elastic{on} 2017

What's evolving in Elasticsearch

Clinton Gormley & Simon Willnauer

Elastic

7 March 2017

@clintongormley & @s1m0nw

Elasticsearch 5.0.0

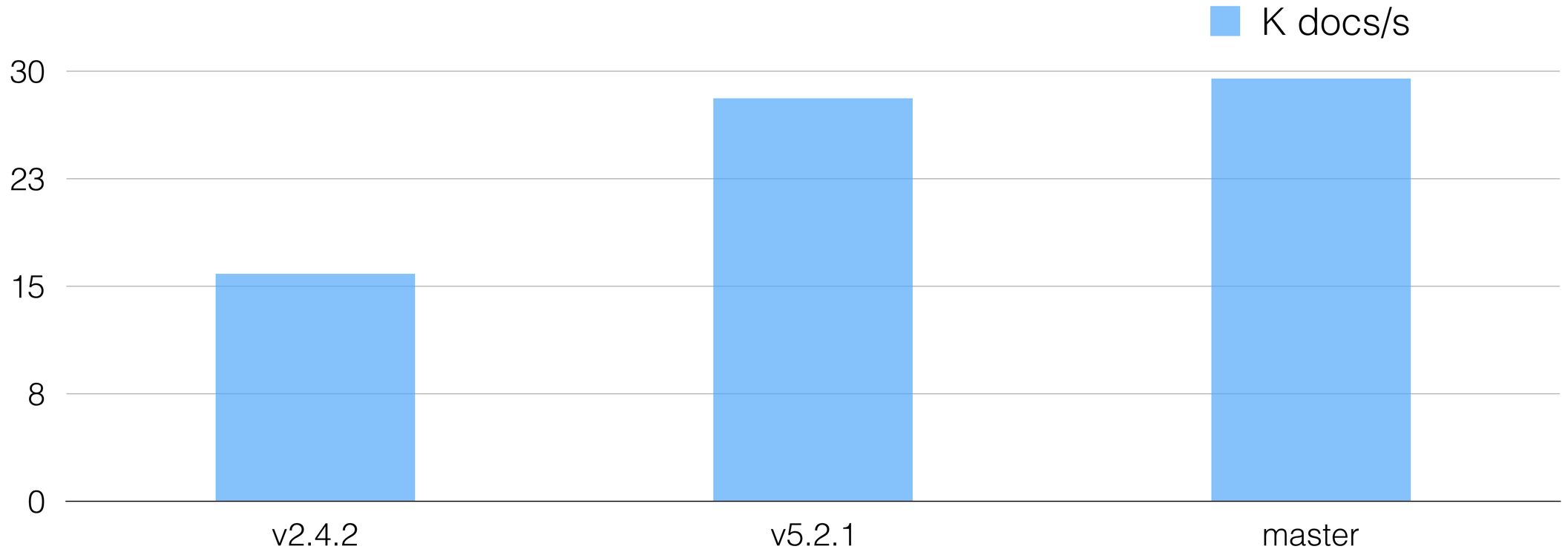
26 October 2016

- Faster
- Friendlier
- Smaller
- Smarter
- Safer

elasticsearch

Append-only indexing

Throughput with one replica on two nodes, with auto-generated IDs



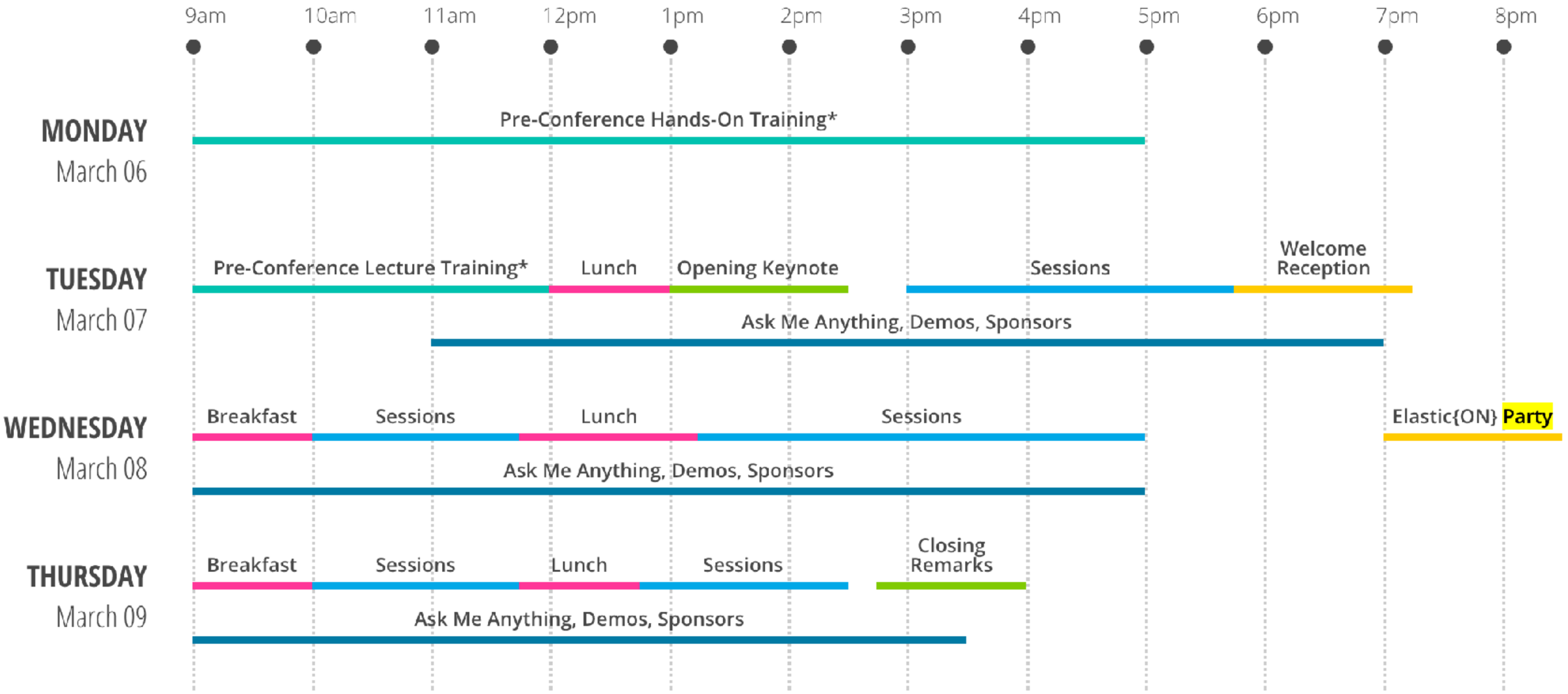
What’s new in 5.x?

Mappings

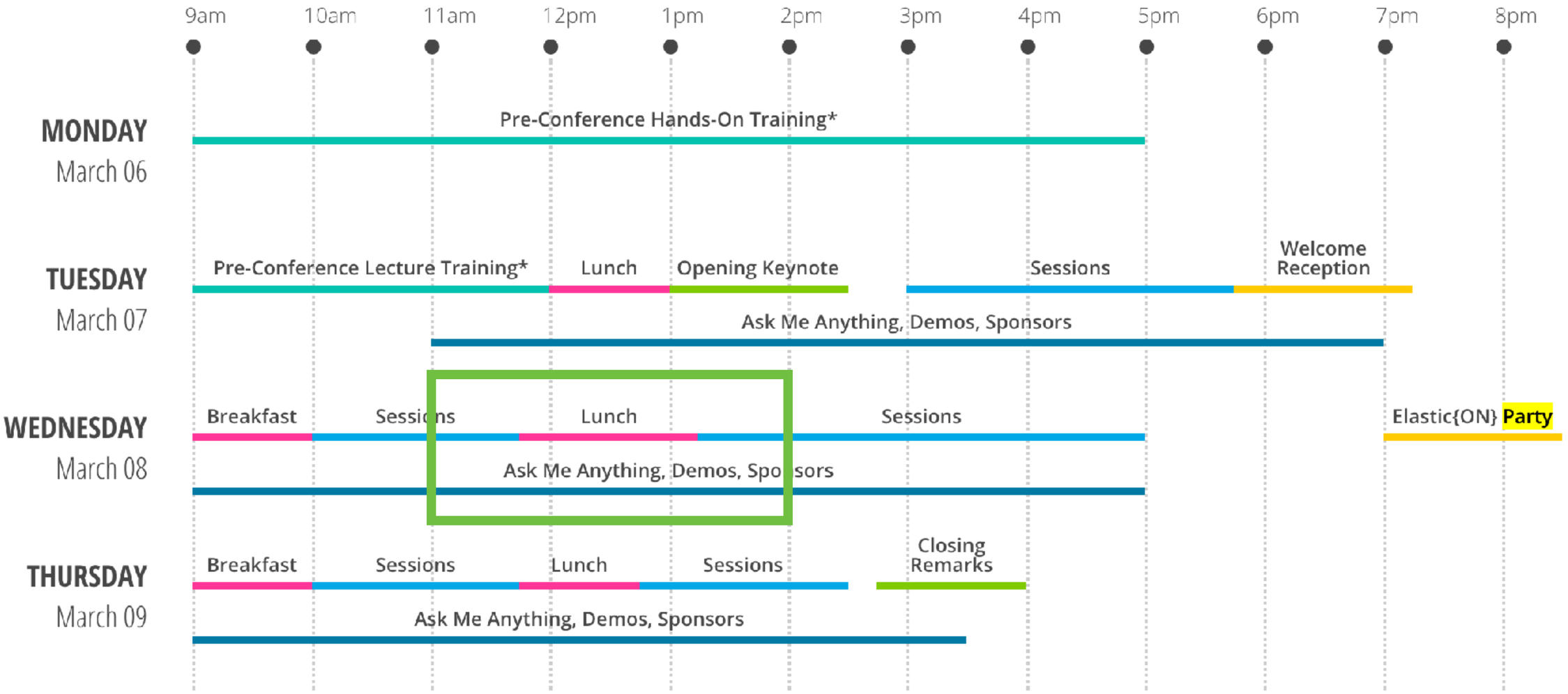
Range Fields & Queries

What's on at Elasticon

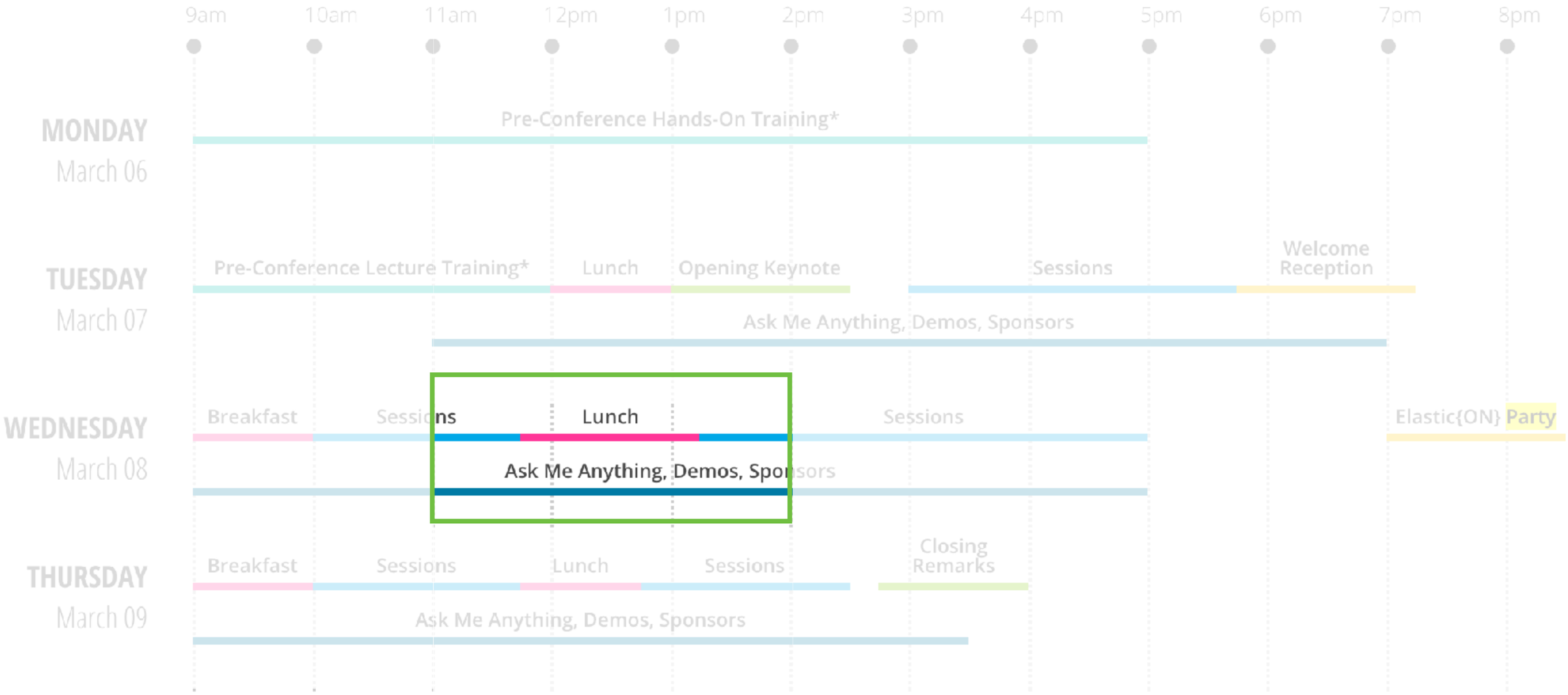
tomorrow between 11am and 2pm?


9am
10am
11am
12pm
1pm
2pm
3pm
4pm
5pm
6pm
7pm
8pm
MONDAY
March 06
Pre-Conference Hands-On Training*
TUESDAY
March 07
Pre-Conference Lecture Training*
Lunch
Opening Keynote
Sessions
Welcome Reception
Ask Me Anything, Demos, Sponsors
WEDNESDAY
March 08
Breakfast
Sessions
Lunch
Sessions
Elastic{ON} Party
Ask Me Anything, Demos, Sponsors
THURSDAY
March 09
Breakfast
Sessions
Lunch
Sessions
Closing Remarks
Ask Me Anything, Demos, Sponsors

