Connecting the Dots: A Secure Protocol for Cross-Team Knowledge Discovery

How to find semantic relationships across an organization without exposing anyone's work

The Agile Monkeys

March 2026

The Problem

In any organization of meaningful size, people routinely solve the same problems independently. Two teams build integrations with the same API. Three engineers research the same library. A project in Madrid runs into the same technical constraint that a team in London resolved six months ago.

This isn't a failure of communication — it's a failure of discoverability. No one is withholding information; they simply don't know who else is working on what. The information exists, scattered across personal notes, project documents, Slack channels, and agent memory. But there is no mechanism to surface these connections without exposing the underlying work.

The naive approach — "dump everything into a shared knowledge base and let people search" — fails for three reasons. First, access control: not everyone should see everything. Sensitive projects, pre-announcement work, and competitive intelligence cannot be made broadly searchable. Second, information overload: a single searchable corpus across a 10,000-person organization produces more noise than signal. Third, and most fundamentally, privacy: people need to control what they share about their work. A system that exposes your work to the entire organization without your consent is one that people will refuse to use — or worse, feed with misleading information.

The challenge is to build a system that can detect "Person A and Person B are working on semantically related things" without revealing to anyone — including the system itself — what either person is actually working on.

Why This Is Harder Than It Looks

The obvious technical approach is to store embeddings (vector representations) of everyone's work topics in a database, run similarity searches, and notify people when matches are found. This approach has a critical flaw: **text embeddings can be inverted to recover the original text.**

State-of-the-art inversion attacks achieve 82-92% semantic information leakage from embeddings alone — meaning an LLM evaluating the reconstructed text judges that it reveals the vast majority of the sensitive content from the original (Zhang et al., 2025). Earlier attacks recover 92% of 32-token inputs exactly (Morris et al., 2023). An

"anonymized" embedding database is, under current attack capabilities, largely recoverable. For the full evidence and analysis, see our companion article "Embeddings Are Not Private."

This means any architecture for cross-team knowledge discovery must treat embeddings with the same security posture as the raw data they represent. Simply removing the original text and keeping only the vectors provides no meaningful privacy.

The Semantic Vault: A Service That Knows But Cannot Tell

We propose an architecture modeled on how the industry already handles secrets: **a vault service.** The same way AWS KMS holds cryptographic keys and performs operations on them without ever disclosing the keys to callers, a Semantic Vault holds topic embeddings and performs similarity matching without ever disclosing the embeddings to anyone — including the service operator.

The core principle: **embeddings go in, only match indicators come out.** The vault computes similarity internally, and the only information that crosses the trust boundary is "you have potential matches" — never the embeddings, never the topics, never the identities of the matching parties (until both consent).

This isn't theoretical. The architectural building blocks are production-ready.

The Building Blocks

Trusted Execution Environments (TEEs) are hardware-isolated enclaves where code and data are protected from everything outside the enclave — including the operating system, the hypervisor, and the cloud provider's administrators. Intel SGX, AMD SEV, and AWS Nitro Enclaves all provide this capability. Fortanix Confidential AI and Anjuna Seaglass offer production platforms for running ML workloads inside TEEs today. Fortanix, for example, enables deployment of proprietary AI models into enterprise environments where the model weights remain encrypted and invisible, even to the infrastructure operator (Fortanix).

Federated vector search is a rapidly maturing field. FedVSE (VLDB 2025), built on the industrial vector database Milvus, deploys local vector databases at each data owner with a TEE-protected central coordinator that routes queries without accessing plaintext embeddings. The system was evaluated across six cloud servers with formal privacy guarantees: clients

learn nothing beyond query answers, and no local database can infer data from others.

Privacy-preserving record linkage is in production at scale. Datavant operates a large-scale privacy-preserving record linkage service in US healthcare, enabling matching across facilities without exposing patient data (Datavant). Their architecture irreversibly transforms data at the source before it leaves the originating organization, and uses a neutral intermediary that performs matching on transformed tokens without accessing the underlying records.

These building blocks range from production systems (Fortanix, Datavant) to peer-reviewed research validated in realistic deployments (FedVSE).

The Protocol

The Semantic Vault protocol has four phases: registration, matching, notification, and connection.

