✨ UNIVERSAL HOSTLESS

SUBSTRATE

FULL STACK WEB APPS *WITHOUT A BACKEND*… AND MORE 🚀

A UNIVERSAL HOSTLESS SUBSTRATE

BROOKLYN ZELENKA, @expede

fission

A UNIVERSAL HOSTLESS SUBSTRATE

BROOKLYN ZELENKA, @expede

- Cofounder/CTO at Fission — https://fission.codes
- Functional Programming
  - Founder of the Vancouver FP meetup
  - Mainly known in FP-land for Witchcraft
  - Fission is very much informed by FP mindset
    - Universality, orthogonality, properties, &c
- PLT, VMs, Distributed Systems
- Previously an Ethereum Core Dev
- Spending a lot of time with IPFS, ECC, CRDTs, & Bloom Clocks

fission

A UNIVERSAL HOSTLESS SUBSTRATE

WHAT WE'RE COVERING

A UNIVERSAL HOSTLESS SUBSTRATE

WHAT WE'RE COVERING

- Content addressing
- Global file system
- User-controlled auth
- Portable compute

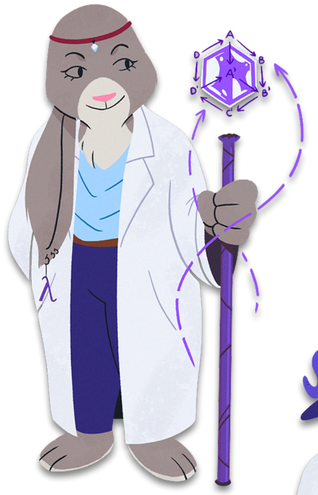
A UNIVERSAL HOSTLESS SUBSTRATE

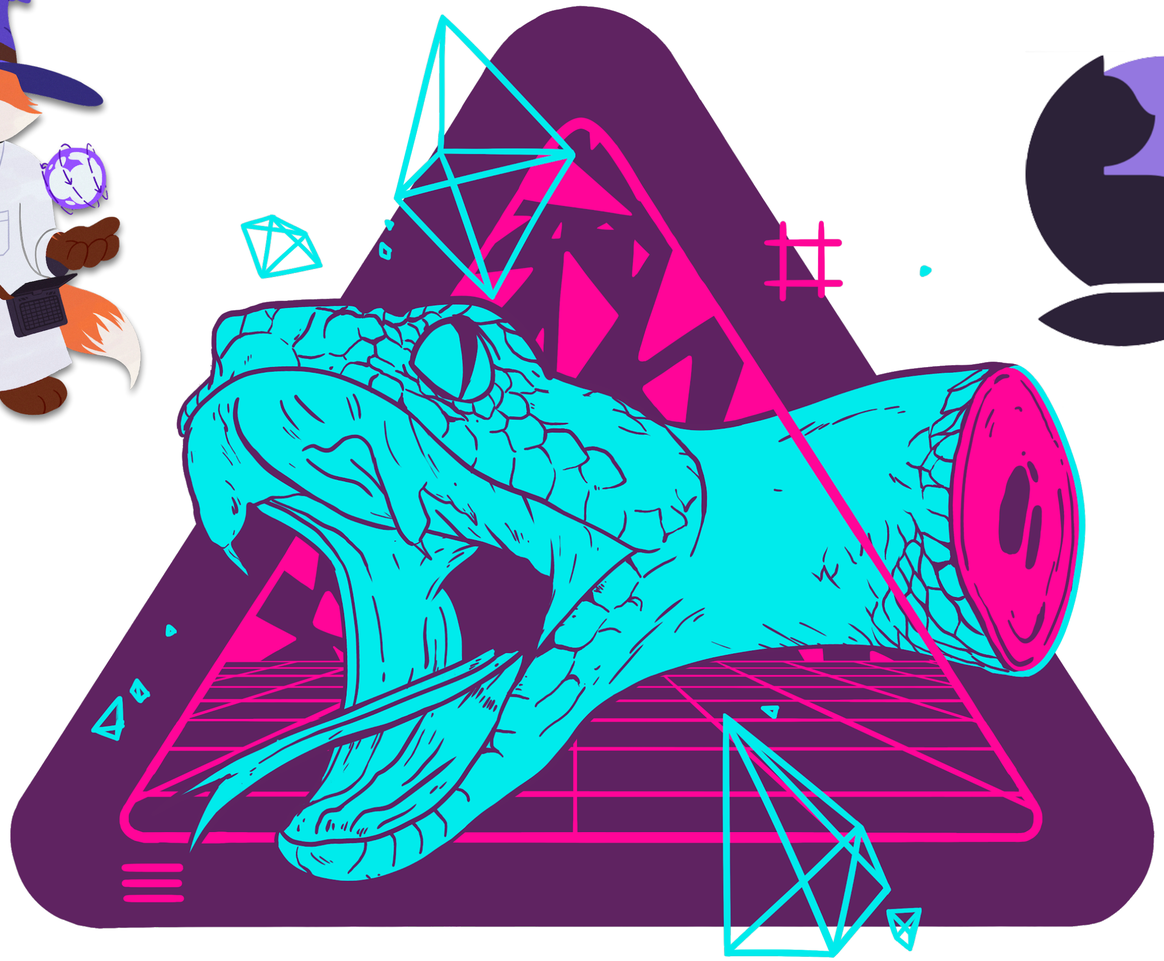
STACK

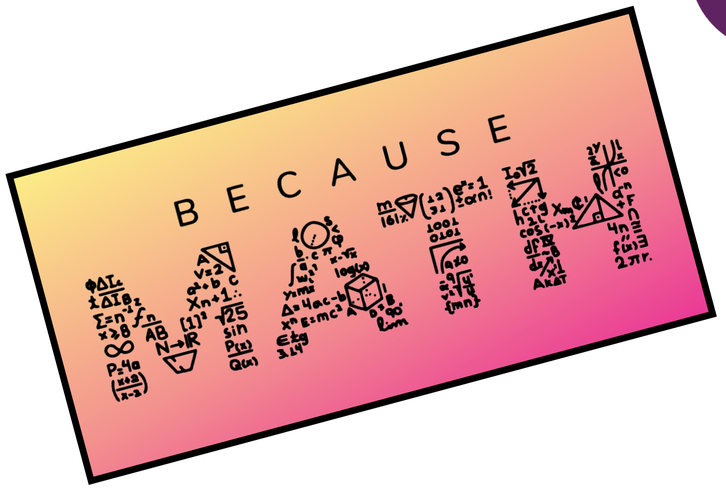
A UNIVERSAL HOSTLESS SUBSTRATE

STACK

- Server — Haskell (servant-server)
- CLI — Haskell (servant-client)
- Fission Drive (web file explorer) — Elm
- Browser-based Authorization — Elm
- Browser SDK — TypeScript
- IPFS Responsive Image Resizer — Rust (native & wasm)



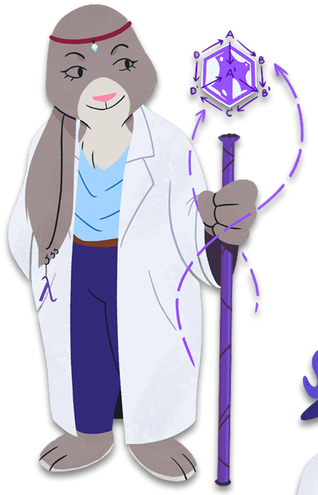


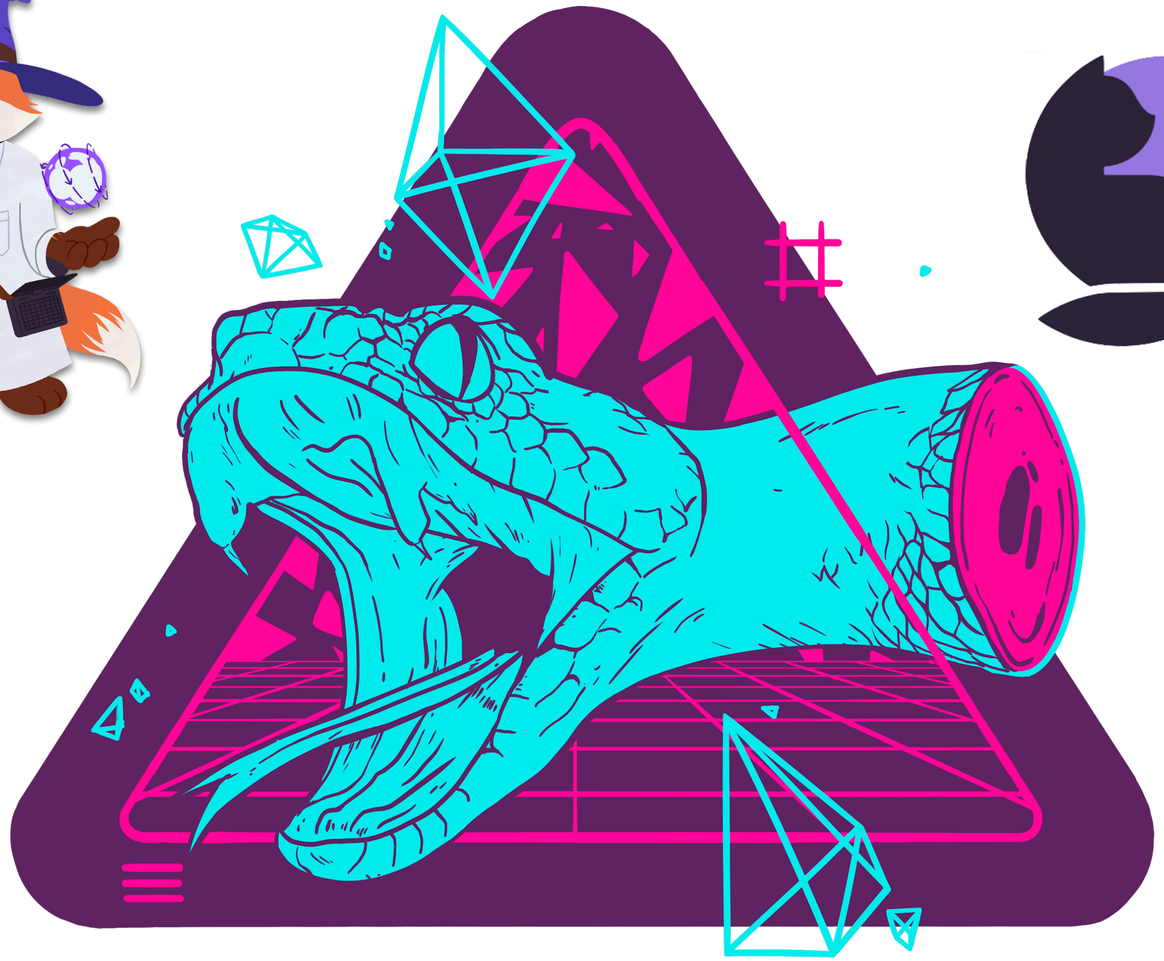


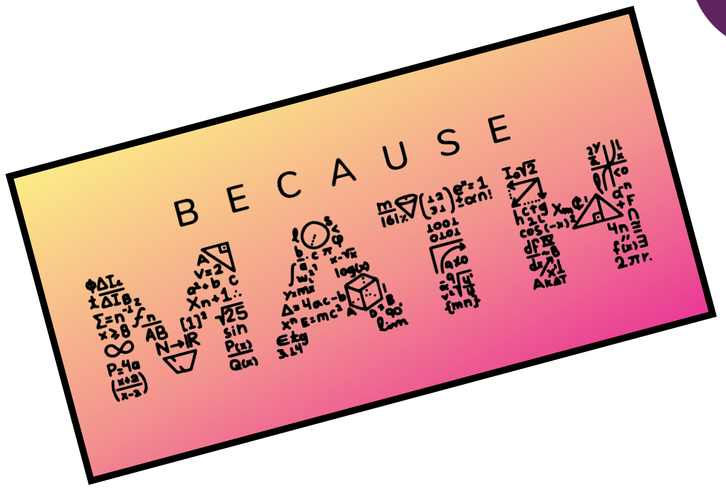
SCREAMING_SNAKE_CASE

WE HAVE STICKERS!

fission







SCREAMING_SNAKE_CASE

WE HAVE STICKERS!

PING ME AND WE'LL MAIL SOME

fission

SOME BACKGROUND CONTEXT

SOME BACKGROUND CONTEXT

WHAT SET OF PROBLEMS IS FISSION SOLVING?

BACKGROUND CONTEXT

SHIPPING A WEB APP IN 2020 IS TOO HARD!

Backends

- Multi-tenant
- Increasingly sharded
- Highly concurrent
- Data leaks everywhere 😱
- ACL complexity & GDPR

DevOps

- Expensive & complex
- Very much its specialty
- We're close to peak Kubernetes

BACKGROUND CONTEXT

SHIPPING A WEB APP IN 2020 I

Backends

- Multi-tenant
- Increasingly sharded
- Highly concurrent
- Data leaks everywhere 😱
- ACL complexity & GDPR

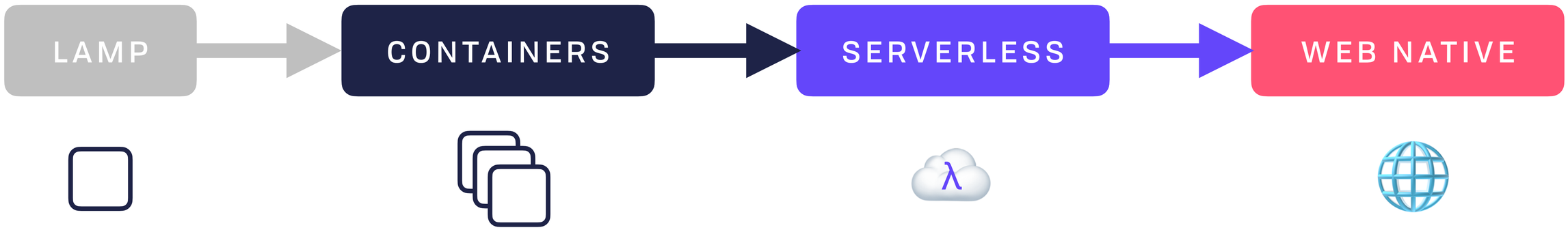
DevOp

- Expe
- Very
- We'

BACKGROUND CONTEXT

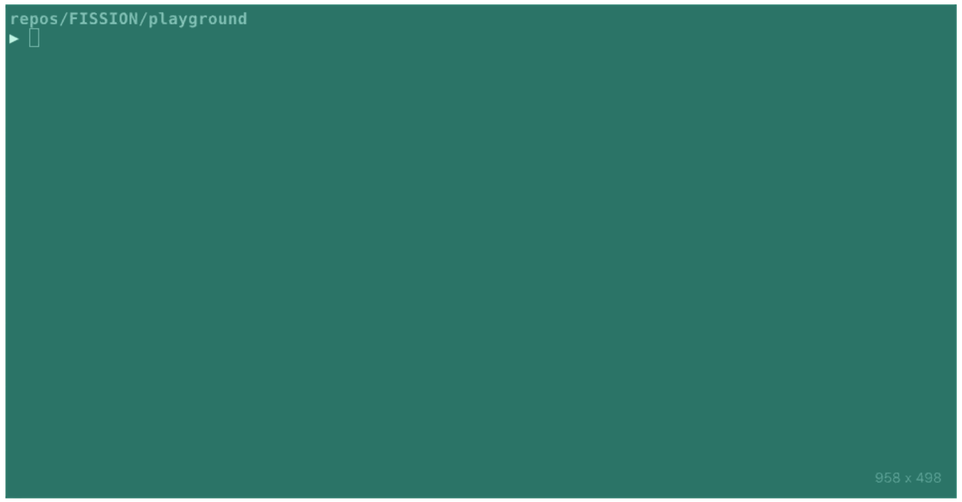
FRONTEND IS EATING THE BACKEND 🍔😋

- Frontend is never going away
- Browsers keep getting *more powerful* (e.g. WebAssembly, WebAuthN, WebCrypto)
- Trend to more granular/edge (Cloudflare Workers or Fastly Edge Cloud)
- Empower front end devs / full stack web apps for the 20's and beyond 🚀



