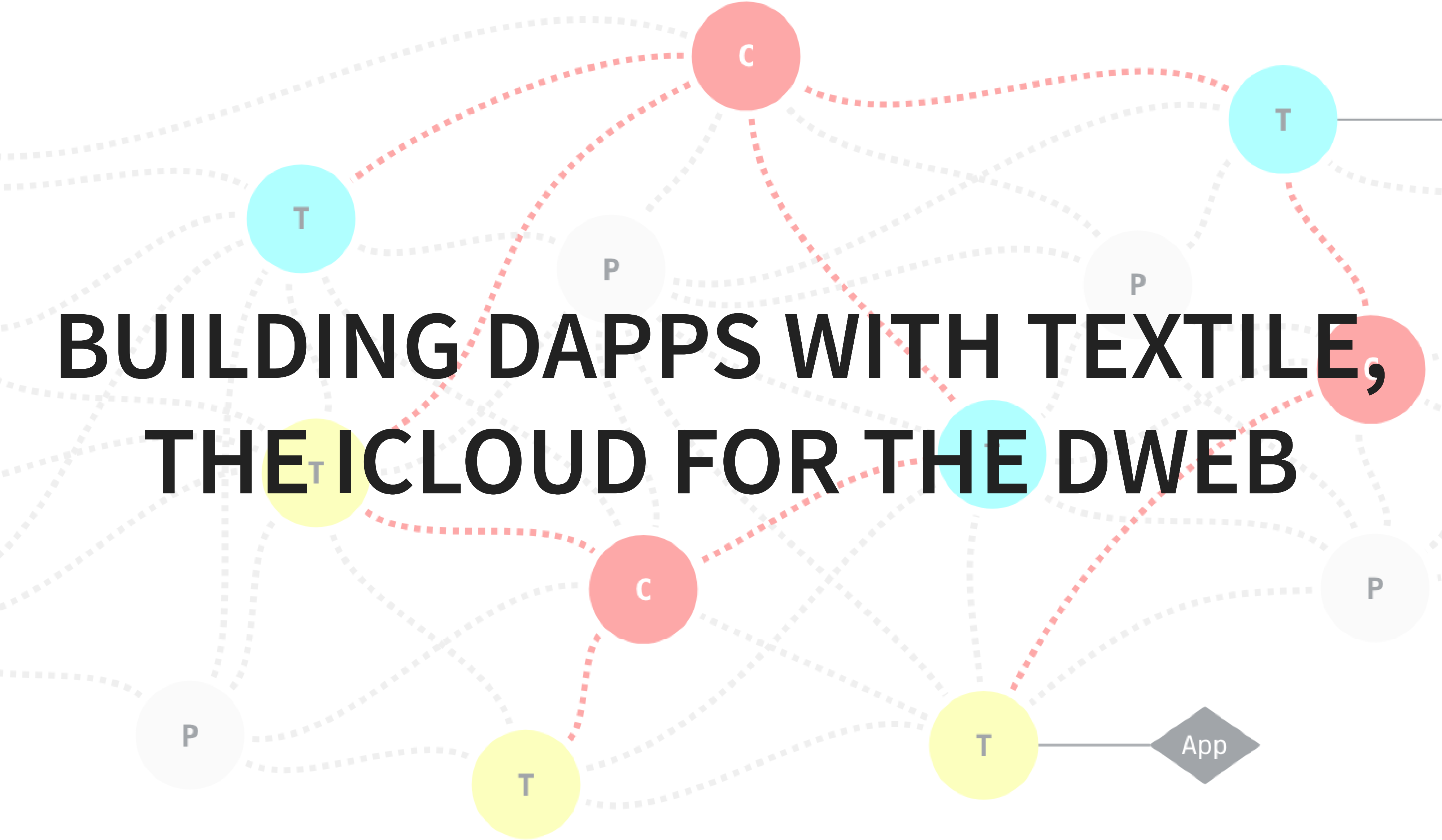
BUILDING DAPPS WITH TEXTILE, THE ICLOUD FOR THE DWEB



WELCOME!

Placeholder Image

TEXTILE

...A SET OF TOOLS AND TRUST-LESS INFRASTRUCTURE FOR BUILDING CENSORSHIP RESISTANT AND PRIVACY PRESERVING APPLICATIONS

INSTRUCTORS

Carson | Andrew | Benjamin

Sander | Aaron | Thomas

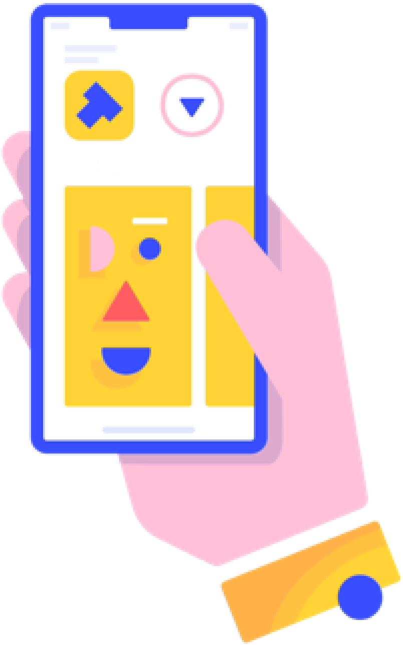
OUTLINE

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

STRUCTURE

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

DEMO!



