Technical Case: Enterprise Intelligent OCR Microservice

Automated Financial Data Extraction & Reconciliation System

---

1. Executive Summary

This project involved the architecture and development of a high-performance **OCR Microservice** designed to automate the ingestion, extraction, and enrichment of unstructured financial documents (invoices, receipts, bank statements).

Moving beyond traditional, brittle OCR solutions (like Tesseract), this system leverages the reasoning capabilities of modern Large Language Models (LLMs) to achieve high-accuracy data extraction, enabling automated financial reconciliation across global markets.

---

2. The Business Challenge

The client faced significant operational bottlenecks in their back-office financial operations:

- **Unstructured Data:** Financial documents arrived in various formats (PDF, JPG, PNG) with no standardized layout.
- **Global Complexity:** The system needed to process multi-lingual documents and handle diverse regional currency formats (e.g., Argentine Pesos vs. Ukrainian Hryvnias).
- **Accuracy Requirements:** Strict requirements for "comparison keys" — normalized amounts, dates, and merchant categories — were essential for downstream automated reconciliation. Standard OCR tools failed to capture context or categorize transactions correctly.

---

3. The Solution: Async-First Architecture

I architected a robust, asynchronous microservice using **Python 3.11+** and **FastAPI**, designed for high concurrency and fault tolerance.

3.1. Multi-LLM Orchestration Layer

To balance cost and performance, I built a unified abstraction layer that orchestrates communication with multiple providers:

- **OpenAI (GPT)**
- **Anthropic (Claude)**
- **Google Vertex AI (Gemini)**

This allows the business to dynamically switch models based on document complexity (e.g., using cheaper models for simple receipts and advanced reasoning models for complex invoices) without code changes.

3.2. LLMOps & Observability (Langfuse)

A critical component of the architecture was the integration of **Langfuse** for professional "LLM Operations":

- **Prompt Management:** Prompts are decoupled from the code and managed via the Langfuse dashboard. This enables version control (e.g., v1-prod vs v2-test) and A/B testing of prompt engineering strategies without backend redeployment.
- **Tracing & Monitoring:** Every extraction request is fully traced, providing visibility into token usage, latency, and detailed execution paths for debugging hallucinations.

3.3. Data Reliability & Validation

- **Strict Typing:** Utilized **Pydantic** to enforce rigorous data schemas. All extracted fields (amounts, currency codes, timestamps) are validated and normalized before reaching the database.
- **Robust Parsing:** Integrated json-repair to gracefully handle and correct malformed JSON outputs from LLMs, ensuring system stability.

---

4. Key Capabilities & Impact

Capability	Business Value
Smart Categorization	Automatically classifies document types and merchant categories for seamless accounting integration.
Confidence Scoring	Provides reasoning metadata and confidence scores for every extraction, flagging low-confidence data for human review.
Format Agnostic	Capable of processing multi-page PDFs and various image formats with built-in normalization utilities (pdf2image, Pillow).
Global Scalability	Successfully handles diverse languages and regional currency formats, enabling international expansion.