Phase 1: Registration

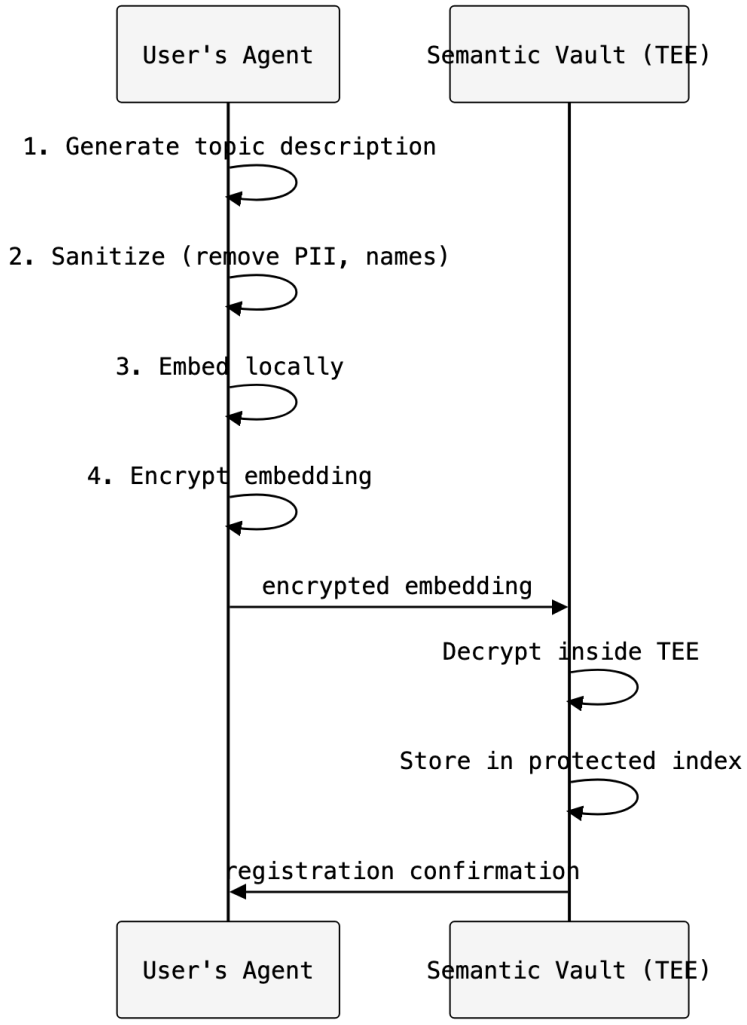
When a person (or their agent) identifies a topic they're working on, they register it with the vault:

The user's agent generates a topic description — an abstract characterization of the work domain, stripped of identifying details. Not "building a Stripe integration for Project Phoenix in the payments team" but "payment gateway integration patterns for recurring billing."

Pre-embedding sanitization removes any remaining PII, project names, or organizational identifiers. This happens on the client side, before anything leaves the user's environment.

The sanitized description is embedded using a standard embedding model. The resulting vector captures semantic meaning without identifying details.

The embedding is encrypted and sent to the vault — either a centralized TEE-protected service or a federated node. In the centralized model, the embedding is decrypted only inside the TEE. In the federated model, the embedding stays in the user's local vector database and is never transmitted.



Phase 2: Matching

The vault continuously (or periodically) computes similarity between all registered topic embeddings. This computation happens **entirely inside the trust boundary** — inside the TEE for the centralized model, or distributed across local nodes for the federated model.

Centralized matching (TEE vault): All embeddings are decrypted and indexed inside the enclave. Standard HNSW or IVF similarity search runs on plaintext vectors — but the plaintext exists only in TEE-protected memory, inaccessible to the operator. The vault identifies pairs of embeddings with similarity above a configurable threshold.

Federated matching (FedVSE-style): Each local node maintains its own vector index. The TEE-protected coordinator distributes similarity queries across nodes, collects local top-k results, and aggregates them — without ever seeing the embeddings themselves. This approach is preferable when organizations cannot send embeddings to a shared service, even a

TEE-protected one.

Both approaches produce the same output: a set of (anonymized_id_A, anonymized_id_B, similarity_score) tuples, where the IDs map to registered topics but not to people or teams.

Phase 3: Notification

When the vault identifies a match above threshold, it notifies each party **independently and anonymously**:

> "Someone in the organization is working on something semantically related to your registered topic [topic_hash]. Similarity confidence: high. Would you like to learn more?"

