The Long Convergence: What Became of the CRDT Idea

The Intellectual Genealogy of Conflict-Free Replicated Data Types

Intellectual Genealogies Project

August 2026 — draft v0.3

Concept by Carlos Baquero · Execution by Claude agents

Seed paper: Marc Shapiro, Nuno Preguiça, Carlos Baquero, Marek Zawirski. "Conflict-Free Replicated Data Types." *SSS 2011*, LNCS 6976, pp. 386–400. doi:10.1007/978-3-642-24550-3_29.

An intellectual genealogy follows ideas through history, not just papers through a citation network.

Abstract. In 2011, a group of four researchers proposed that replicated data can be kept consistent without any synchronization, provided the operations on it are designed around a few mathematical properties. The resulting objects were named Conflict-free Replicated Data Types, or CRDTs. Fifteen years later, the citation curve of the founding paper has still not bent downward; the acronym names technology shipped in industrial databases, JVM middleware, and collaborative editors; and the formal model behind it, Strong Eventual Consistency, appears as a section heading in product documentation. This text reconstructs what happened in between. It traces where the ideas came from, including a correctness crisis in a neighbouring community that the usual origin story omits; how the ideas branched into nine research lines; what each adopting community changed in them; which ideas faded away; and why the paper is now cited more as a landmark than as a source of theorems. Looking at the trajectory as a whole, a consistent pattern emerges. The 2011 synthesis supplied a mechanism, a way of building convergent objects. The communities that inherited it kept asking what that mechanism means, each in its own terms: formal, practical, and even political.

1 The Original Intervention

Eventual consistency makes a simple promise: replicas of the data are allowed to diverge for a while but, once updates stop, all of them will reach the same state. By the late 2000s this promise had become industrial practice. Following the CAP trade-off, large services had learned to live with divergent replicas that would be reconciled at some later time. Amazon's Dynamo had also shown that reconciliation has its own subtle failure modes: deleted items reappeared in shopping carts. The building blocks were all available (version vectors, anti-entropy, last-writer-wins), but what was missing was a general criterion, a way of knowing in advance that a given replicated object would indeed converge. The 2011 papers were quite direct about this state of affairs: "published EC approaches are ad-hoc and error-prone."

If we ask where CRDTs came from, the standard answer is that they "emerged from distributed systems research in 2011." It is a convenient summary, but on two counts the historical record tells a different story. Next, we revisit both.

Old components

Let us start with the components, which were considerably older than the assembled machine. The last-writer-wins register already appears in

1976, in an RFC by Johnson and Thomas (1976) on duplicate databases. The tombstone sets and the stability algorithm date back to the replicated logs and dictionaries of Wuu and Bernstein (1984). The delivery substrate comes from the epidemic protocols introduced by Demers et al. (1987). As for the semilattice idea, the mathematical core of the state-based approach, it was formulated in 1997 by Baquero and Moura (1997), in a technical report from the University of Minho on convergent abstract data types for mobile computing. At the time, that report went mostly unnoticed. There is no dispute over this inheritance; the 2011 papers acknowledge it in plain terms: "The foundations of CvRDTs were introduced by Baquero and Moura. We extend their work with CmRDTs and a number of new results."

A detour through collaborative editing

The second correction is less well remembered. CRDTs did not begin inside distributed systems research; they began in collaborative editing, and they began with a failure. Since the work of Ellis and Gibbs (1989), collaborative editors had been built on Operational Transformation (OT): each site executes its own operations immediately, and incoming remote operations are transformed to compensate. In 2005, Oster et al. (2005) used an automated theorem prover to check the transformation functions that had been published for strings, and found them all to be incorrect. A way out was needed, and one of the candidates was somewhat radical: give up on transforming operations, and instead design the operations so that they commute. Their 2006 system WOOT took exactly that route, with unique identifiers and tombstones, and dispensing with vector clocks. The title makes the intention clear: "Real-Time Group Editors Without Operational Transformation" (Oster et al. 2006).

The future CRDT authors were working on the same problem. Shapiro and Preguiça's Treedoc (Preguiça et al. 2009) was a replicated sequence aimed at cooperative editing, and it was for Treedoc that the term "commutative replicated data type", the first expansion of the acronym, was coined (Leţia et al. 2010). The editing groups and the CRDT authors kept in contact throughout, by citation and also through informal channels that leave no trace in citation databases: in the 2011 catalogue, two of the constructions are credited to Roh, and to Molli, Weiss and Skaf, via "private communication."

