

more power to your edge app

concordant.io



The Concordant vision: more power to your edge app

Concordant is an innovative Backend-as-a-Service (BaaS) platform that makes it easy to develop powerful edge apps.

Concordant provides consistent local-first data that is liberated from cloud barriers. It supports applications at the edge and on the move with advanced collaboration features.



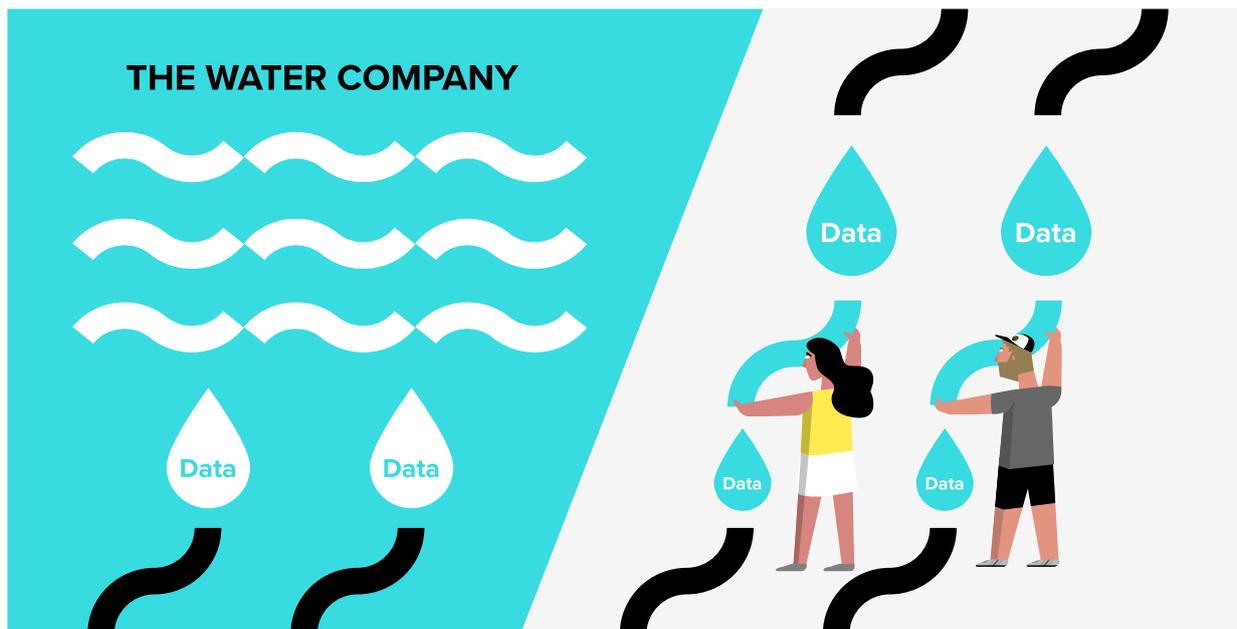
**Mobile
Backend
as a Service**

Edge apps are too complex

Edge-first apps are the future. We want to manage data, work, share, collaborate, and play with others, at home, at work, or on the move. Seamless offline work, without losing data or consistency — this should be easy. However, current edge platforms fall short, because they consider edge devices as just a conduit to the cloud.

