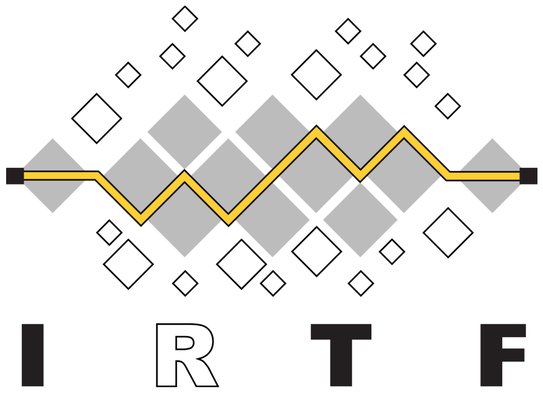
An Analysis of Secure Scuttlebutt as an ICN System

Christian Tschudin, University of Basel
Sep 24, 2018
ICNRG slides



Overview

- Context: Decentralized Web (and one slide on Zooko's triangle)

- Secure Scuttlebutt (SSB)
  - as a social media app
  - **as a technology foundation**
    (names, principles, assets, working model)

- 1:1 comparison with NDN/CCN concepts

- Why SSB is significant

- Outlook
  - SSB challenges
  - ICNRG opportunities

(Re-) decentralizing the Web and more

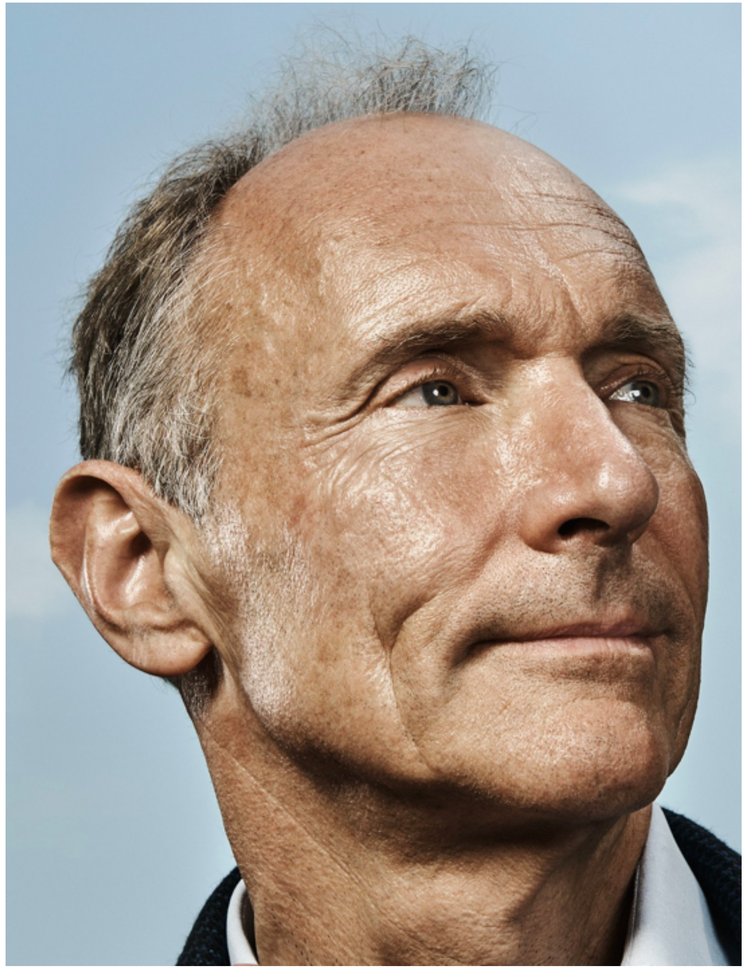
The **2nd Decentralized Web Summit** (Jul/Aug 2018) revealed:

- big names from the past (Vint Cerf, Diffie Whitfield, Tim Berners-Lee)
- a highly motivated crowd of enthusiasts (700 participants)
- big enterprises watching and sometimes already engaging
- hopeful startups
- radical technology tinkerers.

Main theme: “privacy, security and freedom”
- Freedom means: clawing back the Web from the centralization guys
  the GAFA gang (Google, Apple, Facebook and Amazon)

Something important happened here, critical mass got together

“I was devastated” (Tim Berners-Lee, July 2018)



FB, Cambridge Analytica just tip of the iceberg

“The increasing centralization of the Web has ended up producing [..] a large-scale emergent phenomenon which is anti-human.”