The notification reveals:

- That a match exists
- The similarity confidence level
- Optionally, the organizational unit of the other party (e.g., "Engineering" or "EMEA region") — configurable by policy

The notification does **not** reveal:

- Who the other party is
- What the other party's topic is
- Any details beyond the match indicator

Phase 4: Connection

Connection happens only with **bilateral consent** — both parties must independently opt in.

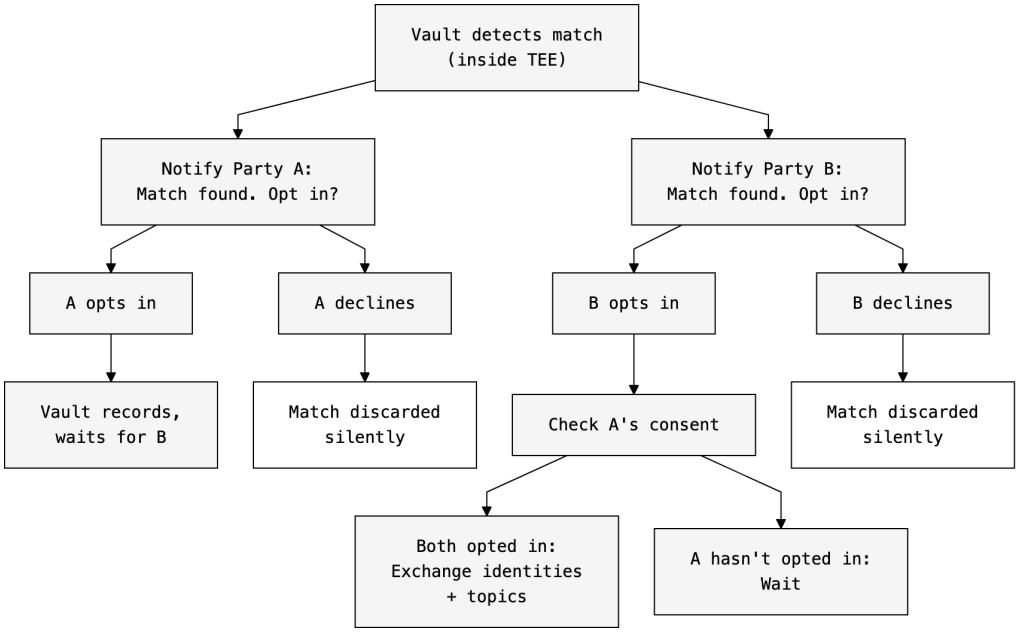
Party A opts in. The vault records consent but takes no action.

Party B opts in. The vault now has bilateral consent.

The vault releases to both parties: the other party's identity (name, team, contact), and the other party's sanitized topic description (the one they registered, stripped of identifiers).

Both parties can now communicate directly — the vault's role is complete.

