Disque

A new distributed message queue

@antirez - Pivotal

Why another one?

Redis roots

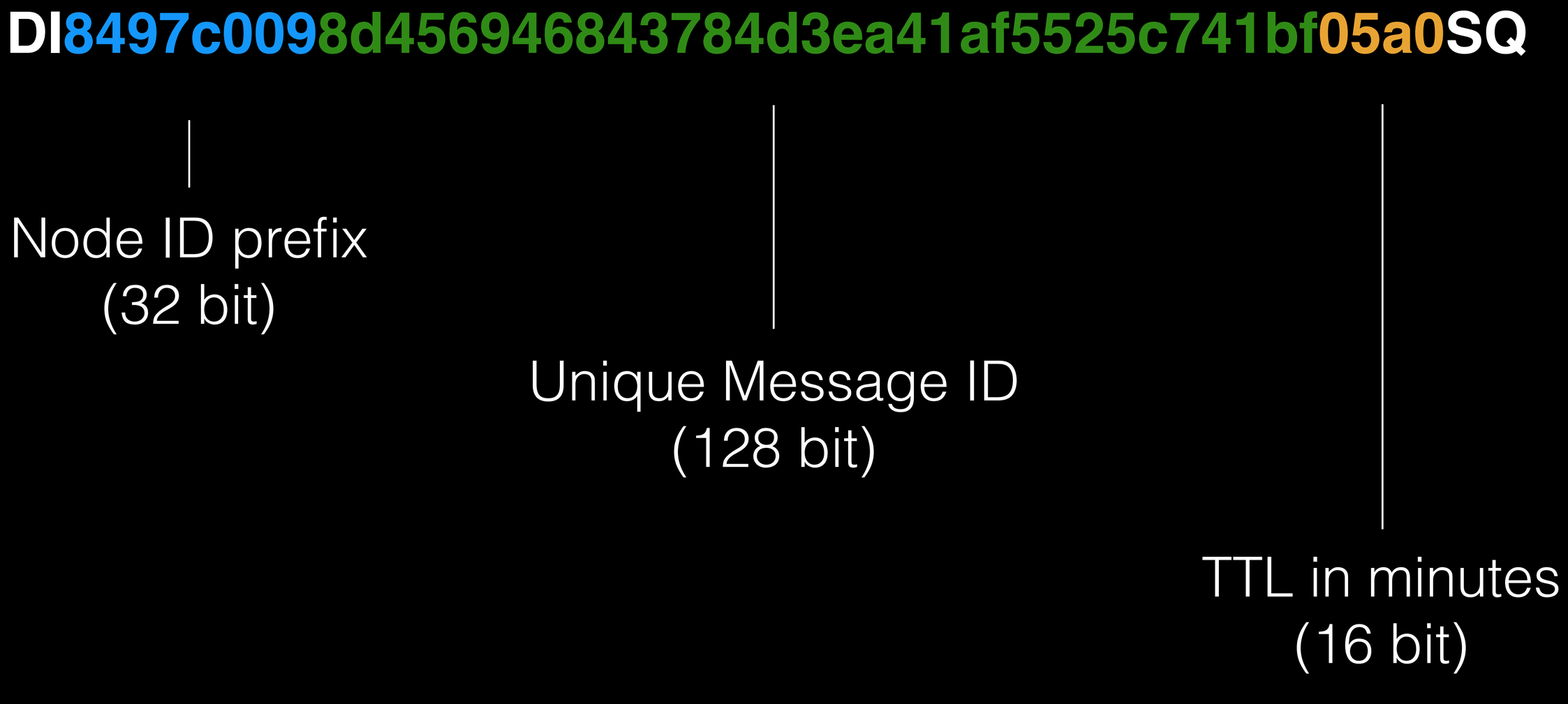
- In memory, optional persistence.
- Same protocol.
- **BSD license**.

Asynchronous jobs execution

API

ADDJOB queue job <timeout>

Disque Job IDs



GETJOB FROM q1 q2

ACKJOB id1 id2 …

Disque is all about **explicit acknowledges**.

Delivery semantics

- **At least once by default**.
- At most once also available.

