

Nexus

Multi-agent RAG Research
Platform



Abdel Rahman Ibrahim

Background

Background

Understanding LLM Queries

User Prompt: You provide a question/instruction

Pre-trained Knowledge: Knowledge is encoded in the model's parameters from training

Response Generation: Predicts next tokens given using attention and learned patterns

Limitations: Knowledge cut-off, access to external data, hallucinations

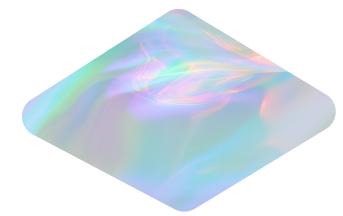


Background

Retrieval-Augmented Generation

RAG systems augment LLM capabilities by connecting them to external knowledge bases.

- Dynamic knowledge updates without retraining
- Grounds responses in source artifacts, reducing hallucination
- Can provide precise citations based on indexed documents



Ingestion & Indexing: Tokens are chunked, and dense vector embeddings are generated

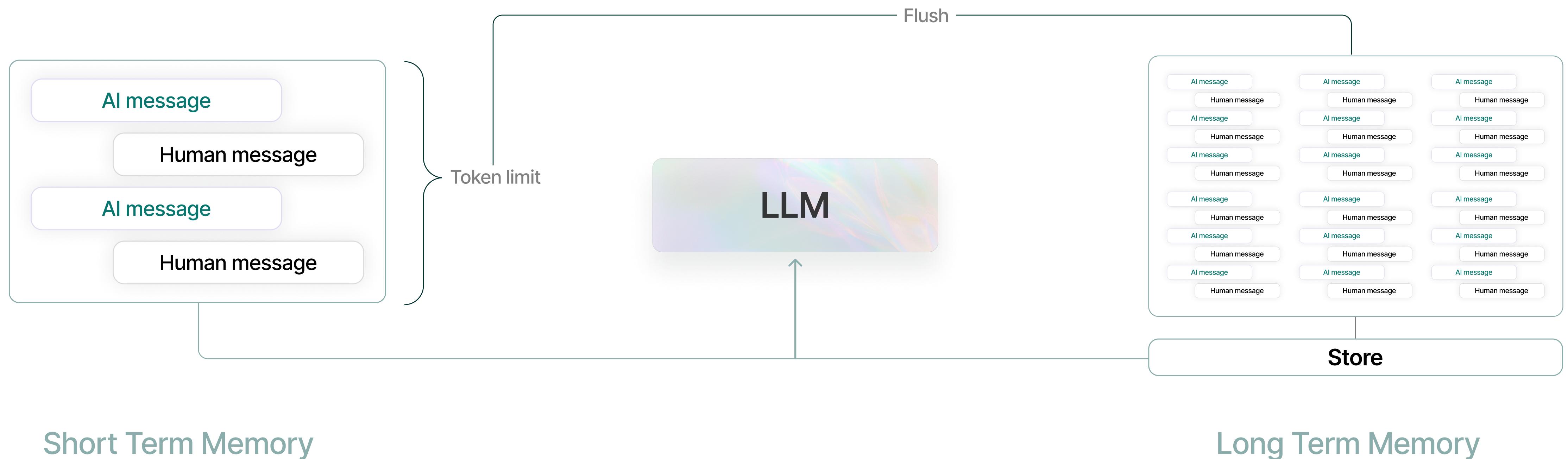
Retrieval: Semantic similarity search is performed to retrieve top-k similar chunks

Generation: Responses are generated with the retrieved information in the context

Background

Memory

Retaining information throughout conversations and interactions is achieved via short and long-term memory solutions



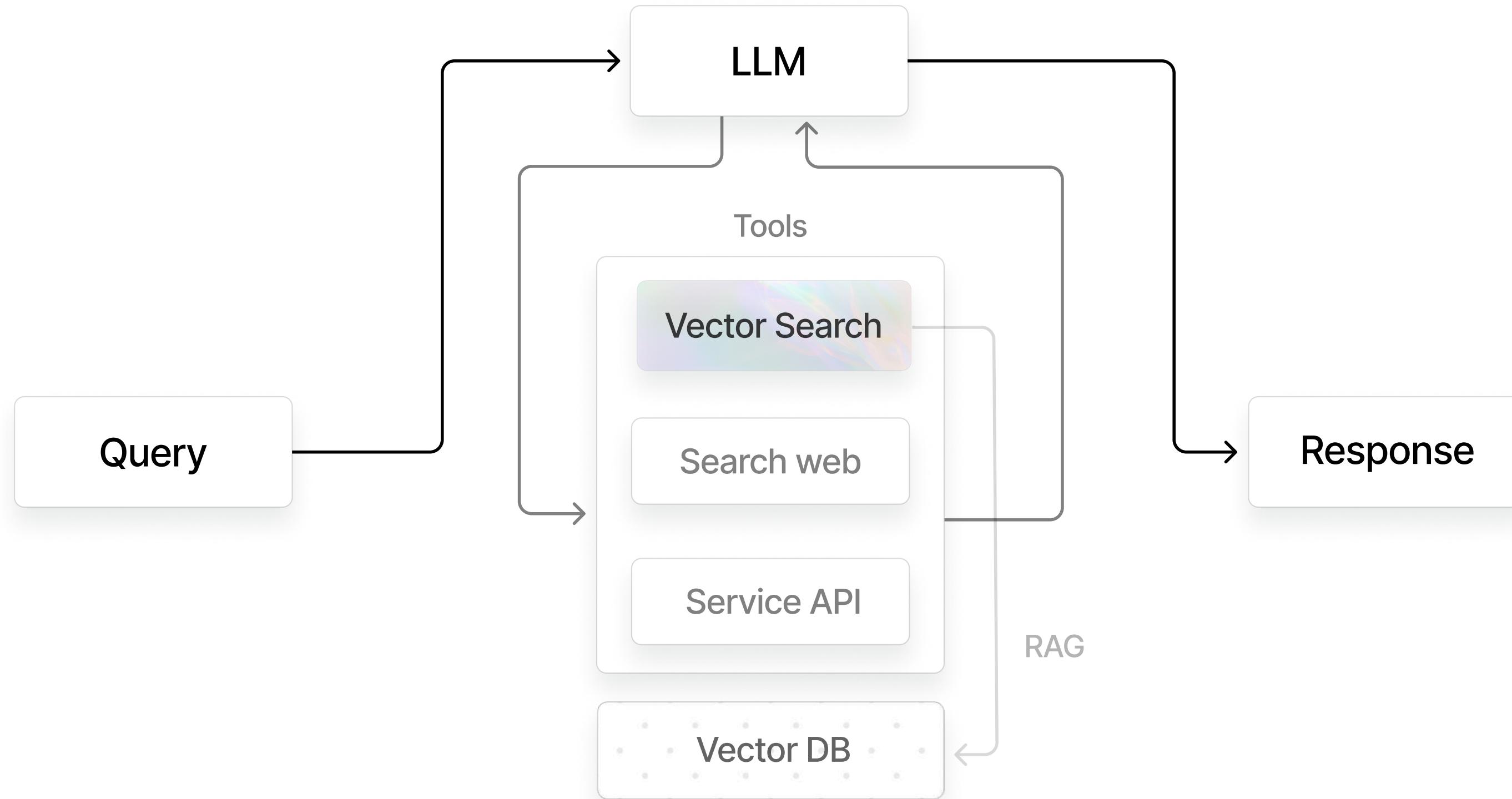
Short Term Memory

Long Term Memory

Background

Agentic AI Systems

RAG systems augment LLM capabilities by connecting them to external knowledge bases.



Putting this all together

The convergence of LLMs, RAG, and agentic systems creates an opportunity for a new class of research tools.

Systems can...

- Understand natural language queries & autonomously retrieve relevant information
- Synthesize insights across multiple documents with verifiable and cited answers
- Maintain conversational context for iterative exploration
- Adapt their search strategies based on query type

Nexus

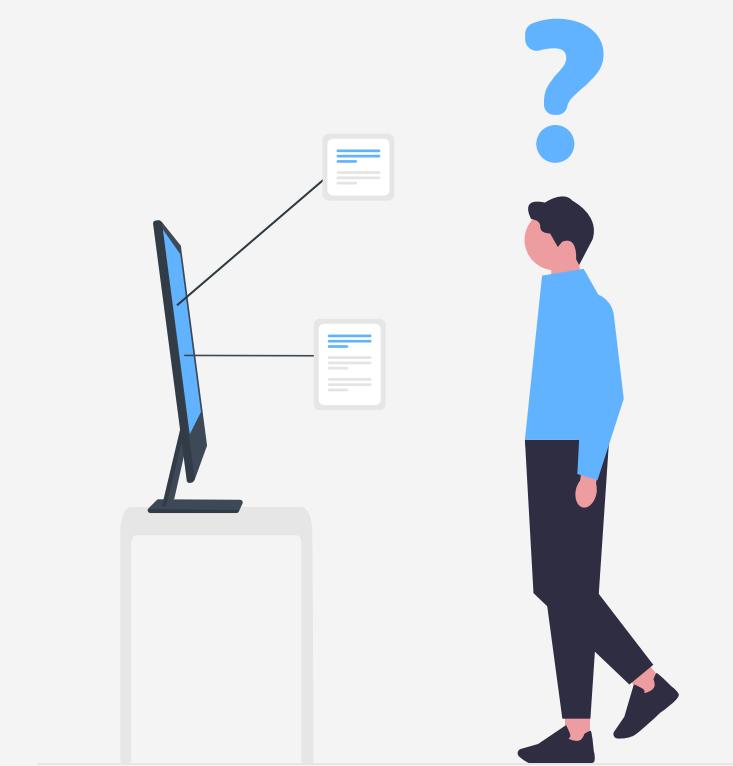
A multi-agent RAG assistant designed for the ResilientDB ecosystem and distributed systems research.



Goals

1

Intelligently query
information from research
papers and other sources



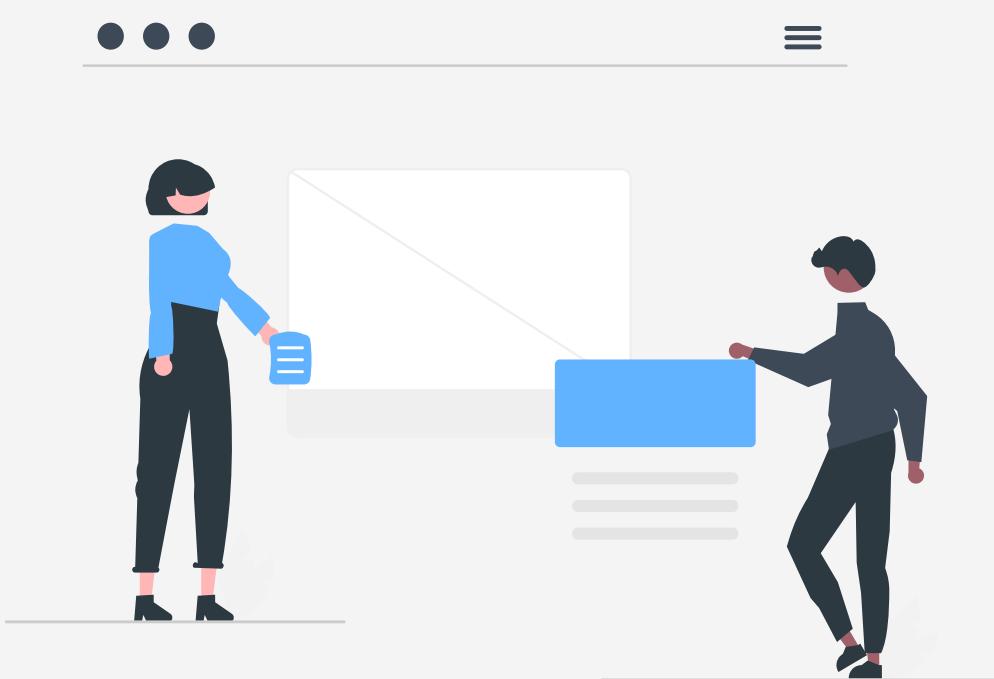
2

Generate code from
academic papers



3

Iterate on ideas with
conversational memory



Architecture

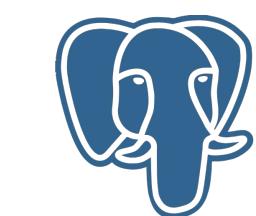
Implementation Stack



Orchestration
Document parsing
Indexing & embedding
Query engines & retrieval



Full-stack framework



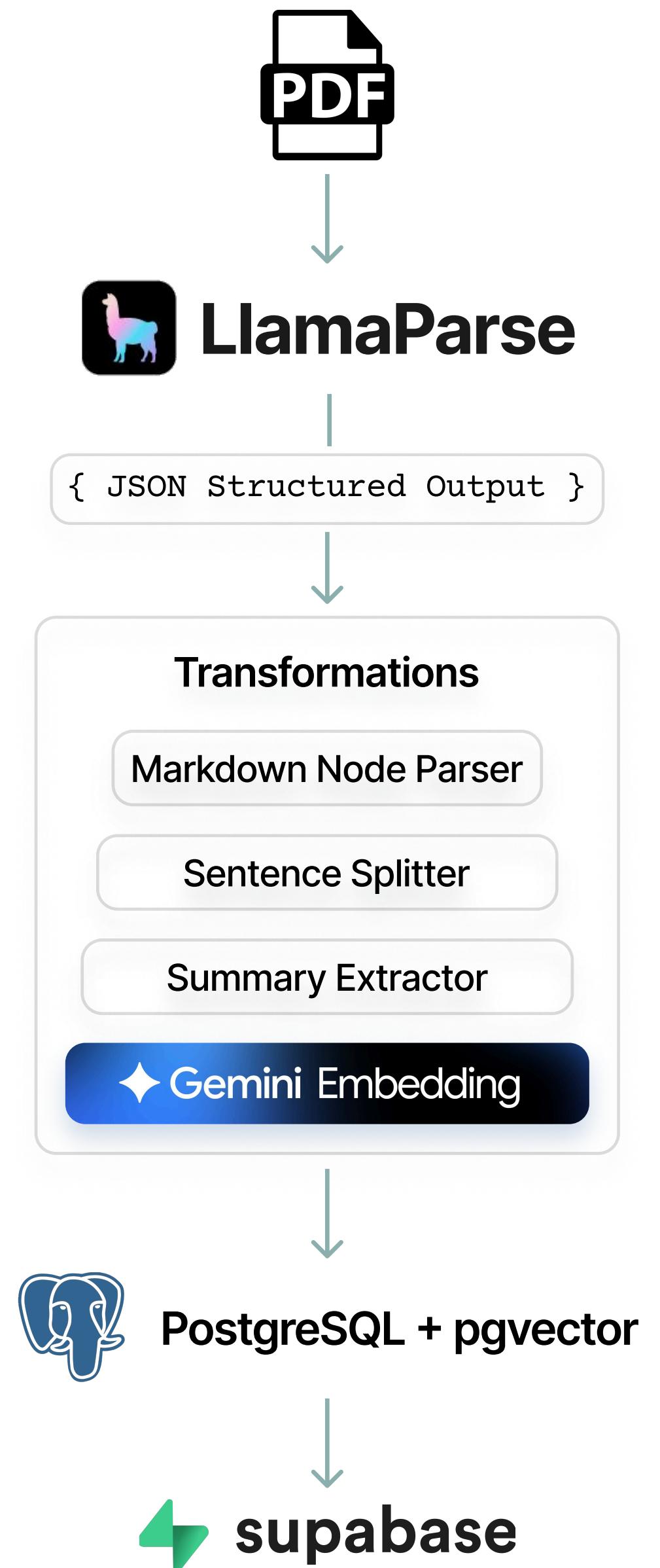
PostgreSQL + pgvector

Persistent Storage
Vector store
Memory



The Ingestion Pipeline

The quality of a RAG system is fundamentally dependent on the quality of the ingested data. Nexus' pipeline is designed to transform unstructured PDFs into queryable vector embeddings.



```
const reader = new LlamaParseReader({
  apiKey: config.llamaCloudApiKey,
  resultType: "json",
  verbose: true
});

const pipeline = new IngestionPipeline({
  transformations: [
    new MarkdownNodeParser(),
    new SentenceSplitter({
      chunkSize: 768,
      chunkOverlap: 20,
    }),
    new SummaryExtractor(),
    Settings.embedModel, // Gemini Embedding 001
  ],
  vectorStore: this.vectorStore, // pgvector
});
```

```
const nodes = await pipeline.run({ documents });
```