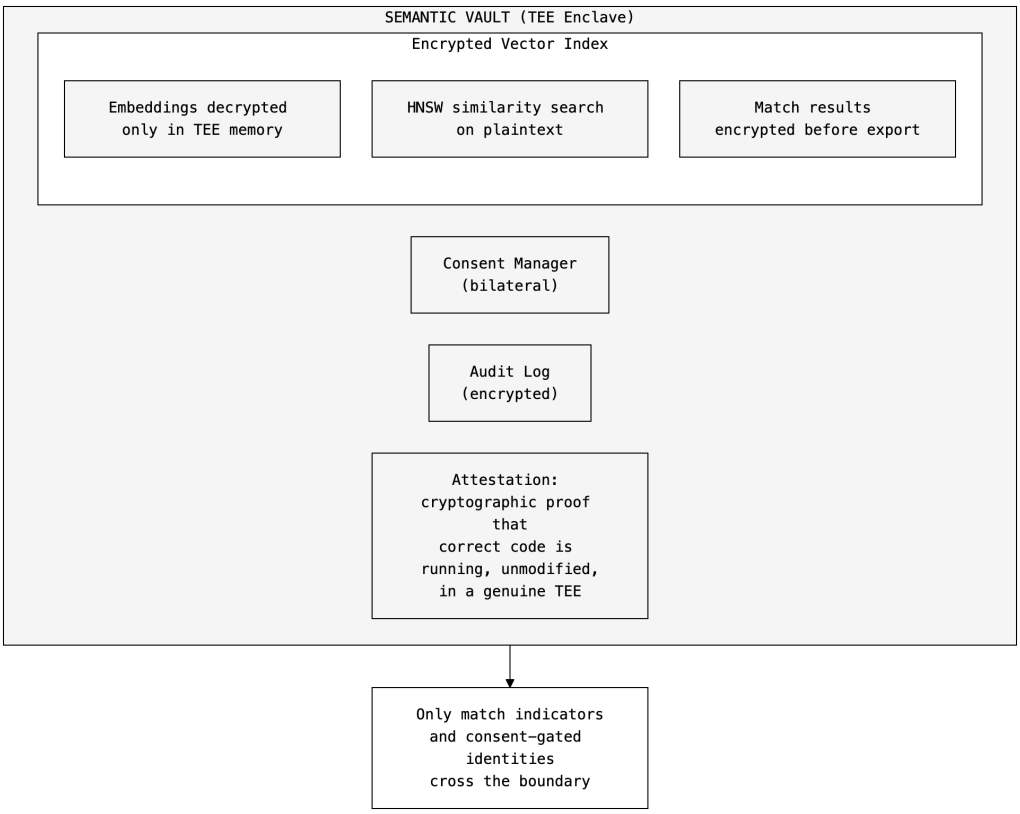
If either party declines, nothing happens. The other party is not informed that a match was found and declined. The vault deletes the match indicator.



Architecture Options

Option A: Centralized Semantic Vault

A single, TEE-protected service holds all topic embeddings for the organization.

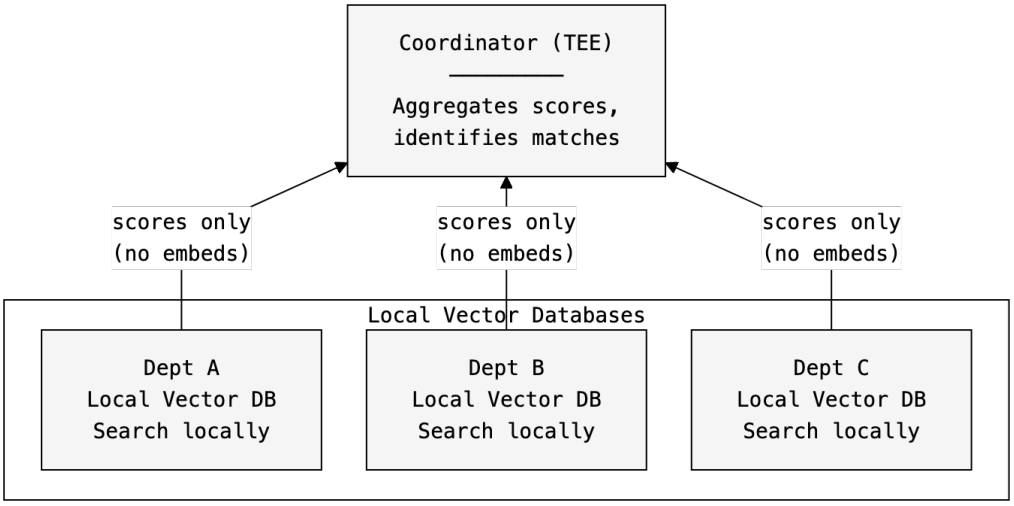


Best for: Single organizations where a central IT team can operate the vault. Simpler to deploy and manage. Lower latency (all embeddings co-located).

Production path: Deploy on AWS Nitro Enclaves or Azure Confidential Computing with Anjuna for enclave management. Use an off-the-shelf vector database (Milvus, Qdrant) inside the enclave. Fortanix provides the confidential AI runtime.

Option B: Federated Semantic Matching

Embeddings never leave their origin. A TEE-protected coordinator distributes queries and aggregates results.