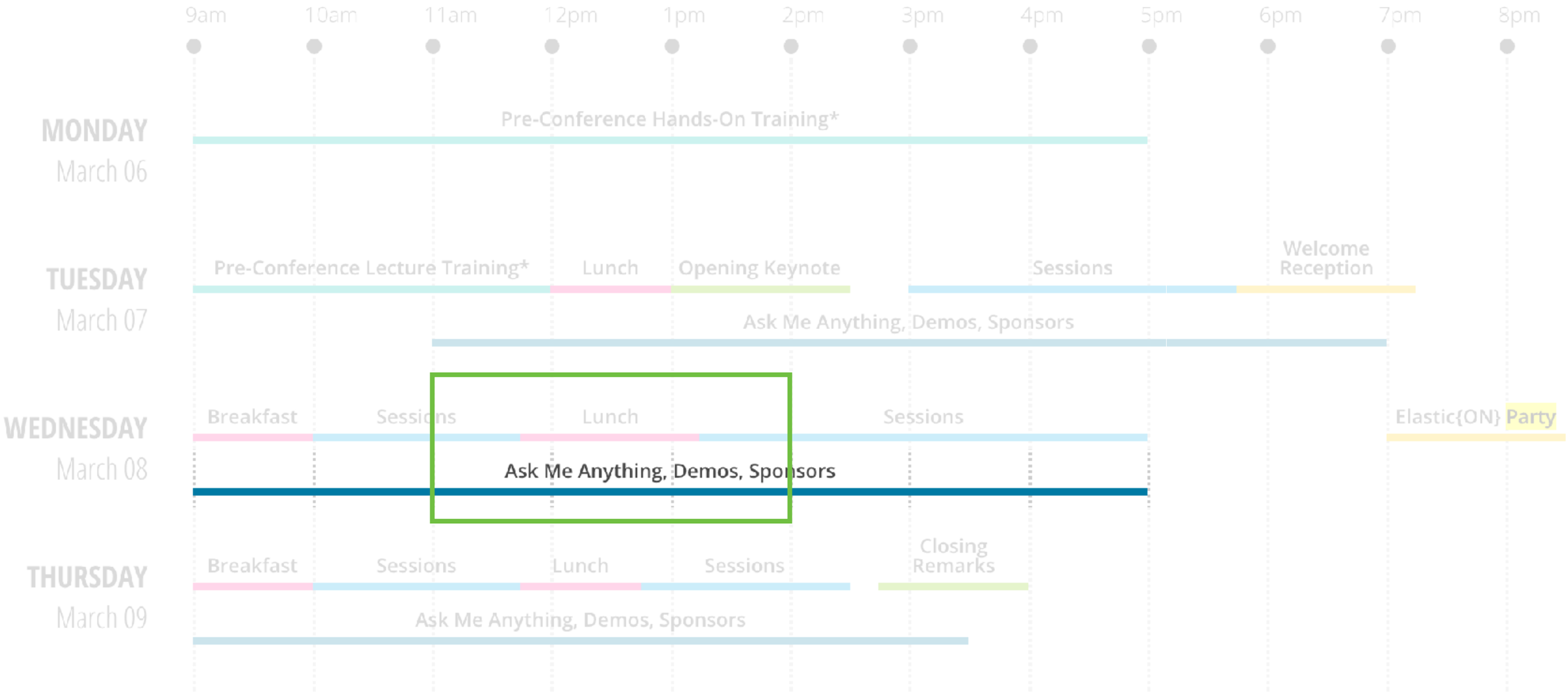
Wednesday 11am - 2pm



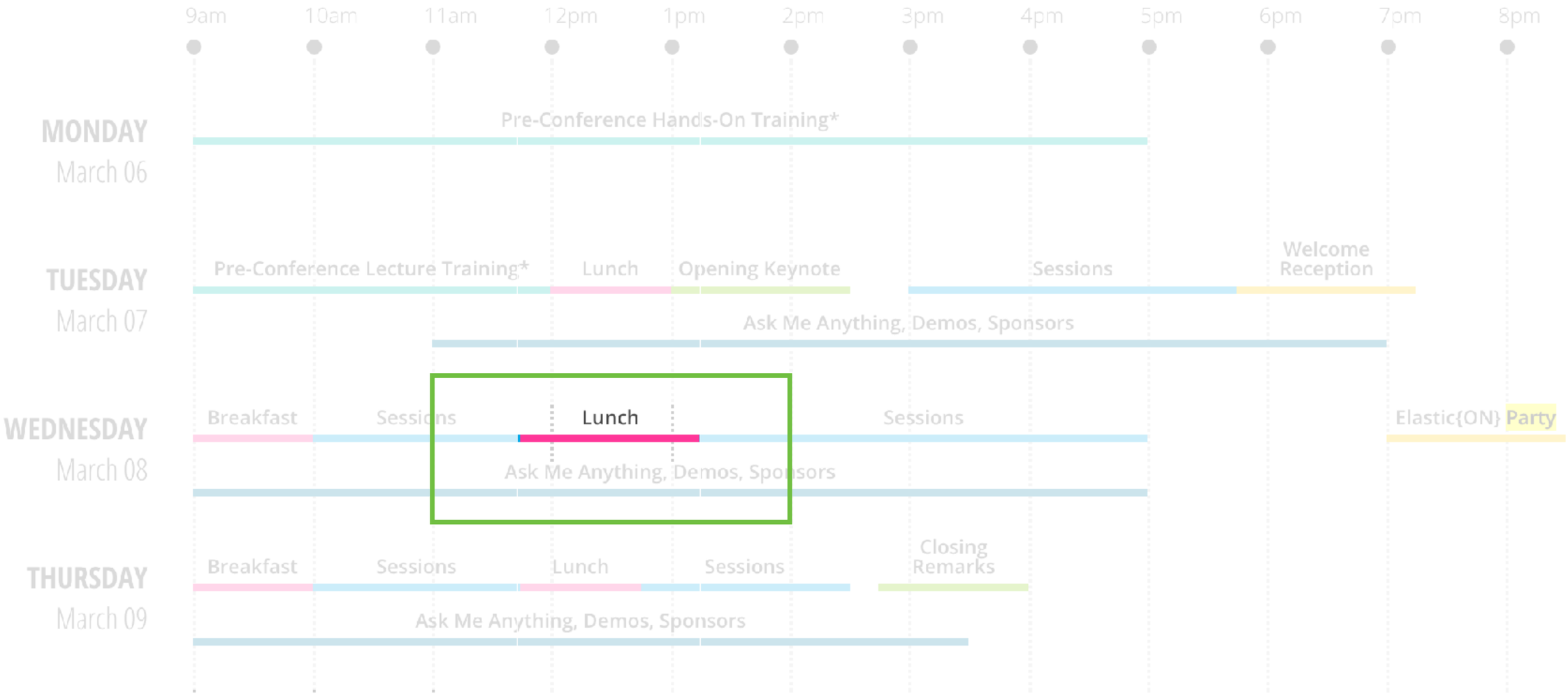
Wednesday 11am - 2pm - INTERSECTS



Wednesday 11am - 2pm - CONTAINS



Wednesday 11am - 2pm - WITHIN



Keyword Normalizers

```
{
  "city": {
    "type":  "string",
    "index": "analyzed",
    "fields": {
      "city.keyword": {
        "type":  "string",
        "index": "not_analyzed"
      }
    }
  }
}
```

```
{
  "city": {
    "type":  "text"
    "fields": {
      "city.keyword": {
        "type":  "keyword"
      }
    }
  }
}
```

```
{
  "city": {
    "type":  "text"
    "fields": {
      "city.keyword": {
        "type":  "keyword"
      }
    }
  }
}
```

Full text queries

Full text analysis

```
{
  "city": {
    "type":  "text"
    "fields": {
      "city.keyword": {
        "type":  "keyword"
      }
    }
  }
}
```

Keyword queries
Aggregations
Sorting

```
{
  "city": {
    "type":  "text"
    "fields": {
      "city.keyword": {
        "type":  "keyword"
      }
    }
  }
}
```

No analysis

```
San Francisco
SAN FRANCISCO
san francisco
San francisc0
```

San Francisco
SAN FRANCISCO
san francisco
San francisc0

san francisco

Search & Aggregations

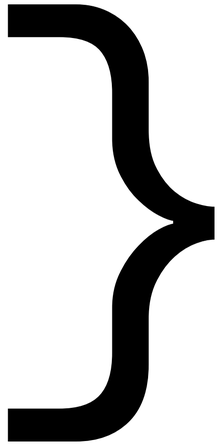
Multi-Word Synonyms

NY

NYC

New York

New York City



Synonyms

Phrase query:

“NYC is OLD!”

Synonym Filter:

```
(ny|nyc|new), (is|york), (old,city)
```

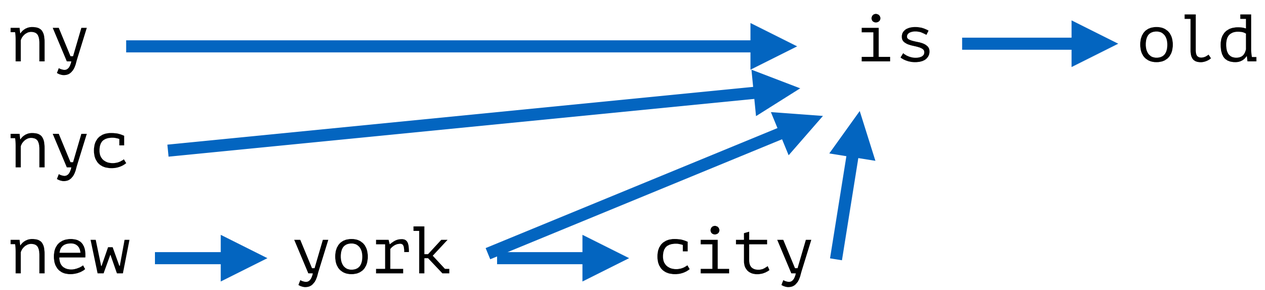
Synonym Filter:

```
(ny|nyc|new), (is|york), (old,city)
```

Synonym Filter:

```
(ny|nyc|new), (is|york), (old,city)
```

Synonym Graph Filter:



More Search Improvements

Query Optimizations

- Smarter query caching
- Faster geo, range, and nested queries
- Unified highlighter
- Field collapsing
- Cancellable searches
- Partitioned term aggs

Operational Improvements

When your cluster is **RED**…

`/_cat/allocation`

`/_cat/indices`

`/_node/stats`

`/_cat/recovery`

`/_cluster/settings`

`/_node`

`/_cat/nodes`

`/_cluster/health`

`/_cat/shards`

`/{index}/_shard_stores`

`/_cluster/state`

`/{index}/_settings`

When your cluster is **RED**…

```
/_cluster/allocation/explain
```

/_cluster/allocation/explain

```
…
"allocate_explanation" : "cannot allocate because allocation
is not permitted to any of the nodes”,
…
{
  "decider" : "filter",
  "decision" : "NO",
  "explanation" : "node does not match index setting
                   [index.routing.allocation.include]
                   filters [_name:\"non_existent_node\"]"
}
…
```

/_cluster/allocation/explain

```
…
"unassigned_info" : {
    "reason" : "NODE_LEFT",
    "at" : "2017-01-04T18:03:28.464Z",
    "details" : "node_left[OIWe8UhhThCK0V5XfmdrmQ]",
    "last_allocation_status" : "no_valid_shard_copy"
  },
  "can_allocate" : "no_valid_shard_copy",
  "allocate_explanation" : "cannot allocate because a
previous copy of the primary shard existed but can no longer
be found on the nodes in the cluster"
…
```

/_cluster/allocation/explain

```
…
"rebalance_explanation" : "cannot rebalance as no target node
exists that can both allocate this shard and improve the
cluster balance",
  "node_allocation_decisions" : [
    {
      "node_id" : "oE3EGFc8QN-Tdi5FFEprIA",
      "node_name" : "node_t1",
      "transport_address" : "127.0.0.1:9401",
      "node_decision" : "worse_balance",
      "weight_ranking" : 1
    }
…
```

Java REST Client

Java REST Client - behind the scenes

- Came late to the party…
- Isn’t nearly as extensive as the Transport Client
- Should have been fixed years ago but hindsight is 20/20
- Maintaining a transport protocol based client causes a massive engineering overhead
  - It’s a “second” entry point into the system
  - Complicates distinguishing between clients and nodes

Java low-level HTTP client

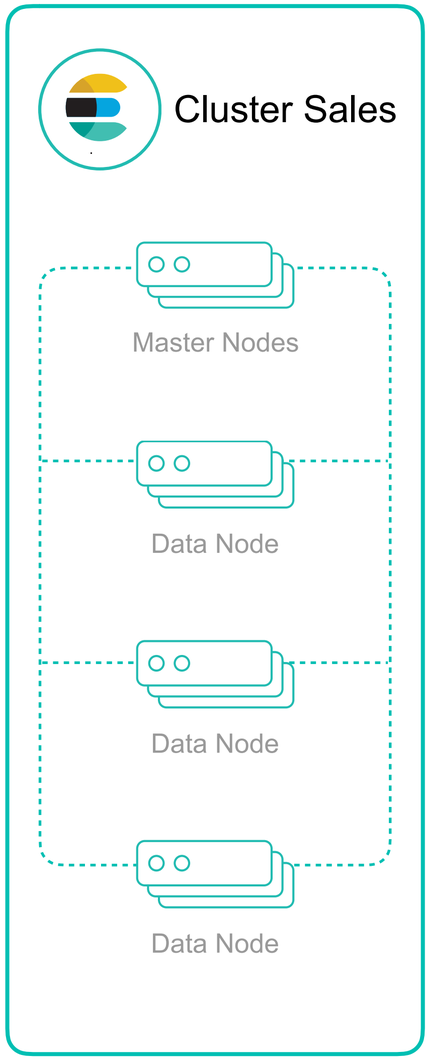
- Released in 5.0.0
- JSON strings only
- Resilient, but not user friendly due to the lack of a higher level API

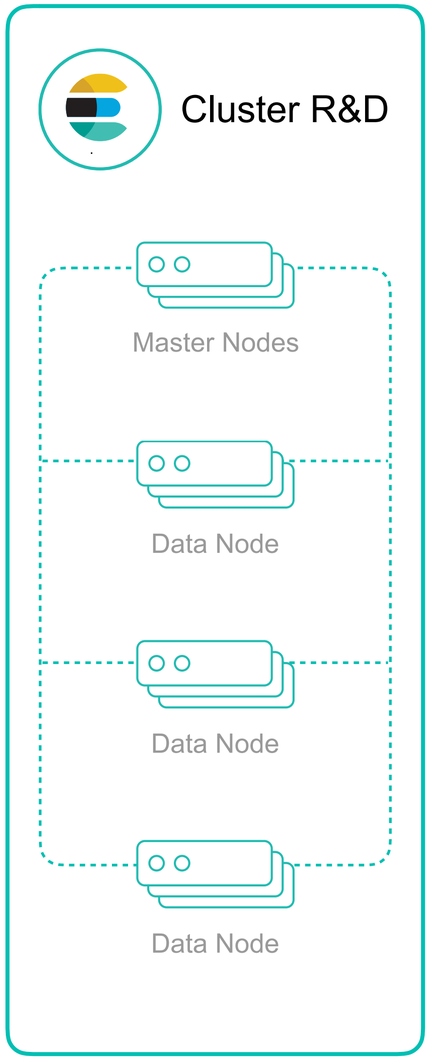
Java high-level HTTP client

- IDE friendly
- Similar API to Transport Client - easy migration
- Based on low-level REST client
- Support CRUD & Search
- Previews in 5.5
- Depends on elasticsearch-core

