Just keep Swimming

:Benchmark OSS Agent Memory on Kubernetes with OpenTelemetry



Henrik Rexed · Cloud Native Advocate

Henrik Rexed

Cloud Native Advocate, CNCF Ambassador

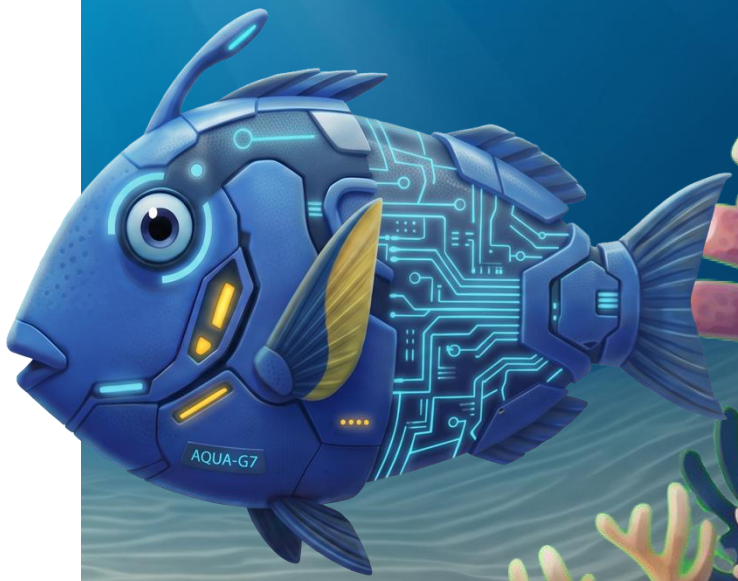


- **15+ years of Performance engineering**
- **Owner of IsitObservable**

  Producer of Perfbytes

Dori the Agent having amnesia

- Without memory :
  - Context Loss
    - The context is lost after each conversation
  - Repeated tool work
    - Keep reusing tooling without considering previous conversation
  - Never consider previous interactions
    - Every conversation is treated as the first one.

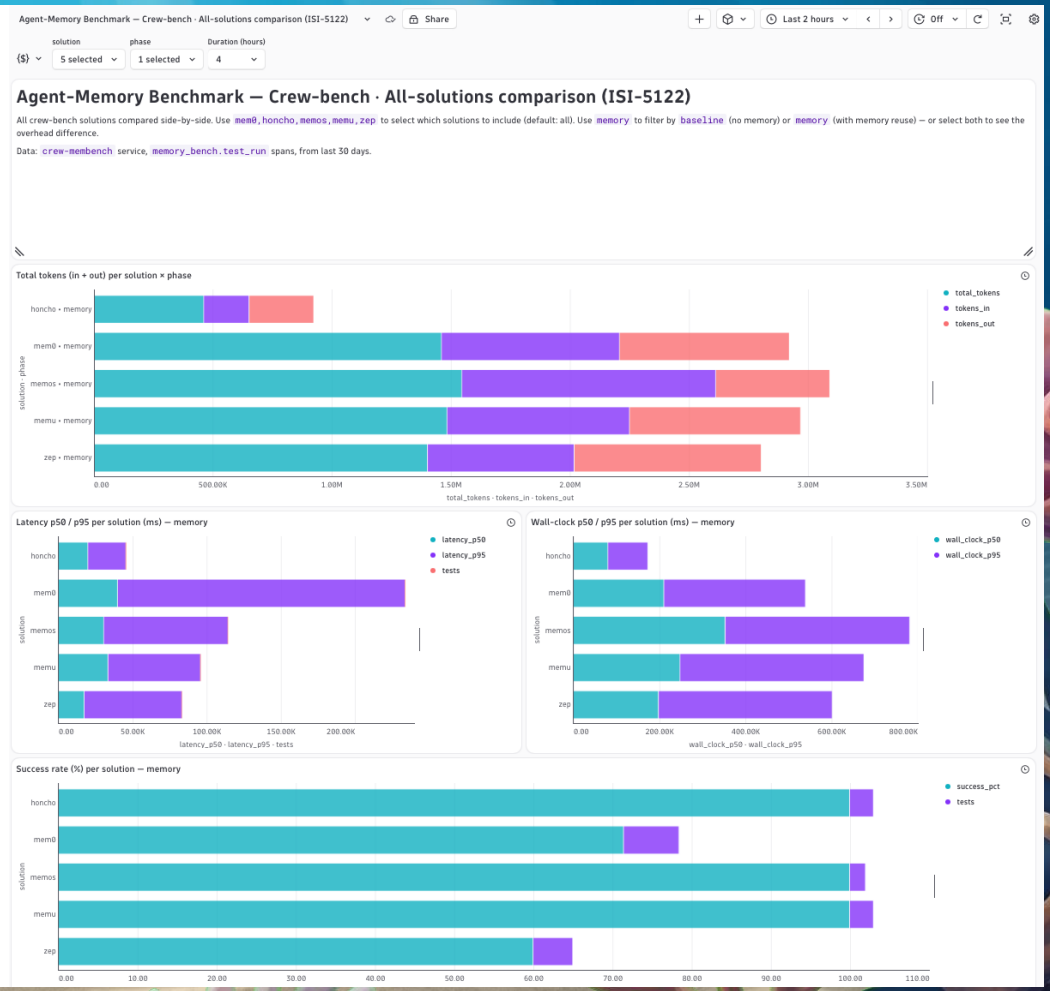


The promise =- Memory server



What are we going to learn

- The memory solutions architecture
- The various solution :
  - Memos
  - Mempalace
  - Zep
  - Mem0
  - MemKeys
- The benchmark environment
- The benchmark Result



