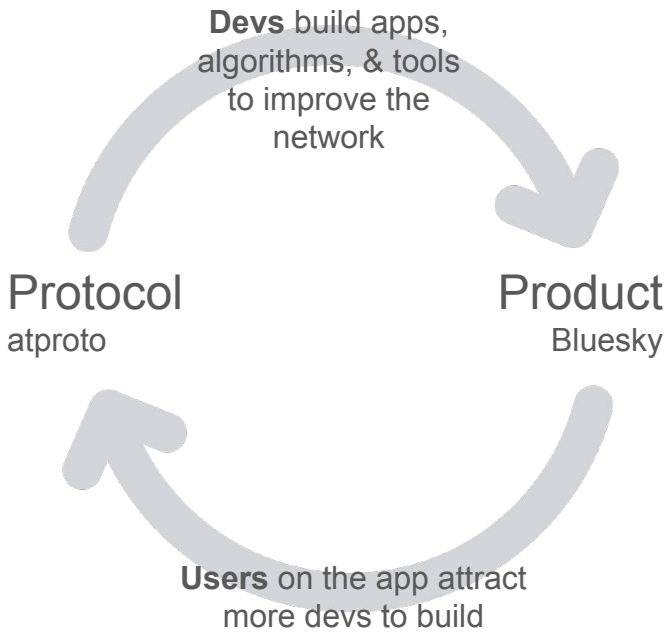
Interoperation in the AT Protocol Ecosystem

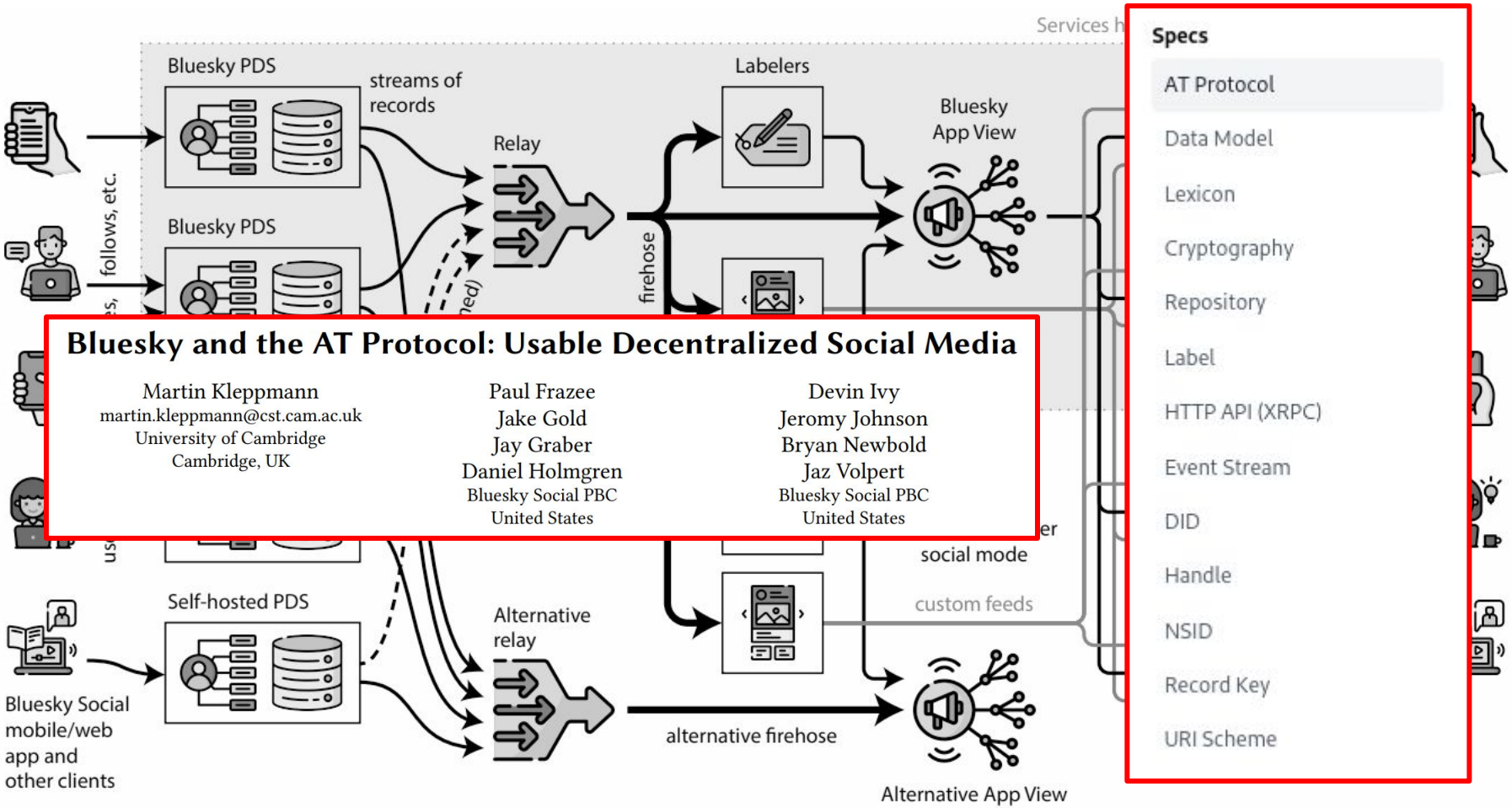
@bnewbold.net / Bluesky

Protocol / Product Co-Development



Characterizing AT Protocol

- Signed location-independent “self-certifying data”
- Authority rooted in user identity
- Split between “data hosting” and “application semantic” layers
- Prioritize **credible exit** over “trustless” or “flat” architectures