Ingestion

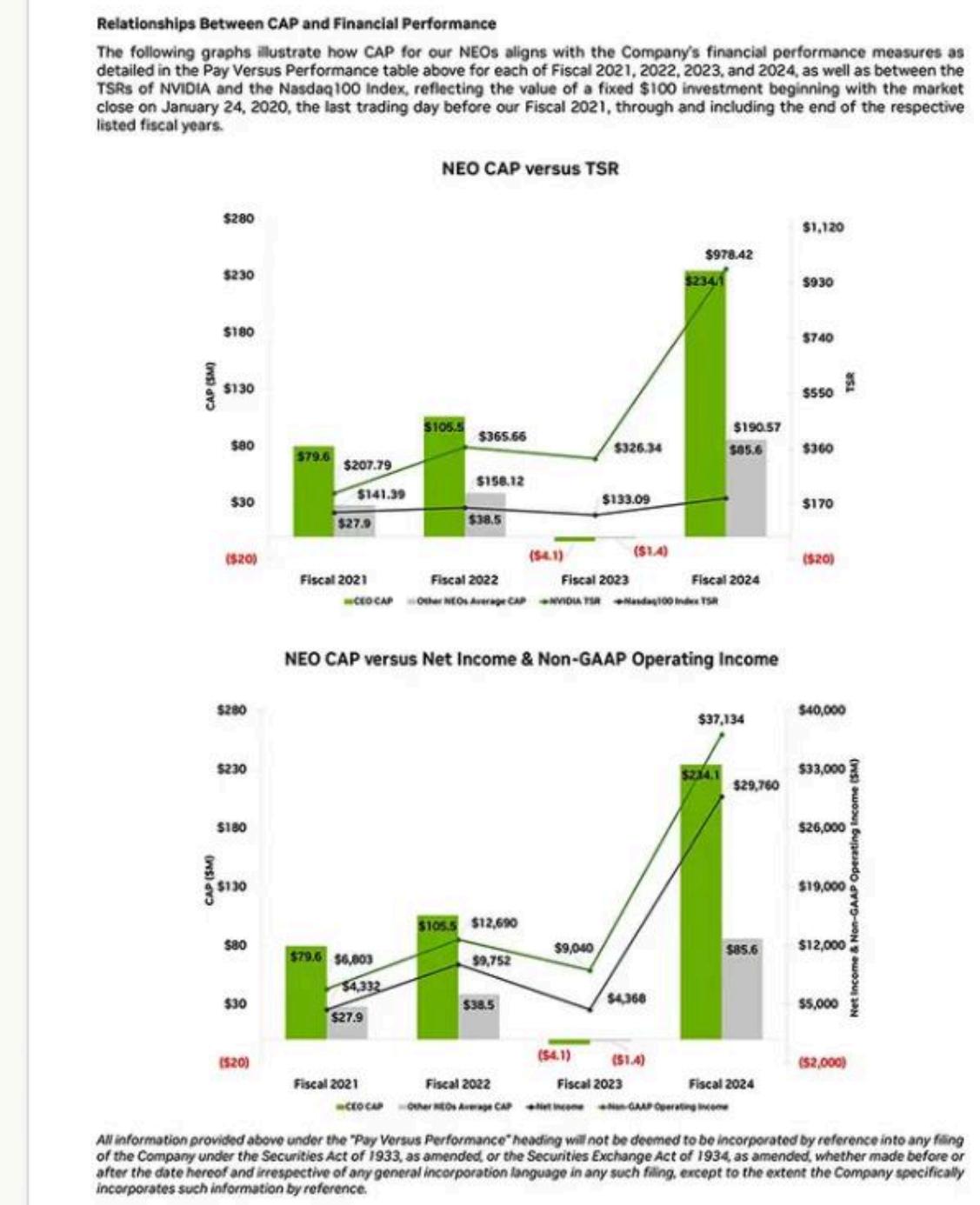
Document Parsing

The pipeline begins with LlamaParseReader, which converts unstructured PDFs into structured JSON

Generates markdown-formatted text that maintains semantic hierarchy

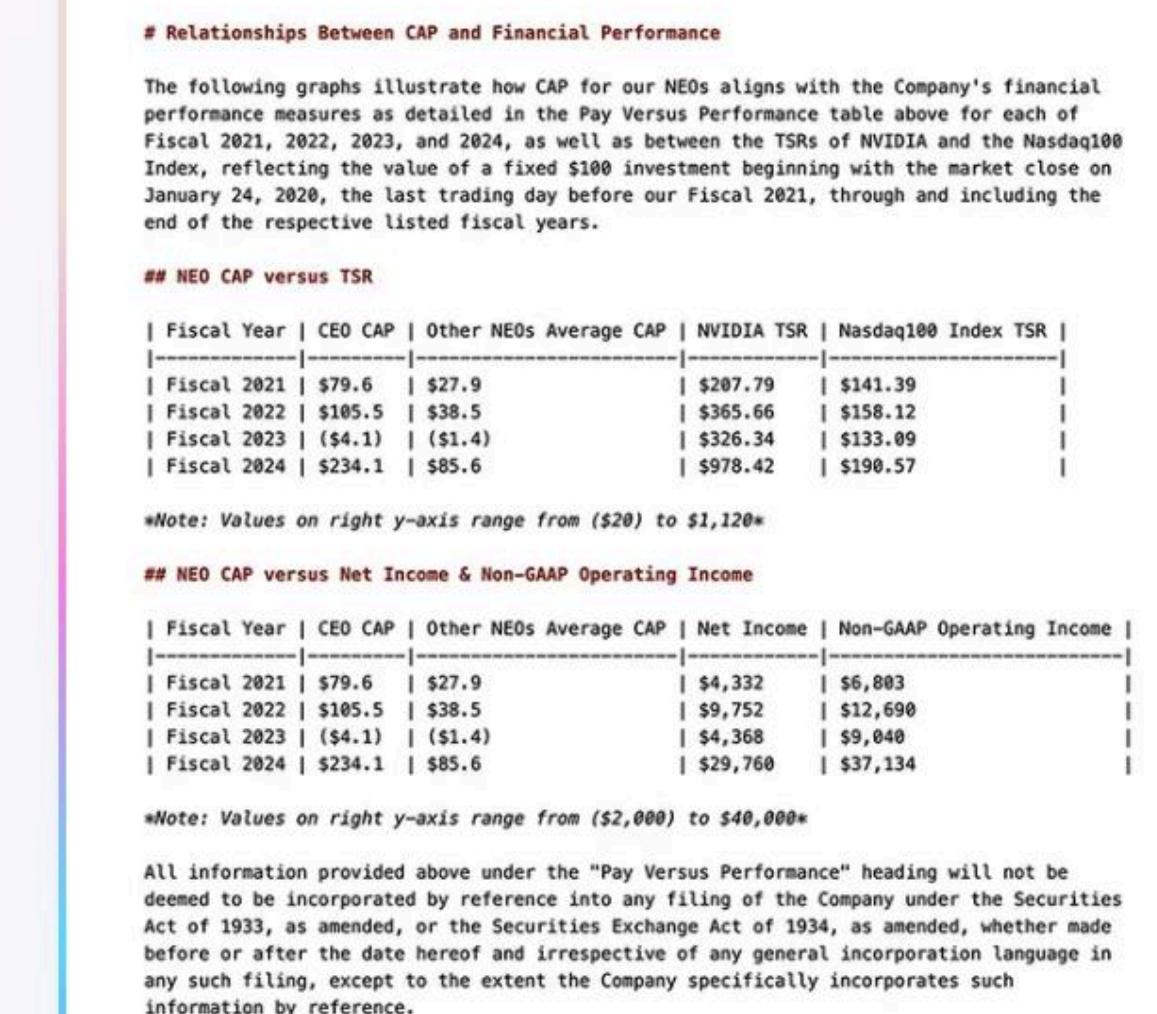
Accurately extracts tables, charts, images, and diagrams into structured formats.

Original Document



63

Parsing Results



63

Ingestion

Transformations



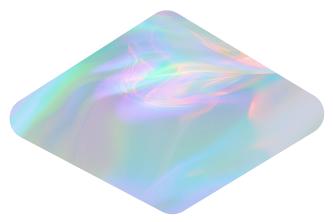
Markdown Node Parser: Respects document hierarchy through semantic heading structures



Sentence Splitter: Intelligently chunks text while preserving semantic coherence



Summary Extractor: Augments rich metadata for relevance and transparency



Gemini Embedding: Converts chunks into semantic vector representations



1. Hierarchical Structure Extraction (Transformation 1): Manabe, T., & Tajima, K. (2015). "Extracting Logical Hierarchical Structure of HTML Documents Based on Headings." *Proceedings of the VLDB Endowment*, 8(12), 1606-1617. <https://www.vldb.org/pvldb/vol8/p1606-manabe.pdf>.

2. Optimal Chunk Sizing (Transformation 2) Bhat, S. R., Rudat, M., Spiekermann, J., & Flores-Herr, N. (2025). "Rethinking Chunk Size For Long-Document Retrieval: A Multi-Dataset Analysis." *arXiv:2505.21700*. <https://arxiv.org/abs/2505.21700>.

3. Metadata-Enriched Retrieval (Transformation 3): Mombaerts, L., Ding, T., Banerjee, A., Felice, F., Taws, J., & Borogovac, T. (2024). "Meta Knowledge for Retrieval Augmented Large Language Models." *arXiv:2408.09017*. <https://arxiv.org/abs/2408.09017>.

Transformations

Markdown Node Parsing

```
# Distributed Systems
Distributed systems are systems whose components are located on different networked computers. They must coordinate to appear as a single coherent system.

## Consensus Protocols
Consensus protocols ensure that multiple nodes agree on a value despite failures. They are critical in blockchain, replicated databases, etc.

### Paxos
Paxos is a family of consensus algorithms that can tolerate failures. It's based on proposals, promises, and acceptances.

### Raft
Raft is designed to be more understandable. It elects leaders, replicates logs, and handles partitioning explicitly.

## Fault Tolerance
Fault tolerance refers to the ability to continue operating despite failures. Techniques include replication, sharding, rollback, etc.

## Case Study: Blockchain Systems
In blockchain systems, consensus and fault tolerance are combined to provide security and availability. Many blockchains use variants of BFT (Byzantine Fault Tolerance) protocols.

# Related Work
In recent years, many papers have studied consensus over unreliable networks, new consensus variants, etc.
```



```
Node (root)
  └─ Node: "Distributed Systems"
    header_path: ["Distributed Systems"]
    content: "Distributed systems are systems ... appear as a single coherent system."

  └─ Node: "Consensus Protocols"
    header_path: ["Distributed Systems", "Consensus Protocols"]
    content: "Consensus protocols ensure ... replicated databases, etc."

  └─ Node: "Paxos"
    header_path: ["Distributed Systems", "Consensus Protocols", "Paxos"]
    content: "Paxos is a family ... proposals, promises, and acceptances."

  └─ Node: "Raft"
    header_path: ["Distributed Systems", "Consensus Protocols", "Raft"]
    content: "Raft is designed ... partitioning explicitly."

  └─ Node: "Fault Tolerance"
    header_path: ["Distributed Systems", "Fault Tolerance"]
    content: "Fault tolerance refers to ... rollback, etc."

  └─ Node: "Case Study: Blockchain Systems"
    header_path: ["Case Study: Blockchain Systems"]
    content: "In blockchain systems ... BFT (Byzantine Fault Tolerance) protocols."

  └─ Node: "Related Work"
    header_path: ["Related Work"]
    content: "In recent years ... new consensus variants, etc."
```

Transformations

Sentence Splitter

Chunk Size: 768 tokens

Overlap: 20 tokens

Sentence-aware splitting



"In distributed systems, Byzantine Fault Tolerance (BFT) handles arbitrary failures including malicious nodes. Practical BFT protocols include PBFT, Tendermint, and HotStuff. PBFT uses a three-phase commit to ensure agreement. Tendermint decouples consensus rounds and uses a rotating proposer. HotStuff pipelines phases and reduces message complexity. BFT protocols rely on cryptographic signatures and quorum certificates to maintain safety and liveness."

"In distributed systems, Byzantine Fault Tolerance (BFT) handles arbitrary failures including malicious nodes. Practical BFT protocols include PBFT, Tendermint, and HotStuff. PBFT uses a three-phase commit to ensure agreement."

"...PBFT uses a three-phase commit to ensure agreement. Tendermint decouples consensus rounds and uses a rotating proposer. HotStuff pipelines phases and reduces message complexity."

"...pipelines phases and reduces message complexity. BFT protocols rely on cryptographic signatures and quorum certificates to maintain safety and liveness."

Summary Extractor

LLM-based annotation of chunk

Enhances result interpretation

Section Summary

Introduces Byzantine Fault Tolerance and PBFT protocol with its three-phase commit mechanism

Section Summary

Describes Tendermint's approach to BFT and HotStuff's pipelined consensus for BFT

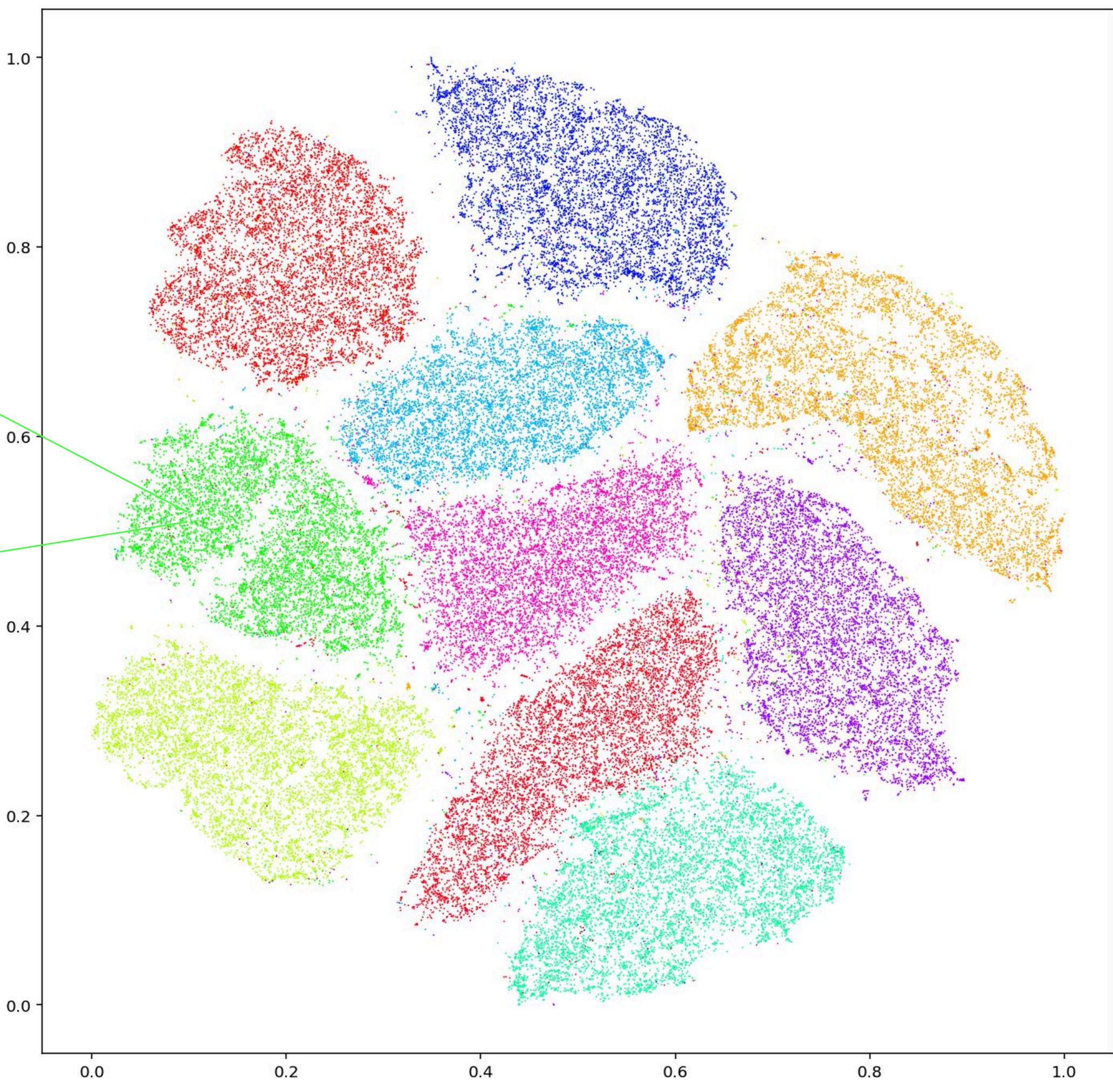
Section Summary

Discusses reliance on cryptographic signatures, view changes, and quorum certificates in BFT protocols.

