Whack WAC

Decentralised, provably secure and consistent Web Access Control

Martel Innovate

Vision

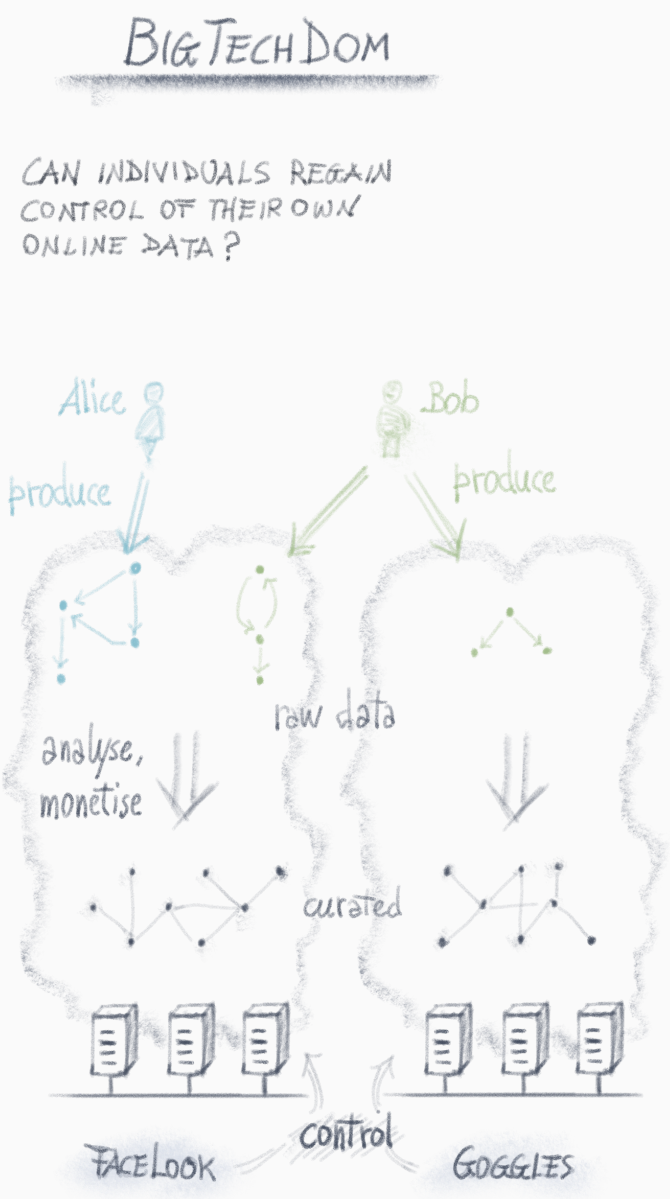
A Shared Vision

TrustChain and Martel share a common vision for a Web that is

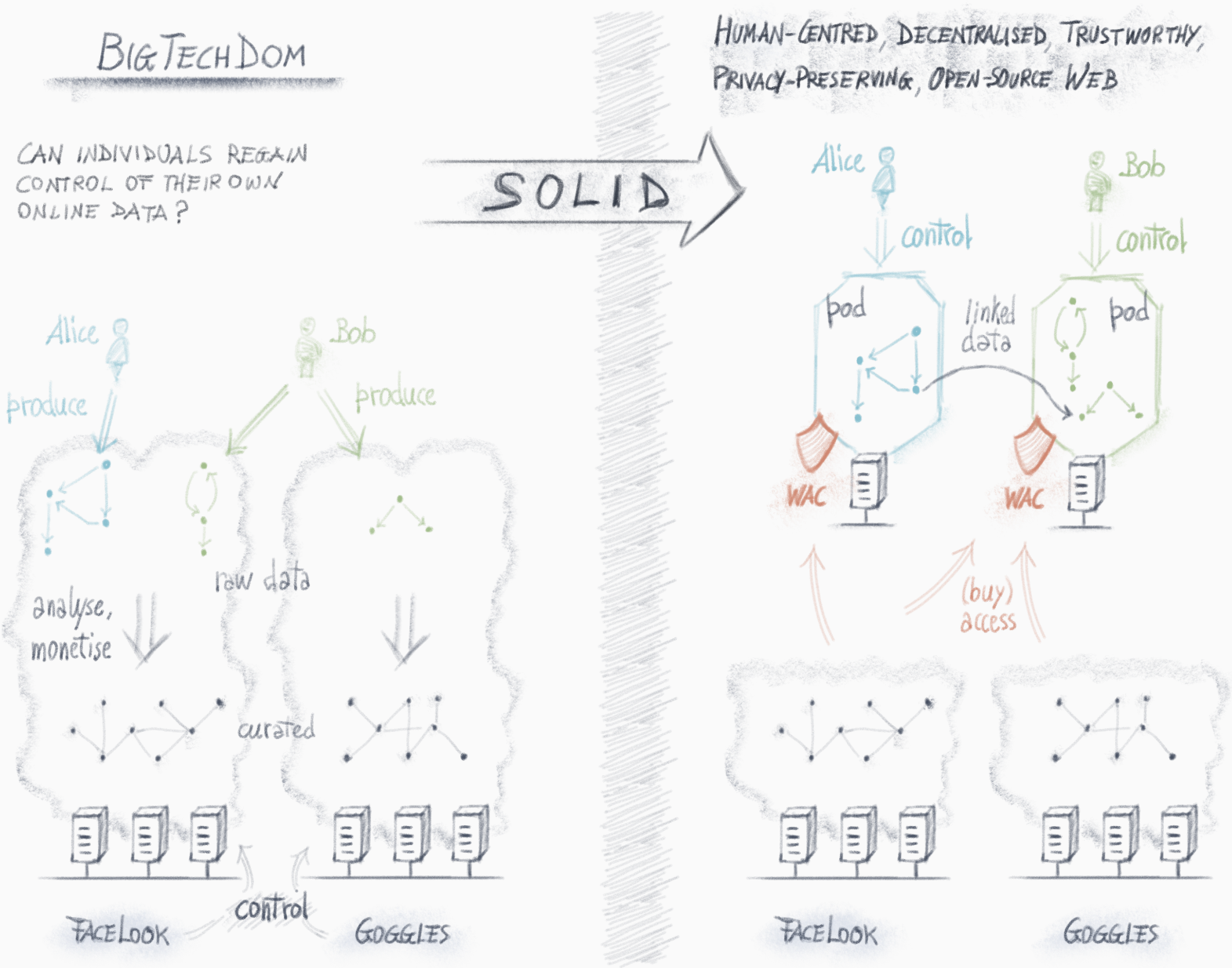
- Human-centred
- Decentralised
- Trustworthy
- Privacy-preserving
- Open-source

where individuals, not big corporations, are truly in control of their data.

The Web Today



The Web Tomorrow?



The Snag

A strong privacy and security architecture is critical to realise this vision.

But how strong is the spec and how secure are WAC implementations themselves?

IBM's 2023 data breach report:

- Bugs cause 40% of the data breaches
- Average data breach cost: $4,000,000

So why should a user trust an implementation to be secure?

Want

For individuals to truly regain control of their online data in a decentralised architecture, WAC implementations should provide

- strong correctness guarantees

But to check whether an implementation is correct, the spec must be

- unambiguous—exactly one interpretation exists
- consistent—free of contradictions