Services h
Bluesky PDS
streams of records
Relay
Labelers
Bluesky App View
follows, etc.
Bluesky PDS
firehose
Bluesky and the AT Protocol: Usable Decentralized Social Media
Martin Kleppmann
martin.kleppmann@cst.cam.ac.uk
University of Cambridge
Cambridge, UK
Paul Frazee
Jake Gold
Jay Graber
Daniel Holmgren
Bluesky Social PBC
United States
Devin Ivy
Jeromy Johnson
Bryan Newbold
Jaz Volpert
Bluesky Social PBC
United States
social mode
custom feeds
Self-hosted PDS
Alternative relay
Bluesky Social mobile/web app and other clients
alternative firehose
Alternative App View
Specs
AT Protocol
Data Model
Lexicon
Cryptography
Repository
Label
HTTP API (XRPC)
Event Stream
DID
Handle
NSID
Record Key
URI Scheme

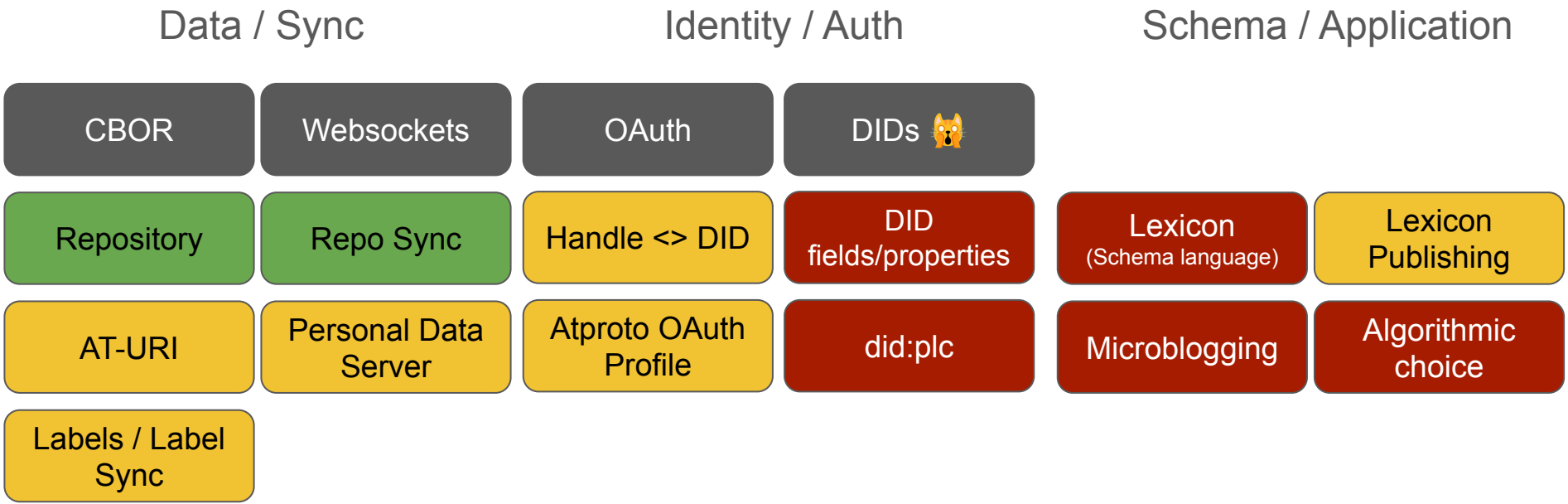
Adoption and Interoperation Status - "Atmosphere"

- 4 distinct PDS implementations operating in the network: pennies/user/year
- 2 distinct full-network Relay implementations operating: $34/month
- 1 independent full-network Bluesky App API server: $200/month
- Can interop independent of any Bluesky network servers
- atproto implementations/SDKs in multiple languages
  - Bluesky: TypeScript, Go
  - Community: Rust, Python
- AtmosphereConf in Spring 2025: 150 developers
- BridgyFed connects atproto and ActivityPub
- Many apps in ecosystem
  - Funded: skylight.social, graze.social
  - Social apps: blogging (whtwnd.com), events (smokesignal.events), photos (grain.social)
  - Other apps: git forge (tangled.sh), video streaming (stream.place)

Why a Standards Body?

- "atproto is just a glorified Bluesky API" is a failure mode
- Confidence for larger players to build and interoperate
- Neutral venue & structure for evolution of protocol
- Expert review and improvements
    - security, extension points, ability to evolve, and general correctness

ATProto Framework



Considerations

1. Is the IETF the right venue for this work?
2. Is the IETF community interested in supporting this work?
3. Are we too early in protocol lifecycle? Too late?
4. Does it make sense to bring a small subset of the framework?
5. How should we describe atproto?
    - "Social web", "Structured data sync", "Global broadcast"
6. What are the implications for future protocol development? i.e. private data?
7. How *likely* are painful network-interop-breaking changes?
8. Are there risks or downsides to "just posting I-Ds"?
9. Is any of this a good idea?

Protocol Ecosystem Open Challenges

- Moderation: most devs are not engaging with moderation
- Extension Points: how small apps can extend popular schemas
- Demand for Protocol Features: private data, group/community
- Scope Creep