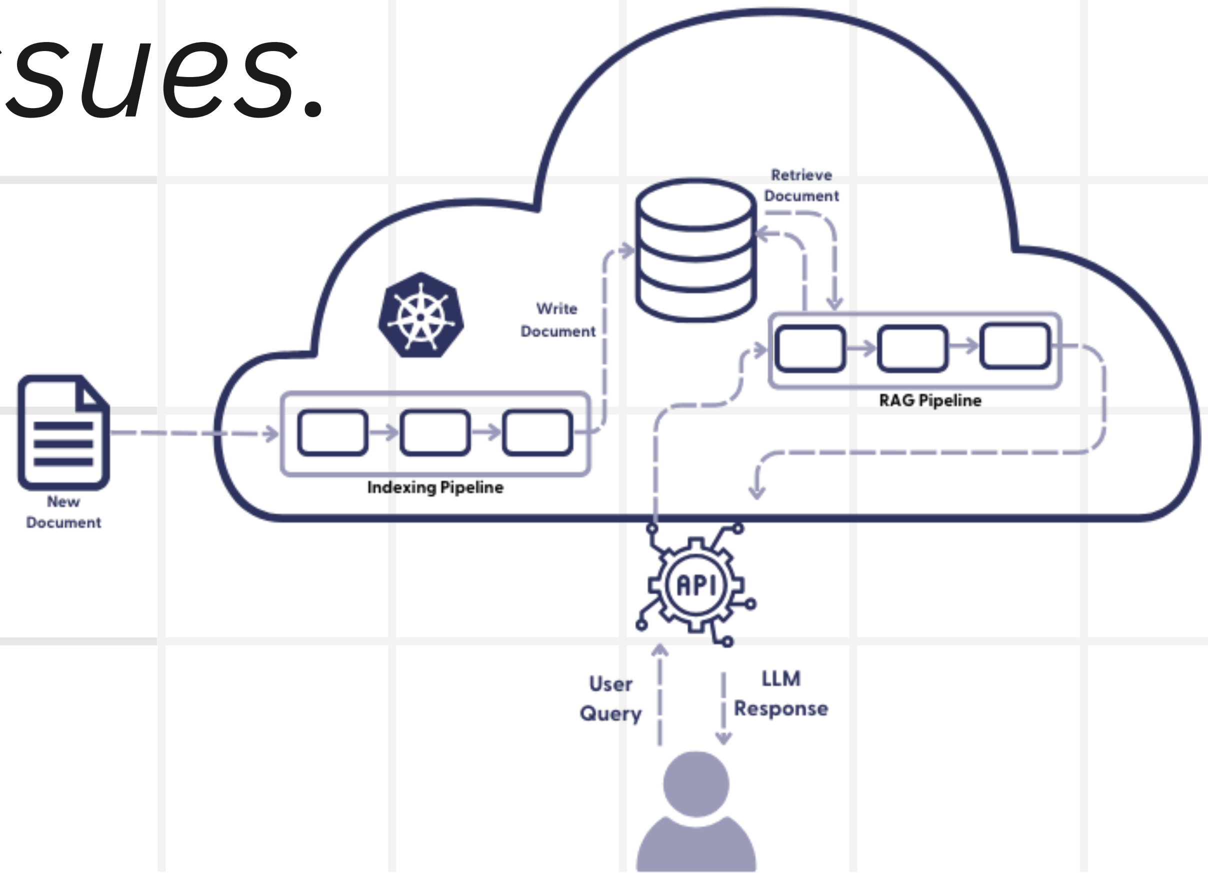
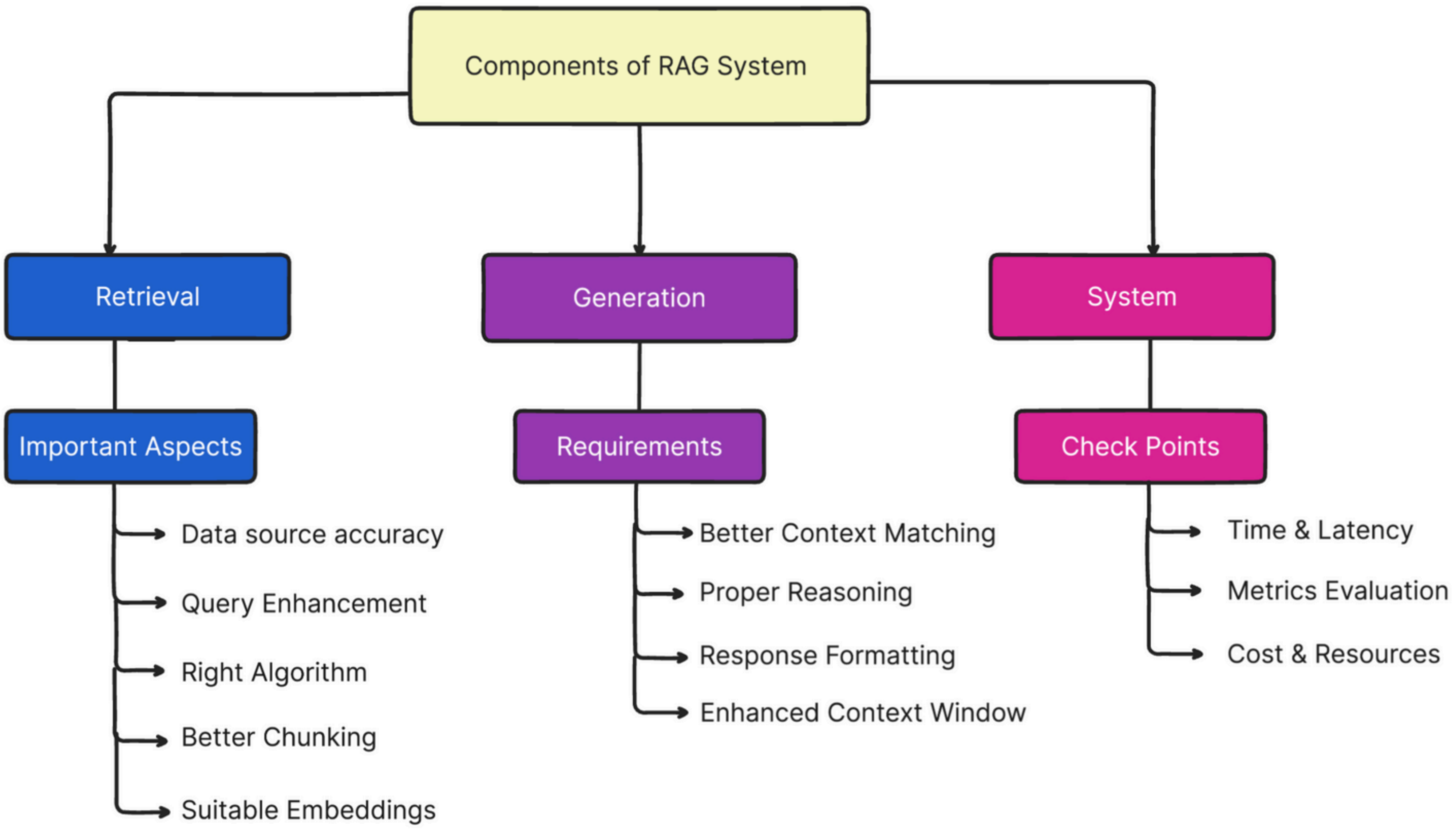
A GUIDE TO BUILDING

A RAG SYSTEM THAT WORKS!

+ Find solutions to fix all your RAG system issues.

