



Tigermeeting

Beyond the Server:

Redefining Enterprise Meeting Management via High Watermark
Distributed Database Architecture

White Paper

Zoltan Arpadffy, CTO

Contents

Contents	2
Introduction	3
Executive Summary	3
The Limitations of Monolithic On-Premises Architectures	4
The Architectural Debt of Centralization	4
The Tigermeeting Paradigm: Serverless On-Premises	4
The Autonomous Edge Node	5
Deep Dive: The High Watermark Distributed Database	5
Data Propagation and Synchronization	5
Decentralized Conflict Resolution.....	6
The Role of the Admin "Node".....	6
Architectural Benefits for Enterprise IT	7
Extreme Fault Tolerance and Resilience.....	7
Linear, Zero-Touch Scalability	7
Network Efficiency and Security	7
Conclusion	8
Technical Appendix: System Requirements	8
Contact information.....	9
Social media	9

Introduction

Tigermeeting is the leading on-premises solution for meeting room management, digital signage and access control — fully decentralized, serverless, and built for extreme scalability.

Designed for enterprise-grade reliability, it offers a perpetual licensing model with no hidden costs or cloud dependencies.

How could we achieve this? The answer is simple: We listen to our customers.

We own the technology. We know the industry. We are passionate about what we do. We consider customer needs. We adjust our product and service roadmap accordingly. Our consistent Blue Ocean strategy and focus on the market earned us respect from both customers and competitors.

We see that our products are able to provide great and affordable service for schools, universities, offices and organizations with simple, functional, efficient and reliable meeting room management solution - that is already highly appreciated worldwide.

Executive Summary

Traditional on-premises meeting room management systems have historically relied on a monolithic, "hub-and-spoke" architecture. A central server (SQL database and application layer) acts as the single source of truth, managing state and traffic for dozens or hundreds of dummy display terminals. While offering local control, this model introduces a critical Single Point of Failure (SPOF) and significant scaling complexity.

Tigermeeting introduces a paradigm shift: a Serverless On-Premises architecture. By leveraging a proprietary High Watermark Distributed Database technology, Tigermeeting shifts the intelligence from the core to the edge. Each meeting room panel functions as an autonomous node within a self-healing mesh network.

This paper examines the technical mechanisms of this decentralized approach, detailing how it achieves superior fault tolerance, linear scalability, and operational resilience without requiring dedicated server infrastructure or external cloud dependencies.

The Limitations of Monolithic On-Premises Architectures

Before analysing the distributed model, it is necessary to define the limitations of the incumbent architecture.

Traditional on-premises solutions generally employ a three-tier architecture housed within the corporate data centre:

- Presentation Tier: "Dumb" endpoint devices (room screens) that poll for data.
- Application Tier: A Windows/Linux server running the booking logic and API handling.
- Data Tier: A central RDBMS (e.g., SQL Server, PostgreSQL) storing schedules, logs, and configurations.

The Architectural Debt of Centralization

While secure due to its on-premises nature, this topology introduces significant technical debt:

- Single Point of Failure (SPOF): A failure in the application, network or data tier (hardware crash, OS patch failure, database corruption, internet outage) results in a complete system-wide outage. Every screen in the facility goes dark simultaneously.
- Scaling Friction: Adding endpoints increases load on the central controller. Scaling requires vertical upgrades (adding CPU/RAM to the server) or complex horizontal scaling (load balancers, database clustering), increasing management overhead.
- Network Bottlenecks: All traffic must hairpin to the central server. In large campuses, simultaneous polling from hundreds of devices can create localized network congestion at the server switch port.

The Tigermeeting Paradigm: Serverless On-Premises

Tigermeeting fundamentally rearchitects this approach by eliminating the dedicated central server entirely. It is "serverless" not in the FaaS (Function-as-a-Service) sense,

but in the literal sense that it requires no permanent OS instance to manage state.

Instead, it utilizes a peer-to-peer (P2P) mesh architecture driven by a High Watermark Distributed Database located at the edge—on the room panel hardware itself.

The Autonomous Edge Node

In the Tigermeeting ecosystem, every endpoint device (e.g., a 10" Android panel outside a meeting room) is a smart node comprising:

- **Local Compute:** Sufficient processing power to handle booking logic and UI rendering independently.
- **Local Storage:** A protected partition holding a complete, encrypted replica of the necessary configuration database and relevant schedule data.
- **Network Awareness:** The ability to discover and communicate directly with peer nodes on the local subnet.

Deep Dive: The High Watermark Distributed Database

The core innovation enabling this architecture is the synchronization mechanism that maintains consistency across autonomous nodes without a central master.

Data Propagation and Synchronization

When a change event occurs on one node (e.g., a user walks up to Panel A and books a room "ad-hoc"), that node does not send a request to a server. Instead:

1. **Local Commit:** The node commits the booking to its local database fragment immediately.
2. **Broadcast:** The node broadcasts encrypted state-change metadata across the LAN segment via UDP/TCP.
3. **High Watermark Consensus:** Peer nodes receive the broadcast. They evaluate the change based on a "high watermark"—a combination of logical clocks and synchronized timestamps—to determine if this new state is the most current.
4. **Replication:** If the data is newer than their current state, peer nodes update their

local replicas accordingly.