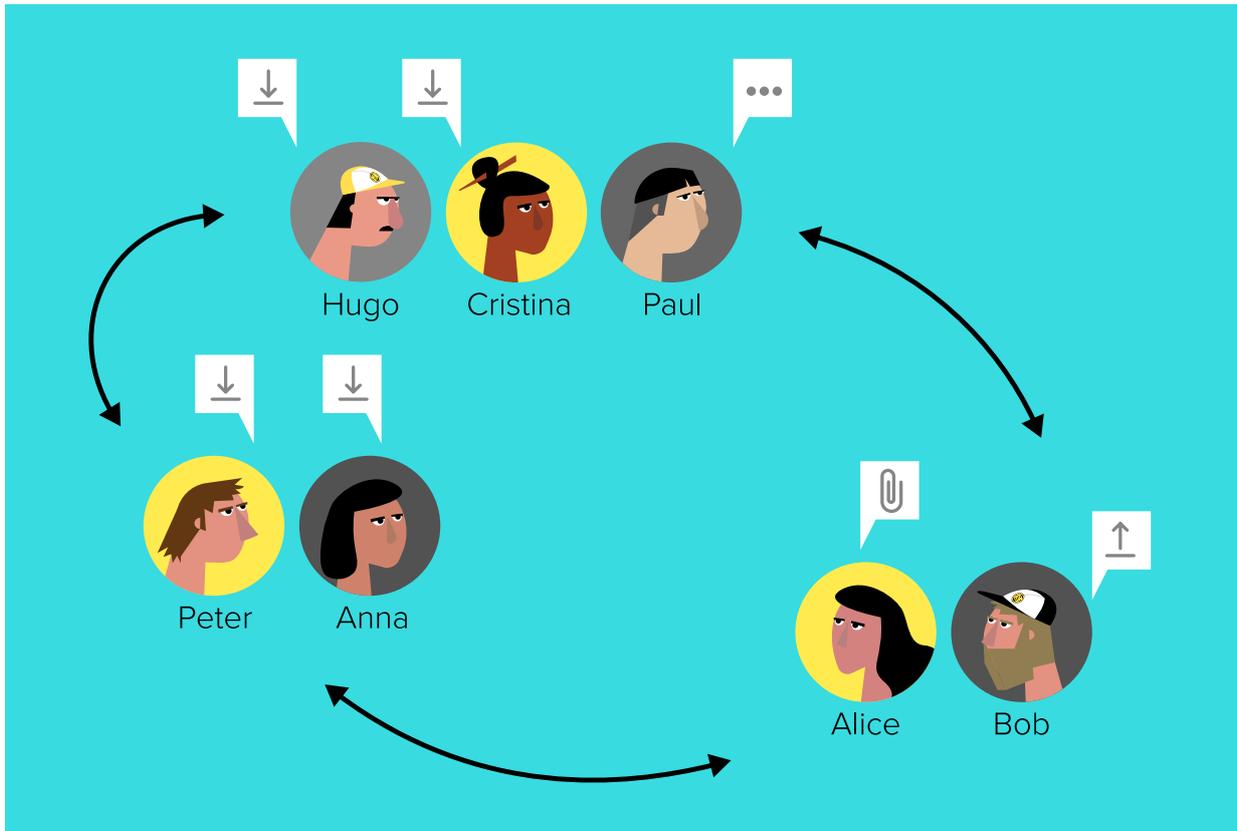
Concordant is *local-first*: by default, data and computation are located and shared at the edge; they go to the cloud only if necessary.

