



# ***The Jump to Hyperspace***

 Light Speed, User Agency, & Moving Past the Cloud 



# The Jump to Hyperspace

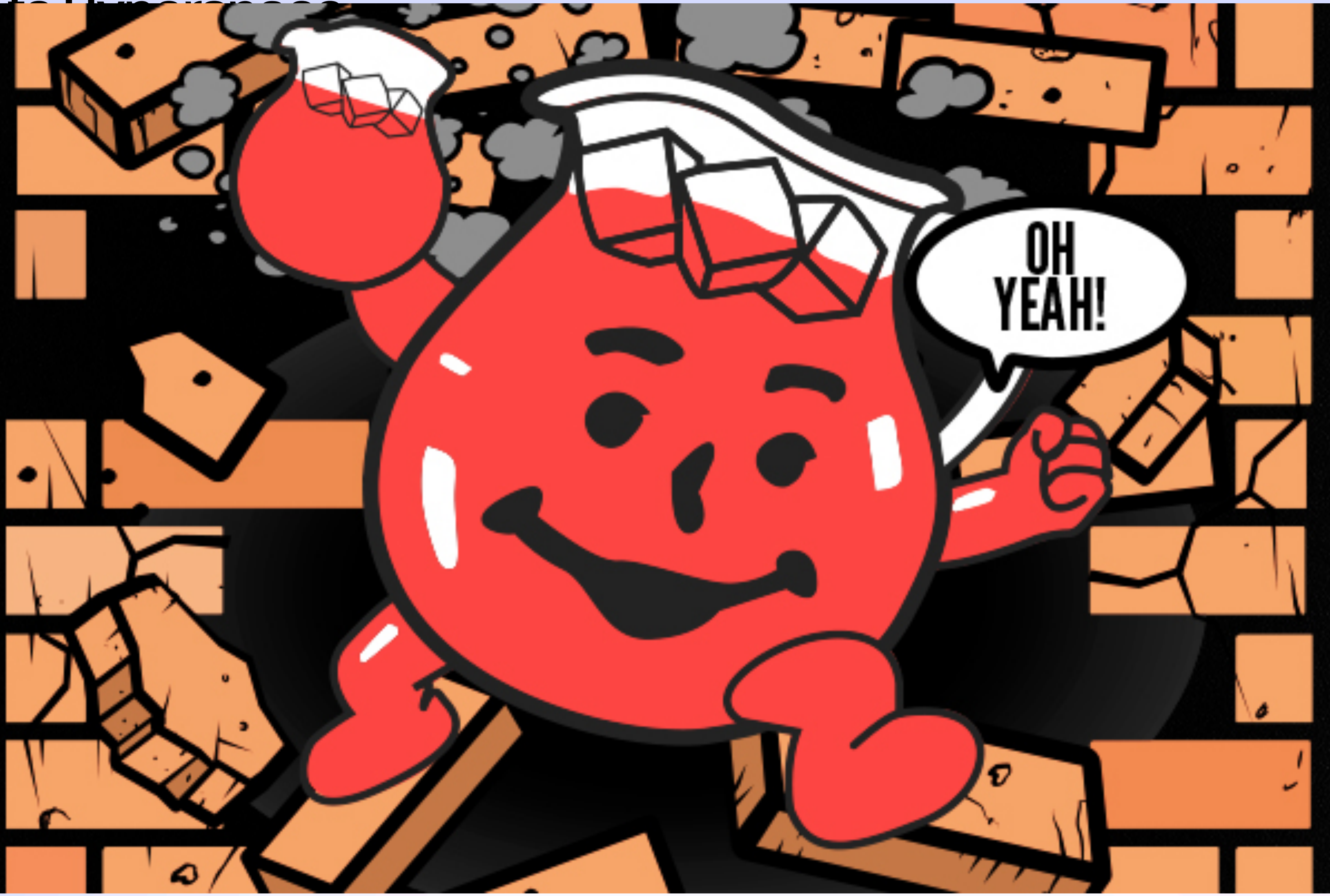


# The Jump to Hyperspace

I suppose it is tempting,  
if the only tool you have is a **hammer**,  
to treat everything as if it were a nail

– Abraham Maslow

The Jump to Happiness



if th  
to

er,  
ail

The Jump to Hyperspace

***Brooklyn Zelenka @expede***

The Jump to Hyperspace

**Brooklyn Zelenka @expede**



[github.com/expede](https://github.com/expede)  
Vancouver 🇨🇦

# The Jump to Hyperspace

## **Brooklyn Zelenka @expede**

- ◆ Auth team lead at Ink & Switch
  - ◆ Beehive: local-first access control 🐝
- ◆ Spec editor at UCAN Working Group
- ◆ Prev. Ethereum core dev
- ◆ PLs and DS are my jam 🙌



[github.com/expede](https://github.com/expede)  
Vancouver 🇨🇦



# The Jump to Hyperspace

## Topics




# The Jump to Hyperspace

## *Topics*

- ◆ **High Level:** Current state of (networked) software
- ◆ **What is LoFi?** Philosophy & high level architecture
  1. Data
  2. Auth
  3. Compute
- ◆ **Looking forward:** developments for the next chapter



# The Jump to Hyperspace

The background of the slide is an abstract, fluid design. It features a series of overlapping, wavy lines that create a sense of depth and movement. The color palette is soft and ethereal, consisting of various shades of light blue, lavender, and pale pink, all set against a very light, almost white, background. The lines flow from the bottom left towards the top right, with some areas appearing more saturated than others, giving the overall effect a dreamlike, nebular quality.