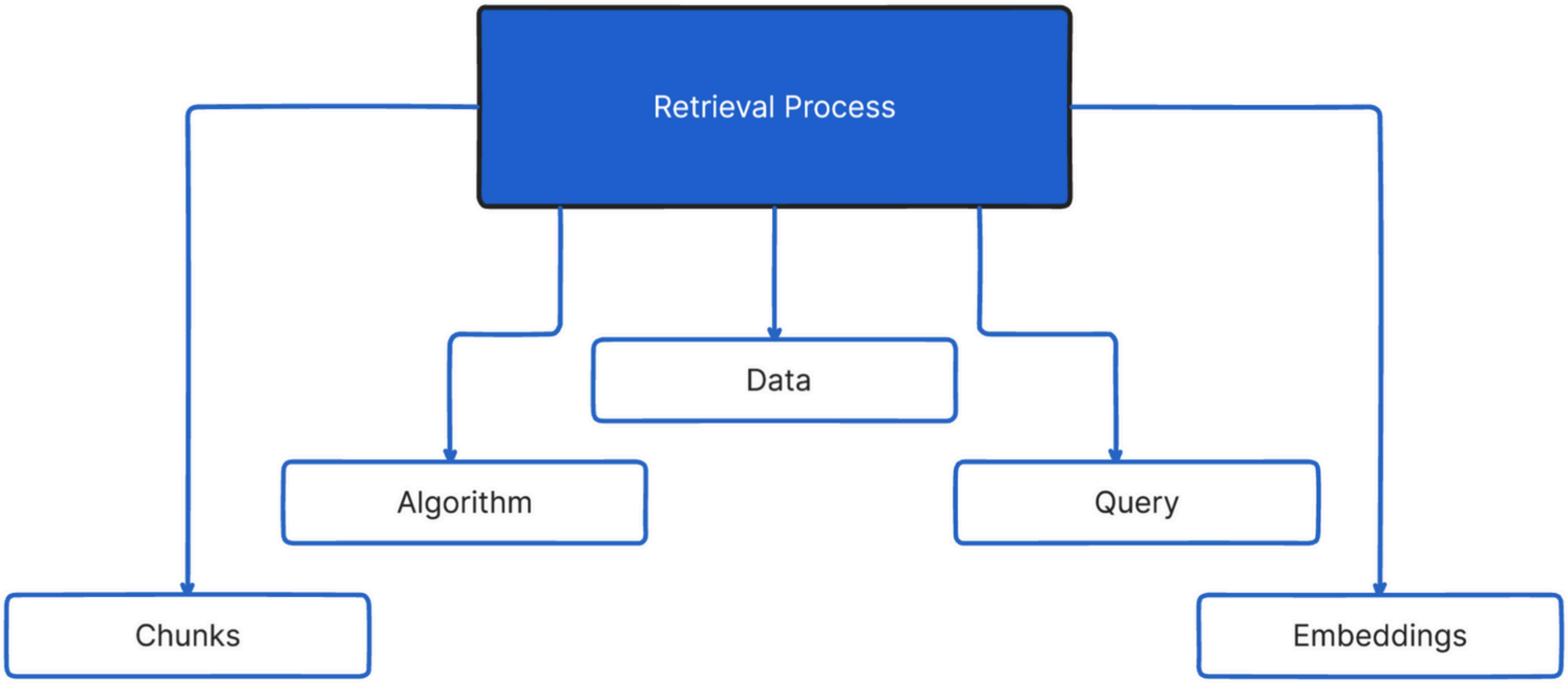


COMPONENTS OF A RAG SYSTEM



Components of RAG Sytems: 1

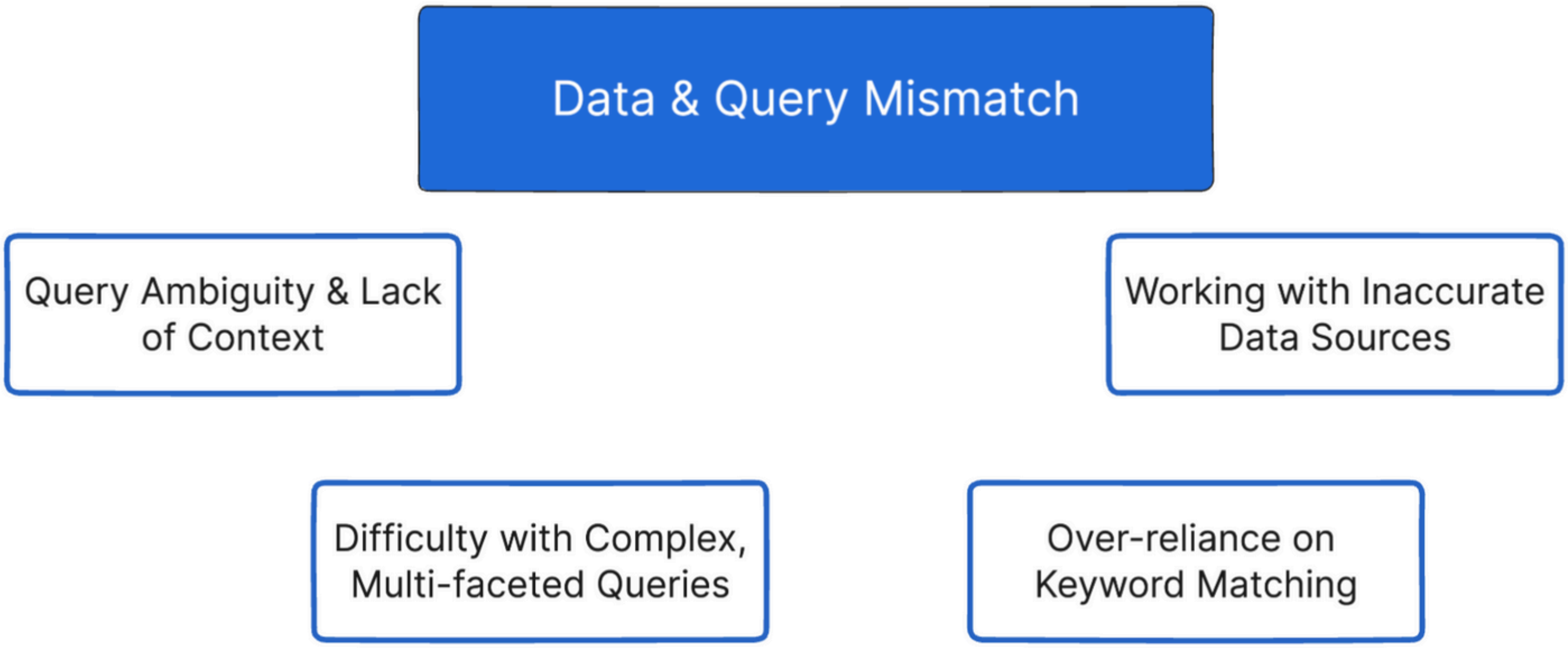
RETRIEVAL PROCESS



Challenges:

- Data & Query Mismatch
- Search/ Retrieval Algorithm Shortcomings
- Challenges in Chunking
- Embedding Problems

