating over the full registry, page by page, to build a compliance report.

Parameter	Required	Description
collection	*MUST*	Identifier of the collection to scan
filter	*MAY*	Simple filter criteria
cursor	*MAY*	Continuation cursor from a prior SCAN response
page_size	*MAY*	Number of items to return per page

Table 3: SCAN Parameters

Response: Page of results with continuation cursor if more items remain. Idempotent: Yes. Error codes: 404, 408.

3.4. PULL

Purpose: Retrieve pending items from a queue or stream. Consumes items from the source; the agent takes ownership of pulled items.

Use case: An order-processing agent pulls the next batch of unprocessed orders from the fulfillment queue to begin packing instructions.

Parameter	Required	Description
source	*MUST*	Queue or stream identifier
max_items	*MAY*	Maximum number of items to pull
visibility_timeout	*MAY*	Seconds for which pulled items are hidden from other consumers

Table 4: PULL Parameters

Response: List of pulled items with item IDs for acknowledgment.
Idempotent: No. Error codes: 404, 409.

3.5. FIND

Purpose: Locate an agent, resource, or entity matching specified criteria. Distinguished from SEARCH by targeting agent or entity discovery rather than content retrieval.

Use case: An orchestrator locates all available analyst agents in the partner governance zone that support the VALIDATE method.

Parameter	Required	Description
criteria	*MUST*	Structured criteria for the entity to find
namespace	*SHOULD*	Registry or namespace to search within
limit	*MAY*	Maximum results to return

Table 5: FIND Parameters

Response: List of matching entities with identifiers and metadata.
Idempotent: Yes. Error codes: 404, 422.

3.6. ANALYZE

Purpose: Perform observational analysis or pattern detection on a dataset or system state. Distinguished from COMPUTE methods such as SUMMARIZE or CLASSIFY by returning findings, patterns, and anomalies rather than transforming the input. ANALYZE does not modify state.

Use case: An agent detects anomalous spending patterns in a quarterly expense dataset by analyzing transaction frequency and amount distributions against historical baselines.

Parameter	Required	Description
target	*MUST*	Dataset, system identifier, or inline data to analyze
analysis_type	*SHOULD*	Pattern type: anomaly, trend, distribution, correlation
time_range	*MAY*	ISO 8601 time window for time-series data
sensitivity	*MAY*	Detection sensitivity threshold (0.0-1.0)
explain	*MAY*	If true, include reasoning behind each finding

Table 6: ANALYZE Parameters

Response: Findings document with detected patterns, anomaly indicators, and confidence scores per finding. Idempotent: Yes. Error codes: 400, 422, 503.

4. COMPUTE Category Methods

COMPUTE methods process or transform information to produce a derived result. Agents use COMPUTE methods to analyze, classify, summarize, and generate outputs from existing inputs. COMPUTE methods are typically idempotent given the same input.

4.1. EXTRACT

Purpose: Pull structured information from unstructured or semi-structured source content.

Use case: An agent extracts party names, dates, and obligation clauses from an unstructured legal contract into a structured JSON schema.

Parameter	Required	Description
source	*MUST*	Content to extract from
schema	*MUST*	Target structure for extracted data
confidence_threshold	*MAY*	Minimum confidence for included extractions

Table 7: EXTRACT Parameters

Response: Extracted data conforming to the declared schema, with confidence scores per field. Idempotent: Yes. Error codes: 400, 422.

4.2. FILTER

Purpose: Apply criteria to a dataset and return only matching records.

Use case: An agent filters a product catalog to items that are in-stock, priced under $200, and tagged with a specific category.

Parameter	Required	Description
data	*MUST*	Dataset to filter (inline or by reference)
criteria	*MUST*	Filter expression
output_format	*MAY*	Format for filtered results

Table 8: FILTER Parameters

Response: Filtered dataset. Idempotent: Yes. Error codes: 400, 422.

4.3. VALIDATE

Purpose: Check input data, a document, or a proposed action against a schema, policy, or rule set. Returns validation results but does not modify state.

Use case: An agent validates an invoice payload against the accounts payable schema and business rules before submitting it for payment.

Parameter	Required	Description
target	*MUST*	Data, document, or action to validate
schema	*SHOULD*	Schema or rule set to validate against
strict	*MAY*	If true, treat warnings as failures

Table 9: VALIDATE Parameters

Response: Validation result with pass/fail status, errors, and warnings. Idempotent: Yes. Error codes: 400, 422.

4.4. TRANSFORM

Purpose: Convert data from one format, schema, or structure to another.

Use case: An agent converts a vendor’s proprietary order payload into the internal canonical order schema before routing to fulfillment.

Parameter	Required	Description
source	*MUST*	Data to transform
source_format	*SHOULD*	Format of the input data
target_format	*MUST*	Desired output format or schema
mapping	*MAY*	Field mapping specification

Table 10: TRANSFORM Parameters

Response: Transformed data in the target format. Idempotent: Yes. Error codes: 400, 422.

4.5. TRANSLATE

Purpose: Convert content between human languages.

Use case: An agent translates product descriptions from English into French and German to prepare a multilingual catalog update.

Parameter	Required	Description
content	*MUST*	Text or structured content to translate
source_language	*SHOULD*	BCP 47 language tag; auto-detected if absent
target_language	*MUST*	BCP 47 language tag of desired output
formality	*MAY*	formal or informal

Table 11: TRANSLATE Parameters

Response: Translated content with source and target language codes. Idempotent: Yes. Error codes: 400, 422.

4.6. NORMALIZE

Purpose: Standardize data values to a canonical form (dates, phone numbers, addresses, currency amounts, units).

Use case: An agent normalizes customer records ingested from three regional CRMs using different date formats, phone conventions, and address schemas into a unified canonical format.

