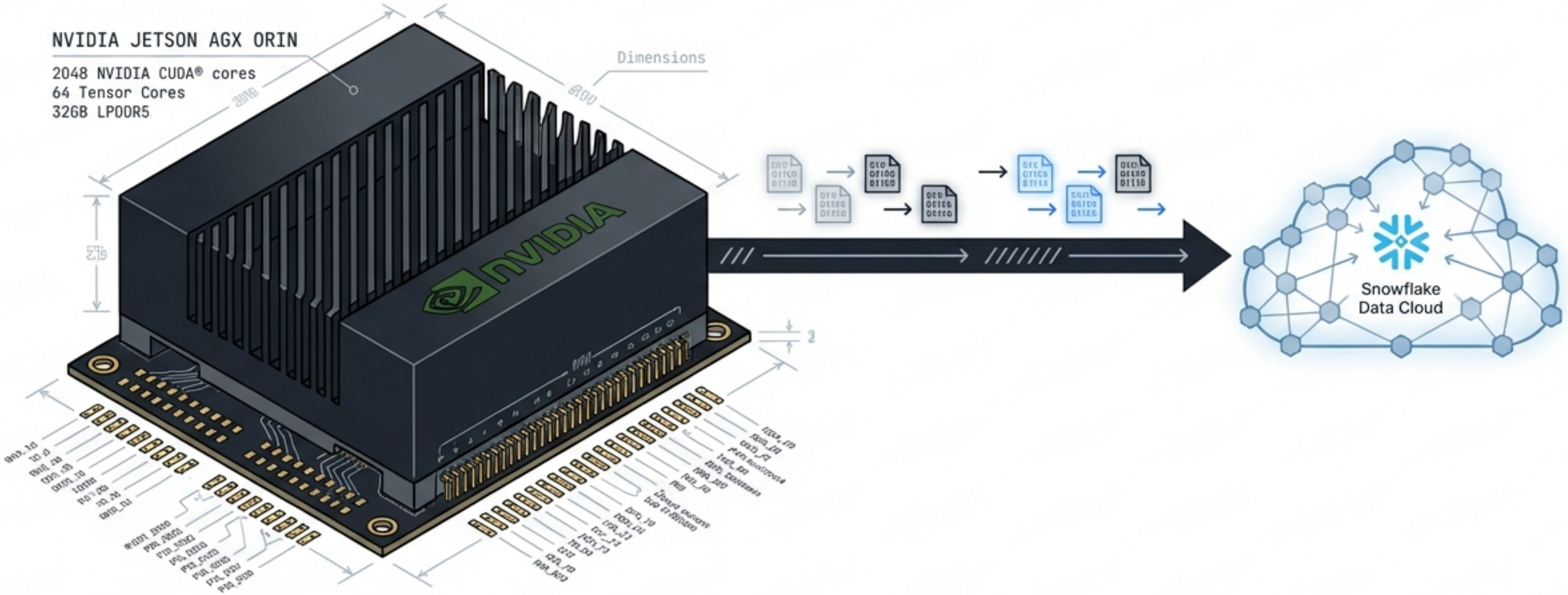
# NVIDIA Orin Edge AI Device Streaming

High-speed data ingestion from Edge to Cloud using Snowpipe Streaming v2 REST API.



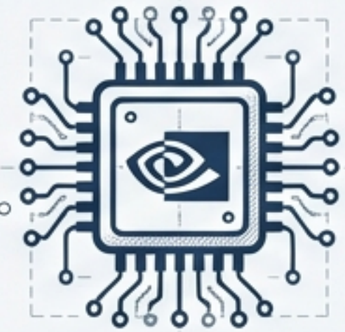

| | |
|---|---|
| TARGET HARDWARE: | NVIDIA JETSON AGX ORIN |
| PROTOCOL: | SNOWPIPE STREAMING V2 (REST) |
| VERSION: | 1.0 |

# Architecture Overview

0 mm 10 20 30 40 50 60 70 80 90 100mm

50 mm | 53 mm | 50 mm | 53 mm

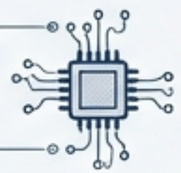
30 mm



## HARDWARE

Targeted for NVIDIA Jetson AGX Orin devices.