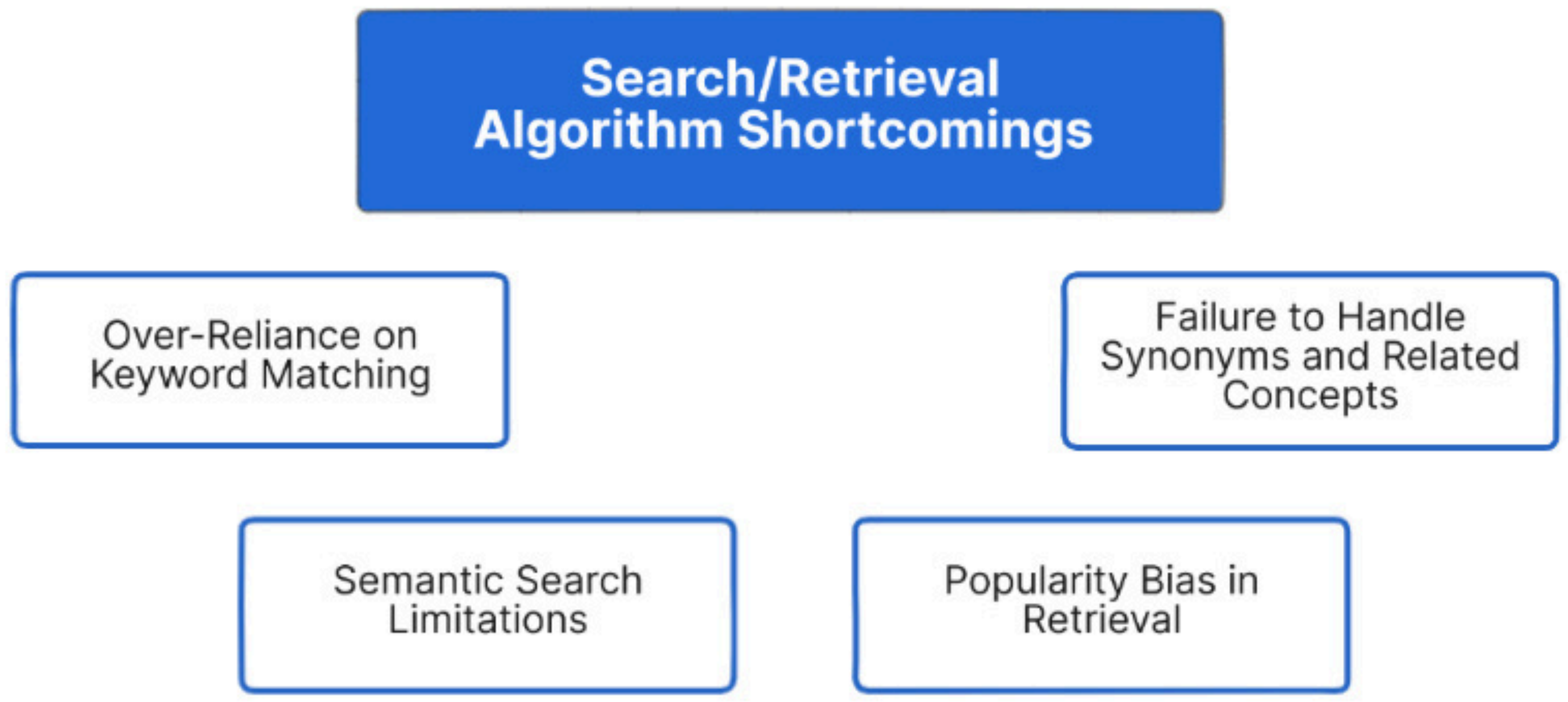
DATA & QUERY


Data & Query Mismatch
Query Ambiguity & Lack of Context
Working with Inaccurate Data Sources
Difficulty with Complex, Multi-faceted Queries
Over-reliance on Keyword Matching

Solution:

- Add Possible Solutions Along with the Query.
- Add Other Similar Queries
- Personalise each query with context
- Consider which data source(s) will be the most relevant for that RAG system.

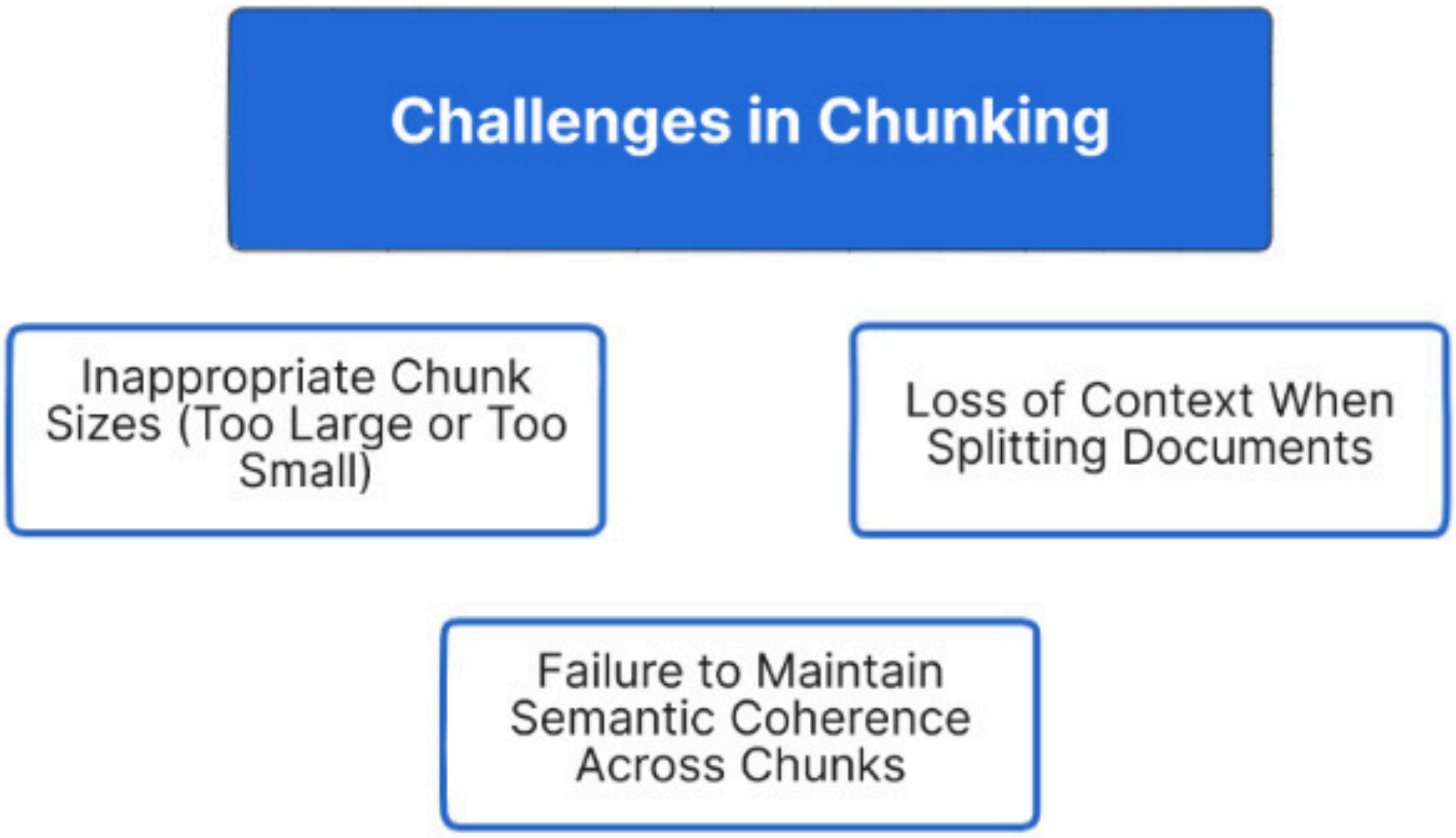
SEARCH/RETRIEVAL ALGORITHMS


Search/Retrieval Algorithm Shortcomings
Over-Reliance on Keyword Matching
Failure to Handle Synonyms and Related Concepts
Semantic Search Limitations
Popularity Bias in Retrieval

Solution:

- Combine keyword (BM25) and semantic search for balanced results.
- Enhance queries (synonyms, context, rephrasing) for better retrieval.
- Use multiple methods (lexical, dense) with re-ranking for improved coverage and relevance.

