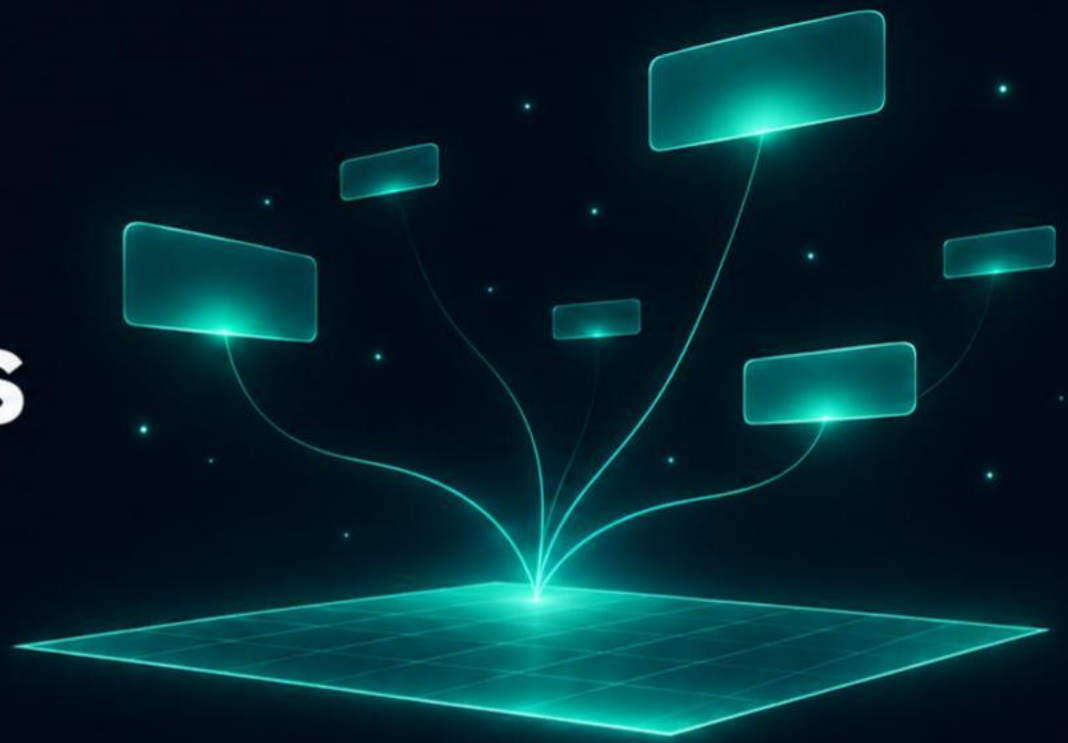
CroakOS

Engineering Impacts of a Distributed Operating System

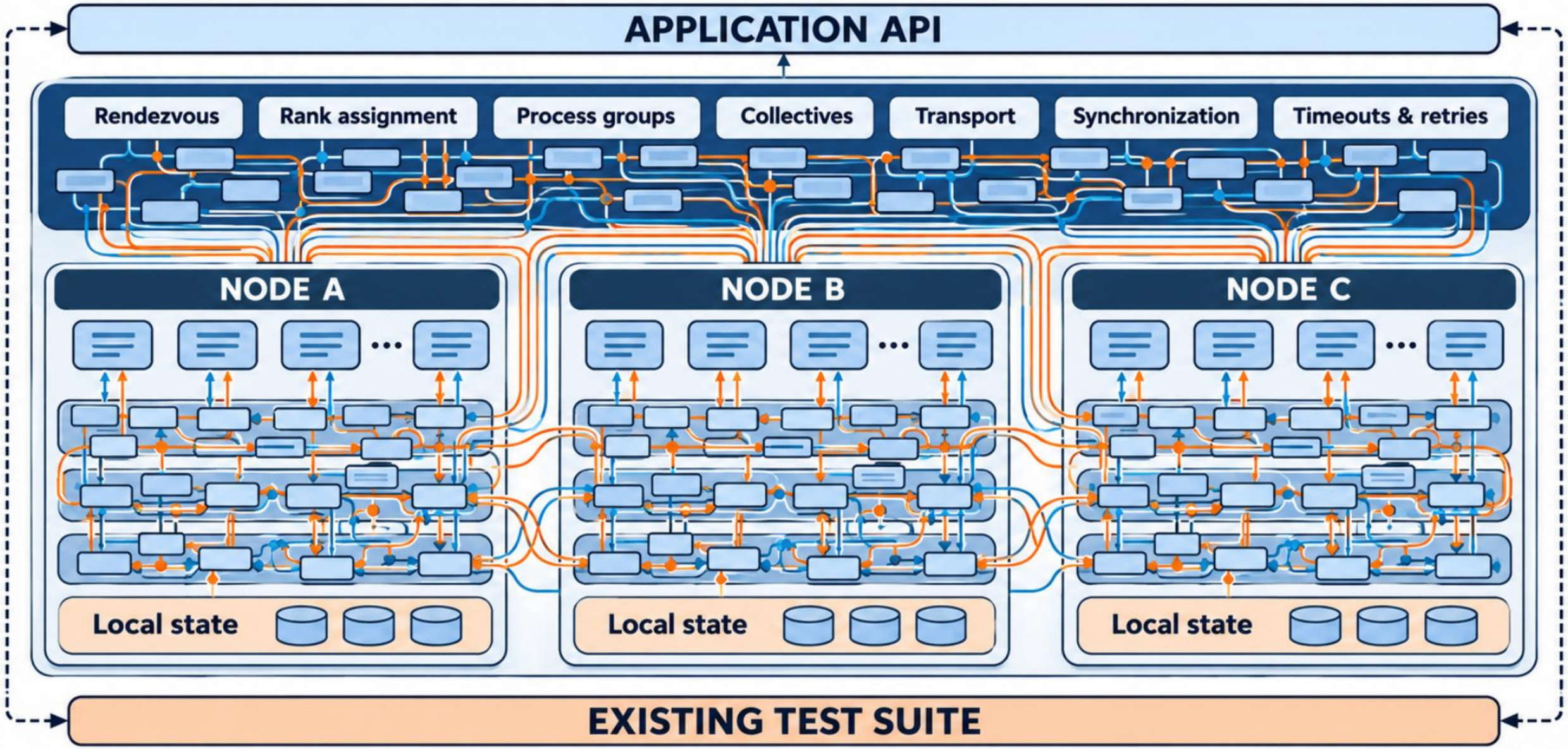


John W. Fawcett
Fawcett Innovations, LLC
john@fawcettinnovations.com

IETF 127 · HotRFC
15 November 2026

TRADITIONAL DISTRIBUTED PROGRAMMING

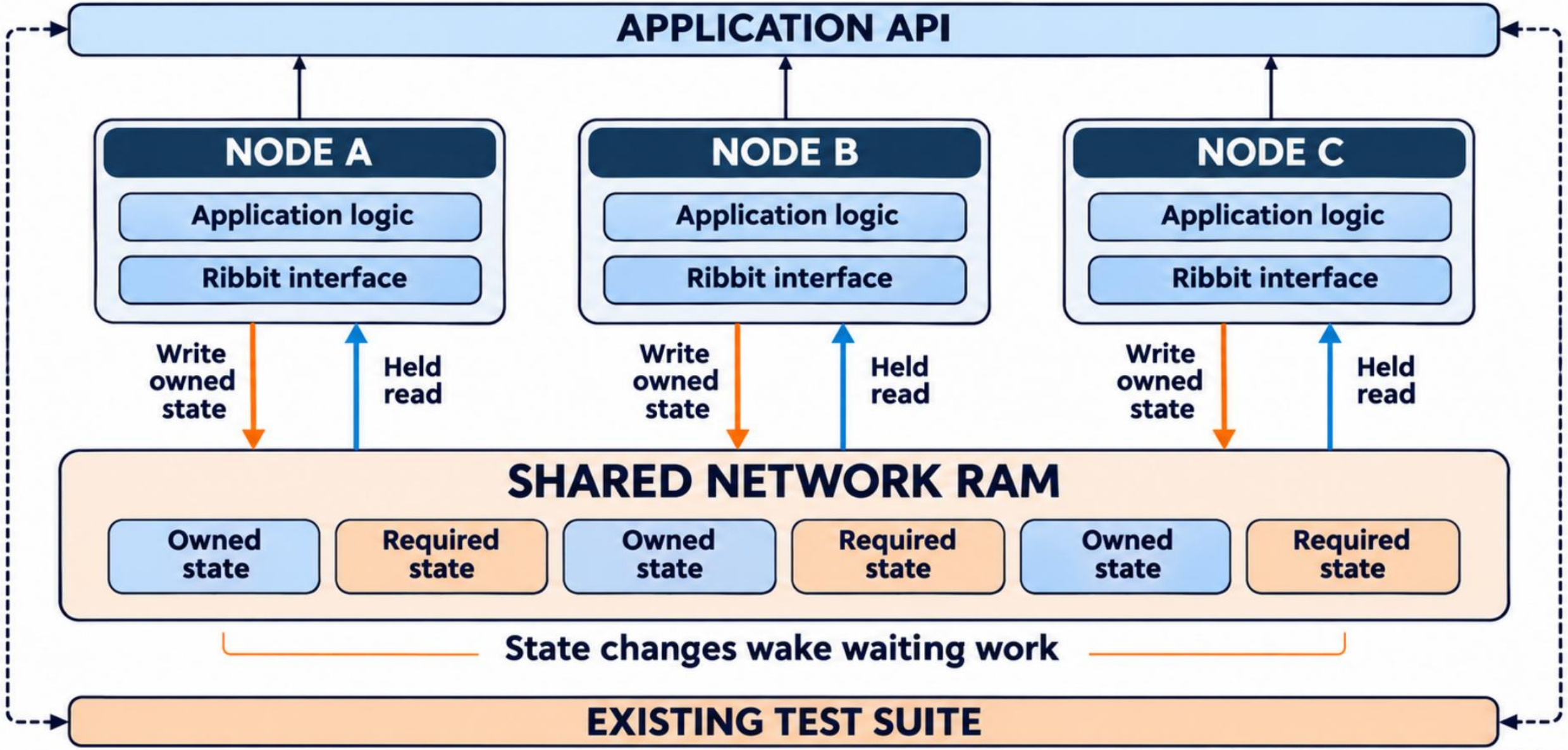
A small API. A large coordination system.



Conceptual view of distributed coordination

CROAKOS METHOD

The same API. The same tests. A Distributed Operating System.



Conceptual view of shared memory and scheduling

Simplification Through Elimination

Remove complexity and expose core algorithms

PyTorch

Distributed training on a five-node Internet mesh

SMALL BATCHES

330X