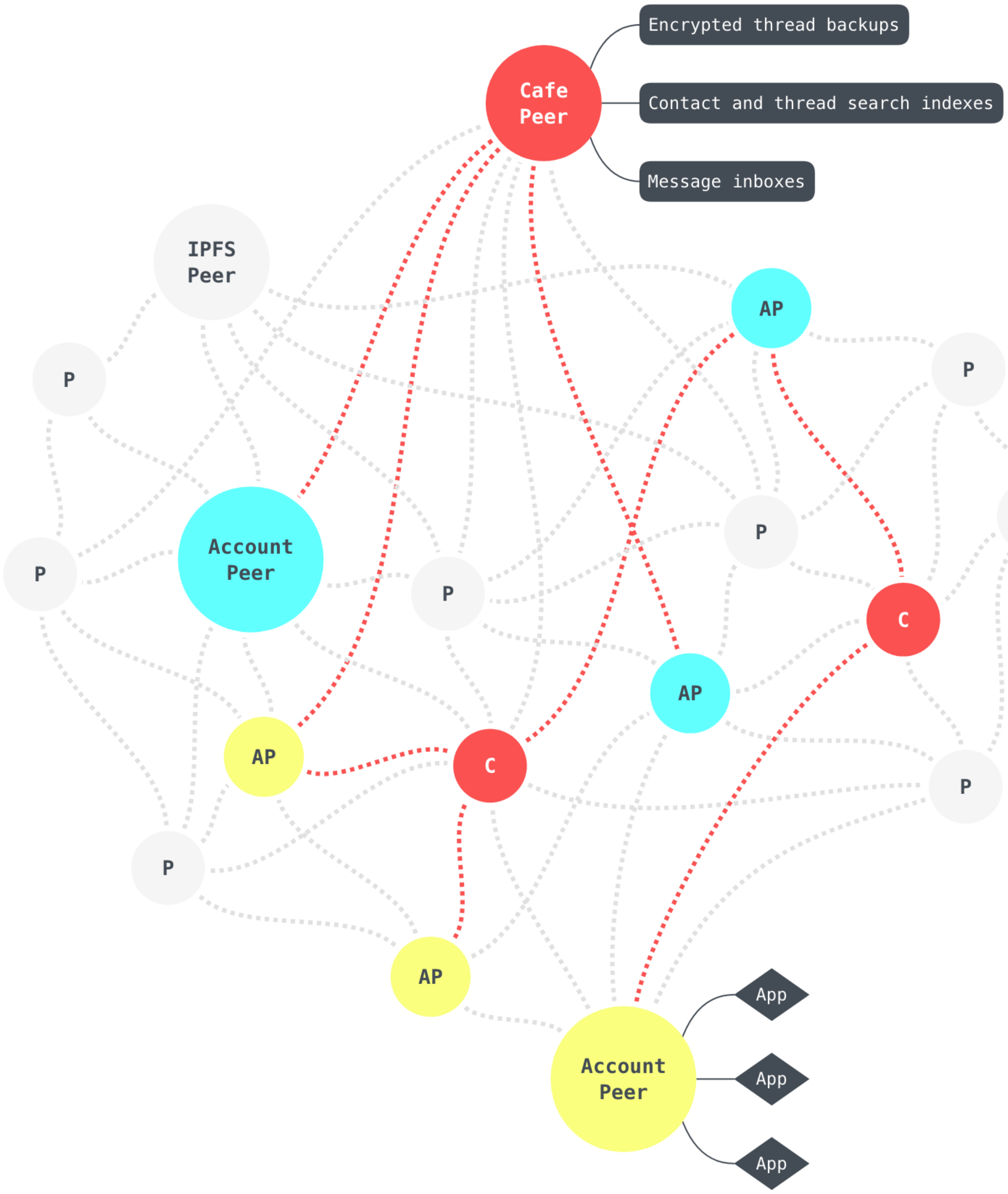
ANATOMY OF A GAME

GAMES ARE ABOUT FUN

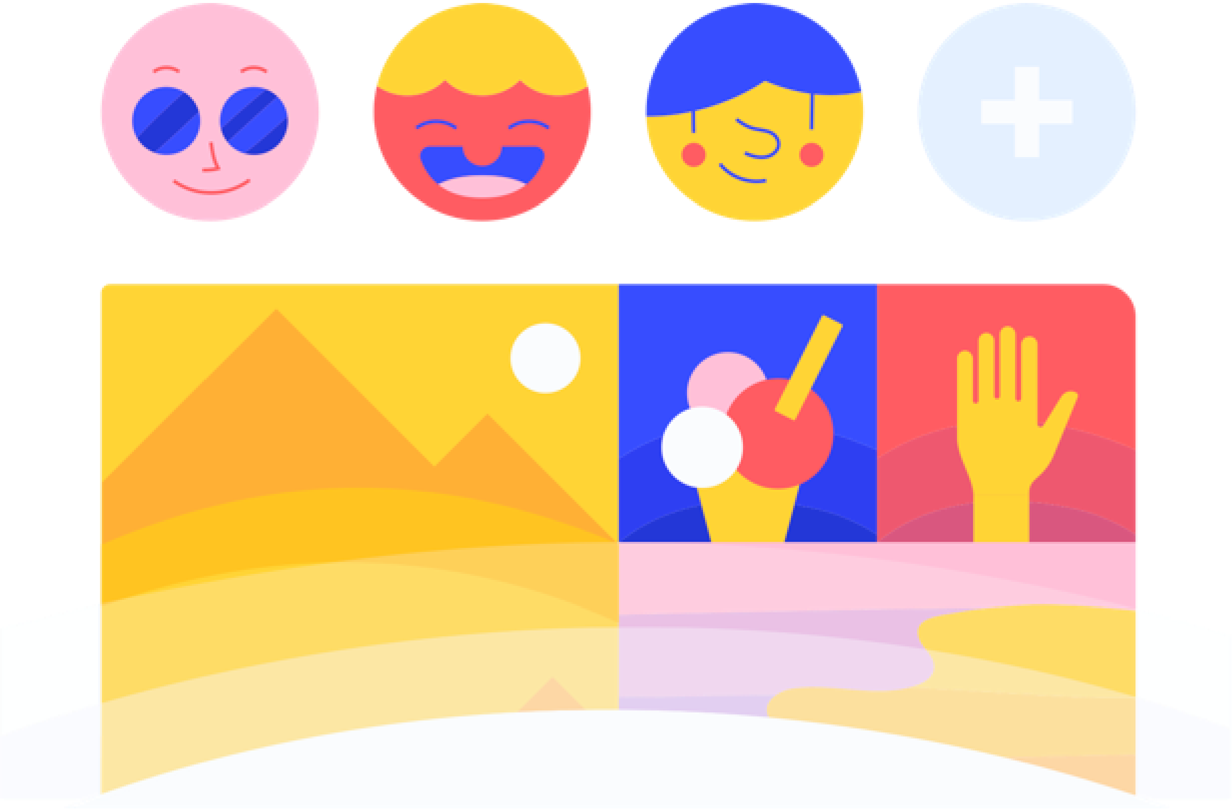
- Only a **few things** needed for IPFS Tag...
  - Way to uniquely identify peers
  - Set of rules and (logically centralized) game environment
  - Means of communicating actions (i.e. *tagging*) and tracking (i.e. *who's it*)
  - Simple user interface

