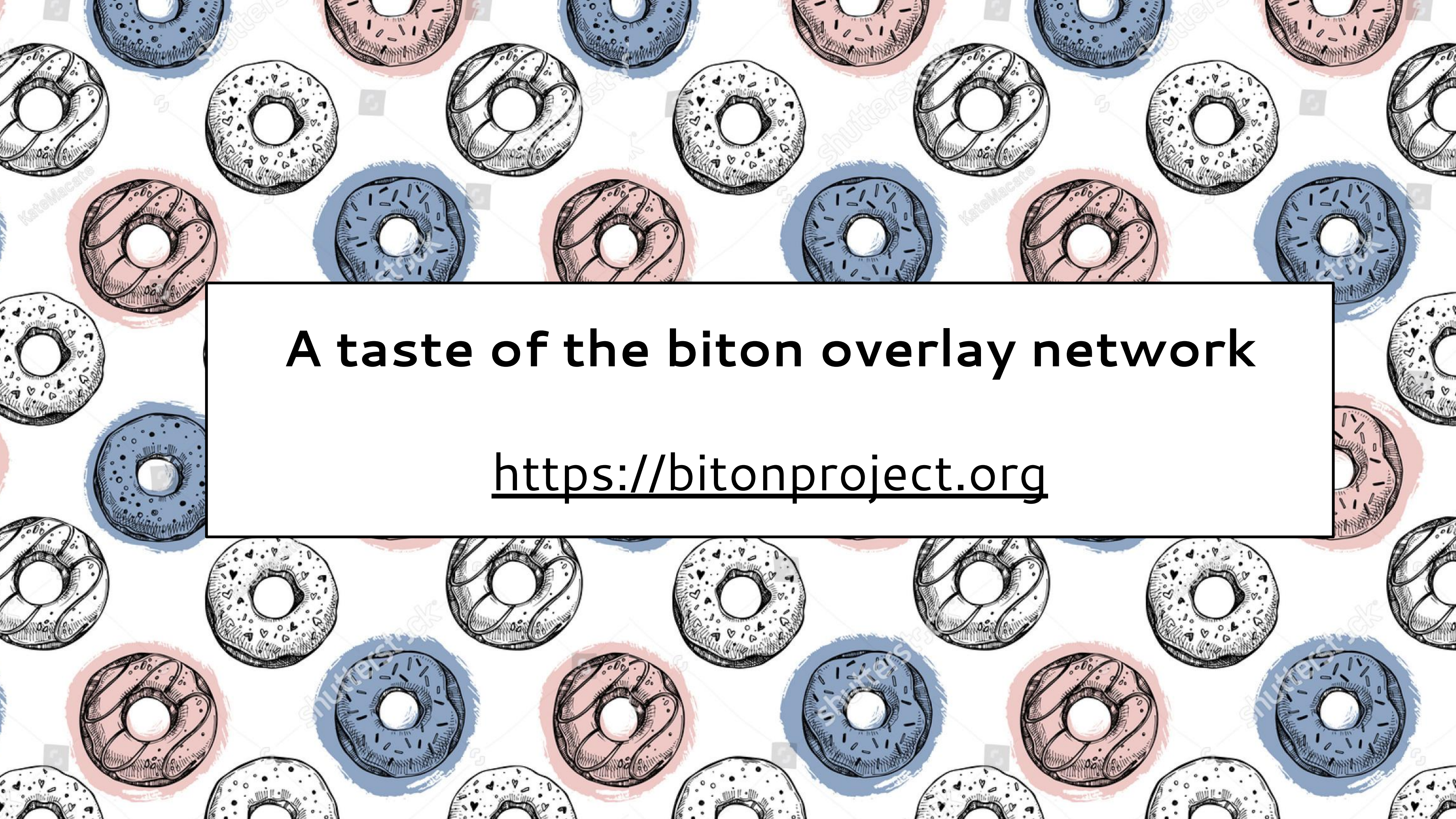


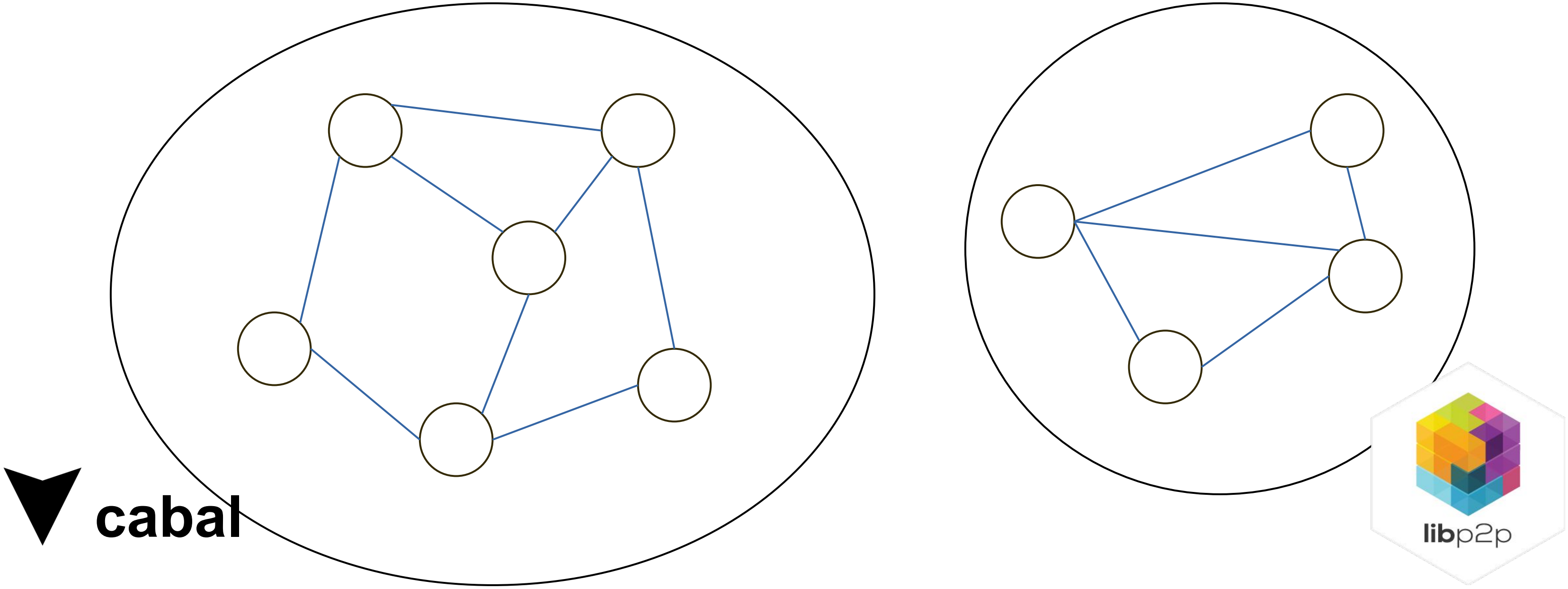
A taste of the biton overlay network

https://bitonproject.org

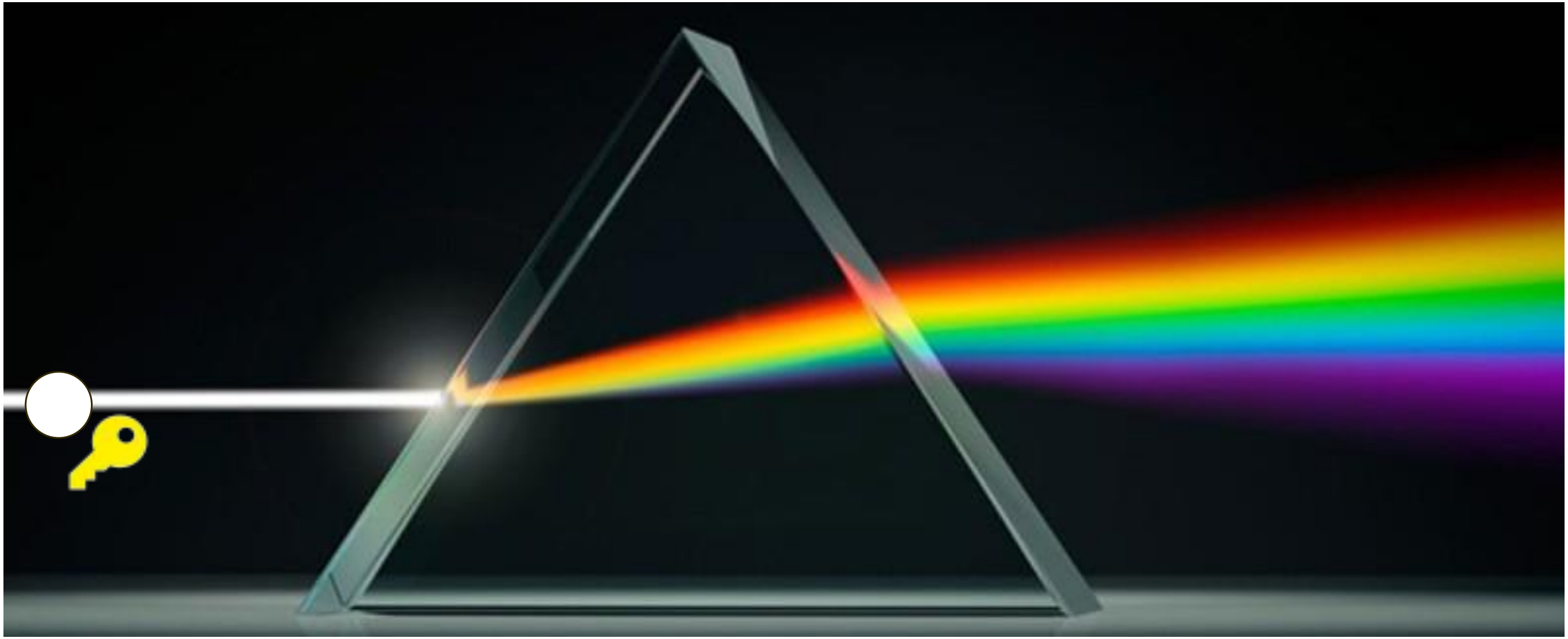
Basic ingredients

- **Local**-first
- Flexible **trust** models
- **Swarms** for application namespaces and overlay partitions
- Low-latency **routing** (service-oriented)
  and redundant **storage** (information-centric)
- **Mixing** messages around sender and receiver, and across swarms
- Greedy routing over a **global overlay** network