“reclaim the Web from corporations and return it to its democratic roots”

SOLID – SOcial LInked Data
- ongoing project at MIT
- started in 2016, lead by TBL

‘decent’ viewpoint (as seen by SOLID)

Paradigm shifts:

1. **End users become data owners**. This is the most well-known decentralization aspect: we store our data in places of our choice, which improves privacy and control.

2. **Apps become views**. As apps become decoupled from data, they start acting as interchangeable views rather than the single gateway to that data.

3. **Interfaces become queries**. Data will be distributed across highly diverse interfaces, so sustainable apps need declarative contracts instead of custom data requests.

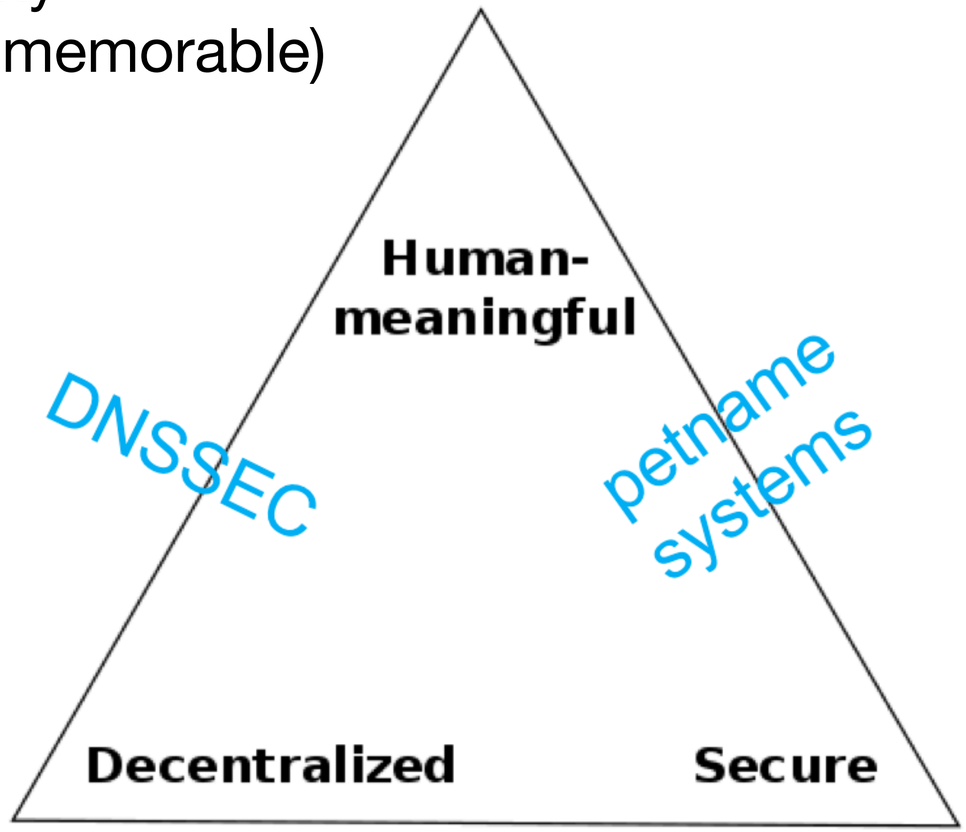
Decent viewpoint on “names”

Natural approach for decentralized identifier (mgmt):

Use public key of random crypto key pair as identity
Problem: crypto keys not human-meaningful (e.g. memorable)

Zooko’s Triangle (Zcash CEO):
Conjectured that no single kind of name or id
can achieve more than two of the three properties.

Example: DNSSec offers a decentralized,
human-meaningful naming scheme, but is
not secure against compromise by the root.



