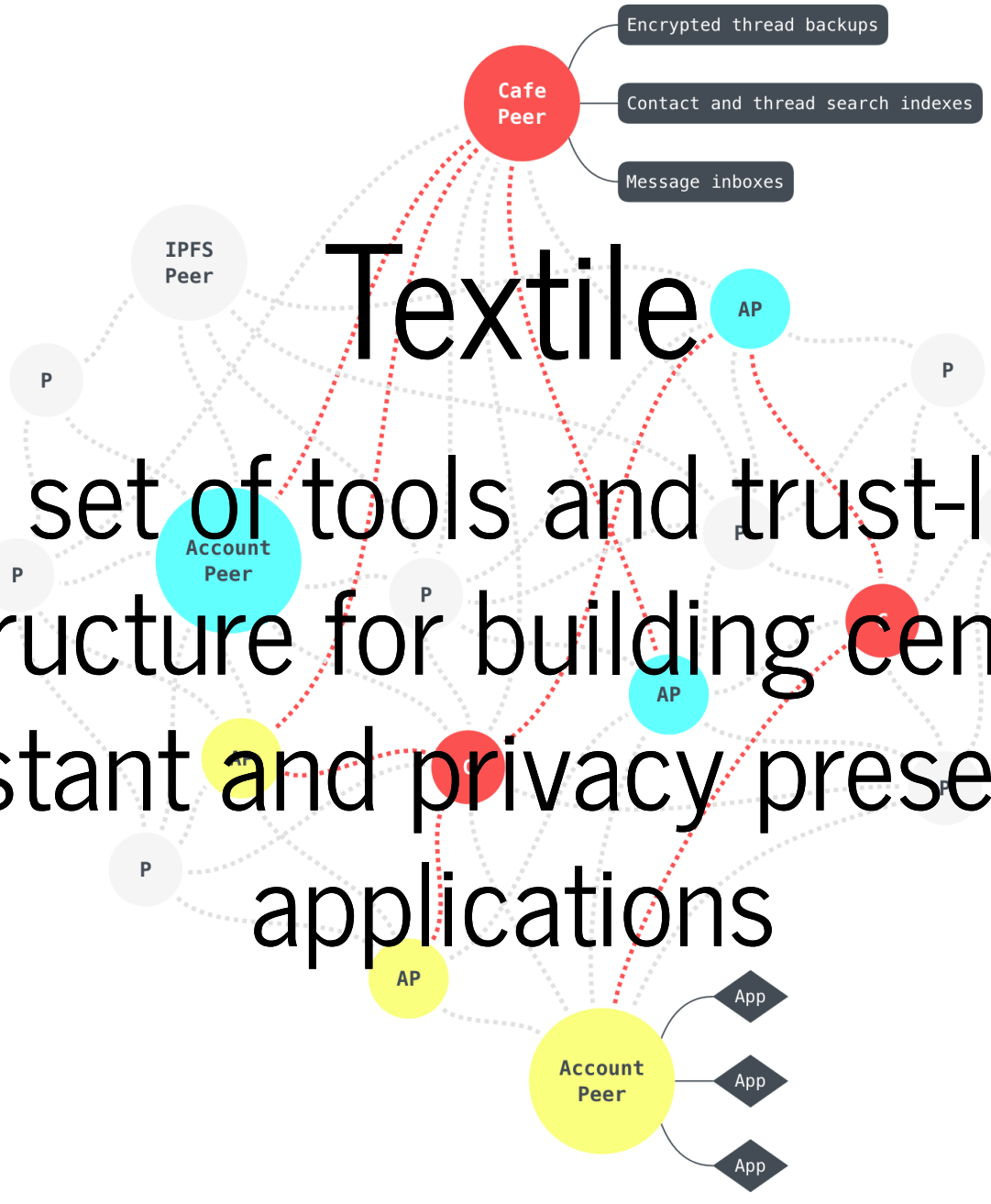
Building DApps with **Textile**, the iCloud for the DWeb

Welcome!

SSID: `Internet Backup - C`

Password: `ShareWithIPFS`



Textile

...a set of tools and trust-less infrastructure for building censorship resistant and privacy preserving applications

Instructors



Carson | Andrew | Benjamin

Sander | Aaron | Thomas

Outline

- Split into two parts, with **break** in the middle
  - First half is **conceptual**/theoretical
  - Second half is **practical**

Structure

1. Welcome & Demo
2. Anatomy of a game/dapp
3. Break & questions
4. Hands on fun/command-line
5. Wrap-up & hackery


21:06
← Back
Hi! I'm free!
Lame
Boo
Send
I
Hey
Yeah
123
space
return
21:06
22:58 REMAINING. YOU ARE IT!
1. Show this screen to anyone playing and they are it.
2. Have them point their native camera at this code to notify the network.
3. Run!
Chat
21:06
waiting for node...
connected
Welcome back, Trackpad
Okay, ready for some Tag?.
You are playing in, IPFS Camp.
The game has started!
Remember, if anyone shows you their red card, you are it!
If the next screen is red... you are it!
Once you are it, you must scan their QR code to take over.
Tap "enter" below to find out who is it!
Ready... set...
Uh oh: node is stopped
connected
Welcome back, Trackpad
Watch out, game on!
You are playing in, IPFS Camp.
Enter

Anatomy of a game

What does it take to build a **Game of Tag** on **IPFS** using decentralized data, content addressing, and encrypted communication?