Consider some example applications



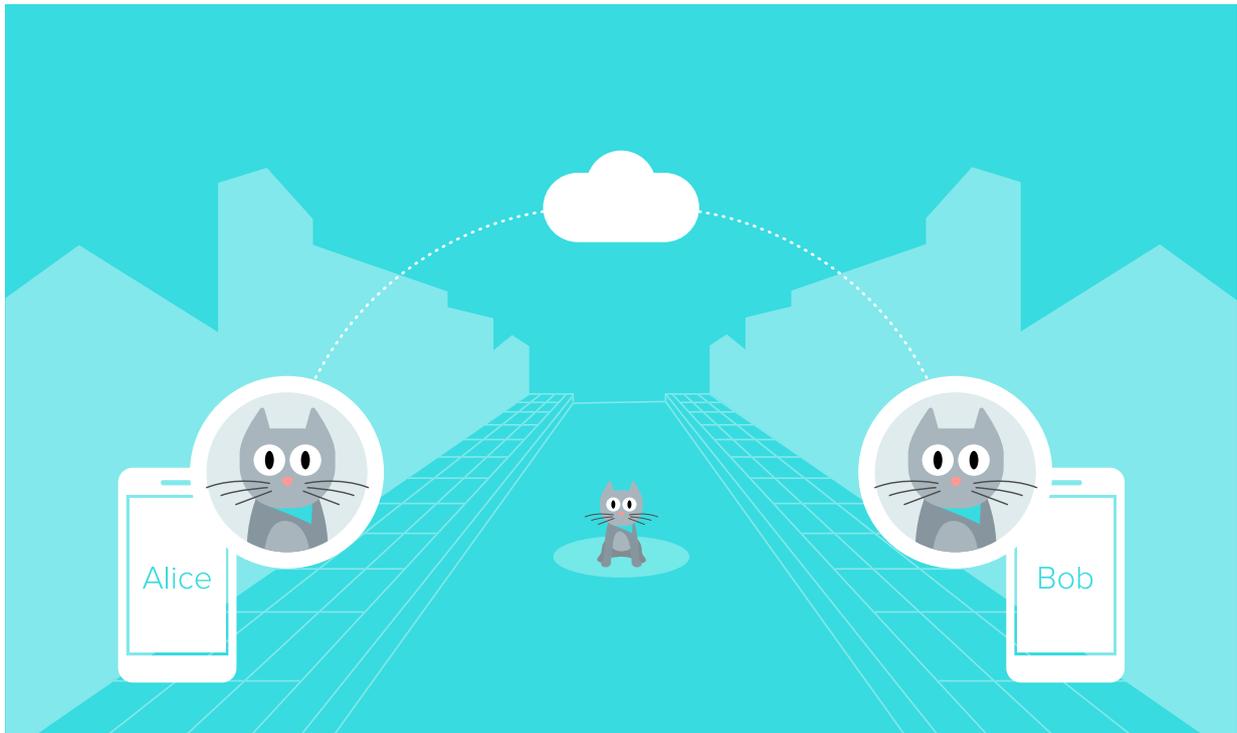
Field agents: Alice and Bob are field inspectors for The Water Company. Using their site-maintenance app, both Alice and Bob annotate and correct the blueprints of the site, update inspection history, add notes and photos, etc. Alice and Bob share this information in real time. This works even when connectivity is flaky, for instance in a basement or in a forest. Alice and Bob's relevant data is always available on their tablets; they are able to update it while offsite; and their devices synchronise directly, even when the cloud is inaccessible. Their data remains consistent, even under intermittent connectivity.

This should be easy, shouldn't it?
**But today's mainstream technology is cloud-centred,
and it doesn't help developers with this.**



Audience engagement: Alice and Bob provide group training sessions for their company's new product. To engage the audience and to encourage collaboration, Alice and Bob use polls, message boards, shared spreadsheets, and so on. Alice and Bob keep their prospects' data confidential. Furthermore, the audience shouldn't be distracted by general internet access. To this effect, they bring along a portable server with its own independent network. A participant's device connects to the local server, and shares only with the other devices in the room. Collaboration just works, and is not dependent on the cloud. It's fast, too, since everything runs locally. The collaboration apps reconcile concurrent updates, and support business-logic rules.

How hard can this be?
**With current cloud-centred technology,
very hard!**



Augmented-reality game: Alice and Bob play their favourite VR game on their mobile phones. The game augments the real landscape, overlaying it with virtual objects and pets. Alice sees a cute cat; she picks it up to adopt it; the cat is hers. But how strange: Bob can adopt the cat, too! It's as if the cat had duplicated! This could never happen in the real world. What's wrong is this: When Alice's app performs "adopt the cat," Bob's app has no way of knowing it, because they communicate only via the cloud, and only occasionally.



Shouldn't application correctness rules be enforced at the edge?
Current cloud-centred technology can't do this, forcing edge-app developers to compromise on their business-logic rules.



The trouble with BaaS

Edge systems are complex in many ways. Therefore, many app developers rely on a mobile Backend-as-a-Service (BaaS) to manage the complexity. However, cloud-based BaaS systems have a problem with data at the edge. Let's try to understand the issues.

Cloud-based systems are limiting: Mainstream edge-support systems are cloud-centric. The edge device is restricted to acting as a sensor and/or user interface; all the crucial storage and processing happen on a cloud server. Your app is captive of the cloud provider; service at the edge depends on network quality; offline service is a second-class citizen. Even close-by devices can communicate only via the cloud.