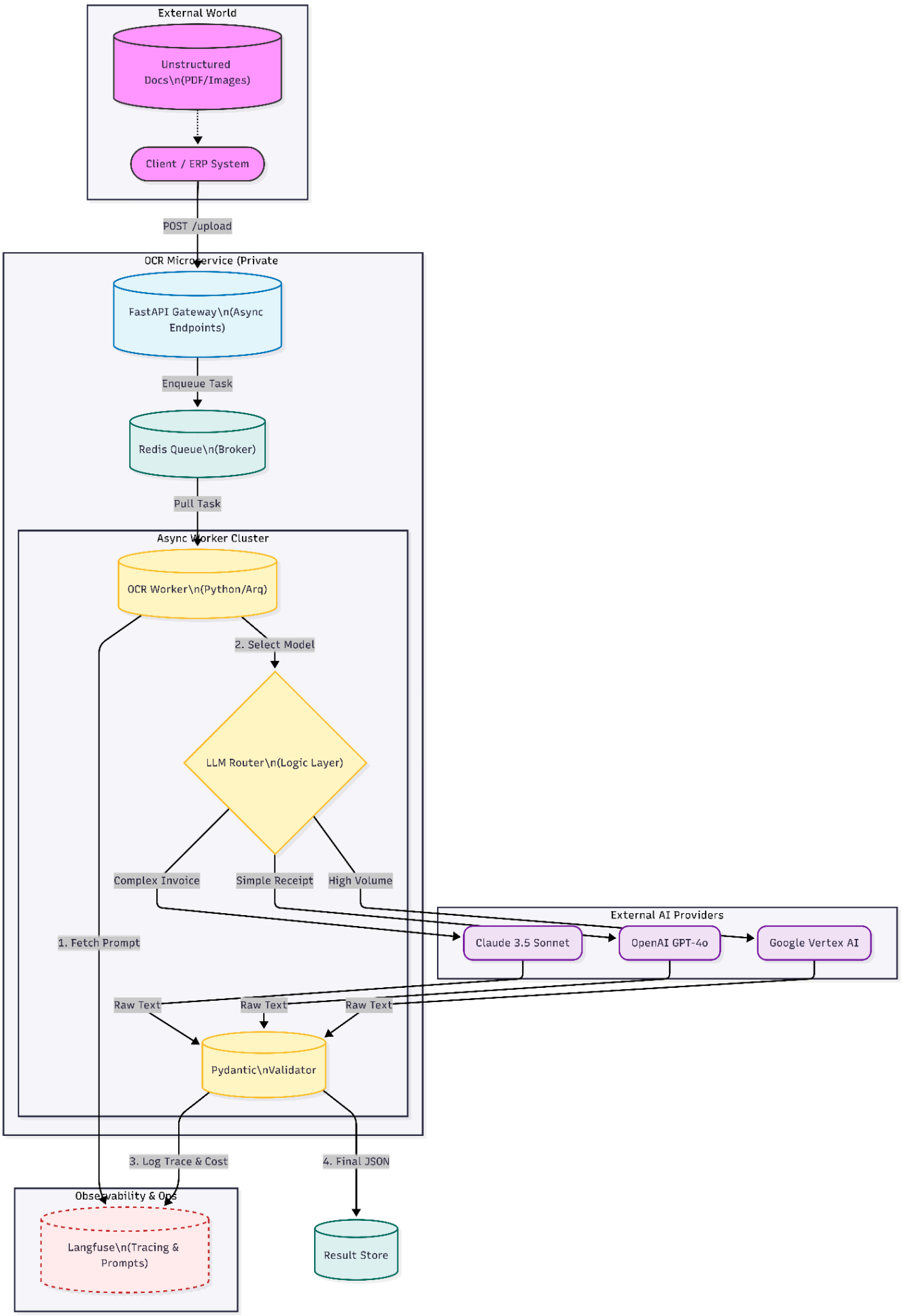
---

5. Technology Stack

- **Core Framework:** Python 3.11+, FastAPI, Asyncio
- **Data Validation:** Pydantic
- **AI Providers:** OpenAI API, Anthropic API, Google Vertex AI
- **LLMOps:** Langfuse (Tracing, Prompts, Evals)
- **Utilities:** json-repair, pdf2image, Pillow
- **Infrastructure:** Docker, Docker Compose

6. System Architecture Flow:

1. Ingestion (FastAPI): The client uploads a document via a non-blocking `POST` endpoint. The API performs basic validation and immediately offloads the processing task to a Redis Queue, returning a `job_id` to ensure zero timeout risks.
2. Async Orchestration (Worker): A dedicated Python worker consumes the task. It utilizes a Smart Router to analyze the document type (e.g., "Invoice" vs. "Receipt") and dynamically selects the most cost-effective LLM provider (OpenAI, Anthropic, or Vertex AI).
3. LLMOps Integration (Langfuse):
    - Before Execution: The worker fetches the latest "Production" version of the prompt from Langfuse, decoupling prompt engineering from code deployment.
    - During Execution: The full trace (Input Image → Prompt → LLM Response → Output) is logged asynchronously to Langfuse for cost tracking and debugging.
4. Validation & Output: The raw LLM output is parsed through json-repair and strictly validated against Pydantic schemas. Validated JSON is stored for the client to retrieve via polling.


