

# THE RECORD SHOP

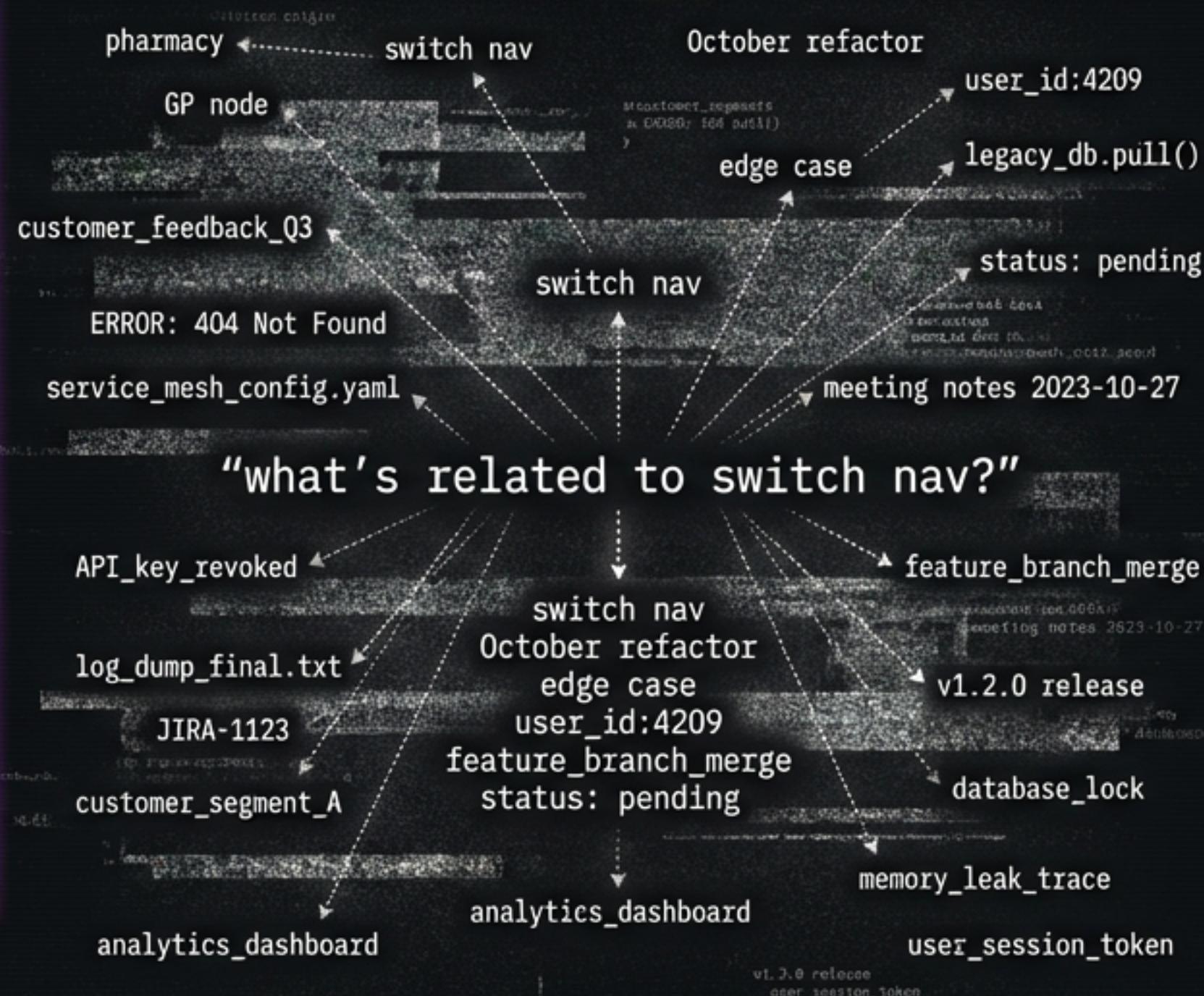
`float.dispatch // Active Context Protocol`

The Record Shop Architecture

```
bridge:: float/techcraft/active-context-protocol
semantic_anchor:: SC-ACTIVECONTEXT-CHIRP-RECORDSHOP
ctx:: 2025-11-29 @ 17:20
```

