

Local-First Access Control

 **Trusted, Safe Collaboration Without Boundaries** 

Local-First Access Control



 **Trusted, Safe Collaboration Without Boundaries** 

Local-First Access Control

Brooklyn Zelenka @expede

Local-First Access Control

Brooklyn Zelenka @expede



github.com/expede

Local-First Access Control

Brooklyn Zelenka @expede

- ♦ Beehive (auth) project lead at Ink & Switch 🐝
 - ♦ Team: John Mumm & Alex Good
 - ♦ Many community collaborators including DXOS, Serenity, Matrix Research, Herb Caudill, Martin Kleppmann, &c
- ♦ Spec editor at UCAN Working Group
- ♦ Prev. Ethereum core dev, mostly EVM but also access control
- ♦ PLs and DSys are my jam 🙌



github.com/expede

What we're not doing right now.

But maybe someone here will figure it out.

- Peer-to-peer sync. (WebRTC requires servers & P2P is unreliable.)
- E2E encryption. (There are trade-offs here but it's doable.)
- Schema migration. (See: Project Cambria.)
- Native mobile apps. (We're living with the web.)



What we're not doing right now.

But maybe someone here will figure it out.

- Peer-to-peer sync. (WebRTC requires servers & P2P is unreliable.)
- E2E encryption. (There are trade-offs here but it's doable.)
- Schema migration. (See: Project Cambria.)
- Native mobile apps. (We're living with the web.)



Local-First Access Control



Local-First Access Control

Cryptography is a tool for turning
lots of different problems into
key management problems

Dr. Lea Kissner, Global Lead of Privacy Technologies at Google

Local-First Access Control



Say the line, Brooke

Cryptogr
lots of c
key man

Cryptography is a tool for turning
lots of different problems into
key management problems

for turning
ms into
blems

Dr. Lea Kissner, C



ologies at Google

Local-First Access Control

Buckle Up

Local-First Access Control

Buckle Up

- ♦ **Context**

- ♦ The cloud approach
- ♦ How local-first is different
- ♦ An emerging area!
- ♦ Spread ideas!

Local-First Access Control

Buckle Up

♦ **Context**

- ♦ The cloud approach
- ♦ How local-first is different
- ♦ An emerging area!
- ♦ Spread ideas!

♦ **Overview of core Beehive design**

- ♦ (Slightly into the weeds because it's novel 😊)
- ♦ Mutation Control
 - ♦ Convergent capabilities
 - ♦ Concurrent revocation
- ♦ Read Control
 - ♦ Causal encryption
 - ♦ Continuous group key agreement (CGKA)
- ♦ A good idea from UCAN™

Local-First Access Control

Buckle Up

♦ **Context**

- ♦ The cloud approach
- ♦ How local-first is different
- ♦ An emerging area!
- ♦ Spread ideas!

♦ **Overview of core Beehive design**

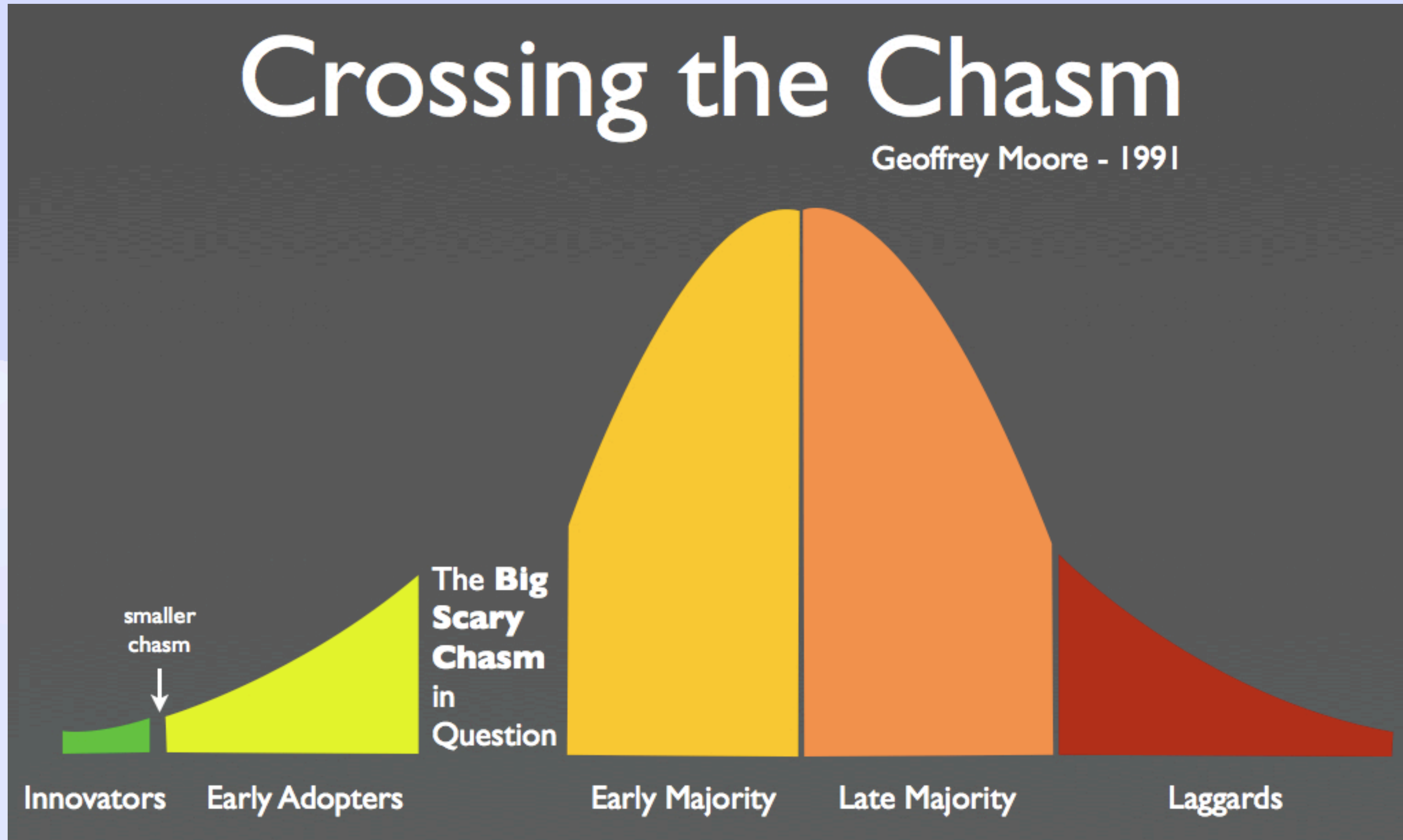
- ♦ (Slightly into the weeds because it's novel 😊)
- ♦ Mutation Control
 - ♦ Convergent capabilities
 - ♦ Concurrent revocation
- ♦ Read Control
 - ♦ Causal encryption
 - ♦ Continuous group key agreement (CGKA)
- ♦ A good idea from UCAN™

♦ **Not covered**

- ♦ Sync
 - ♦ Beelay, PRIBLT, Sedimentree
- ♦ Other systems
 - ♦ e.g. UCAN, WNFS, EIP-1066
- ♦ Deep crypto
- ♦ Future fanciness
 - ♦ FHE, ZKPs, PQC

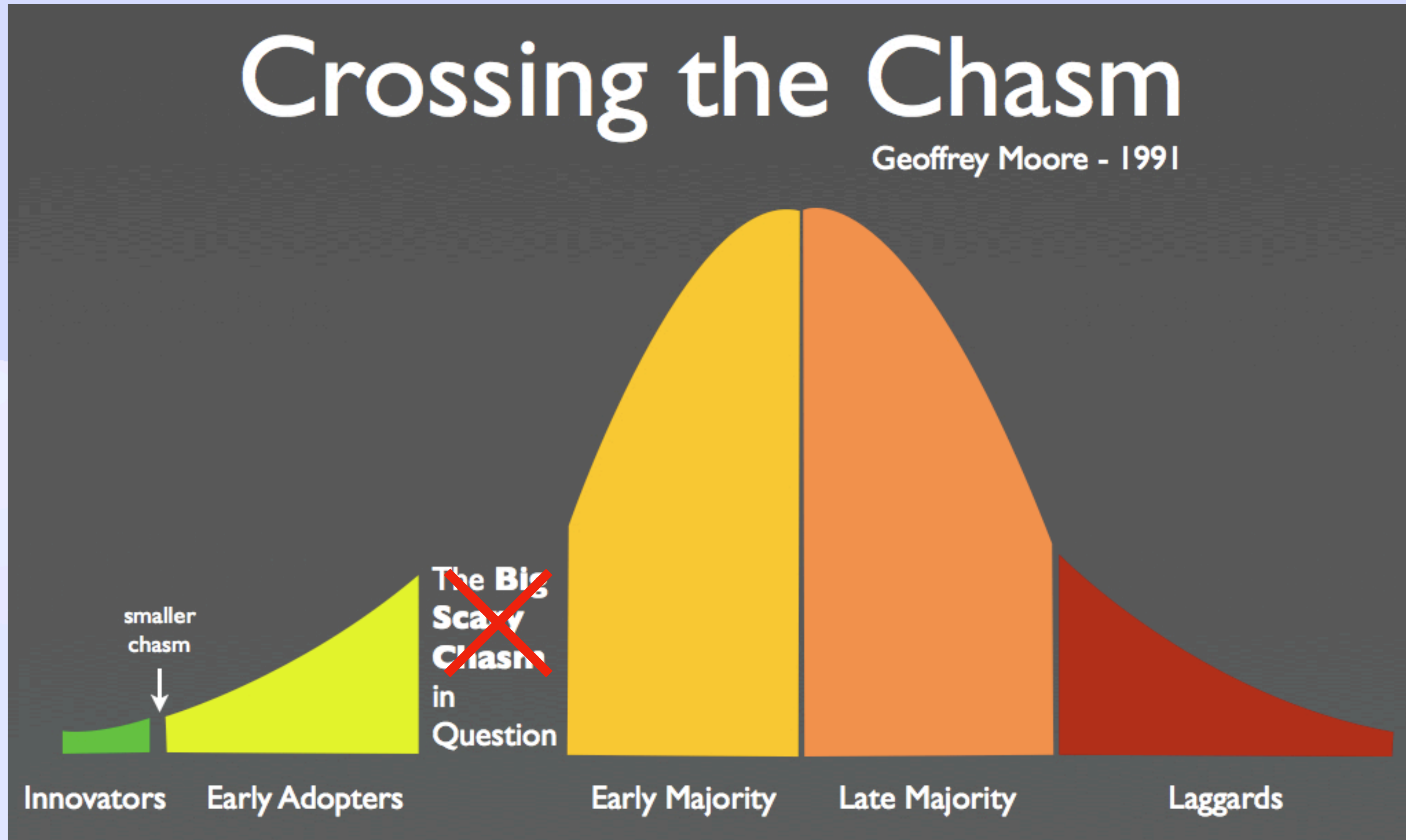
Local-First Access Control

Why Now?



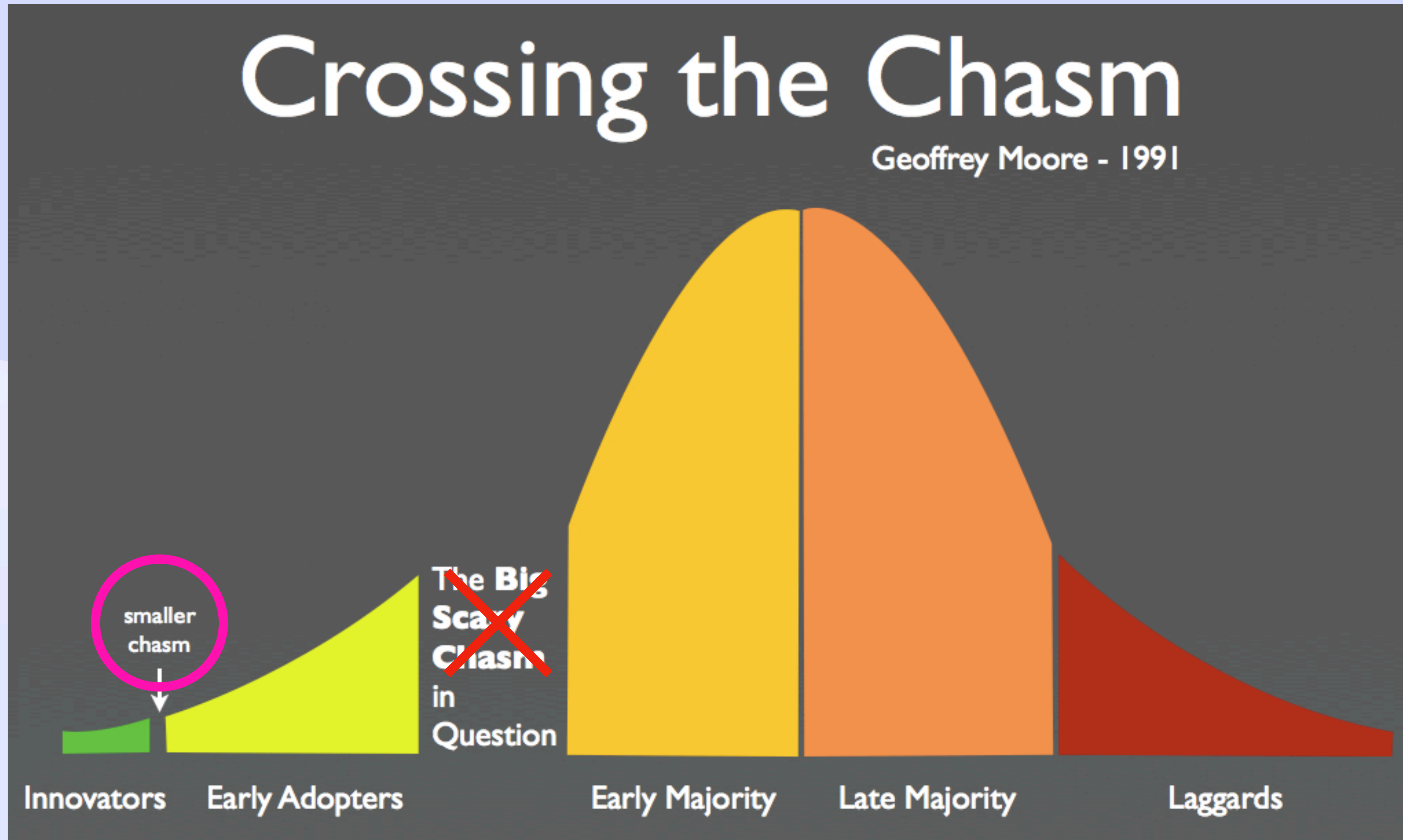
Local-First Access Control

Why Now?



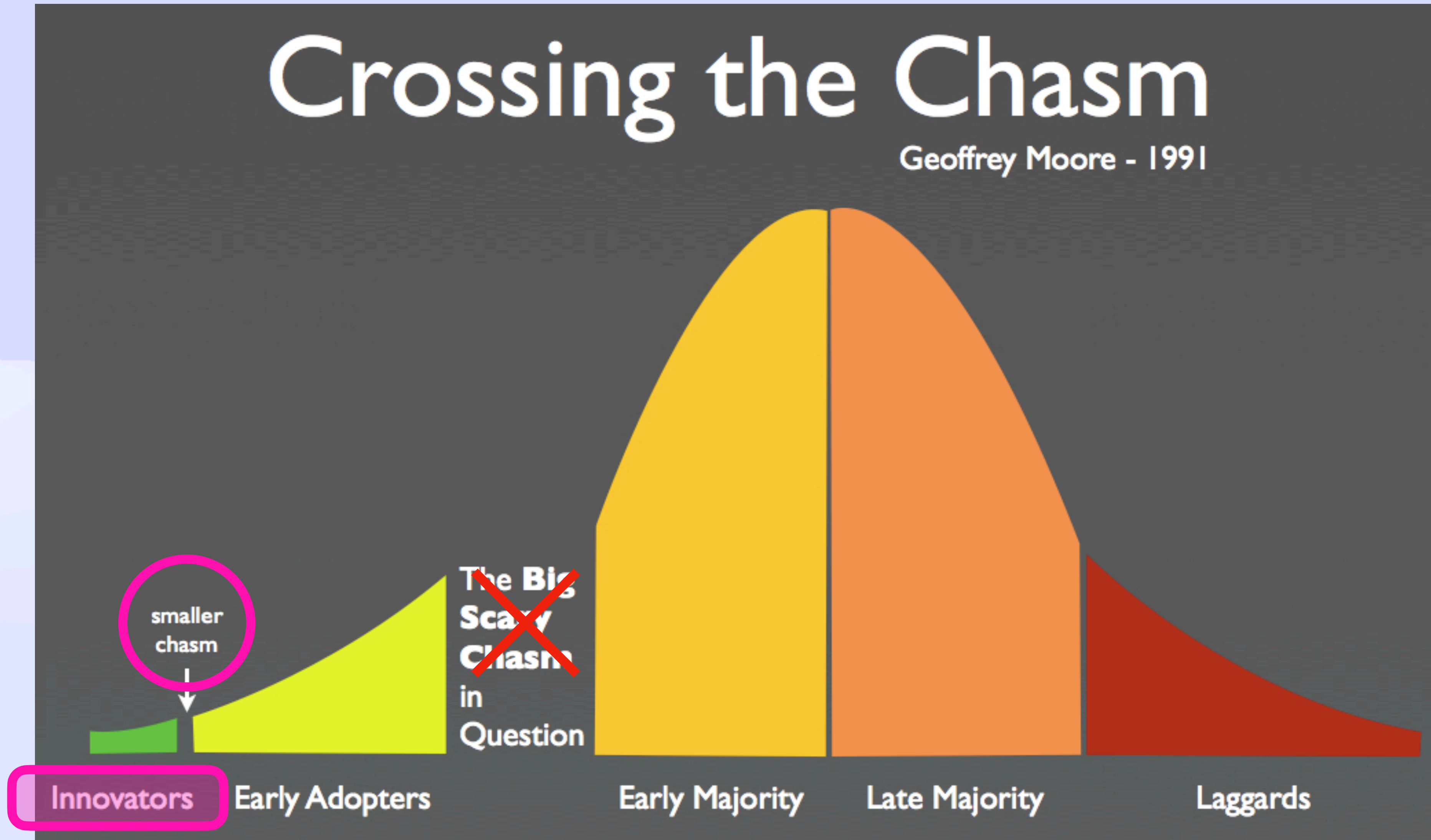
Local-First Access Control

Why Now?



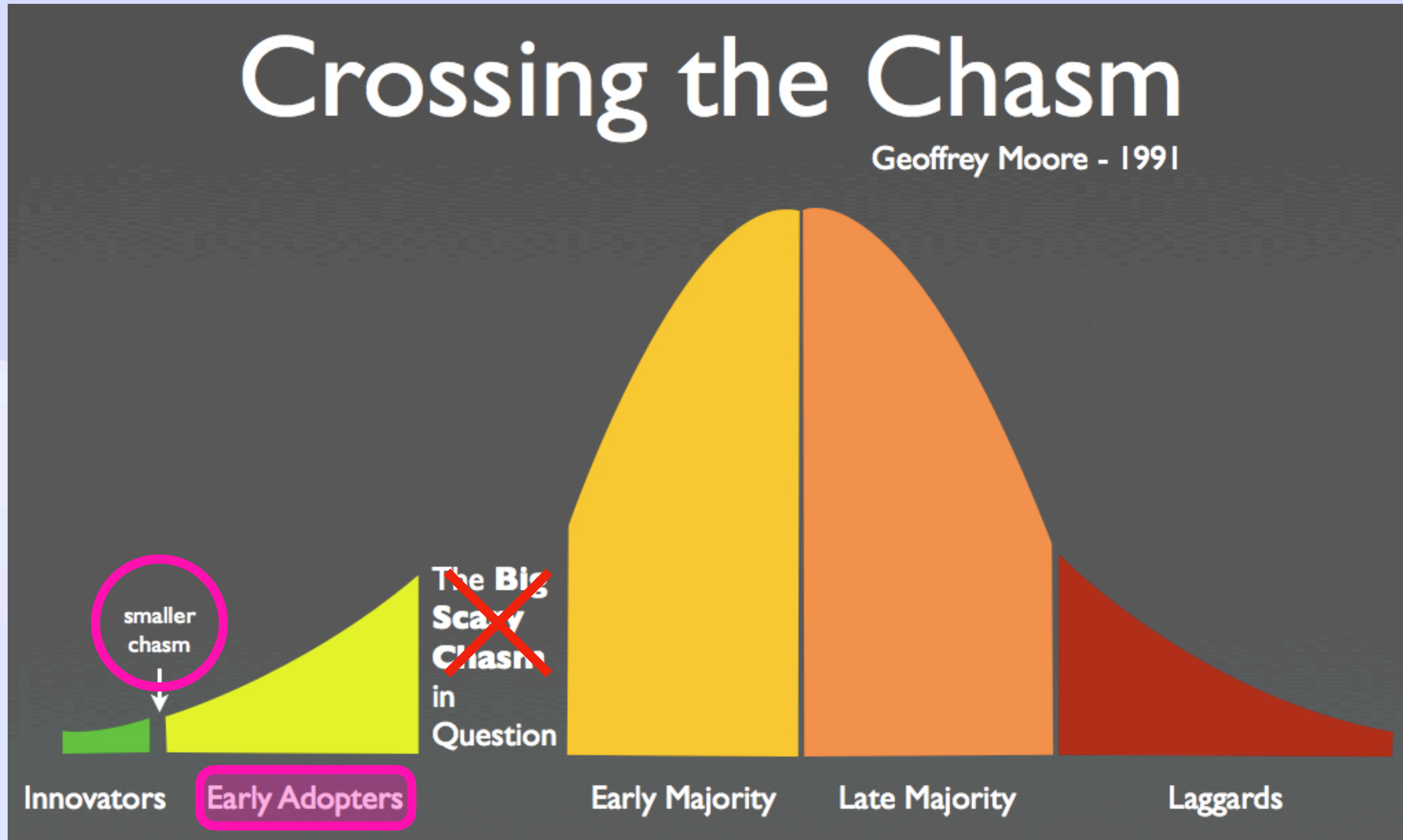
Local-First Access Control

Why Now?



Local-First Access Control

Why Now?



Local-First Access Control

Command & Control

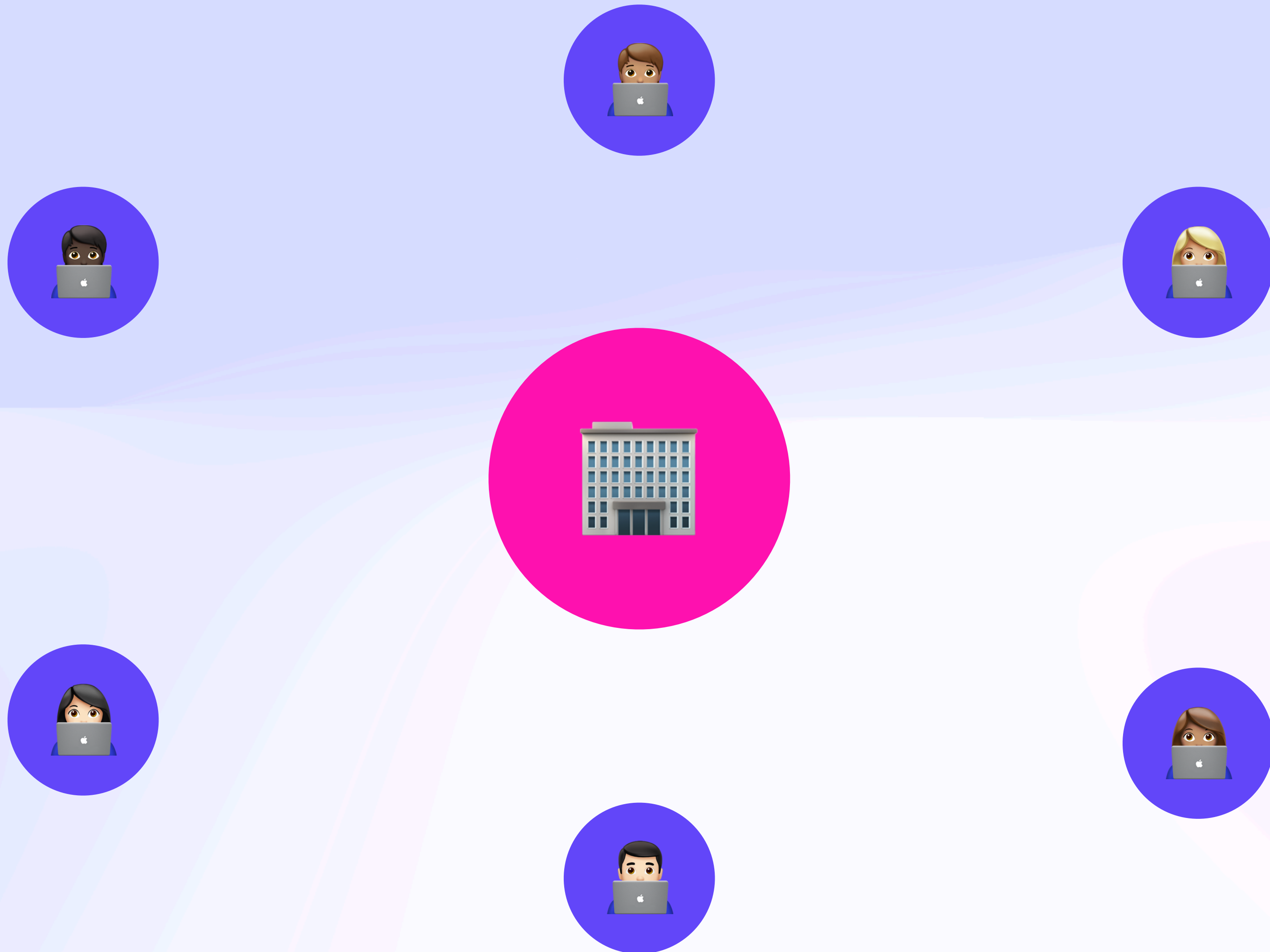
Local-First Access Control

Command & Control



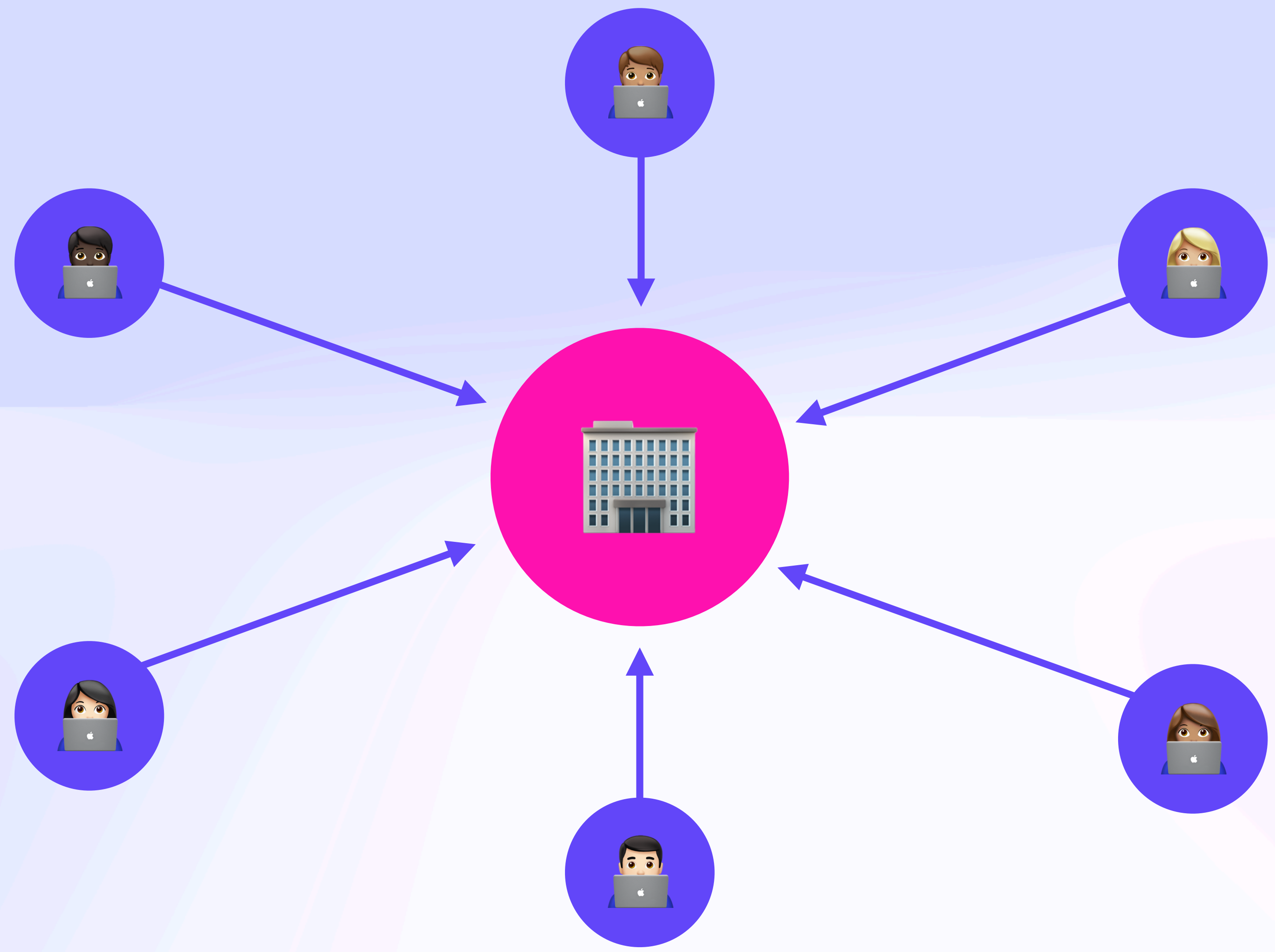
Local-First Access Control

Command & Control



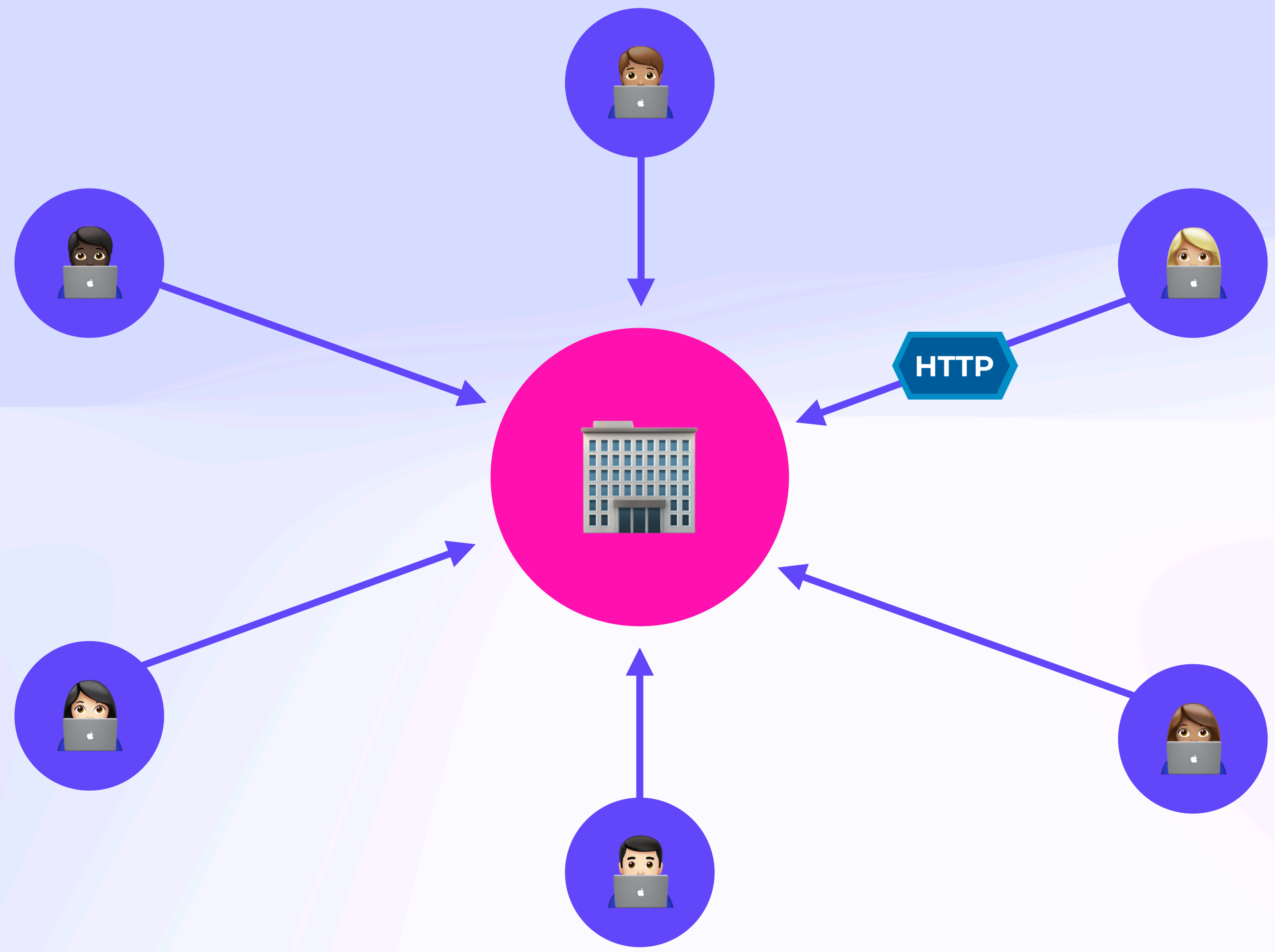
Local-First Access Control

Command & Control



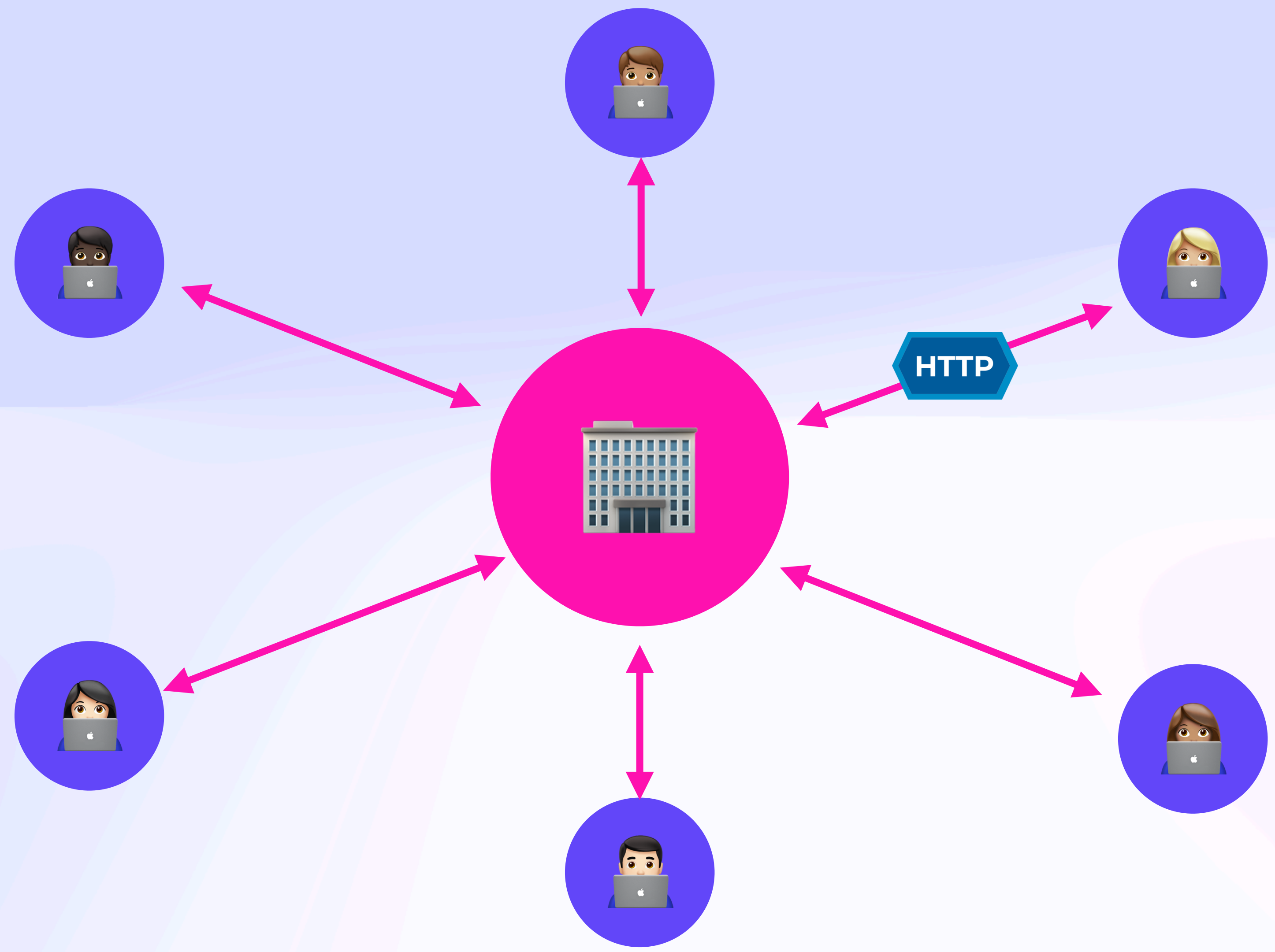
Local-First Access Control

Command & Control



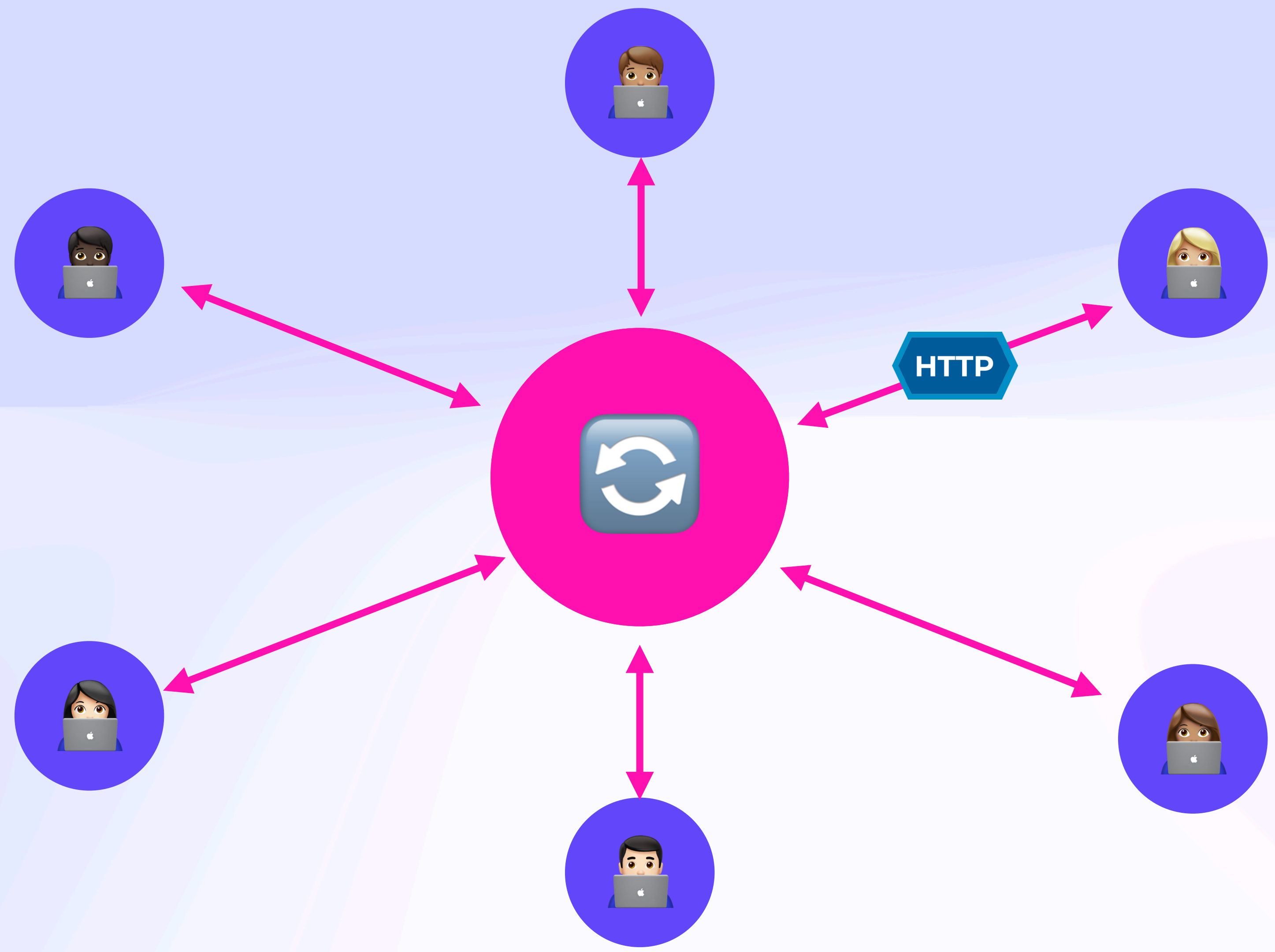
Local-First Access Control

Hive Dynamics



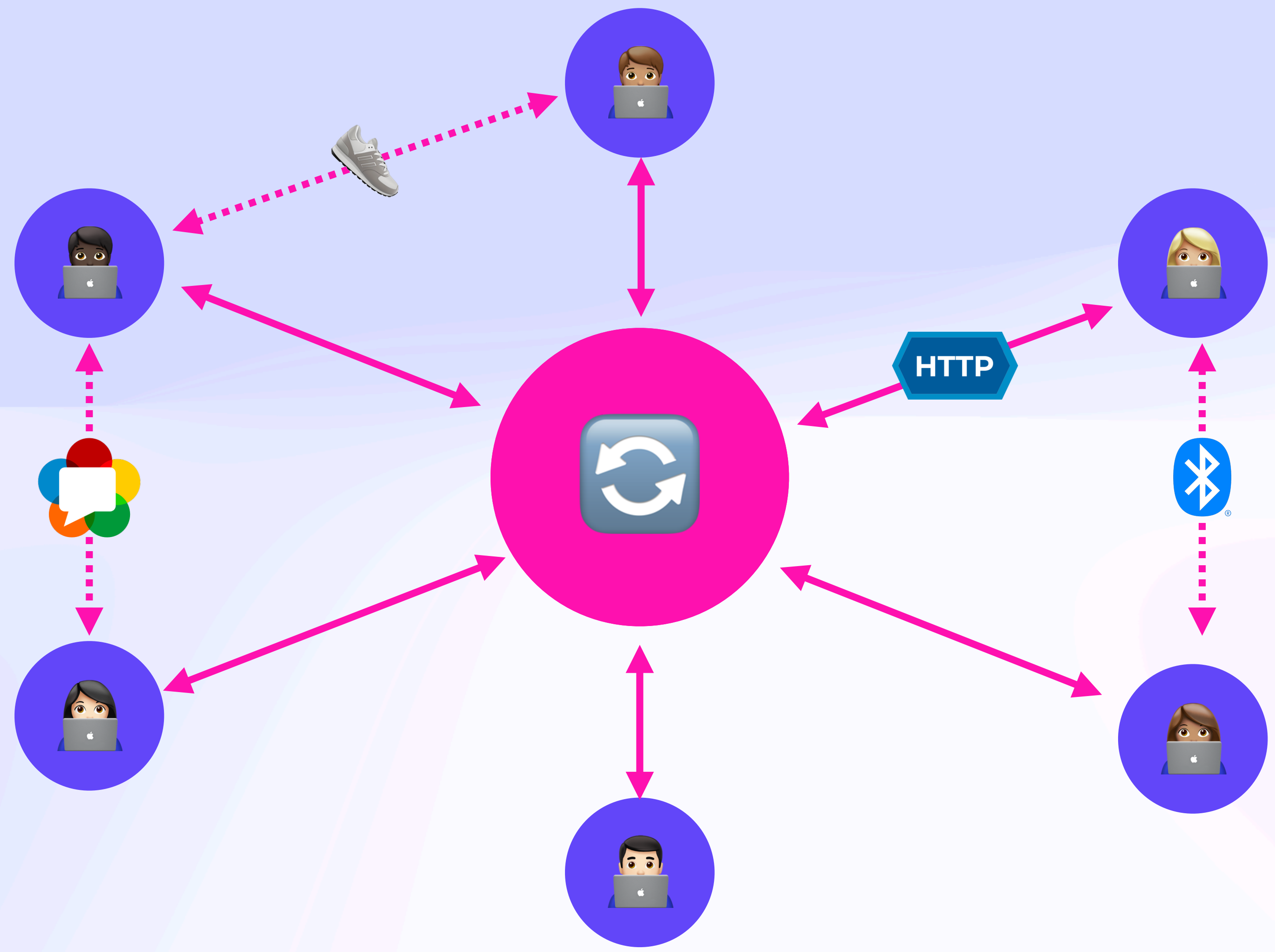
Local-First Access Control

Hive Dynamics



Local-First Access Control

Hive Dynamics



Local-First Access Control

The Highest of Levels

Local-First Access Control

The Highest of Levels

- Comparable **experience** to cloud auth, minus the servers
- **Control** who can read and write to documents
- Compatible with swappable/**interoperable** sync servers
 - Not store data in plaintext on those servers (encryption at rest)
 - ...but still be efficient: support Automerge compaction
- **"Wikipedia-scale"** (100k+ docs, 10k+ E2EE readers, 1k+ writers, 100s admins)
- Secure "at least at the level of the FBI, but not necessarily NSA, MSS, or Mossad"

Local-First Access Control

"Comparable Experience"

Local-First Access Control

"Comparable Experience"

Manage access

Create teamAdd peopleAdd teams

Direct accessOrganization access

Select all

role:admin

TypeRole

alexjg

Role: admin

beehive

@inkandswitch/beehive • 3 members

Role: admin

Brooklyn Zelenka

expede

Role: admin

web

@inkandswitch/web • 4 members

Role: admin

Add people, groups, and calendar events

People with access

Owner

hello@katiewilde.com

hello@katiewilde.com

Brooklyn Zelenka (you)

hello@brooklynzelenka.com

Editor



General access

Restricted

Only people with access can open with the link

Copy link

Done

 Single Source of Truth 

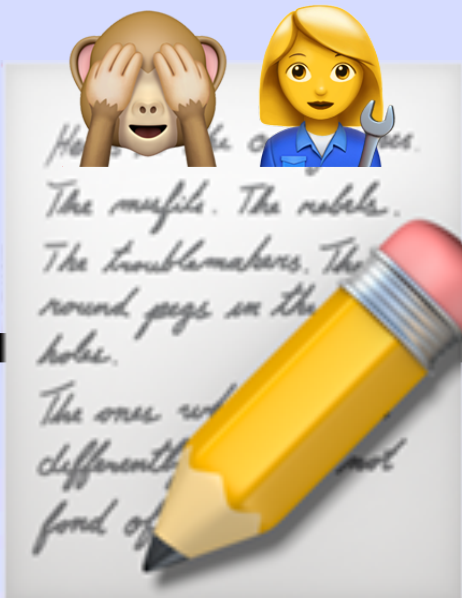
Cloud Auth

Cloud Auth

Cloud Auth Flow ☁️

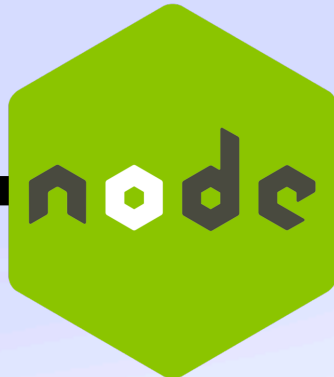
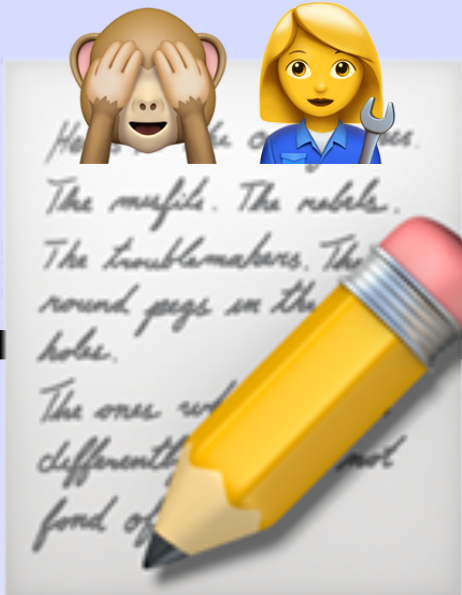
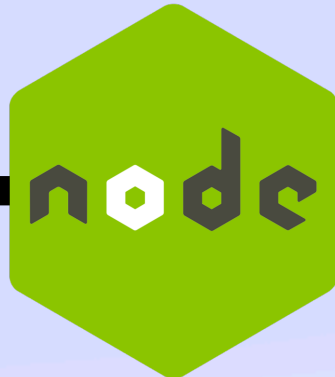
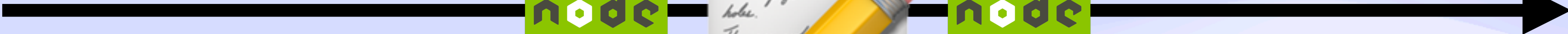
Cloud Auth