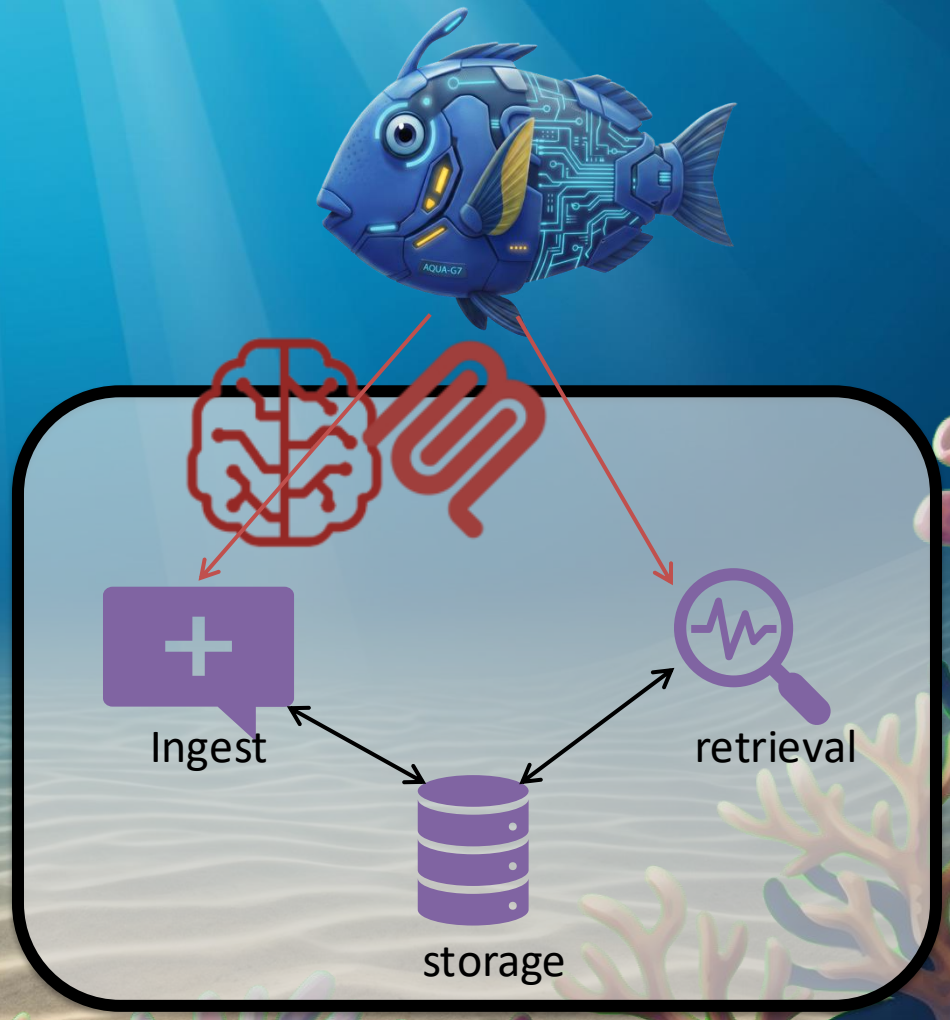

The memory Solutions

The various architecture

Vector + graph storage	• Queryable • Semantic research	• time aware graph	• Vector storage	• List of keys • Vector + GraphStorage
memory OS	Knowledge Graph	Temporal KG	Embedding Layer	Temporal KG with keys

Expected components

- Most of the Memory has dedicated components
  - Ingest to new memory
  - The Storage to store the new memory in to vector database (usually through embedding)
  - A retrieval letting the agent search, the information





The various solution to remember

The desire



MemOS



MemPalace



Zep



Memkeyw

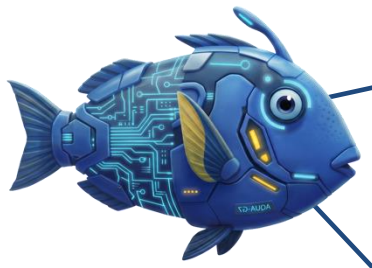


Mem0

Mem0

- Apache v2 license
- The most integrated solution

(AWS, CrewAI , Flowise, Langflow..Etc)



