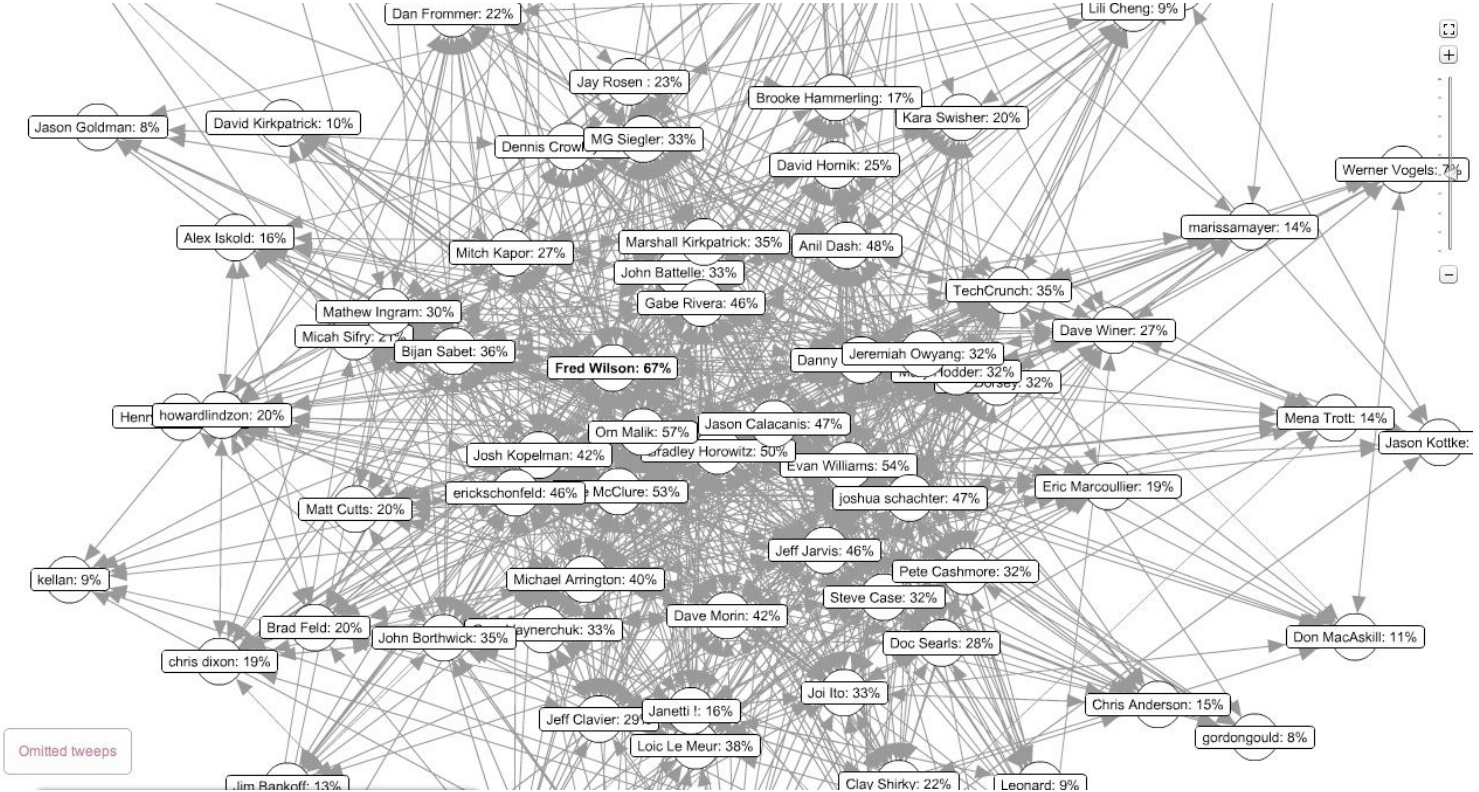
GraphFlow

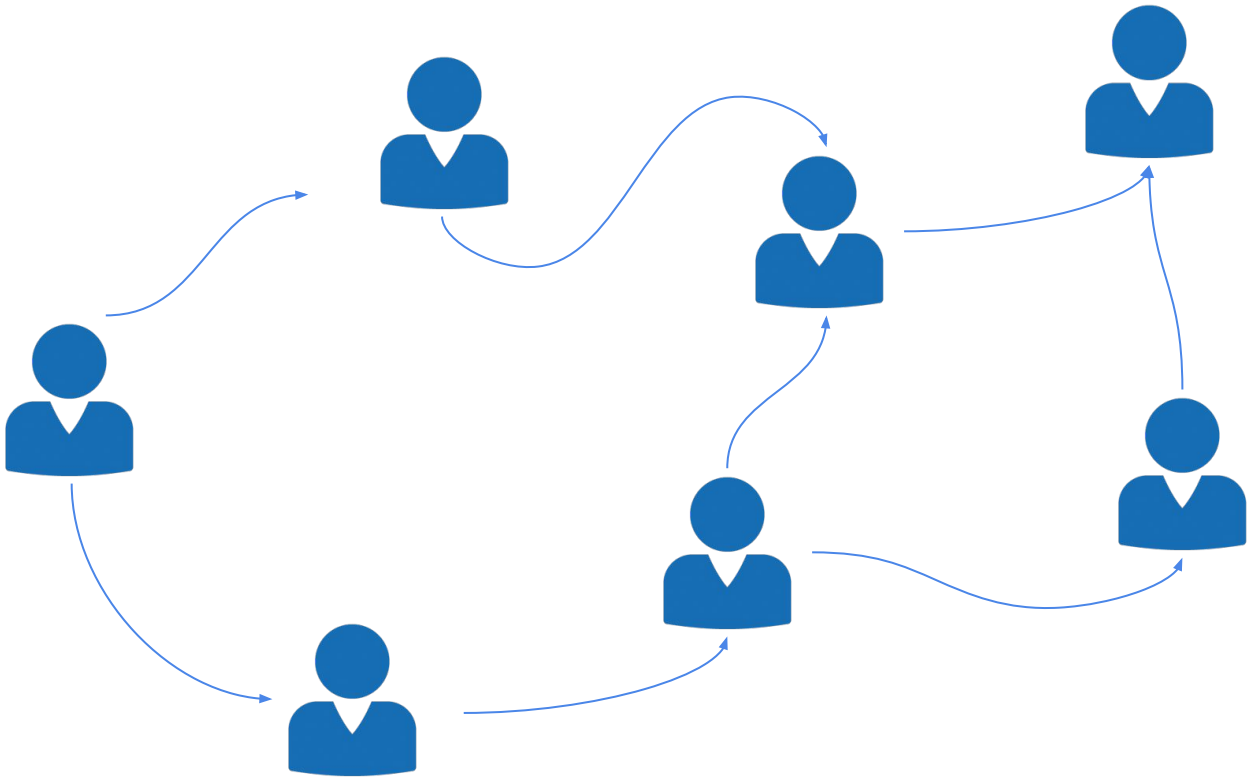
A system for active graph processing

