



jilsovereign.com HTTP API

Operated by JIL Sovereign Technologies, Inc. (Delaware). Phone +1 (855) 572-9636. Measured 2026-10-21.

The machine-readable document is <https://jilsovereign.com/openapi/jilsovereign-api.json>. This PDF is <https://jilsovereign.com/openapi/jilsovereign-api.pdf>. Both describe <https://api.jilsovereign.com>. This website is not that host.

What was measured

GET / returned HTTP 200 and the root document below. GET /health and GET /ready each returned HTTP 200 and {"status":"ok","service":"api-gateway"}. GET /docs returned HTTP 200, text/html, 6211 bytes, title JIL Sovereign API Reference. That page contains no OpenAPI document. Its hrefs are <https://developers.jilsovereign.com> and /.

```
{
  "service": "JIL Sovereign API Gateway",
  "base_url": "https://api.jilsovereign.com",
  "docs": "https://api.jilsovereign.com/docs",
  "catalog": "https://api.jilsovereign.com/v1/catalog",
  "console": "https://developers.jilsovereign.com",
  "auth": {
    "scheme": "hybrid-pq-signature",
    "algorithms": [
      "ed25519",
      "ml-dsa-65"
    ],
    "headers": [
      "X-API-Key",
      "X-API-Timestamp",
      "X-API-Nonce",
      "X-API-Signature",
      "X-API-PQ-Signature"
    ],
    "challenge": "METHOD\\nORIGINAL_URL\\nSHA256(body)\\nTIMESTAMP\\nNONCE",
    "enforced": true
  },
  "apis": [
    "asset-intelligence",
    "igaming",
    "kyc",
    "payment-integrity",
    "attestyx",
    "metering",
    "courtchain",
    "ava"
  ]
}
```

Scheme hybrid-pq-signature. Algorithms ed25519, ml-dsa-65. Headers X-API-Key, X-API-Timestamp, X-API-Nonce, X-API-Signature, X-API-PQ-Signature. The challenge is METHOD, ORIGINAL_URL, SHA256(body), TIMESTAMP, and NONCE, joined by newlines. Enforced is true. No sample signature and no key are printed.

GET /v1/catalog returned HTTP 200, 5474 bytes, and the eleven topics in this document. GET /v1/catalog/asset-intelligence returned HTTP 200. Its plan objects use monthlyQuota and priceEnv. priceEnv names an environment variable. The name is not printed. The list above uses quota and a blurb. No price is printed. GET /v1/catalog/areb, GET /v1/catalog/pieb, and GET /v1/catalog/ava each returned HTTP 404 and {"error":"unknown topic"}.



The catalog text says self-serve topics use POST /v1/subscribe and Stripe, and that a credential is issued on payment. No subscribe request was sent. No price is printed.

Every catalog path under /openapi/ returned HTTP 404, application/json, 92 bytes: {"error": "unknown endpoint", "hint": "GET https://api.jilsovereign.com/ lists available APIs"}. The YAML files linked from this website are the repository specifications. The API host does not serve them.

The repository catalog names areb and pieb. The live catalog response did not include them, so they are not listed as available. The root apis array includes ava. ava is not a catalog topic. This document is not a public L1 JSON-RPC API. developers.jilsovereign.com is the console and does not serve this file.

Live catalog

Topic	Access	Live endpoints	Live summary
asset-intelligence	self-serve	POST /v1/intel/inquiry	On-chain asset discovery, screening, tracing, clustering, and court-ready evidence for a single subject.
igaming	self-serve	POST /v1/igaming/pre-payment	Inline deposit/withdrawal wallet decision: screening + source-of-funds + cluster risk before a payment.
kyc	self-serve	POST /v1/kyc/verify; POST /v1/kyb/verify; POST /v1/kyc/tier; POST /v1/kyc/sanctions; POST /v1/kyc/bec; POST /v1/kyc/bid; POST /v1/kyc/ubo; POST /v1/payments/pre-screen	Individual + business verification, sanctions/PEP, BEC/BID, UBO graph, payment pre-screen.
payment-integrity	request-access	POST /v1/payment-integrity/attestations; POST /v1/payment-integrity/reputation/check; POST /v1/payment-integrity/bad-actors; POST /v1/payment-integrity/creb	Retroactive payment-integrity attestations, reputation checks, bad-actor registry, CREB evidence bundles.
attestyx	request-access	POST /v1/attestyx/*	Institutional attestation flow: engagement-scoped substantiation and CREB-grade evidence.
insureaudit	request-access	none in the live catalog	MCO retroactive payment-integrity engagements (network, premise-mismatch, litigation-path).
pre-clearance	request-access	none in the live catalog	Inter-custody settlement attestation for institutional digital-asset custodians.
federal-grants	request-access	none in the live catalog	Grant-program integrity engines (EB-5, H-1B, WIE, WC) with statutory-cited verdicts.
phantom-employee	request-access	none in the live catalog	MFCU-facing phantom-employee typologies with FRE 902(13) self-authenticating CREB output.
metering	request-access	POST /v1/metering/consume; POST /v1/metering/cases; POST /v1/metering/cases/{id}/agree; POST /v1/metering/cases/{id}/settle; GET /v1/metering/tracker/{clientId}	Report-block consumption + CREB contingency case lifecycle (bill on agreed lawsuit + on settlement). Reported PII-free from the customer SPCS.
courtchain	request-access	POST /v1/courtchain/anchor; GET /v1/courtchain/anchor/{id}	Store the cryptographic CREB record in JIL: anchor a bundle hash to the L1 and receive a tamper-evident receipt. Data stays in-tenant; only the hash is anchored.