FUN WITH TEXTILE

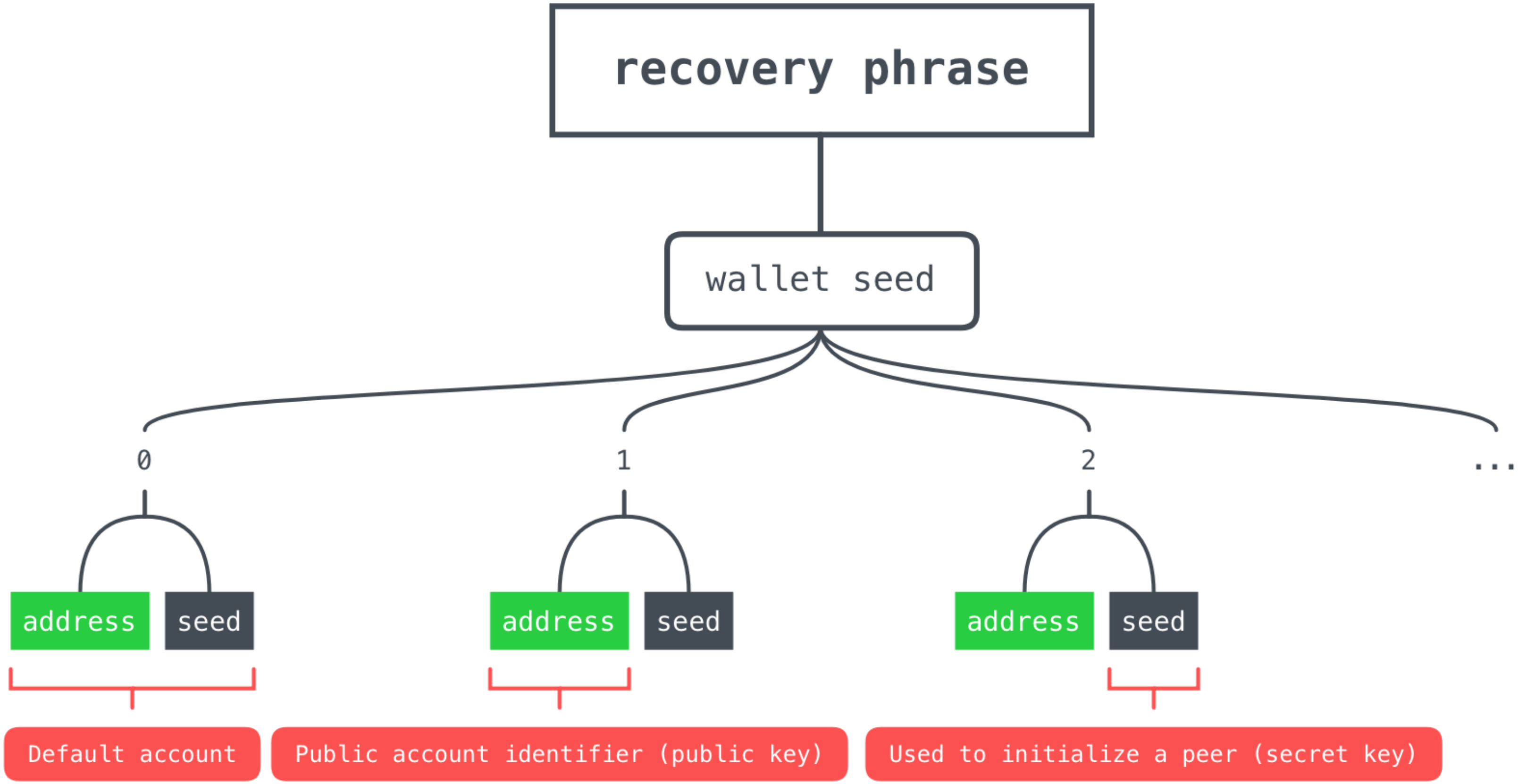
- Textile-based dapps meet requirements with some basic concepts…
  - *Identifying peers* done via data wallets & accounts
  - *Rules* & *game environment* are defined using schemas
  - *Communication* & *score-keeping* done via threads
  - User interface provided via client libraries
    - Today we’ll play with cmdline and javascript


Cafe Peer
Encrypted thread backups
Contact and thread search indexes
Message inboxes
IPFS Peer
Account Peer
AP
P
C
Account Peer
App
App
App

GAMES ARE ABOUT PEOPLE



THE WALLET



recovery phrase

wallet seed

0

1

2

...