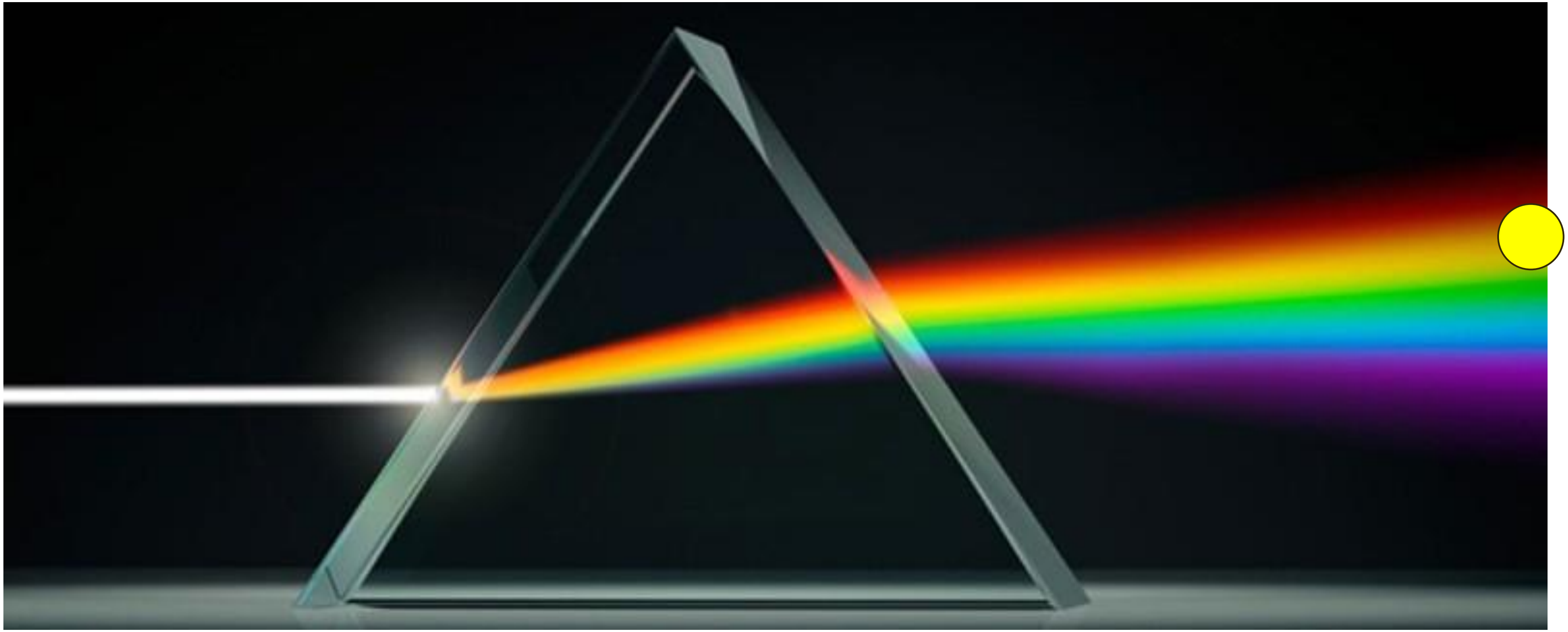
Domain-specific local swarms

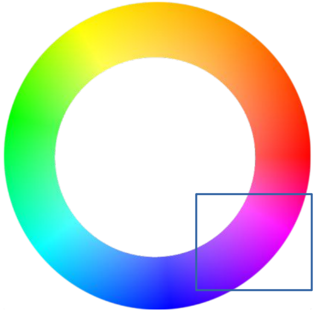


biton overlay address space

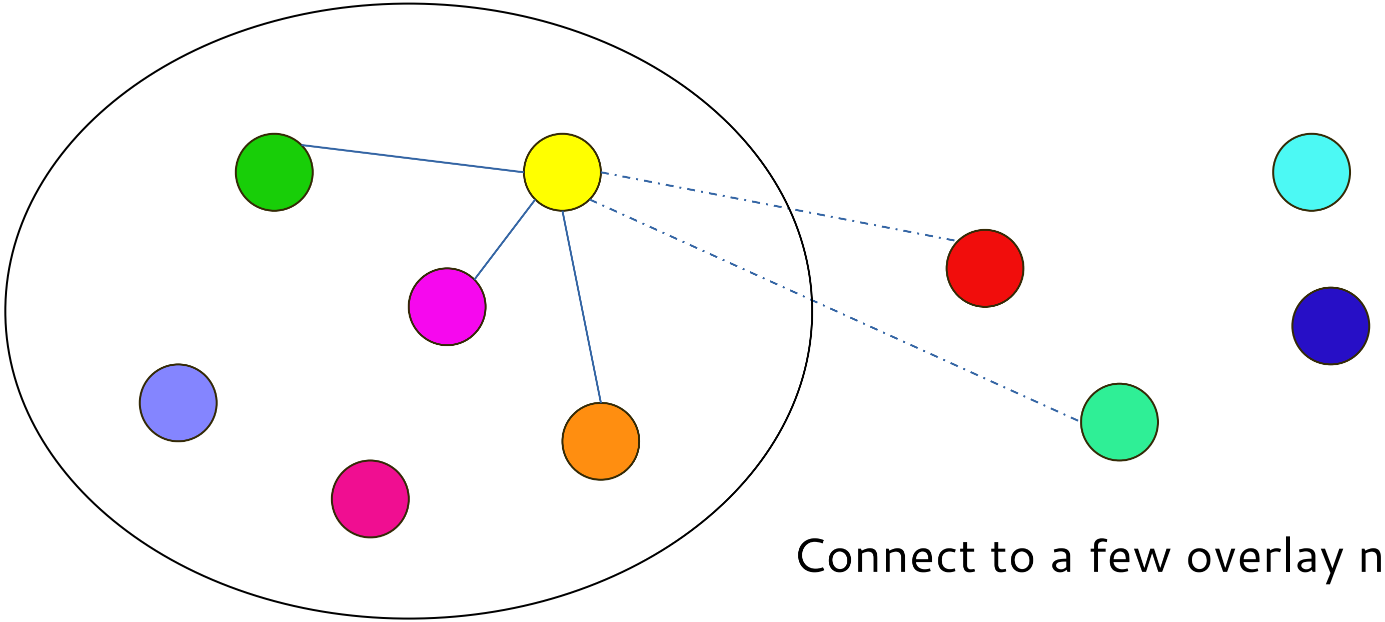


biton overlay addressing



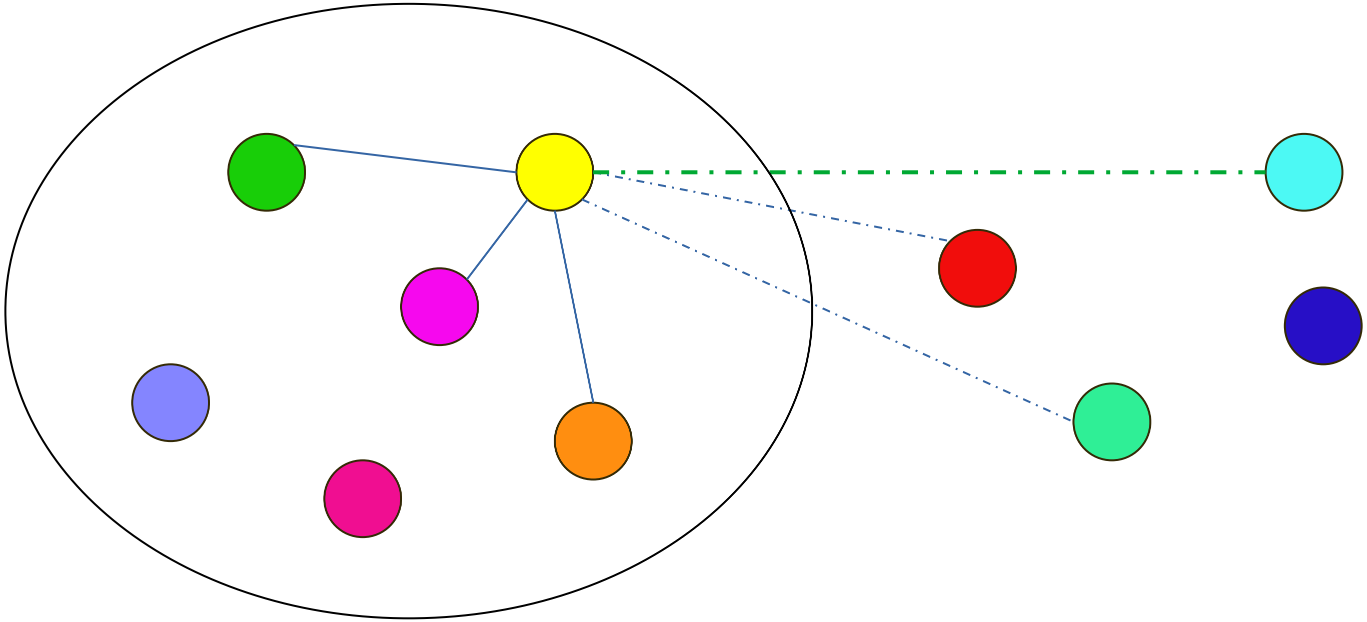


Join spectrum edges



Connect to a few overlay neighbors

Peer trust levels

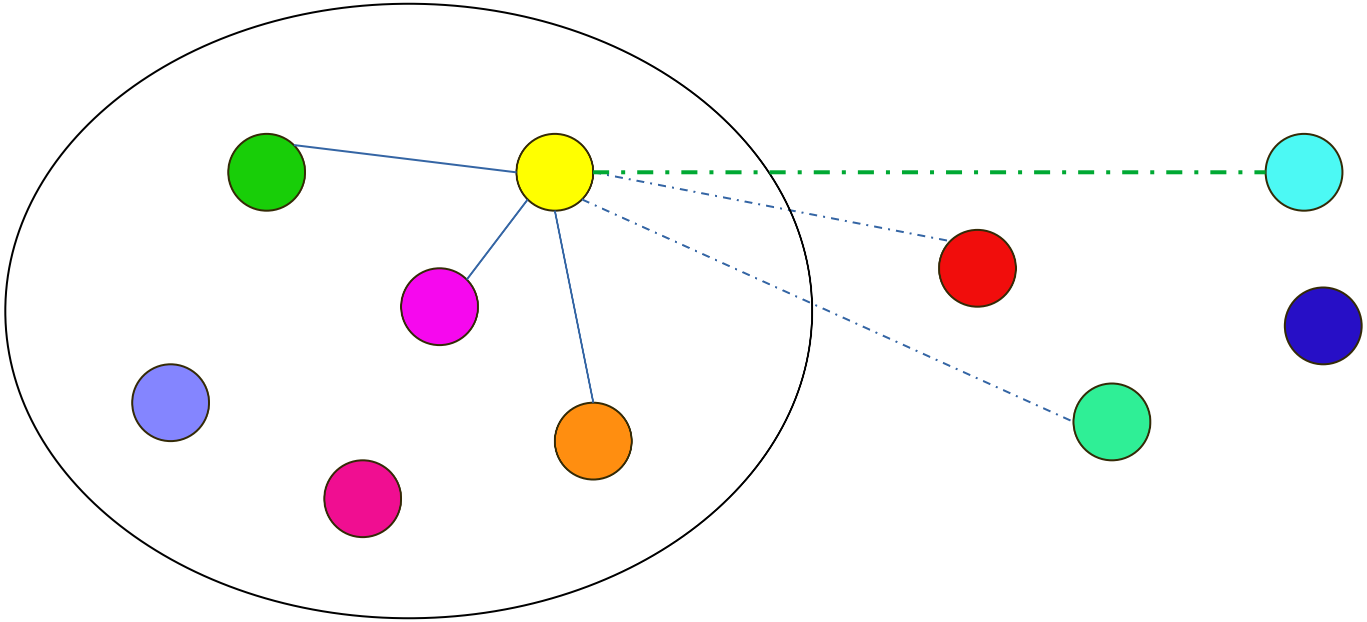


Friends: Trusted connections regardless overlay position

Commoners: Connections in domain-specific swarms

Strangers: Neighbors in the biton overlay

Mixing and storage replication across friends,
domain-specific swarms, and the global overlay