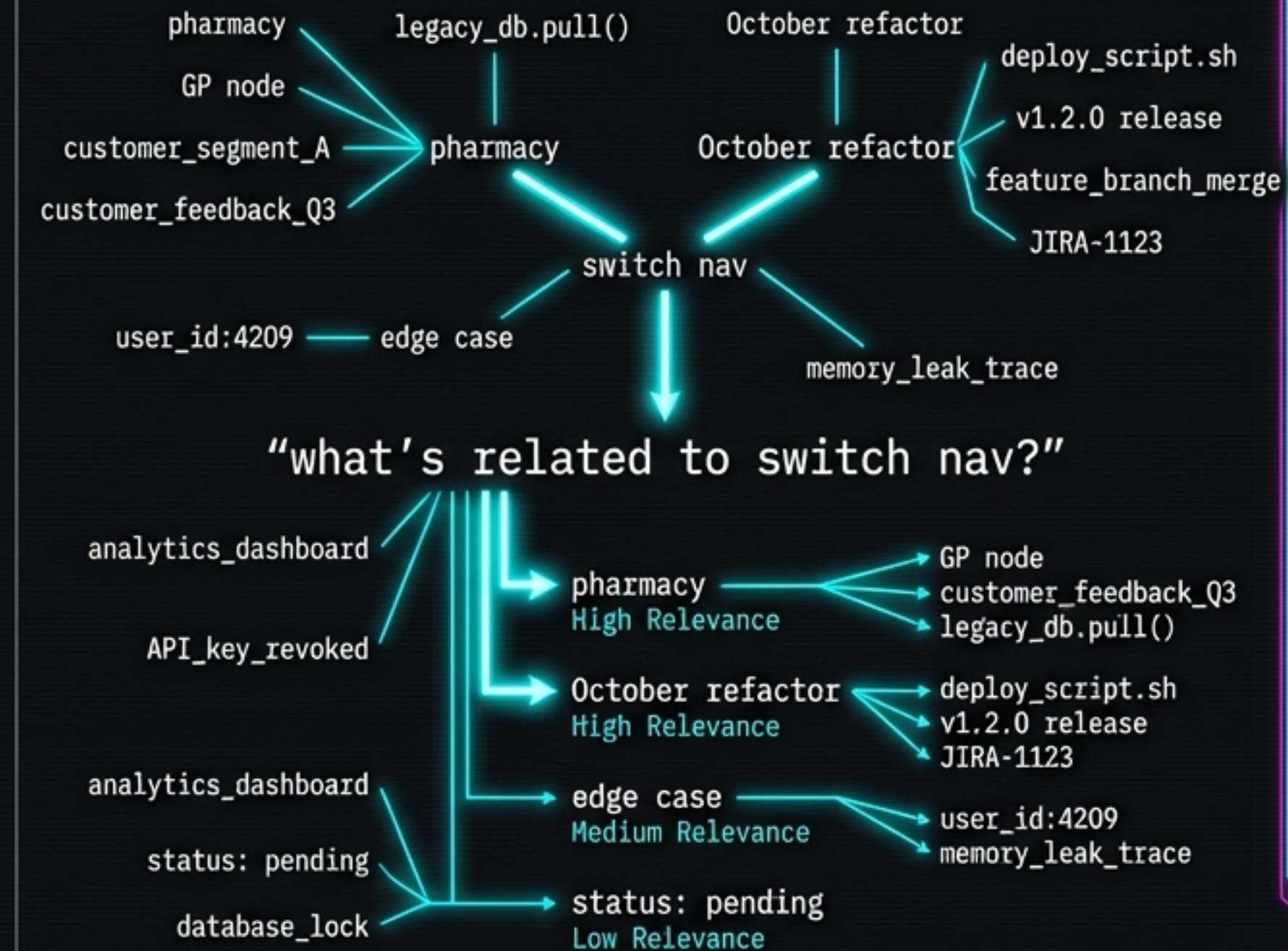
# The Problem :: Context is Fleeting

## 'GENERIC\_QUERY'

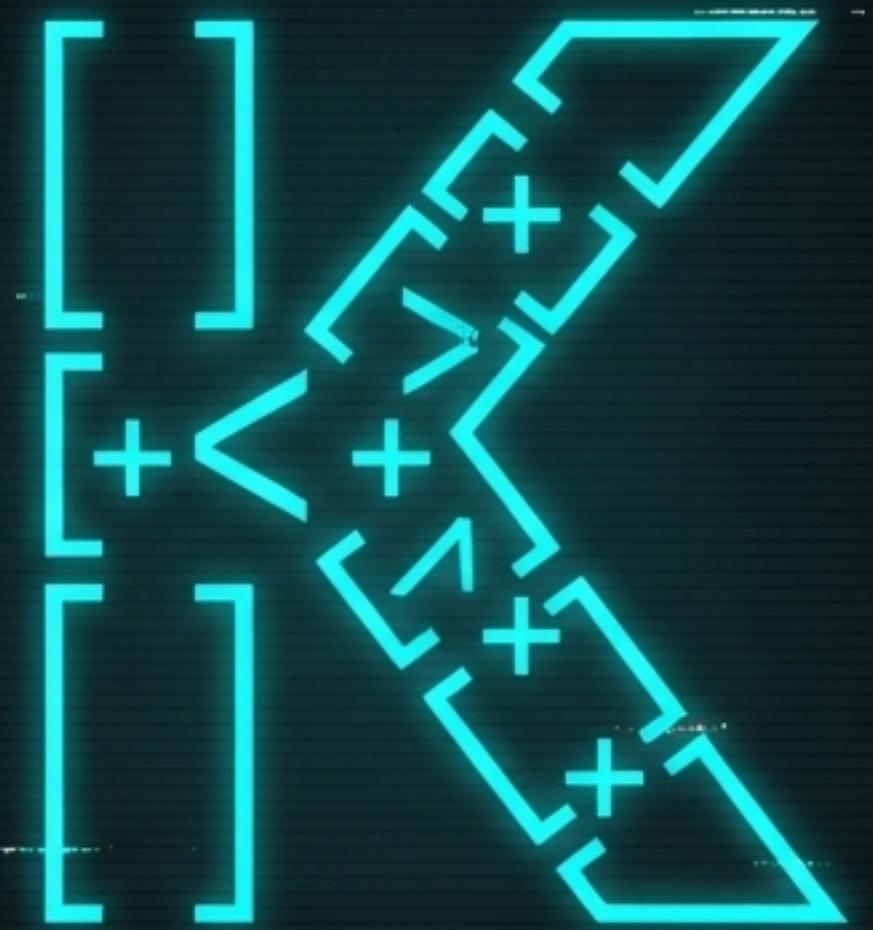


"Standard systems hear the question. A context-aware system knows *why* you're asking."

## 'SHAPED\_RESPONSE'



# The Solution :: A System with a Memory



‘KAREN [FRONT OF HOUSE]’

inventory.know()  
manifest.track()  
preview.generate()

**Knows where things ARE.**



‘EVNA [BACK OF HOUSE]’

connections.find()  
archaeology.run()  
patterns.surface()

**Knows how things CONNECT.**

## Chirps [seed\_context] vs. Requests [harvest\_context]

```
ceremony: LOW  
response_expected: FALSE  
purpose: "Just putting it in the air."
```

> ctx:: just realized HEIC connects to  
v0 work

```
ceremony: HIGHER  
response_expected: TRUE  
purpose: "Harvesting what chirps  
seeded."
```

> What's related to the HEIC  
implementation?

# The Pattern :: Context Accumulates

TIME	TYPE	CONTENT	CONTEXT.DELTA
10:00	chirp	deep in pharmacy GP node	+pharmacy +GP
10:15	chirp	edge case in switch nav	+switch +edge
10:30	chirp	connects to October refactor?	+October
10:45	REQUEST	what's related to switch nav?	

→ SHAPED RESPONSE  
[pharmacy weighting]  
[edge case awareness]  
[October hypothesis]

```
graph LR; P1([+pharmacy]) --- P2([+GP]); P3([+switch]) --- P4([+edge]); P5([+October]) --- R[→ SHAPED RESPONSE<br/>[pharmacy weighting]<br/>[edge case awareness]<br/>[October hypothesis]]
```