Asset Intelligence

Live catalog id asset-intelligence. Status available. Access self-serve. Category Asset Intelligence.

Live catalog summary: On-chain asset discovery, screening, tracing, clustering, and court-ready evidence for a single subject.

Live catalog endpoints: POST /v1/intel/inquiry.

Live catalog plans, quotas only: Standard 100k calls/mo (quota 100000), Scale 1M calls/mo (quota 1000000), Enterprise Unlimited + priority (quota -1). No price is printed.

Repository specification: JIL Sovereign Asset-Intelligence API version 2026.06.08. The API host path /openapi/asset-intelligence.yaml returned HTTP 404 on 2026-10-21. The file on this website is https://jilsovereign.com/openapi/asset-intelligence.yaml.

Specification text: The single licensed Asset-Intelligence front door. One endpoint answers every inquiry; the `type` field selects which engine runs: - screen -- sanctions / bad-actor / known-service verdict for an address (the Chainalysis/TRM



substitute) - prepayment -- synchronous allow/review/block wallet decision (also exposed at /v1/igaming/pre-payment) - trace -- on-chain counterparty trace + subpoena-target attribution - discover -- multi-source asset discovery for a named subject (on-chain, corporate, real-estate, court, etc.) - cluster -- deterministic wallet clustering (common-control heuristics, Daubert-survivable) - creb -- CREB(R) court-ready evidence bundle (enterprise/platform plans; proxied to ...

POST /v1/intel/inquiry

Specification summary: Run one asset-intelligence inquiry (type selects the engine)

Specification text: The request body MUST carry a `type`. Remaining properties are the payload for that inquiry type. Plan entitlement for `type` is enforced before dispatch; a plan that lacks the type returns 403.

Specification example, request, screen: Screen a single address.

```
{
  "type": "screen",
  "address": "0xabc...",
  "chainId": 1
}
```

Specification example, request, prepayment: Pre-payment wallet decision.

```
{
  "type": "prepayment",
  "event": "deposit",
  "chain": "ETH",
  "address": "0xabc...",
  "amountUsd": 5000
}
```

Specification example, request, trace: Trace counterparties.

```
{
  "type": "trace",
  "address": "0xabc...",
  "chainId": 1,
  "depth": 2,
  "direction": "both"
}
```

Specification example, request, discover: Discover assets for a named subject.

```
{
  "type": "discover",
  "subjectId": "subj-001",
  "subjectName": "Jane Roe",
  "jurisdictions": [
    "US-DE"
  ]
}
```

Specification example, request, cluster: Cluster addresses under common control.

```
{
  "type": "cluster",
  "chain": "BTC",
  "address": "bc1q...",
  "maxAddresses": 500
}
```

Specification example, request, creb: Generate a CREB bundle (enterprise).

```
{
  "type": "creb",
  "planId": "plan-001",
  "clusterId": "cl-001"
}
```

Specification example, response, HTTP 200, screen.

```
{
  "inquiry": "screen",
  "licensee": "Acme Investigations LLC",
  "result": {
    "address": "0xabc...",

```



```

"status": "KNOWN_SERVICE",
"risk": "low",
"owner": "Coinbase",
"category": "exchange",
"sanctioned": false,
"badActor": false,
"custodian": true,
"sources": [
  "jil-seed",
  "ofac"
],
"reasons": [
  "matched known-service label"
]
}

```

Specification example, response, HTTP 401, badKey.

```

{
  "error": "invalid or missing API key",
  "header": "X-API-Key"
}

```

Specification example, response, HTTP 403, planScope.

```

{
  "error": "plan 'screening' does not include 'trace' inquiries",
  "allowed": [
    "screen",
    "prepayment"
  ]
}

```

Specification example, response, HTTP 403, expired.

```

{
  "error": "license expired",
  "expiredAt": "2026-01-01T00:00:00Z"
}

```

Specification example, response, HTTP 429, quota.

```

{
  "error": "monthly quota exceeded",
  "quota": 50000,
  "used": 50000
}

```

Specification responses: 200: Inquiry result. The `result` shape depends on the inquiry `type`. 400: Bad request -- unknown/missing inquiry type or invalid payload 401: Invalid or missing API key / signature 403: License suspended/expired, or plan does not include this inquiry type 429: Monthly quota exceeded

iGaming Pre-Payment Integrity

Live catalog id igaming. Status available. Access self-serve. Category iGaming.

Live catalog summary: Inline deposit/withdrawal wallet decision: screening + source-of-funds + cluster risk before a payment.