Cloud Auth Flow ☁️



Cloud Auth

Cloud Auth Flow ☁️



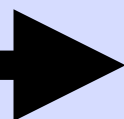
Cloud Auth

Cloud Auth Flow ☁️



Cloud Auth

Cloud Auth Flow ☁️



Cloud Auth

Cloud Auth Flow ☁️



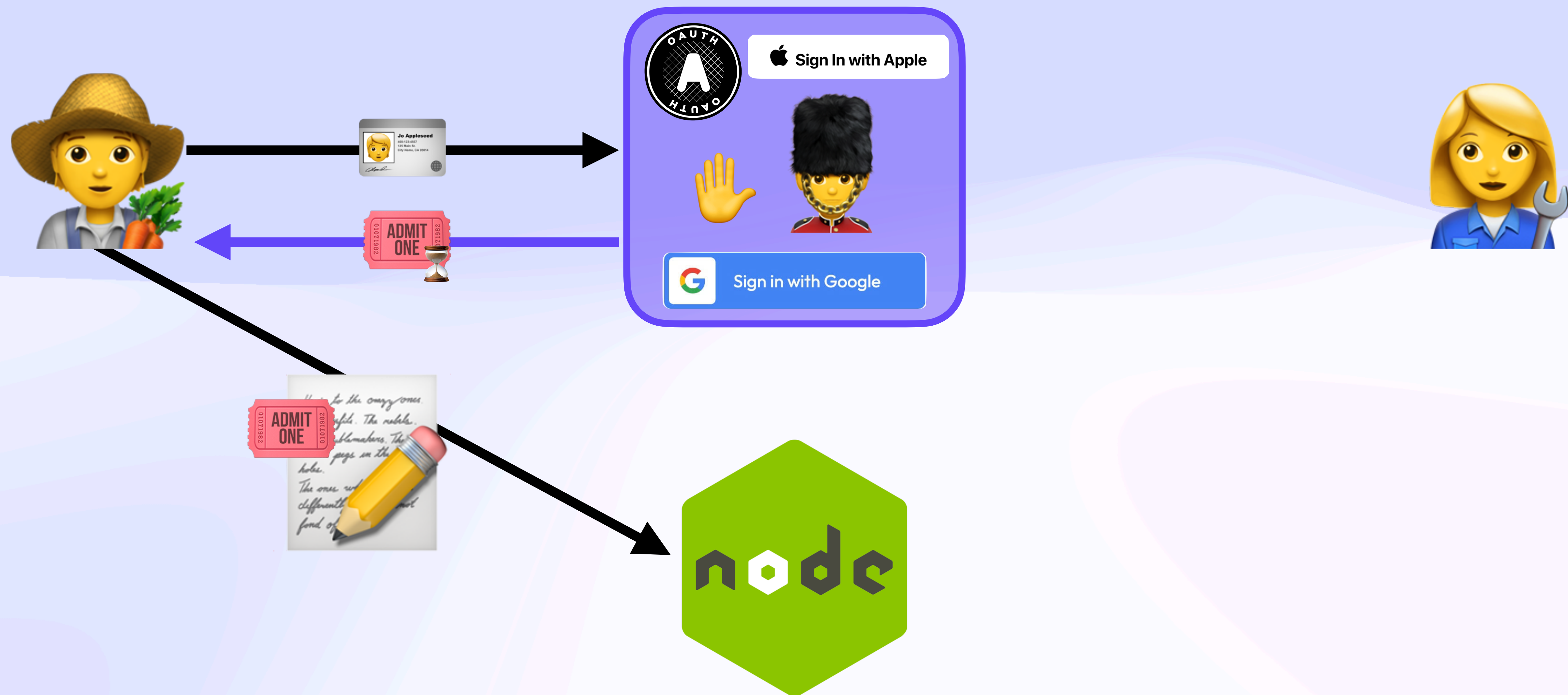
Cloud Auth

Cloud Auth Flow



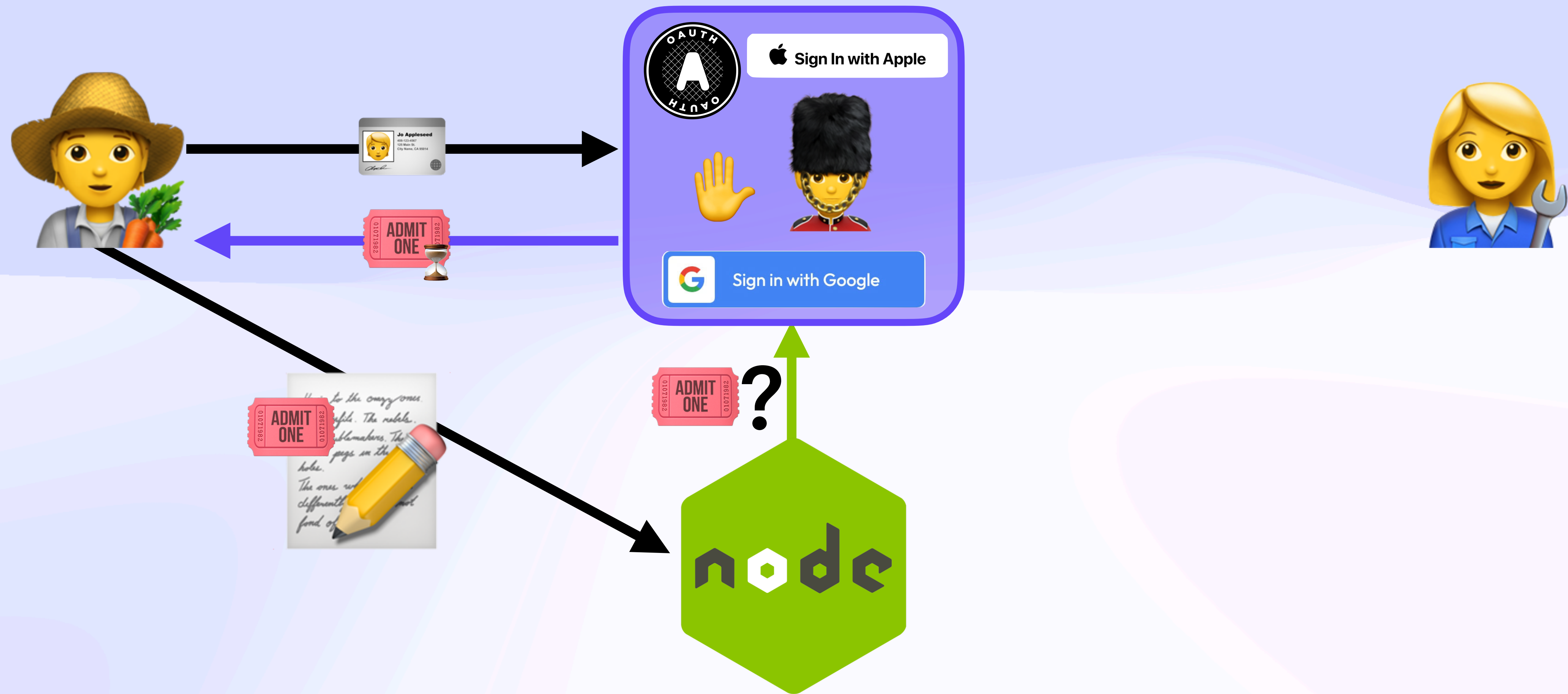
Cloud Auth

Cloud Auth Flow ☁️



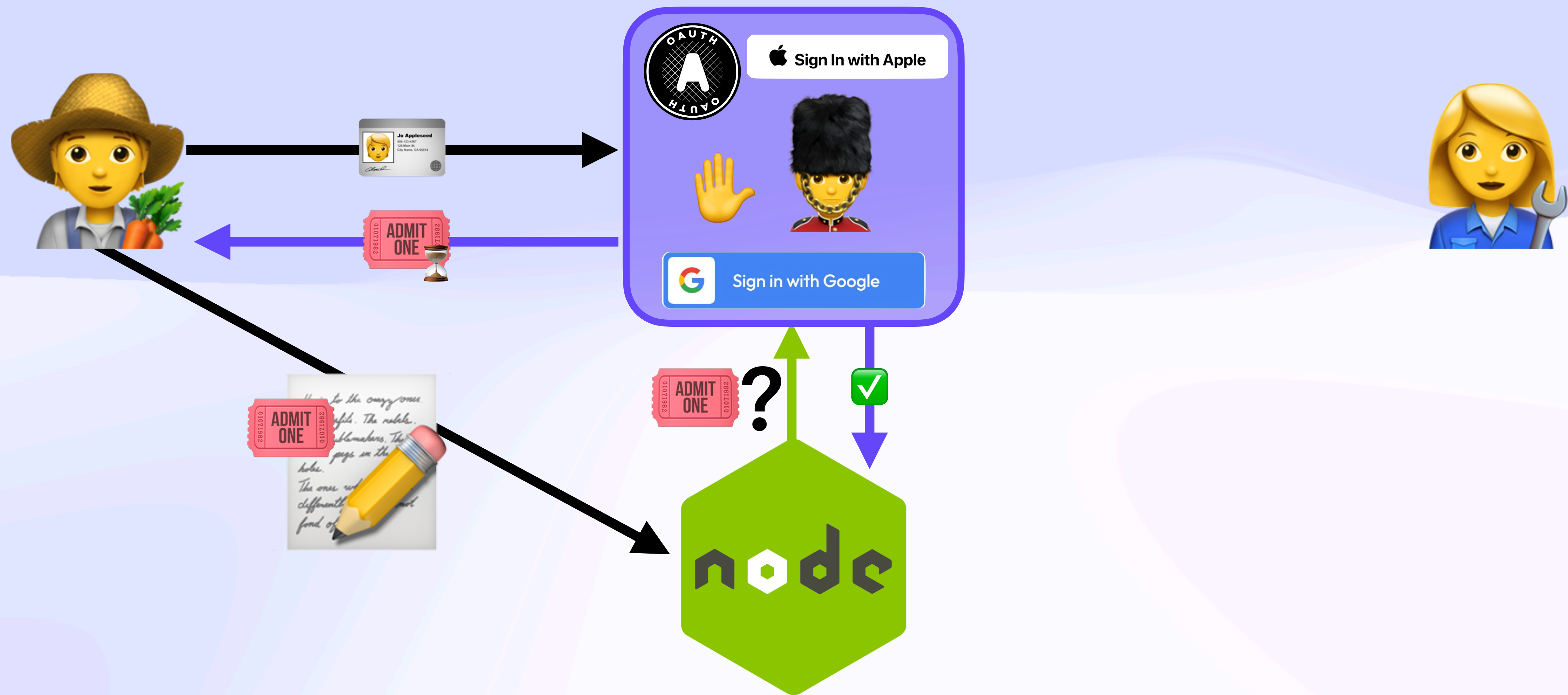
Cloud Auth

Cloud Auth Flow



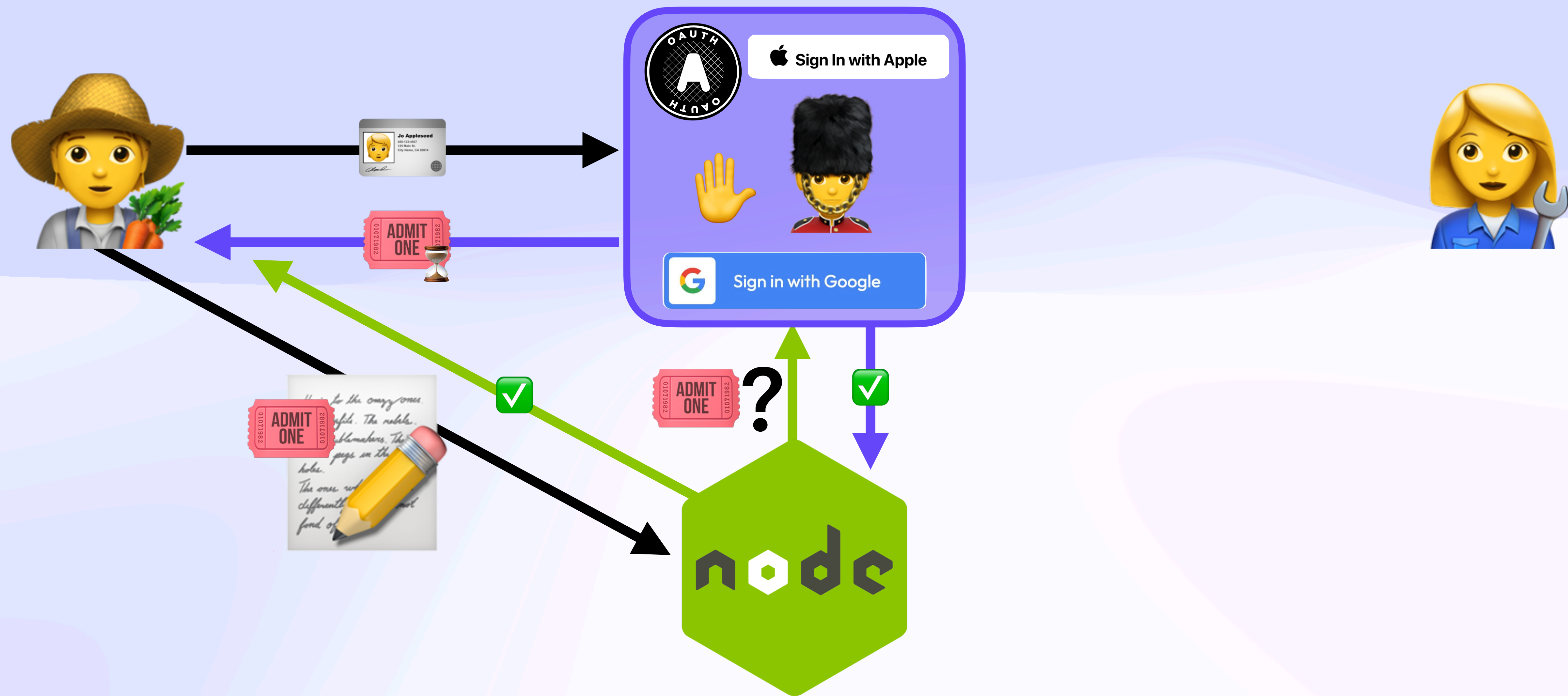
Cloud Auth

Cloud Auth Flow



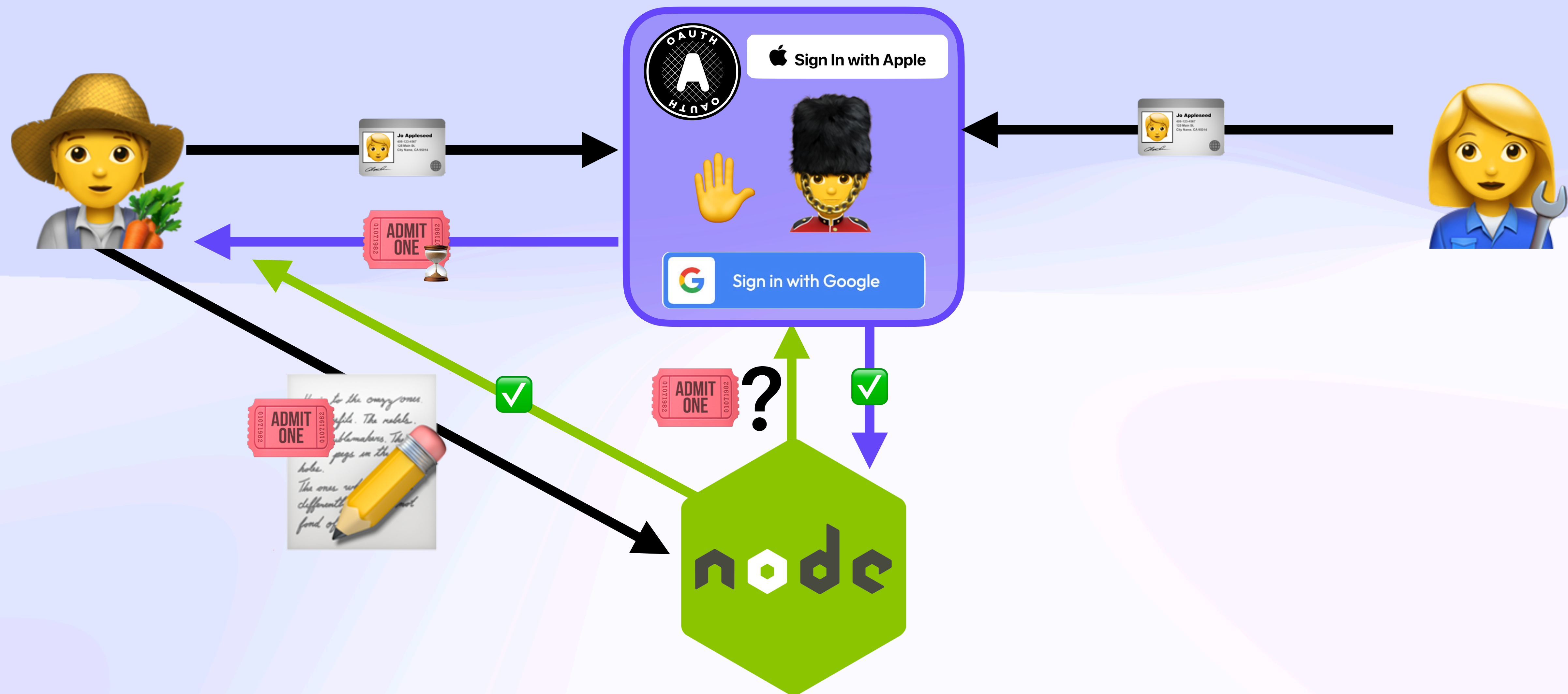
Cloud Auth

Cloud Auth Flow ☁️



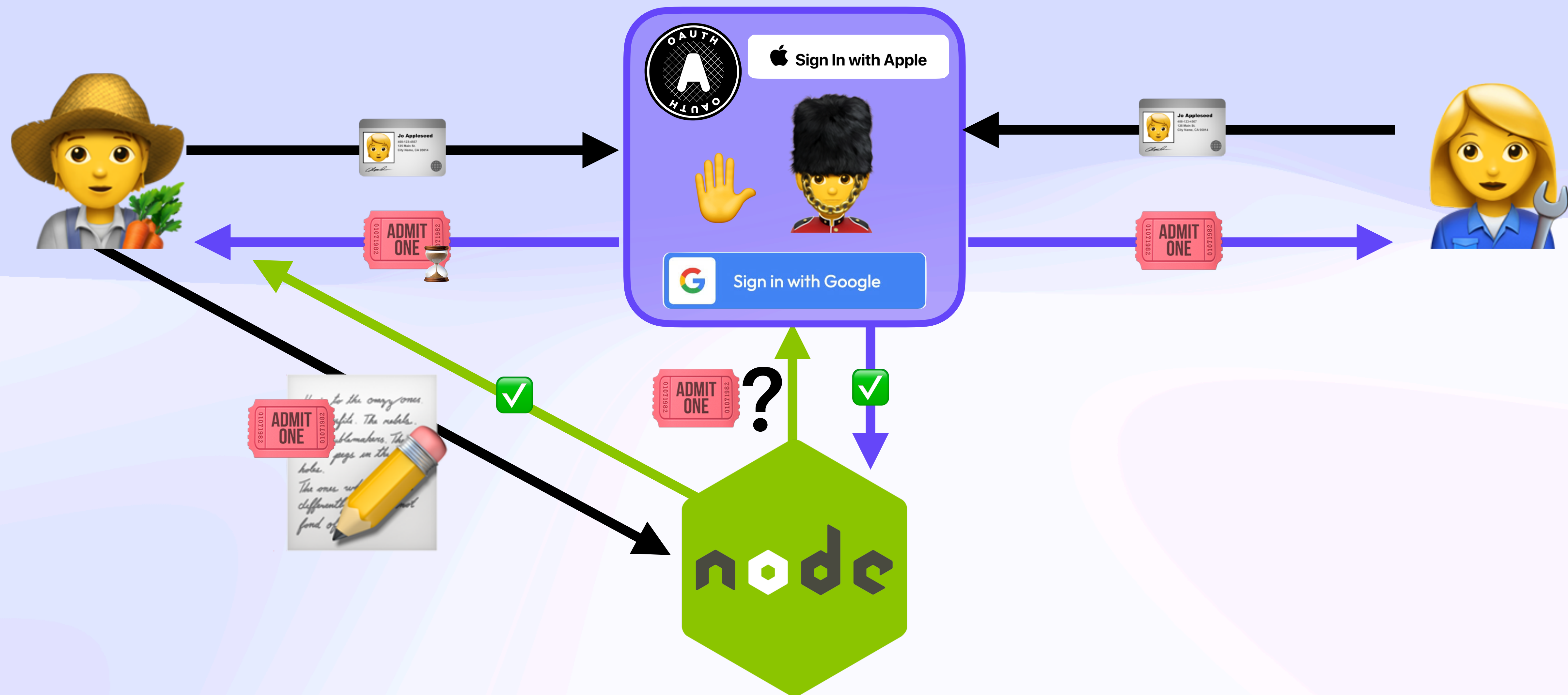
Cloud Auth

Cloud Auth Flow ☁️



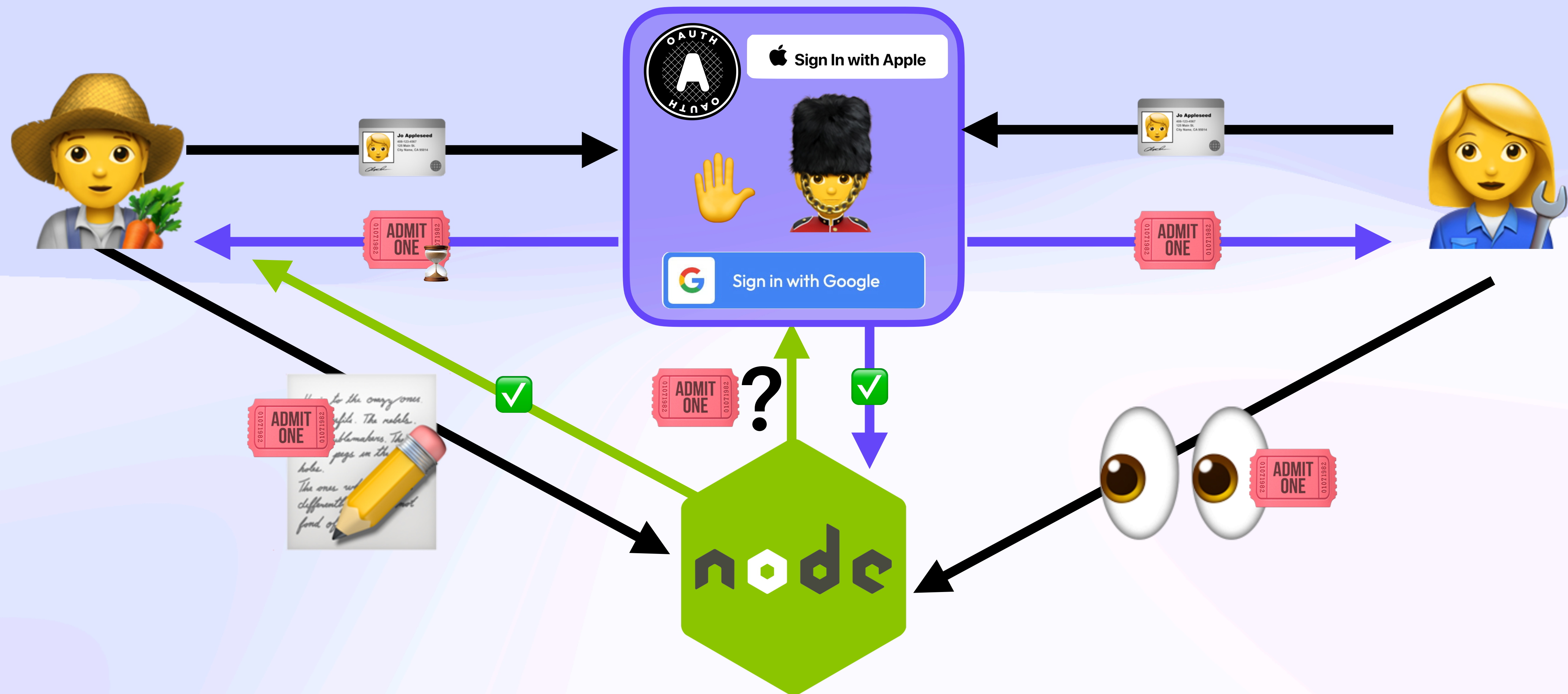
Cloud Auth

Cloud Auth Flow ☁️



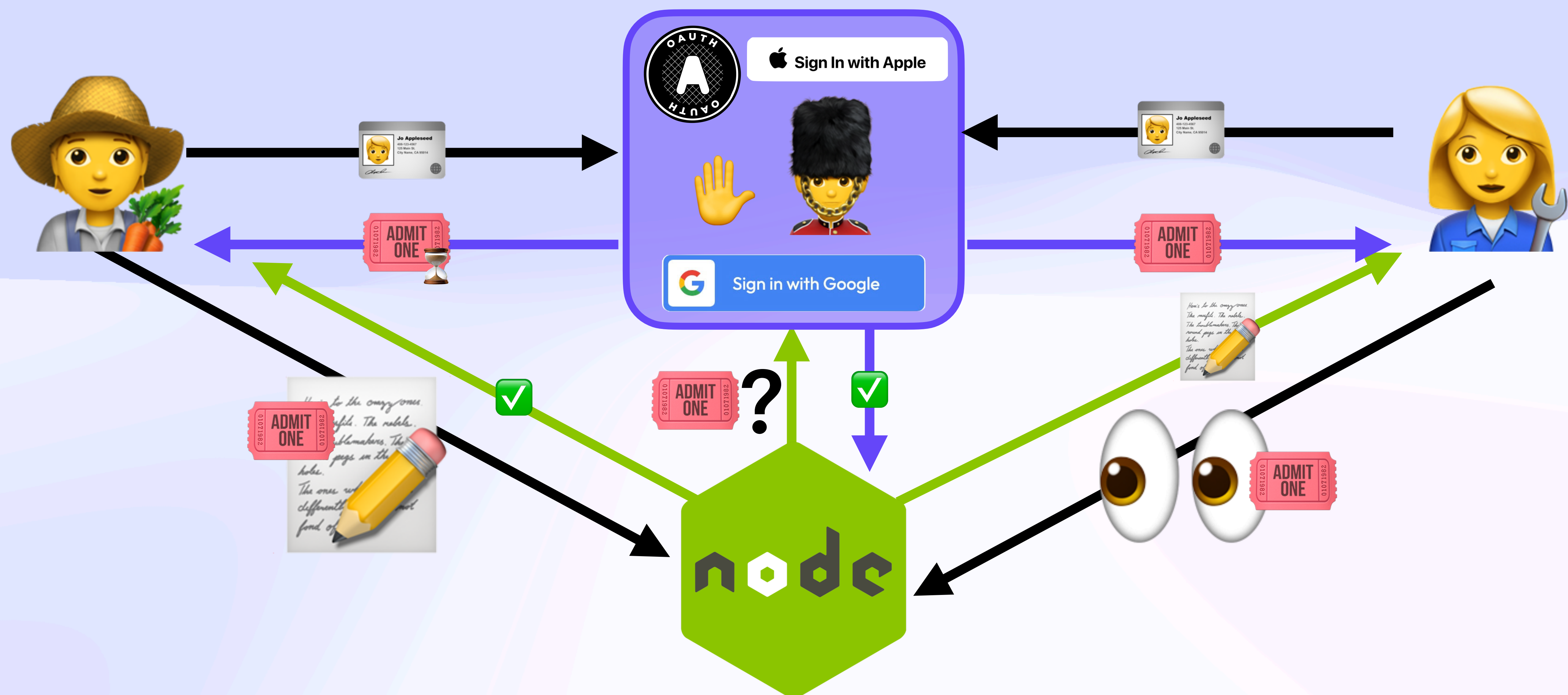
Cloud Auth

Cloud Auth Flow ☁️



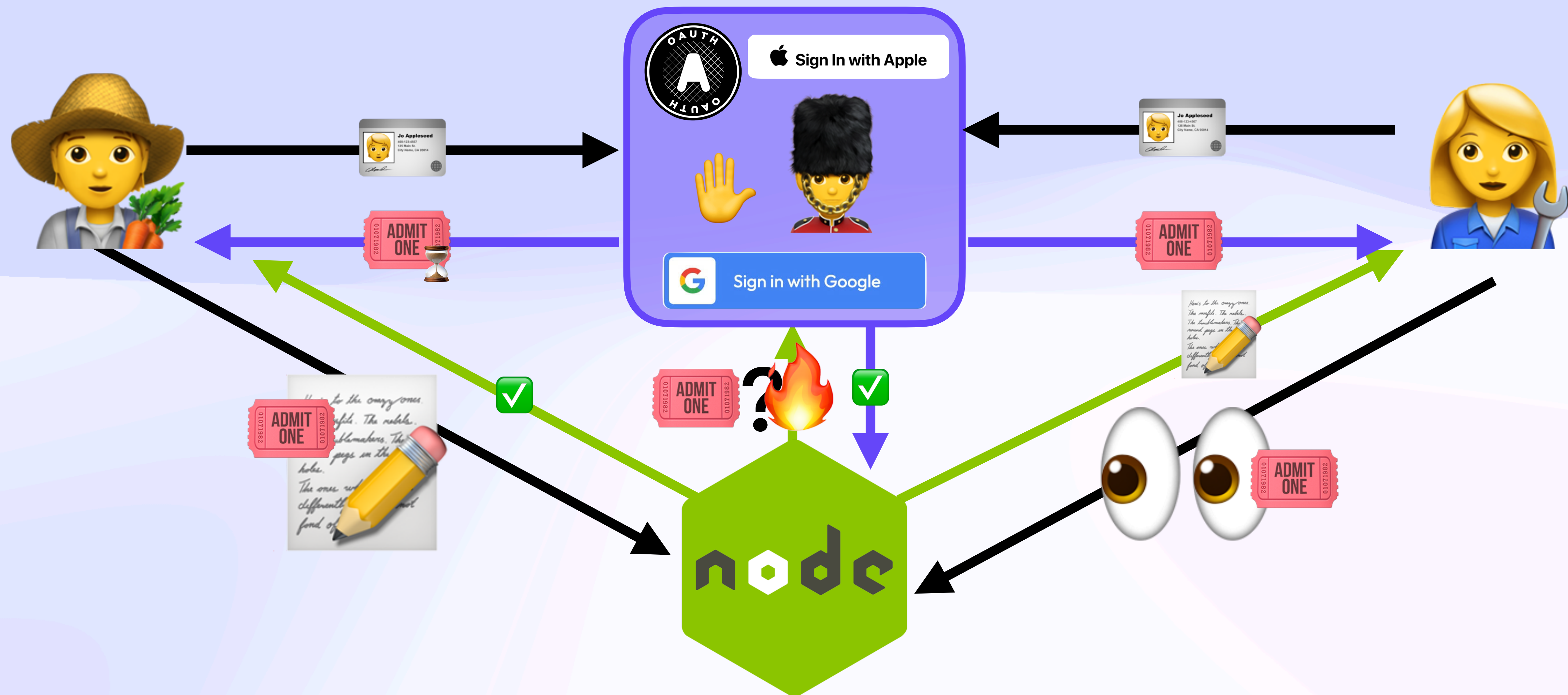
Cloud Auth

Cloud Auth Flow

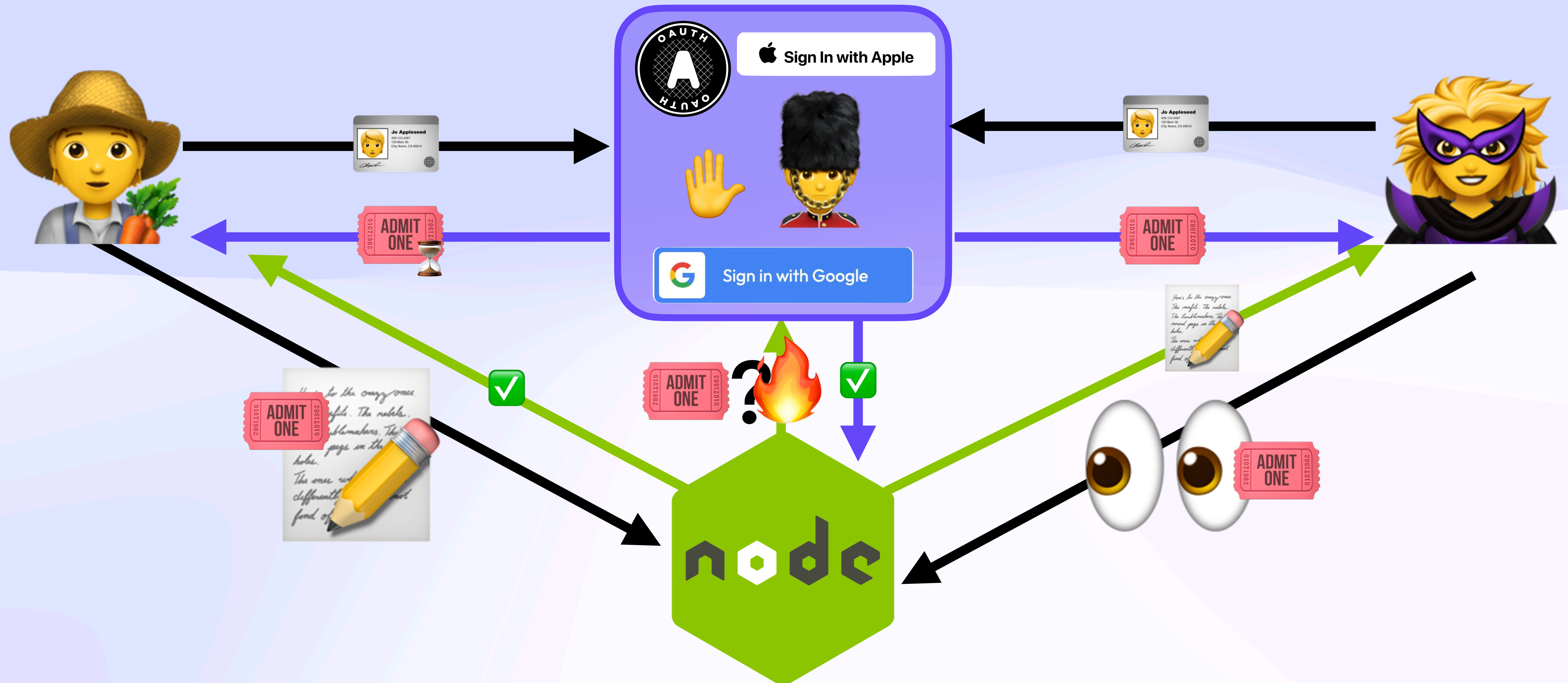


Cloud Auth

Cloud Auth Flow

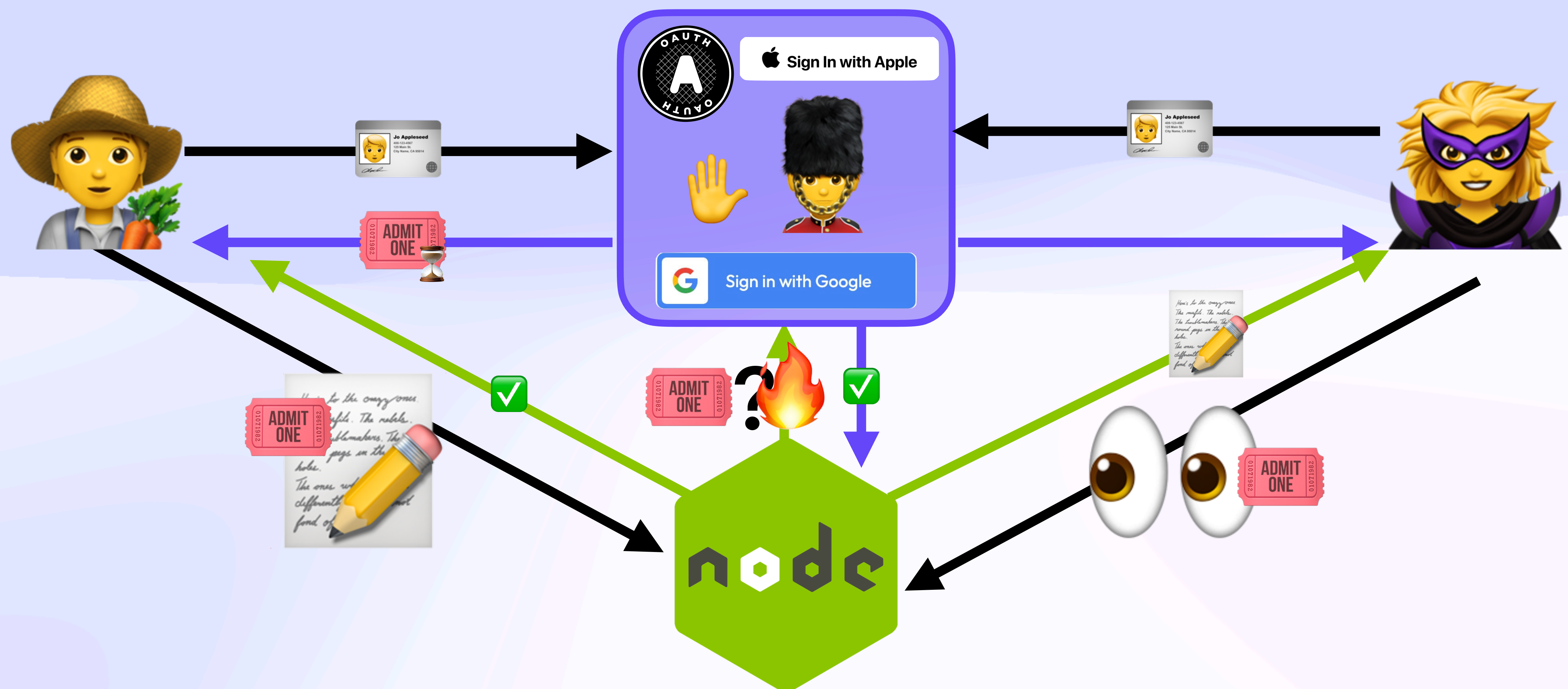


Cloud Auth Flow ☁️

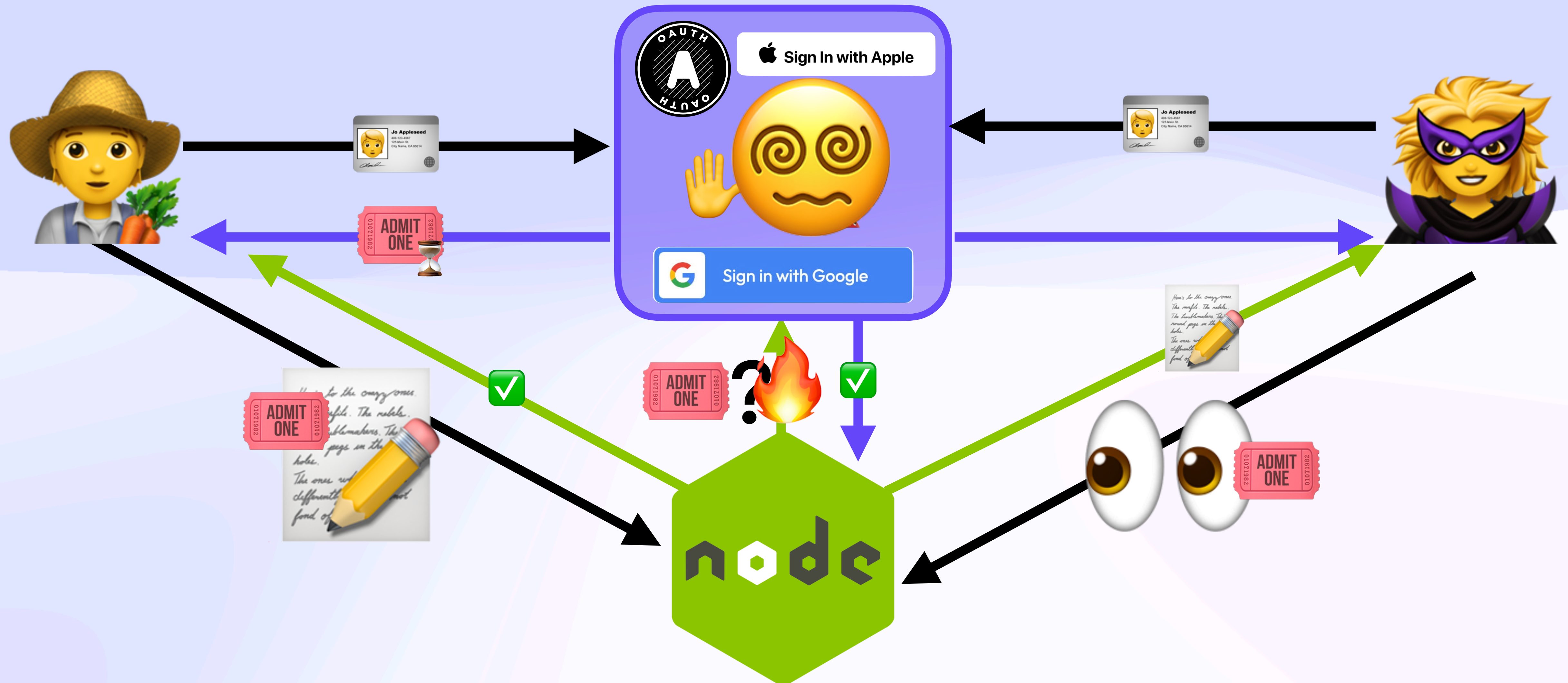


Cloud Auth

Cloud Auth Flow



Cloud Auth Flow ☁️



Cloud Auth

Cloud: Auth-as-Place ☁️

Cloud Auth

Cloud: Auth-as-Place ☁️



Cloud Auth

Cloud: Auth-as-Place ☁️

"Over Here"



"Over There"



Cloud Auth

Cloud: Auth-as-Place ☁️

"Over Here"



"Over There"





SERVERS ? WHERE WE'RE GOING, WE
DON'T NEED **SERVERS**



**SERVERS? WHERE WE'RE GOING, WE
DON'T NEED SERVERS**

Playing By New Rules

A Different Context

A Different Context

Local-First in Pictures

A Different Context

Local-First in Pictures



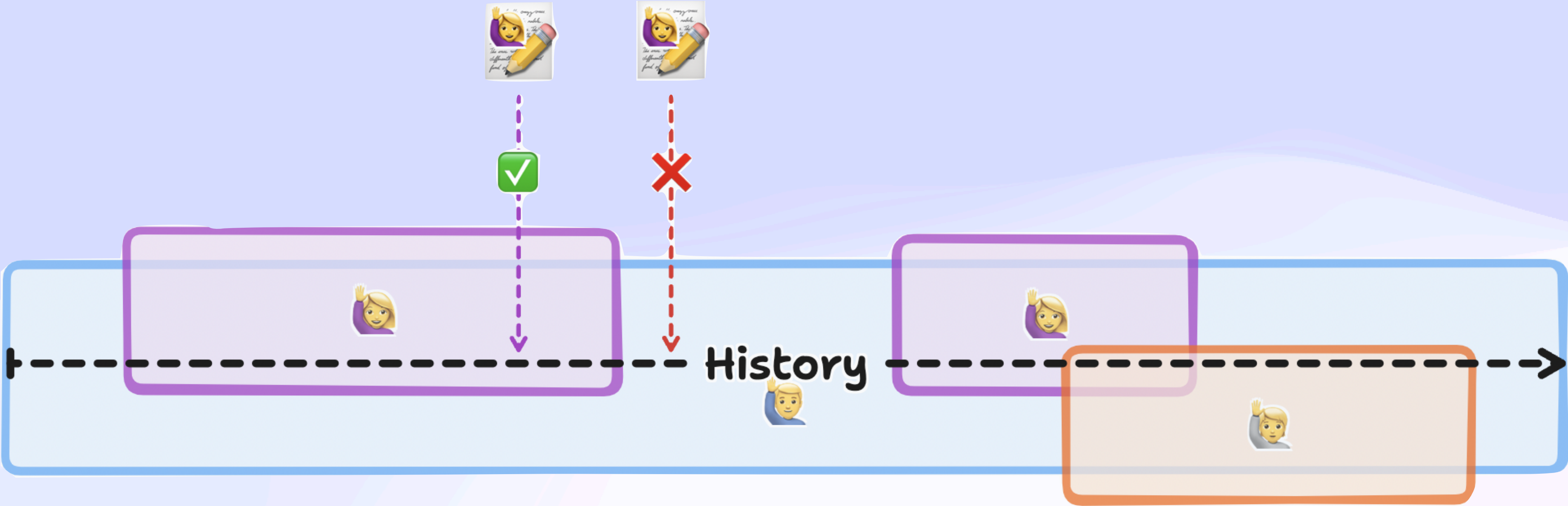
A Different Context

Local-First in Pictures



A Different Context

Adds & Removals Over Time



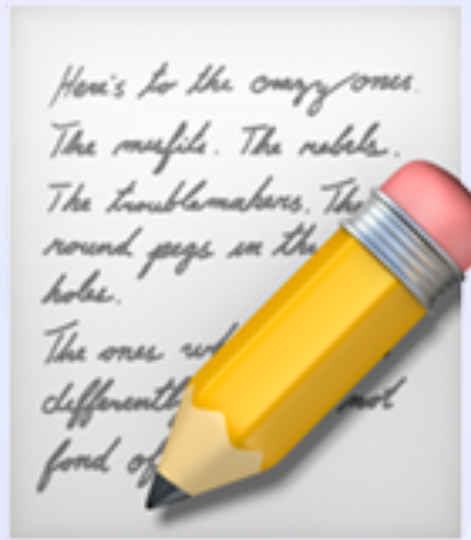
A Different Context

Auth as Data: "Auth Must Travel with Data"



A Different Context

Auth as Data: "Auth Must Travel with Data"



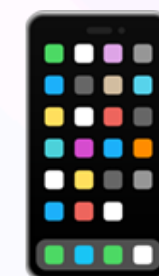
A Different Context

Auth as Data: "Auth Must Travel with Data"



A Different Context

Auth as Data: "Auth Must Travel with Data"



A Different Context

Auth as Data: "Auth Must Travel with Data"



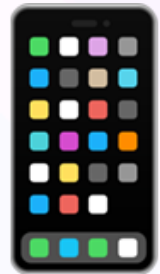
A Different Context

Auth as Data: "Auth Must Travel with Data"

"Auth Here"
"Data Here"



"Auth There"
"Data There"



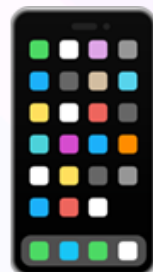
A Different Context

Auth as Data: "Auth Must Travel with Data"

"Auth Here"
"Data Here"



"Auth There"
"Data There"



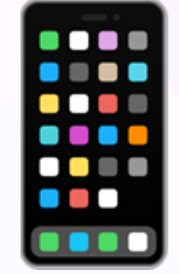
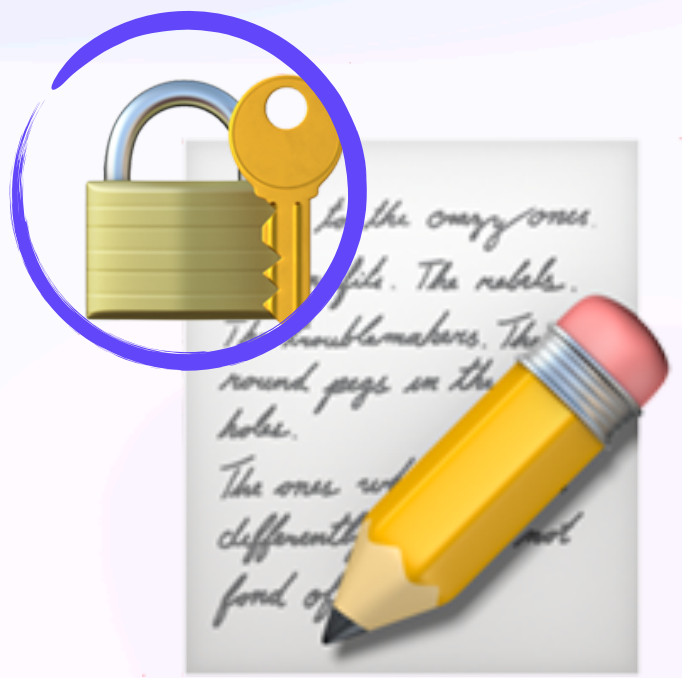
A Different Context

Auth as Data: "Auth Must Travel with Data"

"Auth Here"
"Data Here"



"Auth There"
"Data There"



A Different Context

In Sync Servers We Trust (as little as possible)

A Different Context

In Sync Servers We Trust (as little as possible)

- What do we trust servers to do?
 - Hold our bytes and not delete them
 - Only send those bytes to someone with the right permissions
 - Trust them with the knowledge of our auth graph
 - Server knows who (IP address & public key) requests which doc IDs
- Defence-in-depth strategy against buggy or poorly run sync servers (or break-ins)

A Different Context



A Different Context

Cloud

Local-First

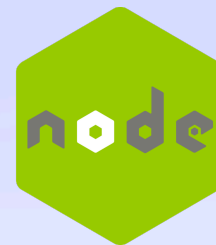
A Different Context

Cloud

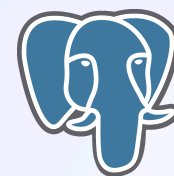
Auth 



Compute 



Data 




Local-First

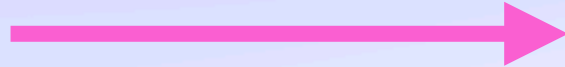
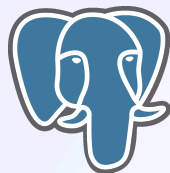
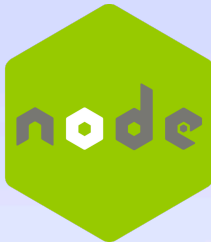
A Different Context

Cloud

Auth 


Compute 

Data 



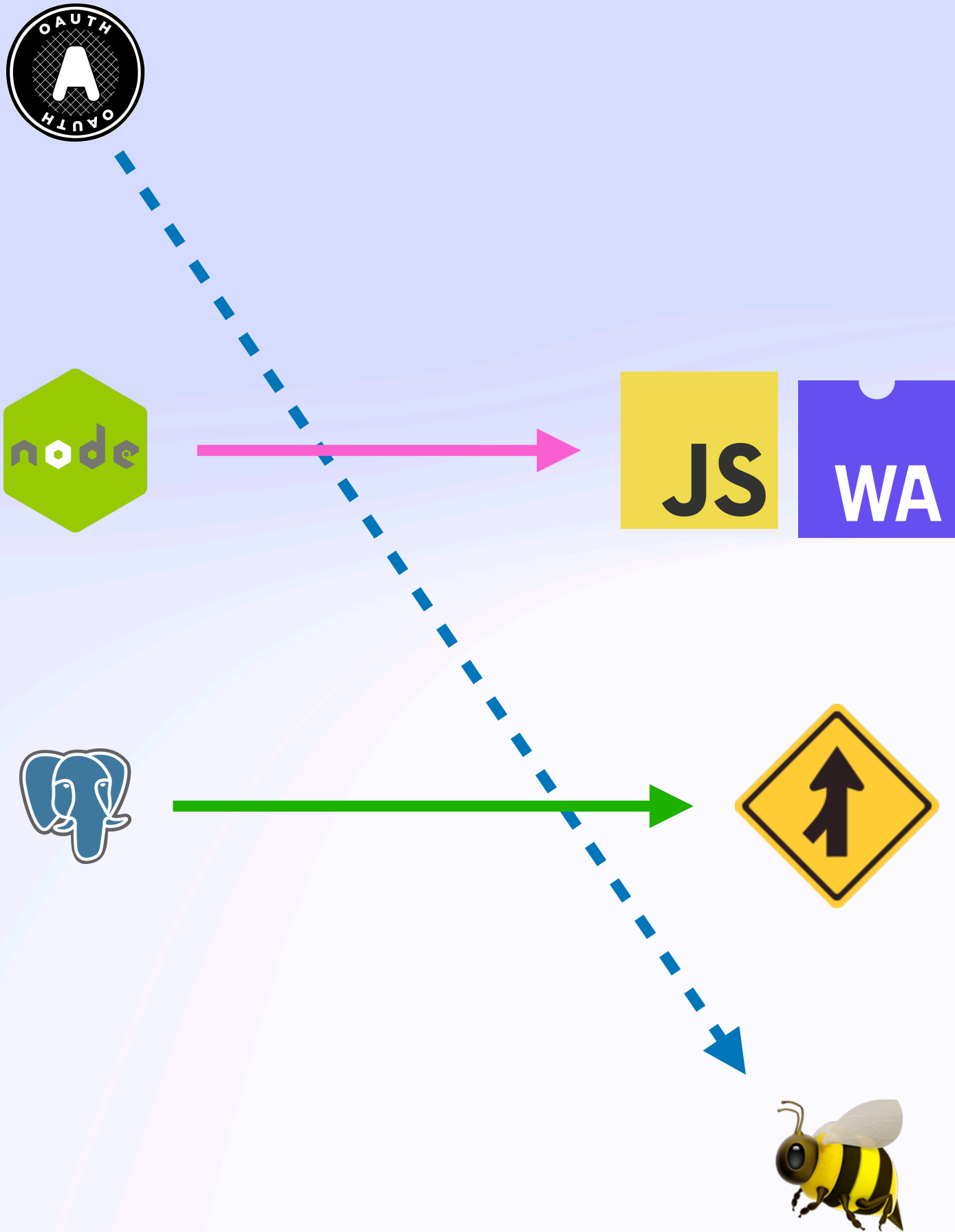
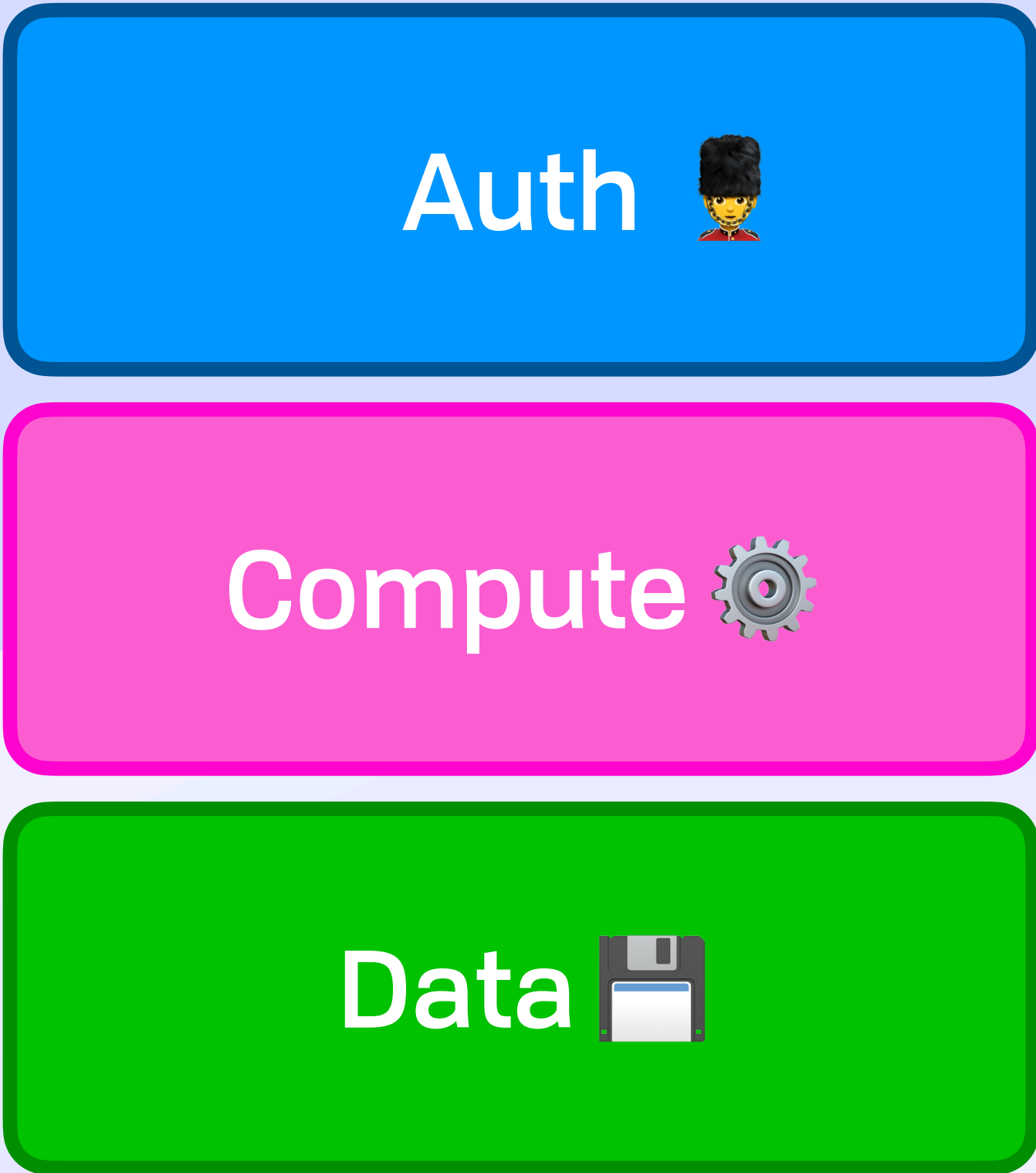
Local-First

Compute 

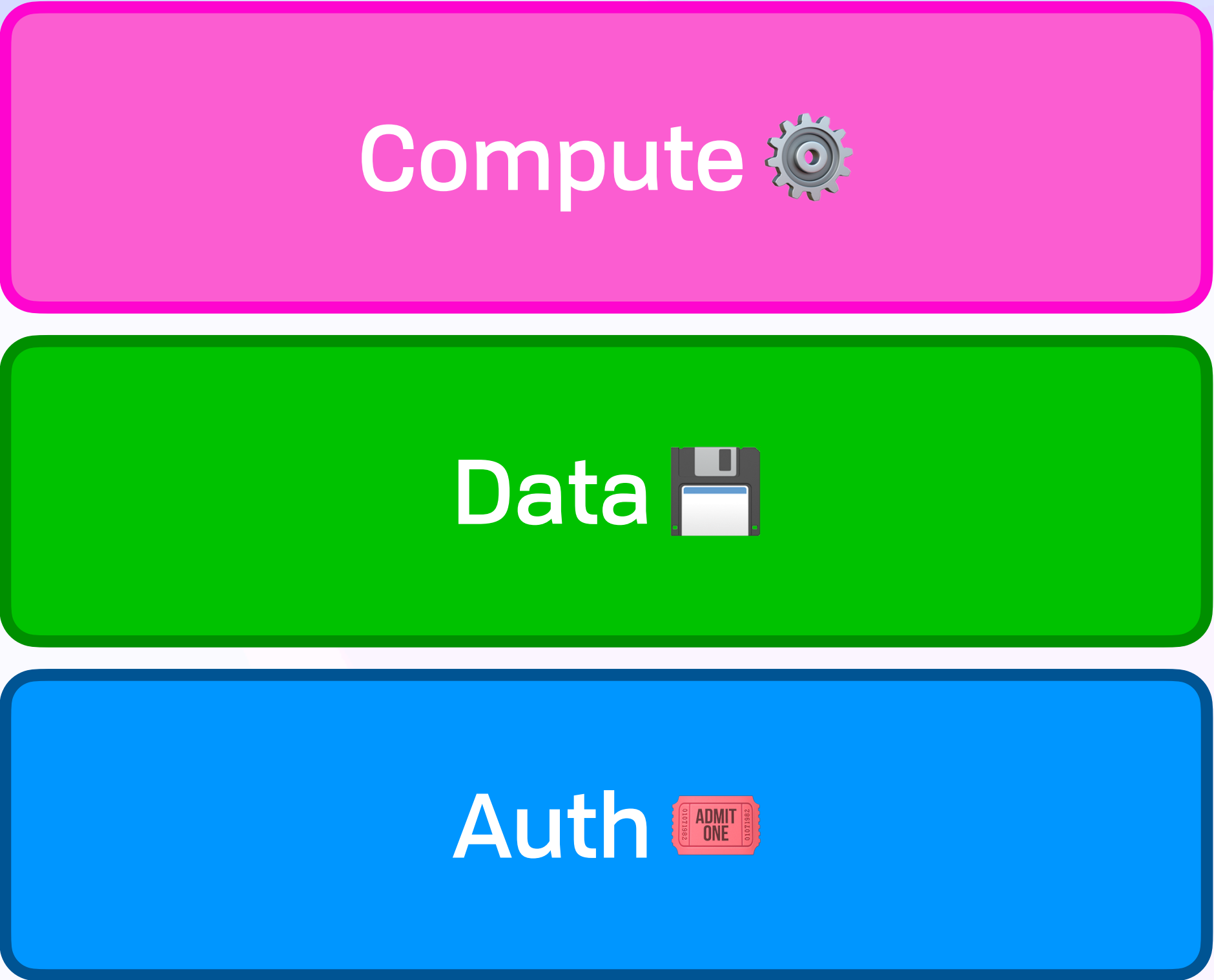
Data 

A Different Context

Cloud



Local-First



Expanding the Beehive, and UCAN Too

Convergent Capabilities



Convergent Capabilities



Convergent Capabilities

Copy
(Request)



