



HISTORY, VISION, INTENTION

A database for the next 50 years.

Truth is the superset of all facts. Every event creates a fact. Facts are connected. Twenty-first Century systems already model facts as immutable records. Immutable, time-aware graph data will be the core of these systems as they evolve. XTDB empowers your business to capture the truth.

WHAT IS XTDB?

XTDB aims to be a database for the next fifty years of human progress.

As computers evolve with us, two streams of advancement repeatedly converge, creating watershed moments in the history of Computer Science.

First, historical physical limitations are reduced to insignificance or eliminated entirely.

Second, our experiential understanding of hardware and software development allows us to break through archaic software development techniques.

We believe we are at such a watershed moment in computing. XTDB is built on a collection of pillars which reify these beliefs.



PILLARS

As of this writing, XTDB has a stable version **1.20.0** release built on these principles:

Disk is cheap.

Archaic data storage systems only supported destructive updates to data because disks were once expensive. This is no longer true. Immutable data is fundamental to simple, deterministic, auditable software. This is data provenance.

Everyone deserves data sovereignty.

Data provenance demands data sovereignty. Individuals deserve privacy and this implies record eviction from an immutable system. Organizations must control their data and storage systems with open formats and open source software.

Events are not states.

Systems of record must understand the difference between an occurrence and the result of that occurrence. Event streams are a natural tool for organizations to begin capturing immutable data, but they are difficult to query because they do not serve state. A system of record must be *state-serving*.

Time is ubiquitous.

All domains and disciplines of human understanding are suffused in time. In all systems, there are (at least) two naturally-occurring timelines. (1) Time as understood by the system itself (“I recorded this fact at 2021-11-29T14:03:01”). (2) Time as understood by human beings (“I recorded this fact on 2021-11-29 about an event from 2021-11-27”). Even in systems which do not use it yet, temporal data exists and must be recorded.

Schema is evolutionary.

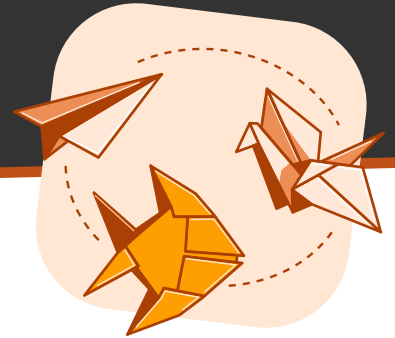
No database schema is static. It is a universal truth that the first version of a database schema is the schema built with the most limited knowledge. Historical immutable records should not be “migrated” because they accurately represent a point in time when knowledge was limited. Archaic systems destroy history by modifying it and only understand a single, contemporaneous point in time. Immutable, temporal data must support schema-on-demand because schema exists at all points in time.

Relationships from graphs.

Facts only encode meaning by value. The same value may be found in two facts. When this happens, those two facts are connected. The emergent form of collections of connected facts is always a graph.

REINFORCED PILLARS

Throughout the development of XTDB, we have been told repeatedly that these pillars hold true for many 21st Century software systems. To grow XTDB into version 2.0, we plan to reinforce our commitment by strengthening the primary pillars.



Some disks are cheaper than others.

Expanding on Pillar 1, we are engaged in R&D efforts to separate storage and compute in XTDB. If storage is perceived to be infinite, XTDB must support remote data chunks so this perception is approximately true. Once storage and compute are separated, only a central object store requires the complete set of XTDB's data. Any number of nodes can run independently, containing only the data they require individually.

Users want to query time itself.

Expanding on Pillar 4, our R&D has converged on a Timeline Index which will support queries across time. XTDB 1.x supports point-in-time bitemporal queries. XTDB 2.x will support queries which ask questions of the temporal plane itself. For example, "when did Jon, Hakan, and Steven all work at RadioShack?"

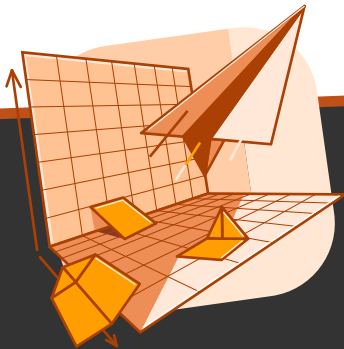
Users want first-class SQL.