**DEVICE:** NVIDIA AGX OBIN
**PROCESSOR ARCH:** ARM64



30 mm


## SOFTWARE

Pure Python client leveraging scoped tokens and standard HTTP requests.

**LANG:** PYTHON 3.10+
**PROTOCOL:** HTTP/HTTPS → HTTP
**AUTH:** SCOPED TOKENS

30 mm

## OUTCOME

Streams system metrics and AI-enriched video data to Snowflake in real-time.

**DATA TYPE:** METRICS & VIDEO
**TARGET:** SNOWFLAKE DATA CLOUD
**LATENCY:** REAL-TIME STREAMING

30 mm

## INTEGRATION

Uses Snowpipe Streaming v2 REST API. No heavy ODBC/JDBC drivers required.

**API:** SNOWPIPE STREAMING V2
**DRIVER:** NONE REQUIRED
**METHOD:** RESTFUL

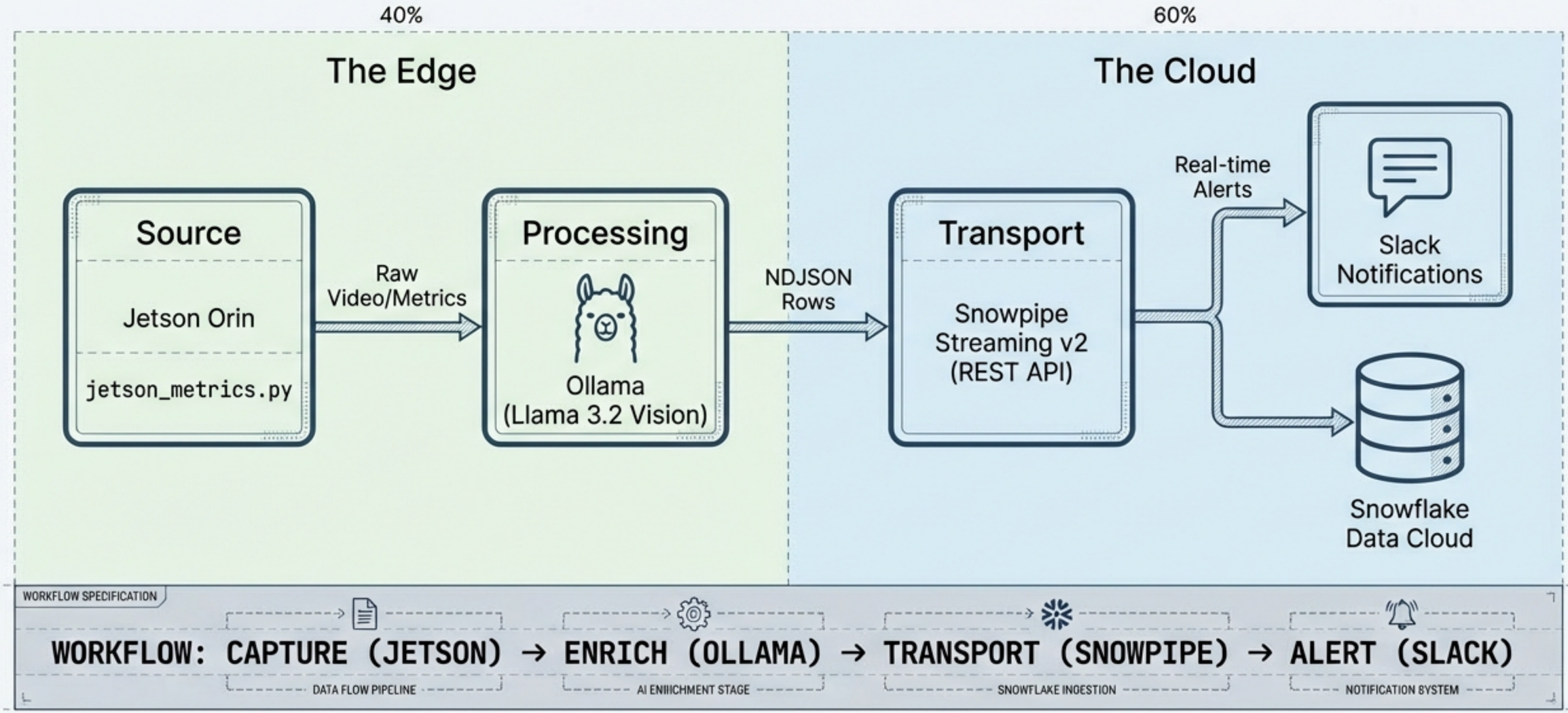
WORKFLOW SPECIFICATION

**WORKFLOW: CAPTURE (JETSON) → ENRICH (OLLAMA) → TRANSPORT (SNOWPIPE) → ALERT (SLACK)**

DATA FLOW PIPELINE | AI ENRICHMENT STAGE | SNOWFLAKE INGESTION | NOTIFICATION SYSTEM

| | |
|---|---|
| PROJECT: ARCHITECTURE OVERVIEW | DRAWING NO: A-101 |
| DATE: OCT 26, 2023 | DESIGNER: AI SYSTEM |

# The Data Pipeline




| | |
|---|---|
| PROJECT: ARCHITECTURE OVERVIEW | DRAWING NO: A-102 |
| DATE: OCT 26, 2023 | DESIGNER: AI SYSTEM |

# Core Capabilities

## High-Speed Ingest

Utilizes the Snowpipe Streaming v2 REST API with a scoped token flow for low-latency data transfer. Bypasses traditional drivers.