# The Jump to Hyperspace

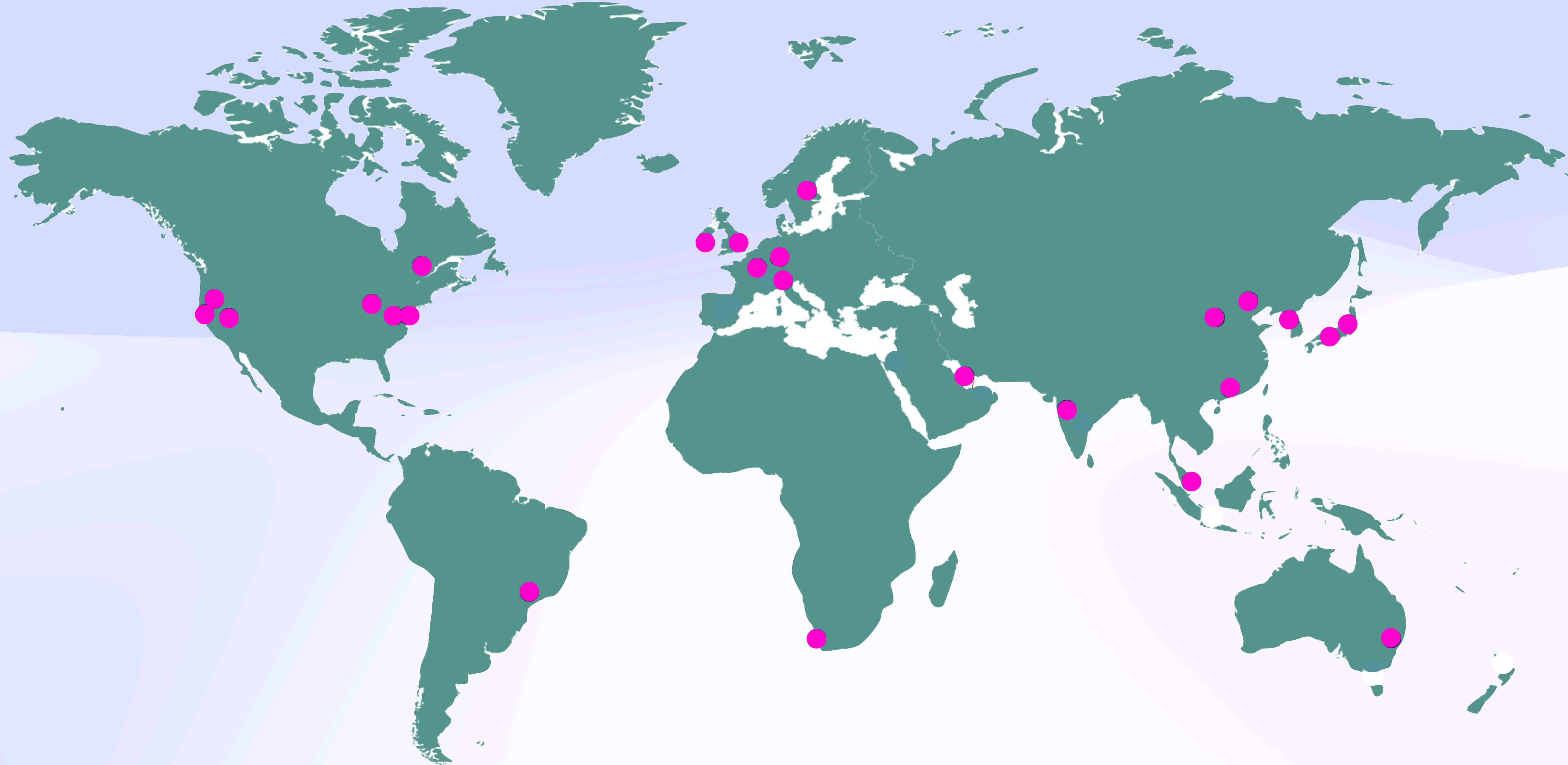
## Audience Poll

# ***Economic Weight Class***

 One Size Does Not Fit All 

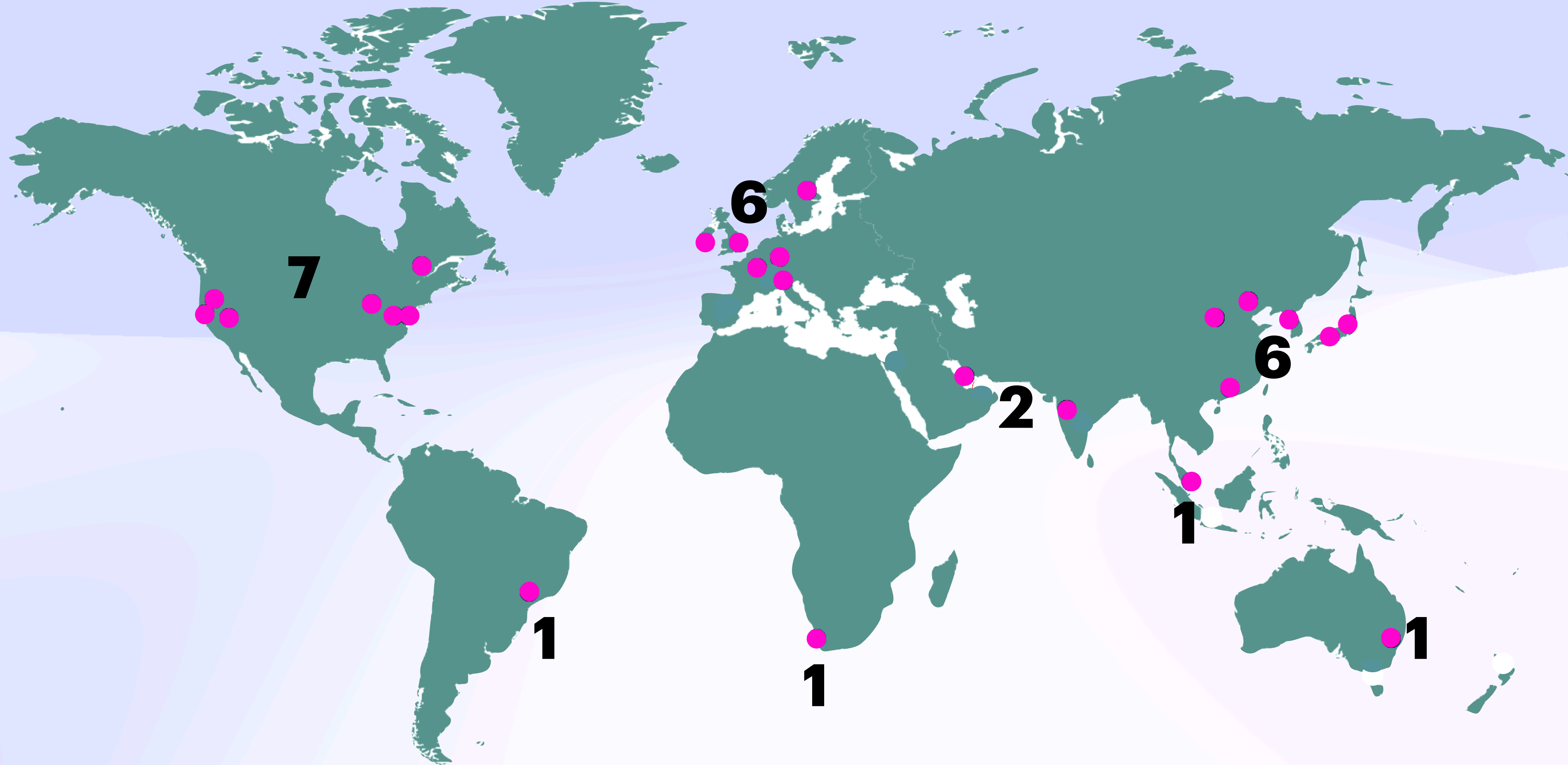
# Economic Weight Class

## *Users vs Cloud Infra*



# Economic Weight Class

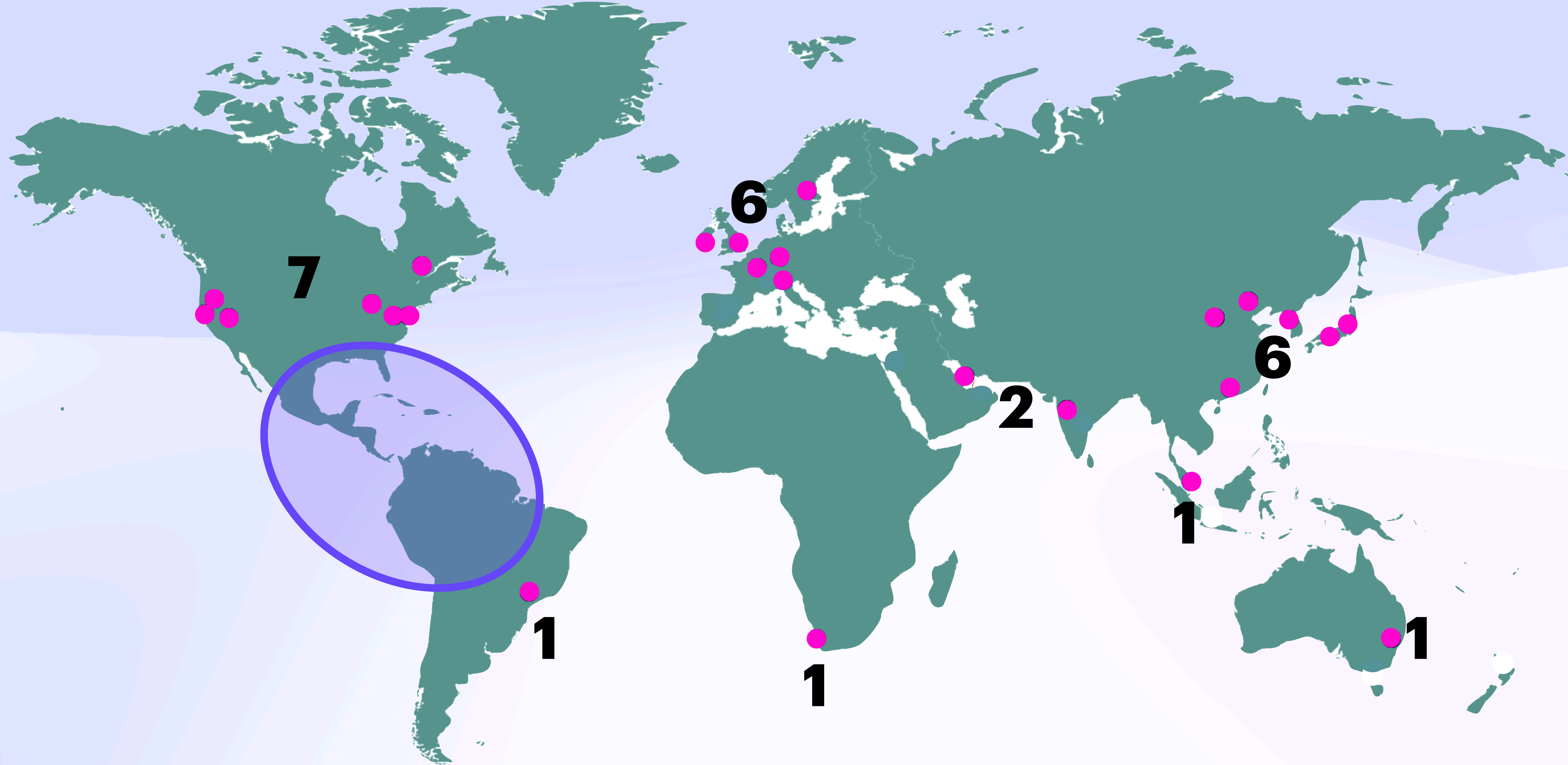
## *Users vs Cloud Infra*



Source: AWS

# Economic Weight Class

## *Users vs Cloud Infra*

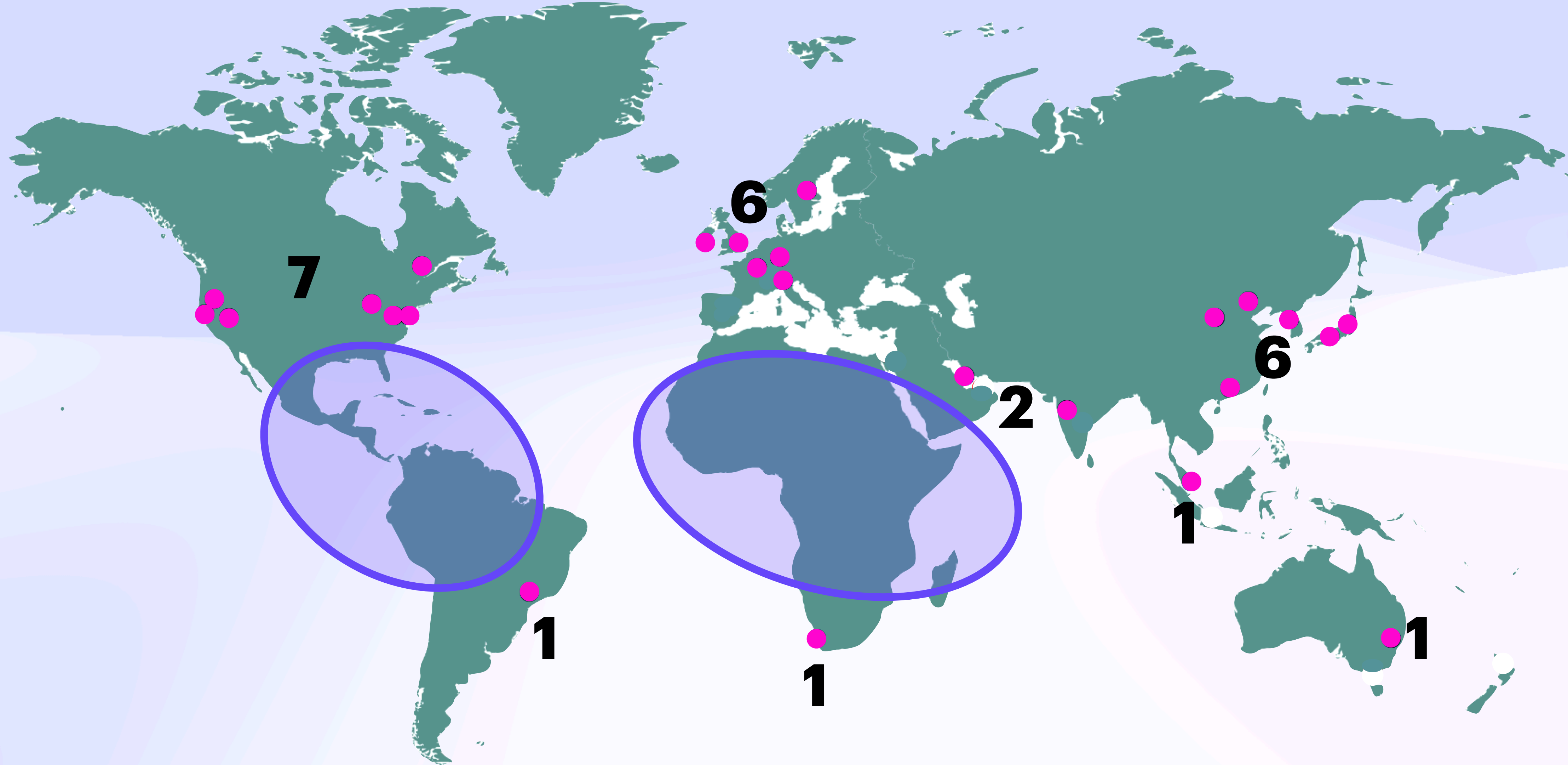


Source: AWS



# Economic Weight Class

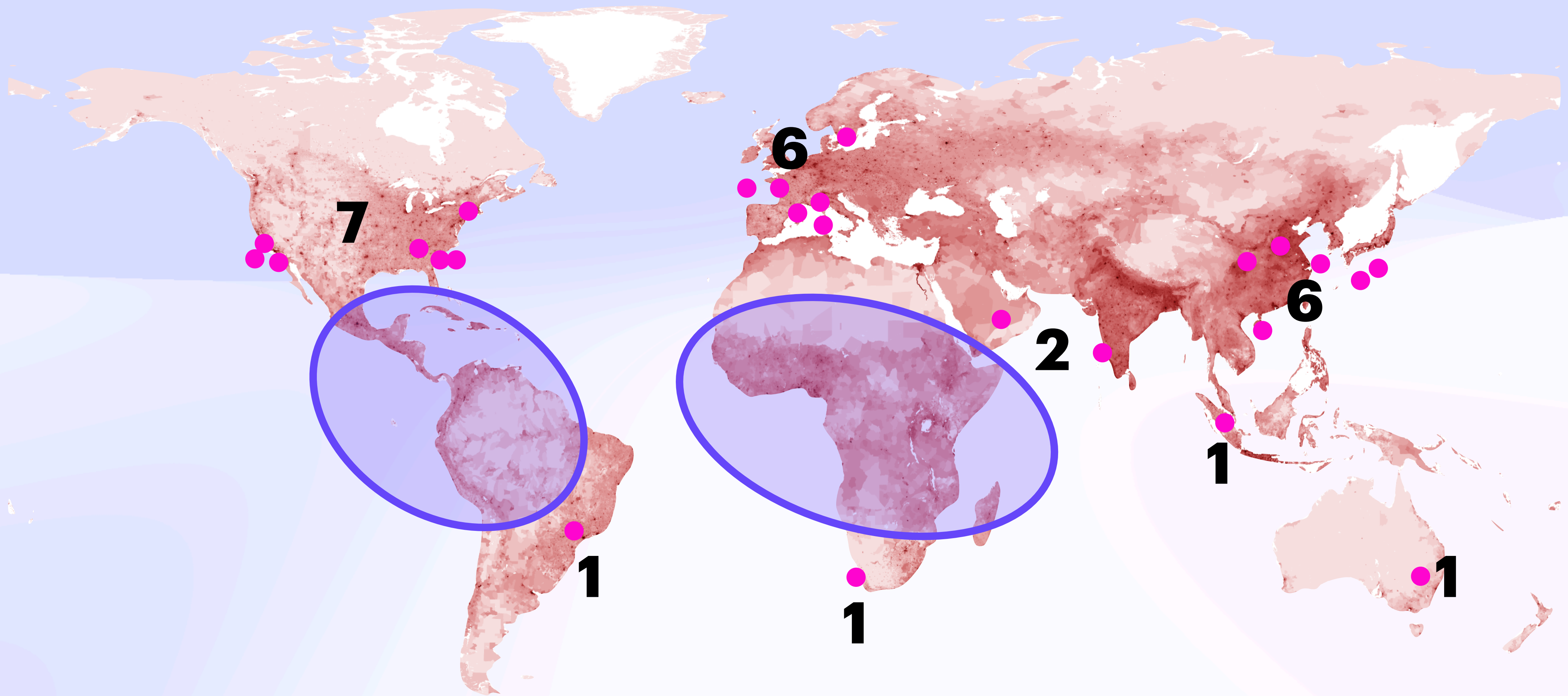
## *Users vs Cloud Infra*



Source: AWS

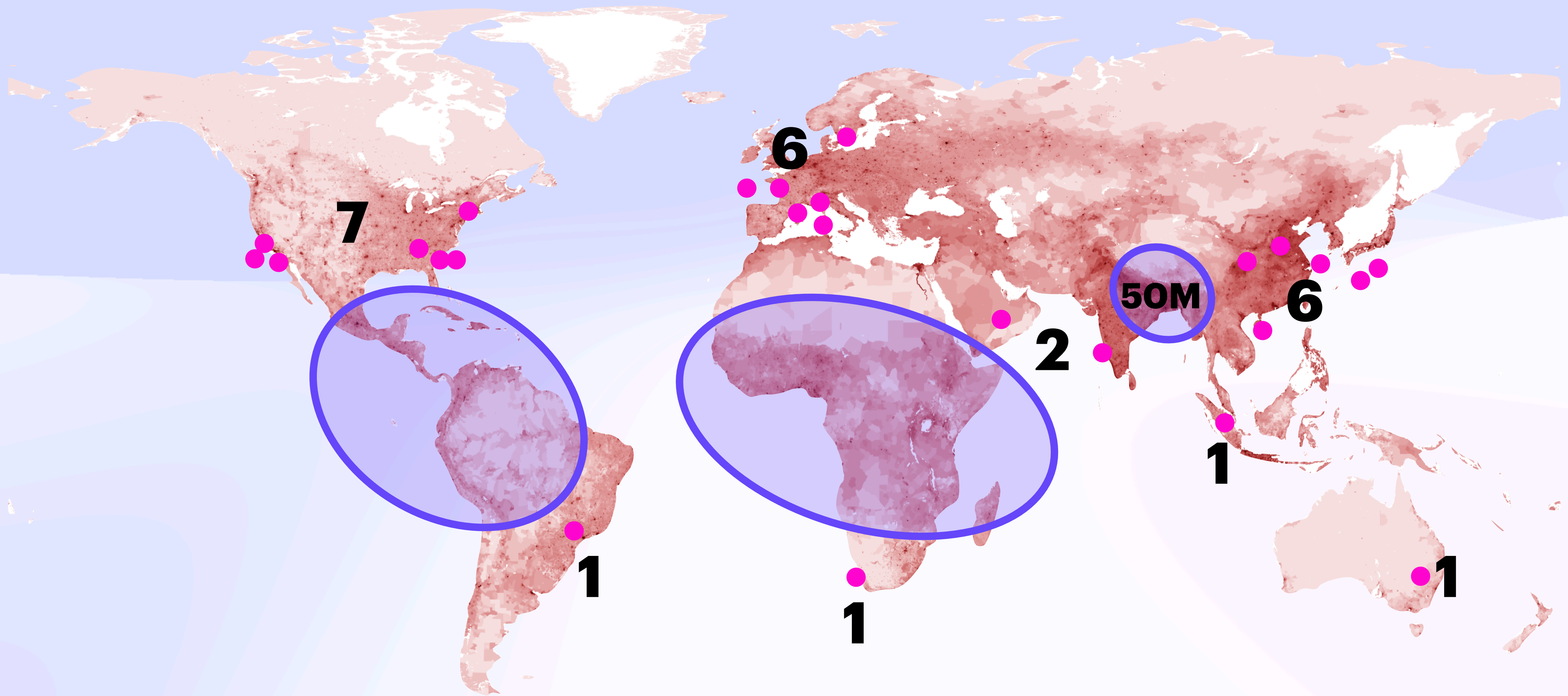
# Economic Weight Class

## *Users vs Cloud Infra*



# Economic Weight Class

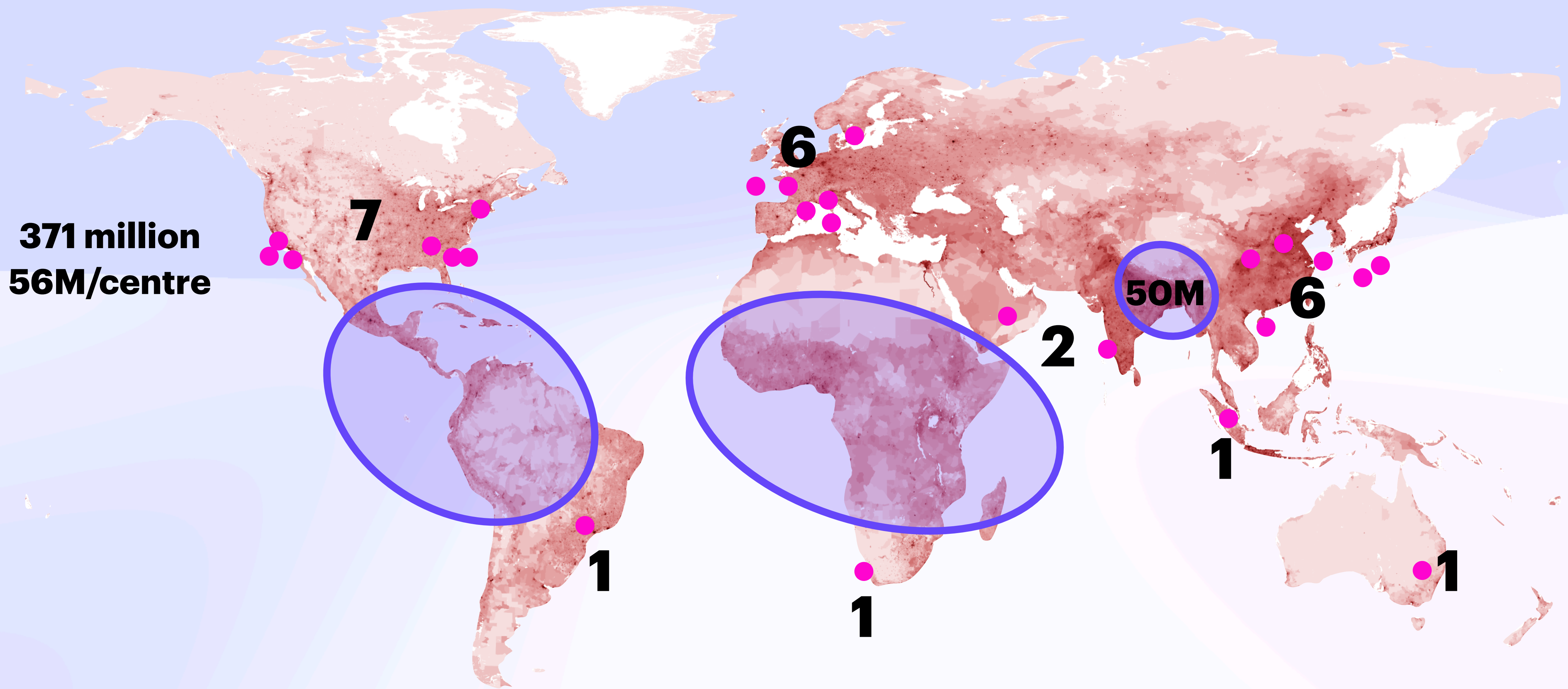
# *Users vs Cloud Infra*



Source: AWS

# Economic Weight Class

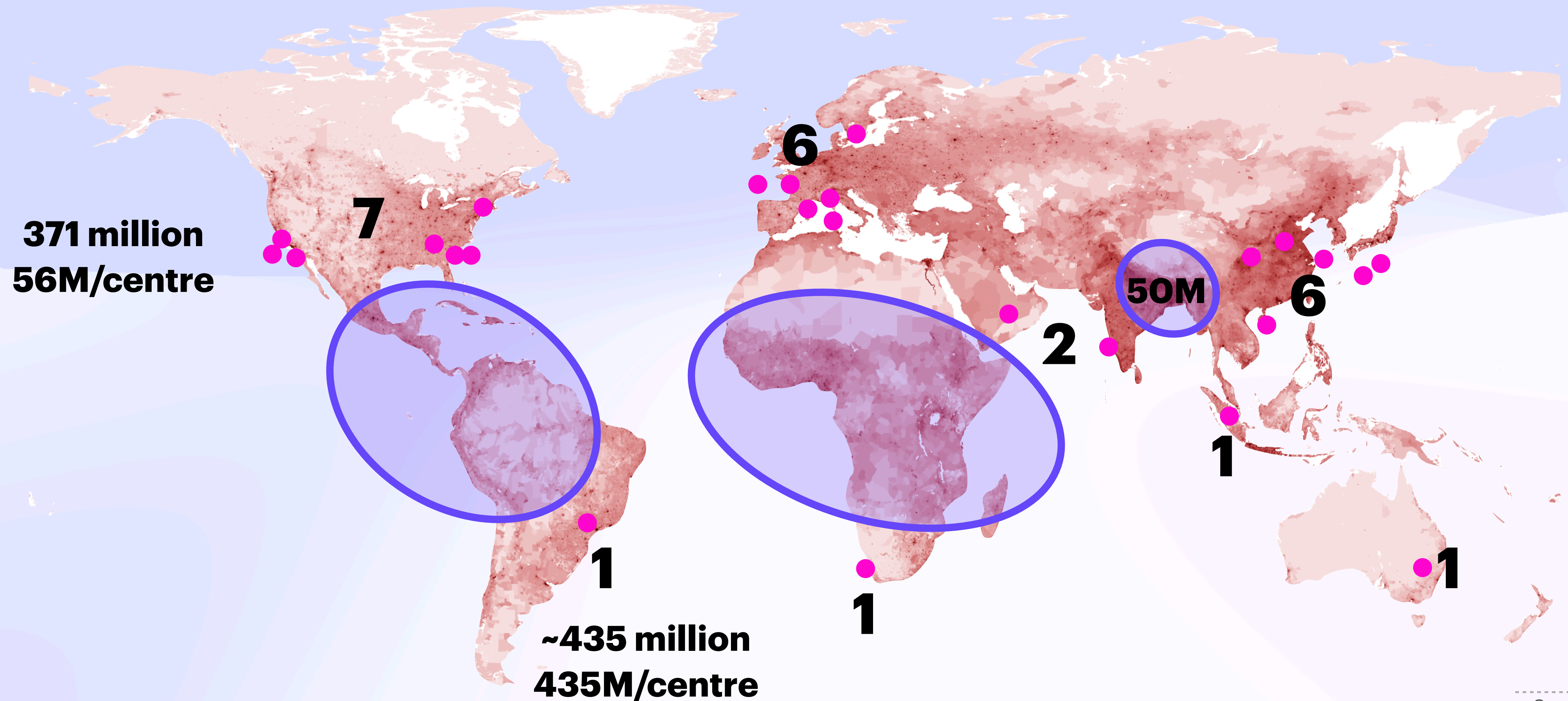
# *Users vs Cloud Infra*



371 million  
56M/centre

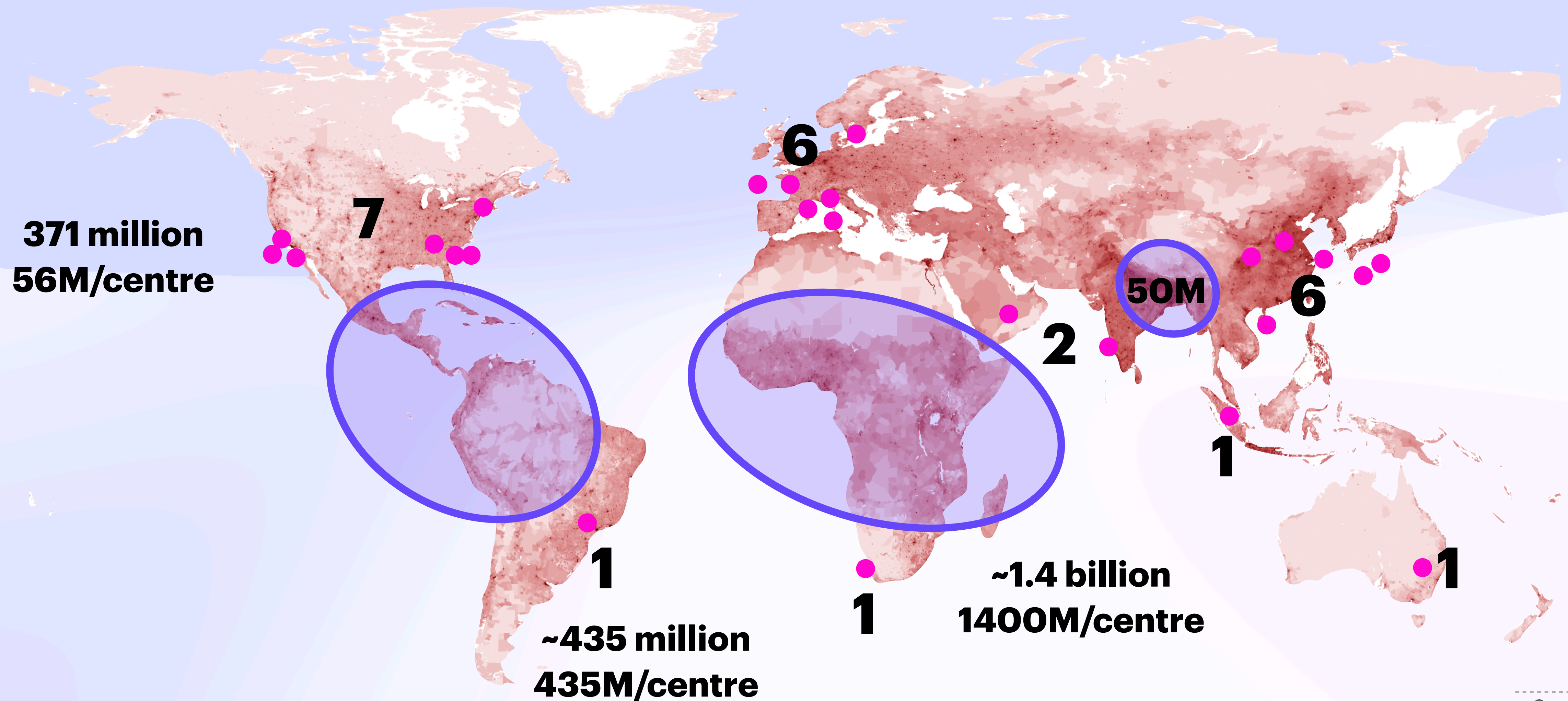
# Economic Weight Class

## *Users vs Cloud Infra*



# Economic Weight Class

## *Users vs Cloud Infra*



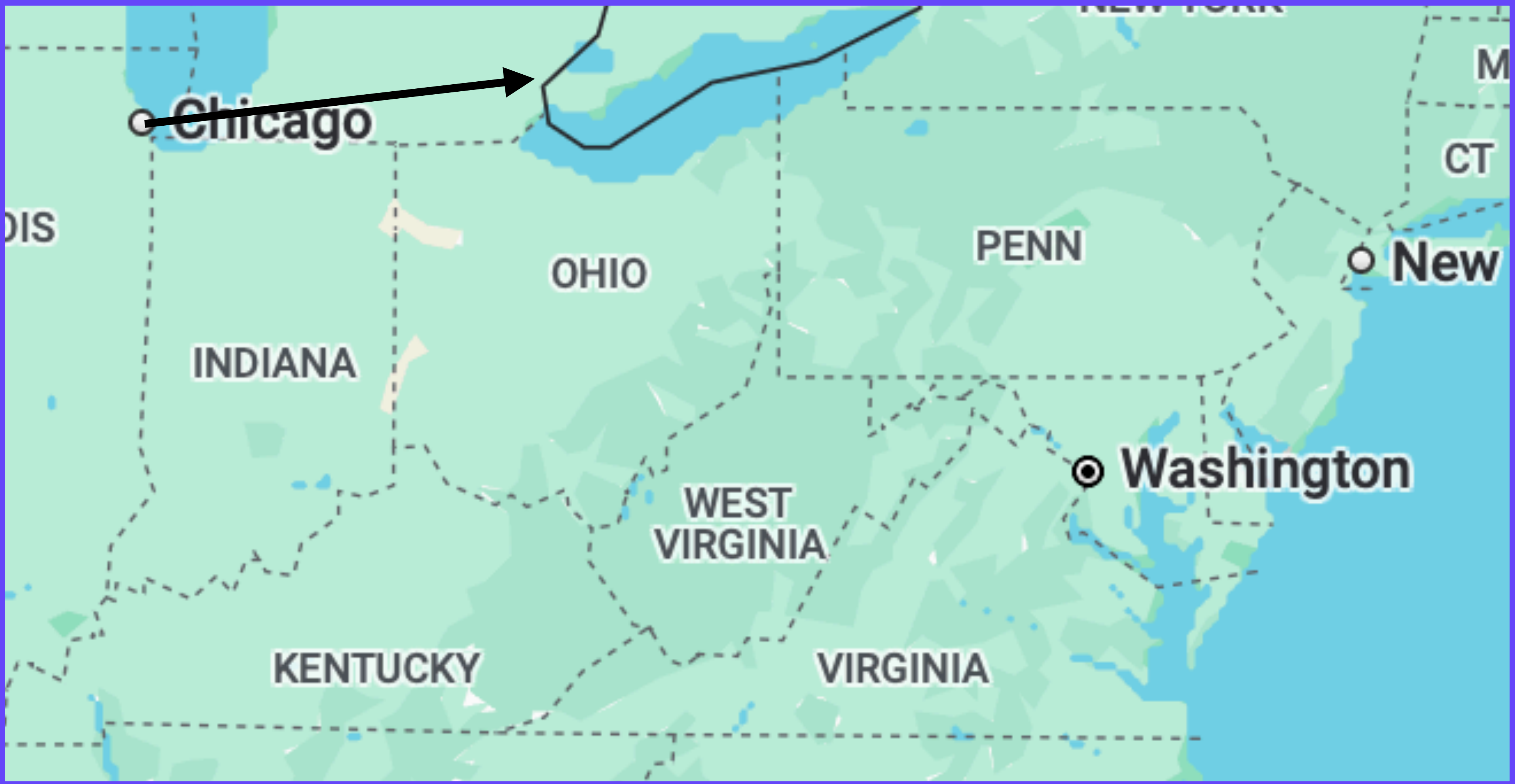
Economic Weight Class

# *Sending a "Direct" Message in 2024*



Economic Weight Class

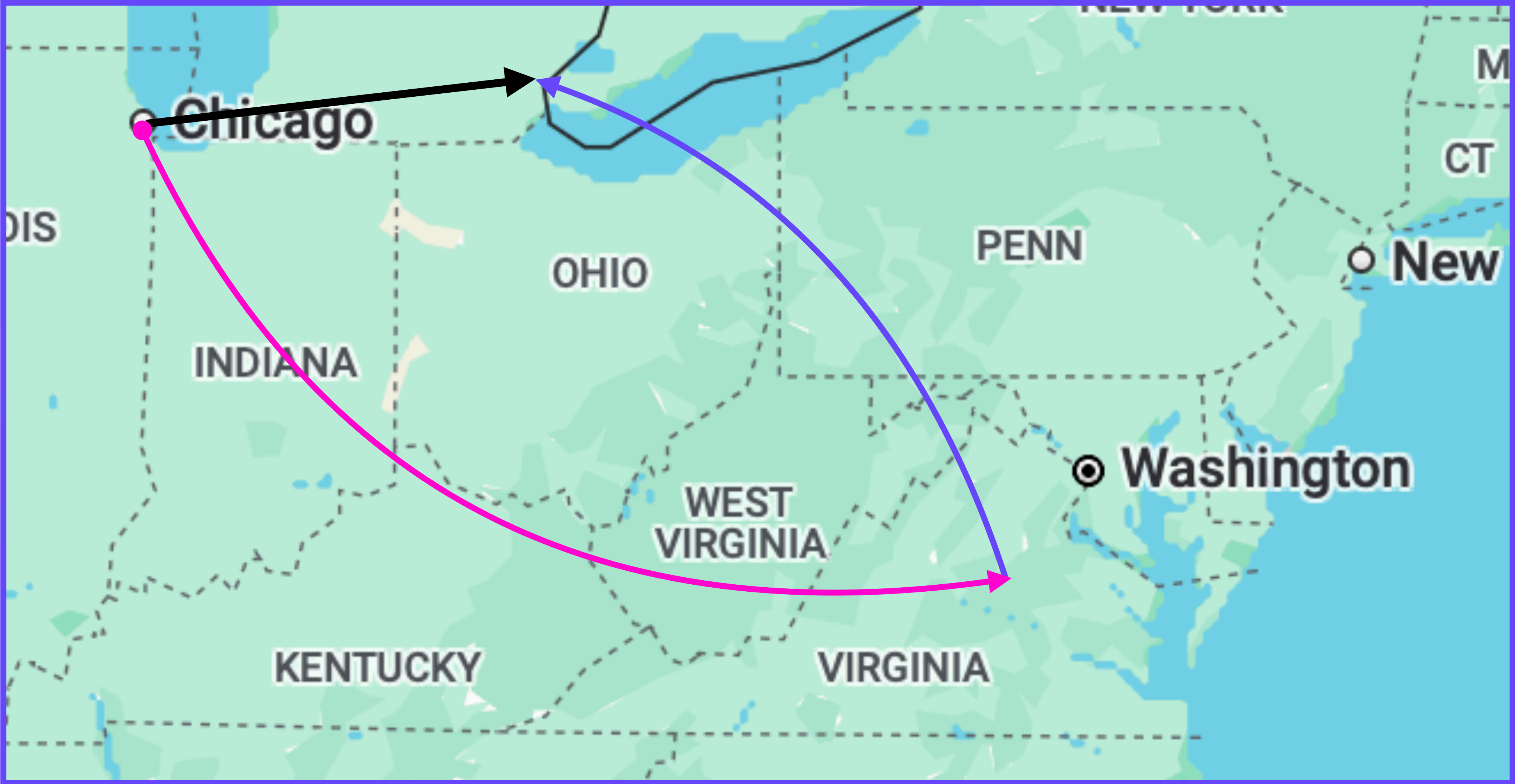
# *Sending a "Direct" Message in 2024*





Economic Weight Class

# *Sending a "Direct" Message in 2024*



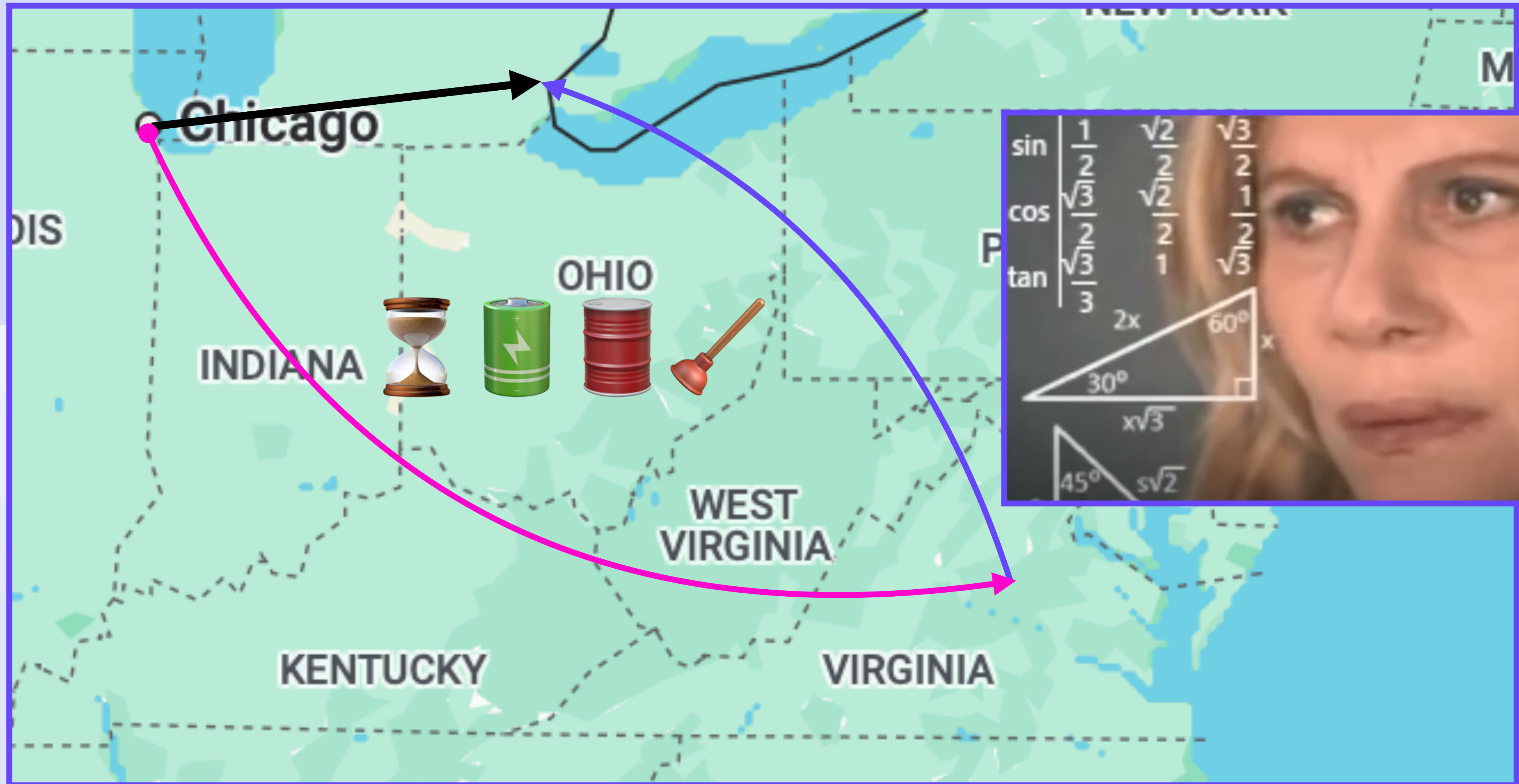
Economic Weight Class