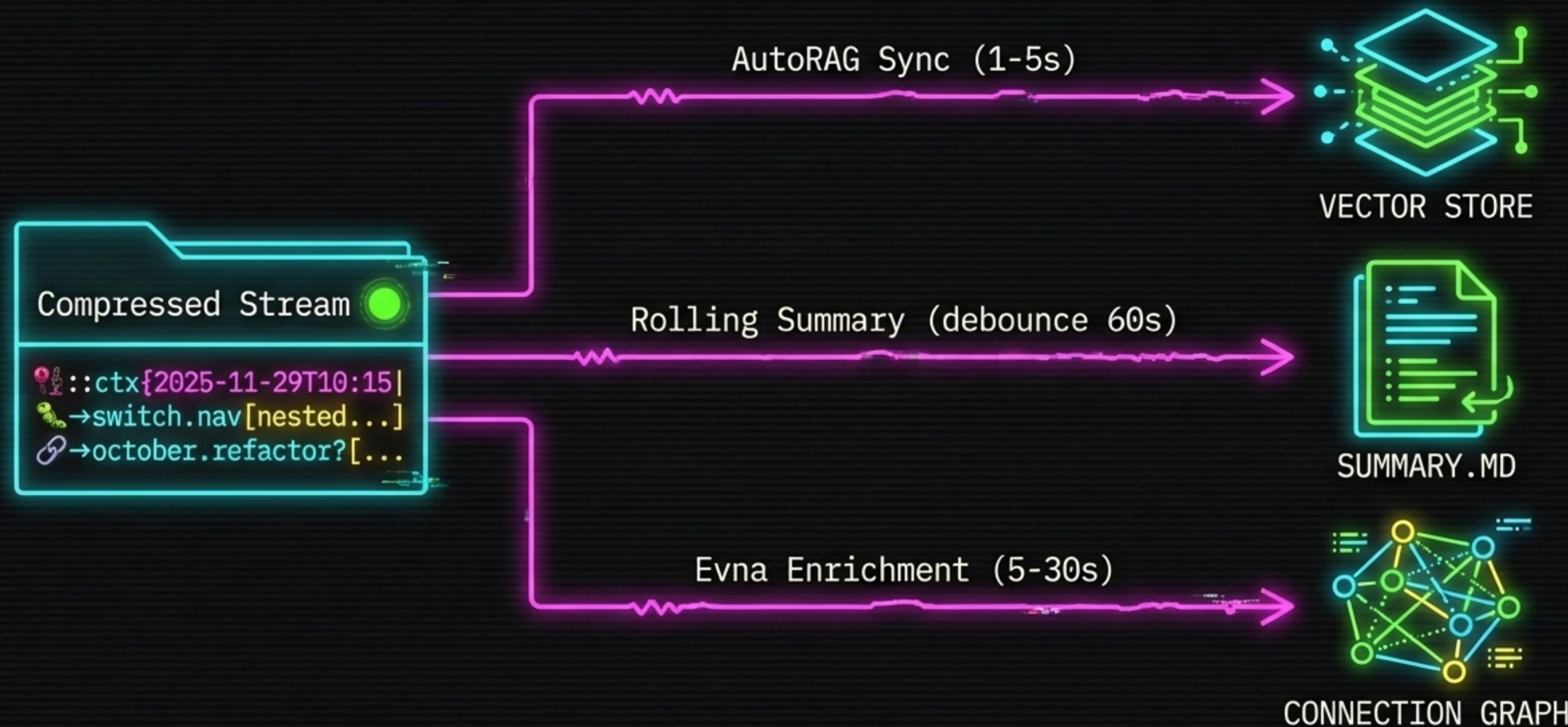
"Without chirps: generic response. With chirps: contextually precise."

