Sentry syndication layer

What problem does it solve?

Most of our date doesn't need to live in the clouds, It can do just fine down here on earth. Stored and synced directly between the devices that we use from day to day. Our civilisation is now in the "cloud era" because it's easy, convenient and profitable. But the cloud era has adverse effects on privacy and security of individuals and corporations.

Whats wrong with the cloud?

The cloud has one point of failure, usually it becomes compromised because of human negligence that eventually results in mass leaks and wind up on the dark web. It's also an easy target for surveillance and espionage from states and corporations mining our data and Intellectual property and reselling it to the highest bidder.

What makes it different?

When we store data with the user, offline and with strong encryption. There simply isn’t one point of failure anymore. It's also impossible for corporations and states to efficiently surveil the encrypted and locally stored data.

Whats the end Goal?

The goal is to facilitate for better local data communication in favour of external data (cloud) communication, if the latter is not implicitly necessary. External network activity will always be prone to leaks and mass surveillance and harms the human condition in the long run.

Why did we build it?

P2P networking is hard. To do it offline over BlueTooth is harder. Making it secure is even harder. Keeping database in sync without a central point of truth is hard too. Storing entire change history of the data adds another layer of complexity as well. The API makes these things almost as easy as using a Cloud API.

How does it work?

The API uses Bluetooth, industry standard encryption and SQL database, we then designed our own offline P2P protocol and customised the SQL database to support a distributed data topology. All encryption and security is handled by Native OS encryption API's.

What is it?

- API for offline P2P networking over bluetooth
- API for syncing databases in a distributed topology

What can it do?

- Discover peers
- Perform secure handshake
- Send encrypted payloads over 256bit E2EE
- Auto reconnect peers
- Connect up to 8 peers in one session
- Operate when devices are in standby / suspended mode
- Keep data in sync across multiple distributed databases w/o merge conflicts
- Maintain low bandwidth usage by only transferring minimum change-sets
- Store data in an encrypted SQL database (secured by your biometry or personal password)

- Full history support ensures all changes ever performed can be rolled back. And nothing is ever lost.
- Facilitate secure one off deliveries of data payloads (Quick-share)

FAQ:

- Why not use WiFi? Wifi does not support direct device communication, and does not work in standby. Important for seamless user experience
- What encryption is used? Industry standard Diffie–Hellman key exchange, ChaCha20-Poly1305, and our own layer of confirm code security when adding peers to subvert any MITM attack vectors.

Appendix:

- Diffie–Hellman key exchange: https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange
- ChaCha20-Poly1305 https://developer.apple.com/documentation/cryptokit/chachapoly