Domain-specific content
does not leave the local swarm

Plausible deniability

The adversary cannot know with probability >50% whether a node is the original sender of a request.

Mixing similar to Crowds [Reiter et al., 1998], but with:

- Long-lived, trusted connections
- Mixing messages across swarms
- Storage replication and overlay requests as cover traffic

BitTorrent transport

- Long-lived, encrypted, high bandwidth connections
- Incentives for contributing resources, prioritizing rare files etc.
- Facilitates decentralised peer discovery
- Wide adoption means high collateral if censored
- Supernodes improve performance and cached content lifetime

Deployment?

- Supernodes and the Internet backbone
- Mesh topologies
- Label switching (MPLS)
- biton “control plane“

Cooking up next

- Mixing protocols
- Strategies for efficient routing in the global overlay
- Distributed storage (caching, lookup algorithms, flooding attacks...)
- Flexible trust models accessible to the network stack
- Sharding blockchain networks

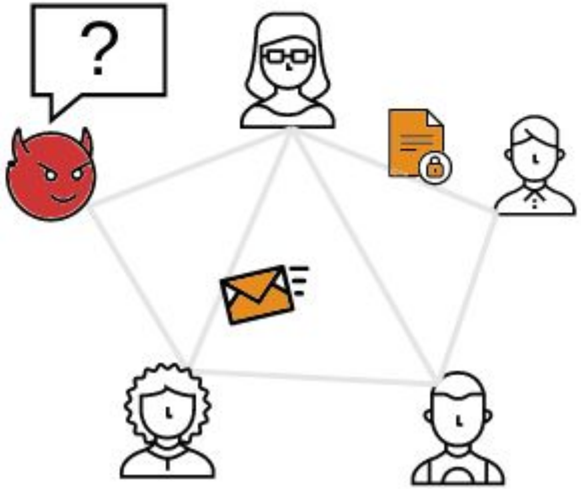
https://demo.bitonproject.org

smtp@bitonproject.org
PGP: 567E 168D 9FE7 6EFA 6784 B977 010F F6C7 E9B4 F3BD

biton is a decentralized network against censorship and surveillance of our online activity.

