

ENTERPRISE TECHNICAL BLUEPRINT • ORIGO LABS

Sub-30ms Intent Routing: The Banking77 Enterprise Blueprint & TCO Guide

How enterprise financial institutions route high-cardinality customer support inquiries across 77 fine-grained categories with **74.51% Top-3 accuracy** on commodity CPU—with **0% cloud data egress** and a **99.6% reduction in operational spend** compared to cloud LLMs.

74.51% Top-3 Evaluated zero-shot across all 3,080 Banking77 test queries	30.2 ms Median Over 250 decisions/sec on a single standard CPU instance	100% Air-Gapped < 64 MB RAM footprint; zero external cloud API calls	> \$250k/yr ROI Flat \$100/mo VPS compute vs. \$20k–\$35k/mo cloud API bills
--	---	--	---

1. Problem Formulation: The High-Cardinality Triage Challenge

Tier-1 financial institutions receive tens of thousands of customer support inquiries daily across conversational web chat, mobile apps, and incident portals. Efficient automated triage requires classifying each message into one of dozens of highly specialized operational intents.

The standard industry benchmark, **Banking77** (PolyAI), mirrors this real-world operational reality:

- **77 Fine-Grained, Overlapping Categories:** Subtle distinctions such as `card_arrival` vs. `card_delivery_estimate`, or `lost_or_stolen_card` vs. `compromised_card` require exact context alignment.
- **Colloquial Customer Speech:** Inquiries are unstructured, informal, and filled with conversational slang ("still waiting on the plastic in the mail").
- **Sub-50ms Response Deadlines:** Real-time conversational workflows require intent classification to complete in under 50 milliseconds to avoid user-perceived lag.
- **Strict Regulatory Data Residency:** Transmitting unredacted customer inquiries with financial identifiers to external multi-tenant LLM APIs violates GDPR, SOC2, and banking compliance mandates.

2. Frontier Cloud LLMs vs. Local Decision Models: TCO Analysis

Enterprise architecture teams evaluating high-volume intent routing face an unsustainable cost and compliance penalty when deploying frontier cloud LLMs (e.g. GPT-4o, Claude 3.5) for high-frequency routing tasks:

EVALUATION DIMENSION	FRONTIER CLOUD LLMs (CLOUD APIS)	LOCAL ZEV + MICRO-EMBEDDING PIPELINE	ENTERPRISE IMPACT
Compute Infrastructure	Dedicated multi-tenant GPU clusters	1x Standard 4-Core CPU VPS (\$40–\$100/mo)	Runs on existing internal compute
Cost at 10M decisions/mo	\$20,000 – \$35,000 / month (\$240k–\$420k/yr)	\$100 / month flat hosting (\$1,200/yr)	99.6% cost reduction (>\$250k ROI)
Median Latency (p50)	800 ms – 2,500 ms (network lag + token generation)	27.0 ms (local in-process / LAN)	40x latency acceleration
Customer PII Data Egress	100% leaves enterprise security perimeter	0% egress (fully air-gapped on-prem)	Zero compliance liability under GDPR/SOC2
Positional Order Bias	12% – 35% variance based on prompt order	0.000006% variance (ICMB guarantee)	Auditable, mathematically invariant scoring

Executive Takeaway: High-frequency decision routing is fundamentally a *classification and verification problem*, not a text generation problem. Paying generative token rates (\$25k/mo) for routing decisions is economically inefficient and creates unnecessary regulatory exposure.

TWO-STAGE ARCHITECTURAL BLUEPRINT

3. The Solution: Two-Stage Cascaded Routing Architecture

To preserve strict order equivariance, ZEV enforces a bounded attention budget of $K \leq 10$ choices per forward pass. For high-cardinality enterprise catalogs (77+ choices), production architectures deploy a two-stage cascaded pipeline:

STAGE 1 • MICRO-EMBEDDING Model2Vec Candidate Filter Computes static 64-dim vector dot product against 77 pre-indexed intents in 14 μs on CPU. Narrows field to Top-8 candidates with 86.5% Recall@8	STAGE 2 • DECISION ENGINE ZEV Calibrated Scoring Executes a single non-autoregressive forward pass across the 8 candidates in 26 ms with Inter-Choice Masked Blindness (ICMB). Produces calibrated probabilities.	POLICY GATE • ACTION TRIAGE Automated Resolution Dispatches automated workflow if confidence ≥ 0.80; surfaces Top-3 suggestions to human agent console if $0.50 \leq \text{conf} < 0.80$; escalates if < 0.50.
--	---	---

Mathematical Scoring Formulation

Let customer query q and pre-indexed intent definitions $\{c_j\}_{j=1}^{77}$ be embedded via static embedding model M . The Top-8 candidate set C_8 is selected via cosine similarity in 14 μ s. ZEV then evaluates C_8 under ICMB attention, emitting calibrated decision probabilities $\mathbf{p}_{\text{zev}}$.