Extract facts
LLM, every add

Reconcile
Add, update, delete

Graph store
Optional, pluggable

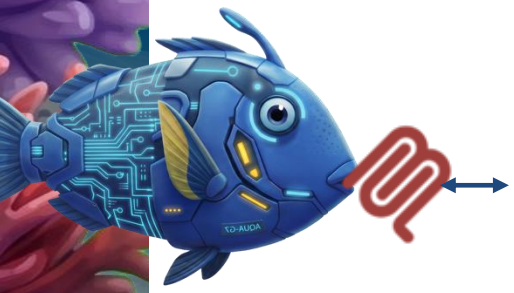
Embed + search
nomic-embed-text

Vector store
Pluggable
Qdrant by default
Our run: in-memory

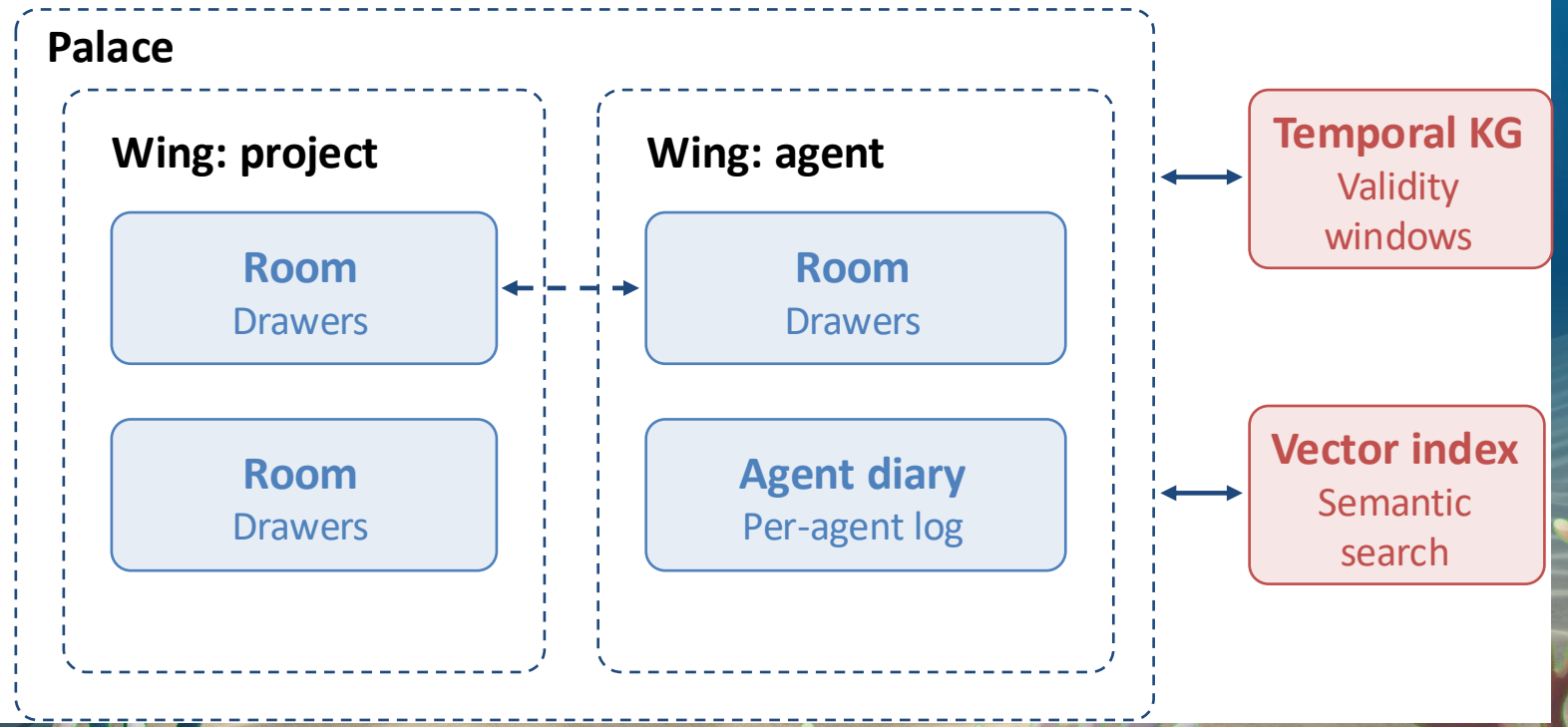
Mempalace



- Mit license
- Agent interacts with MCP

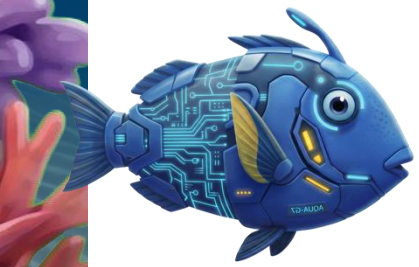






Zep(graphiti





Episodes
Raw messages

Extract
Entities + edges

Hybrid retrieval
Vector + BM25 + graph

Temporal knowledge graph

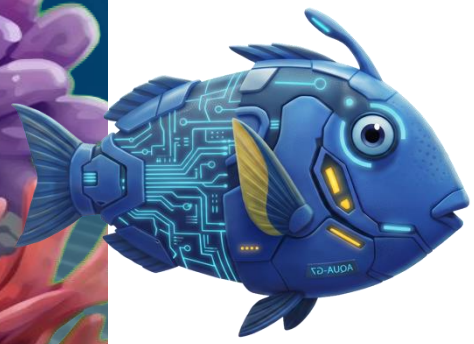
Dana works at

Acme
Invalidated at t2

Globex
Valid from t2

t1 t2 now

MemOs



MOS (memory OS)