External World
Unstructured Docs\n(PDF/Images)
Client / ERP System
POST /upload
OCR Microservice (Private
FastAPI Gateway\n(Async Endpoints)
Enqueue Task
Redis Queue\n(Broker)
Pull Task
Async Worker Cluster
OCR Worker\n(Python/Arq)
2. Select Model
LLM Router\n(Logic Layer)
Complex Invoice
Simple Receipt
High Volume
1. Fetch Prompt
External AI Providers
Claude 3.5 Sonnet
OpenAI GPT-4o
Google Vertex AI
Raw Text
Raw Text
Raw Text
Pydantic\nValidator
3. Log Trace & Cost
4. Final JSON
Observability & Ops
Langfuse\n(Tracing & Prompts)
Result Store

Examples:

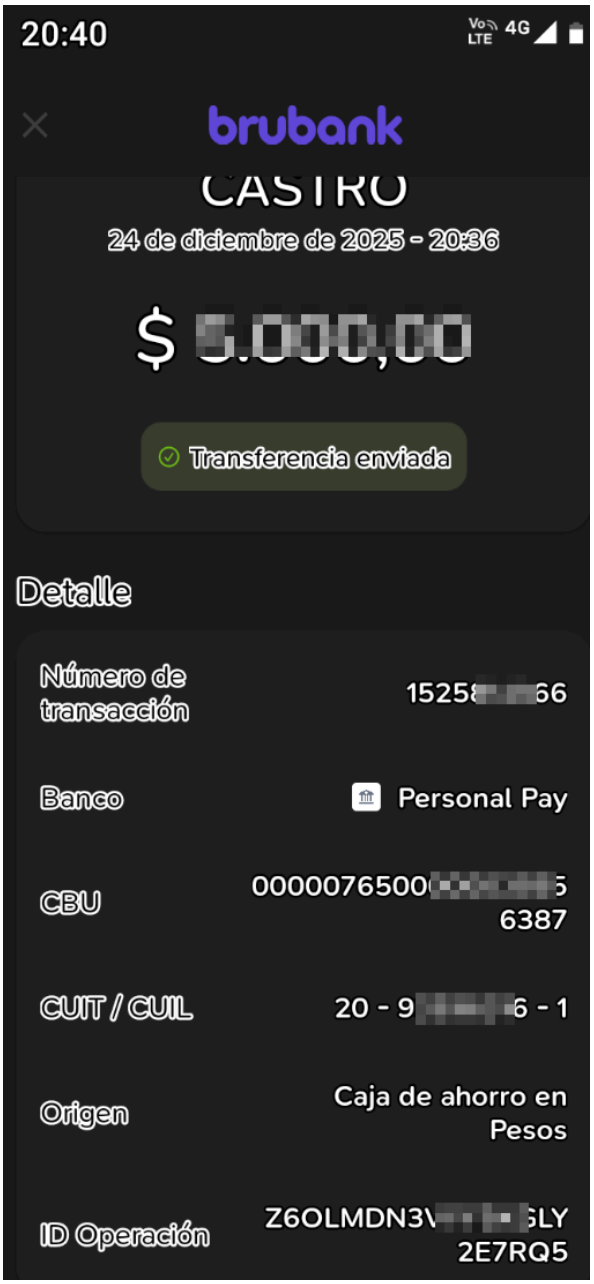
Example 1:


```
{
   "status": "success",
   "response": {
      "total_pages": 1,
      "pages": [
         {
            "page_number": 1,
            "raw_text": "Comprobante de transferencia\nJueves, 25 de diciembre de 2025 a las 04:08 hs\n\n$ XXXX\nMotivo: Varios\n\nDe\nAldXXXXXXrena\nCUIT/CUIL: 20-3XXXX788-0\nMercado Pago\nCVU: 00000031001XXXXXX269\n\nPara\nAndrea Celeste Nievas\nCUIT/CUIL:
```