Parameter	Required	Description
data	*MUST*	Data to normalize
type	*MUST*	Normalization type: date, phone, address, currency, unit
locale	*MAY*	Locale context for normalization

Table 12: NORMALIZE Parameters

Response: Normalized data with original and canonical forms. Idempotent: Yes. Error codes: 400, 422.

4.7. PREDICT

Purpose: Apply a model or function to input data and return a predicted output or probability estimate.

Use case: An agent predicts the likelihood of customer churn for each account in a segment, returning probability scores and top contributing factors for accounts above the risk threshold.

Parameter	Required	Description
input	*MUST*	Input data for the prediction
model_id	*SHOULD*	Identifier of the model to use
confidence_floor	*MAY*	Minimum confidence required to return a result
explain	*MAY*	If true, include feature attribution in response

Table 13: PREDICT Parameters

Response: Prediction result with confidence score and optional explanation. Idempotent: Yes. Error codes: 400, 422, 503.

4.8. RANK

Purpose: Sort or score a set of items by relevance, quality, or a declared criterion.

Use case: An agent ranks a shortlist of job candidates by predicted role fit, ordering them for recruiter review.

Parameter	Required	Description
items	*MUST*	List of items to rank
criterion	*MUST*	Ranking criterion or scoring function
limit	*MAY*	Return only the top-N items

Table 14: RANK Parameters

Response: Ranked list with scores per item. Idempotent: Yes. Error codes: 400, 422.

4.9. CLASSIFY

Purpose: Assign one or more items to categories from a known classification scheme. Distinguished from RANK (which scores and orders items) and PREDICT (which estimates future states) by assigning categorical labels to existing data.

Use case: An agent classifies incoming support tickets into one of eight predefined categories to route them to the appropriate team.

Parameter	Required	Description
items	*MUST*	Items or data records to classify
taxonomy	*MUST*	Classification scheme or label set to use
multi_label	*MAY*	If true, each item may receive more than one label
confidence_threshold	*MAY*	Minimum confidence for a label assignment

Table 15: CLASSIFY Parameters

Response: Classification result with assigned labels and confidence scores per item. Idempotent: Yes. Error codes: 400, 422.

4.10. CALCULATE

Purpose: Perform numeric, logical, or financial computations on structured inputs. Distinguished from PREDICT (probabilistic) and EVALUATE (qualitative comparison) by operating on deterministic mathematical or rule-based logic.

Use case: An agent calculates the total cost of a procurement order including applicable taxes, shipping fees, and bulk discount rules.

Parameter	Required	Description
expression	*MUST*	Computation to perform, as a structured formula object or registered formula ID
inputs	*MUST*	Named input values for the computation
precision	*MAY*	Number of decimal places in the result
currency	*MAY*	ISO 4217 currency code for financial computations

Table 16: CALCULATE Parameters

Response: Computation result with inputs echoed, formula applied, and output value. Idempotent: Yes. Error codes: 400, 422.

4.11. EVALUATE

Purpose: Compare a target against benchmarks, rules, standards, or criteria and return a qualitative or scored assessment. Distinguished from VALIDATE (pass/fail against a schema) by returning a graded assessment with explanatory context.

Use case: An agent evaluates a vendor proposal against a weighted scorecard covering price, delivery timeline, compliance posture, and references, returning a composite score and narrative rationale.

Parameter	Required	Description
target	*MUST*	Item, document, or entity to evaluate
criteria	*MUST*	Evaluation criteria or scorecard definition
weights	*MAY*	Relative weights for each criterion
rubric	*MAY*	Scoring rubric mapping values to qualitative labels

Table 17: EVALUATE Parameters

Response: Evaluation result with per-criterion scores, composite score, and narrative rationale. Idempotent: Yes. Error codes: 400, 422.

4.12. GENERATE

Purpose: Produce textual, structured, or code output from a source specification, dataset, or prompt. Distinguished from SUMMARIZE (which condenses existing content) by creating new content that did not exist in the source input.

Use case: An agent generates API documentation from an OpenAPI specification, producing endpoint descriptions, parameter tables, and example payloads.

Parameter	Required	Description
source	*SHOULD*	Input specification, data, or context for generation
output_type	*MUST*	Type of output: text, code, json, html, markdown
format	*MAY*	Output format details (e.g., language for code generation)
length	*MAY*	Approximate target length or size
style	*MAY*	Style guidance: formal, casual, technical, neutral

Table 18: GENERATE Parameters

Security note: GENERATE with output_type: code *MUST* require human or governance review before passing to RUN. Implementations providing a pipeline from GENERATE to RUN *MUST* insert an explicit VALIDATE or approval step between them.

Response: Generated content with output_type field. Idempotent: Yes (for identical inputs). Error codes: 400, 422, 503.

4.13. RECOMMEND

Purpose: Generate a ranked list of suggestions, options, or actions tailored to a declared context, goal, and constraints. Distinguished from RANK (which scores a provided list) by generating the candidate set itself based on the agent’s understanding of the context.

Use case: An agent recommends three product accessories to a customer based on their purchase history, stated budget, and current browsing context.

Parameter	Required	Description
context	*MUST*	User, session, or situational context for the recommendation
goal	*MUST*	What the recommendation is optimizing for
constraints	*SHOULD*	Hard limits: budget, category, availability, exclusions
limit	*MAY*	Maximum number of recommendations to return
explain	*MAY*	If true, include rationale for each recommendation

Table 19: RECOMMEND Parameters

Response: Ordered list of recommendations with scores, rationale, and confidence per item. Idempotent: Yes. Error codes: 400, 422, 503.