SSB - as a social media app

From Wikipedia, the free encyclopedia

Scuttlebutt in slang usage means rumor or gossip, deriving from the nautical term for the cask used to serve water (or, later, a water fountain).[1][2]

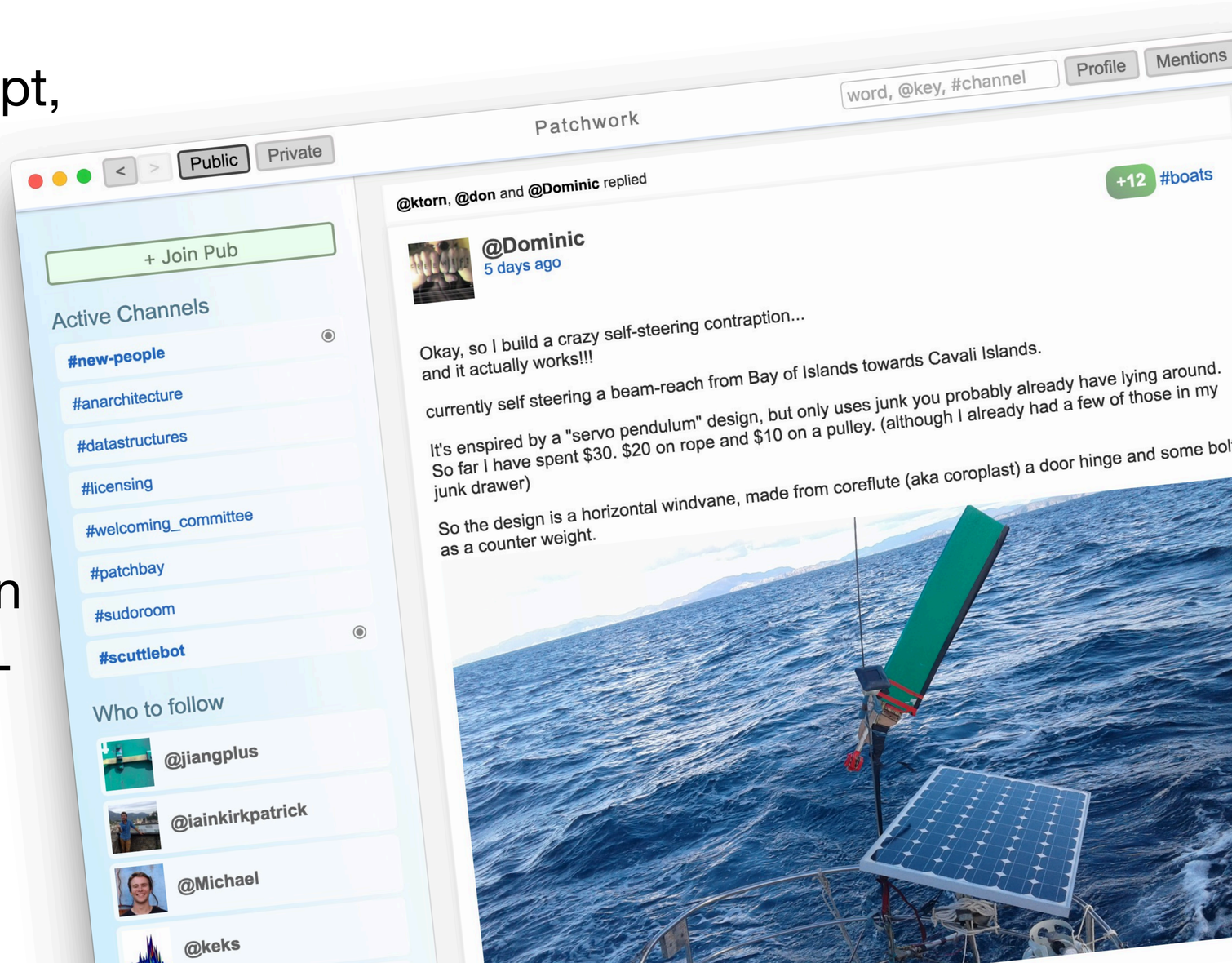
Started 2014 in New Zealand by *Dominic Tarr*
- group of ca 20 core developers with similar social and discourse-aware mindset, “new eco”

- “social viewer” is main app, **for themselves** (—> next slide)

- Other apps running over SSB:
  `git-ssb`, “gathering” (calendar invite), book reviews
  some betas: chess, secret-sharing for key recovery

SSB “Patchwork” (viewer app)

- SSB is almost pure JavaScript, of high quality
- Excellent description of the security, RPC and gossip protocol
- Desktop browser is based on Electron, Android app in pre-beta.