# *Sending a "Direct" Message in 2024*



Economic Weight Class

# ***Sending a "Direct" Message in 2024***



# Economic Weight Class

***Industrial***

USS Nimitz

Crew: ~5k

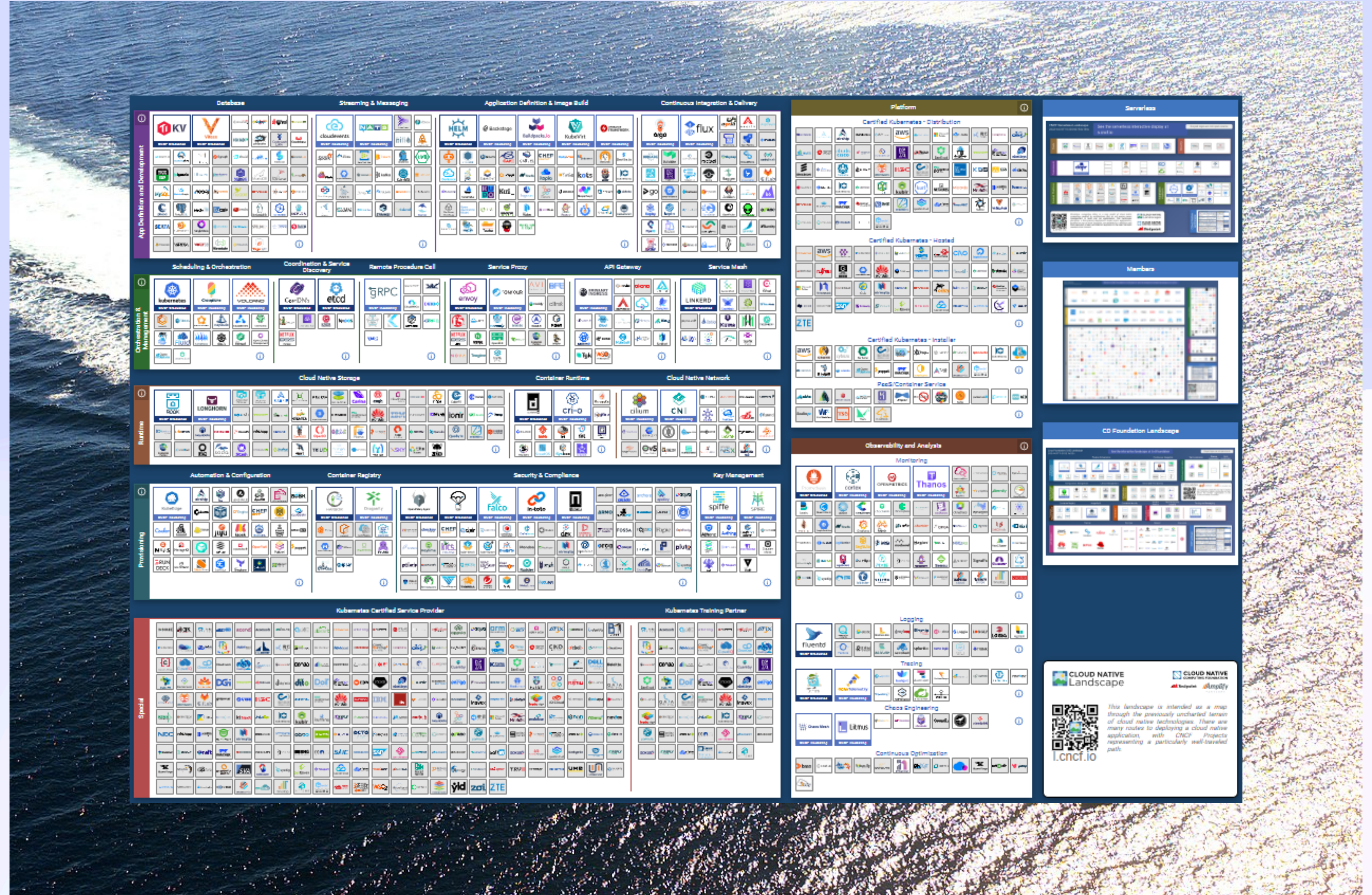
Cost: >\$8.5 billion (USD)



# Economic Weight Class

# *Industrial*

USS Nimitz  
Crew: ~5k  
Cost: >\$8.5 billion (USD)



Economic Weight Class

***Consequences*** 

Economic Weight Class

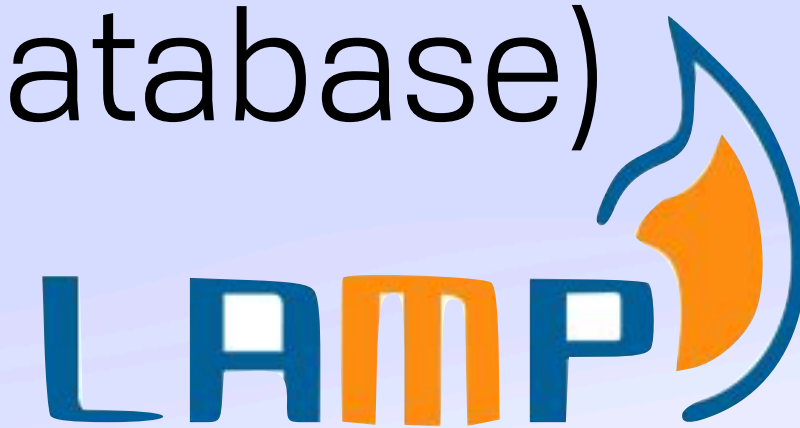
# *Consequences*

- Single source of truth ("**the**" database)

## Economic Weight Class

# *Consequences*

- Single source of truth ("**the**" database)
- Server-centric



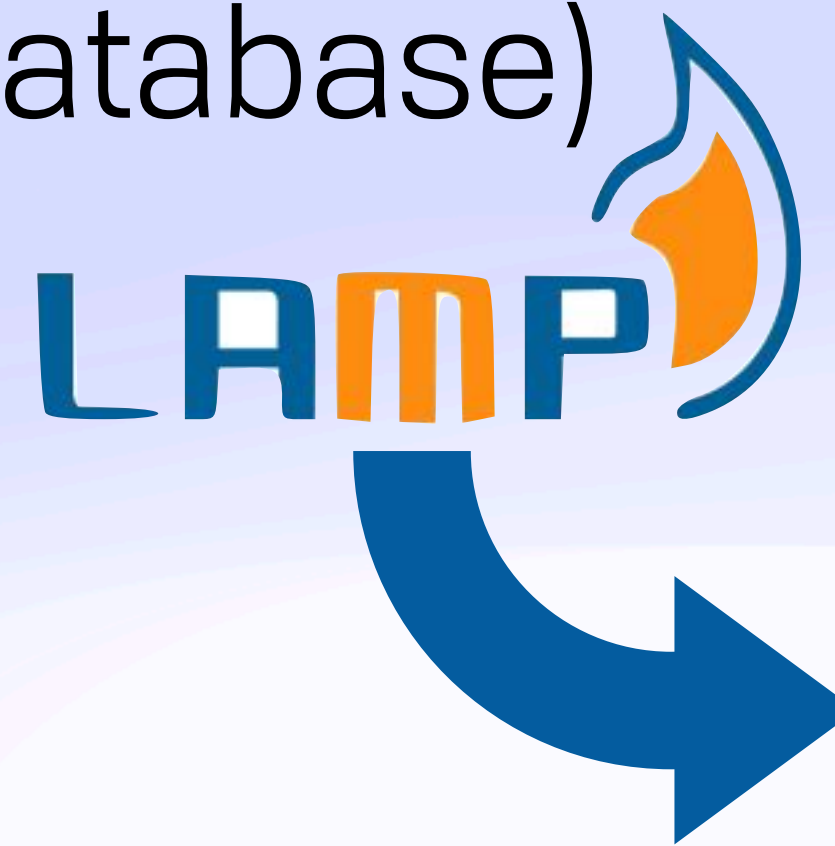
- "Full stack development"
- DevOps, Docker, k8s
- How to train enough engineers?



## Economic Weight Class

# Consequences 🍂

- Single source of truth ("**the**" database)
- Server-centric
- "Full stack development"
- DevOps, Docker, k8s
- How to train enough engineers?




Economic Weight Class

# ***Condensing The Stack***

Economic Weight Class

# *Condensing The Stack*

Users 

Developer 

# Economic Weight Class

# *Condensing The Stack*

Users 

Browser 

REST / JSON-RPC / GraphQL 

Server 

Data Store 

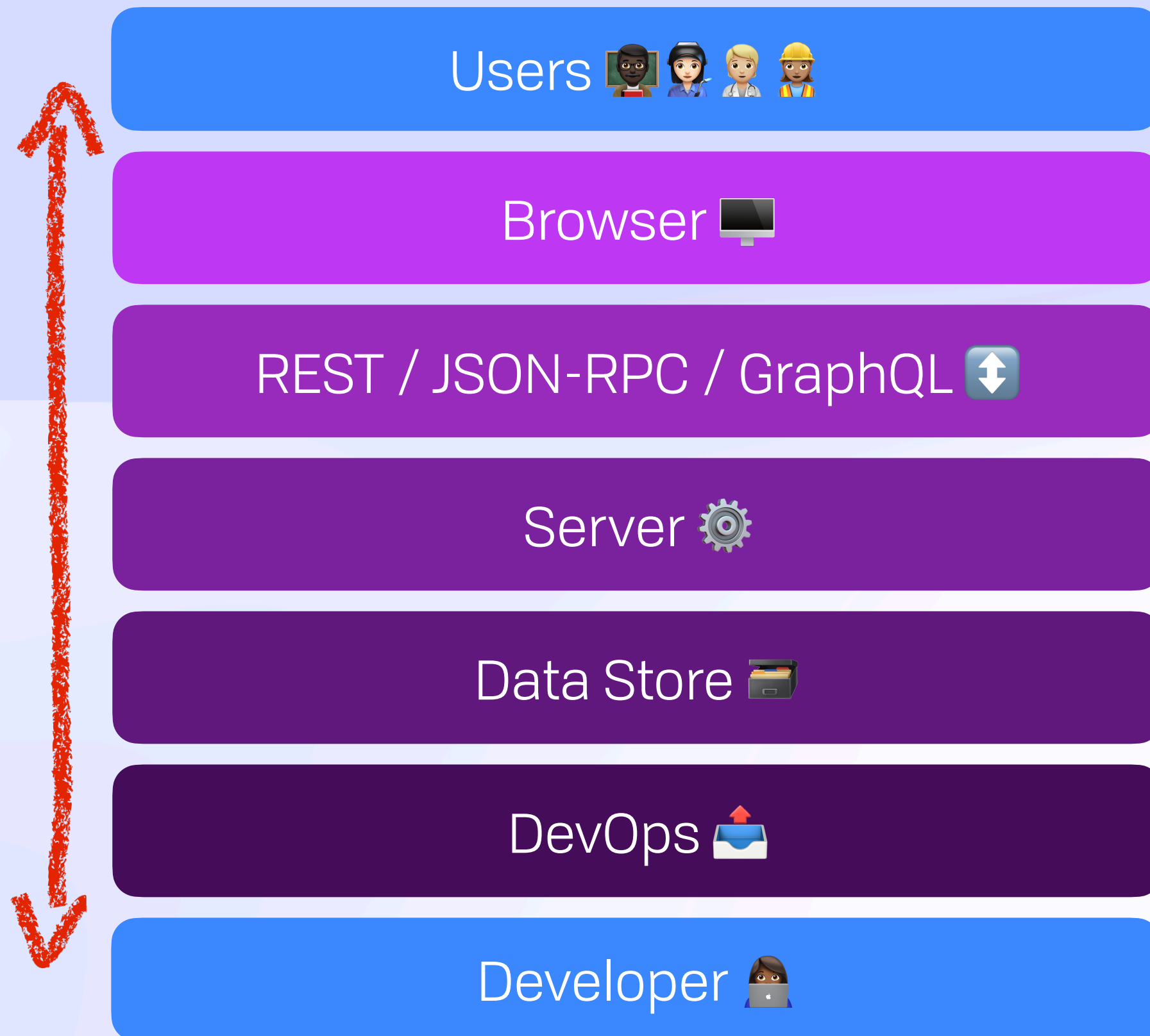
DevOps 

Developer 

# Economic Weight Class

## *Condensing The Stack*

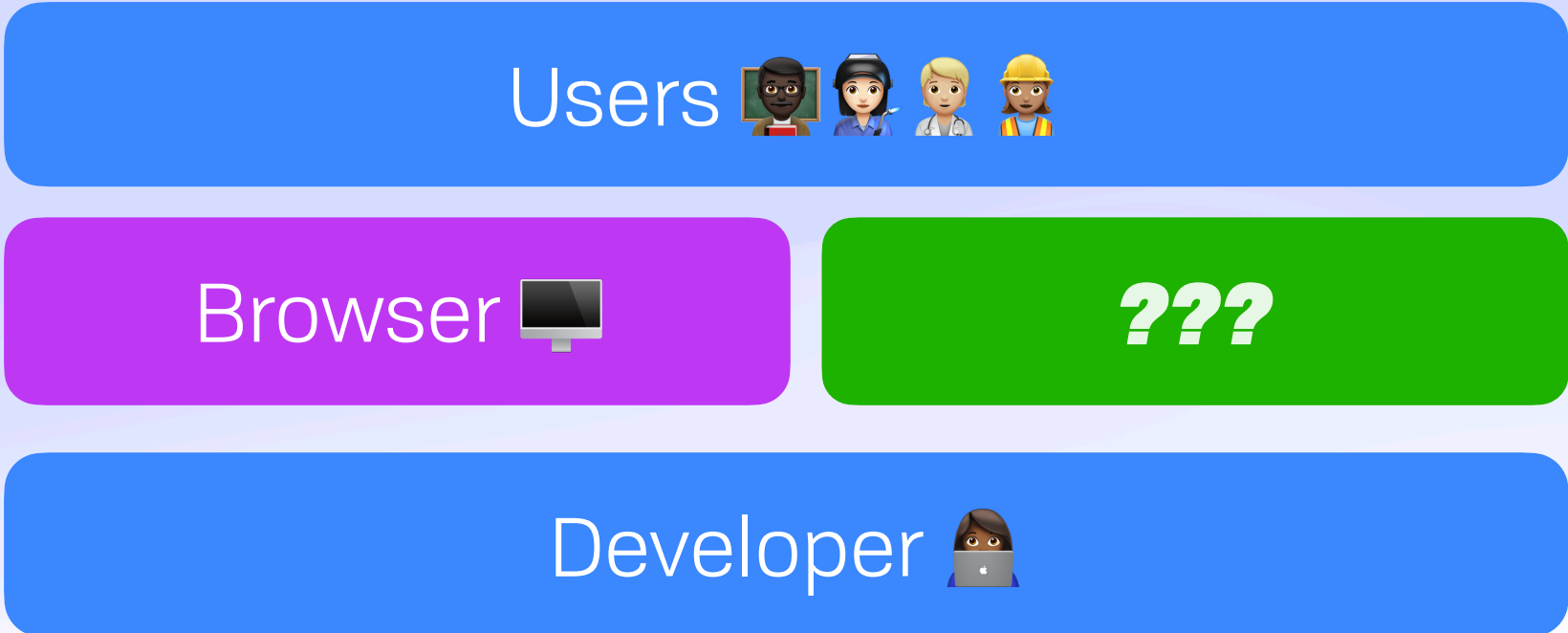
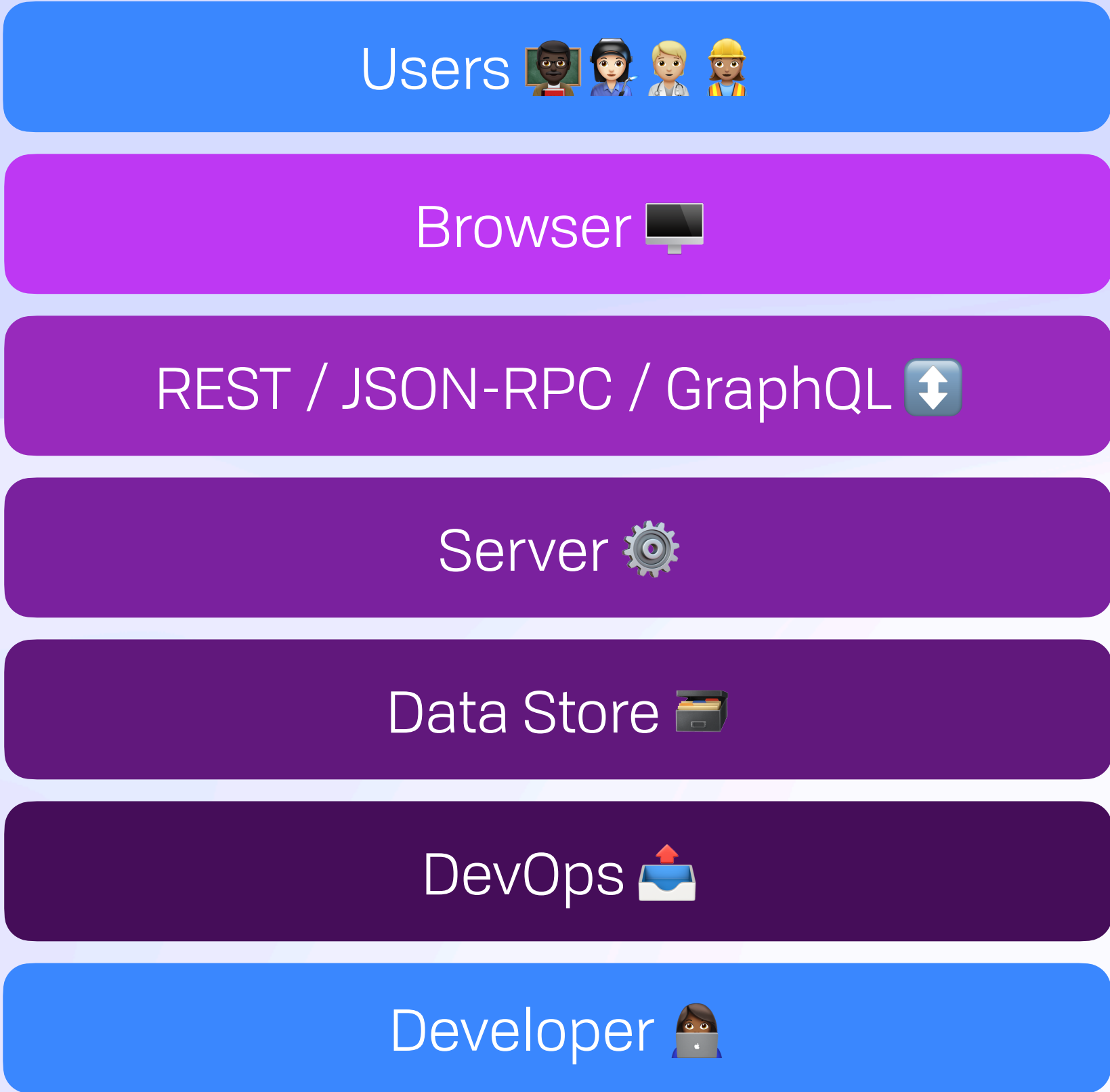
Have