Convergent Capabilities

Copy
(Request)

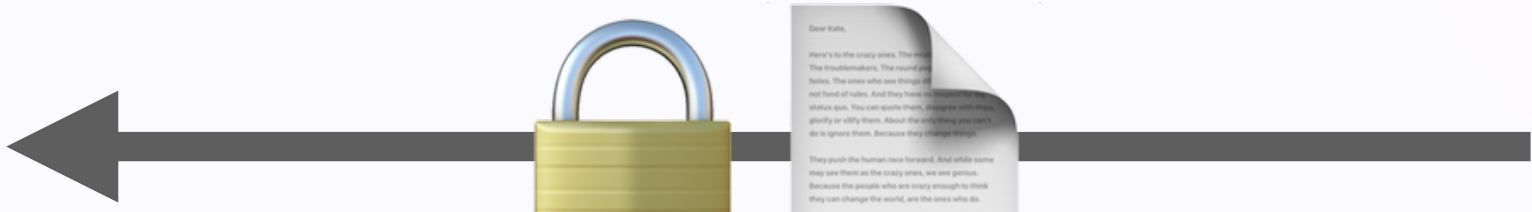


Convergent Capabilities

Read
(Decrypt)

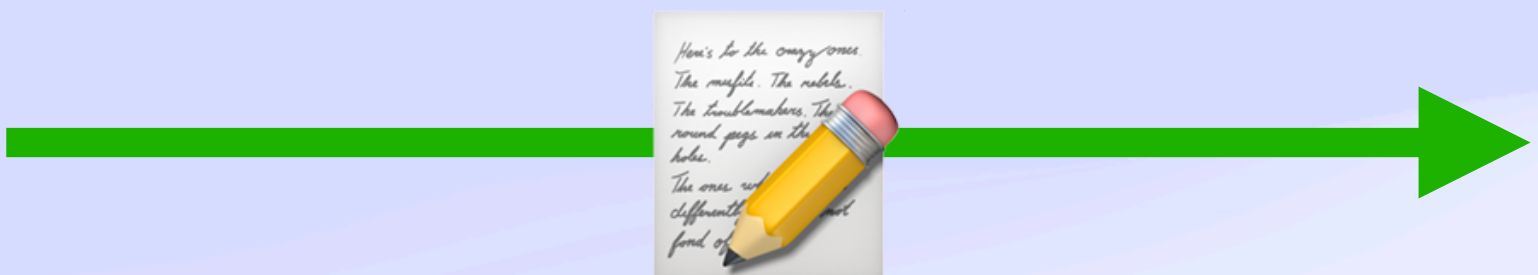


Copy
(Request)

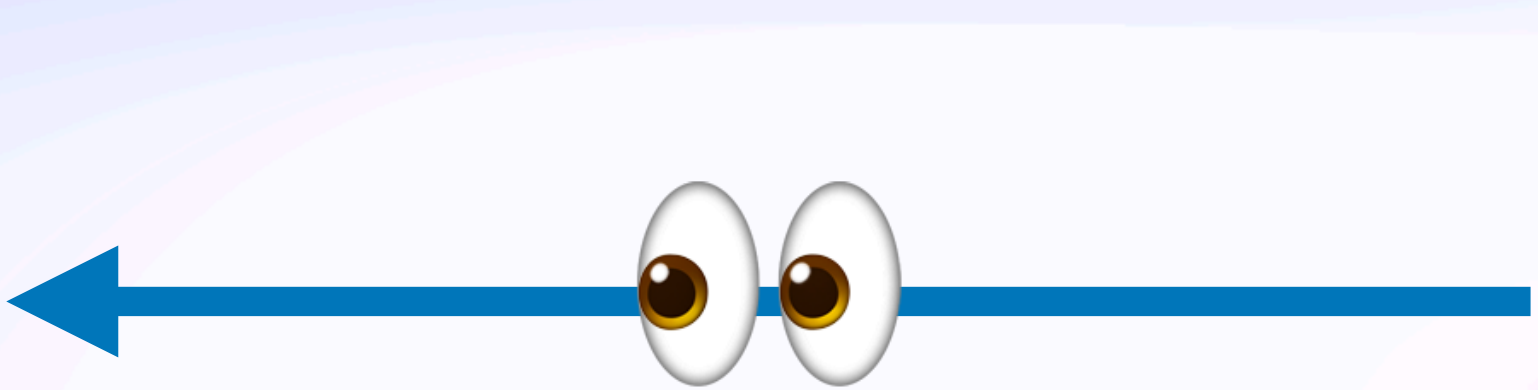


Convergent Capabilities

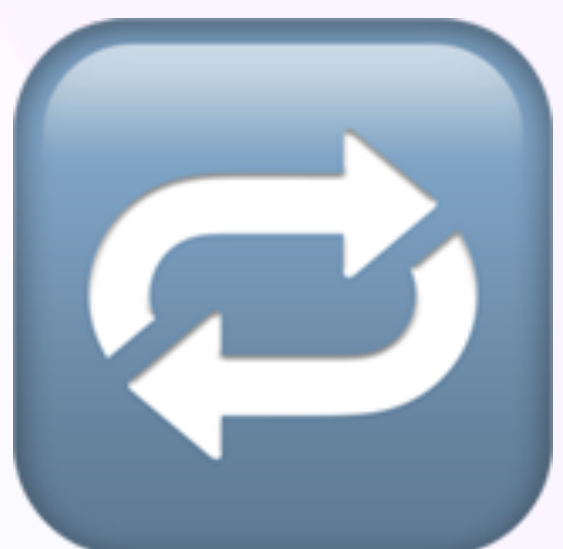
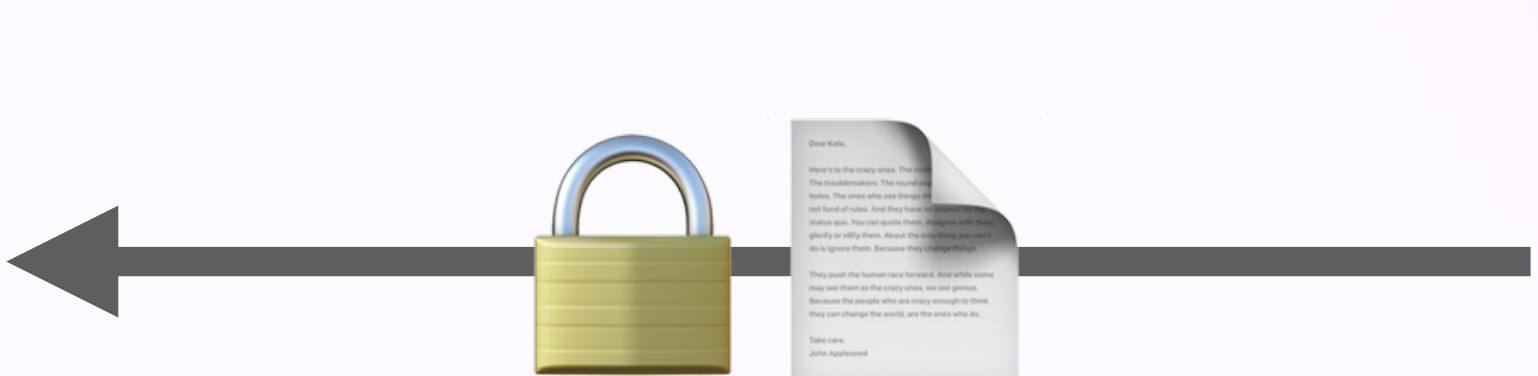
Write
(Update)



Read
(Decrypt)



Copy
(Request)

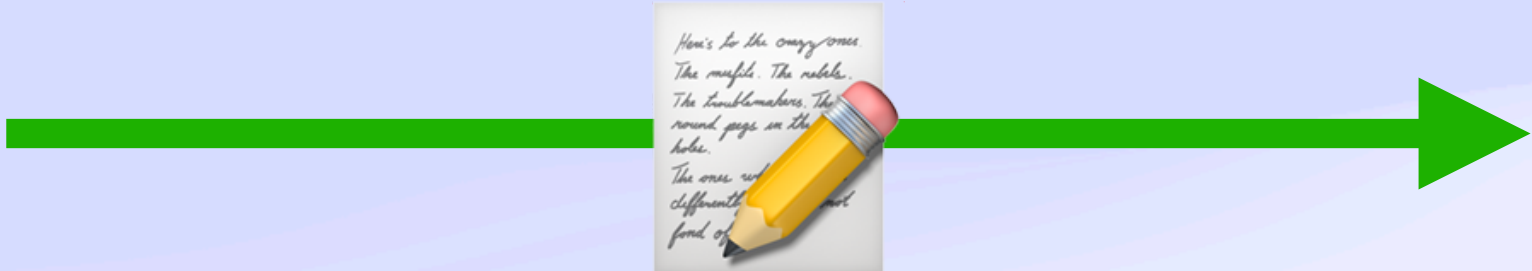


Convergent Capabilities

Admin
(Revoke)



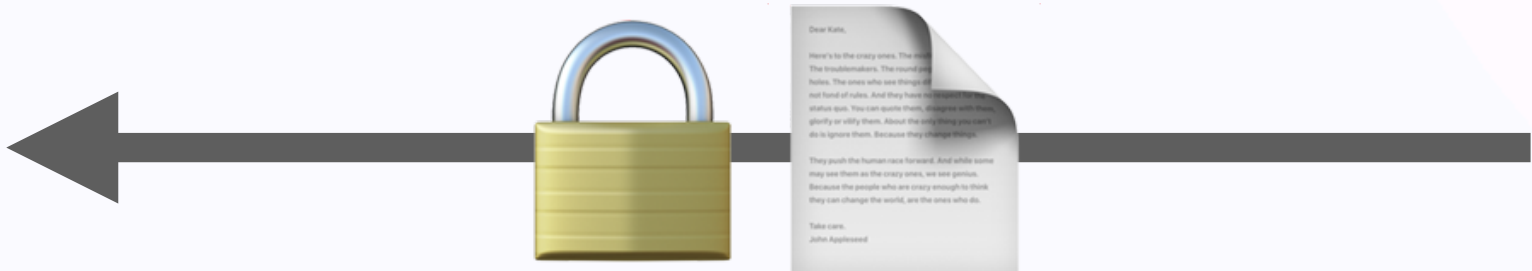
Write
(Update)



Read
(Decrypt)



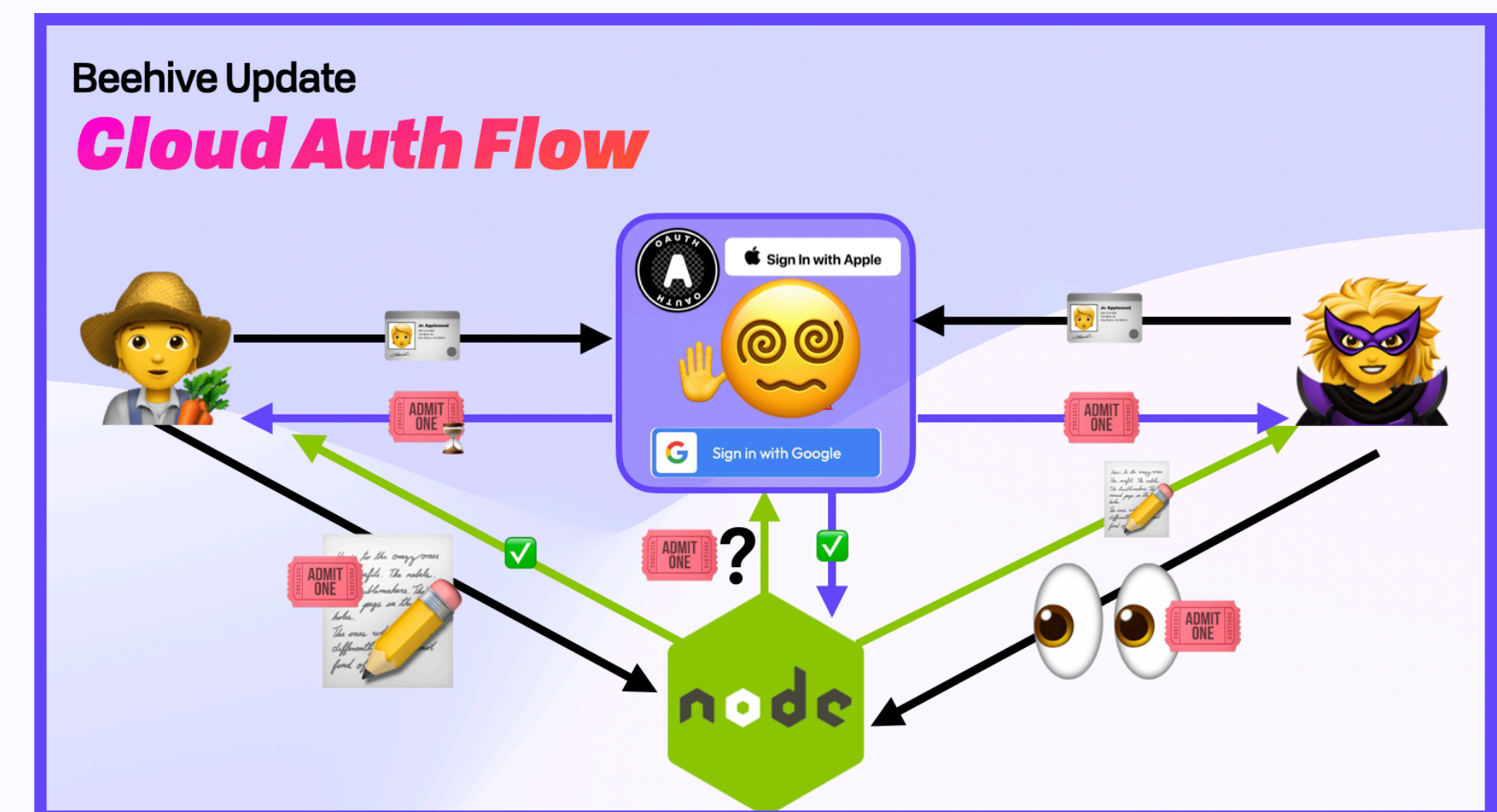
Copy
(Request)



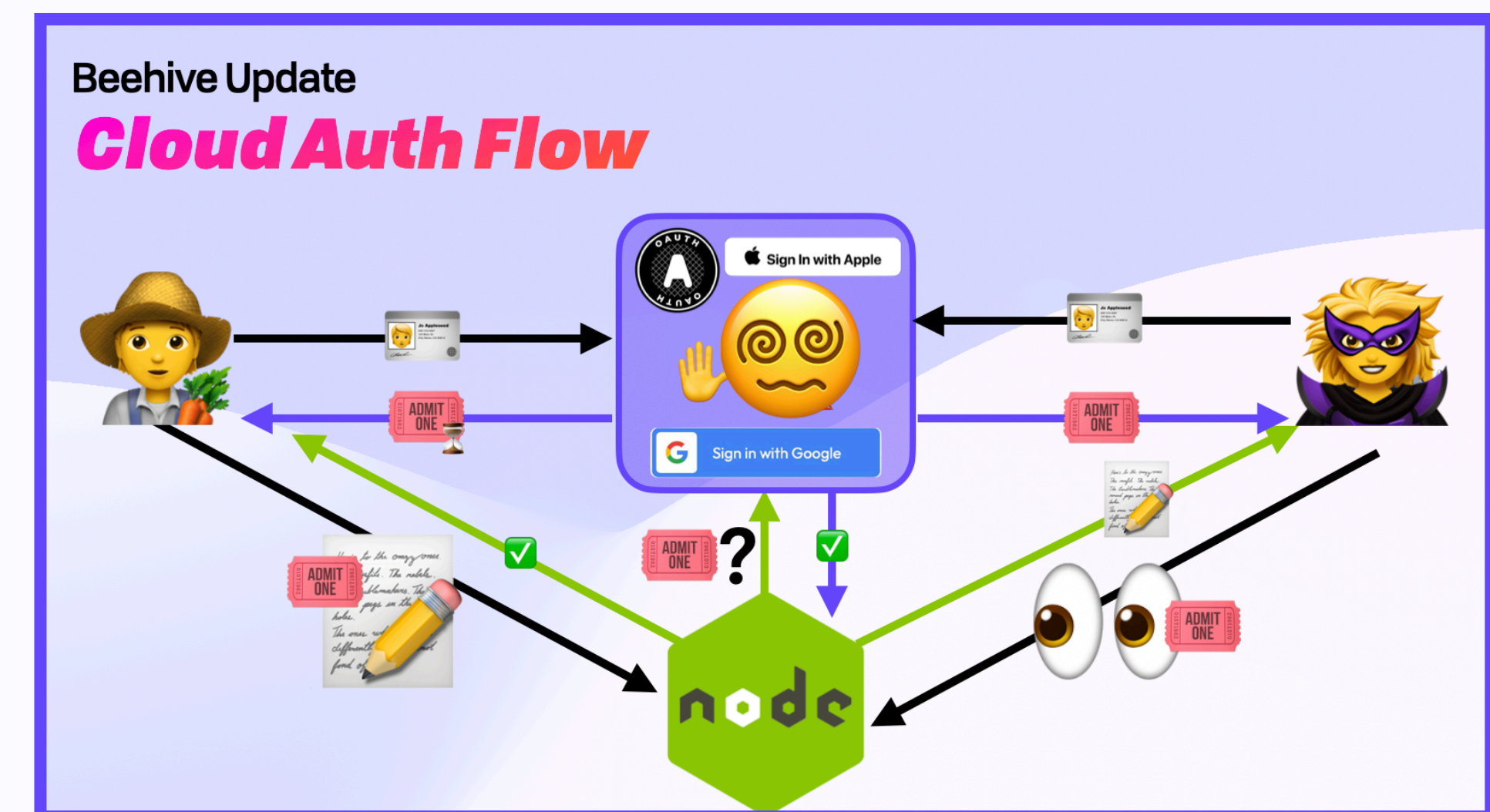
Convergent Capabilities

Self-Authenticating Changes

Self-Authenticating Changes

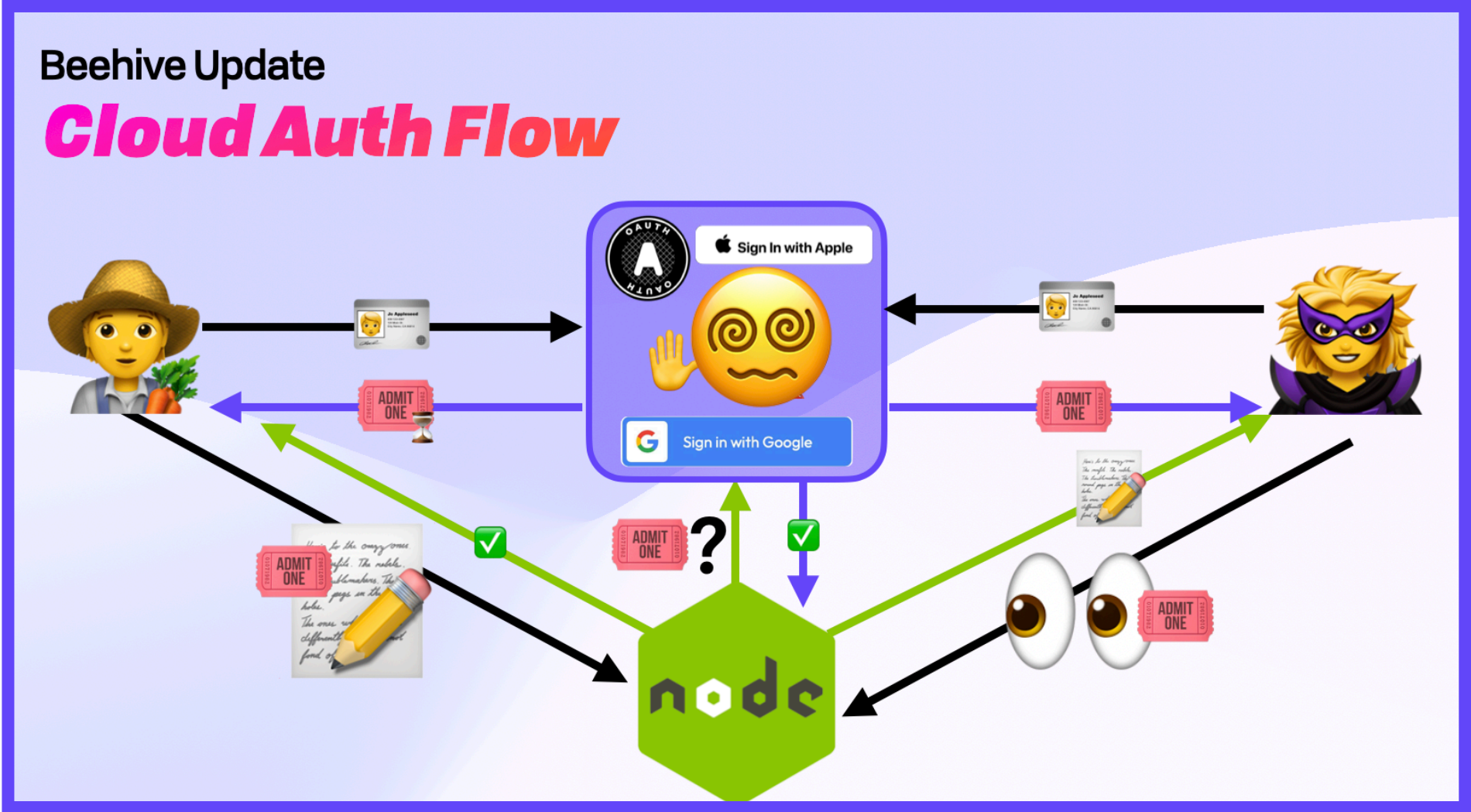


Self-Authenticating Changes



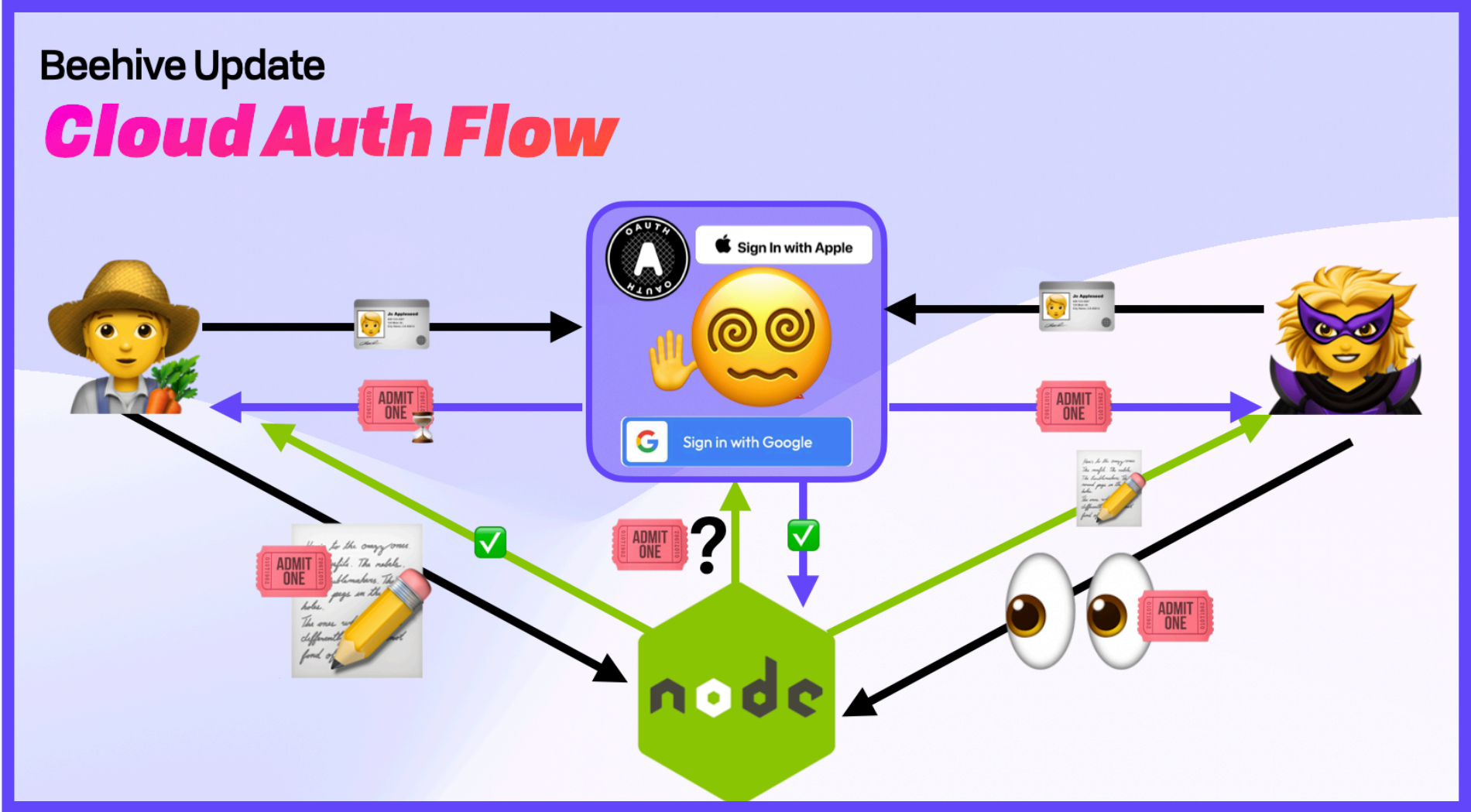
Convergent Capabilities

Self-Authenticating Changes



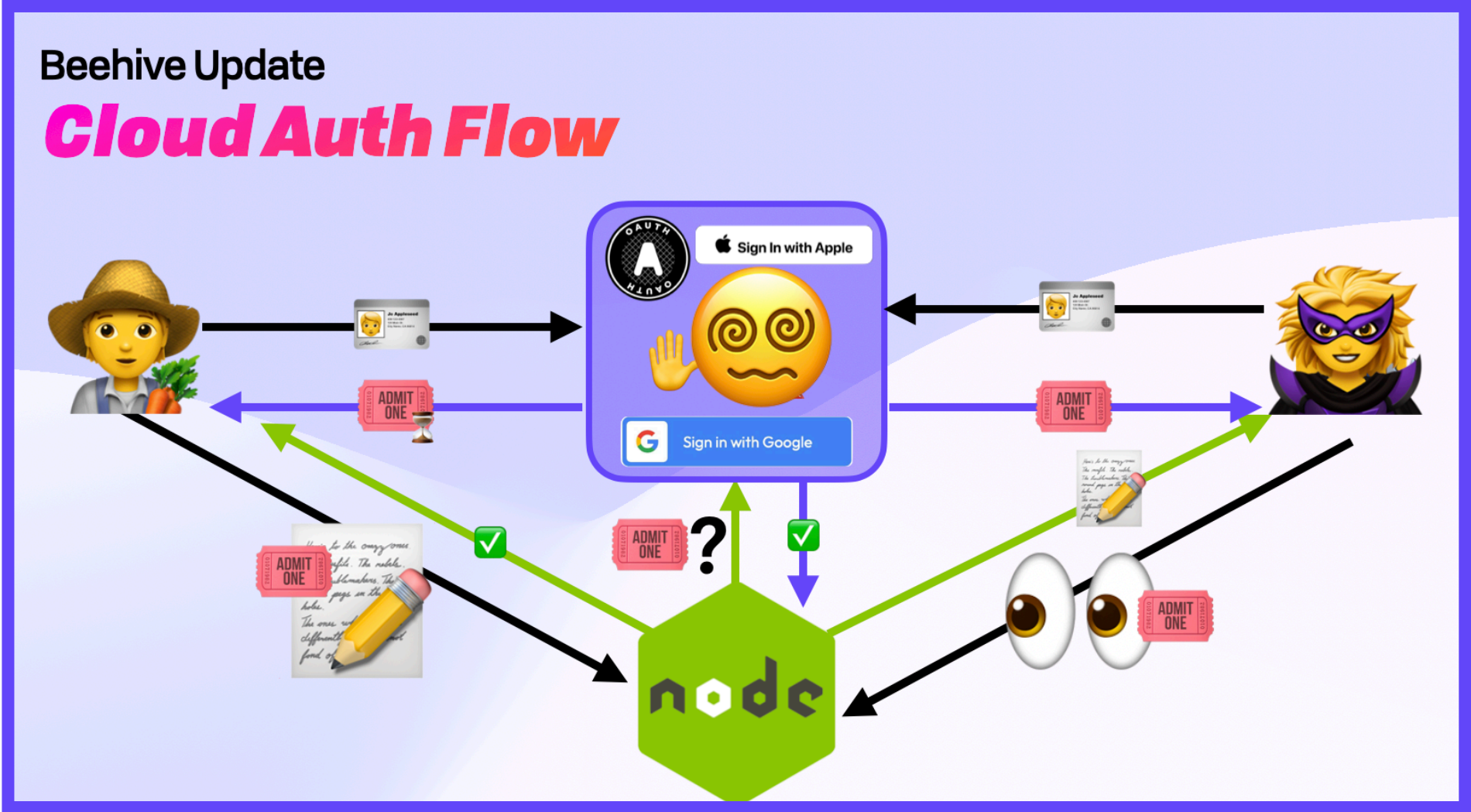
Convergent Capabilities

Self-Authenticating Changes



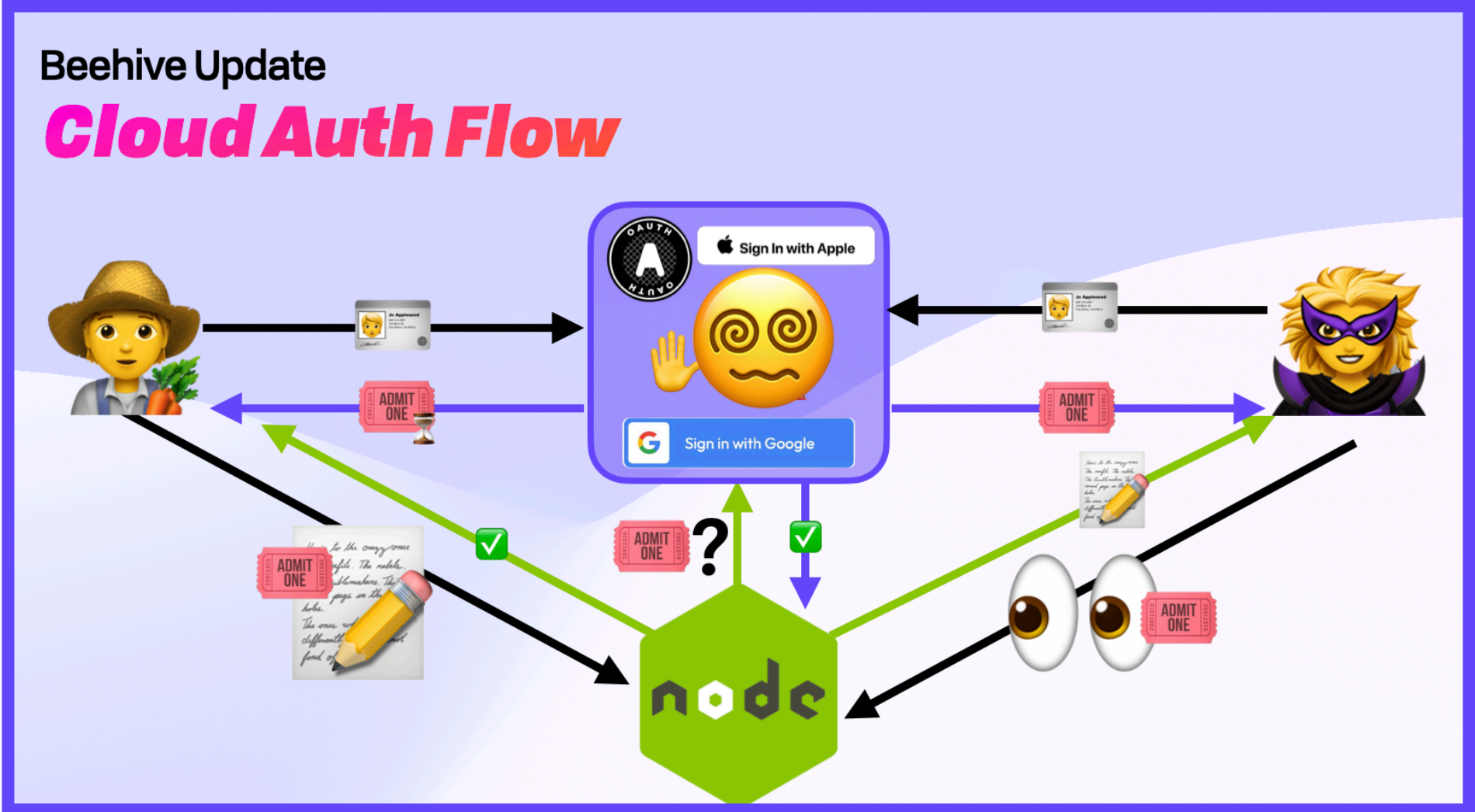
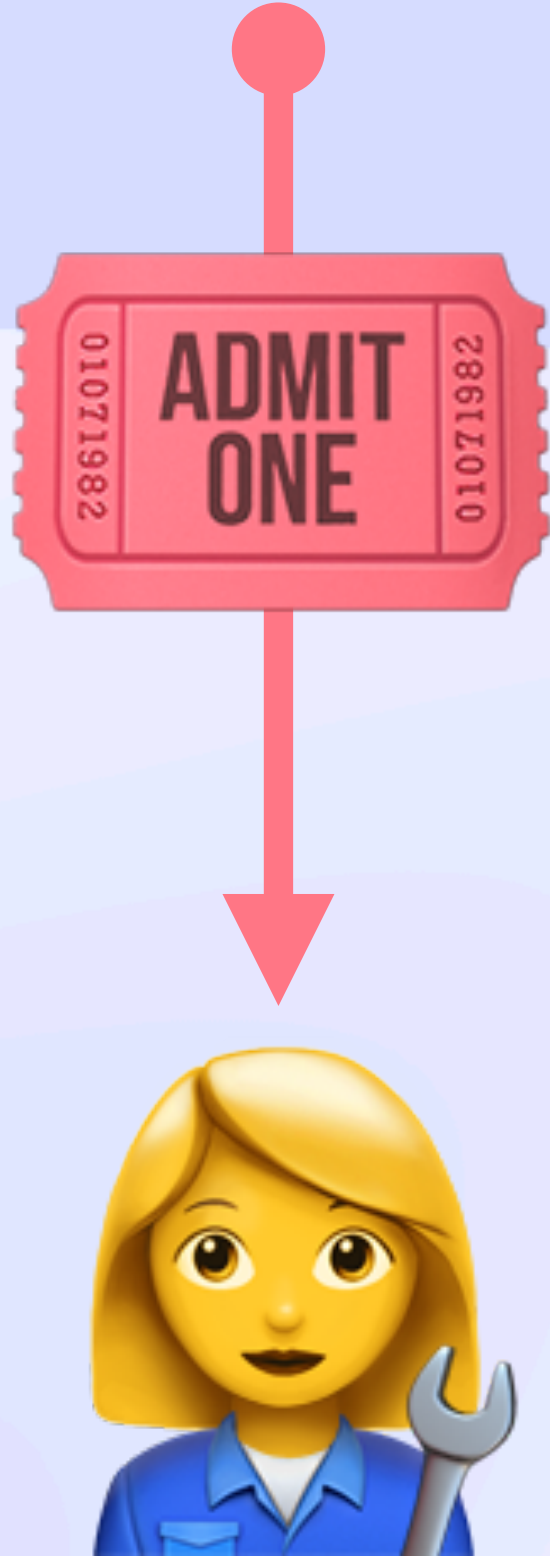
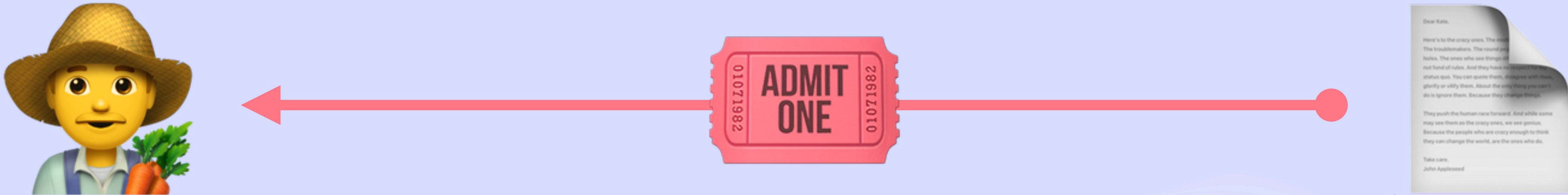
Convergent Capabilities

Self-Authenticating Changes



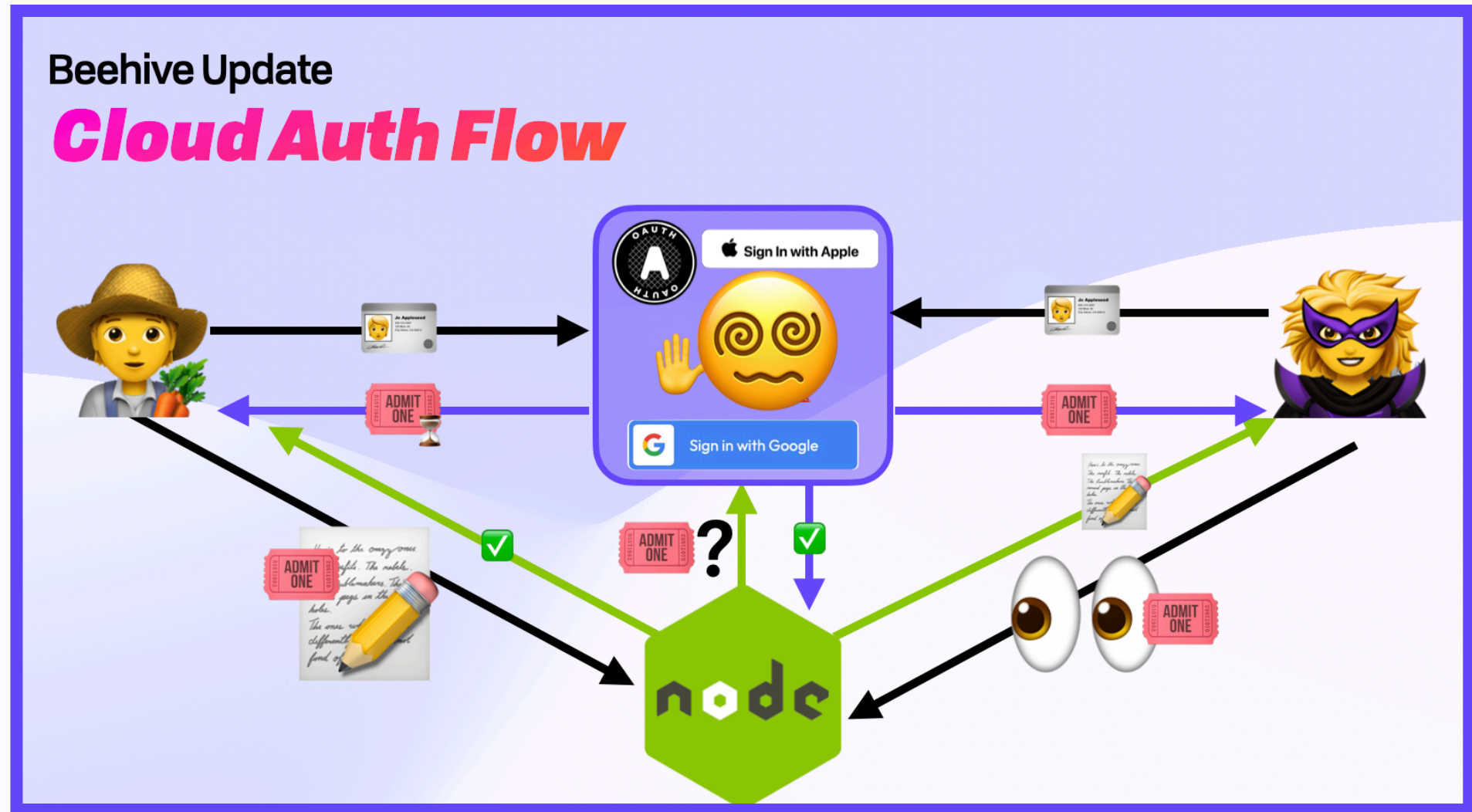
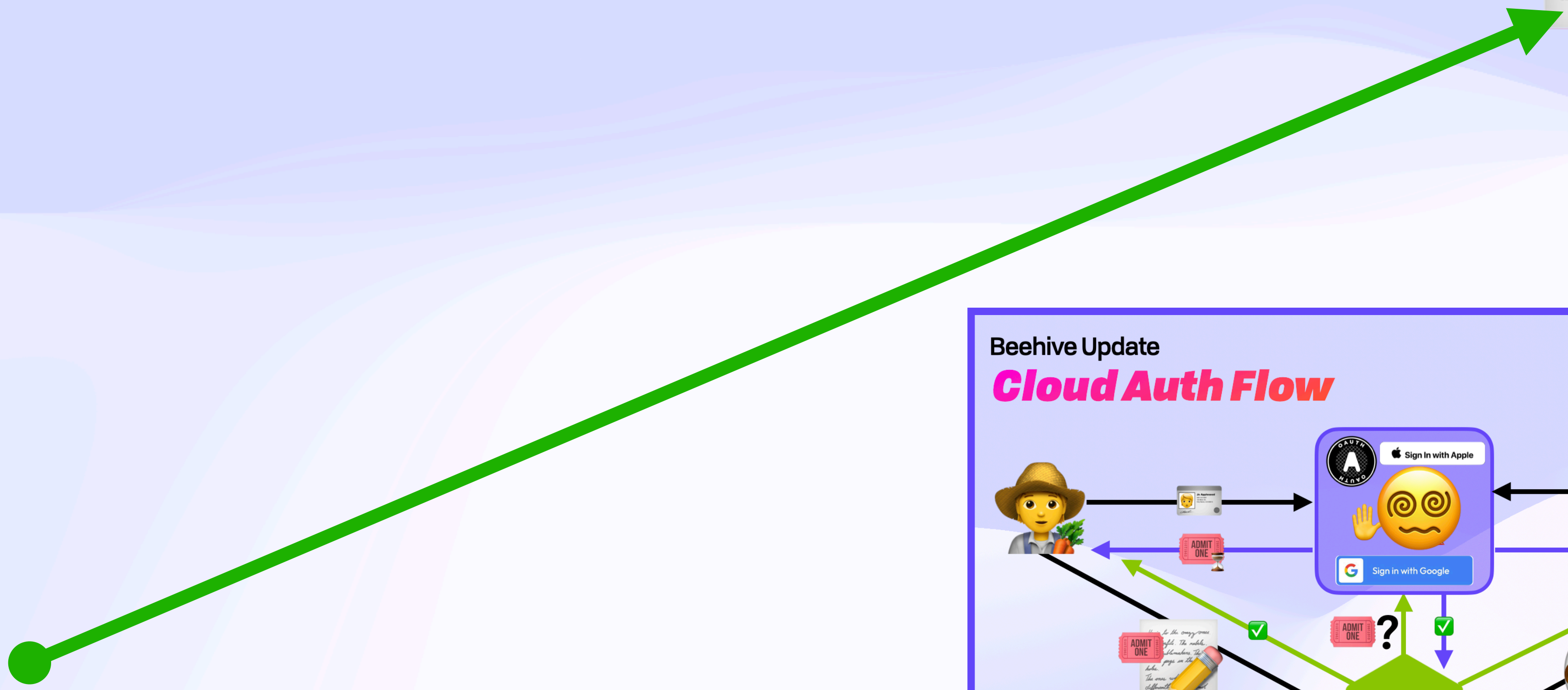
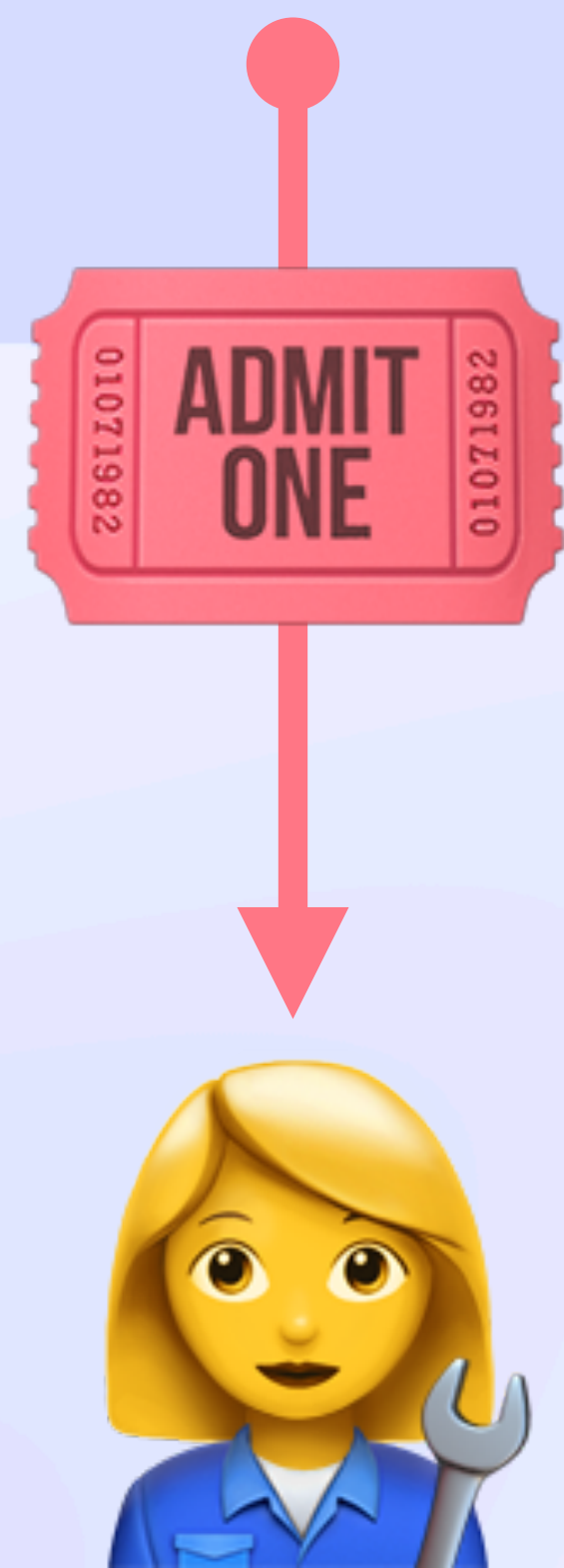
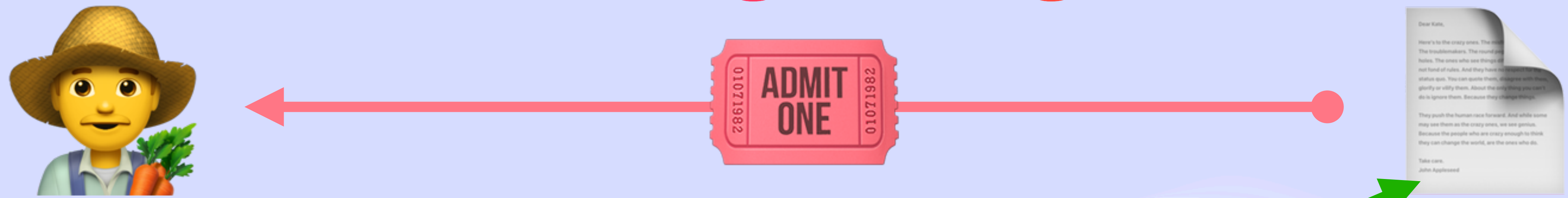
Convergent Capabilities