5. TRANSACT Category Methods

TRANSACT methods perform state-changing operations with external systems, records, or commitments. TRANSACT methods are not idempotent by default.

5.1. QUOTE

Purpose: Pre-flight cost estimation for a proposed method invocation. The requesting agent submits a proposed method call; the server returns a cost estimate without executing the method. Servers supporting Budget-Limit *SHOULD* implement QUOTE.

Use case: An agent estimates the token cost of a large-document SUMMARIZE operation before committing, to verify it stays within the session budget.

Parameter	Required	Description
method	*MUST*	The AGTP method for which a cost estimate is requested
parameters	*MUST*	The parameters that would be passed to that method
budget_units	*SHOULD*	Units in which to express the estimate

Table 20: QUOTE Parameters

Response: Cost estimate in Cost-Estimate response header and response body, with confidence range. Idempotent: Yes. Error codes: 400, 404, 422.

```
AGTP/1.0 200 OK
Task-ID: task-quote-01
Cost-Estimate: tokens=8500 compute-seconds=2.3
Content-Type: application/agtp+json

{
  "status": 200,
  "task_id": "task-quote-01",
  "result": {
    "method": "QUERY",
    "estimated_cost": {
      "tokens": {"min": 7200, "expected": 8500, "max": 11000},
      "compute_seconds": {"min": 1.8, "expected": 2.3, "max": 4.0}
    },
    "confidence": 0.82
  }
}
```

5.2. REGISTER

Purpose: Create a new record, entity, or subscription in a target system.

Use case: An onboarding agent registers a new employee in the identity management system, the access control system, and the payroll platform as three sequential REGISTER calls within a single session.

Parameter	Required	Description
entity_type	*MUST*	Type of entity to register
data	*MUST*	Registration data
idempotency_key	*SHOULD*	Client-provided key for duplicate detection

Table 21: REGISTER Parameters

Response: Registration receipt with new entity identifier. Idempotent: No (use idempotency_key for safe retry). Error codes: 409, 422.

5.3. SUBMIT

Purpose: Deliver data, a document, or a work item to a processing system or queue for handling.

Use case: An agent submits a completed expense report to the finance system’s approval queue for human review.

Parameter	Required	Description
target	*MUST*	Destination system or queue
payload	*MUST*	Data or document to submit
priority	*MAY*	Submission priority
callback	*MAY*	AGTP endpoint for delivery confirmation

Table 22: SUBMIT Parameters

Response: Submission receipt with tracking identifier. Idempotent: No. Error codes: 400, 409, 503.

5.4. AUTHORIZE

Purpose: Approve permissions, credentials, or access for an entity, action, or resource. Distinguished from VALIDATE (which checks conformance) by issuing an active grant rather than a check result.

Use case: An identity agent authorizes a contractor’s access request for a restricted project repository, generating a scoped access token with a defined expiry.

Parameter	Required	Description
subject	*MUST*	Agent-ID, principal, or entity to authorize
resource	*MUST*	Resource or action being authorized
scope	*MUST*	Permissions being granted
ttl	*SHOULD*	Time-to-live for the authorization in seconds
conditions	*MAY*	Conditional constraints on the authorization

Table 23: AUTHORIZE Parameters

Security note: AUTHORIZE *MUST* carry authorization:grant in the agent’s Authority-Scope. The granted scope *MUST NOT* exceed the authorizing agent’s own declared scope (the anti-laundering constraint from [AGTP] Section 7.2.7 applies to all authorization grants). 451 Scope Violation *MUST* be returned if either constraint is violated. Idempotent: No. Error codes: 403, 409, 451.

5.5. CANCEL

Purpose: Revoke or reverse a previously scheduled or committed transaction. Distinguished from lifecycle operations (agent suspension or revocation) by targeting a specific transaction, booking, or reservation rather than an agent’s registry state.

Use case: An agent cancels a conference room booking on behalf of the principal, releasing availability and notifying other attendees.

Parameter	Required	Description
target_id	*MUST*	Identifier of the transaction, booking, or reservation to cancel
reason	*SHOULD*	Structured cancellation reason
notify_parties	*MAY*	If true, NOTIFY relevant parties on cancellation
refund_policy	*MAY*	Refund handling: full, partial, none

Table 24: CANCEL Parameters

Response: Cancellation receipt with confirmation and any refund details. Idempotent: No (cancellation of an already-cancelled item *SHOULD* return 409 Conflict). Error codes: 404, 409, 422.

5.6. TRANSFER

Purpose: Move ownership or custody of a resource from one principal or agent to another.

Use case: An agent transfers custodianship of a finalized contract to the legal archive system, removing it from the active working directory and updating ownership records.

Parameter	Required	Description
resource_id	*MUST*	Resource to transfer
from_principal	*MUST*	Current owner
to_principal	*MUST*	New owner
reason	*SHOULD*	Reason for transfer

Table 25: TRANSFER Parameters

Security note: TRANSFER *MUST* verify that the requesting agent holds Authority-Scope for the resource type being transferred. 451 Scope Violation *MUST* be returned if not. Idempotent: No. Error codes: 403, 404, 409, 451.

5.7. PURCHASE

Purpose: Execute a financial transaction to acquire a resource, service, or allocation.

Use case: A travel agent purchases a confirmed airline seat after the principal approves the itinerary, carrying the principal's payment method identifier.

Parameter	Required	Description
item	*MUST*	Item or service being purchased
principal_id	*MUST*	Principal authorizing the purchase
amount	*MUST*	Purchase amount and currency
payment_method	*SHOULD*	Payment instrument identifier
confirm_immediately	*MAY*	Boolean; if false, creates a hold

Table 26: PURCHASE Parameters

Security note: PURCHASE *MUST* carry payments:purchase in the agent's Authority-Scope. 451 Scope Violation *MUST* be returned if absent. PURCHASE *MUST* validate against the Budget-Limit header if present. Idempotent: No. Error codes: 403, 409, 451, 452.