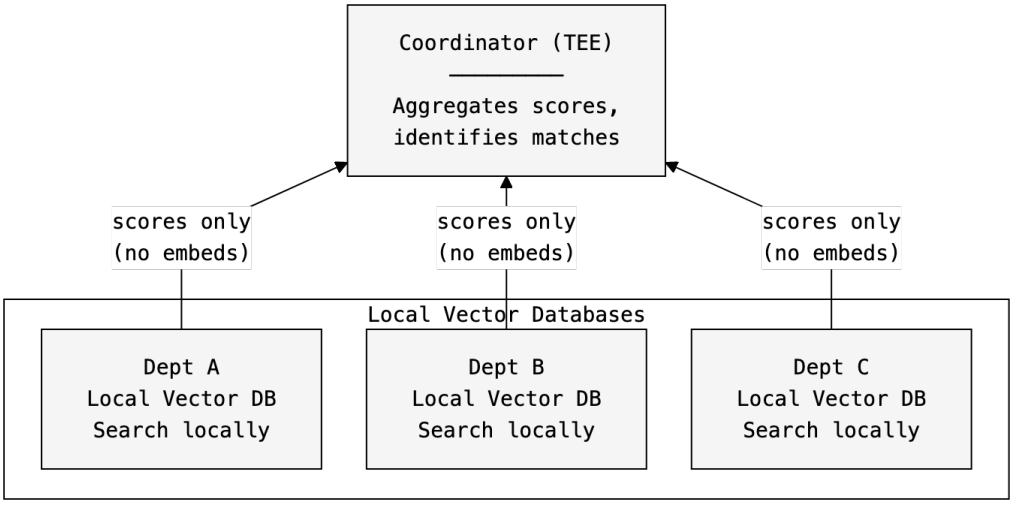


Best for: Multi-subsidiary organizations, cross-company collaborations, or environments where data residency requirements prevent centralization. Higher trust guarantee because embeddings physically stay with their owners.

Production path: FedVSE provides the architecture (built on Milvus, published at VLDB 2025). Each department runs a local Milvus instance. The coordinator runs inside an SGX or Nitro enclave. The FedVS extension (KDD 2025) adds attribute filtering for access-control-aware matching.

Option C: Hybrid

The centralized vault within a single organization, federated matching across organizational boundaries.



Best for: Organizations that want internal knowledge discovery with the option to extend to partner ecosystems later.

What Makes This Different from "Just Build a Search Engine"

A semantic vault is not an enterprise search engine with extra security. The architectural differences are fundamental:

	Enterprise Search	Semantic Vault
Who sees what	Everyone with access sees search results	Nobody sees anything until bilateral consent
Data flow	Data is centralized and indexed for search	Embeddings enter a trust boundary and never leave
Operator access	Search service operator can see the index	Vault operator cannot see the index (TEE enforcement)
Privacy model	Access control (you can or can't search)	Cryptographic isolation (the system itself can't expose data)
Output	Documents and snippets	Match indicators only — "a match exists"
Consent model	Implicit (if you have access, you can search)	Explicit bilateral (both parties must opt in)

The vault doesn't help you find documents. It helps you find people working on related things — and only if both people want to be found.

Practical Considerations

What "Topic Description" Means

The quality of the matching depends entirely on the quality of the topic descriptions that are registered. Too specific, and sanitization strips everything useful. Too vague, and everything matches everything.

In practice, a good topic description captures:

- **The domain**: "payment processing," "container orchestration," "regulatory compliance"
- **The problem type**: "integration," "migration," "performance optimization," "security audit"
- **The technical approach** (optionally): "event-driven architecture," "graph-based," "fine-tuning"

It does **not** include: project names, client names, people's names, internal codenames, deadlines, or business-specific identifiers.

The agent generating the topic description should be trained (or prompted) to produce these abstractions naturally. Over time, organizations will

develop conventions for how topics are described — similar to how tagging systems evolve.

Threshold Tuning

The similarity threshold determines how aggressively the vault surfaces matches. Too low, and everyone is notified about vaguely related work (notification fatigue). Too high, and only near-identical work is matched (missing valuable connections).

The threshold should be configurable per organizational unit and adjustable based on feedback. Early deployments should start with a high threshold (fewer, higher-confidence matches) and lower it as users develop trust in the system.

Scale

For a 10,000-person organization where each person registers an average of 3 active topics, the vault holds ~30,000 embeddings. This is small for modern vector databases — similarity search over 30,000 vectors completes in milliseconds on standard hardware, with TEE overhead adding modest latency.

The computational cost is dominated by embedding generation (at registration time) and periodic re-matching (when new topics are registered). Batch matching on a daily or hourly cycle is practical for any organization size.