Self-Authenticating Changes



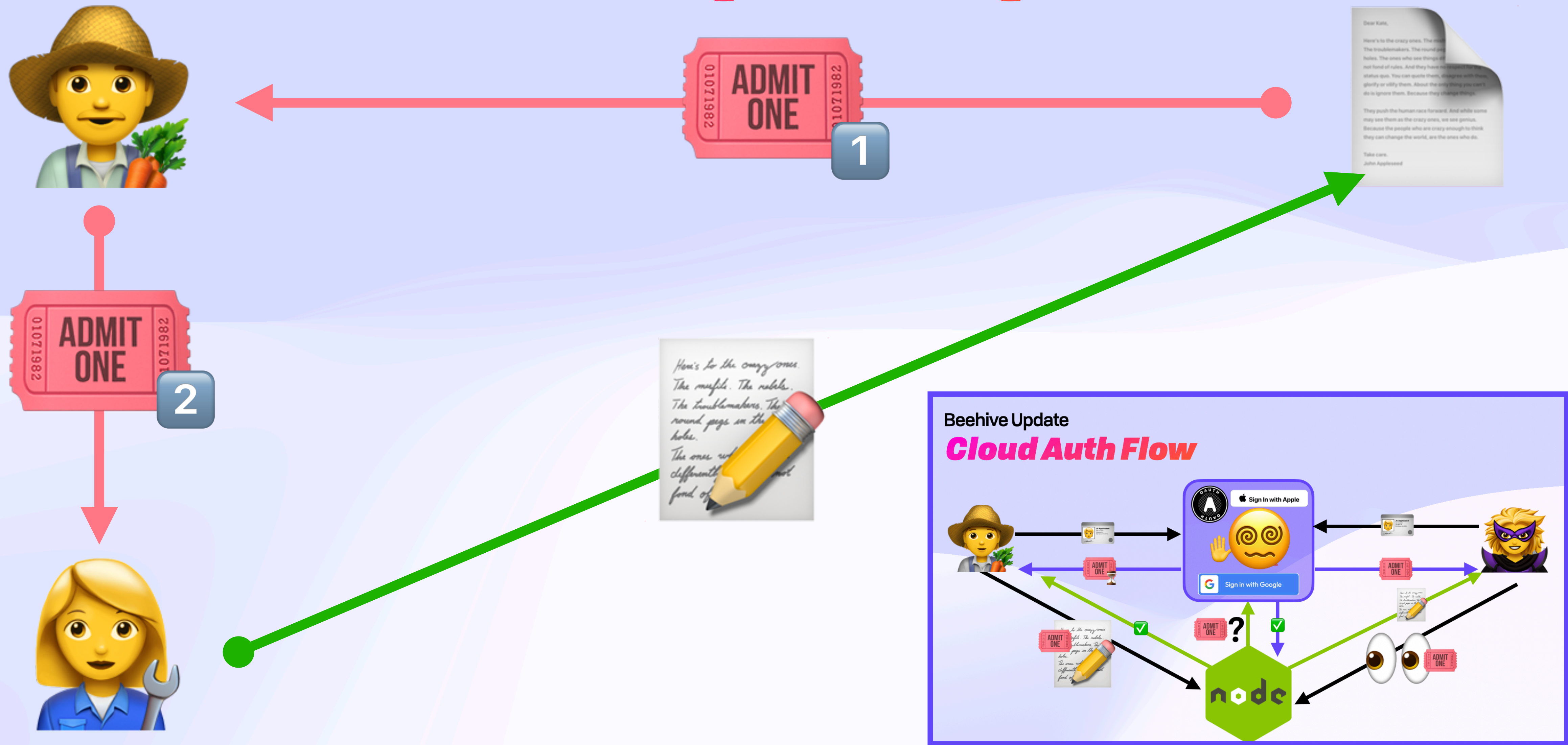
Convergent Capabilities

Self-Authenticating Changes

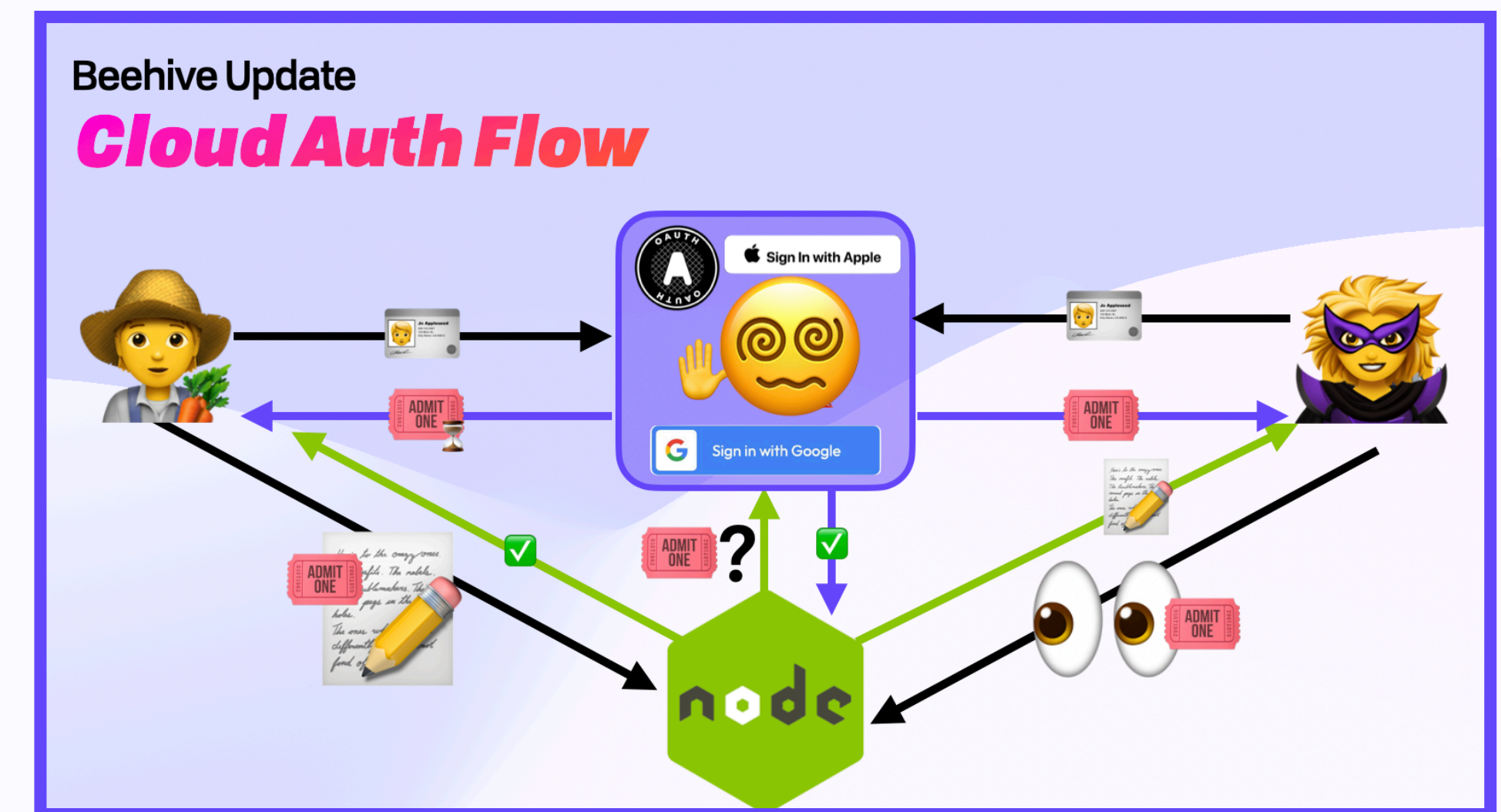


Convergent Capabilities

Self-Authenticating Changes

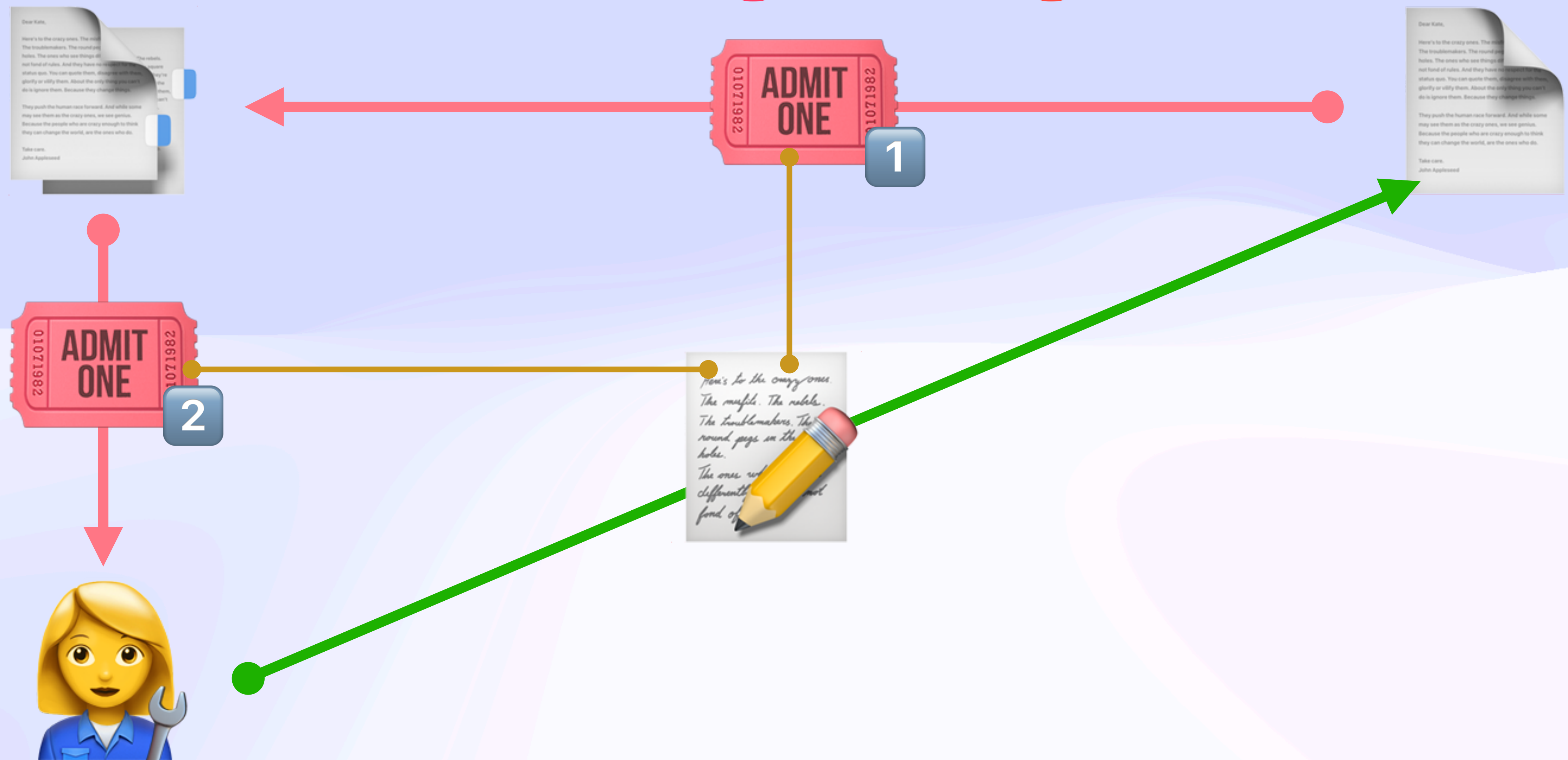


Self-Authenticating Changes



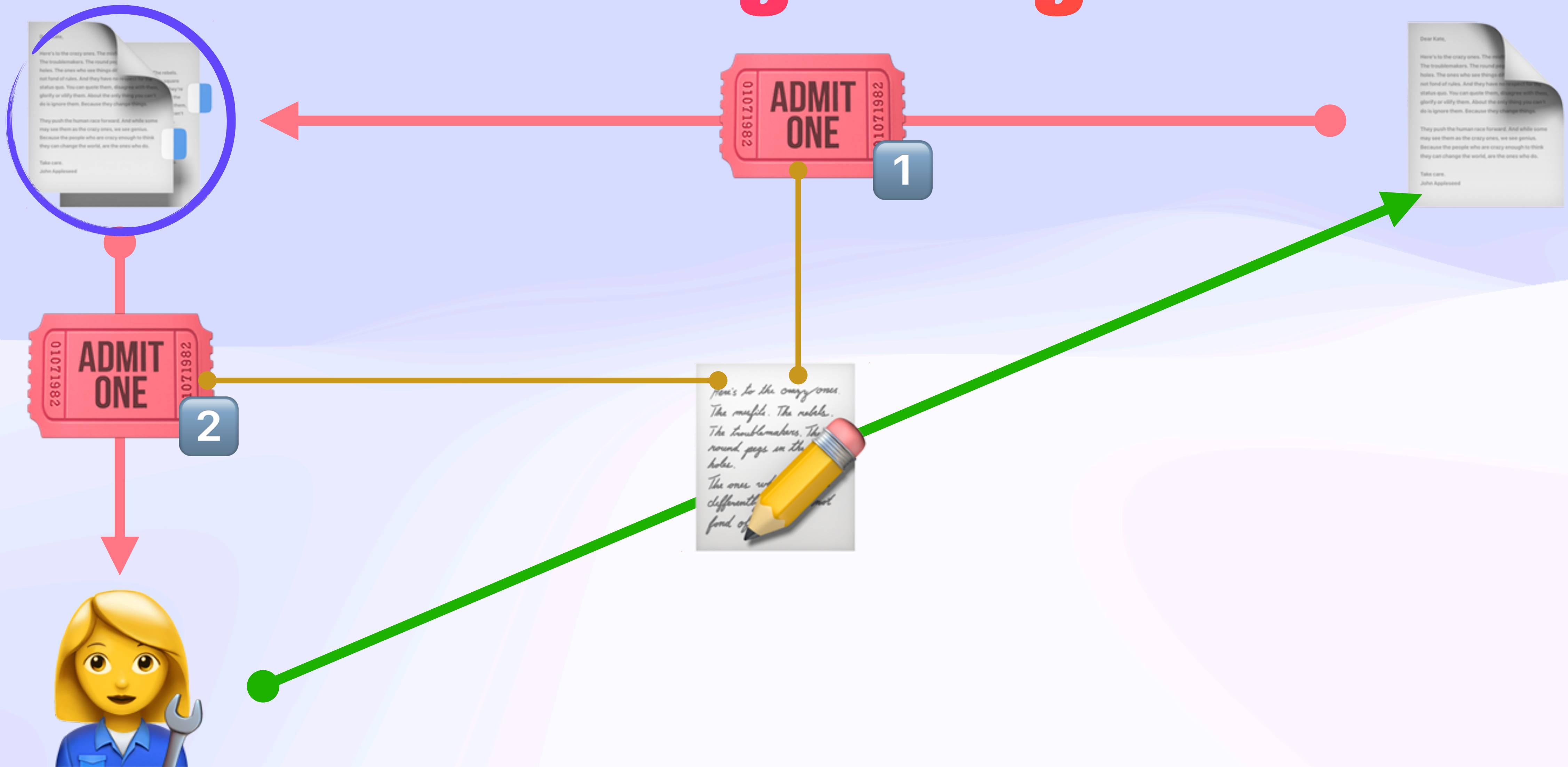
Convergent Capabilities

Self-Authenticating Changes



Convergent Capabilities

Self-Authenticating Changes



Convergent Capabilities

Honest Model, Simple to Use

- "If you have it, you can use it & share it"
- Closely models auth **ground truth**
- Natural **programming** model "like passing handles"

Convergent Capabilities

Example API

Convergent Capabilities

Example API

- **High level**

- `beehiveEssay.addMember(alice, role)?;`
- `beehiveEssay.addText("hello world")?;`

Convergent Capabilities

Example API

- **High level**

- `beehiveEssay.addMember(alice, role)?;`
- `beehiveEssay.addText("hello world")?;`

- **Low level / under the hood**

- `essayCapability.delegateTo(alice);`
- `essayCapability.do(doc ⇒ doc.addText("hello world"));`


Convergent Capabilities

Example API

- **High level**

- `beehiveEssay.addMember(alice, role)?;`
- `beehiveEssay.addText("hello world")?;`

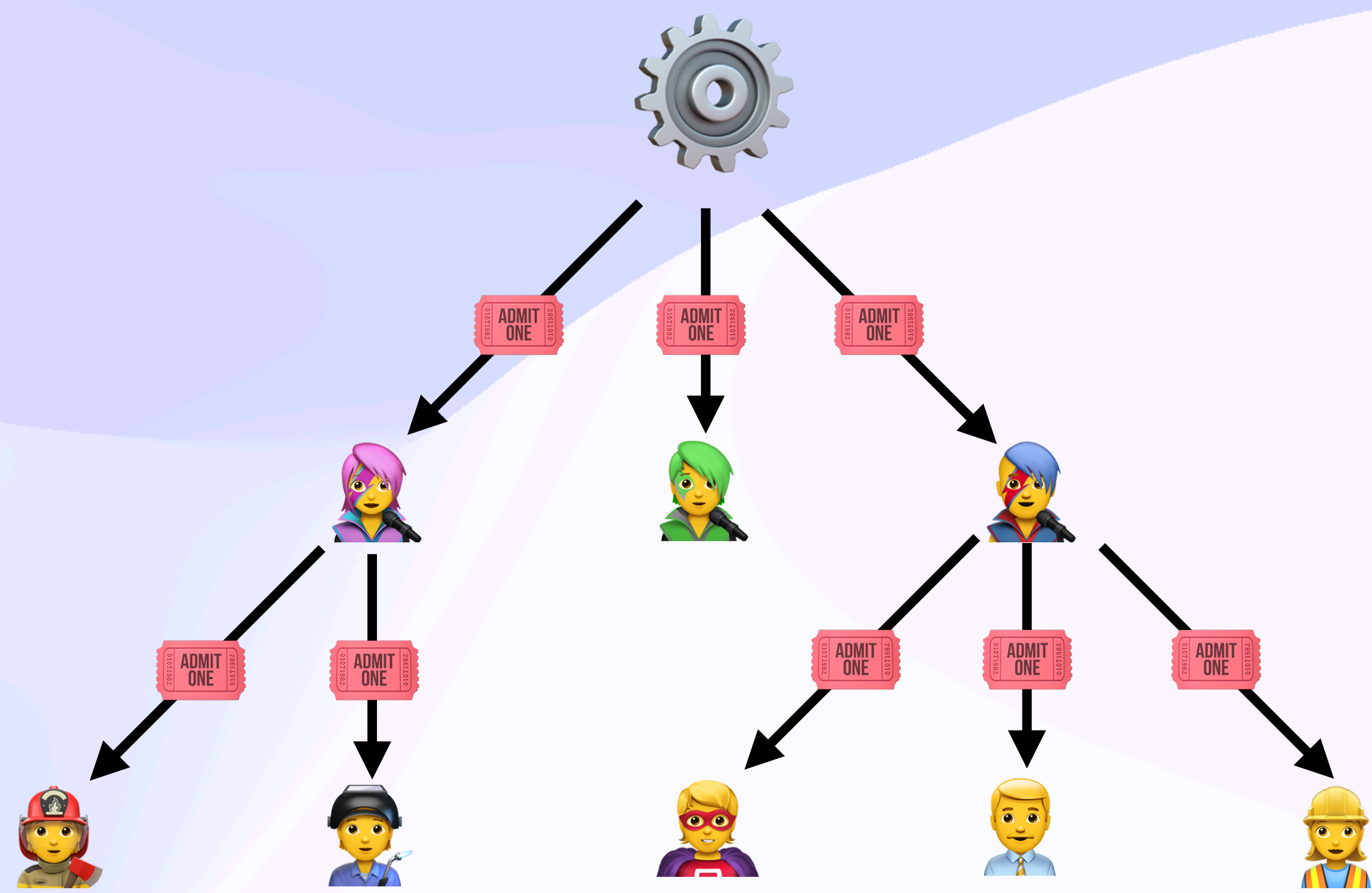
- **Low level / under the hood**

- `essayCapability.delegateTo(alice);` 
- `essayCapability.do(doc ⇒ doc.addText("hello world"));`



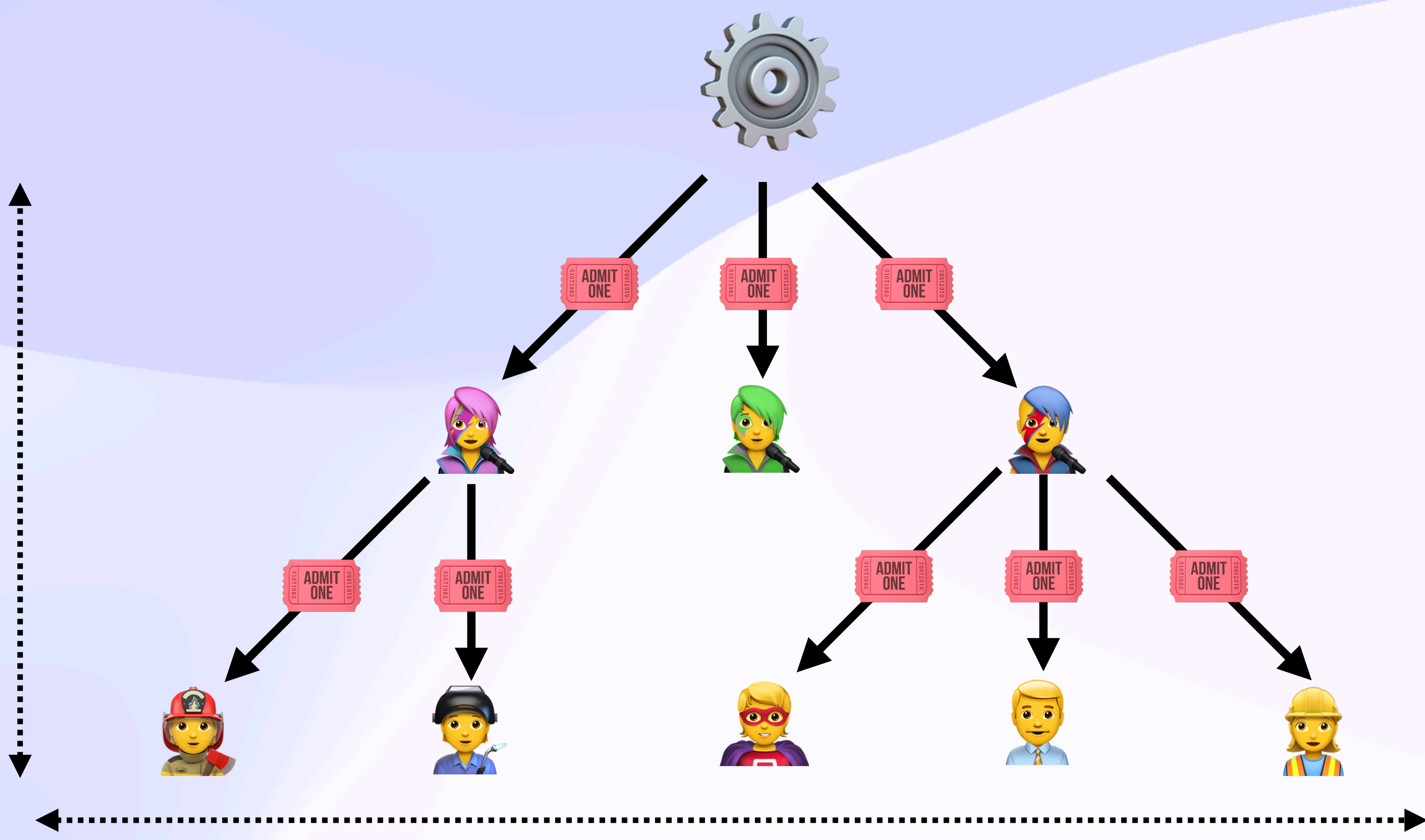
Convergent Capabilities

On a Need to Know Basis



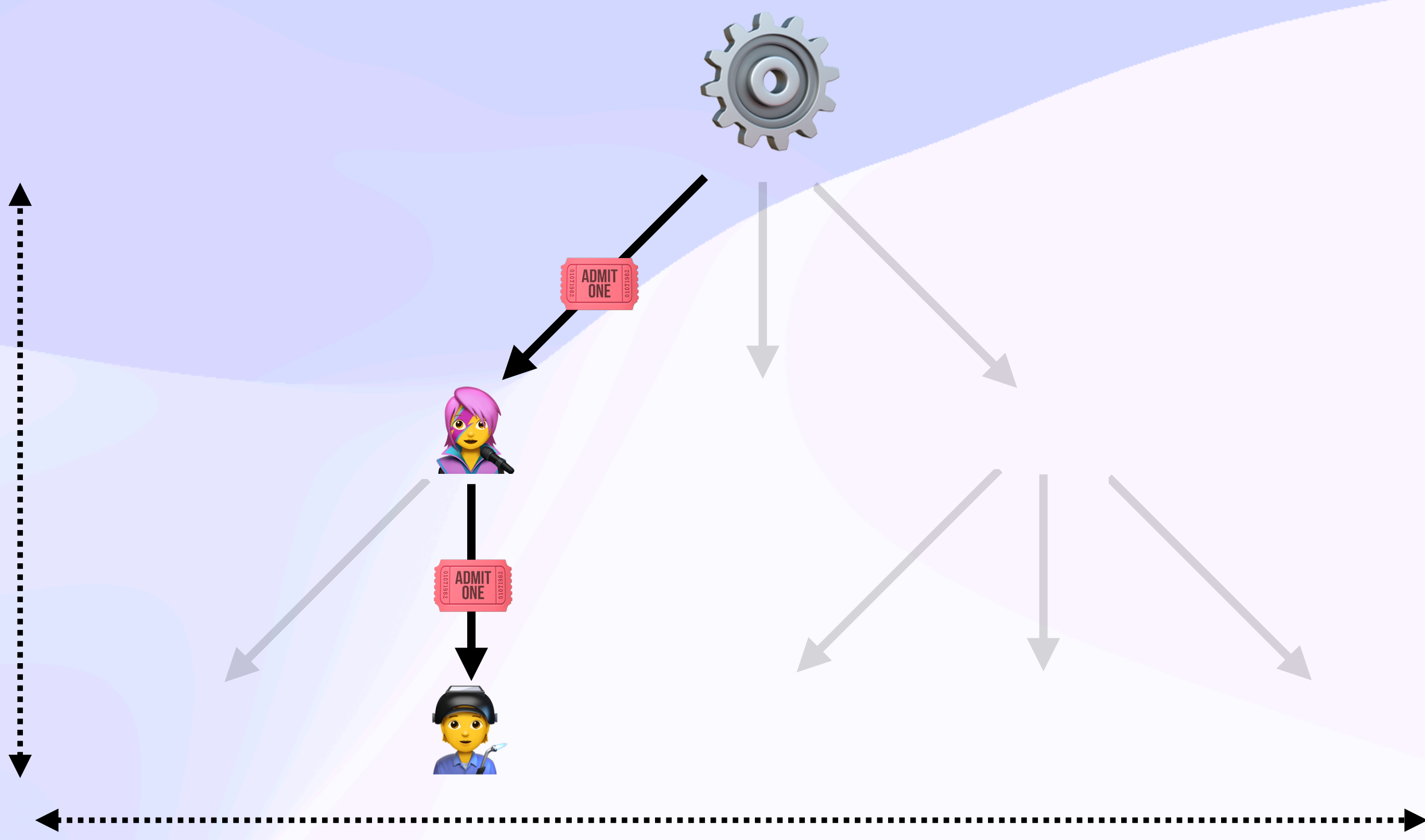
Convergent Capabilities

On a Need to Know Basis



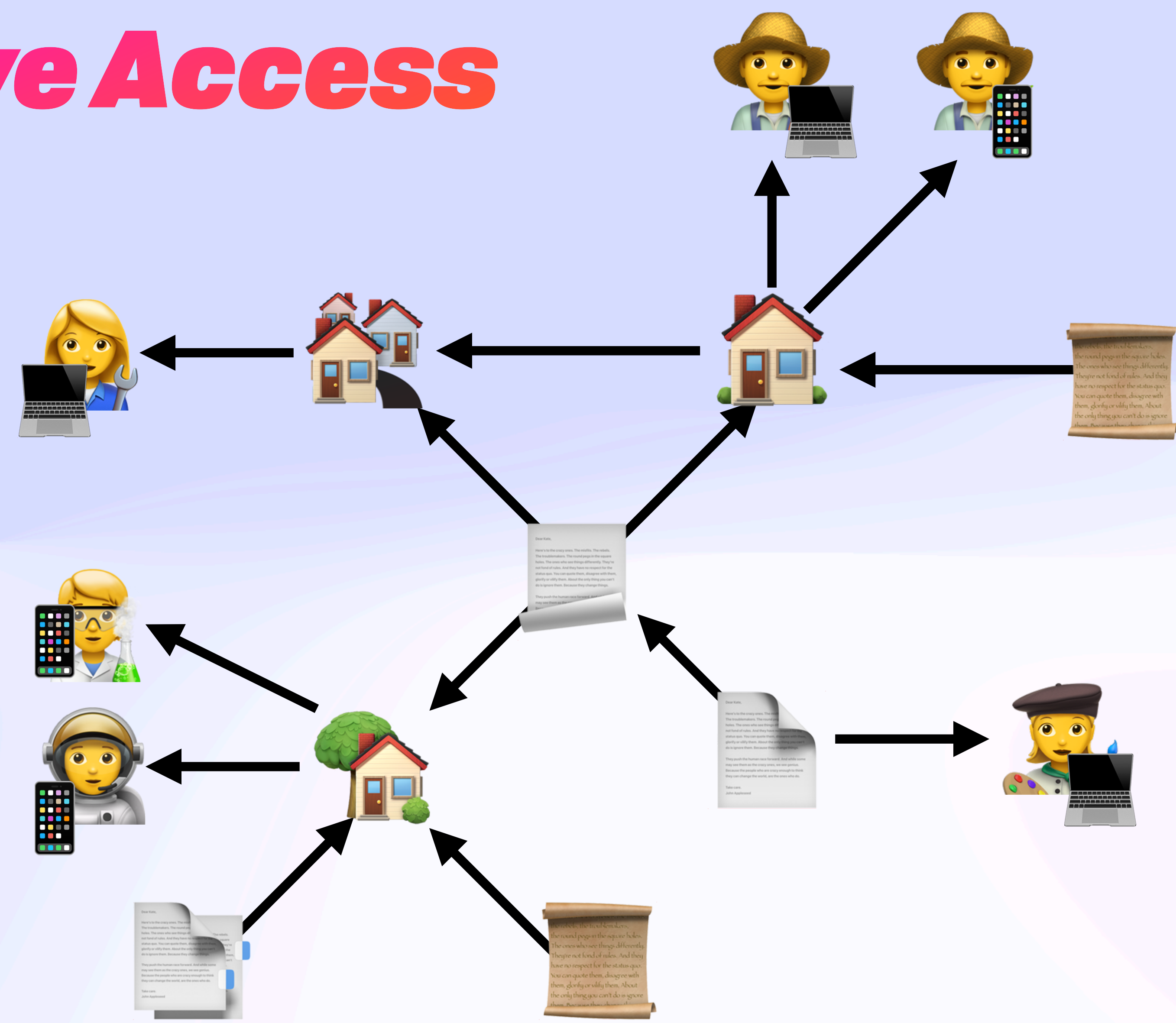
Convergent Capabilities

On a Need to Know Basis

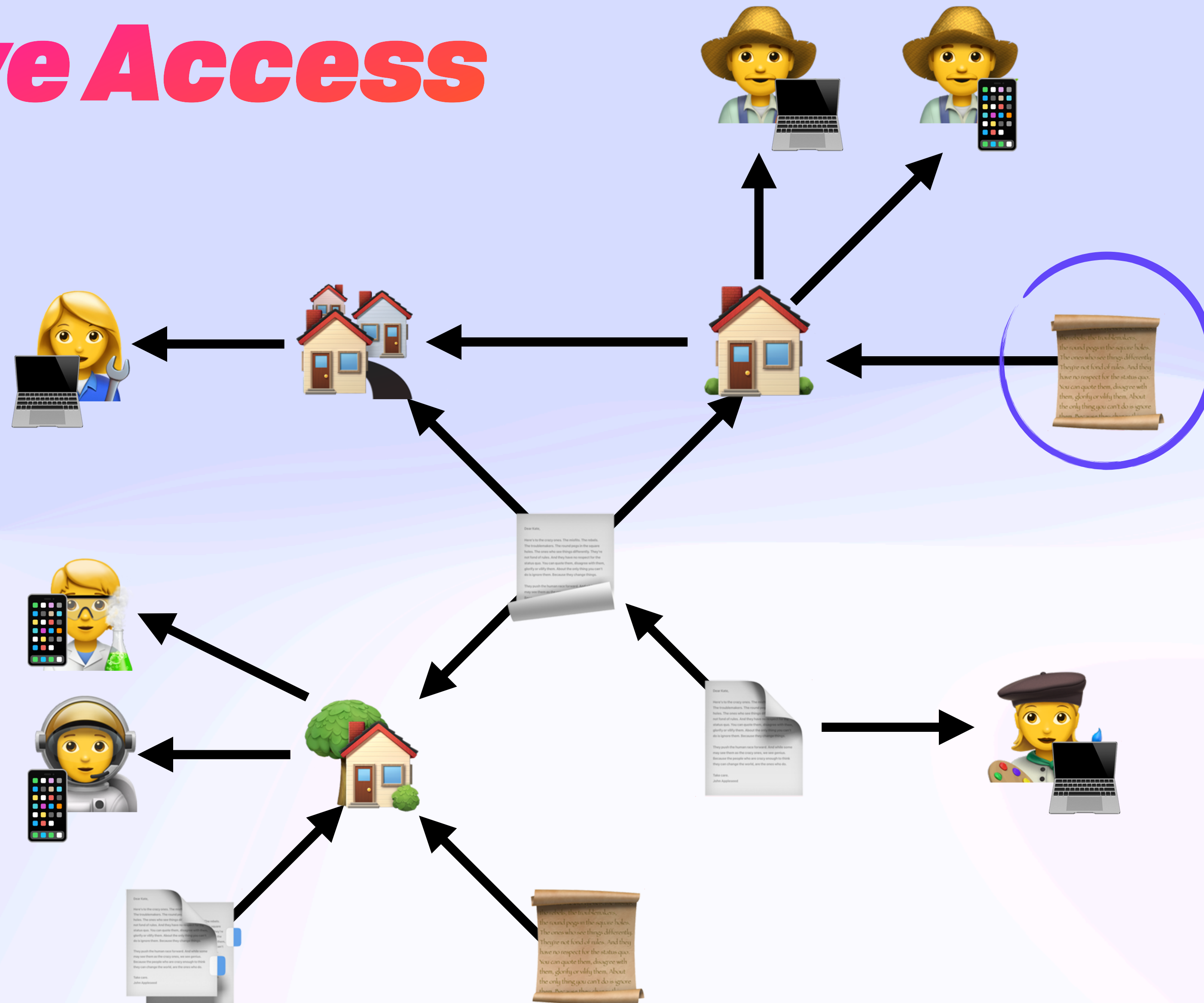


Convergent Capabilities

Transitive Access

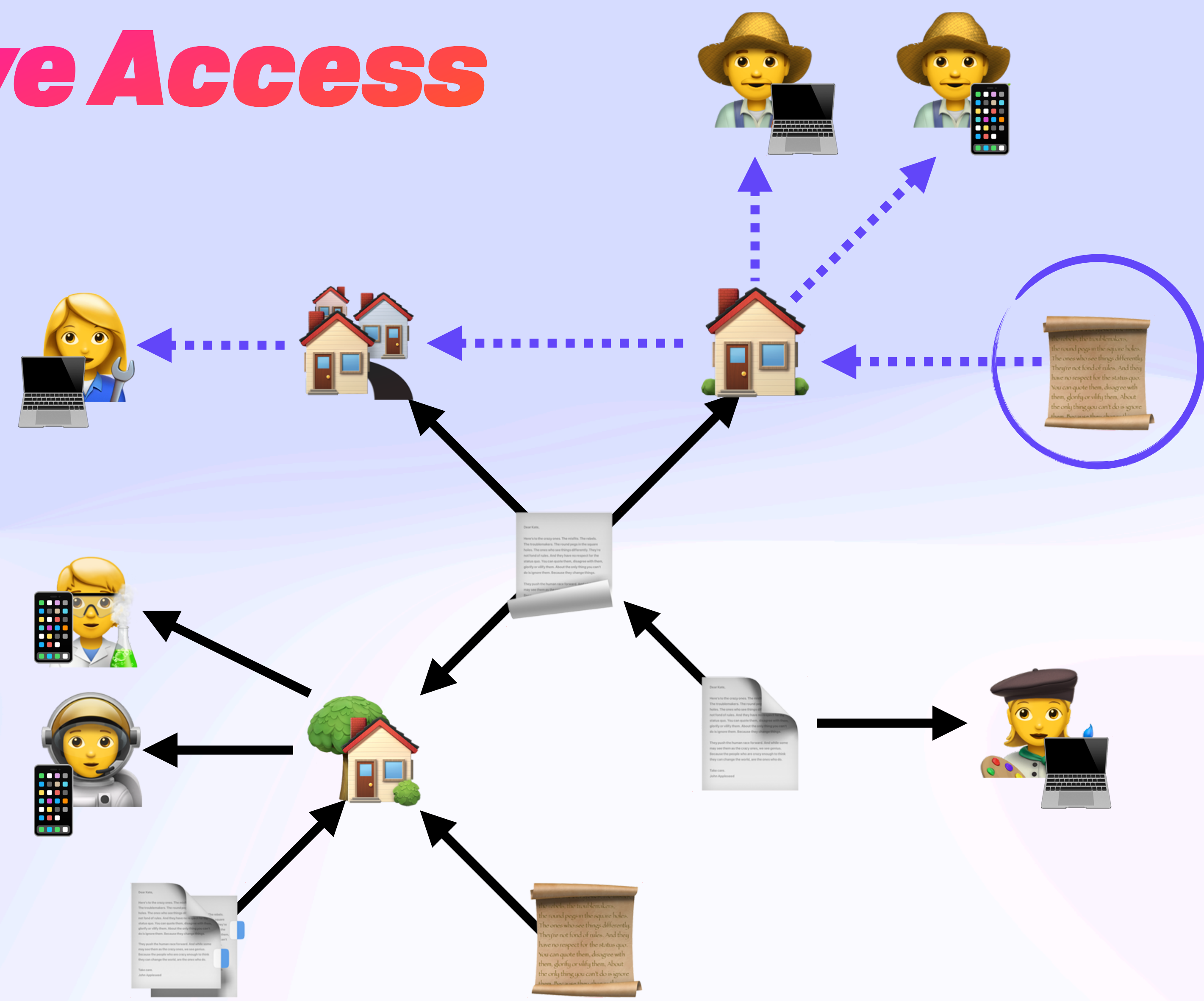


Transitive Access



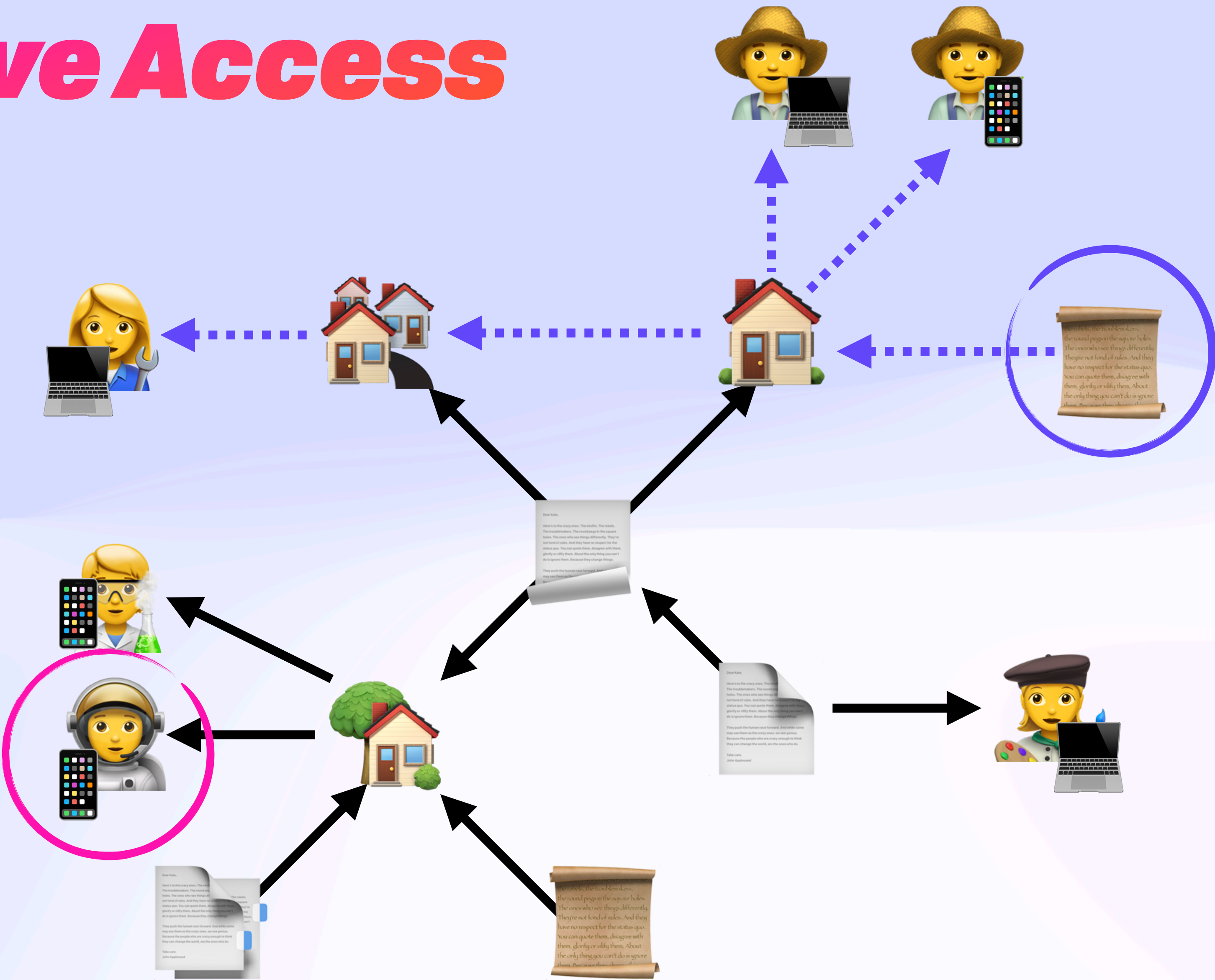
Convergent Capabilities

Transitive Access



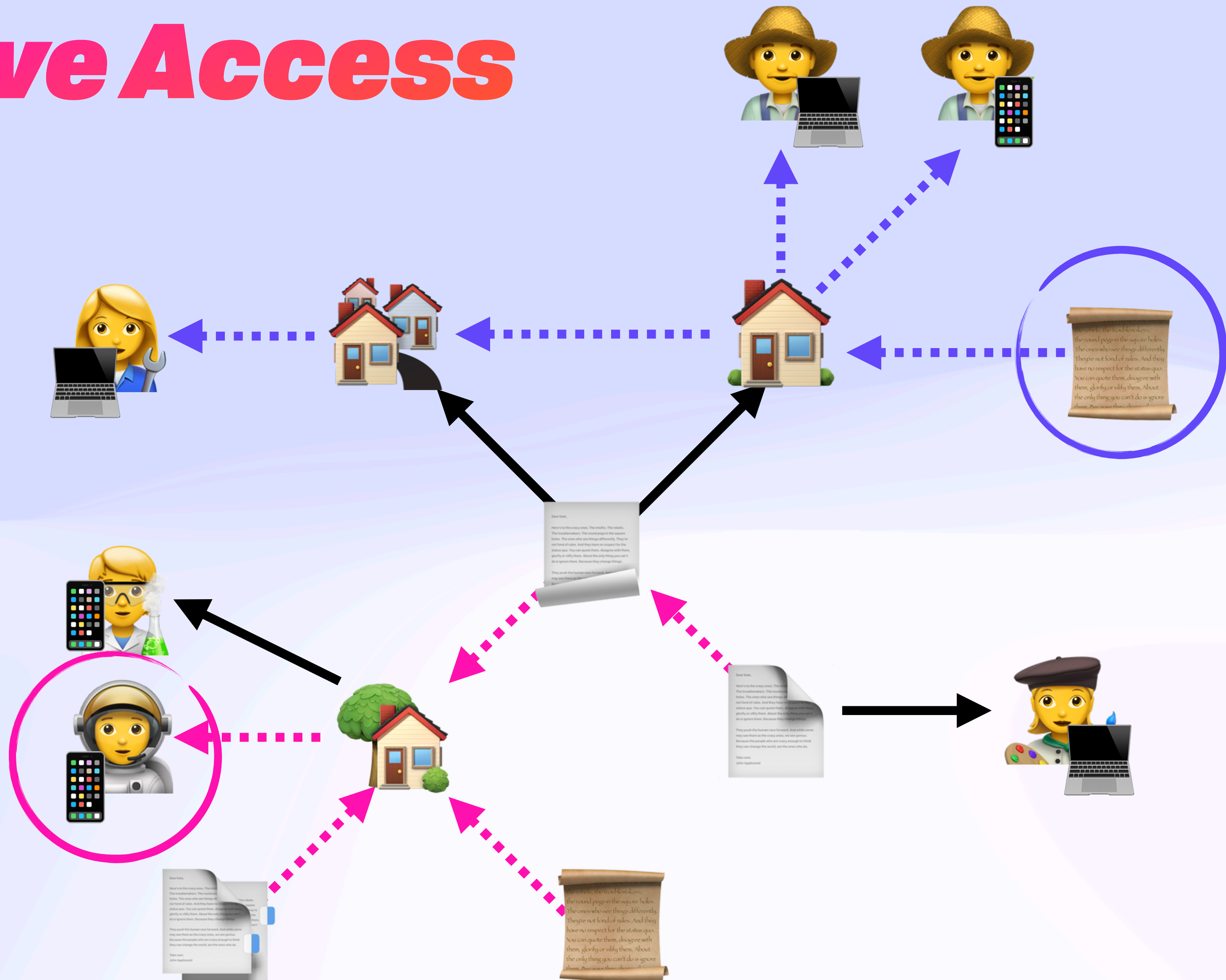
Convergent Capabilities

Transitive Access



Convergent Capabilities

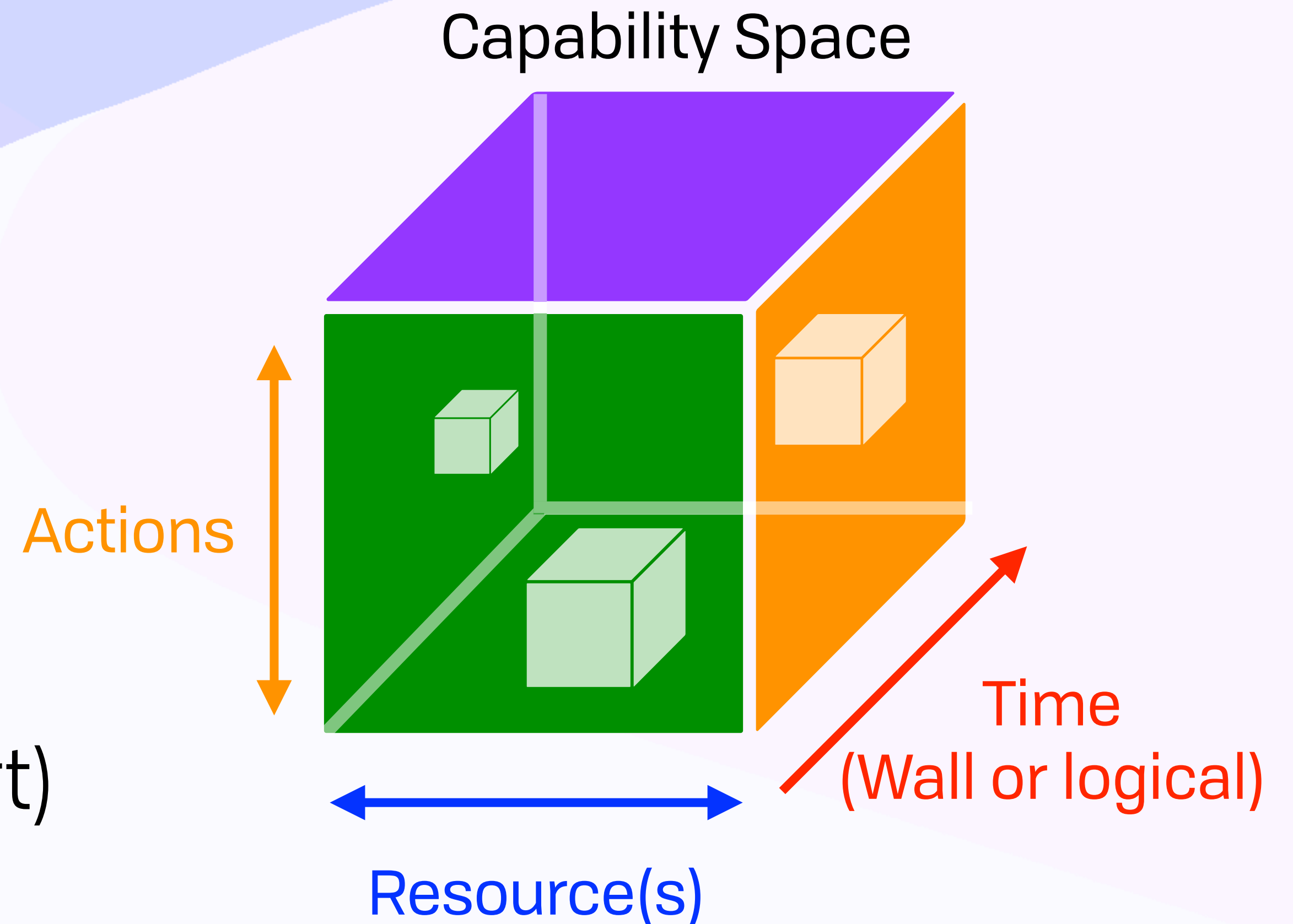
Transitive Access



Convergent Capabilities

Capability Space 🪐

- Make the box as small as possible!
- Fewest resources
- Fewest actions
- Least amount of time
- (Revocation considered a last resort)



Revocation

Redaction, Post-Compromise Security, and Cycle Breaking

Revocation

Eventually Consistent Revocation

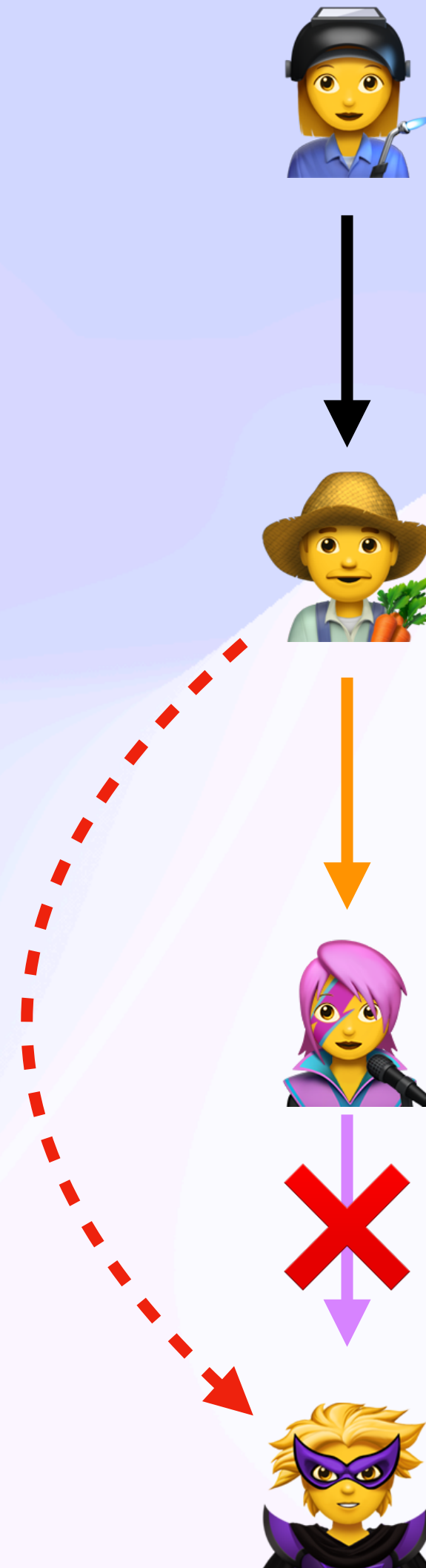
Revocation

Eventually Consistent Revocation



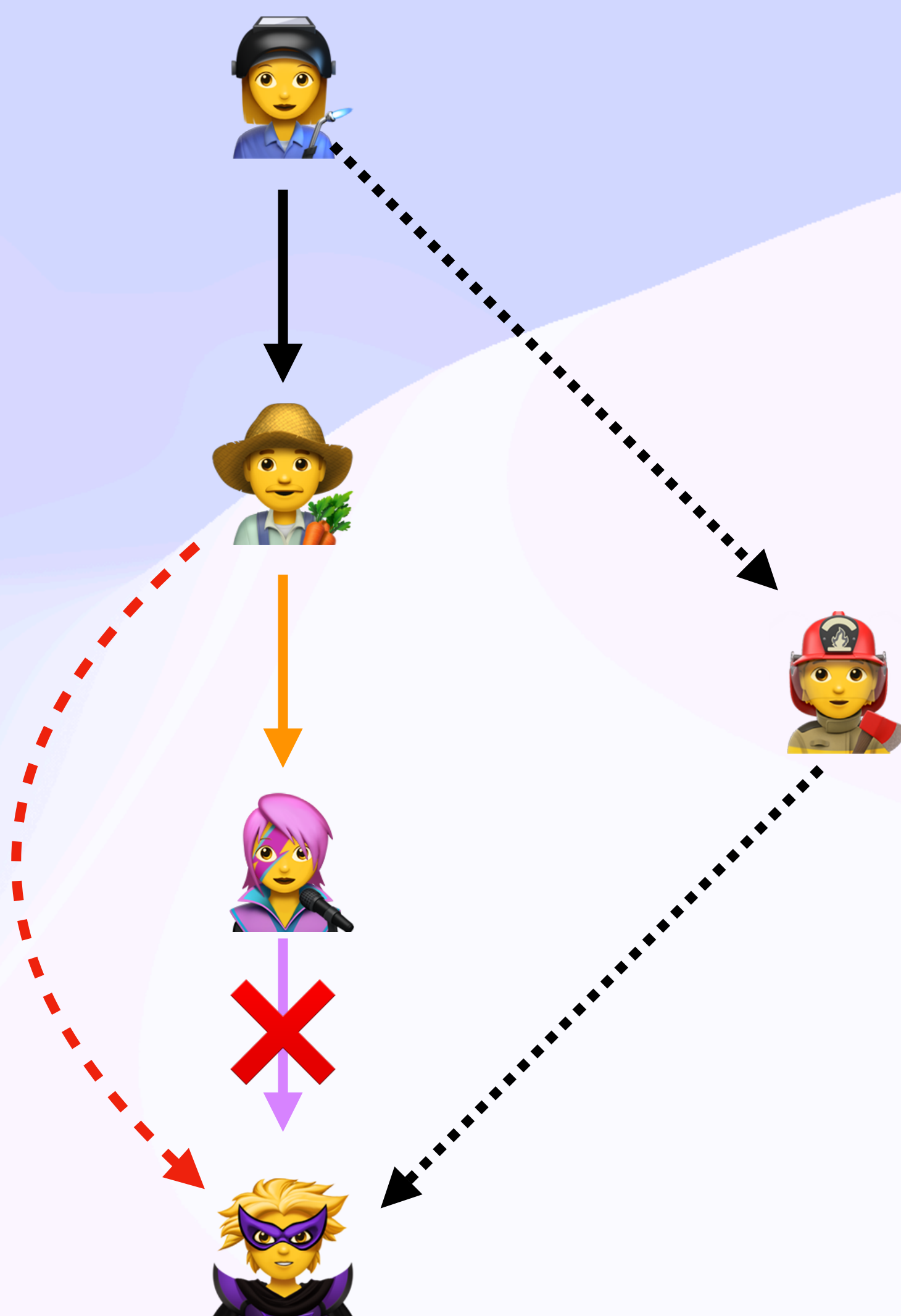
Revocation

Eventually Consistent Revocation



Revocation

Eventually Consistent Revocation



Revocation

What To Do With Revocation Cycles?

Revocation

What To Do With Revocation Cycles?



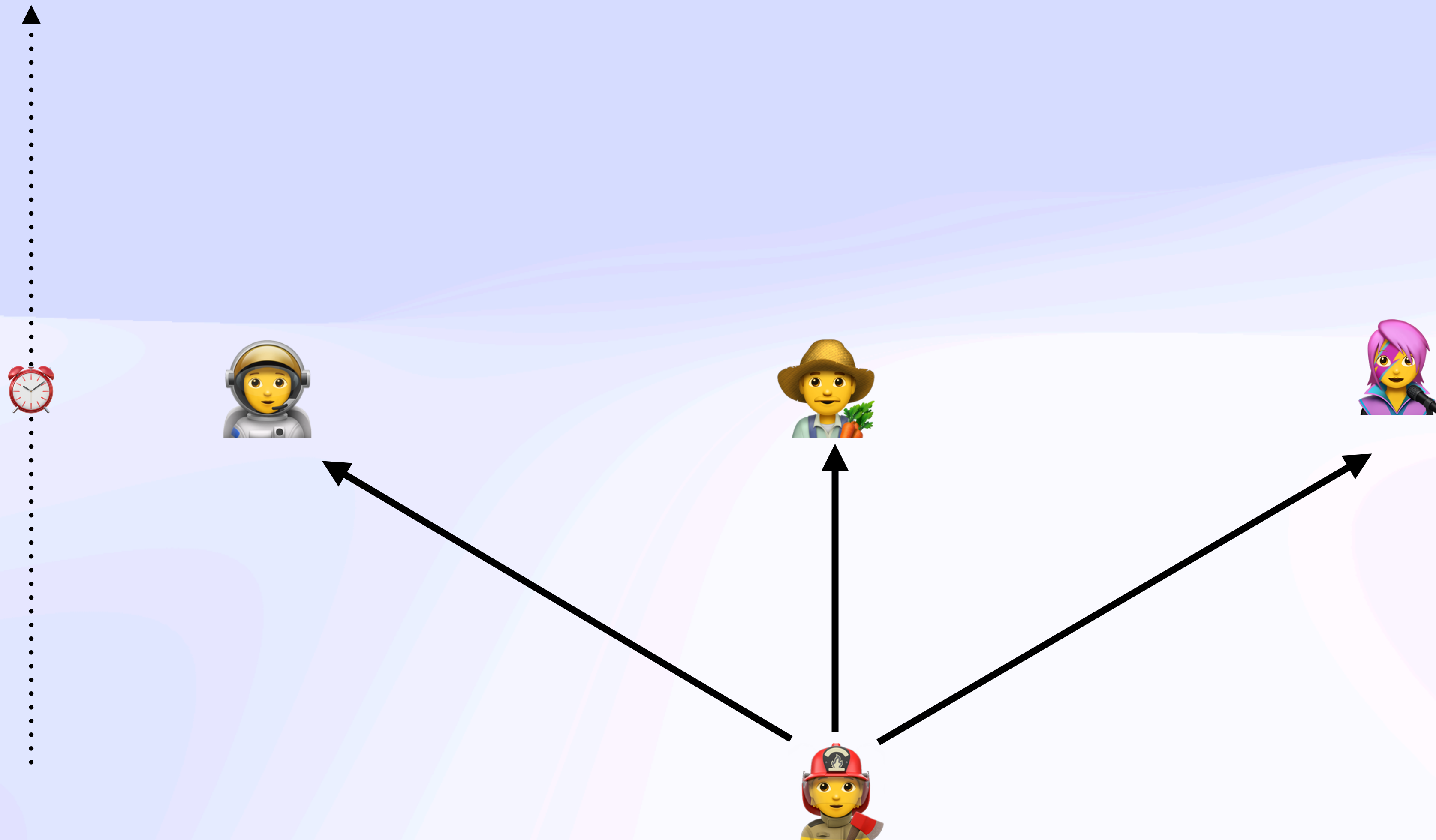
Revocation

What To Do With Revocation Cycles?