Live catalog endpoints: POST /v1/igaming/pre-payment.

Live catalog plans, quotas only: Standard 100k calls/mo (quota 100000), Scale 1M calls/mo (quota 1000000), Enterprise Unlimited + priority (quota -1). No price is printed.

Repository specification: JIL Sovereign iGaming Pre-Payment API version 2026.06.08. The API host path /openapi/igaming.yaml returned HTTP 404 on 2026-10-21. The file on this website is <https://jilsovereign.com/openapi/igaming.yaml>.

Specification text: One synchronous allow/review/block wallet decision an iGaming operator calls inline before a deposit or withdrawal. The verdict combines a sanctions / bad-actor screen, source-of-funds tracing (first funder, mixer-funded), and deterministic multi-account clustering (which of the operator's other player wallets share common control with this wallet). Requires a license whose plan includes the `prepayment` inquiry type and a hybrid-PQ signature (see api-gateway.yaml).



Usage is metered per call.

POST /v1/igaming/pre-payment

Specification summary: Synchronous pre-payment wallet decision (ALLOW / REVIEW / BLOCK)

Specification text: Call inline before settling a deposit/withdrawal. `address` and `chain` are required. `chain` accepts ETH or BTC (SOL is not supported for this decision). Supply `knownPlayerAddresses` to run the multi-accounting check against the operator's other player wallets.

Specification example, request, deposit: Deposit decision with multi-accounting check.

```
{
  "event": "deposit",
  "playerId": "player-7781",
  "chain": "ETH",
  "address": "0xabc...",
  "amountUsd": 5000,
  "knownPlayerAddresses": [
    "0xdef...",
    "0x123..."
  ],
  "engagementId": "eng-2026-09",
  "permissiblePurpose": "AML/KYC pre-transaction screening"
}
```

Specification example, request, withdrawal: Minimal withdrawal decision.

```
{
  "event": "withdrawal",
  "chain": "BTC",
  "address": "bc1q..."
}
```

Specification example, response, HTTP 200, review.

```
{
  "decision": "REVIEW",
  "risk": "medium",
  "reasons": [
    "funded via mixer-adjacent address",
    "2 linked player accounts"
  ],
  "sanctions": "CLEAR",
  "clusterAccounts": 3,
  "linkedPlayerAccounts": [
    "0xdef...",
    "0x123..."
  ],
  "sourceOfFunds": {
    "firstFunder": "0x999...",
    "funderLabel": "unknown EOA",
    "mixerFunded": true
  },
  "event": "deposit",
  "playerId": "player-7781",
  "address": "0xabc...",
  "chain": "ETH",
  "generatedAt": "2026-06-08T12:00:00Z",
  "latencyMs": 412,
  "warnings": []
}
```

Specification example, response, HTTP 400, noAddress.

```
{
  "error": "address required"
}
```

Specification example, response, HTTP 400, noChain.

```
{
  "error": "chain required (ETH or BTC)"
}
```



Specification example, response, HTTP 401, badKey.

```
{
  "error": "invalid or missing API key",
  "header": "X-API-Key"
}
```

Specification example, response, HTTP 403, planScope.

```
{
  "error": "plan 'investigations' does not include 'prepayment' inquiries",
  "allowed": [
    "screen",
    "trace",
    "discover",
    "cluster"
  ]
}
```

Specification example, response, HTTP 403, expired.

```
{
  "error": "license expired",
  "expiredAt": "2026-01-01T00:00:00Z"
}
```

Specification example, response, HTTP 429, quota.

```
{
  "error": "monthly quota exceeded",
  "quota": 1000000,
  "used": 1000000
}
```

Specification responses: 200: Decision returned 400: Bad request -- missing address/chain or invalid payload 401: Invalid or missing API key / signature 403: License suspended/expired, or plan does not include the prepayment inquiry type 429: Monthly quota exceeded

KYC / KYB

Live catalog id kyc. Status available. Access self-serve. Category Identity.

Live catalog summary: Individual + business verification, sanctions/PEP, BEC/BID, UBO graph, payment pre-screen.

Live catalog endpoints: POST /v1/kyc/verify; POST /v1/kyb/verify; POST /v1/kyc/tier; POST /v1/kyc/sanctions; POST /v1/kyc/bec; POST /v1/kyc/bid; POST /v1/kyc/ubo; POST /v1/payments/pre-screen.

Live catalog plans, quotas only: Standard 100k calls/mo (quota 100000), Scale 1M calls/mo (quota 1000000), Enterprise Unlimited + priority (quota -1). No price is printed.

Repository specification: JIL Sovereign KYC / KYB Verification version 2026.06.08. The API host path /openapi/kyc.yaml returned HTTP 404 on 2026-10-21. The file on this website is <https://jilsovereign.com/openapi/kyc.yaml>.