# The Handshake :: Immediate, Dual-Write Capture



**SYNC: Immediate (< 100ms)**

# The Back Room :: Asynchronous Enrichment



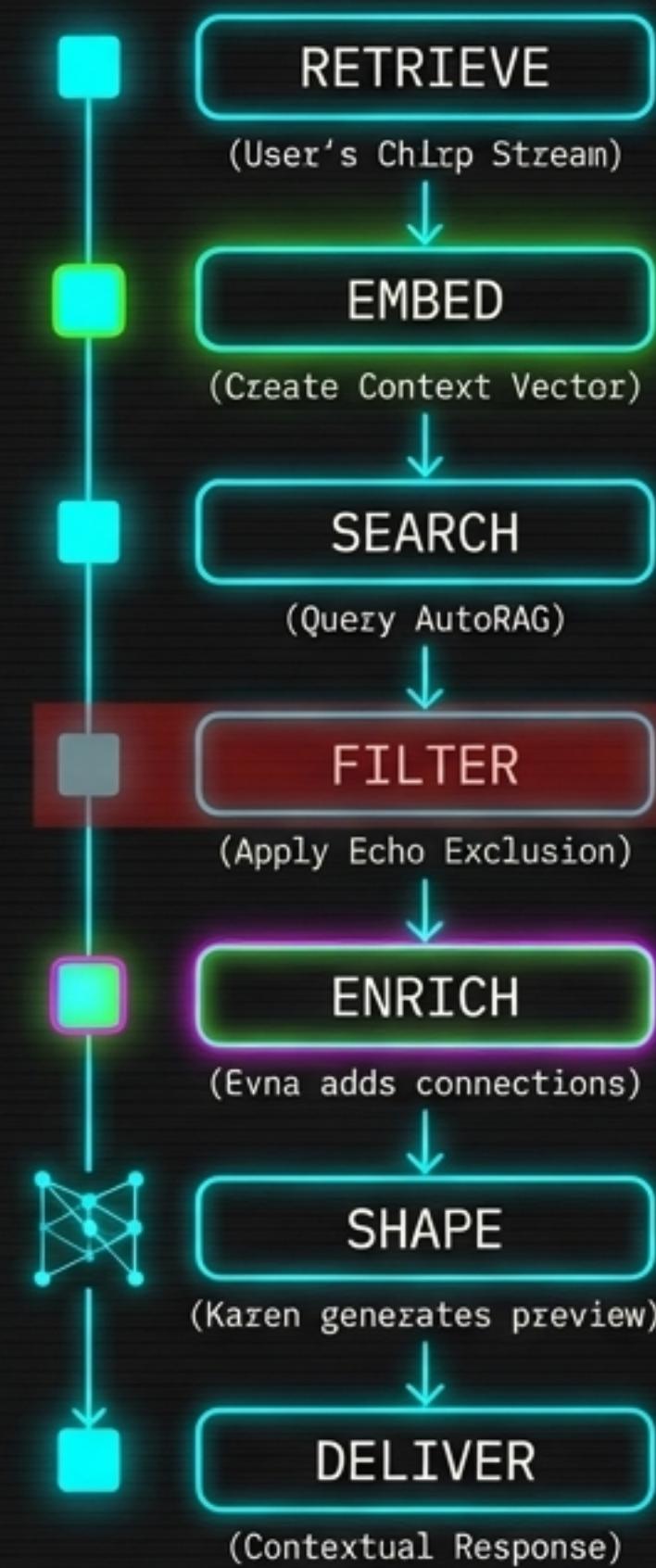
**“The user doesn’t wait. The system gets smarter in the background.”**

# Retrieval Logic :: No Echo, Shaped Response



—“You just said it. You know you said it.  
— Show me what \*changed\*.”

# The Shaping Flow :: A Request's Journey



# The Architecture :: Filesystem & Guarantees

## Directory Structure

```
/active-context/
└── streams/
    ├── {user}-full.log
    ├── {user}-compressed.log
    └── {user}-summary.md
└── enriched/
    └── {timestamp}-enriched.json
└── feeds/
    └── {user}-feed.json
```

## Timing Guarantees

```
synchronous_operations:
    dual_write: < 100ms
    ack_to_user: < 150ms
asynchronous_operations:
    autorag_sync: 1-5s
    summary_update: debounce 60s
echo_window:
    default: 30 minutes
```

# The Interface :: API Surface

**POST** /active-context/chirp

**body**: {  
  content: string,  
  markers: object  
}

**response**: {  
  ack: true,  
  hash: string  
}

**latency**: < 150ms

**POST** /active-context/request

**body**: {  
  query: string  
}

**response**: {  
  results: array,  
  context\_applied: object  
}

**latency**: < 2s

**GET** /active-context/feed

**params**: {  
  user\_id,  
  limit,  
  exclude\_self  
}

**response**: {  
  items: array,  
  context\_vector\_applied:  
  bool  
}

# The Language :: FreakAST Compressed Reference



**::dual.op**

Simultaneous operation across two parallel data streams.



**::chirp**

Lightweight, rapid data packet transmission or notification.



**::request**

Standard API query or data retrieval command.



**::dual.write**

Concurrent data writing to primary and secondary storage.



**::echo.exclusion**

Mechanism to prevent data feedback loops or duplicates.



**::feed.shape**

Defines the structure and format of data delivery streams.



**::loop**

Controlled iterative process or continuous data cycle.

The Loop :: chirp → context → request → SHAPED RESPONSE



The train is always HERE. The question is velocity.