BACKGROUND CONTEXT

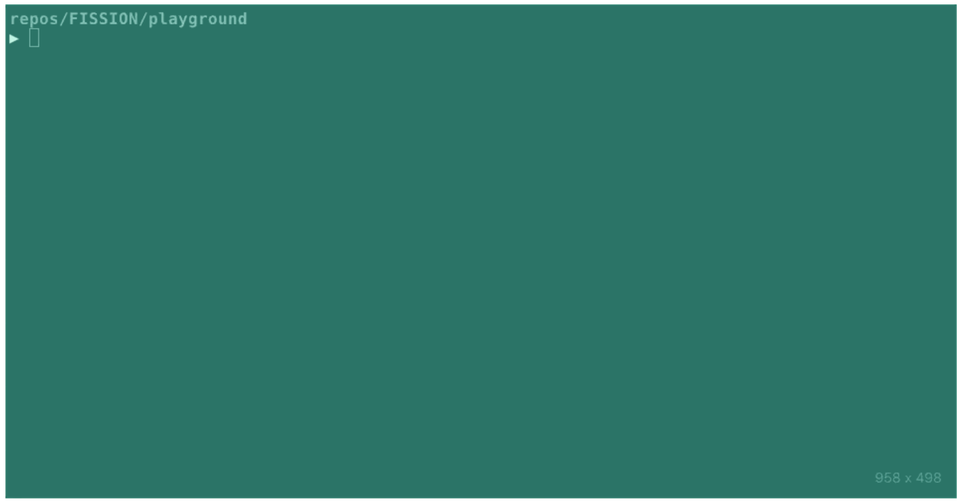
CONSTRAINTS



BACKGROUND CONTEXT

CONSTRAINTS

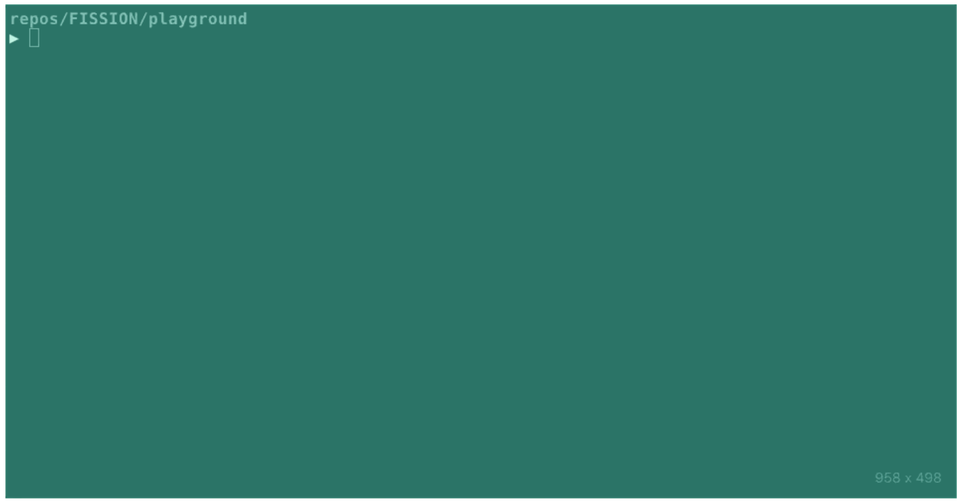
- **Everything** for a modern web app **directly** in the browser
- Vanilla browsers only — **no plug-ins**
- **As secure *or better*** than traditional cloud infra
- Equal or **easier UX**
- **Users fully control** their data
- Apps must work both **offline** and **networked**
- Infrastructure agnostic — “local is the same as prod”



BACKGROUND CONTEXT

CONSTRAINTS

- **Everything** for a modern web app **directly** in the browser
- Vanilla browsers only — **no plug-ins**
- **As secure *or better*** than traditional cloud infra
- Equal or **easier UX**
- **Users fully control** their data
- Apps must work both **offline** and **networked**
- Infrastructure agnostic — “local is the same as prod”



BACKGROUND CONTEXT

WE HAVE SOME NEW BUILDING BLOCKS!

- Start thinking “universally”
- WebCrypto API
- Self-sovereign identity / DID
- Content addressing
- Macaroons & SPKI auth
- CRDTs
- Immutable (functional) data structures at web scale!

(Disclaimer: taken care of under the hood, but interoperable)

BACKGROUND CONTEXT

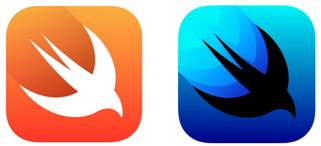
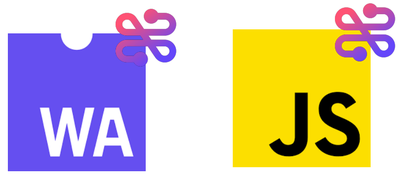
WE HAVE SOME NEW BUILDING BLOCKS!

- Start thinking “universally”
- WebCrypto API
- Self-sovereign identity / DID
- Content addressing
- Macaroons & SPKI auth
- CRDTs
- Immutable (functional) data structures at web scale!

(Disclaimer: taken care of under the hood, but interoperable)

BACKGROUND CONTEXT

"WEB NATIVE"

	iOS	+ fission
COMPUTE		
IDENTITY		
STORAGE		

Build web apps more like native mobile & desktop

Password-less login, end-to-end encryption, secure by default

Local-first, secure, user controlled, global file & hosting platform

BACKGROUND CONTEXT

UPSHOT?

BACKGROUND CONTEXT

UPSHOT?

- Go from zero to production *on a plane* ✈️
- Move data to compute and vice versa 🔄
- Scale sub-linearly 📈
- Serve areas that lack sufficient cloud hardware 🖥️
- *Anyone* can be a service provider (lower bar to entry) 💁‍♀️💁‍♂️

CONTENT ADDRESSING

CONTENT ADDRESSING

MEET THE INTERPLANETARY FILE SYSTEM 👋

CONTENT ADDRESSING

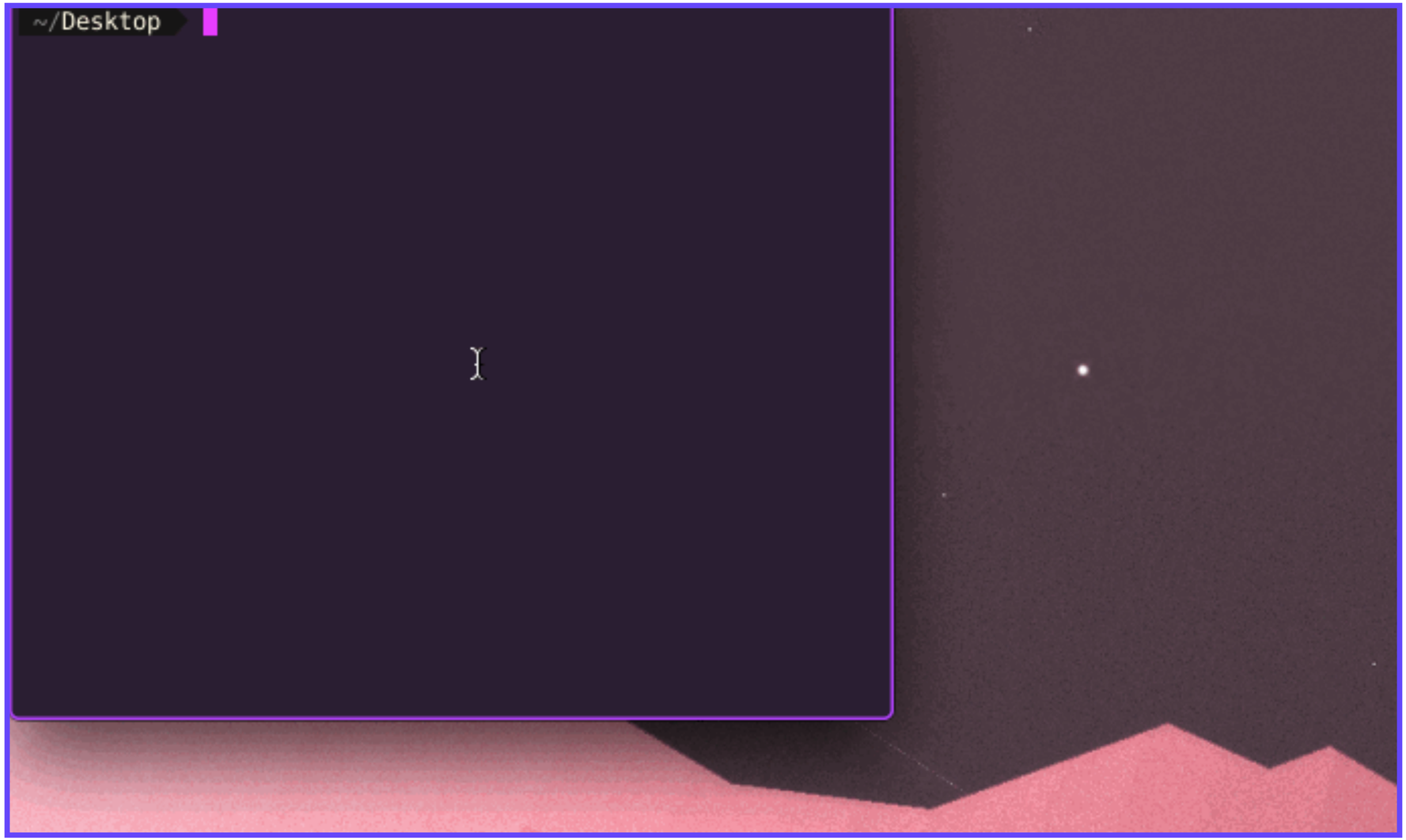
INTERPLANETARY FILE SYSTEM

CONTENT ADDRESSING

INTERPLANETARY FILE SYSTEM

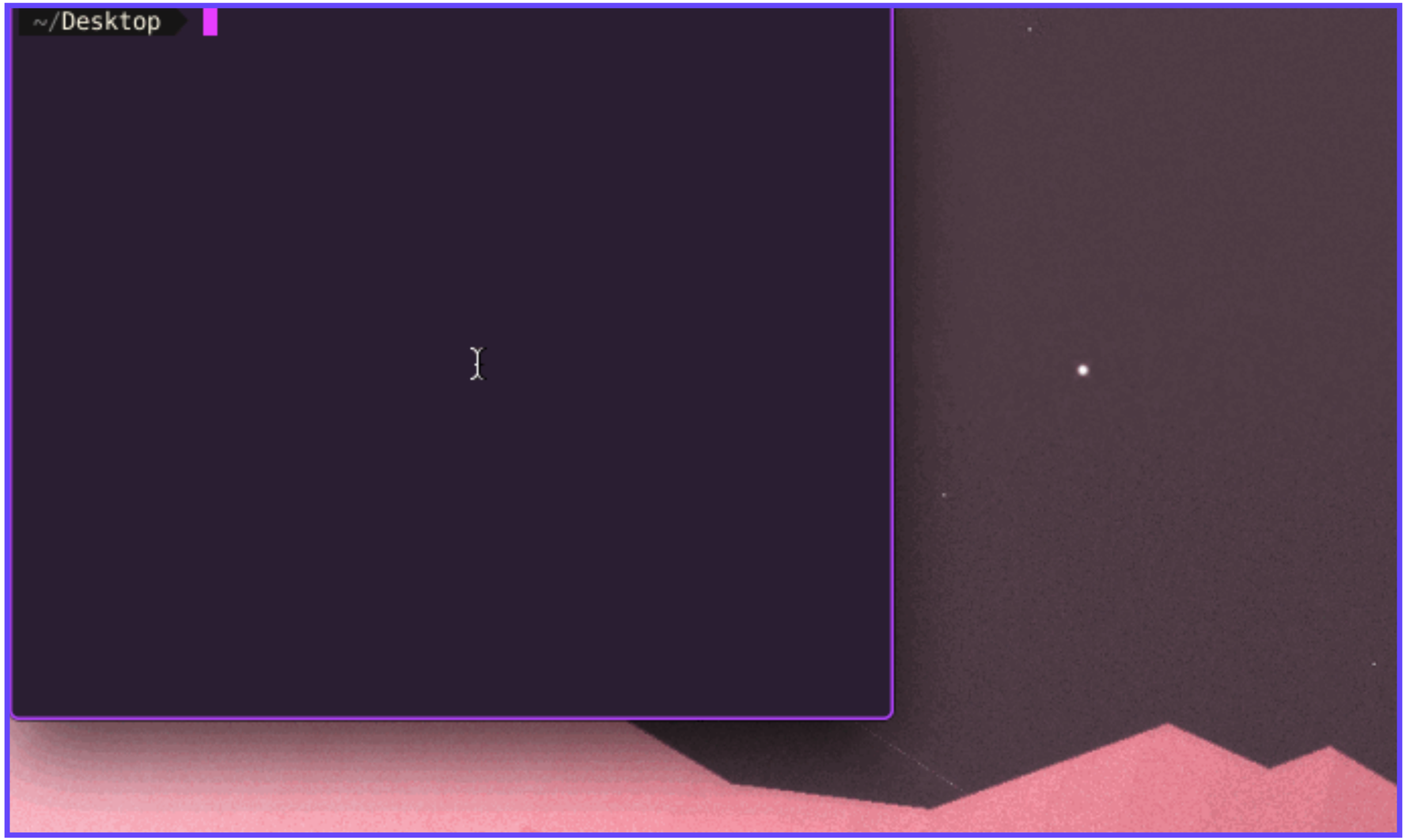
CONTENT ADDRESSING

INTERPLANETARY FILE SYSTEM



CONTENT ADDRESSING

INTERPLANETARY FILE SYSTEM



CONTENT ADDRESSING

LOCATION ADDRESSING

CONTENT ADDRESSING

LOCATION ADDRESSING

- Predominantly single-source (per file) server/client

CONTENT ADDRESSING

LOCATION ADDRESSING

- Predominantly single-source (per file) server/client
- Like a key/value store `{ip => {path => content}}`

CONTENT ADDRESSING
LOCATION ADDRESSING

- Predominantly single-source (per file) server/client
- Like a key/value store `{ip => {path => content}}`
- “Location addressing”
  - DNS maps names to IP addresses
  - Focused on the physical network

VIRTUAL ADDRESS

PHYSICAL LOCATION

CONTENT ADDRESSING

LOCATION ADDRESSING

- Predominantly single-source (per file) server/client
- Like a key/value store `{ip => {path => content}}`
- “Location addressing”
  - DNS maps names to IP addresses
  - Focused on the physical network
- Mutable addressing
  - www.foo.com/baz may be JSON today, but a video tomorrow
  - …or altered content

VIRTUAL ADDRESS

PHYSICAL LOCATION

CONTENT ADDRESSING

CONTENT IDS

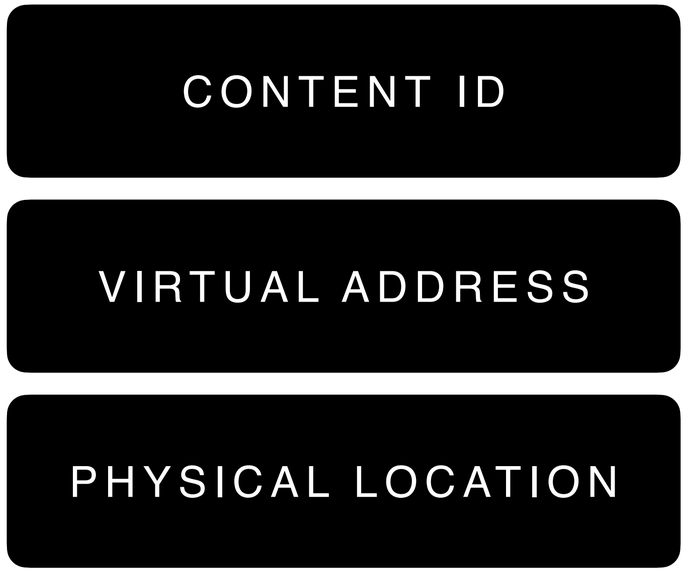
VIRTUAL ADDRESS

PHYSICAL LOCATION

CONTENT ADDRESSING

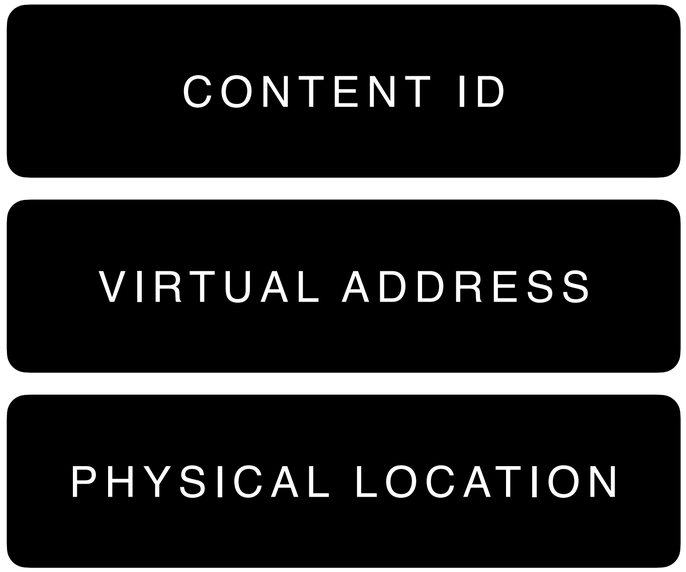
CONTENT IDS

- A layer of abstraction above location



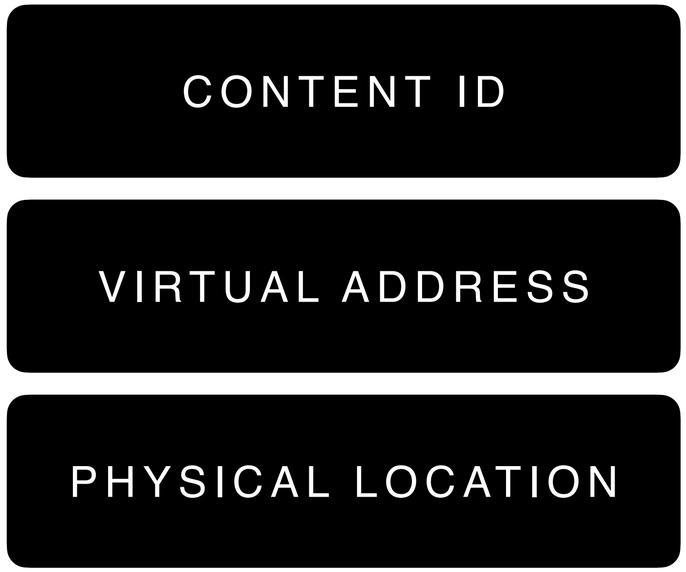
CONTENT ADDRESSING
CONTENT IDS

- A layer of abstraction above location
- Like a key/value store `{hash(content) => content}`
  - Content hash AKA “content identifier” or CID
  - Special “universal” relationship to content