Two steps to a synthesis

The synthesis itself arrived in two steps, six months apart. In January 2011, the technical report "A comprehensive study of Convergent and Commutative Replicated Data Types" (Shapiro et al. 2011a) gathered

around twenty specifications, among them the G-Counter, PN-Counter, LWW-Register, MV-Register, 2P-Set and OR-Set, plus graphs and sequences, and settled, on the theory side, for "quiescent consistency." In July, the companion paper, later published at SSS (Shapiro et al. 2011b), kept the acronym but gave it a new expansion, "Conflict-free". It defined Strong Eventual Consistency (SEC), proved two sufficient conditions for achieving it (either the replica states form a monotonic semilattice, or concurrent operations commute under causal delivery), proved the equivalence of the two styles, and claimed "a solution to the CAP problem." Interestingly, while the acronym remained stable from the very beginning, the expansion behind it changed three times in the space of four years. "Quiescent consistency" survived for only six months; the authors themselves retired it in their next paper.

Neighbours, not descendants

Not every contemporary work belongs in this family tree, and two cases deserve mention because they sit just outside it. Roh et al. (2011) in Korea had "independently developed the Replicated Abstract Data Type concept, which is quite similar to CRDT"; the 2011 papers acknowledge the parallel invention, and the RGA construction was welcomed into the catalogue. And at SOSP, in the autumn of the same year, the COPS system (Lloyd et al. 2011) defined causal+ consistency, the combination of causal consistency with convergent conflict handling. COPS cites no paper from the CRDT cluster. It grew in parallel from common ancestors, Bayou and Dynamo, and it was left to later observers to point out the family resemblance to SEC (Viotti and Vukolić 2016). As so often in the history of science, the same problem was maturing in several places at once.

What the 2011 papers contributed was therefore not a first spark. It was a synthesis: older components, a hard lesson learned in a neighbouring community, and a mathematical criterion, finally assembled under a single name. In the sections that follow, we trace what the communities that received this synthesis went on to do with it.

2 The First Descendants

How does a research community receive a synthesis of this kind? In this case, the reception had three properties worth separating: it was fast, it was broad, and it was surprisingly selective.

Fast and broad are easy to document. Within two years, the citing literature already contained, in miniature, nearly everything that the following decade would develop at scale. Walter (Sovran et al. 2011) brought commutative sets into a transactional geo-replicated store. At

Microsoft, Burckhardt's group began treating eventually consistent objects as a programming-language problem. At Berkeley, Hellerstein's group connected the semilattices to its own logic-based work on coordination avoidance (Conway et al. 2012). Applications appeared for semantic stores, for file systems, and for trees. Even the CAP discourse, which the paper had explicitly addressed, took notice within a year: Eric Brewer's own retrospective, "CAP Twelve Years Later" (Brewer 2012), cites the work.

The selectivity only becomes visible when one reads the citing sentences instead of just counting them. We examined roughly 2,500 citation-context sentences, and the proportions are instructive. The CAP claim, which the paper lists as its first contribution, appears in about eleven of those sentences. The equivalence theorems are essentially absent. Two-thirds of the citations we could label by intent are background references, typically of the form "special-purpose types like CRDTs." In other words, the citing literature adopted the concept and left the theorems largely untouched. The paper was becoming a landmark that authors salute in passing, rather than a source from which results are taken.

The two-part cluster also produced a reception pattern of its own. During the first five years, the comprehensive technical report was cited nearly as often as the published paper. This makes sense, since what implementers needed was the catalogue rather than the theorems; Akka's documentation links the report's PDF to this day (*Akka documentation: Distributed Data* n.d.). After 2016, the citations consolidated onto the published paper, and after 2021 even the habit of citing both faded away. In the end, a single canonical reference was left standing. Figure 1 shows both movements at once: the volume that never declines, and the internal shift of the canonical reference.

3 The Branching

By the middle of the decade, the descendants can be sorted into distinct research lines. Our count arrives at nine, a number that should be read with some caution: works were assigned to branches from metadata and sampled texts, and the corresponding entry in our evidence ledger remains open. Table 1 lays out the whole structure: the ancestry, the seed cluster, the branches with their kinds of connection, and the cases that belong just outside, with each branch's measured eclipse of the seed.