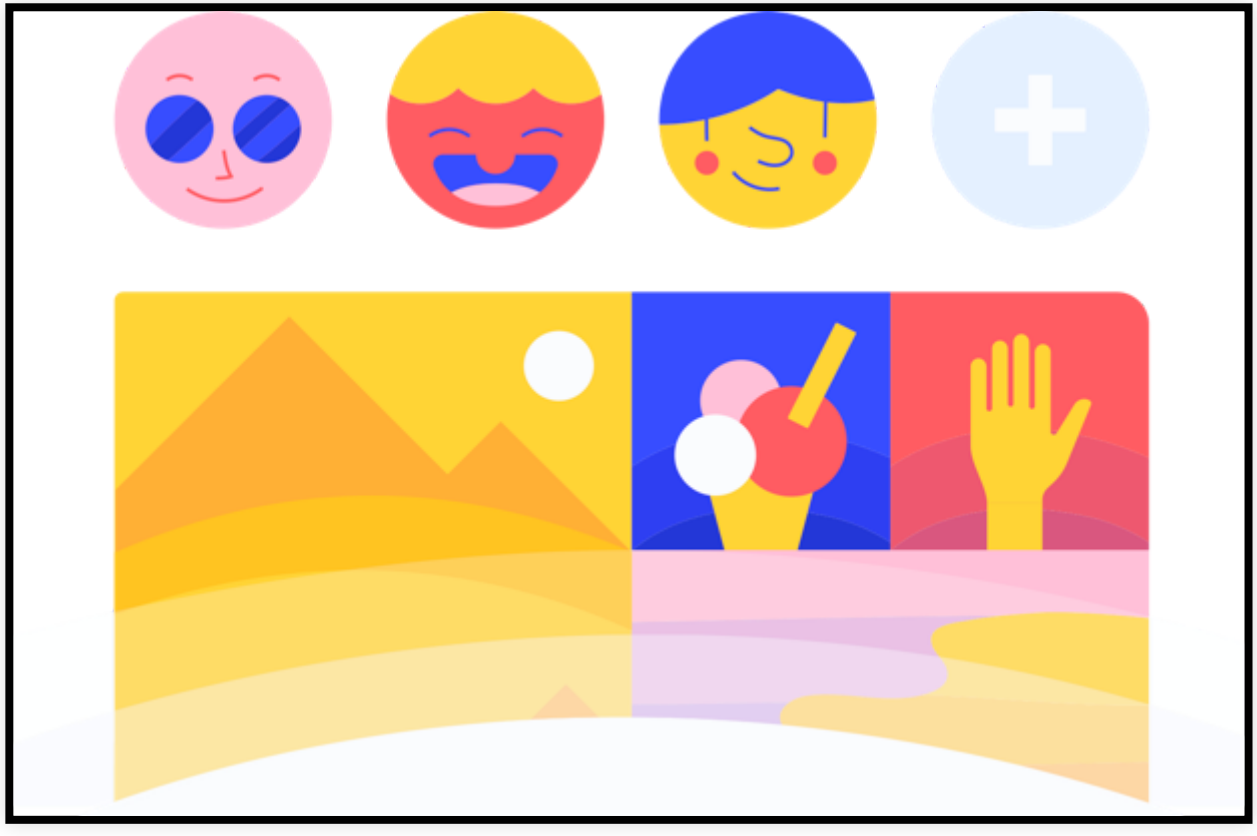
What is a game of tag?

- A group of individuals,
- Agreeing on a set of rules,
- With a shared record or state, and
- A way to communicate & verify game play

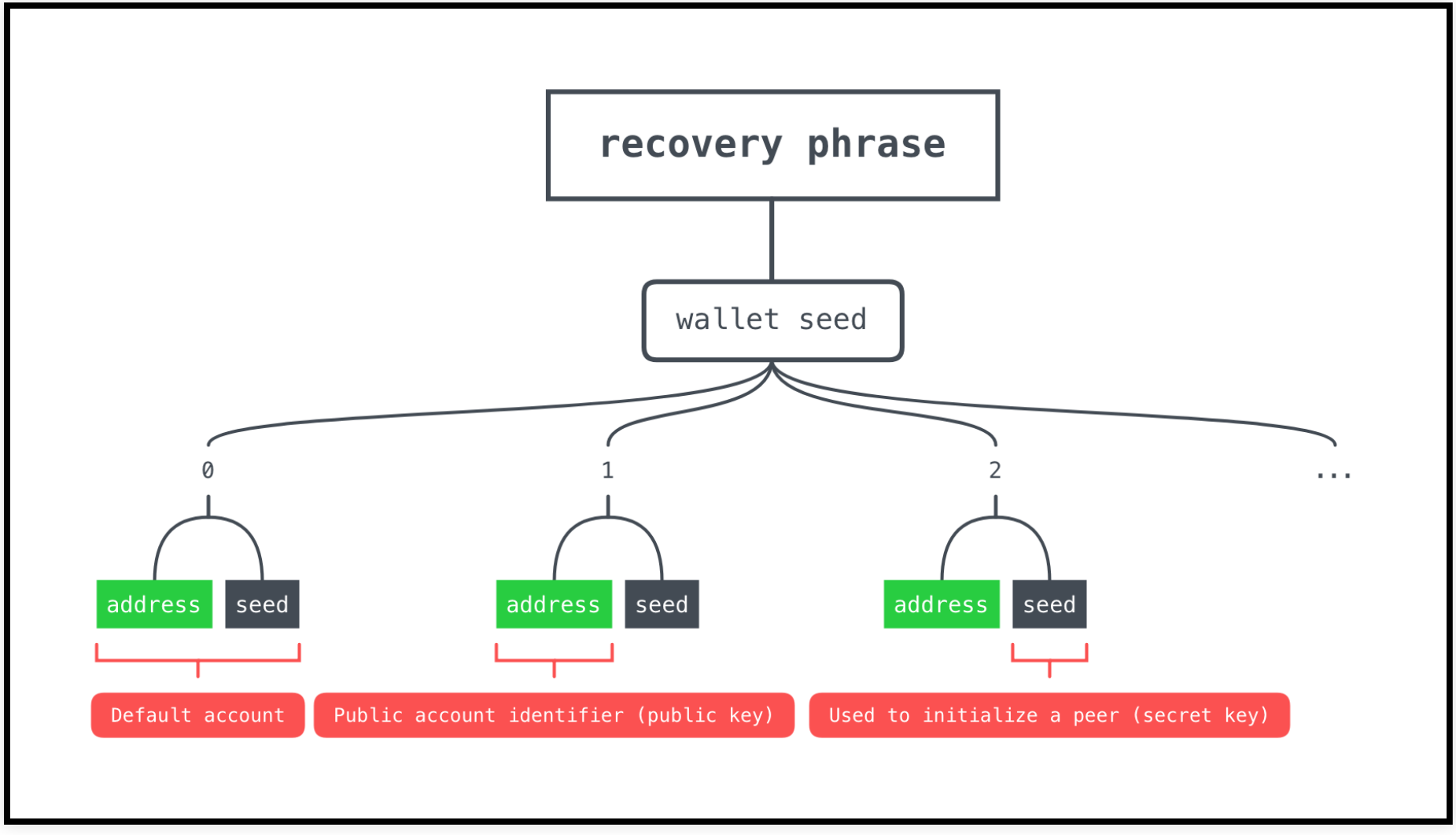
How does Textile do it?