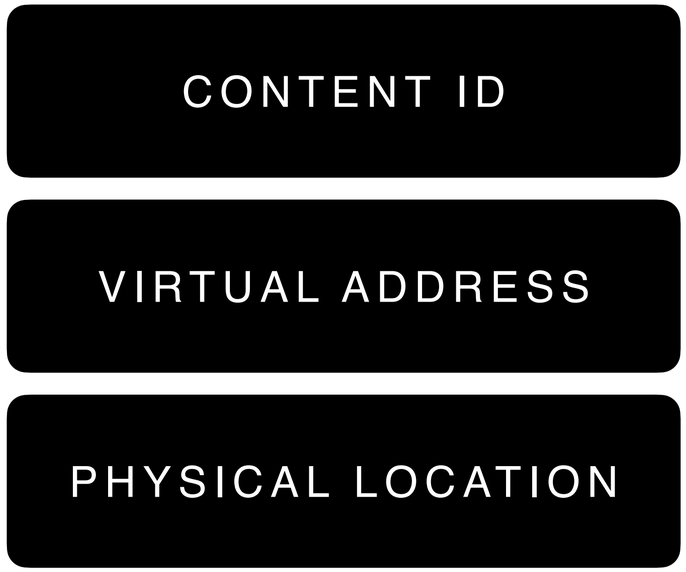
CONTENT ADDRESSING
CONTENT IDS

- A layer of abstraction above location
- Like a key/value store `{hash(content) => content}`
  - Content hash AKA “content identifier” or CID
  - Special “universal” relationship to content
- Focused on *the data*



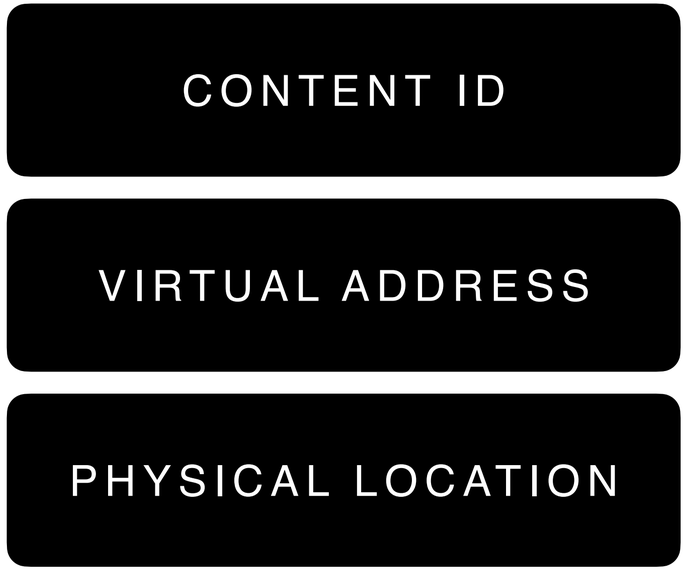
CONTENT ADDRESSING
CONTENT IDS

- A layer of abstraction above location
- Like a key/value store `{hash(content) => content}`
  - Content hash AKA “content identifier” or CID
  - Special “universal” relationship to content
- Focused on *the data*
- Does not care where it lives



CONTENT ADDRESSING
CONTENT IDS

- A layer of abstraction above location
- Like a key/value store `{hash(content) => content}`
  - Content hash AKA “content identifier” or CID
  - Special “universal” relationship to content
- Focused on *the data*
- Does not care where it lives
- Still have paths
  - Immutable DAG
  - Why no loops?

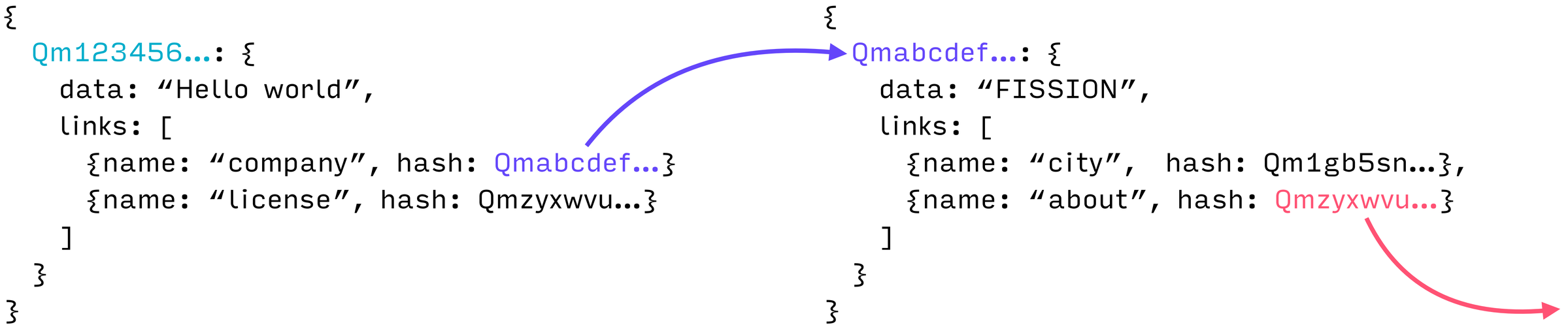


CONTENT ADDRESSING
LINKED DATA

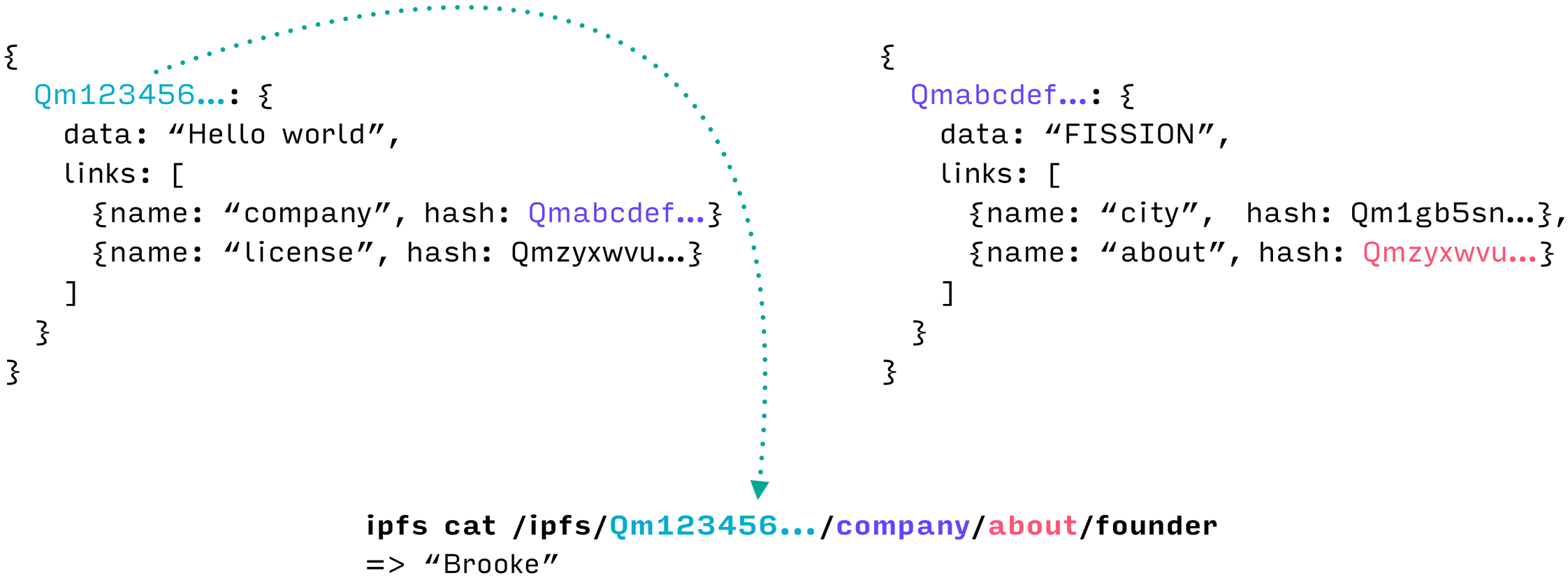
CONTENT ADDRESSING
LINKED DATA

```
{
  Qm123456…: {
    data: “Hello world”,
    links: [
      {name: “company”, hash: Qmabcdef…}
      {name: “license”, hash: Qmzyxwvu…}
    ]
  }
}
```