address seed

address seed

address seed

Default account

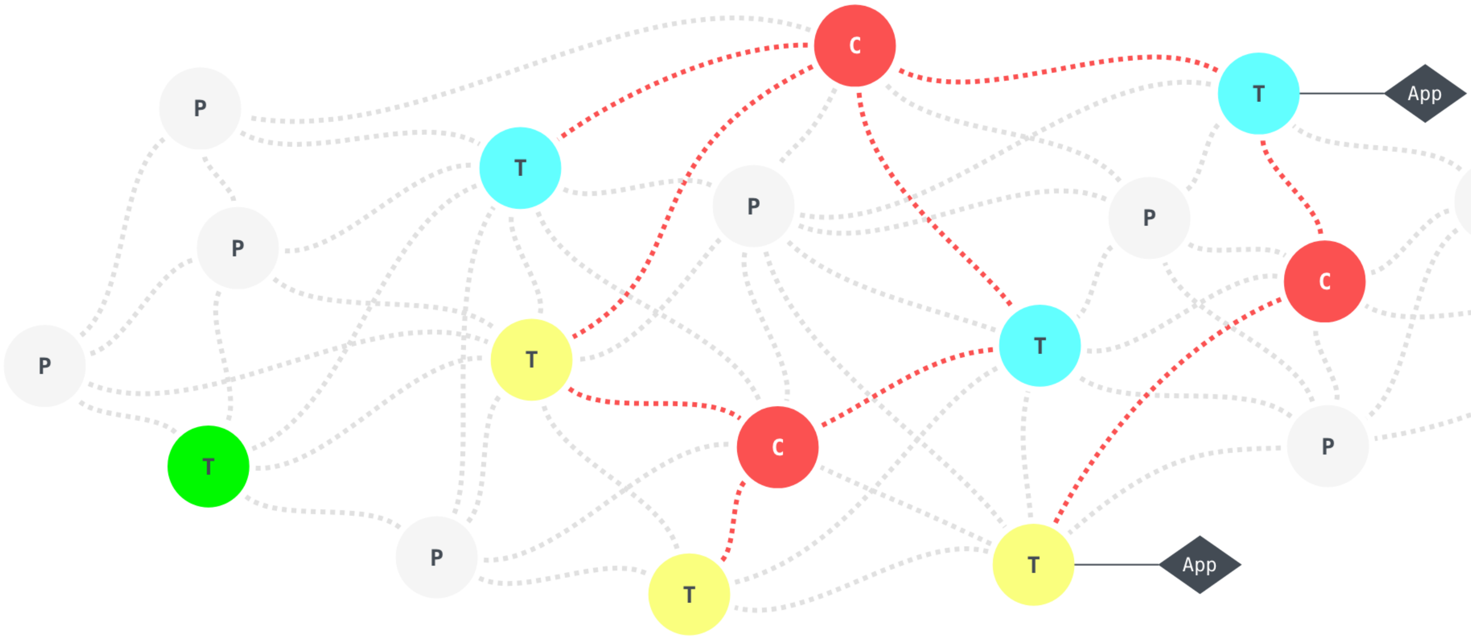
Public account identifier (public key)

Used to initialize a peer (secret key)

ACCOUNTS

Placeholder Image

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...