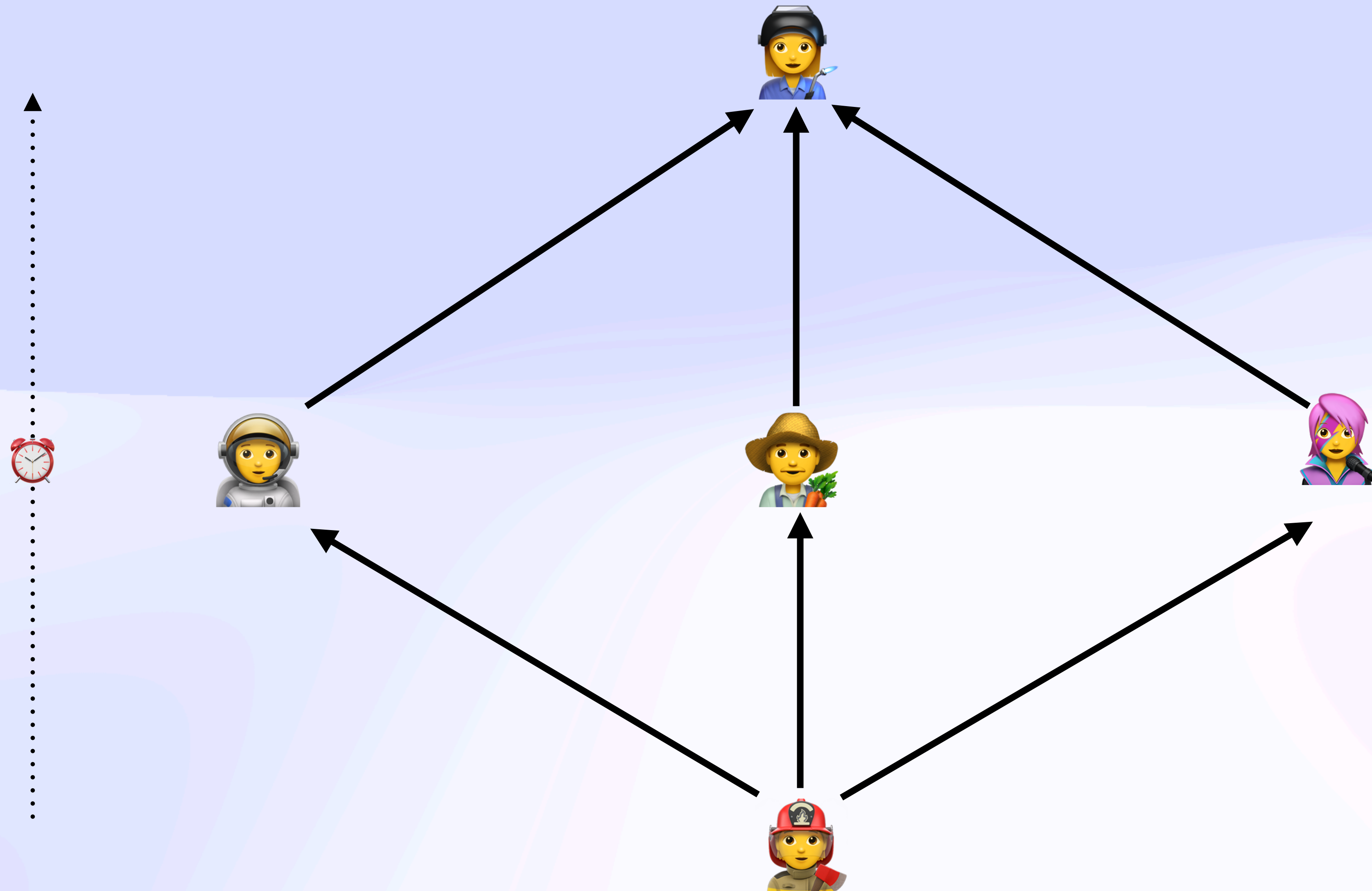
Revocation

What To Do With Revocation Cycles?



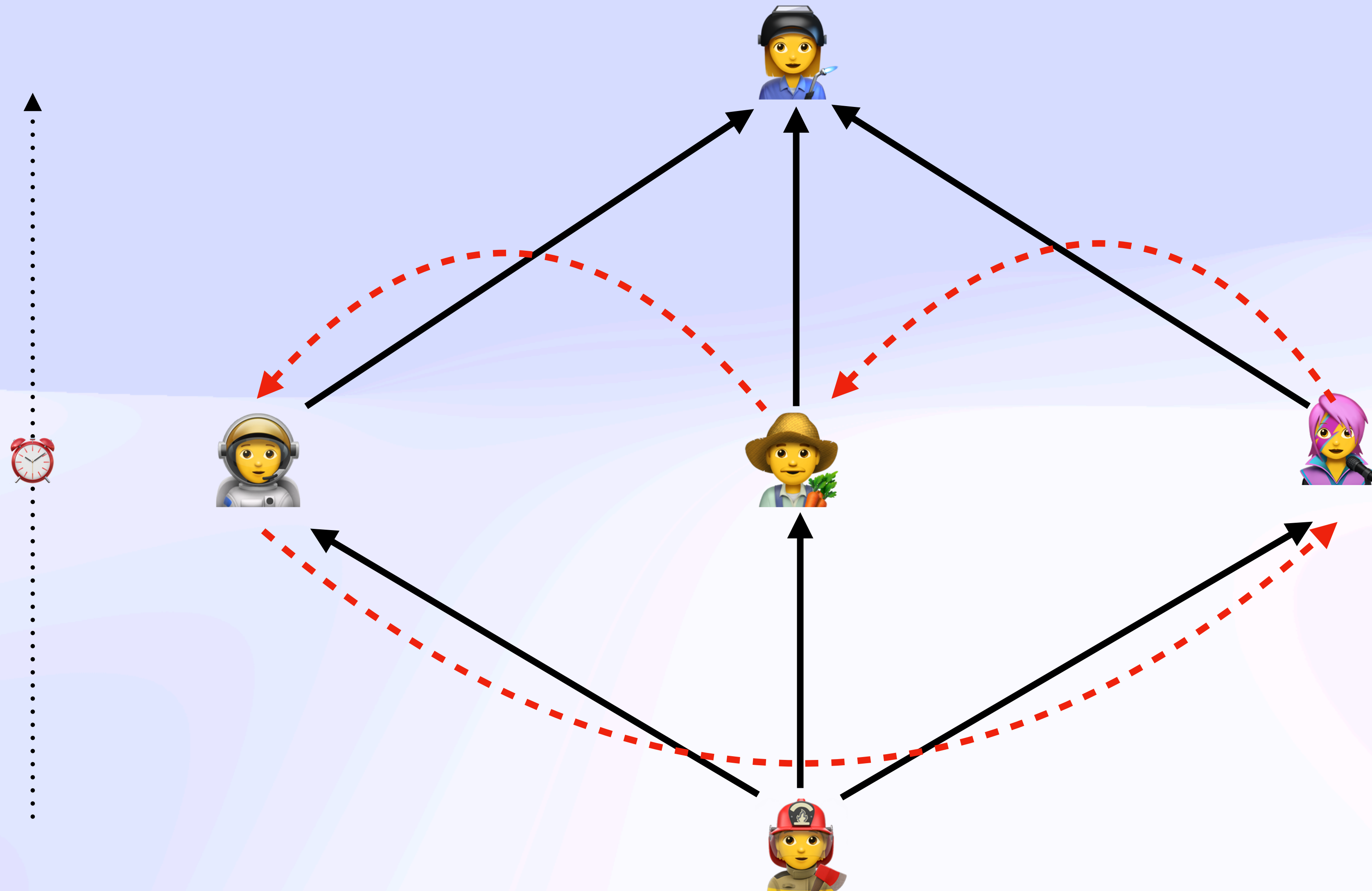
Revocation

What To Do With Revocation Cycles?



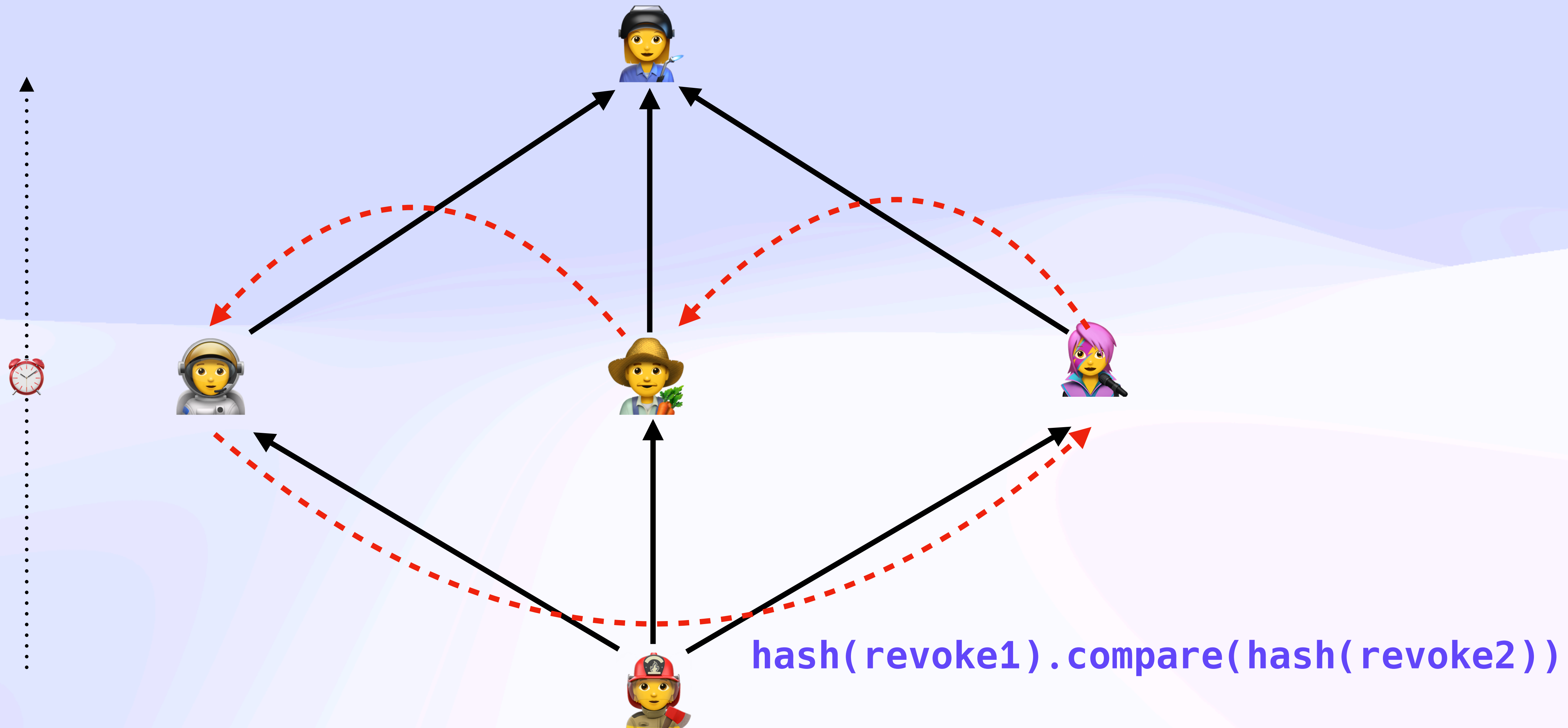
Revocation

What To Do With Revocation Cycles?



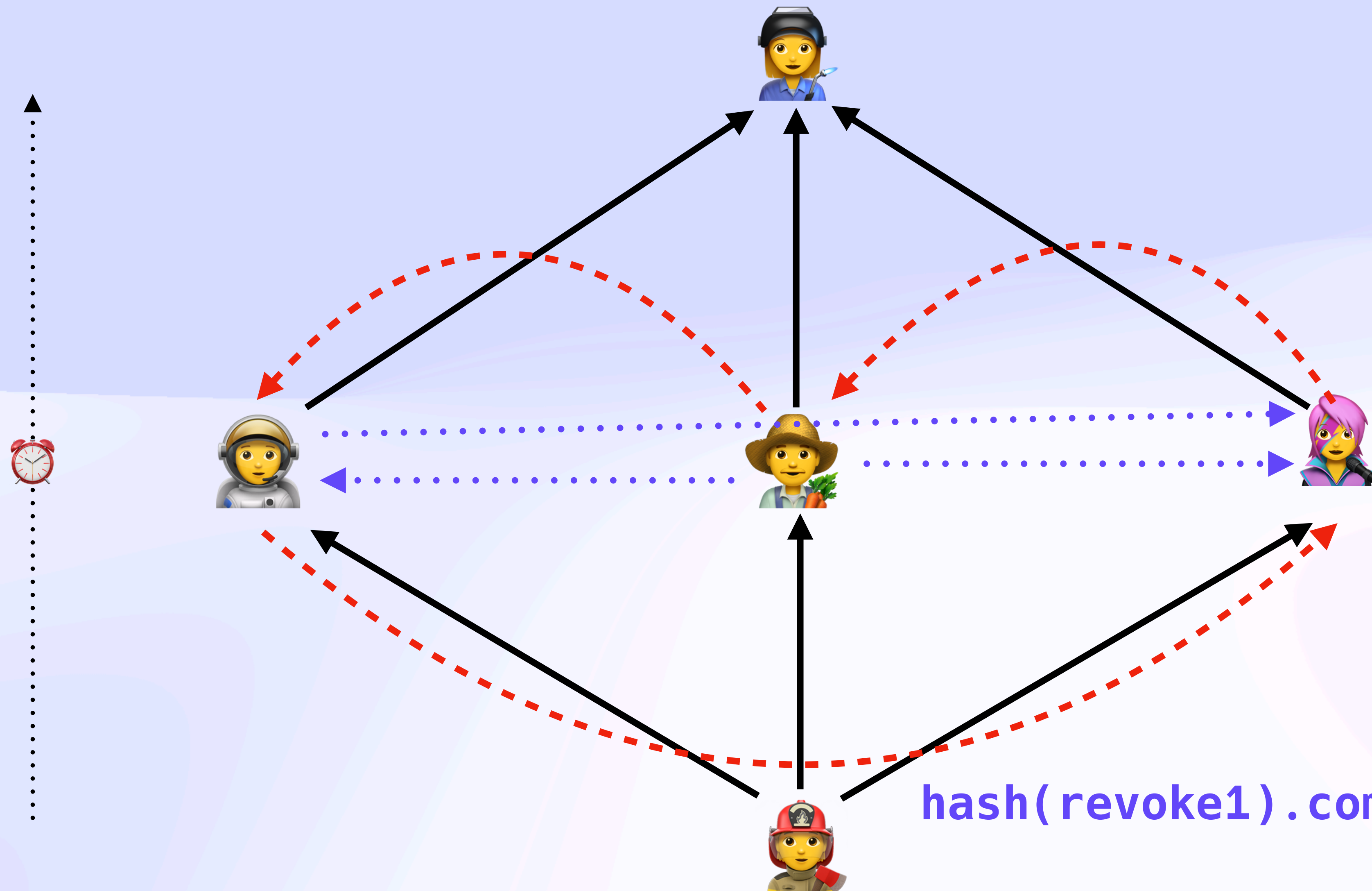
Revocation

What To Do With Revocation Cycles?



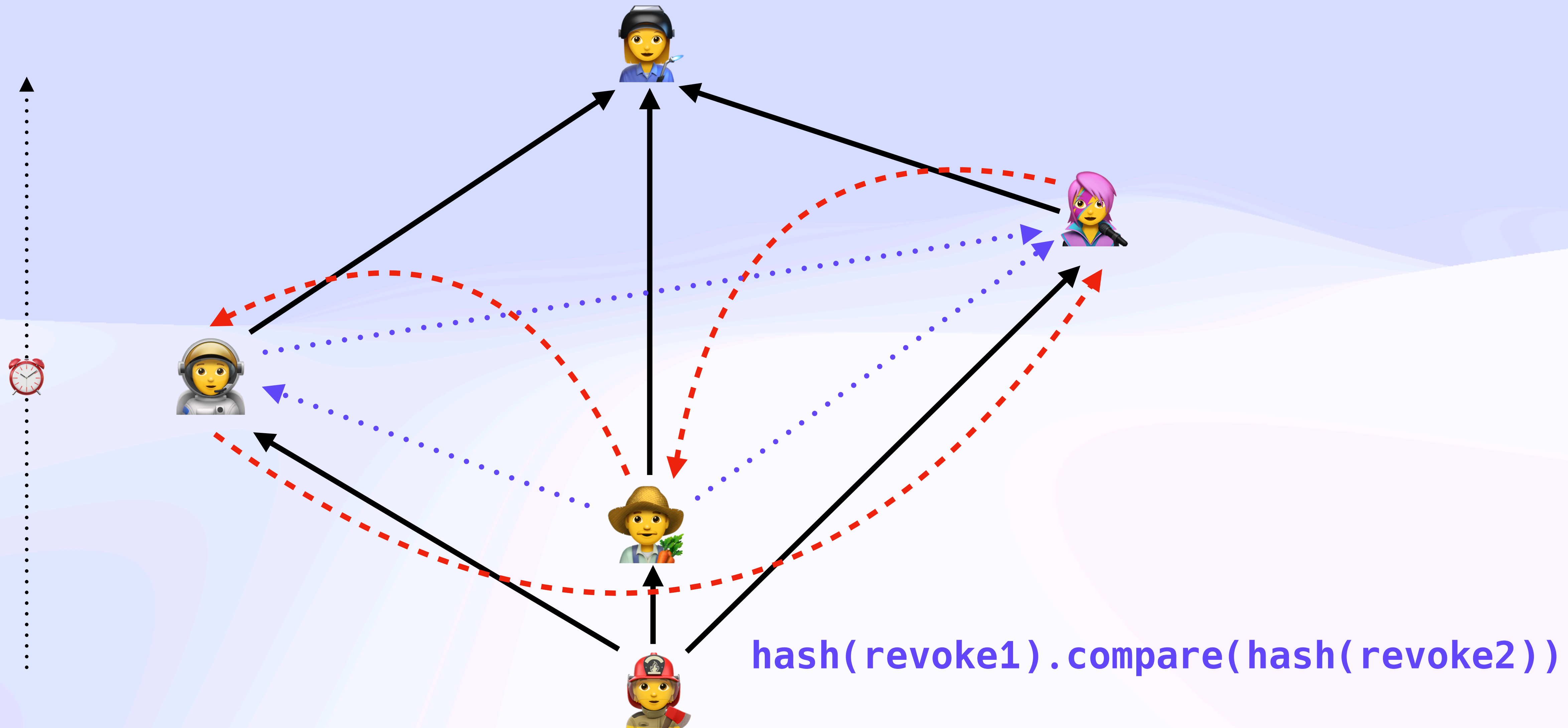
Revocation

What To Do With Revocation Cycles?



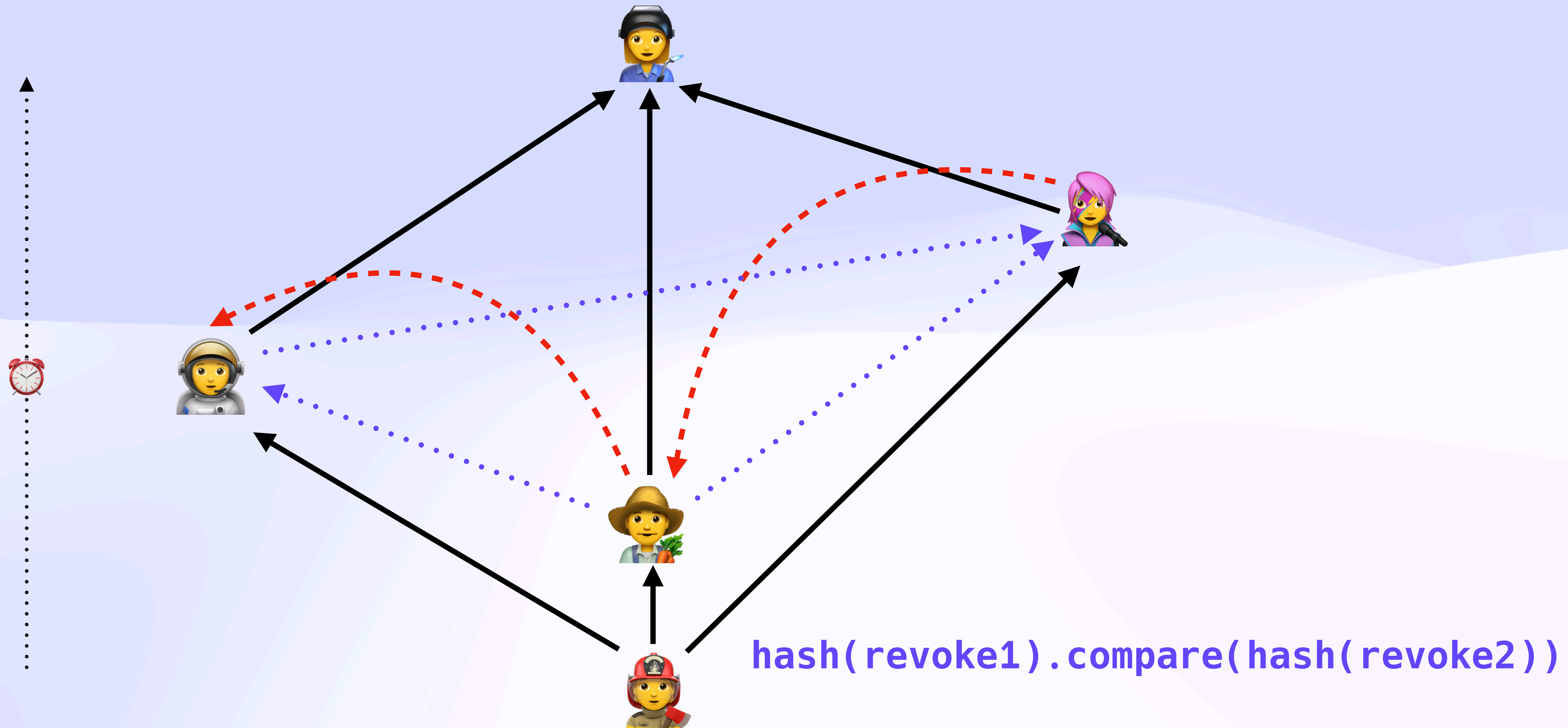
Revocation

What To Do With Revocation Cycles?



Revocation

What To Do With Revocation Cycles?



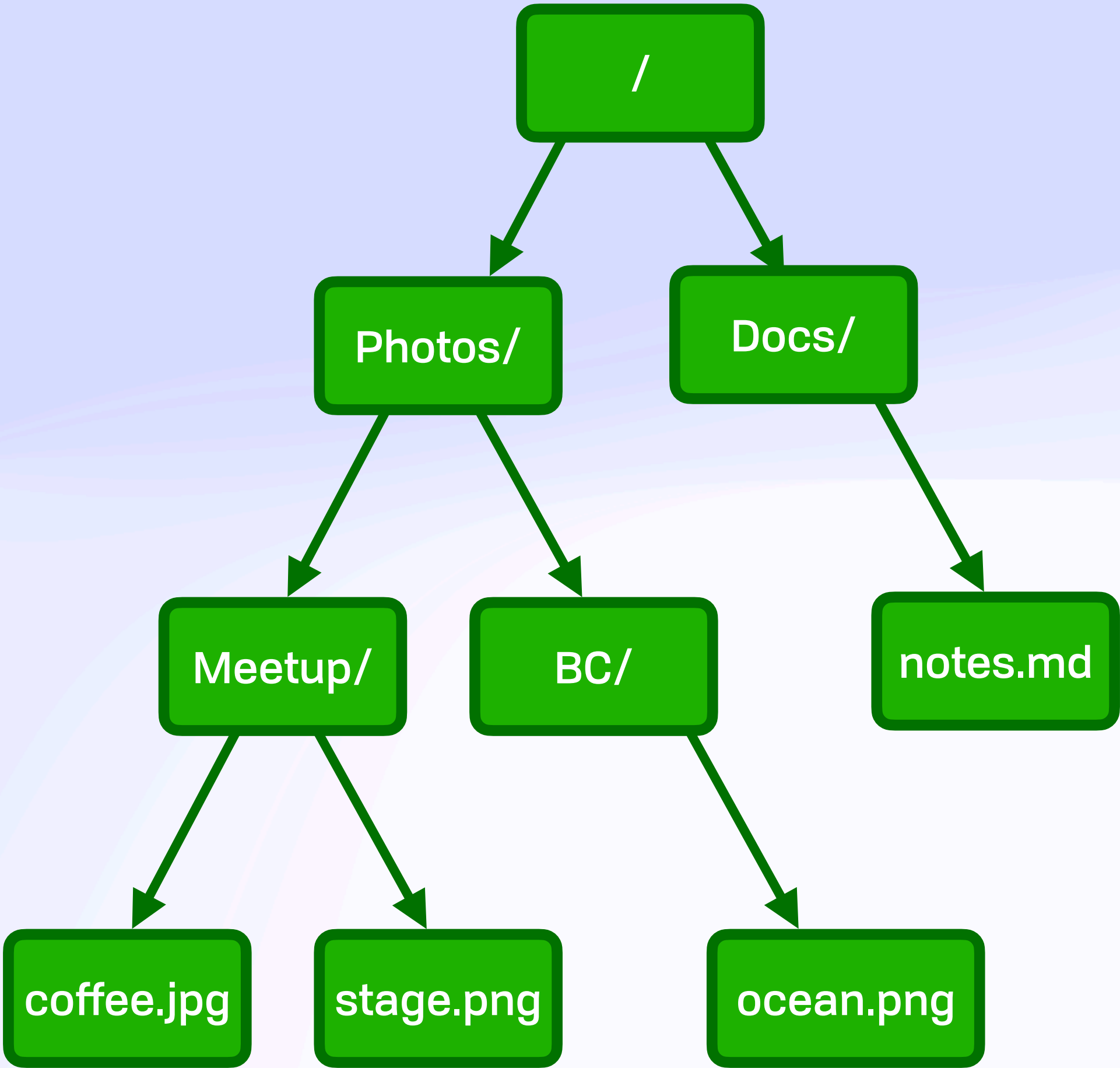
Self-Healing Concurrent Group Encryption

BeeKEM



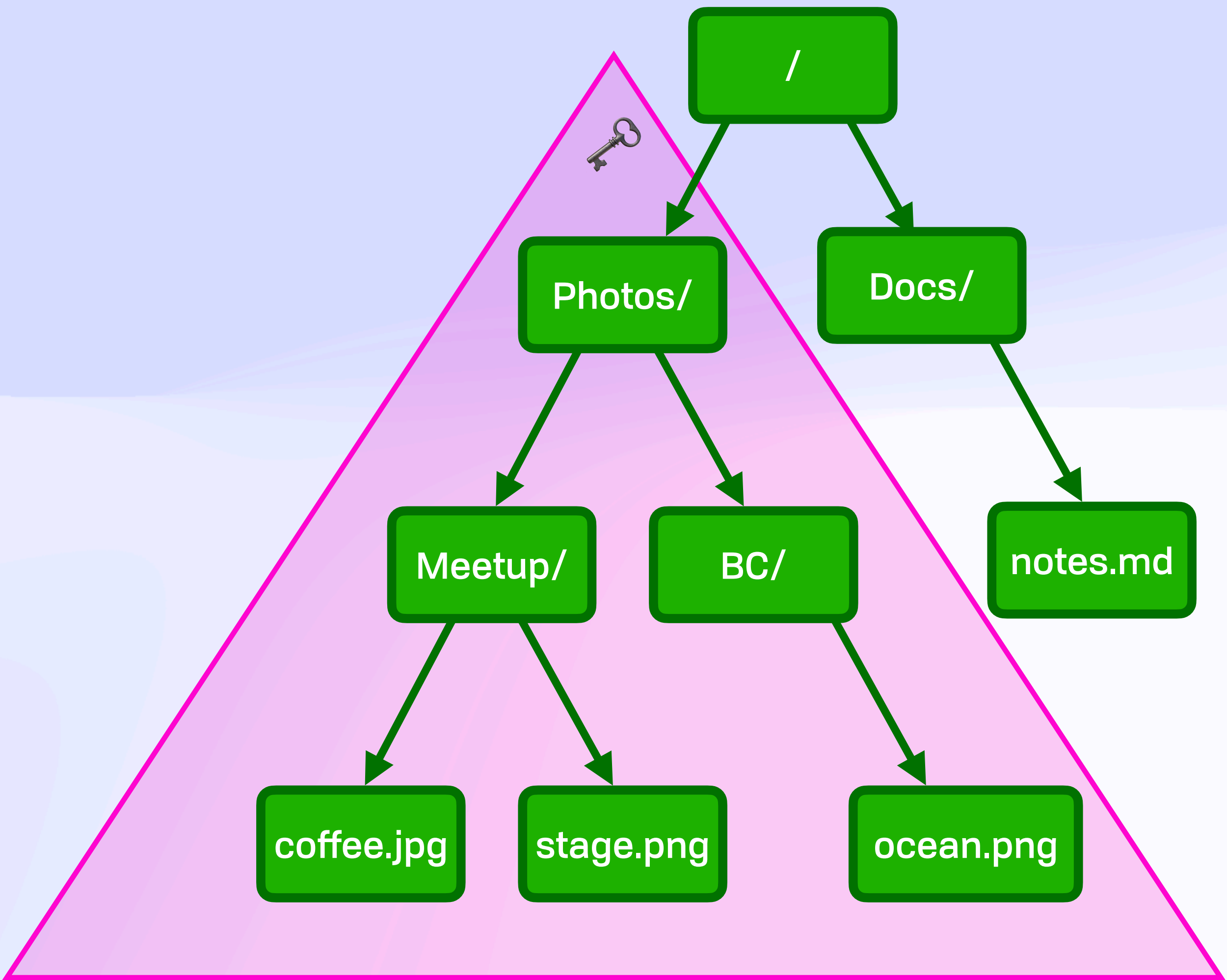
Self-Healing Concurrent Group Encryption

Transitive Read Control



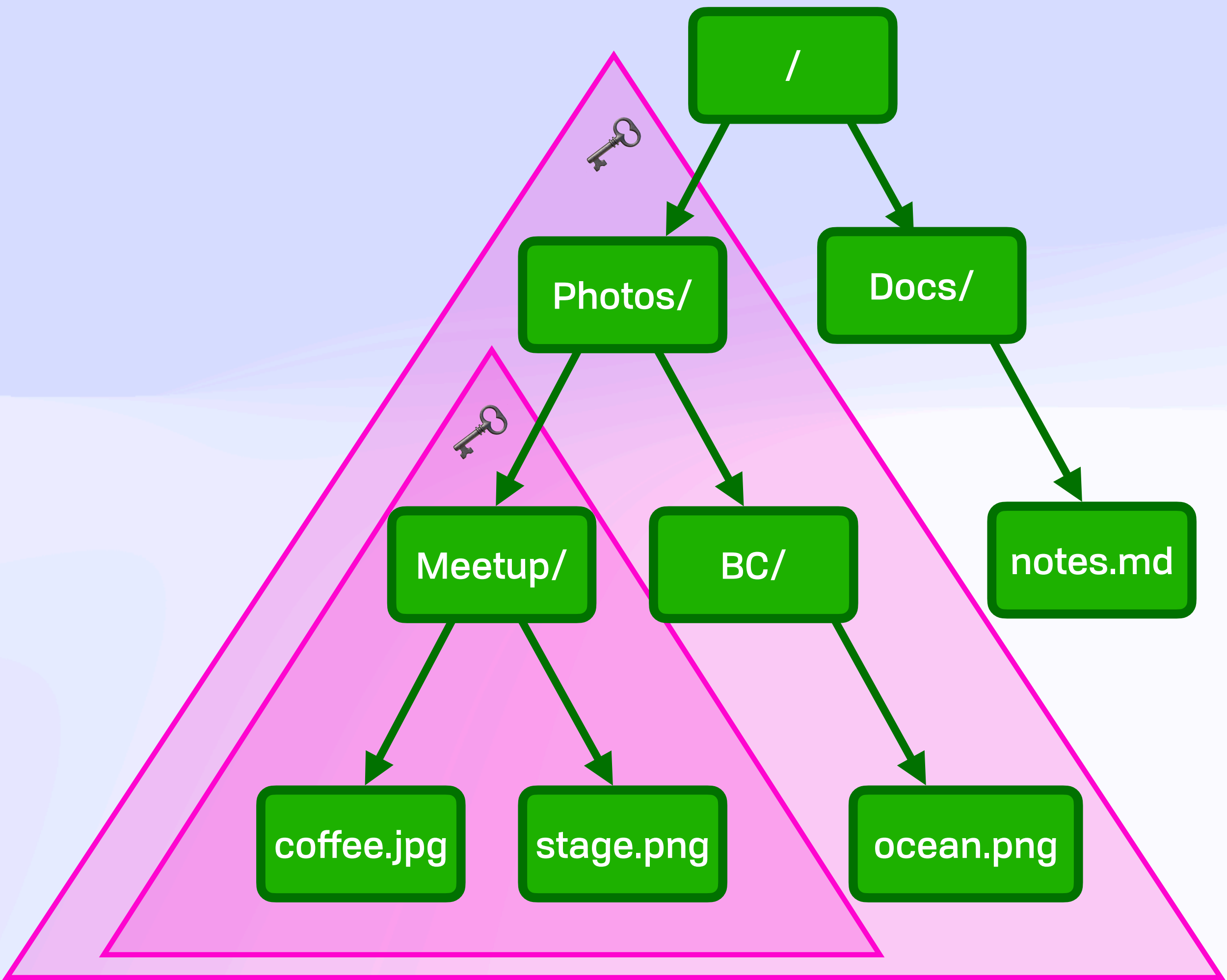
Self-Healing Concurrent Group Encryption

Transitive Read Control



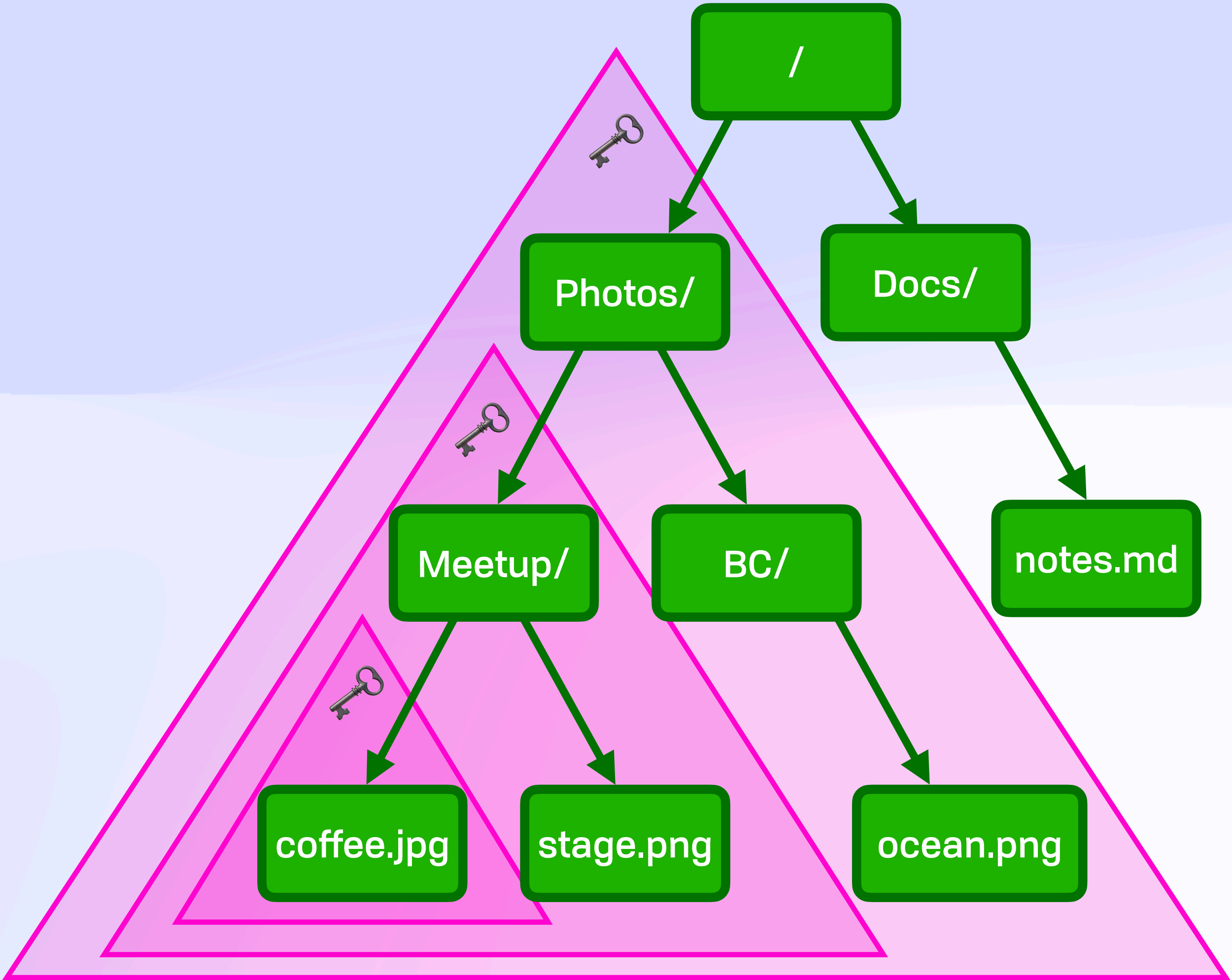
Self-Healing Concurrent Group Encryption

Transitive Read Control



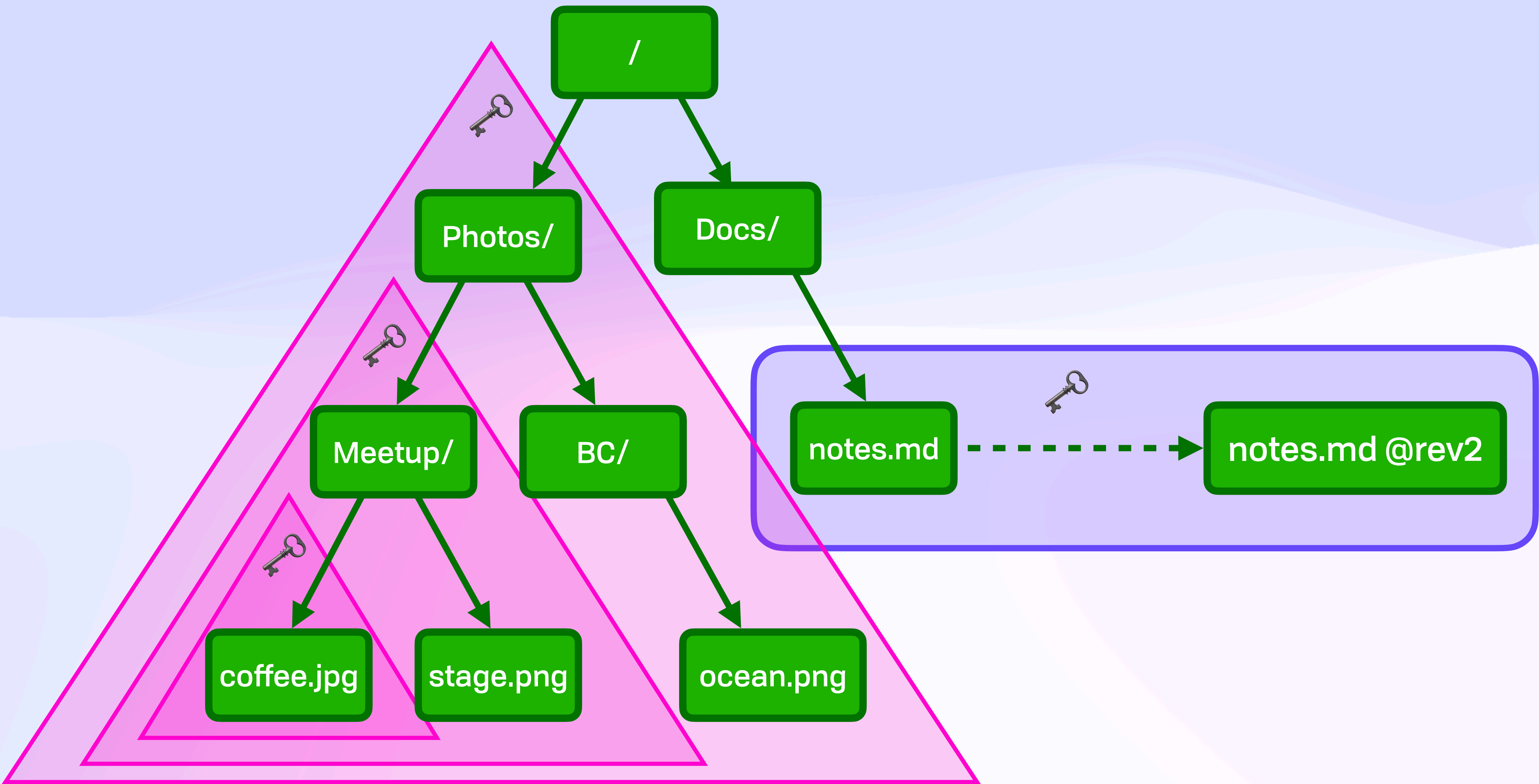
Self-Healing Concurrent Group Encryption

Transitive Read Control



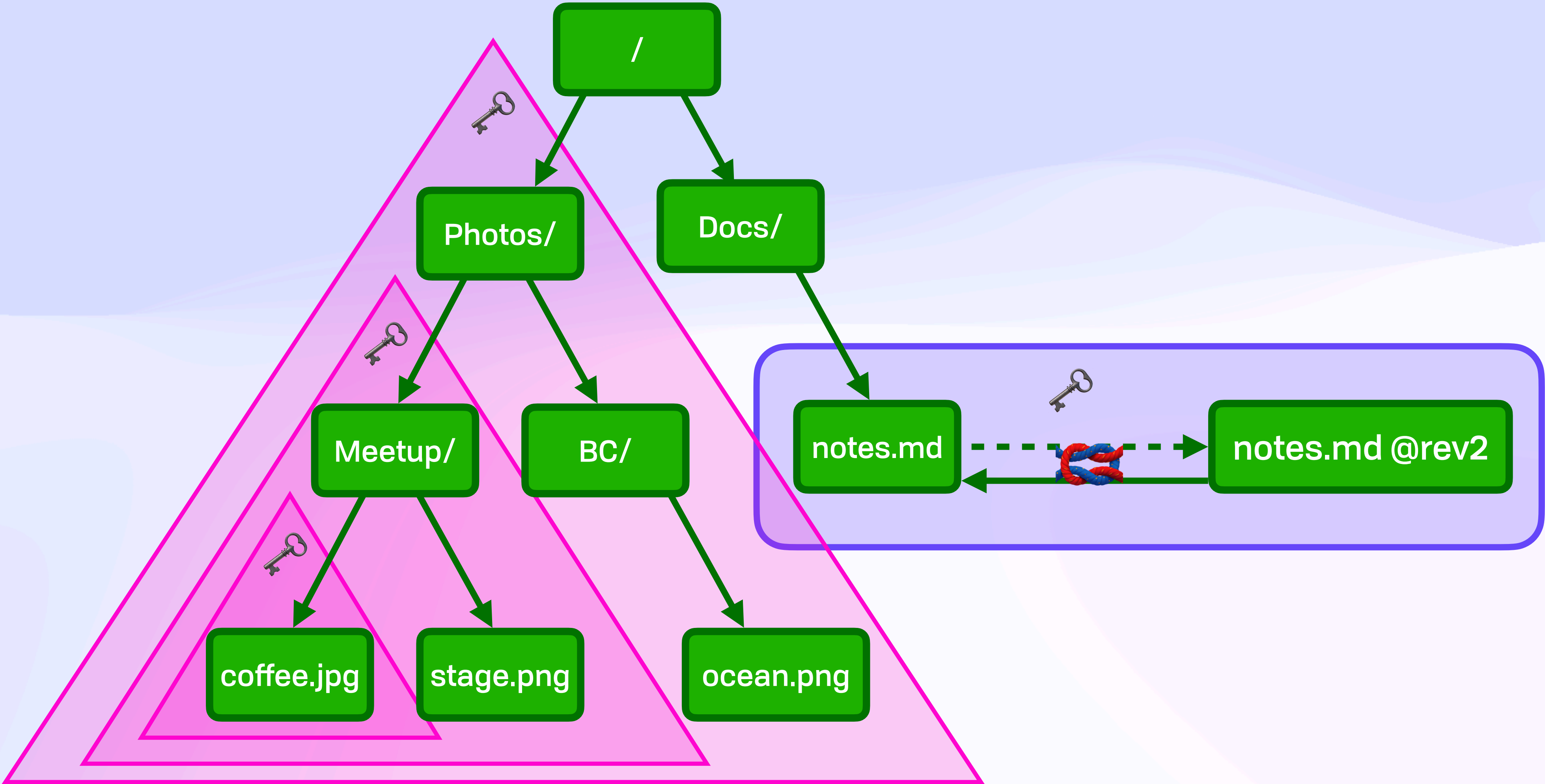
Self-Healing Concurrent Group Encryption

Transitive Read Control



Self-Healing Concurrent Group Encryption

Transitive Read Control



Self-Healing Concurrent Group Encryption

Security Over Time

Self-Healing Concurrent Group Encryption

Security Over Time

Dear Kate,

Here's to the crazy ones. The mad. The troublemakers. The round pegs in square holes. The ones who see things others don't. And they have not regard for status quo. You can quote them, disagree with them, glorify or vilify them. About the only thing you can't do is ignore them. Because they change things.

They push the human race forward. And while some may see them as the crazy ones, we see genius. Because the people who are crazy enough to think they can change the world, are the ones who do.

Take care,
John Appleseed

Dear Kate,

Here's to the crazy ones. The mad. The troublemakers. The round pegs in square holes. The ones who see things others don't. And they have not regard for status quo. You can quote them, disagree with them, glorify or vilify them. About the only thing you can't do is ignore them. Because they change things.

They push the human race forward. And while some may see them as the crazy ones, we see genius. Because the people who are crazy enough to think they can change the world, are the ones who do.

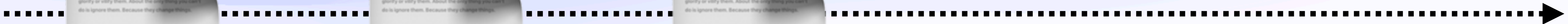
Take care,
John Appleseed

Dear Kate,

Here's to the crazy ones. The mad. The troublemakers. The round pegs in square holes. The ones who see things others don't. And they have not regard for status quo. You can quote them, disagree with them, glorify or vilify them. About the only thing you can't do is ignore them. Because they change things.

They push the human race forward. And while some may see them as the crazy ones, we see genius. Because the people who are crazy enough to think they can change the world, are the ones who do.

Take care,
John Appleseed



Self-Healing Concurrent Group Encryption

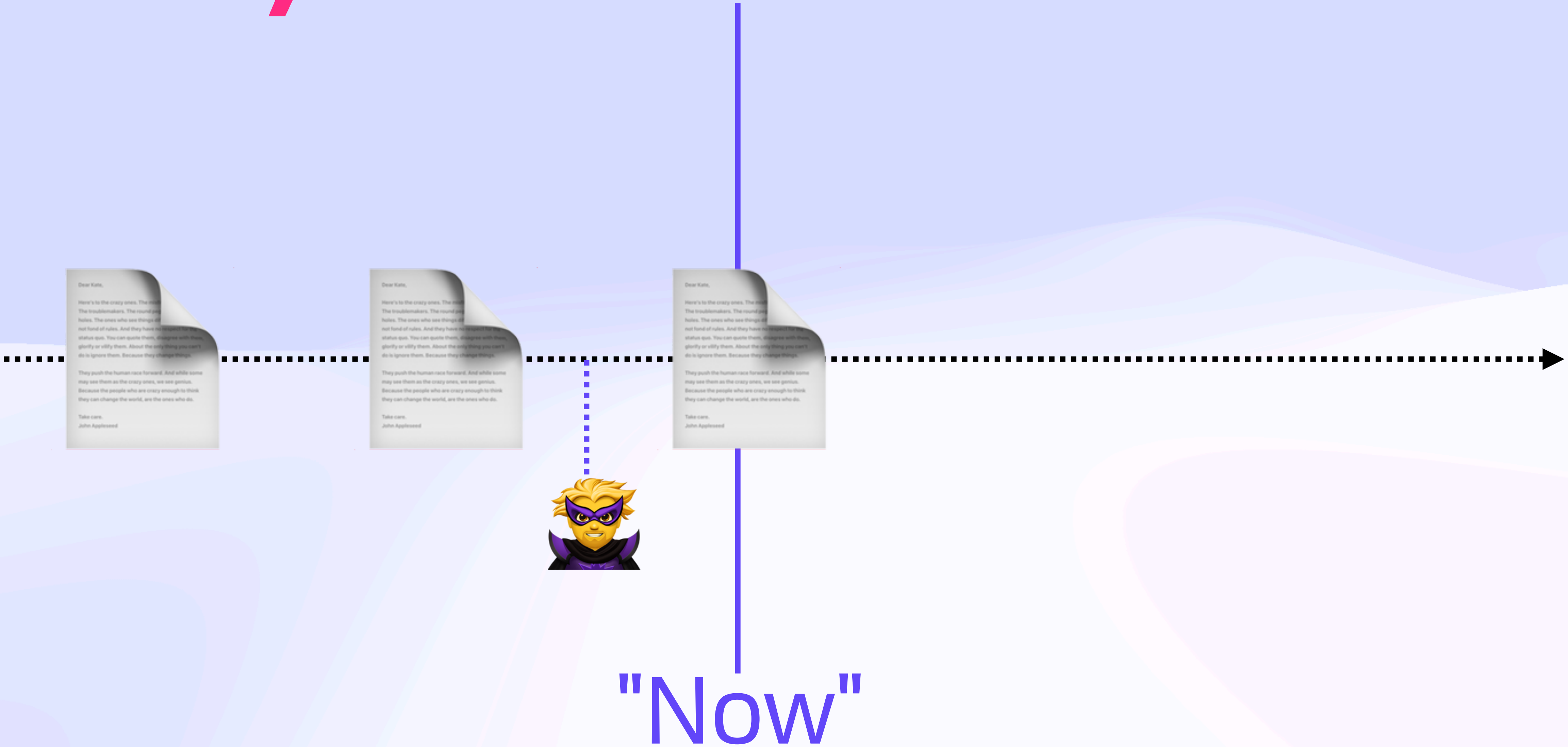
Security Over Time



"Now"

Self-Healing Concurrent Group Encryption

Security Over Time



Dear Kate,

Here's to the crazy ones. The mad. The troublemakers. The round pegs in square holes. The ones who see things of not fond of rules. And they have no respect for status quo. You can quote them, disagree with them, glorify or vilify them. About the only thing you can't do is ignore them. Because they change things.

They push the human race forward. And while some may see them as the crazy ones, we see genius. Because the people who are crazy enough to think they can change the world, are the ones who do.

Take care,
John Appleseed

Dear Kate,

Here's to the crazy ones. The mad. The troublemakers. The round pegs in square holes. The ones who see things of not fond of rules. And they have no respect for status quo. You can quote them, disagree with them, glorify or vilify them. About the only thing you can't do is ignore them. Because they change things.

They push the human race forward. And while some may see them as the crazy ones, we see genius. Because the people who are crazy enough to think they can change the world, are the ones who do.

Take care,
John Appleseed



Dear Kate,

Here's to the crazy ones. The mad. The troublemakers. The round pegs in square holes. The ones who see things of not fond of rules. And they have no respect for status quo. You can quote them, disagree with them, glorify or vilify them. About the only thing you can't do is ignore them. Because they change things.

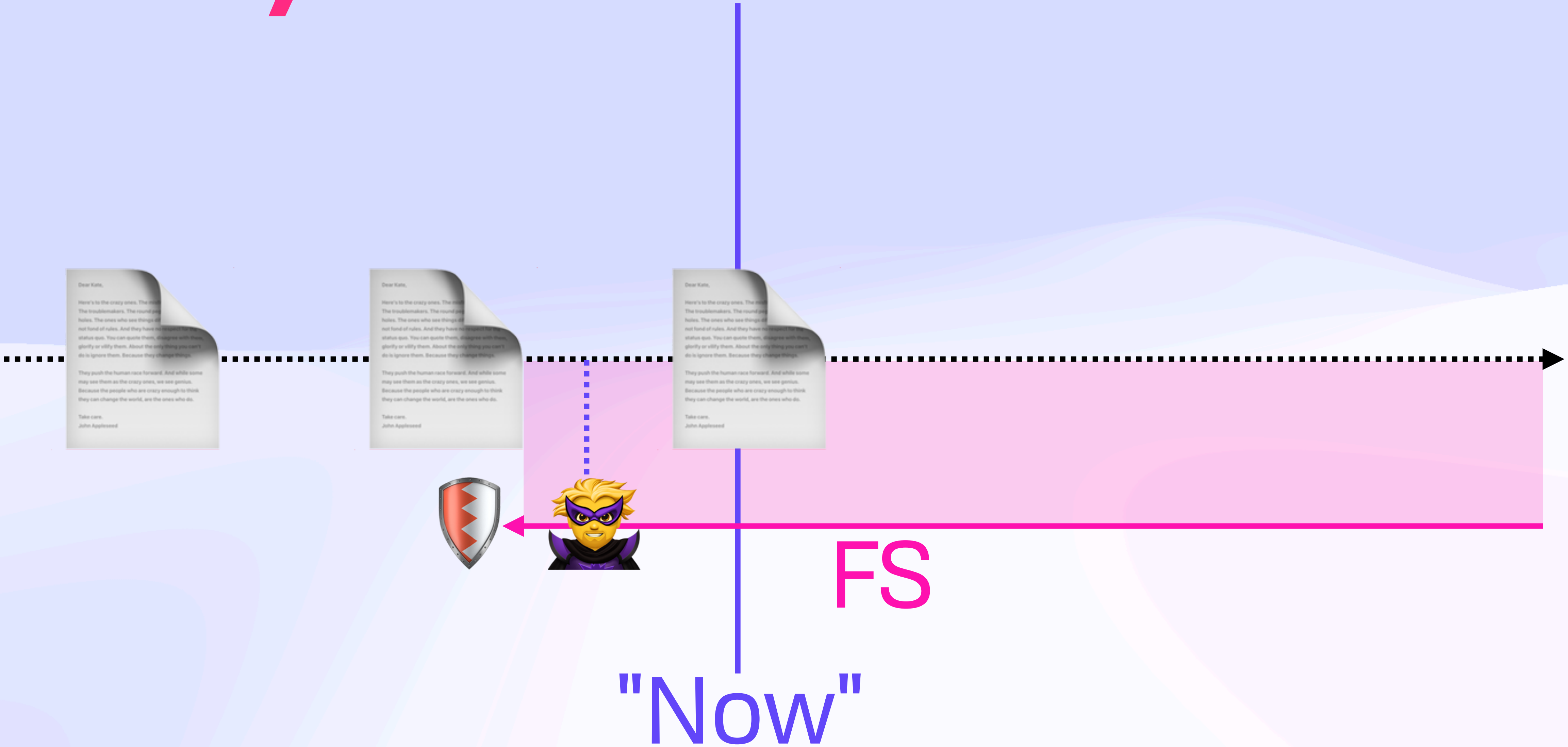
They push the human race forward. And while some may see them as the crazy ones, we see genius. Because the people who are crazy enough to think they can change the world, are the ones who do.

