# Semantic UUIDs: Max–Min Codebooks, Meaning–Sortable Identifiers, and Pointer–Driven LLMs

Draft for discussion — October 2025

**Abstract.** We propose a practical pipeline for *meaning■addressable* computation that pairs (i) a diversity■first lexical codebook of 4096 words for human I/O, (ii) a 128■bit *meaning■sortable* identifier (SUID) for storage and range queries, and (iii) a *semantic pointer protocol* for large language models (LLMs) that replaces verbose chains of thought with compact handles. The codebook is built by greedy max–min selection over normalized embeddings, yielding a speakable, error■checked base■4096 codec with semantic snapping at decode time. SUID projects embeddings to PCA(2), quantizes via a 2D Hilbert curve (60 bits), and packs a 128■bit key that sorts by meaning before time. We discuss multi■resolution prefixes (sCIDR), compositional variants (multi■SUID triangulation and PQ■style codes), and an LLM protocol that treats IDs as first■class pointers into retrieval packets.

# 1. Motivation

Conventional identifiers (UUIDv4/7, content hashes) ignore meaning; name■based IDs (UUIDv5) are deterministic but syntax■bound. Modern pipelines manipulate semantic objects (documents, arguments, tasks). We seek small, deterministic handles that are (a) human■friendly, (b) sortable by approximate meaning in a B■tree, and (c) actionable by LLMs as reusable pointers.

## Design goals:

- Speakable: robust voice/phone channel (base■4096 words, CRC/ECC).
- Sortable: B■tree friendly via locality■preserving Hilbert index.
- Deterministic: stable given frozen embedder + projection + version.
- Composable: multi■resolution prefixes; multi■handle triangulation; PQ codes.
- Toolable: LLMs emit/expand handles as first■class actions.

# 2. A 4096■Word Diversity■First Codebook

We build a speakable codebook by greedy farthest■point (max–min) selection over L2■normalized embeddings. Near■duplicate suppression (edit distance and high cosine similarity) improves robustness. Mapping integers [0..4095] to words yields a 12■bit symbol; 11 words encode a 128■bit payload plus a 4■bit CRC.

**Semantic decoding without exact words.** At decode■time, free■form phrases are embedded and snapped to the nearest codeword per slot, enabling ‘meaning■only’ channels.

# 3. A Meaning■Sortable 128■bit Identifier (SUID)

Fix an embedder and fit PCA(2) on the codebook vectors; record min/max. For any item: embed → PCA(2) → normalize to [0,1]^2 → quantize (30 bits/axis) → 2D Hilbert index (60 bits) → pack into 128 bits: 4 (version) **|** 60 (hilbert) **|** 48 (unix_ms) **|** 16 (rand). Lexicographic sort clusters by meaning first, then time.

**Multi■resolution prefixes (sCIDR).** Use prefix length ■≤60 to denote coarse■to■fine cells; range■scan on the prefix, then refine by cosine.

**Compositional variants.** (i) Multi■SUID triangulation (2–4 IDs, intersect neighborhoods). (ii) PQ■style keys (split vector into M subspaces, 12 bits each).

# 4. LLMs as Semantic Pointer Machines

Define a semantic pointer protocol (SPP): models *emit* compact handles when reasoning and *expand* them via tools into retrieval packets (neighbors, curated summary, citations). Chains of thought become chains of pointers, reducing tokens/latency while improving consistency.

**Training.** Supervise (text→handle) and (handle→expansion). Toolformer■style traces teach when to emit vs. expand.

# 5. Security, Ethics, and Versioning

SUID and codebook phrases are not cryptographic; use content hashes for security boundaries. Freeze embedder, PCA, and normalization; bump a 4■bit version on any change. Enforce authorization at expansion time; audit bias and consider domain■specific versions (e.g., v=2■law).

# 6. Evaluation Plan

- B■tree prefilter vs. ANN (HNSW/FAISS): recall@k, latency, cost.
- Dedup/canonicalization by sCIDR bucket vs. cosine thresholds.
- LLM token savings: replace boilerplate with handles; measure accuracy vs. CoT.
- Human I/O robustness: 11■word codec with CRC/ECC vs. base■2048.

# Appendix: Pseudocode

**Base■4096 (human mnemonic).**

```
UUID(128b) → 11 words (12b each) + CRC■4 nibble
bits = UUID_128 << 4 | crc4(UUID_bytes)
digits = base_4096(bits, length=11)
words = [ codebook[d] for d in digits ]
```

**SUID (meaning■sortable).**

```
v = embed(text); v = v / ||v||
z = normalize01( PCA2(v); mins, maxs )
(x,y) = quantize_30bits(z)
h = hilbert2D(x,y) # 60 bits
SUID = pack( ver=1 (4b), h (60b), unix_ms (48b), rand16 (16b) )
```

# References

- Gonzalez (1985) – Clustering to minimize the maximum intercluster distance.
- Arthur & Vassilvitskii (2007) – k■means++.
- Jégou et al. (2011) – Product Quantization for NN Search.
- Johnson et al. (2019) – Billion■scale similarity search with GPUs (FAISS).
- Kulesza & Taskar (2012) – Determinantal Point Processes.
- Butz (1969) – Hilbert curve integer algorithms.
- BIP■39 (2013) – Mnemonic code.
- RFC 4122 (2005) – UUID Namespace.