Why Cloudflare chose an unusual consensus algorithm, and what it teaches about design

17 November 2026 · 5 min read · blogs.iictp.org/cloudflare-quepaxa-meerkat-leaderless-consensus

Cloudflare published a post on 11 November 2026 explaining why its experimental distributed consensus service, Meerkat, uses an algorithm called QuePaxa rather than Raft, Multi-Paxos or Viewstamped Replication.

TL;DR The short version

- Cloudflare explained on 11 November 2026 why Meerkat, its experimental consensus service, uses QuePaxa rather than Raft or Multi-Paxos.
- Common algorithms depend on one leader with predictable latency, which fails Cloudflare's global control plane when a leader is slow or cut off.
- Do not copy the choice, but copy the method: write down the failure you expect and choose the design against it.

Read online, listen to it, or change the text size: https://blogs.iictp.org/cloudflare-quepaxa-meerkat-leaderless-consensus

What was reported

- Cloudflare says Meerkat is designed to manage its internal global control plane. Replicas agree on an ordered log of events, and the log exposes applications such as a key-value store, in the same family as Consul, etcd and Apache ZooKeeper.
- Cloudflare says common algorithms rely on a single leader being available with predictable latency. If the leader goes down, slows or is disconnected, it says the system slows or becomes unavailable until a new leader is in place.
- Meerkat is meant to run across continents on links with high and variable latency, which Cloudflare says is unusual for consensus systems.
- QuePaxa comes from published research (Tennage, Băsescu et al., 2023). Cloudflare says it promises better stability under volatility, in large part by not relying on a leader to make progress.

A short primer

Consensus lets a group of machines agree on one shared truth even when some crash or disconnect. The guarantee that matters is that machines never decide conflicting values, so replicated logs never diverge.

What a practitioner should do

- Do not copy this choice. Cloudflare calls QuePaxa an unusual pick for its own conditions; for most teams a mature Raft-based system such as etcd is the sound default.
- Do copy the method: write down the failure you expect (a slow or partitioned leader) and choose the design against that, not against fashion.
- Ask of any coordination dependency in your stack what happens when its leader is slow, not only when it is dead.
- Prefer well-understood, mature implementations unless a measured requirement says otherwise, and treat “experimental” in a vendor post as exactly that.

The wider lesson

Architecture decisions are expensive to undo, so the reasoning behind them matters as much as the choice. This sits in the Architecture & Systems Design domain of the TPBOK.

> Facts in this post were read from the sources listed below on 17 November 2026; this is a developing story or product and details may have changed since.

Sources and further reading

1. Cloudflare. Choosing QuePaxa, an unusual consensus algorithm. 11 November 2026. blog.cloudflare.com/choosing-quepaxa/

Published by IICTP, an independent technical credential body, on 17 November 2026. The current version of this post is at https://blogs.iictp.org/cloudflare-quepaxa-meerkat-leaderless-consensus. More guides: blogs.iictp.org.