27-44XXXX36-5\nFIWIND PAY SA\nCVU: 0000026XXXXXXXX35317\n\nNúmero de operación de Mercado Pago\n138787XXXXX79\nCódigo de identificación\n1LMP68NKVXXXXXXXX7OEV",
"language": "es",
"institution": "Mercado Pago",
"structured_data": {
"Header Section": {
"Comprobante de transferencia": "",
"Fecha": "Jueves, 25 de diciembre de 2025 a las 04:08 hs",
"Monto": "$ XXXX",
"Motivo": "Varios"
},
"Sender Section": {
"De": "AldXXXXXXena",
"CUIT/CUIL": "20-35XXXXXX8-0",
"Banco": "Mercado Pago",
"CVU": "0000003XXXXXX8269"
},
"Recipient Section": {
"Para": "AndXXXXXXXXvas",
"CUIT/CUIL": "27-447XXXXX-5",
"Banco": "FIWXXXXXAY SA",
"CVU": "000002XXXXX5317"
},
"Transaction List": [],
"Footer Section": {
"Número de operación de Mercado Pago": "13878XXXXX79",
"Código de identificación": "1LMP68NKVXXXXXXXR7OEV"
}
},
"enrichment": {
"normalized_timestamp": "2025-12-25T04:08:00-03:00",
"normalized_date": "2025-12-25",
"normalized_total_amount": XXXXXX,
"currency_code": "ARS",
"country_code": "AR",
"timezone": "America/Argentina/Buenos_Aires",
"document_type": "receipt",
"merchant_category_code": "transfers"
},
"inference": {
"confidence_score": 95,
"reasoning": "The text is clear and well-structured, with all necessary information visible.",
"model_used": "openai/gpt-4o-mini",
"image_size": 386961,
"image_dimensions": "921x2048",
"tokens_in_image": 1445,
"tokens_in_prompt": 2192,
"tokens_out_response": 469,
"duration_seconds": 10.53452
}

```
        }
      ]
    }
}
```

Example 2:

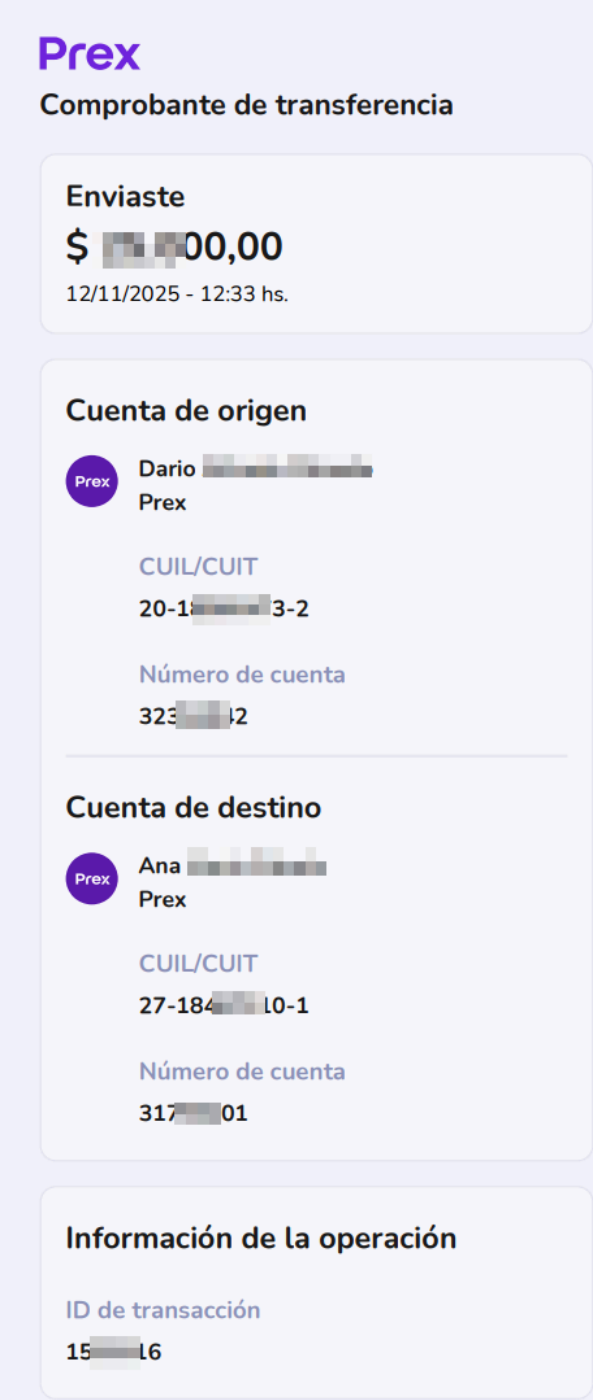



```
{
   "status": "success",
   "response": {
      "total_pages": 1,
      "pages": [
         {
            "page_number": 1,
```

```
            "raw_text": "brubank\n\nCASTRO\n24 de diciembre de 2025 - 20:36\n\n$ XXXX,00\n\n✔️ Transferencia enviada\n\nDetalle\n\nNúmero de transacción\n1525814366\n\nBanco\nPersonal Pay\n\nCBU\n0000076XXXXXXXX58637\n\nCUIT / CUIL\n20 - 93XXXXX16 - 1\n\nOrigen\nCaja de ahorro en Pesos\n\nID Operación\nZ6OLMDN3VYY5KGLY 2E7RQ5",
            "language": "es",
            "institution": "brubank",
            "structured_data": {
               "Header Section": {
                  "Payment Date": "24 de diciembre de 2025 - 20:36",
                  "Amount": "$ XXX,00",
                  "Status": "Transferencia enviada"
               },
               "Sender Section": {
                  "CUIT / CUIL": "20 - 9XXXX16 - 1",
                  "Origen": "Caja de ahorro en Pesos"
               },
               "Recipient Section": {},
               "Transaction List": [],
               "Footer Section": {
                  "Número de transacción": "1525814366",
                  "Banco": "Personal Pay",
                  "CBU": "0000076XXXXX58637",
                  "ID Operación": "Z6OLMDN3VYY5KGLY 2E7RQ5"
               }
            },
            "enrichment": {
               "normalized_timestamp": "2025-12-24T20:36:00-03:00",
               "normalized_date": "2025-12-24",
               "normalized_total_amount": XXXX,
               "currency_code": "ARS",
               "country_code": "AR",
               "timezone": "America/Argentina/Buenos_Aires",
               "document_type": "receipt",
               "merchant_category_code": "financial"
            },
            "inference": {
               "confidence_score": 95,
               "reasoning": "The image quality is good, and the text is clear and well-structured, allowing for accurate extraction of all relevant data.",
               "model_used": "openai/gpt-4o-mini",
               "image_size": 361226,
               "image_dimensions": "921x2048",
               "tokens_in_image": 1445,
               "tokens_in_prompt": 2192,
               "tokens_out_response": 399,
               "duration_seconds": 10.31528
            }
         }
      ]
   }
```

