GLYCOS

an extensible, resilient and private peer-to-peer online social network

Ruben De Smet Ann Dooms An Braeken Jo Pierson

ToxCon 2019

VUB VRIJE UNIVERSITEIT BRUSSEL

WHOAMI

```
$ whoami
rsmet
```

WHOAMI

```
$ whoami
rsmet
```

Ruben De Smet

- PhD topic: privacy engineering
  - for decentralised social media
  - for Internet of Things
- MSc topic: peer-to-peer social network

CLONING FACEBOOK

Components:

CLONING FACEBOOK

Components:

- web server stack (nginx, Apache, …)

CLONING FACEBOOK

Components:

- web server stack (nginx, Apache, …)
- relational and/or NoSQL database (PostgreSQL, MySQL, …, MongoDB, …)

CLONING FACEBOOK

Components:

- web server stack (nginx, Apache, …)
- relational and/or NoSQL database (PostgreSQL, MySQL, …, MongoDB, …)
- back-end stack (PHP with Kohana, Ruby on Rails, …)

CLONING FACEBOOK

Components:

- web server stack (nginx, Apache, …)
- relational and/or NoSQL database (PostgreSQL, MySQL, …, MongoDB, …)
- back-end stack (PHP with Kohana, Ruby on Rails, …)
- front-end stack (React, Angular, …)

CLONING FACEBOOK

Components:

- web server stack (nginx, Apache, …)
- relational and/or NoSQL database (PostgreSQL, MySQL, …, MongoDB, …)
- back-end stack (PHP with Kohana, Ruby on Rails, …)
- front-end stack (React, Angular, …)

Let’s make it private.

CLONING FACEBOOK

Components:

- web server stack (nginx, Apache, …)
- relational and/or NoSQL database (PostgreSQL, MySQL, …, MongoDB, …)
- back-end stack (PHP with Kohana, Ruby on Rails, …)
- front-end stack (React, Angular, …)

Let's make it private.

- Give users a great ToS

CLONING FACEBOOK

Components:

- web server stack (nginx, Apache, …)
- relational and/or NoSQL database (PostgreSQL, MySQL, …, MongoDB, …)
- back-end stack (PHP with Kohana, Ruby on Rails, …)
- front-end stack (React, Angular, …)

Let's make it private.

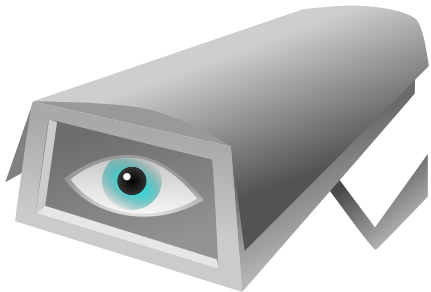
- Give users a great ToS; OR
- give users control over their data: decentralisation.

cloning Facebook: private version

VRIJE
UNIVERSITEIT
BRUSSEL

GOAL

private

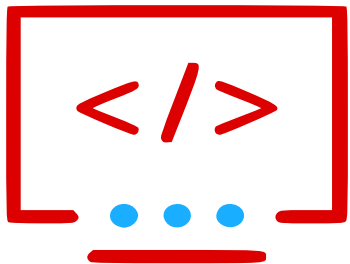


GOAL

performant

GOAL

extensible



STATE-OF-THE-ART

federated Mastodon, Diaspora*;

STATE-OF-THE-ART

federated Mastodon, Diaspora*;

peer-to-peer overlay (PeerSoN, Buchegger et al., 2009), friend-to-friend (RetroShare).

PEER-TO-PEER

FRIEND-TO-FRIEND: "RETROSHARE"

friend-to-friend (f2f), 100 % decentralised
Since October 2017: GXS, “Generic data eXchange System” (Soler, 2017):

Services defines groups
Groups a structured collection of messages
Messages hierarchical data items belonging to a group
Identities an “account”, user identification
Circles a set of identities

introducing glycos

VRIJE
UNIVERSITEIT
BRUSSEL

GOAL

- developer convenience: abstractions, tools; AND
- generality of data model; AND
- guarantees on privacy

GOAL

- developer convenience: abstractions, tools; AND
- generality of data model; AND
- guarantees on privacy

additionally: performance ⇒ mobile friendliness.