CONTENT ADDRESSING
LINKED DATA



CONTENT ADDRESSING
LINKED DATA



CONTENT ADDRESSING

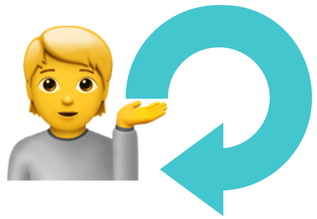
ROUTING & LOOKUP 🔍🌊

CONTENT ADDRESSING

ROUTING & LOOKUP 🔍🌊

CONTENT ADDRESSING

ROUTING & LOOKUP 🔍🌊

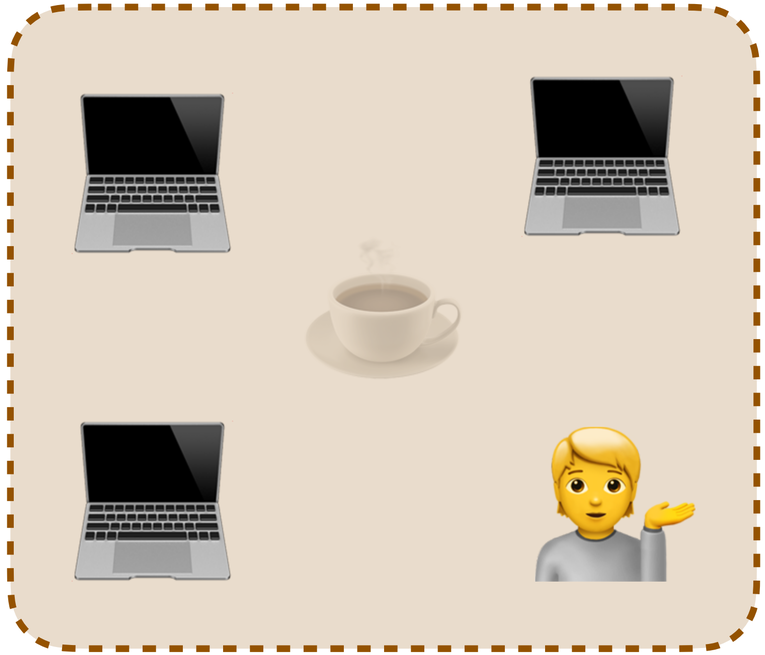


CONTENT ADDRESSING

ROUTING & LOOKUP 🔍🌊

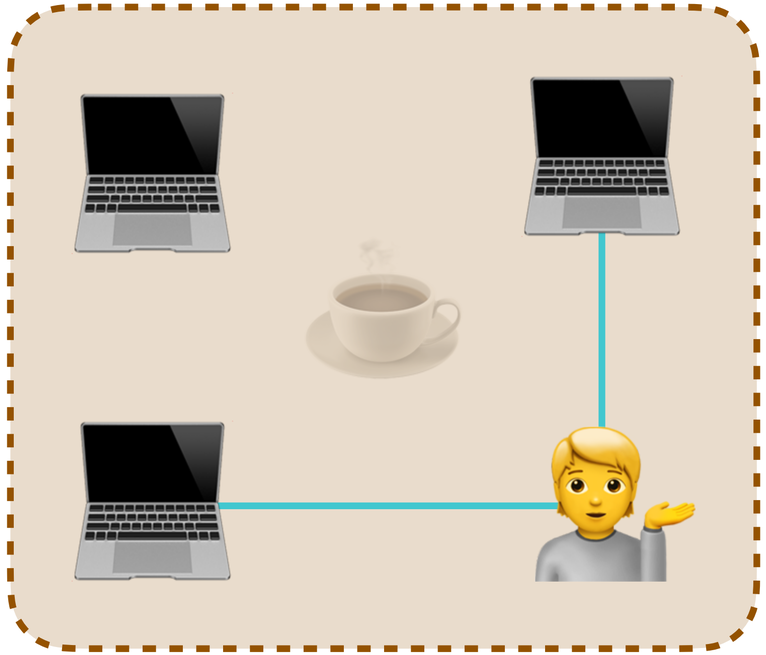
CONTENT ADDRESSING

ROUTING & LOOKUP 🔍🌊



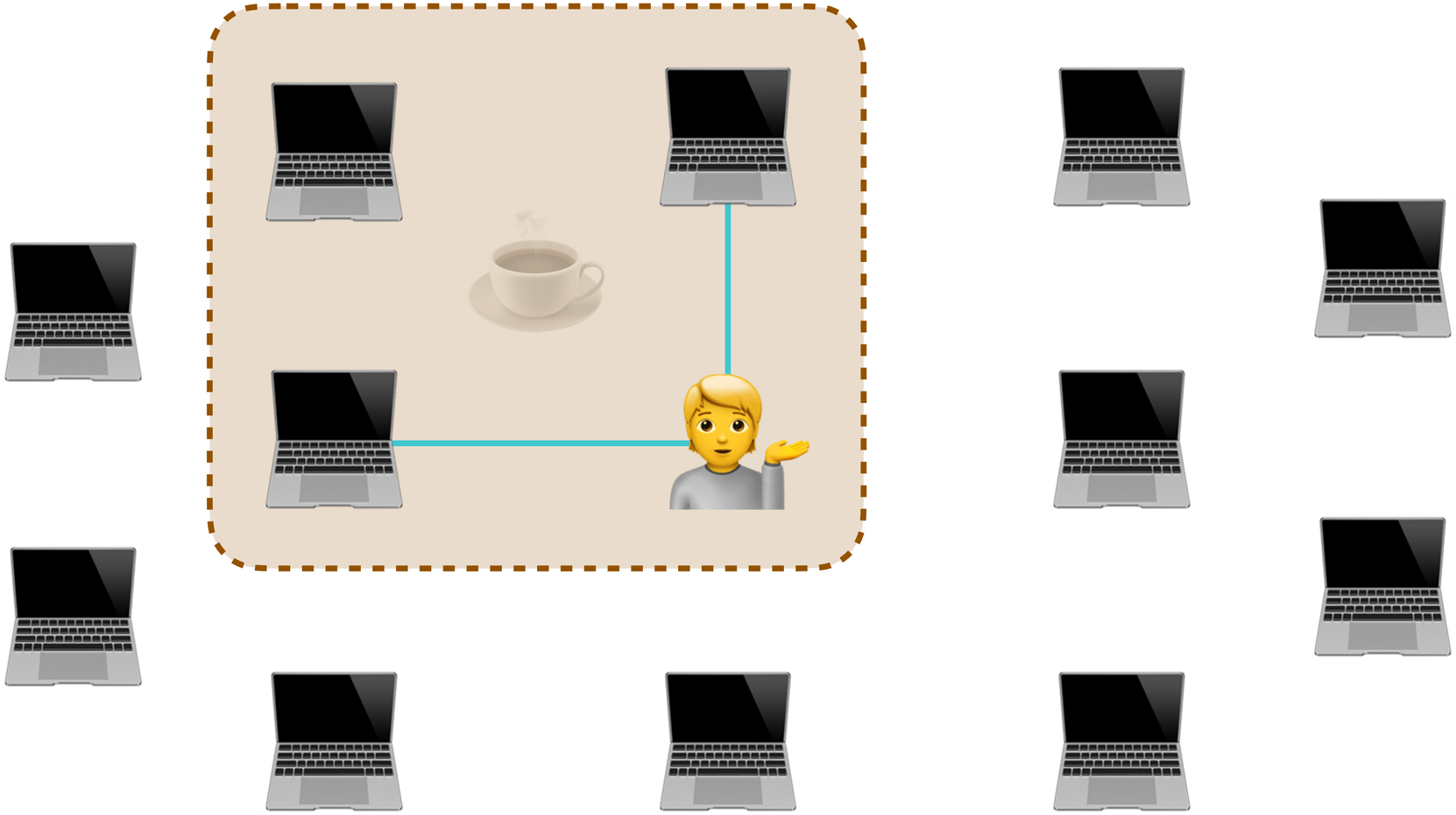
CONTENT ADDRESSING

ROUTING & LOOKUP 🔍🌊



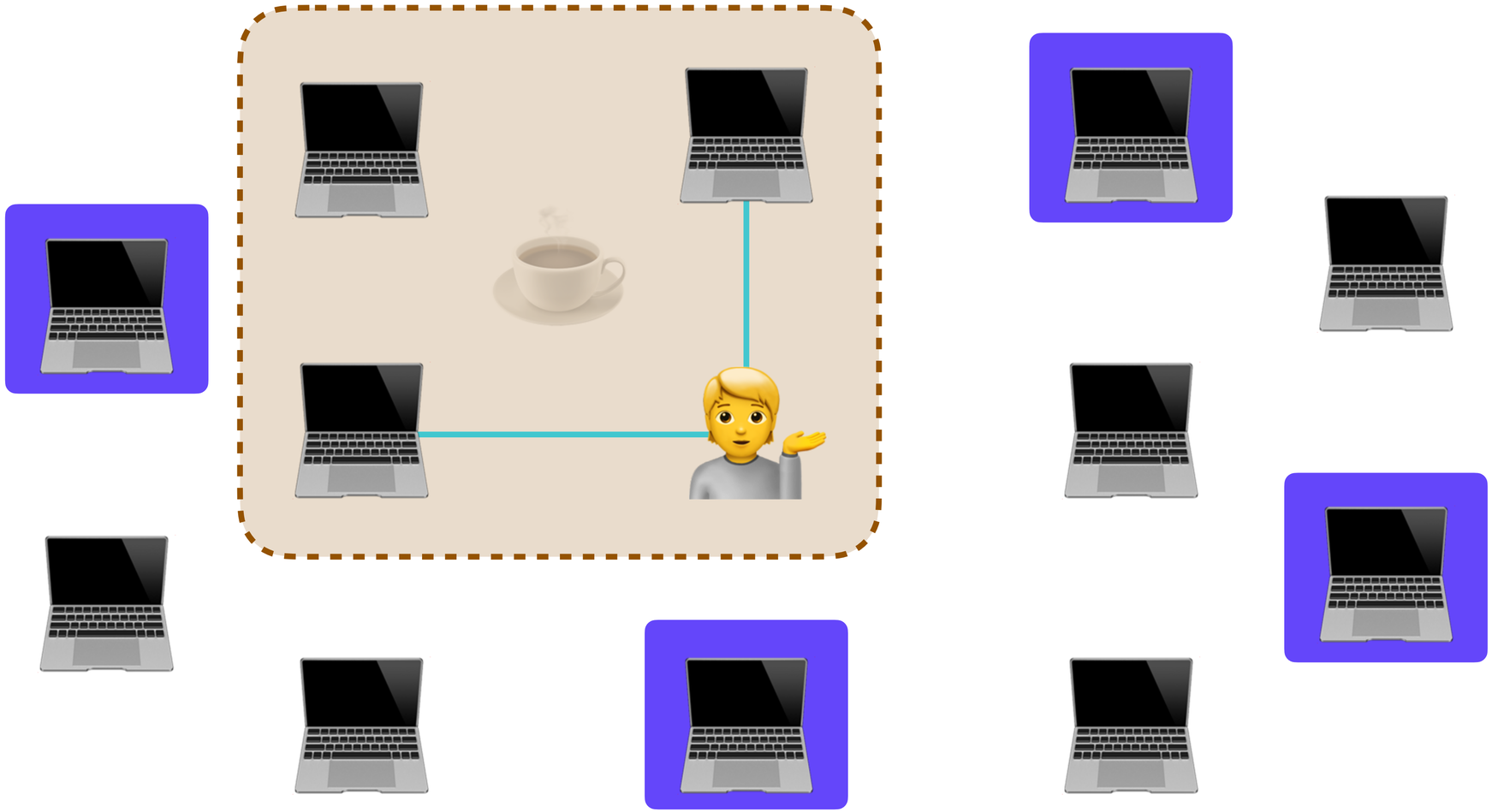
CONTENT ADDRESSING

ROUTING & LOOKUP 🔍🌊



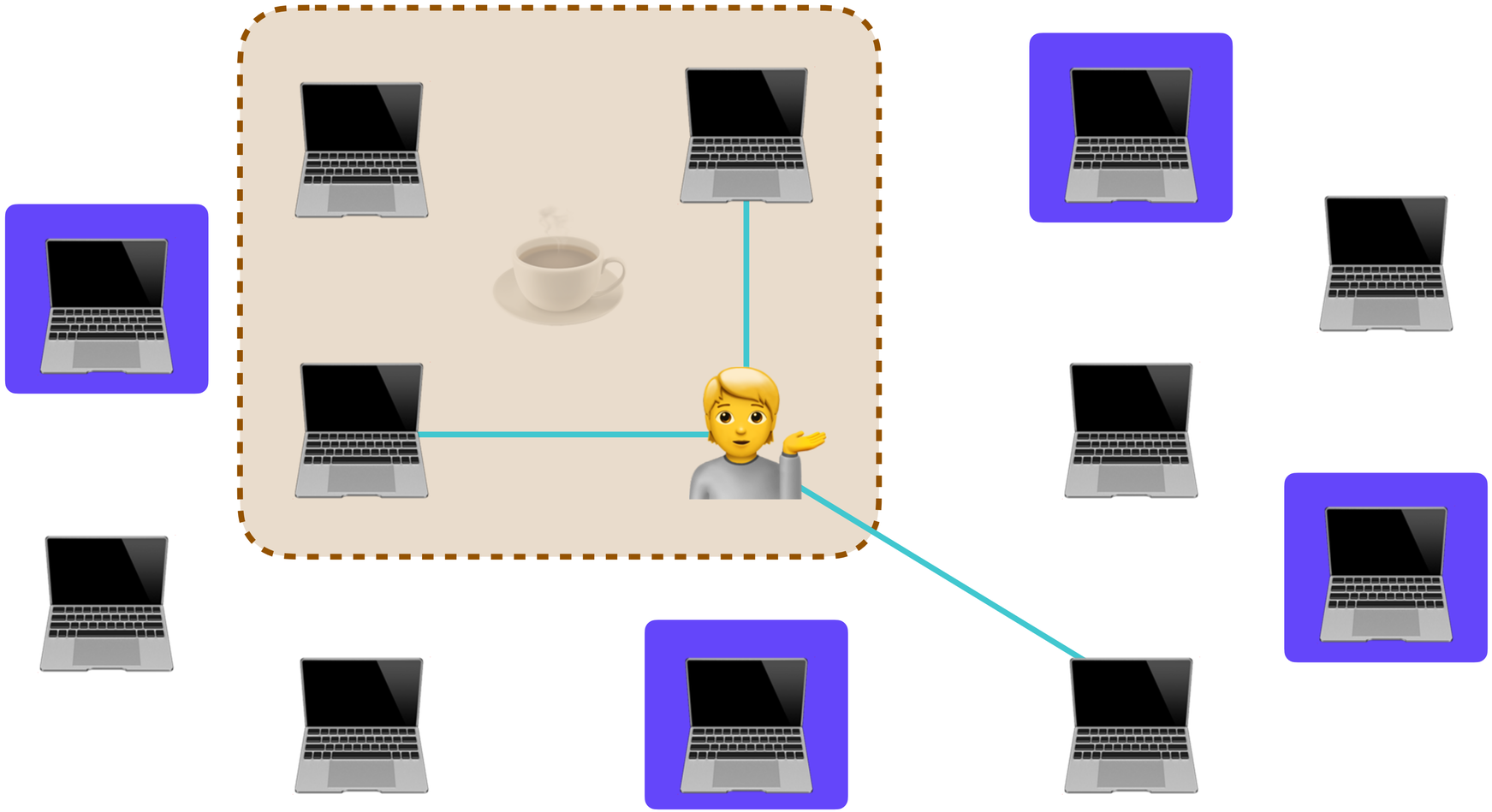
CONTENT ADDRESSING

ROUTING & LOOKUP 🔍🌊



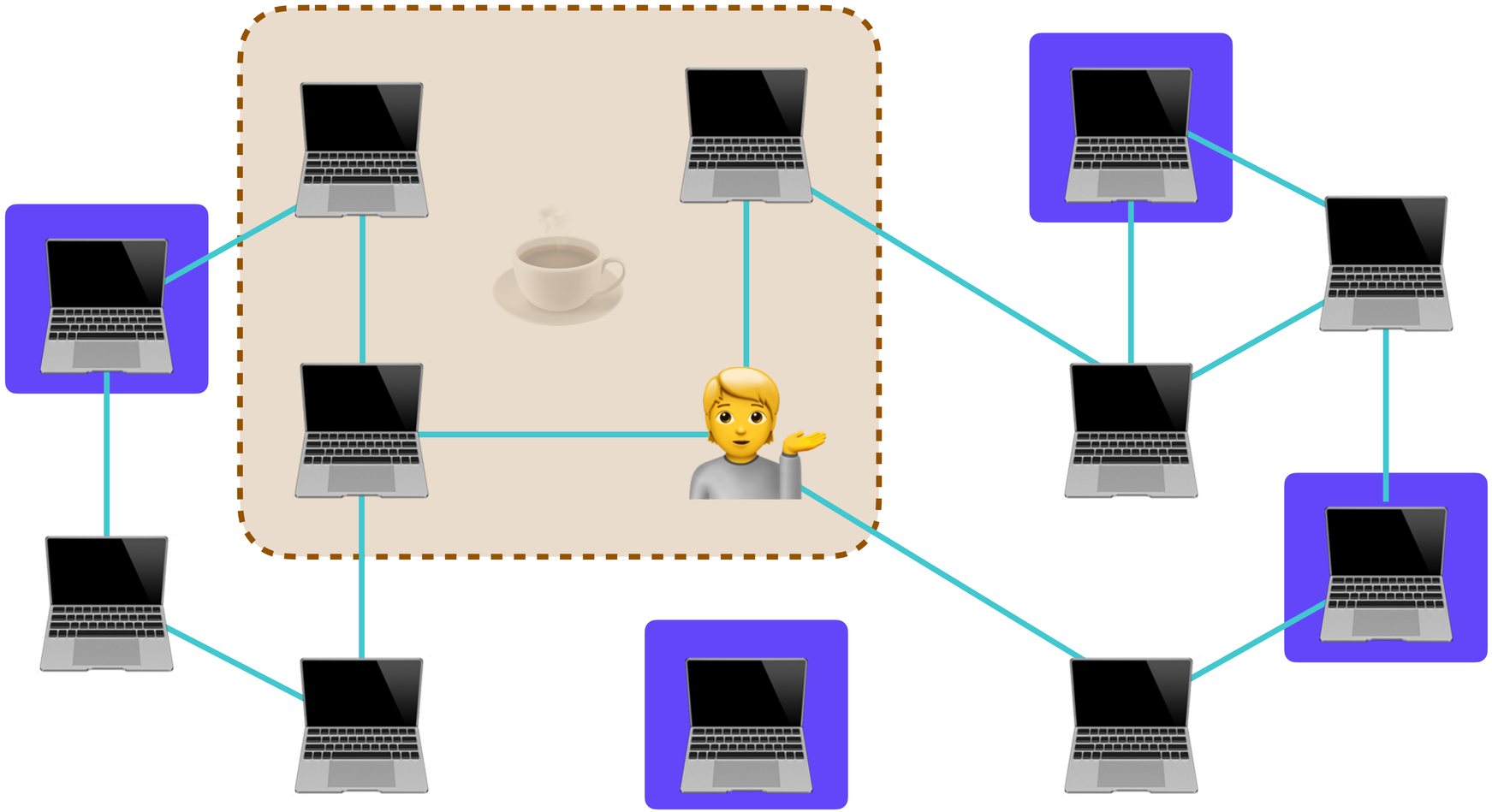
CONTENT ADDRESSING

ROUTING & LOOKUP 🔍🌊



CONTENT ADDRESSING

ROUTING & LOOKUP 🔍🌊

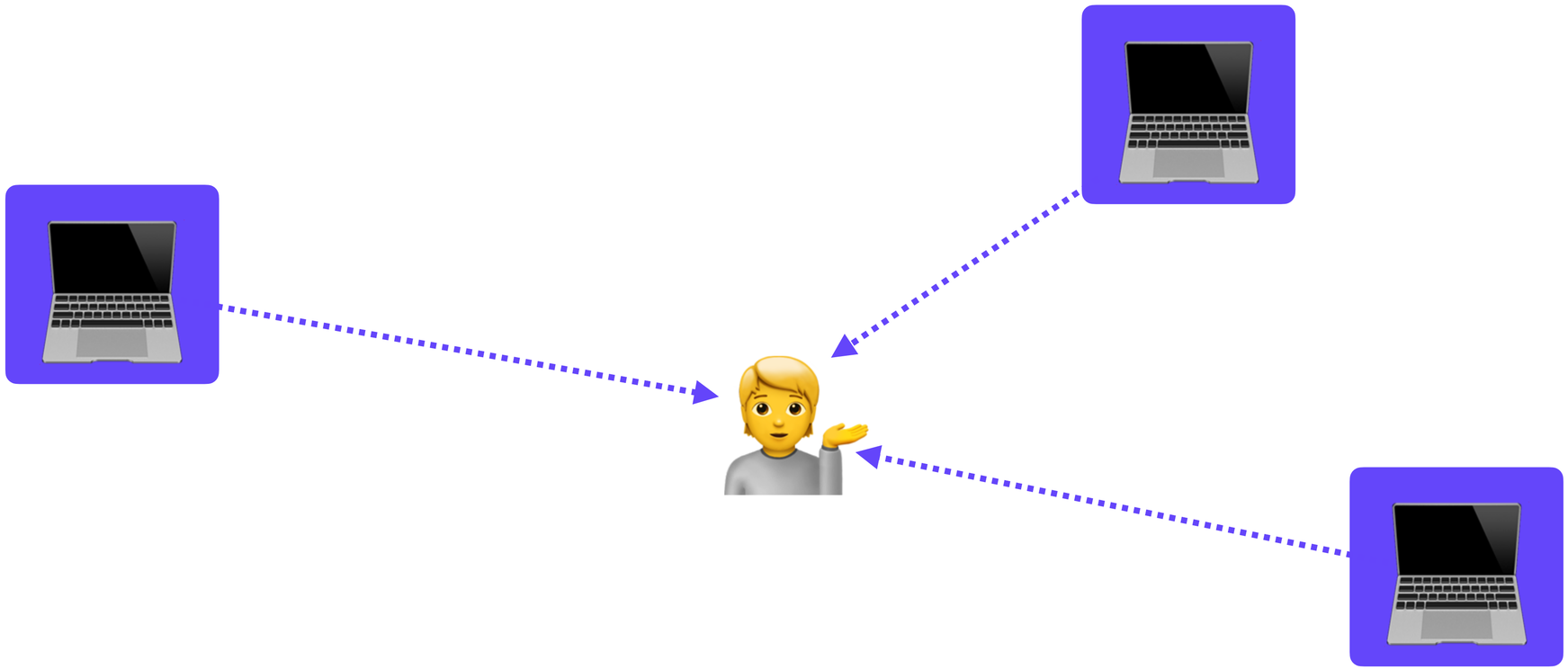


CONTENT ADDRESSING

ROUTING & LOOKUP 🔍🌊

CONTENT ADDRESSING

ROUTING & LOOKUP 🔍🌊



CONTENT ADDRESSING

ONE *HUGE* NAMESPACE TO RULE THEM ALL 💍🌋

CONTENT ADDRESSING

ONE *HUGE* NAMESPACE TO RULE THEM ALL 💍🌋

- Same file = same hash
  - No matter when 🕓
  - No matter where 🌍🌌
  - No matter who 👩🤖👨‍🚀🦸🦊
- Zero file duplication per node
- Replication = CDN-ish features
- Immutable data structures
- Structural sharing = dedup blocks
- Files & data living together!
- Automerge, OrbitDB, &c

CONTENT ADDRESSING

FISSION FILE SYSTEM

CONTENT ADDRESSING

FISSION FILE SYSTEM

- Unix file system interface

CONTENT ADDRESSING

FISSION FILE SYSTEM

- Unix file system interface
- Same files across apps (think native apps instead of trad. web apps)

CONTENT ADDRESSING

FISSION FILE SYSTEM

- Unix file system interface
- Same files across apps (think native apps instead of trad. web apps)
- Fully immutable, temporal data structure
  - Peek & rollback (like Apple Time Machine for the web)