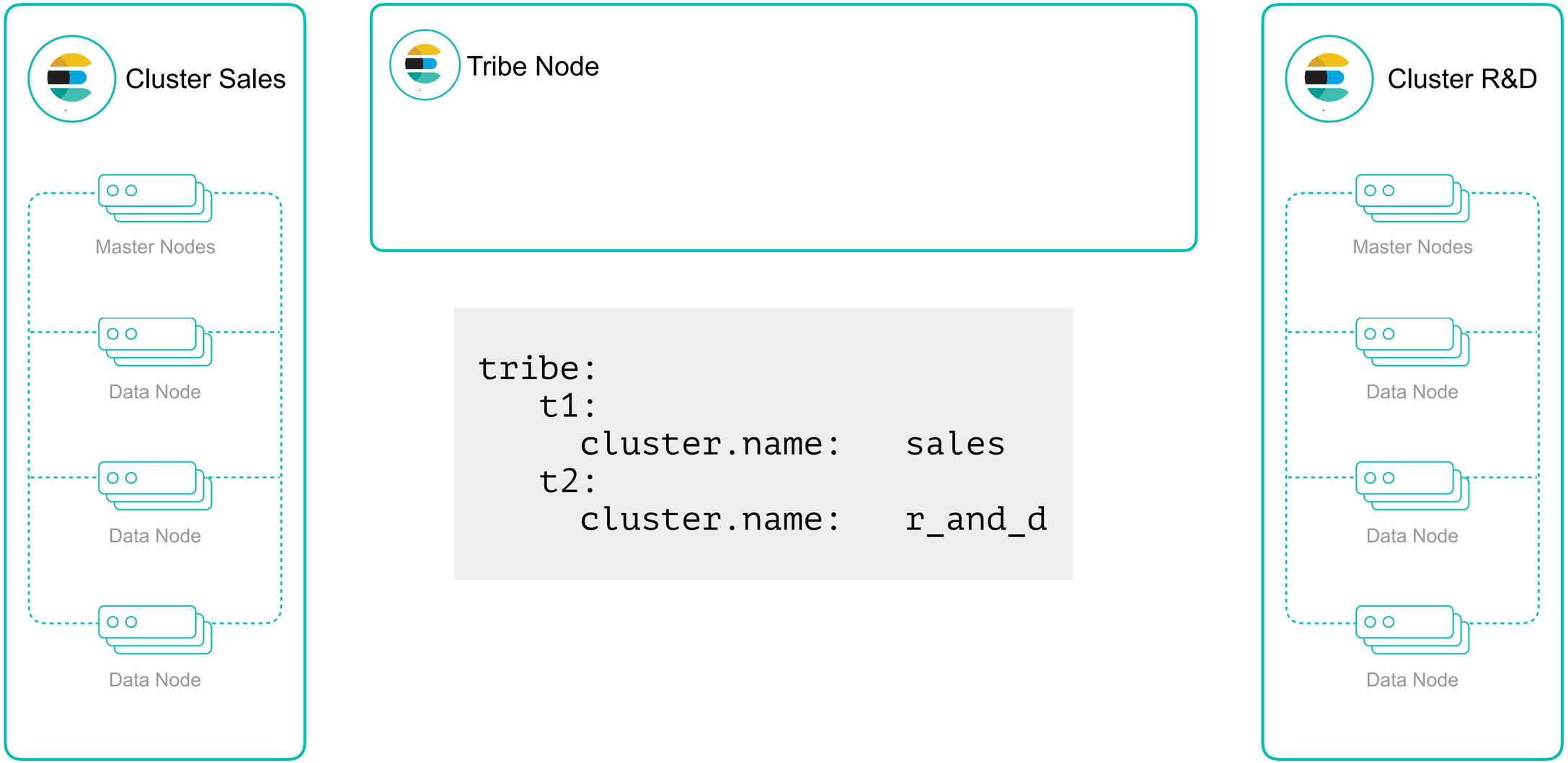
Tribe Node

How the Tribe Node works

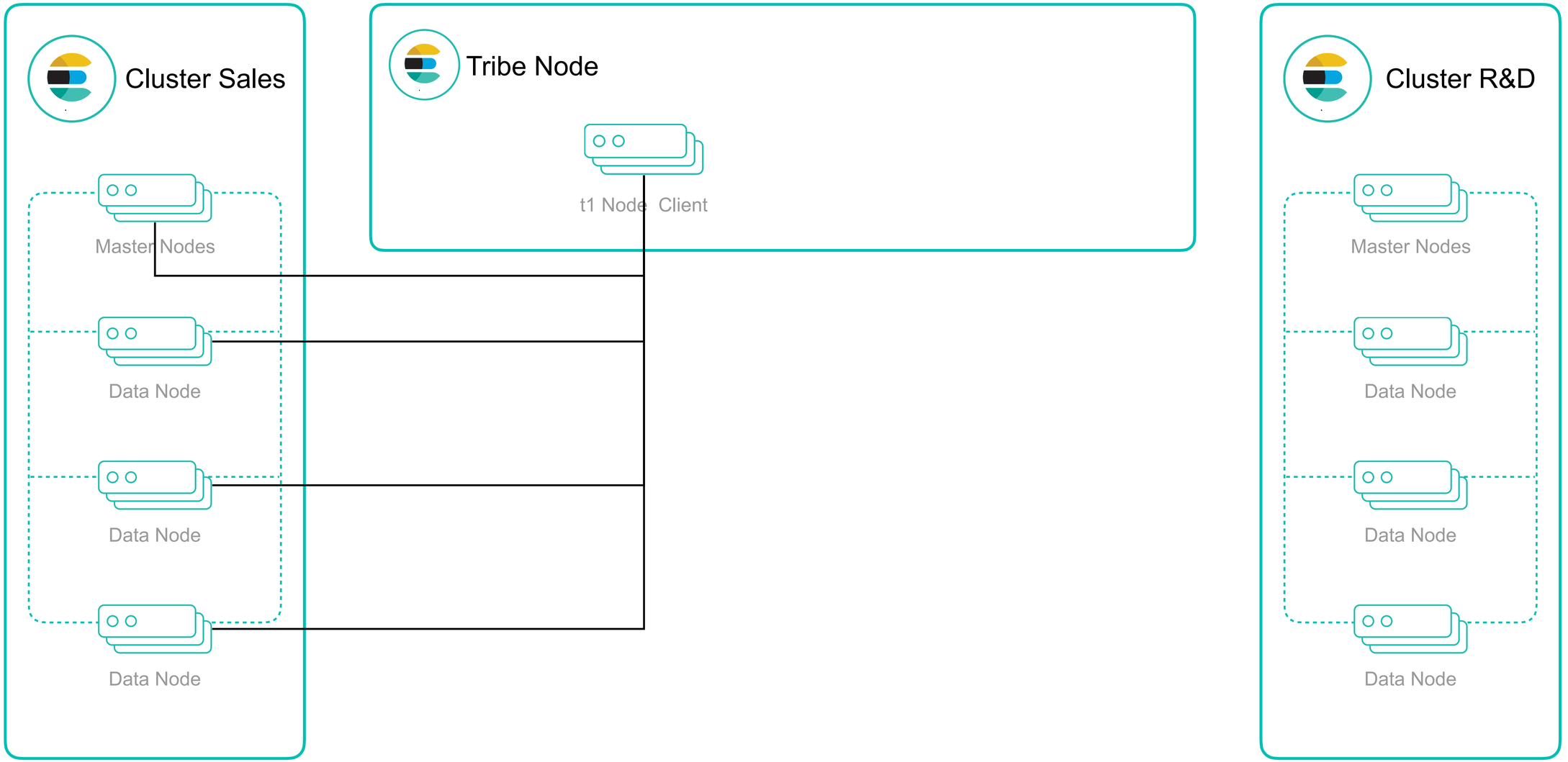




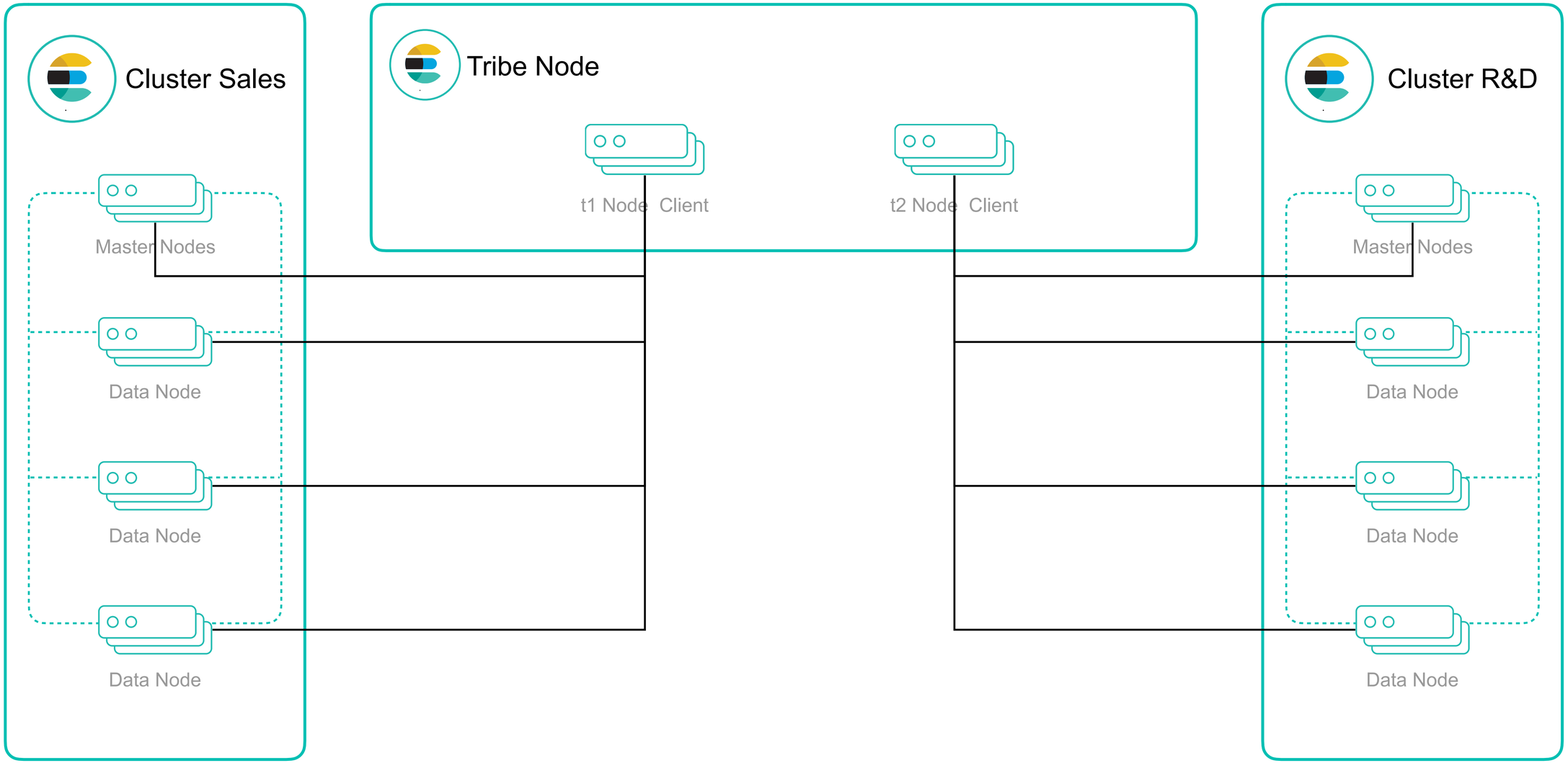
How the Tribe Node works



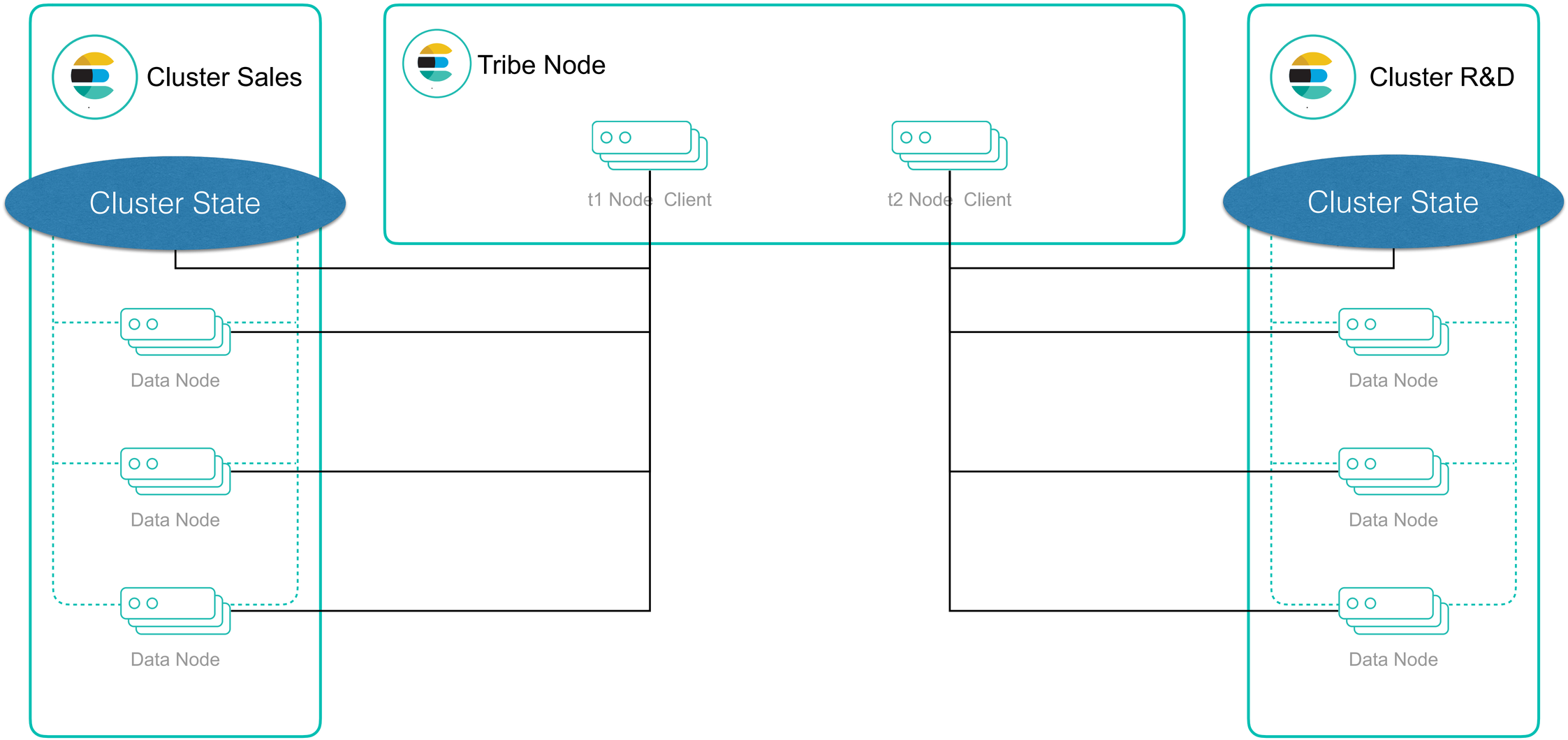
How the Tribe Node works



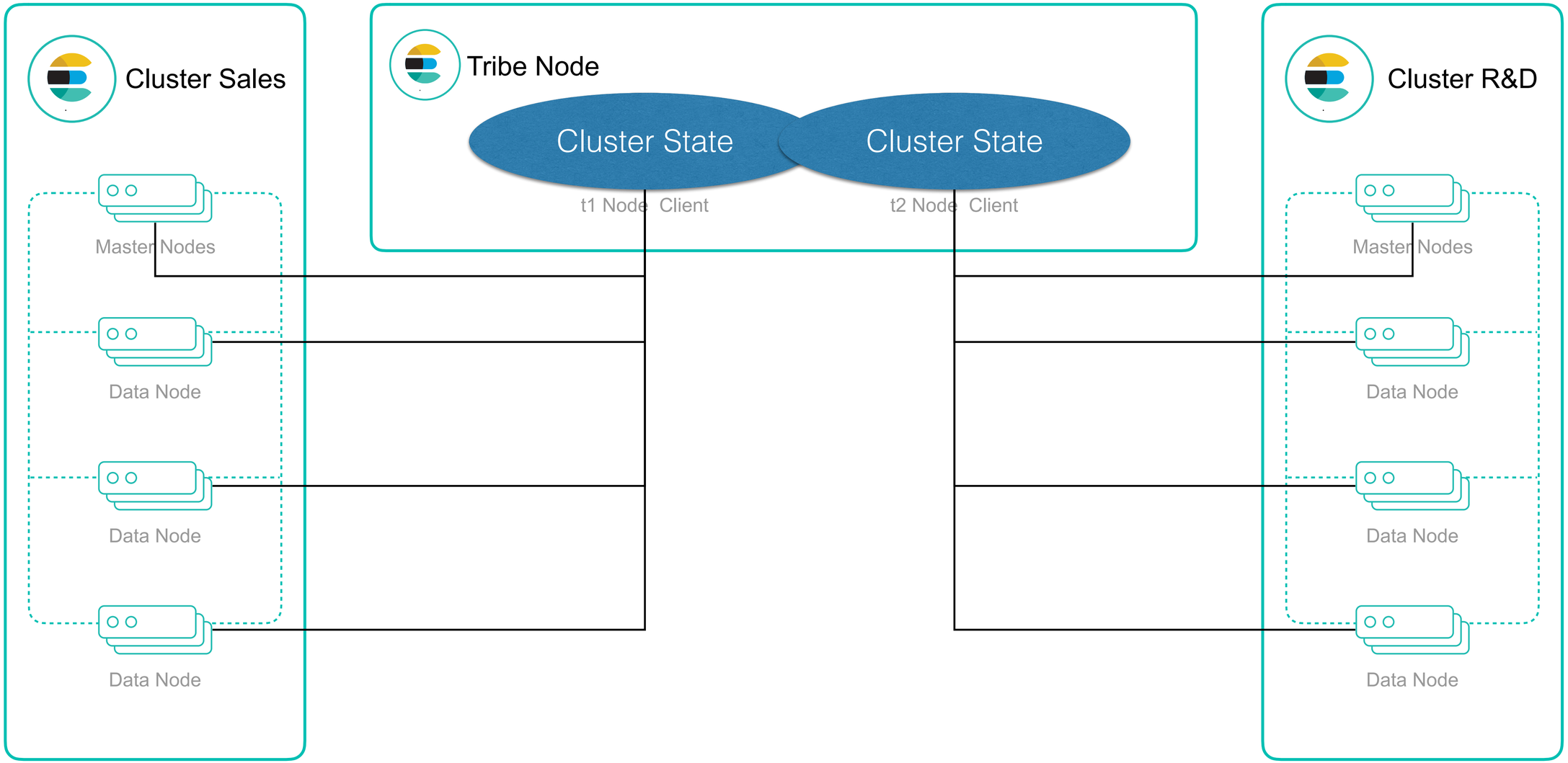
How the Tribe Node works



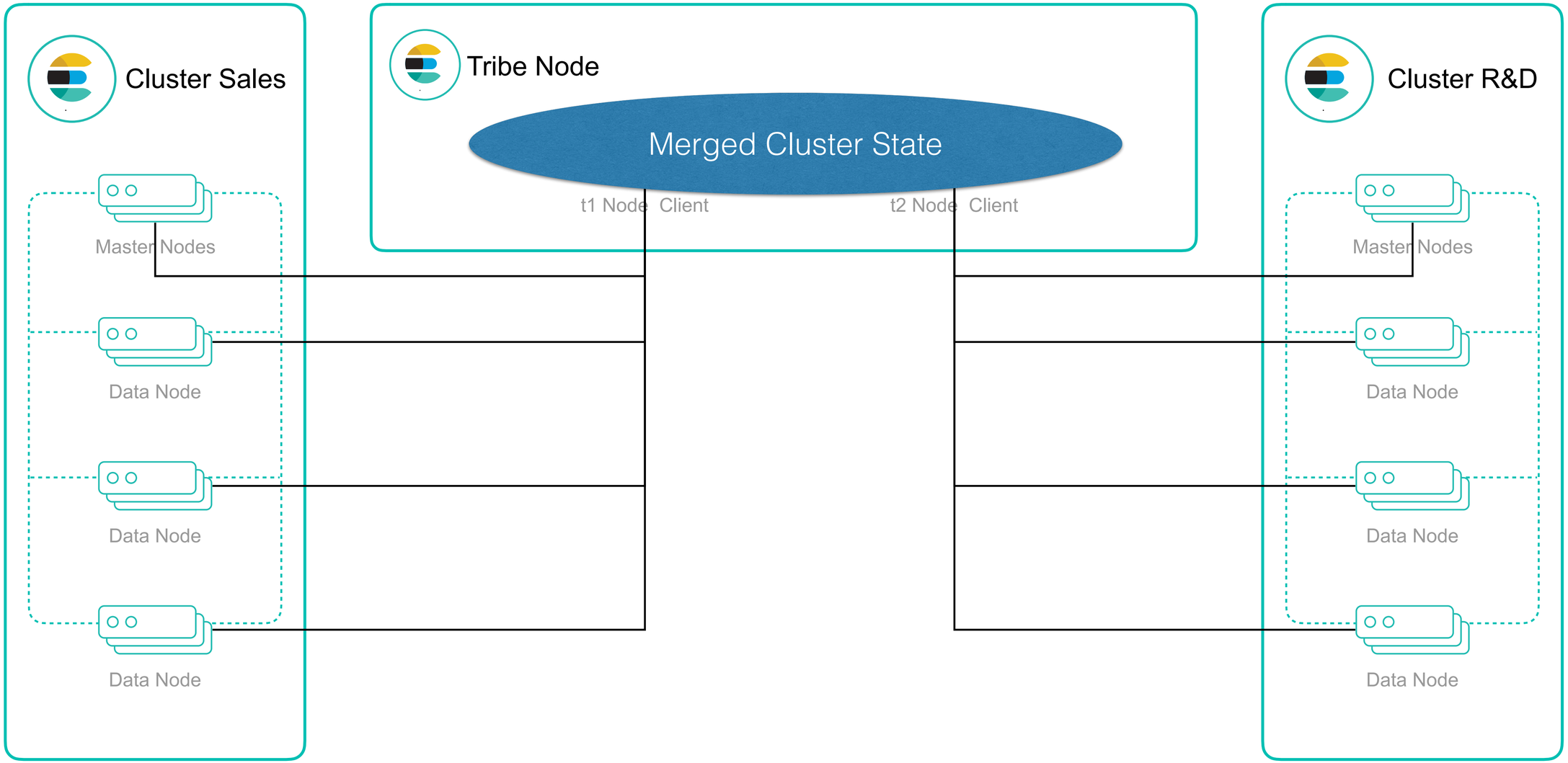
How the Tribe Node works



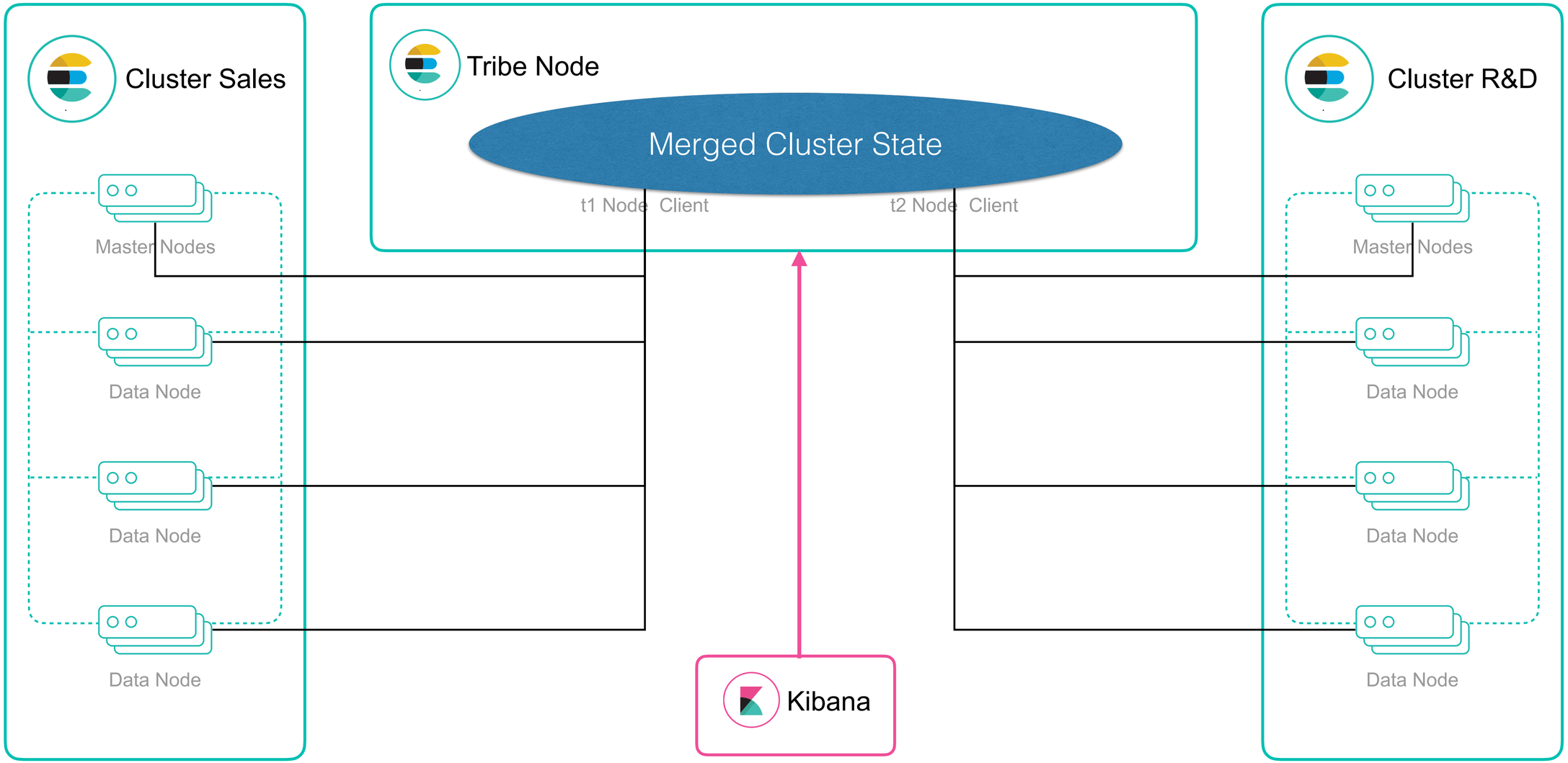
How the Tribe Node works



