RADICALBIT

Deep, Different, Disruptive.

NSDb (Natural Series Db)

A time series streaming oriented database optimized for the serving layer

Roberto Bentivoglio - @robbenti
Saverio Veltri - @save_veltri

Berlin – 13 September 2017

WHO WE ARE

Radicalbit is a highly **specialized software firm**, founded in Milan, Italy, in 2015.

We are focussed on the design and development of **Fast Data products and solutions**, using the best-of-breed **streaming technologies (and of course Flink)**.

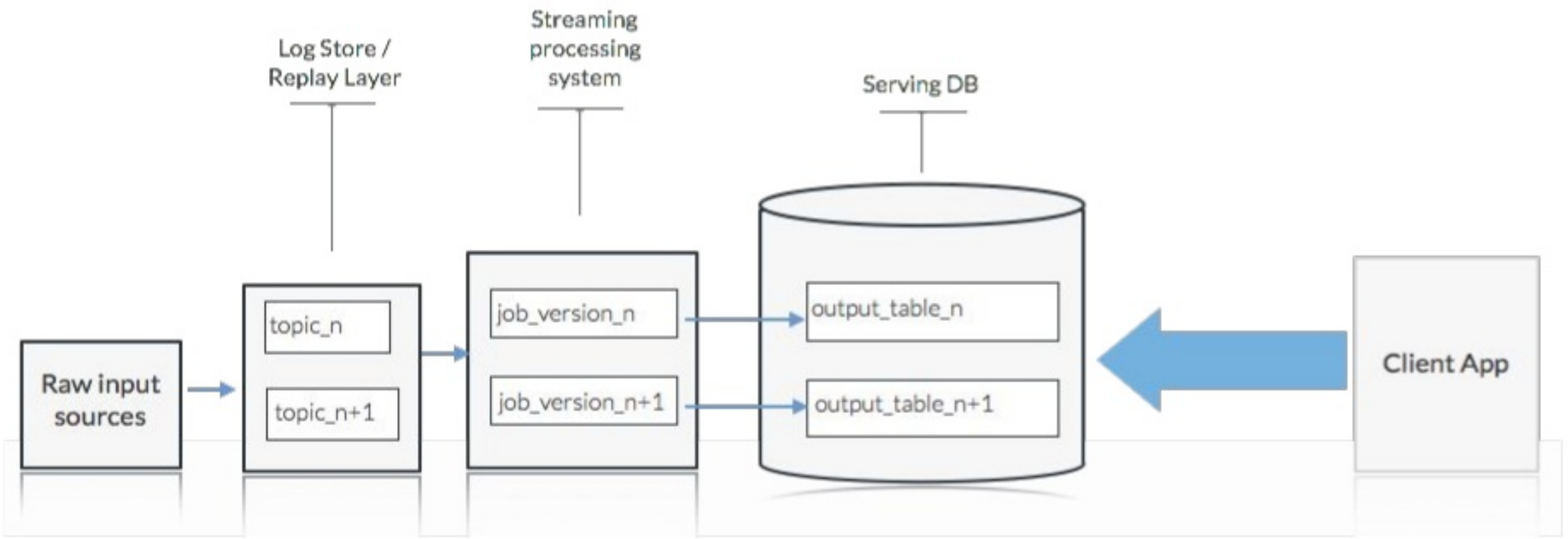
We're rapidly growing, offering our enterprise solutions to data intensive companies.

RADICALBIT
Deep. Different. Disruptive.

NSDb - MOTIVATIONS

- Our Fast Data distribution lacked of Time Series functionalities
- Too many licensing and pricing issues exploring third-party OEM solutions
- Third-party solutions don't completely fit
  - our non-functional requirements
  - our streaming real-time analytics needs
- Have a deep technical ownership of the solution
- Publish / Subscribe model
- Push capabilities
- Create a community around Radicalbit products
- We'd like to grow up our Scala codebase (we're Scala guys)