BACKGROUND

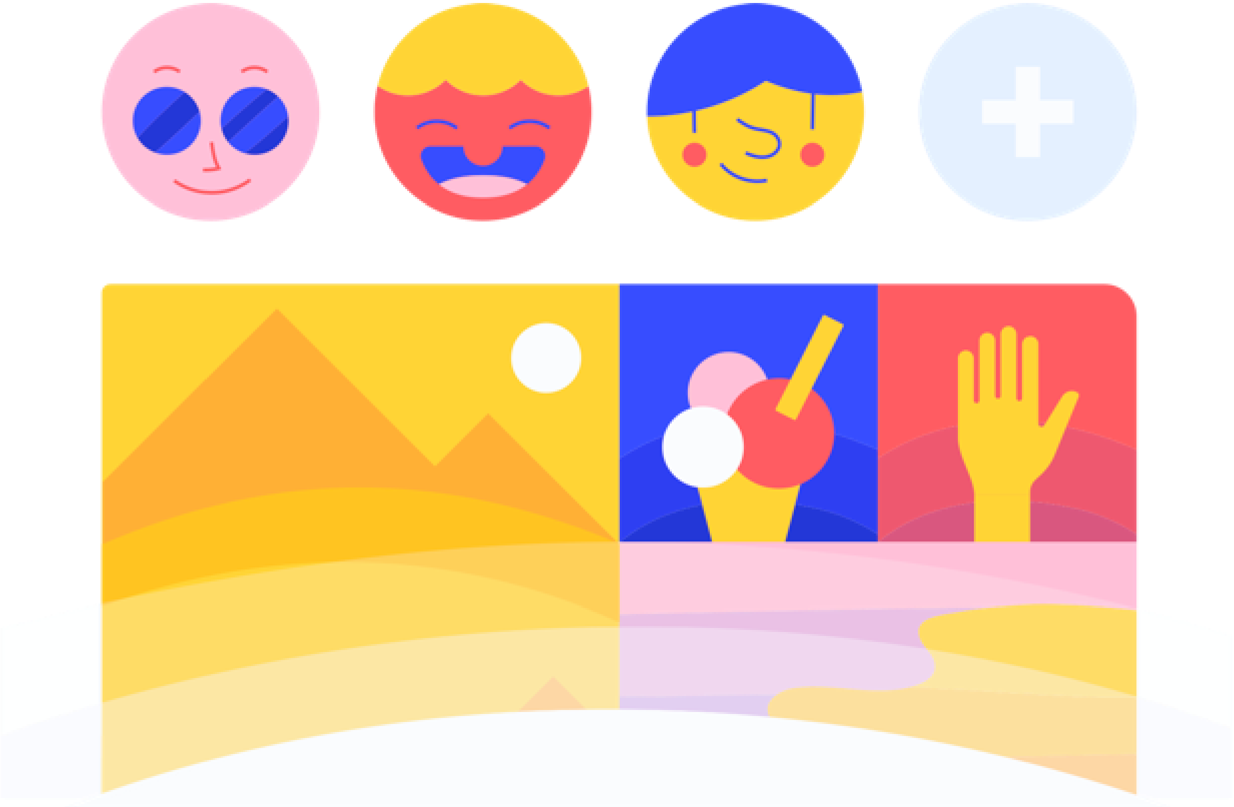
REQUIREMENTS

ACCESS CONTROL

BLOCKS

FILES

GAMES ARE ABOUT RULES

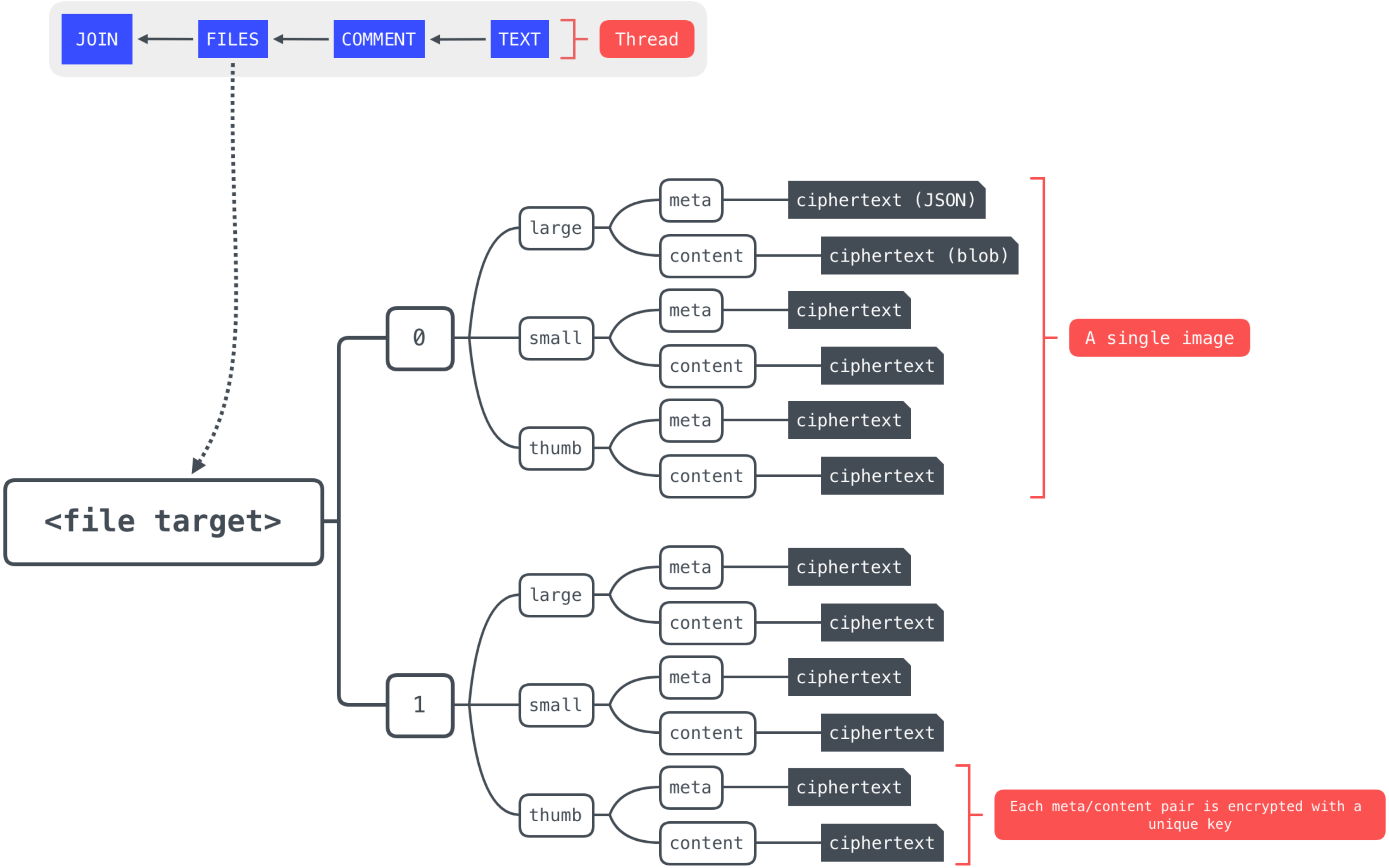


Placeholder Image

SCHEMAS

```
{
  "name": "media",
  "pin": true,
  "links": {
    "large": {
      "use": ":file",
      "mill": "/image/resize",
      "opts": {
        "width": "800",
        "quality": "80"
      }
    },
    "small": {
      "use": ":file",
```

MILLS

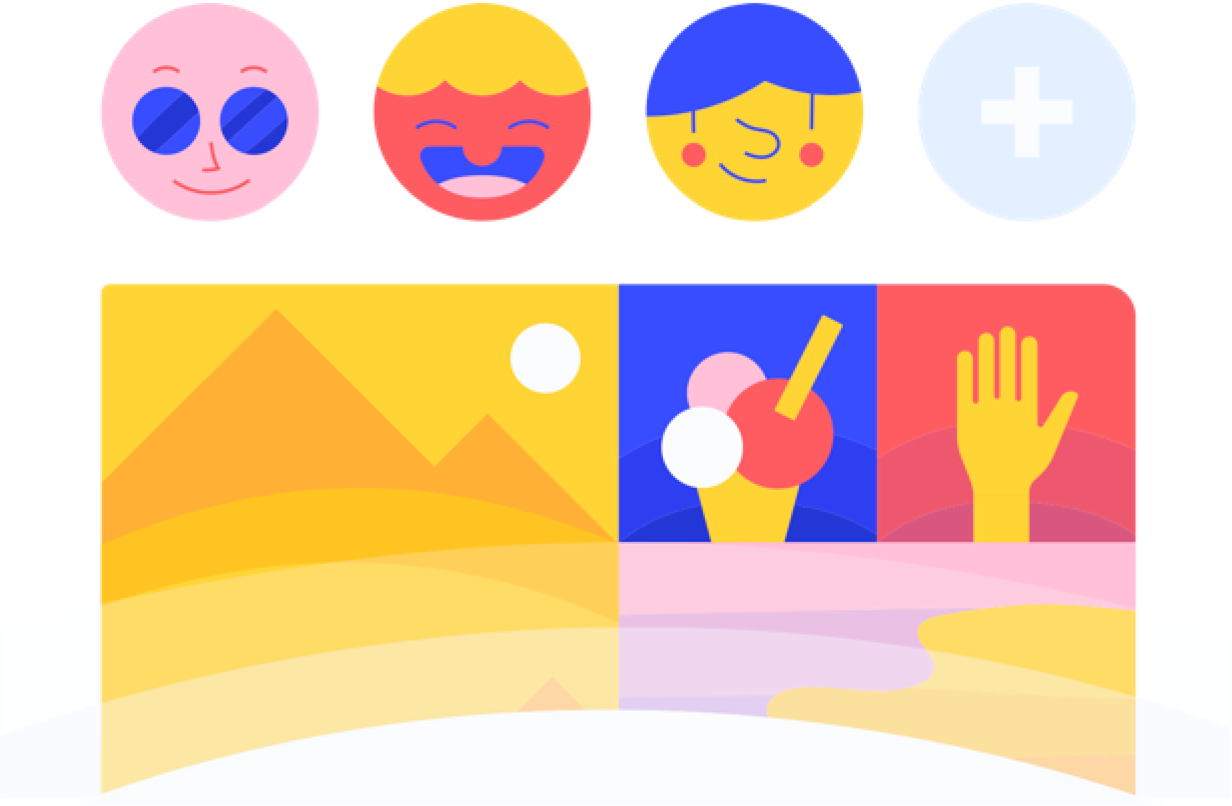

JOIN
FILES
COMMENT
TEXT
Thread
<file target>
0
large
meta
ciphertext (JSON)
content
ciphertext (blob)
small
meta
ciphertext
content
ciphertext
thumb
meta
ciphertext
content
ciphertext
A single image
1
large
meta
ciphertext
content
ciphertext
small
meta
ciphertext
content
ciphertext
thumb
meta
ciphertext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