MemCube

mem_reader
Extraction LLM

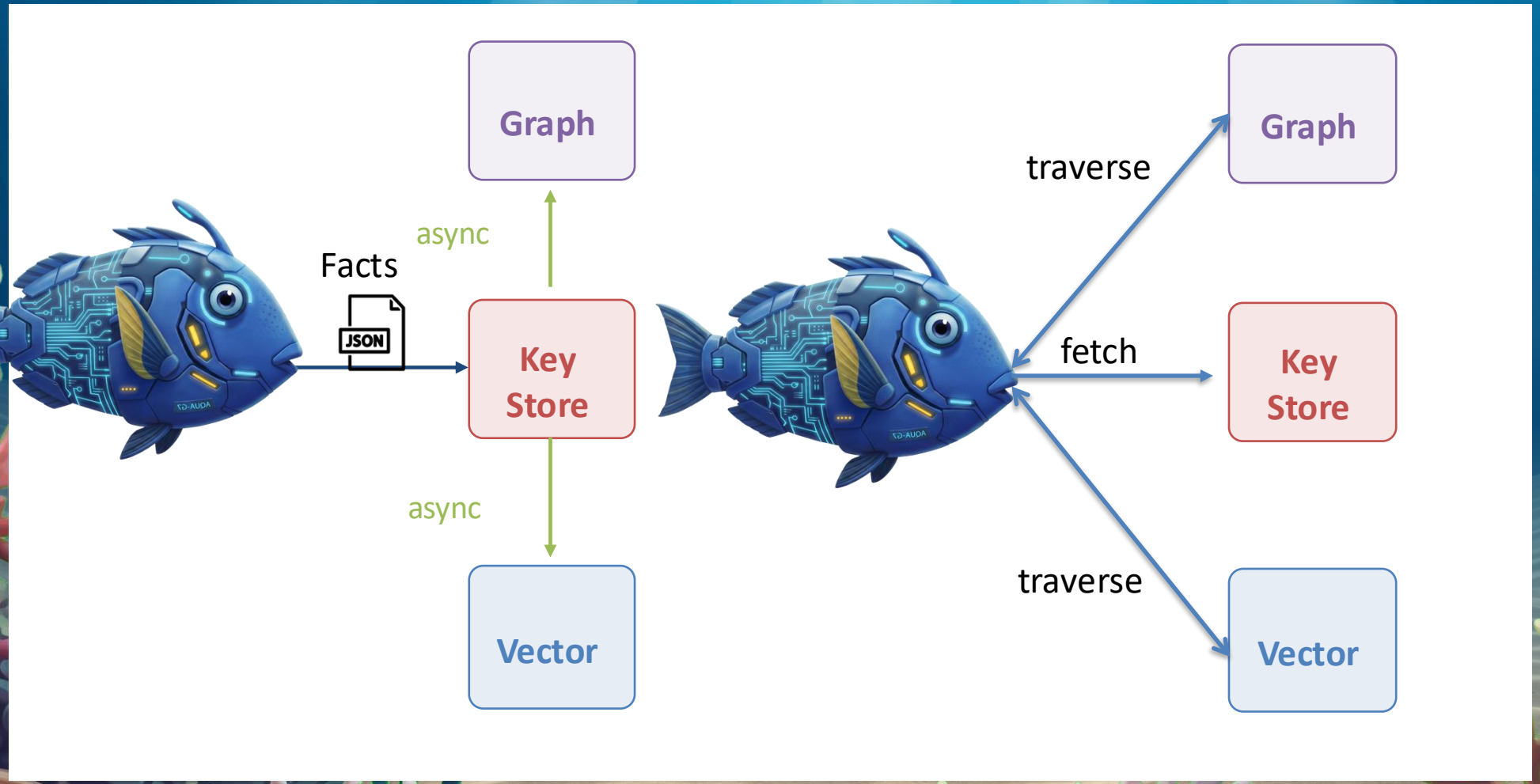
Retriever
Fine search mode

TreeTextMemory

Graph: Neo4j

Vector: Qdrant

MemKeys


Graph
async
Facts
JSON
Key Store
async
Vector
traverse
Graph
fetch
Key Store
traverse
Vector