# Economic Weight Class

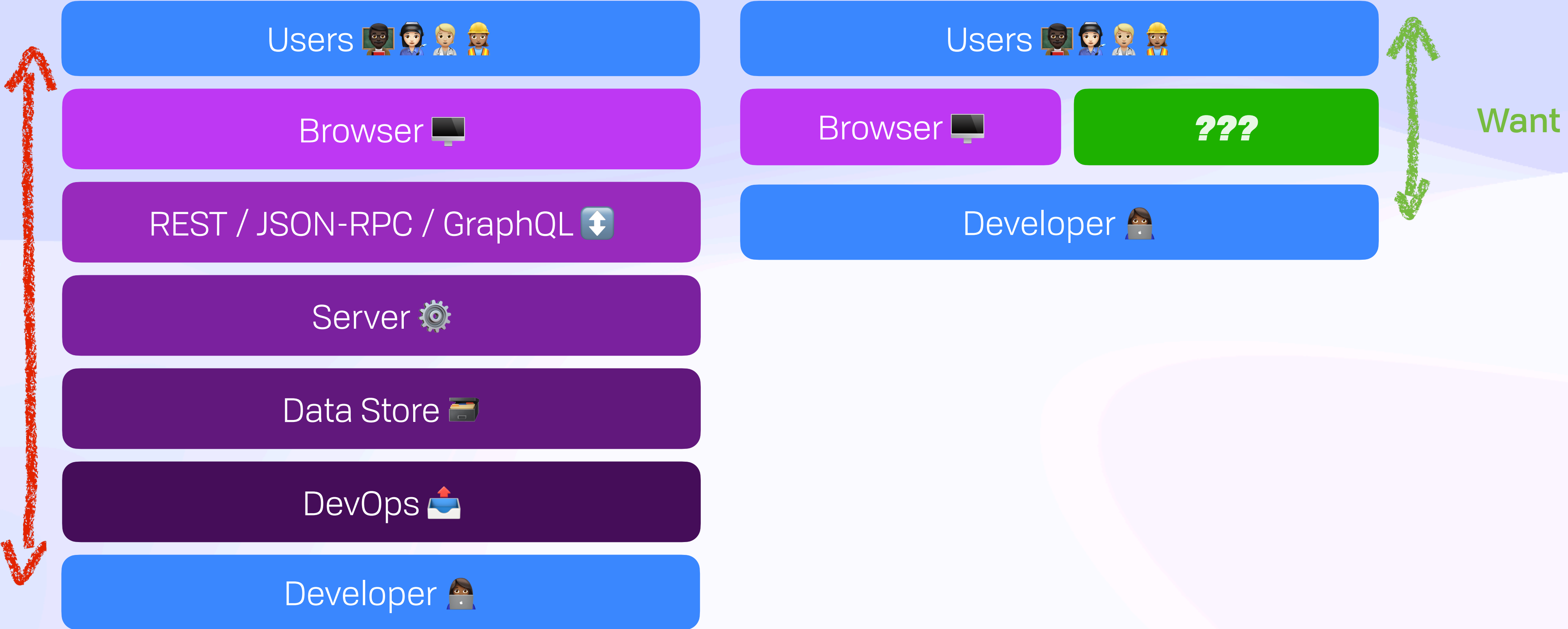
# Condensing The Stack

Have



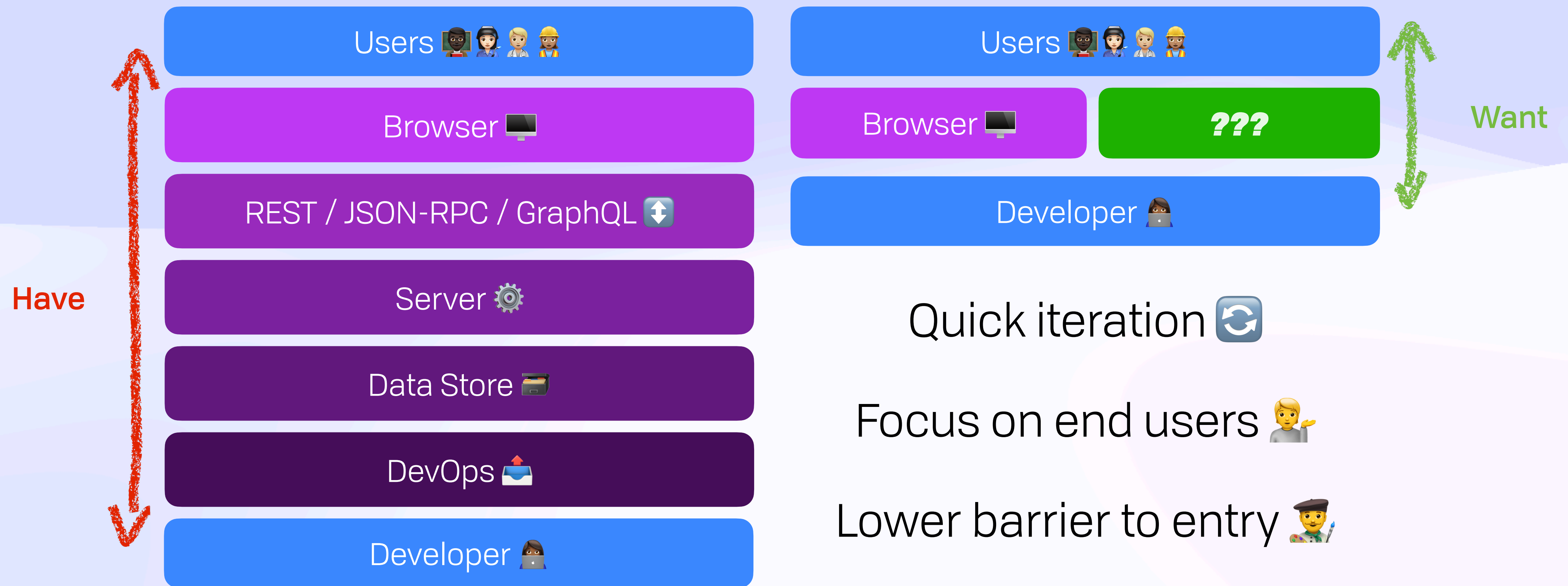
# Economic Weight Class

# Condensing The Stack



# Economic Weight Class

## *Condensing The Stack*





# Economic Weight Class

# *An Incredible Journey... Into a Dark Age*

The screenshot shows the 'Killed by Google' website interface. At the top, there is a search bar and a filter dropdown menu currently set to 'All (295)'. Below the search bar, a list of discontinued products is displayed, each with an icon, a title, a date range, a category tag, and a brief description.

| Product           | Date           | Category | Description  |
|-------------------|----------------|----------|--|
| Google Jamboard   | December 2024  | App      | Off to the glue factory in 5 months, Google Jamboard was a web and native whiteboard app that offered a rich collaborative experience. It will be about 8 years old.   |
| Jamboard          | September 2024 | Hardware | Scheduled to be killed in 2 months, Jamboard was a digital 4K touchscreen whiteboard device that allowed to collaborate using Google Workspace services. It will be over 7 years old.  |
| VPN by Google One | 2020 - 2024    | Service  | Killed about 1 month ago, VPN by Google One was a virtual private network service that provided users encrypted transit of their data and network activity and allowed them to mask their IP address. It was over 3 years old. |
| DropCam           | 2009 - 2024    | Hardware | Killed 4 months ago, Dropcam was a line of Wi-Fi video streaming cameras acquired by Google in 2014. It was about 15 years old.  |
| Google Podcasts   | 2018 - 2024    | App      | Killed 4 months ago, Google Podcasts was a podcast hosting platform and an Android podcast listening app. It was almost 6 years old.   |

All (295)

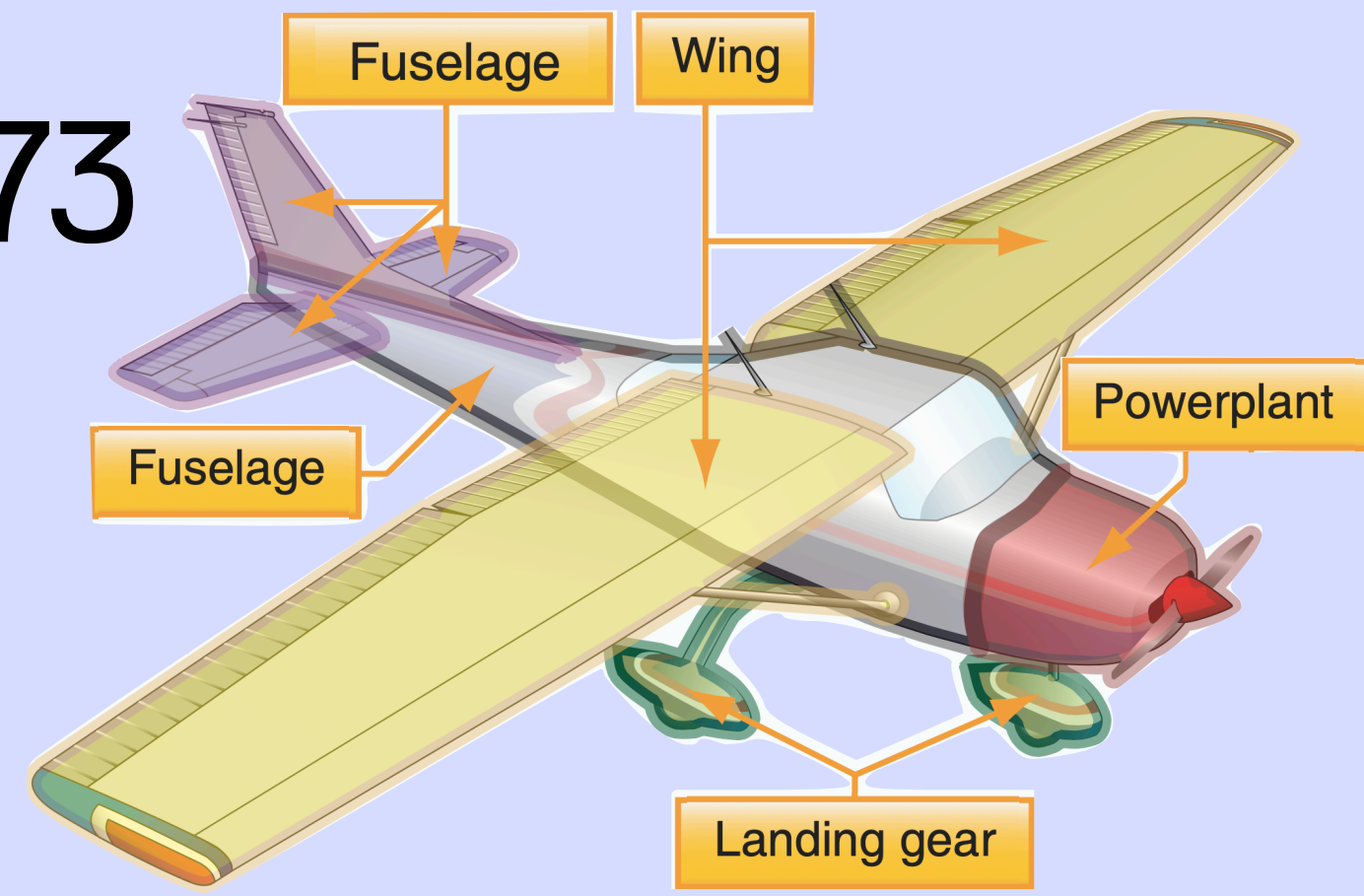
Economic Weight Class

***Functional Fixedness***

Economic Weight Class

# *Functional Fixedness*

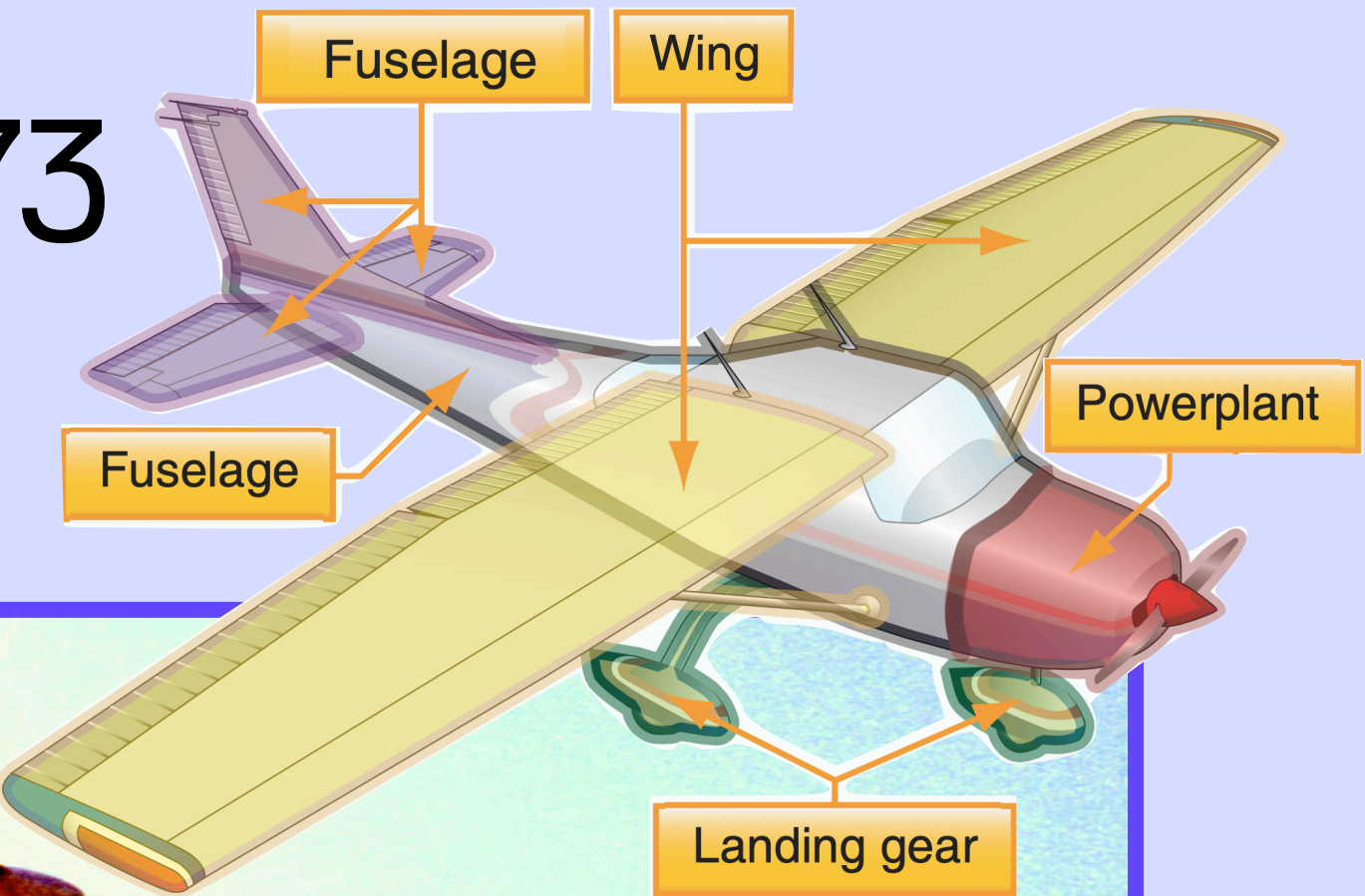
1959 - 1973



Economic Weight Class

# Functional Fixedness

1959 - 1973

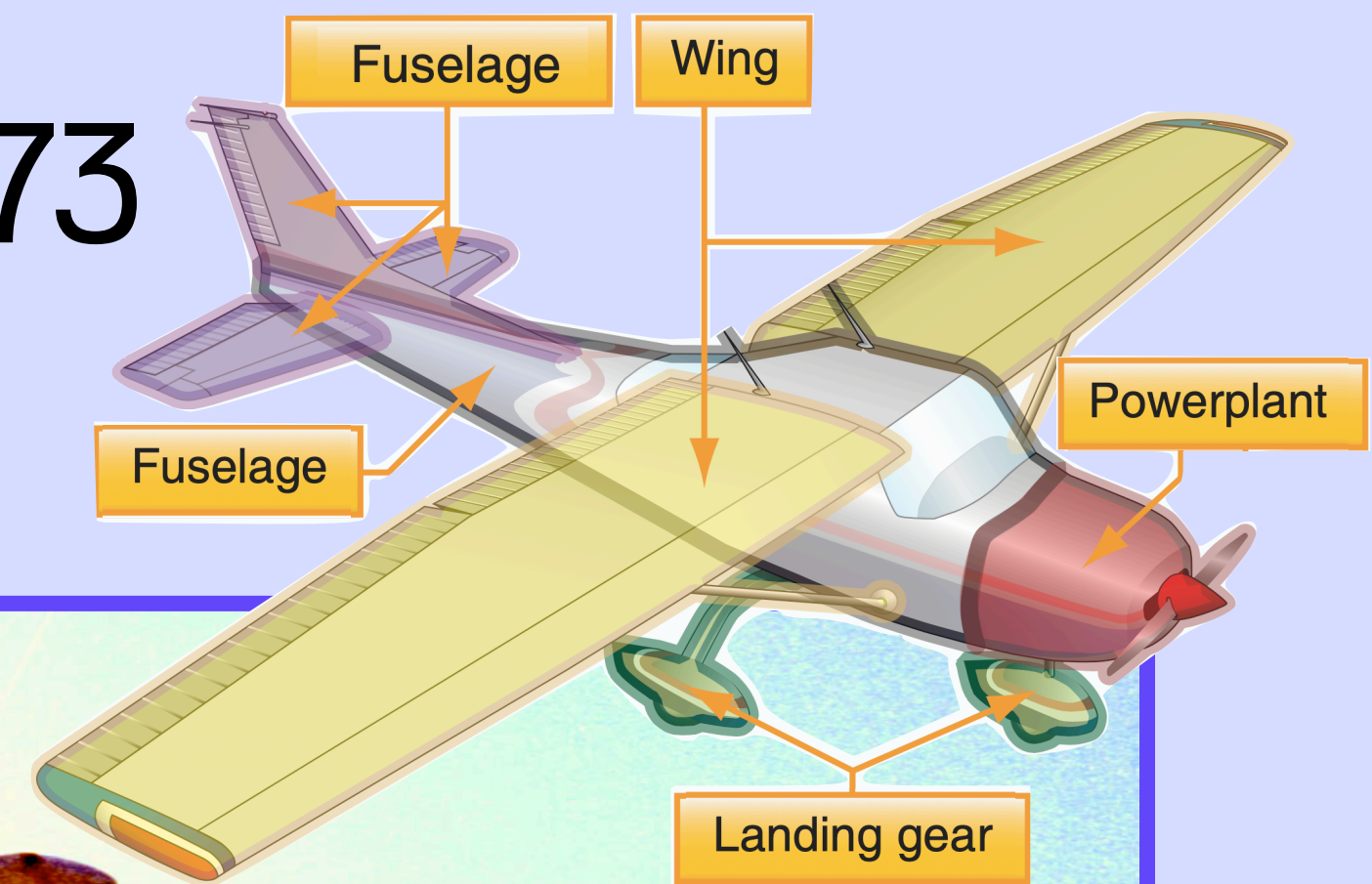


Economic Weight Class

# *Functional Fixedness*

"Everyone else was trying to make an airplane. We were trying to do human powered flight!"