KAPPA ARCHITECTURE



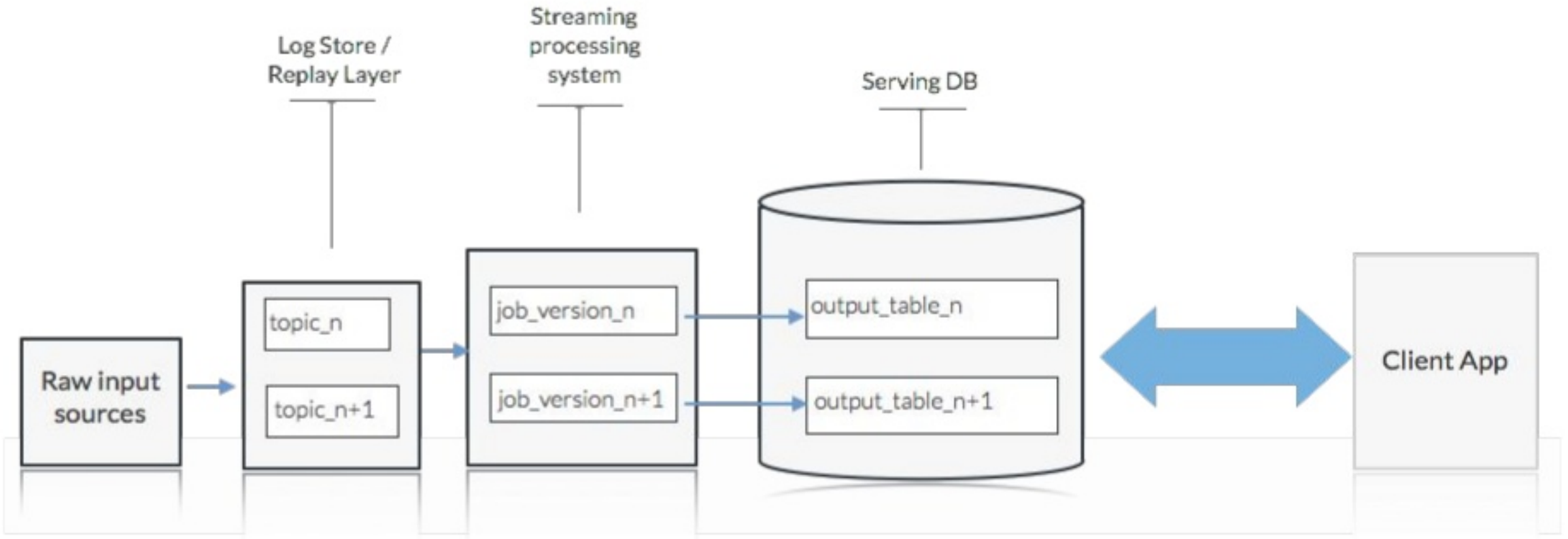
RADICALBIT

CHALLENGES IMPLEMENTING KAPPA ARCHITECTURE

- Everything is handled as a **streaming event**. This is great!
- There are plenty of solutions for handling real time writes / journaling (Some of them are being presented concurrently to our talk :))
- If a traditional database is being used, any possible **real time capability** will be lost on the **serving layer**
- Any application must query periodically the DB for gathering data in a **pull fashion**

RADICALBIT

KAPPA ARCHITECTURE REVISITED



RADICALBIT

NSDb - MAIN FEATURES

- Optimized time series management
- Focus on read performance
- High availability and clustering
- Ad-hoc publish-subscribe streaming feature (using WebSockets)
- SQL like query language
- Fluent Java / Scala Api

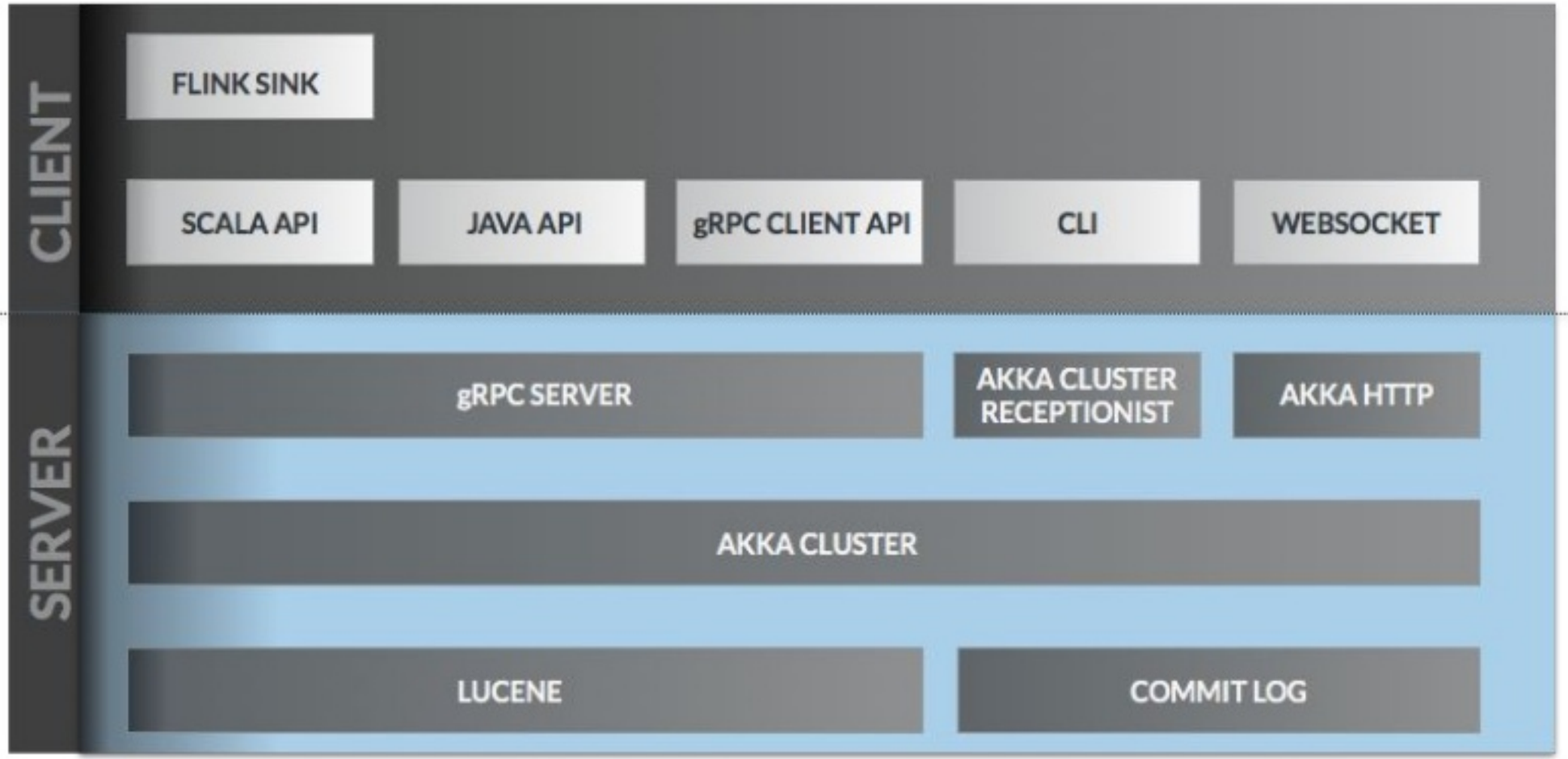
RADICALBIT
Deep. Different. Disruptive.