5.8. SIGN

Purpose: Apply a cryptographic signature to a document, assertion, or data artifact using the agent's or principal's signing key.

Use case: An agent signs a finalized service agreement on behalf of the principal using the organization's governance key, producing a timestamped, non-repudiable signature record.

Parameter	Required	Description
payload	*MUST*	Data to sign
key_id	*SHOULD*	Identifier of the signing key to use
algorithm	*MAY*	Signature algorithm; defaults to implementation default

Table 27: SIGN Parameters

Response: Signed artifact with signature metadata. Idempotent: No (signatures include timestamps). Error codes: 403, 422.

5.9. LOG

Purpose: Write a structured record to an audit trail or operational log. Agents *SHOULD* use LOG for governance-significant events rather than embedding log data in other method payloads.

Use case: A financial agent logs each step of a multi-stage wire transfer approval workflow as discrete LOG events, creating an audit trail for compliance reporting.

Parameter	Required	Description
event_type	*MUST*	Structured event category
data	*MUST*	Event data
severity	*SHOULD*	info, warning, error, or critical
correlation_id	*MAY*	ID linking this event to a parent task or session

Table 28: LOG Parameters

Response: Log receipt with record identifier. Idempotent: No. Error codes: 400, 503.

5.10. PUBLISH

Purpose: Make content or data available to a defined audience or channel.

Use case: An agent publishes a finalized research summary to the organizational knowledge base, making it discoverable to all users with knowledge:read scope.

Parameter	Required	Description
content	*MUST*	Content to publish
channel	*MUST*	Target channel or audience identifier
format	*SHOULD*	Content format
schedule_at	*MAY*	ISO 8601 timestamp for deferred publication

Table 29: PUBLISH Parameters

Response: Publication receipt with content identifier and channel confirmation. Idempotent: No. Error codes: 400, 403, 409.

6. INTEGRATE Category Methods

INTEGRATE methods connect, synchronize, or unify data and services across system boundaries. Agents use INTEGRATE methods to eliminate fragmentation across silos, align state across environments, and establish relationships between entities. The INTEGRATE category is distinct from TRANSACT (no external commitment or irreversible state change) and from COMPUTE (no transformation of content into a new analytical form). INTEGRATE methods concern structural alignment and relational connectivity.

6.1. MERGE

Purpose: Combine two or more datasets, documents, or resources into a unified result, resolving conflicts according to a declared policy.

Use case: An agent merges duplicate customer profiles detected across two acquired subsidiary databases into a single canonical record, applying a "most recently updated" conflict resolution strategy.

Parameter	Required	Description
sources	*MUST*	List of resources or inline data to merge
strategy	*SHOULD*	Merge strategy: union, intersection, latest, manual
conflict_resolution	*MAY*	Policy for conflicting fields

Table 30: MERGE Parameters

Response: Merged result with conflict report if applicable. Idempotent: No. Error codes: 400, 409, 422.

6.2. LINK

Purpose: Create a persistent association or relationship between two entities in a system of record. Distinguished from MAP (which defines structural schema correspondence) by creating a relationship record between specific instances.

Use case: An agent links a user’s enterprise SSO identity to their account in a third-party analytics platform, establishing a persistent cross-system identity association.

Parameter	Required	Description
source	*MUST*	Source entity identifier
target	*MUST*	Target entity identifier
relationship	*MUST*	Type of relationship
metadata	*MAY*	Additional relationship attributes

Table 31: LINK Parameters

Response: Link record with relationship identifier. Idempotent: No. Error codes: 404, 409.

6.3. SYNC

Purpose: Reconcile the state of a local resource with a remote authoritative source, aligning records bidirectionally or unidirectionally.

Use case: An agent synchronizes the local product inventory cache with the warehouse management system, pulling delta changes since the last sync timestamp and pushing any local updates that occurred offline.

Parameter	Required	Description
resource	*MUST*	Resource to synchronize
remote	*MUST*	Authoritative source URI
direction	*SHOULD*	pull, push, or bidirectional
conflict_policy	*MAY*	remote_wins, local_wins, or manual

Table 32: SYNC Parameters

Response: Sync receipt with change summary. Idempotent: No. Error codes: 404, 409, 503.

6.4. IMPORT

Purpose: Bring external data into the agent’s operational context or a designated storage target from a non-AGTP-native source. Distinguished from FETCH (which retrieves a known resource by ID from an AGTP-native address) by ingesting from external systems.

Use case: An agent imports a CSV export of contact records from a legacy CRM into the new system, resolving format differences and logging import statistics.

Parameter	Required	Description
source	*MUST*	URI or inline data to import
target	*MUST*	Destination identifier within the agent's context
format	*SHOULD*	Format of the source data
conflict_policy	*MAY*	Behavior on conflict: overwrite, skip, merge

Table 33: IMPORT Parameters

Response: Import receipt with record count and any conflict resolutions. Idempotent: No. Error codes: 400, 409, 422.

6.5. MAP

Purpose: Define a structural correspondence between two schemas, data models, or field sets. Distinguished from TRANSFORM (which applies a mapping to convert a specific payload) by producing a reusable mapping definition rather than executing a one-time conversion.

Use case: An agent generates a field-level mapping between an external vendor's order schema and the internal canonical order schema, producing a reusable mapping document that subsequent TRANSFORM calls can reference.