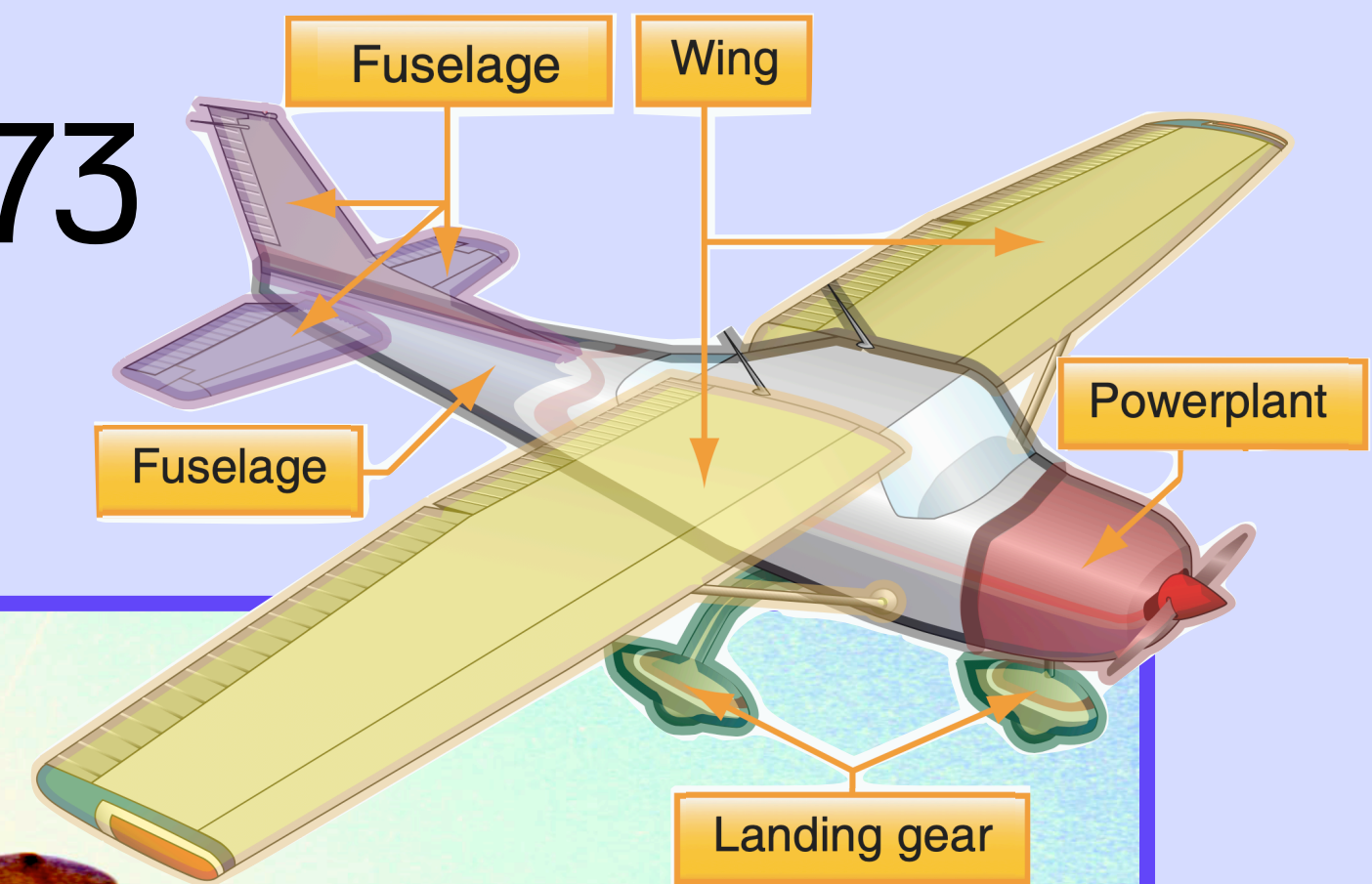
1959 - 1973



Economic Weight Class

# *Functional Fixedness*

1959 - 1973



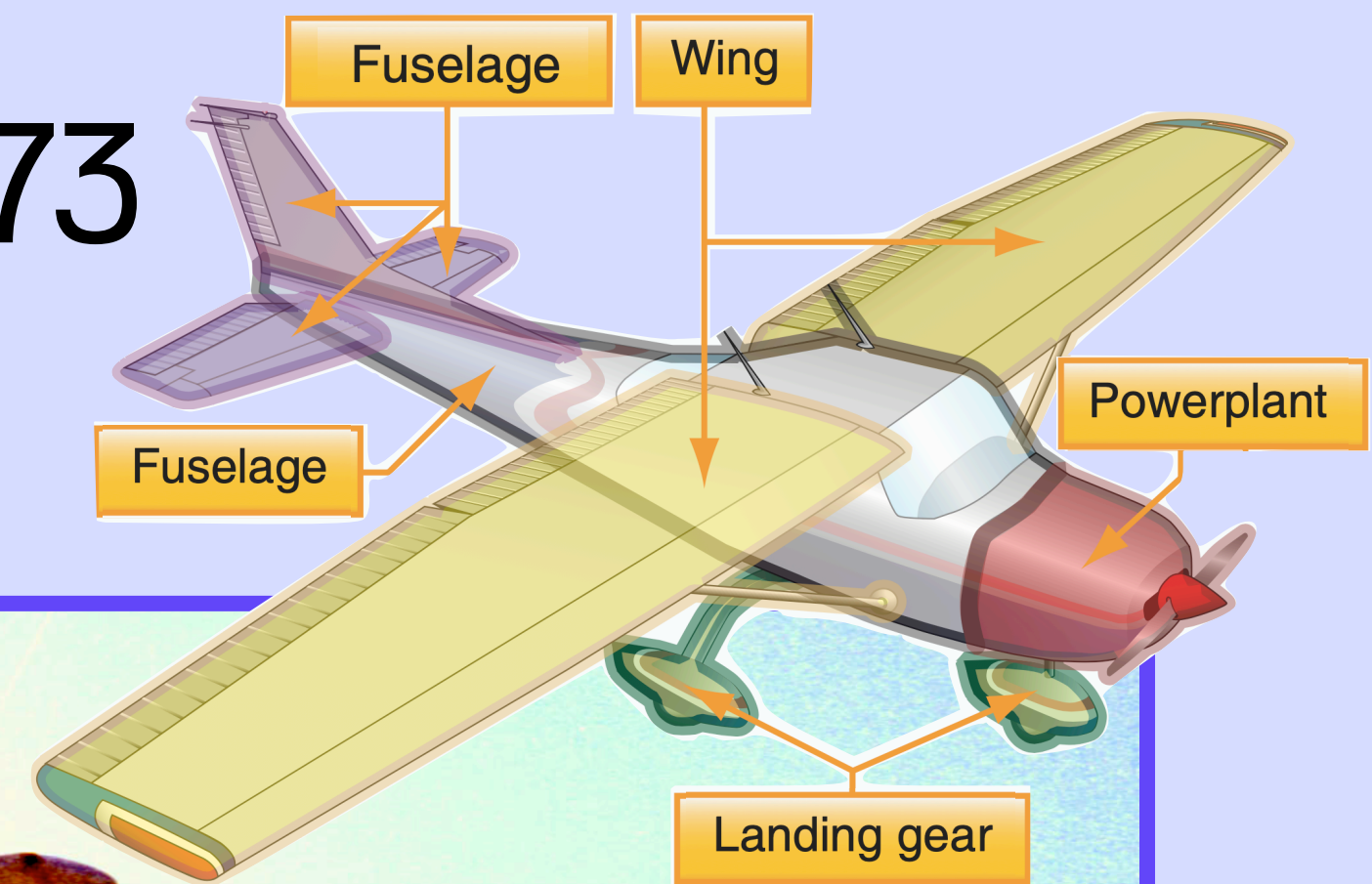
"Everyone else  
was trying to  
make **Google**.  
We were trying to  
do human  
powered flight!"



Economic Weight Class

# *Functional Fixedness*

1959 - 1973



"Everyone else  
was trying to  
make **Google**.  
We were trying to

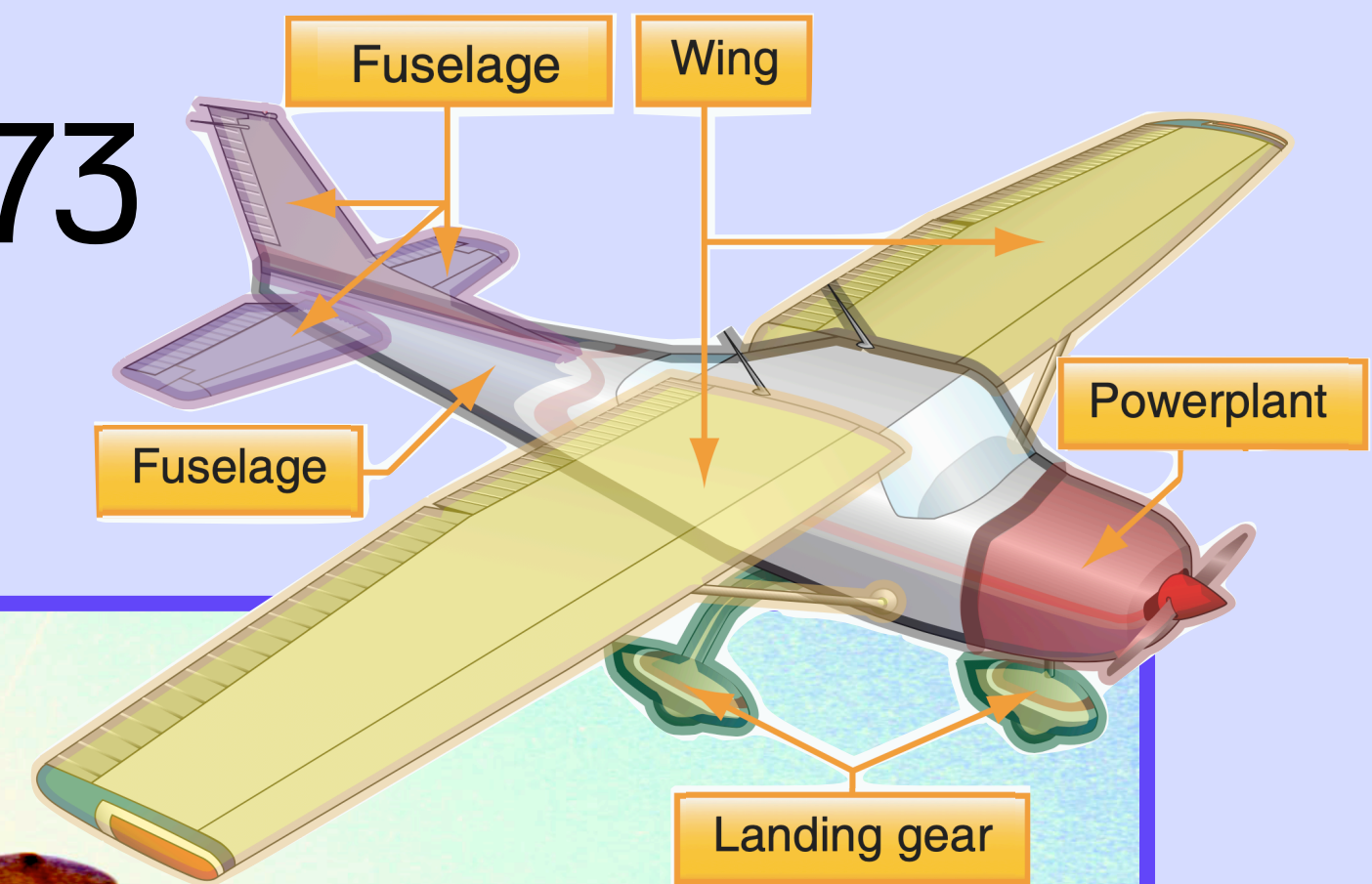
**build apps**"



Economic Weight Class

# *Functional Fixedness*

1959 - 1973



"Everyone else  
was trying to  
make **Google**.  
We were trying to

**solve user  
problems**"





# Economic Weight Class

## *Human*

A bike  
Crew: 1  
Cost: <\$8.5 billion (USD)