Two of the lines stayed close to the founding school. The systems line built platforms. SwiftCloud, Cure and AntidoteDB composed CRDTs with causal consistency and transactions; later came the invariants, with Indigo's "explicit consistency" and the Bounded Counter, answering an admission the paper itself had made, that some guarantees require "small doses of synchronisation." The efficiency line went after the costs. Delta-

	emerged	relation	eclipse of the seed*
SOURCES OF THE SEED (1976–2009)			
LWW register — Johnson & Thomas	1976	documented	
replicated logs, tombstones — Wuu & Bernstein	1984	documented	
epidemic protocols — Demers et al.	1987	documented	
convergent ADTs, semilattices — Baquero & Moura	1997	documented	
OT correctness crisis; WOOT	2005–06	documented	
Treedoc, "commutative RDT"	2007–09	documented	
BRANCHES OF THE LINEAGE			
systems platforms — founding school; Antidote, invariants	2011→	documented	— (self-propelled)
PL specification — renamed "replicated data types"	2012→	documented	37%
collaborative editing → local-first	2013→	documented	31%
efficiency: delta, pure op-based	2014→	documented	22–31%
industrial adoption — Riak, Redis, Akka	2014→	documented	hub-attributed
verification & synthesis	2017→	documented	54%
Byzantine fusion — new trust model	2019→	documented	young line
CALM, coordination avoidance — fellow traveller; LVars offshoot	2011↔	mutual	
OT counter-line — provoked repairs, not retreat	2018–20	critique	
DORMANT — FORECAST LINES THAT NEVER GREW			
complexity-classes agenda	—	dormant	
self-stabilization bridge	—	dormant	
OUTSIDE THE LINEAGE			
RADT — Roh et al.; RGA absorbed into the catalogue	2011	parallel invention	
COPS, causal+ — kinship noted later, by others	2011	parallel invention	

Table 1: The genealogy at a glance — a view of the evidence ledger. Relations use the ledger's own vocabulary (documented descent, mutual reinforcement, critique, dormant, parallel invention), not citation links. *Eclipse of the seed: the share of the branch canon's citers that does not cite the seed cluster, where a canon was measured (bar length proportional; details in the text).

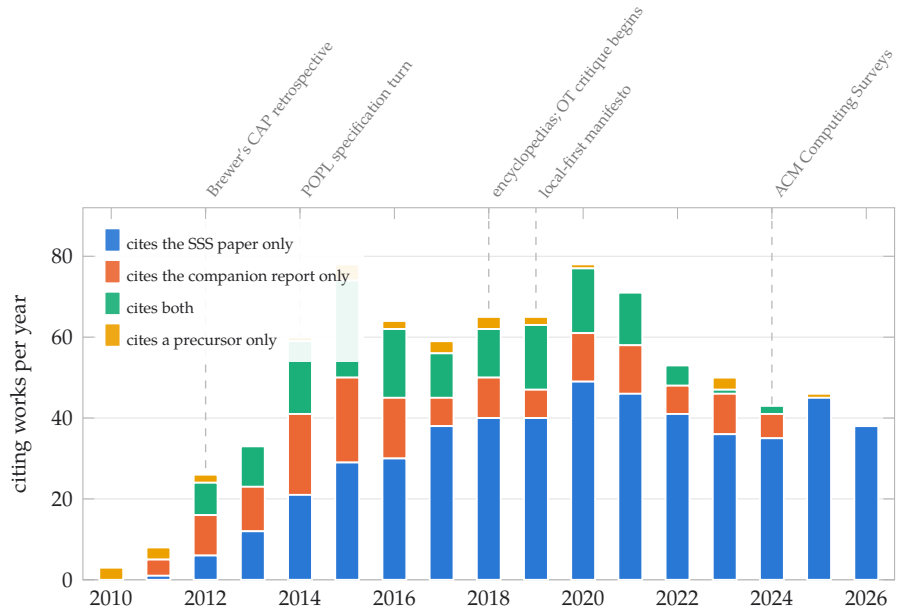

Figure 1: Citing works per year for the merged seed cluster (840 works, OpenAlex), by which cluster member a work cites. The companion technical report carries much of the early reception and fades after 2016; the habit of citing both collapses after 2021, leaving the SSS paper as the single canonical reference. Markers along the top are intellectual events from the genealogy's timeline. 2026 is a partial year.

CRDTs propagate semilattice increments rather than full states (Almeida et al. 2018), and pure operation-based CRDTs reduce the metadata to the bare operations, resorting to causal stability to garbage-collect what is no longer needed. Delta-CRDTs came from Almeida, Shoker and Baquero, the same Baquero of the 1997 report; the line that had opened the state-based idea returned, twenty years later, to repair its costs.