Specification text: Identity (KYC), business (KYB), sanctions/PEP, Business Email Compromise (BEC), Business Identity Document (BID), Ultimate Beneficial Owner (UBO), and payment pre-screen surfaces, fronted by the kyc-service. Verdicts carry a combined risk score 0-100 and a decision (auto-approve / auto-hold / auto-reject). Risk-based tiering runs the free in-house checks first and returns the MINIMUM verification tier required, so the paid doc+biometric vendor is only invoked for flagged or high-value subjects. This document covers the PUBLIC gateway surface only; analyst case-review, biometric, batch, and registration admin paths are excluded. Every operation requires a subscription credential and a ...

POST /v1/kyc/verify

Specification summary: Individual KYC verification (sanctions screen + risk score + decision)

Specification responses: 200: KYC verdict

Specification field examples, not a full request: flags = ["SANCTIONS_HIT"].

POST /v1/kyb/verify

Specification summary: Business KYB verification (BEC + BID + sanctions -> combined risk)

Specification responses: 200: KYB verdict



No published request or response example.

POST /v1/kyc/tier

Specification summary: Risk-based verification tiering (returns the minimum tier required)

Specification text: Runs the free in-house checks (sanctions/PEP + email/domain BEC signals), scores risk 0-100, and decides the minimum verification tier so the paid doc+biometric vendor is only triggered for flagged or high-value subjects. Tier 0 = clear (in-house sufficient), Tier 1 = authoritative database match, Tier 2 = gov-ID document auth + liveness selfie. A sanctions hit returns action `reject`. At least one of `fullName` or `email` is required.

Specification responses: 200: Tier decision

No published request or response example.

POST /v1/kyc/sanctions

Specification summary: Sanctions / PEP screening (Yente / OpenSanctions with OFAC fallback)

Specification responses: 200: Screening result

Specification field examples, not a full request: source = "yente".

POST /v1/kyc/bec

Specification summary: Business Email Compromise / vendor-impersonation check

Specification text: Combines RDAP domain age + MX/SPF/DMARC + lookalike detection + disposable-email blocklist into a BEC risk score.

Specification responses: 200: BEC risk score + components

No published request or response example.

POST /v1/kyc/bid

Specification summary: Business Identity Document validation (registry + LEI + jurisdiction)

Specification responses: 200: BID validation result

No published request or response example.

POST /v1/payments/pre-screen

Specification summary: Pre-screen a payment instruction (IBAN/BIC validity + corridor risk)

Specification text: Validates the destination IBAN (MOD-97) and SWIFT BIC (ISO 9362), assesses the cross-border corridor risk, and returns an action.

Specification responses: 200: Pre-screen verdict

No published request or response example.

Payment Integrity

Live catalog id payment-integrity. Status available. Access request-access. Category Healthcare / Payment Integrity.

Live catalog summary: Retroactive payment-integrity attestations, reputation checks, bad-actor registry, CREB evidence bundles.

Live catalog endpoints: POST /v1/payment-integrity/attestations; POST /v1/payment-integrity/reputation/check; POST /v1/payment-integrity/bad-actors; POST /v1/payment-integrity/creb.

Repository specification: JIL Sovereign Payment Integrity Attestation API version 2026.06.08. The API host path /openapi/payment-integrity.yaml returned HTTP 404 on 2026-10-21. The file on this website is <https://jilsovereign.com/openapi/payment-integrity.yaml>.

Specification text: Real-time, pre-settlement trust layer that sits between a merchant platform and its payment processors, payout providers, and settlement workflows. Every payment, payout, refund, or account event receives a deterministic Pass / Review / Block / Escalate attestation with a risk score, trust score, reason codes, a sanctions result (OFAC / EU / UK / UN screening), and a CREB-eligible evidence record. The JIL token is the settlement and metering rail only (configurable bps per attestation); no custody, no processing. The public gateway exposes this service under the `v1/payment-integrity/`



namespace (the namespace prefix is stripped before reaching the backend). Requests require a subscription ...

The repository catalog summary for this topic is a merchant decision. The live response on this date used the shorter summary above. This document follows the live response.

POST /v1/payment-integrity/attestations

Specification summary: Create a pre-settlement attestation (deterministic PASS / REVIEW / BLOCK / ESCALATE verdict)

Specification text: Scores a payment, payout, refund, or account event before money moves. Runs sanctions screening, reputation-graph lookup, bad-actor matching, device/IP intelligence, and velocity in parallel, then returns a sealed deterministic verdict. Idempotent on the `Idempotency-Key` header (a replay returns the original response with `X-Idempotent-Replay: true`).