NSDb - MAIN FEATURES

- **Apache Flink** integration (sink)
- Integration with many languages thanks to gRPC (Go, Python, C++, ...)
- Metrics versioning
- Native support for Apache Avro (ingestion, schema publishing and versioning)
- JDBC support
- Import / Export features

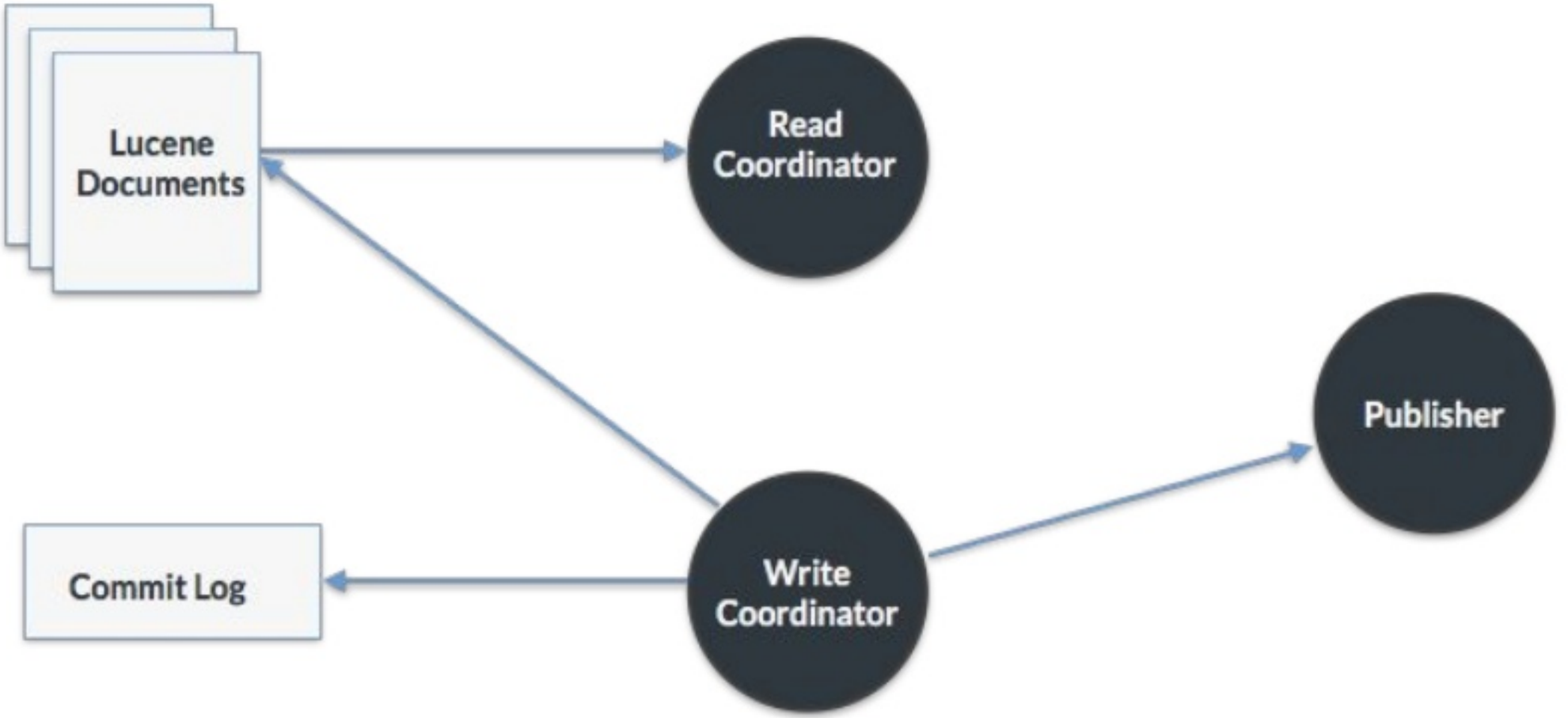
RADICALBIT
Deep, Different. Disruptive.