Three other lines built canons of their own, in other communities. The programming-languages line runs from Burckhardt's Cloud Types to the POPL 2014 paper by Burckhardt et al. (2014), the last of these a bridge author coming from the founding team. The question changed: no longer how to build a convergent object, but how to specify one, using visibility and arbitration relations, in a style reminiscent of weak memory models. The title of that paper deserves a second look: "Replicated Data Types," with no qualifier attached. The verification line starts in 2017, when Gomes et al. (2017) mechanized SEC in Isabelle/HOL. Their stated motivation could have been written by any veteran of the OT episode: "many published algorithms have later been shown to be incorrect, even some that were accompanied by supposed mechanised proofs." The line then moved into synthesis, with Katara deriving CRDTs by verified

lifting (Laddad et al. 2022), and into verification languages. And the collaborative-editing line closed a circle: the sequence CRDTs (LSEQ, Yjs (Nicolaescu et al. 2016), the JSON CRDT (Kleppmann and Beresford 2017), Automerge) returned the framework to peer-to-peer editing, the very problem from which WOOT had set out, and carried it, eventually, into mainstream software.

The remaining lines are of three different kinds. Berkeley's coordination-avoidance program, with CALM, Bloom^L and invariant confluence (Bailis et al. 2014), is best described as a fellow traveller rather than a descendant: it has a root and a logic of its own, and its exchange of citations with the CRDT line is documented in both directions. The industrial line, visible in Riak, Redis and Akka, adopted the catalogue itself, and we return to it below. The critical line, the counter-attack mounted by the OT school, will get a section of its own. And the youngest line removes one of the original assumptions: the Byzantine CRDTs, to which we also return.

4 Migration and Transformation

Every community that adopted CRDTs also changed them, and the changes all point in the same direction: away from how the objects are built, and towards what the objects mean.

Consider first the specification turn, which moved the objects from systems papers into semantics. What the 2011 papers had treated as a design discipline (prove the semilattice, prove the commutativity) the POPL 2014 framework recast as a specification problem, complete with lower bounds and optimality results. In the programming-languages literature the field is now called, simply, "replicated data types." That phrase appears more than five hundred times in our citation contexts, more often than "conflict-free" itself. The qualifier was dropped along the way; the concept no longer needed it.

The working vocabulary made a similar move. The catalogue's "Observed-Remove Set" is named after its mechanism: a removal only affects the elements it has observed. The literature increasingly prefers "add-wins set," a name for what the user obtains when an add and a remove race with each other. It is the same object under a new name, and the new name describes the outcome rather than the internals.

Industry, for its part, adopted the artifacts and dispensed with the citations. Akka implements the catalogue under the original names, `GCounter`, `PNCounter`, `ORSet` and `LWWRegister`, and links the 2011 technical report from its documentation (*Akka documentation: Distributed Data* n.d.). Redis built a product formerly named CRDB, documents it as "based on CRDT technology," and gives one of its documentation sections the title "Strong eventual consistency." Its only attribution is a link to Wikipedia (*Redis*

documentation: Active-Active geo-distributed Redis n.d.). The ideas and their names survived the trip intact; the citing conventions did not.

The largest reframing came from the editing line. The 2019 local-first essay, by Kleppmann et al. (2019), makes the case for software in which the primary copy of the data lives on the user's own device, and it names its enabling technology without hesitation: "we think CRDTs may be the foundation for collaborative software that gives users full ownership of their data" — the essay's stated analogy being packet switching. Nothing in the 2011 lists of future work, which mention complexity classes, invariants and a library, anticipates anything of the kind. A mechanism designed for convergence under network partitions was now serving as an argument about who should control the data.

The final import reversed an explicit assumption. The system model of 2011 states that processes are non-Byzantine. From 2019 onwards, the Merkle Search Trees (Auvolat and Taïani 2019), the Merkle-CRDTs of the IPFS ecosystem, and Kleppmann (2022) brought together CRDT convergence and the hash-linking of content-addressed storage, so that adversarial replicas could be tolerated. The idea of convergence came through intact; the assumptions about trust did not.

5 The Lost Citation Chain

If we follow the ideas beyond the reach of the citation indexes, two patterns emerge.