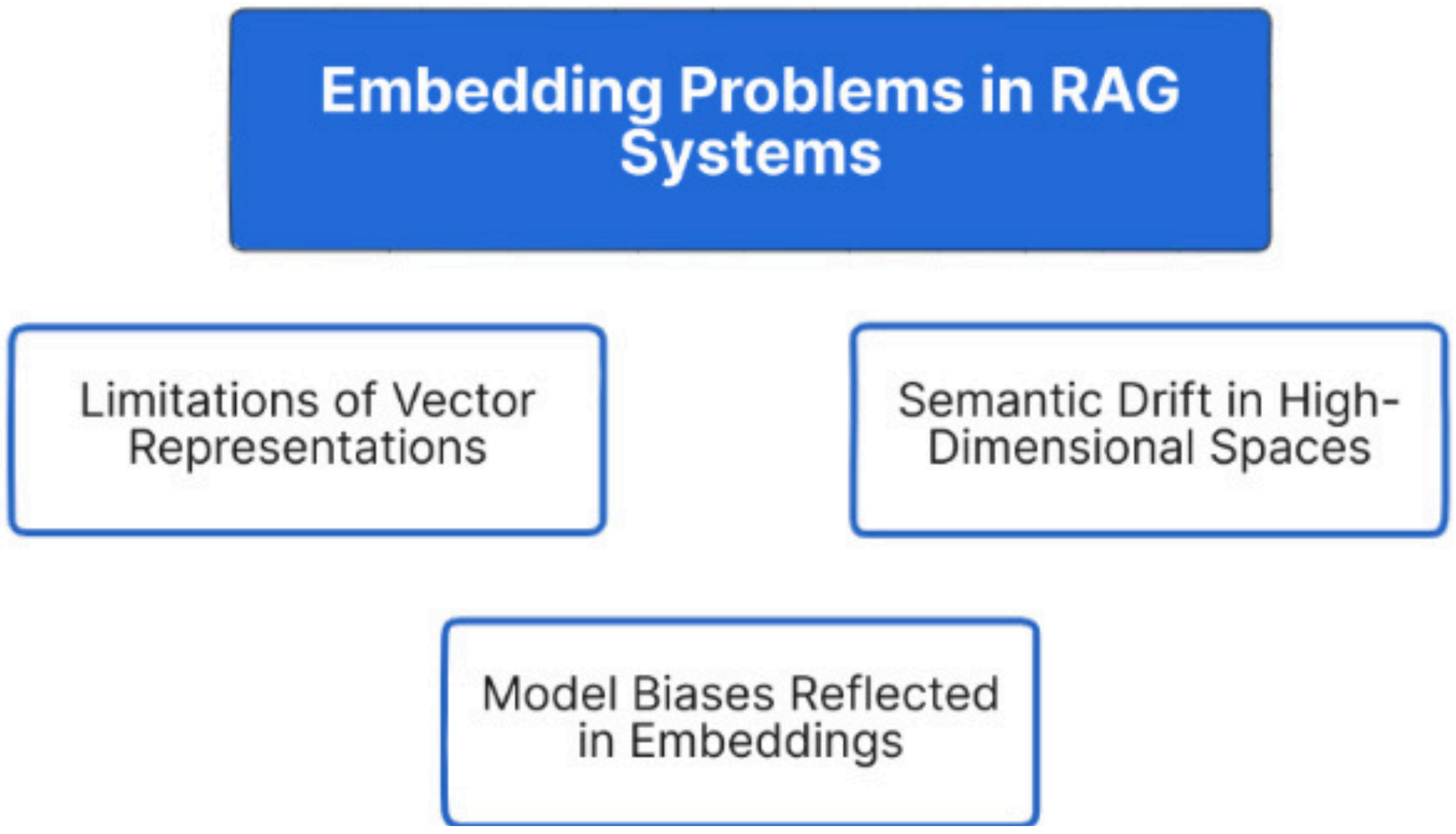
CHUNKING


Challenges in Chunking
Inappropriate Chunk Sizes (Too Large or Too Small)
Loss of Context When Splitting Documents
Failure to Maintain Semantic Coherence Across Chunks

Solution:

- Use NLP to find natural breakpoints, creating meaningful chunks.
- Divide structured documents along existing sections/titles.
- Add overlapping text between chunks to maintain context/references.
- Employ AI to adapt chunk size based on topic shifts for relevance.

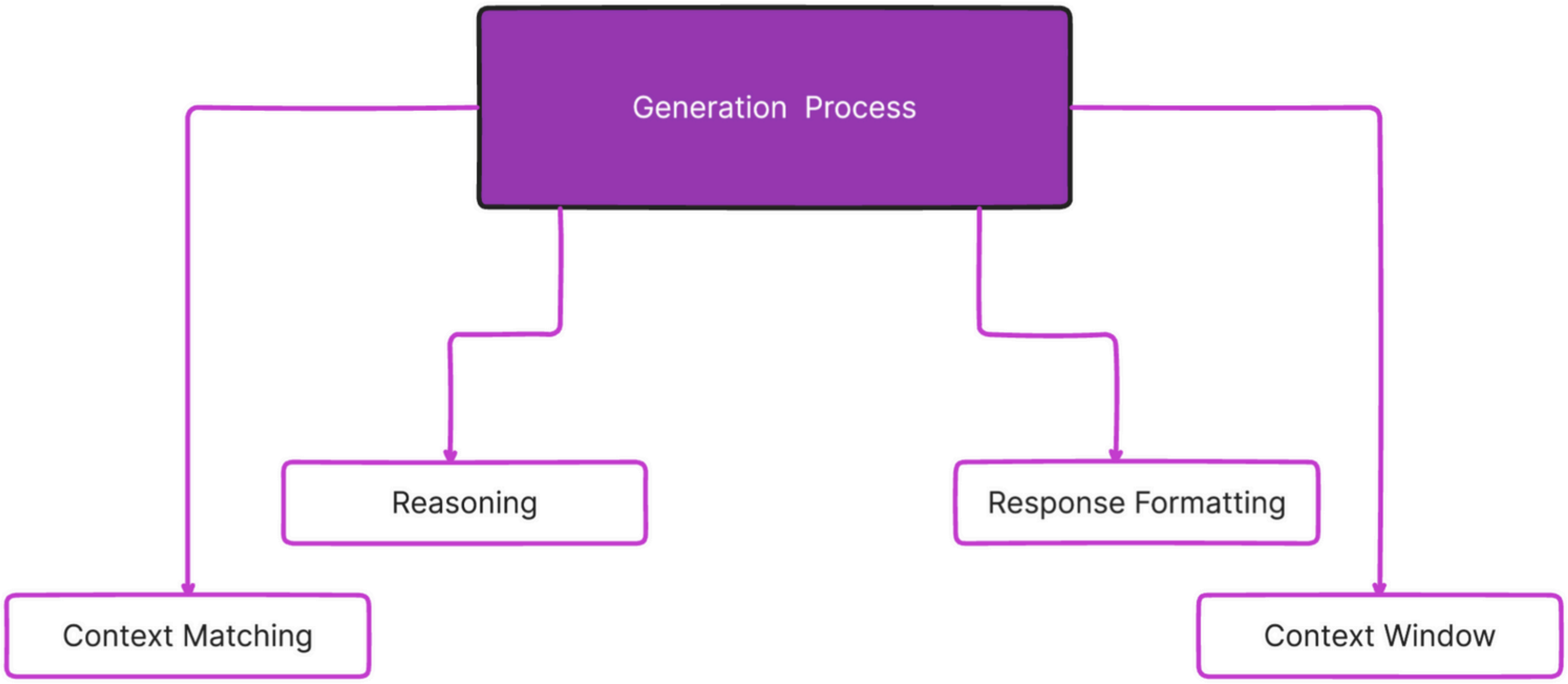
EMBEDDING


Embedding Problems in RAG Systems
Limitations of Vector Representations
Semantic Drift in High-Dimensional Spaces
Model Biases Reflected in Embeddings

Solution:

- Adapt embeddings using domain-specific data for accuracy.
- Re-embed knowledge frequently to stay current.
- Combine traditional and contextual models for better understanding.