Take care,
John Appleseed

"Now"

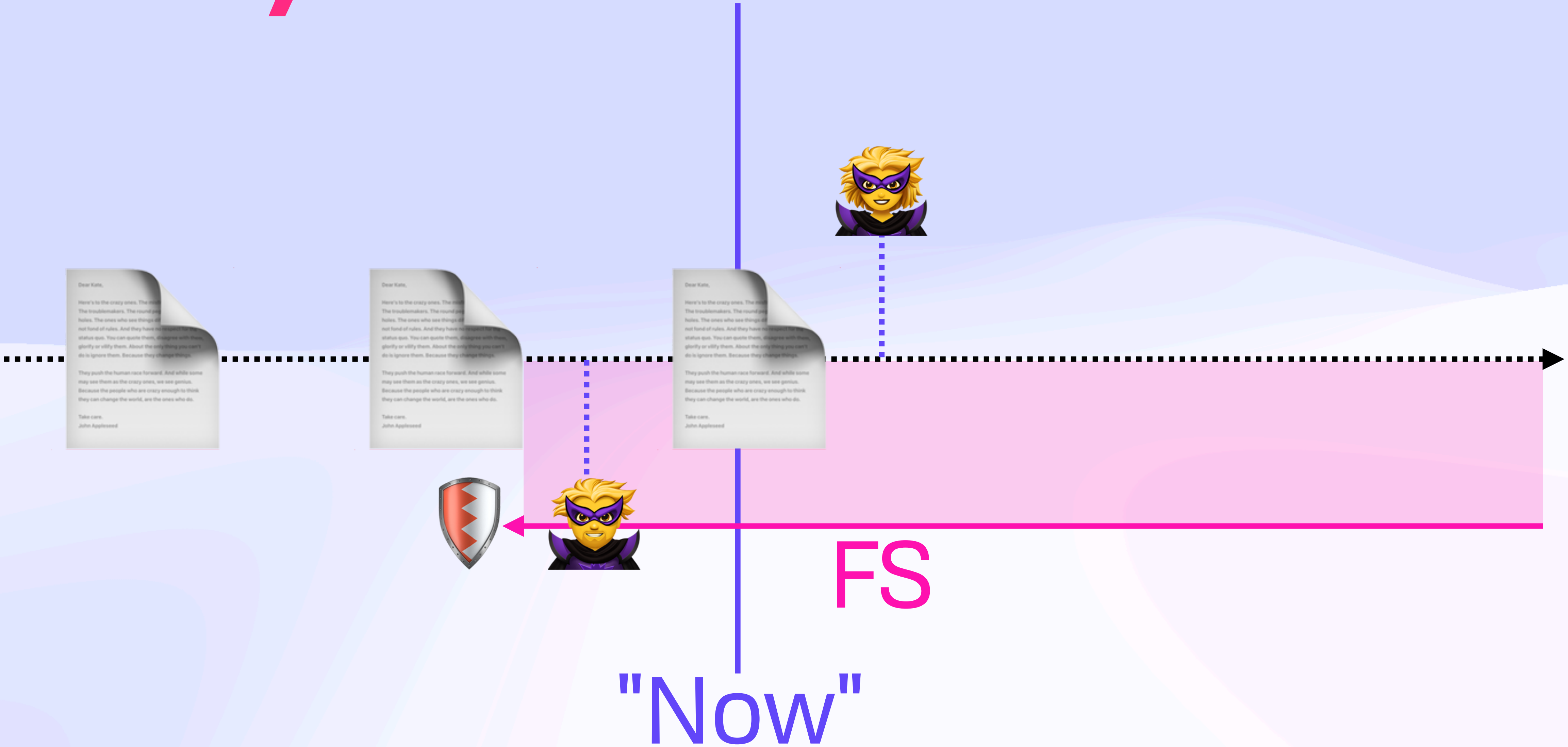
Self-Healing Concurrent Group Encryption

Security Over Time



Self-Healing Concurrent Group Encryption

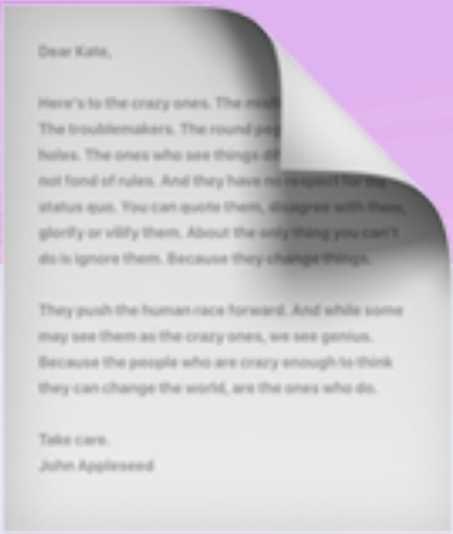
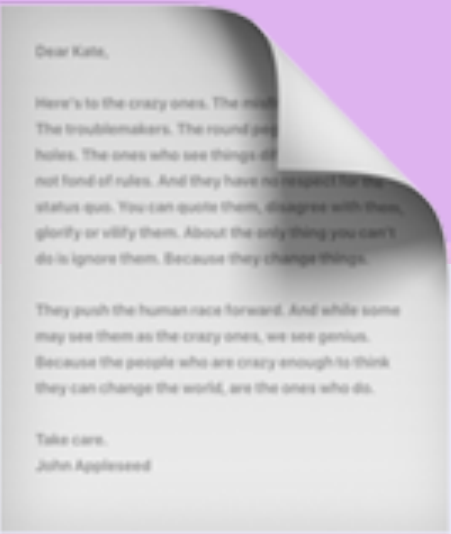
Security Over Time



Self-Healing Concurrent Group Encryption

Security Over Time

PCS



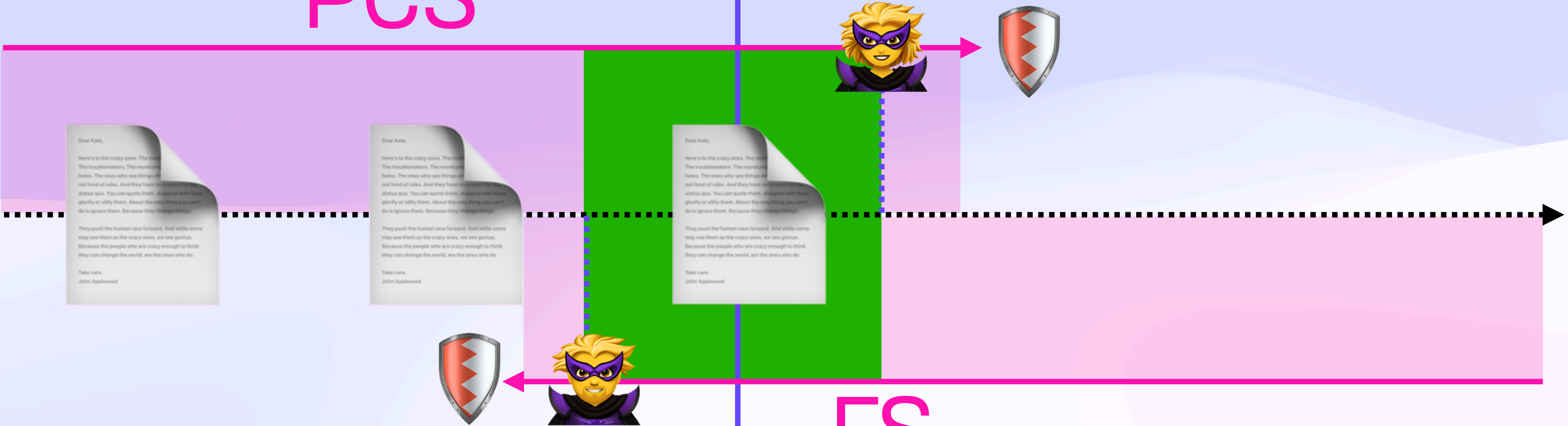
FS

"Now"

Self-Healing Concurrent Group Encryption

Security Over Time

PCS



FS

"Now"

Self-Healing Concurrent Group Encryption



Self-Healing Concurrent Group Encryption



Self-Healing Concurrent Group Encryption

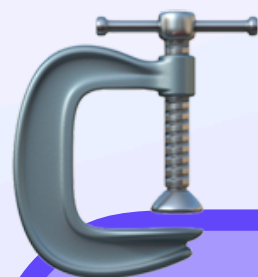


Here's to the crazy ones.
The mavericks. The rebels.
The troublemakers. The
round pegs in the
holes.
The ones who
different. They're not
fond of

Here's to the crazy ones.
The mavericks. The rebels.
The troublemakers. The
round pegs in the
holes.
The ones who
different. They're not
fond of

Here's to the crazy ones.
The mavericks. The rebels.
The troublemakers. The
round pegs in the
holes.
The ones who
different. They're not
fond of

Self-Healing Concurrent Group Encryption

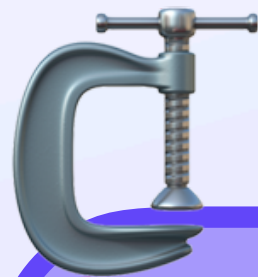


Here's to the crazy ones.
The mavericks. The rebels.
The troublemakers. The
round pegs in the
holes.
The ones who
different. They're not
fond of

Here's to the crazy ones.
The mavericks. The rebels.
The troublemakers. The
round pegs in the
holes.
The ones who
different. They're not
fond of

Here's to the crazy ones.
The mavericks. The rebels.
The troublemakers. The
round pegs in the
holes.
The ones who
different. They're not
fond of

Self-Healing Concurrent Group Encryption

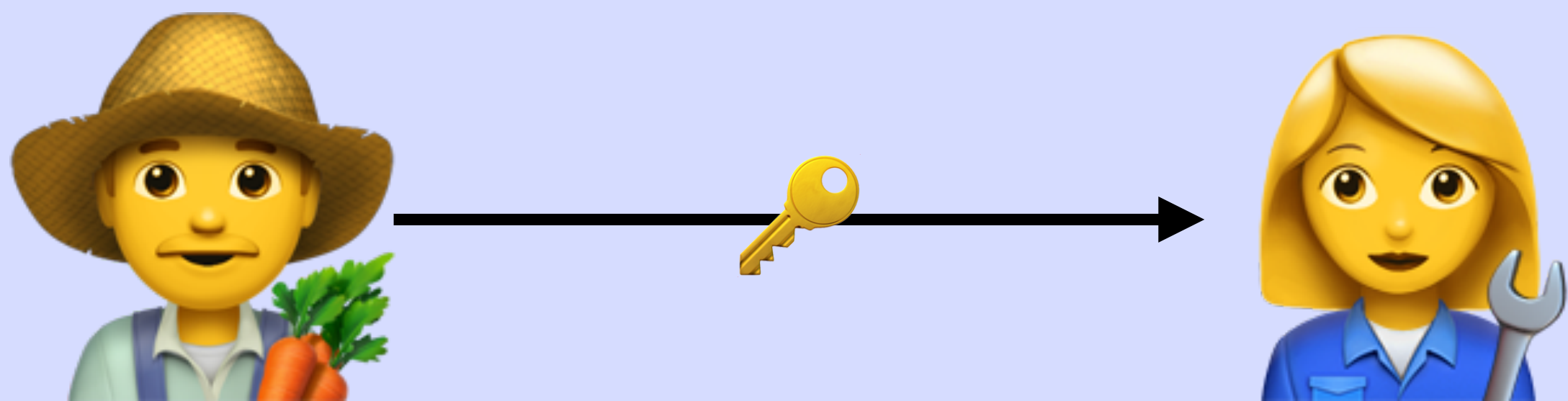


Here's to the crazy ones.
The mavericks. The rebels.
The troublemakers. The
round pegs in the
holes.
The ones who
different. They're not
fond of

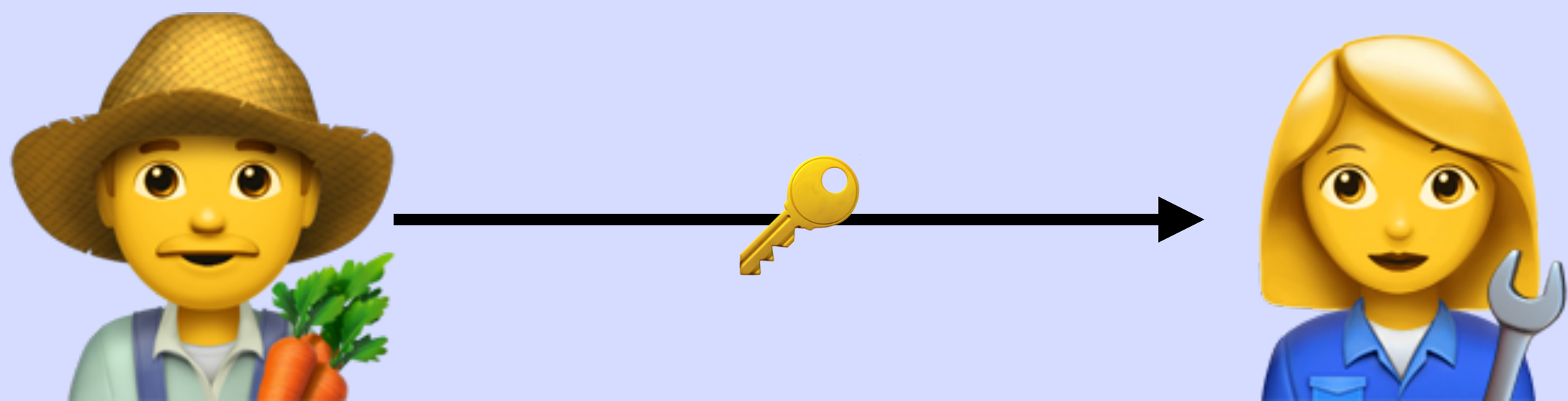
Here's to the crazy ones.
The mavericks. The rebels.
The troublemakers. The
round pegs in the
holes.
The ones who
different. They're not
fond of

Here's to the crazy ones.
The mavericks. The rebels.
The troublemakers. The
round pegs in the
holes.
The ones who
different. They're not
fond of

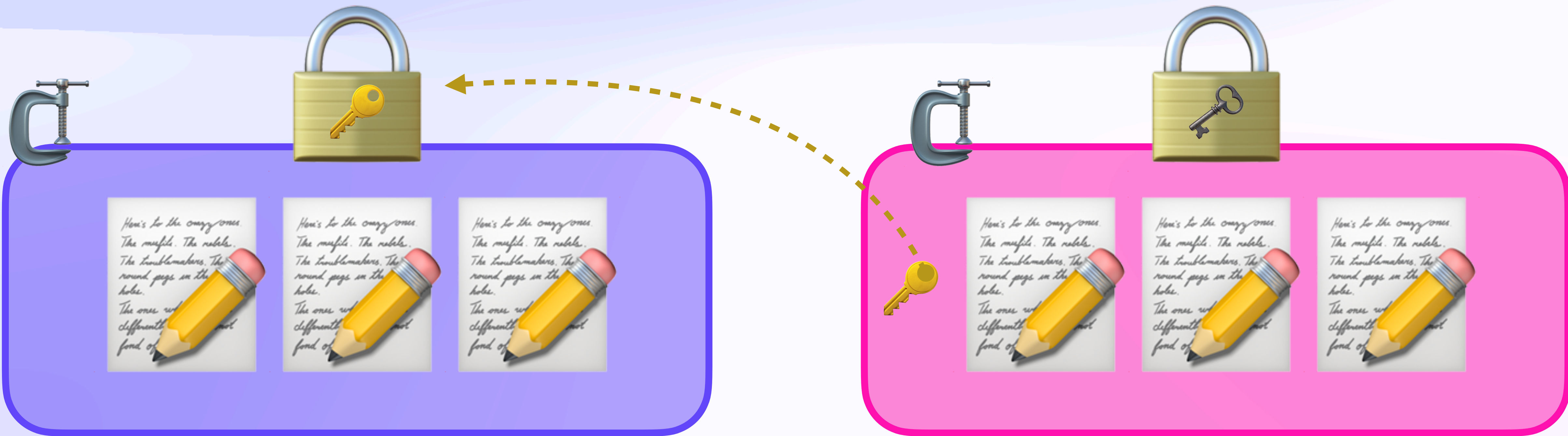
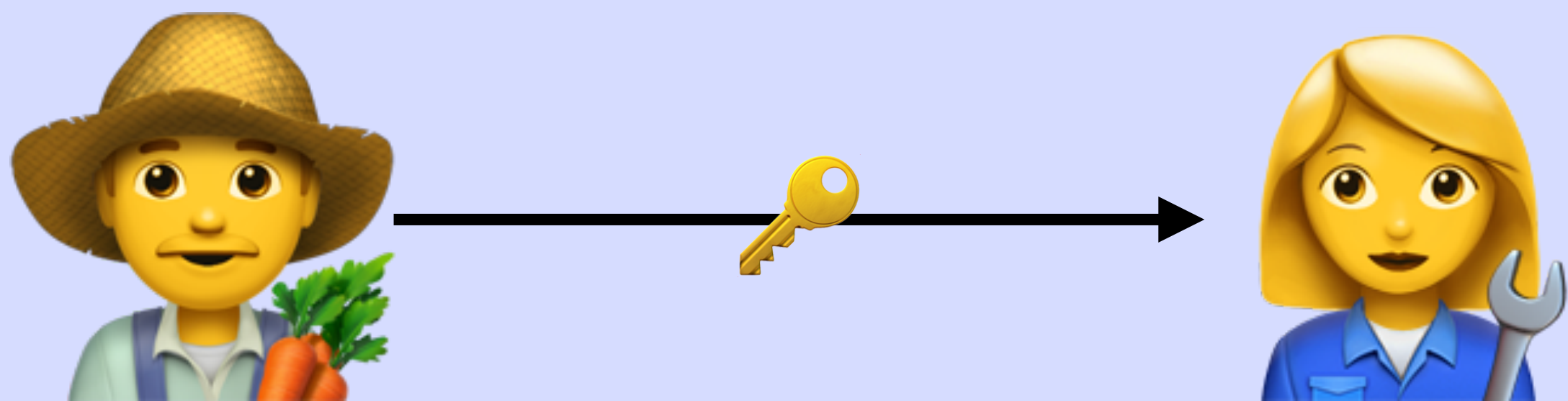
Self-Healing Concurrent Group Encryption



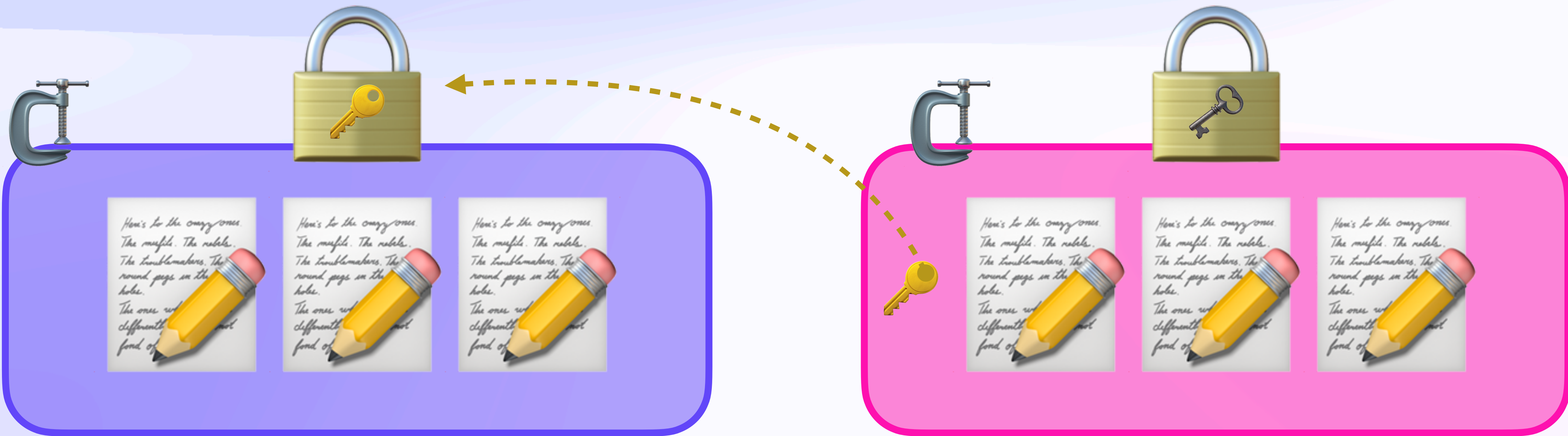
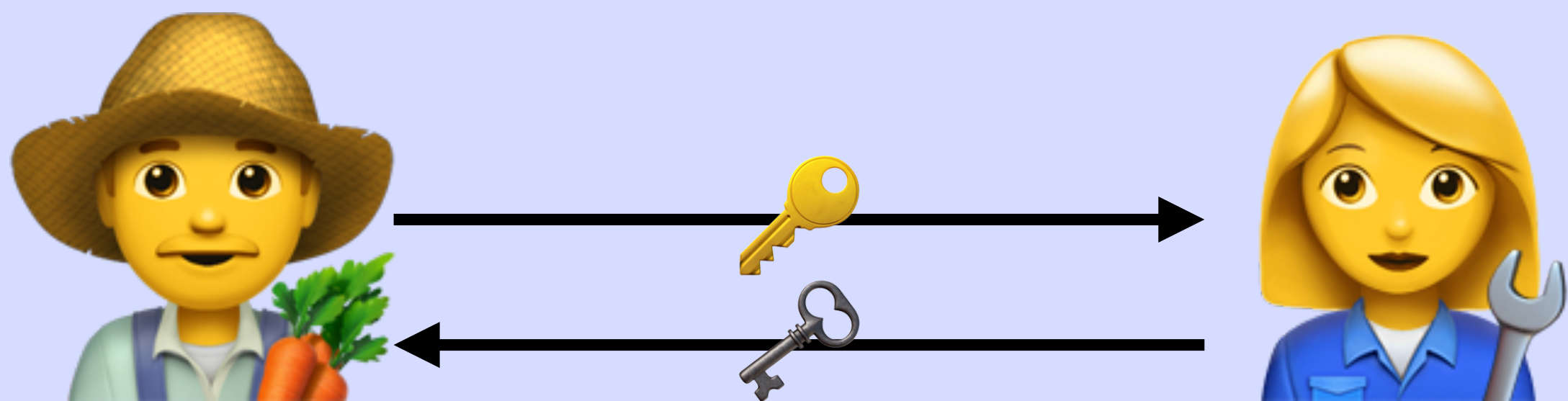
Self-Healing Concurrent Group Encryption



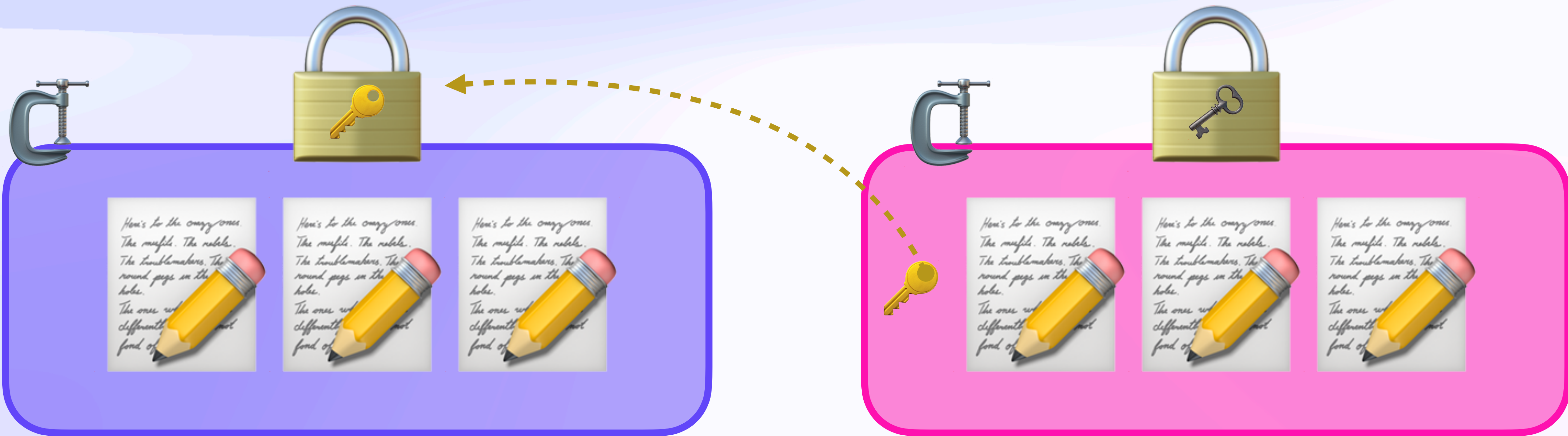
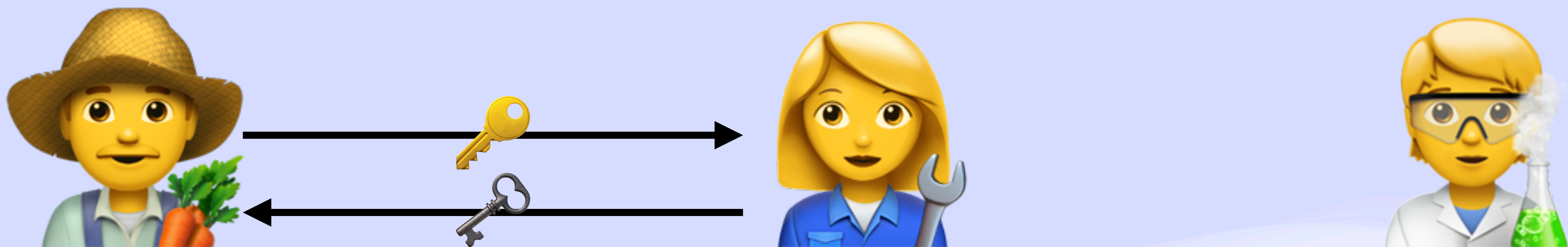
Self-Healing Concurrent Group Encryption



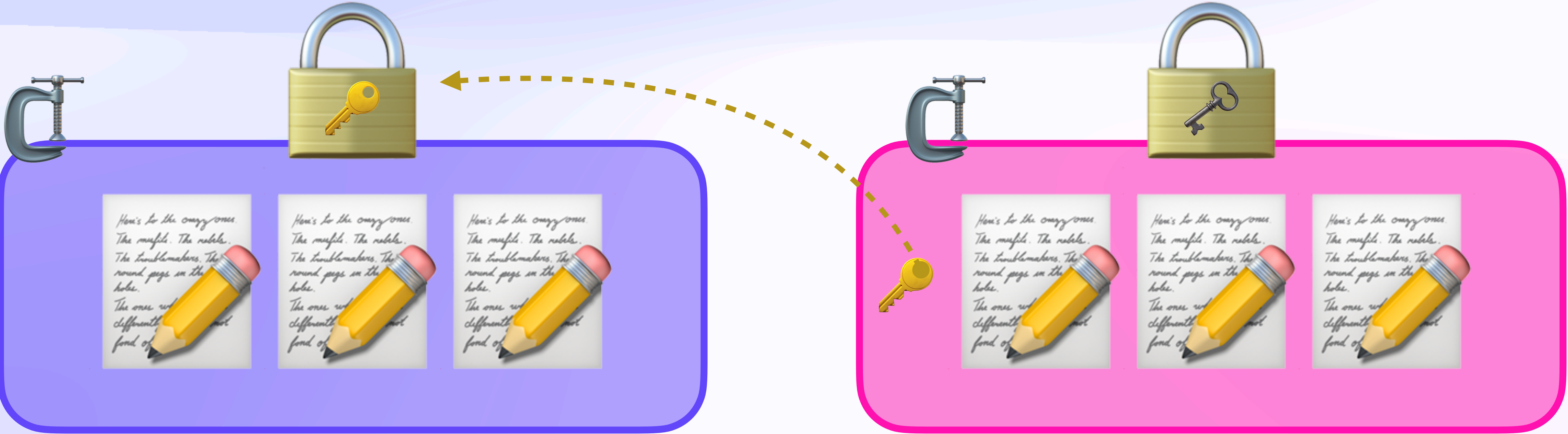
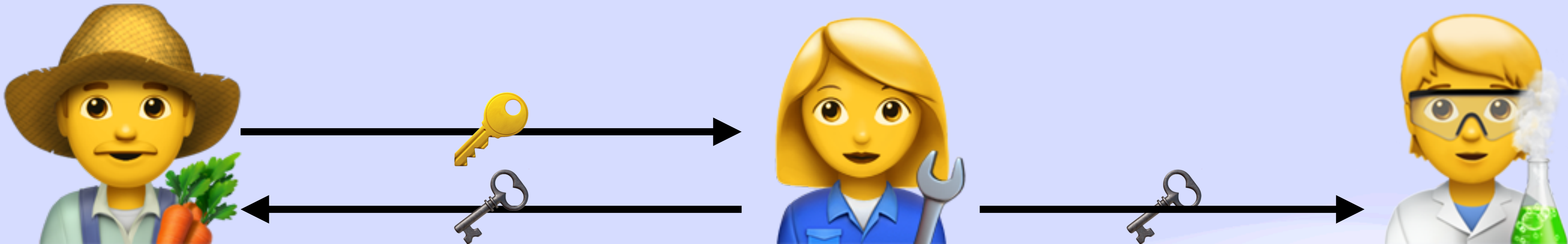
Self-Healing Concurrent Group Encryption



Self-Healing Concurrent Group Encryption

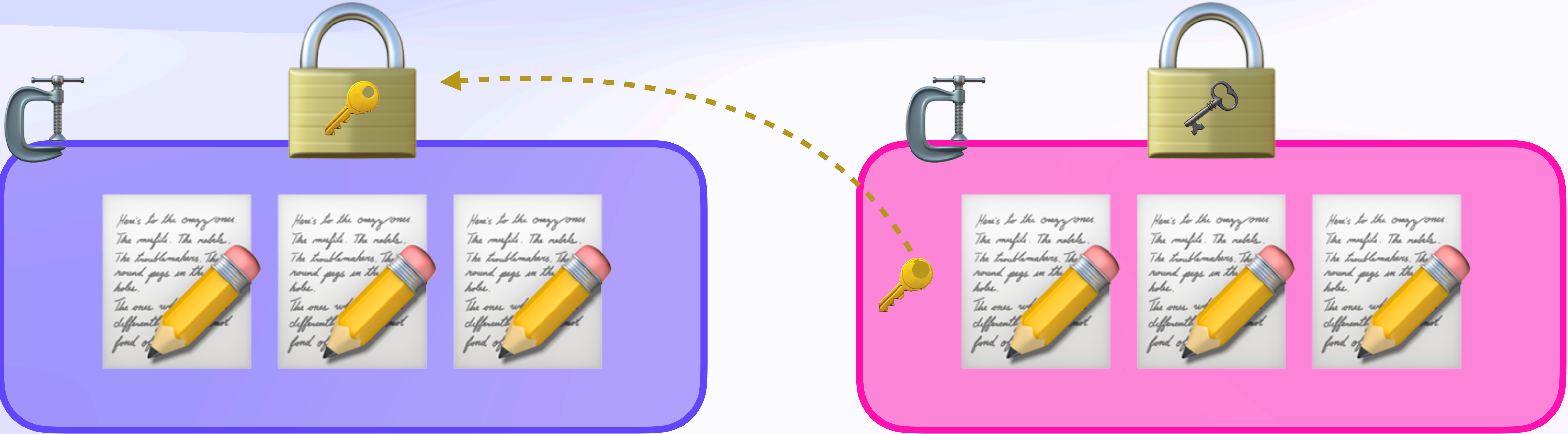
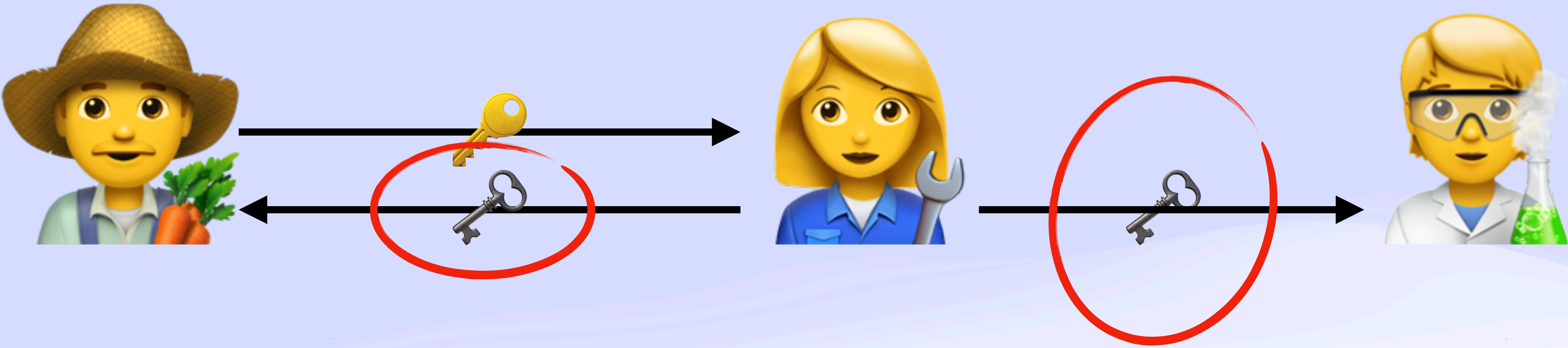


Self-Healing Concurrent Group Encryption

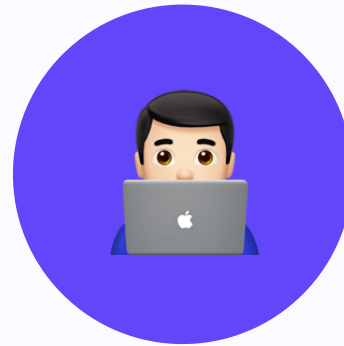
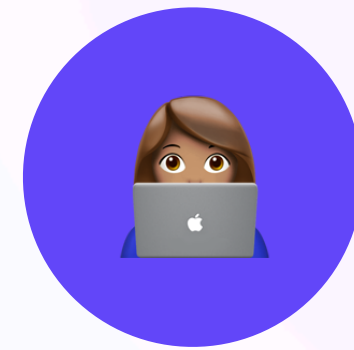


Self-Healing Concurrent Group Encryption

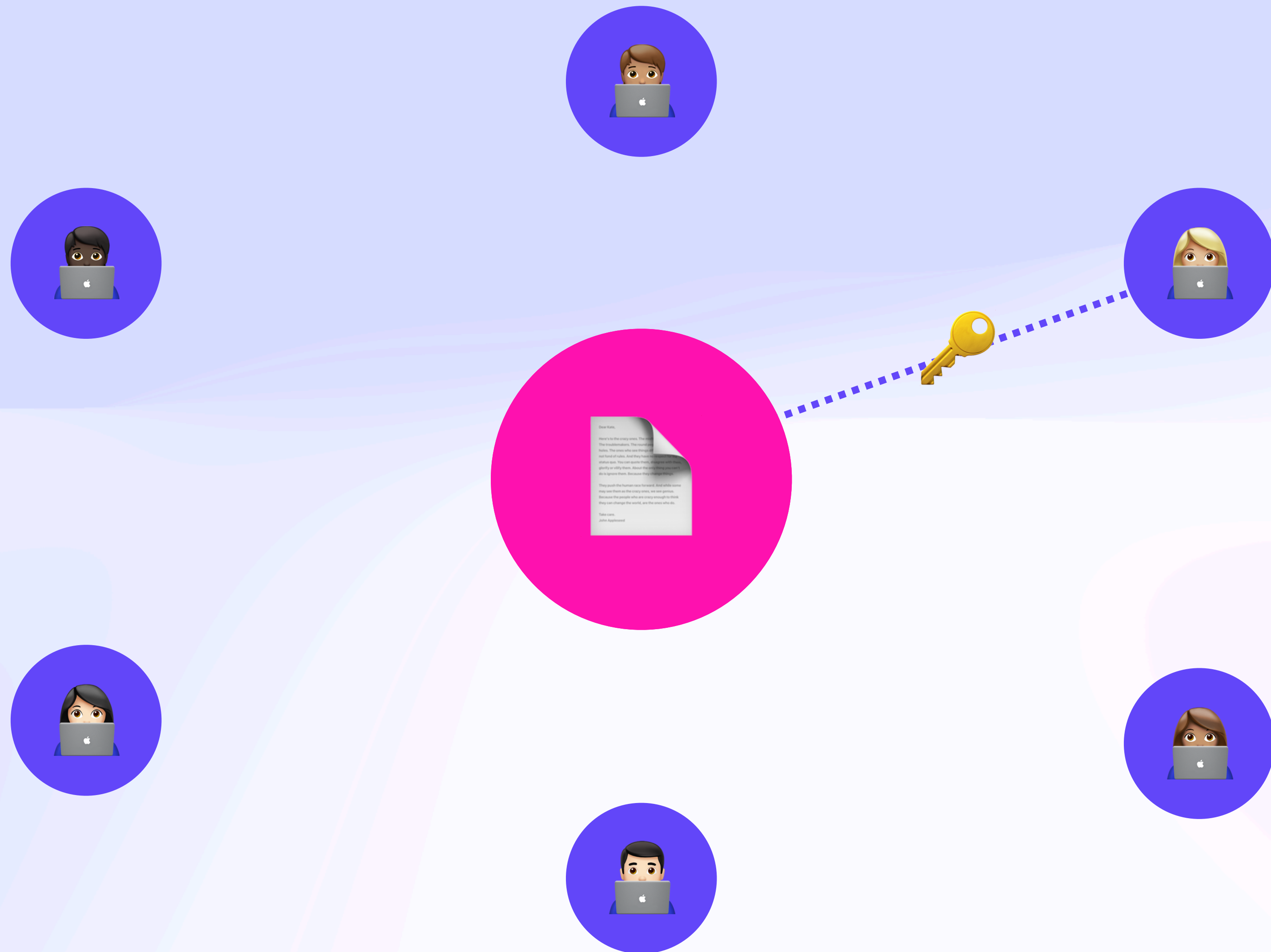
Well that's not going to scale to Wikipedia size



Naive Solution

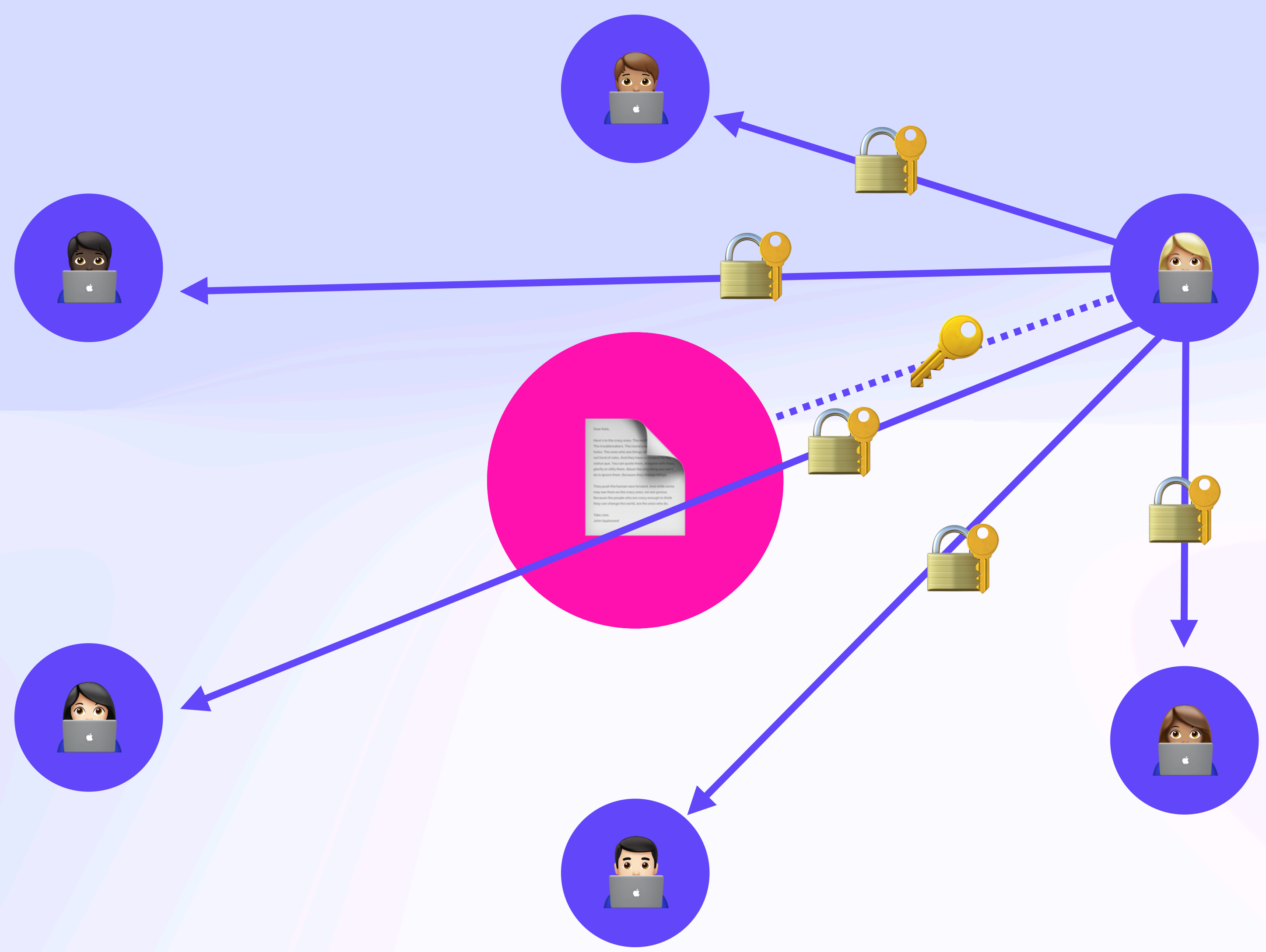


Naive Solution



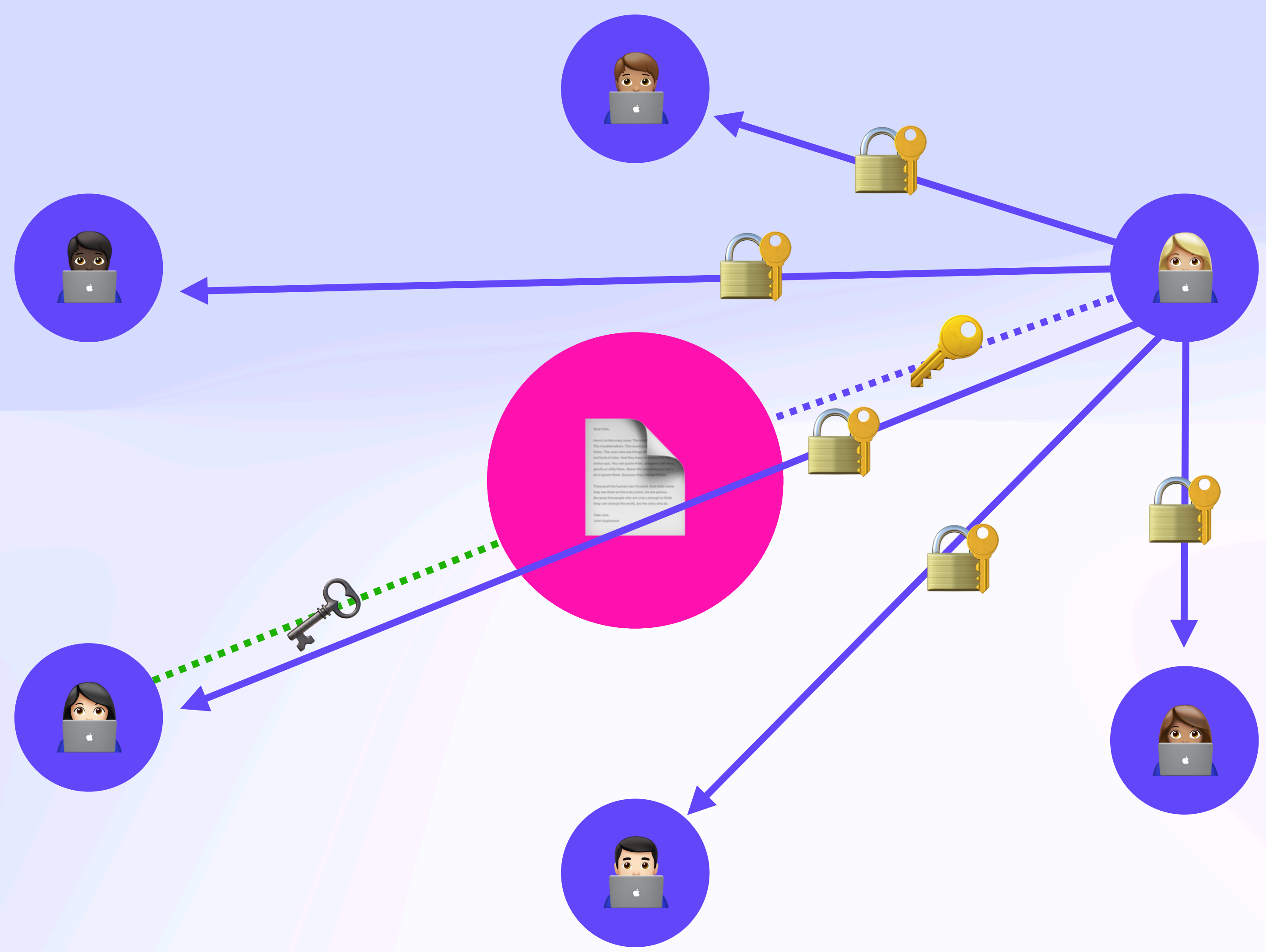
Self-Healing Concurrent Group Encryption

Naïve Solution



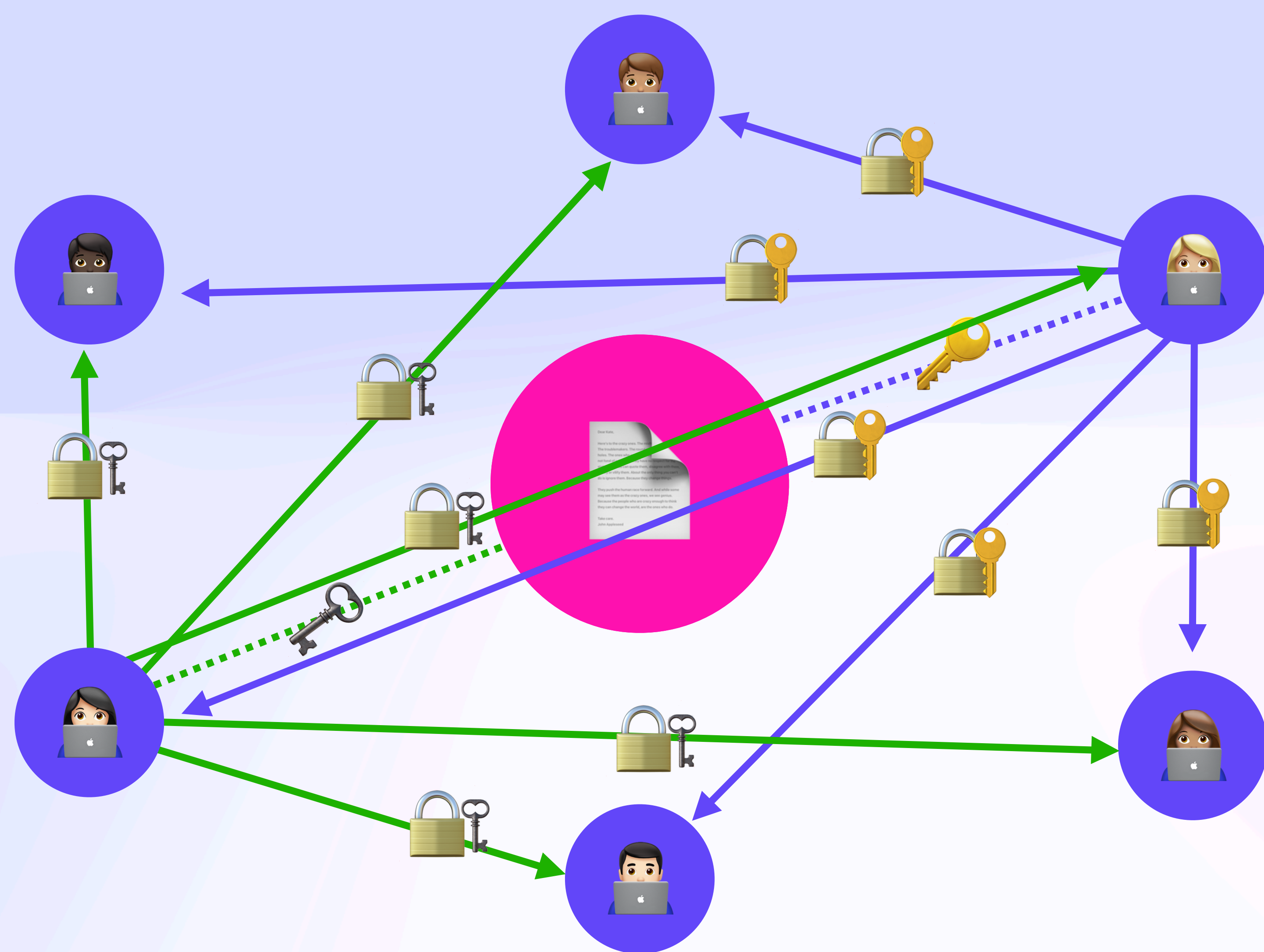
Self-Healing Concurrent Group Encryption

Naïve Solution



Self-Healing Concurrent Group Encryption

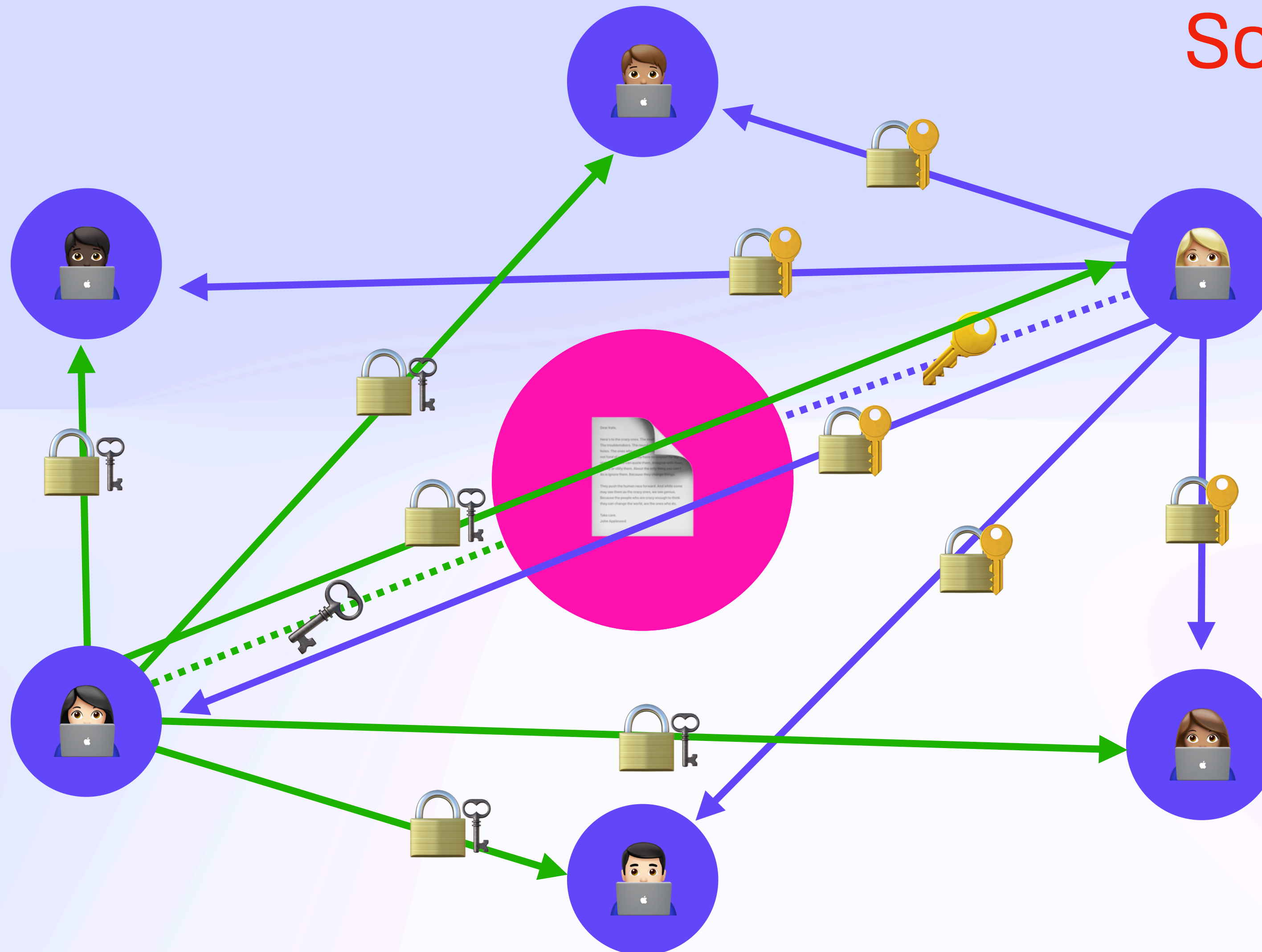
Naïve Solution



Self-Healing Concurrent Group Encryption

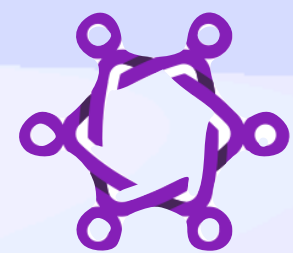
Naïve Solution

Scales in $O(n)$



Self-Healing Concurrent Group Encryption

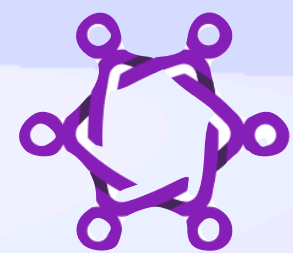
TreeKEM



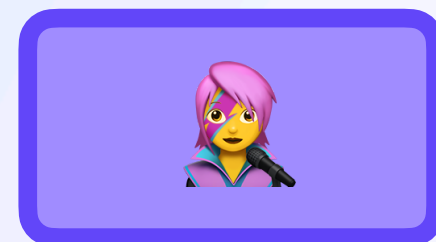
MLS

Self-Healing Concurrent Group Encryption

TreeKEM

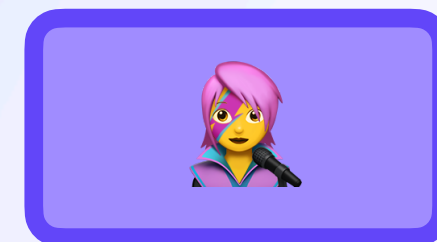


MLS



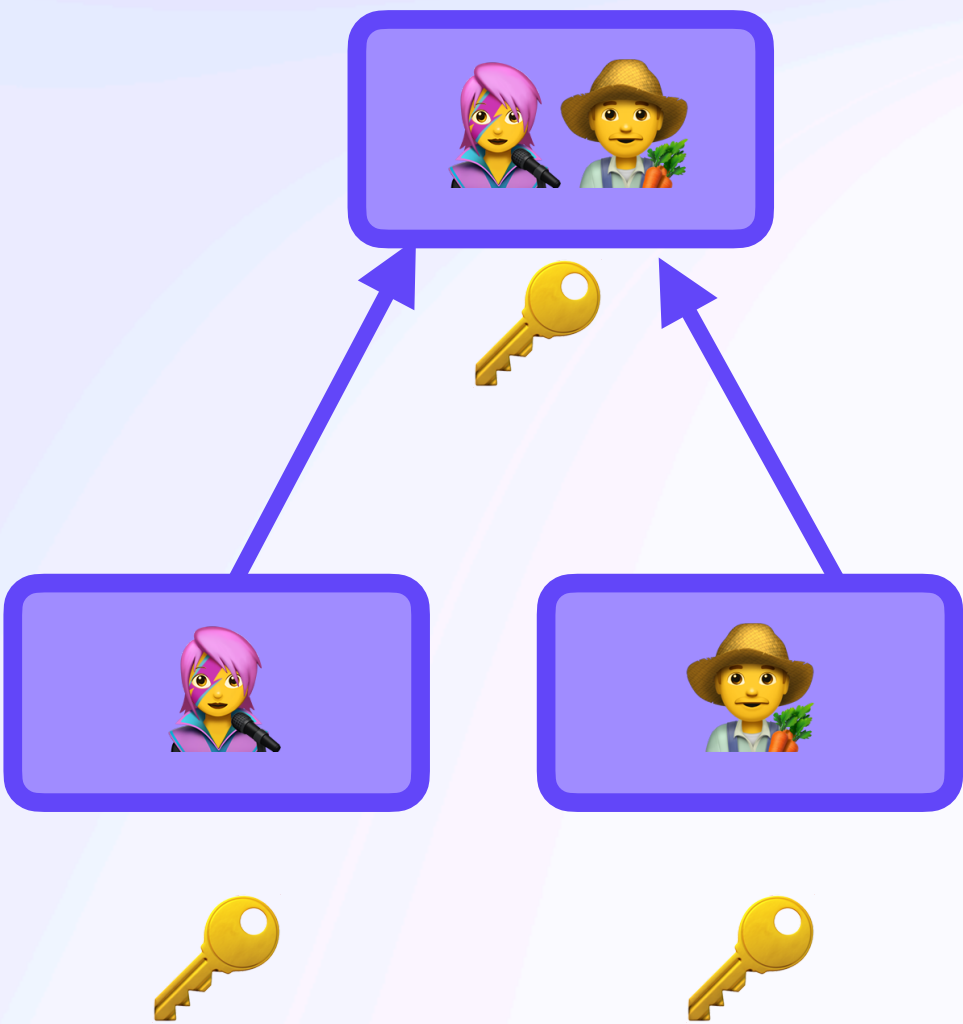
Self-Healing Concurrent Group Encryption