CONTENT ADDRESSING

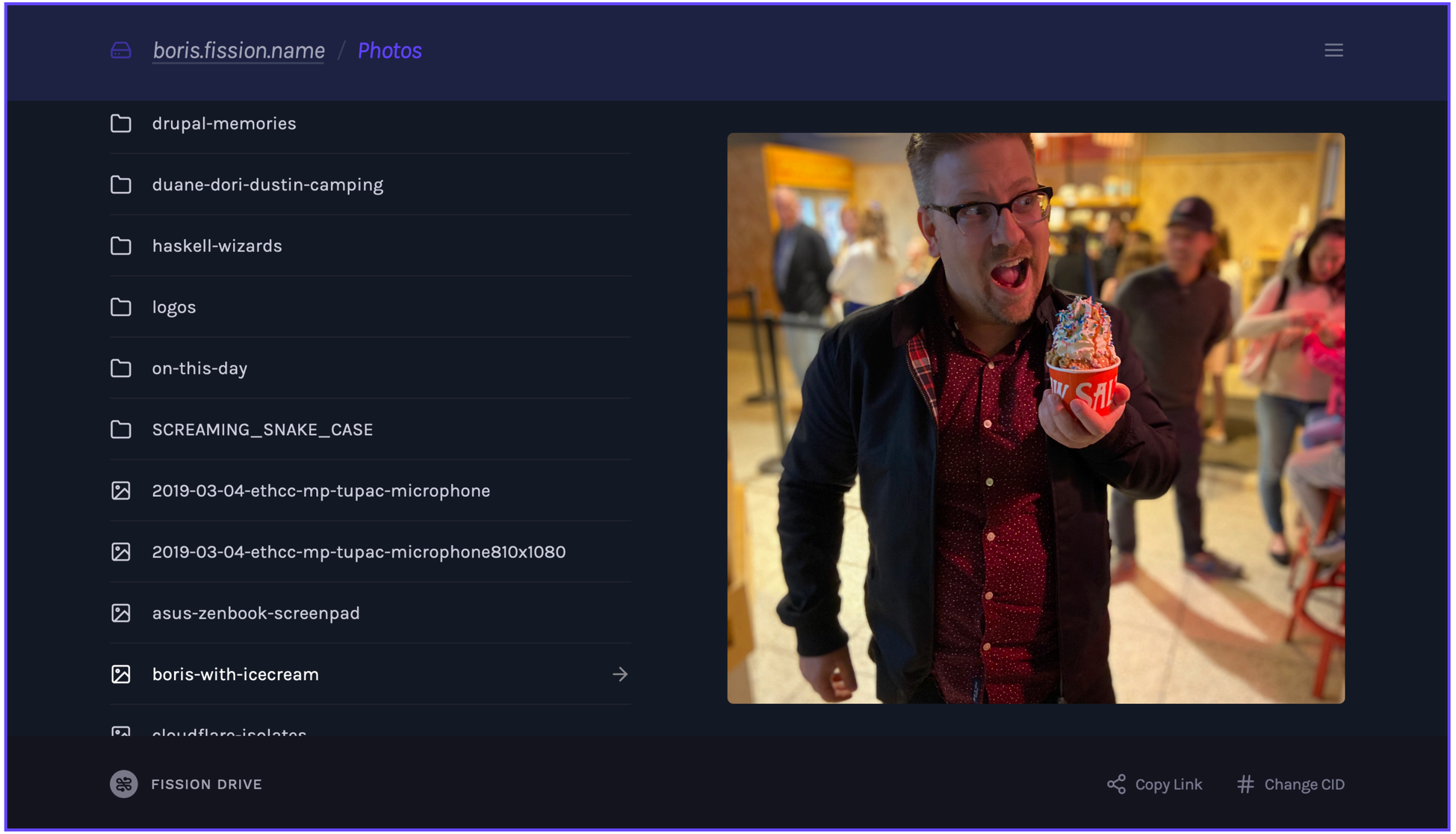
FISSION FILE SYSTEM

- Unix file system interface
- Same files across apps (think native apps instead of trad. web apps)
- Fully immutable, temporal data structure
  - Peek & rollback (like Apple Time Machine for the web)
- Event-sourced file system (apps can consume stream)

CONTENT ADDRESSING

DRIVE — "FULLY USER CONTROLLED DROPBOX"

https://drive.fission.codes



UCAN

UCAN

🔑 CLIENT-SIDE AUTHN & AUTHZ 🦸‍♀️

STEP ONE

USER IDS WITHOUT A DATABASE

USER IDS WITHOUT A DATABASE

STANDARDIZATION 🏢

USER IDS WITHOUT A DATABASE

STANDARDIZATION 🏛

- W3C, Microsoft, BC, etc
- For users, devices, and more
- Based on public-key cryptography
- Truly “universal” UUIDs
- Agnostic about backing

EXAMPLE 2: Minimal self-managed DID Document

```
{
  "@context": "https://w3id.org/did/v1",
  "id": "did:example:123456789abcdefghi",
  "publicKey": [{
    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "RsaVerificationKey2018",
    "owner": "did:example:123456789abcdefghi",
    "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
  }],
  "authentication": [{
    // this key can be used to authenticate as DID ...9938
    "type": "RsaSignatureAuthentication2018",
    "publicKey": "did:example:123456789abcdefghi#keys-1"
  }],
  "service": [{
    "type": "ExampleService",
    "serviceEndpoint": "https://example.com/endpoint/8377464"
  }]
}
```

USER IDS WITHOUT A DATABASE

SELF-SOVEREIGN IDENTITY (SSI) 🔑🦸‍♀️

USER IDS WITHOUT A DATABASE

SELF-SOVEREIGN IDENTITY (SSI) 🔑🦹‍♀️

- Generate your own globally-unique, verifiable user ID!

USER IDS WITHOUT A DATABASE

SELF-SOVEREIGN IDENTITY (SSI) 🔑🦸‍♀️

- Generate your own globally-unique, verifiable user ID!
- As many as you like 🤷‍♀️

USER IDS WITHOUT A DATABASE

SELF-SOVEREIGN IDENTITY (SSI) 🔑🤦‍♀️

- Generate your own globally-unique, verifiable user ID!
- As many as you like 🤷‍♀️
- Many methods — we're starting with **`did:key`**

USER IDS WITHOUT A DATABASE

SELF-SOVEREIGN IDENTITY (SSI) 🔑🤦‍♀️

- Generate your own globally-unique, verifiable user ID!
- As many as you like 🤷‍♀️
- Many methods — we're starting with **`did:key`**
- Not super readable, so publicize over DNS TXT record

```
_did.USERNAME.fission.name
```

USER IDS WITHOUT A DATABASE

SELF-SOVEREIGN IDENTITY (SSI) 🔑🤦‍♀️

- Generate your own globally-unique, verifiable user ID!
- As many as you like 🤷‍♀️
- Many methods — we're starting with **`did:key`**
- Not super readable, so publicize over DNS TXT record

```
_did.USERNAME.fission.name
```

```
did:key:zBR4m3DNZHT1G8Nb2RHzgKK7TrWxEmJjZskgvFdncTthzUHzngyNKmKx4VKWEJE6sk4SE4Ka3kH92MxU2YC7CcePHy77GzZy8
```

```
Ed25519 — AAAAC3NzaC1lZDI1NTE5AAAAIB7/gFUQ9llI1BTrEjW7Jq6fX6JLsK1J4wXK/dn9JMcO
```

STEP TWO

DISTRIBUTED READ CONTROL

DISTRIBUTED READ CONTROL

OCAP / READ KEYS

DISTRIBUTED READ CONTROL

OCAP / READ KEYS

- ACLs
  - "Reactive access control”
  - Authority by association

DISTRIBUTED READ CONTROL

OCAP / READ KEYS

- ACLs
  - "Reactive access control”
  - Authority by association
- OCAP
  - “Proactive" access control
  - Authority by possession
    - “You either have the key, or you don’t”

DISTRIBUTED READ CONTROL

OCAP / READ KEYS

- ACLs
  - "Reactive access control”
  - Authority by association
- OCAP
  - “Proactive" access control
  - Authority by possession
    - “You either have the key, or you don’t”
- “Just” normal AES-256 keys

DISTRIBUTED READ CONTROL

MORE GRANULAR ACCESS: CRYPTREES 🔐🌳

DISTRIBUTED READ CONTROL

MORE GRANULAR ACCESS: CRYPTREES 🔐🌳

- Public keys playing double duty: IDs and secure key exchange!

DISTRIBUTED READ CONTROL

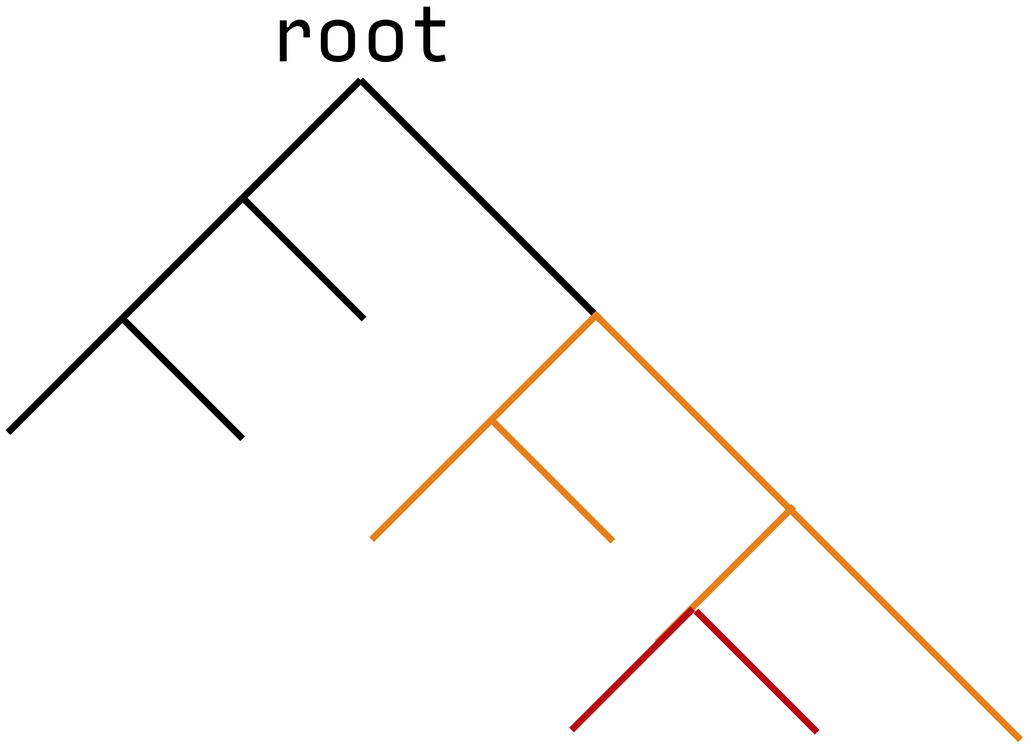
MORE GRANULAR ACCESS: CRYPTREES 🔐🌳

- Public keys playing double duty: IDs and secure key exchange!
- Encrypt the encryption with more encryption
  - Each layer (file or dir) is encrypted with a key
  - Dirs contain keys for each sub dir / file
  - Recurse!

DISTRIBUTED READ CONTROL

MORE GRANULAR ACCESS: CRYPTREES 🔐🌳

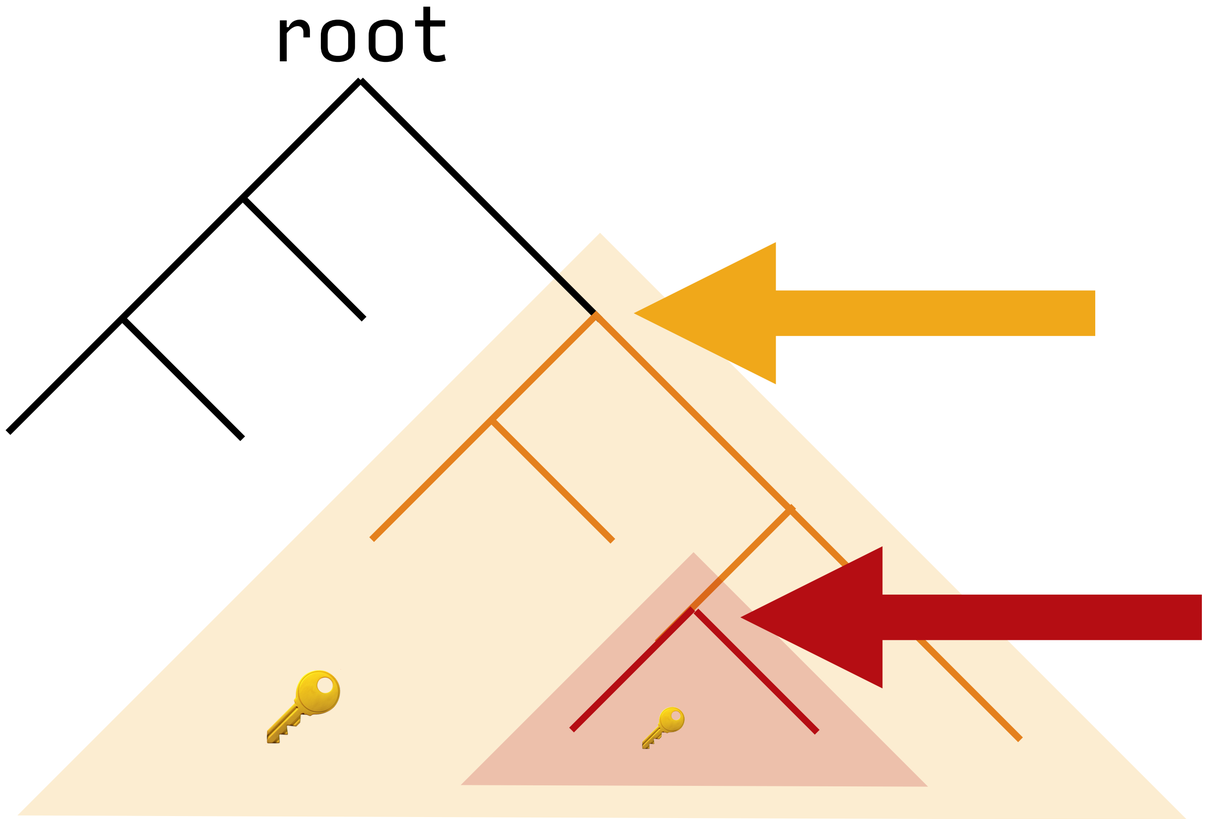
- Public keys playing double duty: IDs and secure key exchange!
- Encrypt the encryption with more encryption
  - Each layer (file or dir) is encrypted with a key
  - Dirs contain keys for each sub dir / file
  - Recurse!



DISTRIBUTED READ CONTROL

MORE GRANULAR ACCESS: CRYPTREES 🔐🌳

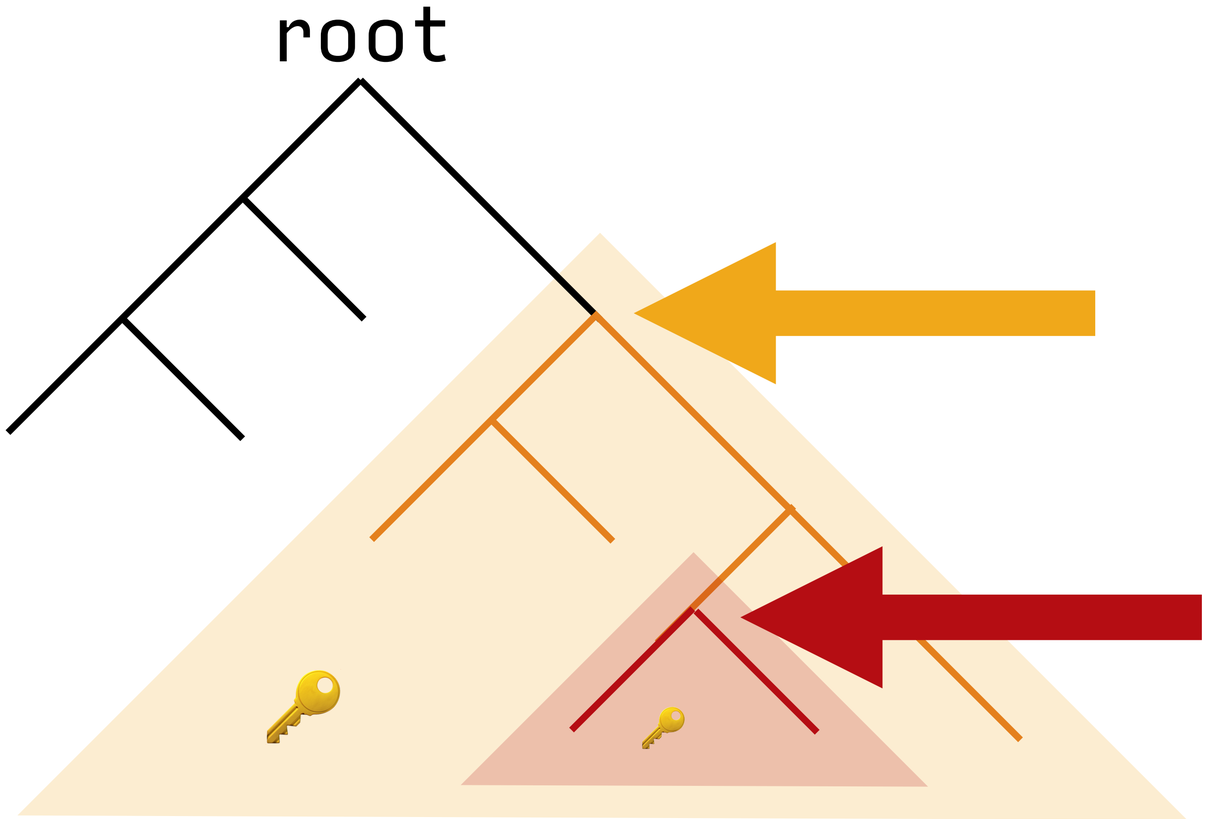
- Public keys playing double duty: IDs and secure key exchange!
- Encrypt the encryption with more encryption
  - Each layer (file or dir) is encrypted with a key
  - Dirs contain keys for each sub dir / file
  - Recurse!