Chathura Kankanamge
Siddhartha Sahu

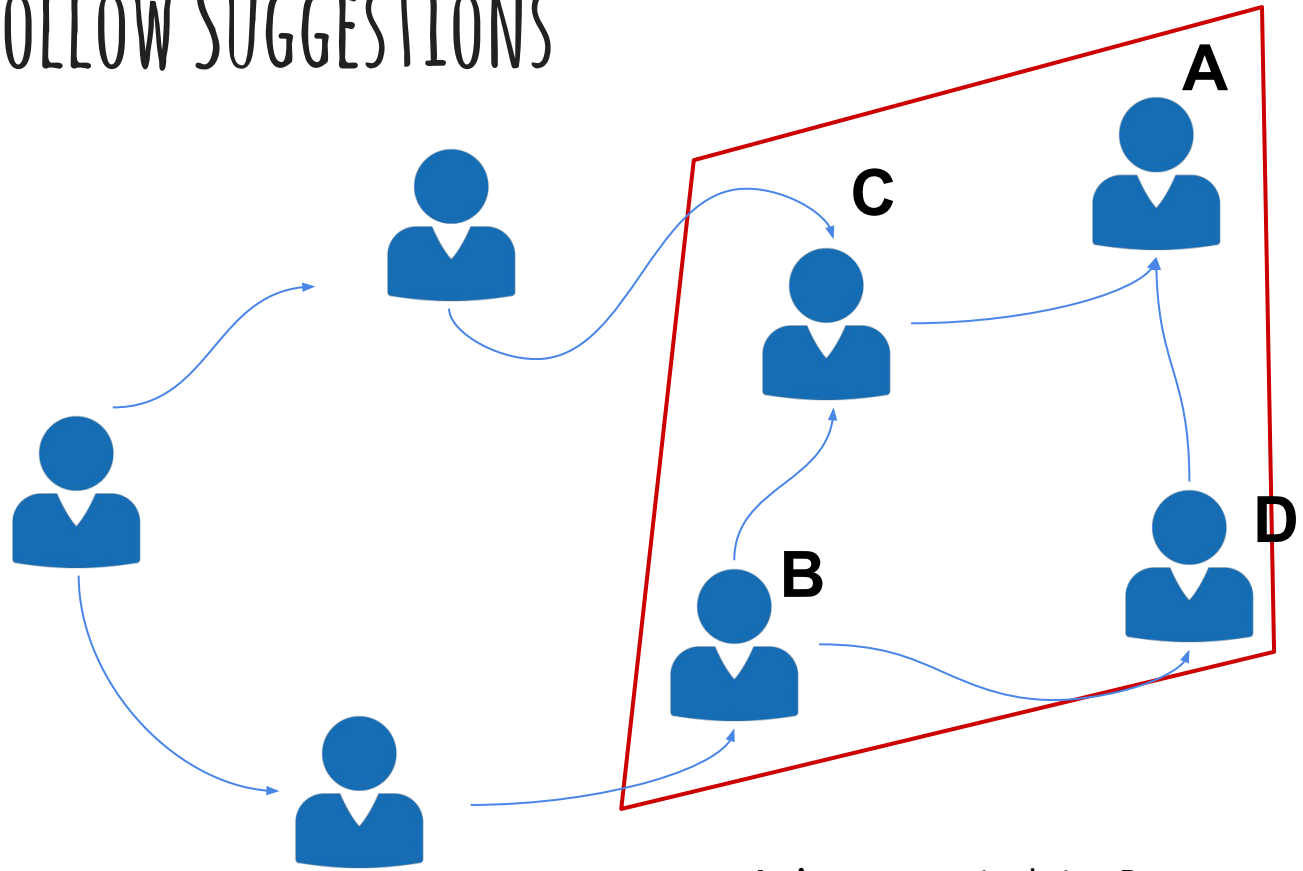
The Twitter Connections Graph



Twitter Follow Suggestions