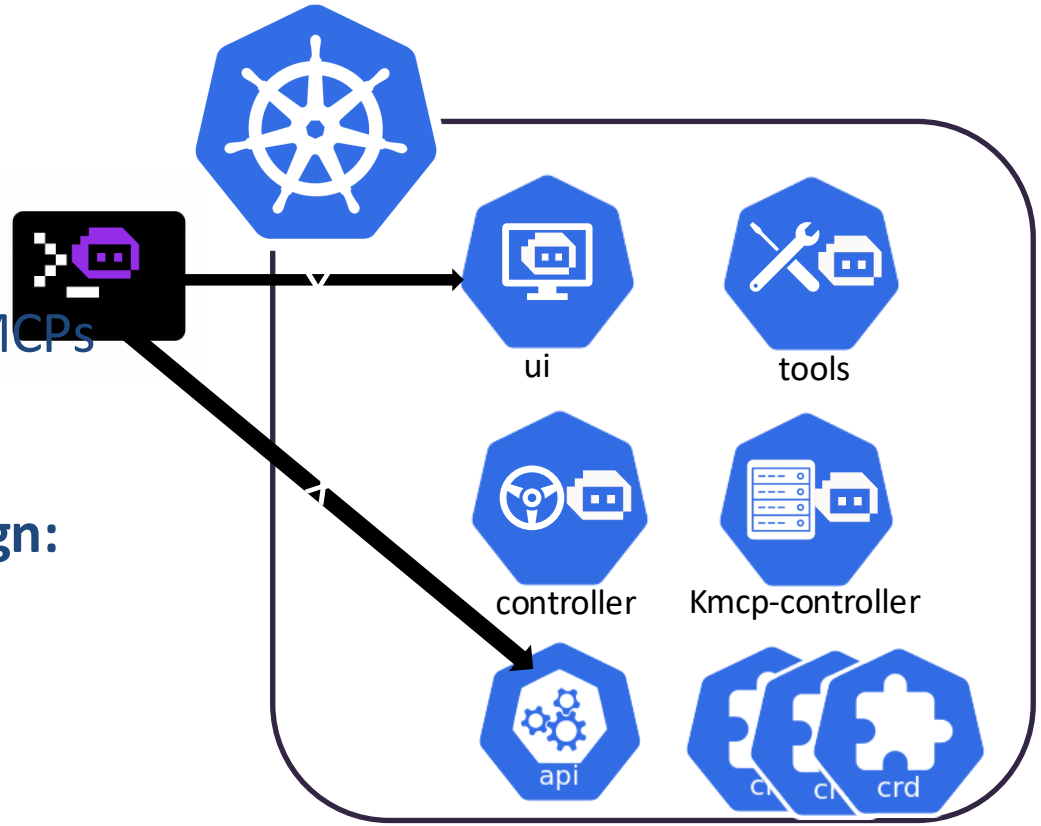
Summary

backend	Technology	Webui	MCP
memos	Neo4j graph + Qdrant vector + mem_reader extraction LLM;	✓	✓
zep	temporal knowledge graph over Postgres+pgvector;	✗	✗
mempalace	vector + knowledge graph	✗	✓
mem0	pure vector; in-process in-memory Qdrant + LLM	✓	✓
memkeys	temporal keyed + temporal fact graph	✓	✓

The Benchmark environment

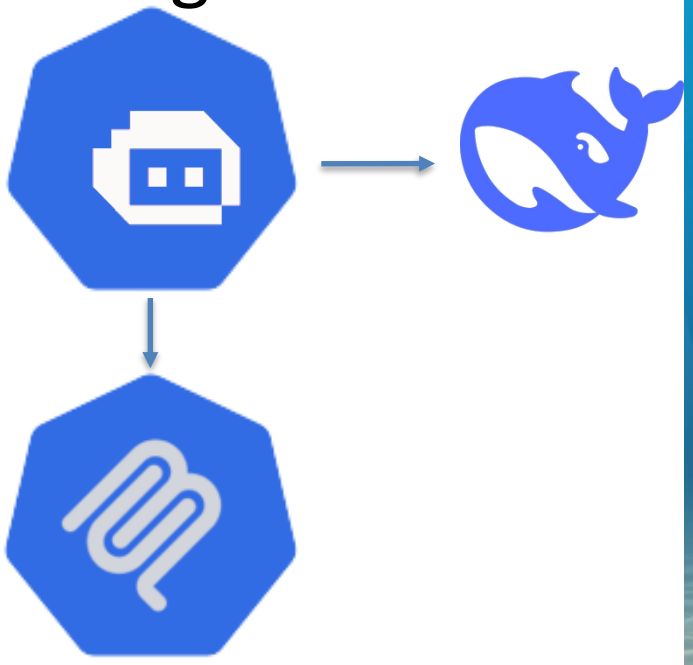
Kagent

- **Kagent provides a CLI for easy onboarding**
- **it will install several core components :**
  - The Kagent ui
  - The default tooling
  - The controllers managing the CRDS and MCPs
- **it provides several CRD helping us in the design:**
  - RemoteMCPserver
  - ModelConfig
  - Agent



The reality

- Benchmark was done using:
  - Kagent

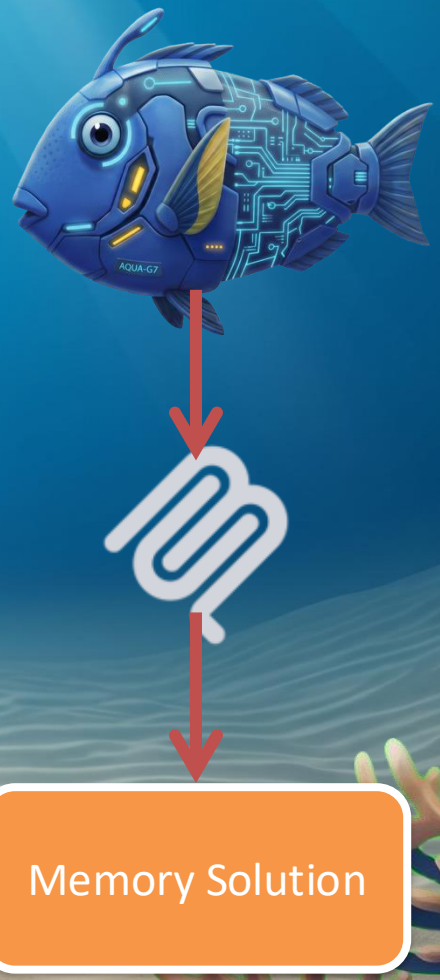


- 2 sets of tests:
  - Single Agent
  - Crew orchestration
- Kagent has predefined memory solution