- *Identifying individuals* done via data wallets & accounts
- *Rules* defined using schemas
- *Shared record* & *communication* done via threads
- *Game environment* provided via client libraries
    - Today we'll play with cmdline

Games are about people



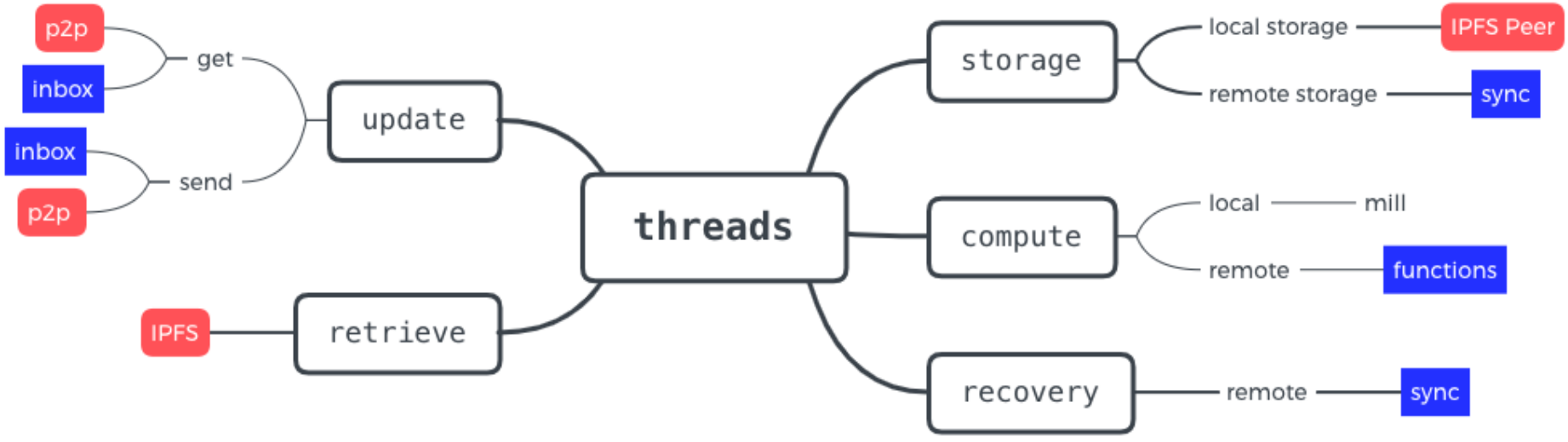
Wallet & Accounts



Games are about connections

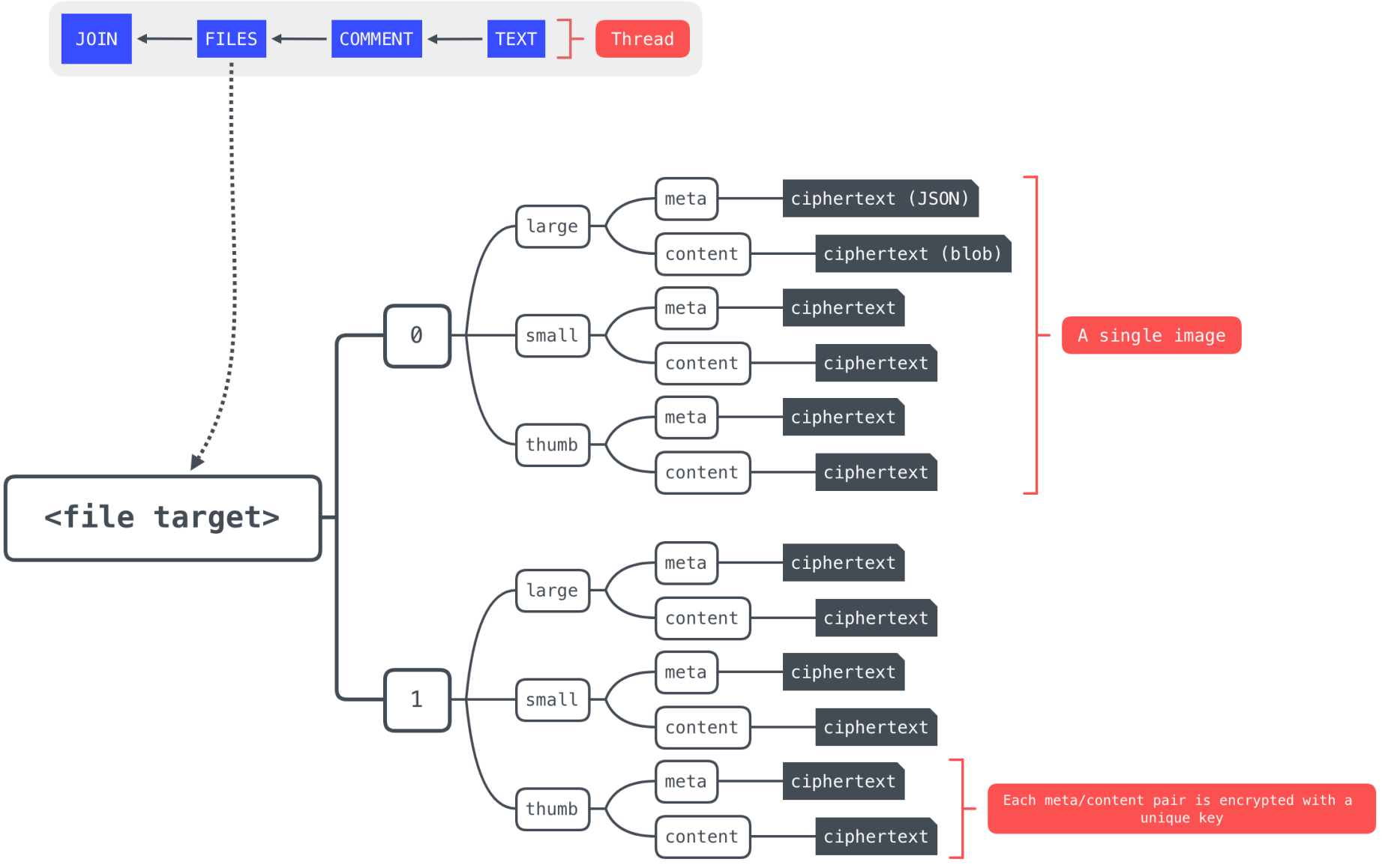
Threads

- Decentralized database layer that supports…
  - Replication (*who's it?*)
  - p2p updates (*tag you're it!*)
  - Conflict resolution (*no, you're it!*)
  - Queries (*wait, who's it?*)
  - Access controls (*can I play too?*)
  - Offline edits, and more…