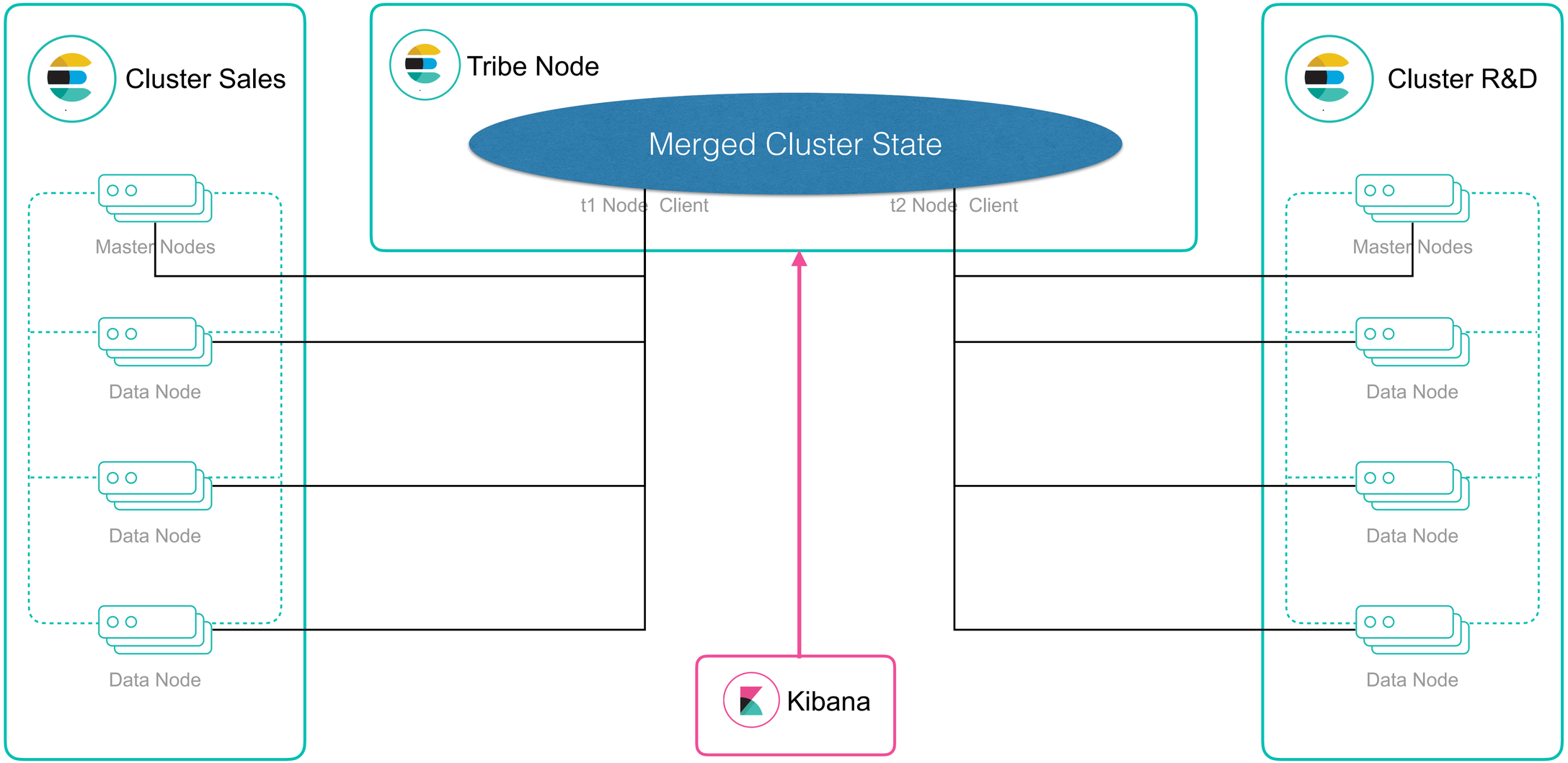
How the Tribe Node works



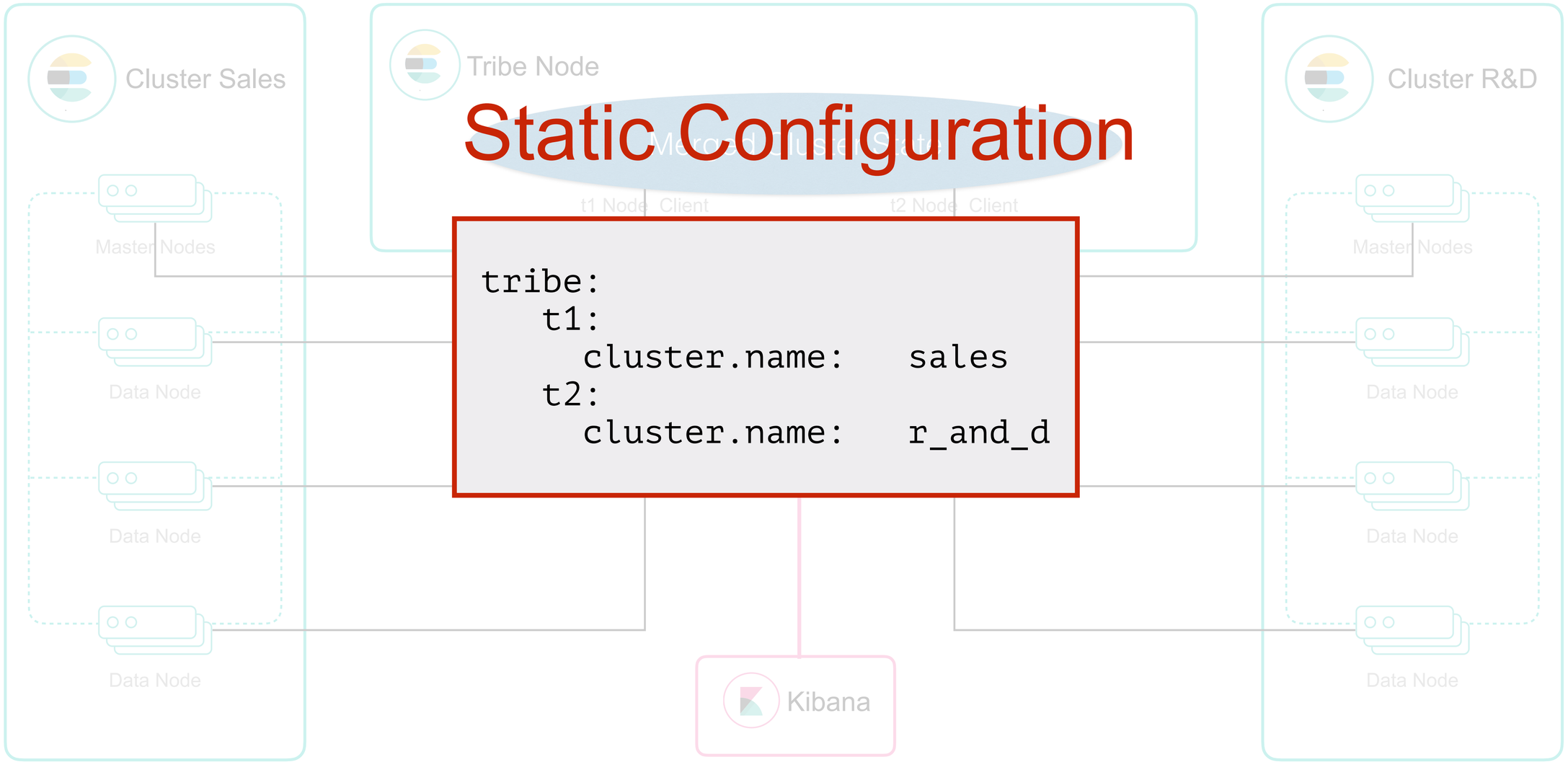
How the Tribe Node works



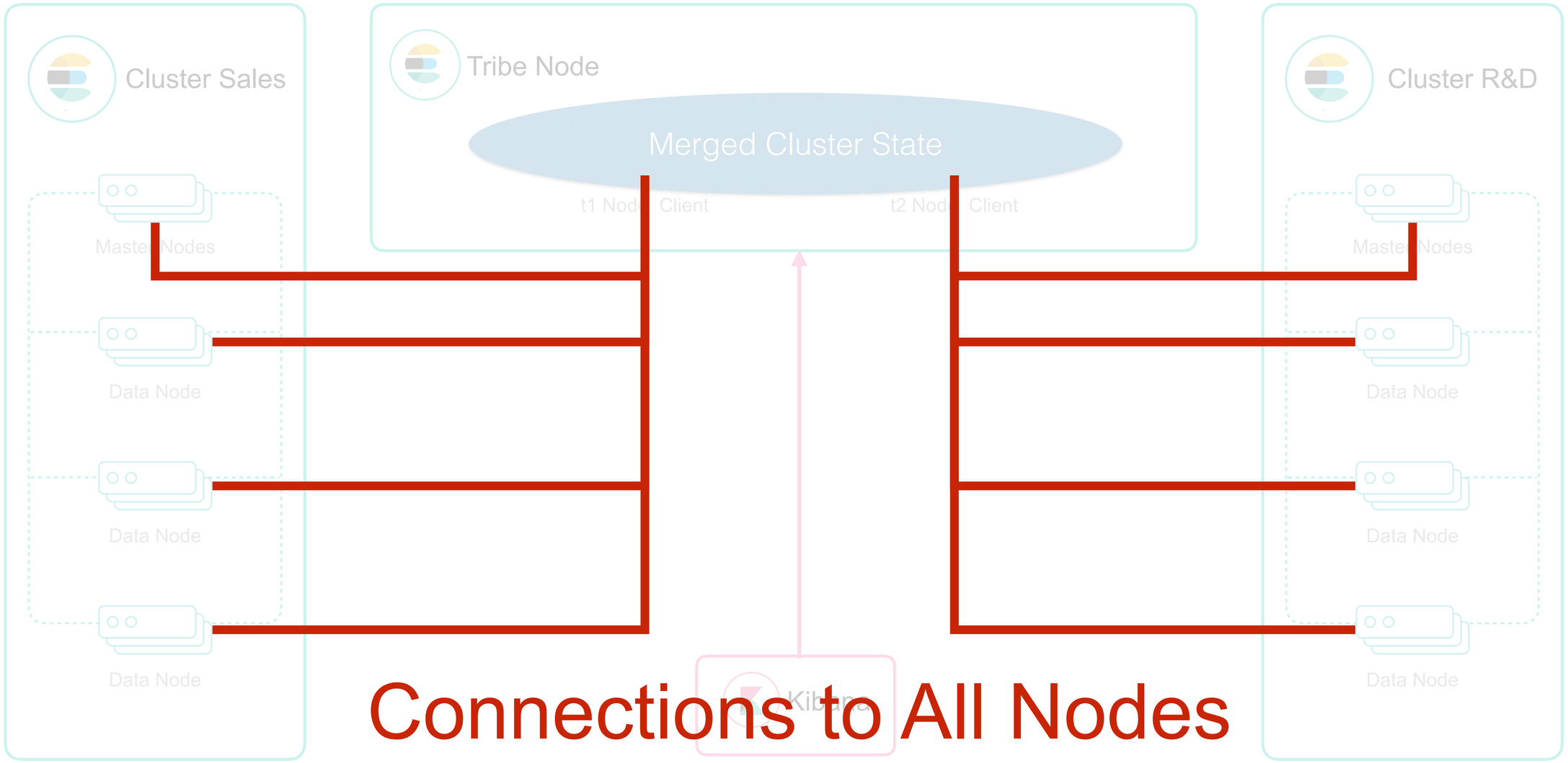
Problems With How the Tribe Node works



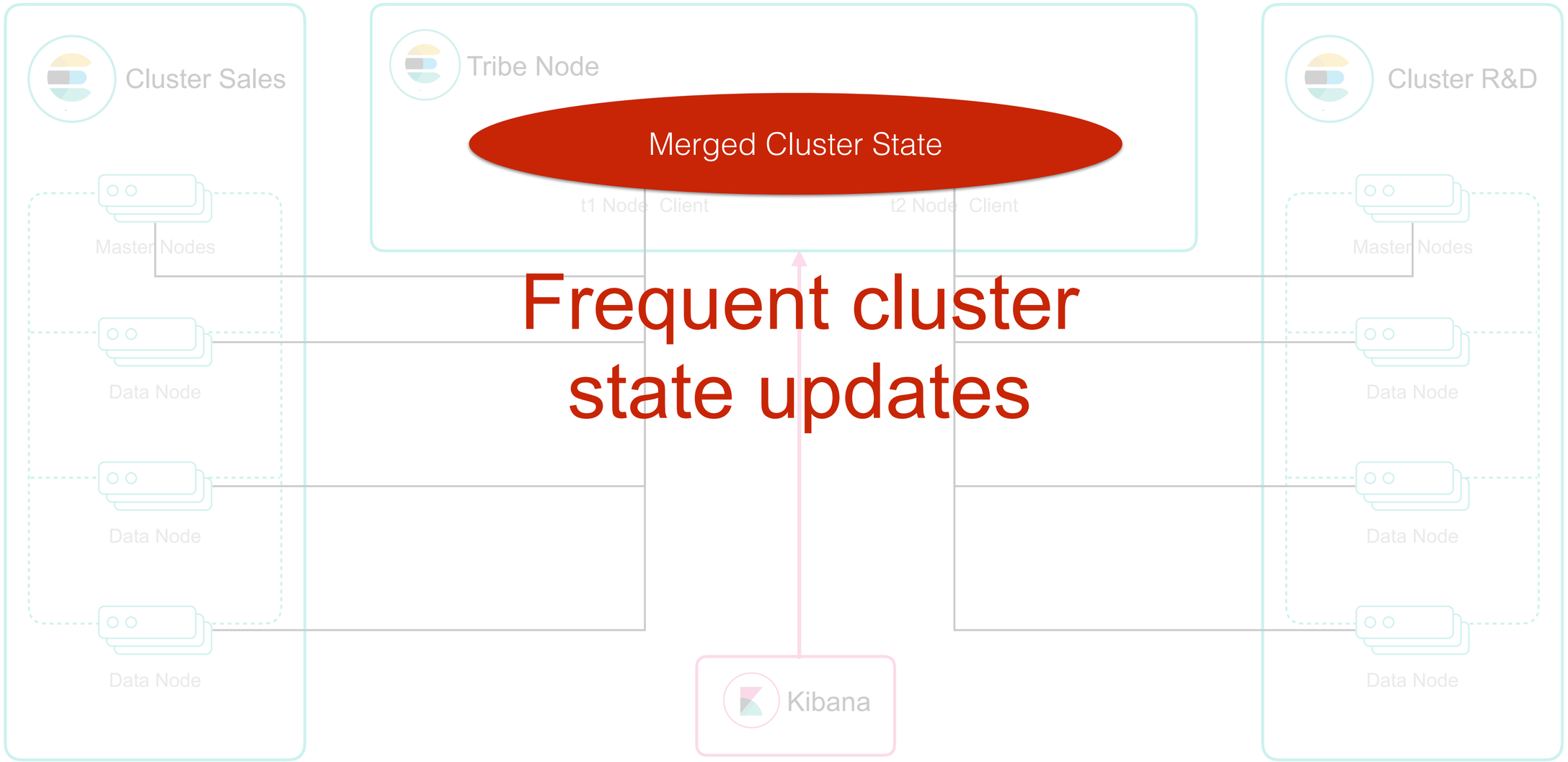
Problems With How the Tribe Node works



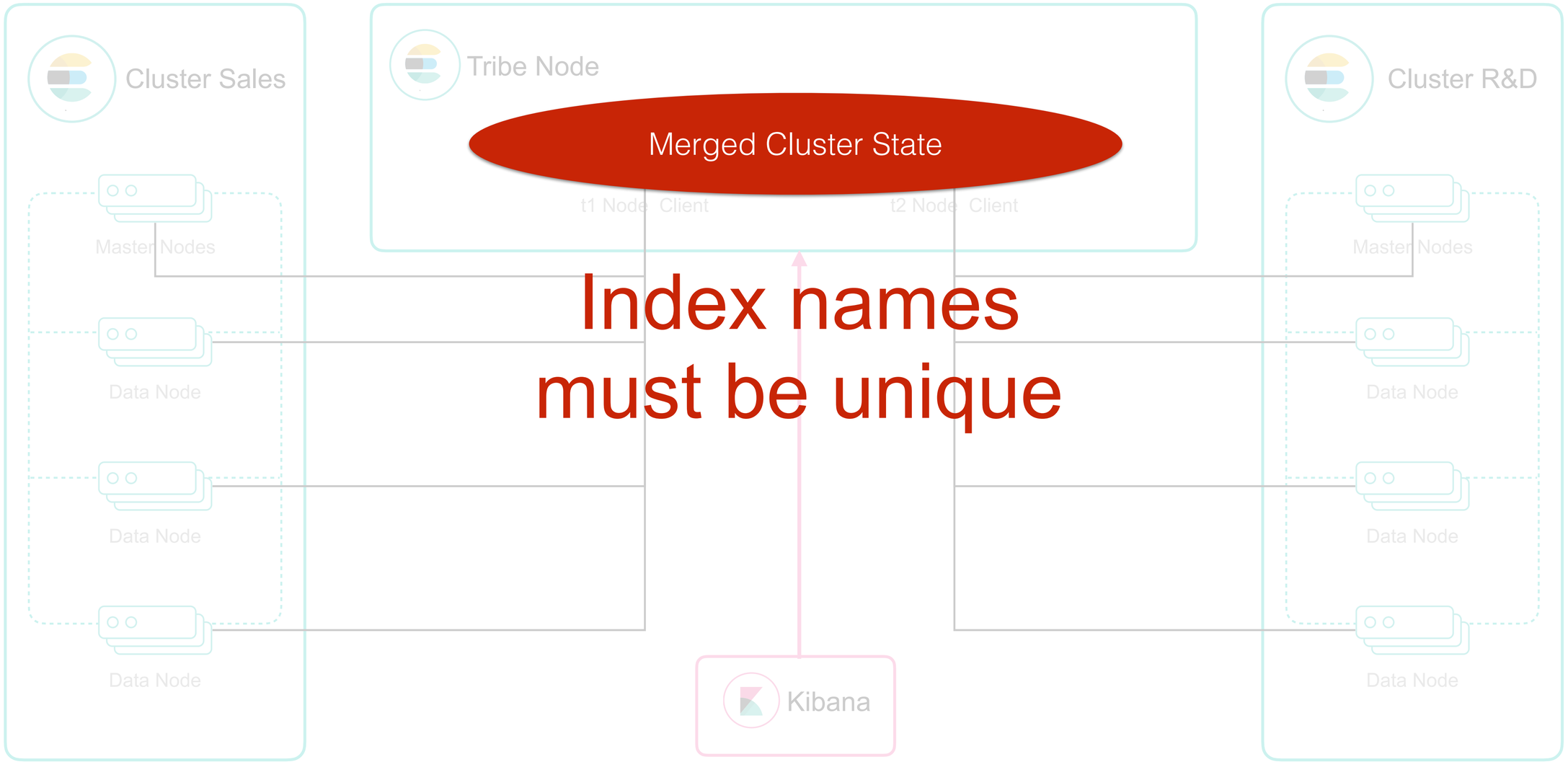
Problems With How the Tribe Node works



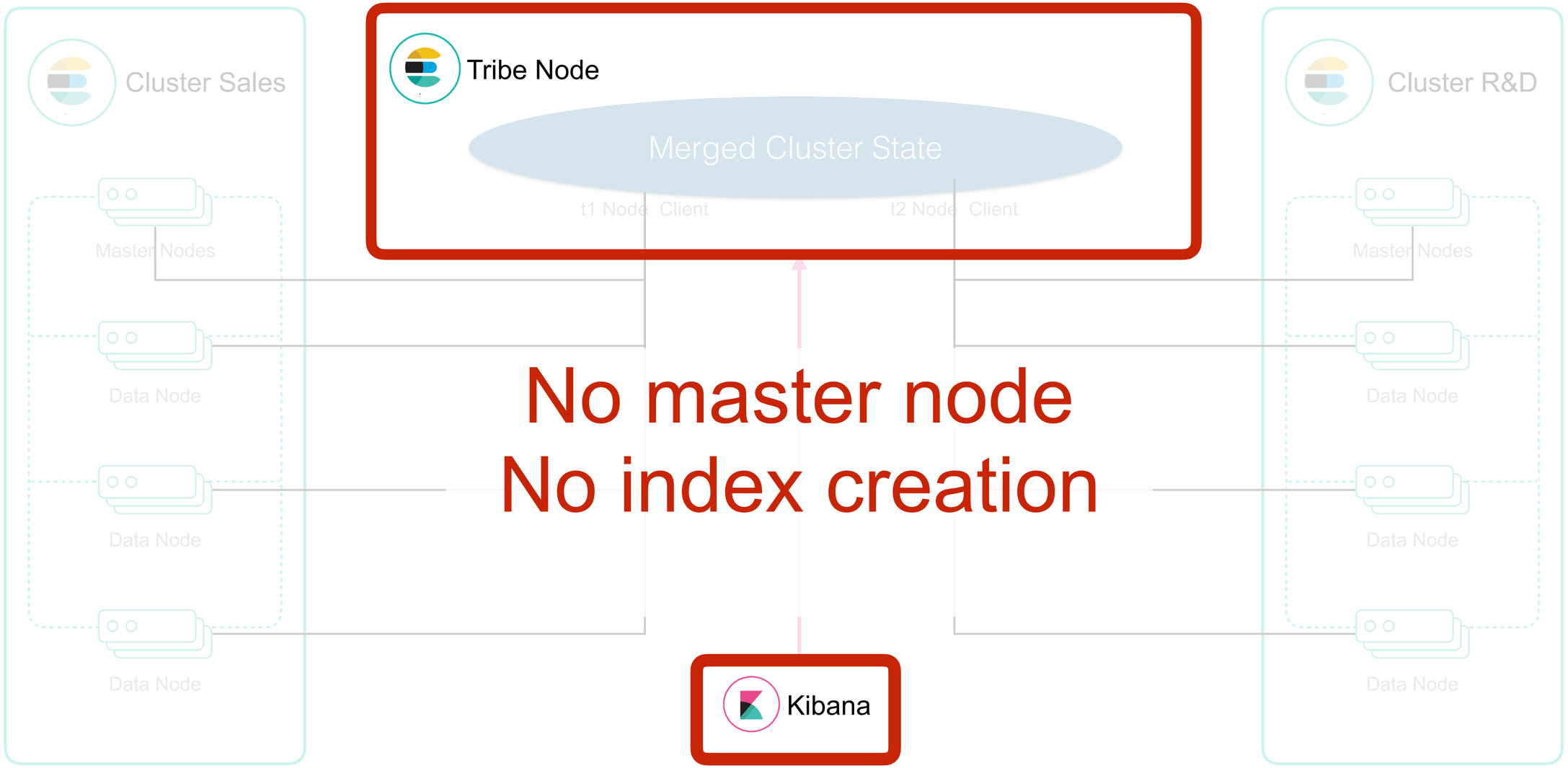
Problems With How the Tribe Node works