NSDb - OVERALL ARCHITECTURE



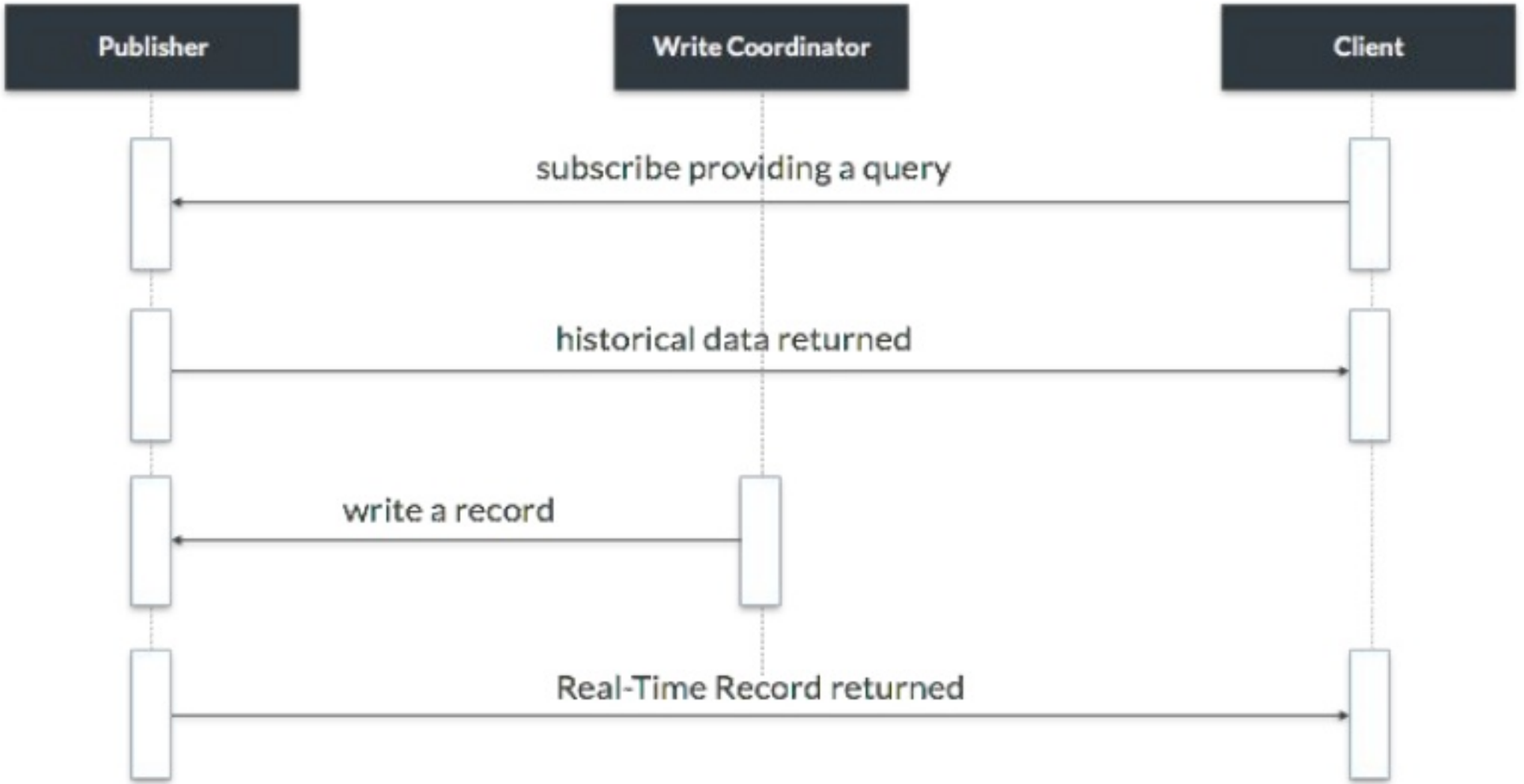
RADICALBIT
Deep. Different. Disruptive.

NSDb - CORE ARCHITECTURE



RADICALBIT
Deep. Different. Disruptive.