The production scoring ensemble blends the static semantic similarity p_{m2v} with ZEV's contrastive probabilities with blending parameter $\alpha = 0.3$:

$$P_{\text{production}}(c_i \mid q) = (1 - \alpha) \times p_{\text{m2v}}(c_i \mid q) + \alpha \times p_{\text{zev}}(c_i \mid q)$$

4. Empirical Performance Scorecard: All 3,080 Test Queries

Evaluated across the complete held-out Banking77 test set (3,080 inquiries, 77 categories) on commodity multi-core CPU:

PIPELINE CONFIGURATION	STAGE 1 FILTER	STAGE 1 RECALL@8	TOP-1 ACCURACY	TOP-3 ACCURACY	MEDIAN LATENCY	THROUGHPUT (CPU)
Legacy Lexical + ZEV	BM25 Token Overlap	68.80%	14.40%	50.40%	22.7 ms	165 dec/sec
Production (2M Embeddings + ZEV)	potion-base-2M	82.24%	48.90%	68.77%	31.6 ms	240 dec/sec
Production (8M Embeddings + ZEV)	potion-base-8M	86.49%	53.60%	74.51%	30.2 ms	254 dec/sec
Hierarchical Macro-Tree	ZEV 10-Domain Tree	27.60% (Domain)	4.40%	14.00%	66.1 ms	58 dec/sec

The 74.51% Assistive Multiplier: In agent triage consoles (Zendesk, Salesforce Service Cloud), presenting three pre-calculated macros allows representatives to resolve tickets with a single click. ZEV places the true intent within the top 3 suggestions in **74.51% of all customer interactions** in ~30 ms.

PRODUCTION IMPLEMENTATION & GOVERNANCE

5. 20-Line Production Implementation Blueprint

The entire two-stage pipeline can be deployed as an in-process Python/Rust service or lightweight Docker container:

```
import numpy as np
from model2vec import StaticModel
import urllib.request, json

# 1. Load static embedding model (~8MB RAM) and pre-index canonical intents
embedder = StaticModel.from_pretrained("minishlab/potion-base-8M")
intent_vecs = embedder.encode(ALL_77_INTENTS)
intent_vecs = intent_vecs / (np.linalg.norm(intent_vecs, axis=1, keepdims=True) + 1e-9)

def route_customer_inquiry(inquiry_text: str, k: int = 8, alpha: float = 0.3) -> dict:
    # Stage 1: Microsecond candidate filter (14 µs on CPU)
    q_vec = embedder.encode([inquiry_text])
    q_vec = q_vec / (np.linalg.norm(q_vec, axis=1, keepdims=True) + 1e-9)
    top_indices = np.argsort(-np.dot(intent_vecs, q_vec[0]))[:k]
    candidates = [ALL_77_INTENTS[i] for i in top_indices]

    # Stage 2: ZEV decision verification (26 ms on CPU)
    payload = {
        "state": inquiry_text,
        "questions": {
            "intent": {
                "type": "choice",
                "instructions": "Identify specific banking issue for department routing:",
                "options": [f"{c}: {c.replace('_', ' ').capitalize()}" for c in candidates]
            }
        }
    }
    req = urllib.request.Request("http://127.0.0.1:8788/v1/systemone",
                                data=json.dumps(payload).encode(),
                                headers={"Content-Type": "application/json"})
    with urllib.request.urlopen(req, timeout=2.0) as resp:
        zev_res = json.loads(resp.read().decode())

    # Stage 3: Ensemble scoring and policy triage
    zev_scores = [zev_res["answers"]["intent"]["probabilities"].get(f"{c}: {c.replace('_', ' ').capitalize()}", 0) for c in candidates]
    m2v_probs = np.exp(np.dot(intent_vecs[top_indices], q_vec[0]) * 10.0)
    blended = (1.0 - alpha) * (m2v_probs / np.sum(m2v_probs)) + alpha * np.array(zev_scores)
    best_idx = int(np.argmax(blended))
    top3_idx = np.argsort(-blended)[:3]

    return {
        "best_intent": candidates[best_idx],
        "confidence": float(blended[best_idx]),
        "top3_suggestions": [candidates[i] for i in top3_idx],
        "auto_dispatch": float(blended[best_idx]) >= 0.80
    }
```

6. Enterprise Governance & Compliance Verification

- **Zero Cloud Data Egress (100% Air-Gapped):** All weights and embeddings run entirely in local RAM. Zero customer messages or telemetry leave the on-prem VPC.
- **Auditable Order Equivariance:** Inter-Choice Masked Blindness guarantees 0.000006% positional variance across choice permutations, eliminating prompt recency bias.
- **Deterministic Cold Boot:** Full engine and embedding initialization completes in under 350 ms on cold process start.
- **Commercial Licensing Options:** Available under self-hosted perpetual enterprise licenses, managed cloud PAYG via Opper AI, or custom dedicated cluster agreements.

Start an Empirical Evaluation on Your Data

Evaluate ZEV against your internal support queues, intent taxonomies, or tool catalogs on your own infrastructure.

Web: <https://zev.origolabs.com/evaluate/> · Inquiries: licensing@origolabs.com