Economic Weight Class

***Command & Control***

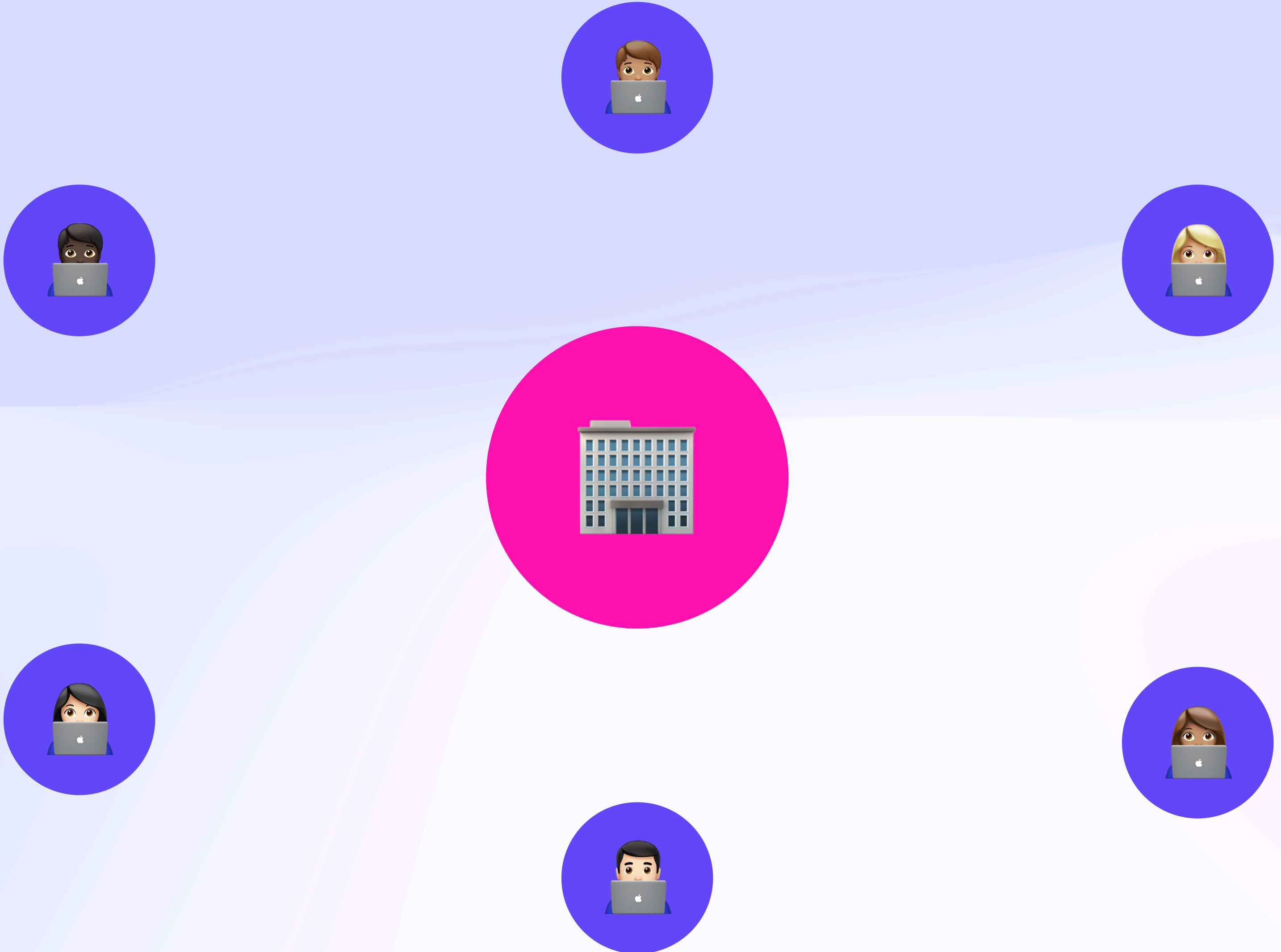
Economic Weight Class

***Command & Control***



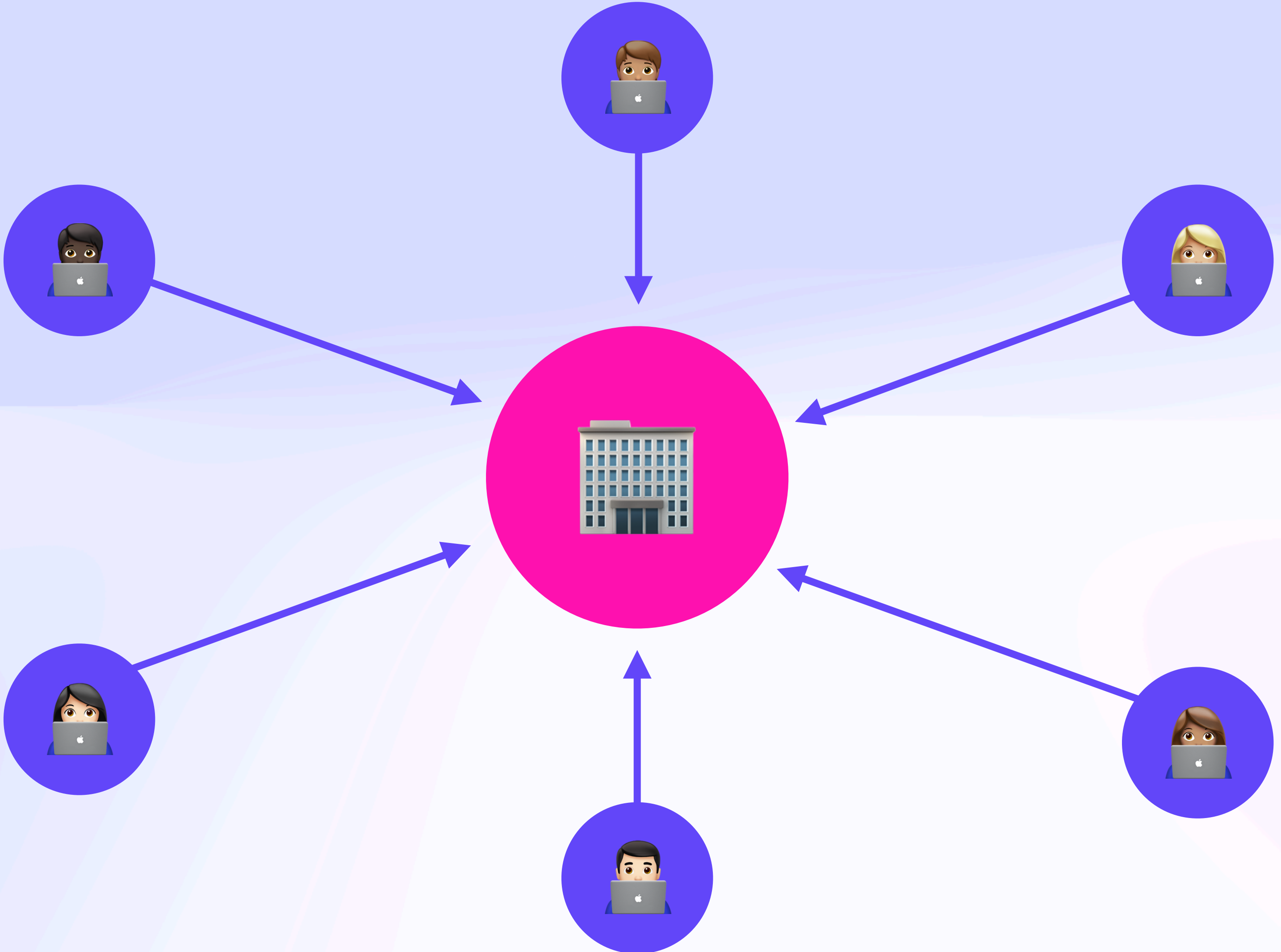
Economic Weight Class

# *Command & Control*



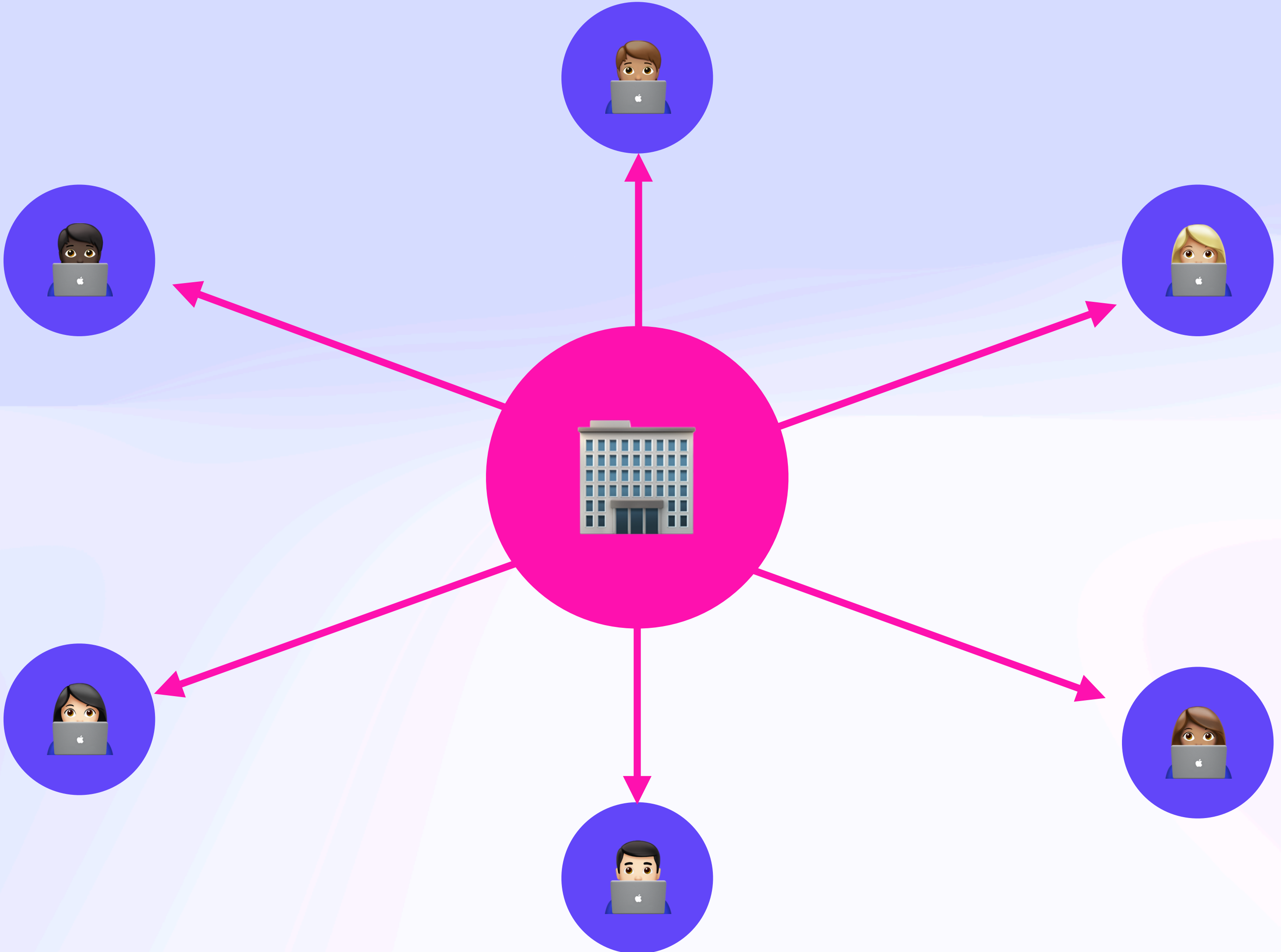
Economic Weight Class

# *Command & Control*



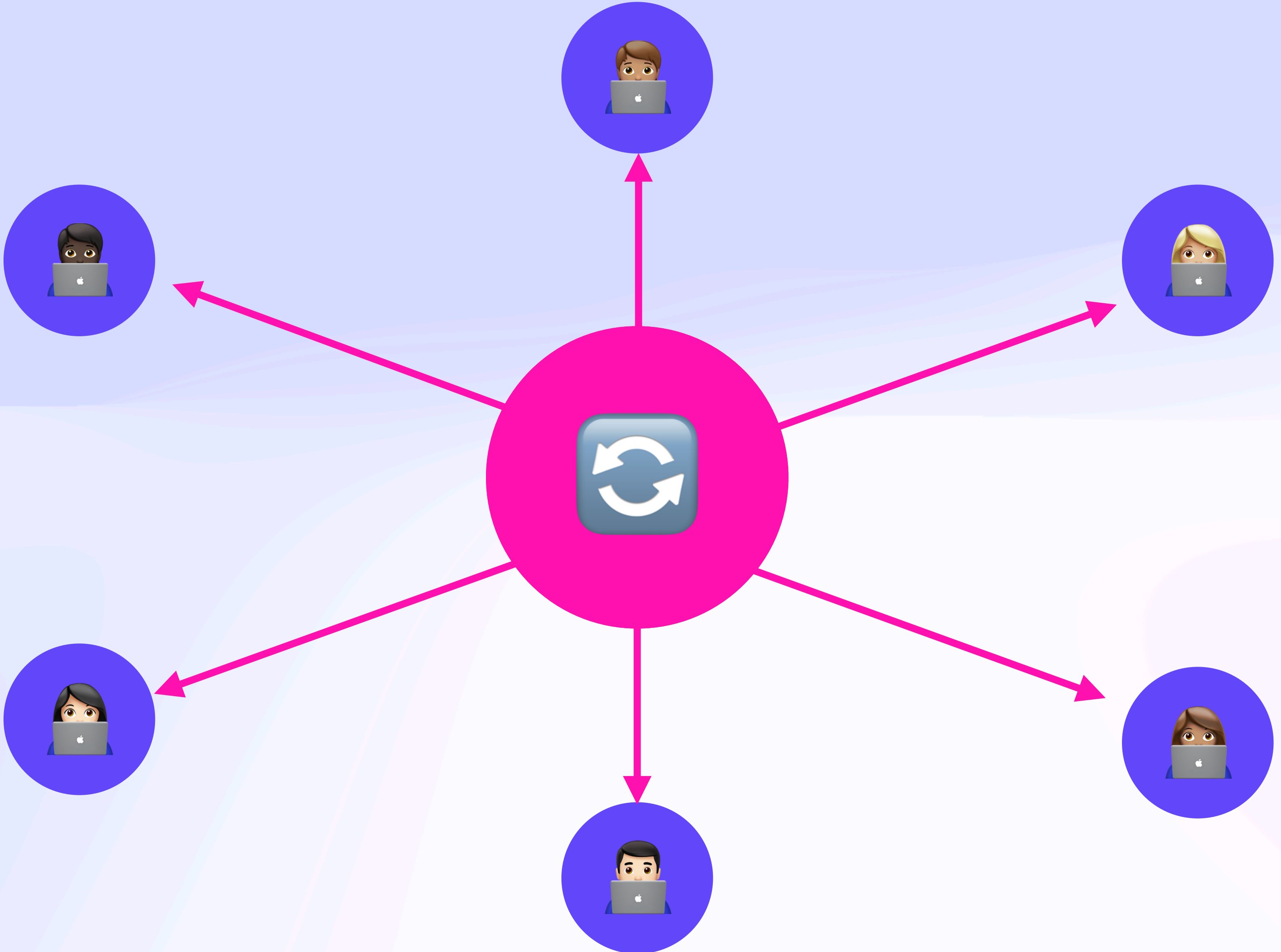
# Economic Weight Class

## *Hive Dynamics*



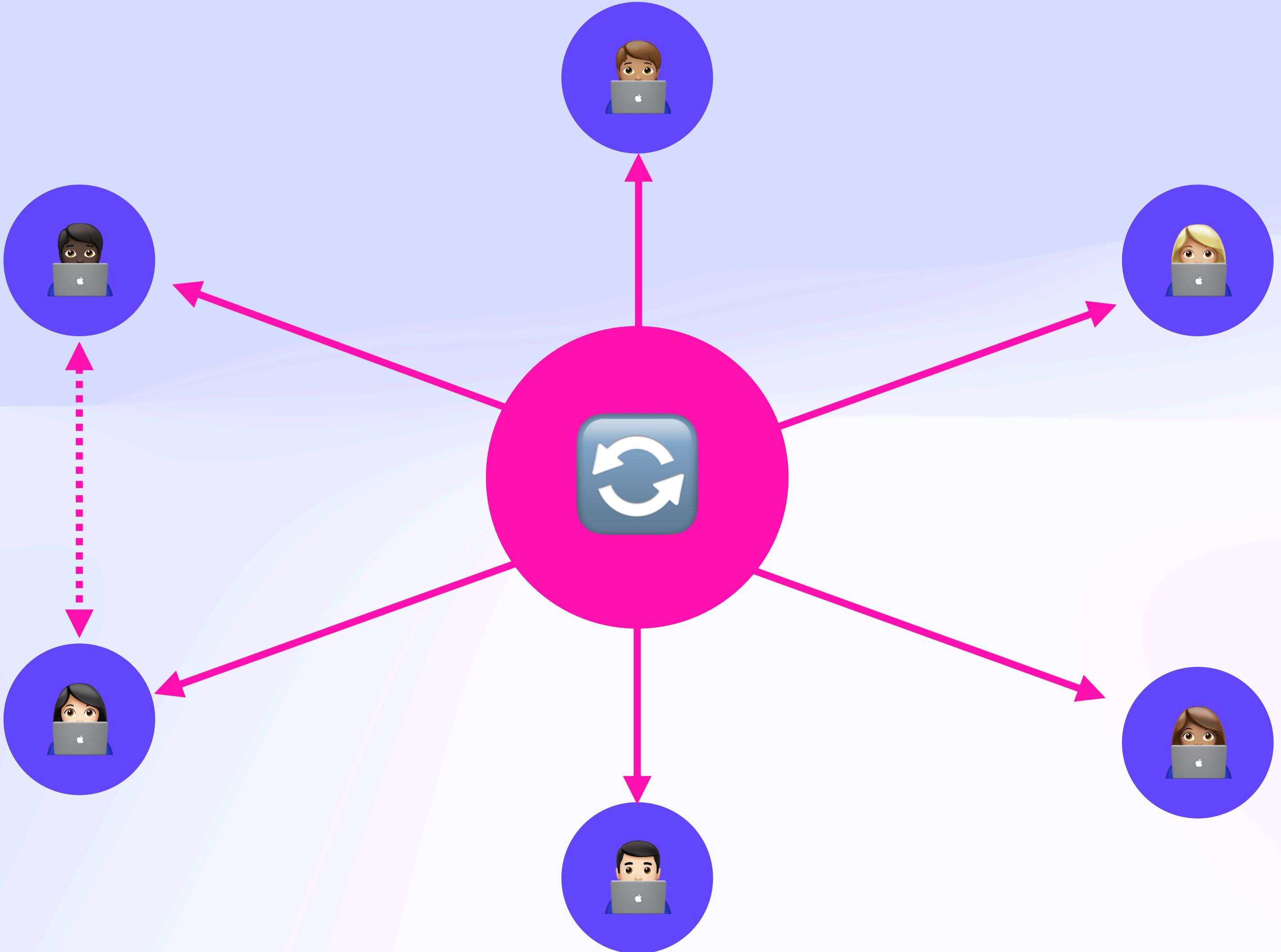
# Economic Weight Class

## *Hive Dynamics*



# Economic Weight Class

## *Hive Dynamics*





Economic Weight Class

# ***Solving Lamport's Problem***

Economic Weight Class

# ***Solving Lamport's Problem***

A distributed system is one in which the failure of a computer you didn't even know existed can ***render your own computer unusable***

— Leslie Lamport

Economic Weight Class

## ***Solving Lamport's Problem***

A distributed system is one in which the failure of a computer you didn't even know existed can ***render your own computer unusable***

— Leslie Lamport

...by this definition,

***LoFi isn't a distributed system*** 🤔

— Martin Kleppmann

Economic Weight Class

# ***A Litmus Test***

...by this definition,

***LoFi isn't a distributed system*** 

— Martin Kleppmann

Economic Weight Class

# ***What If Computing Were Interpersonal?***

# Economic Weight Class

# ***What If Computing Were Interpersonal?***

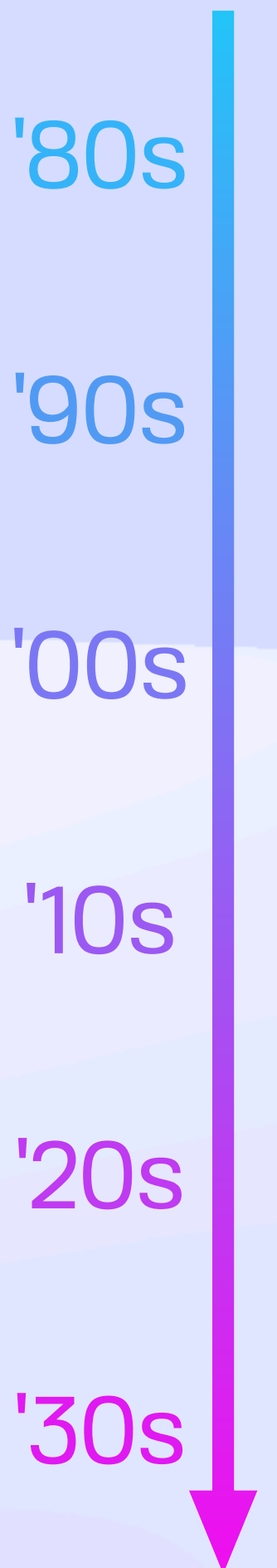


Image credit: <https://bestofjs.org/projects/tldraw>

# Economic Weight Class

## ***What If Computing Were Interpersonal?***

'80s

'90s

'00s

'10s

'20s

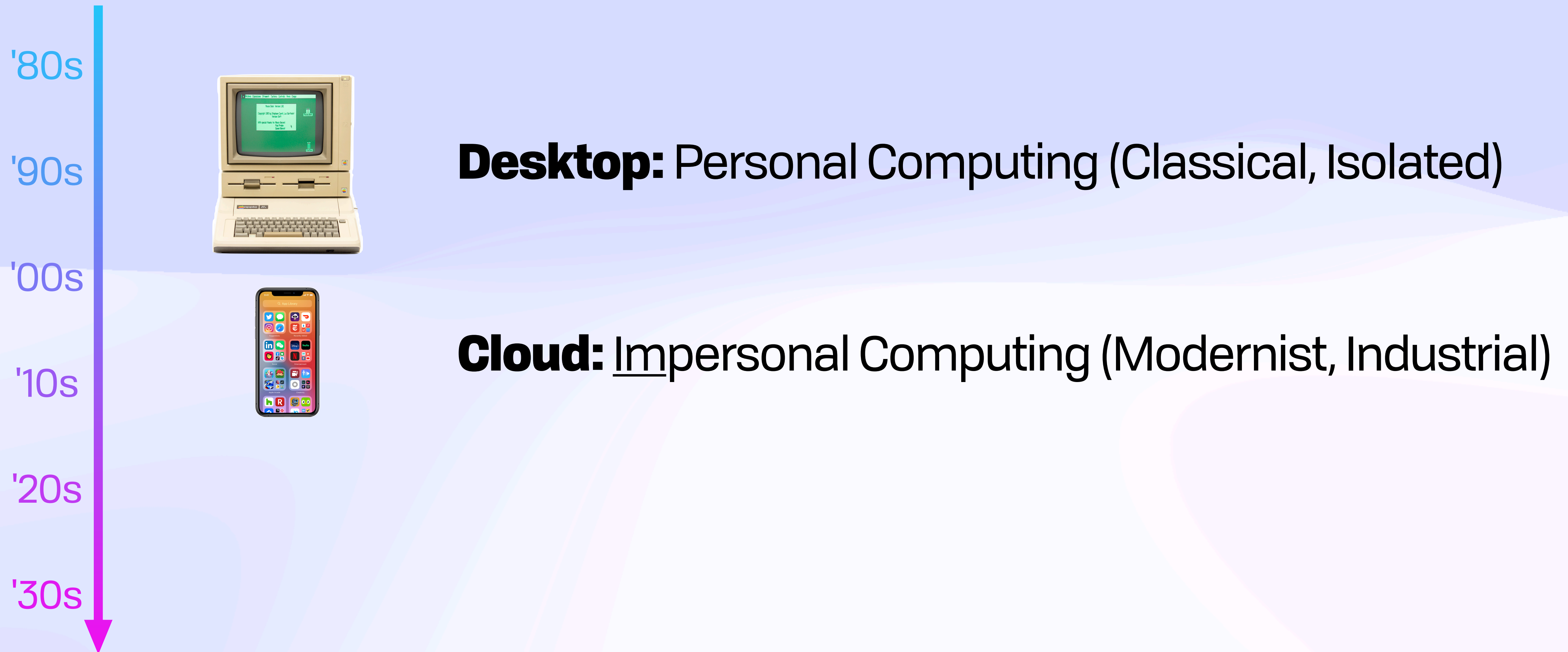
'30s



**Desktop:** Personal Computing (Classical, Isolated)

# Economic Weight Class

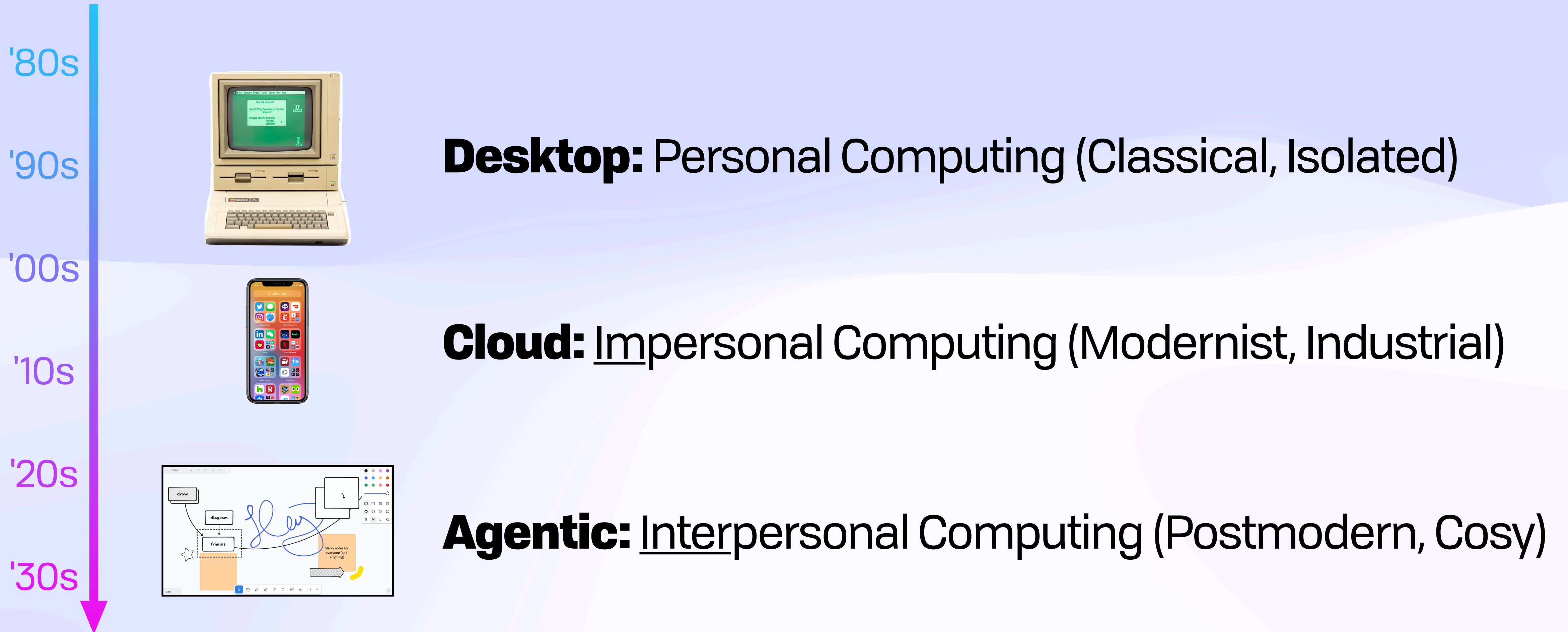
## *What If Computing Were Interpersonal?*





# Economic Weight Class

## *What If Computing Were Interpersonal?*



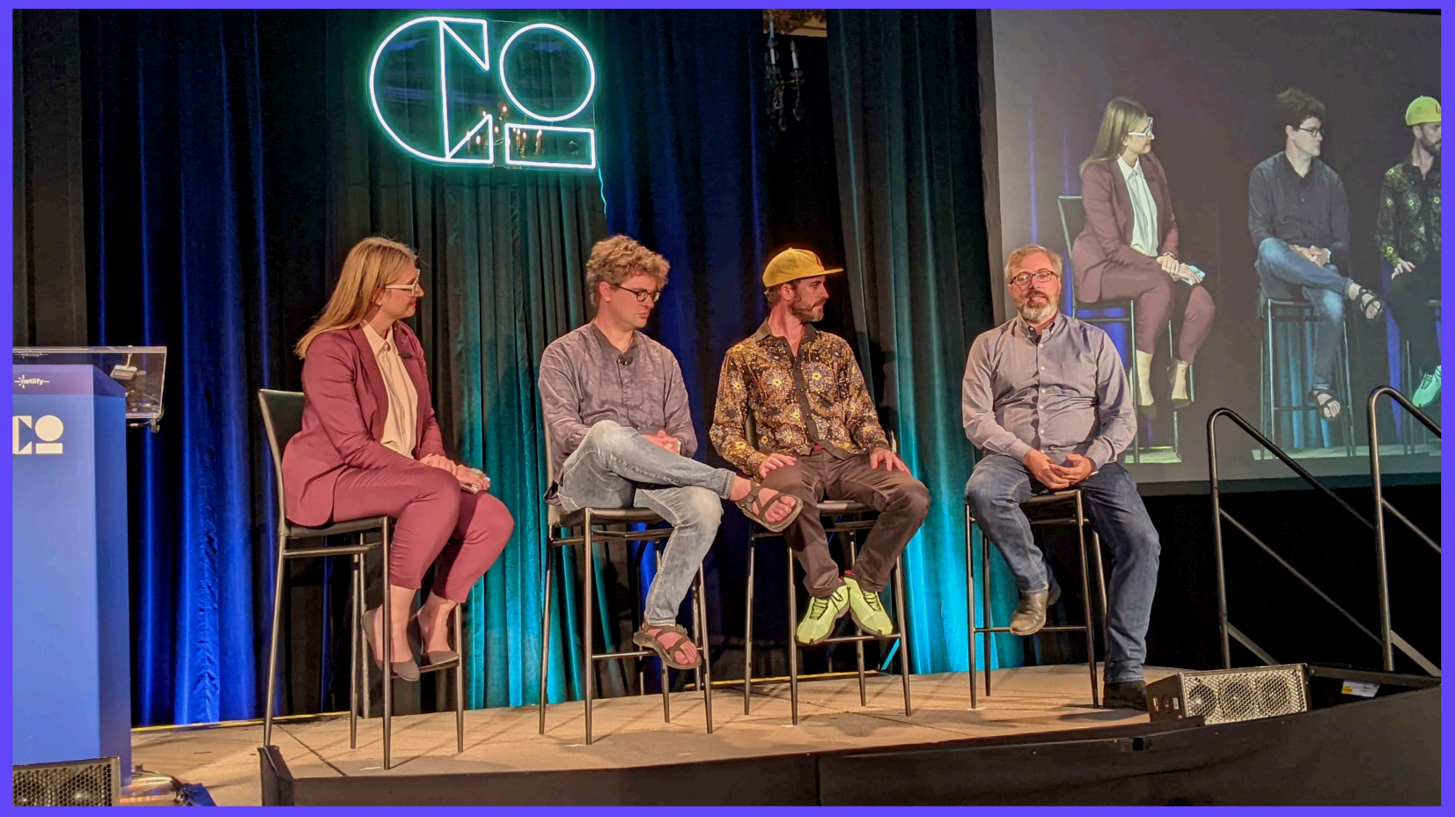
Finally...

# *A New Hope*



A New Hope

# *Early But Exciting*



# A New Hope

## *Early But Exciting*



VIDEO: ARDHIRA PUTRA

GREGORY BARBER BACKCHANNEL AUG 3, 2023 6:00 AM

### The Cloud Is a Prison. Can the Local-First Software Movement Set Us Free?



A New Hope

# Early But Exciting



VIDEO: ARDHIRA PUTRA

GREGORY BARBER BACKCHANNEL AUG 3, 2023 6:00 AM

## The Cloud Is a Prison. Can the Local-First Software Movement Set Us Free?



A New Hope

# Early But Exciting

A new breed of apps.

## Local-first

A fully local database provides an instant-on app experience, especially when combined with PWA or hybrid app architecture.

Unabashedly local-first software

## Local-First

Set `clientPersistence: true` and data is also stored on the client, providing instant ("local-first") startup, navigation, and offline support.



The best way to  
**local-first**

Local-first sync for Postgres  
from the inventors of CRDTs

VIDEO: ARDHIRA PUTRA

GREGORY BARBER BACKCHANNEL AUG 3, 2023 6:00 AM

The Cloud Is a Prison. Can the Local-First Software Movement Set Us Free?