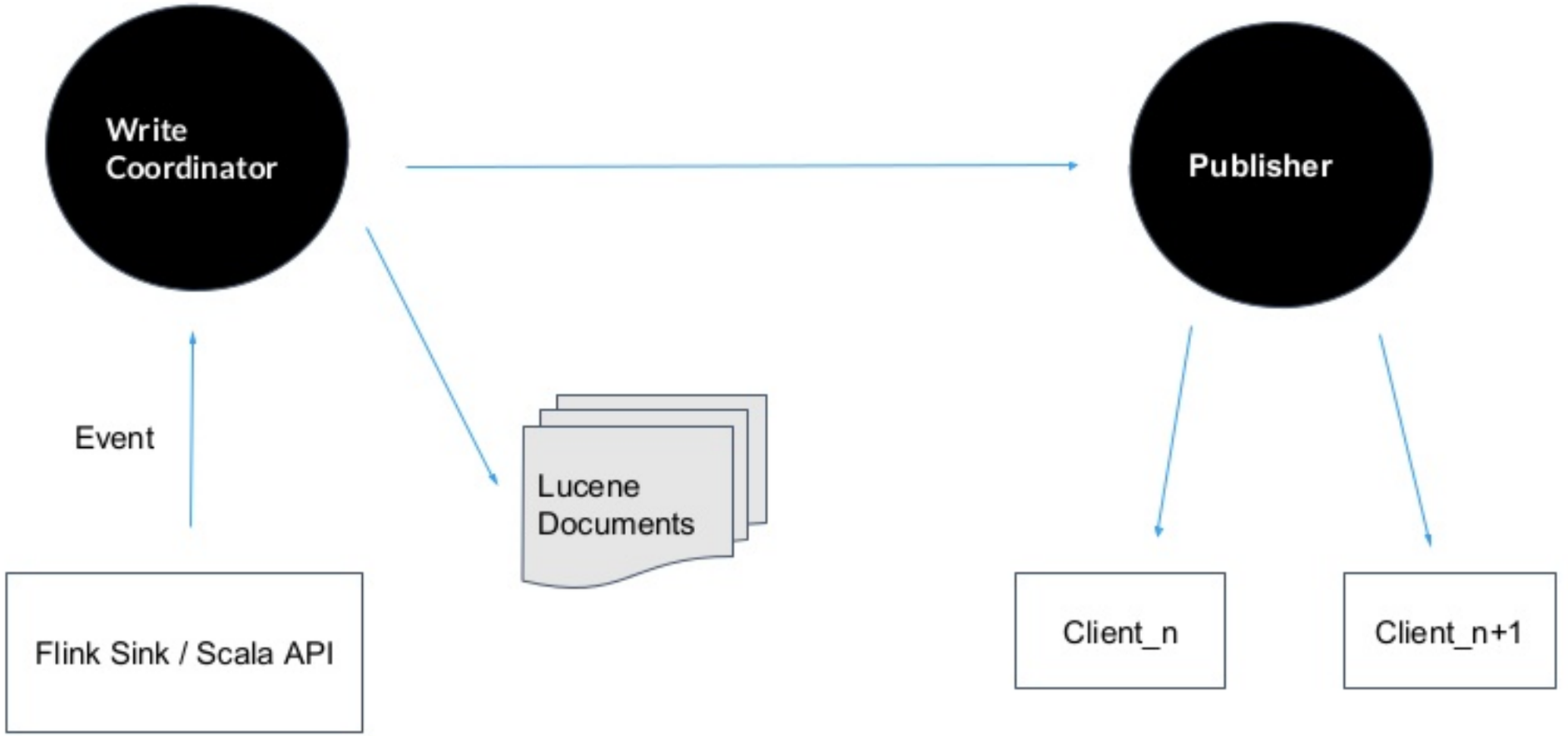
NSDb | PUBLISH-SUBSCRIBE



RADICALBIT

NSDb - CONSISTENCY

- Eventual consistency
- Real time delivery for subscribed client



RADICALBIT

SCALING

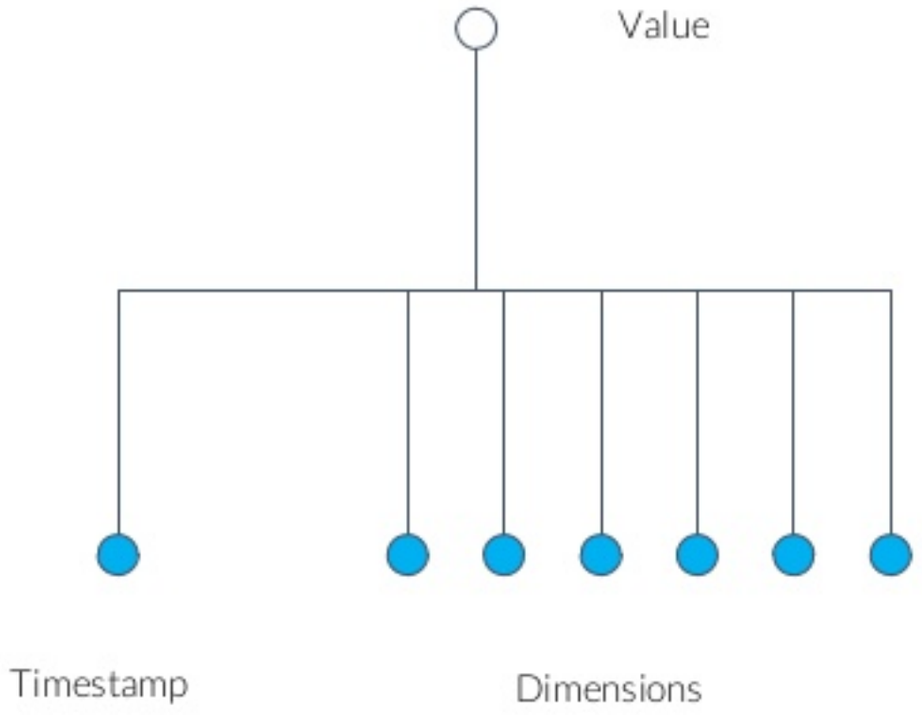
- Scaling out is a challenging topic
- How would it be possible to scale a generic Sql or NoSql workload ?

¯_(ツ)_/¯

- In most of the well known NoSql Databases data partitioning is something demanded to the user
- Unless boundaries are being defined

NSDb - BIT

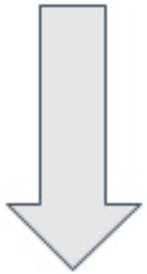
MultiDimensional time series value



RADICALBIT
Deep. Different. Disruptive.

NSDb - SCALING

- NSDB's Bits are immutable. New data continuously arrives, and will be always inserted and never updated.
- Time series data are naturally ordered by time



Time is Totally a Gentleman!

RADICALBIT
Deep. Different. Disruptive.

NATURAL TIME SHARDING

- Workloads can be partitioned across time and space
- space partitioning will be demanded to Lucene indexes
- Time partitioning leverages intrinsic time linearity

RADICALBIT
Deep. Different. Disruptive.