The first can be called hub attribution. Automerge's documentation states that Automerge is a CRDT and cites no paper at all; it links instead to crdt.tech (*CRDT.tech: Conflict-free Replicated Data Types — resources and community* n.d.). That site, in turn, credits the 2011 technical report and a 2018 overview, and it is maintained by Kleppmann, Bieniusa and Shapiro. Redis, as we saw, attributes to Wikipedia. Practitioner memory is thus organized in two tiers: the tools point to the hubs, and the hubs point to the founders. The pattern is familiar from everyday life, where we quote the encyclopedia and let the encyclopedia remember the sources. It is worth noticing, though, who runs the main hub here: the memory of the lineage is curated by the founding school itself. Seen from a citation database, the industrial adoption of CRDTs is almost invisible. Seen from the ground, no other community preserved the original vocabulary so faithfully, down to the exact spelling of `GCounter`.

The second pattern is the broken final hop, and LVars is the illustrative case. LVars, the lattice-based variables for deterministic parallelism proposed by Kuper and Newton (2013), place least-upper-bound joins at the centre of a programming model, and cite no member of the founding cluster. They do cite CALM and Bloom^L, which cite it. On the documented

record, then, the idea travelled from the 2011 cluster to Bloom^L, and from Bloom^L to LVars, with the last link back missing. To be fair, LVars also has independent roots of its own, in I-structures and in dataflow, and we class it as strong indirect descent with mixed ancestry. The general lesson deserves to be kept: when a single citation hop goes missing, an entire lineage becomes invisible to direct citation analysis.

We also record one negative result, as a matter of discipline. The convergent conflict handling of COPS resembles strong convergence, but COPS is not a descendant: it grew in parallel, from shared ancestors, and it was third parties who later drew the connection to SEC. Similarity is not inheritance, and a genealogy is only informative if it can turn down a plausible candidate now and then.

6 Convergence, Fusion, and Reinterpretation

The strongest reinterpretations came from rivals, one working inside the lineage and one attacking from outside.

The inside case is that of the Mergeable Replicated Data Types, presented by Kaki et al. (2019). MRDTs keep the destination, replicated types that converge on their own, but change the engine that takes us there: convergence now comes from a three-way merge over explicit version histories, not from designing operations that commute. In essence, this is the answer familiar from Git, transported to replicated types. The paper cites the founding cluster and the POPL framework, so the descent is documented; yet it is a descent that abandons the very mechanism its ancestors saw as the heart of the matter.

The outside case is the OT school answering back. Between 2018 and 2020, Sun et al. (2020) published a series of papers claiming to have "revealed facts and evidences that refute CRDT claims over OT on all accounts": the correctness advantage was not real, the complexity was understated, and the deployed co-editors, which mostly run OT, had been ignored by CRDT advocacy. The choice of venue is revealing, since the papers appeared in the CSCW space, the community where OT had been born. The outcome is revealing as well: the editing line answered with repairs rather than retreats. The interleaving anomalies on which the critique pressed hardest were analysed and fixed within the CRDT line itself, with Peritext for rich text and Fugue for interleaving (Weidner and Kleppmann 2023). In the end, the counter-attack did not stop the line; it made it better.

Underneath both episodes, the verification line was replaying the founding pattern. In 2005, a theorem prover broke the published OT functions. After 2017, it was the turn of published CRDT designs to be broken by theorem provers, including some that had arrived with

supposedly mechanized proofs, and to be rebuilt on machine-checked foundations. A field born out of one correctness crisis reached maturity by putting itself through another, this time of its own making.

7 Decline, Survival, or Canonization

Of decline, there is none to report. The merged citing corpus holds steady at fifty to eighty works per year from 2014 through 2026, and the volume of citation contexts actually peaks in 2025–26. Fifteen years after publication, the curve has not started to bend. What changed is the genre of the attention: two encyclopedia entries in 2018, one of them written by the founders; a place in the NoSQL survey canon; an ACM Computing Surveys treatment in 2024; and the product documentation already described. The concept migrated from research papers into works of reference, which is where a field stores what everyone is expected to know. It is a quieter form of success than a citation spike, and probably a more durable one.

1976–1989 The components appear: LWW, replicated logs and tombstones, epidemic propagation; OT is born for group editing.

1997 Semilattice foundations formulated for mobile computing; few notice.

2005–06 A theorem prover breaks the published OT functions; WOOT escapes by designing operations that commute.

2007–10 Gestation in cooperative editing; "commutative replicated data type" is coined.

2011 The synthesis, in two steps: the catalogue in January; "conflict-free" and SEC in July. Parallel inventions mark the boundary.

2011–14 Every later branch appears in embryo; Brewer's CAP retrospective cites the work; POPL renames the field.