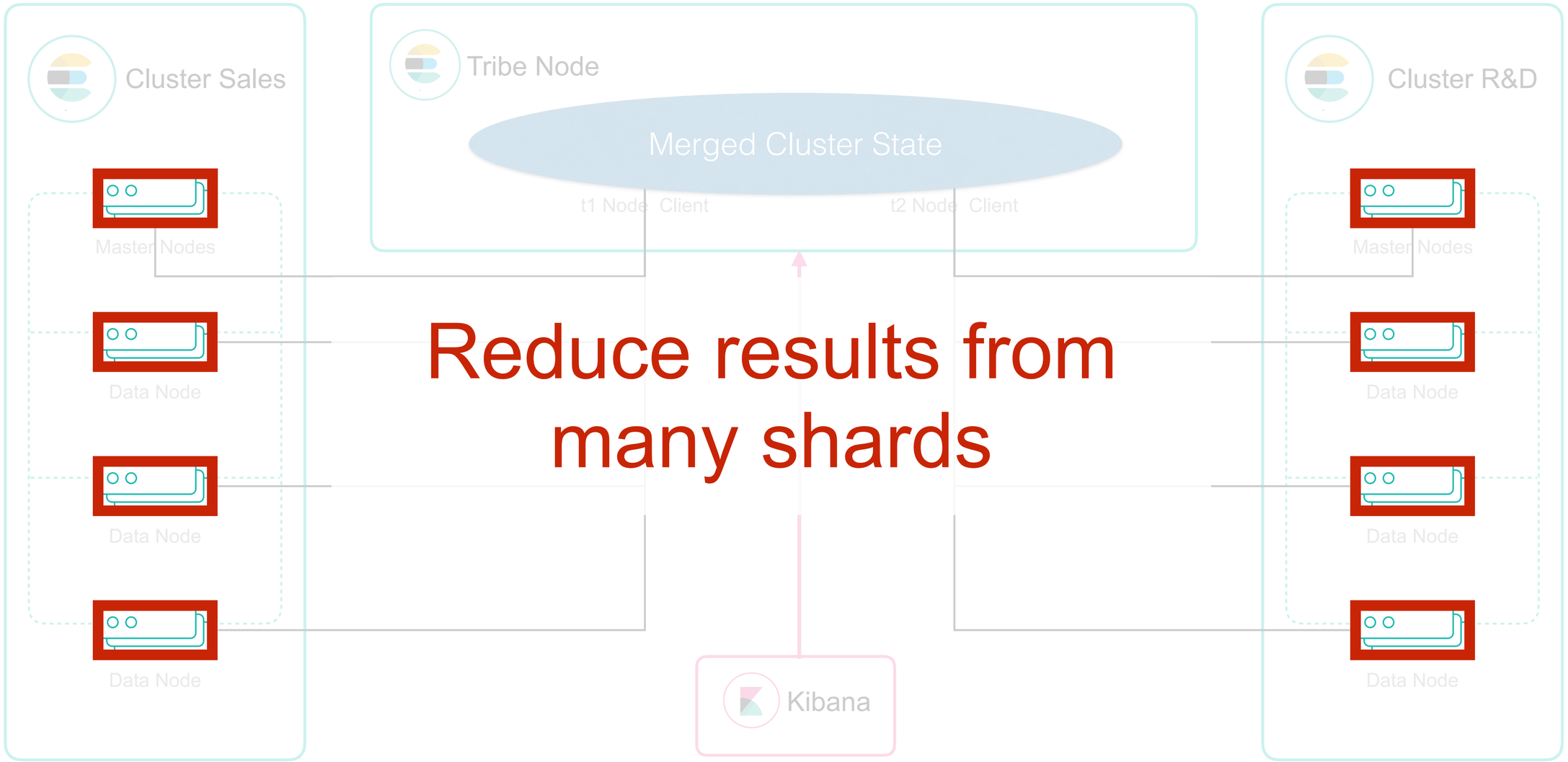
Problems With How the Tribe Node works



Problems With How the Tribe Node works



Problems With How the Tribe Node works



The Tribe Node is Dead

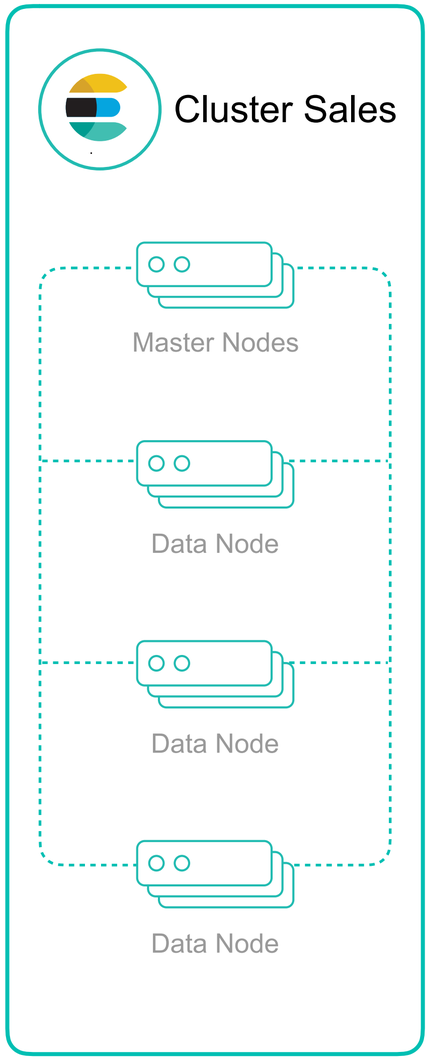
Long Live
Cross-Cluster Search!

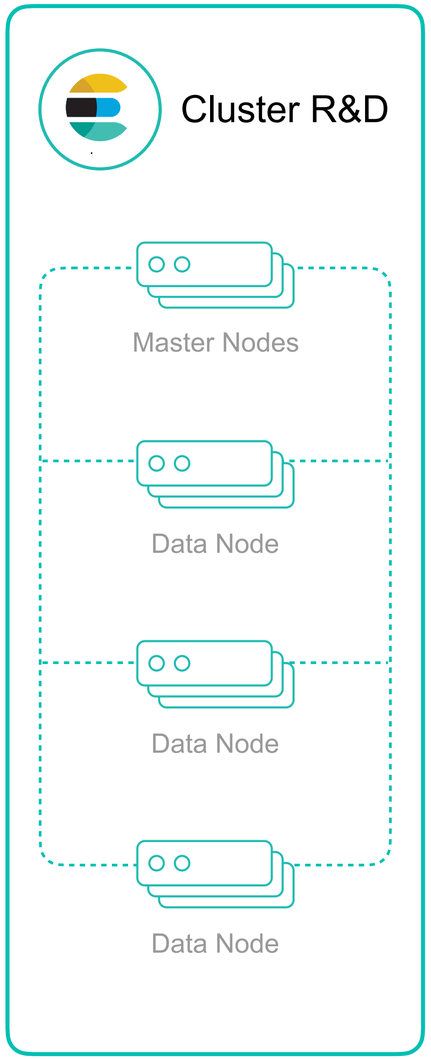
Minimal viable solution to supersede tribe

Reduces the problem domain to query execution

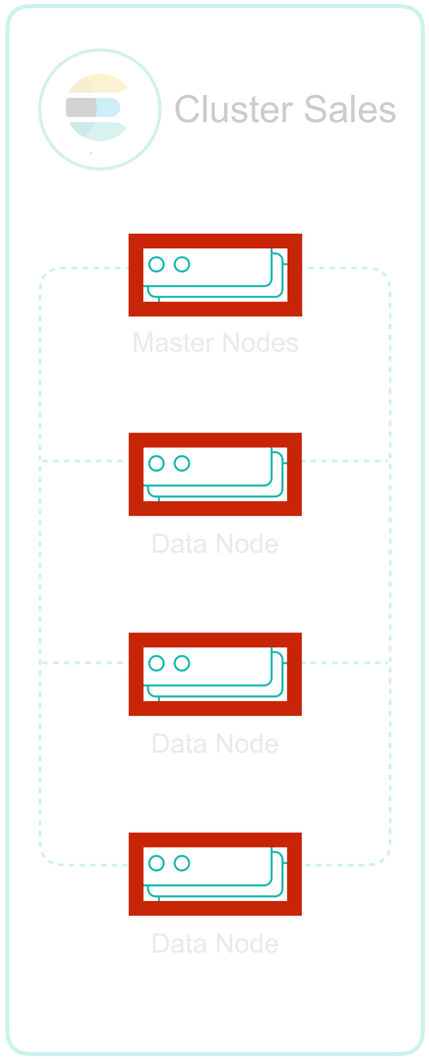
Cluster related information is reduced to a namespace