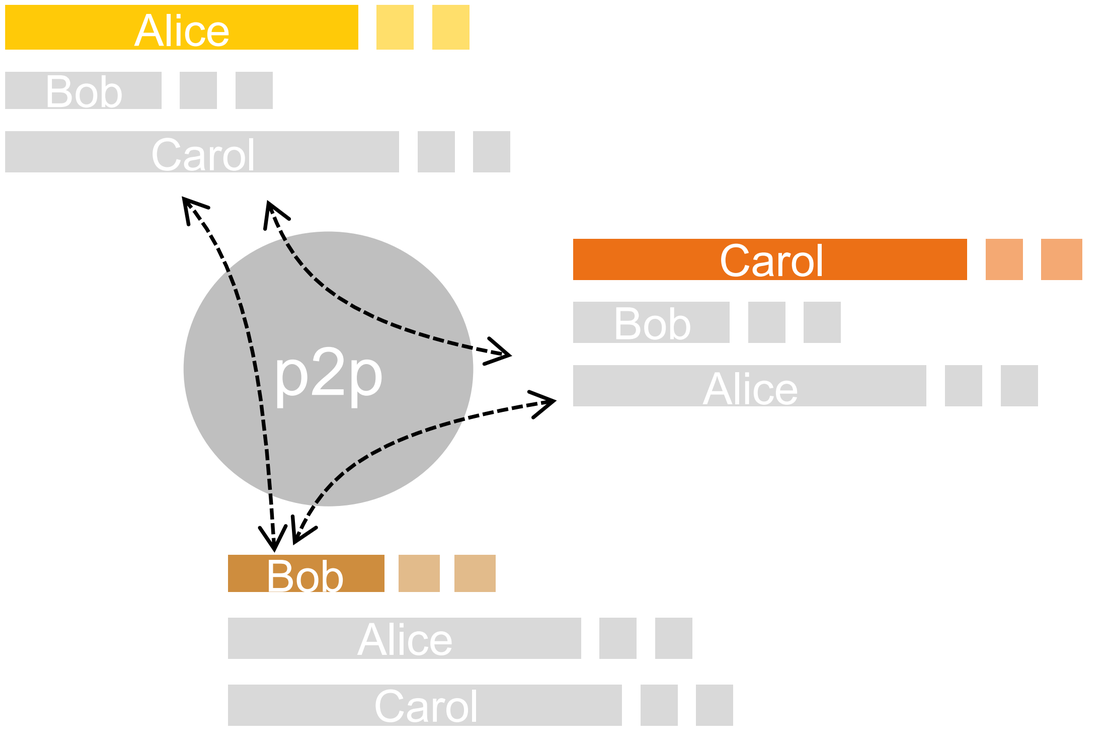


Replicated Logs and Subjective Readers

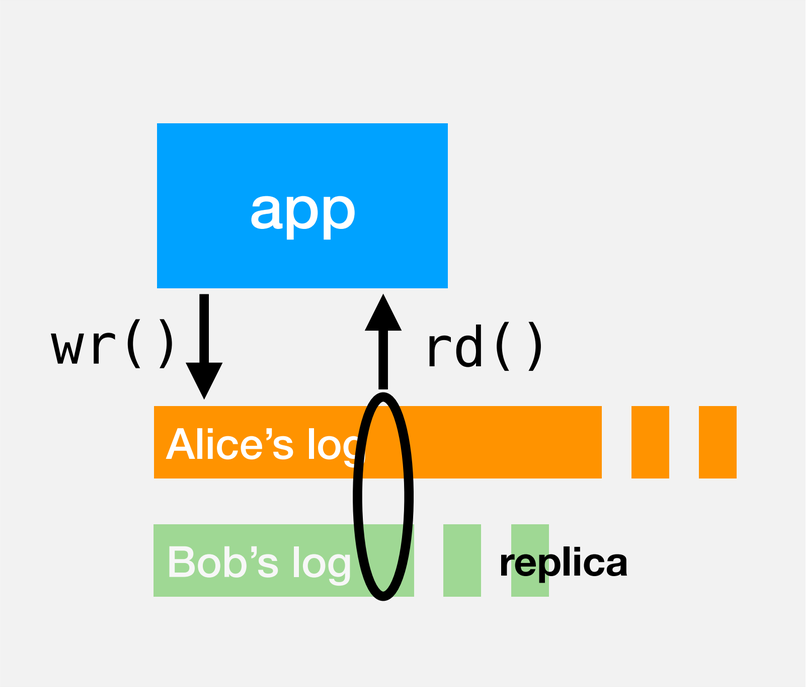
SSB has exactly one data pod per user

- Ground truth are the individual **append-only logs:**
  - hash-chained signed messages
  - replication via peer-to-peer fabric