Parameter	Required	Description
source_schema	*MUST*	Schema or data model to map from
target_schema	*MUST*	Schema or data model to map to
strategy	*SHOULD*	Mapping strategy: exact, semantic, custom
unmapped_policy	*MAY*	Behavior for fields with no target: ignore, flag, preserve

Table 34: MAP Parameters

Response: Mapping document with per-field correspondence and confidence scores. Idempotent: Yes. Error codes: 400, 422.

6.6. CONNECT

Purpose: Establish a conduit, session, or integration channel between two systems, services, or agents. Distinguished from LINK (which creates a record-level association) by establishing an active communication or data pathway.

Use case: An agent establishes a streaming data connection between a sensor telemetry feed and the observability platform, configuring the channel parameters and returning a handle for subsequent operations.

Parameter	Required	Description
source	*MUST*	Source system or agent identifier
destination	*MUST*	Destination system or agent identifier
channel_type	*SHOULD*	Type of connection: stream, webhook, polling, event-bus
config	*MAY*	Channel-specific configuration parameters
ttl	*MAY*	Connection lifetime in seconds; persists until CANCEL if absent

Table 35: CONNECT Parameters

Response: Connection handle with connection ID and channel metadata. Idempotent: No. Error codes: 404, 409, 503.

6.7. EMBED

Purpose: Insert one component, dataset, or agent capability into another as a nested or composed element. Used when an integration requires one system to host or expose the functionality of another within its own operational context.

Use case: An agent embeds a third-party risk-scoring model’s output into the organization’s credit decisioning workflow as a named sub-component, making its scores available inline within decisioning records.

Parameter	Required	Description
source	*MUST*	Component, dataset, or capability to embed
target	*MUST*	Container or context to embed into
binding	*SHOULD*	Named reference or mount point within the target
version	*MAY*	Specific version of the source to embed

Table 36: EMBED Parameters

Response: Embedding receipt with binding reference and version. Idempotent: No. Error codes: 400, 404, 409.

7. COMMUNICATE Category Methods

COMMUNICATE methods deliver signals, messages, or structured outputs to recipients. Agents use COMMUNICATE methods to push information, respond to incoming requests, and surface results to humans and systems.

7.1. ALERT

Purpose: Send a high-priority, time-sensitive message to a recipient or group. Distinguished from NOTIFY (Tier 1, general-purpose async push) by signaling urgency and requiring acknowledgment handling if not acknowledged within a declared timeout.

Use case: An infrastructure monitoring agent sends a critical alert to the on-call team when database latency exceeds the SLA threshold, carrying metric value, threshold, and recommended escalation path.

Parameter	Required	Description
recipient	*MUST*	Target Agent-ID, human endpoint, or group
message	*MUST*	Alert content with severity and context
severity	*MUST*	critical, high, or warning
acknowledge_by	*SHOULD*	ISO 8601 deadline for acknowledgment
escalation_path	*MAY*	Fallback recipient if not acknowledged by deadline

Table 37: ALERT Parameters

Response: Alert delivery receipt with alert ID. Implementations *SHOULD* track acknowledgment state and trigger escalation on timeout. Idempotent: No. Error codes: 400, 404, 503.

7.2. BROADCAST

Purpose: Disseminate information simultaneously to multiple recipients or a defined group. Distinguished from NOTIFY (targeted at a specific recipient) by addressing an audience rather than an individual and not expecting individual responses.

Use case: An agent broadcasts a scheduled maintenance window notification to all registered subscriber endpoints across three governance zones.

Parameter	Required	Description
audience	*MUST*	Group identifier, zone, or list of recipient identifiers
content	*MUST*	Broadcast payload
channel	*SHOULD*	Delivery channel: agtp, email, webhook, sms
expiry	*MAY*	Timestamp after which the broadcast should not be delivered

Table 38: BROADCAST Parameters

Response: Broadcast receipt with message ID and delivery count. Idempotent: No. Error codes: 400, 403, 503.

7.3. REPLY

Purpose: Provide a direct, traceable response to an incoming NOTIFY, ESCALATE, or QUERY that explicitly requested a reply. Creates a response thread linked to the originating task.

Use case: A human-in-the-loop handler sends a REPLY to a pending ESCALATE, providing a decision and carrying the original escalation ID for thread continuity.

Parameter	Required	Description
in_reply_to	*MUST*	Task-ID of the message being replied to
content	*MUST*	Reply content
urgency	*SHOULD*	critical, informational, or background

Table 39: REPLY Parameters

Response: Reply receipt. Idempotent: No. Error codes: 404, 400.

7.4. SEND

Purpose: Deliver a message or payload to a recipient through a declared external channel. Distinguished from NOTIFY (AGTP-native async push) by targeting non-AGTP delivery channels.

Use case: An agent sends an email confirmation to a customer's registered email address after completing a booking, using a declared template identifier.

Parameter	Required	Description
recipient	*MUST*	Channel-specific address
channel	*MUST*	Delivery channel: email, webhook, sms, push
content	*MUST*	Message content
template_id	*MAY*	Template identifier for formatted delivery

Table 40: SEND Parameters

Response: Delivery receipt with channel-specific confirmation. Idempotent: No. Error codes: 400, 404, 503.

7.5. REPORT

Purpose: Generate and deliver a structured summary or analysis document to a principal or system. Distinguished from NOTIFY (raw payload push) by producing a formatted output artifact with declared structure.

Use case: An agent generates and delivers a weekly sales performance summary to the sales leadership distribution list in PDF format.

Parameter	Required	Description
report_type	*MUST*	Report category
scope	*MUST*	Data scope or time range for the report
recipient	*MUST*	Target principal or endpoint
format	*SHOULD*	Output format: pdf, json, html, markdown

Table 41: REPORT Parameters

Response: Report delivery receipt with document identifier.
Idempotent: No. Error codes: 400, 422, 503.