2015–18 Platforms, proofs, and products; verification begins; citations consolidate on the SSS paper; encyclopedias arrive.

2018–22 Local-first reframing; the OT counter-critique; MRDTs swap the mechanism; Byzantine fusion.

2023– Normal science, running controversy, no decline.

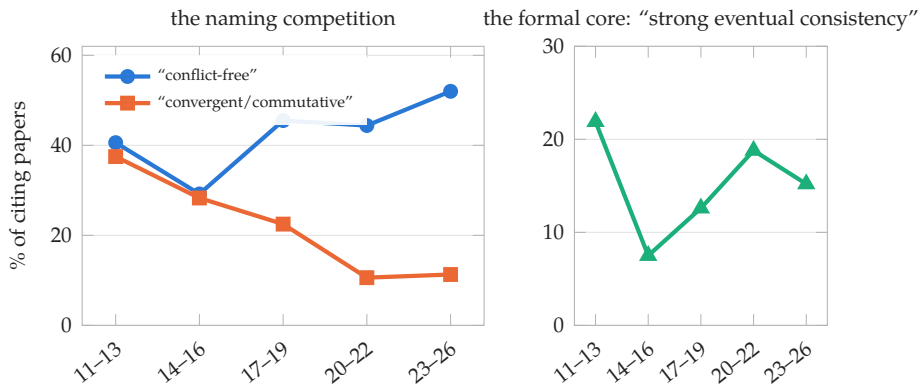

Figure 2: Terminology in the citation contexts of the SSS paper, in three-year bins (share of context-bearing citing papers whose citing sentences mention each term group; bin sizes 32–204 papers, 2026 partial). Left: the naming competition — near parity in 2011–13, five to one for “conflict-free” by 2023–26. Right: “strong eventual consistency” — strong early formal attention, a dip during the systems-adoption years, and a return with the verification era. The mechanism-to-policy rename of the set types (“observed-remove” to “add-wins”) is too rare in contexts to chart at this corpus size.

Survival, on the other hand, was quite selective, and the names mattered. The catalogue’s counters, registers, sets and sequences all made good careers; the graph types left only a thin trail. “Conflict-free” won over “convergent/commutative” in reception by roughly eight to one, though both survive technically as CvRDT and CmRDT. “Quiescent consistency” did not even survive its own cluster. The RADT name of the parallel invention died as well, while its artifact, RGA, prospers under the rival’s name. Two forecast lines never grew at all: the complexity-class agenda announced in the paper’s own future work, and the self-stabilization connection — the paper was, after all, published at a self-stabilization symposium, and the connection that its own venue seemed to invite never turned into a research line. (A note of caution is due on these dormancy claims: they are bounded by the coverage of our corpus, and absence of evidence is not evidence of absence.)

Strong Eventual Consistency had a career of its own. The formal literature adopted it as a named model, one that strengthens plain eventual consistency (Viotti and Vukolić 2016); the verification work turned it into the theorem that CRDT proofs establish (Gomes et al. 2017); and Redis prints it as a heading in its documentation. Few definitions can claim a similar path. Both stories are visible in the citation contexts themselves (Figure 2): the naming competition resolves, and the formal core dips and returns.

8 The Intellectual Legacy

What became, then, of the original contribution? It helps to consider its three parts separately, since each had a different fate.

The concept became infrastructure. The claim that replicated objects can be correct by construction, without coordination, travelled from contested proposal to the default frame in which distributed data types are now taught, specified, verified, shipped and disputed. The catalogue became a shared vocabulary, and part of it went further, into industrial identifiers. The formal core had the strangest fate of the three: rarely cited in ordinary contexts, it turned into the property that verification papers set out to prove and that the taxonomies of consistency models place on their maps. Its influence is strongest just where citation counts are blind.

If the whole genealogy has a through-line, it is the move from mechanism to meaning. The 2011 papers answered a question of how, with semilattices, commutativity and causal delivery. Each of the strong descendants then re-centred the discussion on a different what. What does a replicated type mean, formally? That was the specification turn. What should its conflict policy be called? That gave us add-wins. What is the technology for? Local-first ownership. What can be trusted? The Byzantine import. And what is the approach worth, when measured against a rival's engineering record? That question was posed by the OT counter-line.