The Datavant Precedent

Datavant's privacy-preserving record linkage service is the closest production analog to what we propose. They handle sensitive healthcare data at scale across thousands of facilities (Datavant). The key architectural lessons:

- **Transformation at the source**: Data is irreversibly transformed before it leaves the originating organization. For our vault, sanitization + embedding + encryption serves the same function.
- **Neutral intermediary**: The matching service operates on transformed representations, never raw data, and returns only match indicators. Our vault operates the same way — it holds only encrypted embeddings (decrypted only in TEE) and exports only match indicators.
- **Scoped matching**: Matching operates within defined project or organizational boundaries. For our vault, this maps to configurable matching scopes (within a department, across the organization, across partner companies).

The pattern has been validated in healthcare environments with stringent security requirements and regulatory scrutiny.

Limitations and Honest Caveats

TEE side-channel attacks exist. Intel SGX has documented side-channel vulnerabilities (speculative execution, cache timing). AMD SEV and AWS Nitro Enclaves use different architectures with different security properties, but no TEE is immune to all side-channel risks. TEE protection raises the bar significantly but is not absolute. Defense-in-depth (encrypting embeddings even inside the TEE) mitigates this.

The "small world" problem persists. If only one person in the organization works on "quantum-resistant key exchange for embedded medical devices," no amount of architectural protection hides who registered that topic when a match is found. The vault can detect this (single-registration topics) and decline to match them, but this limits utility for highly specialized work.

Topic description quality varies. If users register vague topics ("working on backend stuff"), the matching produces noise. If they register overly specific topics, sanitization strips the useful signal. This is a UX and organizational culture challenge as much as a technical one.

Bilateral consent has an asymmetry problem. If Party A opts in and Party B never responds, Party A knows a match was proposed (since they received the notification). They know someone is working on something related, even though they don't know who. In some contexts, even this metadata leakage is unacceptable.

Operational complexity. Running TEE infrastructure requires specialized expertise. Attestation, enclave lifecycle management, key management, and enclave-specific debugging are non-trivial operational burdens. Using managed services (Fortanix, Anjuna) reduces but doesn't eliminate this.

Where to Start

Phase 1: Single department pilot. Deploy a centralized vault (AWS Nitro Enclave) for one department. Use pre-embedding sanitization + TEE isolation. Skip the federated layer — it's unnecessary within a single department. Set the similarity threshold high. Measure: how many matches surface? How many lead to useful connections? What's the notification-to-connection ratio?

Phase 2: Cross-department expansion. Extend the vault to additional departments. Tune thresholds based on pilot feedback. Develop organizational conventions for topic descriptions. Integrate with personal agent dashboards (Pillar 1) so agents can register topics automatically based on the user's current work context.

Phase 3: Federated cross-organization. For organizations with subsidiaries or partner ecosystems, deploy the federated model. Each organization runs a local vault; a TEE-protected coordinator matches across boundaries. This phase requires data sharing agreements and cross-organizational trust frameworks.

References

- Fan, Z. et al. "FedVSE: Federated Vector Search Engine." PVLDB 18(12), 2025. ACM DL
- Zhu, Z. et al. "FedSQ: Secure Federated Vector Similarity Queries." PVLDB 17(12), 2024. VLDB
- Zhao, D. "FRAG: Federated Vector Database for Secure RAG." 2024. arXiv:2410.13272
- Fan, Z. et al. "FedVS: Federated Vector Similarity Search with Filters." KDD 2025. HufuDB
- Fortanix. "Confidential AI." fortanix.com
- Anjuna Security. "Seaglass Platform." anjuna.io
- Datavant. "Privacy-Preserving Record Linkage." datavant.com
- AWS. "LLM Inference Over Confidential Data Using Nitro Enclaves." 2024. AWS ML Blog
- Serengil, S. & Ozpinar, A. "Encrypted Vector Similarity with PHE." 2025. arXiv:2503.05850
- Broker-Based Private Matching. Springer, 2011. Link
- Snowflake. "Data Clean Rooms." docs.snowflake.com
- Cisco. "Securing Vector Databases." 2024. Cisco
- Morris, J.X. et al. "Text Embeddings Reveal (Almost) As Much As Text." EMNLP 2023. arXiv:2310.06816
- Zhang et al. "ZSinvert: Universal Zero-Shot Embedding Inversion." 2025. arXiv:2504.00147