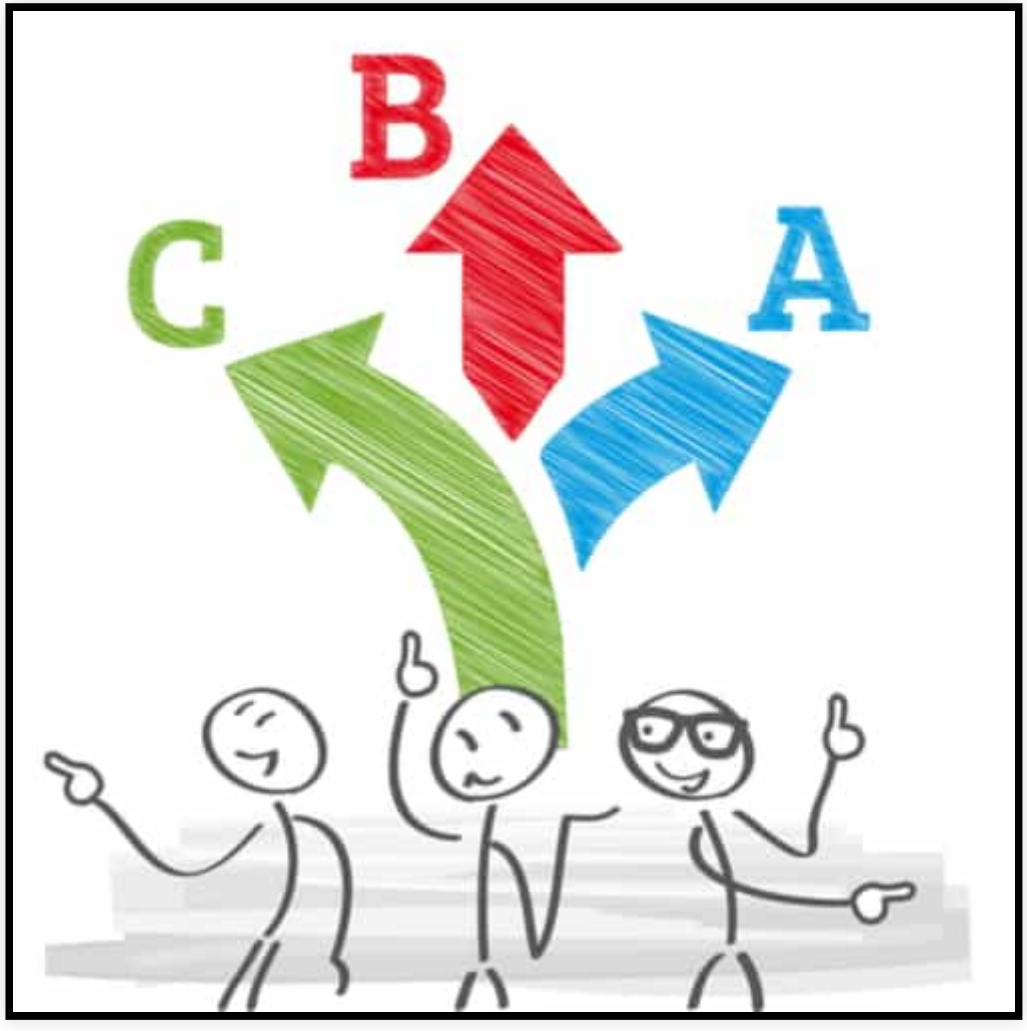

p2p
inbox
get
update
inbox
p2p
send
IPFS
retrieve
threads
storage
local storage
IPFS Peer
remote storage
sync
compute
local
mill
remote
functions
recovery
remote
sync

Access Control

- Thread
  - Backed by its own Keypair
  - Array of immutable blocks
  - Block
    - Metadata + Content
    - Types
      - Joins, Leaves, Data, Messages, etc.
  - Locally indexed
    - Exposed via API + SDKs


JOIN
FILES
COMMENT
TEXT
Thread
<file target>
0
1
large
small
thumb
meta
content
ciphertext (JSON)
ciphertext (blob)
ciphertext
A single image
Each meta/content pair is encrypted with a unique key