Transformations

Embedding

“Consensus algorithm”
“Protocol for agreement”



T-SNE embeddings of MNIST dataset

```
const reader = new LlamaParseReader({
  apiKey: config.llamaCloudApiKey,
  resultType: "json",
  verbose: true
});

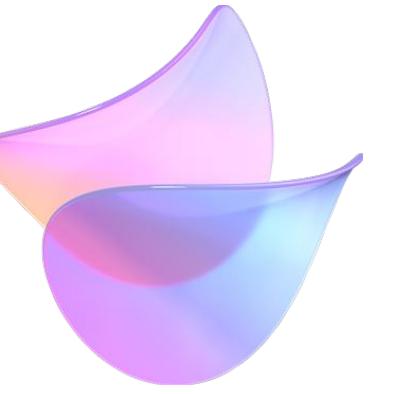
const pipeline = new IngestionPipeline({
  transformations: [
    new MarkdownNodeParser(),
    new SentenceSplitter({
      chunkSize: 768,
      chunkOverlap: 20,
    }),
    new SummaryExtractor(),
    Settings.embedModel, // Gemini Embedding 001
  ],
  vectorStore: this.vectorStore,
});

const nodes = await pipeline.run({ documents });
```

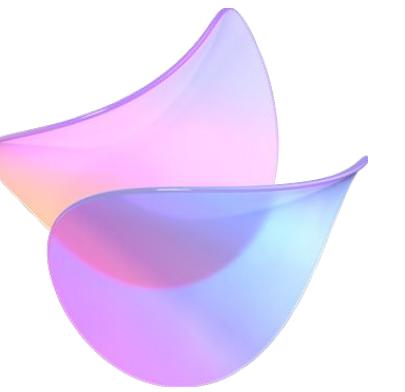
llama-service.ts

```
{
  "page": 1,
  "doc_id": "1fa671ba-7d5e-4811-92f1-8f885e051282",
  "_node_type": "TextNode",
  "ref_doc_id": "1fa671ba-7d5e-4811-92f1-8f885e051282",
  "document_id": "1fa671ba-7d5e-4811-92f1-8f885e051282",
  "_node_content": "{Full content of the node}",
  "sectionSummary": "Based on the information provided, the summary is: This is the citation for a document from the 2021 IEEE 37th International Conference on Data Engineering (ICDE). The source document can be found at a specific Google Drive URL. The document has the DOI 10.1109/ICDE51399.2021.00124 and its ISBN is 978-1-7281-9184-3. The copyright is held by IEEE in 2021.",
  "source_document": "https://drive.google.com/uc?export=download&id=1Hrjs8nrxZeUgxv009pQ4RDe9YEhP2qEd"
}
```

[0.045054592,-0.037427053,-0.013345831,0.0143943885,0.018287357,0.029309465,0.048970364,-0.037848596,-0.019507779,0.0113140205,0.035685625,-0.03730628,0.009060913,-0.004251276,-0.053073347,0.0041570244,0.0042670136,-0.019425694,0.0010801862,-0.002995339,0.013110867,0.029987669,-0.00990041,-0.025567
861,-0.034190807,-0.029335614,-0.08502919,-0.008585837,0.027368942,-0.05662335,0.013088469,-0.051867794,0.03535009,0.014644119,-0.032793853,0.01055421,-0.012371013,-0.007235346,0.026198002,-0.012811754,-0.010937583,-0.02808223,-0.01896981,0.039751187,-0.043444984,-0.022184132,0.013809059,-0.024870366,-0.0487
83895,0.022366347,0.0067822896,0.04106069,-0.007834533,0.01799001,-0.04948757,-0.010346083,0.007883295,-0.006438088,-0.005141225,0.035576552,-0.0019174676,-0.014232455,0.042864587,0.016796304,-0.09872204,-0.03837689,0.02860502,0.07425296,-0.025598068,0.022391243,-0.03297029,0.045841474,-0.027404854,-0.029566487
,-0.10796287,-0.028835956,0.046248853,0.024407145,0.0074812323,-0.05330227,-0.024366908,-0.02978933,-0.053448245,-0.035102926,0.027269455,-0.029894818,-0.023163186,-0.04269084,0.07213797,-0.030355155,-0.02714559,0.010290779,-0.07747191,0.022822179,0.013990647,-0.026422637,-0.05888465,0.025511084,-0.04785892,0.
0124828825,-0.038271274,-0.051607203,0.015518795,0.034668203,-0.004377671,-0.009526284,0.12317518,-0.0047018193,0.078035325,-0.055175867,-0.01891247,-0.00029407177,-0.018717857,0.0007655383,-0.012075507,0.008285081,0.058566436,0.023364715,0.033559185,0.029901728,0.017532561,0.03234473,0.04382169,0.014114258,0.0
3204593,0.003327568,0.038020156,0.0372865,0.007181305,0.017110392,-0.03277441,0.016188368,0.04711224,0.028664077,0.029156938,0.075805284,-0.03341327,0.037957773,-0.01627511,-0.022529798,0.017403739,-0.0061310846,0.046118904,-0.0037693498,-0.008794241,-0.007576191,-0.0032796722,-0.011361023,-0.008712466,-0.00531
98203,-0.009755723,-0.040887404,0.0008500466,0.097240634,0.017320853,0.037568334,0.019258434,0.021889027,-0.03445499,0.0500844,-0.029758293,0.018537201,0.03250169,-0.0020674502,-0.026933065,-0.024907917,-0.007103617,-0.035283454,-0.0008090982,0.007664101,0.035244327,-0.035026923,-0.03144913,-0.040258393,-0.0851
6306,-0.03267669,-0.021905037,0.004988663,-0.009189612,-0.030797394,-0.054115765,0.014310109,0.0042979065,0.0355872,0.018680645,0.07851912,-0.03337334,-0.023975963,-0.01966598,-0.01888664,-0.059043188,-0.022329143,0.0319194,0.03187782,-0.0019656005,0.0040531694,-0.006010702823,-0.03802987,-0.020168
75,0.07300549,0.013751715,-0.008735685,0.0147932945,-0.005515184,0.07152479,-0.030746933,-0.03560285,0.04086823,-0.03956056,0.02867413,-0.014506769,0.013544234,0.06874336,0.034635056,0.048954625,0.030250989,-0.0010706063,-0.04914125,0.013543987,-0.030106647,0.04696871,-0.021720454,0.07954436,0.00
703646,0.07125526,0.02636805,-0.07379131,-0.026850386,0.07269728,0.004102216,-0.0077199712,0.03288888,0.023787932,0.020121578,0.008748846,-0.01549048,-0.021043194,-0.030820658,0.019099075,0.068082996,0.01126815,0.00940413,-0.03722482,-0.017794564,0.017709829,-0.012688128,0.039006114,-0.033292647,-0.044369206,0.
016692474,0.022746982,-0.07513609,0.05912447,-0.019595174,0.04614273,-0.017746298,-0.011867442,-0.009712143,0.002754626,0.022896009,0.014657753,0.0027460032,-0.019785373,-0.015390676,-0.048480373,0.018830214,0.045807634,0.0066392156,-0.06413011,0.090011306,-0.031681526,-0.024073428,0.021701533,-0.015281158,0.01
7038567,0.030059038,-0.038760137,0.027436843,0.014223371,0.034190394,-0.03219567,-0.042360112,0.026365388,-0.035607334,-0.017529335,0.045531705,-0.057289217,-0.06562186,-0.012903989,-0.032509863,-0.0530657,-0.01407934,-0.00014810845,-0.023117712,0.04438752,0.032558948,-0.03404656,-0.023415254,-0.07849372,-0.032
53125,-0.060576756,0.04160789,0.013371682,0.02573322,-0.05465939,0.029874722,-0.01560937,0.022386799,-0.046060972,0.011346813,0.0137707265,0.035842016,0.09351039,-0.010192759,-0.021037346,-0.0350597,0.059227377,0.00078544865,0.06038948,0.054985803,-0.0032905715,0.0119425645,0.075784445,0.051990777,
0.0122802025,-0.011467843,-0.03406338,0.01795881,0.026757637,0.082981005,-0.0016308475,-0.014446405,-0.07070057,0.0116018844,0.016237399,0.0182094,-0.030773092,0.02354314,-0.04891349,-0.032076247,0.044919893,-0.009181673,-0.03900997,0.00030786626,0.058908224,0.0403234844,-0.068179056,-0.04094745,-0.0323
09007,0.007712316,-0.051450156,0.0537784,-0.023740565,0.04273752,-0.040737465,-0.06738708,-0.0041401573,-0.02618207,0.01717939,0.0069641173,-0.036676306,0.018683707,-0.017575175,0.033253703,0.023126086,-0.065285355,-0.026396237,0.01440618,-0.0027057,-0.03860217,-0.041935902,-0.057071984,0.012493779
,-0.011752237,-0.039252765,0.0028293333,0.05411652,0.053826913,-0.022509953,-0.037713446,0.030920686,0.053161576,0.019579945,0.01260849,-0.007281518,0.019452939,0.08218685,0.02308802,0.007931069,-0.024230702,-0.020788118,-0.016631367,0.024988005,-0.017825384,-0.007240692,-0.01959386,-0.05011076,-0.013817169,-0.
032445792,-0.07646976,0.013313386,-0.017470196,-0.02093293,-0.028017629,0.0035983787,0.01117247,0.010292836,-0.058555096,-0.11271633,-0.038810983,0.069106236,0.02557883,0.028680619,0.013681855,0.018544815,0.0018574293,0.012374663,0.013054026,-0.028657388,-0.040082816,-0.0062189335,0.0024682074,-0.015364334,0.0
31207373,0.062739,-0.015524038,0.0130072633,-0.01809759,0.0102650225,0.023741893,-0.030264704,-0.010358395,0.031403538,0.01973742,0.04966875,-0.04652603,-0.003872562,-0.009862989,0.014547517,-0.09980768,-0.016205423,0.09105247,-0.055878308,-0.040080924,-0.026893718,-0.035933528,0.00036020
373,-0.06168844,-0.026456162,0.007182898,-0.010995649,-0.002983411,-0.011188277,-0.06541244,0.001516591,-0.08976748,-0.023578487,-0.015529637,0.03591153,0.031522673,-0.0403457,-7.193734e-05,-0.027666721,-0.006020446,-0.03366468,-0.102536545,0.028259235,-0.030026074,-0.0096109
6,0.015797133,-0.004182767,0.055151064,0.0063886866,-0.004777197,-0.030313851,0.029739888,-0.01595901,-0.004005257,0.062608756,0.006220815,-0.0008798068,-0.031121006,-0.008652619,-0.032712642,-0.0016169531,0.064101376,0.020249922,0.037716374,0.020256096,-0.012706188,-0.024337597,0.015873158,0.00563
29127,-0.040229924,-0.0128125865,0.029432332,-0.00055092433,0.023659173,0.031508077,0.01256879,0.010107785,-0.010990974,0.035192057,-0.012700568,0.029228175,0.012813209,-0.00087613205,-0.04917383,0.037603926,-0.020790461,-0.106091425,-0.01262407,0.0790496,-0.0874164,-0.020312544,-0.0020256403,-0.035526313,0.030
947171,-0.032963928,0.07150111,-0.014730626,-0.01573197,0.0030830104,-0.023236265,0.016022556,0.014490172,0.015567217,0.0018819771,-0.025489943,-0.024203936,0.03898399,0.023267135,-0.03945135,0.023060305,0.04862085,-0.13168655,0.03320365,-0.049504496,0.014047868,0.00873216,0.03553588,-0.05772851,-0.00825874,-0.
01469883,-0.0077147055,0.023112953,-0.0086371275,0.034750972,-0.032370675,0.034282997,-0.026942294,-0.023103466,0.06321355,0.048506934,-0.04112983,-0.011174838,0.008434346,-0.031543124,-0.013377556,-0.014334358,0.026171777,-0.00084897055,0.0640999,-0.03544731,-0.04463825,0.011161612,-0.0065544466,0.009464226,0.
032932416,0.013662266,0.035507467,-0.005249748,-0.031130593,0.059457634,0.017921094,-0.0013427715,0.01974341,-0.042643473,-0.039844267,0.05142641,-0.06126553,0.018240137,0.029167835,-0.02327895,-0.006842381,0.013611945,-0.03209337,0.027143601,0.025044475,-0.04205668,0.046993285,0.04123182,0.01494545
31,-0.02407854,0.033128876,0.0104922615,-0.012089555,0.002982733,0.005163093,0.016039446,-0.07537757,0.00226335,-0.030400475,0.002179893,0.02090096,0.002308598,-0.02503054,-0.0014868418,0.031053694,-0.0075833253,0.017454293,0.035265602,-0.0134925395,-0.04815737,-0.026426874,0.045011364,0.042029987,
0.052294258,-0.024112785,0.0099