## Edge AI Enrichment

Optional integration with Ollama to process images locally using 'llama3.2-vision' before streaming context to the cloud.

## Authentication Flexibility

Supports Key-pair JWT for production security and Programmatic Access Tokens (PAT) for rapid development.

## Multi-Modal Data

Simultaneously handles structured system metrics (JSON) and unstructured video frame data in the same pipeline.


| | |
|---|---|
| PROJECT: ARCHITECTURE OVERVIEW | DRAWING NO: A-102 |
| DATE: OCT 26, 2023 | DESIGNER: AI SYSTEM |

# Project Structure

- nvidiastreaming
  - main.py — Entry point and CLI handler.
  - snowpipe_streaming_client.py — Manages REST API lifecycle (Token exchange, Channel management).
  - jetson_metrics.py — Collects hardware stats from the Orin.
  - ollama_client.py — Interface for AI model interaction.
  - snowflake_jwt_auth.py — Generates JWTs for key-pair authentication.
  - setup_snowflake.sql — Database preparation script.


| | |
|---|---|
| PROJECT: ARCHITECTURE OVERVIEW | DRAWING NO: A-102 |
| DATE: OCT 26, 2023 | DESIGNER: AI SYSTEM |

# Edge AI Layer: Ollama Integration

Input Frame

**ollama_client.py**

model: llama3.2-vision

JetBrains Mono

Output Summary

A server rack with blue lights indicating normal operation.

**Code Note**

Command to activate:

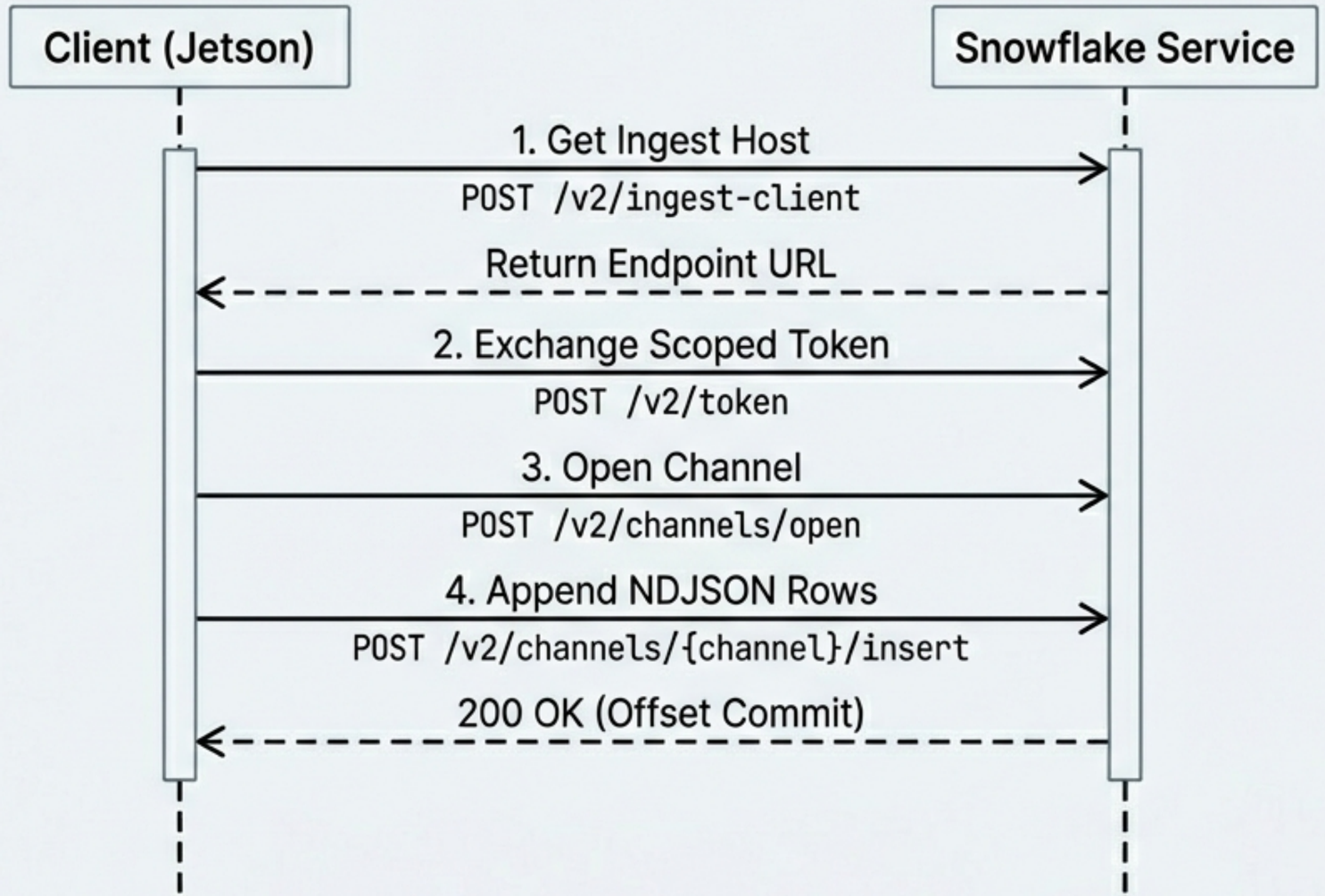
```
python main.py --ollama-model llama3.2-vision
```

Benefit: Enrichment happens at the edge, reducing cloud bandwidth usage.