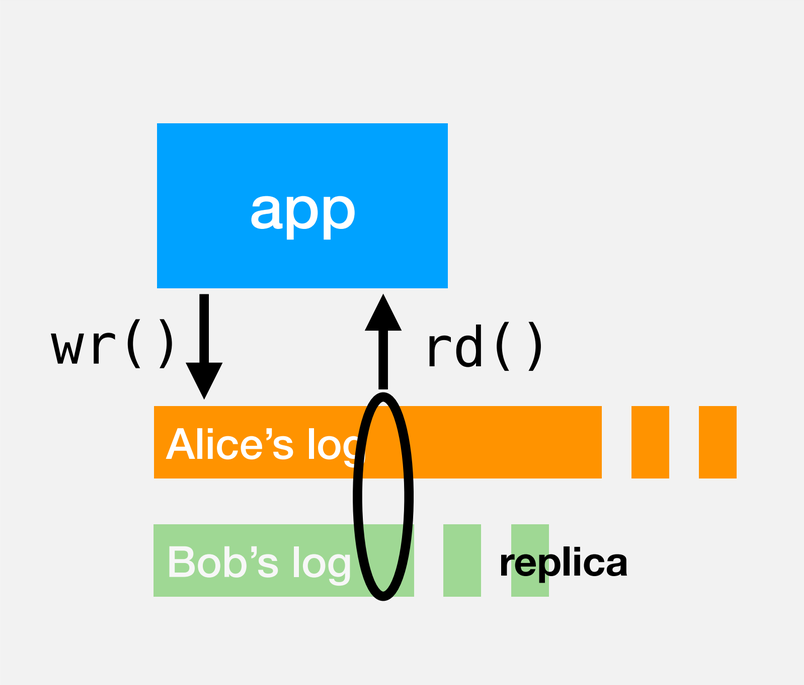
- “subjective reader”:
  locally reconstruct app-level data types (e.g. chat dialogue) from stitching together entries from each participant’s log