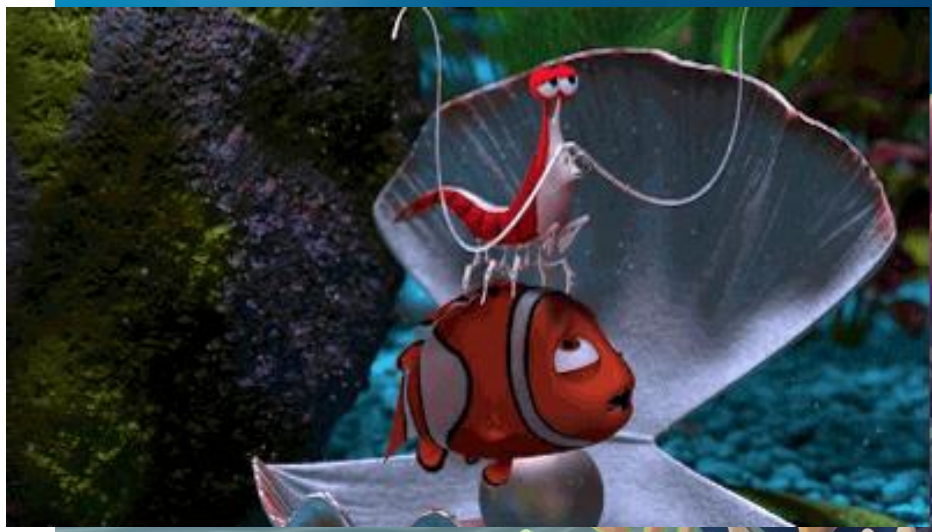
The solution to connect crewai

- One MCP server connecting the Agent to the memory solution is required.
- The tools expose are ：
  - Add
  - search
  - Get
  - Update
  - Delete
  - Stats



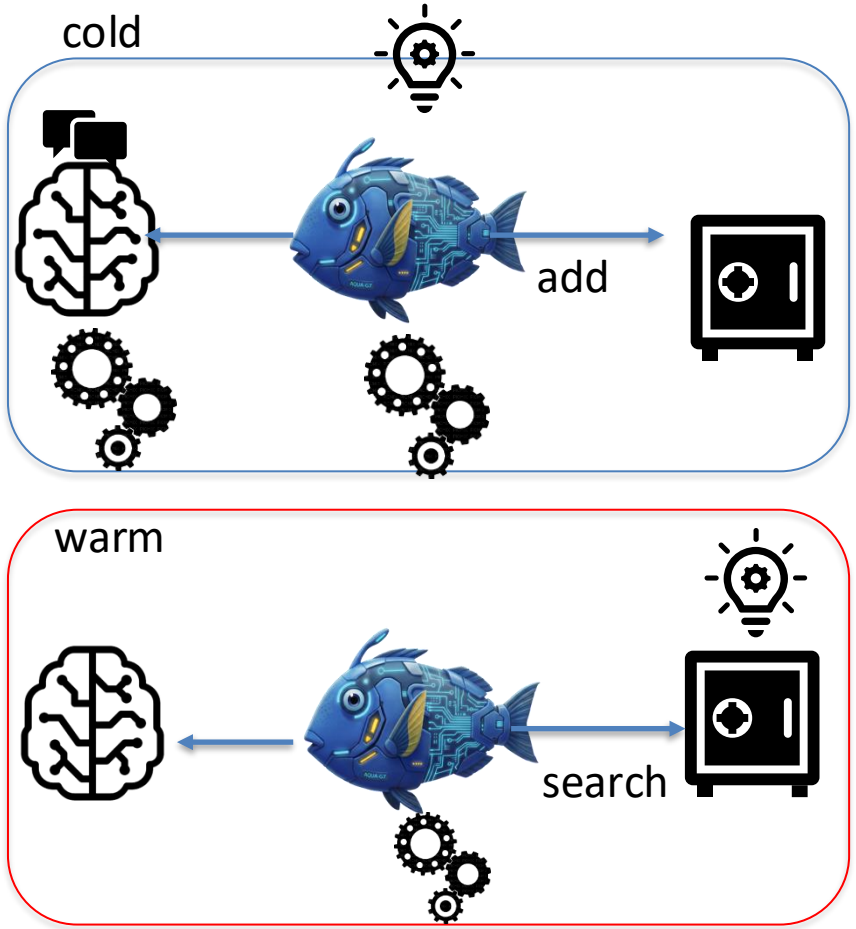
We need kpi to compare

- For each solution Add Opentelemetry instrumentation
- Each memory solution expose spans :
  - Add
  - Search
  - Recall
- Collect stats on Memory operations



Process to measure the memory

- Each Test require:
  - 1 Cold run: Without memory
  - 1 Warm run: With Memory
- The efficiency is measured by :
  - Delta= Warm - Cold