A New Hope

***In the Wild*** 

A New Hope

*In the Wild* 🦁



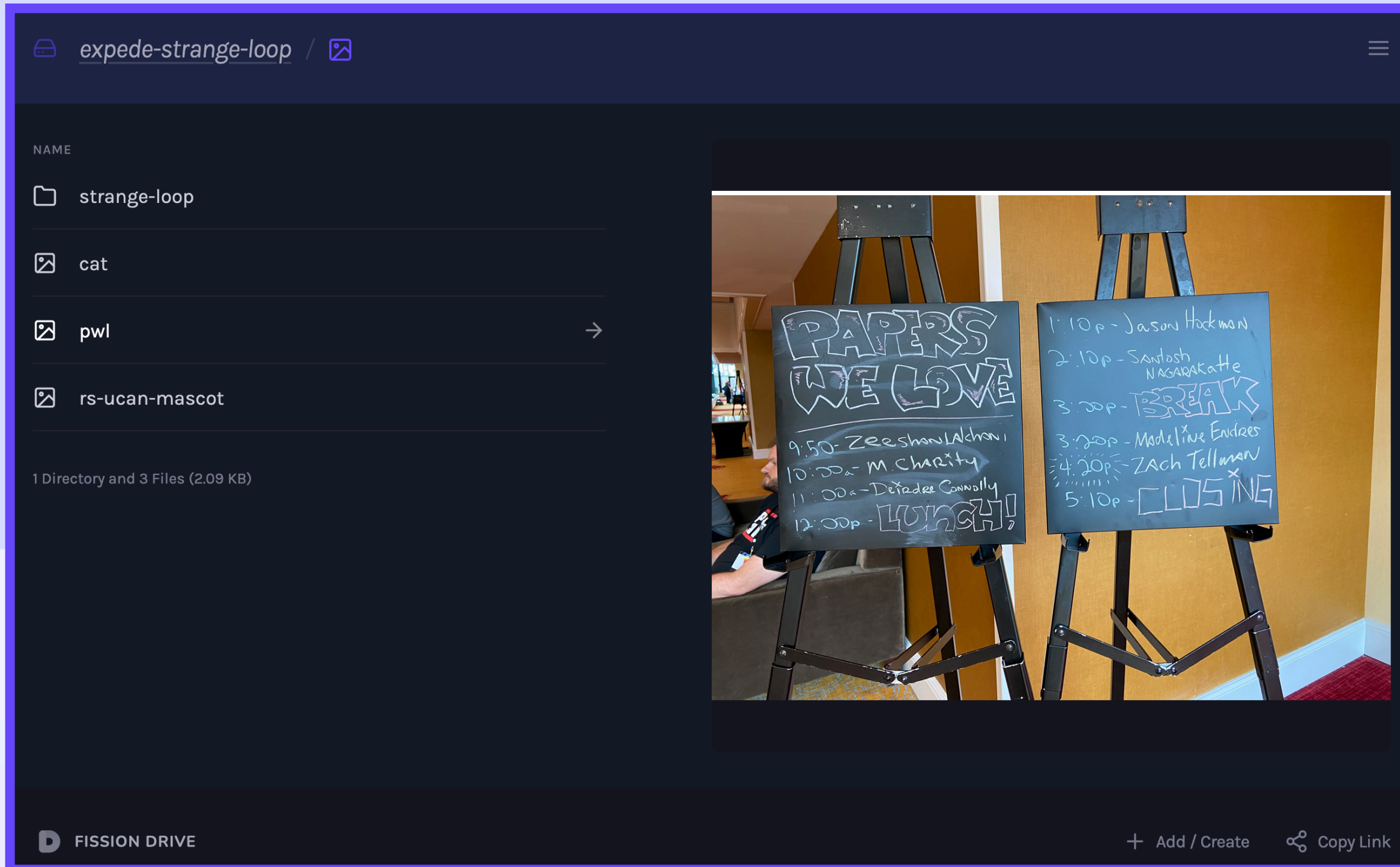


A New Hope

***In the Wild*** 

# A New Hope

# In the Wild



expede-strange-loop /

NAME

- strange-loop
- cat
- pwl
- rs-ucan-mascot

1 Directory and 3 Files (2.09 KB)

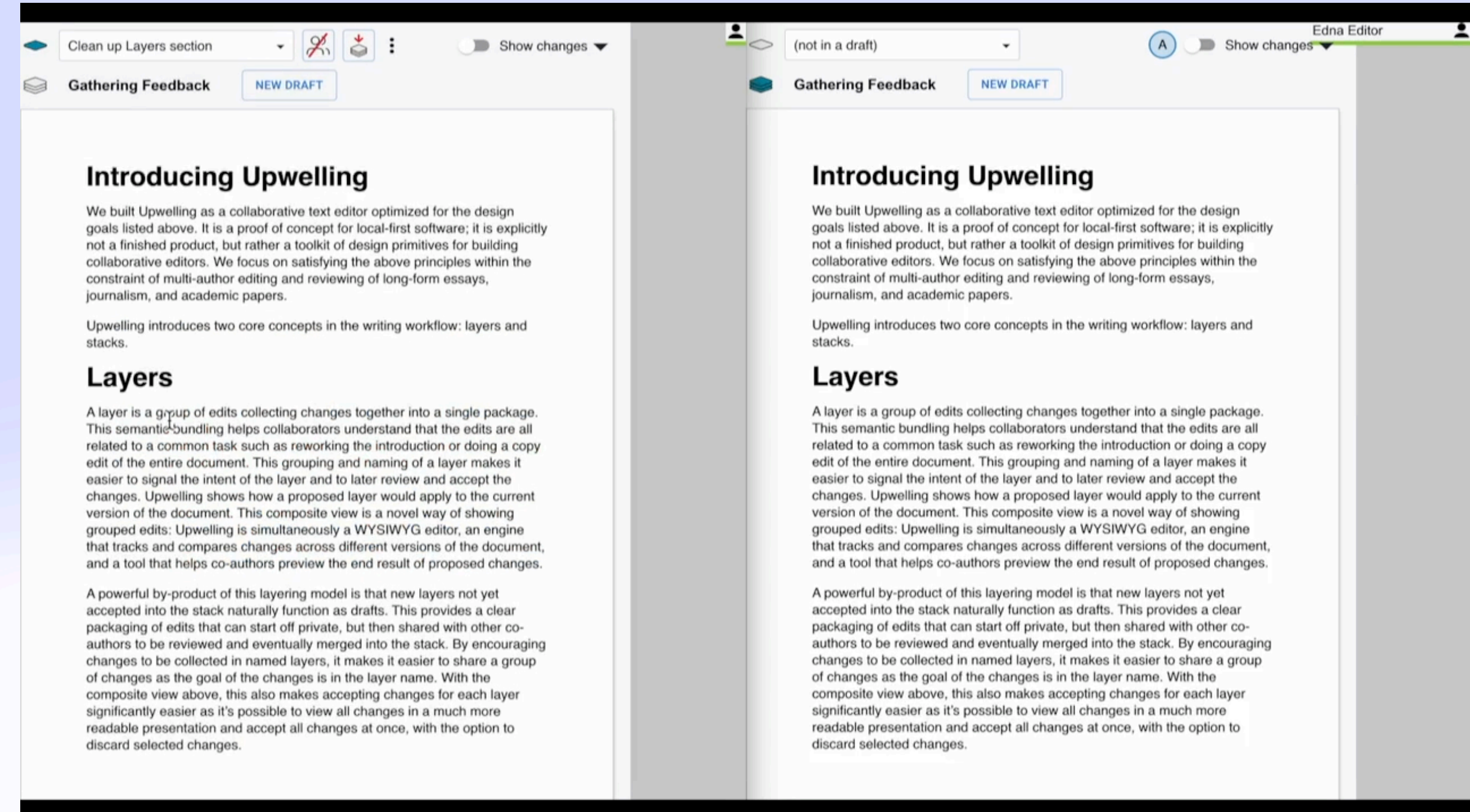
FISSION DRIVE

+ Add / Create Copy Link

**PAPERS WE LOVE**

9:50 - Zeeshan Akhori  
10:00 - M. Charity  
11:00 - Detazee Connolly  
12:00 - LOUNGE!

1:10p - Jason Hackman  
2:10p - Santosh NAGARAKATTE  
3:00p - BREAK  
3:20p - Madeline Endres  
4:20p - Zach Tellman  
5:10p - CLOSING



Clean up Layers section Show changes

Gathering Feedback NEW DRAFT

## Introducing Upwelling

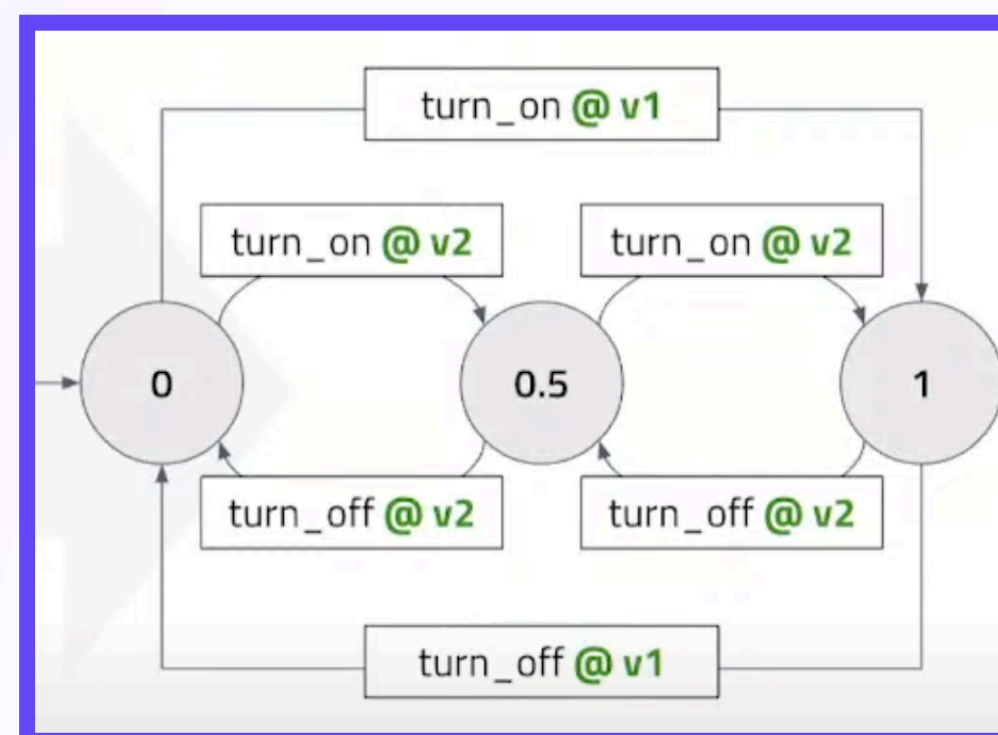
We built Upwelling as a collaborative text editor optimized for the design goals listed above. It is a proof of concept for local-first software; it is explicitly not a finished product, but rather a toolkit of design primitives for building collaborative editors. We focus on satisfying the above principles within the constraint of multi-author editing and reviewing of long-form essays, journalism, and academic papers.

Upwelling introduces two core concepts in the writing workflow: layers and stacks.

### Layers

A layer is a group of edits collecting changes together into a single package. This semantic bundling helps collaborators understand that the edits are all related to a common task such as reworking the introduction or doing a copy edit of the entire document. This grouping and naming of a layer makes it easier to signal the intent of the layer and to later review and accept the changes. Upwelling shows how a proposed layer would apply to the current version of the document. This composite view is a novel way of showing grouped edits: Upwelling is simultaneously a WYSIWYG editor, an engine that tracks and compares changes across different versions of the document, and a tool that helps co-authors preview the end result of proposed changes.

A powerful by-product of this layering model is that new layers not yet accepted into the stack naturally function as drafts. This provides a clear packaging of edits that can start off private, but then shared with other co-authors to be reviewed and eventually merged into the stack. By encouraging changes to be collected in named layers, it makes it easier to share a group of changes as the goal of the changes is in the layer name. With the composite view above, this also makes accepting changes for each layer significantly easier as it's possible to view all changes in a much more readable presentation and accept all changes at once, with the option to discard selected changes.



A New Hope

# ***Slicing the Problem***

## Seven ideals for local-first software

1. No spinners: your work at your fingertips
2. Your work is not trapped on one device
3. The network is optional
4. Seamless collaboration with your colleagues
5. The Long Now
6. Security and privacy by default
7. You retain ultimate ownership and control

A New Hope

# *Slicing the Problem*

## Seven ideals for local-first software

1. No spinners: your work at your fingertips
2. Your work is not trapped on one device
3. The network is optional
4. Seamless collaboration with your colleagues
5. The Long Now
6. Security and privacy by default
7. You retain ultimate ownership and control

A New Hope

# *Slicing the Problem*

## Seven ideals for local-first software

1. No spinners: your work at your fingertips
2. Your work is not trapped on one device
3. The network is optional
4. Seamless collaboration with your colleagues
5. The Long Now
6. Security and privacy by default
7. You retain ultimate ownership and control



A New Hope

# *Slicing the Problem*

## Seven ideals for local-first software

1. No spinners: your work at your fingertips
2. Your work is not trapped on one device
3. The network is optional
4. Seamless collaboration with your colleagues
5. The Long Now
6. Security and privacy by default
7. You retain ultimate ownership and control



A New Hope

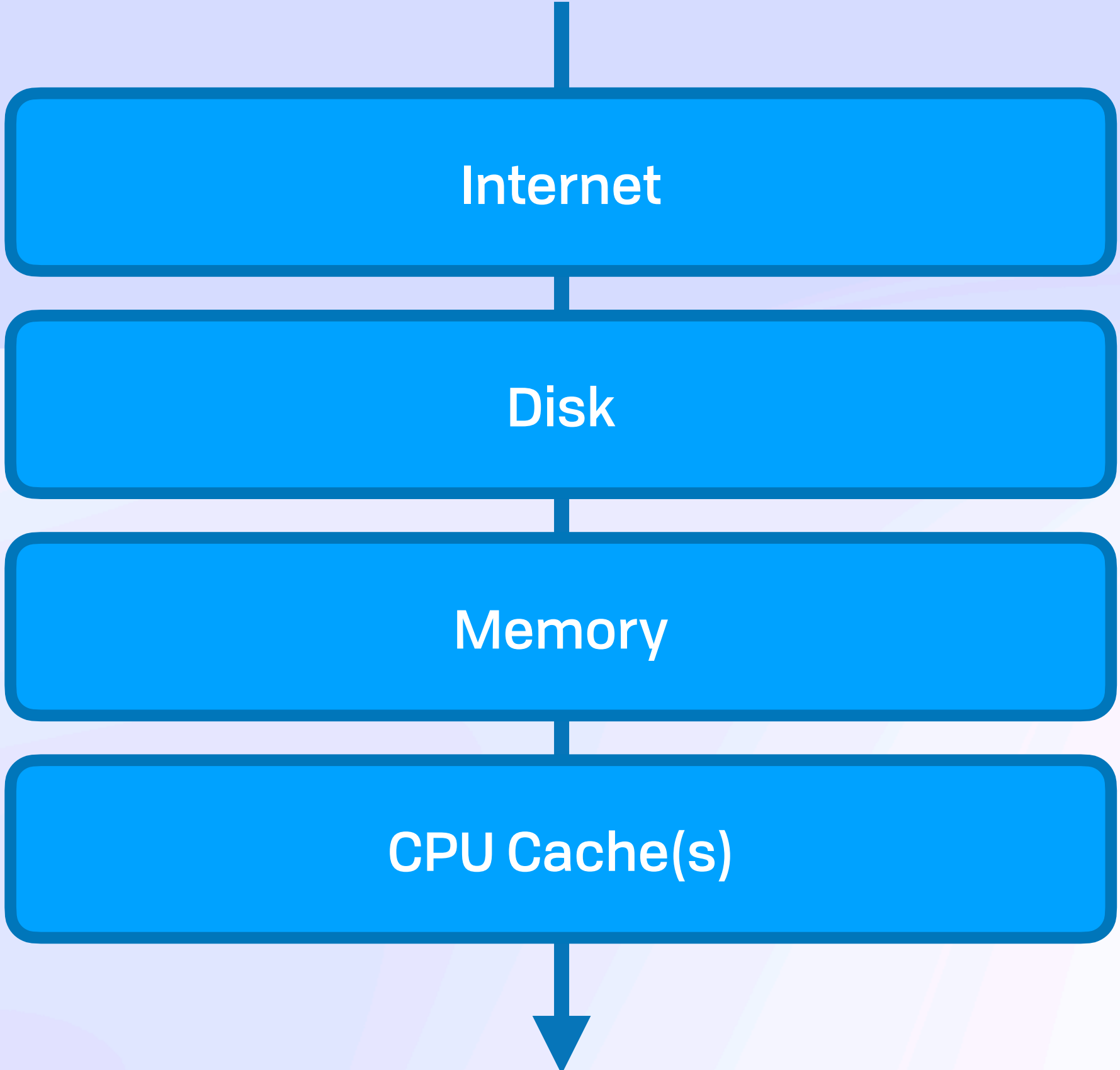
# ***New Metaphors***

**Layered / "Cache for the Internet"**

A New Hope

# *New Metaphors*

Layered / "Cache for the Internet"



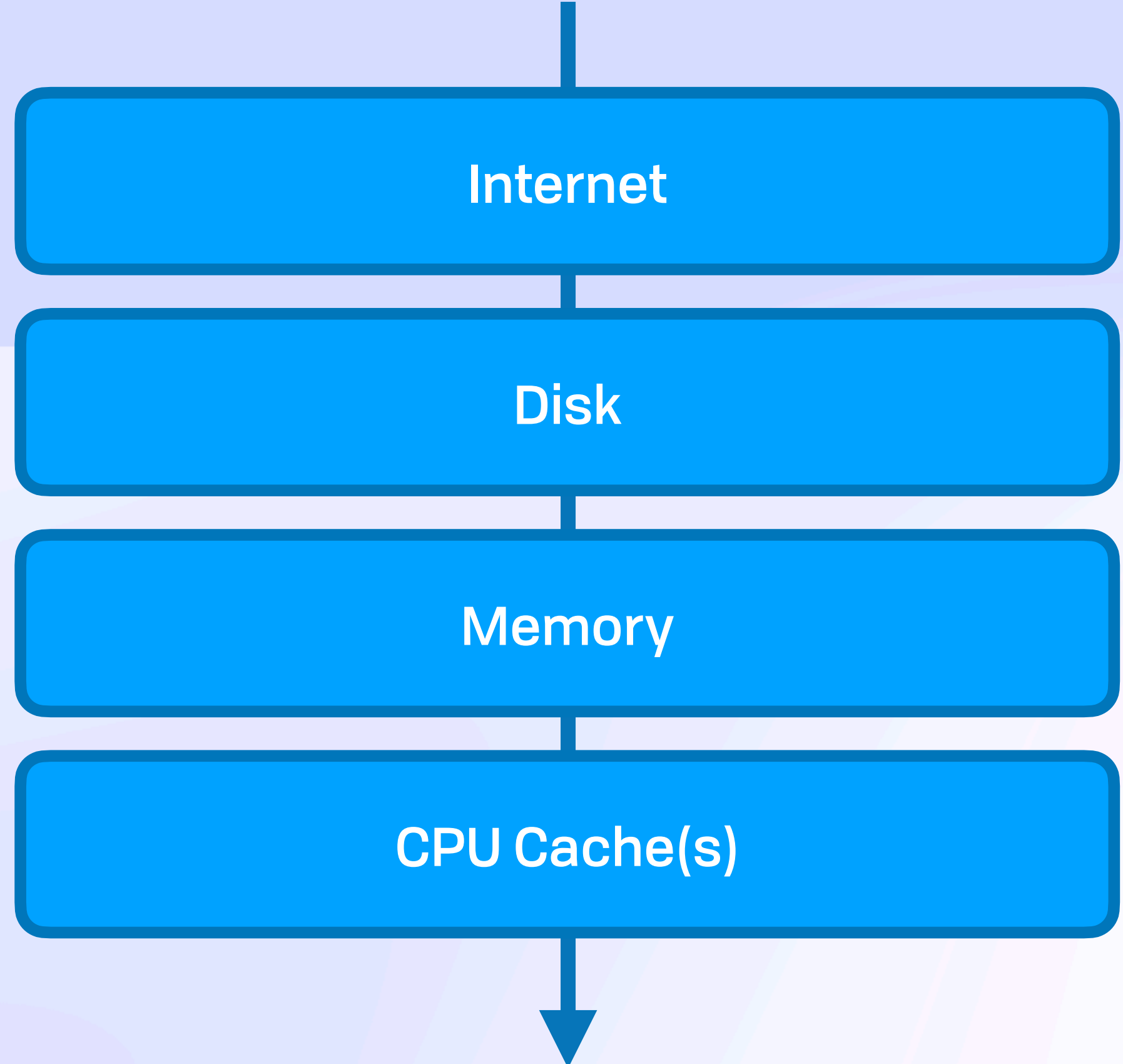


A New Hope

# *New Metaphors*

Layered / "Cache for the Internet"

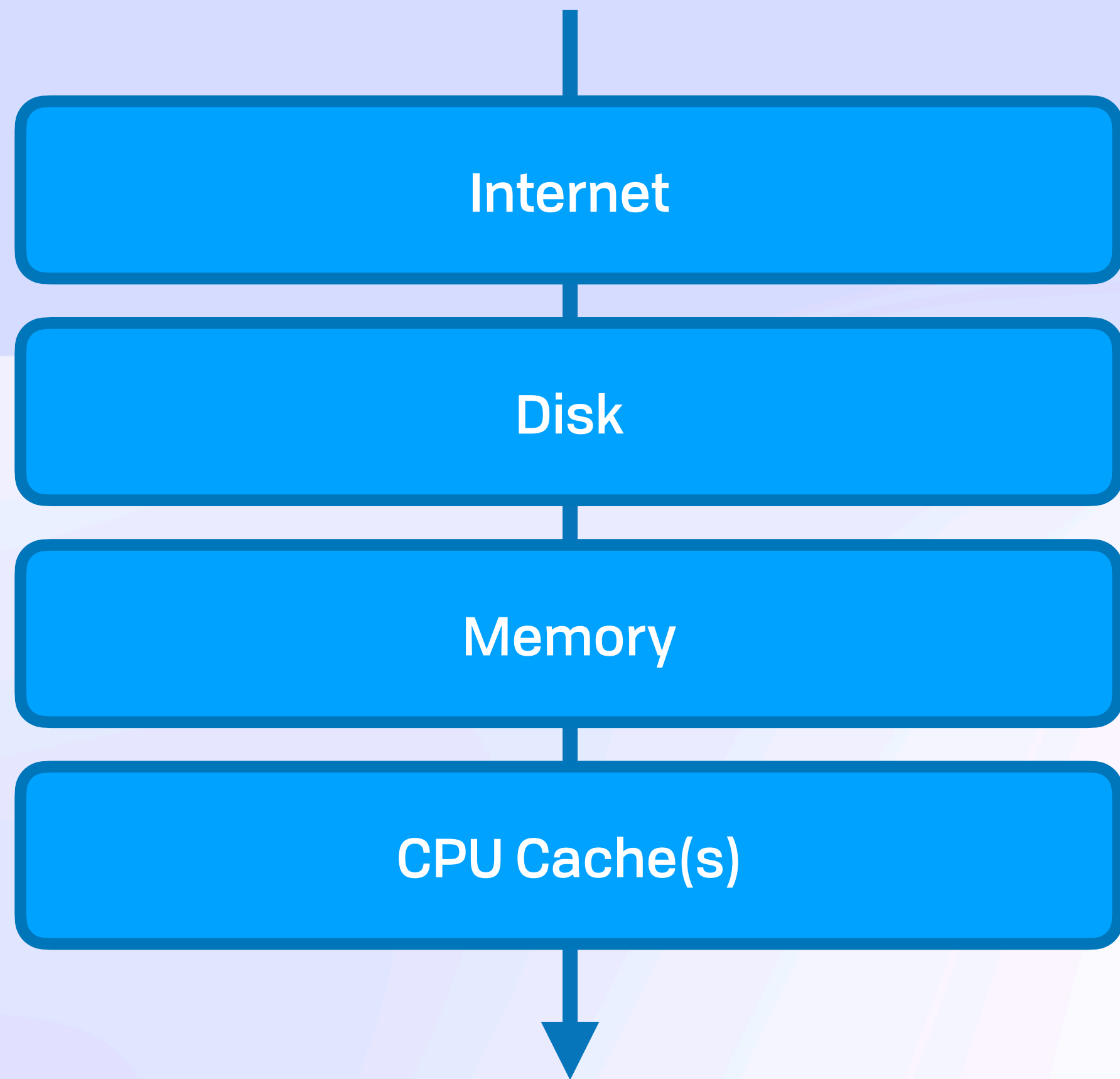
Cellular / P2P



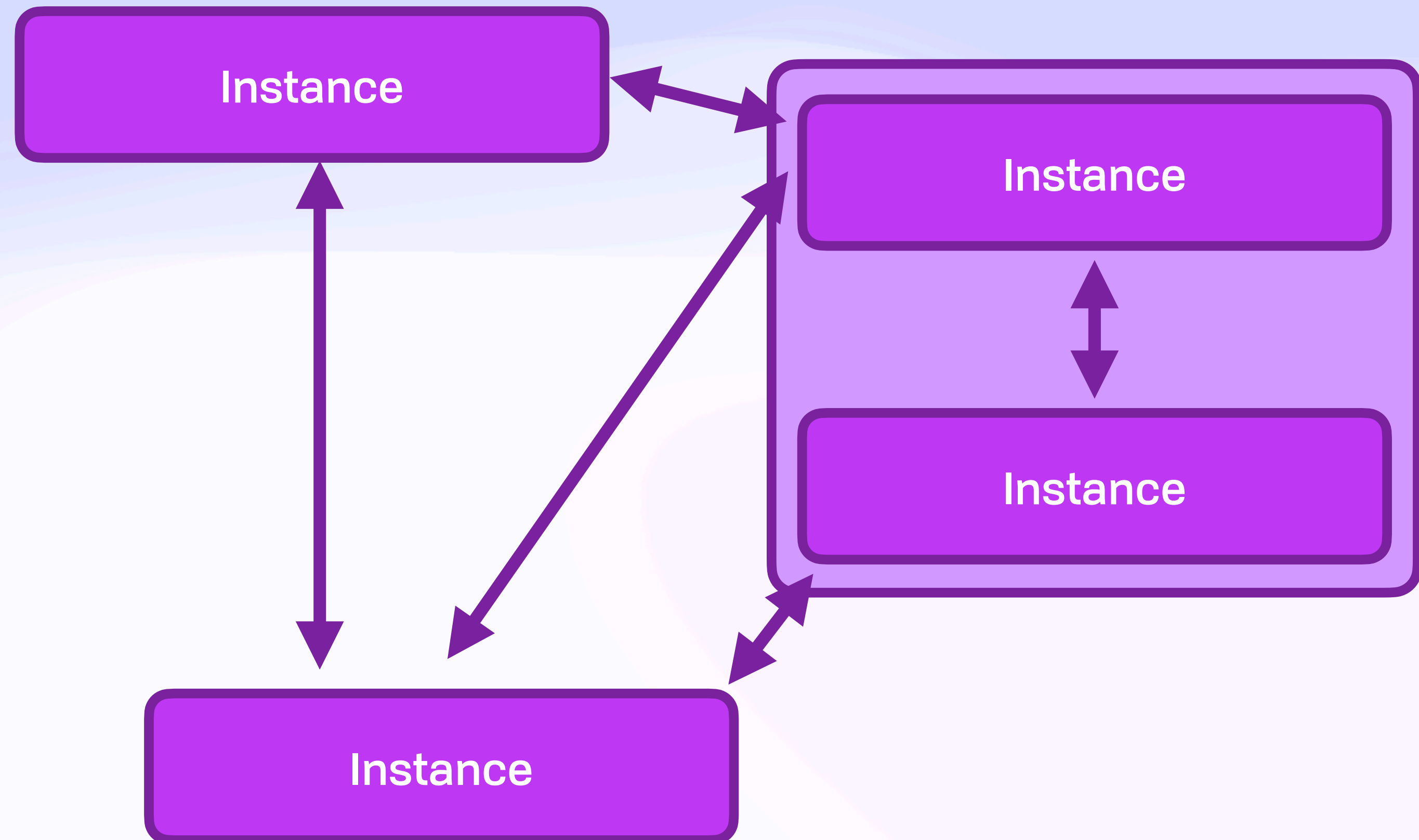
A New Hope

# *New Metaphors*

Layered / "Cache for the Internet"