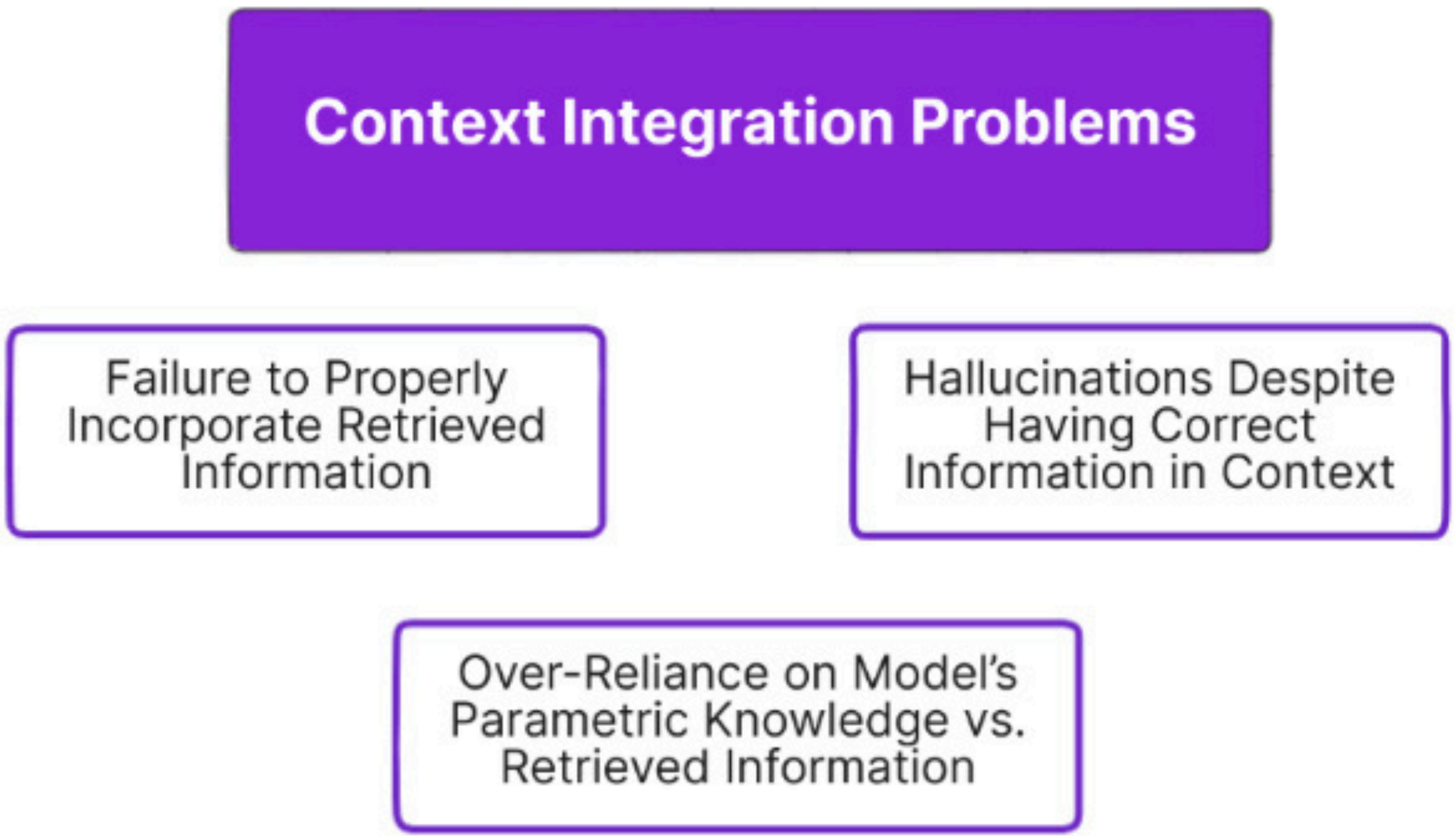
GENERATION PROCESS FAILURES



Reasons:

- Context Integration Problems
- Reasoning Limitation
- Response Formatting Issues
- Context Window Utilization

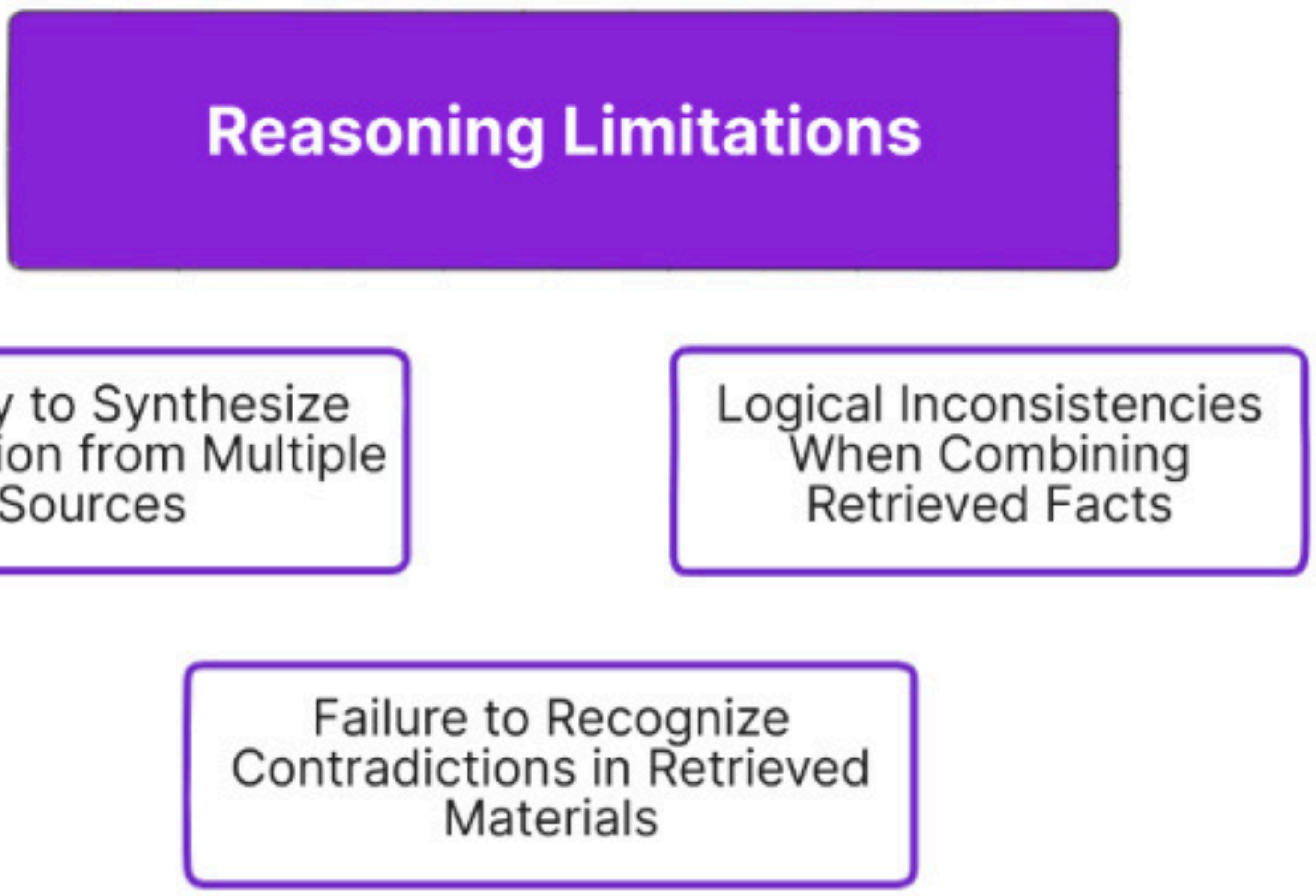
CONTEXT INTEGRATION


Context Integration Problems
Failure to Properly Incorporate Retrieved Information
Hallucinations Despite Having Correct Information in Context
Over-Reliance on Model's Parametric Knowledge vs. Retrieved Information

Solution:

- Supervised FineTuning for Better Grounding
- Fact Verification Post-Processing
- Retrieval-Aware Training