8. ORCHESTRATE Category Methods

ORCHESTRATE methods coordinate, sequence, and manage workflows, tasks, and agents across time and system boundaries. Agents use ORCHESTRATE methods to build composite execution flows, manage failure recovery, and direct work to appropriate handlers.

8.1. CHAIN

Purpose: Link a defined sequence of AGTP method invocations into a composite workflow where each step’s output may be used as input for subsequent steps. Distinguished from BATCH (which executes unrelated tasks in parallel) by enforcing sequential dependency between steps.

Use case: An agent chains SEARCH, followed by ANALYZE, followed by REPORT into a single declared workflow for weekly competitor intelligence, where each step uses the prior step’s output.

Parameter	Required	Description
steps	*MUST*	Ordered list of AGTP method calls with parameters
on_failure	*SHOULD*	Behavior if a step fails: abort, skip, retry, escalate
context_propagation	*MAY*	If true, each step’s output is available to subsequent steps
timeout	*MAY*	Maximum total execution time across all steps in seconds

Table 42: CHAIN Parameters

Response: Execution receipt with chain ID and per-step status. Idempotent: No. Error codes: 400, 408, 422.

8.2. BATCH

Purpose: Group multiple independent AGTP method calls for concurrent or bulk execution, reducing round-trip overhead. Distinguished from CHAIN (sequentially dependent) by executing steps independently without output dependency.

Use case: An agent batches twenty TRANSLATE calls for product descriptions into a single BATCH invocation, reducing latency from twenty serial round-trips to a single parallel execution.

Parameter	Required	Description
calls	*MUST*	List of independent AGTP method calls to execute
execution_mode	*SHOULD*	parallel or sequential; default parallel
partial_success	*MAY*	If true, return results for successful calls even if others fail

Table 43: BATCH Parameters

Response: Batch result containing individual outcome per call. Idempotent: Conditional (depends on idempotency of component calls). Error codes: 400, 408, 422.

8.3. MONITOR

Purpose: Establish ongoing observation of an agent, resource, or condition. Returns a subscription handle; the AGTP server delivers updates via NOTIFY when observed conditions change.

Use case: An agent establishes a monitor on the behavioral trust score of a set of sub-agents, receiving NOTIFY updates when any agent's score drops below the declared threshold.

Parameter	Required	Description
target	*MUST*	Agent-ID, resource URI, or condition expression
events	*MUST*	Event types to observe
callback	*MUST*	AGTP endpoint for NOTIFY delivery
threshold	*MAY*	Condition value that triggers notification
interval	*MAY*	Polling interval in seconds if event-based delivery is unavailable

Table 44: MONITOR Parameters

Response: Monitor subscription receipt with subscription identifier. To cancel monitoring, send NOTIFY with event type monitor_cancel and the subscription identifier. Idempotent: No. Error codes: 404, 400.

8.4. ROUTE

Purpose: Direct a task, message, or work item to the appropriate handler or agent based on declared routing criteria.

Use case: An agent routes an incoming customer inquiry to the appropriate specialized handling agent based on topic classification, account tier, and declared availability of candidate handlers.

Parameter	Required	Description
payload	*MUST*	Item to route
criteria	*MUST*	Routing criteria or rules
candidates	*SHOULD*	List of candidate Agent-IDs or endpoints
fallback	*MAY*	Default handler if no candidate matches

Table 45: ROUTE Parameters

Response: Routing decision with selected handler identifier. Idempotent: No. Error codes: 404, 422, 503.

8.5. RETRY

Purpose: Re-attempt a previously failed AGTP method invocation using a prior Task-ID. Distinguished from a new invocation by carrying the original Task-ID for idempotency checking at the server.

Use case: An agent retries a failed BOOK call after a 503 Unavailable, carrying the original Task-ID so the booking system can detect and suppress any duplicate if the original call was partially processed.

Parameter	Required	Description
original_task_id	*MUST*	Task-ID of the failed invocation
delay	*MAY*	Seconds to wait before retrying
max_attempts	*MAY*	Maximum total attempts including this one

Table 46: RETRY Parameters

Response: New task receipt linked to the original task. Idempotent: Yes (for idempotent original methods); No for non-idempotent originals. Error codes: 404, 409.

8.6. PAUSE

Purpose: Temporarily halt a scheduled or repeating workflow without terminating it. Distinguished from SUSPEND (Tier 1, which targets a session) by targeting a workflow or schedule record.

Use case: An agent pauses a recurring data pipeline workflow during a maintenance window, setting a resume timestamp at the scheduled end of the window.

Parameter	Required	Description
workflow_id	*MUST*	Identifier of the workflow or schedule to pause
reason	*SHOULD*	Reason for pause
resume_at	*MAY*	ISO 8601 timestamp for automatic resumption

Table 47: PAUSE Parameters

Response: Pause receipt with workflow status. Idempotent: No. Error codes: 404, 409.

8.7. RESUME

Purpose: Restart a paused workflow or a suspended session.

Use case: An agent resumes a paused procurement workflow after the required human approval has been received, using the resumption nonce issued at PAUSE time.

Parameter	Required	Description
workflow_id	*MUST*	Workflow or session identifier
resumption_nonce	*MUST* (for sessions)	Nonce issued at SUSPEND or PAUSE
checkpoint	*MAY*	State override for resumption context

Table 48: RESUME Parameters

Response: Resumption receipt with current workflow status. Idempotent: No. Error codes: 404, 408, 409.

8.8. RUN

Purpose: Execute a named, registered procedure or automation script. Implementations *MUST NOT* accept free-form execution strings; the procedure *MUST* be identified by a registered procedure_id.