How Cross-Cluster search works

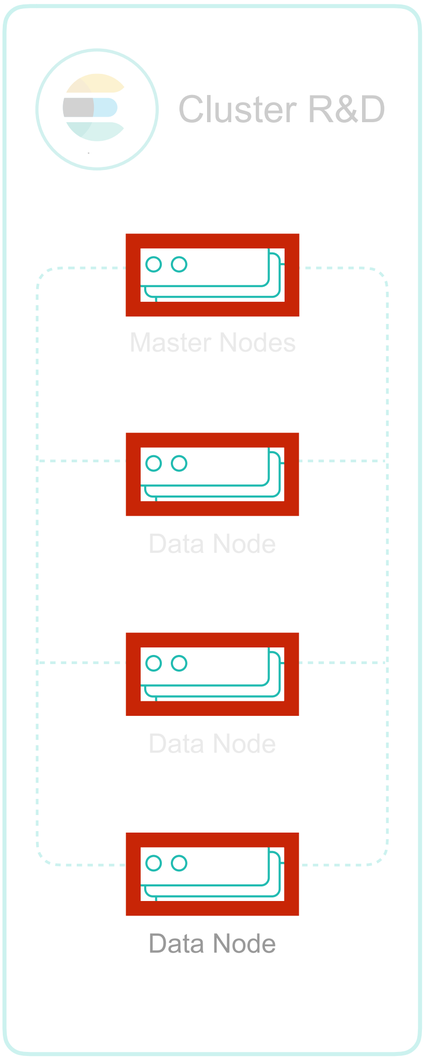




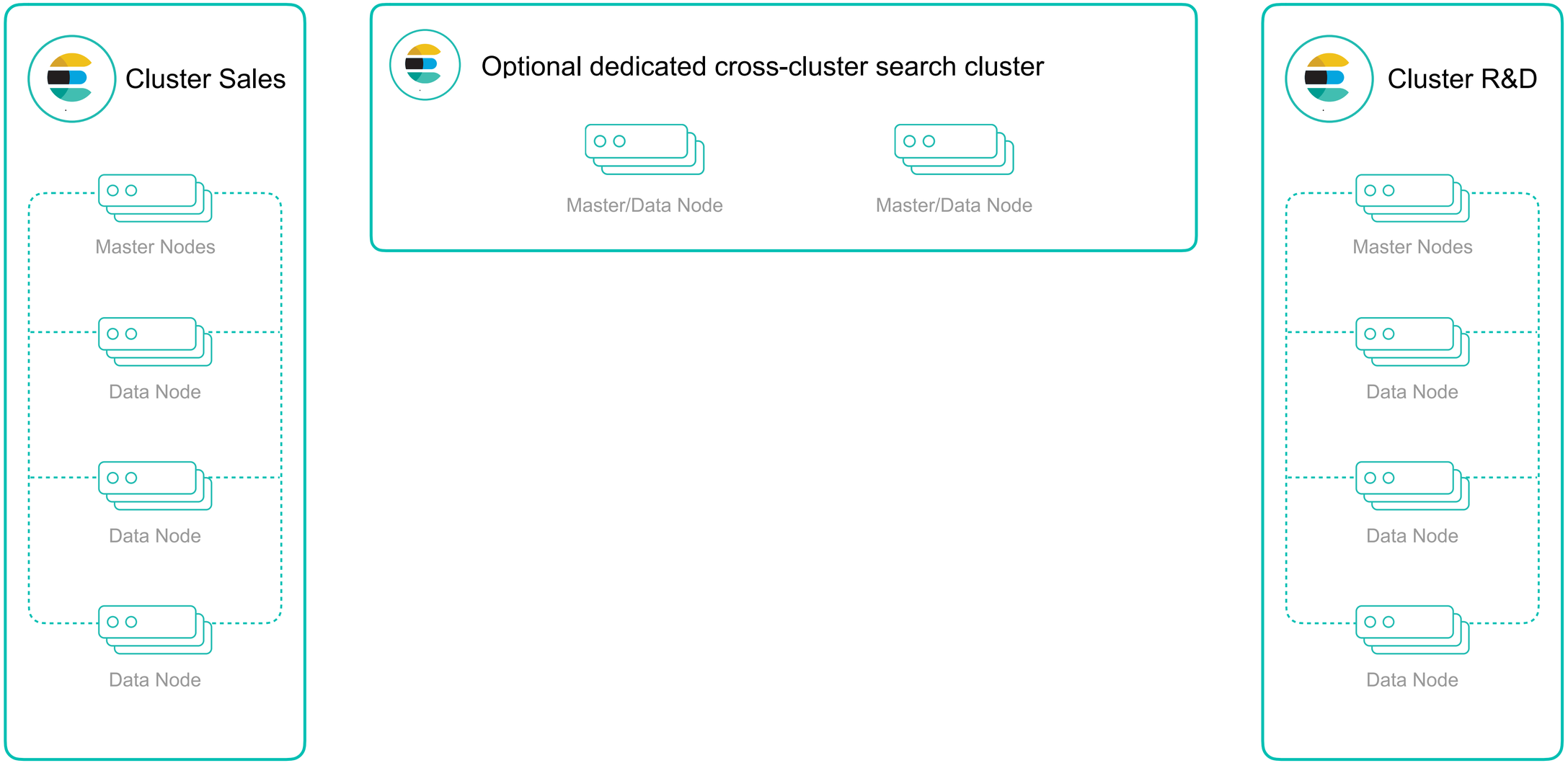
How Cross-Cluster search works



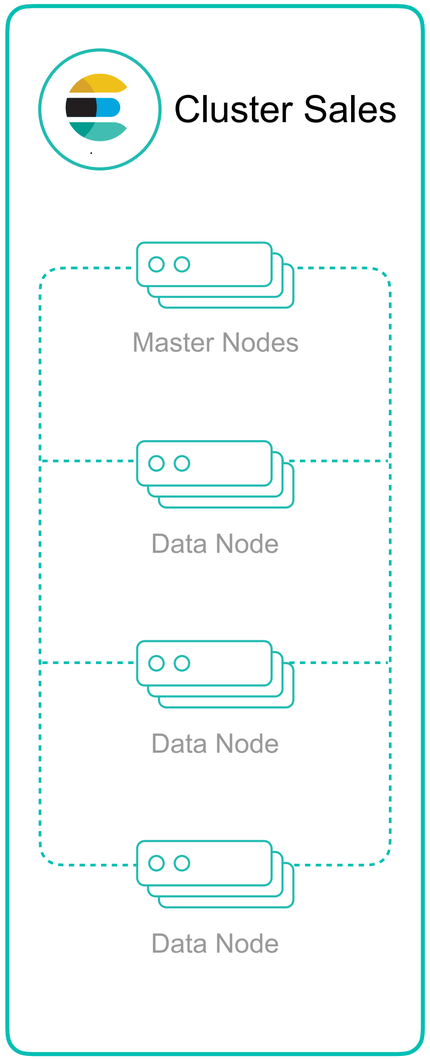
Any node can perform cross-cluster search

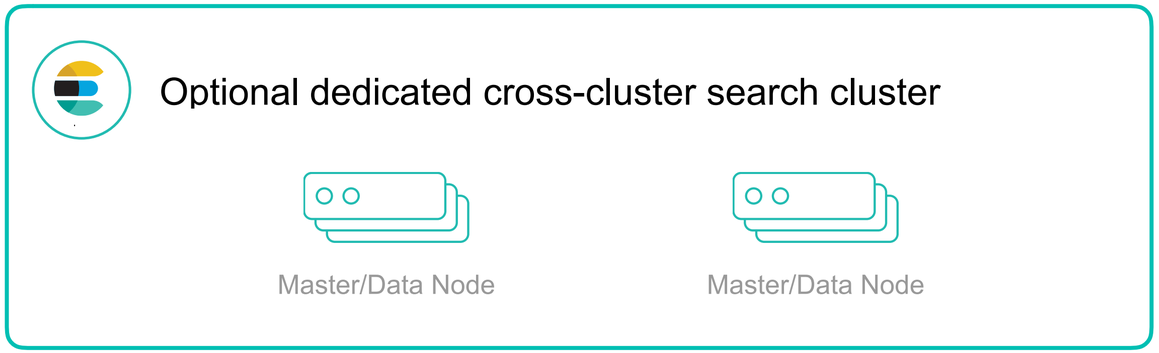


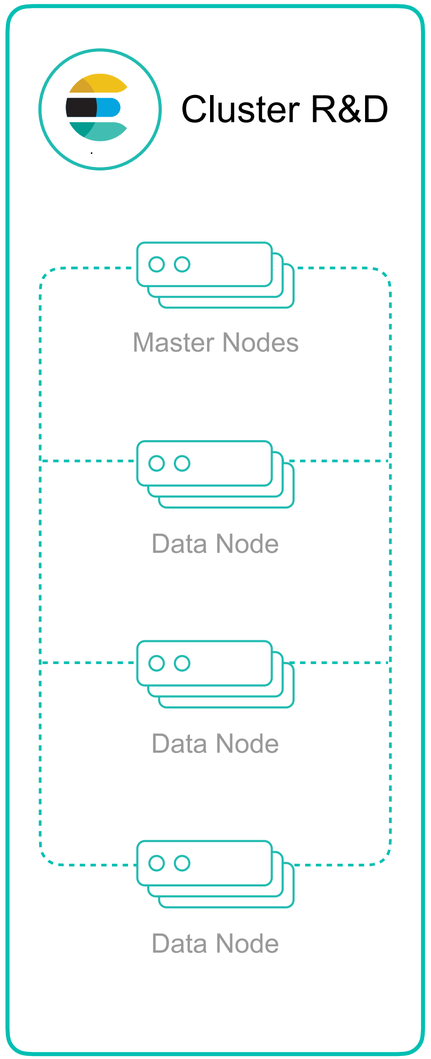
How Cross-Cluster search works



How Cross-Cluster search works



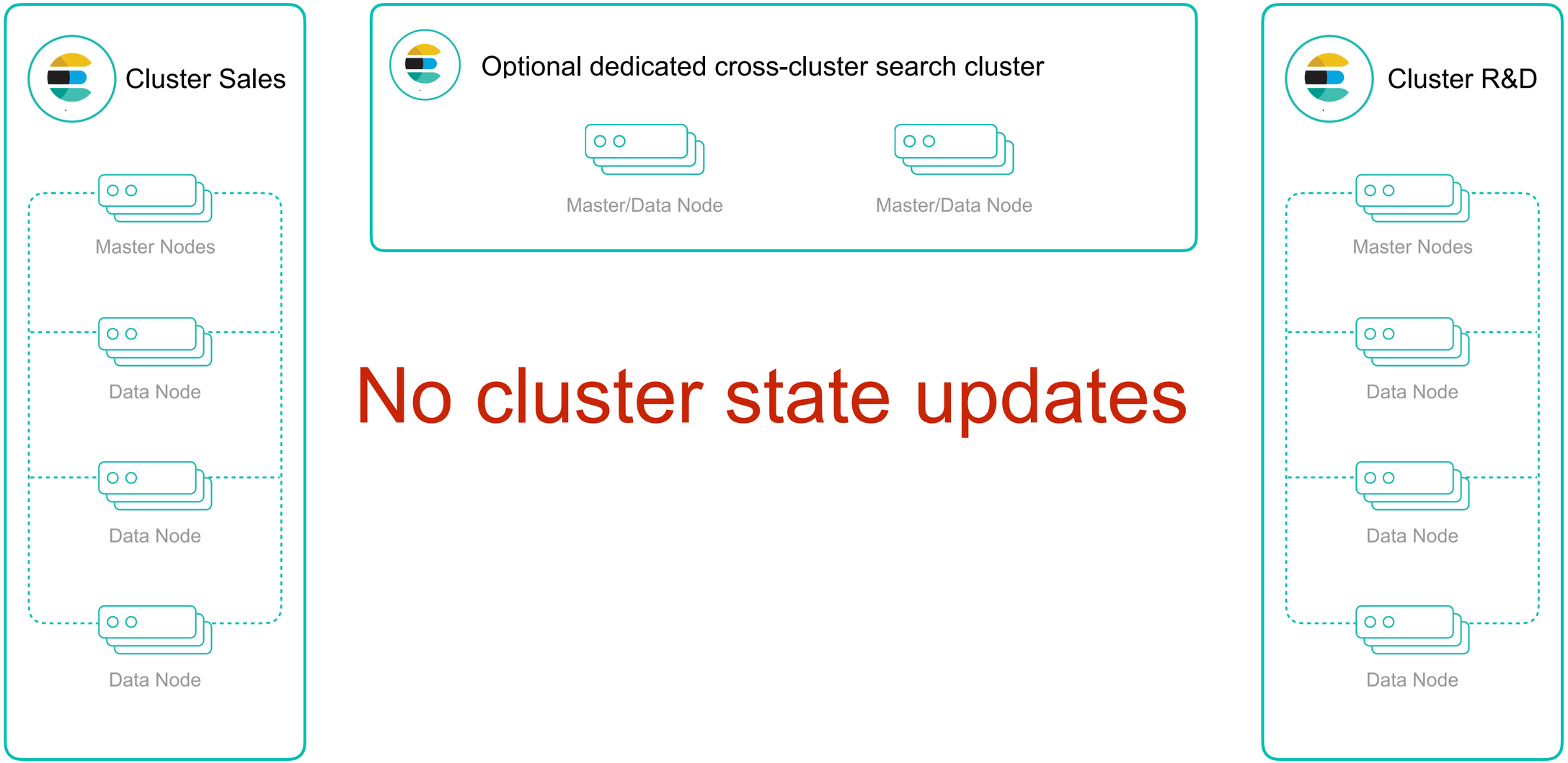





```
PUT _cluster/settings
{
  "transient": {
    "search.remote": {
      "sales.seeds":   "10.0.0.1:9300",
      “r_and_d.seeds”: "10.1.0.1:9300"
    }
  }
}
```