The scenarios 1

- 1 Single SRE agent running investigation



K8s failures

crashloop

imagepull

oomkill

pending

svc-conn

Gateway API issues

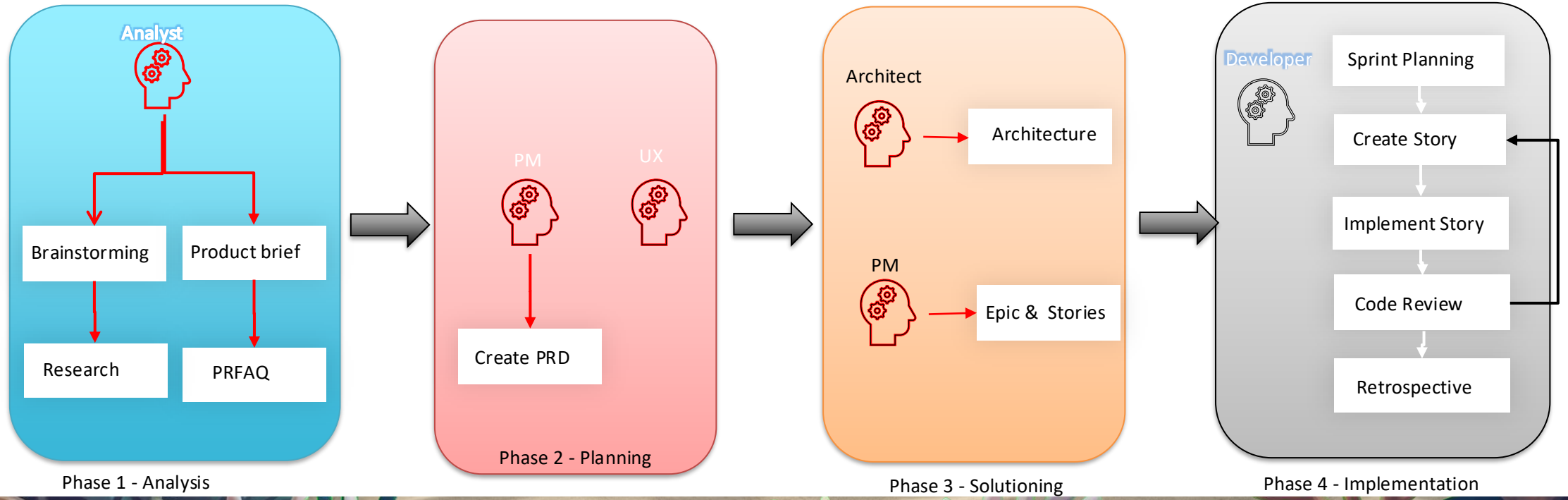
httproute

gateway

grpcroute

The scenarios 2

- CrewAI team building technical solution using the BMAD methodology.



The scenarios 2

- Crewai Team building and expense-approval app

- 1st test: Introduce error during team execution
  - Run without memory
  - Run memory

- 2sd test: Change facts
  - Run with memory
  - Introduce a change
  - Ask the team about architecture decision

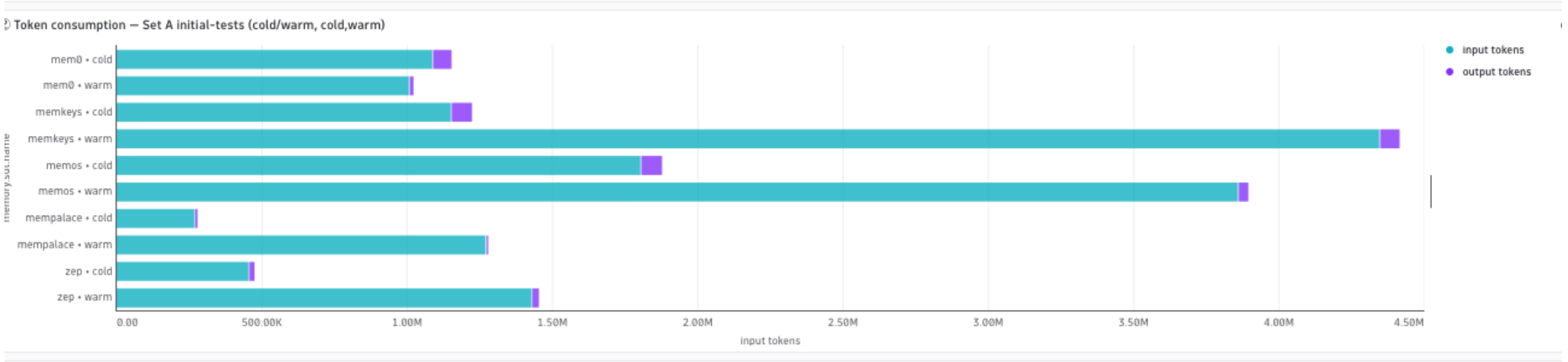


The Benchmark result

Scenario 1


Memory working-set (MiB) by solution — SINGLE-AGENT runs (last 30d)
graphiti
memos
zep
honcho
mem0
letta
mempalace
memkeys
memobase
solution
0.00
50.00
100.00
150.00
200.00
250.00
300.00
350.00
mem_mib
CPU (millicores) by solution — SINGLE-AGENT runs (last 30d)
memkeys
zep
graphiti
memobase
memos
honcho
mempalace
letta
mem0
solution
0.00
5.00
10.00
15.00
20.00
25.00
30.00
35.00
40.00
45.00
50.00
55.00
60.00
millicores

Token


Token consumption — Set A initial-tests (cold/warm, cold,warm)
mem0 • cold
mem0 • warm
memkeys • cold
memkeys • warm
memos • cold
memos • warm
mempalace • cold
mempalace • warm
zep • cold
zep • warm
0.00
500.00K
1.00M
1.50M
2.00M
2.50M
3.00M
3.50M
4.00M
4.50M
input tokens
input tokens
output tokens

Scenario 1

backend	cold	warm	Δ	payoff
memos	0.525	0.565	**+0.040**	✓
zep	0.308	0.425	**+0.117**	✓
mempalace	0.333	0.358	**+0.025**	✓
mem0	0.517	0.433	**-0.083**	✗
memkeys	0.405	0.343	**-0.062**	✗

