ChatGPT

Data Guide (v5) – Synthetic Invoice Approval Dataset & Metrics

This document describes the enhanced synthetic dataset used to replicate Mimica's analytics platform and extends it to support cross-region process standardisation. It defines the schema, variant logic, derived metrics, data processing steps and mapping to UI components.

1. Raw Dataset Overview

The file `new_synthetic_invoice_data.json` contains **4 153 step records** across **500 invoice approval transactions**. Each record captures the following fields:

Field	Description
`task_id`	Always `invoice_approval`. Allows combining multiple tasks in future.
`transaction_id`	Unique ID for each end-to-end process execution.
`region`	Region of the performer (Americas, EMEA, APAC, LATAM, North America).
`user_id`	Anonymised identifier; multiple transactions can be performed by the same user.
`role`	Role of the performer (AP clerk, Manager, Supervisor, Analyst).
`variant`	Label (A–E) representing structural differences in the process.
`step_index`	Order of the step within the transaction (1-based).
`action_name`	Name of the action (e.g., `read_email`, `manager_approval`).
`application`	Application used (Outlook, SAP, Excel, PDF Viewer, Notepad, "-").
`duration_sec`	Duration of the action in seconds.
`start_time_sec` / `end_time_sec`	Cumulative start/end time of the step relative to transaction start.
`decision_outcome`	Outcome for decision steps (within_tolerance, exceeds_tolerance, approved, requires_review); blank for non-decision steps.
`auto_score`	Synthetic score (0–1) indicating how automatable the step is.

Variants

Five variants model real-world deviations:

- **A (Standard)** – Baseline flow: read email → download & open invoice → validate data → check tolerance → approve → record log → notify.
- **B (Second Approval)** – Inserts `manager_approval` after approval; reflects stricter policies.
- **C (Local Log)** – Adds `update_local_log` after recording log; reflects regional habit of keeping a separate local log.
- **D (Early Termination)** – If tolerance check fails, branches to `send_for_review` then ends.
- **E (Summary Report)** – Adds `compile_summary` before notification; summarises the invoice for stakeholders.

Variant probabilities vary by region to simulate regional behaviours.

Step Type Classification

Each action is classified as one of four types:

Type	Criteria	Examples
Action	Direct tasks with clear instructions.	`read_email`, `download_attachment`, `approve_invoice`, `record_log`, `notify_requester`
Semi-structured Input	Tasks requiring data entry or validation with some variability.	`validate_invoice_data`, `update_local_log`, `compile_summary`
Decision	Steps involving a branching decision or approval.	`check_tolerance`, `manager_approval`, `send_for_review`
Virtualised Action	Conceptual or end steps with no user input.	`end_process`

This classification enables calculation of metrics such as ease of deployment gauge and counts of actions, semi-structured inputs and decisions.

2. Processed Metrics

Aggregated metrics are computed and stored in `/public/data/processed/` for efficient consumption by the front-end. Each file is a JSON array of objects matching the TypeScript interfaces defined in `lib/types.ts`.

2.1 Region Metrics (`processed_region_metrics.json`)

Fields: `region`, `avg_duration`, `median_duration`, `max_duration`, `min_duration`, `avg_step_count`, `transaction_count`.

Usage: Bar charts comparing average cycle time per region; summary cards for median/min/max durations and step counts; filter menus.

2.2 Variant Distribution (`processed_variant_distribution.json`)

Structure: Each object has `region` and counts for variants A–E. The counts represent number of transactions in each region following that variant.

Usage: Stacked bar chart showing variant composition by region; variant filter controls.

2.3 Variant Metrics (`processed_variant_metrics.json`)

Fields: `variant`, `avg_duration`, `median_duration`, `step_count`, `transaction_count`.

Usage: Compare performance across variants; identify candidates for best practice; inform recommendation logic.

2.4 Step Metrics (`processed_step_metrics.json`)

Fields: `action_name`, `avg_duration`, `median_duration`, `count`, `type` (action/semi-structured/decision/virtualised).