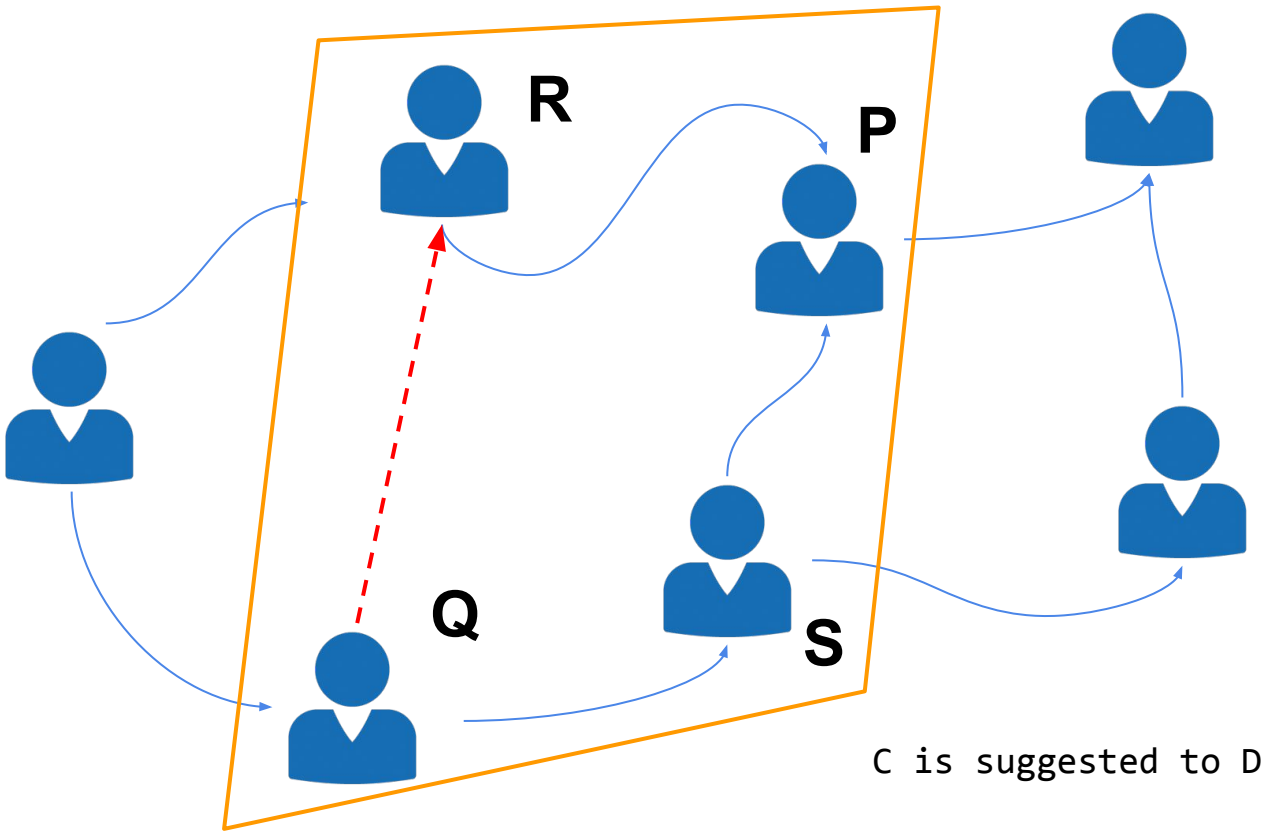


Twitter Follow Suggestions



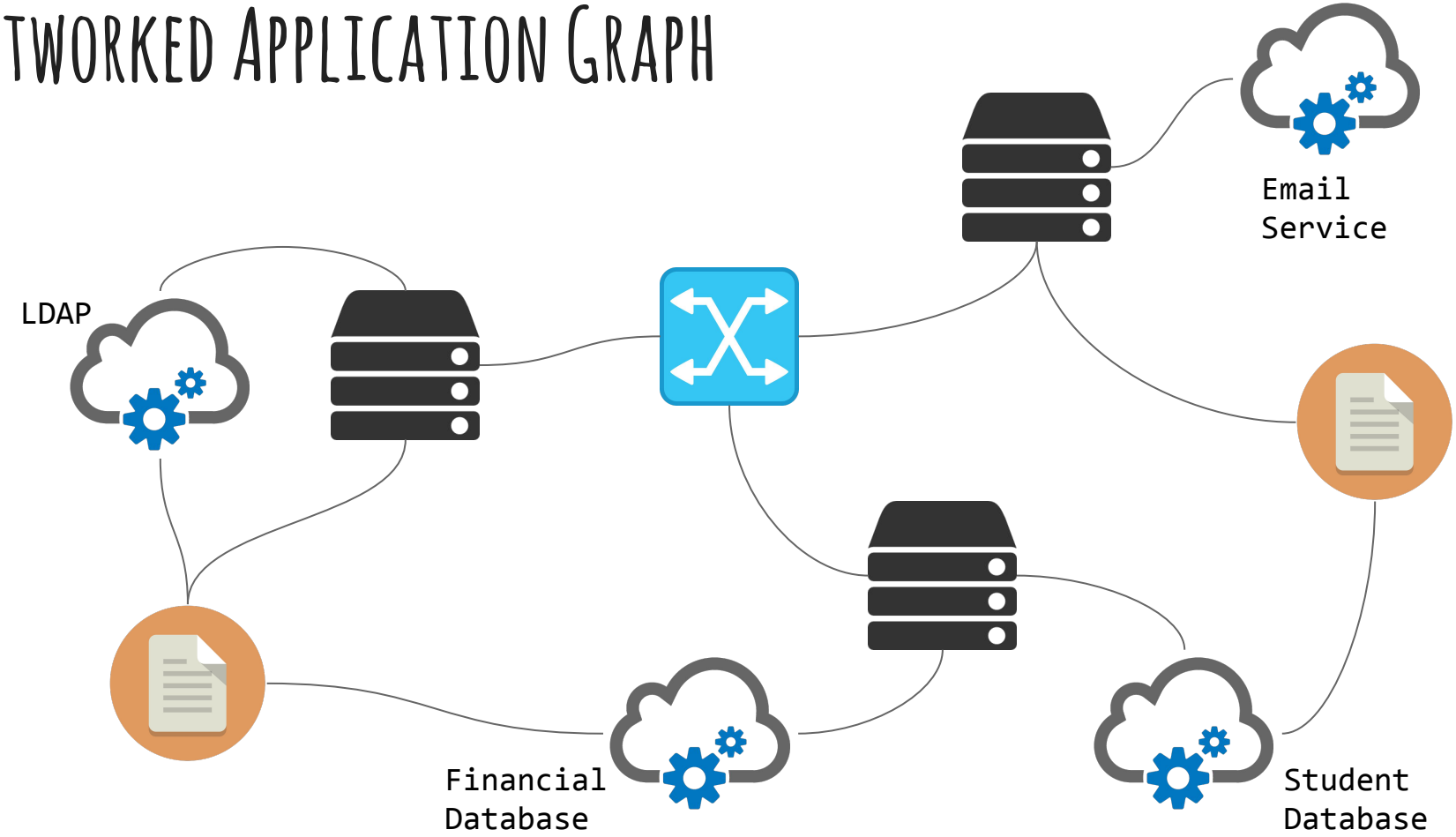
A is suggested to B.