DISTRIBUTED READ CONTROL

MORE GRANULAR ACCESS: CRYPTREES 🔐🌳

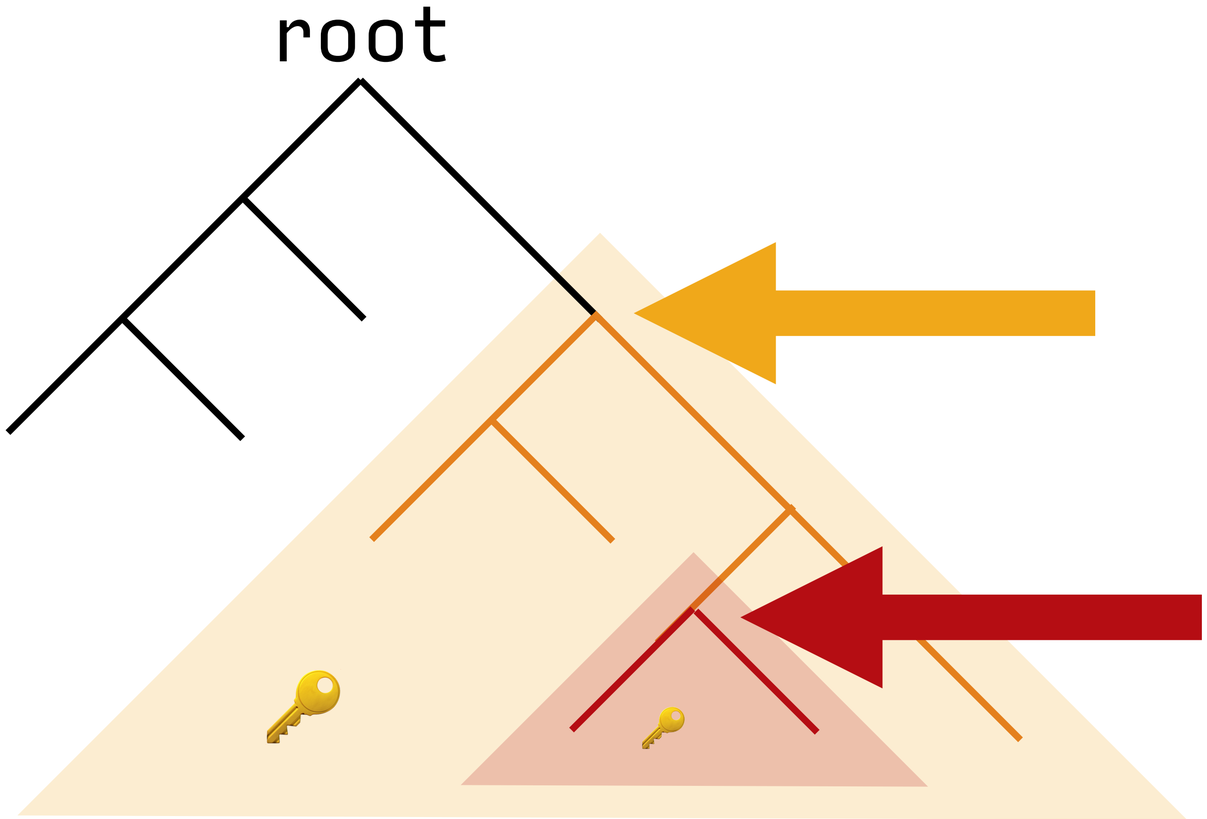
- Public keys playing double duty: IDs and secure key exchange!
- Encrypt the encryption with more encryption
  - Each layer (file or dir) is encrypted with a key
  - Dirs contain keys for each sub dir / file
  - Recurse!
- Access granted to a directory and below
  - i.e. Same UX Dropbox/Google Drive
  - Full user controlled



DISTRIBUTED READ CONTROL

MORE GRANULAR ACCESS: CRYPTREES 🔐🌳

- Public keys playing double duty: IDs and secure key exchange!
- Encrypt the encryption with more encryption
  - Each layer (file or dir) is encrypted with a key
  - Dirs contain keys for each sub dir / file
  - Recurse!
- Access granted to a directory and below
  - i.e. Same UX Dropbox/Google Drive
  - Full user controlled
- Revocation = key rotation & DH exchange