Collaborative apps at the edge need more: local-first data, edge presence, conflict resolution, access control for groups, and direct local communication. To support fast response and disconnection, the BaaS maintains a cache on the device, enabling the app to read and write locally. What happens when two devices update their cache with conflicting changes? Reconciling conflicts is hard: the common approach ("Last Writer Wins") is to pick one as the winner; the other one is just lost! Some systems support custom conflict-resolution procedures, but writing them is difficult and error-prone.

Per-object consistency is not enough: Many apps need to maintain consistency between objects, e.g., when **A** depends on **B**. Old-school databases have serialisable transactions or foreign-key, primary-key and check constraints; however, this does not scale well to the edge. Therefore, most BaaS platforms manage each object individually; for instance, reconciling conflicts per key-value pair (if at all!). This compromises the correctness of application data, and can lead to data loss.

Smart caching is needed: It is important to pre-load the cache with the necessary data, otherwise, offline operation will fail. But what is the content that should be pre-loaded? Many BaaS systems delegate this decision to the app developer; but how can an app developer know in advance what data the user will actually need in the field?



Mainstream BaaSes limit application functionality and generate poor user experience, unless developers create complicated workarounds.



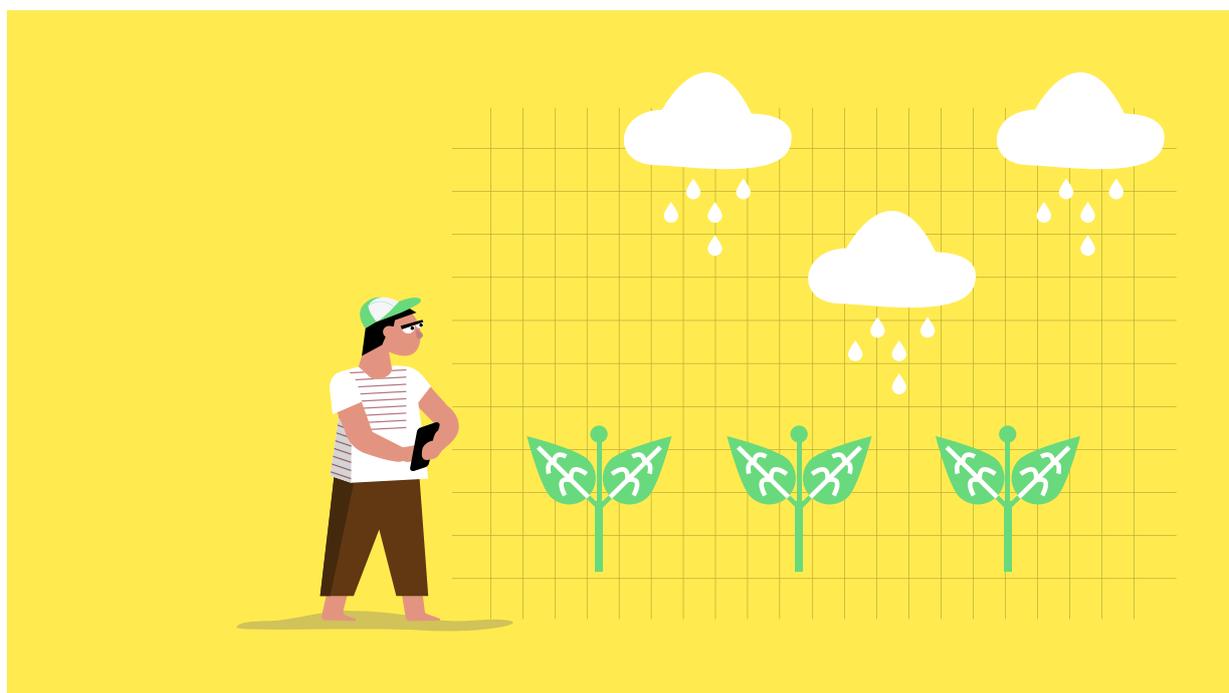
Enter Concordant:

- Concordant is edge-centric.
- It uses a *mathematically-sound and industry-tested* approach to reconciliation.
- Concordant's *highly-available transactions are observed in the correct order*, without compromising availability or response time.
- Concordant provides *predictive caching*.