XTDB 1.x supports a dialect of SQL via Apache Calcite [1]. While this provides a basic SQL interface to XTDB, we have heard from a number of people that this solution feels incomplete and they would prefer that SQL was a first-class citizen of the database. To enable first-class SQL, we are rewriting our expression engine. This rewrite is an attempt to yoke the SQL specification to the XTDB data model directly. As a happy side-effect, the modern SQL spec supports our pillars. SQL:2011 supports temporal query. SQL:2016 supports nested data and evolutionary schema. SQL:2022 (proposed) supports property graph queries. Thus, full SQL support indirectly strengthens pillars 4, 5, and 6.

WHY BUILD XTDB? WHY NOW?

Existing data stores are of two broad types. First, there are databases like Postgres and SQL Server that have evolved out of data storage and query techniques of the 70s and 80s. Second, there are new-age databases like MongoDB, Neo4j, Datomic, Rockset, Terminus, ImmuDB, etc. that address one of last century's data store inadequacies.

Neither category of data store solves the fundamental problems of 21st century data storage. Because event ledgers, immutable records, temporal data, evolutionary schema, graph queries, data sovereignty, and storage-vs-compute are deeply interlinked, a data store which solves only for one (or even a subset) leaves the application developer to solve the others manually.



EVERYTHING ALL AT ONCE? NOT QUITE.

We do not intend to build an everything-for-everyone database and, as such, XTDB will not be “onesize-fits-all”. [2] There are a number of purposes XTDB intentionally does not serve.

Although XTDB is temporal to its very core, we are not building a high-throughput streaming time series database like Clickhouse. [3] Although XTDB is intended for massive volumes of data, we are not building a realtime, horizontal-scaling OLTP database like VoltDB. [4] Although XTDB is faulttolerant, we are not tackling the problem of geo-distribution, as CockroachDB does. [5]

Some features the core of XTDB lacks may be added as extensions or separate open source products altogether, over time. However, the base of XTDB is intended to be a general-purpose HTAP database. If it helps, think of our competitors as the “boring old database” crowd: Postgres, MongoDB, Neo4j, and friends. (Note: We don't really consider closed-source databases meaningful competitors on a 50-year timescale.)

We are building XTDB to solve acute problems at the core of data storage technology. These problems cannot be solved by extending Postgres... or we would do that. Postgres is a very good database, but its underpinnings belong to another era. Our work is based on the past forty years of database research and we hope, humbly, to give back to Computer Science everything we have learned, and more.

BUSINESS PLAN

How do we plan to sustain this?

Building XTDB isn't free. JUXT Ltd has generously provided seed and Series A funding for the development of XTDB, but research, development, maintenance, and support of XTDB is a full-time endeavour. As XTDB matures, it is becoming self-sufficient. We like to know the products we use will be commercially supported and sustainable - open source is a requirement, but it isn't enough on its own. Our intention in explaining our business plan (in brief) is to extend a courtesy to our users: these are our intentions.



Future-focused domains

XTDB is a general-purpose database, but some industries tend to inherently tread on near-future territory in software development. One such industry is high finance. It is our explicit intention to target high-value trade infrastructure, risk, and financial machine learning systems. Such systems already demand immutable, auditable records with (bi-)temporal storage and queries today.

Services

Since JUXT Ltd is financing the research and development of XTDB, it only makes sense that it should be the premier consultancy to provide services and support for XTDB. Of course, XTDB will always be open source and we encourage other consultancies to build bespoke systems on top of XTDB as well.

Enterprise Licensing & Support

XTDB Ltd provides enterprise licenses to customers with significant XTDB installations. Such customers want the security of knowing they have a long-term contract to support their business.

Timeline

2018 - 2020 During this period, we built XTDB up from an alpha product to its current, production-ready 1.20 release. Throughout these years we established the need for a data product like XTDB and refined our vision through market validation.

2021 - 2023 This three-year window represents a heavy investment in R&D and the release of the 2.x Series of XTDB. We intend to grow the XTDB 1.x market share and establish a foothold in the industry such that XTDB is completely bootstrapped and self-sustaining.

Financing

For the time-being, the financing of XTDB will be provided by JUXT Ltd. By bootstrapping and establishing the market for XTDB from 2018, we have ensured the vision for a 50-year database was not accidentally compromised by a myopic profit goal. If XTDB ever receives external (outside of JUXT Ltd) investment, we will ensure that its current trajectory continues without such a compromise.

Any questions?
Contact our team for more information.



1. <https://calcite.apache.org/>
2. http://cs.brown.edu/~ugur/fits_all.pdf
3. <https://clickhouse.com/>
4. <https://www.voltodb.com/>
5. <https://www.cockroachlabs.com/>