Building the Nexus Agent



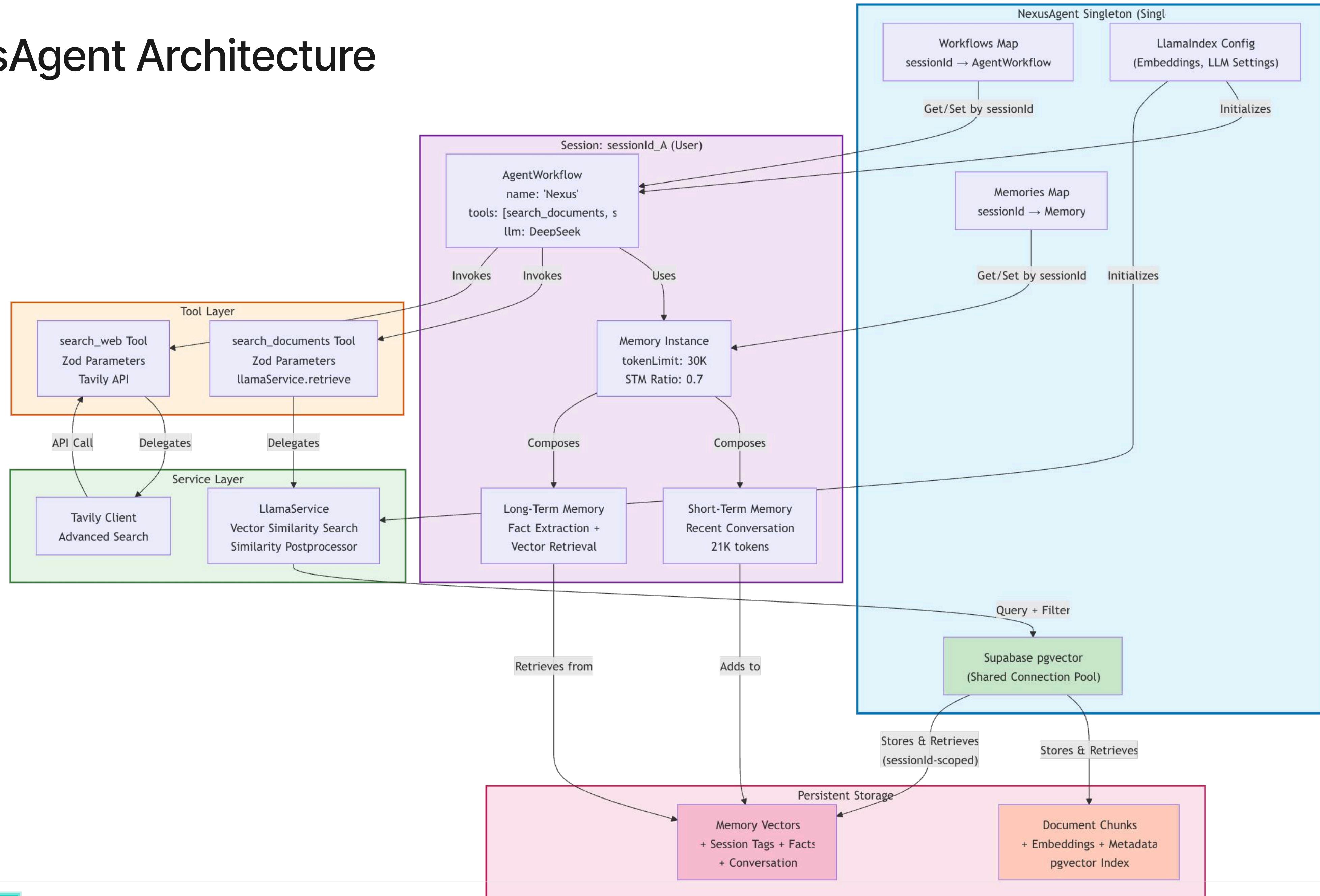
Building the Nexus Agent

Autonomy in Tool Selection

Iterative Reasoning

Memory-Augmented Context

NexusAgent Architecture



NexusAgent Architecture

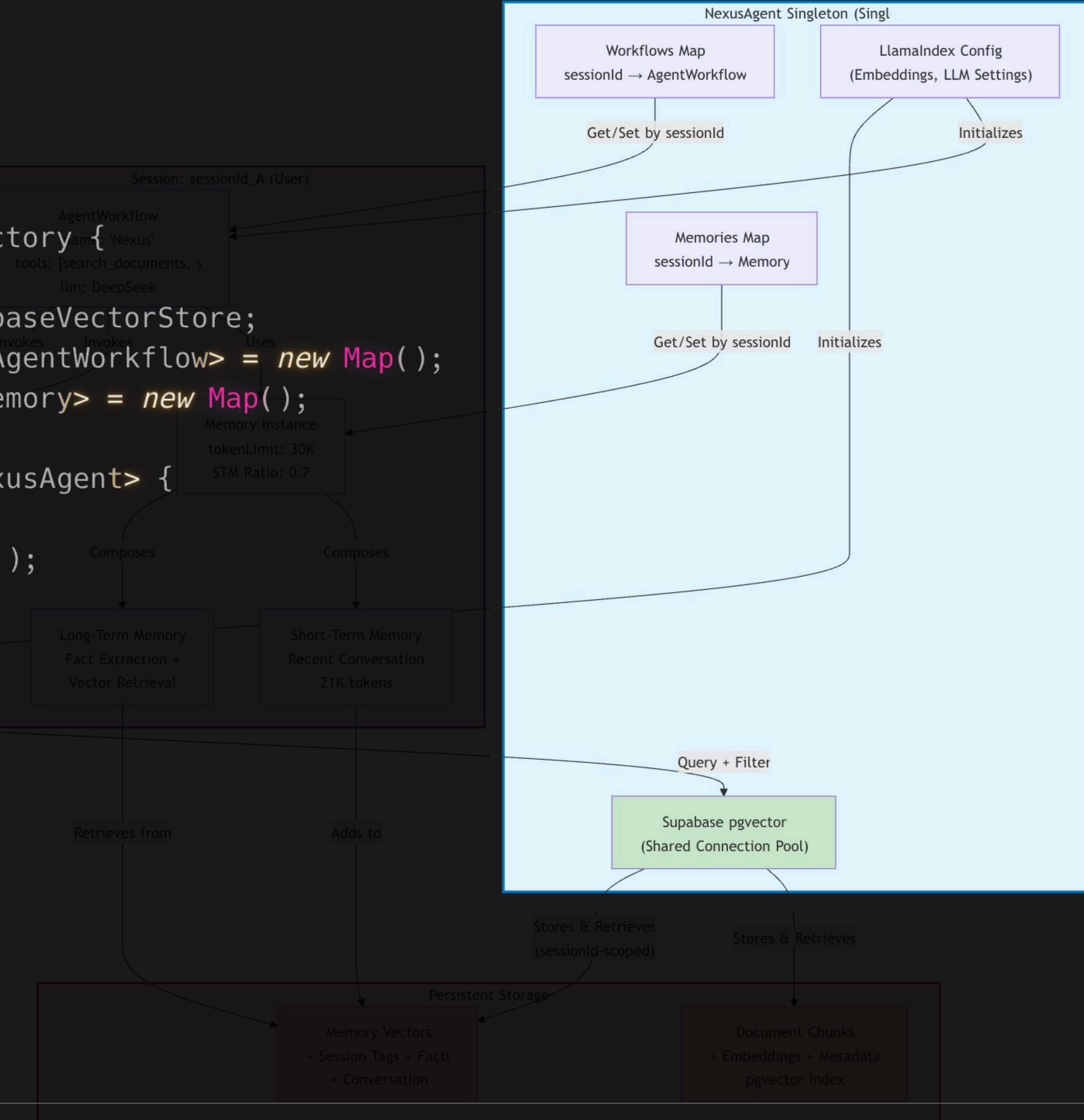
```

export class NexusAgent implements AgentFactory {
    private static instance?: NexusAgent;
    private readonly memoryVectorStore: SupabaseVectorStore;
    private readonly workflows: Map<string, AgentWorkflow> = new Map();
    private readonly memories: Map<string, Memory> = new Map();

    public static async create(): Promise<NexusAgent> {
        if (!NexusAgent.instance) {
            NexusAgent.instance = new NexusAgent();
        }
        return NexusAgent.instance;
    }
}

```

The code defines the `NexusAgent` class which implements `AgentFactory`. It contains static fields for `instance`, `memoryVectorStore`, `workflows`, and `memories`. The `create()` method returns the static `instance`.



NexusAgent Architecture

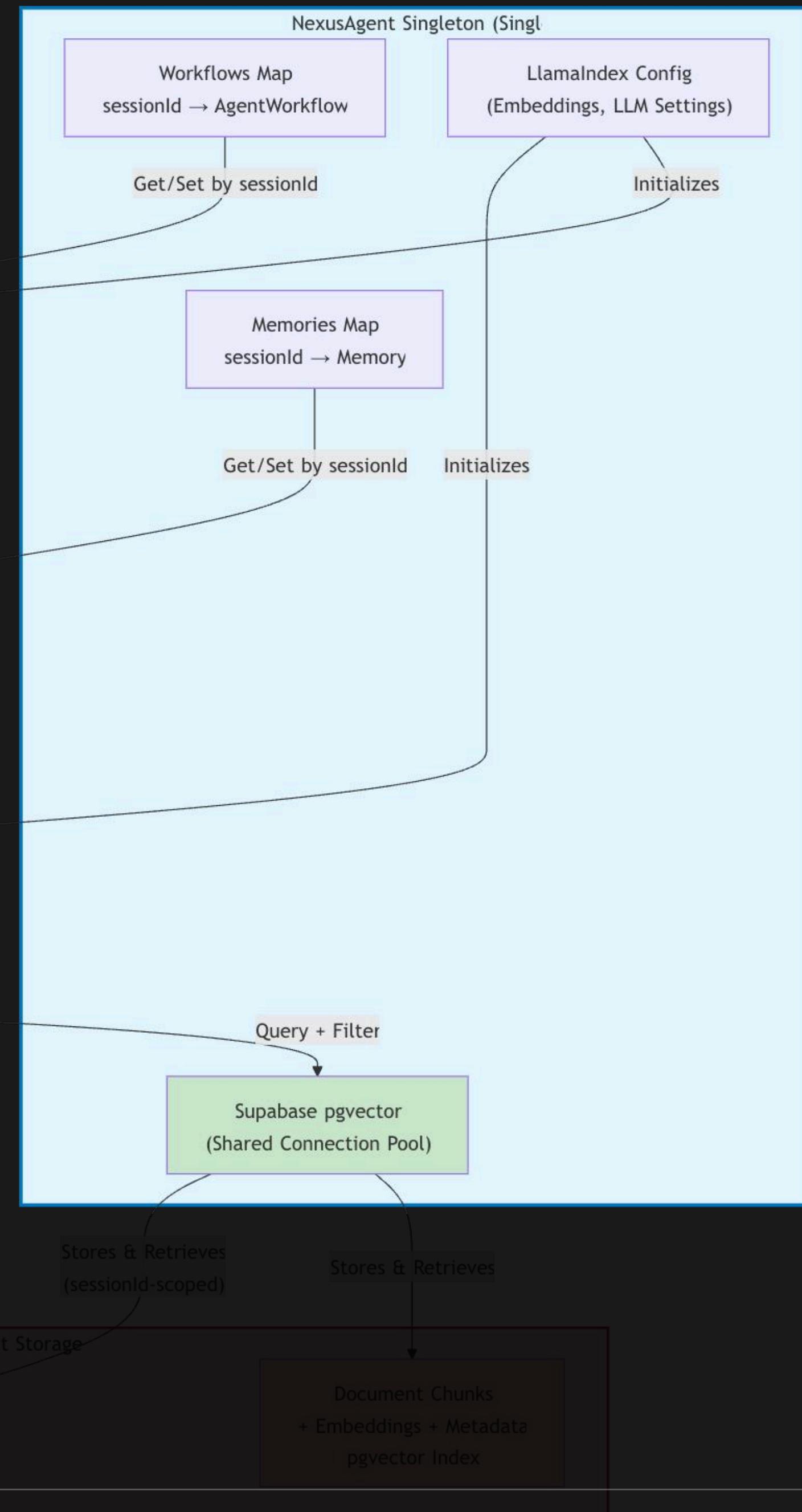
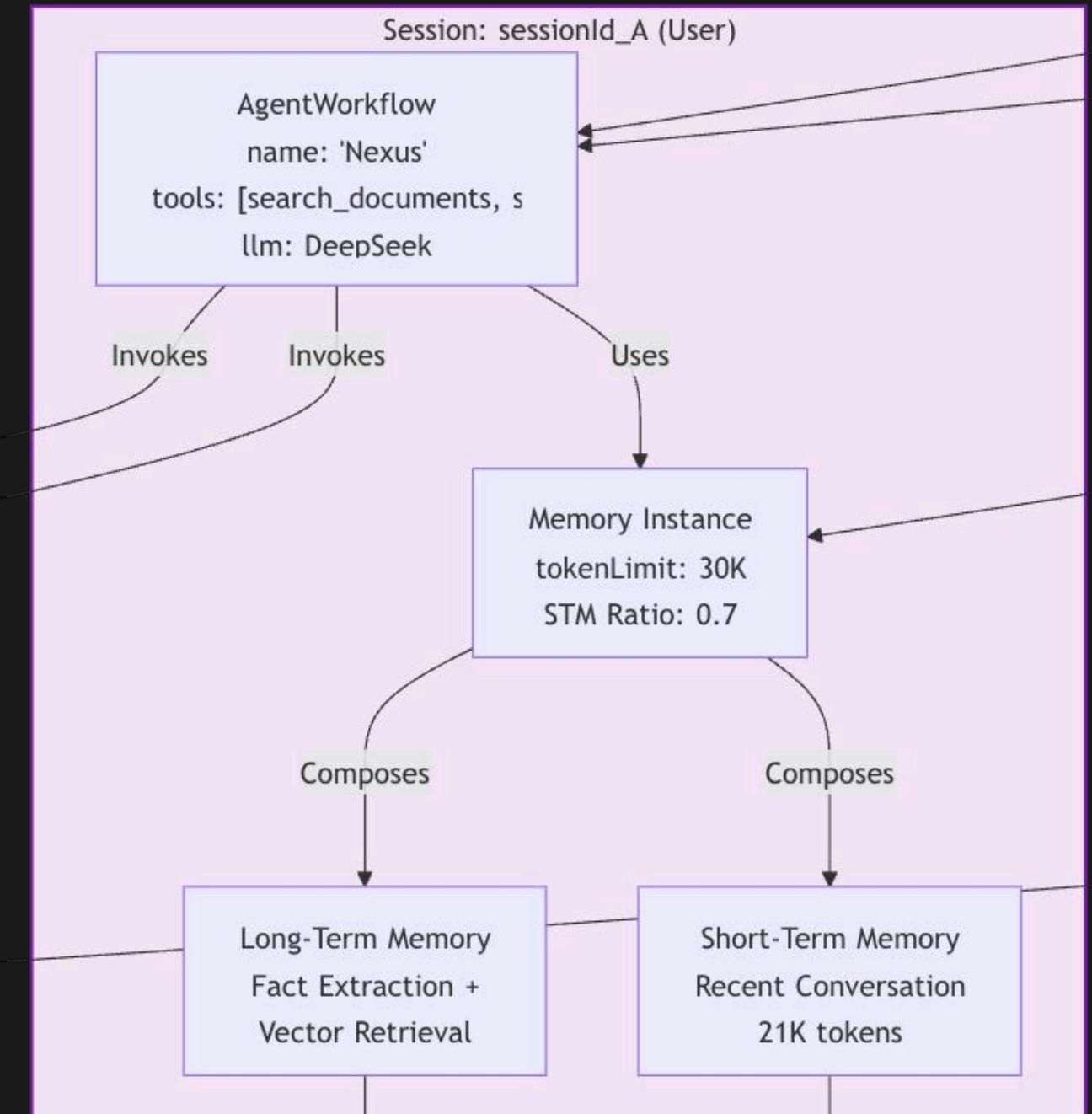
```

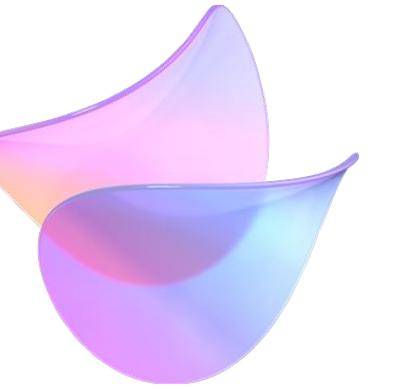
private readonly workflows: Map<string, AgentWorkflow> =
new Map();

public async createAgent(documents: string[], sessionId: string): Promise<AgentWorkflow> {
    const existingWorkflow =
this.workflows.get(sessionId);
    if (existingWorkflow) {
        return existingWorkflow;
    }

// Create new workflow if not present
const workflow = agent({
    name: "Nexus",
    tools: [documentSearchTool, webSearchTool],
    systemPrompt: AGENT_RESEARCH_PROMPT,
    llm: deepseek({ model: config.deepSeekModel }),
    memory,
});
    this.workflows.set(sessionId, workflow);
    return workflow;
}