Games are about rules



Schemas

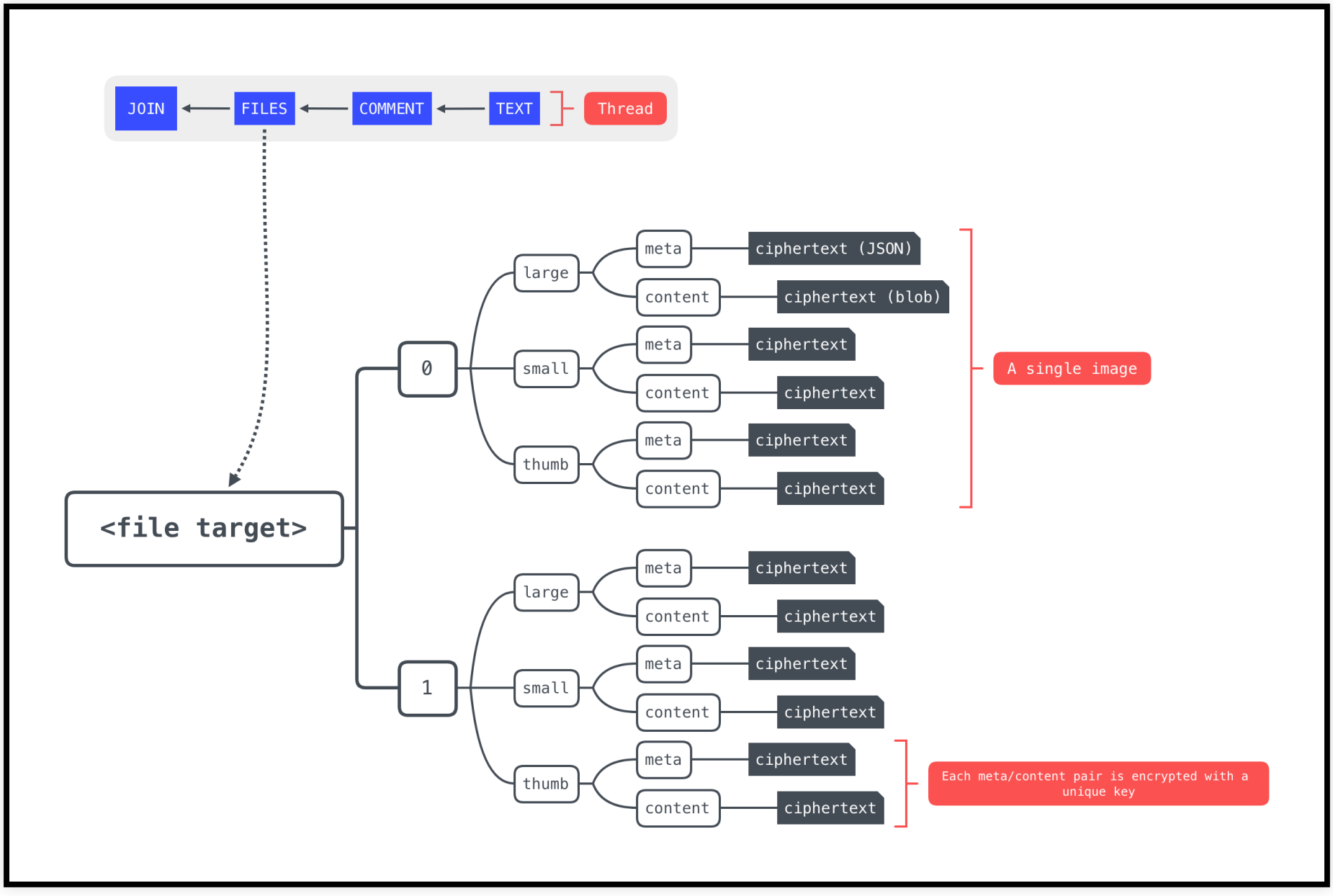
- Two main functions
  - Define a Thread's data DAG structure
  - Define the order of mills (transforms) needed to produce this structure

```
// Schema definition (for media in Textile Photos)
{
  "name": "media",
  "pin": true,
  "links": {
    "large": {
      "use": ":file",
      "mill": "/image/resize",
      "opts": {
        "width": "800",
        "quality": "80"
      }
    },
    "small": {
     "use": ":file",
     "mill": "/image/resize"
```

Mills

Decentralized compute

- Three distinct purposes
  - Validate
  - Transform
  - Index


JOIN
FILES
COMMENT
TEXT
Thread
<file target>
0
1
large
small
thumb
meta
content
ciphertext (JSON)
ciphertext (blob)
ciphertext
A single image
Each meta/content pair is encrypted with a unique key

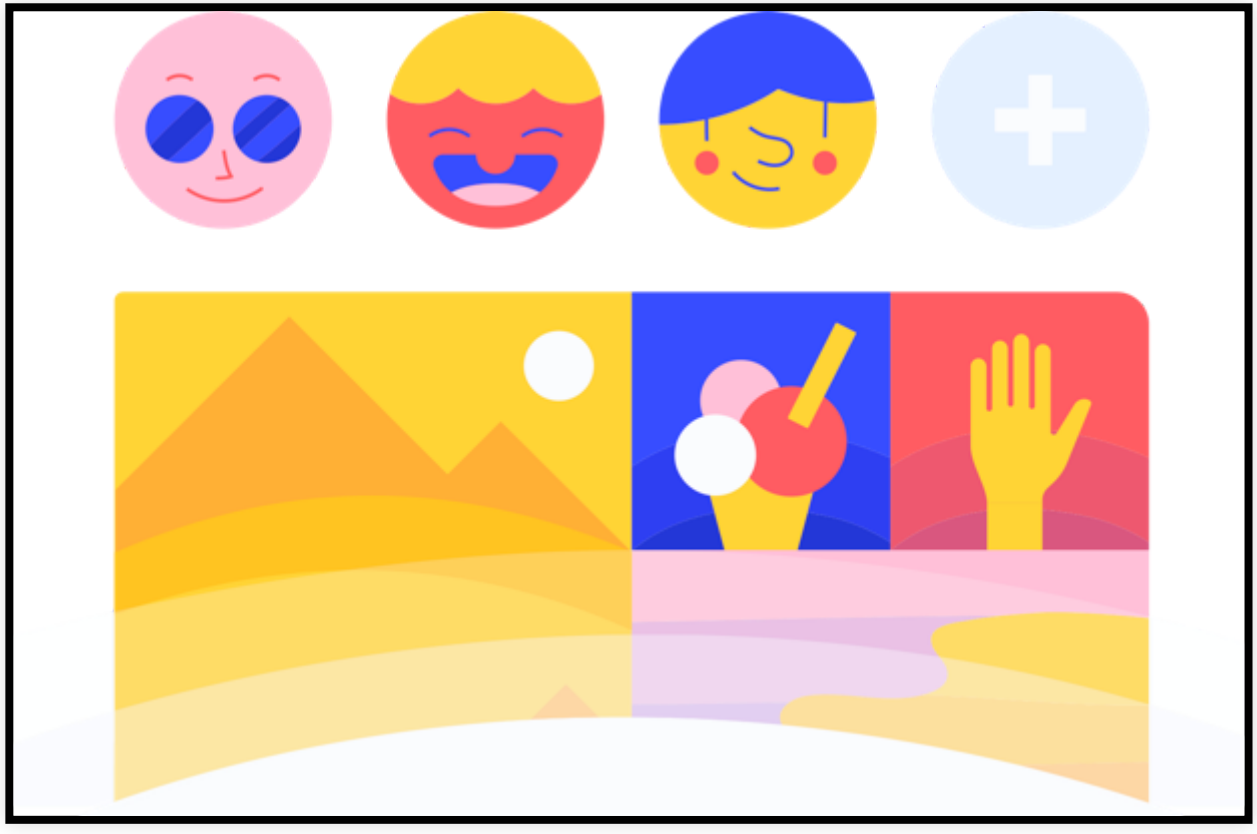
File Indexes


```
{
    "links": {
        "large": {
            "mill": "/image/resize",
            "checksum": "EqkWwbMQoSosYnu85XHpdTsM3NDKTRPk5j4RQjN6
            "source": "D4QdxGCAFnGwCHAQxrros1V6zEf78N4ugK3GwZyT5d
            "opts": "21uBAuSeQUdw5aDu5CYPxEfeiLVeuvku1T26nWtJC84C
            "hash": "QmcvoHe333KRf3tfNKrtrM7aMUVnrB4b1JyzhSFybepv
            "key": "6cCnusZVHwp6udnKv3eYhurHK6ArJyFxCYRWTUFG8ZuMw
            "media": "image/jpeg",
            "name": "clyde.jpg",
            "size": "84222",
            "added": "2019-03-17T01:20:17.061749Z",
            "meta": {
                "height": 600,
                "width": 800
```