This process ensures eventual consistency across the mesh in near real-time (milliseconds latency on a standard LAN), without hair pinning traffic through a central choke point.

Decentralized Conflict Resolution

A critical challenge in distributed systems is handling simultaneous conflicting requests (the "Double Booking Problem"). If two users attempt to book the same room at the exact same second from different interfaces, traditional systems rely on database row locking on the central server.

Tigermeeting handles this at the edge through deterministic algorithms. Because every node possesses the booking logic and a replica of the schedule:

The nodes involved in the conflict communicate instantly as peers.

Based on the precise timestamp of the request origin and predetermined tie-breaking logic, one request is validated and propagated as the winning state.

The losing request receives an immediate "room unavailable" notification at the edge device.

The Role of the Admin "Node"

If there is no server, how is the system managed?

The Tigermeeting Admin App acts as a transient peer on the network. It is not a constantly running service.

When an administrator launches the app on their laptop, it joins the mesh network, authenticates, and gains authority to push configuration changes (themes, room names, intricate settings) to the peer nodes. Once the admin closes the app, it leaves the mesh, and the ecosystem continues to run autonomously.

Architectural Benefits for Enterprise IT

For systems architects and network engineers, this distributed approach translates into tangible technical advantages.

Extreme Fault Tolerance and Resilience

The architecture inherently creates an N+1 redundancy model where N equals the total number of devices.

Limited Blast Radius: If a single room panel suffers a hardware failure, it affects only that specific room. The rest of the network is completely unaware and unaffected.

Self-Healing: When a failed node is replaced, the new device simply joins the network, discovers its peers, and automatically synchronizes the current configuration and schedule state from the mesh.

Linear, Zero-Touch Scalability

Scaling a centralized system is logarithmic; scaling a distributed system is linear. To add 50 new meeting rooms, IT simply installs 50 new panels and connects them to the VLAN. The distributed database automatically expands to accommodate the new nodes without requiring re-architecture or infrastructure upgrades at the core.

Network Efficiency and Security

East-West Traffic Profile: Nearly all synchronization traffic is localized "east-west" traffic within the LAN segment. It does not traverse the core router or firewalls unless spanning multiple subnets.

Air-Gap Readiness: Because the system requires no connection to a vendor cloud for operation, it is ideally suited for high-security, air-gapped networks often found in defense, aerospace, financial institutions and other critical infrastructure sectors.

Conclusion

The monolithic server model for facility management is an artifact of a previous IT era. It imposes unnecessary fragility and management overhead on modern enterprises.

Tigermeeting's utilization of a High Watermark Distributed Database represents the maturation of edge computing in the smart office space. By decentralizing data and logic, it provides IT leaders with an architecture that is inherently more resilient, easier to scale, and secure by design. It delivers the reliability of on-premises ownership without the burden of server management.

Technical Appendix: System Requirements

Component	Requirement	Notes
Network Topology	LAN / Private VLAN	Multicast/Broadcast capability recommended within the subnet for efficient peer discovery.
Ports	TCP/UDP (Specific ports defined in deployment guide)	Ports must be open between peer devices on the local network segments. No inbound Internet ports required.
Server OS	None Required	No Windows Server or Linux distro needed.
Database	None Required	No external SQL Server, Oracle, or MySQL instance needed.
External Access	Optional	Outbound 443 only required if syncing with cloud calendars (O365/Google Workspace) or for optional license validation.

Contact information

Email: info@tigermeeting.app

Web: <https://tigermeeting.app/en/contact>

Customer Support: support@tigermeeting.app

More information can be obtained under “About” menu in the Admin Application.



TIGERMEETING ADMIN VERSION: 3.3.3

Tigermeeting AB

A Swedish company, that brings high-end meeting management
and calendar products to affordable level.
Please, take contact with us for any inquiry.

Address: Edbovägen 47, 142 63 Stockholm, Sweden

info@tigermeeting.app | support@tigermeeting.app

[Release Notes](#) [Terms of Service](#)
[Customer Support](#) [Privacy Policy](#)
[Open Source Licenses](#) [Cookie Policy](#)

Let us shine up your meeting rooms.
Global presence with Scandinavian quality.

Social media

Follow us on social media to get event updates on product news and new releases.

LinkedIn <https://www.linkedin.com/company/tigermeeting/>

Facebook <https://www.facebook.com/tigermeeting/>

Instagram <https://www.instagram.com/tigermeeting>

Reddit <https://www.reddit.com/u/tigermeeting/>

GitHub <https://www.github.com/tigermeeting>

X(Twitter) <https://x.com/tigermeeting>

YouTube <https://youtube.com/@tigermeeting>

Threads <https://www.threads.net/@tigermeeting>

Tik Tok <https://www.tiktok.com/@tigermeeting>

Telegram <https://t.me/tigermeeting>

Pinterest <https://www.pinterest.com/tigermeetingroom/>

WhatsApp

<https://www.whatsapp.com/channel/0029VanwIDn6LwHgKMtMF90S>

Weixin / WeChat