Usage: Bottleneck detection table; highlight slow steps on the map; compute ease of deployment gauge (based on counts of step types).

2.5 Top Bottlenecks (`processed_top_bottlenecks.json`)

Subset of step metrics containing only the top five actions by average duration. Used to populate the bottleneck table and emphasise optimisation targets.

2.6 Step Type & Application Usage Metrics

These metrics are computed on the fly or stored in additional processed files:

- **Step Type Counts per Region** – Number of actions, semi-structured inputs, decisions and virtualised actions executed in each region. Supports the ease of deployment gauge and counts panel.
- **Application Usage per Region** – Share of total transaction time spent in each application (SAP, Outlook, Excel, etc.). Used for application donut charts. For websites, treat certain applications (e.g., SharePoint) as website proxies or add a `website` field to the raw data.

- **Decision Path Counts** – Count unique paths through decision points by grouping transactions on decision outcomes (e.g., `within_tolerance` vs. `exceeds_tolerance`); summarised by region.

3. Data Processing Flow

1. **Load raw step records** from `new_synthetic_invoice_data.json` using pandas/JavaScript.
2. **Group by transaction** to compute total duration and step count. Summarise by region and variant to produce region and variant metrics.
3. **Classify steps** using the step type table; compute counts of each type per region.
4. **Aggregate by action_name** to compute average and median durations and frequency; sort to identify top bottlenecks.
5. **Compute application usage** by summing durations per application and dividing by total duration per region. Create percentage values for donut charts.
6. **Derive decision paths** by grouping transactions based on sequences of decision outcomes. Count distinct paths to determine "decision path" metric.
7. **Export processed files** to `public/data/processed/` as JSON. These files are used by the Next.js app to build dashboards.

4. Mapping Metrics to UI Components

UI Component	Data Source	Notes
Process List Table	Raw data aggregated by process (hardcoded single process in this mock)	Use `region_metrics` for summarising time spent.
Summary Panel – Time Saved	Requires baseline vs. improved durations (not in dataset)	For mock, compute time saved as difference between average duration and minimum duration times transaction count.
Automatability Rating	Compute percentage of steps with `auto_score` > 0.8	Categorise as Low/Medium/High/Very High.
Ease of Deployment Gauge	Step type counts per region; compute percentage shares	Visualise using four segments for actions, semi-structured, decisions, virtualised.
Counts Cards	Step type counts, number of applications, websites, decision paths	Derived metrics; websites approximated from applications or added field.
Applications & Websites Donut Charts	Application usage metrics	If `website` missing, treat certain apps as websites or augment dataset.
Region Comparison Bar Chart	`processed_region_metrics.json`	Plot `avg_duration` and `transaction_count` per region.

UI Component	Data Source	Notes
Variant Distribution Stacked Chart	`processed_variant_distribution.json`	Bars represent variant counts by region.
Variant Metrics Table/Chart	`processed_variant_metrics.json`	Display average duration and step count per variant.
Bottleneck Table	`processed_top_bottlenecks.json`	List top slow actions; clicking highlights nodes on map.
Process Map	Raw step sequences aggregated into directed graph	Use graph construction algorithm described in RESEARCH_v5.md.

5. Extending the Dataset

To fully emulate Mimica's features, future iterations could augment the dataset with:

- **Screenshot identifiers** – Link steps to screenshot assets for display in the step details panel.
- **Website domain** – Distinguish between applications and website pages (e.g., `sharepoint.com`, `sap.mycompany.com`).
- **Baseline durations** – Provide an alternative execution time representing the best practice; necessary for computing time saved metrics.
- **Transaction timestamps** – Add `timestamp` for each transaction to compute frequency per day and identify time-of-day patterns.
- **SME metadata** – Include SME names or IDs and the number of days they were recorded; needed to compute per-SME/day metrics.

This data guide ensures consistent understanding of the mock data and provides clear pathways from raw steps to the analytics and visualisations used in the MVP.