Delta-Based CRDT Sync API

Documentation for Geo-Distributed Real-Time Synchronization

Overview

The **Delta-Based CRDT Sync API** enables **efficient conflict-free data replication** across **geo-distributed nodes** by syncing **state deltas** of **Conflict-Free Replicated Data Types (CRDTs)**. Ideal for **low-latency systems**, **offline-first applications**, **multi-master replication**, and **eventually consistent systems**.

This API is optimized for:

- Distributed edge systems
- Real-time collaborative editors
- Peer-to-peer (P2P) messaging apps
- Multi-region databases with offline sync

High-Value Keywords:

CRDT API, delta synchronization, eventual consistency, geo-distributed database, conflict-free replication, multi-region data sync, P2P sync protocol, real-time collaborative systems, distributed systems API, edge computing API, high availability data replication

---

Key Features

- **Delta-State Synchronization**: Transfer minimal diffs instead of full state
- **Multi-Region Support**: Handle cross-AZ and cross-cloud replication
- **Conflict-Free Merging**: CRDTs automatically resolve concurrent updates
- **Secure Peer Authentication**: Built-in mutual TLS + nonce challenges
- **Offline Mode**: Queue updates for deferred merge on reconnection
- **Real-Time Metrics Endpoint**: Monitor replication lag, conflicts, and merge throughput

---

Supported CRDT Types

CRDT Type	API Identifier	Use Case Example
Grow-only Counter	gcounter	Page view counter
Last-Write-Wins	lww_register	User profile fields
Observed-Remove Set	orset	Tag management, chat members
Map of CRDTs	crdt_map	Nested objects (e.g., JSON structure)
Sequence	rga, seq_crdt	Collaborative text editing

Authentication

All endpoints require:

- **Mutual TLS Authentication**
- API Key in header: X-CRDT-API-Key: <your_key>
- Nonce challenge system to prevent replay attacks

Endpoint: POST /crdt/sync

Purpose: Sync delta state of CRDT with remote peer node.

Request

http

CopyEdit

```
POST /crdt/sync HTTP/1.1

Host: sync.api.mycrdt.net

Content-Type: application/json
```

```
X-CRDT-API-Key: abc123

{
  "type": "orset",
  "object_id": "room:4398:tags",
  "delta": {
    "adds":
    "removes":
  },
  "timestamp": "2025-06-05T12:01:22.904Z",
  "source_node": "us-west-edge-003"
}
```

Parameters

Field	Type	Description
type	string	CRDT type (gcounter, orset, etc.)
object_id	string	Unique ID for the CRDT object
delta	object	Minimal diff to apply
timestamp	string	ISO 8601 UTC timestamp
source_node	string	Logical node ID pushing the delta

Response

json

CopyEdit

```
{
  "status": "synced",
  "merged_into": "room:4398:tags",
  "conflicts_resolved": 0,
  "replicated_to": ["eu-central-edge-001", "ap-south-edge-002"]
}
```

Error Codes

Code	Message	Meaning
401	Unauthorized	Invalid API key or TLS handshake failed
409	Delta Conflict	Merge failed due to malformed delta
422	CRDT Type Not Supported	Provided type not available for syncing
500	Internal Server Error	Unexpected failure during merge

Endpoint: GET /crdt/metrics

Purpose: Fetch sync stats for observability and real-time monitoring.

http

CopyEdit

```
GET /crdt/metrics?object_id=room:4398:tags
```

Response:

json

CopyEdit

```
{
  "object_id": "room:4398:tags",
  "merge_lag_ms": 42,
  "replication_factor": 3,
  "conflicts_last_hour": 0,
  "last_synced_at": "2025-06-05T12:01:29.504Z"
}
```

Endpoint: GET /crdt/object/:object_id

Purpose: Retrieve current state of a CRDT object.

Advanced Topics

1. Delta Compression and Compaction

To prevent delta growth over time:

- Enable **automatic compaction** with POST /crdt/compact
- Use **snapshot checkpoints** every 100 ops

2. Cross-Cloud CRDT Tunneling

Supports peering between:

- AWS → GCP
- GCP → Azure
- Hybrid on-prem setups

3. Causal Consistency Layer

Every CRDT object maintains:

- **Lamport clocks**
- Optional **vector clocks** for strong causal merging

Sync Strategy Guide

Strategy	Description
Push-Pull Hybrid	Nodes both broadcast deltas and respond to pulls
Gossip-Based Sync	Uses probabilistic peer broadcast cycles
Tree-Based Relay	Useful for IoT mesh networks
Direct Peer Sync	Fastest, point-to-point delta sync

Use Case: Real-Time Document Collaboration

Imagine a global collaborative editor (like Google Docs) backed by seq_crdt.
This API:

- Allows **typing latency < 50ms** even on spotty networks
- Ensures **merge correctness** on concurrent character inserts
- Enables **offline edit queueing** and auto-sync later

Rate Limits

Plan	Requests/min	Notes
Developer	60	For test networks
Pro	1,000	Suitable for production P2P
Enterprise	10,000+	With SLA, support, custom TTL

SDKs & Integrations

- **Node.js CRDT SDK**
- **Rust Delta Merge Library**

- **Python Async Client**
- **WASM Frontend Binding** for React & Flutter apps

Support

- Security Contact: security@mycrdt.net
- Docs Help: support@mycrdt.net
- Status Page: status.mycrdt.net

SEO-Optimized Summary

The **Delta-Based CRDT Sync API** is a high-performance, geo-aware synchronization API for **conflict-free real-time replication**. It's tailored for **distributed databases**, **edge computing**, and **collaborative software**. Backed by advanced **delta-based synchronization algorithms**, it ensures **high availability**, **low latency**, and **eventual consistency** across distributed systems.

Conclusion

If you're building scalable, offline-friendly, real-time applications with **multi-region replication**, this API is the synchronization backbone you need. Designed for engineers. Documented for clarity. Powered by CRDTs.