PRDJECT: ARCHITECTURE DVERVIEW | DRAWING NO: A-102

BATE: OCT 26, 2028 | DCSIGNER: AI 9YSTEM

# Streaming Layer: Snowpipe v2 REST API

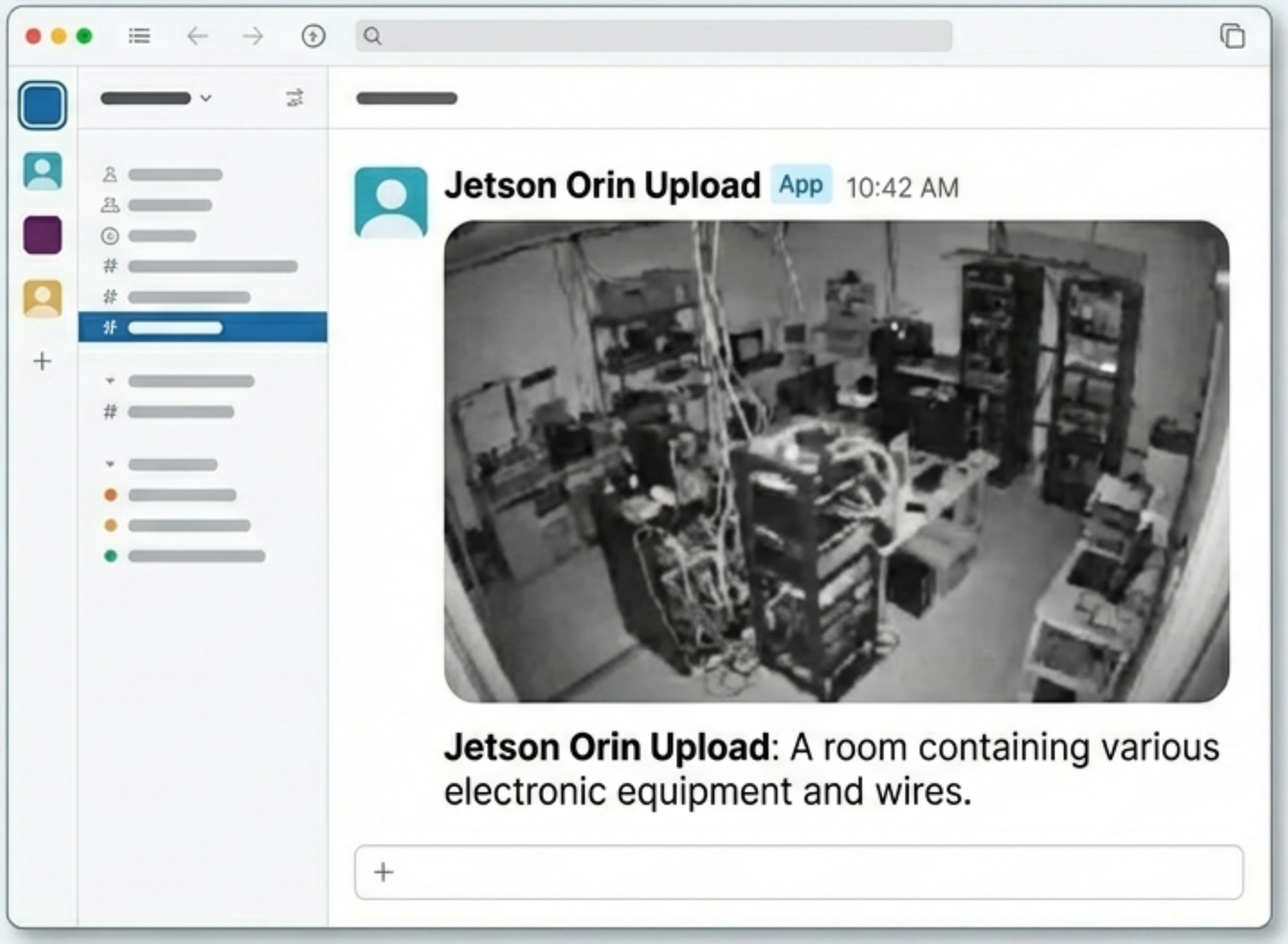



**Technical Advantage:**

This flow completely bypasses traditional JDBC/ODBC wrappers, making it lightweight and ideal for ARM64 architecture.

PROJECT: ARCHITECTURE OVERVIEW | DRAWING NO: A-102
DATE: OCT 26, 2023 | DESIGNER: AI SYSTEM

# Real-Time Alerts: Slack Integration




**Configuration Required (snowflake_config.json):**

- "slack.enabled": true
- "slack.bot_token": "xoxb-..."
- "slack.channel": "C012345..."

PROJECT: ARCHITECTURE OVERVIEW | DRAWING NO. A-104
DATE: OCT 26, 2028 | DESIGNER: AI SYSTEM

# Prerequisites & Environment Setup

```
$ python3 -m venv venv
$ source venv/bin/activate
(venv) $ pip install -r requirements.txt

# In a separate terminal session
$ ollama serve
$ ollama pull llama3.2-vision
```