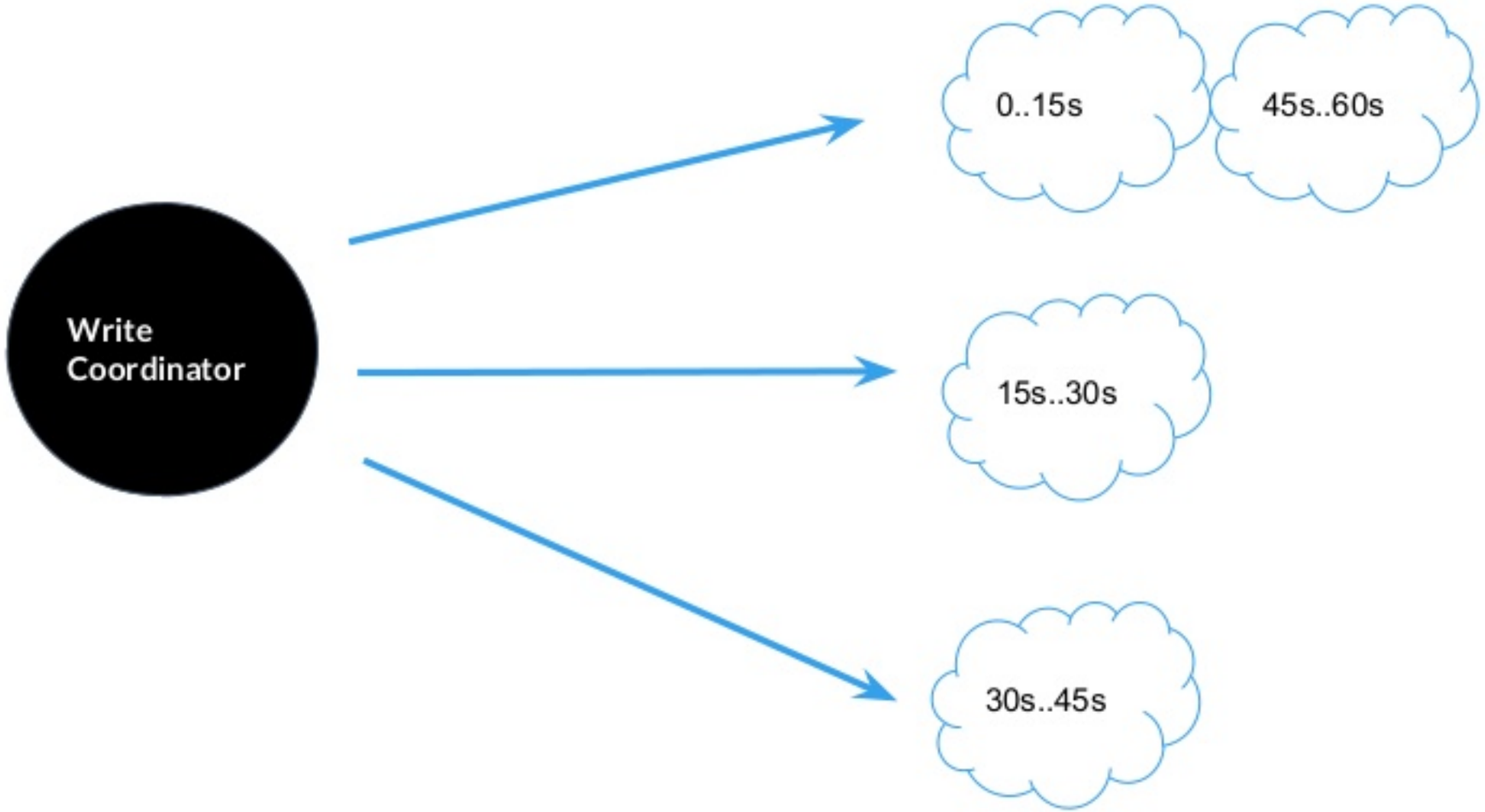
NSDb - NATURAL TIME SHARDING

```
sharding {
 ...
 interval = 15s
}
```

- A node is picked up for writing until time length reaches configured interval
- Autoscaling is supported

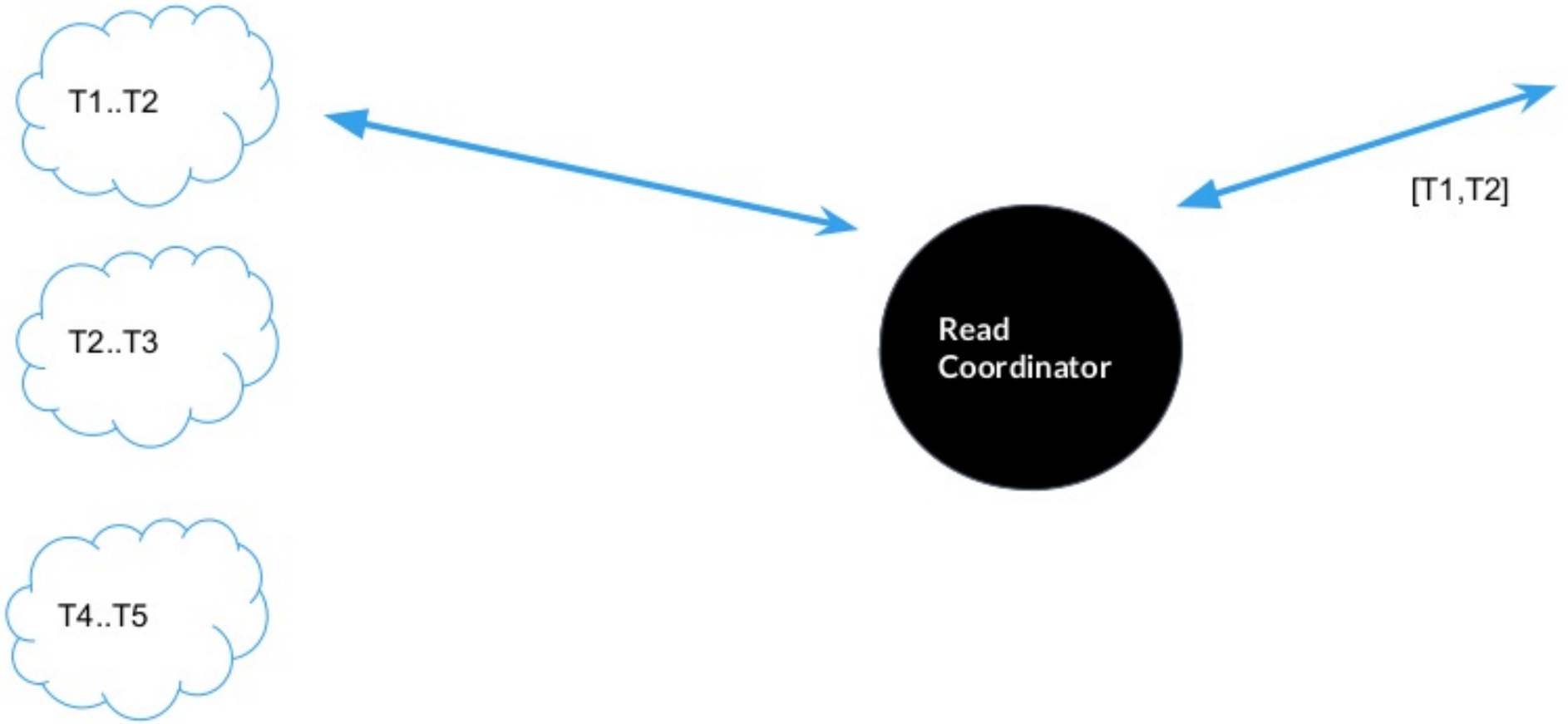
Natural Time Sharding - Data Partitioning

```
interval = 15s #nodes = 3
```