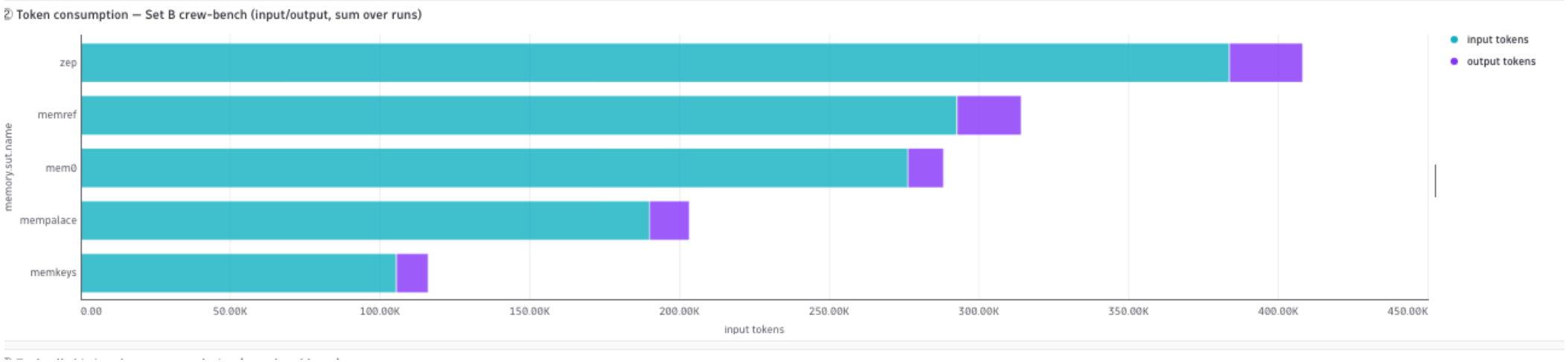
Scenario 2

backend	build	resume	restart	factchange
zep	0.33	0.52	0.00	0.42
mempalace	0.00	0.33	0.00	0.42
memkeys *	0.02	0.33	0.33	**1.00**
mem0	0.00	0.00	0.00	0.50
memref	0.00	0.00	0.00	0.08

Scenario 2


Memory working-set (MiB) by solution — CREW run (2026-10-27)
memos
graphiti
zep
mempalace
memkeys
solution
0.00
50.00
100.00
150.00
200.00
250.00
300.00
350.00
mem_mib
CPU (millicores) by solution — CREW run (2026-10-27)
memkeys
zep
memos
graphiti
mempalace
solution
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
20.00
22.00
millicores

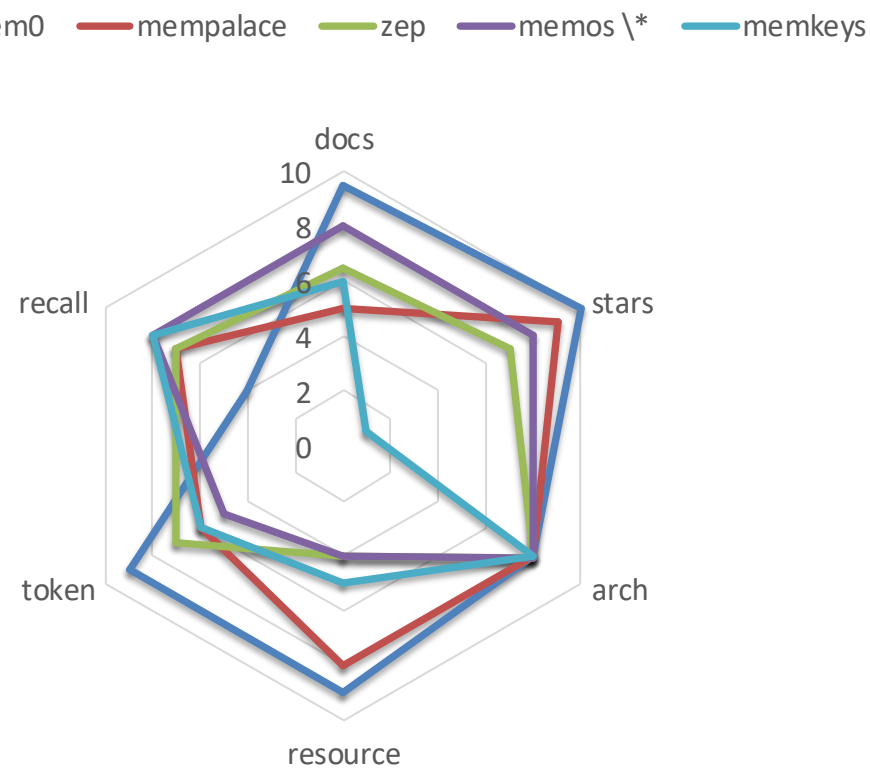
Token


Token consumption — Set B crew-bench (input/output, sum over runs)
input tokens
output tokens
memory.sut.name
zep
memref
mem0
mempalace
memkeys
0.00
50.00K
100.00K
150.00K
200.00K
250.00K
300.00K
350.00K
400.00K
450.00K
input tokens

Memory Operation



Conclusion


mem0
mempalace
zep
memos *
memkeys
docs
10
8
6
4
2
0
stars
arch
resource
token
recall

Is It Observable

- **Looking for educational content on Observability?**
  Checkout the YouTube Channel :