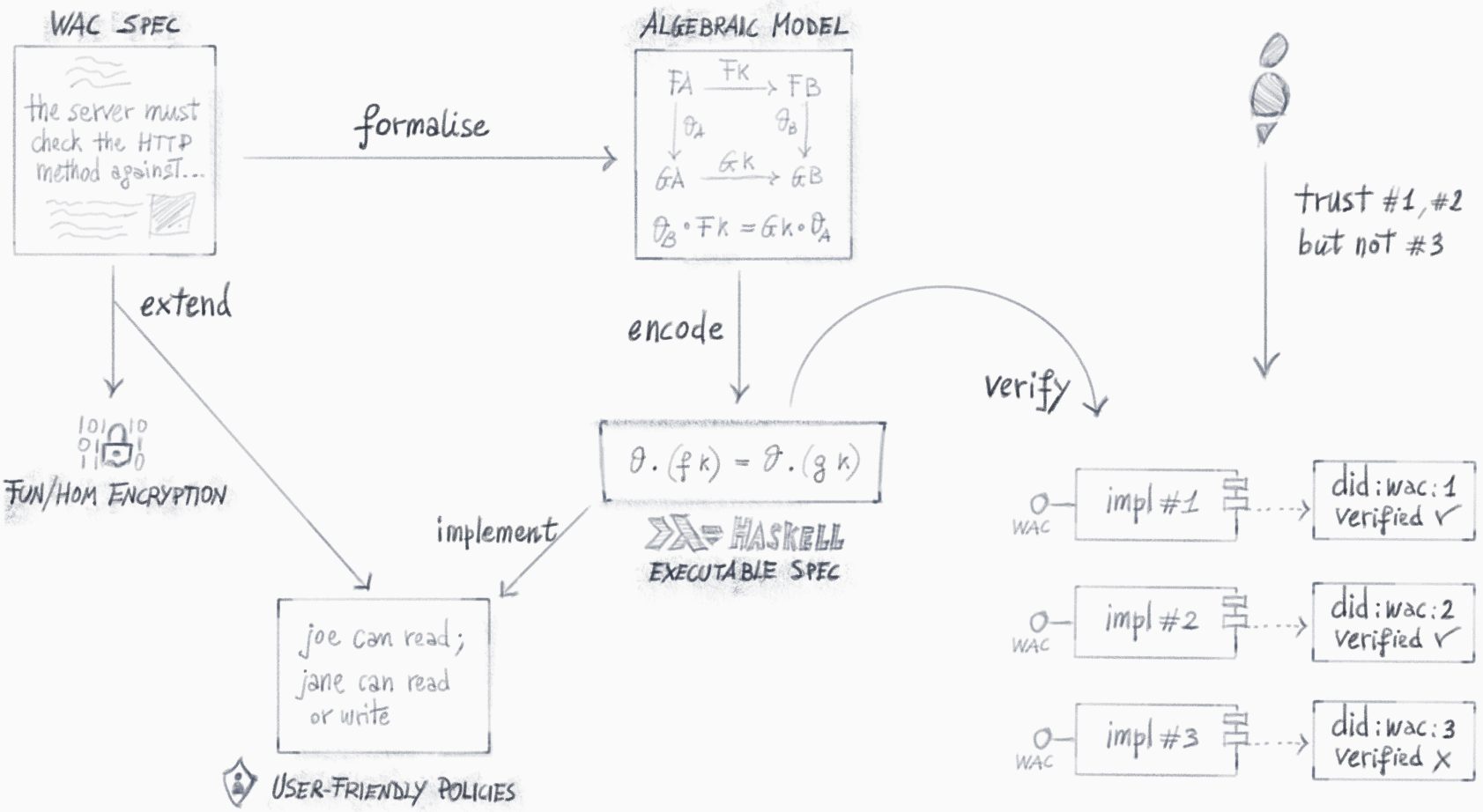
Mission

Contribute to the Solid journey by improving & extending WAC

- Decentralised
- Provably secure
- Consistent
- User-friendly

How?

The Master Plan



Proof of Concept

Whack WAC is open-source

- https://github.com/c0c0n3/whack-wac
- Initial algebraic model
- Trivial Haskell executable spec
- Simple Haskell policy EDSL

Formal Methods Teaser

Approach: Adopt mathematical techniques to produce software users can trust to be correct.

Hard: How to get from plain English to equations?

Here's a trivial example to illustrate the approach.