FASTER

LARGE BATCHES

5.5X

FASTER

80%

LESS CODE

50%

LESS COMPLEXITY

Distributed heterogeneous networks
Drop does not kill the run

LISPERS.NET

Map-server throughput over the Internet

LOOKUPS

17X

THROUGHPUT

REGISTRATIONS

7.9X

THROUGHPUT

68%

LESS CODE

41%

LESS COMPLEXITY

Flat CPU. Distributed map servers.
One registration database.

Simplicity Brings Speed Without Compromise

Code velocity skyrockets and we move past "Hello, World"

SYSTEM	CONVERSION TIME	RESULT
DHCPv6 failover	**2 days**	Reference implementation
Redis 7.2.11	**4 days**	10.9× fewer lines; Redis's own suite
PyTorch distributed	**5 days**	Performance parity; role-based rewrite up to 330×
rlite	**5 days**	rlite's own tests: 30 of 30
Mosquitto	**5 days**	3.7× stock at QoS 1
lispers.net	**6 days**	17× lookups, 7.9× registrations over the Internet
Meshtastic 2.8.2	**In progress**	Passes Meshtastic's own suites so far

HUMAN-DIRECTED ENGINEERING AI writes the code. Existing tests judge the result.

The Ask

Bring Your Complexity

Let’s see what happens when we write programs instead of plumbing

Seeking collaborators, partners, and adopters

John W. Fawcett, Fawcett Innovations, LLC
john@fawcettinnovations.com
github.com/FawcettJohnW/FrogNet-Living-Network

FOLLOW-UP SIDE MEETING

Tuesday 17 November
09:30–10:30 PST

Continental 7 · Webex
ietf.webex.com/meet/sidemeetings1

I would like to express my gratitude to Dino Farinacci and John Day.