Concordant: The next generation

Concordant turns the traditional BaaS design on its head. Concordant is local-first; it enjoys fast response and is always available. It seamlessly supports online and offline mode for apps, and is designed for collaboration. However, Concordant can leverage the cloud when advantageous: for stable storage, notification services, or data analytics, among others.

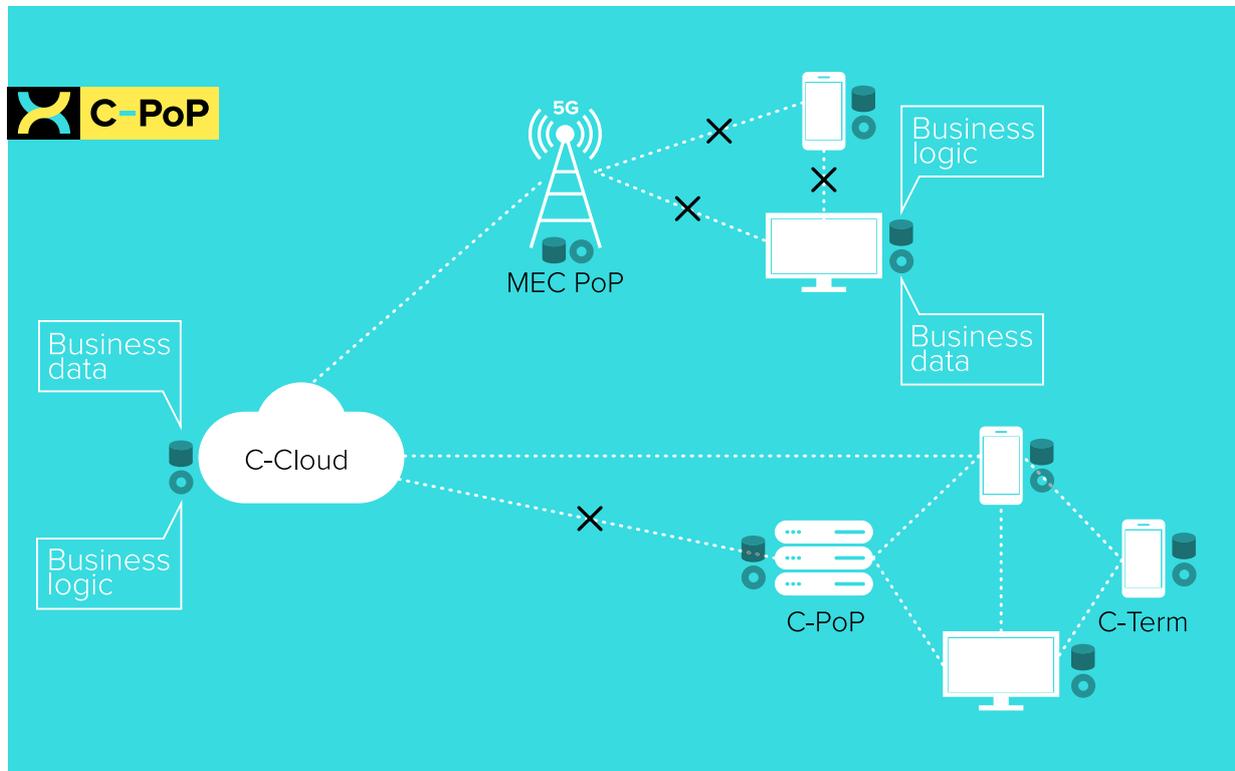


Concordant simplifies the development of new mobile applications, or enriching an existing cloud-based application with edge and mobile capabilities.

Concordant's unique features empower the developer of edge and mobile apps, while making development easier than ever. Let's take a look of what's under the hood.