Formal Methods Teaser

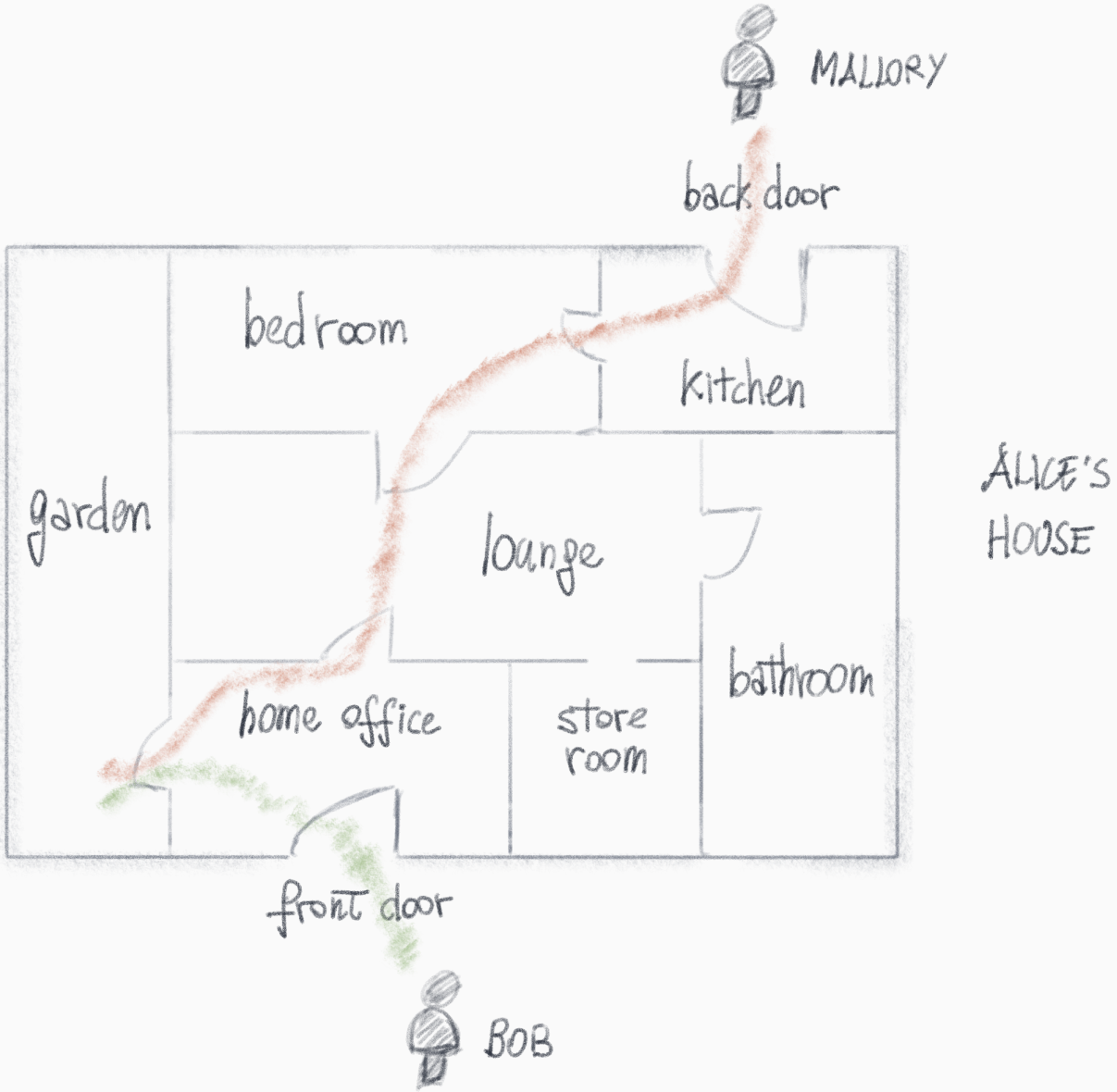
Alice runs her business from home.

She's got a nice garden where she receives customers.

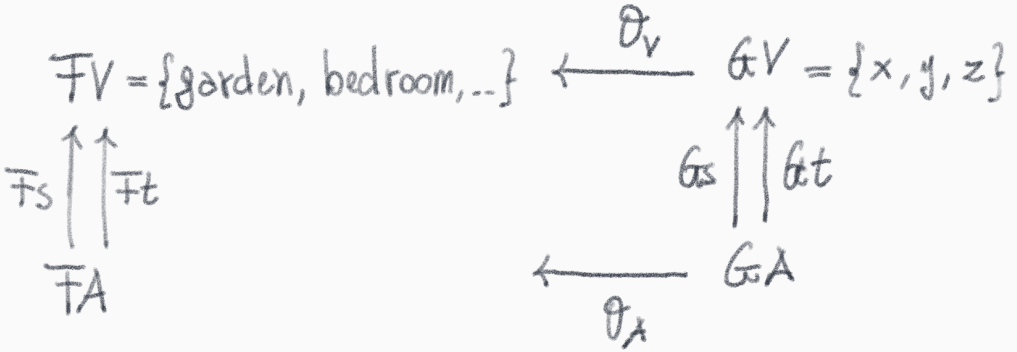
She's hired a new butler, Alfie.

Alice instructs Alfie: *"Usher customers to the garden."*

Formal Methods Teaser



Formal Methods Teaser



$$\theta_V \circ G_s = F_s \circ \theta_A$$
$$\theta_V \circ G_t = F_t \circ \theta_A$$
$$\theta_V(z) = \text{garden}$$

Fire Your Questions :-)