ADDJOB queue job 0
RETRY 3600

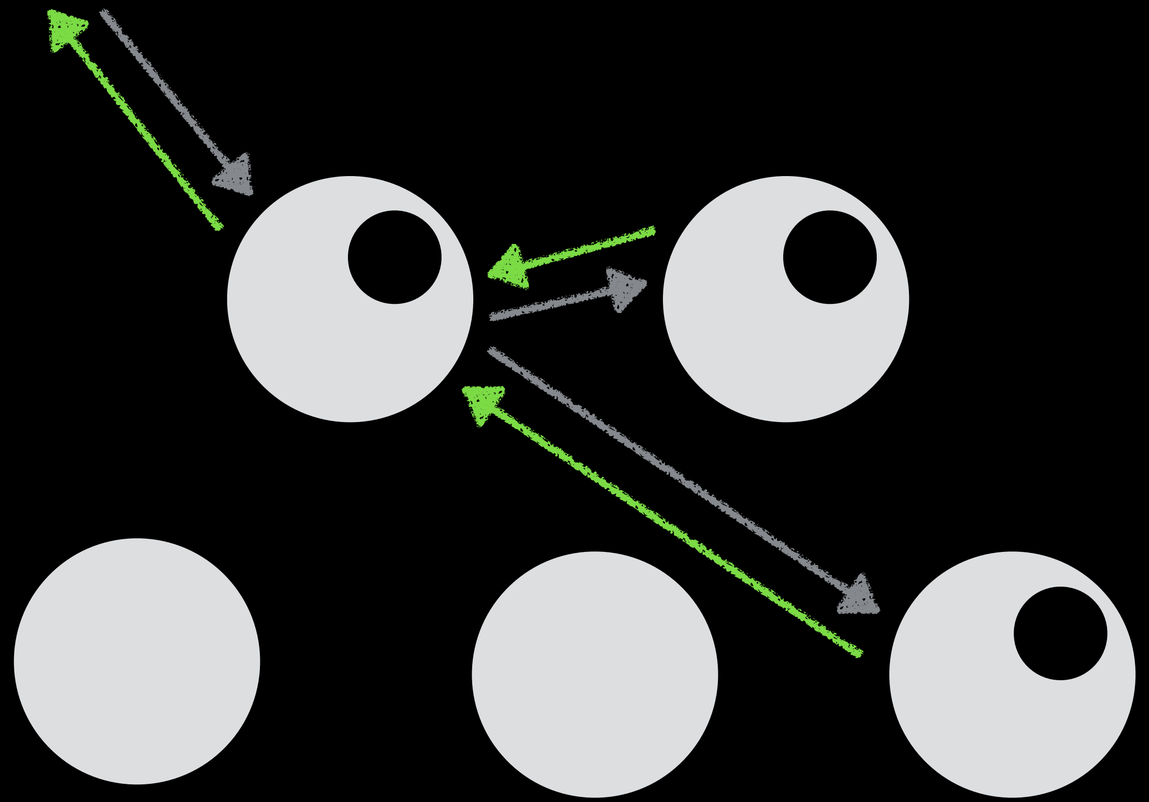
ADDJOB queue job 0

TTL 86400

ADDJOB queue job 0
DELAY 3600

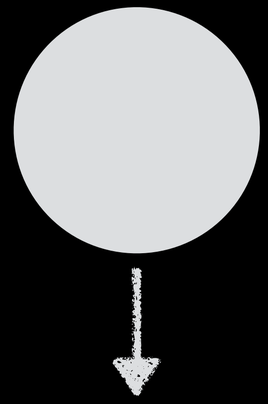
Synchronous replication

ADDJOB myqueue task1 **REPLICATE 3**



ADDJOB queue job 0
ASYNC

Optional persistence



Append Only File

LOADJOB … data …

DELJOB … id …

Replication + Persistence =

REPLICATE N means N-1 nodes can fail.

Persistence controls amnesia on restart.

Many other commands

- Introspection: **SHOW**, **QPEEK**, **QSCAN**, …
- Job management: **DEQUEUE**, **ENQUEUE**, …
- Job visibility: **WORKING** id.
- Connection handling: **HELLO**.

Disque & C**AP**

- **AP**.
- Immutable messages (mostly).
- Converge to **ACK** state.
- CAP “A” availability (single node partition).

Federation:
all nodes are **really** the same

Always available

Automatic multiple delivery

Single queue partitioning

Where is the catch?

Best effort ordering

Main Design Sacrifice

WIP features

- **Alternative for explicit dead letters**.
- Introspection & debugging.
- Simpler nodes removal.
- Stability.

github.com/antirez/disque