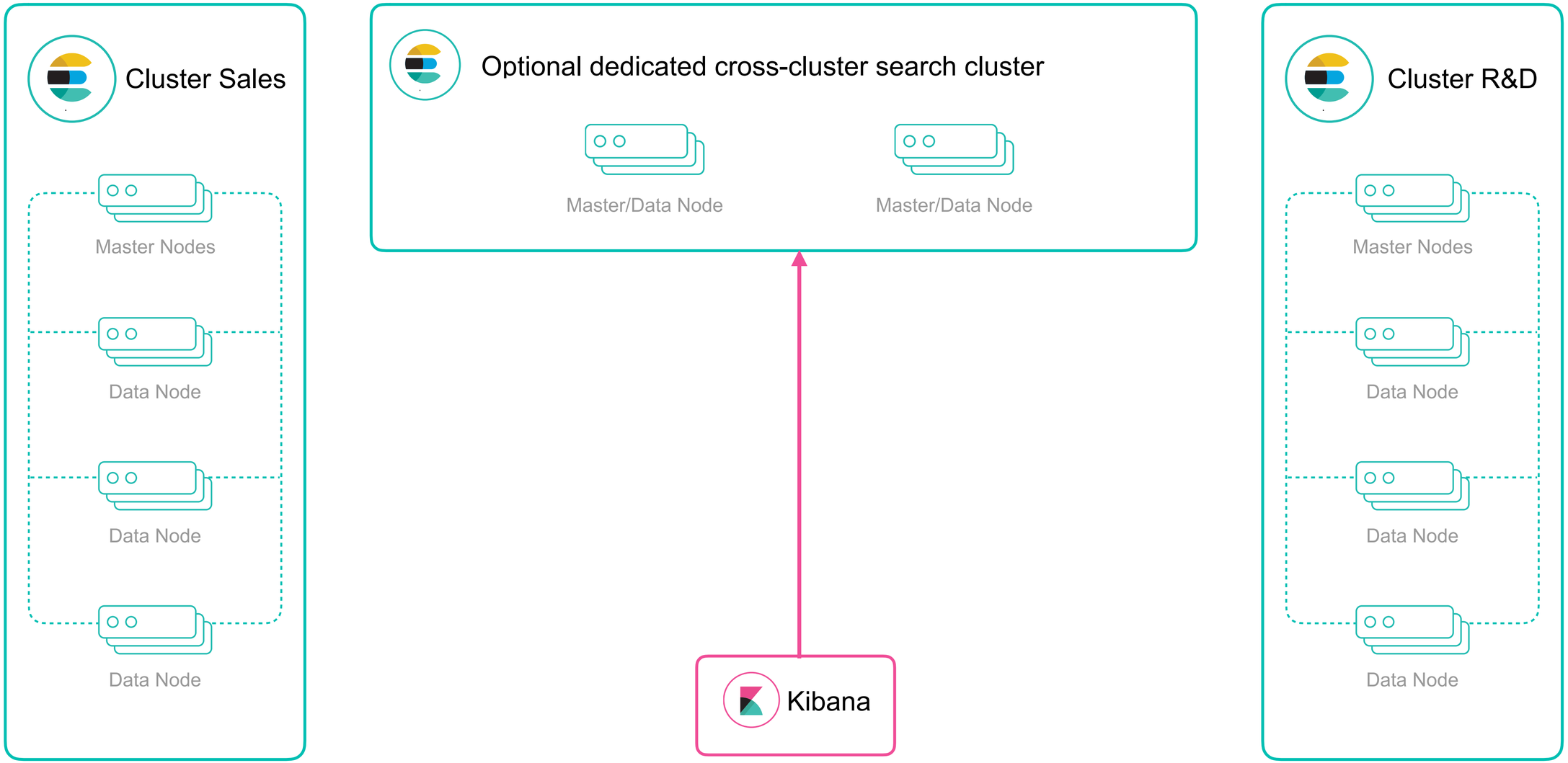
Dynamic settings

How Cross-Cluster search works

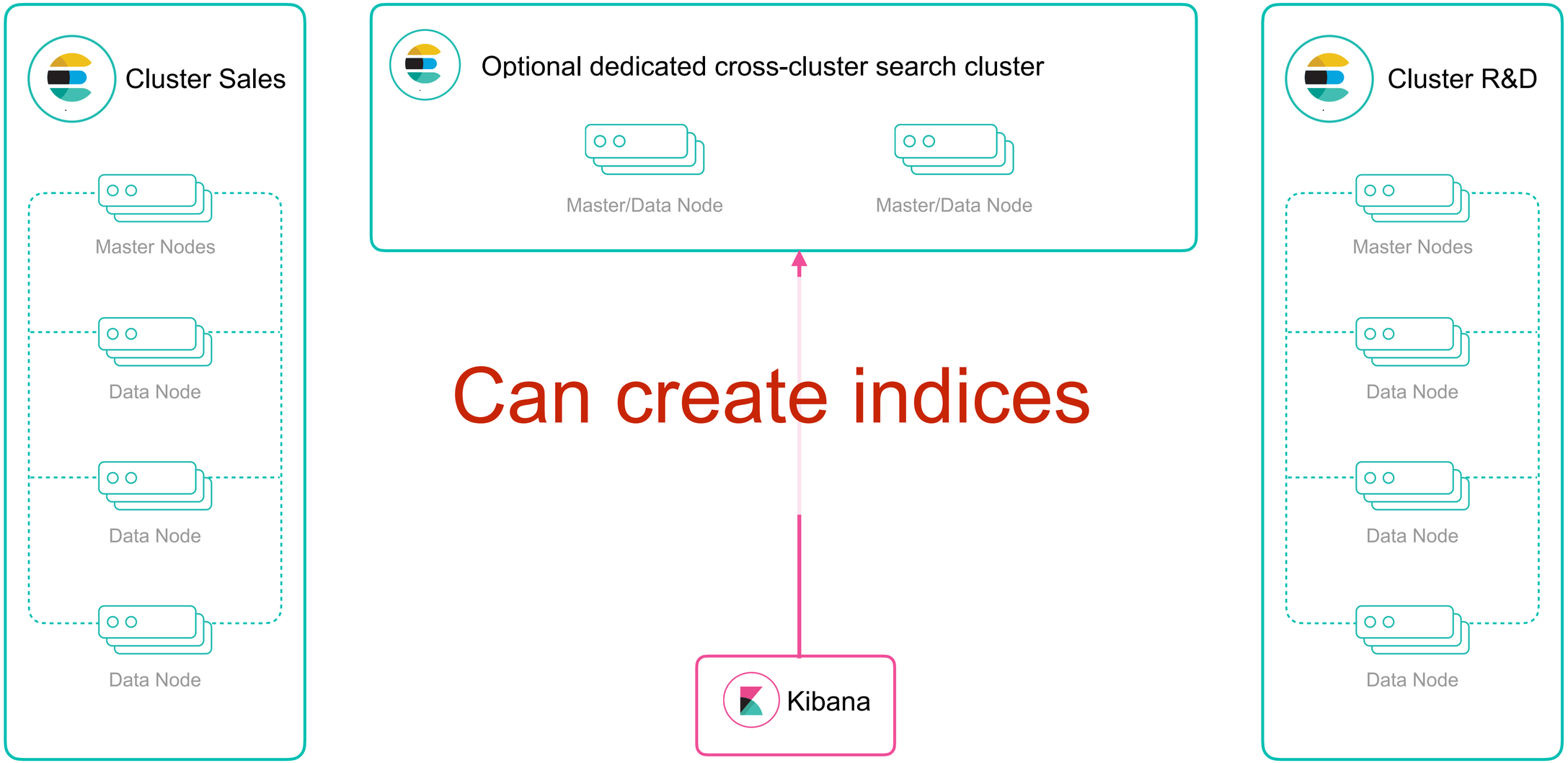


No cluster state updates

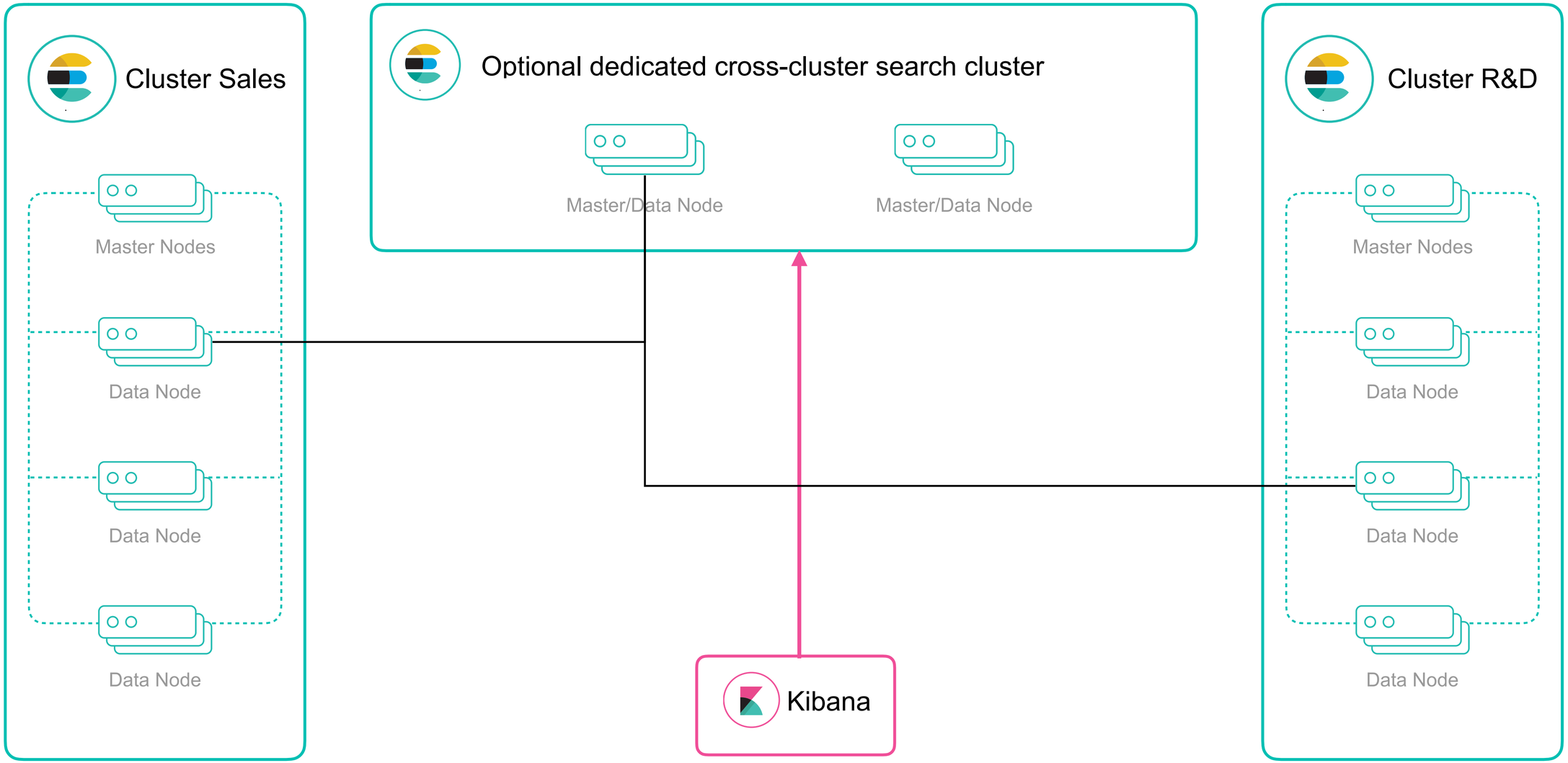
How Cross-Cluster search works



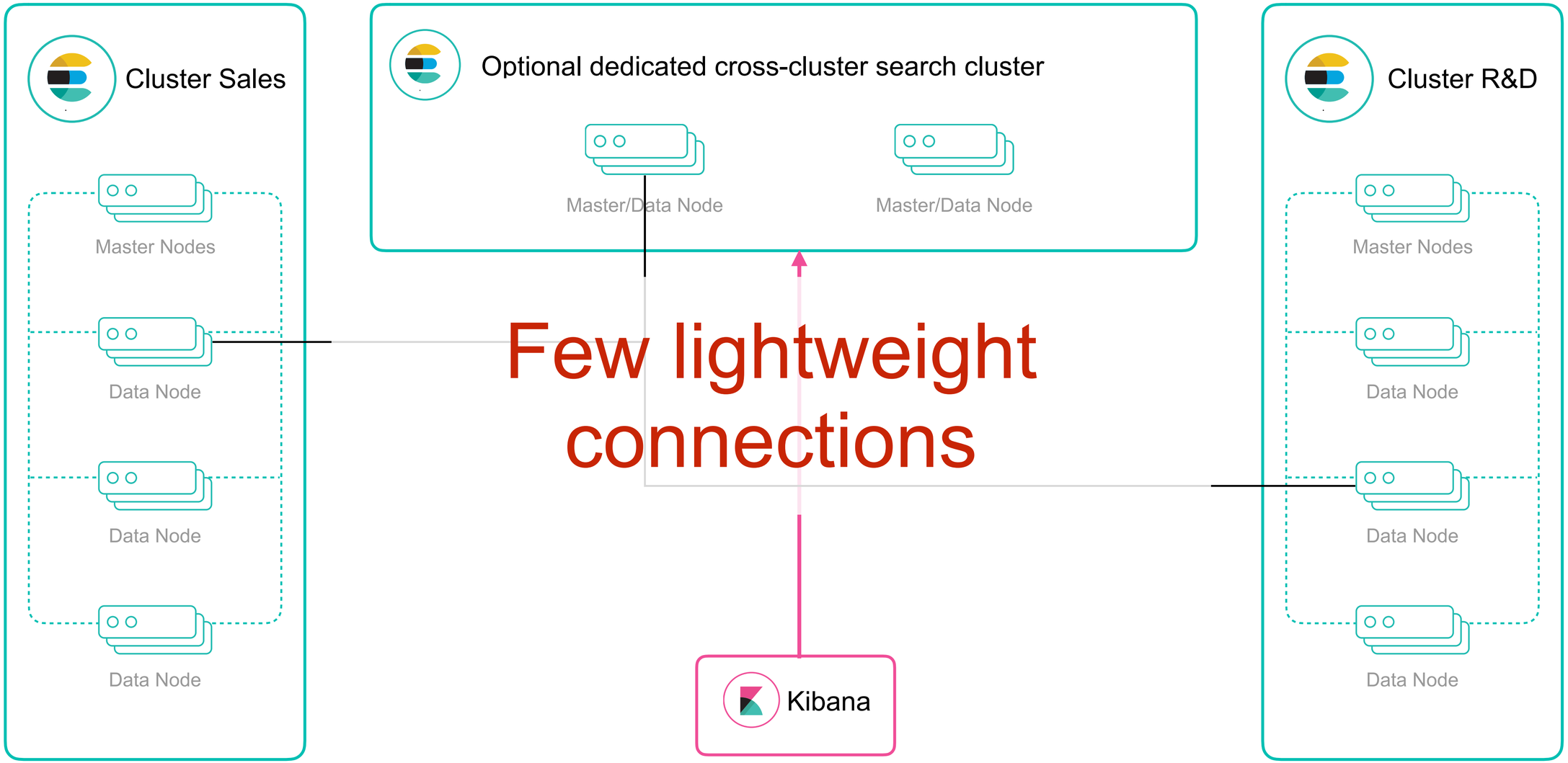
How Cross-Cluster search works



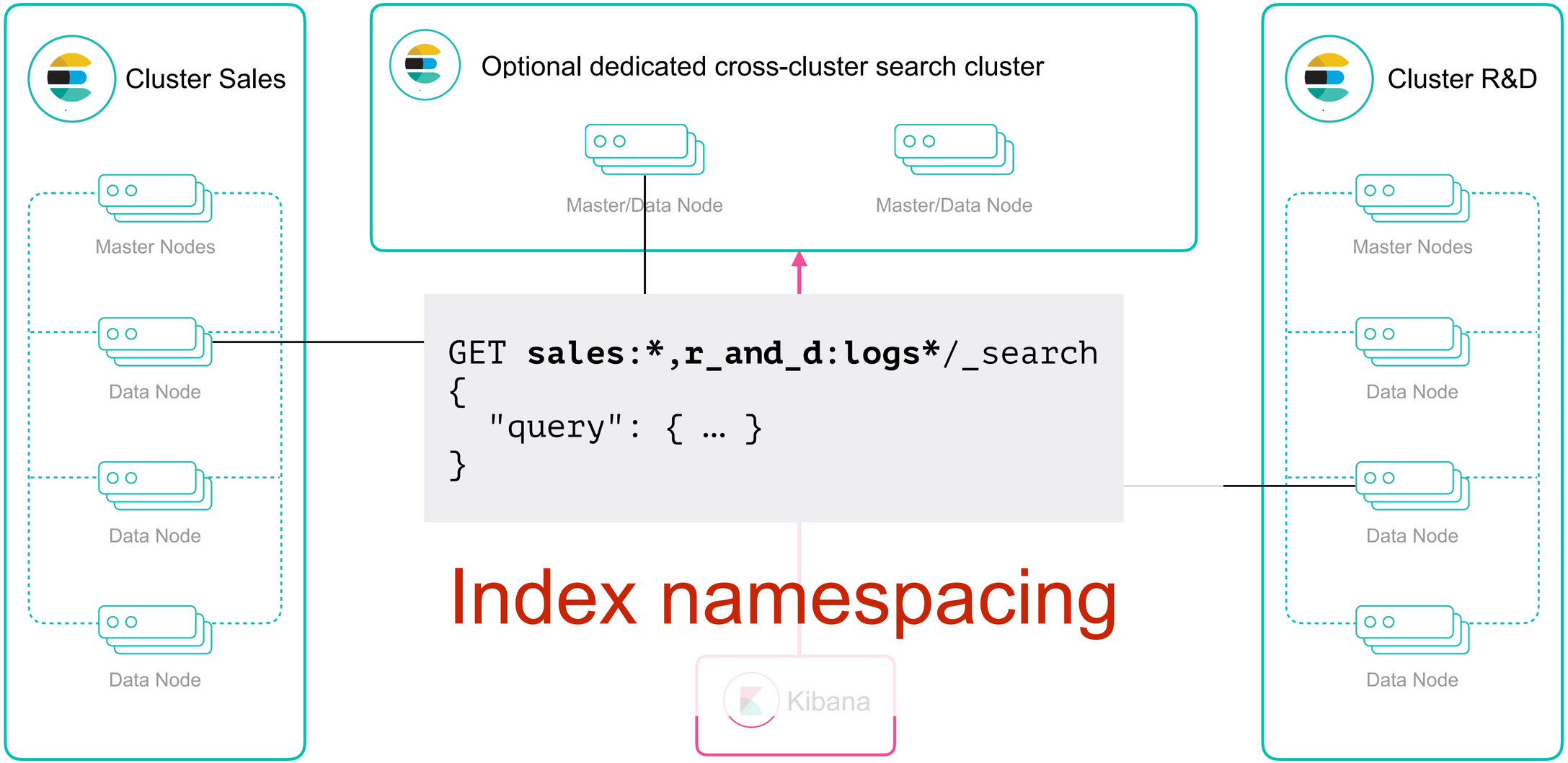
How Cross-Cluster search works



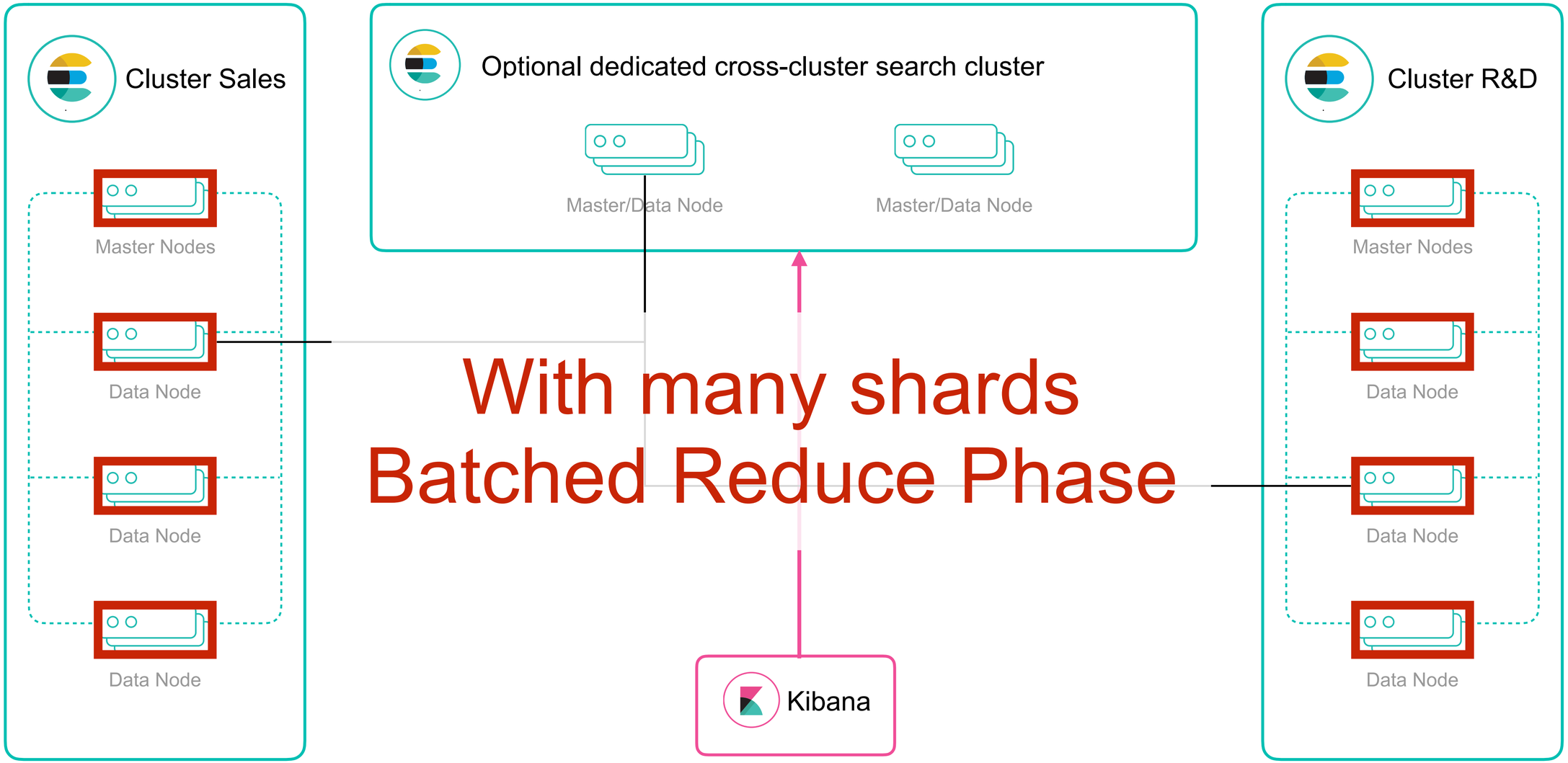
How Cross-Cluster search works



How Cross-Cluster search works



How Cross-Cluster search works



Cross-Cluster Search
v5.3.0

Batched Reduce Phase
v5.4.0

v6 and beyond

Doc Values
v2.x

Doc Values

- Columnar store
- Fast access to a field’s value for many documents.
- Used for aggregations, sorting, scripting, and some queries
- Written to disk at index time.
- Cached in the file-system cache

Doc Values - Dense Values

Segment 1

Docs	Field 1	Field 2
1	One	A
2	Two	B
3	Three	C

Segment 2

Docs	Field 1	Field 2
1	Four	D

Doc Values - Dense Values

Segment 1

Docs	Field 1	Field 2
1	One	A
2	Two	B
3	Three	C

Segment 2

Docs	Field 1	Field 2
1	Four	D

Merged Segment 3

Docs	Field 1	Field 2
1	One	A
2	Two	B
3	Three	C
4	Four	D

Doc Values - Sparse Values

Segment 1

Docs	Field 1	Field 2
1	One	A
2	Two	B
3	Three	C

Segment 2

Docs	Field 3	Field 4	Field 5
1	Foo	Bar	Baz

Doc Values - Sparse Values

Segment 1

Docs	Field 1	Field 2
1	One	A
2	Two	B
3	Three	C

Segment 2

Docs	Field 3	Field 4	Field 5
1	Foo	Bar	Baz

Merged Segment 3

Docs	Field 1	Field 2	Field 3	Field 4	Field 5
1	One	A	Null	Null	Null
2	Two	B	Null	Null	Null
3	Three	C	Null	Null	Null
4	Null	Null	Foo	Bar	Baz