Specification example, request, trusted_purchase: Established account, US card, clean history.

```
{
  "event_type": "PAYMENT_PURCHASE",
  "merchant_id": "epic-games",
  "external_transaction_id": "txn_10293",
  "user": {
    "user_id": "u_good",
    "account_age_days": 900,
    "prior_chargeback_count": 0,
    "prior_purchase_count": 50
  },
  "payment": {
    "amount": 19.99,
    "currency": "USD",
    "payment_method_type": "CARD",
    "card_country": "US"
  },
  "network": {
    "ip_address": "203.0.113.10",
    "country": "US"
  },
  "context": {
    "channel": "WEB"
  }
}
```

Specification example, request, sanctioned_wallet: Crypto payout to an OFAC-listed address.

```
{
  "event_type": "WALLET_PAYOUT",
  "merchant_id": "epic-games",
  "external_transaction_id": "txn_55012",
  "payment": {
    "amount": 5000,
    "currency": "USD",
    "payment_method_type": "CRYPTO_WALLET"
  },
  "wallet": {
    "wallet_address": "0x8589427373d6d84e98730d7795d8f6f8731fda16",
    "blockchain": "Ethereum"
  },
  "network": {
    "ip_address": "203.0.113.77",
    "country": "US"
  },
  "context": {
    "channel": "API"
  }
}
```

Specification responses: 201: Attestation created

POST /v1/payment-integrity/reputation/check

Specification summary: Look up the trust / risk reputation of one or more entities



Specification text: Returns the current reputation projection (reputation score, risk score, status, reason codes, last-seen) for each requested entity. Entities are users, devices, IPs, wallets, payment methods, creators, or hashed contact identifiers.

Specification example, request, two_entities.

```
{
  "entities": [
    {
      "entity_type": "USER",
      "entity_id": "u_bad"
    },
    {
      "entity_type": "WALLET",
      "entity_id": "0x8589427373d6d84e98730d7795d8f6f8731fda16"
    }
  ]
}
```

Specification responses: 200: Reputation results

POST /v1/payment-integrity/bad-actors

Specification summary: Record (upsert) a known bad actor in the reputation graph

Specification text: Adds or updates a bad-actor record keyed by entity type and id. The record feeds future attestations as a hard signal. Returns CREATED or UPDATED.

Specification example, request, confirmed_fraud.

```
{
  "entity_type": "WALLET",
  "entity_id": "0x8589427373d6d84e98730d7795d8f6f8731fda16",
  "reason_code": "CONFIRMED_FRAUD",
  "severity": "CRITICAL",
  "source": "chargeback-ops"
}
```

Specification responses: 201: Bad actor recorded

POST /v1/payment-integrity/creb

Specification summary: Assemble a Court Ready Evidence Bundle (CREB) for a payment-integrity case

Specification text: Queues a CREB for a confirmed-fraud, chargeback, sanctions, refund-abuse, or legal-response case. The bundle assembles the attestation record, the rule-execution trace, the evidence index, and a cryptographic chain of custody, seals it with a deterministic bundle hash, and anchors it on CourtChain under FRE 902(14). Returns 202 with a creb_id; the bundle is retrievable shortly after via GET /v1/payment-integrity/creb/{id}.

Specification example, request, confirmed_fraud_crebe.

```
{
  "attestation_id": "att_3f9c1d2e4b5a6c7d8e9f01",
  "case_type": "CONFIRMED_FRAUD",
  "requested_formats": [
    "PDF",
    "JSON"
  ],
  "requester": "siu-desk"
}
```

Specification responses: 202: CREB queued

Attestyx

Live catalog id attestyx. Status available. Access request-access. Category Attestation / CREB.

Live catalog summary: Institutional attestation flow: engagement-scoped substantiation and CREB-grade evidence.

Live catalog endpoints: POST /v1/attestyx/*.

Repository specification: JIL Sovereign Attestyx Attestation API version 2026.06.08. The API host path /openapi/attestyx.yaml returned HTTP 404 on 2026-10-21. The file on this website is <https://jilsovereign.com/openapi/attestyx.yaml>.



Specification text: Attestyx is the institutional attestation + Court Ready Evidence Bundle (CREB) flow backed by the JIL attestation-service. It clears settlements with a deterministic approved / denied / hold verdict, runs the fan-out check pipeline, issues court-portable receipts (PDF / JSON / TXT / IMG), provisions per-customer engagements with frozen LOB-profile check snapshots, surfaces AVA plan receipts, and generates per-cluster CREB bundles sealed with a CourtChain anchor. The measured gateway names ed25519 and ml-dsa-65. The public gateway exposes this service under the ``v1/attestyx/`` namespace. The gateway strips only the ``v1/attestyx`` prefix, so the backend's own ``/api/v1/...`` paths are ...

POST `/v1/attestyx/api/v1/settle`

Specification summary: Clear a settlement in one call (deterministic approved / denied / hold verdict)

Specification text: The main client endpoint. Runs sender + recipient credential checks and settlement-policy evaluation in parallel, stores the attestation, and on approval issues a finality receipt and records the metering fee. Emits a signed `settlements.requested` event so the fan-out check services produce sub-verdicts that the aggregator correlates into a court-portable bundle.

Specification example, request, `cross_border`.

```
{
  "tx_id": "tx_88210",
  "sender_id": "party_us_4471",
  "sender_country": "US",
  "recipient_id": "party_de_9930",
  "recipient_country": "DE",
  "amount_usd": 125000,
  "corridor": "US-DE",
  "client_id": "acme-treasury"
}
```

Specification responses: 201: Settlement cleared

POST `/v1/attestyx/api/v1/attestations`

Specification summary: Enqueue an attestation for async pipeline processing

Specification text: Inserts a pending attestation row and emits `attestations.requested` for the per-node Kafka worker. Returns 202 with a `poll_url`. Receipts become available once the pipeline drains.