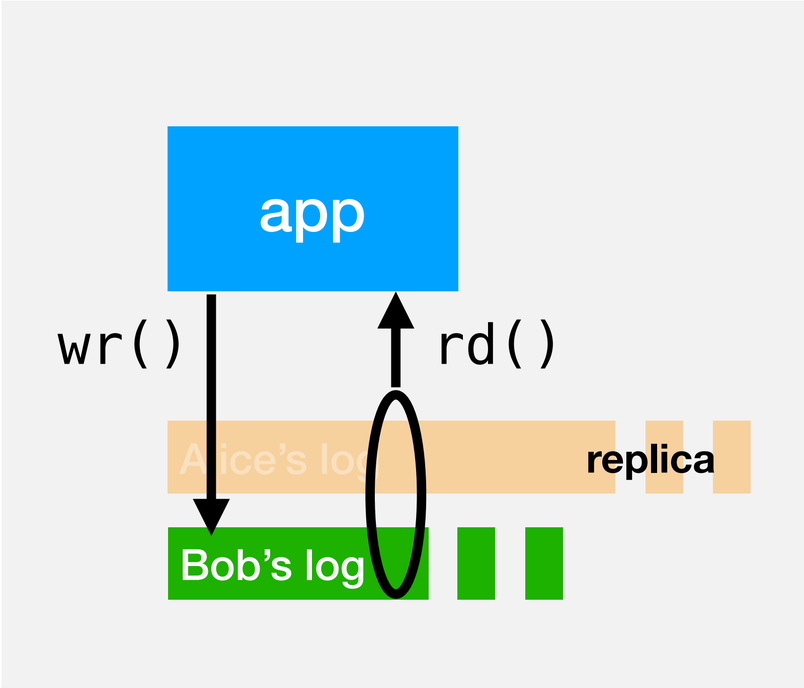


Replicated append-only logs

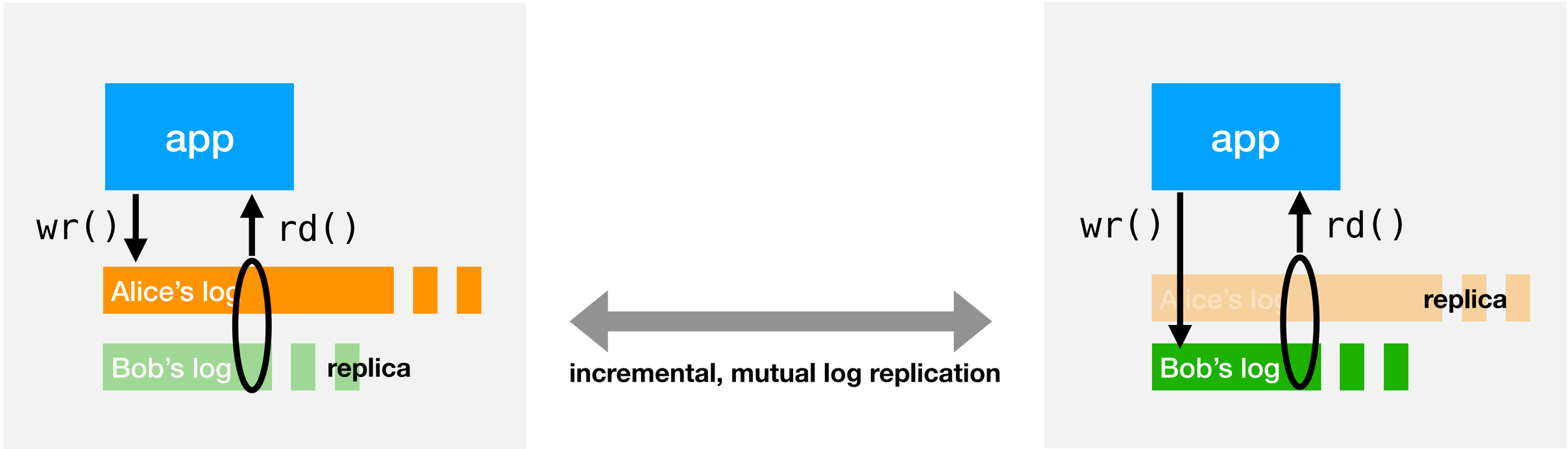


Replicated append-only logs

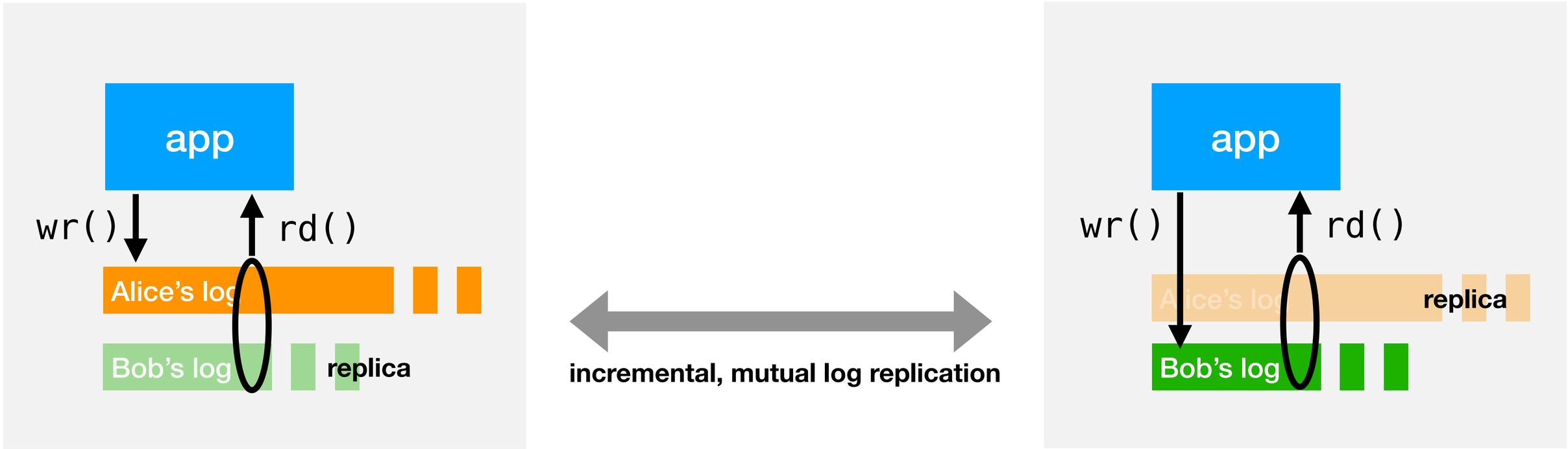




Replicated append-only logs

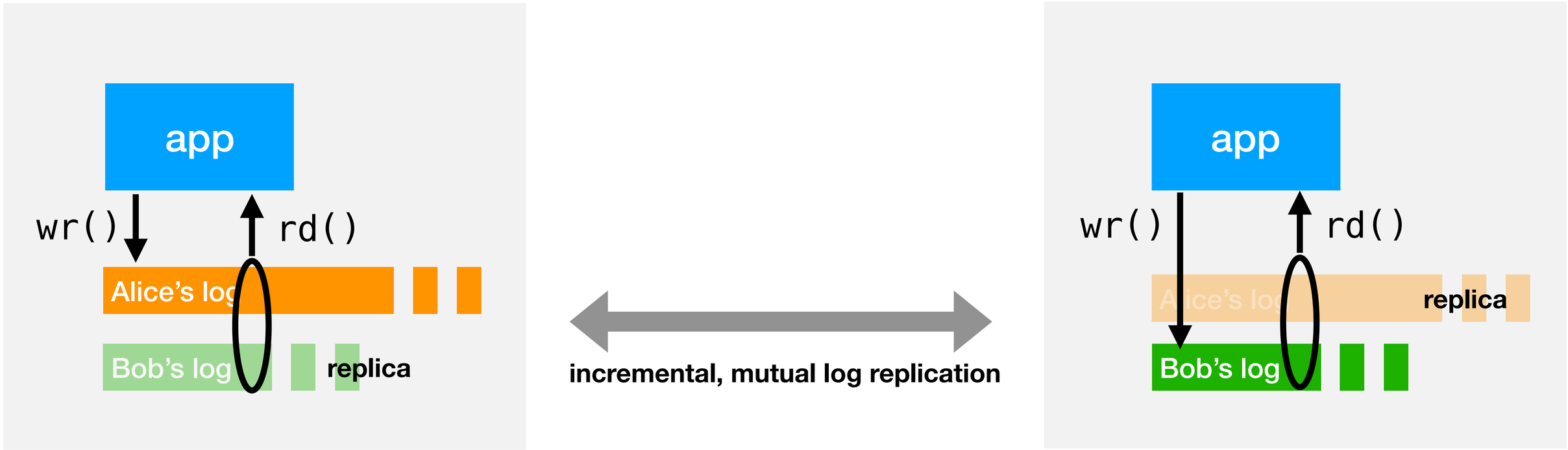


Replicated append-only logs



Abstraction from how (protocol-wise) information updates are propagated:
what matters is that *extensions* of remote logs are securely brought to you

Replicated append-only logs



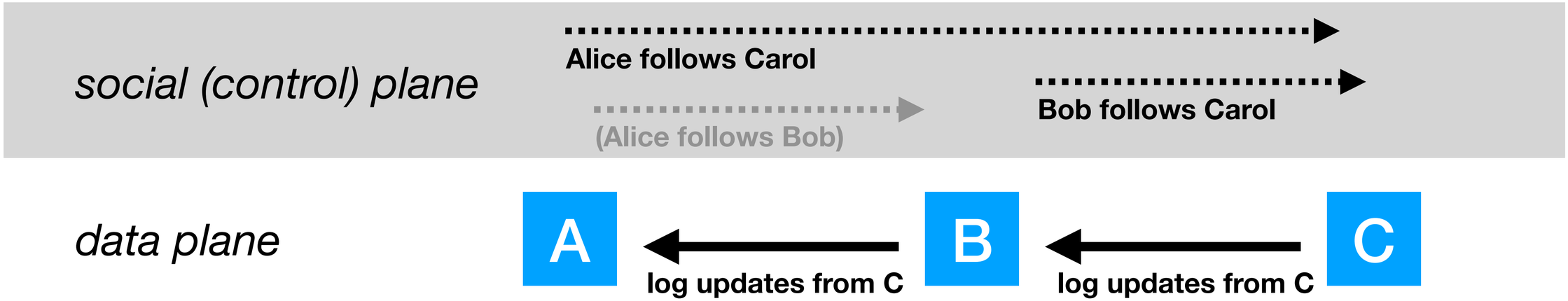
Abstraction from how (protocol-wise) information updates are propagated:
what matters is that *extensions* of remote logs are securely brought to you