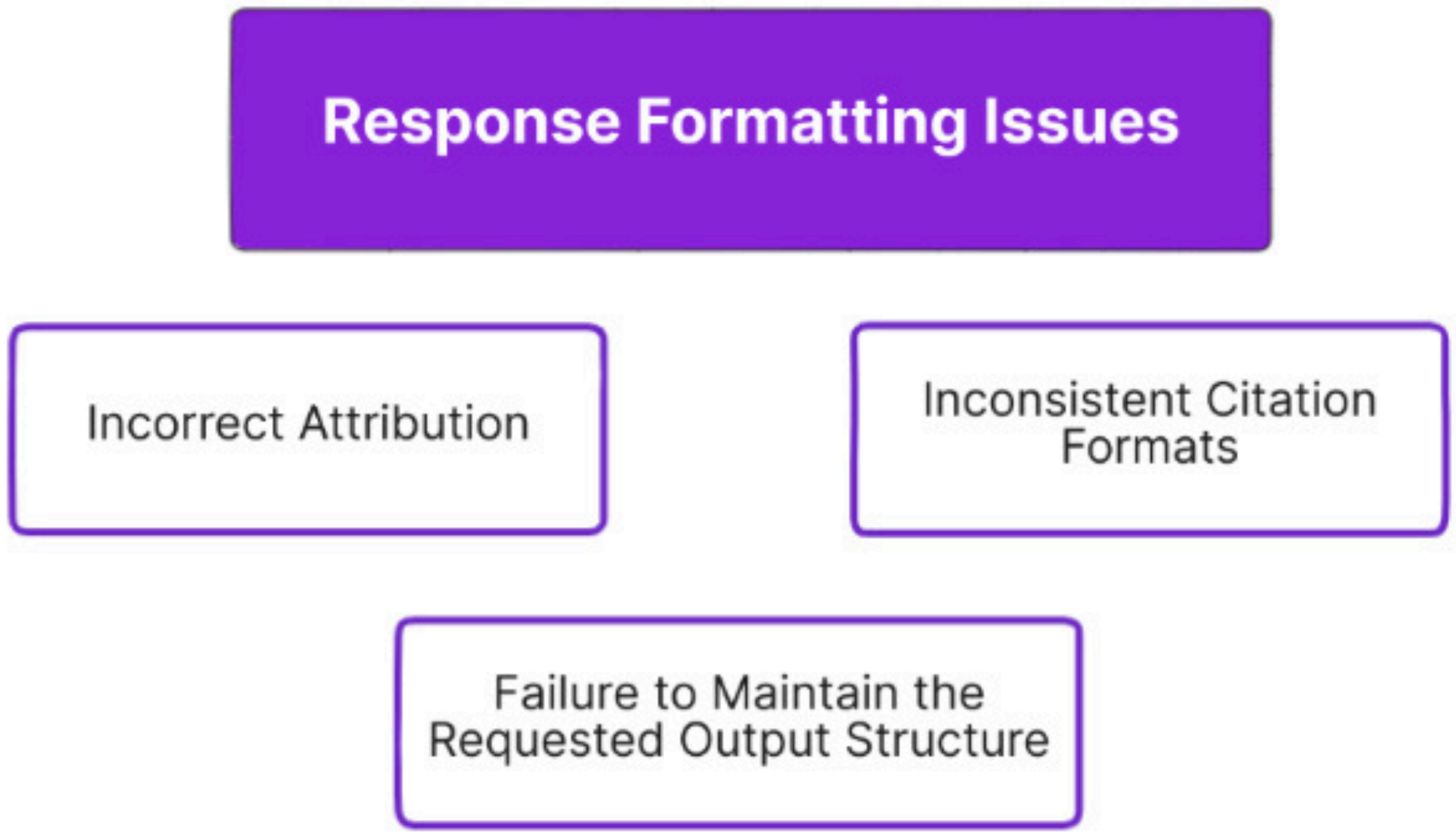
REASONING


Reasoning Limitations
Inability to Synthesize Information from Multiple Sources
Logical Inconsistencies When Combining Retrieved Facts
Failure to Recognize Contradictions in Retrieved Materials

Solution:

- Chain-of-thought Prompting
- Multi-step Reasoning Frameworks
- Contradiction Detection Mechanisms

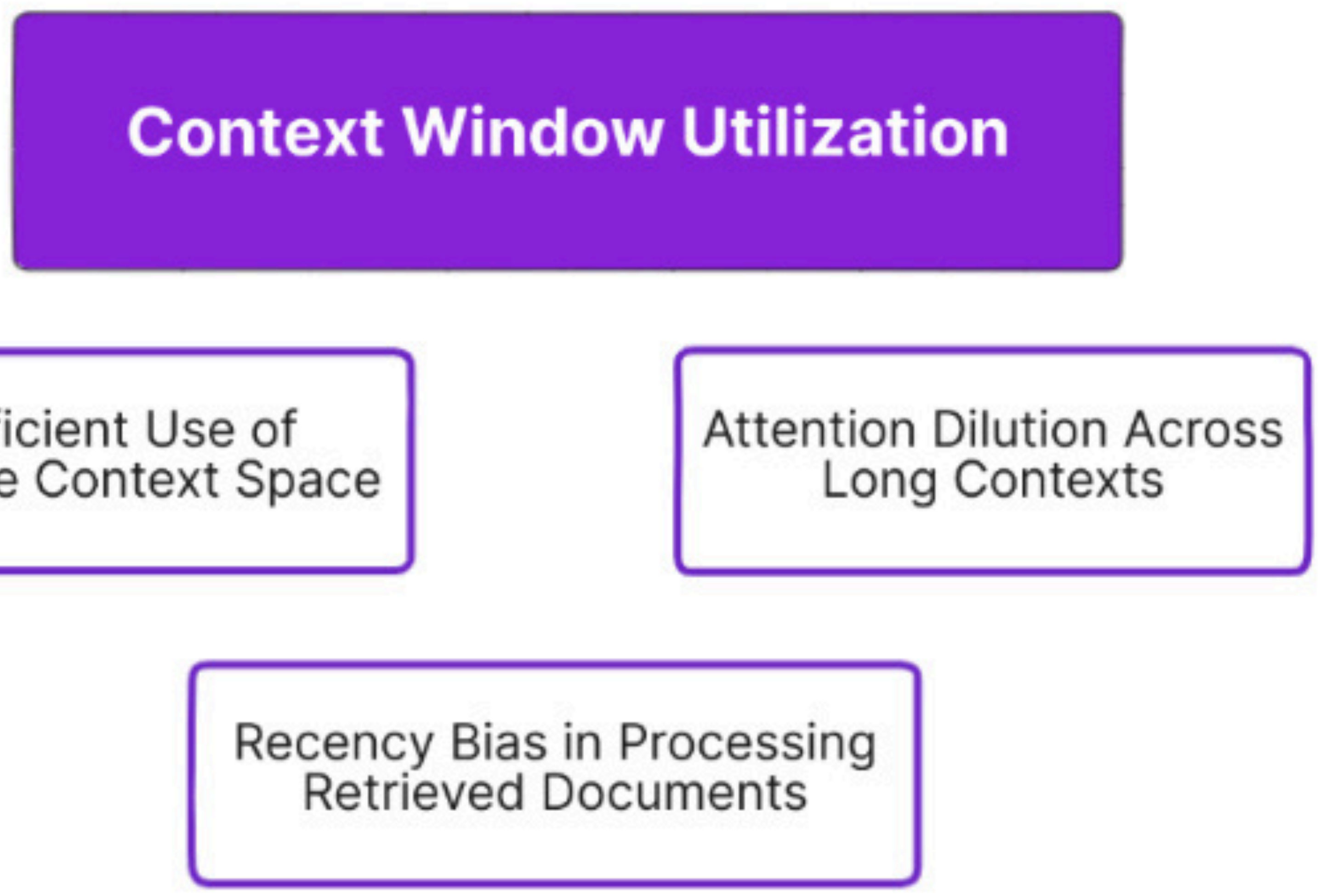
RESPONSE FORMATTING


Response Formatting Issues
Incorrect Attribution
Inconsistent Citation Formats
Failure to Maintain the Requested Output Structure

Solution:

- Enforce structured formatting by using predefined templates
- model with prompt engineering for output formatting
- Automatically checks and corrects attribution, citations, and structure