Specification responses: 202: Attestation enqueued

Specification field examples, not a full request: `status = "pending"`.

POST `/v1/attestyx/api/v1/attestations/{attestationId}/rerun`

Specification summary: Re-run a prior attestation against the delta of still-unverified required checks

Specification text: Computes `required_checks` (caller override or vendor-declared) minus the checks that already passed on the latest run, bumps `run_sequence`, and emits the delta. Rejects with 400 if the delta is empty.

Specification responses: 202: Rerun enqueued 400: Empty rerun delta (nothing to re-run).

Specification field examples, not a full request: `status = "pending"`.

POST `/v1/attestyx/api/v1/engagements`

Specification summary: Provision a per-customer engagement (returns a scoped API key once)

Specification text: Creates an engagement with a frozen snapshot of the LOB profile's T1/T2 check codes, T2 evidence packs, and T3 CREB template (engagement immutability boundary). Returns a scoped bearer API key exactly once; the server stores only its SHA-256 hash. Requires admin auth.

Specification example, request, `by_profile`.

```
{
  "client_id": "state-mfcu-tx",
  "lob_profile_id": "prof_medicaid_um",
  "engagement_label": "TX MFCU 2026 retro PI",
  "company_name": "Texas MFCU",
  "billing_contact_email": "ops@example.gov"
}
```

Specification responses: 201: Engagement provisioned 409: `engagement_id` already exists.



POST /v1/attestyx/api/v1/plans/{plan_id}/authorize

Specification summary: Authorize a subset of clusters for Tier 2 spend

Specification text: Customer approves specific clusters with a budget cap and an authorizer attestation. Requires the engagement-scoped bearer key (or admin).

Specification responses: 200: Clusters authorized

No published request or response example.

POST /v1/attestyx/api/v1/plans/{plan_id}/crebs/{cluster_id}

Specification summary: Generate a Court Ready Evidence Bundle (CREB) for one cluster

Specification text: Assembles and seals a CREB for the cluster (Merkle root, CourtChain anchor, SDV upload). Requires admin or the engagement-scoped bearer key for the plan's engagement.

Specification responses: 200: CREB generated

No published request or response example.

InsuredAudit

Live catalog id insuredaudit. Status request-access. Access request-access. Category Healthcare / Payment Integrity.

Live catalog summary: MCO retroactive payment-integrity engagements (network, premise-mismatch, litigation-path).

The live catalog lists no endpoints for this topic. The repository specification names the paths below. They are not listed as callable routes, and this document prints no request example for them.

Repository specification: JIL Sovereign InsuredAudit version 2026.06.08. The API host path /openapi/insuredaudit.yaml returned HTTP 404 on 2026-10-21. The file on this website is <https://jilsovereign.com/openapi/insuredaudit.yaml>.

Specification text: request-access (provisioned per engagement). MCO retroactive payment-integrity engagements: network analysis, premise-mismatch detection, and litigation-path decisioning over the customer's claims data. AI accelerates operations only; outcomes are determined by deterministic rules, statutory citations, and human signoff, with CREB(R) court-ready evidence on promotion. PLACEHOLDER SPEC -- InsuredAudit is not yet built. The gateway catalogs it with status `request-access` and an empty public endpoint list, and the backend `INSUREDAUDIT\_URL` route is gated/unset. The endpoints below describe the intended engagement API surface (engagement intake, network, premise-mismatch, litigation-path); ...

Method	Repository path	Specification summary
POST	/v1/insuredaudit/engagements	[PLANNED] Engagement intake (open an MCO retro payment-integrity engagement)
GET	/v1/insuredaudit/engagements/{id}/network	[PLANNED] Network explorer (entity/provider/bank-fingerprint graph for the engagement)
GET	/v1/insuredaudit/engagements/{id}/premise-mismatch	[PLANNED] Premise-mismatch findings (premise-business compatibility violations)
POST	/v1/insuredaudit/engagements/{id}/litigation-path	[PLANNED] Litigation-path decision (recommend tier/path with statutory authority)

JIL Corridor (Pre-Clearance)

Live catalog id pre-clearance. Status request-access. Access request-access. Category Financial / Settlement.

Live catalog summary: Inter-custody settlement attestation for institutional digital-asset custodians.

The live catalog lists no endpoints for this topic. The repository specification names the paths below. They are not listed as callable routes, and this document prints no request example for them.

Repository specification: JIL Corridor (Pre-Clearance) version 2026.06.08. The API host path /openapi/pre-clearance.yaml returned HTTP 404 on 2026-10-21. The file on this website is <https://jilsovereign.com/openapi/pre-clearance.yaml>.