Use case: An agent runs a registered data quality validation procedure against a freshly imported dataset, passing the dataset ID as input and receiving a structured pass/fail report.

Parameter	Required	Description
procedure_id	*MUST*	Registered identifier of the procedure to run
input	*MAY*	Input parameters for the procedure
timeout	*MAY*	Maximum execution time in seconds

Table 49: RUN Parameters

Security note: RUN *MUST* require activation:run in the agent's Authority-Scope. Free-form execution strings *MUST NOT* be accepted under any circumstances. Idempotent: No. Error codes: 403, 404, 408, 422, 451.

8.9. CHECK

Purpose: Query the status or health of an agent, resource, workflow, or external dependency. Returns a structured status response.

Use case: An agent checks the availability of all three downstream payment processors before initiating a purchase, using depth: shallow to minimize latency.

Parameter	Required	Description
target	*MUST*	Agent-ID, resource URI, or system identifier
depth	*MAY*	shallow (reachability only) or deep (full dependency check)

Table 50: CHECK Parameters

Response: Status document with health indicators per component.
Idempotent: Yes. Error codes: 404, 408.

9. Method Summary

Method	Category	State-Modifying	Idempotent	Key Constraints
FETCH	Acquire	No	Yes	
SEARCH	Acquire	No	Yes	
SCAN	Acquire	No	Yes	
PULL	Acquire	Yes	No	Consumes items
FIND	Acquire	No	Yes	
ANALYZE	Acquire	No	Yes	
EXTRACT	Compute	No	Yes	
FILTER	Compute	No	Yes	
VALIDATE	Compute	No	Yes	
TRANSFORM	Compute	No	Yes	
TRANSLATE	Compute	No	Yes	
NORMALIZE	Compute	No	Yes	
PREDICT	Compute	No	Yes	
RANK	Compute	No	Yes	

CLASSIFY	Compute	No	Yes	
CALCULATE	Compute	No	Yes	
EVALUATE	Compute	No	Yes	
GENERATE	Compute	No	Yes	code output requires review before RUN
RECOMMEND	Compute	No	Yes	
QUOTE	Transact	No	Yes	No execution
REGISTER	Transact	Yes	No	Use idempotency_key
SUBMIT	Transact	Yes	No	
AUTHORIZE	Transact	Yes	No	Requires authorization:grant; scope subset only
CANCEL	Transact	Yes	No	
TRANSFER	Transact	Yes	No	Requires scope
PURCHASE	Transact	Yes	No	Requires payments:purchase
SIGN	Transact	Yes	No	
LOG	Transact	Yes	No	
PUBLISH	Transact	Yes	No	
MERGE	Integrate	Yes	No	
LINK	Integrate	Yes	No	
SYNC	Integrate	Yes	No	
IMPORT	Integrate	Yes	No	
MAP	Integrate	No	Yes	Produces mapping definition; does not convert data
CONNECT	Integrate	Yes	No	

EMBED	Integrate	Yes	No	
ALERT	Communicate	No	No	Acknowledgment tracking required
BROADCAST	Communicate	No	No	
REPLY	Communicate	No	No	
SEND	Communicate	No	No	
REPORT	Communicate	Yes	No	
CHAIN	Orchestrate	Yes	No	
BATCH	Orchestrate	Yes	Conditional	
MONITOR	Orchestrate	Yes	No	
ROUTE	Orchestrate	Yes	No	
RETRY	Orchestrate	Yes	Conditional	
PAUSE	Orchestrate	Yes	No	
RESUME	Orchestrate	Yes	No	
RUN	Orchestrate	Yes	No	No free-form strings
CHECK	Orchestrate	No	Yes	

Table 51: Tier 2 Method Summary

10. Security Considerations

10.1. PURCHASE Authorization

The PURCHASE method carries financial consequences and *MUST* be subject to strict scope enforcement. Implementations *MUST* reject PURCHASE requests that do not carry payments:purchase in the Authority-Scope header with 451 Scope Violation. Budget-Limit validation *MUST* occur before execution; 452 Budget Exceeded *MUST* be returned if the purchase amount would exceed the declared budget.

10.2. AUTHORIZE Scope Constraint

The AUTHORIZE method grants permissions to other entities and is subject to the same anti-laundering constraint that applies to DELEGATE in the Tier 1 core: the scope granted by an AUTHORIZE call *MUST NOT* exceed the authorizing agent's own Authority-Scope. Any implementation receiving an AUTHORIZE request where the grant scope exceeds the authorizing agent's declared scope *MUST* return 451 Scope Violation and *MUST* log the event.

10.3. RUN Method Safety

The RUN method is the highest-risk Tier 2 method. Implementations *MUST* maintain a registry of permitted procedure_id values and *MUST NOT* execute procedures not in that registry. Free-form execution strings *MUST* be rejected. Each RUN invocation *MUST* be logged with the full procedure_id, input parameters, and executing Agent-ID.

10.4. GENERATE Code Output Safety

When GENERATE is invoked with output_type: code, the resulting code *MUST NOT* be passed directly to RUN without human or governance review. Implementations providing a pipeline from GENERATE to RUN *MUST* insert an explicit VALIDATE or approval step between them.

10.5. MONITOR Callback Verification

The callback parameter in MONITOR specifies an AGTP endpoint to receive updates. Implementations *MUST* verify that the callback endpoint is reachable and that the requesting agent has authority to receive notifications at that endpoint before establishing a monitoring subscription. Unverified callbacks are a potential exfiltration vector.

10.6. TRANSFER and Ownership Chain Integrity

TRANSFER operations modify resource ownership and create audit obligations. Implementations *MUST* record the complete ownership chain for each transferred resource in the governance audit trail. The Attribution-Record for TRANSFER *MUST* include both from_principal and to_principal.