Twitter Follow Suggestions

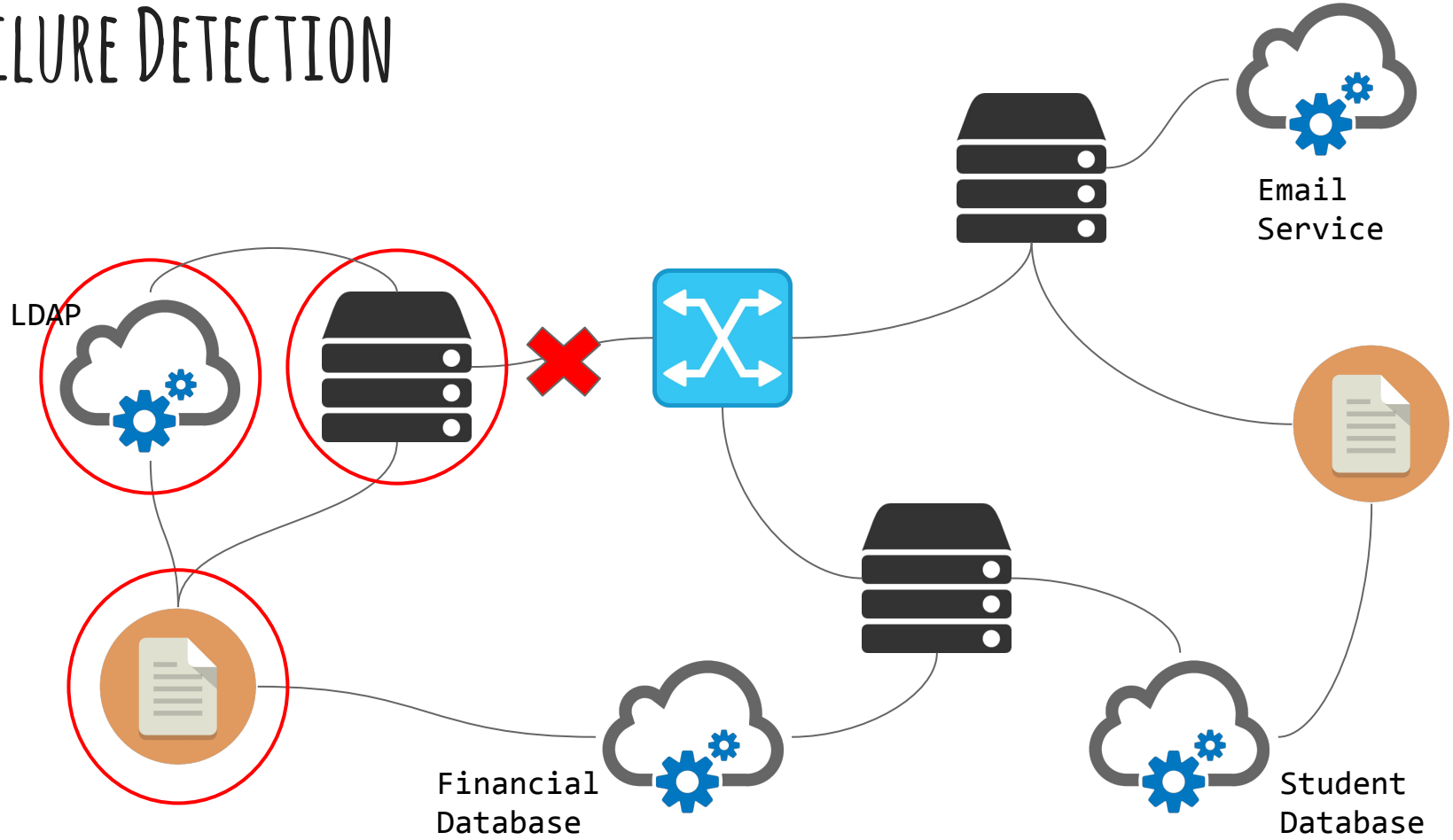


C is suggested to D.

Networked Application Graph



Failure Detection



Problem Statement

Run continuous queries on a large graph database with rapidly changing data and perform user defined actions based on the results.

Examples:

1. New follow suggestions.
2. Cancel transactions which create a circular pattern and email account manager.
3. Send emails to application admins upon server failure.

Previous Solutions

Polling queries

Application-specific optimizations

    Eg. Graphjet at Twitter

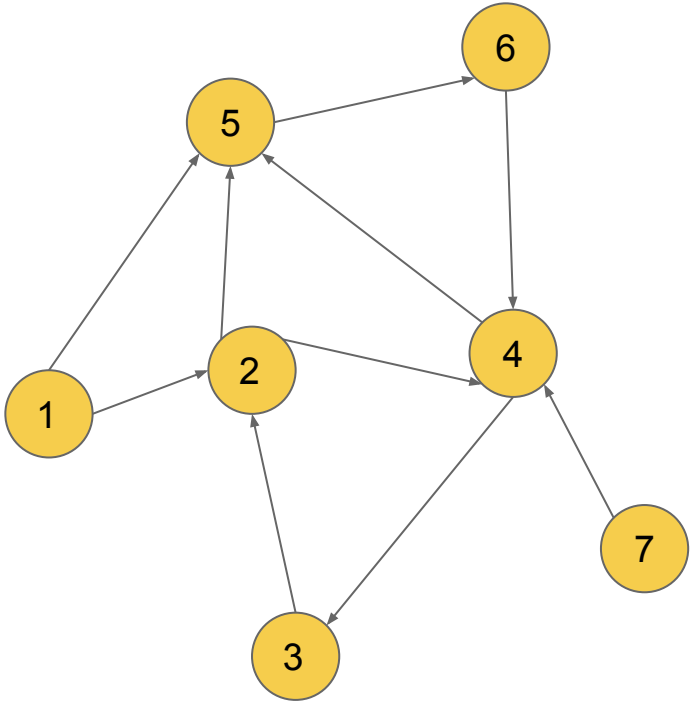
Generic Join Algorithm

Worst-case optimal join algorithm

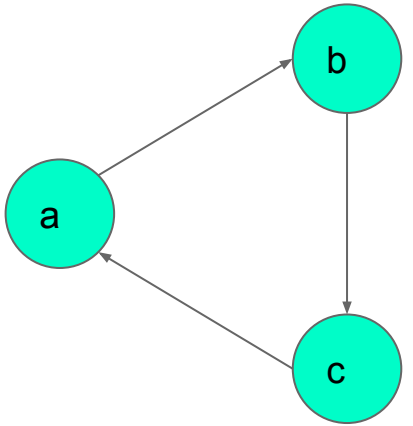
$O(n^{3/2})$ space complexity

MATCH Queries

Graph:

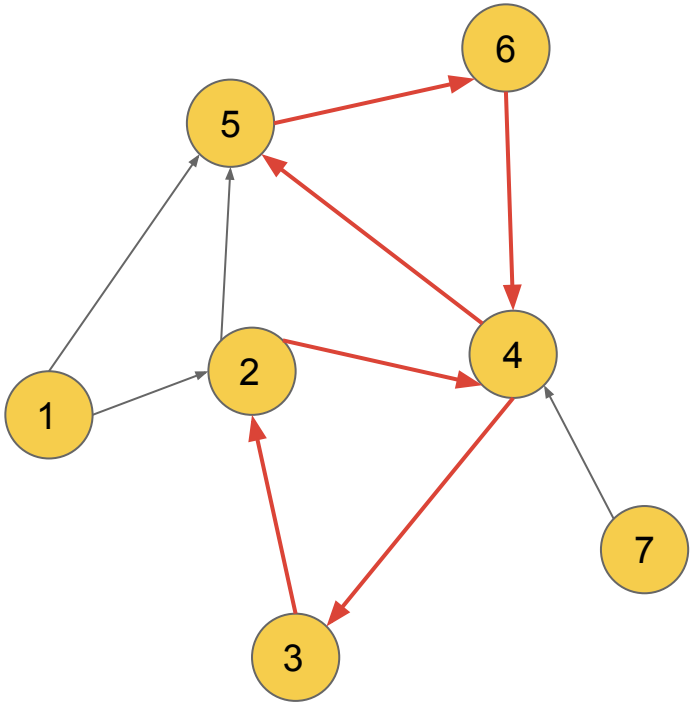


Query:

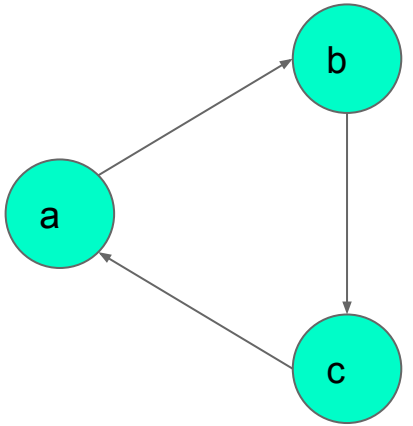


MATCH QUERIES

Graph:



Query:



Goal: find all motifs that match abc

{5,6,4} and {2,4,3}

Naive Join

R1
a->b

R2
b->c

{1,2}

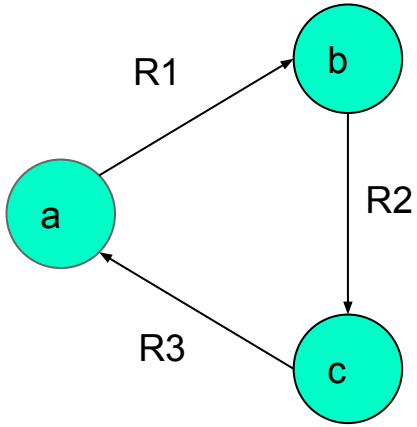
{1,2}
{1,5}
{2,5}
{2,4}
{3,2}
{4,3}
{4,5}
{5,6}
{6,4}
{7,4}

{1,5}

{1,2}
{1,5}
{2,5}
{2,4}
{3,2}
{4,3}
{4,5}
{5,6}
{6,4}
{7,4}

.
.
.
.

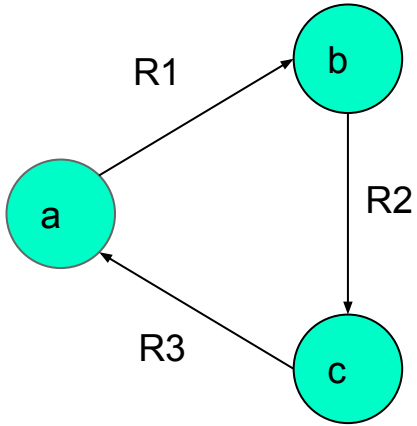
{2,5}
{2,4}
{3,2}
{4,3}
{4,5}
{5,6}
{6,4}
{7,4}



Space complexity $O(n^2)$, Time complexity $O(n^2)$

Generic Join Algorithm

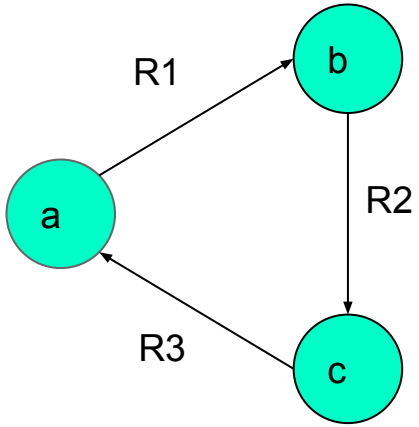
R1 a->b	R2 b->c	R3 c->a
{1,2}	{1,2}	{1,2}
{1,5}	{1,5}	{1,5}
{2,5}	{2,5}	{2,5}
{2,4}	{2,4}	{2,4}
{3,2}	{3,2}	{3,2}
{4,3}	{4,3}	{4,3}
{4,5}	{4,5}	{4,5}
{5,6}	{5,6}	{5,6}
{6,4}	{6,4}	{6,4}
{7,4}	{7,4}	{7,4}



Generic Join Algorithm

R1 a->b	R2 b->c	R3 c->a
{1,2}	{1,2}	{1,2}
{1,5}	{1,5}	{1,5}
{2,5}	{2,5}	{2,5}
{2,4}	{2,4}	{2,4}
{3,2}	{3,2}	{3,2}
{4,3}	{4,3}	{4,3}
{4,5}	{4,5}	{4,5}
{5,6}	{5,6}	{5,6}
{6,4}	{6,4}	{6,4}
{7,4}	{7,4}	{7,4}

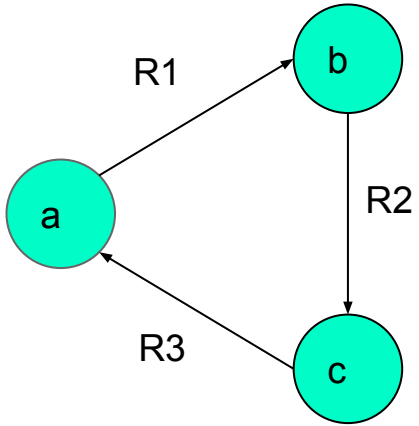
{2,_}
{3,_}
{4,_}
{5,_}
{6,_}



Generic Join Algorithm

R1 a->b	R2 b->c	R3 c->a
	{1,2}	{1,2}
{2,5}	{1,5}	{1,5}
{4,5}	{2,5}	{2,5}
{2,4}	{2,4}	{2,4}
{6,4}	{3,2}	{3,2}
{3,2}	{4,3}	{4,3}
{4,3}	{4,5}	{4,5}
{5,6}	{5,6}	{5,6}
	{6,4}	{6,4}
	{7,4}	{7,4}