}

Example 3:



{
"status": "success",
"response": {
"total_pages": 1,
"pages": [
{
"page_number": 1,
"raw_text": "Prex\nComprobante de transferencia\n\nEnviaste\n$ XXX00,00\n12/11/2025 - 12:33 hs.\n\nCuenta de origen\nDario Alfredo Olmedo\nPrex\nCUIL/CUIT\n20-1XXXX73-2\nNúmero

de cuenta\n323XXX42\n\nCuenta de destino\nAna XXXX\nPrex\nCUIL/CUIT\n27-18487110-1\nNúmero de cuenta\n31XXX1\n\nInformación de la operación\nID de transacción\n155XXXX16",
"language": "es",
"institution": "Prex",
"structured_data": {
"Header Section": {
"Enviaste": "$ XXX00,00",
"Fecha": "12/11/2025 - 12:33 hs."
},
"Sender Section": {
"Nombre": "Dario XXXX Olmedo",
"CUIL/CUIT": "20-18XXXX3-2",
"Número de cuenta": "323XXXX42"
},
"Recipient Section": {
"Nombre": "Ana XXXX Zabala",
"CUIL/CUIT": "27-18XXX10-1",
"Número de cuenta": "317XXX1"
},
"Transaction List": [],
"Footer Section": {
"ID de transacción": "15XXX16"
}
},
"enrichment": {
"normalized_timestamp": "2025-11-12T12:33:00-03:00",
"normalized_date": "2025-11-12",
"normalized_total_amount": XXX00,
"currency_code": "ARS",
"country_code": "AR",
"timezone": "America/Argentina/Buenos_Aires",
"document_type": "receipt",
"merchant_category_code": "transfer"
},
"inference": {
"confidence_score": 95,
"reasoning": "The image quality is high, and the text is clear and well-structured, allowing for accurate extraction.",
"model_used": "openai/gpt-4o-mini",
"image_size": 88421,
"image_dimensions": "625x1521",
"tokens_in_image": 1105,
"tokens_in_prompt": 2192,
"tokens_out_response": 395,
"duration_seconds": 7.652675
}
}
]
}

```
}
```