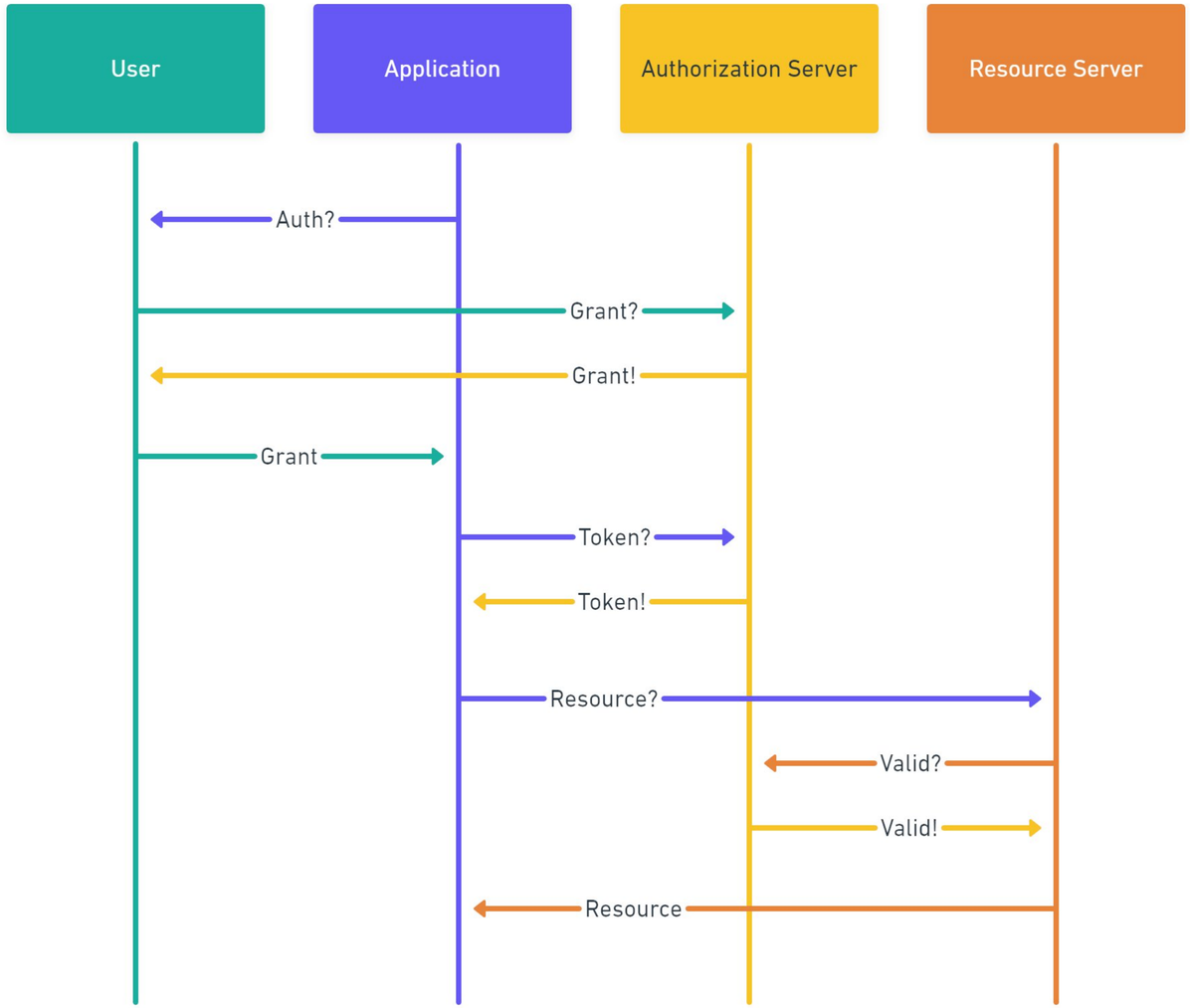


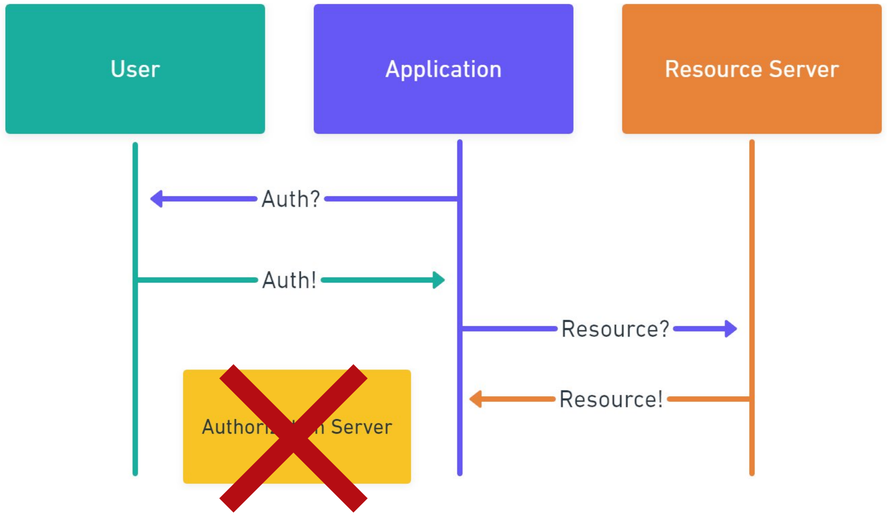
STEP THREE

DELEGATED WRITE ACCESS

DELEGATED WRITE ACCESS

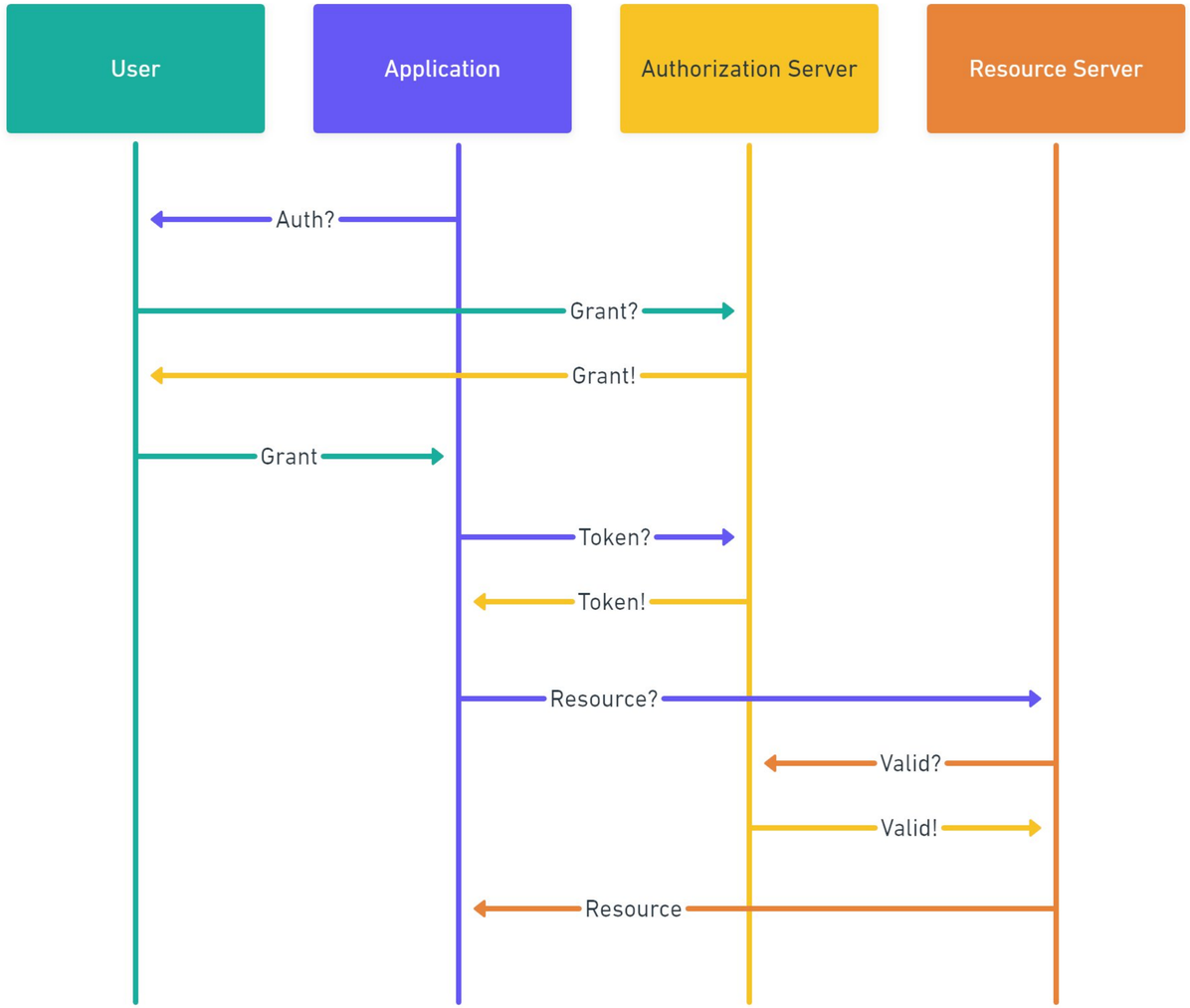
SIDE-BY-SIDE

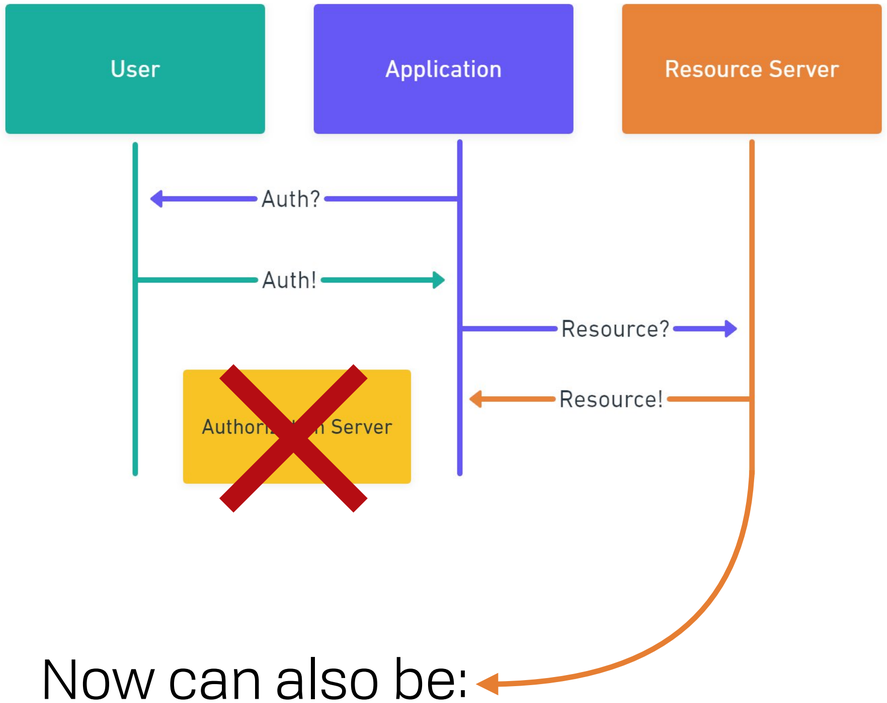




DELEGATED WRITE ACCESS

SIDE-BY-SIDE



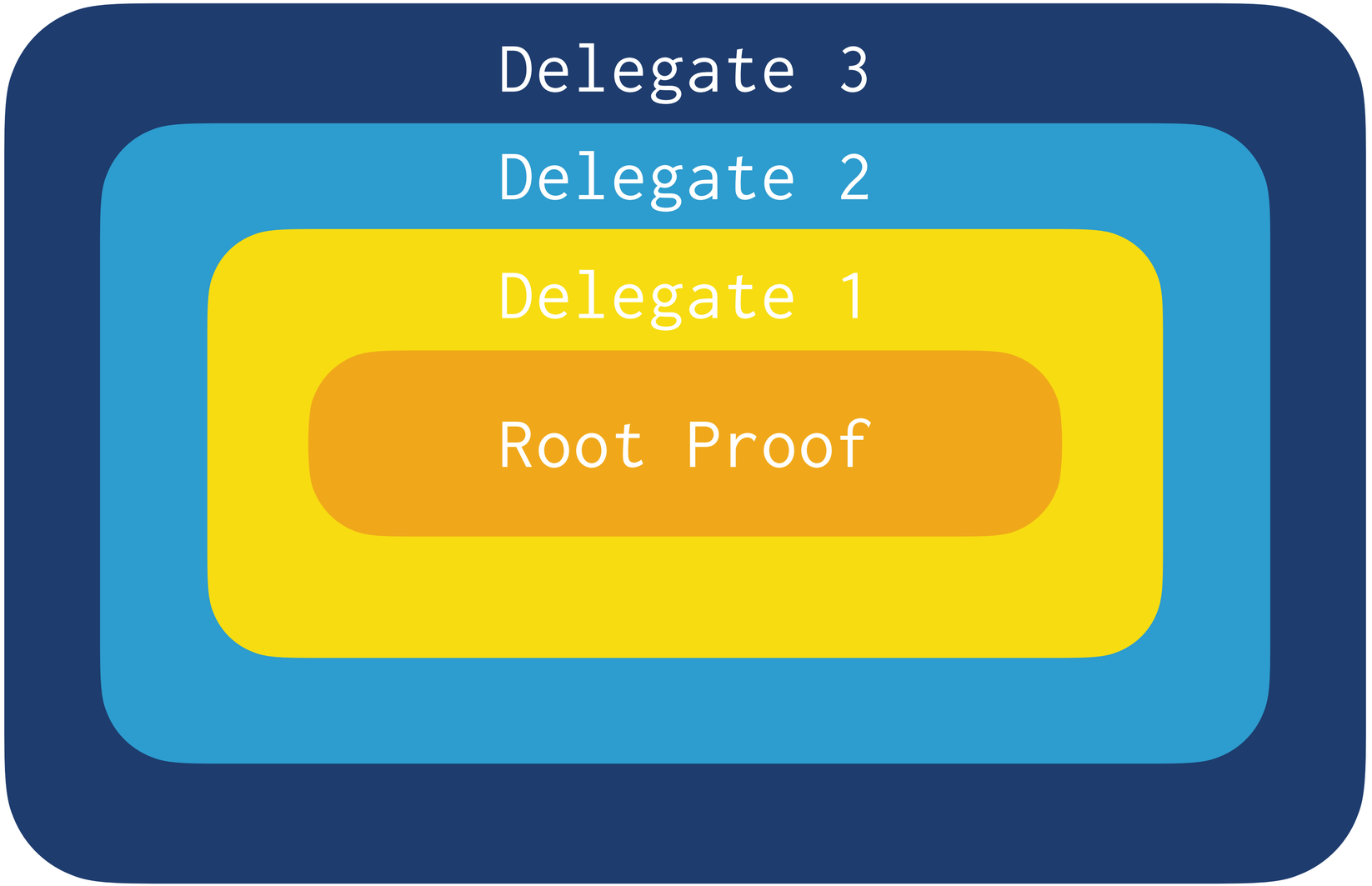


Now can also be:

- Another device (same human)
- A user's peer (different human)
- Some service

DELEGATED WRITE ACCESS

UCAN: USER CONTROLLED AUTHORIZATION NETWORK 🦜



- Solves for user-centrism
- Decentralized delegation
- Attenuation
- Shrink size with CIDs
- Assumes PKI

DELEGATED WRITE ACCESS

EACH LAYER FOLLOWS THIS FORM

```
{
  “alg”: “RS256”,
  “typ”: “JWT“,
  “cty”: “JWT“
}
{
  “iss”:“did:key:z1MdJPaWBebKxtE33AszRWYTF67wCLeFdcsqc3R87hyLKzBK…”,
  “aud”:“did:key:zBR4m3DNZHT1G8Nb2RHzgKK7TrWxEmJjZskgvFdncTthzUH…”,
  “scp”:“/public/photos/covid2020/”,
  “pty”:”APPEND_ONLY”,
  “prf”:<JWT PROOF>,
  “exp”:1589423547
}
<SIGNATURE>
```

DELEGATED WRITE ACCESS

EACH LAYER FOLLOWS THIS FORM

```
{
  “alg”: “RS256”,
  “typ”: “JWT“,
  “cty”: “JWT“
}
{
  “iss”:“did:key:z1MdJPaWBebKxtE33AszRWYTF67wCLeFdcsqc3R87hyLKzBK…”,
  “aud”:“did:key:zBR4m3DNZHT1G8Nb2RHzgKK7TrWxEmJjZskgvFdncTthzUH…”,
  “scp”:“/public/photos/covid2020/”,
  “pty”:”APPEND_ONLY”,
  “prf”:<JWT PROOF>,
  “exp”:1589423547
}
<SIGNATURE>
```

DELEGATED WRITE ACCESS

EACH LAYER FOLLOWS THIS FORM

```
{
  “alg”: “RS256”,
  “typ”: “JWT“,
  “cty”: “JWT“
}
{
  “iss”:“did:key:z1MdJPaWBebKxtE33AszRWYTF67wCLeFdcsqc3R87hyLKzBK…”,
  “aud”:“did:key:zBR4m3DNZHT1G8Nb2RHzgKK7TrWxEmJjZskgvFdncTthzUH…”,
  “scp”:“/public/photos/covid2020/”,
  “pty”:”APPEND_ONLY”,
  “prf”:<JWT PROOF>,
  “exp”:1589423547
}
<SIGNATURE>
```

DELEGATED WRITE ACCESS

EACH LAYER FOLLOWS THIS FORM

```
{
  “alg”: “RS256”,
  “typ”: “JWT“,
  “cty”: “JWT“
}
{
  “iss”:“did:key:z1MdJPaWBebKxtE33AszRWYTF67wCLeFdcsqc3R87hyLKzBK…”,
  “aud”:“did:key:zBR4m3DNZHT1G8Nb2RHzgKK7TrWxEmJjZskgvFdncTthzUH…”,
  “scp”:“/public/photos/covid2020/”,
  “pty”:”APPEND_ONLY”,
  “prf”:<JWT PROOF>,
  “exp”:1589423547
}
<SIGNATURE>
```

Recursive

Problem: gets pretty big

DELEGATED WRITE ACCESS

HASHING IT DOWN

```
{
  “alg”: “RS256”,
  “typ”: “JWT“,
  “cty”: “JWT“
}
{
  “iss”:“did:key:z1MdJPaWBebKxtE33AszRWYTF67wCLeFdcsqc3R87hyLKzBK…”,
  “aud”:“did:key:zBR4m3DNZHT1G8Nb2RHzgKK7TrWxEmJjZskgvFdncTthzUH…”,
  “scp”:“/public/photos/covid2020/”,
  “pty”:”APPEND_ONLY”,
  “prf”:”QmaEmBULputJ5sAJX4bRQYwwWV2DUPnwNSz2R2eTvHV4DT”,
  “exp”:1589423547
}
<SIGNATURE>
```

DELEGATED WRITE ACCESS

HASHING IT DOWN

```
{
  “alg”: “RS256”,
  “typ”: “JWT“,
  “cty”: “JWT“
}
{
  “iss”:“did:key:z1MdJPaWBebKxtE33AszRWYTF67wCLeFdcsqc3R87hyLKzBK…”,
  “aud”:“did:key:zBR4m3DNZHT1G8Nb2RHzgKK7TrWxEmJjZskgvFdncTthzUH…”,
  “scp”:“/public/photos/covid2020/”,
  “pty”:”APPEND_ONLY”,
  “prf”:”QmaEmBULputJ5sAJX4bRQYwwWV2DUPnwNSz2R2eTvHV4DT”,
  “exp”:1589423547
}
<SIGNATURE>
```

PORTABLE COMPUTE

PORTABLE COMPUTE

⚡ ADD MORE POWER TO JS & WASM AND STIR 🤖

PORTABLE COMPUTE
DYNAMIC FAAS

PORTABLE COMPUTE
DYNAMIC FAAS

- Run everything locally by default
  - Good for devs with powerful machines
  - Slow for students with Chromebooks

PORTABLE COMPUTE
DYNAMIC FAAS

- Run everything locally by default
  - Good for devs with powerful machines
  - Slow for students with Chromebooks
- Farm out longer running computation to service providers
  - …dynamically at runtime

PORTABLE COMPUTE

DYNAMIC FAAS

- Run everything locally by default
  - Good for devs with powerful machines
  - Slow for students with Chromebooks
- Farm out longer running computation to service providers
  - …dynamically at runtime
- Heavy compute, parallel workloads, &c

PORTABLE COMPUTE

APPROACH & TRADE-OFFS

PORTABLE COMPUTE
APPROACH & TRADE-OFFS

- Code-as-data
- Memoization
- Compiler techniques at web scale (“world computer”)
- Network latency (normally zero, now x)
- Restricted subset (e.g. total)
- Event-based w/ two-phase commit
- Trusted (incl. AWS Lambda 🤷‍♀️)

PORTABLE COMPUTE
TOTALITY

PORTABLE COMPUTE
TOTALITY

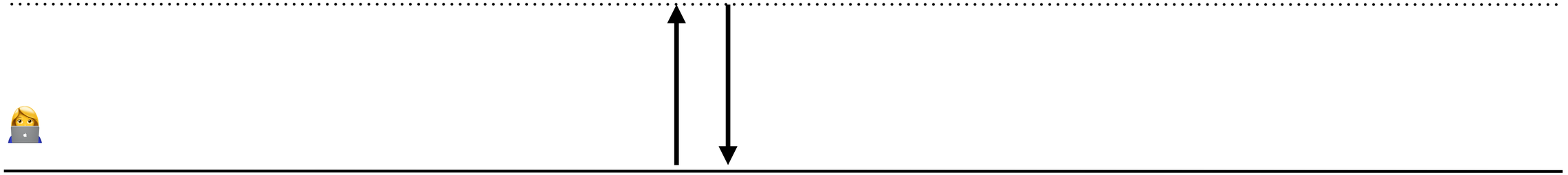
Side Effects

Pure Functions

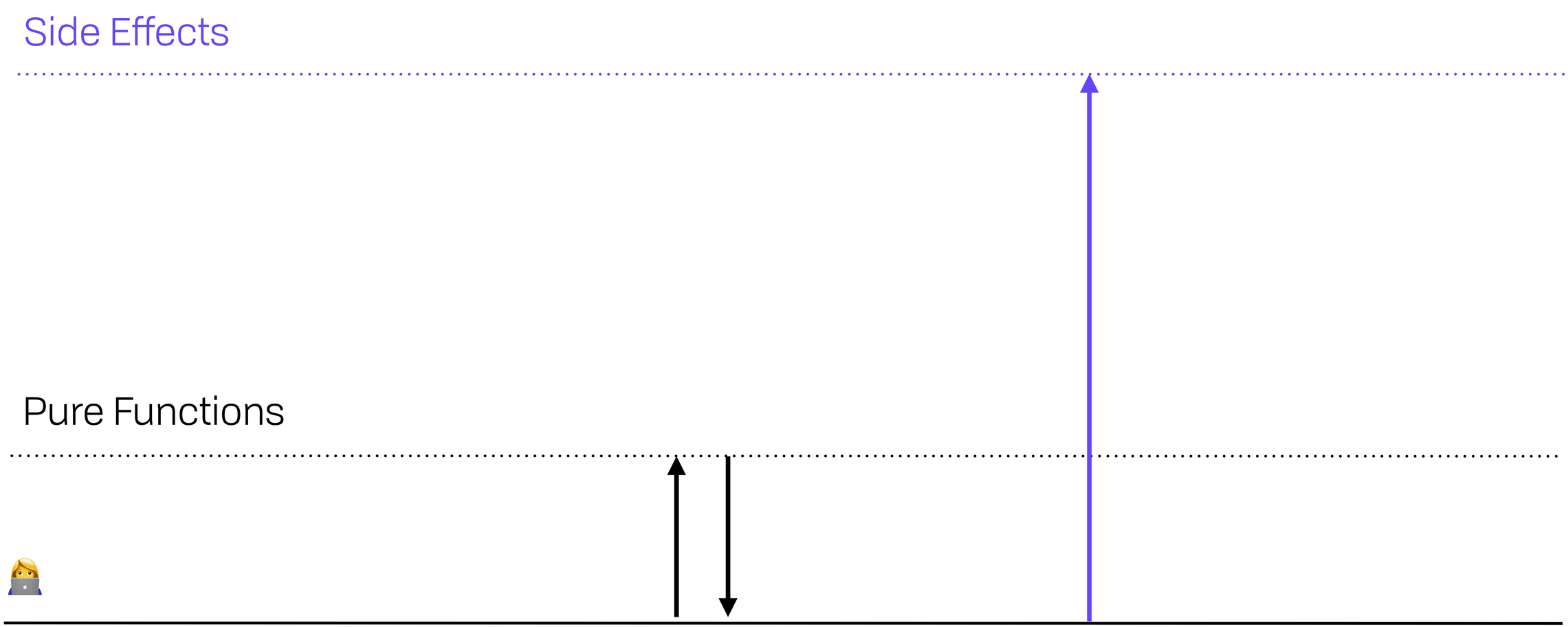
PORTABLE COMPUTE
TOTALITY

Side Effects

Pure Functions



PORTABLE COMPUTE
TOTALITY



PORTABLE COMPUTE

EVENT BASED (ABSTRACT USER STREAM, CRDTS)

PORTABLE COMPUTE

EVENT BASED (ABSTRACT USER STREAM, CRDTS)

Off-Platform Side Effect Stream

Platform Effect Stream

Pure Function Stream

Base Event Stream

PORTABLE COMPUTE

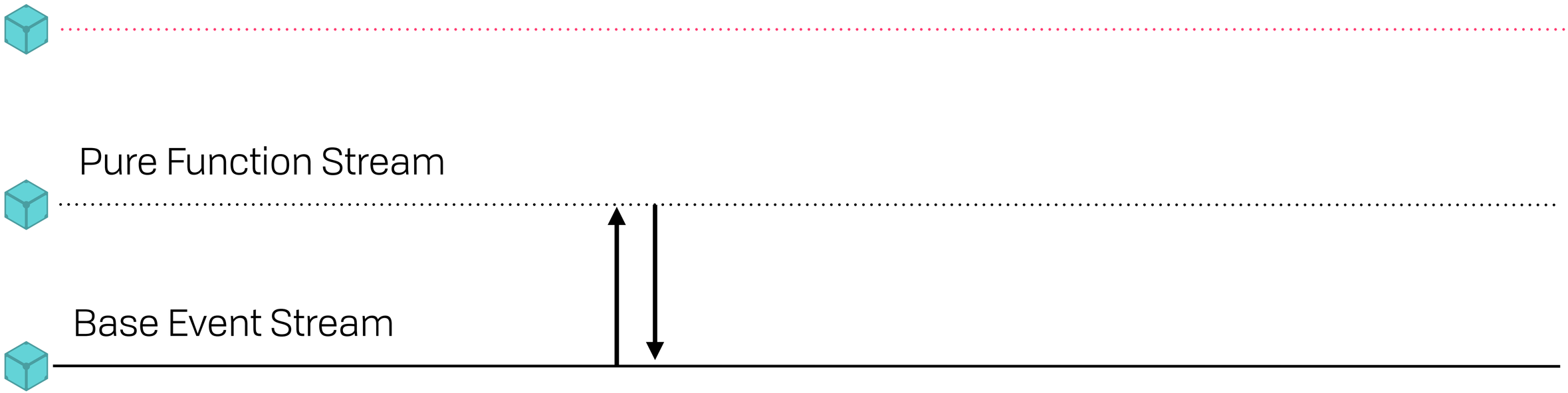
EVENT BASED (ABSTRACT USER STREAM, CRDTS)