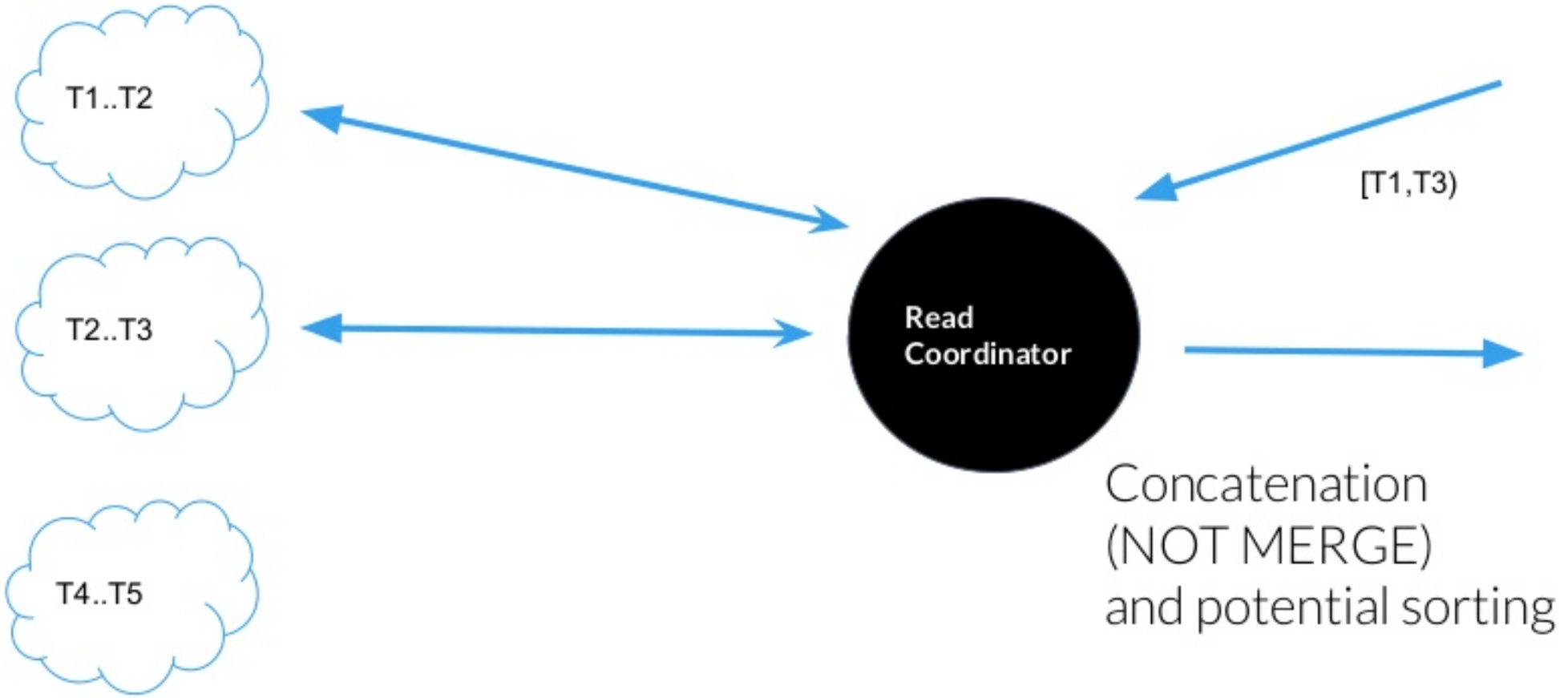
RADICALBIT
Deep. Different. Disruptive.