Specification text: request-access (provisioned per engagement). Inter-custody settlement attestation for institutional digital-asset custodians (Fireblocks, Anchorage, BitGo). The asset never enters JIL: the engagement attests on a candidate transfer, the custodian decides. No custody, no transmission, no bridge. Surface grounded in three live services: - corridor-engine (:8710) -- route discovery and optimization across settlement corridors (lowest_cost, fastest, most_reliable,



compliant_first), with outcome feedback. - preclearance-cosigner (:8972) -- threshold co-signature quorum status and keygen ceremony. Threshold-share signing is scaffold/dev-mode today; real CGGMP21 + FROST + hybrid-PQ co-signing ...

Method	Repository path	Specification summary
POST	/v1/corridor/optimize	Select the best settlement route for an origin/destination/amount under a strategy
GET	/v1/corridor/list	List all corridors with status and route counts
GET	/v1/corridor/{id}	Corridor detail with all routes
GET	/v1/corridor/{id}/routes	Routes configured for a corridor
GET	/v1/corridor/stats	Corridor statistics for the last 24 hours (selections, fees, strategy/outcome mix)
POST	/v1/corridor/route/{routeId}/outcome	Record the realized outcome for a prior route selection (feeds rolling success rate + latency)
GET	/v1/cosign/quorum-status	Co-signer quorum configuration and scheme status (ECDSA / FROST / hybrid-PQ)
POST	/v1/cosign/keygen	Run a threshold keygen ceremony (dev-mode CGGMP21 today; HSM-backed n=20 in Phase 2)
POST	/v1/cosign/threshold-share/ecdsa	Request an ECDSA signature share from a validator (dev-mode deterministic ack; real CGGMP21 sign in Phase 2)
POST	/v1/cosign/threshold-share/frost	Request a FROST Ed25519 signature share (scaffold: returns 501)
POST	/v1/cosign/verify-pq-attestation	Verify a hybrid Ed25519 + ML-DSA-65 attestation (scaffold: returns 501)
GET	/v1/oob/cooling-periods	Catalog of per-asset cooling periods for out-of-band confirmation
GET	/v1/oob/cooling-period/{asset}	Cooling period for a single asset
POST	/v1/oob/initiate	Open an out-of-band confirmation request for a high-risk transfer (admin-scoped)
GET	/v1/oob/status/{request_id}	Status and ack tally for an out-of-band request
POST	/v1/oob/acknowledge	Officer acknowledges or denies a request with an Ed25519 signature over the canonical payload

Federal Grants Integrity

Live catalog id federal-grants. Status request-access. Access request-access. Category Federal / Government.

Live catalog summary: Grant-program integrity engines (EB-5, H-1B, WIE, WC) with statutory-cited verdicts.

The live catalog lists no endpoints for this topic. The repository specification names the paths below. They are not listed as callable routes, and this document prints no request example for them.

Repository specification: JIL Sovereign Federal Grants Integrity version 2026.06.08. The API host path /openapi/federal-grants.yaml returned HTTP 404 on 2026-10-21. The file on this website is <https://jilsovereign.com/openapi/federal-grants.yaml>.

Specification text: request-access (provisioned per engagement). Grant-program integrity engines with statutory-cited verdicts for federal verticals (EB-5, H-1B, WIE, WC). A foundation/operator runs programs and applications through deterministic check engines and the grants-verdict-engine, then emits court-ready CREB(R) evidence. AI accelerates operations only; outcomes are determined by deterministic rules, statutory citations, and human signoff. Surface grounded in the live grants-api (:8980), which fronts the eb5-engine, h1b-engine, wie-service, grants-engine, and grants-verdict-engine. Endpoints below are the engagement-relevant subset (programs, applications, attestations, CREB); the service also exposes ...

Method	Repository path	Specification summary
POST	/v1/grant-programs	Create a grant program (foundation-scoped)
GET	/v1/grant-programs/mine	List programs owned by the calling foundation
GET	/v1/grant-programs/public	List public (open) grant programs
GET	/v1/grant-programs/{program_id}	Get a single program
POST	/v1/applications	Create a grant application (grantee-scoped)
GET	/v1/applications/mine	List the calling grantee's applications
GET	/v1/applications/queue	Foundation review queue



Method	Repository path	Specification summary
GET	/v1/applications/{id}	Get an application
POST	/v1/applications/{id}/submit	Grantee submits an application for review
POST	/v1/applications/{id}/precheck	Run the integrity engines against an application (foundation/operator-scoped)
POST	/v1/applications/{id}/decide	Foundation records an approve/deny decision
POST	/v1/attestations	Produce a grant-integrity attestation (runs categorization, BEC, BID, bank, and the verdict engine)
GET	/v1/attestations/{id}	Get an attestation
GET	/v1/creb/{attestation_id}	Get the CREB(R) evidence bundle for an attestation
GET	/v1/creb/{attestation_id}/receipt	Get the tamper-evident CourtChain receipt for a CREB(R) bundle

Phantom Employee Detection

Live catalog id phantom-employee. Status request-access. Access request-access. Category Healthcare / MFCU.

Live catalog summary: MFCU-facing phantom-employee typologies with FRE 902(13) self-authenticating CREB output.

The live catalog lists no endpoints for this topic. The repository specification names the paths below. They are not listed as callable routes, and this document prints no request example for them.