Off-Platform Side Effect Stream

Platform Effect Stream

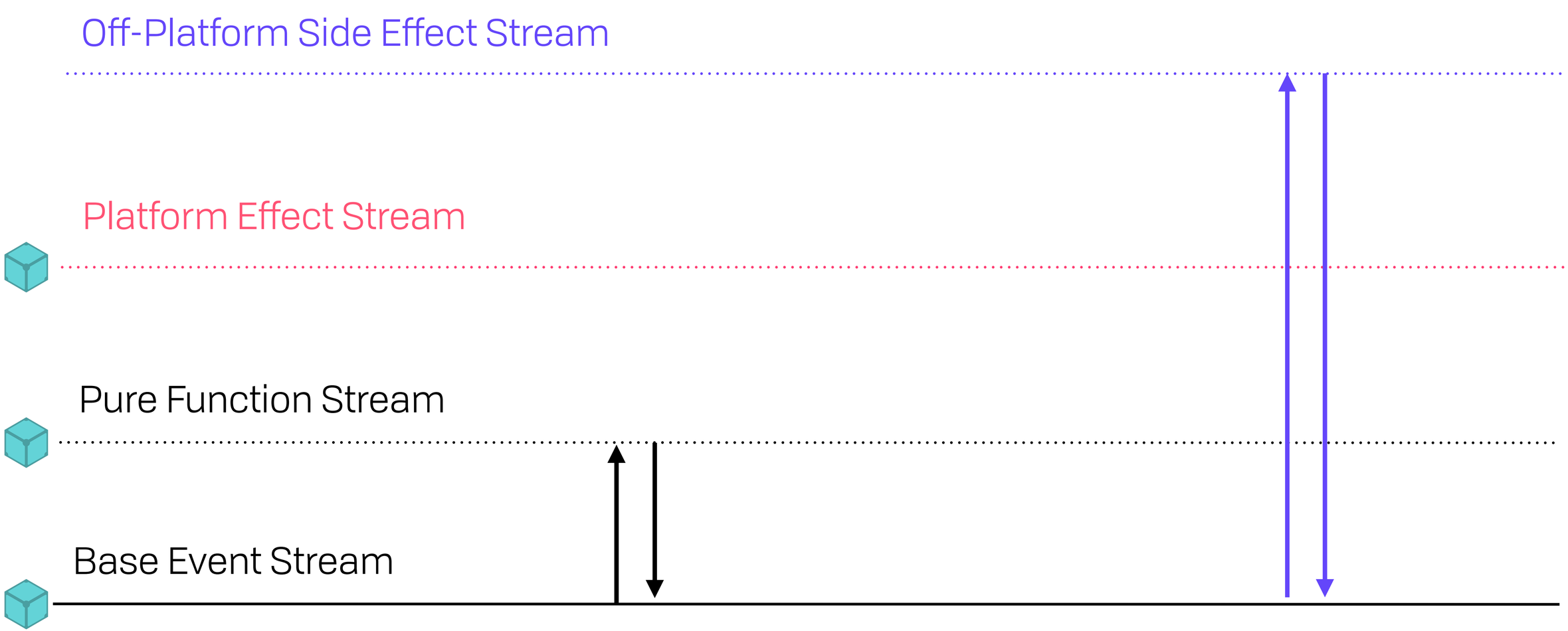
Pure Function Stream

Base Event Stream



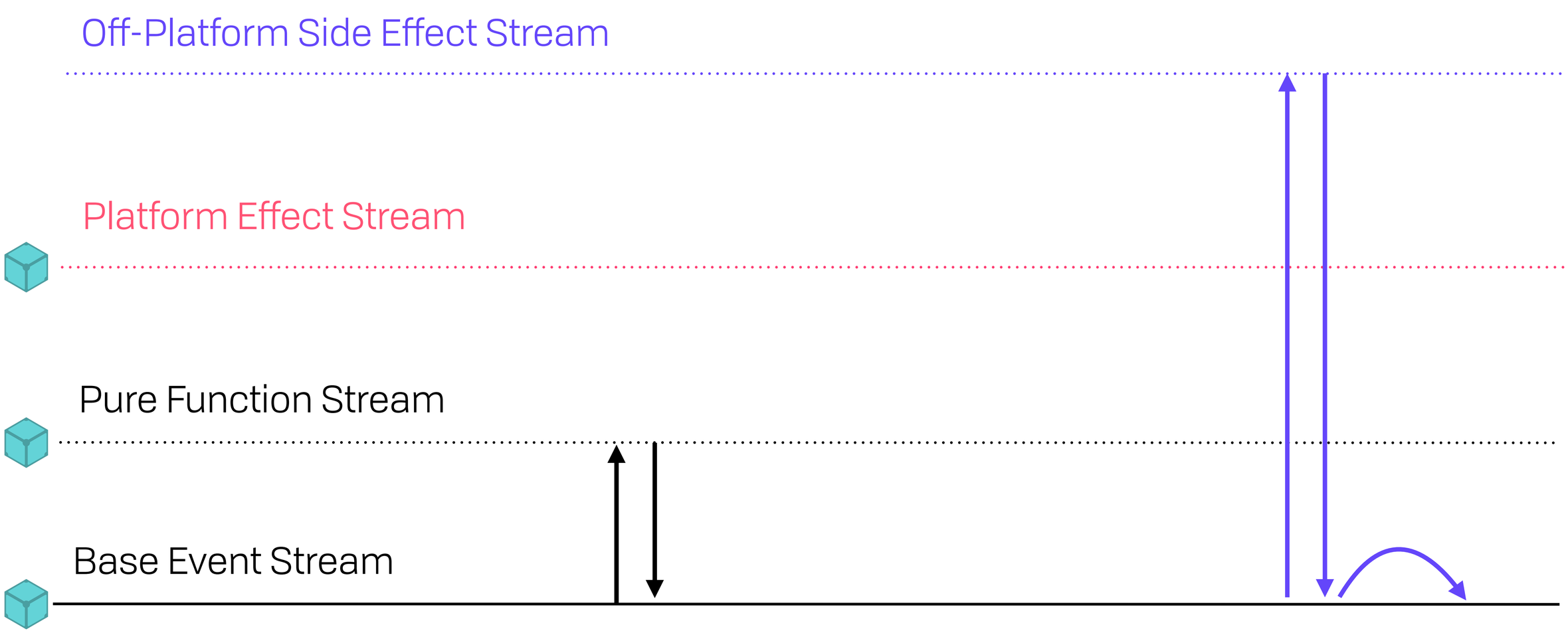
PORTABLE COMPUTE

EVENT BASED (ABSTRACT USER STREAM, CRDTS)



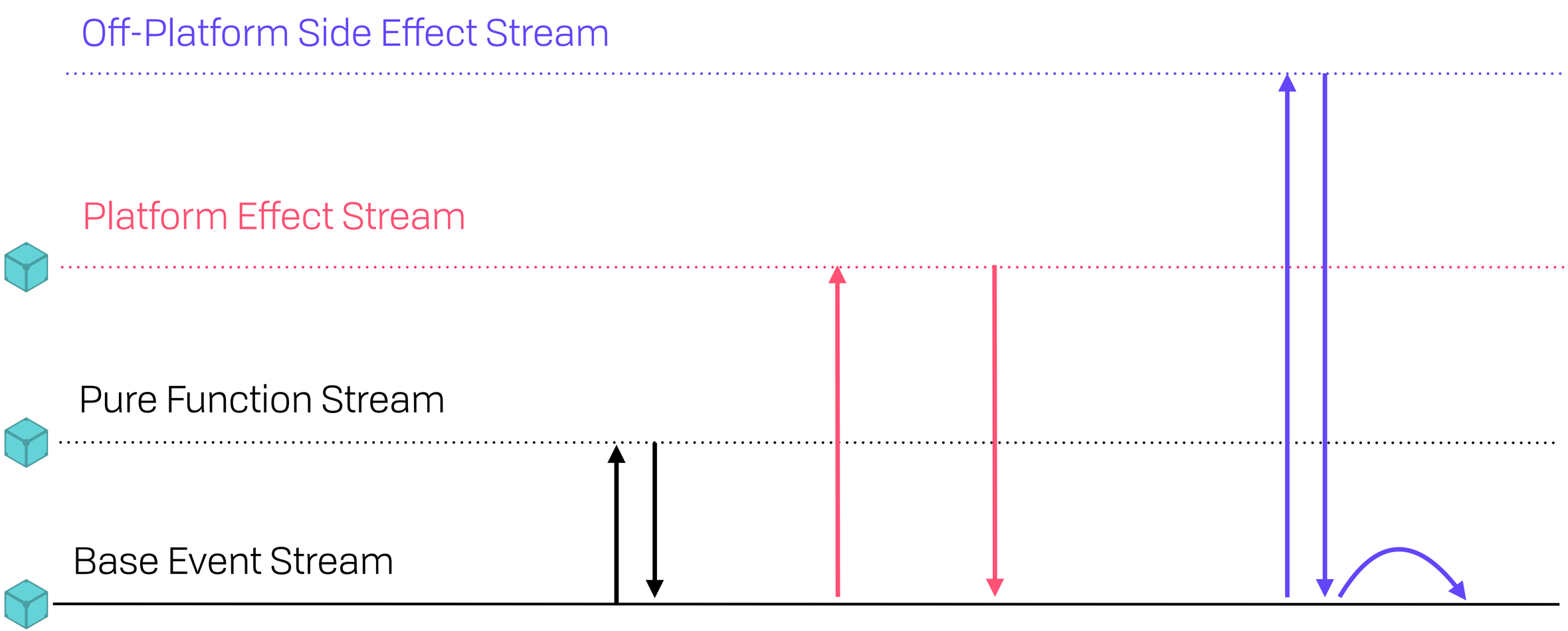
PORTABLE COMPUTE

EVENT BASED (ABSTRACT USER STREAM, CRDTS)



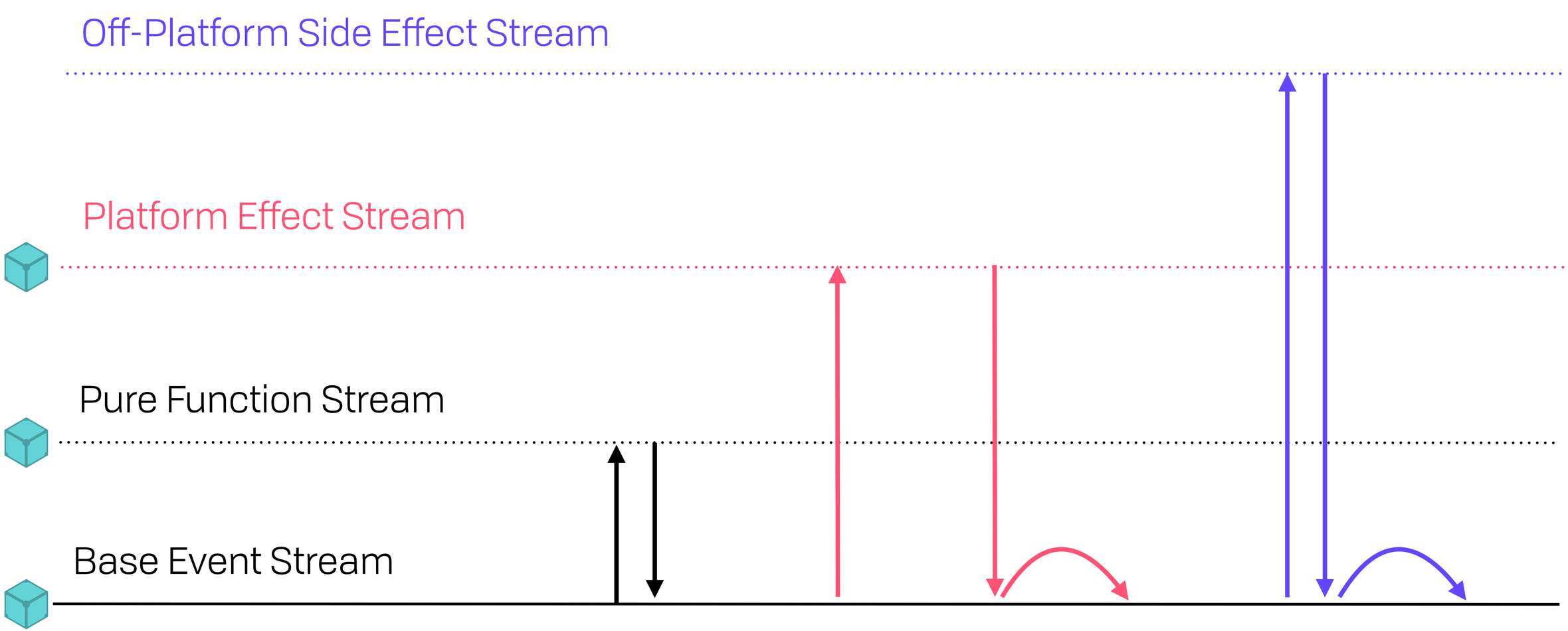
PORTABLE COMPUTE

EVENT BASED (ABSTRACT USER STREAM, CRDTS)



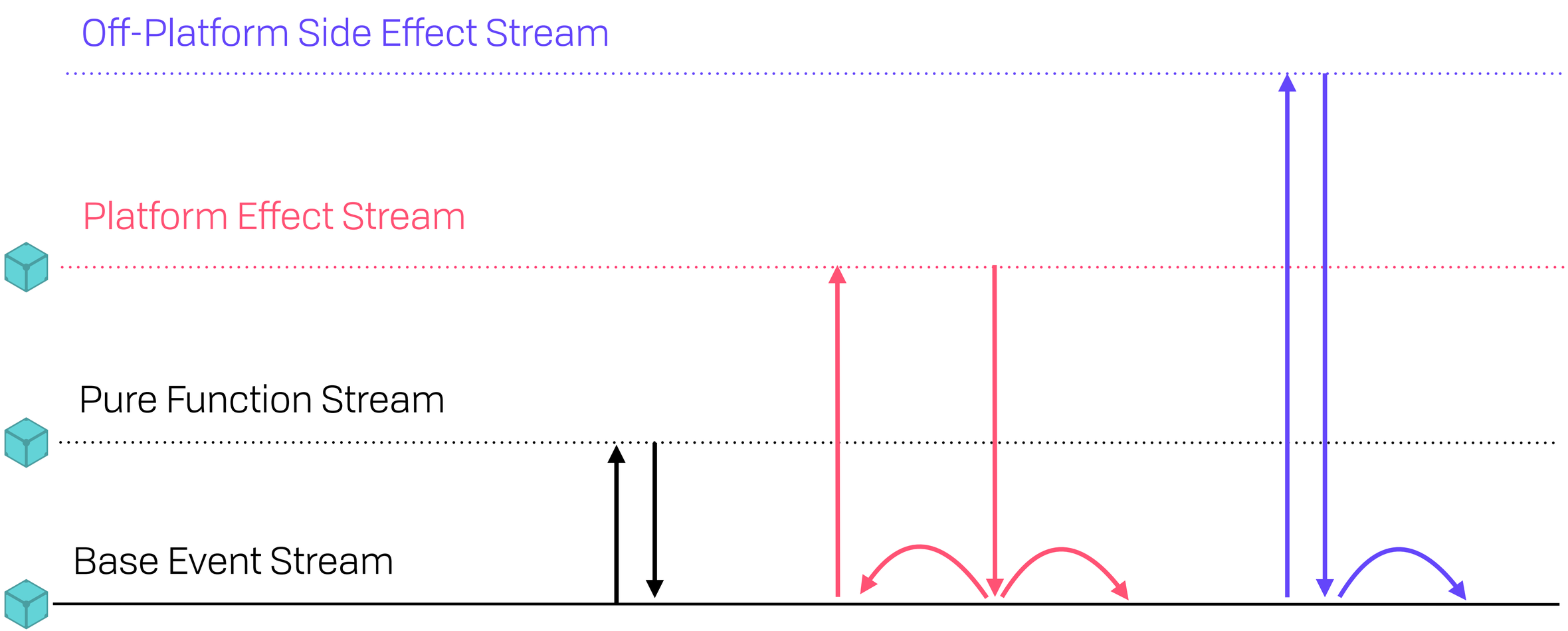
PORTABLE COMPUTE

EVENT BASED (ABSTRACT USER STREAM, CRDTS)



PORTABLE COMPUTE

EVENT BASED (ABSTRACT USER STREAM, CRDTS)



A UNIVERSAL HOSTLESS SUBSTRATE

RECAP

A UNIVERSAL HOSTLESS SUBSTRATE

RECAP

- Build and use fully locally
- No such thing as "deployment"
- Zero config
- Default: only serve the app, not data
- Efficient bandwidth

A UNIVERSAL HOSTLESS SUBSTRATE

RECAP

- Build and use fully locally
- No such thing as "deployment"
- Zero config
- Default: only serve the app, not data
- Efficient bandwidth

- One identity for all services
- Flexible client-side auth
- Military-grade security
- User owned data
- Share nothing with site/app by default
- Flexible FaaS without pre-deployment
- Offline-first and local-network aware

https://fission.codes
https://talk.fission.codes
https://discord.gg/zAQBDEq

🇩🇪 THANK YOU, BERLIN FP 🎉

brooklyn@fission.codes
github.com/expede
@expede

https://fission.codes
https://talk.fission.codes
https://discord.gg/zAQBDEq

🇩🇪 THANK YOU, BERLIN FP 🎉

brooklyn@fission.codes
github.com/expede
@expede