TECHNICAL REFERENCE · VERSION 1.5

Infrastructure Architecture Overview

Request routing, usage settlement, crypto payment verification and task ledger design.

> Chapters 1–3 describe the model gateway and checkout flows. Chapter 4 describes the agent liquidity framework. The workspace displays current service availability.
>
> The workspace separates API credits from pool positions. A pool position retains the agreed terms, invoice, settlement reference and final payable. A payout request uses the network of the funded position.

Contents

System boundaries and lifecycle 02

Request validation and metering 03

Crypto invoice reconciliation 04

Agent records and operational handoffs 05

CHAPTER 01

System boundaries and lifecycle

Aether AI manages customer access, billing and request validation. The inference provider processes approved model calls. A separate payment service reports on-chain transfers for API credit invoices. Access to the upstream provider remains behind the gateway.

Resource and margins flow

Source
Procure eligible model access and API capacity.

→

Allocate
Publish capacity and authorize task budgets.

→

Execute
Agents use approved models and vendors.

→

Reconcile
Match cash and costs before eligible allocations.

Component	Inputs	Outputs
Workspace	Sign-in session and selected model or invoice option.	Account controls, supported models and activity.
Model gateway	Aether bearer key or workspace session, text and model ID.	Text completion with metering, or structured error.
Inference ledger	Available credit, reservation and measured usage.	Captured charge, released reservation and current balance.
Payment adapter	Invoice parameters and signed processor notifications.	Matched payment status and account credit.
Task ledger framework	Task authorization, commitments, costs and customer receipts.	Task margin with evidence and eligible allocation record.

CHAPTER 02

Request validation and metering

Clients list text models with GET /api/v1/models and call POST /api/v1/chat/completions. The workspace saves chats and accepts up to three files of 2 MB each; compatible models can receive images. Catalog listings and prices refresh daily at 03:10 UTC.

Request flow

1. Resolve the Aether model ID to its approved upstream model. Validate the message list and requested output limit.
2. Estimate maximum usage cost and create an atomic reservation. This protects the same account balance from overlapping requests.
3. Send the request to the upstream provider. Validate the assistant text and token counters before capturing a charge.
4. Settle the measured charge and release unspent credit, or release the reservation on failure.

```
POST /api/v1/chat/completions
Authorization: Bearer $AETHER_API_KEY
Content-Type: application/json

{"model":"MODEL_ID_FROM_CATALOG","messages":[{"role":"user","content":"Summarize this
task."}],"max_tokens":128}
```

Requests accept up to 20 messages and a 10,000-byte body. The gateway returns a single JSON response. Usage records include response identifiers and measured token counts for reconciliation. See the API contract for current fields and errors.

CHAPTER 03

Crypto invoice reconciliation

API Access creates an invoice with one receiving address, an exact amount, a network, an asset and an expiry. Native-asset quotes use time-limited pricing. Customers should confirm the net amount delivered to the recipient, since wallets may subtract fees.

Payment processing sequence

1. Read the receiving address from the payment-route configuration and bind the network, amount and purpose to the invoice and account.
2. Verify the processor notification signature and match the receiving address, chain, asset and exact amount.
3. Wait for the confirmation threshold. Send eligible mismatches to review.
4. Use chain, transaction ID and transfer position to prevent duplicate credits. Post the account credit and mark the invoice credited in one ledger transaction.

Exception	Handling
Processor callback is repeated	An already credited transfer position does not generate another credit.
Wallet sends the wrong token or chain	The transfer may fail to match. Recovery is route dependent.
Received amount differs	The confirmed transfer can require review. A user should preserve the invoice and transaction IDs.
Native quote expired	Timing is checked against the invoice. Late matched transfers can require review.

See the payment status guide for customer-facing troubleshooting.

CHAPTER 04

Agent records and operational handoffs

The proposed pool ledger uses a task identifier from authorization through settlement. Each commitment and its release or expense use an operation reference so a reviewer can account for open and closed spending without counting the same movement twice.

Record	Required linkage
Authorization	Task scope, approved amount, currency, deadline, spending policy and approver.
Commitment	Task and operation IDs, amount, vendor, hold status and expiry.
Expense	Operation, receipt, service, amount, currency and recognized timestamp.
Collection	Task, customer invoice, cleared amount, payment reference and clearance timestamp.
Adjustment	Original entry, reason, effective date, amount and reviewer.

How operators handle an incomplete workflow

Image and video jobs use a separate queue: a quote precedes an API-credit reservation. Status and results persist in the workspace. A confirmed rejection releases credit; an ambiguous submission enters reconciliation without an automatic generation retry.

Keep open commitments visible until they are settled or safely released. When provider metering is unavailable, reconcile against the available receipt evidence before closing. When customer funds have yet to clear, keep the unpaid invoice separate from realized task margin.

See the task ledger guide and the settlement calculation.