Natural Time Sharding - Best Case



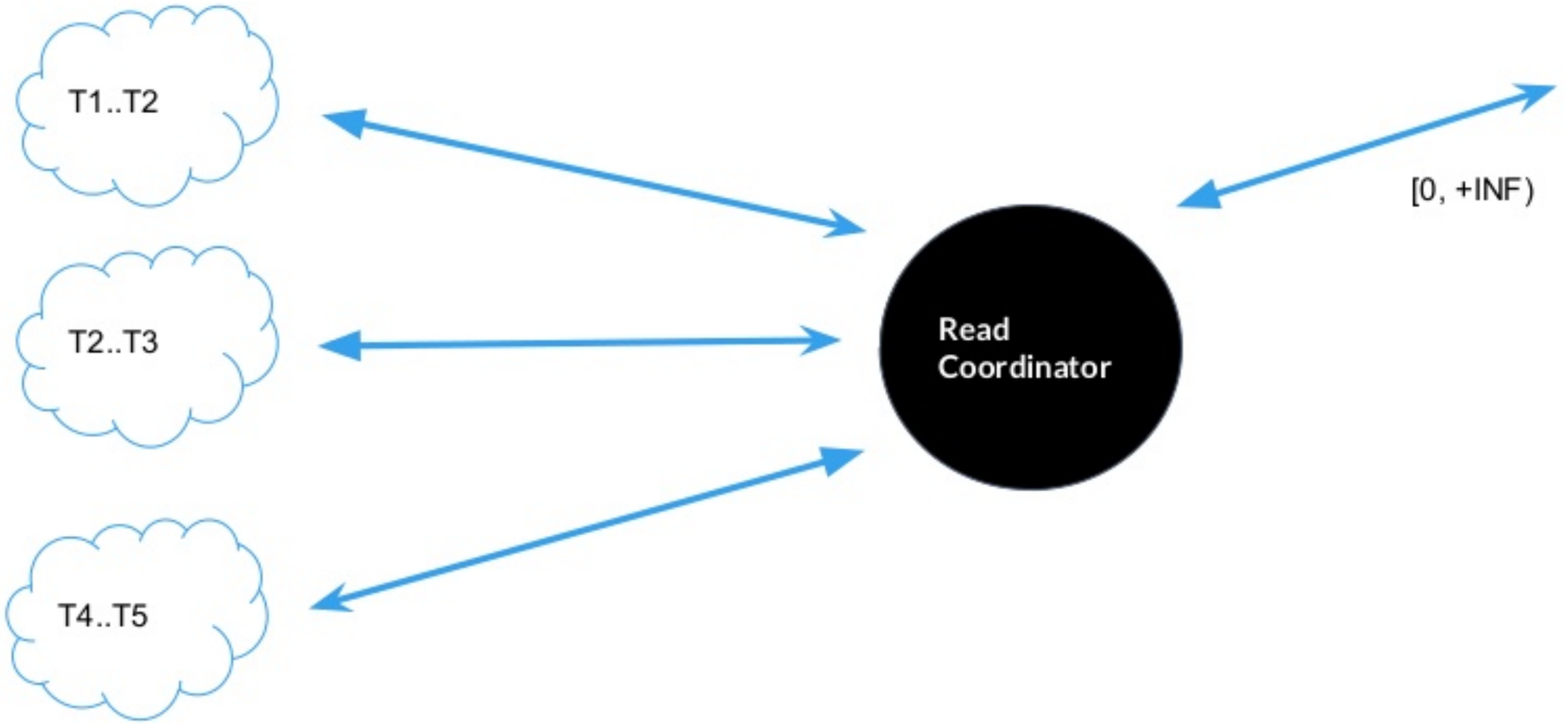
RADICALBIT
Deep, Different. Disruptive.

Natural Time Sharding - Average Case



RADICALBIT
Deep. Different. Disruptive.

Natural Time Sharding - Worst Case



RADICALBIT
Deep. Different. Disruptive.

NSDb - TIME SERIES MANAGEMENT

- **Namespace**: high level structure grouping homogeneous metrics
- **Metric**: a series of Bit (Point)
  - **Bit**: logical construct used into the API. It groups together:
    - **Timestamp**: the event time
    - **Value**: the numerical value being measured
    - **Dimensions**: a dynamic list of queryable String -> Value pairs

NSDb - SCALA API

```
val nsdb = NSDB.connect(host = "127.0.0.1", port =
7817)(ExecutionContext.global)

val series = nsdb.namespace("registry")
  .bit("order amount")
  .value(34.20)
  .dimension("city", "Berlin")
  .dimension("gender", "M")

val res: Future[RPCInsertResult] = nsdb.write(series)
res onComplete {
 case Success(r) => // do something here...
 case Failure(t) => // manage the errors here...
}
```


RADICALBIT
Deep. Different. Disruptive.

NSDb - FLINK SINK

```
val sentiment: DataStream[TweetSentiment] =
sentimentalAnalysisByHashtag.flatMap(new
ScoreText[TweetByHashtag])

val sentimentSink = new NSDBSink[TweetSentiment]
("localhost", 7817)
{
  (s: TweetSentiment) =>
    Bit(namespace = "twitter", metric = "TweetSentiment")
      .dimension("key", s.score.toString) // Positive,
Negative, Neutral
      .value(1)
}
sentiment.addSink(sentimentSink)
```


RADICALBIT
Deep. Different. Disruptive.

NSDb - DEMO

Live demo showing a stream of tweets

RADICALBIT
Deep, Different, Disruptive.

NSDb - ROADMAP

The first Release Candidate will published around November / December 2017 (Apache License v2.0) and it will offer:

- Clustering
- Java and Scala + gRPC API
- **Flink connector** (sink)
- WebSocket
- CLI

The following points will be released in the following versions:

- Integration to stream seamlessly from Pravega and Apache Kafka
- Native Apache Avro management (schema publishing)
- JDBC API thanks to Apache Calcite
- SSL and Kerberos support
- many others

RADICALBIT
Deep, Different, Disruptive.

Q&A

RADICALBIT
Deep. Different. Disruptive.

DANKE!

<radicalbit.team/>

info@radicalbit.io

RADICALBIT