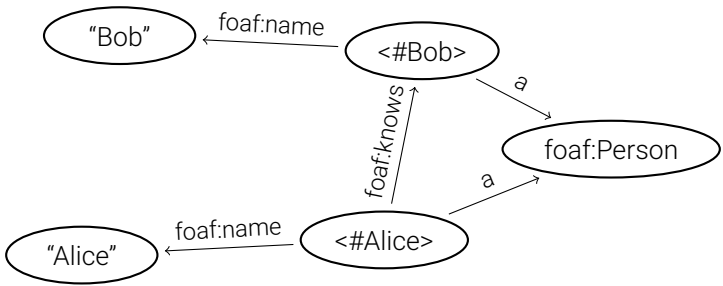


Figure: An example RDF graph. This graph signifies that Alice, a person, knows Bob, another person. Both persons have a name and type.

GRAPH DATABASES (CONT.)

Graph databases are well studied (Angles & Gutierrez, 2008; Lanthaler, Cyganiak, & Wood, 2014; Lassila & Swick, 1997).

They can represent arbitrary structured data.

Glycos couples RDF graphs with modern cryptography to provide access control and anonymity w.r.t. outsiders.

EXAMPLE

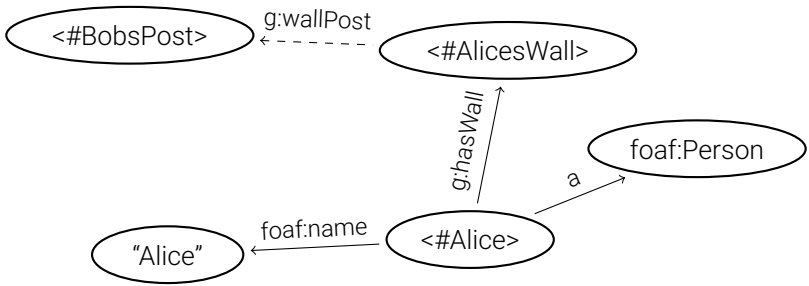


Figure: Bob writes a message on Alice's wall. This is only possible if Alice has granted Bob the rights to do so; otherwise, the network will not accept Bobs post (`<#BobsPost>`). The definition of those access rights are contained within every vertex.

On the `<#AlicesWall>` vertex, Alice has defined who are allowed to append other vertices: an access control list.

PSEUDONYMISATION

Bob generates an ephemeral (one-time) public key for Alice (with $A = aG$) as follows (van Saberhagen, 2013):

$$
\begin{aligned}
r &\leftarrow [0, \ell - 1] \\
R &\leftarrow rG, \\
pk_{\mathrm{OT}}^{\mathrm{alice}} &\leftarrow \mathcal{H}_s(rA)G + A, \\
sk_{\mathrm{OT}}^{\mathrm{alice}} &\leftarrow \mathcal{H}_s(aR) + a.
\end{aligned}
$$

EDGES AND VERTICES

Vertices are stored with their edges on a DHT.

Vertex

- Identified by a random owner key;
- Contains a pseudonymised ACL.

EDGES AND VERTICES

Vertices are stored with their edges on a DHT.

Vertex

- Identified by a random owner key;
- Contains a pseudonymised ACL.

Edge

- Encrypted “predicate” and “object”;
- ring signature over the subject’s ACL.

RING SIGNATURES

Ring signatures (Rivest, Shamir, & Tauman, 2001) prove knowledge of one key in a set *R*.

RING SIGNATURES

Ring signatures (Rivest et al., 2001) prove knowledge of one key in a set *R*.

Provides unlinkability: two edges from the same author are undistinguishable from two edges from different authors.

IMPLICATIONS AND SUMMARY

Graph databases are well studied, manipulation is **easy**, and they are generic. Tooling can be provided for developers, (roughly) same abstraction level as web development.

Basic implementation written in Rust, tested cross-platform (Intel/ARM), including networking, cryptography, very basic API.

https://gitlab.com/glycos/glycos

RESULTS

Benchmarks[1]:

	lower bound	median	upper bound
decrypt vertex	3198 s^{-1}	3341 s^{-1}	3493 s^{-1}
edge verification	108 µs	113 µs	118 µs

[1] Intel Xeon, single threaded, 95 % confidence interval

c&c, q&a

mailto:rubedesm@vub.ac.be @rubdos

Slides at https://rubdos.gitlab.io/papers/toxcon-2019.pdf