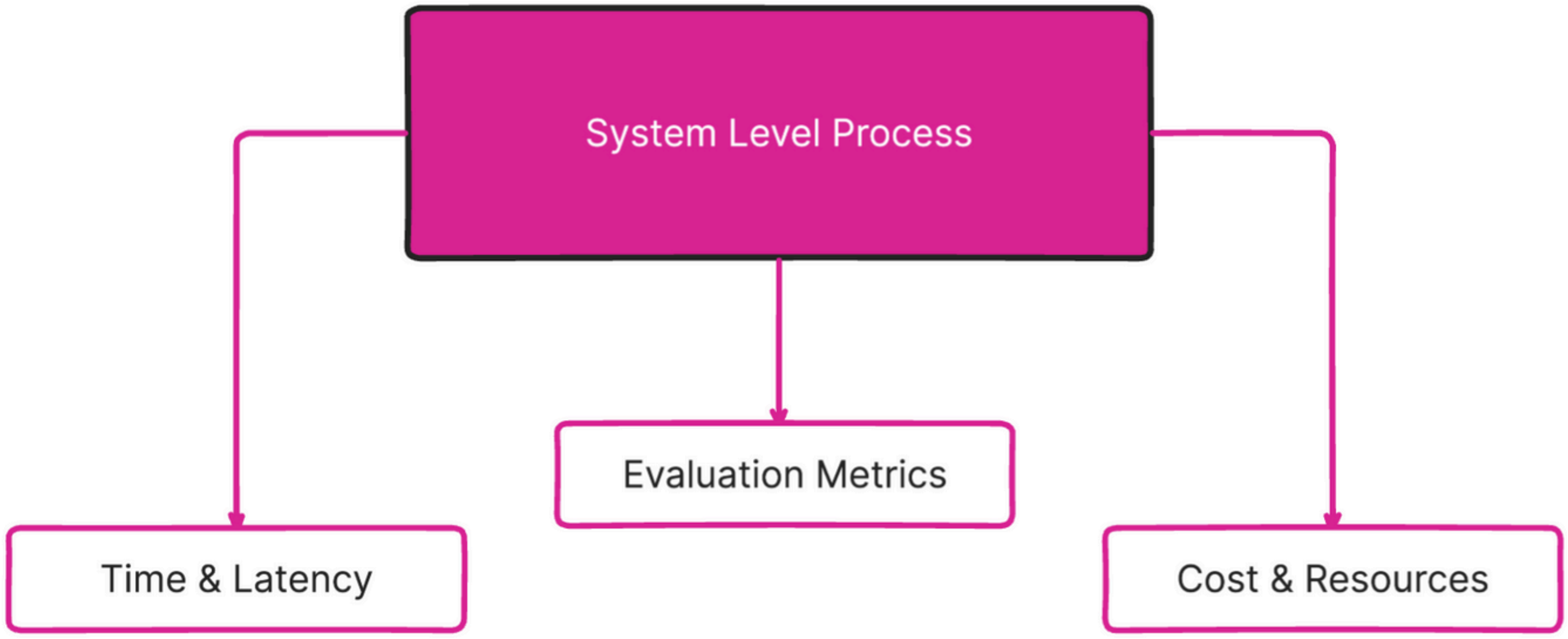
CONTEXT WINDOW


Context Window Utilization
Inefficient Use of Available Context Space
Attention Dilution Across Long Contexts
Recency Bias in Processing Retrieved Documents

Solution:

- Position key information where the model focuses most
- Maximize value by prioritizing important content and reducing redundancy
- Use structured prompts to direct the model's focus to essential sections.

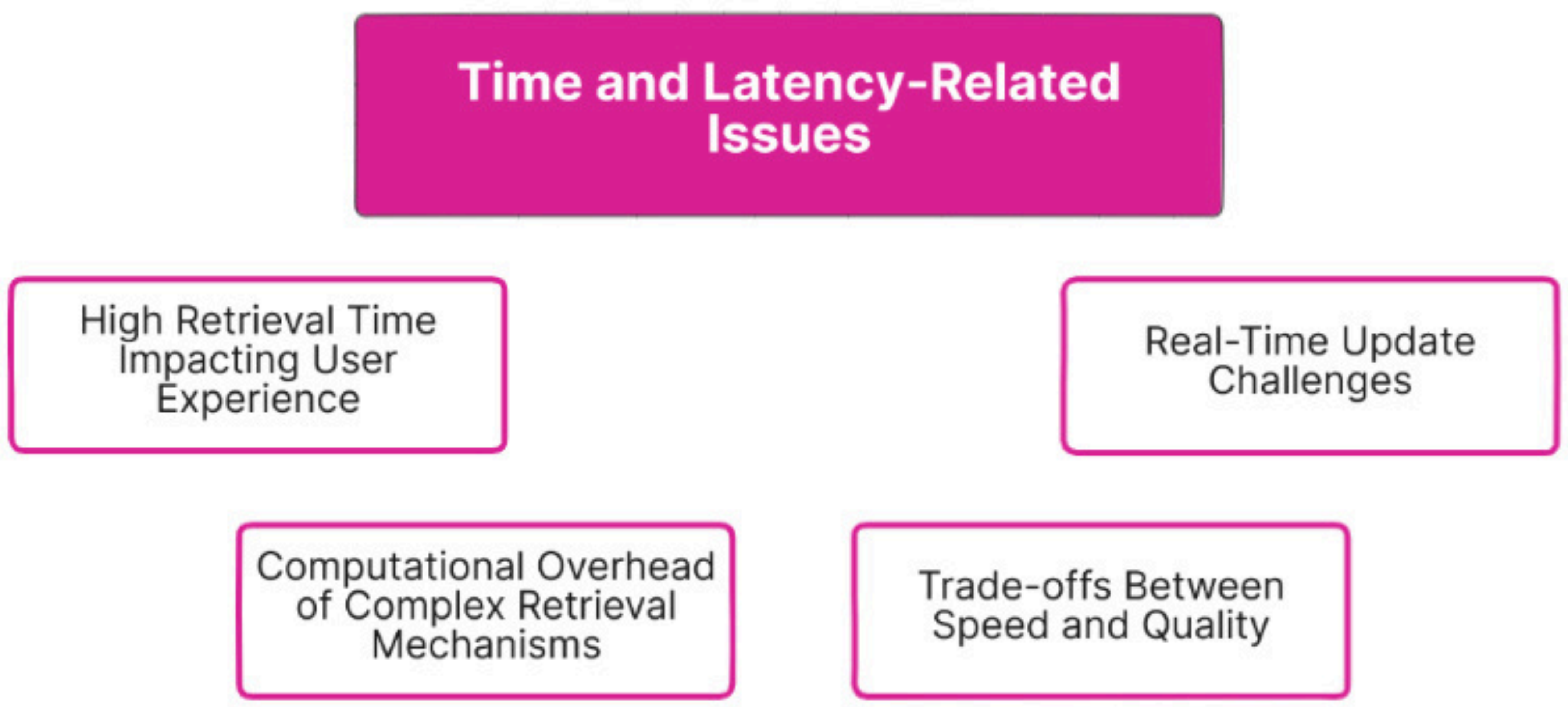
SYSTEM LEVEL FAILURES



Reasons:

- Time & Latency Related Issues
- Evaluation Challenges
- Architectural Limitations
- Cost & Resource Efficiency