TreeKEM



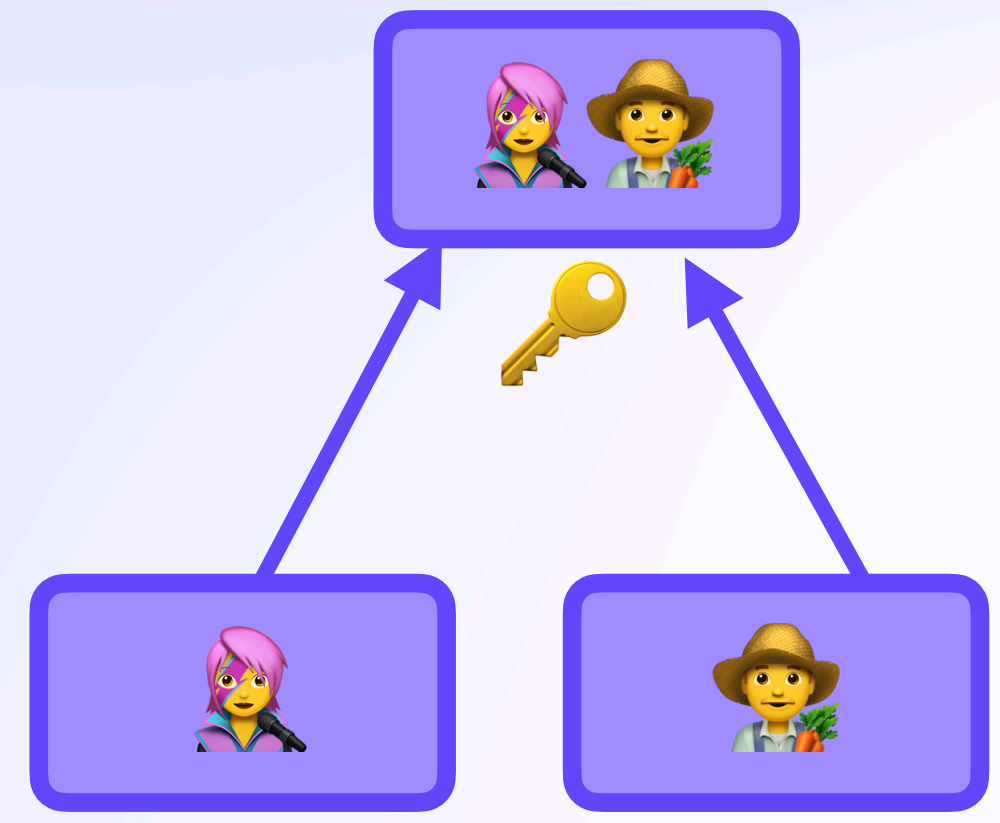
Self-Healing Concurrent Group Encryption

TreeKEM



Self-Healing Concurrent Group Encryption

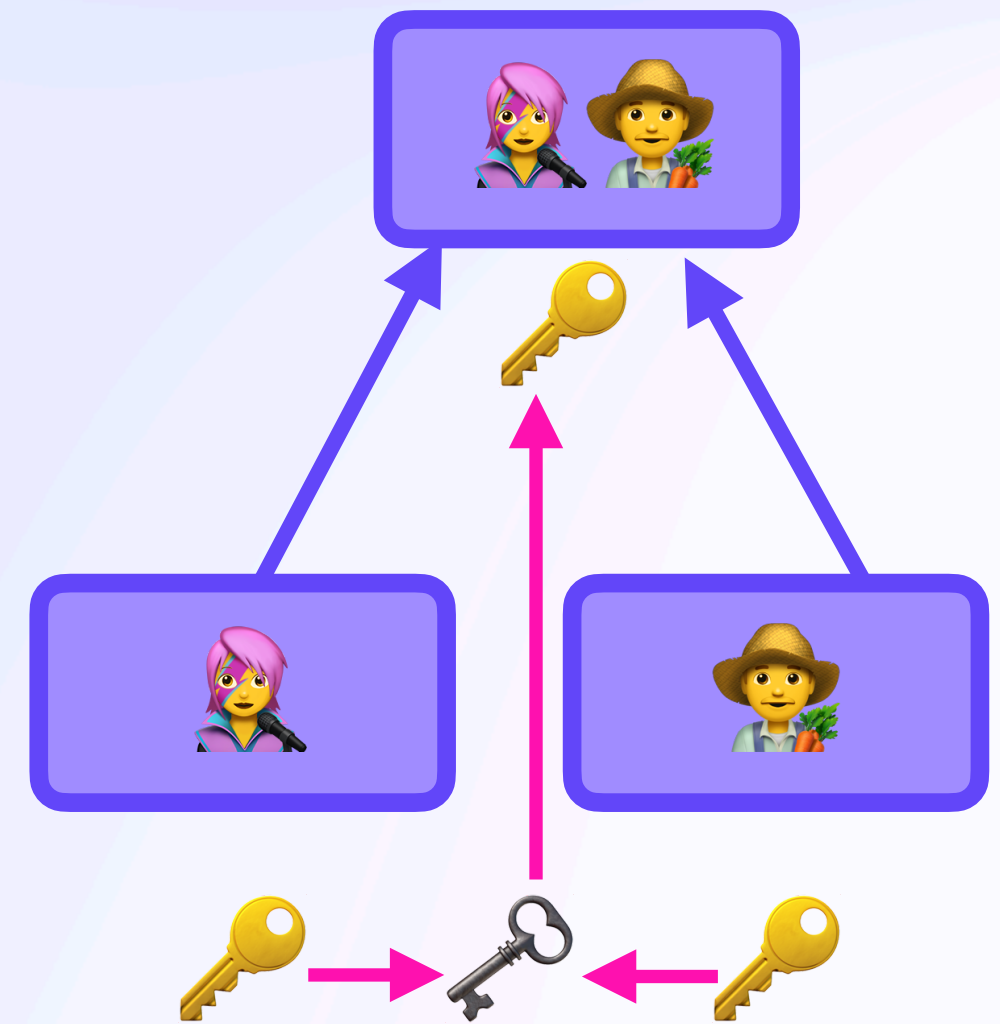
TreeKEM



Diffie Hellman

Self-Healing Concurrent Group Encryption

TreeKEM

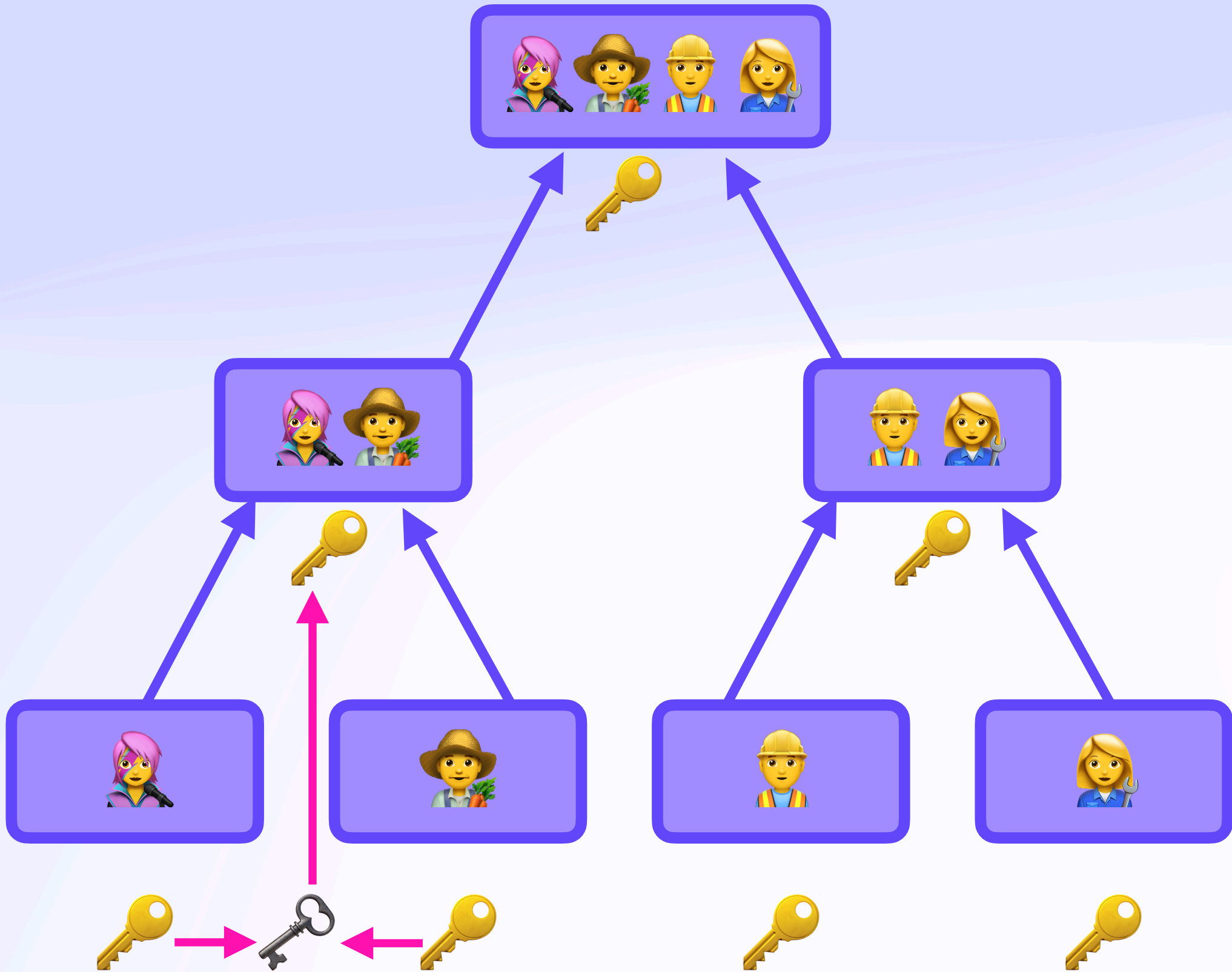


Diffie Hellman

Self-Healing Concurrent Group Encryption

TreeKEM

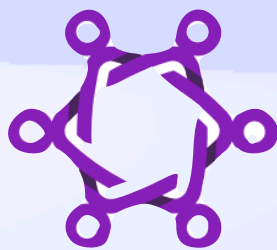
 **MLS**



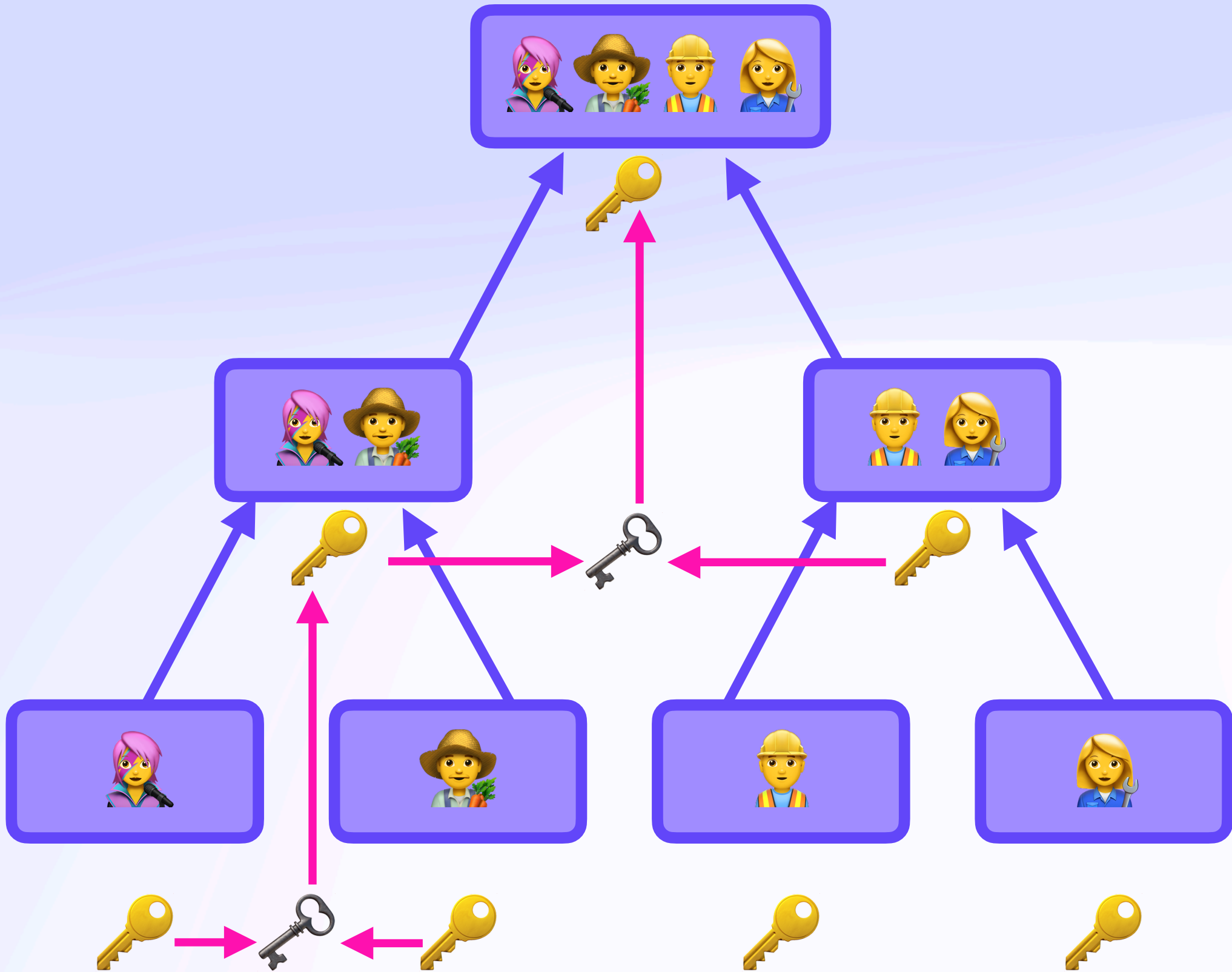
Diffie Hellman

Self-Healing Concurrent Group Encryption

TreeKEM



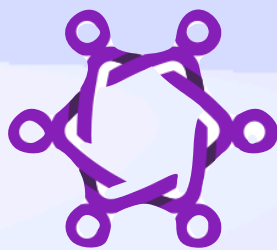
MLS



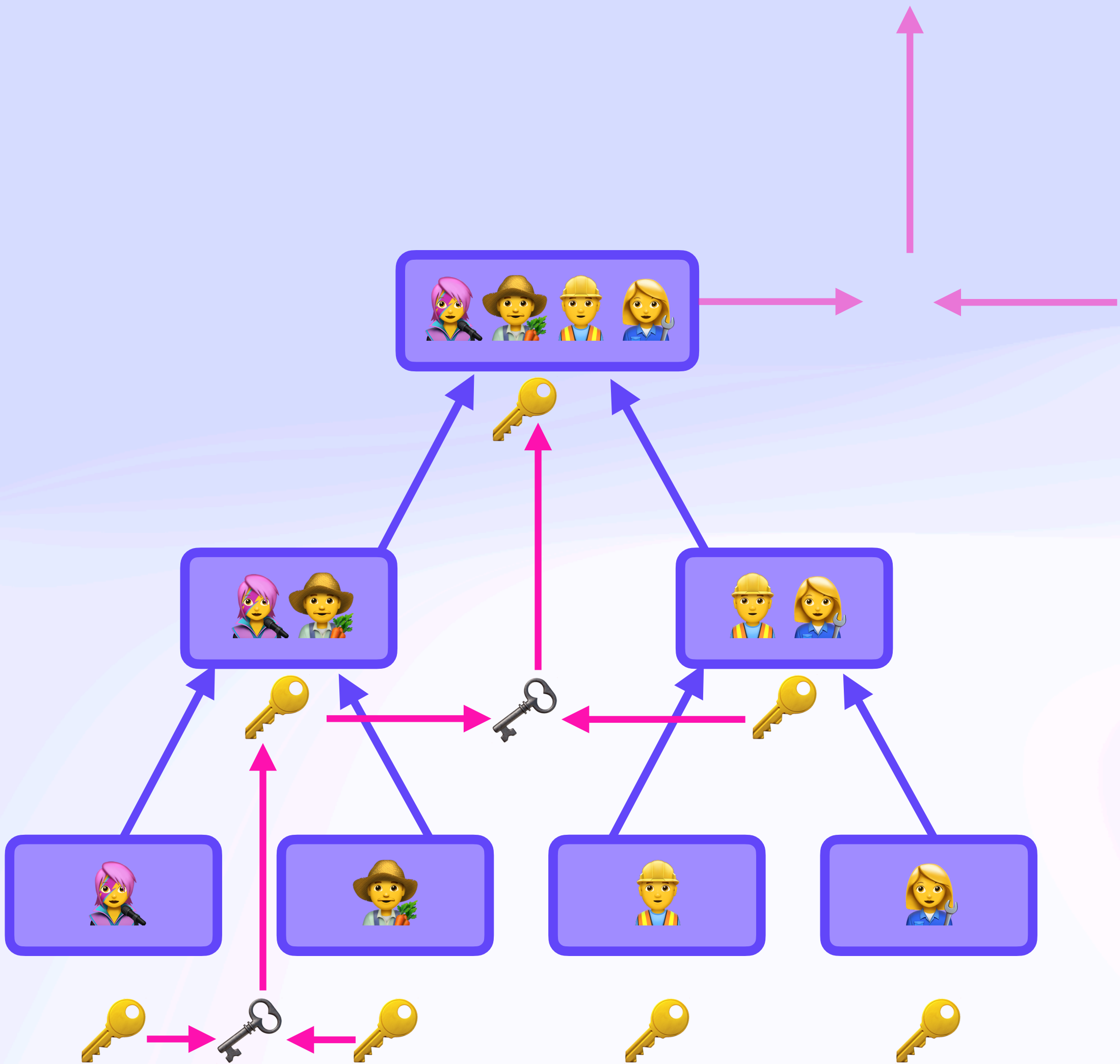
Diffie Hellman

Self-Healing Concurrent Group Encryption

TreeKEM



MLS

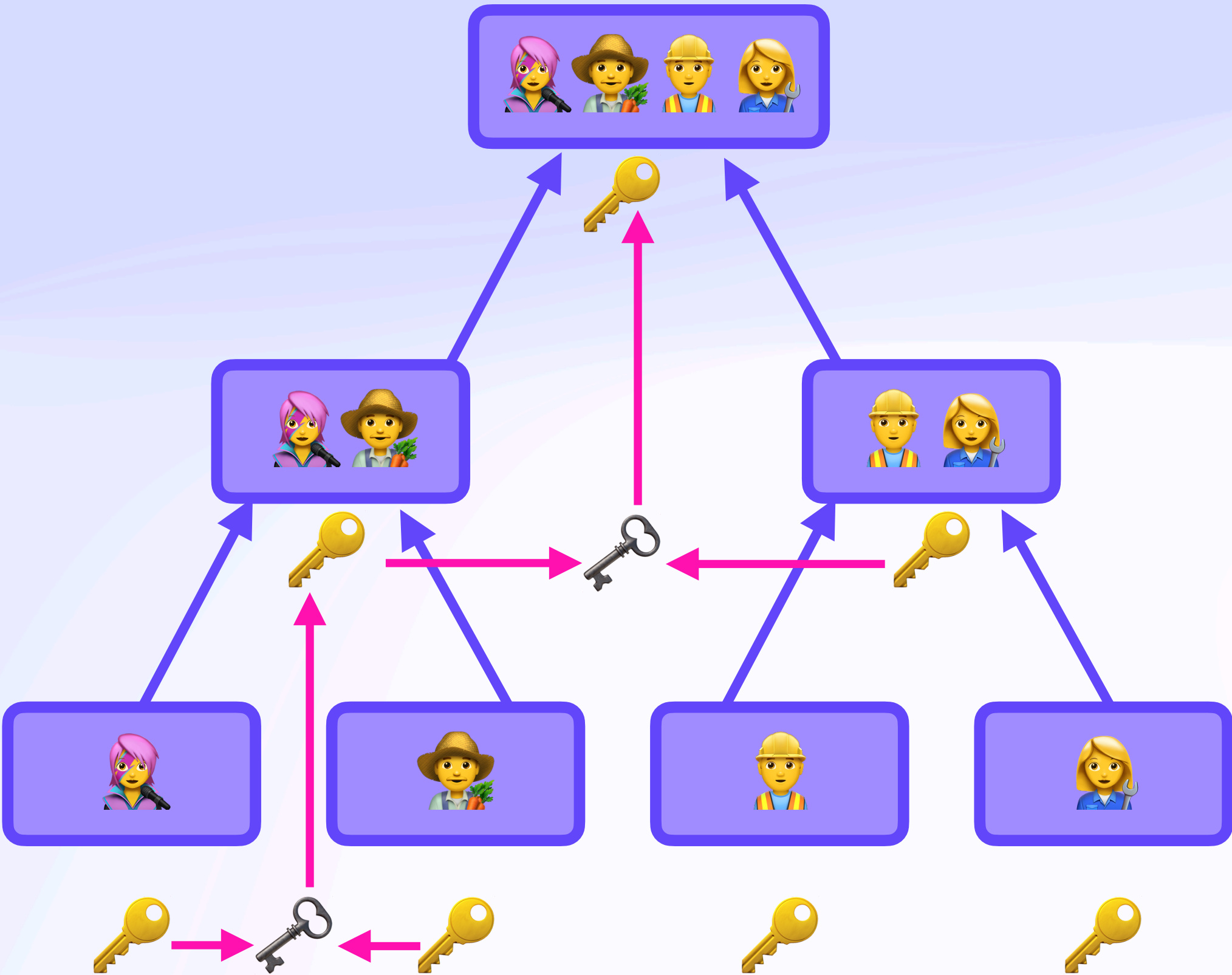


Diffie Hellman

Self-Healing Concurrent Group Encryption

TreeKEM

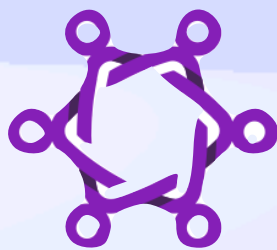
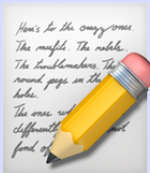
 **MLS**



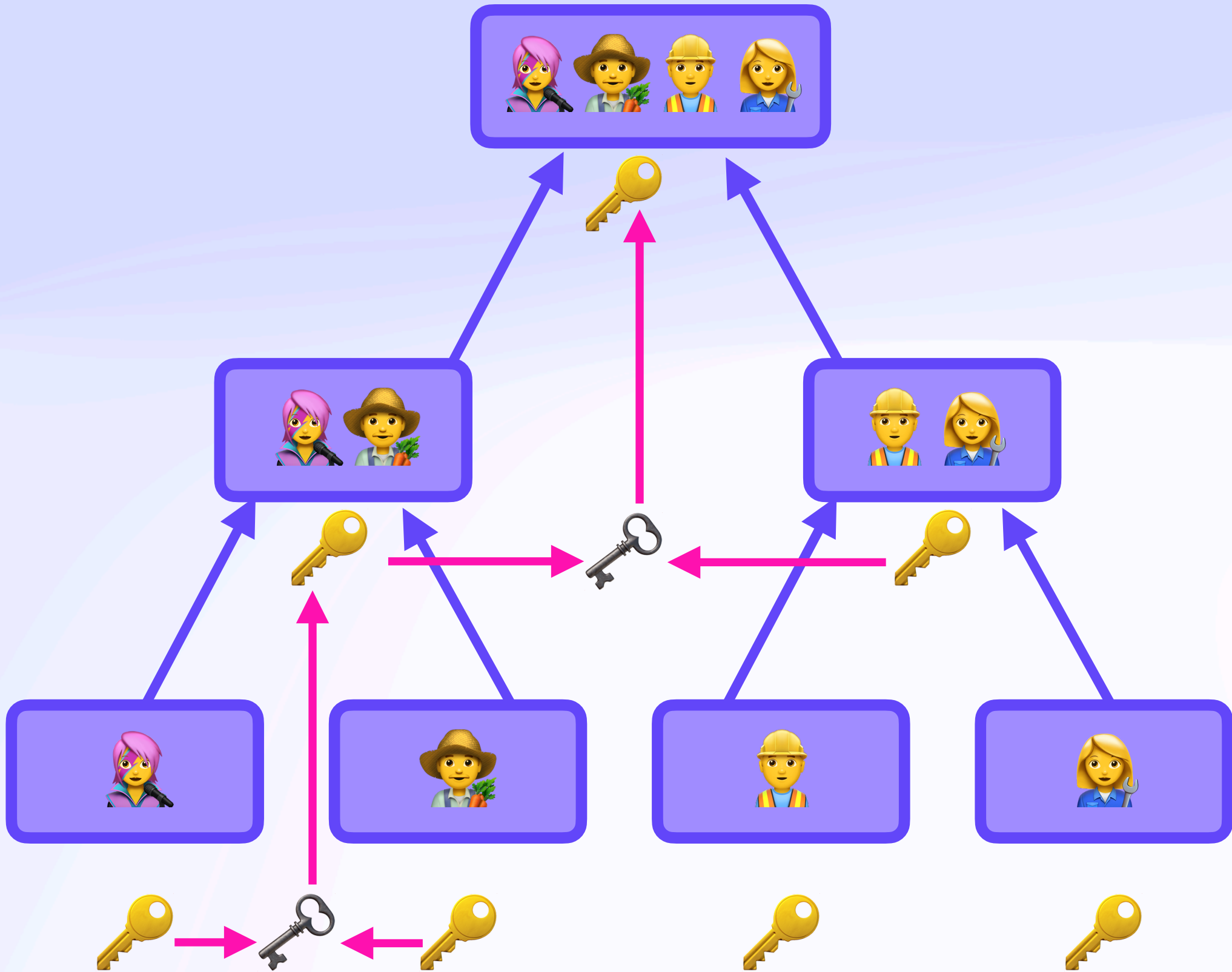
Diffie Hellman

Self-Healing Concurrent Group Encryption

TreeKEM

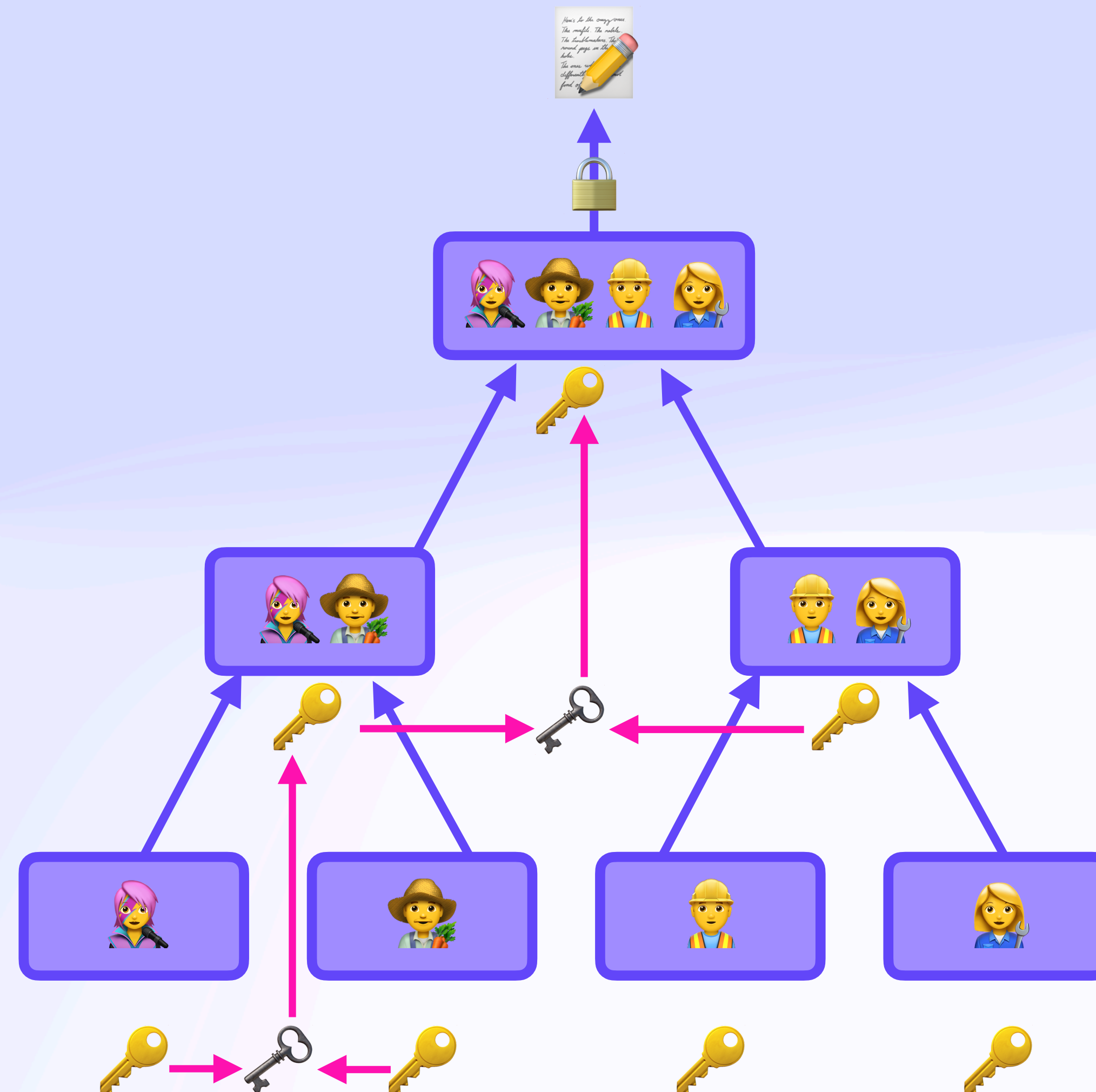


MLS



Diffie Hellman

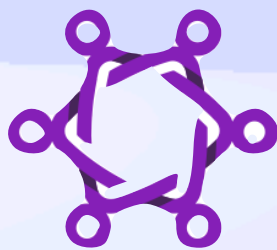
TreeKEM



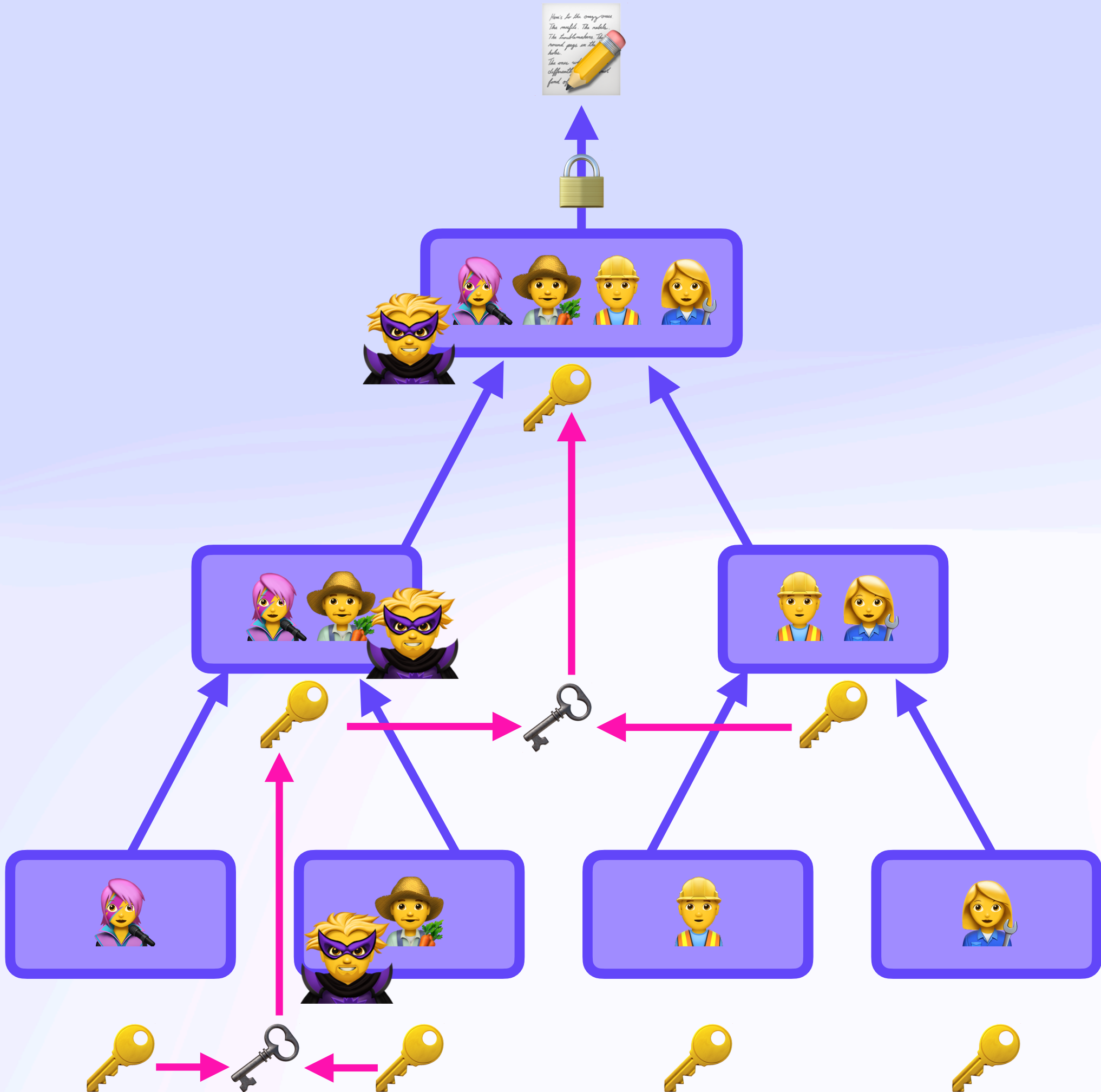
Diffie Hellman

Self-Healing Concurrent Group Encryption

TreeKEM



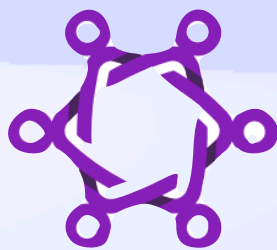
MLS



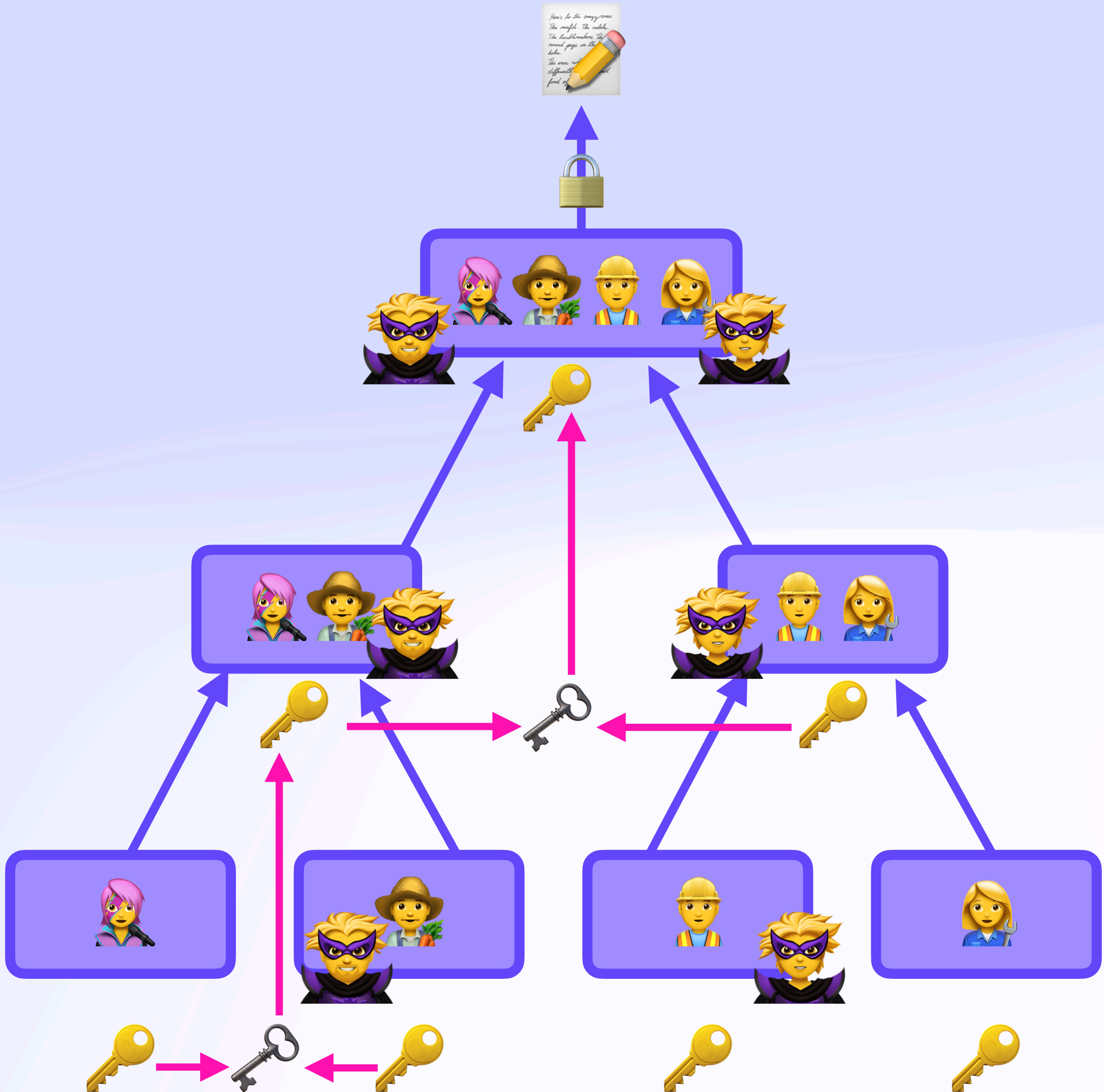
Diffie Hellman

Self-Healing Concurrent Group Encryption

TreeKEM



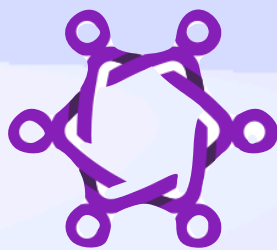
MLS



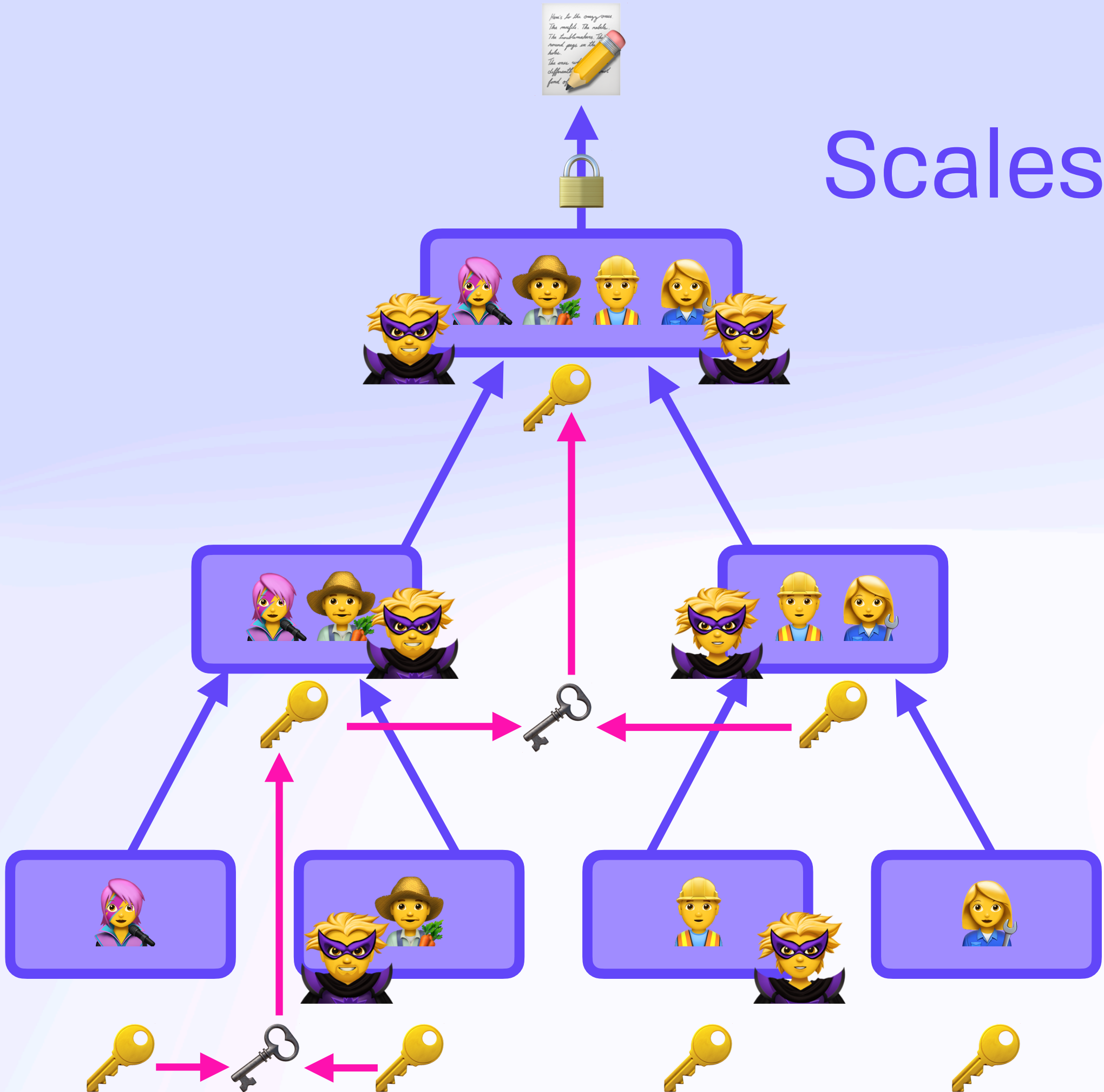
Diffie Hellman

Self-Healing Concurrent Group Encryption

TreeKEM



MLS



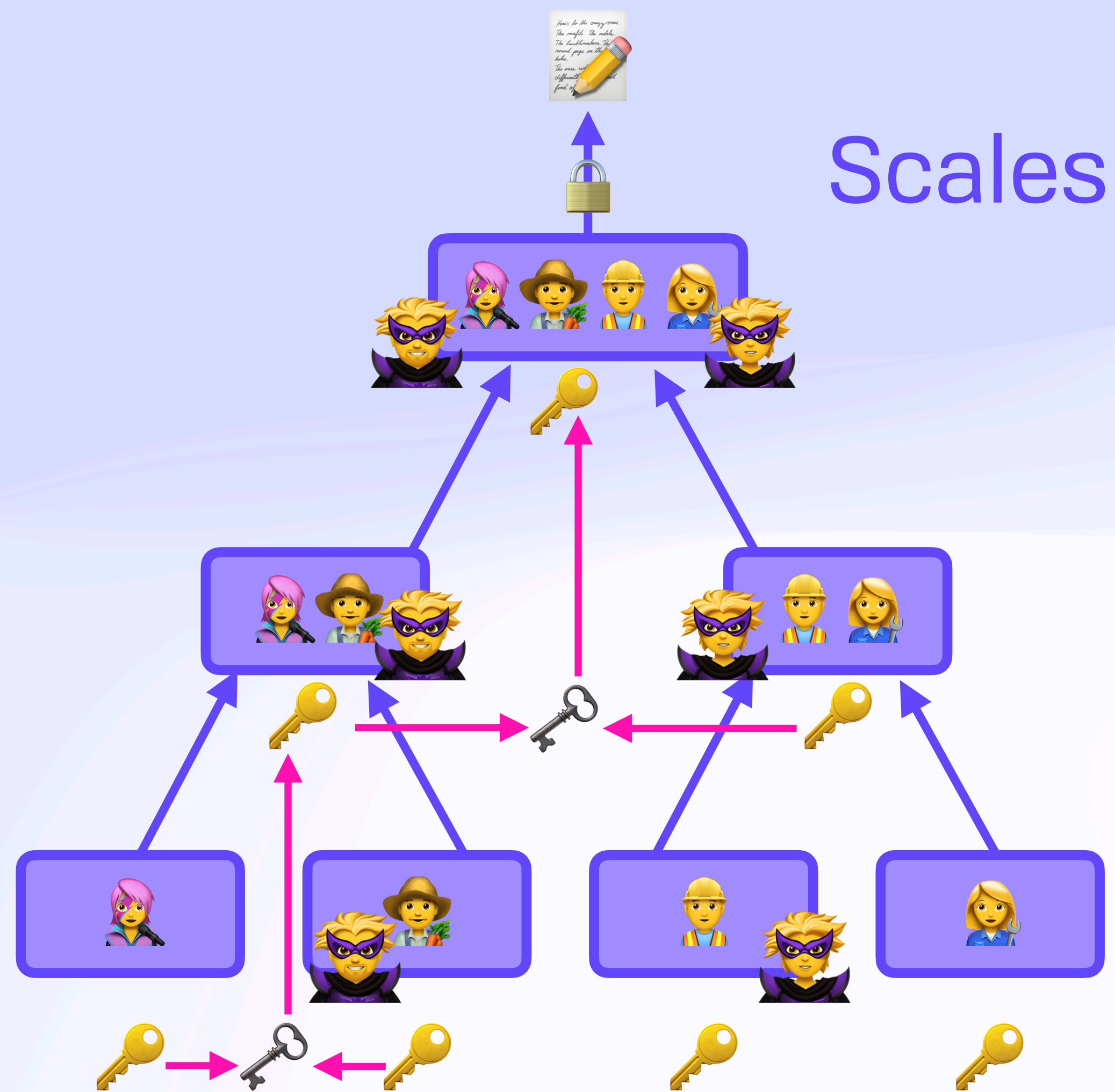
Scales in $O(\log(n))$ 🎉

Diffie Hellman

Self-Healing Concurrent Group Encryption

TreeKEM

 **MLS**

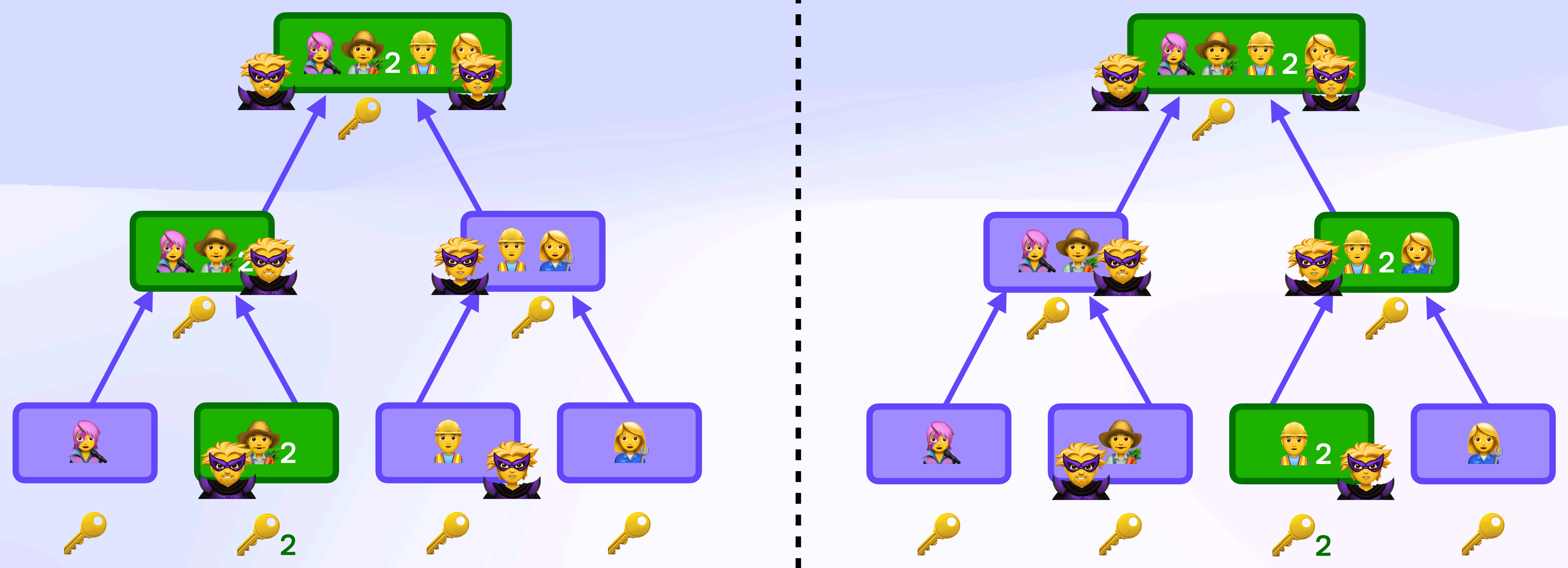


Scales in $O(\log(n))$ 🎉
...but...

Diffie Hellman

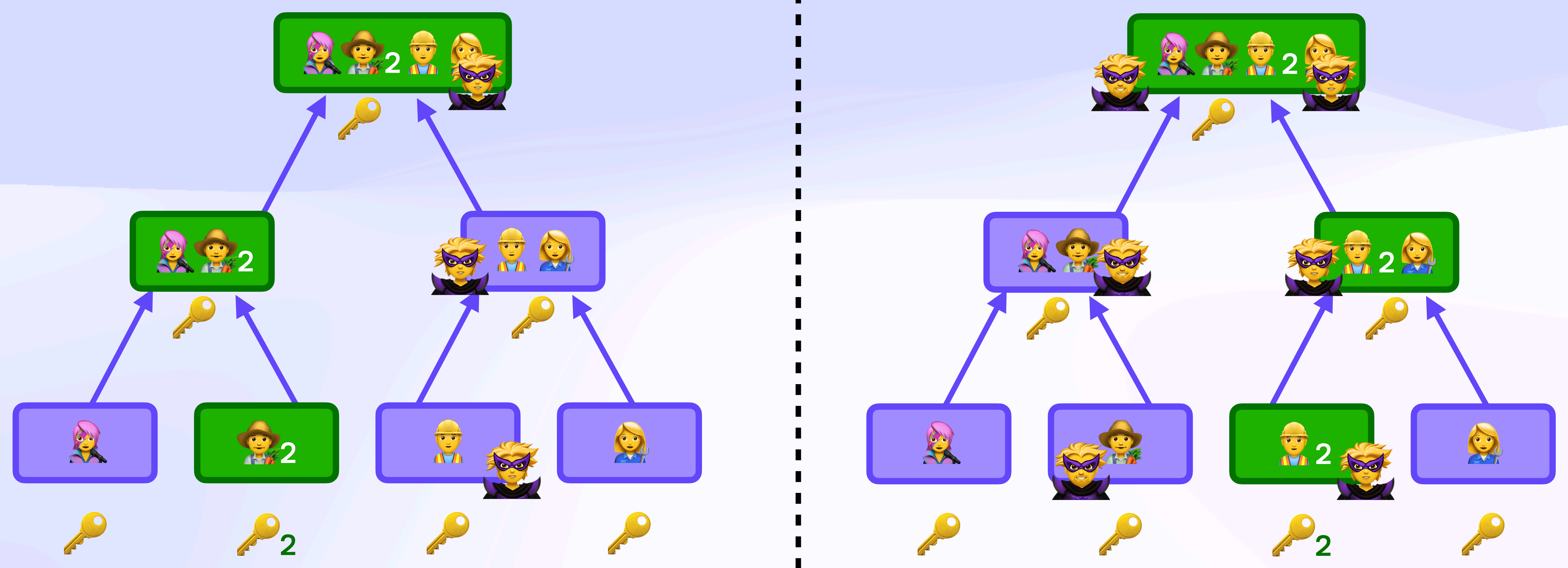
Self-Healing Concurrent Group Encryption

Concurrency Problem!



