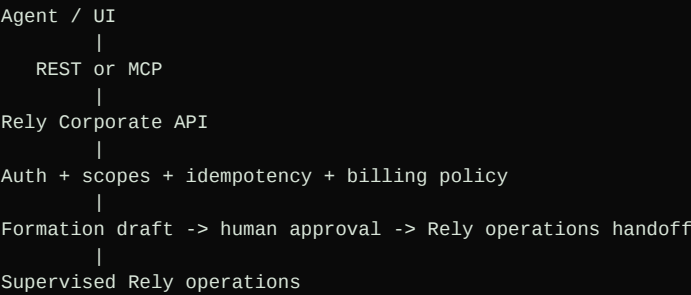


# Technical Brief

Closed beta en hardening · API-first · MCP como adaptador · decisiones humanas

## 1. Modelo de integración

La Corporate API es la fuente de verdad. MCP y CLI consumen el mismo contrato; no acceden a la base de datos ni a proveedores.



El partner utiliza credenciales emitidas por Rely. Corporate Tools, IRS y otros provider credentials nunca se exponen.

## 2. Contrato V1 objetivo

| Método | Ruta                                  | Uso                      |
|--------|---------------------------------------|--------------------------|
| GET    | /tools                                | Descubrir capacidades    |
| POST   | /formation-orders                     | Crear draft LLC + EIN    |
| GET    | /formation-orders                     | Listar y filtrar órdenes |
| GET    | /formation-orders/{id}                | Estado y blockers        |
| POST   | /formation-orders/{id}/submit-request | Pedir revisión           |
| GET    | /runs/{runId}                         | Ejecución sanitizada     |
| GET    | /approvals                            | Sólo lectura             |
| GET    | /ledger                               | Audit trail              |

### Autenticación

X-Rely-Agent-Key: rac\_<prefix>\_<secret>

La key se entrega una sola vez. Rely persiste únicamente prefix y hash. Sandbox y producción usan credenciales diferentes; el environment es metadata, no parte del prefix.

## Crear una orden

```
POST /api/corporate/v1/formation-orders
X-Rely-Agent-Key: rac_0123456789abcdef_<64-char-secret>
Content-Type: application/json
```

```
{
  "idempotencyKey": "northstar-001",
  "payload": {
    "externalCustomerId": "cus_northstar",
    "externalOrderId": "order_001",
    "companyName": "Northstar Labs LLC",
    "formationState": "WY",
    "entityType": "LLC",
    "applicantEmail": "founder@example.com",
    "einRequested": true
  }
}

{
  "success": true,
  "data": {
    "runId": "91c4a5d2-7b3e-4f10-8c62-5a9d1e7f3b20",
    "status": "succeeded",
    "output": {
      "formationOrderId": "7f2a9c31-1e4b-4c8d-9a71-2f6e8b5d3c10",
      "externalOrderId": "order_001",
      "environment": "sandbox",
      "status": "draft_ready",
      "approvalRequired": false,
      "nextAction": "manual_submit_request"
    }
  }
}
```

Misma key + mismo payload: resultado original. Misma key + payload distinto: 409.

## 3. MCP

El binario MCP stdio llama exclusivamente a la Corporate API y hereda RELY\_API\_BASE\_URL y RELY\_AGENT\_API\_KEY del entorno; la raw key no se copia al JSON del host.

```
rely-mcp
```

### Tools iniciales

- rely.formation\_order.create
- rely.formation\_order.list
- rely.formation\_order.read
- rely.formation\_order.submit\_request
- rely.run.status.read
- rely.approvals.list
- rely.ledger.read

No existe una tool de approval decision para la API key del partner.

## 4. Seguridad y operación

- Aislamiento por tenant y credential.
- Idempotencia durable antes de cualquier side effect.
- Keys revocables, scopes cerrados y output sanitizado.
- Producción bloqueada sin contrato activo o créditos. La misma orden puede promocionarse desde sandbox y el crédito prepago se descuenta sólo con aprobación humana; replay, rechazo y pedido de información no cobran.
- Rate limit por credential: sandbox 60/min, standard 120/min y partner 300/min; fallback por IP.
- Una aprobación humana cambia la orden a `approved_for_fulfillment` y deja el handoff `ready_for_operations`.
- El handoff crea evidencia para operaciones; V1 no escribe al provider desde API/MCP.
- Sandbox nunca crea pagos ni presenta una LLC o EIN real.

## 5. Release gate del piloto

Antes de entregar una sandbox key:

- 01 Integrar la rama del conector con dev y pasar CI.
- 02 Crear, leer y listar una orden mediante REST.
- 03 Repetir la orden y probar conflicto de idempotencia.
- 04 Leer run y ledger; ejecutar una tool MCP.
- 05 Confirmar 403 en producción sin billing.
- 06 Confirmar `approval_pending` y `providerSubmissionPerformed=false` con billing.
- 07 Revocar la key disposable y confirmar 401.

### Límite V1

La decisión humana se prueba por separado del smoke de partner keys. Polling primero; webhooks salientes, onboarding self-service y ejecución automática quedan fuera del piloto inicial.