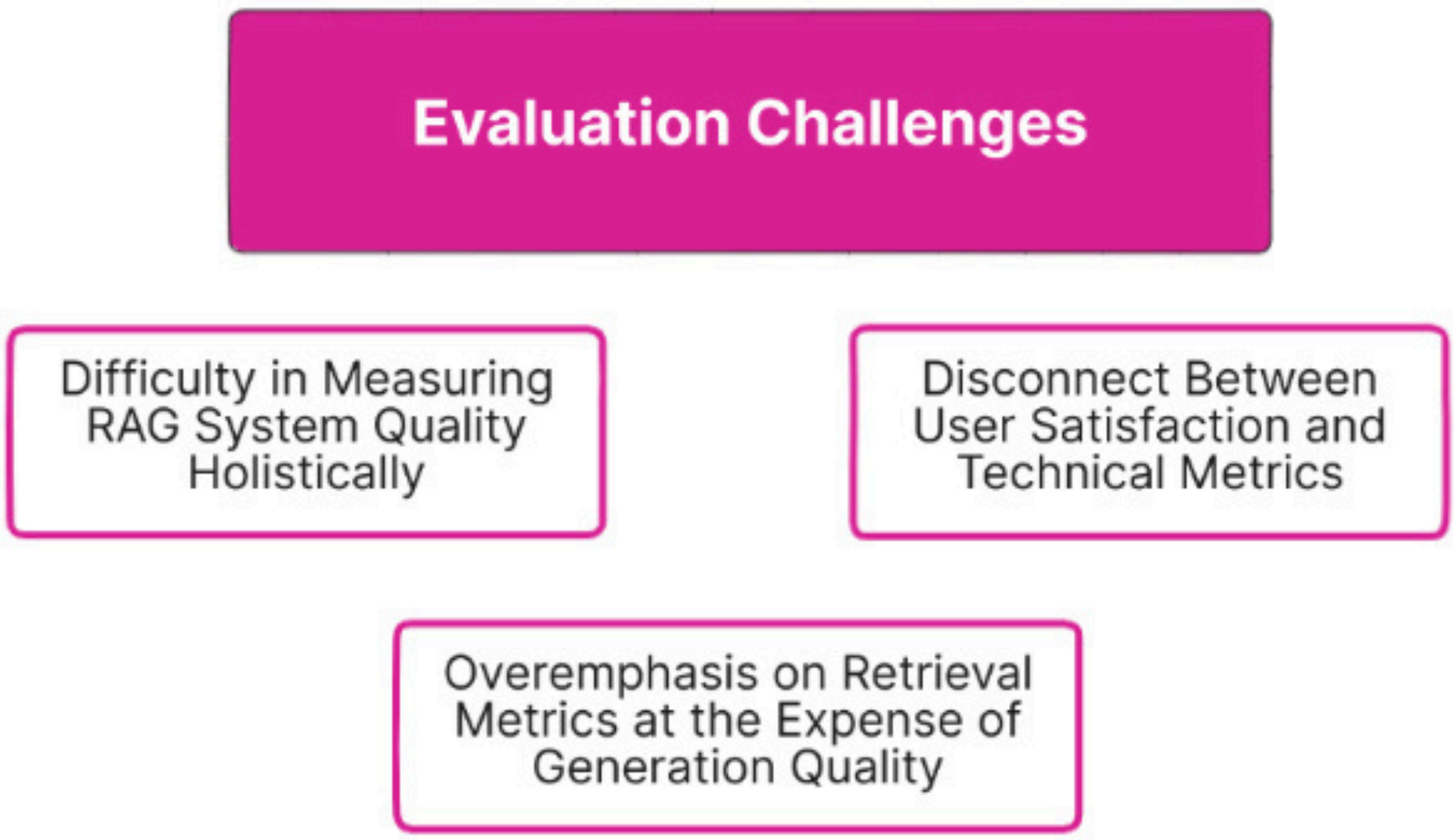
TIME & LATENCY RELATED ISSUES


Time and Latency-Related Issues
High Retrieval Time Impacting User Experience
Real-Time Update Challenges
Computational Overhead of Complex Retrieval Mechanisms
Trade-offs Between Speed and Quality

Solution:

- Store common data in memory for faster access.
- Adjust retrieval complexity based on the query.
- Get quick results first, then refine if necessary.
- Refresh knowledge in the background without slowing responses.

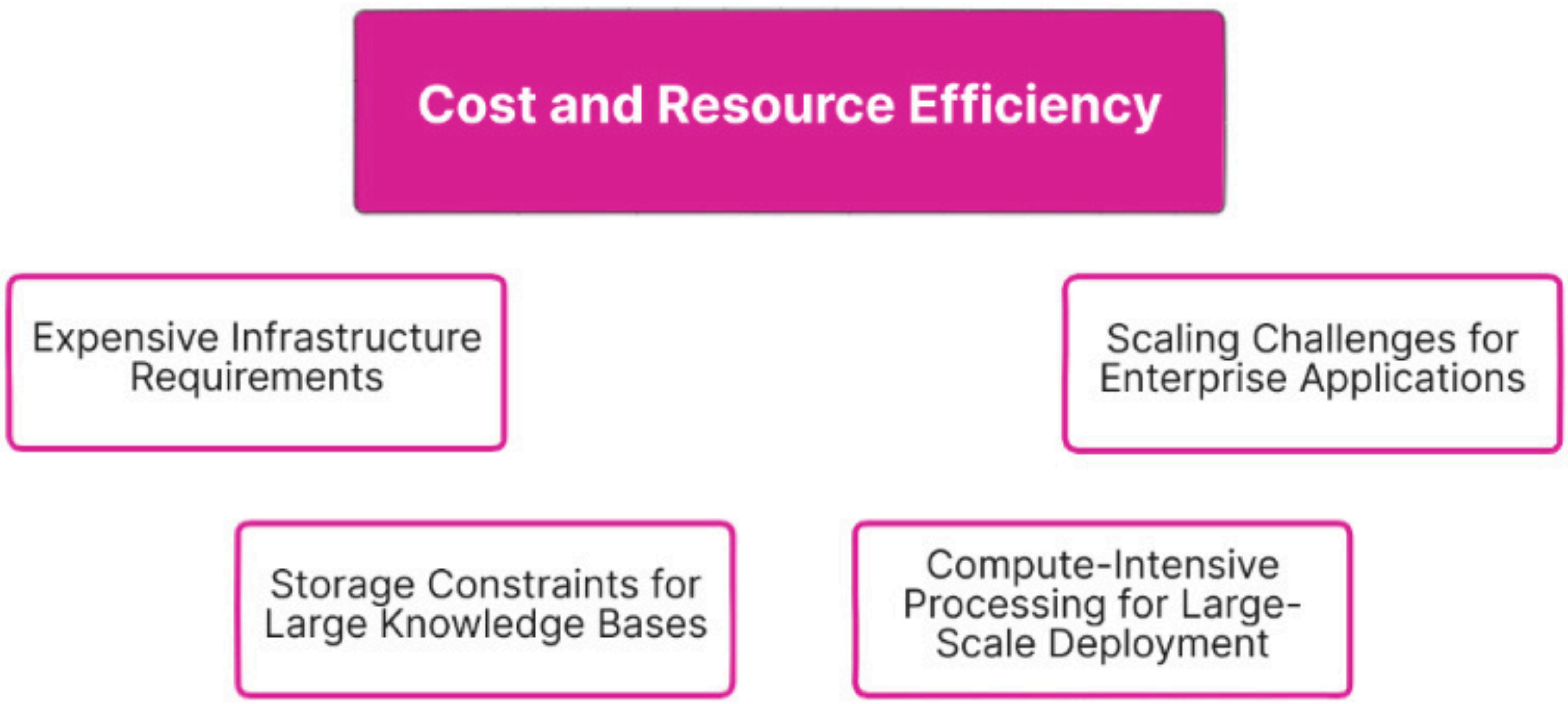
EVALUATION CHALLENGES


Evaluation Challenges
Difficulty in Measuring RAG System Quality Holistically
Disconnect Between User Satisfaction and Technical Metrics
Overemphasis on Retrieval Metrics at the Expense of Generation Quality

Solution:

- Assess RAG using retrieval quality, accuracy, coherence, and user engagement.
- Measure user satisfaction via A/B tests, preference modeling, and feedback.
- Test system robustness and grounding by varying retrieval conditions

COST & RESOURCE EFFICIENCY


Cost and Resource Efficiency
Expensive Infrastructure Requirements
Scaling Challenges for Enterprise Applications
Storage Constraints for Large Knowledge Bases
Compute-Intensive Processing for Large-Scale Deployment

Solution:

- Use fast, approximate search first, then precise retrieval.
- Compress large models into smaller, efficient versions.
- Utilize methods like BM25/hybrid search to reduce compute/memory.
- Speed up retrieval and lower costs with optimized indexes (ANN, etc.)

READ THE BLOG TO UNDERSTAND HOW TO BUILD AN EFFICIENT RAG SYSTEM

READ MORE →

Build a RAG System That Works!