Repository specification: JIL Sovereign Phantom Employee Detection version 2026.06.08. The API host path /openapi/phantom-employee.yaml returned HTTP 404 on 2026-10-21. The file on this website is <https://jilsovereign.com/openapi/phantom-employee.yaml>.

Specification text: request-access (provisioned per engagement). MFCU-facing vertical targeting State Medicaid Fraud Control Units and State AG offices. Five typologies (Identity, Eligibility, Service Delivery, Cost Report, Cross-Entity) emit a CREB(R) Phantom-Employee evidence bundle that is FRE 902(13) self-authenticating. AI accelerates operations only; outcomes are determined by deterministic rules, statutory citations, and human signoff. PLACEHOLDER SPEC -- no dedicated phantom-employee backend API service exists in the repository today (the vertical is marketing/product pages plus the shared attestation + CREB stack). The endpoints below describe the intended engagement API surface a customer would

...

Method	Repository path	Specification summary
POST	/v1/phantom-employee/engagements	[PLANNED] Open a phantom-employee engagement (MFCU / State AG)
POST	/v1/phantom-employee/engagements/{id}/scan	[PLANNED] Run the five-typology scan over the in-tenant dataset
GET	/v1/phantom-employee/engagements/{id}/findings	[PLANNED] List deterministic findings grouped by typology
POST	/v1/phantom-employee/engagements/{id}/creb	[PLANNED] Emit the CREB(R) Phantom-Employee evidence bundle (FRE 902(13))

Usage Metering & Billing

Live catalog id metering. Status available. Access request-access. Category Engagement / Billing.

Live catalog summary: Report-block consumption + CREB contingency case lifecycle (bill on agreed lawsuit + on settlement). Reported PII-free from the customer SPCS.

Live catalog endpoints: POST /v1/metering/consume; POST /v1/metering/cases; POST /v1/metering/cases/{id}/agree; POST /v1/metering/cases/{id}/settle; GET /v1/metering/tracker/{clientId}.

Repository specification: JIL Sovereign Usage Metering & Billing version 2026.06.08. The API host path /openapi/metering.yaml returned HTTP 404 on 2026-10-21. The file on this website is <https://jilsovereign.com/openapi/metering.yaml>.

Specification text: Engagement billing for MCO customers. Attestation compute runs in the customer's Snowflake SPCS; the in-tenant app reports PII-free usage here. Two modes: prepaid report blocks, and CREB contingency billed across a lawsuit lifecycle (a per-customer percentage on the agreed lawsuit amount and on the settlement). Requires a subscription credential scoped to `metering` and a hybrid-PQ signature (see api-gateway.yaml).

POST /v1/metering/consume

Specification summary: Consume one report from the customer's prepaid block (idempotent on artifact_sha256)



Specification responses: 201: Consumed 200: Replay (already counted) 402: No active prepaid block (exhausted)
No published request or response example.

POST /v1/metering/cases

Specification summary: Open a CREB case (links the CourtChain anchor of the cryptographic CREB record)

Specification responses: 201: Case created

No published request or response example.

POST /v1/metering/cases/{id}/agree

Specification summary: Record the agreed lawsuit amount and bill creb_pct% of it

Specification responses: 200: Billed creb_lawsuit

No published request or response example.

POST /v1/metering/cases/{id}/settle

Specification summary: Record the settlement amount and bill creb_pct% of it

Specification responses: 200: Billed creb_settlement

No published request or response example.

GET /v1/metering/tracker/{clientId}

Specification summary: Per-customer site tracker (reports purchased/used/remaining + case status + CREB fee accrual)

Specification responses: 200: OK

No published request or response example.

CourtChain (CREB Record Anchor)

Live catalog id courtchain. Status available. Access request-access. Category Attestation / CREB.

Live catalog summary: Store the cryptographic CREB record in JIL: anchor a bundle hash to the L1 and receive a tamper-evident receipt. Data stays in-tenant; only the hash is anchored.

Live catalog endpoints: POST /v1/courtchain/anchor; GET /v1/courtchain/anchor/{id}.

Repository specification: JIL Sovereign CourtChain (CREB Record Anchor) version 2026.06.08. The API host path /openapi/courtchain.yaml returned HTTP 404 on 2026-10-21. The file on this website is <https://jilsovereign.com/openapi/courtchain.yaml>.

Specification text: Store the cryptographic CREB record in JIL. The customer's in-tenant (SPCS) app submits a bundle HASH (not the data) plus metadata; JIL anchors it to the L1 and returns a tamper-evident receipt (anchor_id, l1_tx_hash, timestamp, signature). PII/PHI never leaves the customer tenant. Requires a subscription credential scoped to `courtchain` and a hybrid-PQ signature (see api-gateway.yaml).

POST /v1/courtchain/anchor

Specification summary: Anchor a CREB record (bundle hash + metadata) to the JIL L1

Specification responses: 200: Anchored 401: Unauthorized

No published request or response example.

GET /v1/courtchain/anchor/{id}

Specification summary: Read back an anchored CREB record receipt

Specification responses: 200: OK 404: Not found

No published request or response example.