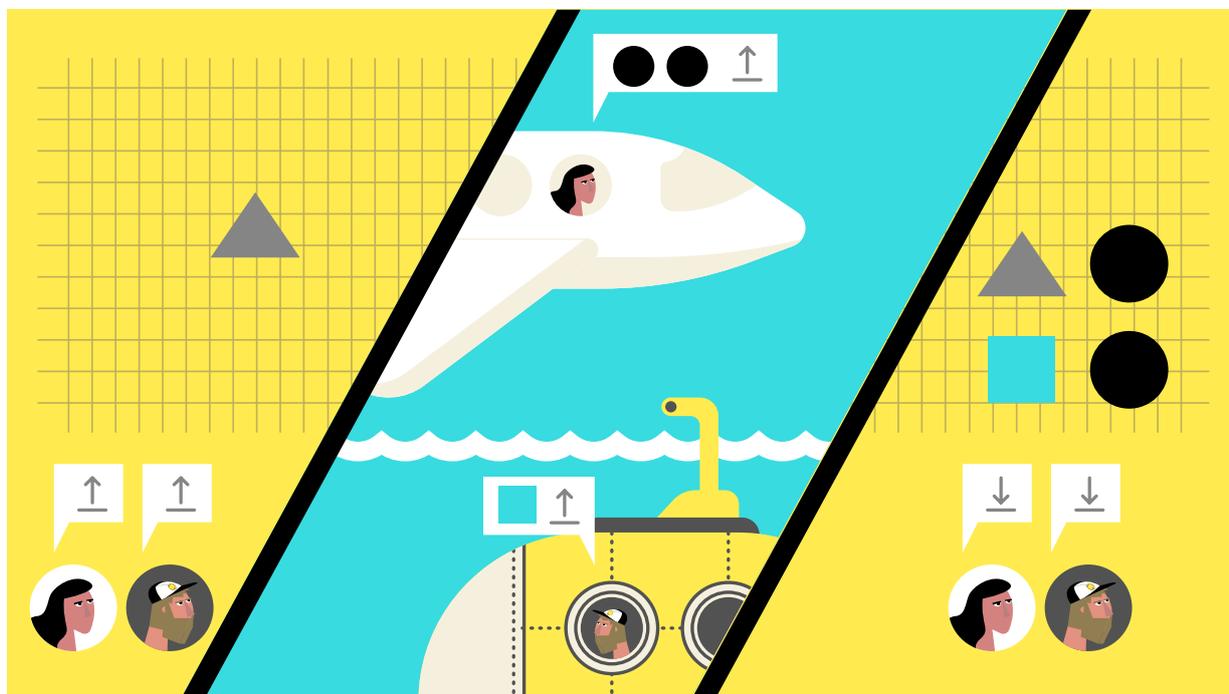
Local-first design



Thanks to its local-first design, Concordant provides a smooth experience to users. Data on the device remains available and consistent, even across device restarts, and even when connectivity is poor. The app on the device does not need to wait for a cloud server. This asynchronous data-sharing model guarantees *seamless transition between online and offline periods*. Close-by devices exchange data directly, without depending on a pre-existing infrastructure. Data is synchronised and backed up in order of priority, in order to make the best of available bandwidth, even over flaky networks.



CRDT data model



Concordant uses the CRDT data model. With CRDTs, data located on a device can always be modified, even in the presence of failures and disconnection. CRDTs implement *automatic conflict resolution*, using mathematically-proven rules. CRDTs are an industry-tested technique that has been adopted in multiple commercial products. Concordant offers an *extensive set of efficient CRDT data types*, with *resolution policies adapted to your app*. CRDTs ensure that, even when multiple users update the same data, it converges on all devices, concordant with business logic and with user expectations.

Consistency beyond CRDTs

CRDTs provide the foundation, convergence; but guaranteeing the correctness of an application may require more. Concordant provides consistency beyond CRDTs, and enforces correctness across objects, with built-in support for: *highly available transactions*, providing atomic updates while remaining available in the presence of failures and disconnection; *referential integrity*, making sure that cross-object references remain valid despite concurrent updates; *check constraints*, guaranteeing that the value of a data item remains valid at all times; and *causal consistency*, ensuring that devices observe all updates in the correct order.