FILE INDEXES

```
{
    "links": {
        "large": {
            "mill": "/image/resize",
            "checksum": "EqkWwbMQoSosYnu85XHpdTsM3NDKTRP
            "source": "D4QdxGCAFnGwCHAQxrros1V6zEf78N4ug
            "opts": "21uBAuSeQUdw5aDu5CYPxEfeiLVeuvku1T2
            "hash": "QmcvoHe333KRf3tfNKrtrM7aMUVnrB4b1Jy
            "key": "6cCnusZVHwp6udnKv3eYhurHK6ArJyFxCYRW
            "media": "image/jpeg",
            "name": "clyde.jpg",
            "size": "84222",
            "added": "2019-03-17T01:20:17.061749Z",
```

SUMMARY

GAMES ARE MEANT TO BE PLAYED



Placeholder Image

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

- Download and extract the latest release for your OS and architecture (or use `wget` etc...)
- macOS and Linux
  - Extract the tarball (manually or via...)

    ```
    👩‍💻 tar xvfz go-textile_0.4.0_{os}-amd64.tar.gz`)
    ```

  - Move `textile` anyplace in your `PATH` (or via...)

    ```
    👨‍💻 ./install
    ```

- Windows
  - Extract the zip file and move `textile.exe` anyplace in your `PATH`

