On "local–first"

Beyond the CRDT silver bullet

Part 1

Exposition

Target

- documents
- files
- personal data repositories

Not: banking services, e-commerce, social networking, ride-sharing, etc.

7 ideals for local-first software

1 – no spinners: your work at your fingertips

2 – your work is not trapped on one device

3 – the network is optional

4 – seamless collaboration with your colleagues

5 – the long now

6 – security and privacy by default

7 – you retain ultimate ownership and control

Towards a better future

CRDTs (Conflict–free Replicated Data Types)
as a Foundational Technology

Use case

```
# in node A and node B
s = "Hello, World"

# in node A
s = "Hello, Alice"

# in node B
s = "Hello, Bob"
```

How to reconcile those?

- Hello, ABloibce
- Hello, AliceBob
- Hello, BobAlice
- Hello, Alice

...

Existing CRDTs differ

- performance
- storage
- compression
- metadata overhead

Hint towards the "automerge" CRDT

show comparison table, page 9

Part 2

Critique

Software license

"In our opinion, maintaining control and ownership of data does not mean that the software must necessarily be open source."

Example 1 – intentional restriction

```
#!/bin/sh
TODAY="$(date +%s)"
LICENSE_EXPIRATION="$(date -d 2020-10-27 +%s)"
if [ "$TODAY" -ge "$LICENSE_EXPIRATION" ]; then
 echo 'License expired!'
 exit 1
fi
echo $((2 + 2))
```

```
# today
$ ./useful-adder.sh
4
# tomorrow
$ ./useful-adder.sh
License expired!
```

Example 2 – unintentional restriction

```
# today
$ useful-program
# ... useful output ...

# tomorrow, with more data
$ useful-program
ERROR: Panic!  Stack overflow!
```

local–first *requires* free software

Otherwise "The Long Now" (ideal n°5) is lost

Denial of existing solutions

"In principle it is possible to collaborate without a repository service, e.g. by sending patch files by email, but the majority of Git users rely on GitHub."

`Solution: either GitHub+CRDTs or git send-email`

Plain–text formats

"Git is highly optimized for code and similar line–based text file"

```
It even pulls software to the plain text
directtion, e.g.:
- delivery-templates
- common-core.protocols.config

Why not exploit that more?
```

Ditching of web applications

"The architecture of web apps remains fundamentally server−centric

`Disagree.  Contrast PouchDB with Android Instant Apps`

Costs are underrated

- storage
- backups
- maintenance

Example: blog vs vlog

Real–time collaboration a bit overrated

It is only possible on the presence of reliable, medium-quality network connection

"X also works when inside an elevator, subway or plane!"

On CRDTs and developer experience

"For an app developer, how does the use of a CRDT−based data layer compare to existing storage layers like a SQL database, a filesystem, or CoreData? Is a distributed system harder to write software for?

YES.

`See "A Note on Distributed Computing"`

Conclusion

Why this is a "paper I love": it took offline-first and ran with it.

But a pinch of CRDT won’t make the world local-first.

The tricky part is the end of the sentence: "in spite of the Cloud".

References

1. "Local-First Software: You Own Your Data, in spite of the Cloud", by M. Kleppmann, A. Wiggins, P. Van Hardenberg and M. F. McGranaghan
2. The Morning Paper article
3. "A Note on Distributed Compiting", by J. Waldo, G. Wyant, A. Wollrath and S. Kendall
4. these slides: euandre.org/slide/
5. prose version of this presentation
6. view source