```





Building the Nexus Agent

Memory

Memory Blocks

Two-block Architecture

Fact Extraction

Vector conversation

30k token budget

Injection Into Context Window

```
const memoryBlocks: BaseMemoryBlock[] = [
  robustFactExtractionBlock({
    id: 'retrieved-facts',
    priority: 1,
    llm: deepseek({ model: config.deepSeekModel }),
    maxFacts: 10,
    isLongTerm: true,
  }),
  vectorBlock({
    id: sessionId,
    vectorStore: this.memoryVectorStore,
    priority: 2,
    retrievalContextWindow: 5,
    queryOptions: {
      similarityTopK: 3,
      mode: "hybrid",
    },
  }),
];
const memory = createMemory({
  tokenLimit: 30000,
  shortTermTokenLimitRatio: 0.7,
  memoryBlocks,
});
```

agent.ts

Memory Blocks

Fact Extraction

Continuous Extraction During Conversation

Long-Term Persistence

Injection Into Context Window

“The user’s name is Alice”

“User prefers to focus on high performance fault tolerant systems”

“ResilientDB uses PBFT with 3-phase commit”

```
const memoryBlocks: BaseMemoryBlock[] = [
    robustFactExtractionBlock({
        id: 'retrieved-facts',
        priority: 1,
        llm: deepseek({ model: config.deepSeekModel }),
        maxFacts: 10,
        isLongTerm: true,
    }),
    vectorBlock({
        id: sessionId,
        vectorStore: this.memoryVectorStore,
        priority: 2,
        retrievalContextWindow: 5,
        queryOptions: {
            similarityTopK: 3,
            mode: "hybrid",
        },
    }),
];
const memory = createMemory({
    tokenLimit: 30000,
    shortTermTokenLimitRatio: 0.7,
    memoryBlocks,
});
```

agent.ts

Memory Blocks

Vector Store

Embedding Conversational Turns

Semantic Similarity Retrieval

Retrieval Context Window

```
const memoryBlocks: BaseMemoryBlock[] = [
  robustFactExtractionBlock({
    id: 'retrieved-facts',
    priority: 1,
    llm: deepseek({ model: config.deepSeekModel }),
    maxFacts: 10,
    isLongTerm: true,
  }),
  vectorBlock({
    id: sessionId,
    vectorStore: this.memoryVectorStore,
    priority: 2,
    retrievalContextWindow: 5,
    queryOptions: {
      similarityTopK: 3,
      mode: "hybrid",
    },
  }),
];
```

```
const memory = createMemory({
  tokenLimit: 30000,
  shortTermTokenLimitRatio: 0.7,
  memoryBlocks,
});
```

agent.ts

<https://arxiv.org/abs/2506.14086>

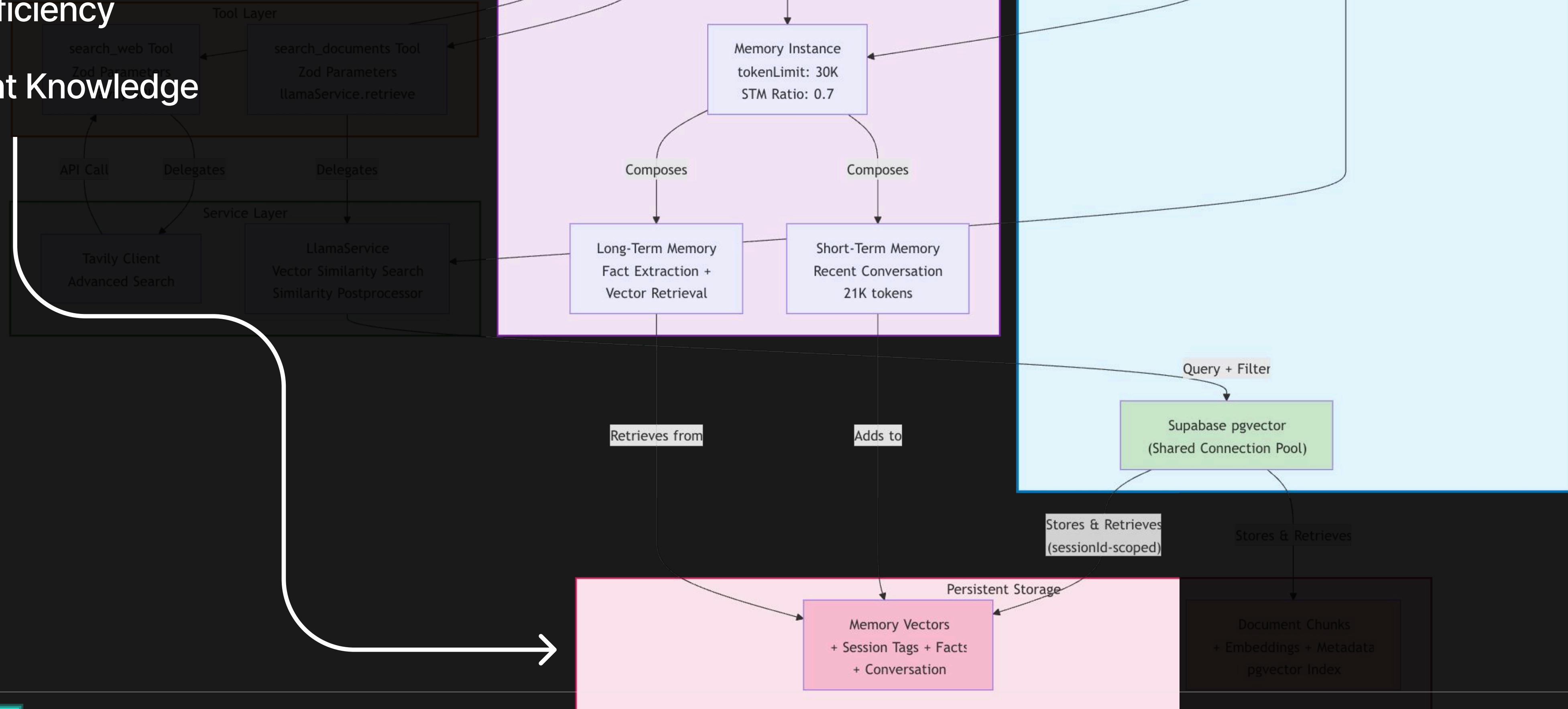
NexusAgent Architecture

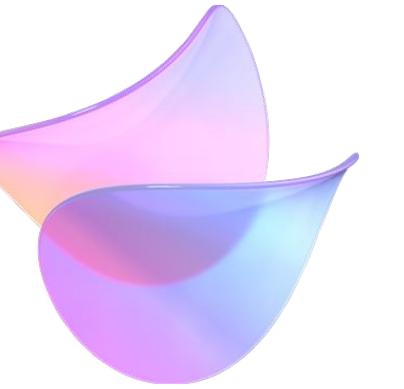
Memory Blocks

Complementary Design

Token Efficiency

Persistent Knowledge

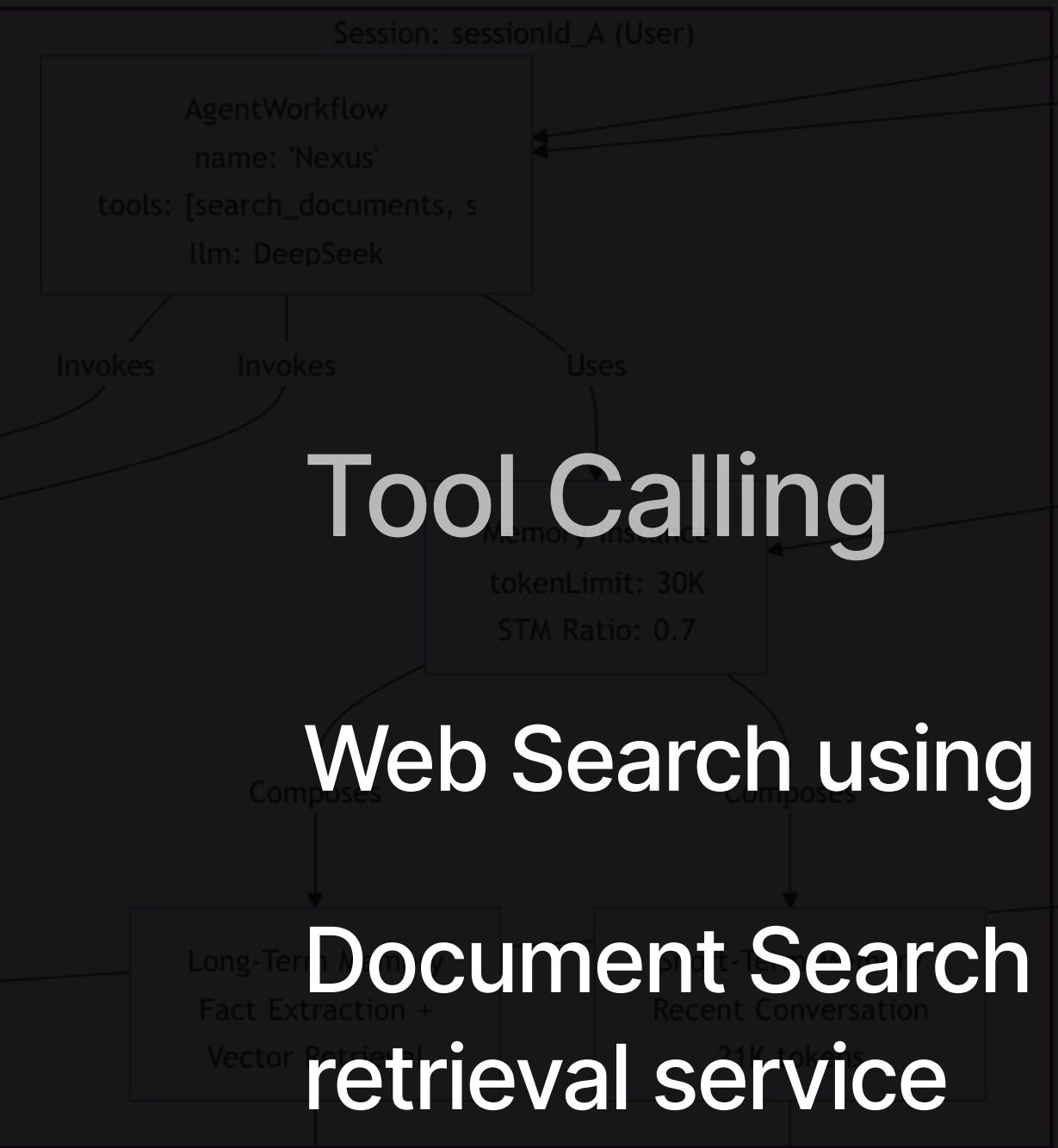
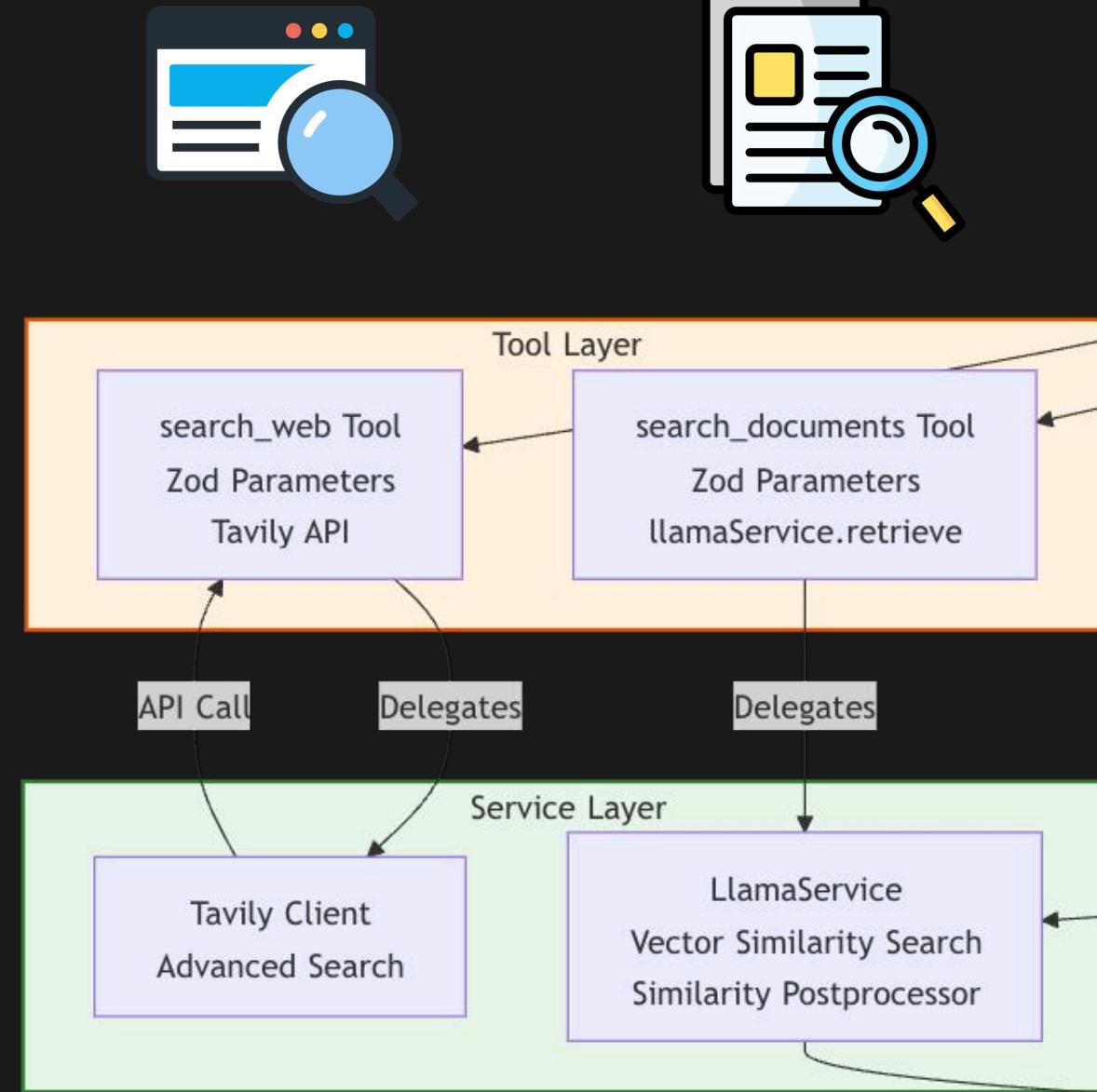




Building the Nexus Agent

Tools

NexusAgent Architecture



Tool Calling

Web Search using Tavily API

Document Search using custom retrieval service

Tools

Semantic Document Search

Nexus' primary mechanism for retrieving information from indexed papers & documents

All attributes are interpreted by the agent as it reasons!