11. IANA Considerations

This document requests registration of the following methods in the IANA AGTP Method Registry established by [AGTP] Section 9.2:

Method	Category	Status	Description
FETCH	Acquire	Permanent	Retrieve a known resource by identifier from an AGTP-native address
SEARCH	Acquire	Permanent	Execute a structured query against a data source using field filters and sort criteria
SCAN	Acquire	Permanent	Iterate over a collection for completeness; prioritizes coverage over relevance
PULL	Acquire	Permanent	Consume pending items from a queue or stream; agent takes ownership of pulled items
FIND	Acquire	Permanent	Locate agents or entities matching criteria in a registry or namespace
ANALYZE	Acquire	Permanent	Detect patterns, anomalies, or trends in a dataset without modifying it
EXTRACT	Compute	Permanent	Pull structured fields from unstructured or semi-structured source content
FILTER	Compute	Permanent	Return only records from a dataset that satisfy declared criteria
VALIDATE	Compute	Permanent	Check data or a proposed action against a schema or rule set; returns pass/fail
TRANSFORM	Compute	Permanent	Convert a specific payload from one format or schema to another

TRANSLATE	Compute	Permanent	Convert content between human languages
NORMALIZE	Compute	Permanent	Standardize data values to canonical form: dates, phones, addresses, currency
PREDICT	Compute	Permanent	Apply a model to input data and return a probability estimate or forecast
RANK	Compute	Permanent	Score and order a provided list of items by a declared criterion
CLASSIFY	Compute	Permanent	Assign categorical labels to items from a known classification scheme
CALCULATE	Compute	Permanent	Perform deterministic numeric, logical, or financial computation on structured inputs
EVALUATE	Compute	Permanent	Score a target against a weighted rubric; returns graded assessment with rationale
GENERATE	Compute	Permanent	Produce new text, structured data, or code from a source specification or prompt
RECOMMEND	Compute	Permanent	Generate a ranked candidate set tailored to a declared context and goal
QUOTE	Transact	Permanent	Return a cost estimate for a proposed method call without executing it
REGISTER	Transact	Permanent	Create a new record, entity, or subscription

			in a target system
SUBMIT	Transact	Permanent	Deliver a document or work item to a processing system or queue
AUTHORIZE	Transact	Permanent	Issue an active permission grant; scope granted must not exceed authorizing agent's scope
CANCEL	Transact	Permanent	Revoke or reverse a specific prior transaction, booking, or reservation
TRANSFER	Transact	Permanent	Move ownership or custody of a resource between principals
PURCHASE	Transact	Permanent	Execute a financial transaction; requires payments:purchase in Authority-Scope
SIGN	Transact	Permanent	Apply a cryptographic signature to a document or artifact
LOG	Transact	Permanent	Write a structured record to an audit trail for governance or compliance purposes
PUBLISH	Transact	Permanent	Make content available to a defined audience or channel
MERGE	Integrate	Permanent	Combine two or more datasets into a unified result with declared conflict resolution
LINK	Integrate	Permanent	Create a persistent association between two specific entity instances
SYNC	Integrate	Permanent	Reconcile local and

			remote resource state bidirectionally or unidirectionally
IMPORT	Integrate	Permanent	Ingest data from a non-AGTP-native external source into the agent's context
MAP	Integrate	Permanent	Define a reusable structural correspondence between two schemas or data models
CONNECT	Integrate	Permanent	Establish an active communication channel or data pathway between two systems
EMBED	Integrate	Permanent	Insert a component or capability into another system as a named sub-component
ALERT	Communicate	Permanent	Send a high-priority, time-sensitive message requiring acknowledgment
BROADCAST	Communicate	Permanent	Disseminate a message simultaneously to a group or zone without expecting replies
REPLY	Communicate	Permanent	Provide a traceable response to a prior NOTIFY, ESCALATE, or QUERY
SEND	Communicate	Permanent	Deliver a message through an external channel: email, webhook, SMS, or push
REPORT	Communicate	Permanent	Generate and deliver a structured summary document to a principal or system

CHAIN	Orchestrate	Permanent	Execute a sequence of method calls with output-to-input dependency between steps
BATCH	Orchestrate	Permanent	Execute multiple independent method calls concurrently to reduce round-trip overhead
MONITOR	Orchestrate	Permanent	Establish ongoing observation of a resource or condition; delivers updates via NOTIFY
ROUTE	Orchestrate	Permanent	Direct a task or message to the appropriate handler based on declared routing criteria
RETRY	Orchestrate	Permanent	Re-attempt a failed invocation using its original Task-ID for idempotency checking
PAUSE	Orchestrate	Permanent	Temporarily halt a scheduled or repeating workflow without terminating it
RESUME	Orchestrate	Permanent	Restart a paused workflow or suspended session using a resumption nonce
RUN	Orchestrate	Permanent	Execute a registered procedure by ID; free-form execution strings are prohibited
CHECK	Orchestrate	Permanent	Query the health or availability of an agent, resource, or external dependency

Table 52: Tier 2 Method Registry Entries

12. References

12.1. Normative References

[AGTP] Hood, C., "Agent Transfer Protocol (AGTP)", Work in Progress, Internet-Draft, draft-hood-independent-agtp-02, 2026, <https://datatracker.ietf.org/doc/html/draft-hood-independent-agtp-02>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

12.2. Informative References

[AGENTIC-API] Hood, C., "AgenticAPI: A Task-Centric Framework for Scalable Agent Integrations", 2025, <https://agenticapi.io>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/rfc/rfc6749>.

Author’s Address

Chris Hood
Nomotic, Inc.
Email: chris@nomotic.ai
URI: https://nomotic.ai