Concordant does not impose a single, universal (and overkill) model of consistency, but adapts to the requirements of your app.



Smart predictive caching

Your client devices cache the data the user will need; this is key to a smooth user experience. Since data is available locally, response time is minimal; there is no need to fetch it from a server. When offline, the app continues to function normally.

Flaky networks do not impact performance. Concordant builds on advanced predictive cache pre-load techniques to reduce cache misses. It identifies patterns of data access, and uses this information predictively to pre-fetch data on client devices.

Edge presence

Concordant supports *peer-to-peer collaboration* between devices in close proximity, for instance over Bluetooth or Wi-Fi Direct. The group can work together on the move, without depending on some local infrastructure (often underpowered) to connect to the cloud. Furthermore, Concordant's edge technology can create a local point of presence (PoP) for the group on a PC or a micro-data-center, to replicate the app's database and to connect mobile devices, in particular those that lack peer-to-peer capability.

Concordant also powers *C-PoP, a portable PoP appliance* that enables users to move along with their data halo. With C-PoP, nearby devices automatically find and share data together even without a pre-existing infrastructure.

The right level of security

Concordant is *secure by design*. All network communication uses industry-standard secure channels. Furthermore, you can *select the level of data security* that is most appropriate for your usage. To protect against data theft, Concordant devices can store data in an encrypted format (*security-at-rest*). For maximum security, Concordant supports end-to-end encryption between trusted devices; untrusted cloud servers or gateways cannot decrypt the data that they store or transmit.



Freedom of choice

Concordant runs an open-source stack that is agnostic to provider. At the developer's choice, it will run on-premise, in a private cloud, or in a public one.

The app can use any of several familiar storage systems; data is not captive. Our AntidoteDB geo-replicated database supports advanced consistency features at geo-scale; alternatively, Concordant supports legacy cloud databases, with a lesser level of integration; regardless, the app enjoys offline operation, peer-to-peer collaboration and advanced caching features.



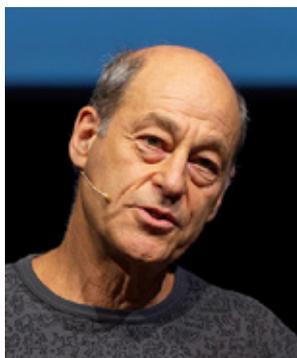
The Concordant team: inventors of CRDTs, and more

The Concordant team consists of international experts with decades of experience in distributed and edge systems.

They developed many of the Concordant technologies within the SyncFree and LightKone European projects. Some of their contributions, such as CRDTs for distributed sharing, or the Hyparview/Plumtree P2P protocols, are used successfully and widely in industry.

They ran several research projects on highly-available and scalable systems, including the highly innovative Antidote geo-distributed database.

Building on this expertise, Concordant is the platform that helps create innovative edge apps with power, speed, and availability.



Marc Shapiro
CHAIRMAN & CHIEF SCIENTIFIC OFFICER



Peter Lash
CHIEF EXECUTIVE OFFICER



Nuno Pregoça
SENIOR TECHNICAL ADVISOR



Annette Bieniusa
SENIOR TECHNICAL ADVISOR



João Leitão
SENIOR TECHNICAL ADVISOR



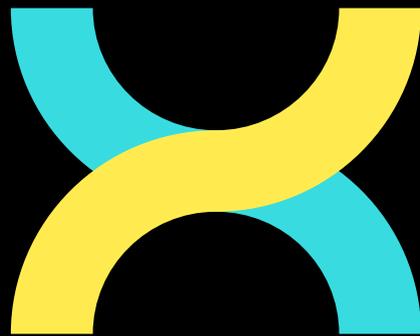
Ludovic Le Frioux
SENIOR DEVELOPER



Ilyas Toumlilt
SENIOR DEVELOPER



Florent Coriat
DEVELOPER



more power to your edge app

concordant.io