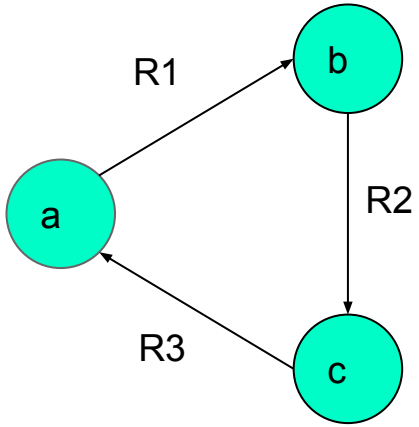
{2,5}
{4,5}
{2,4}
{6,4}
{3,2}
{4,3}
{5,6}



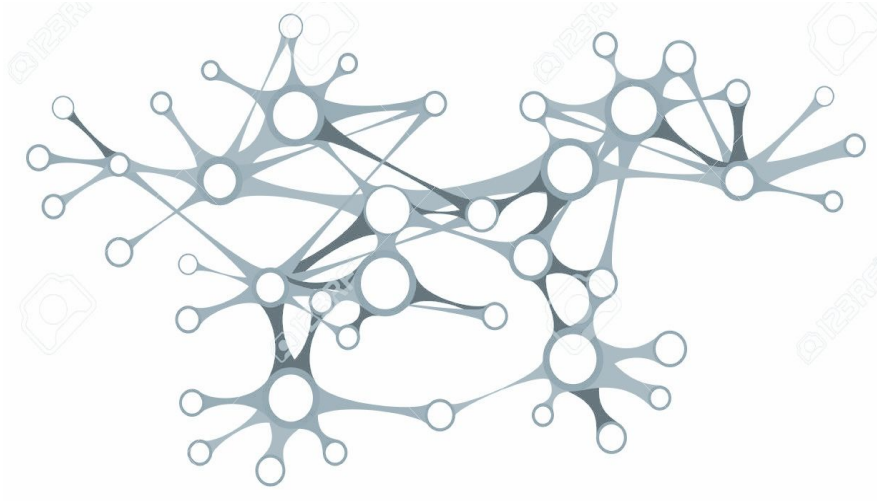
Generic Join Algorithm

R1 a->b	R2 b->c	R3 c->a
	{2,5}	{1,2}
{2,5}	{2,4}	{1,5}
{4,5}	{6,4}	{2,5}
{2,4}	{3,2}	{2,4}
{6,4}	{4,5}	{3,2}
{3,2}	{4,3}	{4,3}
{4,3}	{5,6}	{4,5}
{5,6}		{5,6}
		{6,4}
		{7,4}

{3,2,4}
{5,6,4}
{4,3,2}
{6,4,5}
{2,4,3}
{4,5,6}



GraphFlow



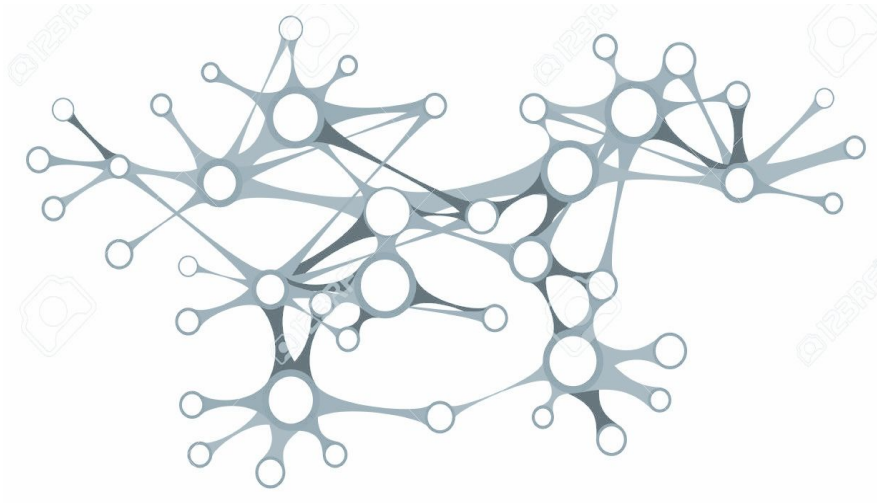
http://previews.123rf.com/images/andriocolt/andriocolt1505/andriocolt150500022/40320562-Abstract-network-of-interconnected-nodes-Stock-Vector.jpg

GraphFlow

A new graph database implemented from scratch.

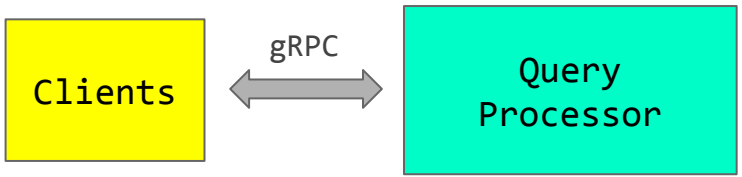
One-time MATCH queries
    Uses Generic Join algorithm