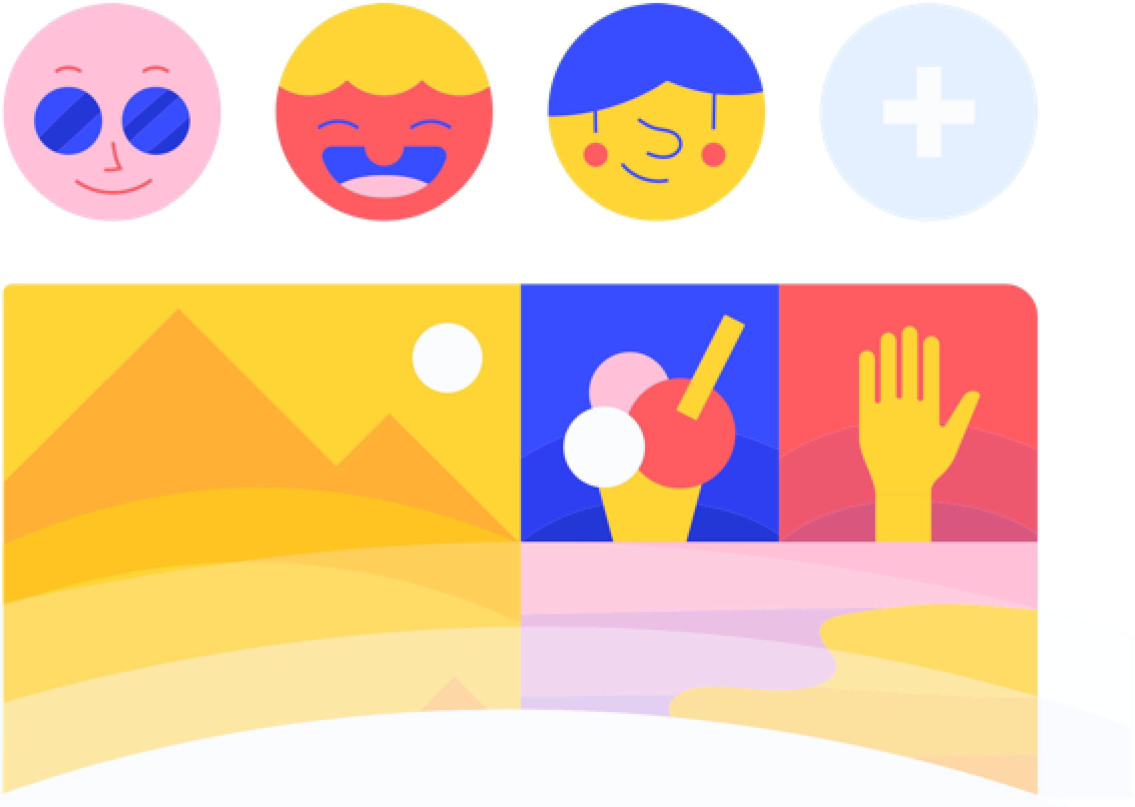
EXTRAS

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

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

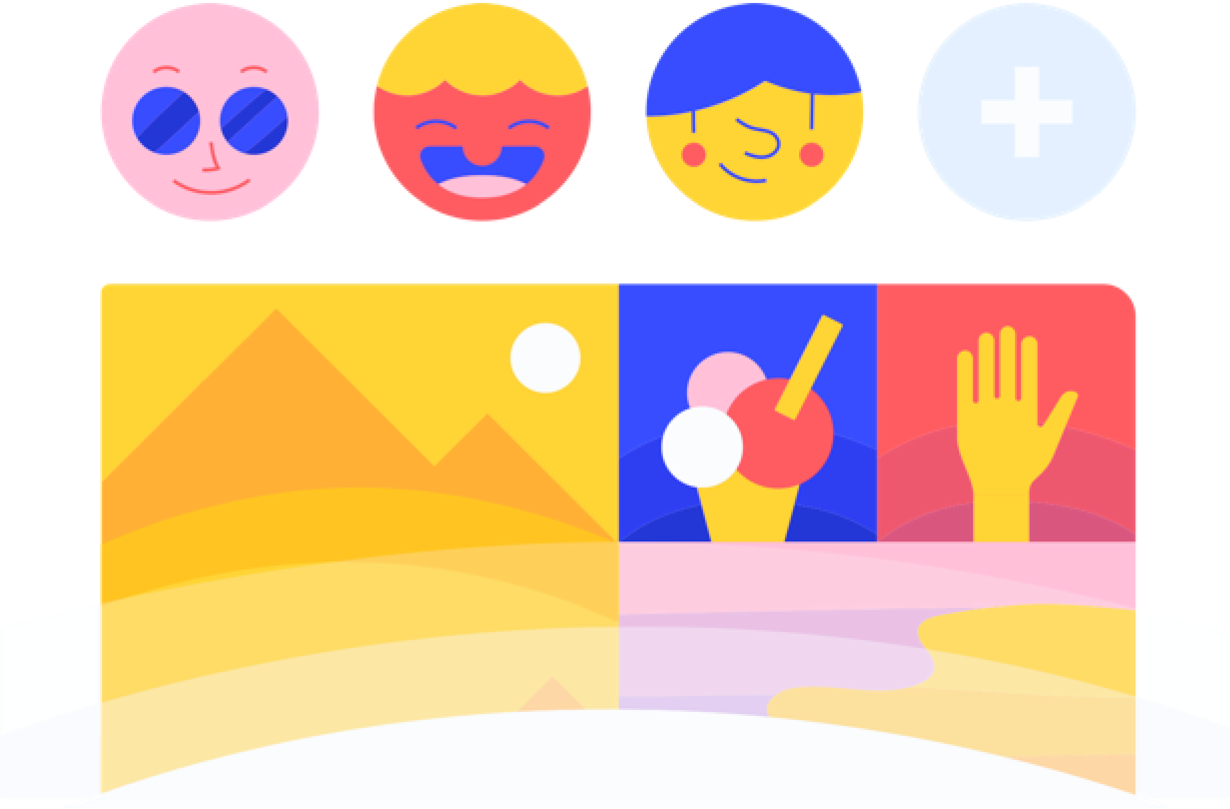
3. Get ready to play around...

BREAK!



Placeholder Image

LET'S PLAY



Placeholder Image

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
WARNING! Anyone who has access to these words can access

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

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

INIT + RUN

```
👨‍💻 textile init Sxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
Initialized account with address Pxxxxxxxxxxxxxxxxxxxxxxx:
```

```
👩‍💻 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:  12D3Kxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx:
Account: Pxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx:
```

PROFILE

```
👨‍💻 textile profile get
```

```
{
    "id": "12D3KooWCMVLfMV8uzYpFN38qn2eMs48tAuHdVZdj3aF6r
    "address": "P8wW5FYs2ANDan2DV8D45XWKtFFYNTMY8RgLCRcQI
    "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"
```

DATA

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

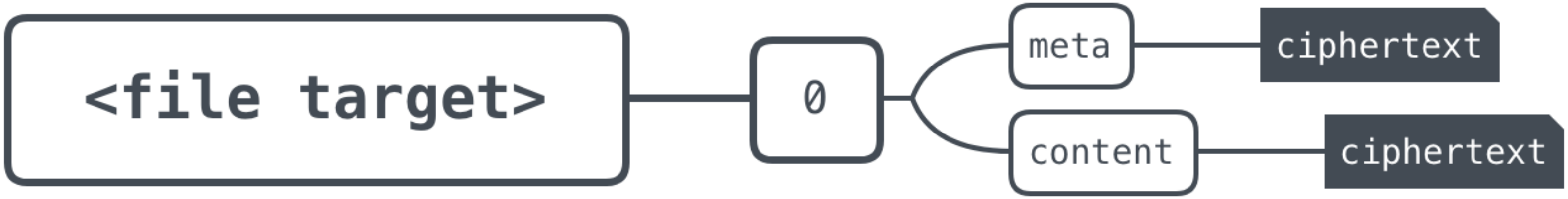
```
{
    "block": "Qmhash",
    "target": "Qmhash",
    "date": "2019-06-14T21:58:14.375745Z",
    "user": {
        "address": "Pxxx",
        "name": "carson"
    },
    "files": [
        {
            "file": {
                "mill": "/blob",
                "checksum": "xxx",
```