Sparse Doc Values
Lucene 7

Index Sorting
Lucene 7

Index sorting

- Sort index by e.g. weight, recency, or popularity
- Ultra-fast search - can terminate once enough hits found

Index sorting

- Sort index by e.g. weight, recency, or popularity
- Ultra-fast search - can terminate once enough hits found
- Even helps with total count and aggregations
- Sort index by low cardinality terms - faster search
- Better sparse index compression
- Slower indexing, good for static indices

Sequence Numbers
v6.0.0

Sequence Numbers

- Internal Feature
- Every operation gets a sequence number
- In 6.0: Fast replica recovery on active indices
- Lays groundwork for:
  - Primary-Replica syncing when Primary fails
  - Cross Data-Centre Recovery
  - Changes API

Upgrading

Rolling Upgrades
v6.0.0

Rolling Upgrades

- Upgrade from **5.latest** to **6.latest**, without a full cluster restart
- Why now and not earlier?
  - Testing needs to be ready
  - The team and the code must be ready
  - Growing user-base and faster release cycles required less painful upgrades

Rolling Upgrades

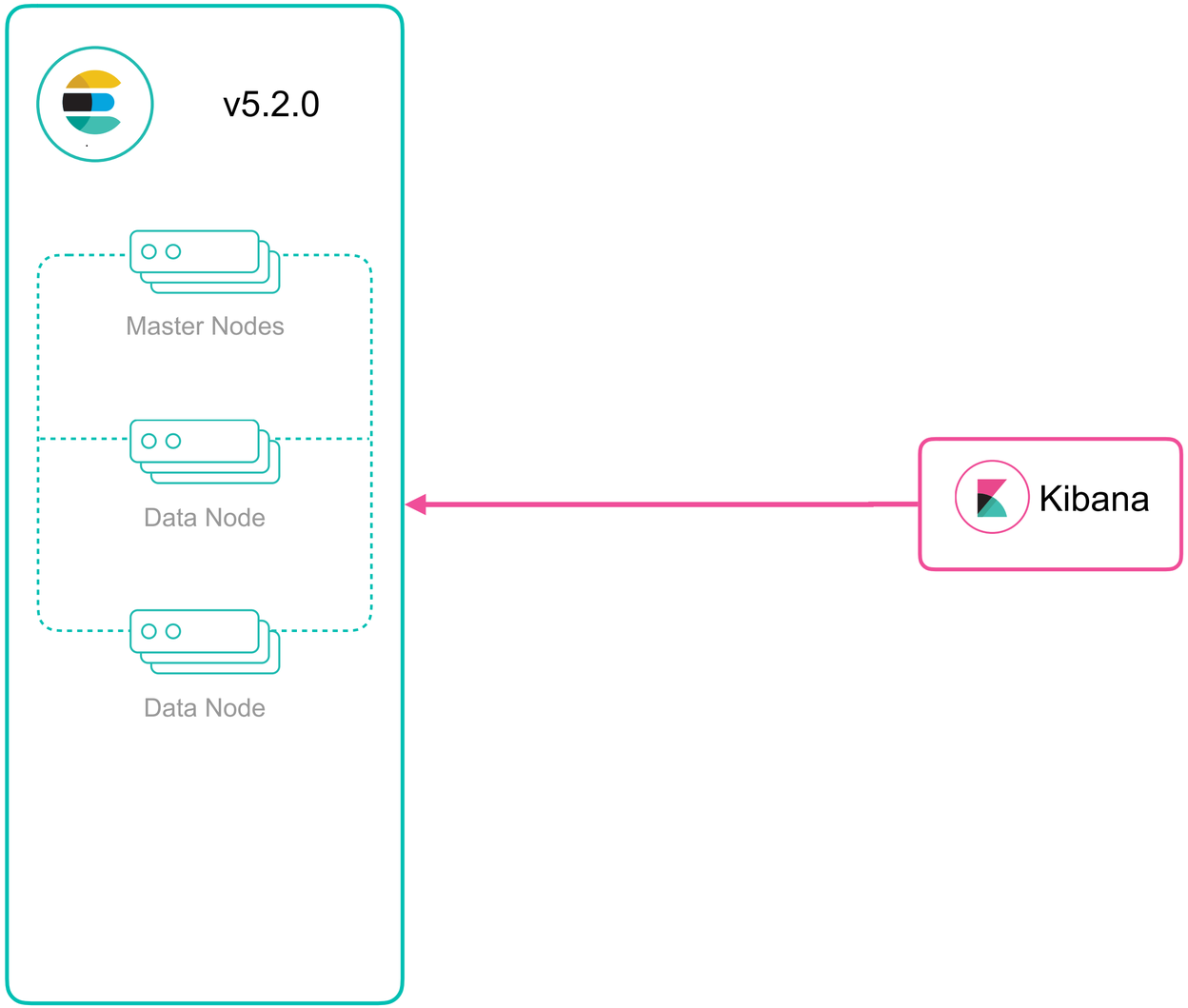
- What is 5.latest?
  - It’s the latest release of 5.x that is GA once 6.0.0 goes GA
  - All 6.x releases will allow upgrading from that 5.x release
  - There might be subsequent 5.x releases that are also eligible for upgrades to 6.x

Rolling Upgrades

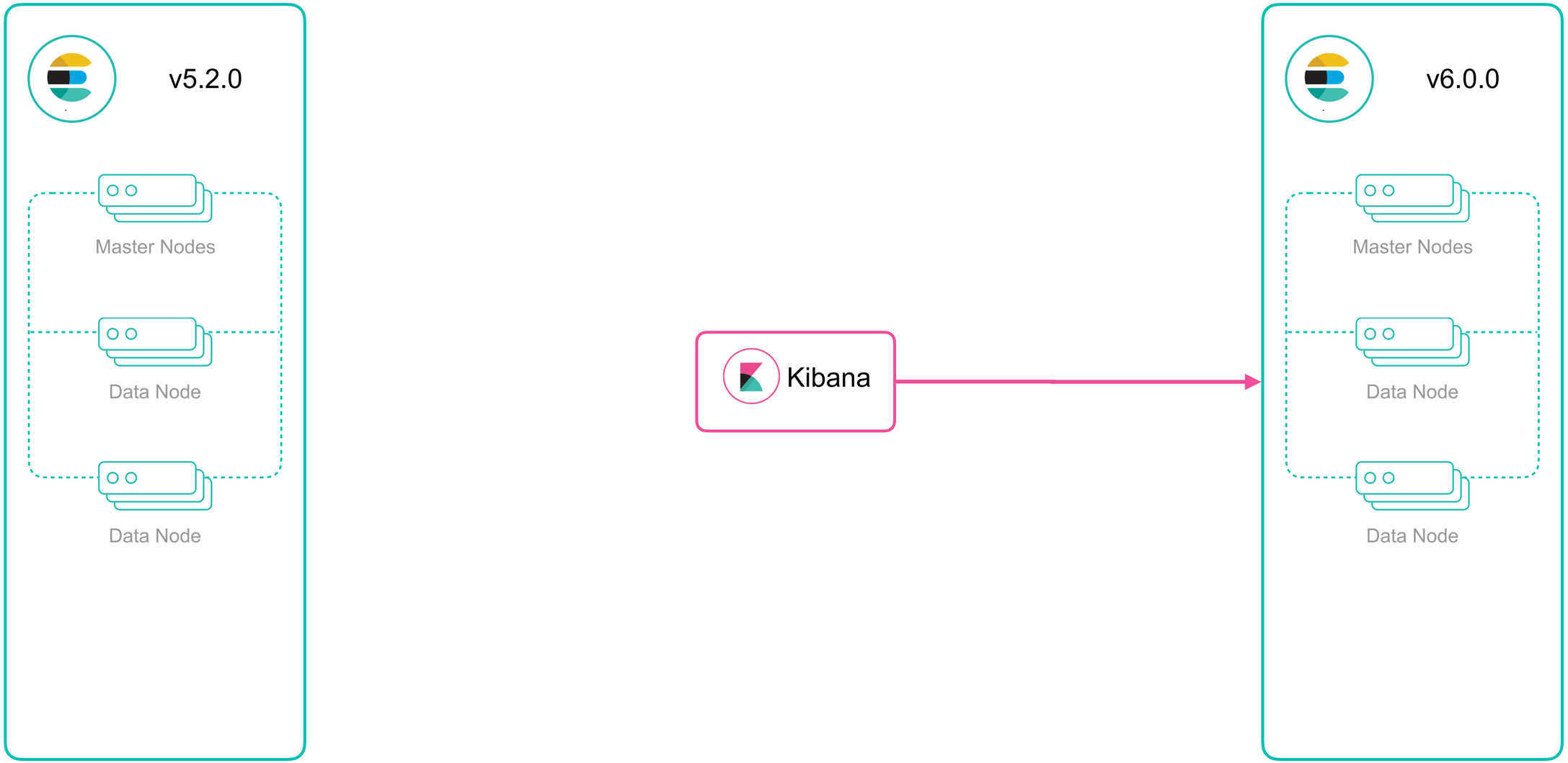
- Caveats:
  - If using security, must have TLS enabled
  - Reserve the right to require full cluster restart in the future - but only if absolutely necessary
  - All nodes must be upgraded to 5.latest in order to upgrade
  - Indices created in 2.x still need to be reindexed before upgrading to 6.x

Cross Major Version Search
v6.0.0

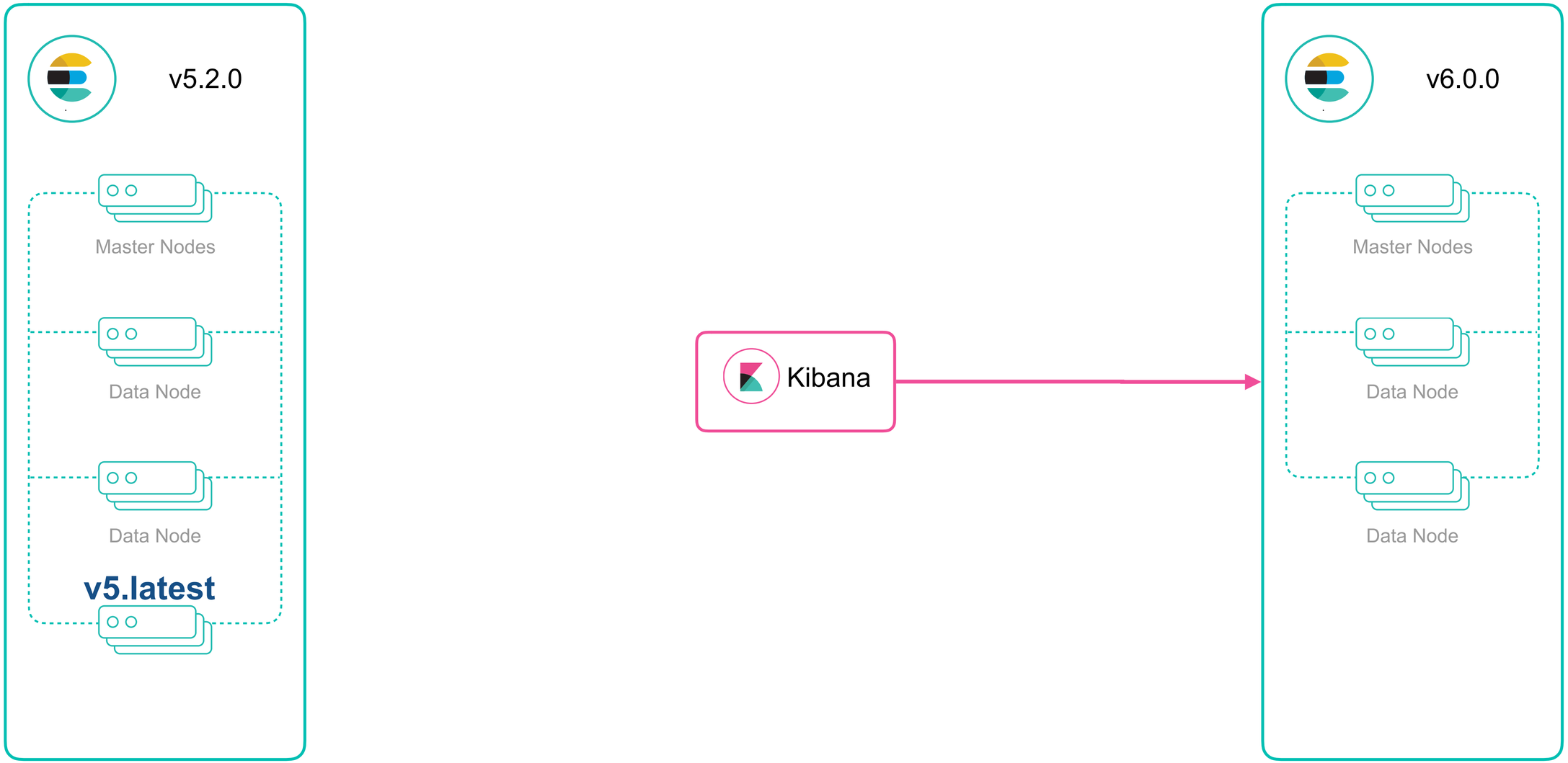
Cross Major Version Search



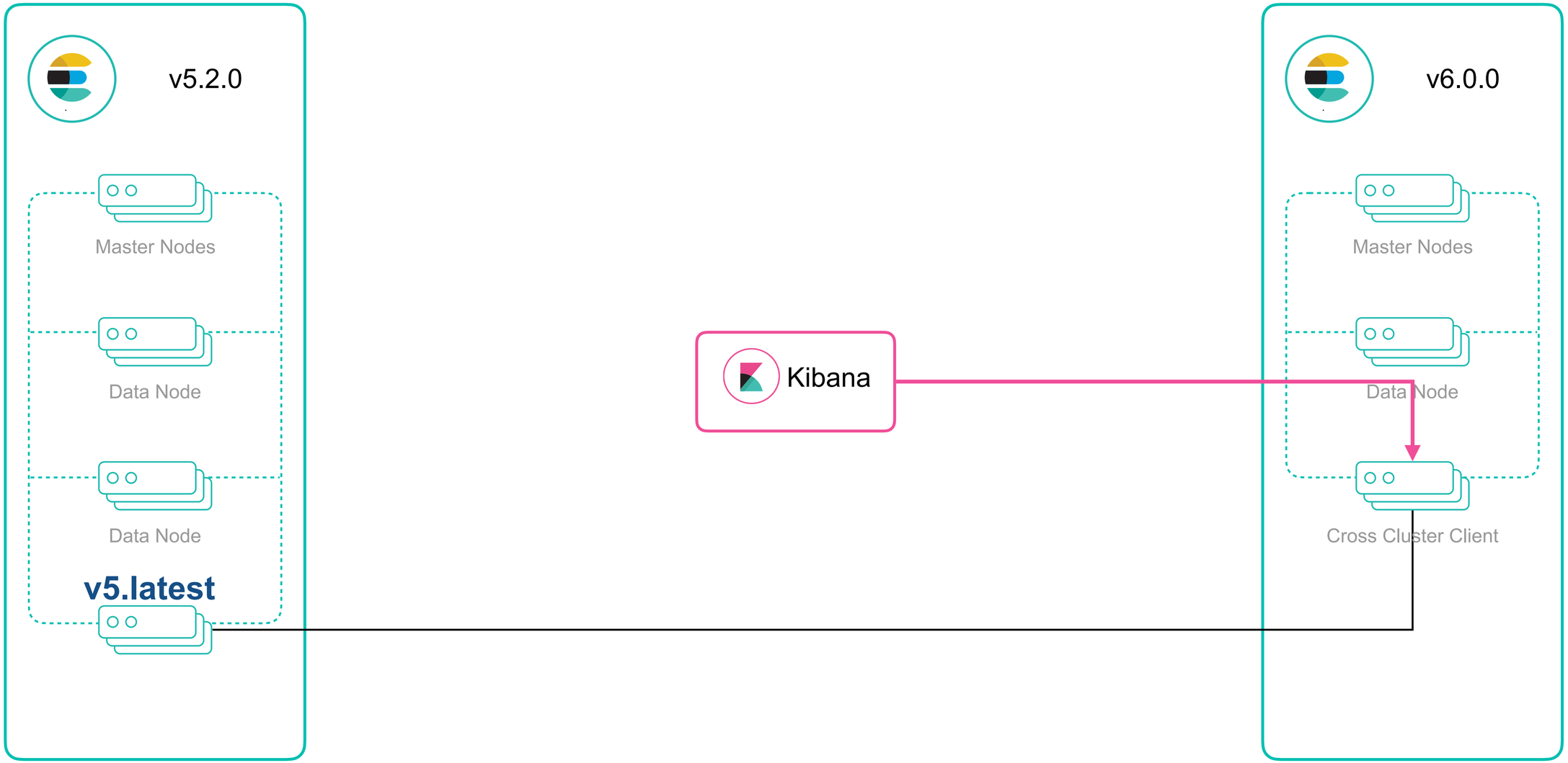
Cross Major Version Search



Cross Major Version Search



Cross Major Version Search



Questions?

Other Talks You Should See

- “Get the Lay of the Lucene Land”
  Adrien Grand - Wednesday

- “Consensus and Replication in Elasticsearch”
  Boaz Leskes, Jason Tedor, and Yannick Welsch - Wednesday

- “Elasticsearch Search Improvements”
  Jim Firenczi, Lee Hinman, Nick Knize - Thursday

- “Secure, Fast, and Painless”
  Nik Everett - Thursday

More Questions?

Visit us at the AMA

www.elastic.co

Please attribute Elastic with a link to elastic.co