1. Python 3.8+ Environment
2. Dependencies installed via pip
3. Local Ollama instance running

PROJECT: ARCHITECTURE OVERVIEW | DRAWING NO. A-100
DATE: OCT 26, 2028 | DESIGNER: AI SYSTEM

# Snowflake Object Setup

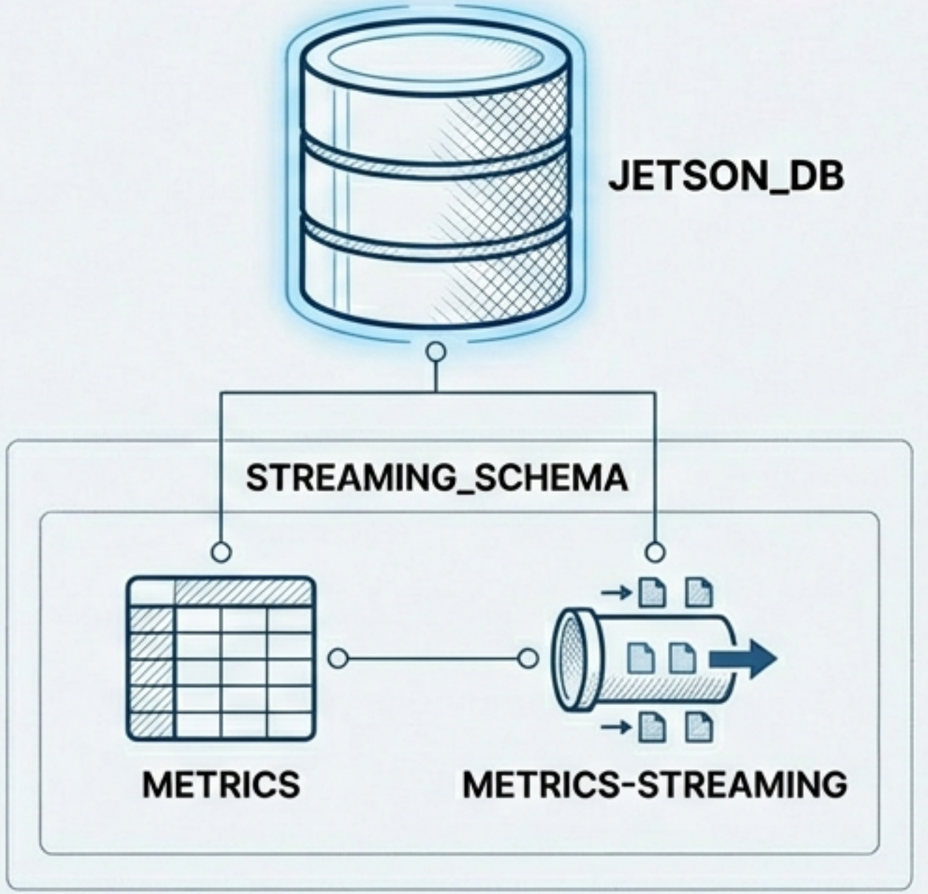
source: setup_snowflake.sql

```
CREATE DATABASE JETSON_DB;

CREATE SCHEMA STREAMING_SCHEMA;

CREATE TABLE METRICS (...);

CREATE PIPE METRICS_PIPE ...
```




**Action:** Run '**setup_snowflake.sql**' in Snowsight worksheets.

| | |
|---|---|
| PROJECT: ARCHITECTURE OVERVIEW | DRAWING NO. A-100 |
| DATE: OCT 26, 2028 | DESIGNER: AI SYSTEM |

# Configuration: snowflake_config.json

```
{
  "account": "ORG-ACCOUNT",
  "user": "jetson_user",
  "database": "JETSON_DB",
  "schema": "STREAMING_SCHEMA",
  "pipe": "METRICS_PIPE",
  "video_capture": {
    "enabled": true
  }
}
```

**Critical**: Use ORG-ACCOUNT format.

Important Tip: If your account hostname contains underscores (e.g., region_id), replace them with dashes in the ingest host URL.

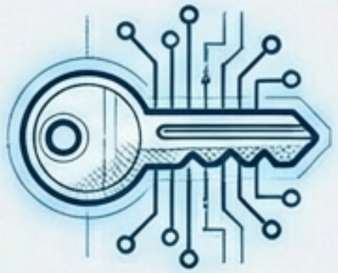
# Authentication Methods

## Method 1: Key-Pair JWT

**Best for:** Production Security

- **Requires:** "`private_key_path`" in config.
- **Mechanism:** Generates distinct JWTs via `snowflake_jwt_auth.py`.

## Method 2: Programmatic Access Token (PAT)



**Best for:** Development / Testing

- **Requires:** "`url`" and "`pat`" fields.
- **Format:** Matches "`RPIWeatherStreaming`" example.