Executes `llamaService.retrieve` with given parameters

```
private createDocumentSearchTool() {  
    return tool({  
        name: "search_documents",  
        description: "This tool can retrieve detailed information from the selected documents.",  
        parameters: z.object({  
            query: z.string().describe("The query to retrieve information from the document's vector embeddings."),  
            documentPaths: z.array(z.string()).describe("The list of document paths to search in"),  
        }),  
        execute: llamaService.retrieve,  
    });  
}
```

Semantic Document Search

Step 1

```
{  
  "tool": "search_documents",  
  "parameters": {  
    "query": "What consensus algorithm does ResilientDB use?",  
    "documentPaths": ["documents/resilientdb-paper.pdf"]  
  }  
}
```

Semantic Document Search

Step 2

Step 1
Tool Call

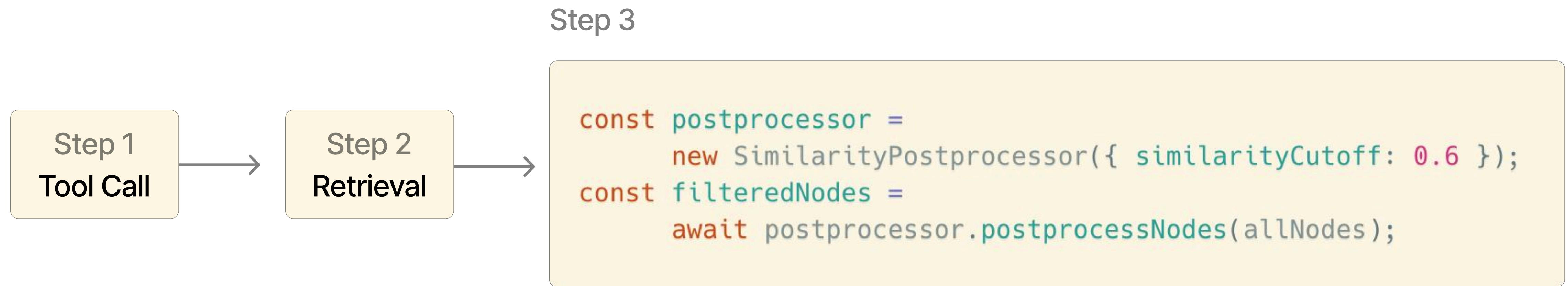


```
const allNodes: NodeWithScore[] = [];
const index = await VectorStoreIndex.fromVectorStore(this.getVectorStore());

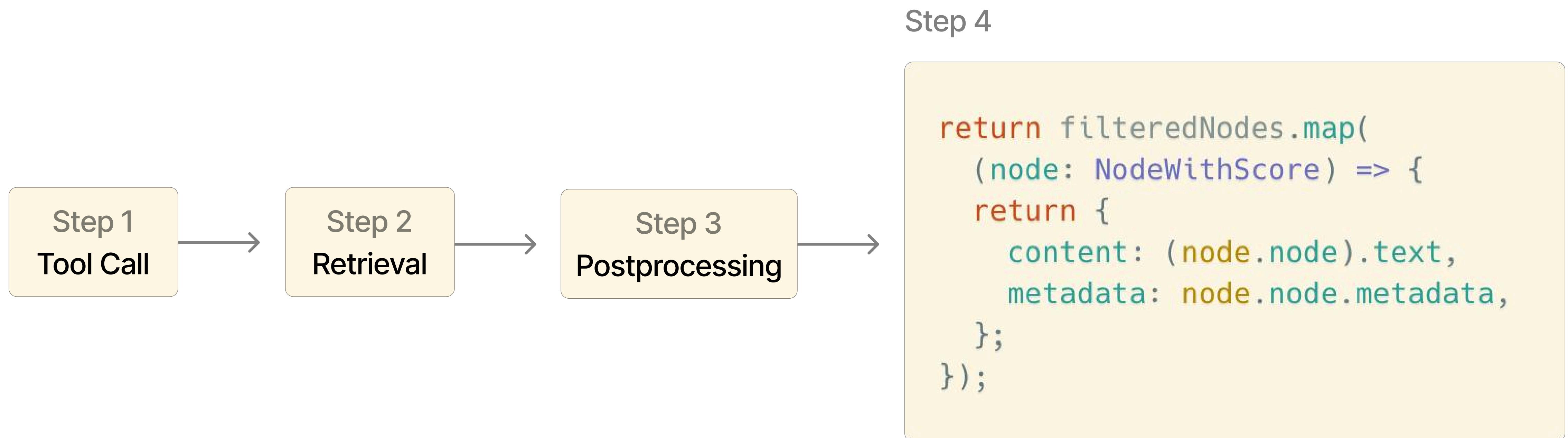
for (const docPath of parameters.documentPaths) {
  const retriever = index.asRetriever({
    similarityTopK: 5, // Get 5 most similar chunks per document
    filters: [
      {
        key: "source_document",
        operator: "==",
        value: docPath,
      },
    ],
  });
}

const nodes = await retriever.retrieve({ query: parameters.query });
allNodes.push(...nodes);
}
```

Semantic Document Search



Semantic Document Search



Tools

Web Search

Nexus' mechanism for retrieving up to date information from the web

Utilizes Tavily's advanced search API

Executes `llamaService.searchWeb` with given parameters

```
private createWebSearchTool() {
  return tool({
    name: "search_web",
    description: "Search the web for information",
    parameters: z.object({
      query: z.string().describe("The search query optimized for web search"),
    }),
    execute: async (parameters: { query: string }) => {
      try {
        const results = await llamaService.searchWeb(parameters);
        return results;
      } catch (error) {...}
    },
  });
}
```

```

searchWeb = async (parameters: { query: string }) => {
  const tavily =
    new TavilyClient({
      apiKey: config.tavilyApiKey
    });
  const response =
    await tavily.search({
      query: parameters.query,
      search_depth: "advanced",
      include_answer: true,
      include_images: true,
      max_results: 10,
    });
  return response;
};

```

lamaService.searchWeb

```

[WebSearchTool] Found 10 results for query: "Byzantine Generals Problem 2025 recent developments updates blockchain consensus"
[WebSearchTool] Results: [
  {
    "url": "https://tangem.com/en/glossary/byzantine-general-problem/",
    "title": "Byzantine Generals' Problem Meaning in Crypto - Tangem",
    "content": "# What is Byzantine Generals' Problem\n\nUpdated Feb 4, 2025\n\nThe Byzantine Generals' Problem is a fascinating conundrum in the realm of computer science and game theory, highlighting the issues in achieving consensus within a distributed network with unreliable participants.\n\n## Byzantine Generals' Problem Meaning [...]\nWithin the sphere of blockchain technology, the Byzantine Generals' Problem emphasizes the need for consensus mechanisms. Blockchains, as decentralized networks, must rely on algorithms that ensure all participants (or nodes) agree on the submitted data, even if some players are faulty or malicious. Solutions to this issue, such as proof-of-work and proof-of-stake, are central to the validation processes in cryptocurrencies like Bitcoin and Ethereum. They ensure that transactions and data [...]",
    "score": 0.8021998
  },
  {
    "url": "https://www.elibrary.imf.org/view/journals/001/2025/186/article-A001-en.xml",
    "title": "Blockchain Consensus Mechanisms: A Primer for Supervisors (2025 ...",
    "content": "The Byzantine General's Problem (BGP) sets out these problems which are applicable to public blockchain networks where distributed",
    "score": 0.6952351
  },
  {
    "url": "https://www.rsinc.com/byzantine-fault-tolerance.php",
    "title": "Byzantine fault tolerance 2025",
    "content": "Consensus in blockchains isn't just about agreeing on a block's hash or timestamp. It's about affirming that every transaction is valid across dozens, hundreds, or even thousands of nodes. BFT protocols evaluate proposed blocks, confirm consistency, and only then allow addition to the chain. [...] In any distributed system, nodes must agree on a single data value or decision to function coherently. This process, known as consensus, ensures consistency and order in the presence of delays, disconnections, or data duplication. While consensus is relatively straightforward in fault-free systems, the situation changes drastically when nodes may act arbitrarily or maliciously—conditions modeled by the Byzantine fault class.\n\n## Why Consensus Gets Complicated with Byzantine Faults [...]\nByzantine Fault Tolerance (BFT) refers to a system's ability to continue operating correctly and reach consensus even when some of its components (or nodes) fail or act maliciously. More formally, a system achieves BFT if it can guarantee consensus among multiple nodes, despite up to  $f$  faulty nodes out of a total of  $3f + 1$  nodes. This threshold comes from the fundamental result proven in the 1982 paper on Byzantine Generals by Lamport, Shostak, and Pease.",
    "score": 0.61112726
  }
]

```

search_web result

Tools

Selection Autonomy

Agent Context: Tools are passed as a list in the AgentWorkflow instance

```
const workflow = agent({
  name: "Nexus",
  tools: [documentSearchTool, webSearchTool],
  systemPrompt: AGENT_RESEARCH_PROMPT,
  llm: deepseek({ model: config.deepSeekModel }),
  memory,
});
```

Tools

Selection Autonomy

Agent Context: Tools are passed as a list in the AgentWorkflow instance

Function Calling: Deepseek-Chat supports and recognizes JSON tool interfaces

```
{  
  "tool": "search_documents",  
  "parameters": {  
    "query": "consensus algorithm Byzantine fault tolerance PBFT",  
    "documentPaths": ["documents/resilientdb-paper.pdf"]  
  }  
}
```

Tools

Multi-Tool Interactions

The agent can perform multiple tool calls in sequence, depending on the information requested

Compare ResilientDB's consensus approach to recent blockchain systems.

I'll help you compare ResilientDB's consensus approach to recent blockchain systems. Let me start by searching the web for current information about blockchain consensus mechanisms, then examine the selected document about ResilientDB.

 *Searched the web*

Now let me search the selected document to understand ResilientDB's consensus approach.

 *Read 1 document*

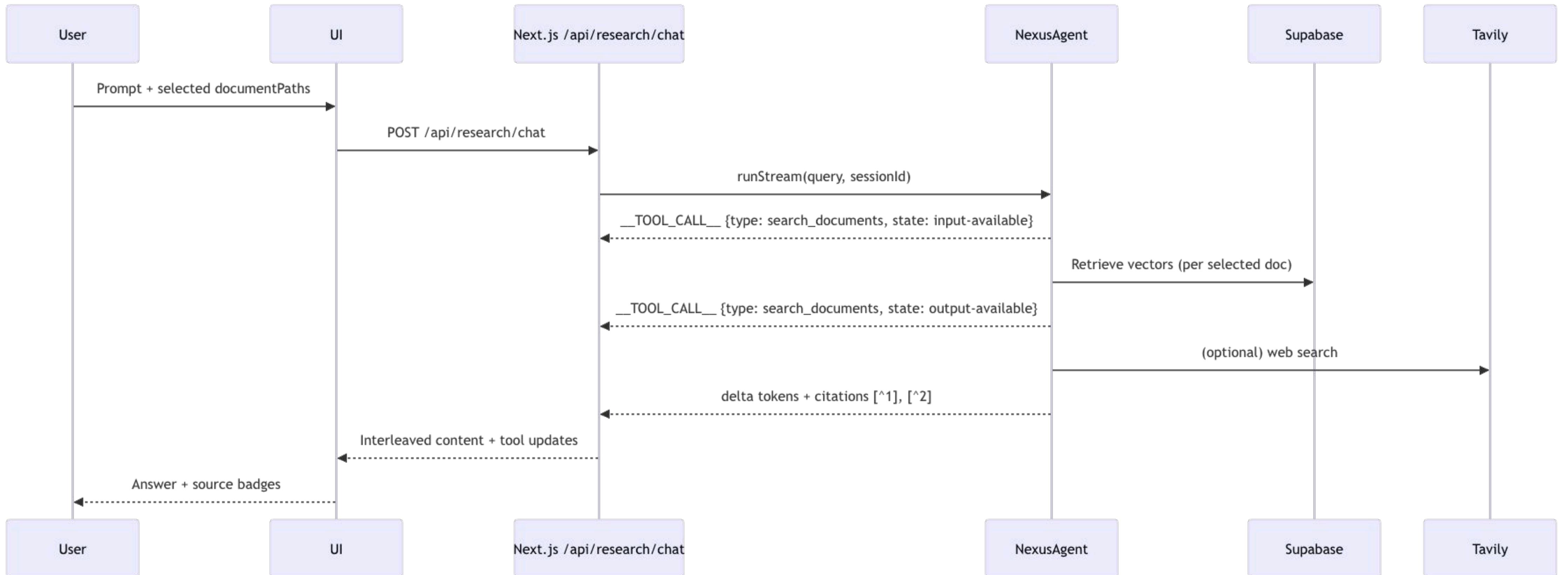
Based on my research from both web sources and the ResilientDB document, here's a comprehensive comparison of ResilientDB's consensus approach to recent blockchain systems:



Building the Nexus Agent

Prompt Engineering

System Architecture



[Demo - Research Agent, Web Search Preference]