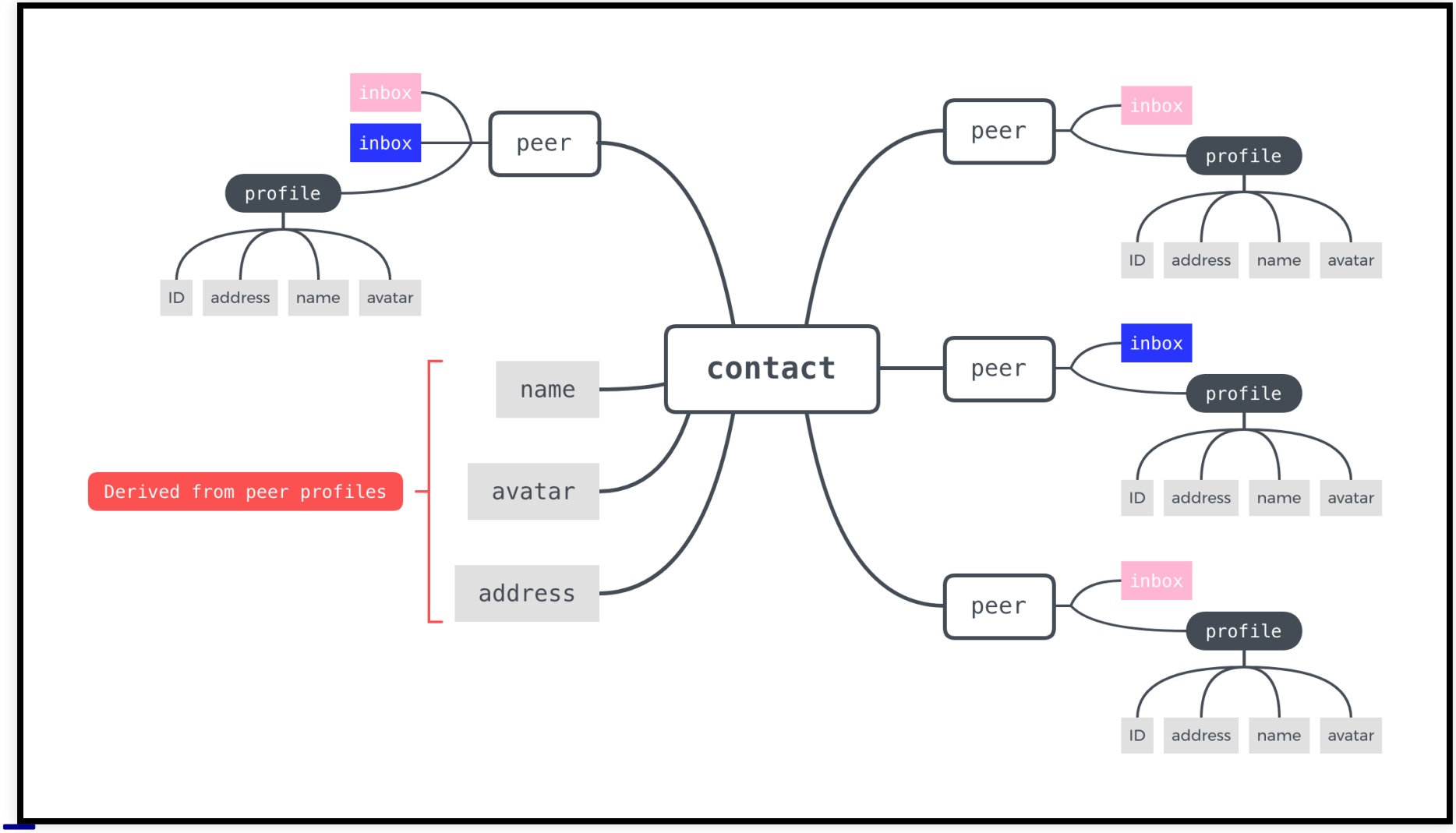
Recap

- Schemas
  - Threads
    - Blocks
      - Data

Games are about friends



Contacts


inbox
inbox
peer
profile
ID
address
name
avatar
peer
inbox
profile
ID
address
name
avatar
contact
name
avatar
address
Derived from peer profiles
peer
inbox
profile
ID
address
name
avatar
peer
inbox
profile
ID
address
name
avatar