Continuous MATCH queries
    Uses Delta Generic Join algorithm

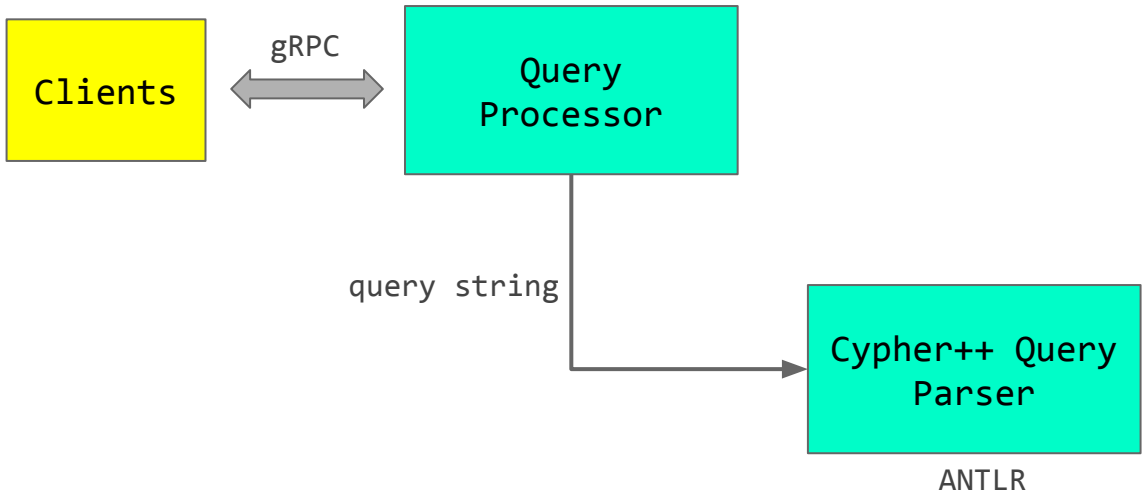


http://previews.123rf.com/images/andriocolt/andriocolt1505/andriocolt150500022/40320562-Abstract-network-of-interconnected-nodes-Stock-Vector.jpg

GraphFlow ARCHITECTURE



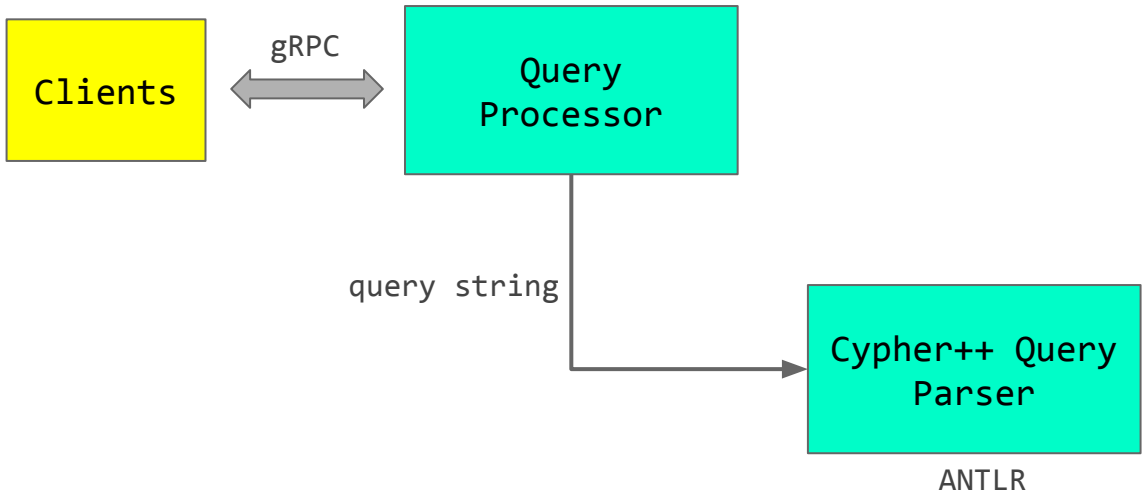
GraphFlow ARCHITECTURE



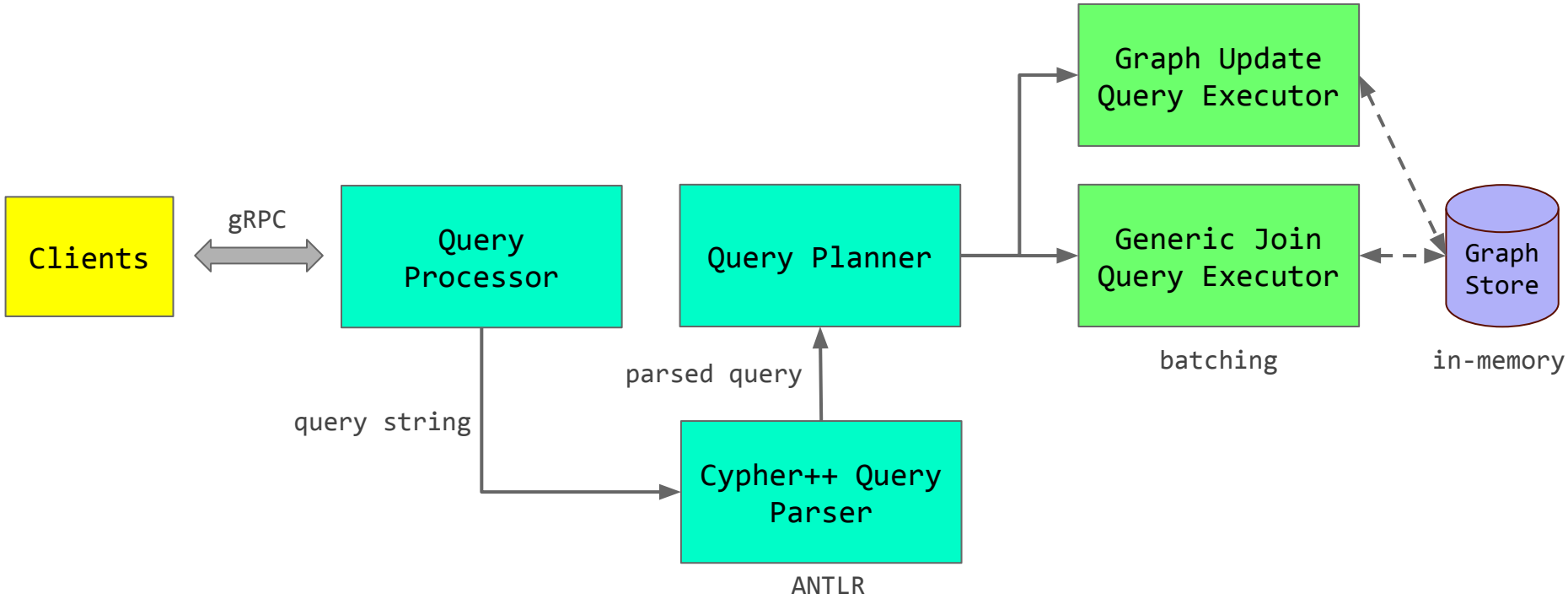
Cypher++

```
CREATE (1)->(2), (2)->(3), (2)->(4), (4)->(8);

DELETE (2)->(3);

MATCH (a)->(b),(a)->(c),(b)->(d),(c)->(d);

CONTINUOUS MATCH (a)->(b),(b)->(c),(c)->(a) FILE triangle_query;
```