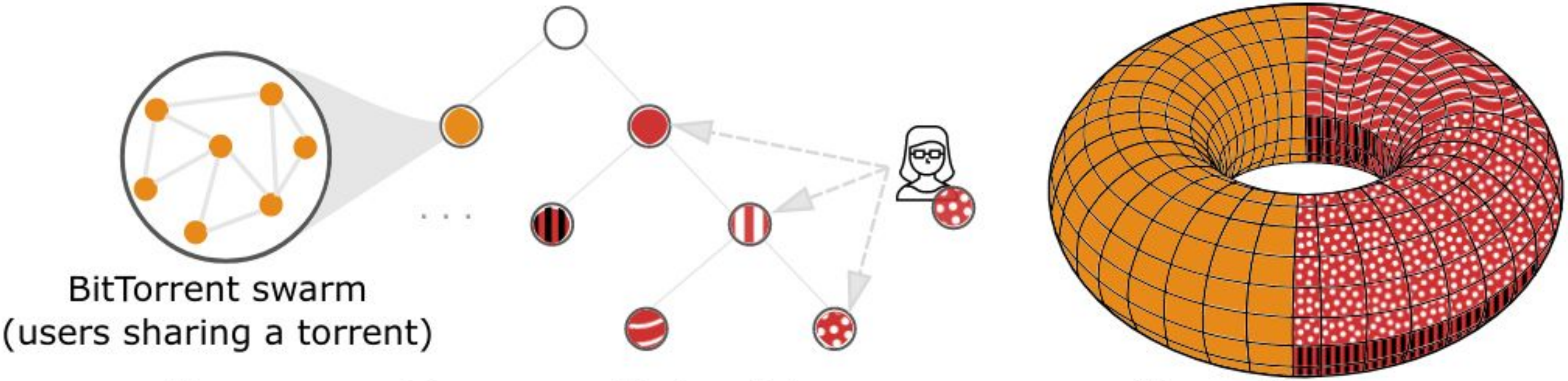
Users connect directly with each other over encrypted BitTorrent connections. They can share files and route traffic through other users, while network operators cannot identify or block their requests.

- Static content is retrieved from the decentralized cache
- Replication protects metadata about user activity
- Routing messages and traffic through others, e.g. through users with access to the uncensored Internet, does not expose the identities of the sender or recipient
- biton can be deployed over community networks, and in that way function during Internet shutdowns



The biton overlay network

We consecutively split the biton address space and map each partition at each level to a BitTorrent swarm. Users connect to swarms that match a prefix of their biton address. Routing across swarms follows the Content Addressable Network design.



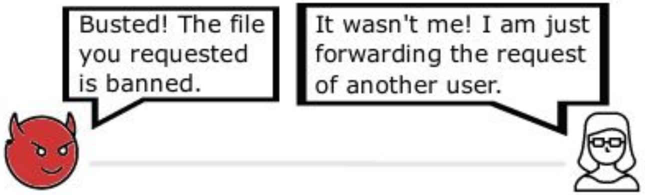
BitTorrent swarm
(users sharing a torrent)

Users connect to swarms that match
a prefix of their biton address

biton address space
at three levels of partitioning

- Same addressing mechanism for storing content and proxying traffic
- The overlay can scale, as users connect to swarms matching longer prefixes
- Direct connections to friends across remote swarms improve routing efficiency

Caching and proxying with plausible deniability


Busted! The file you requested is banned.
It wasn't me! I am just forwarding the request of another user.

Adversaries cannot eavesdrop on peer-to-peer links, and therefore cannot trace requests back to the original sender.

Store this encrypted file so that it will be available after I disconnect.
Are they exchanging messages or replicating the files in their swarm?

Storage replication serves as cover traffic. Files are stored with redundancy within the swarms that match their biton address prefix.

Forward this message to a swarm that is closer to the recipient's biton address.

Users are reachable by their biton address.
biton provides a scalable strategy for handling incoming requests.