# Running the Streamer

## Standard Run:

**Command Prompt**

```
> python main.py --config snowflake_config.json --batch-size 25 --interval 5.0
```

## With AI Enrichment:

**Command Prompt**

```
> python main.py --ollama-model llama3.2-vision
```

## Debug Mode:

**Command Prompt**

```
> python main.py --debug --batch-size 10
```

Debug Flag: Prints verbose progress bars and configuration hints to the console for troubleshooting.

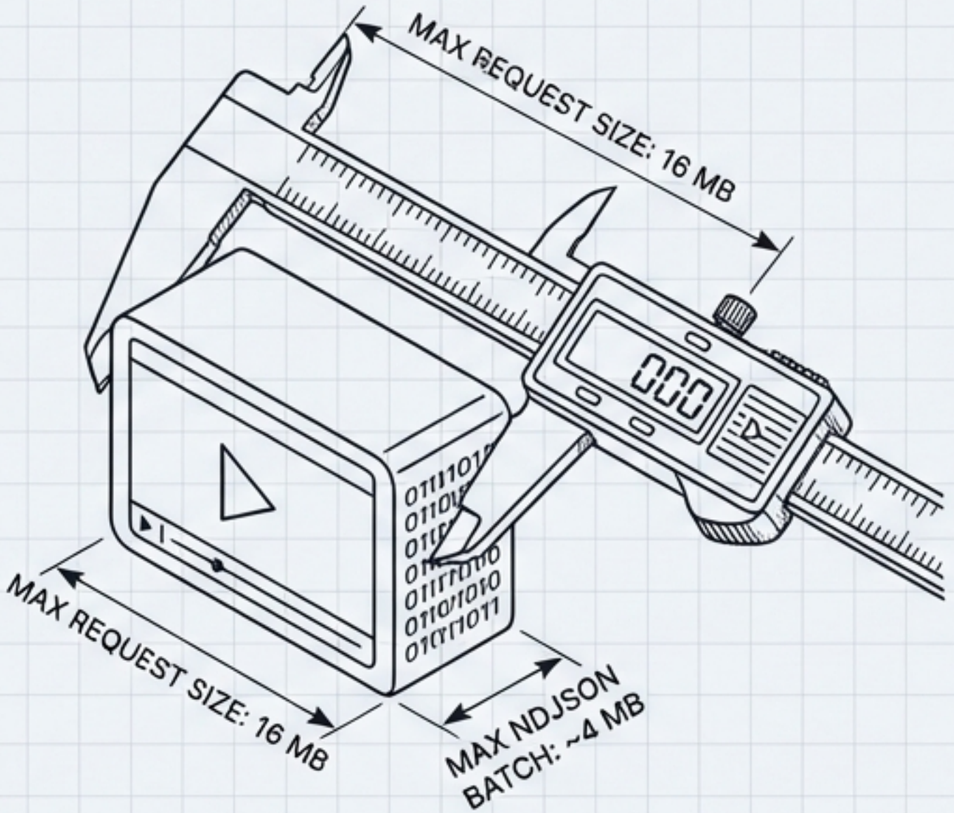
# Video Capture & Constraints

## 1. Video Config

Enable via JSON:

```
'video_capture': {
  'enabled': true,
  'device_index': 0
}
```

Files are saved locally to "output_dir".




## 2. Payload Limits (CRITICAL)

**Max Request Size:** 16 MB

**Max NDJSON Batch:** ~4 MB

**Constraint Advice:** Keep batch sizes small when streaming video data to avoid API rejection.

PROJ. NO.: ARCHITECT-DVCS-OVERVIEW | DRAWING NO. A-103

DATE: OCT 26, 2028 | DESIGNER: AI SYSTEM

# Resources & Documentation

## 1. GitHub Repository

`tspannhw/nvidiastreaming`

GitHub Repo

## 2. Snowflake Documentation

Snowpipe Streaming REST API & JWT Flow

Snowflake Docs

## 3. Ollama

Local LLM Management Library

Ollama Library

**Clone the repo, configure your JSON, and start streaming AI-enriched data from the edge today.**

| | |
|---|---|
| PRD.JSCT: ARDINTECTURB DVCRVIEW | RRAMINC NO. A-104 |
| BATE: OCT 26, 2028 | UCSIGNER: AI NYSTEM |