Two sociological observations stand out. The first is that this lineage is unusually self-propelled. Within its own citing corpus, no community has been more productive than the founding school itself, which also maintains the hub through which practitioners remember the field. Part of the canonization, in other words, was carried out at home. The second is that the origin story is flatter than the origins. The sentence "CRDTs emerged in 2011" erases the 1997 semilattice report, the gestation inside the editing community, and the OT correctness crisis that forced the turn to commutativity. We should confess that our own first reconstruction made exactly the same mistake, and that it was a correction from one of the authors that fixed it. There is no better measure of how completely the flat story has won.

Did the 2011 paper cause the decade that followed? It did not. The times were rich in lattices, causal stores and editing algebras, and at least one sibling was invented independently. What the paper did was to give practitioners a criterion to design against, a catalogue of designs that meet it, and a name that let both circulate. The components had long been available; what was new was the synthesis. And this suggests a general rule for reading any famous paper: counting who cited it is only the beginning. One must follow the ideas themselves, and pay attention to

the several names under which they travel.

Provenance

Every lineage claim in this text is backed by an entry in the project's evidence ledger, 31 claims in total, of which 29 are verified and 2 remain open as synthesis at survey depth, with the evidence type and a confidence level recorded for each claim. Claims classed as "strong" rather than "documented" are worded with hedges to match. The full research notes, the data pulls, and the expert-validation record accompany this document as research provenance. Margin ledger references can be rendered by compiling with the `provenance` option.

Sources

Akka documentation: Distributed Data (n.d.). Accessed 2026-08-16. URL: https://doc.akka.io/libraries/akka-core/current/typed/distributed-data.html.

Almeida, Paulo Sérgio, Ali Shoker, and Carlos Baquero (2018). "Delta state replicated data types". In: *Journal of Parallel and Distributed Computing* 111, pp. 162–173. DOI: 10.1016/j.jpdc.2017.08.003.

Auvolat, Alex and François Taïani (2019). "Merkle Search Trees: Efficient State-Based CRDTs in Open Networks". In: *IEEE Symp. on Reliable Distributed Systems (SRDS)*, pp. 221–230. DOI: 10.1109/srds47363.2019.00032.

Bailis, Peter et al. (2014). "Coordination avoidance in database systems". In: *Proceedings of the VLDB Endowment* 8.3, pp. 185–196. DOI: 10.14778/2735508.2735509.

Baquero, Carlos and Francisco Moura (1997). *Specification of convergent abstract data types for autonomous mobile computing*. Tech. rep. Departamento de Informática, Universidade do Minho.

Brewer, Eric (2012). "CAP twelve years later: How the "rules" have changed". In: *Computer* 45.2, pp. 23–29. DOI: 10.1109/mc.2012.37.

Burckhardt, Sebastian et al. (2014). "Replicated Data Types: Specification, Verification, Optimality". In: *ACM Symp. on Principles of Programming Languages (POPL)*, pp. 271–284. DOI: 10.1145/2535838.2535848.

Conway, Neil et al. (2012). "Logic and lattices for distributed programming". In: *ACM Symp. on Cloud Computing (SoCC)*. DOI: 10.1145/2391229.2391230.

CRDT.tech: Conflict-free Replicated Data Types — resources and community (n.d.). Maintained by M. Kleppmann, A. Bieniusa, and M. Shapiro. Accessed 2026-08-16. URL: https://crdt.tech/.

Demers, Alan et al. (1987). "Epidemic algorithms for replicated database maintenance". In: *Symp. on Principles of Distributed Computing (PODC)*, pp. 1–12. DOI: 10.1145/41840.41841.

Ellis, Clarence A. and Simon J. Gibbs (1989). "Concurrency control in groupware systems". In: *ACM SIGMOD Int. Conf. on Management of Data*, pp. 399–407. DOI: 10.1145/67544.66963.

Gomes, Victor B. F. et al. (2017). "Verifying strong eventual consistency in distributed systems". In: *Proceedings of the ACM on Programming Languages* 1.OOPSLA, 109:1–109:28. DOI: 10.1145/3133933.

Johnson, Paul R. and Robert H. Thomas (1976). *The maintenance of duplicate databases*. Internet RFC 677, Information Sciences Institute.

Kaki, Gowtham et al. (2019). "Mergeable replicated data types". In: *Proceedings of the ACM on Programming Languages* 3.OOPSLA, 154:1–154:29. DOI: 10.1145/3360580.

Kleppmann, Martin (2022). "Making CRDTs Byzantine fault tolerant". In: *Workshop on Principles and Practice of Consistency for Distributed Data (PaPoC)*. DOI: 10.1145/3517209.3524042.