Games are meant to be
played

Setup

- Groups of ~3-4 *by OS*, or cats vs dogs, or …
- What you'll (definitely) *need*
  - A terminal/bash/whatever
  - `go-textile` cli tools
- What you'll (maybe) *want*
  - **IPFS Tag** mobile app
  - Node.js + `npm` tooling

Install

Extras

1. https://github.com/textileio/ipfs-camp-2019
2. Clone the repo

```
👩‍💻 git clone https://github.com/textileio/ipfs-camp-2019
👨‍💻 cd ipfs-camp-2019
```

3. Get ready to play around…


COFFEE
BREAK


21:05
← Back
Hi! I'm free!
Lame
Boo
Send
I
Hey
Yeah
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
space
return
21:05
22:58 REMAINING. YOU ARE IT!
1. Show this screen to anyone playing and they are it.
2. Have them point their native camera at this code to notify the network.
3. Run!
Chat
21:05
waiting for node...
connected
Welcome back, Trackpad
Okay, ready for some Tag?.
You are playing in, IPFS Camp.
The game has started!
Remember, if anyone shows you their red card, you are it!
If the next screen is red... you are it!
Once you are it, you must scan their QR code to take over.
Tap "enter" below to find out who is it!
Ready... set...
Uh oh: node is stopped
connected
Welcome back, Trackpad
Watch out, game on!
You are playing in, IPFS Camp.
Enter

Start

```
👩‍🔒 textile wallet create
```

```
-----------------------------------------------------
| xxxx xxxx xxx xxx xxxx xxxx xxx xxx xxx xxxx xxx xxx |
-----------------------------------------------------
WARNING! Store these words above in a safe place!
WARNING! If you lose your words, you will lose access to data in
WARNING! Anyone who has access to these words can access your wal

Use: `wallet accounts` command to inspect more accounts.

--- ACCOUNT 0 ---
Pxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Sxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Init + Run

```
👩‍💻 textile init Sxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
Initialized account with address Pxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
👩‍💻 textile daemon
```

```
go-textile version: vx.x.x
Repo version: xx
Repo path: /path/to/.textile/repo
API address: 127.0.0.1:40600
Gateway address: 127.0.0.1:5050
System version: amd64/{darwin,linux,windows}
Golang version: go1.12.x
PeerID:  12D3Kxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Account: Pxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Cafes

```
👩‍🔧 textile cafe add 12D3KooWGN8VAsPHsHeJtoTbbzsGjs2LTmQZ6wFKvuPicl
```


```
{
    "access": "xxx",
    "cafe": {
        "address": "Pxxx",
        "api": "v1",
        "node": "0.5.3",
        "peer": "12D3Kxxx",
        "protocol": "/textile/cafe/1.0.0",
        "url": "https://us-west-dev.textile.cafe"
    },
    "exp": "2019-07-26T10:25:19.333555816Z",
    "id": "12D3Kxxx",
    "refresh": "xxx",
    "rexp": "2019-08-23T10:25:19.333555816Z",
    "subject": "12D3Kxxx",
    "type": "JWT"
```

Profile

```
👩‍💻 textile profile get
```

```
{
    "id": "12D3KooWCMVLfMV8uzYpFN38qn2eMs48tAuHdVZdj3aF6nex6zay",
    "address": "P8wW5FYs2ANDan2DV8D45XWKtFFYNTMY8RgLCRcQHjyPZe5j"
    "created": "2019-04-19T21:44:46.310082Z",
    "updated": "2019-04-19T21:44:46.310082Z"
}
```

```
👩‍💻 textile profile set name "Carson"
```

```
ok
```

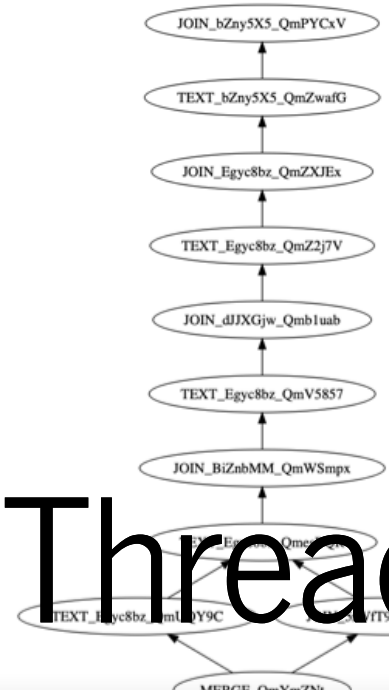
```
👩‍💻 textile profile set avatar "path/to/an/image"
```

```
ok
```

Account

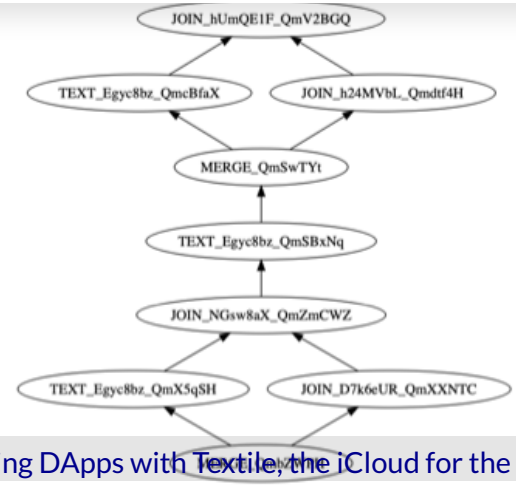
```
🤷‍♀️ textile account get
```


```
{
    "address": "Pxxx",
    "name": "Carson",
    "avatar": "Qmhash",
    "peers": [
        {
            "id": "12D3Kxxx",
            "address": "Pxxx",
            "name": "Carson",
            "avatar": "Qmhash",
            "created": "2019-04-19T21:44:46.310082Z",
            "updated": "2019-04-20T00:31:34.699845Z"
        }
    ]
}
```



Threads

```
{
    "name": "blob",
    "pin": true,
    "mill": "/blob"
}
```



```
👨‍💻 textile threads add "Name" --blob --key="ipfs.camp.tag"
```


```
{
    "block_count": 1,
    "head": "Qmhash",
    "head_block": {
        "author": "12D3Kxxx",
        "date": "2019-06-14T21:55:44.358843Z",
        "id": "Qmhash",
        "parents": [],
        "thread": "12D3Kxxx",
        "type": "JOIN",
        "user": {
            "address": "Pxxxx",
            "name": "carson"
        }
    },
    "id": "12D3Kxxxx"
```