Multi-agent

The Coding Agent



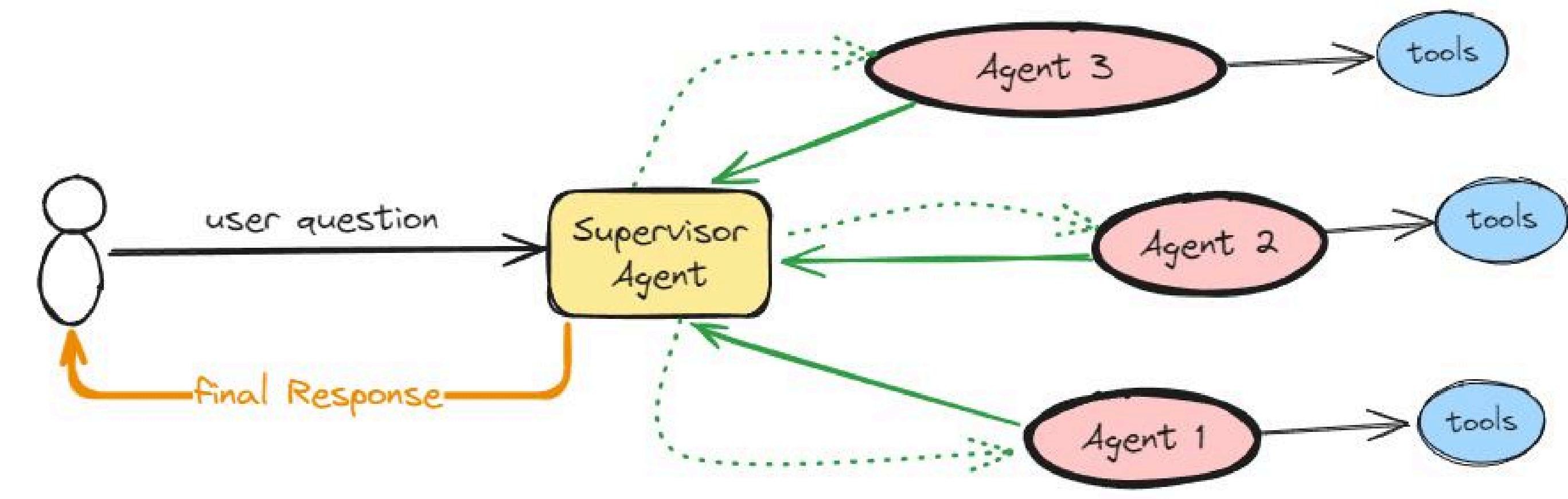
Coding Agent

Multi-Agent Motivation

Task Decomposition: Improves output quality for complex workflows

"Roles" over single-pass: Mimic real-world behavior to separate concerns

Modularity: Attach specialized prompts/tools for each phase

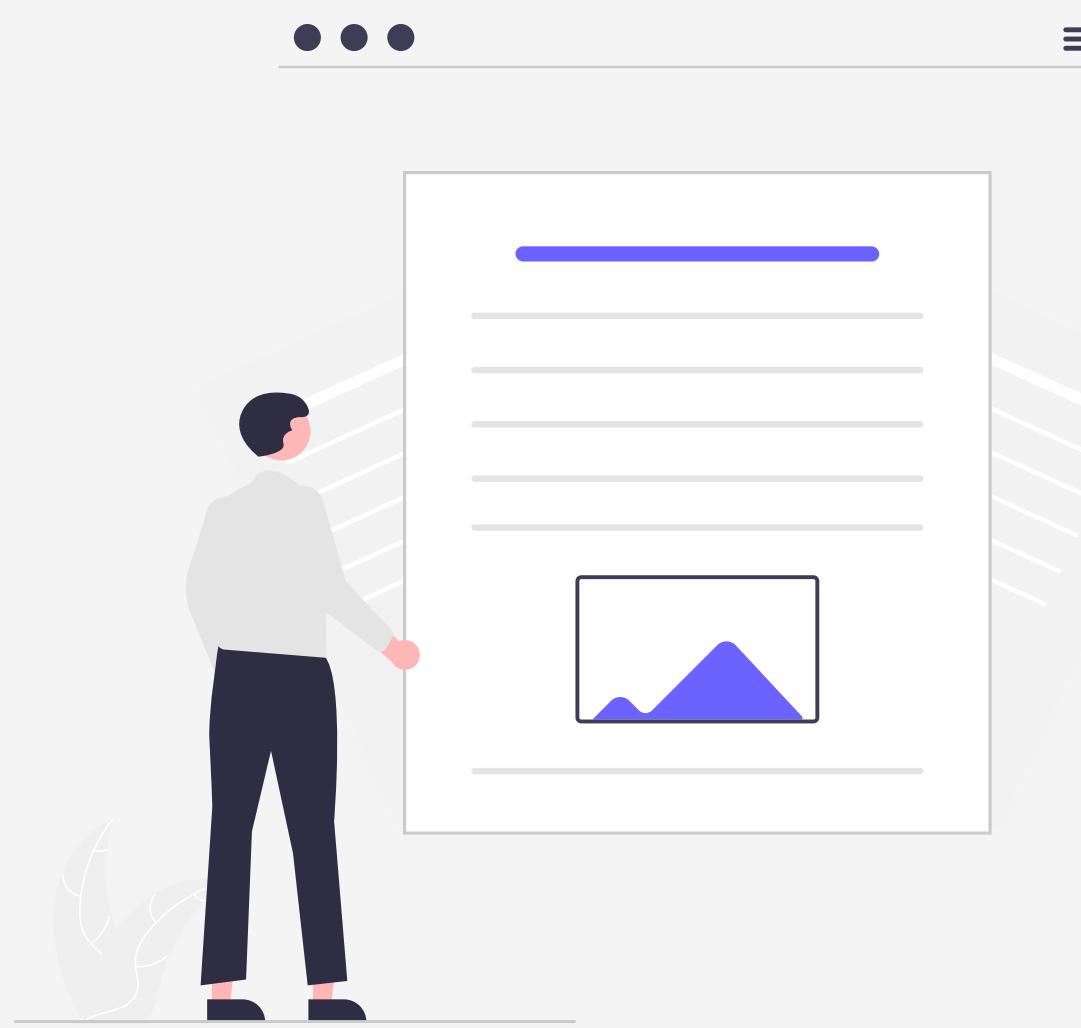


Rana Nameer Hussain Khan, Dawood Wasif, Jin-Hee Cho, Ali Butt. "Multi-Agent Code-Orchestrated Generation for Reliable Infrastructure-as-Code" *arXiv preprint arXiv:2510.03902*, 2025.

Vali Tawosi, Xiaomo Liu, Keshav Ramani, Salwa Alamir. "ALMAS: an Autonomous LLM-based Multi-Agent Software Development Framework." *arXiv preprint arXiv:2510.03463*, 2025.