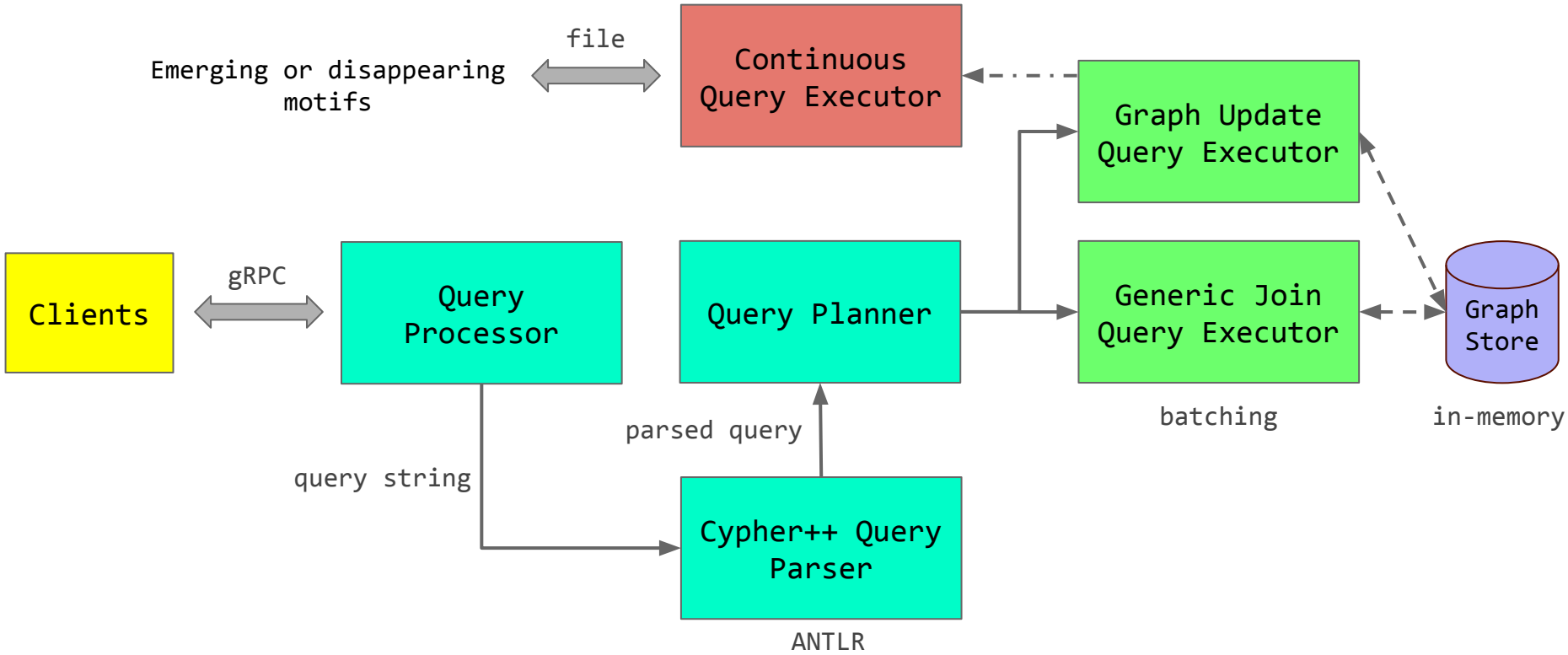
GraphFlow ARCHITECTURE



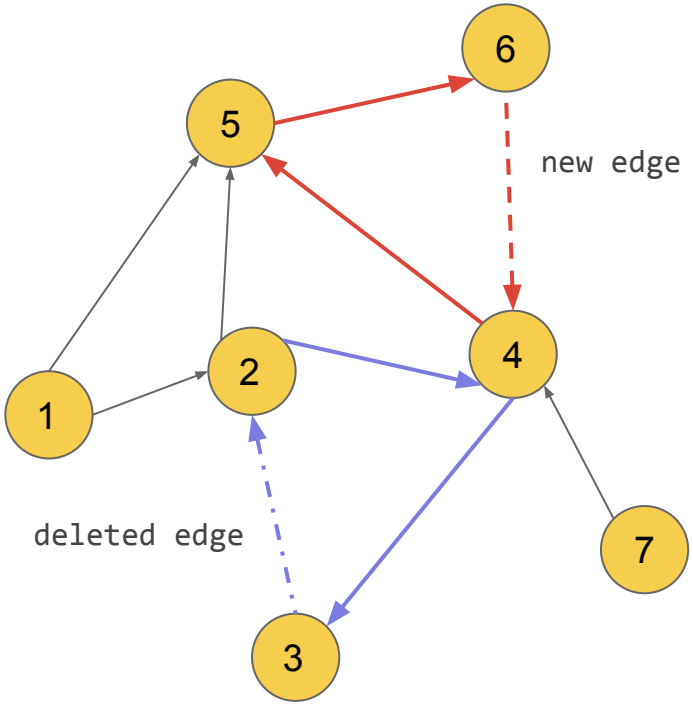
GraphFlow ARCHITECTURE

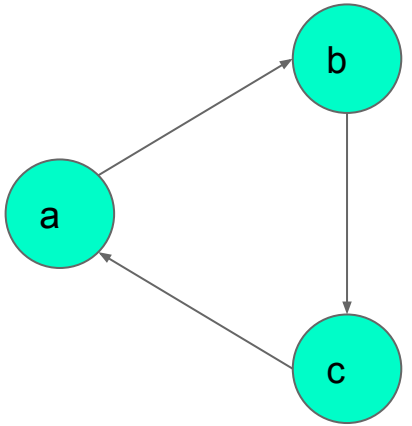


GraphFlow ARCHITECTURE



DEMO





Future Work

- Engineering
  - Add support for vertex and edge labels.
  - Plan viewer
  - UI to visualize continuous query results.
- Performance evaluation and comparisons with:
  - Neo4j
  - Incremental view maintenance.
- Transactions support
  - Abort transactions based on output motifs.
- Streaming support
  - CQL-like windowing semantics

Future Work

- Engineering
  - Add support for vertex and edge labels.
  - Plan viewer
  - UI to visualize continuous query results.
- Performance evaluation and comparisons with:
  - Neo4j
  - Incremental view maintenance.
- Transactions support
  - Abort transactions based on output motifs.
- Streaming support
  - CQL-like windowing semantics

Thank you.