Data

```
👩‍💻 echo "mmm, bytes..." | textile files add <thread-id>
```

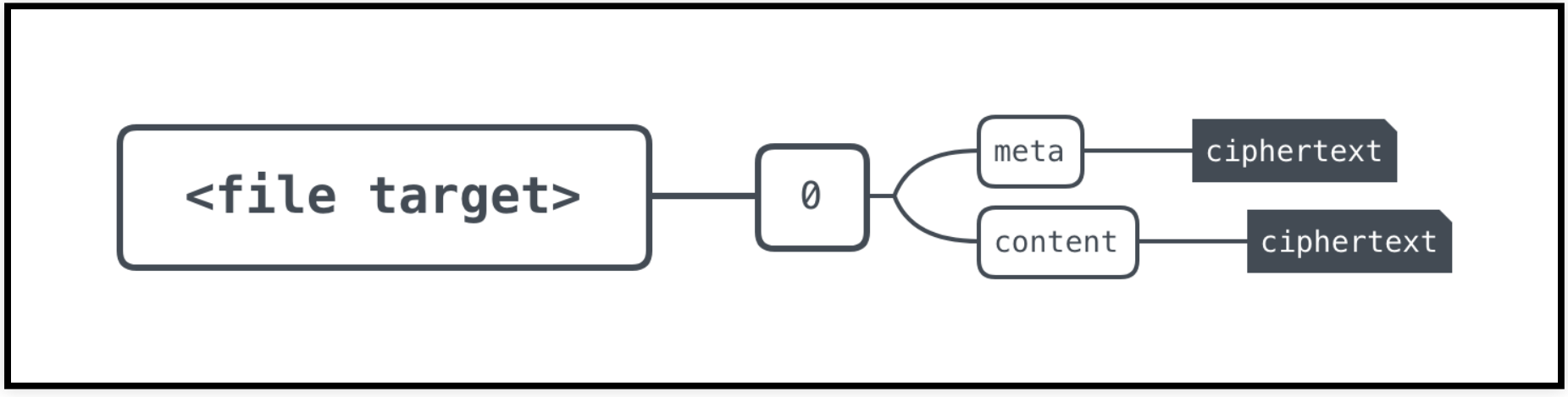

```
{
    "block": "Qmhash",
    "target": "Qmhash",
    "date": "2019-06-14T21:58:14.375745Z",
    "user": {
        "address": "Pxxx",
        "name": "carson"
    },
    "files": [
        {
            "file": {
                "mill": "/blob",
                "checksum": "xxx",
                "source": "xxx",
                "opts": "xxx",
                "hash": "Qmhash"
```

Encryption

```
👨‍💻 textile files keys <target-hash>
```

```
{
    "files": {
        "/0/": "xxx"
    }
}
```

DAGs



Rules

```
{
    "name": "cmd-line-tag",
    "mill": "/json",
    "plaintext": true,
    "json_schema": {
        "title": "CMD Line Tag Mechanics",
        "description": "Possible events in cmd line tag.",
        "type": "object",
        "required": [ "event" ],
        "properties": {
            "event": {
                "type": "string",
                "description": "event type identifier"
            },
            "target": {
                "type": "string"
```

Schemas

```
🤦 textile threads add "Tag" --schema-file=/path/to/tag.json --ty
```


```
{
    "block_count": 1,
    "id": "12D3Kxxx",
    "initiator": "Pxxx",
    "key": "xxx",
    "name": "Tag",
    "peer_count": 1,
    "schema": "Qmhash",
    "schema_node": {
        "json_schema": {
            "description": "Possible events in cmd line tag.",
            "properties": {
                "event": {
                    "description": "event type identifier",
                    "type": "string"
                }
```

Adding

```
👨‍💻 echo '{ "event": "tag", "target": "<address>" }' | textile file
```


```
{
    "block": "Qmhash",
    "target": "Qmhash",
    "date": "2019-06-18T17:51:33.424170Z",
    "user": {
        "address": "Pxxx",
        "name": "carson"
    },
    "files": [
        {
            "file": {
                "mill": "/json",
                "checksum": "xxx",
                "source": "xxx",
                "opts": "xxx",
                "hash": "Qmhash"
```

Friends

- Create a peerpad to share thread information
- Use me:
  - `P4YL7j6fGAwA8WUo9vLGEaDDoKUFrcWEZj`
- Invite them directly

```
👩‍💻 textile invite create <thread-id> --address=<neighbor-peer-id
```

```
ok
```

- or create external invite

```
👨‍💻 textile invites create <thread-
```

```
{
  "id": "Qmhash",
  "inviter": "Pxxx",
  "key": "xxx"
}
```

Messages

```
🤷‍♀️ textile messages add <thread-id> "game on"
```


```
{
    "block": "Qmhash",
    "body": "Game on",
    "comments": [],
    "date": "2019-06-14T21:37:37.053367Z",
    "likes": [],
    "user": {
        "address": "Pxxx",
        "name": "carson"
    }
}
```

Explore!

- List thread blocks (`textile thread blocks`)
- List contacts (`textile contacts list`)
- (more) messages (`textile messages --help`)
- (more) data (`textile files --help`)
- View a feed (`textile feed`)
- Observe real-time updates (`textile observe`)
- Get help (`textile <sub command> --help`)

Group Game

Join

```
👩‍💻 textile invites accept <invite-id> --key=<invite-key>
```

Check

```
👩‍💻 sh am-i-it.sh <thread-id>
```

Play

```
👩‍💻 sh tag-peer.sh <thread-id> <address>
```

Hack and break things!

- If you want to try out the mobile app
  - http://t.txtl.us/
- If you want to hack on the mobile app
  - …/README.md
- If you want to hack on a tag leader-board
  - `cd demo-leaderboard` & follow directions
- If you want to play from cli
  - `cd cli` & follow directions

Learn More

- Concepts
- Take the Tour
- @textileio
  - Slack
  - GitHub
  - Twitter
  - Blog