ENCRYPTION

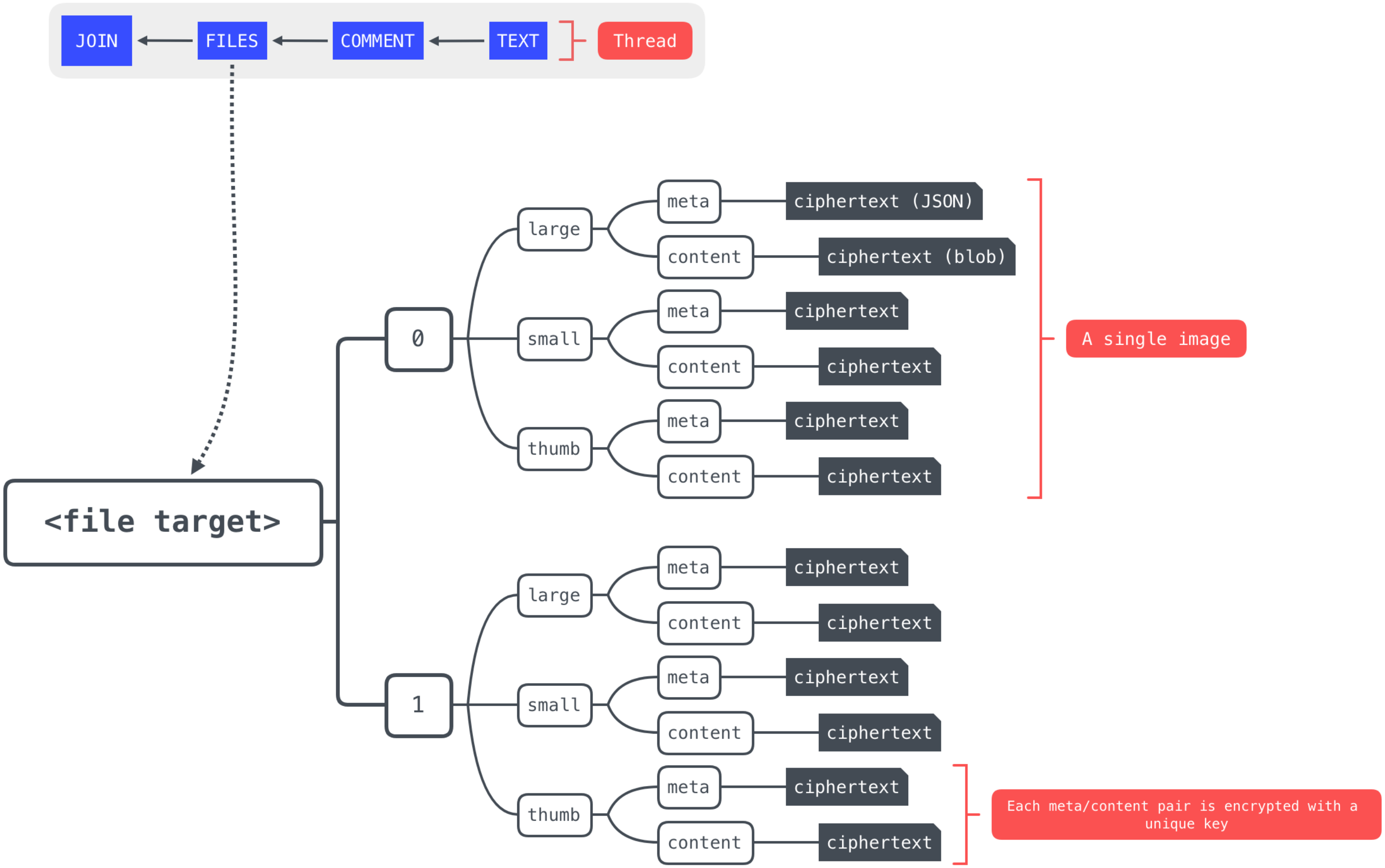
```
textile files keys Qmahash
```

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

DAGS



MEDIA


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

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"
            }
```

SCHEMAS

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

ADDING

```
echo '{ "event": "tag", "target": "'<peer-id>'" }' | tex
```

FRIENDS

```
textile invite create <thread-id> --address=<neighbor-pee
```

MESSAGES

```
👩‍💻 textile messages add -t <thread-id> "game on"
```

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

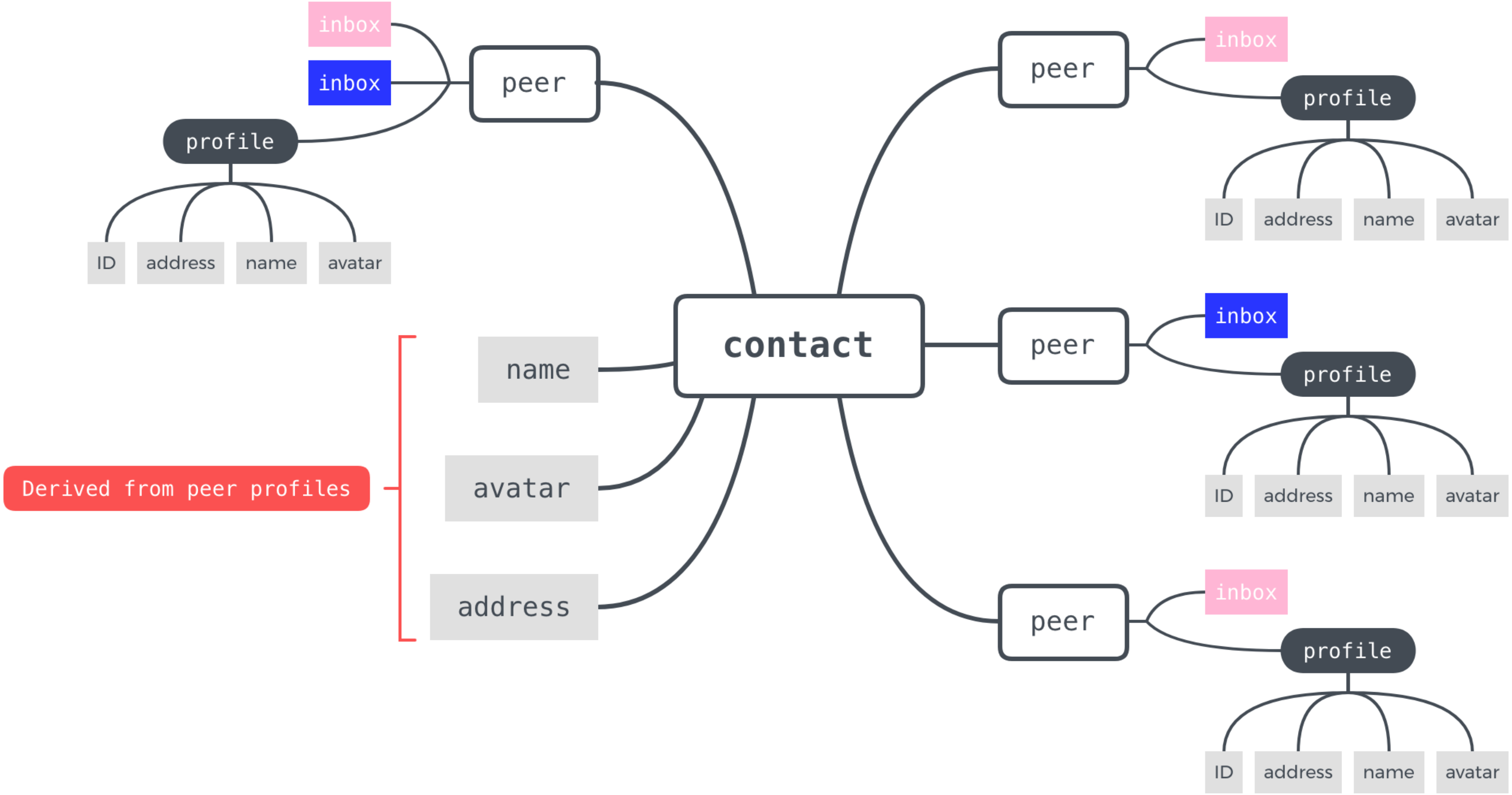
EXPLORE

- List thread blocks
- List contacts
- Add messages
- Add data

QUERY

- Feed API
- Observe API

CONTACT



inbox
inbox
peer
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
peer
inbox
profile
ID
address
name
avatar

-->