Kleppmann, Martin and Alastair R. Beresford (2017). "A Conflict-Free Replicated JSON Datatype". In: *IEEE Transactions on Parallel and Distributed Systems* 28.10, pp. 2733–2746. DOI: 10.1109/tpds.2017.2697382.

Kleppmann, Martin et al. (2019). "Local-first software: you own your data, in spite of the cloud". In: *ACM SIGPLAN Onward!*, pp. 154–178. DOI: 10.1145/3359591.3359737.

Kuper, Lindsey and Ryan R. Newton (2013). "LVars: lattice-based data structures for deterministic parallelism". In: *ACM SIGPLAN Workshop on Functional High-Performance Computing (FHPC)*, pp. 71–84. DOI: 10.1145/2502323.2502326.

Laddad, Shadaj et al. (2022). "Katara: synthesizing CRDTs with verified lifting". In: *Proceedings of the ACM on Programming Languages* 6.OOPSLA2. DOI: 10.1145/3563336.

Leţia, Mihai, Nuno Preguiça, and Marc Shapiro (2010). "Consistency without concurrency control in large, dynamic systems". In: *ACM SIGOPS Operating Systems Review* 44.2, pp. 29–34. DOI: 10.1145/1773912.1773921.

Lloyd, Wyatt et al. (2011). "Don't settle for eventual: scalable causal consistency for wide-area storage with COPS". In: *ACM Symp. on Operating Systems Principles (SOSP)*, pp. 401–416. DOI: 10.1145/2043556.2043593.

Nicolaescu, Petru et al. (2016). "Near Real-Time Peer-to-Peer Shared Editing on Extensible Data Types". In: *ACM Int. Conf. on Supporting Group Work (GROUP)*, pp. 39–49. DOI: 10.1145/2957276.2957310.

Oster, Gérald et al. (2005). *Proving correctness of transformation functions in collaborative editing systems*. Tech. rep. RR-5795. LORIA – INRIA Lorraine.

— (2006). "Data Consistency for P2P Collaborative Editing". In: *ACM Conf. on Computer-Supported Cooperative Work (CSCW)*, pp. 259–268. DOI: 10.1145/1180875.1180916.

Preguiça, Nuno et al. (2009). "A Commutative Replicated Data Type for Cooperative Editing". In: *Int. Conf. on Distributed Computing Systems (ICDCS)*, pp. 395–403. DOI: 10.1109/icdcs.2009.20.

Redis documentation: Active-Active geo-distributed Redis (n.d.). Accessed 2026-08-16. URL: https://redis.io/docs/latest/operate/rs/databases/active-active/.

Roh, Hyun-Gul et al. (2011). "Replicated abstract data types: Building blocks for collaborative applications". In: *Journal of Parallel and Distributed Computing* 71.3, pp. 354–368. DOI: 10.1016/j.jpdc.2010.12.006.

Shapiro, Marc et al. (2011a). *A comprehensive study of Convergent and Commutative Replicated Data Types*. Tech. rep. RR-7506. INRIA. URL: https://inria.hal.science/inria-00555588.

— (2011b). "Conflict-Free Replicated Data Types". In: *Stabilization, Safety, and Security of Distributed Systems (SSS 2011)*. Vol. 6976. LNCS, pp. 386–400. DOI: 10.1007/978-3-642-24550-3_29.

Sovran, Yair et al. (2011). "Transactional storage for geo-replicated systems". In: *ACM Symp. on Operating Systems Principles (SOSP)*, pp. 385–400. DOI: 10.1145/2043556.2043592.

Sun, David et al. (2020). "Real Differences between OT and CRDT in Correctness and Complexity for Consistency Maintenance in Co-Editors". In: *Proceedings of the ACM on Human-Computer Interaction* 4.GROUP, 21:1–21:30. DOI: 10.1145/3392825.

Viotti, Paolo and Marko Vukolić (2016). "Consistency in Non-Transactional Distributed Storage Systems". In: *ACM Computing Surveys* 49.1, 19:1–19:34. DOI: 10.1145/2926965.

Weidner, Matthew and Martin Kleppmann (2023). *The Art of the Fugue: Minimizing Interleaving in Collaborative Text Editing*. arXiv:2305.00583.

Wuu, Gene T. J. and Arthur J. Bernstein (1984). "Efficient solutions to the replicated log and dictionary problems". In: *Symp. on Principles of Distributed Computing (PODC)*, pp. 233–242.