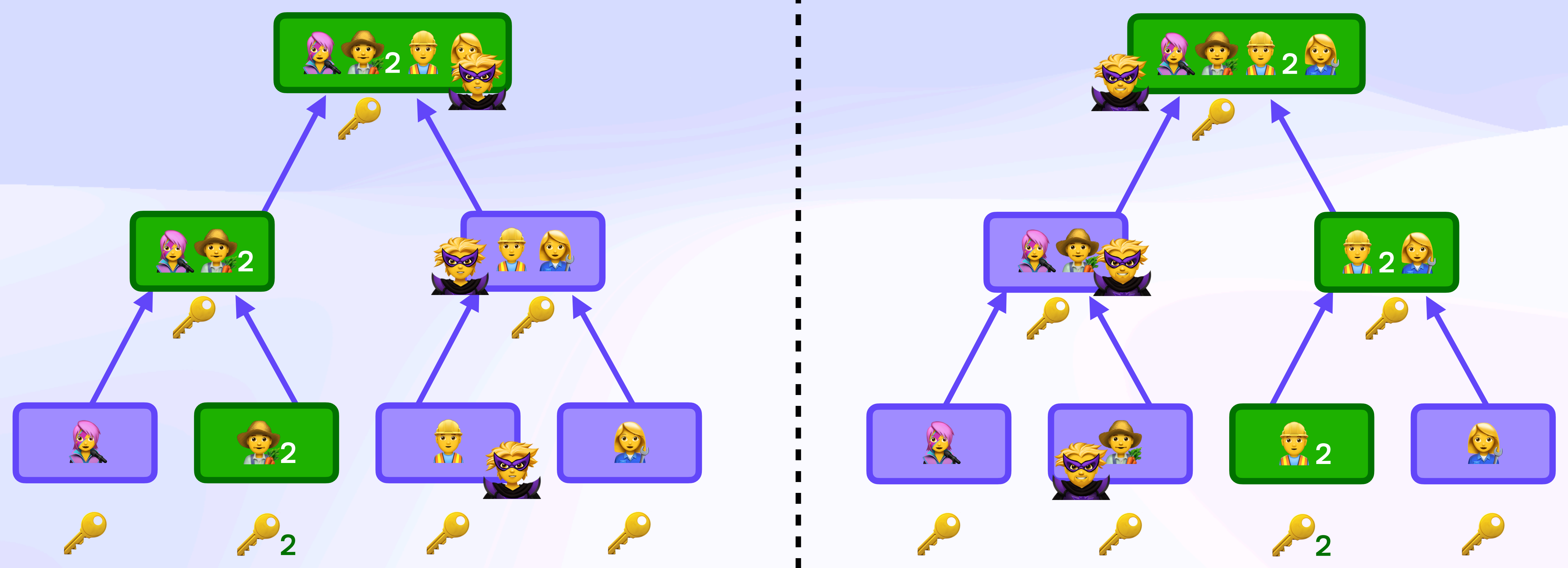
Self-Healing Concurrent Group Encryption

Concurrency Problem!



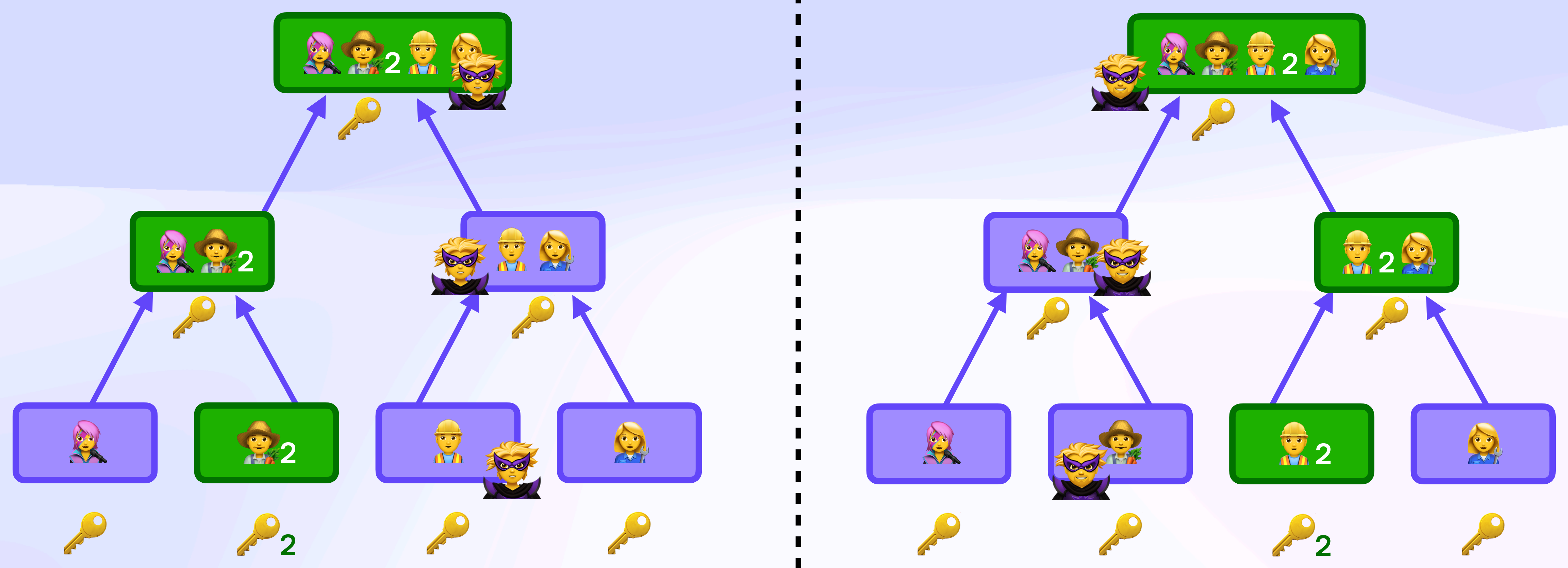
Self-Healing Concurrent Group Encryption

Concurrency Problem!



Self-Healing Concurrent Group Encryption

Concurrency Problem!



Self-Healing Concurrent Group Encryption

Middle Ground



Self-Healing Concurrent Group Encryption

Middle Ground



Self-Healing Concurrent Group Encryption

Middle Ground



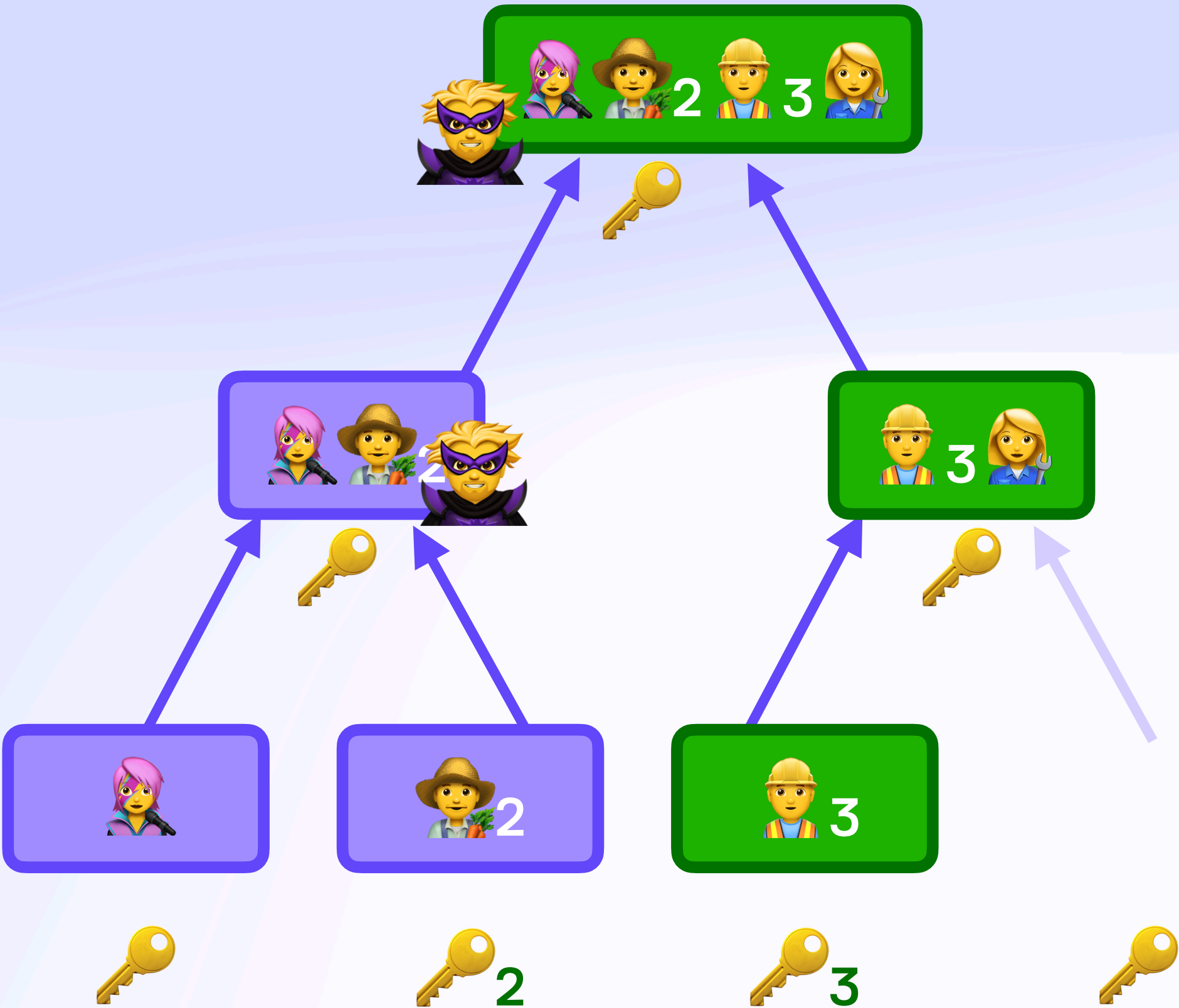
Self-Healing Concurrent Group Encryption

Middle Ground



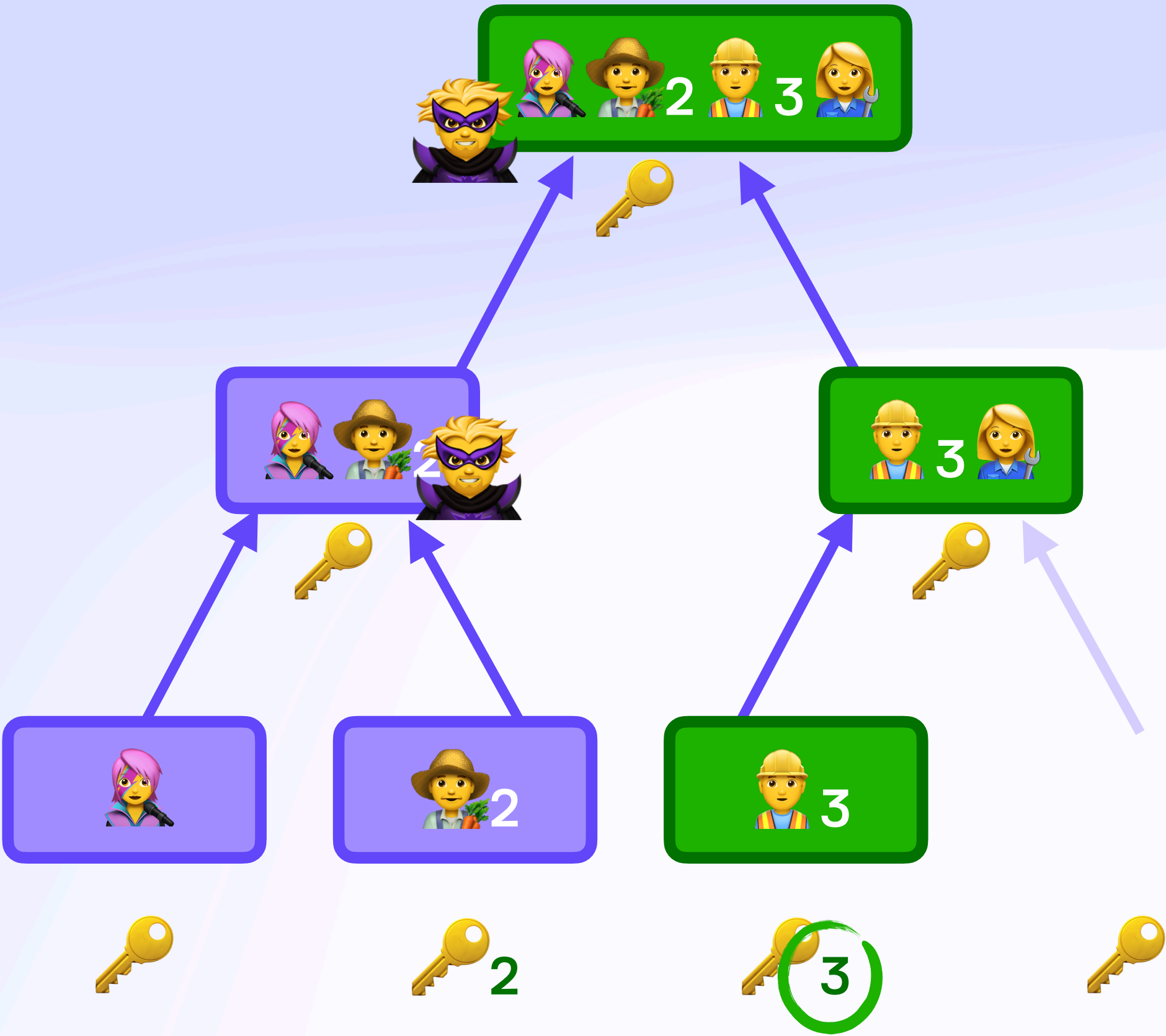
Self-Healing Concurrent Group Encryption

BeeKEM



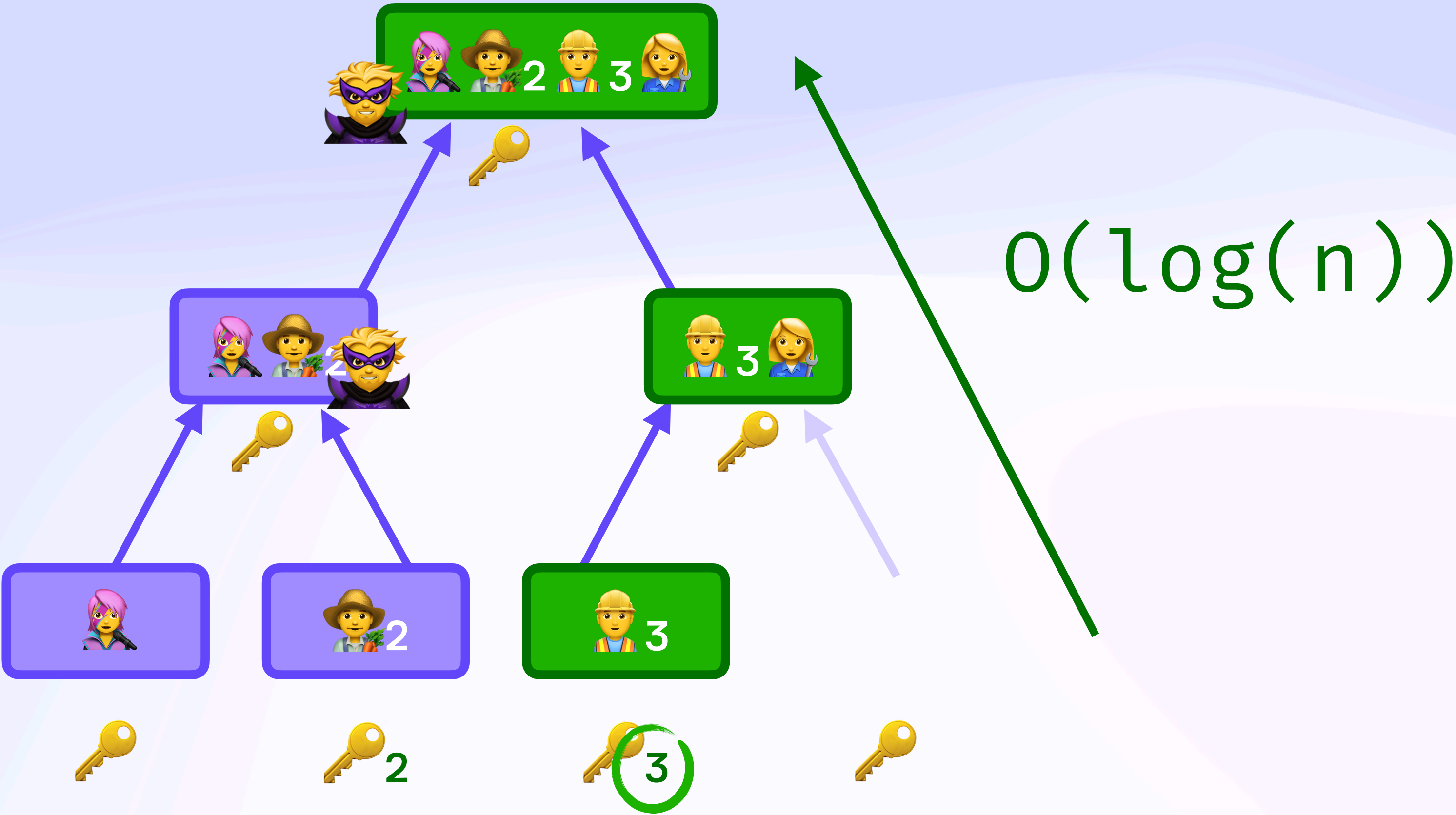
Self-Healing Concurrent Group Encryption

BeeKEM



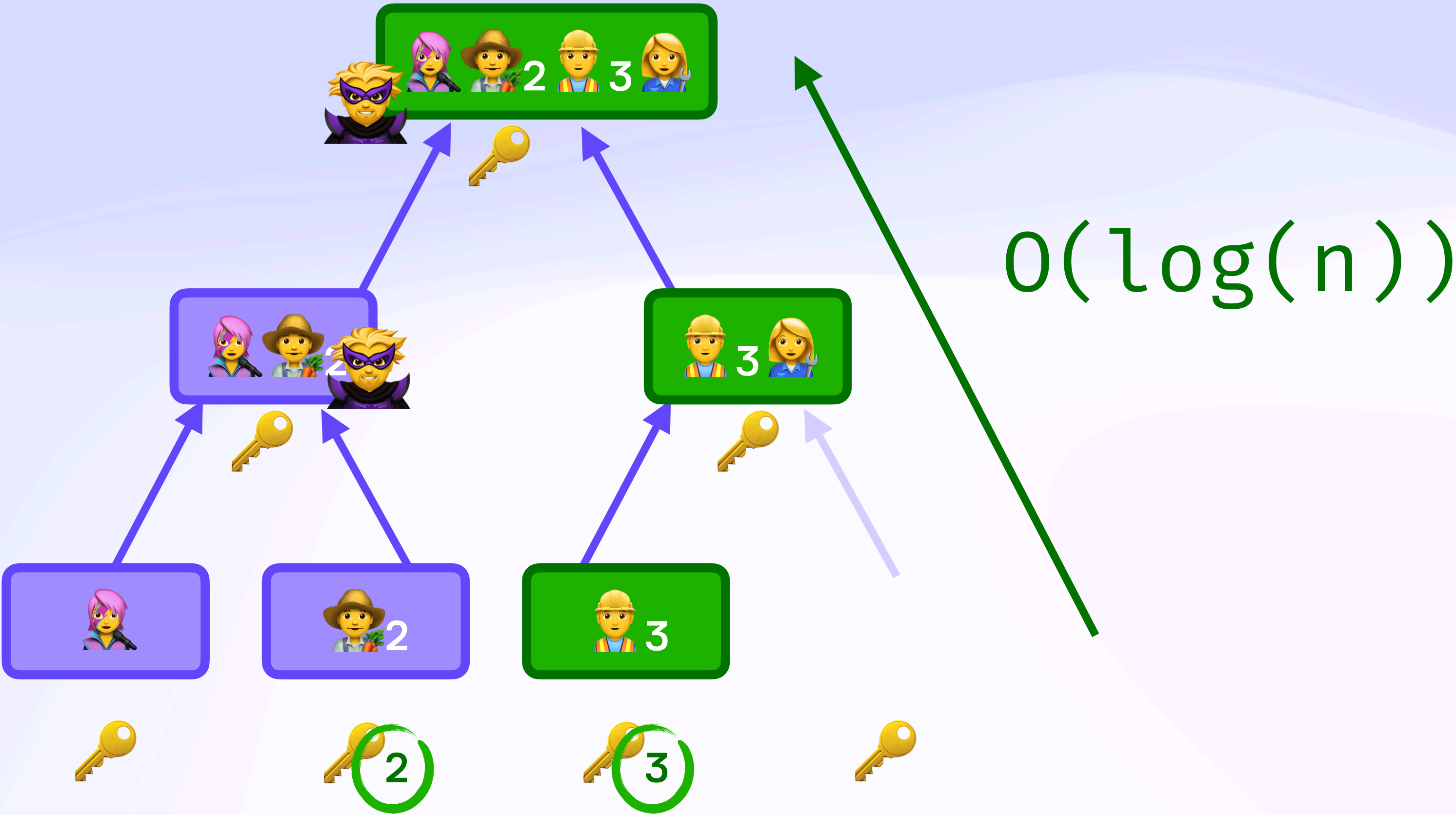
Self-Healing Concurrent Group Encryption

BeeKEM



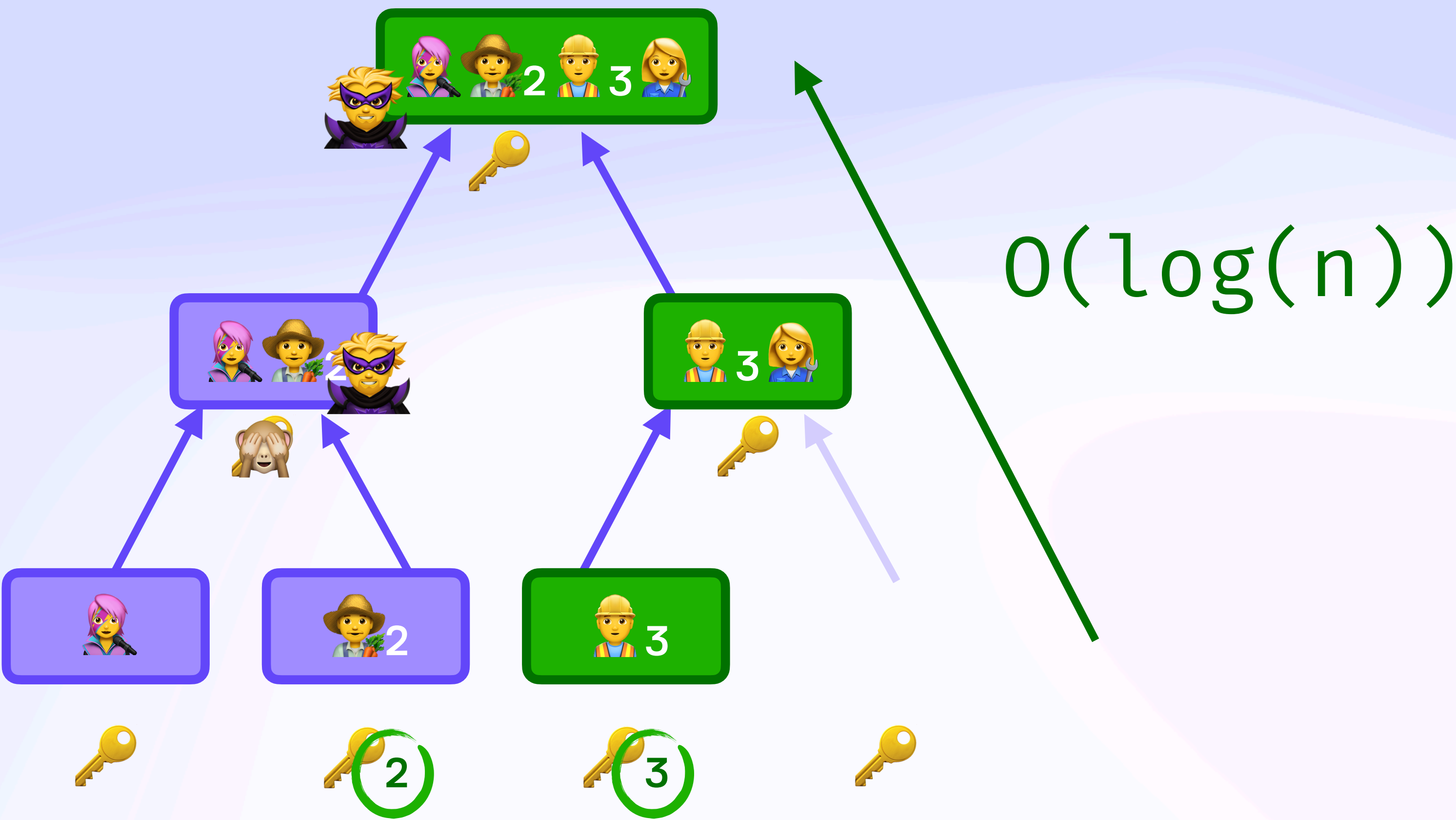
Self-Healing Concurrent Group Encryption

BeeKEM



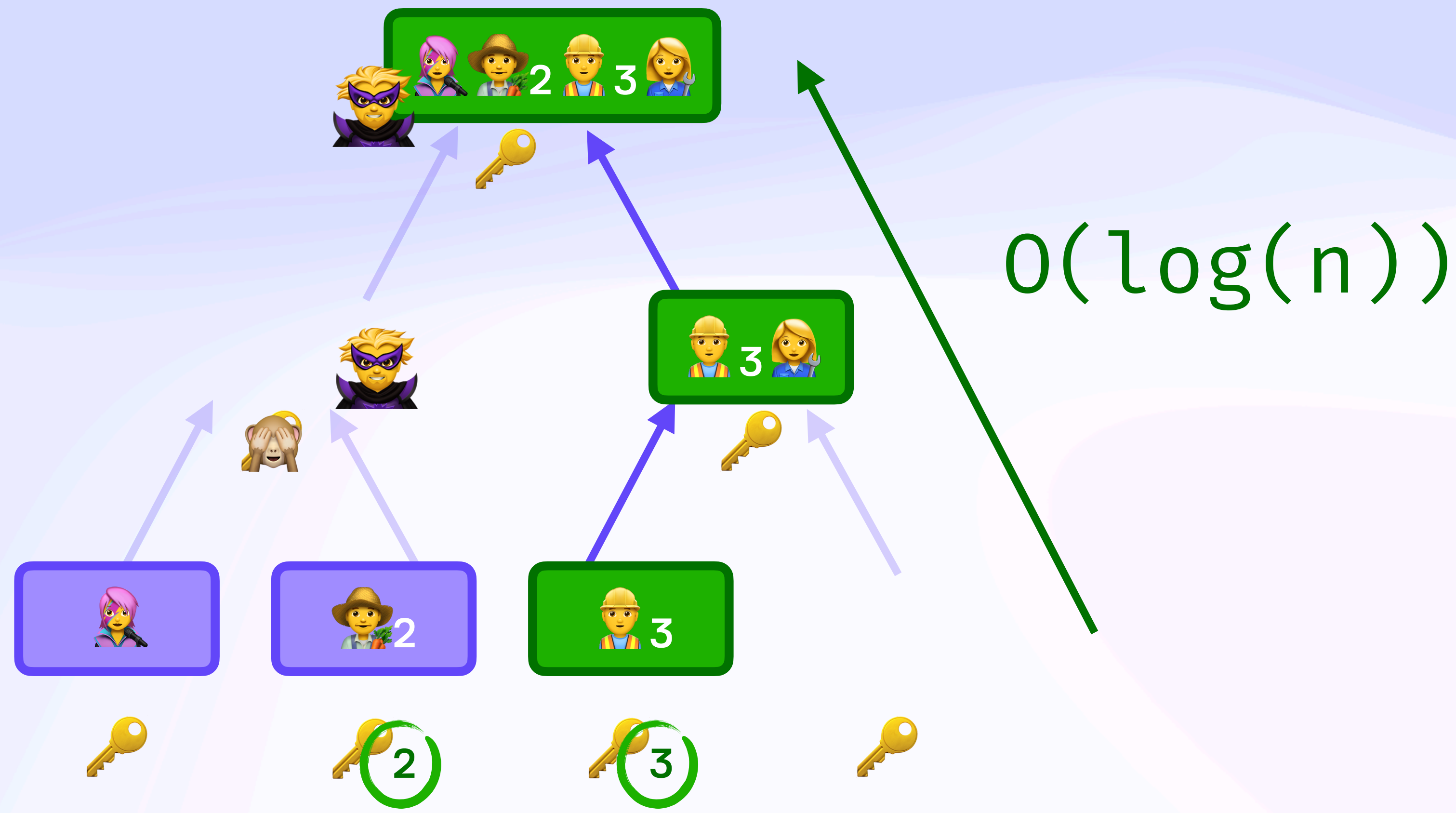
Self-Healing Concurrent Group Encryption

BeeKEM



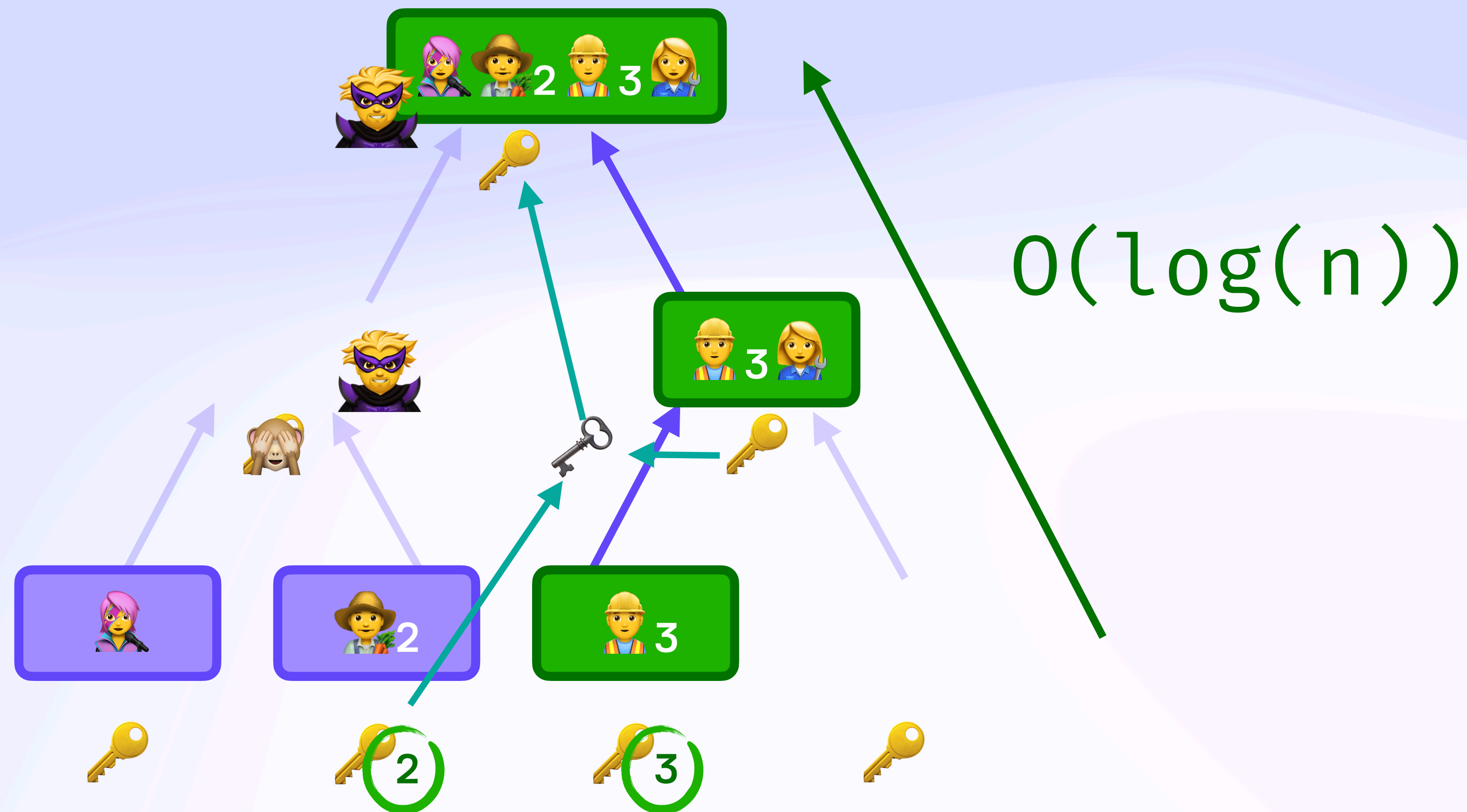
Self-Healing Concurrent Group Encryption

BeeKEM



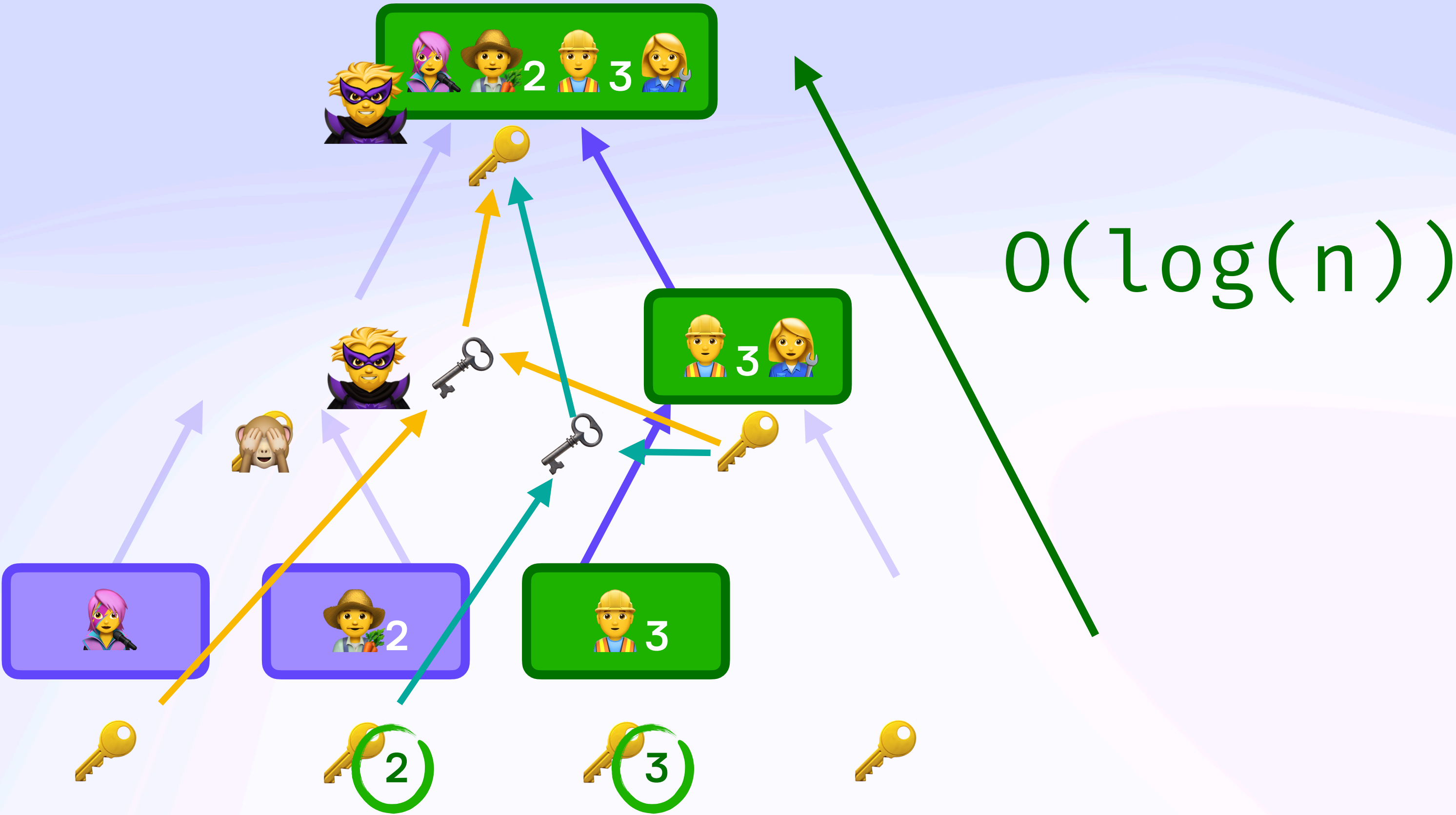
Self-Healing Concurrent Group Encryption

BeeKEM



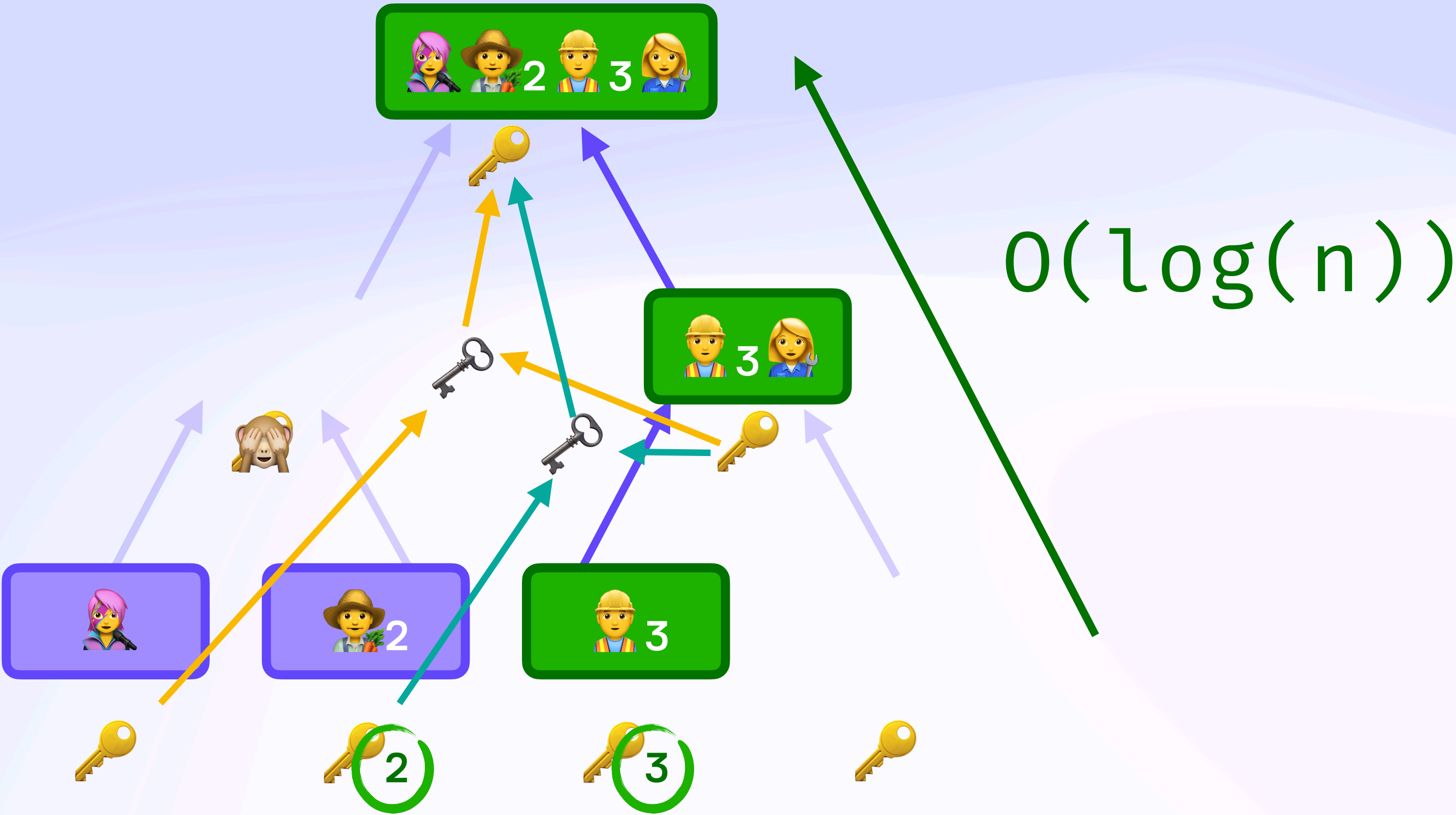
Self-Healing Concurrent Group Encryption

BeeKEM



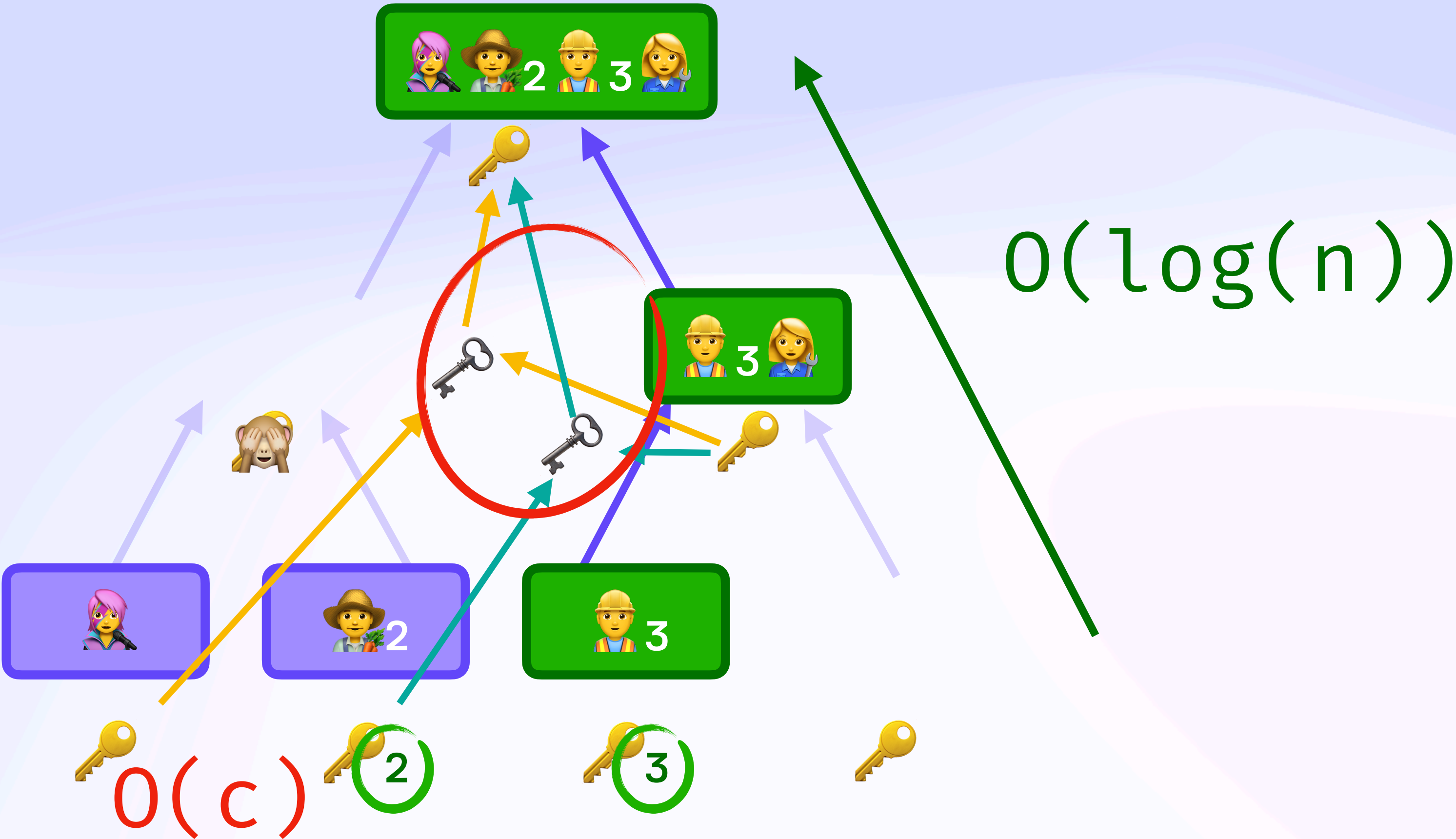
Self-Healing Concurrent Group Encryption

BeeKEM



Self-Healing Concurrent Group Encryption

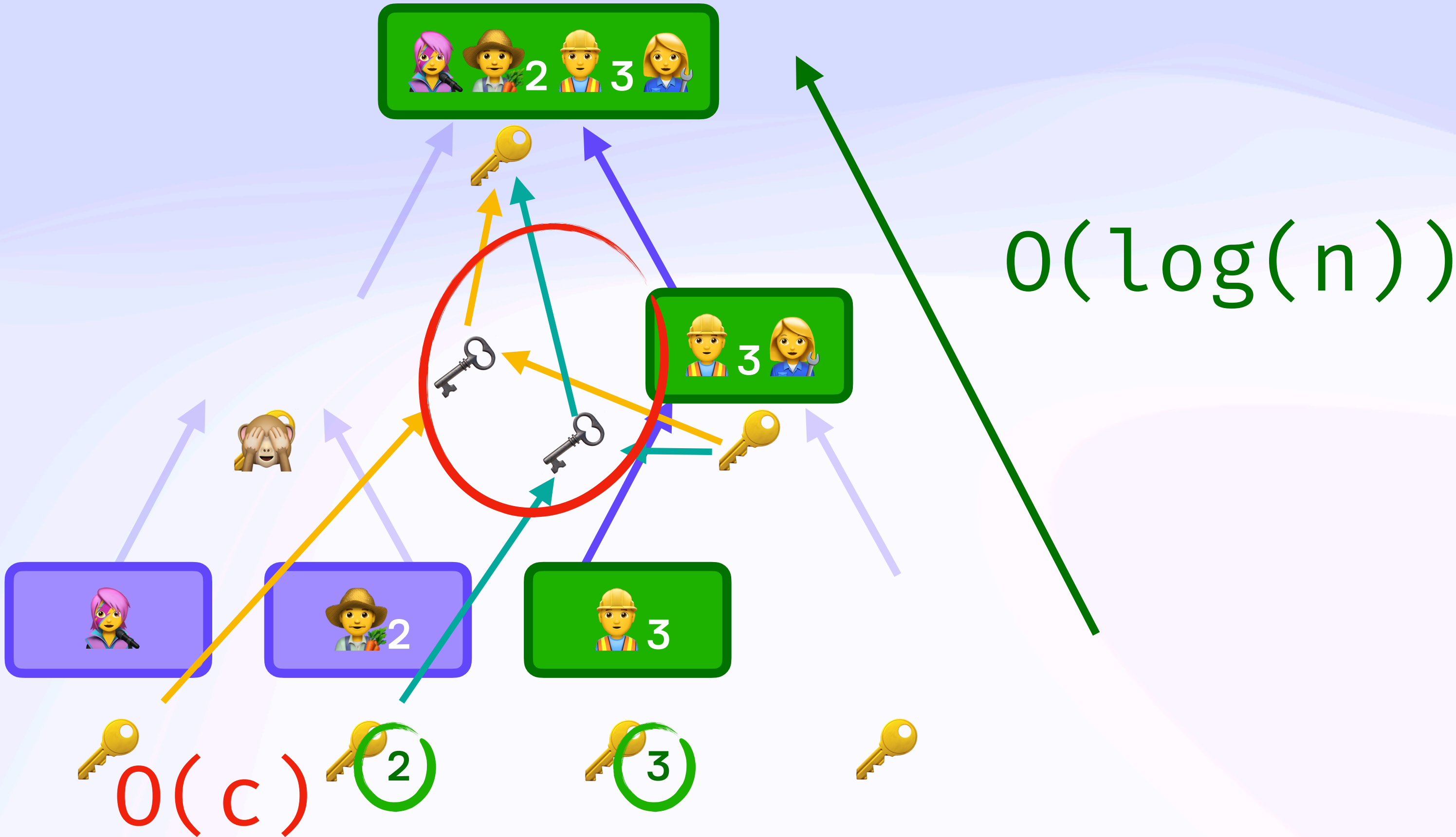
BeeKEM

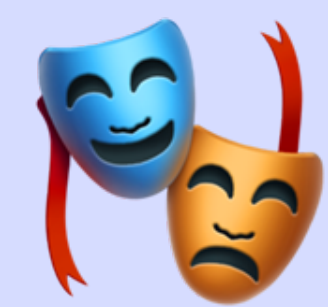


Self-Healing Concurrent Group Encryption

BeeKEM

Scales in $O(c + \log(n))$





Expressive & Extensible

The unreasonable effectiveness of syntactic checking

Expressive & Extensible

Policy Language

```
["==", ".planet.name", "Saturn"]
```

```
[  
  "and", [  
    [">=", ".team[9]?.size", 4],  
    ["==", ".ceo.first_name", "Boris"]  
  ]  
]
```

```
[  
  "every", ".recipient", [  
    "or", [  
      ["match", ".email", "*@example.com"],  
      ["==", ".email", "fraud@not.example.com"]  
    ]  
  ]  
]
```

Expressive & Extensible

Policy Language

```
[ "==", ".planet.name", "Saturn" ]
```

```
[  
  "and", [  
    [">=", ".team[9]?.size", 4],  
    ["==", ".ceo.first_name", "Boris"]  
  ]  
]
```

```
[  
  "every", ".recipient", [  
    "or", [  
      ["match", ".email", "*@example.com"],  
      ["==", ".email", "fraud@not.example.com"]  
    ]  
  ]  
]
```

jq-style selectors



Expressive & Extensible

Policy Language

```
["==", ".planet.name", "Saturn"]
```

```
[  
  "and", [  
    [">=", ".team[9]?.size", 4],  
    ["==", ".ceo.first_name", "Boris"]  
  ]  
]
```

=, match not every
>, ≥ and some
<, ≤ or

```
[  
  "every", ".recipient", [  
    "or", [  
      ["match", ".email", "*@example.com"],  
      ["==", ".email", "fraud@not.example.com"]  
    ]  
  ]  
]
```

jq-style selectors

Expressive & Extensible

Syntactically Driven

```
{
  iss: "did:key:alice",
  aud: "did:key:bob",
  sub: "did:key:doc"
  cmd: "/automerge/update",
  pol: [
    ["gt", ".foo[1].bar", "42"],
    ["eq", ".prev", 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4]
  ]
}
```

```
{
  iss: "did:key:bob",
  aud: "did:key:doc",
  sub: "did:key:doc"
  prf: [hash(delegation), hash(more_chain)]
  cmd: "/automerge/update",
  args: {
    foo: [
      {bar: 100}, // > 42
      {quux: "unconstrained"}
    ],
    prev: 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4
  }
}
```

Expressive & Extensible

Syntactically Driven

```
{  
  iss: "did:key:alice",  
  aud: "did:key:bob",  
  sub: "did:key:doc"  
  cmd: "/automerge/update",  
  pol: [  
    ["gt", ".foo[1].bar", "42"],  
    ["eq", ".prev", 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4]  
  ]  
}
```

```
{  
  iss: "did:key:bob",  
  aud: "did:key:doc",  
  sub: "did:key:doc"  
  prf: [hash(delegation), hash(more_chain)]  
  cmd: "/automerge/update",  
  args: {  
    foo: [  
      {bar: 100}, // > 42  
      {quux: "unconstrained"}  
    ],  
    prev: 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4  
  }  
}
```


Expressive & Extensible

Syntactically Driven

```
{
  iss: "did:key:alice",
  aud: "did:key:bob",
  sub: "did:key:doc",
  cmd: "/automerge/update",
  pol: [
    ["gt", ".foo[1].bar", "42"],
    ["eq", ".prev", 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4]
  ]
}
```

```
{
  iss: "did:key:bob",
  aud: "did:key:doc",
  sub: "did:key:doc",
  prf: [hash(delegation), hash(more_chain)]
  cmd: "/automerge/update",
  args: {
    foo: [
      {bar: 100}, // > 42
      {quux: "unconstrained"}
    ],
    prev: 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4
  }
}
```

Expressive & Extensible

Syntactically Driven

```
{  
  iss: "did:key:alice",  
  aud: "did:key:bob",  
  sub: "did:key:doc",  
  cmd: "/automerge/update",  
  pol: [  
    ["gt", ".foo[1].bar", "42"],  
    ["eq", ".prev", 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4]  
  ]  
}
```



```
{  
  iss: "did:key:bob",  
  aud: "did:key:doc",  
  sub: "did:key:doc",  
  prf: [hash(delegation), hash(more_chain)]  
  cmd: "/automerge/update",  
  args: {  
    foo: [  
      {bar: 100}, // > 42  
      {quux: "unconstrained"}  
    ],  
    prev: 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4  
  }  
}
```

Wrap Up



Wrap Up


Wrap Up

Wrap Up

Wrap Up


- ♦ Mutation Control
 - ♦ Convergent capabilities
 - ♦ Concurrent revocation
- ♦ Read Control
 - ♦ Causal encryption
 - ♦ Continuous group key agreement (CGKA)

People with access



hello@katiewilde.com
hello@katiewilde.com


Owner



Brooklyn Zelenka (you)
hello@brooklynzelenka.com


Editor ▼

General access



Restricted ▼

Only people with access can open with the link

 Copy link

Done

🦋 @expede.wtf

🐘 @expede@types.pl

✍️ notes.brooklynzelenka.com



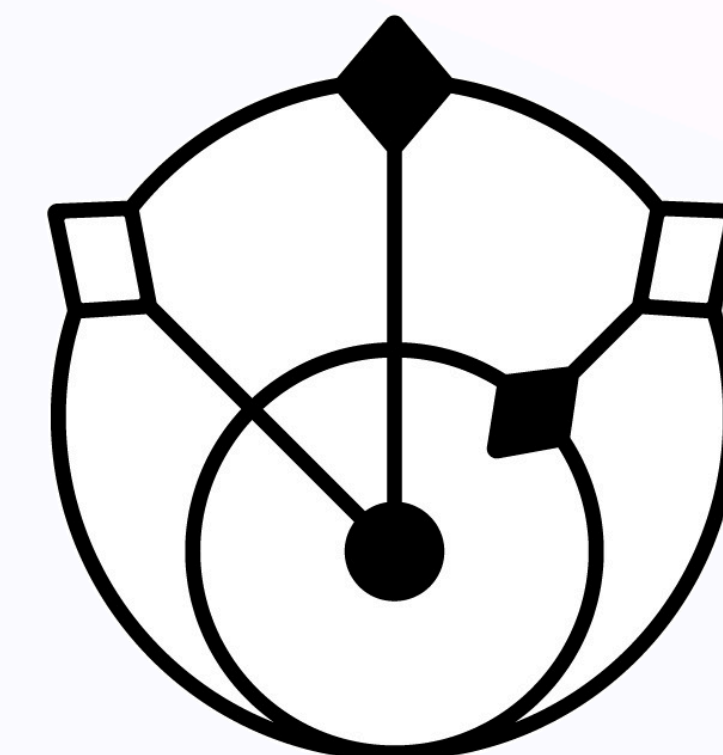
Thank You, DWWeb YVR



github.com/ucan-wg



inkandswitch.com/beehive



* On the unceded ancestral lands of the xʷməθkʷəy̓əm, Skwxwú7mesh, and səliłwətał Nations