Make **secure syncing on log extension** the core network service
(you MUST exclude all other data, anyway)

SSB - social graph-based connectivity

Pure **receiver-driven approach**: only replicate what you are interested in (you never see content from a peer you are not interested in)

- "expressing an Interest" = "to **follow** a peer"
  - put a public "follow" statement in your log
  - forwarding elements will only forward log updates from followed peers
  - establishes a one-directional content delivery gradient, transitive



SSB - onboarding problem

Social graph-based connectivity is unforgiving and cruel:

- at birth (=when trying to join SSB), nobody will talk to you:
  a) you do not know any peer's public key
  b) even if you knew some peer (out of band), they would not serve content to you because they don't follow you

- you need to be introduced (adopted) by an existing SSB user

- Technically: use mDNS/local broadcast/QR codes for first encounter and starting to replicate a peer's log

- **Real** social control plane: PGP's web of trust, signing party style onboarding

Overview of SSB's technical merits

- Uses fast **ED25519** elliptic curve crypto (2012), the "decent de-facto standard":
  1 key pair used for all (!) of: DH key exchange for RPC, encryption+signing of msgs

- e2e encryption and **fully privacy-preserving handling of meta-data** (up to 8 rcpts)
  (must attempt decrypting all messages from friends, set of friends is public, though.

- **Gossip**-based content dissemination
  - scaling comes from p2p and restricting replication to your friends

- Extremely **delay-tolerant**: works over the Internet or "pocket switching" (USB sticks)

- Highly **resilient**: I destroyed my append-only log, and got it back from my "followers" (who keep a full replica of my log).

Format of SSB log records

JSON-based encoding

```
{
  "author": "@AiBJDta+4boyh2USNGwIagH/wKjeruTcDX2Aj1r/haM=.ed25519",
  "sequence": 48,
  "previous": "%9itfeYbt8EXCy8v04TrUevsw37momPxBoM/NFX3cRpE=.sha256",
  "timestamp": 1534460709199,
  "hash": "sha256",
  "content": "RDKMZ4gcfdb…B44V3A==.box",
  "signature": “Vwih8S1UOAzVqRvbYnQg…l3dj==.sig.ed25519"
}
```

Note:
- hash chain (‘previous’ field)
- full name of producer (‘author’ field), so that signature can be validated without cert

RPC - Secure Handshake (SHS)

- peers connect via homebrewn secure RPC protocol

- Secure Handshake (SHS): mutual authentication between the peers
  - important for privacy:
    an observer cannot not see whose log was extended

- SHS establishes a bydirectional RPC channel:
  - both sides can send requests
  - one-time requests
  - also long lived streams (open-ended streams of updates -> notification)

- feels like “app-level multicast” with immediate notification (see “pub” relays)

`https://ssbc.github.io/scuttlebutt-protocol-guide/`

SSB over the Internet

- LAN: mDNS

- otherwise: SHS over TCP/IP port 8008

- “pub” nodes: serve as super node
  - stable Internet presence
  - their presence also put into logs, have a peer id (to be trusted)

- Recently: “EBT” — epidemic broadcast tree
  - gossiping content, reduce redundant propagation paths

Three (flat) Namespaces

- Principles (peers): public key “@abcde….ed25519”
- Log records (messages): hash value “%….==.sha256”
- file names “XX”
  (blobs shipped outside the gossip channel because too large)

1:1 comparison with ICN concepts

- tbd

Conclusions

ZERO need for an intermediary like FB, Twitter, Gmail etc,

yet **secure, scalable, resilient, delay-tolerant, privacy-preserving ICN**

- = *everything a “decent-aware end user” wishes for*
- = *everything an attractive ICN system should expose, running NOW*

Important SSB contribution IMHO:

- identifying replicated append-only logs as foundational ICN service

- secure propagation of the “data growth frontier”, sync at the same time