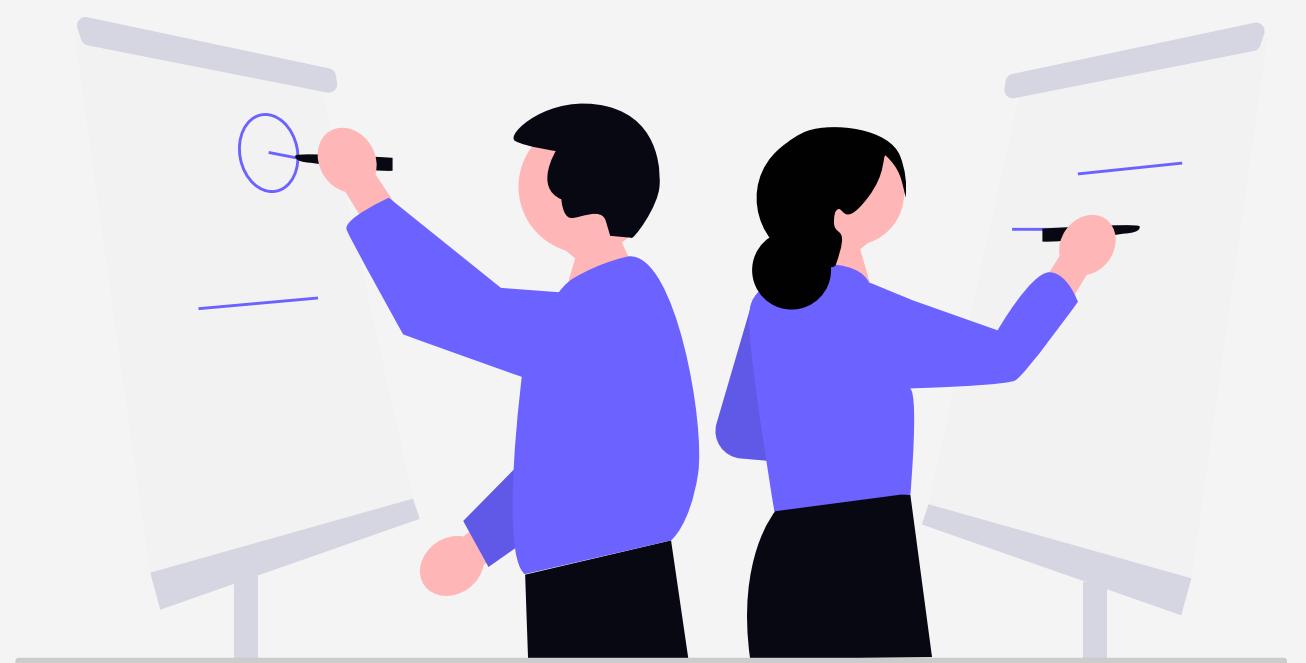
1 Planning

Research, retrieve, and understand dense concepts



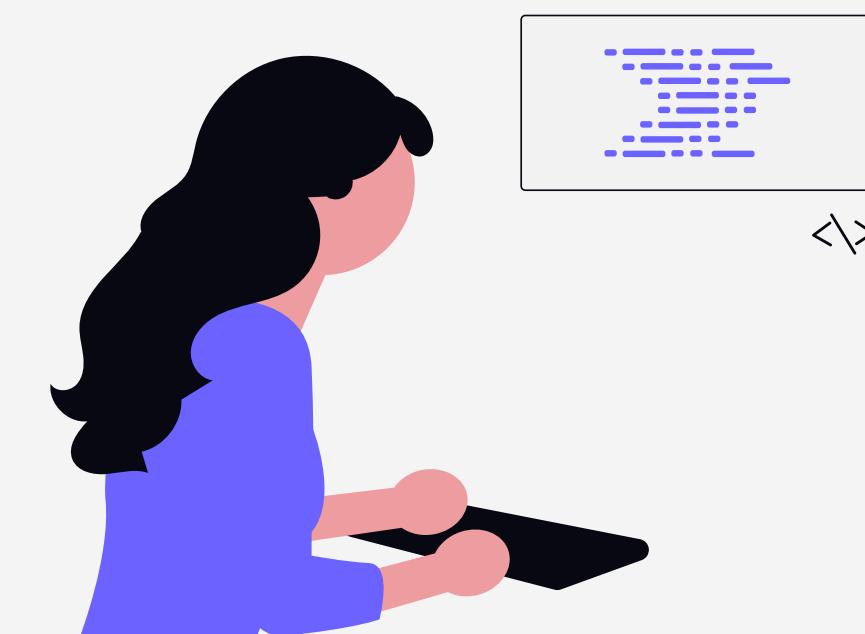
2 Pseudocode

Create a structured, high-level outline for the next agent



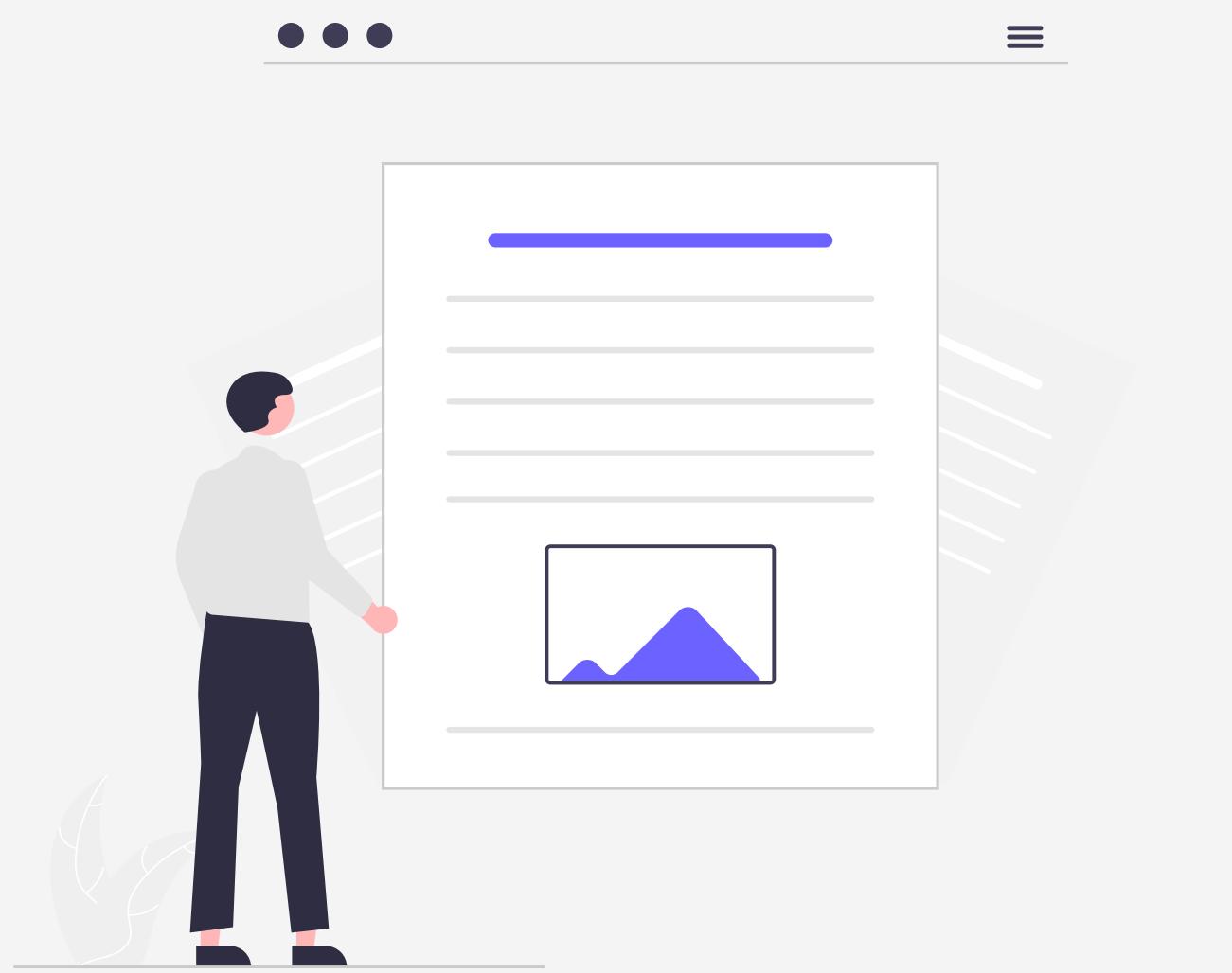
3 Implementation

Consolidate findings, generate final code



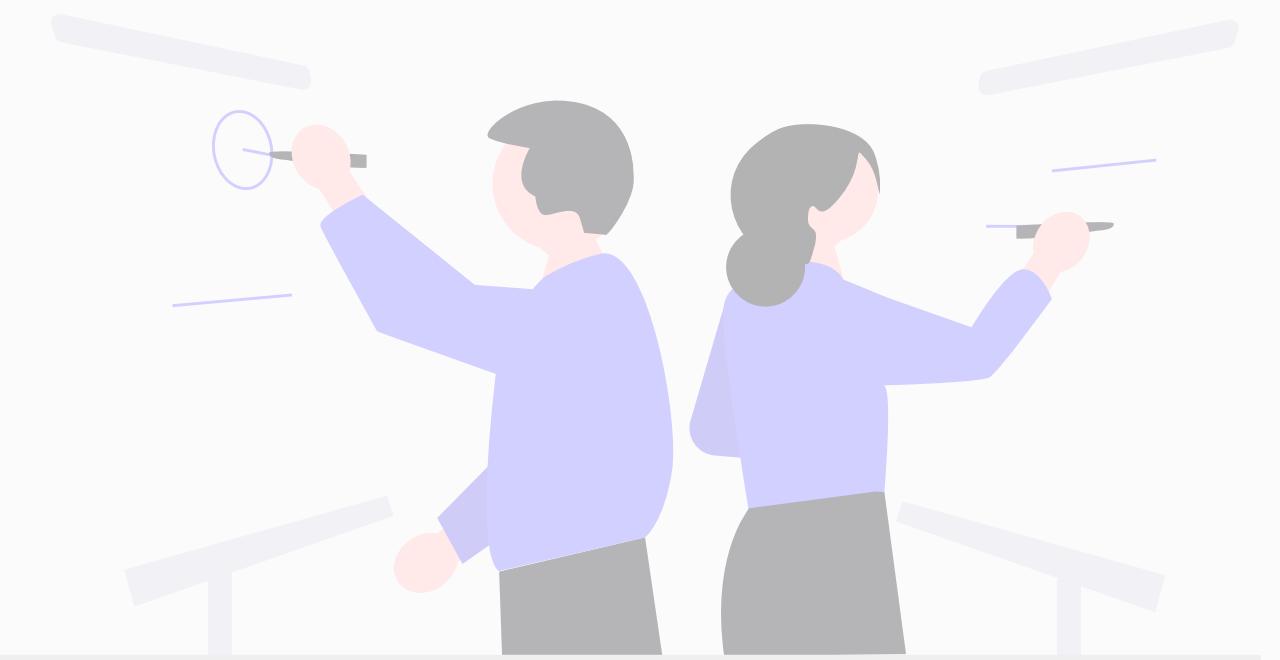
1 Planning

Research, retrieve, and understand dense concepts



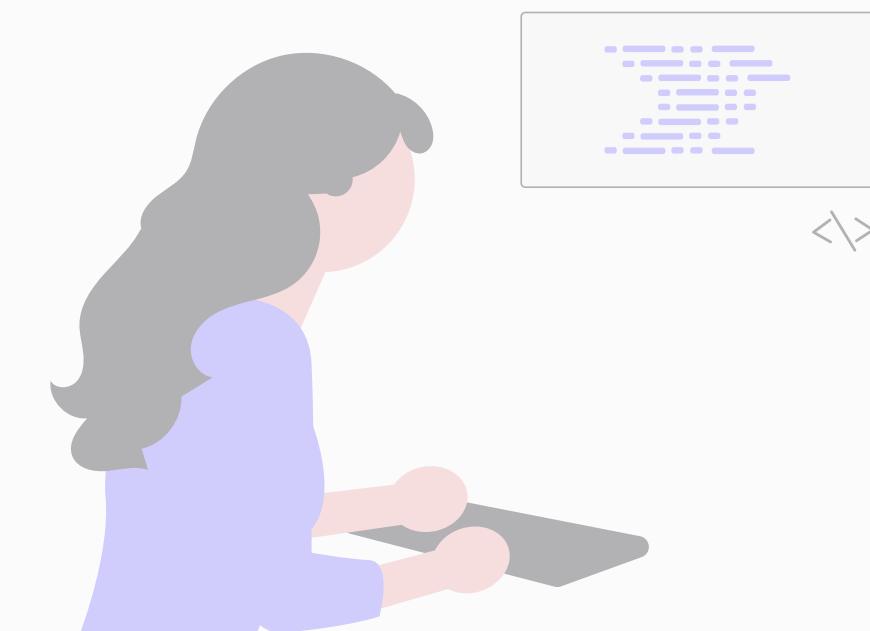
2 Pseudocode

Create a structured, high-level outline for the next agent



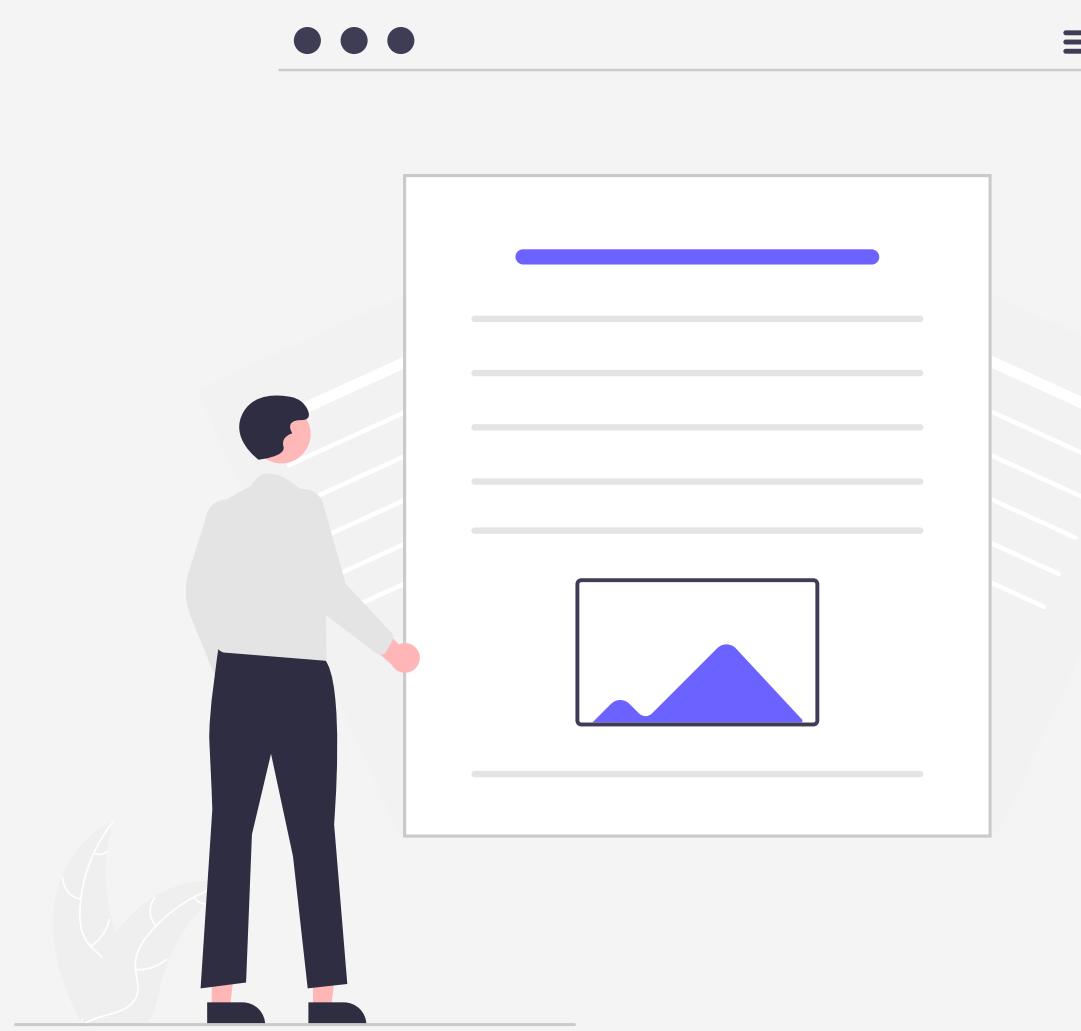
3 Implementation

Consolidate findings, generate final code



1 Planning

Research, retrieve, and understand dense concepts



2 Pseudocode

Create a structured, high-level outline for the next agent



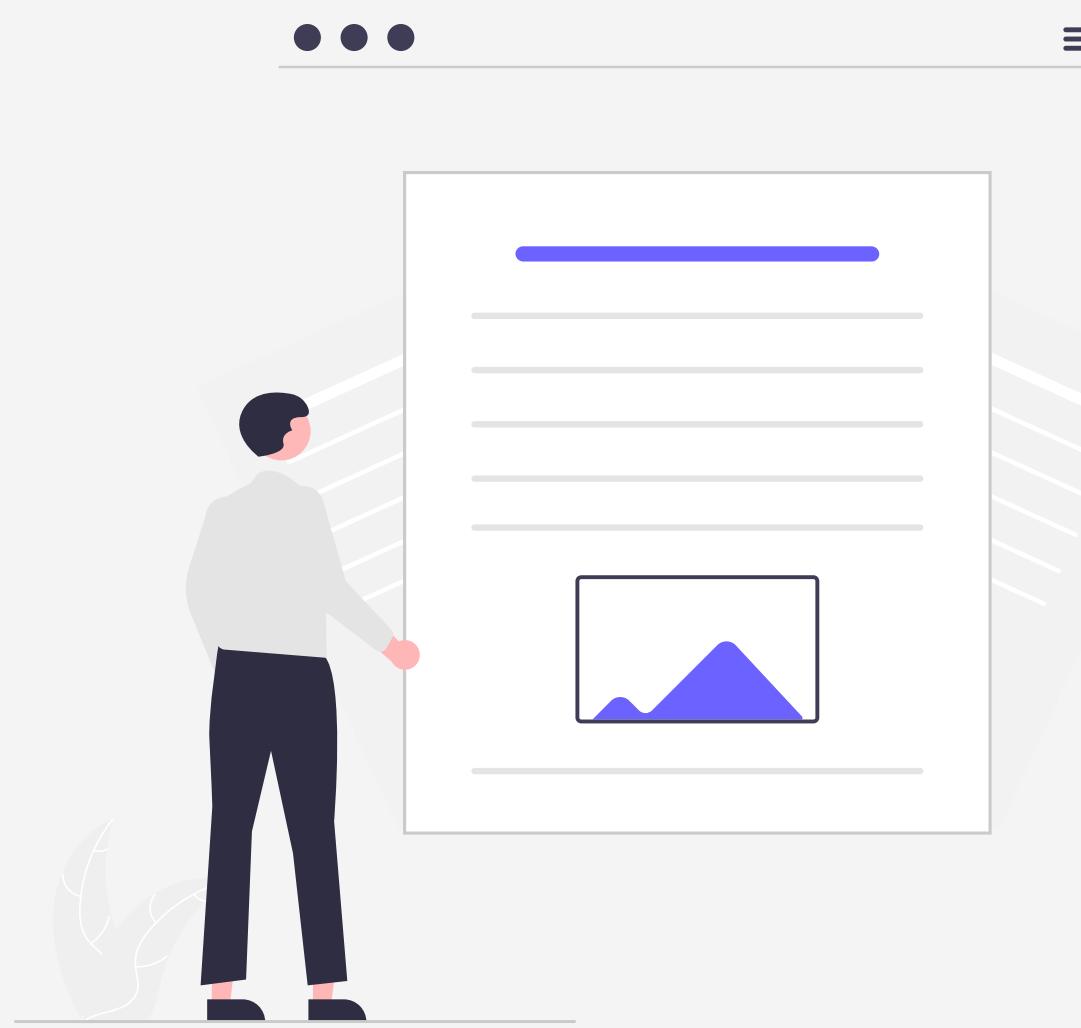
3 Implementation

Consolidate findings, generate final code



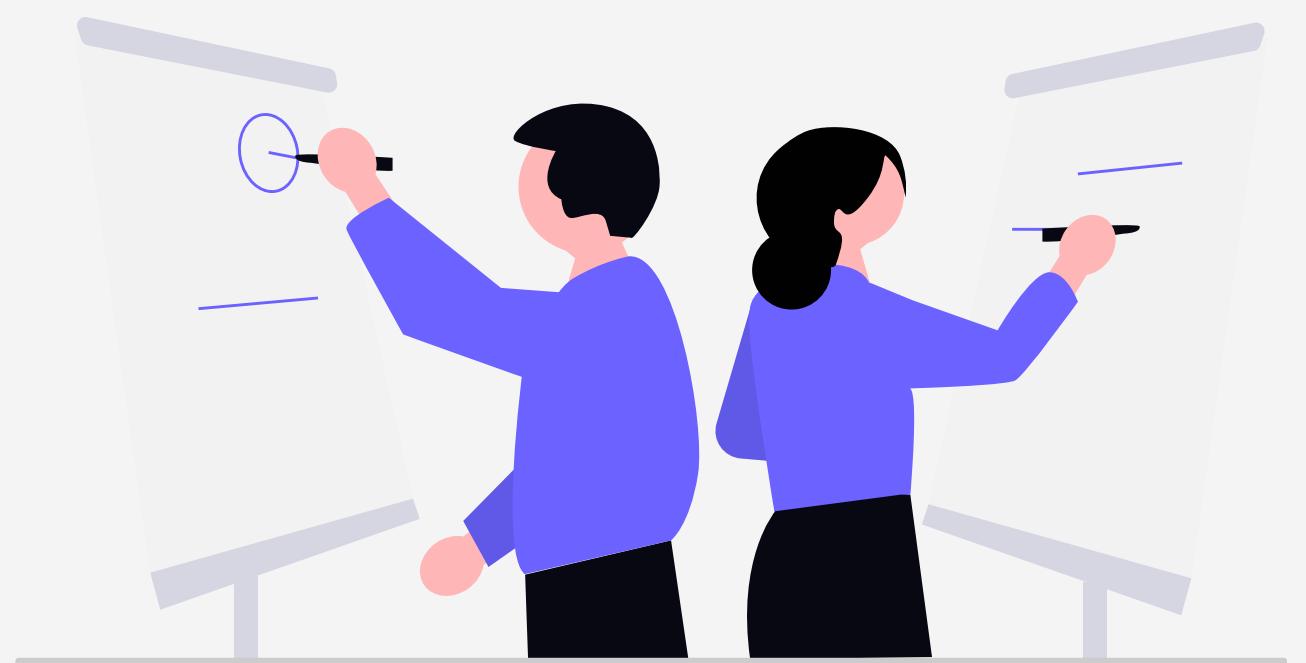
1 Planning

Research, retrieve, and understand dense concepts



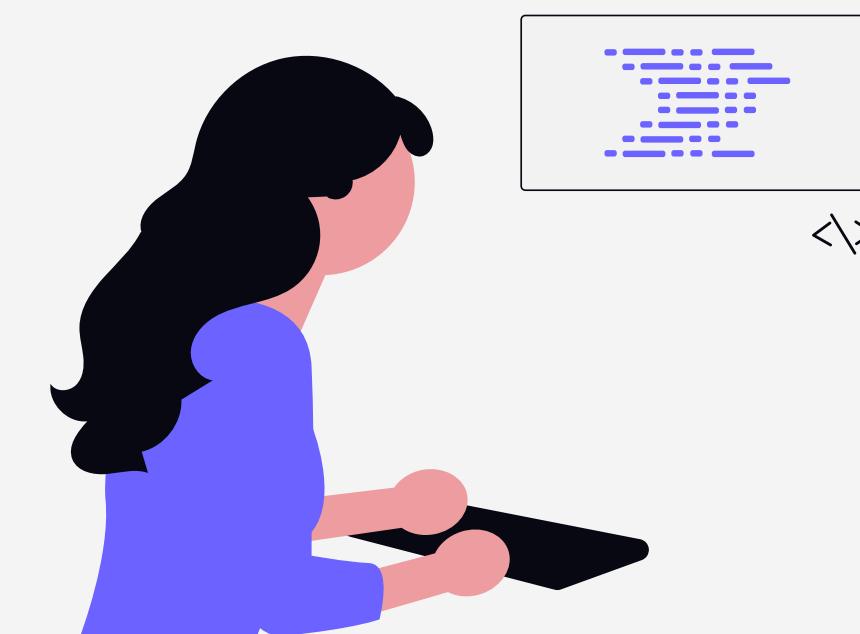
2 Pseudocode

Create a structured, high-level outline for the next agent



3 Implementation

Consolidate findings, generate final code



Handoff and Shared Memory

```
const workflow = multiAgent({
  agents: [plannerAgent, pseudoCodeAgent, codeAgent],
  rootAgent: plannerAgent,
  verbose: true,
  memory: createMemory({
    tokenLimit: 30000,
    shortTermTokenLimitRatio: 0.7,
  })
});
```



Nexus

Code generation started. Check the preview panel to see live progress. You may continue chatting after the code generation is complete.

Thanks!

You're very welcome! I'm glad I could help you with your questions about Apache ResilientDB and blockchain technology. If you have any more questions about ResilientDB, distributed systems, consensus protocols, or any other blockchain-related topics, feel free to ask. I'm here to assist with your research and learning needs. Is there anything specific about the documents you'd like me to help you explore further?

Ask questions about RCC- Resilient Concurrent Consensus for High-Throughput Secure Transaction Processing....

Research

⊕ Search

Preview & Code

</> Python

RCC- Resilient Conc...

⌚ Query

Please implement replica failure.

⌚ Reading Documents ✓ Documents processed

Plan

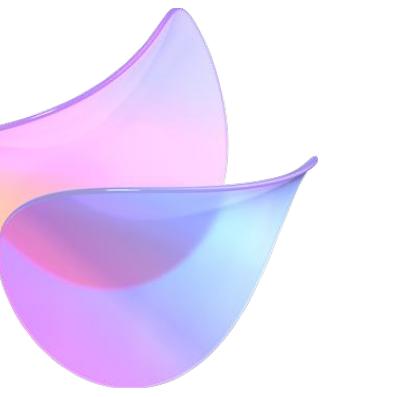
Pseudocode

</> Implementation (Python)

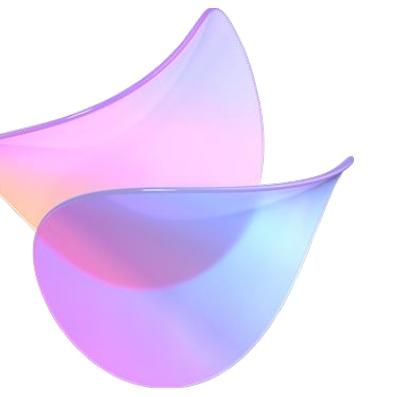
```
import asyncio
import time
import logging
from typing import Dict, List, Set, Optional, Any
from dataclasses import dataclass
from enum import Enum
import random

class InstanceStatus(Enum):
    RUNNING = "running"
    HALTED = "halted"
    RECOVERING = "recovering"

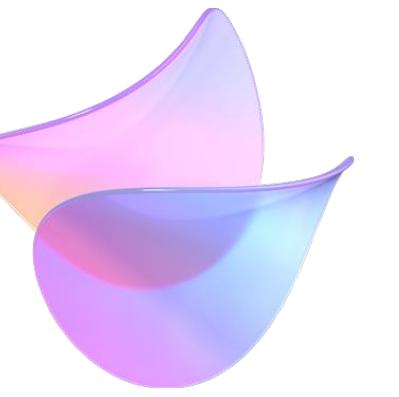
@dataclass
class FailureMessage:
    instance_id: int
    round_number: int
    state: Dict[str, Any]
    sender_id: int
    timestamp: float
```



Conclusion



Future Directions



Retrospective

Thank You