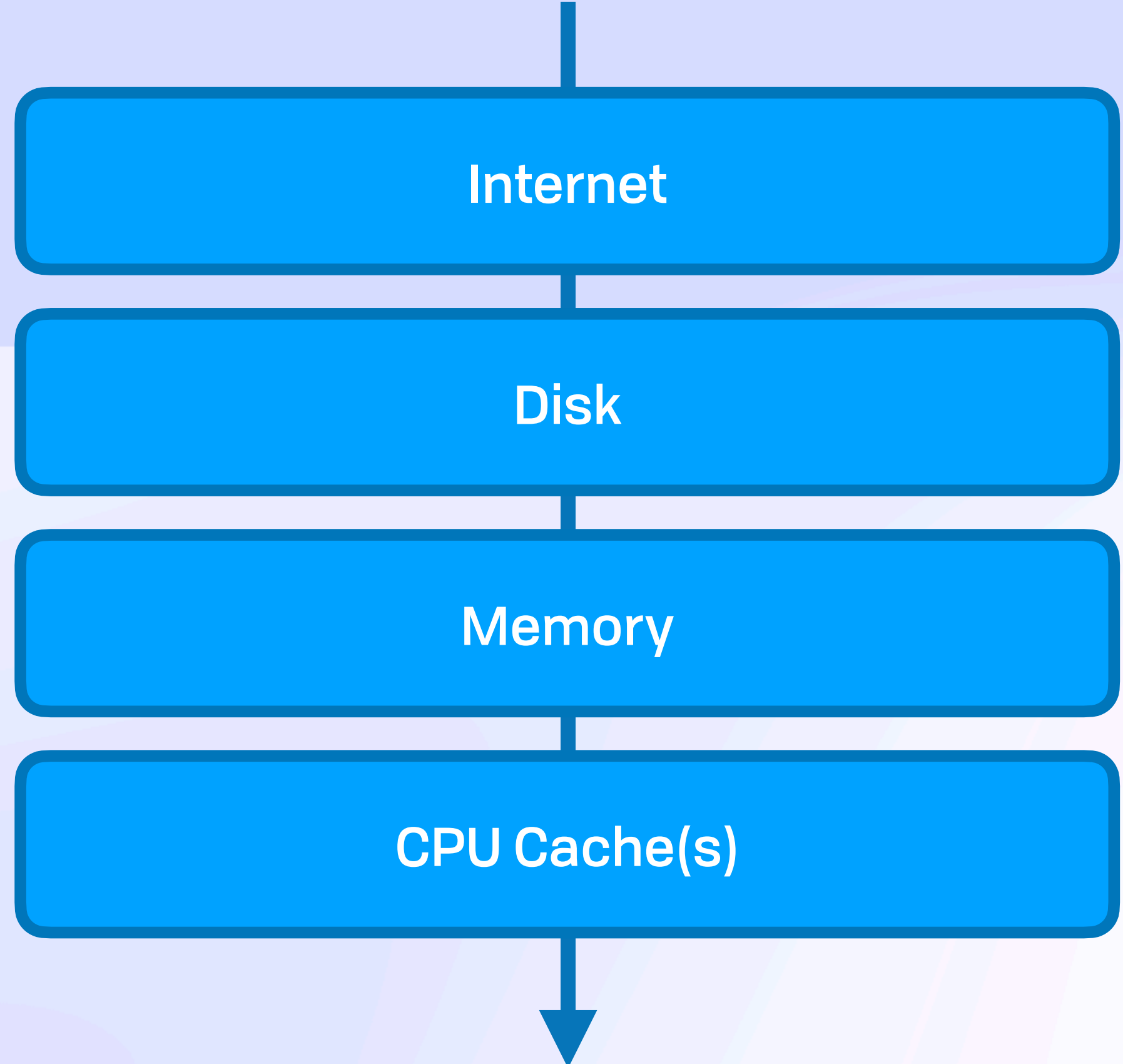
Cellular / P2P



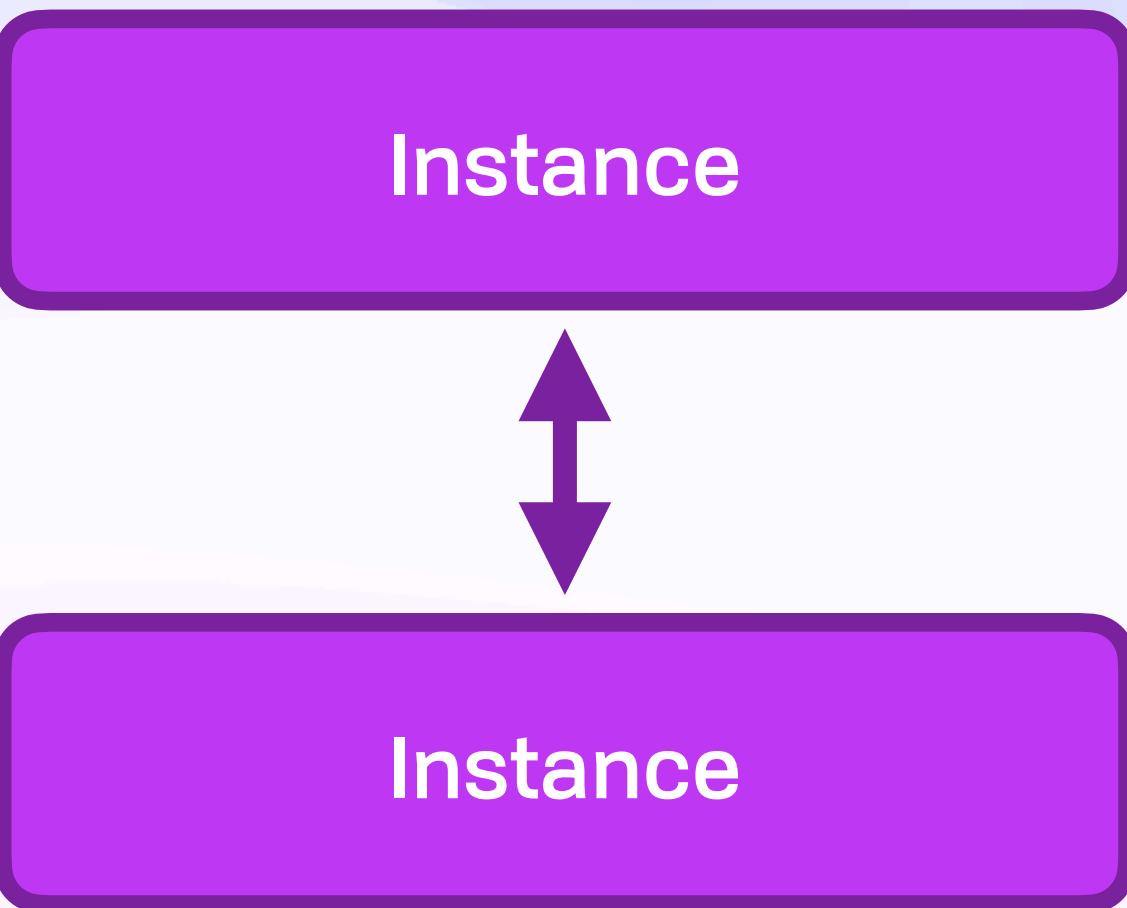
# A New Hope

# *New Metaphors*

Layered / "Cache for the Internet"



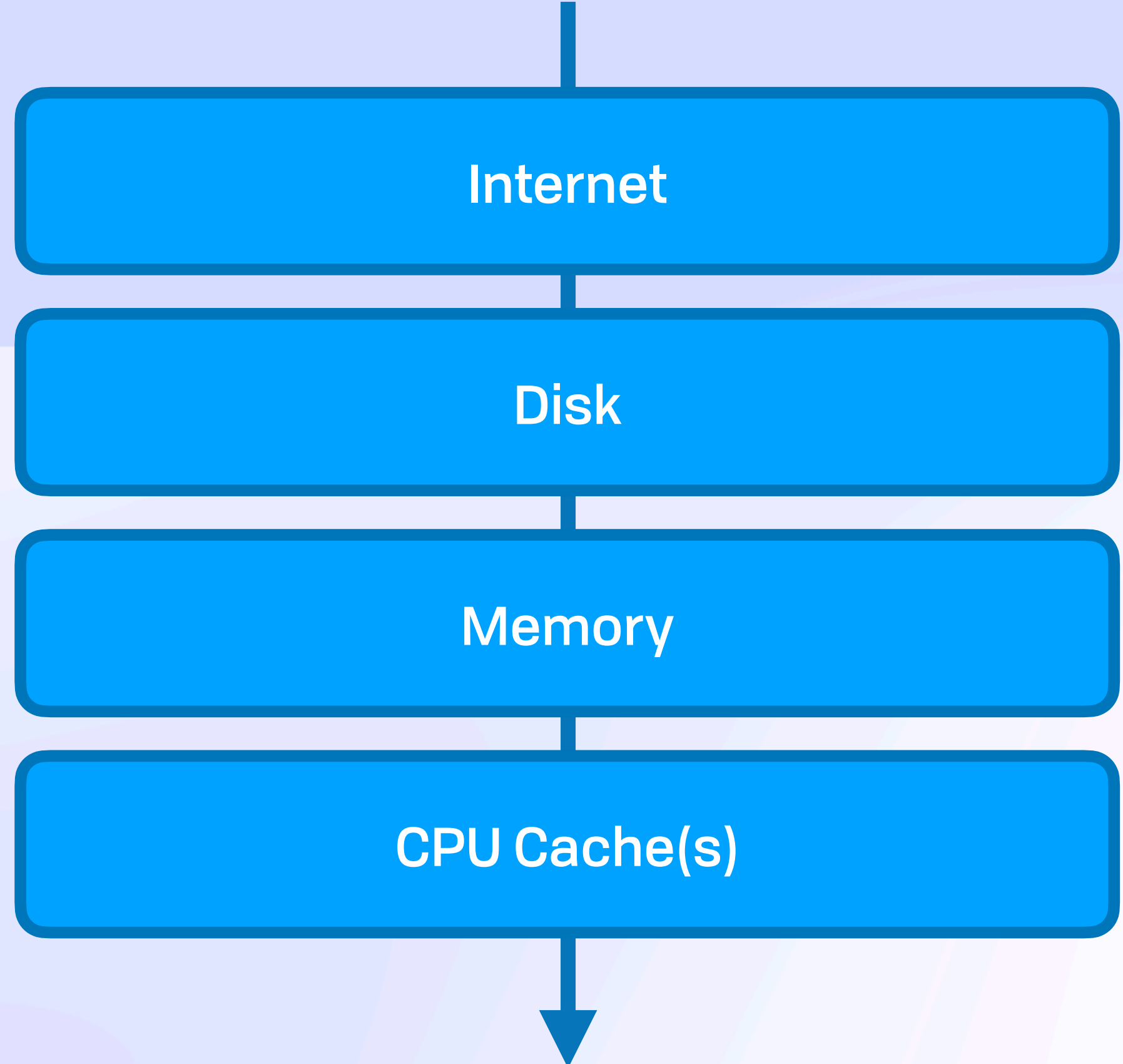
Cellular / P2P



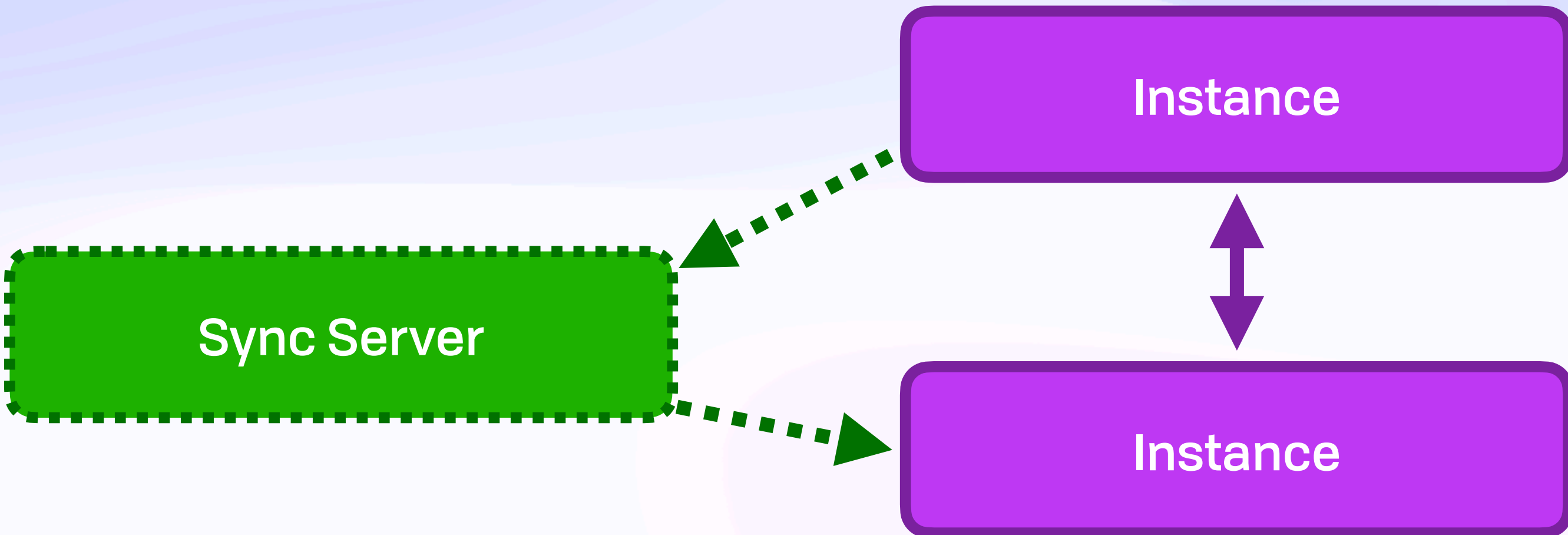
# A New Hope

# *New Metaphors*

Layered / "Cache for the Internet"



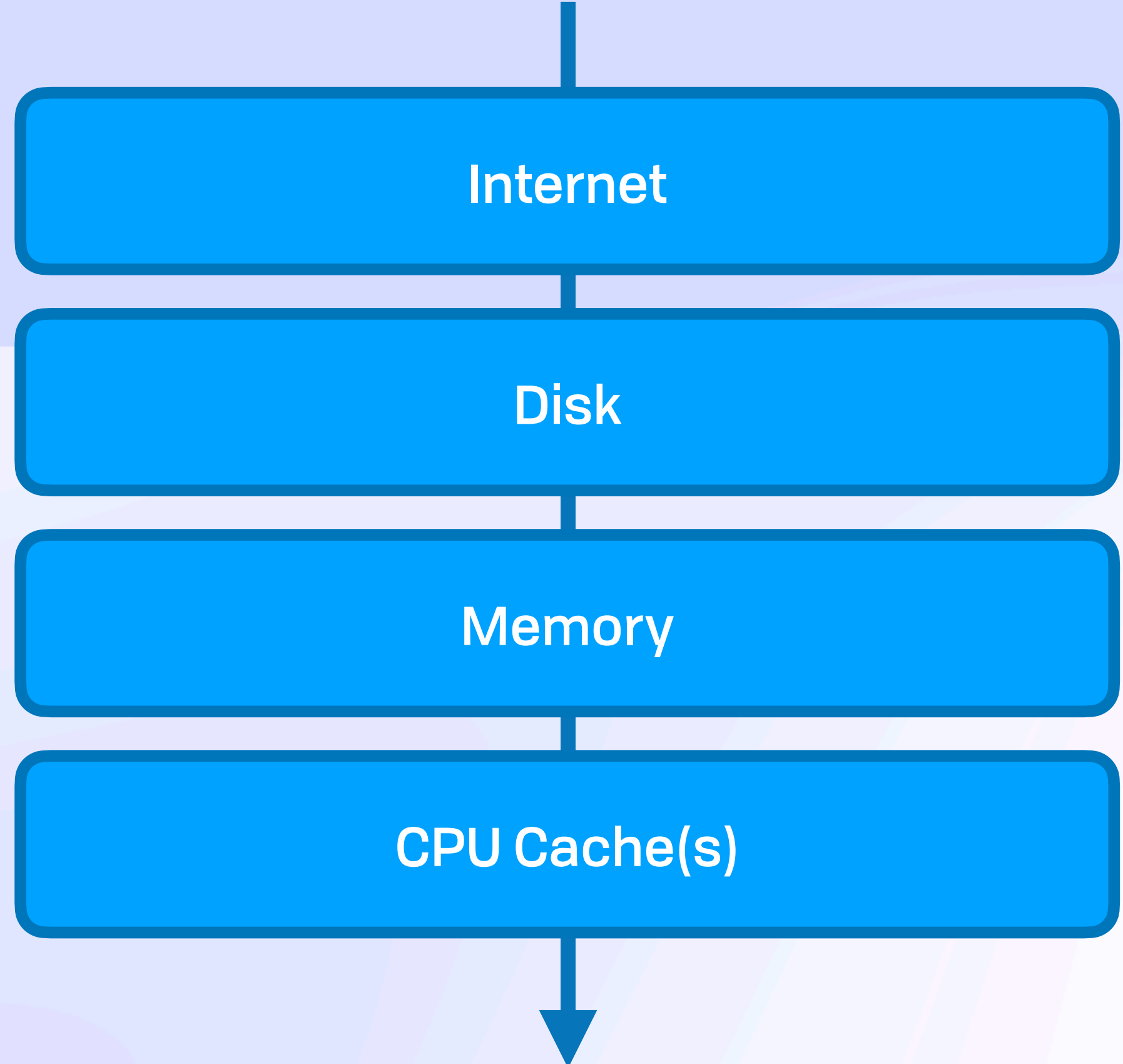
Cellular / P2P



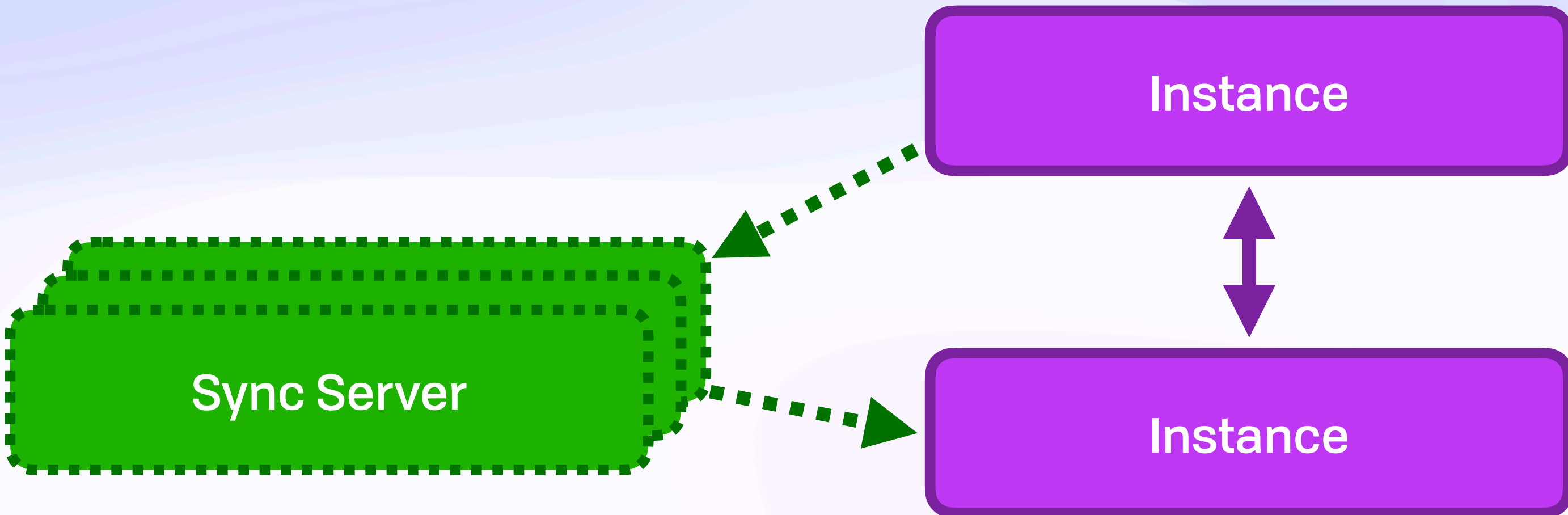
# A New Hope

# *New Metaphors*

Layered / "Cache for the Internet"



Cellular / P2P



A New Hope

***Back to Our Roots***

A New Hope

# ***Back to Our Roots***

- 1. Decentralisation***
- 2. Non-discrimination***
- 3. Bottom-up Design***
- 4. Universality***
- 5. Consensus***

– **The Web Foundation**, History of the Web

A New Hope

# *Back to Our Roots*

**1. Decentralisation**

**2. Non-discrimination**

**3. Bottom-up Design**

**4. Universality**

**5. Consensus**

| Layer        |   | Protocol data unit (PDU) |                   |
|--------------|---|--------------------------|-------------------|
| Host layers  | 7 | Application              | Data              |
|              | 6 | Presentation             |                   |
|              | 5 | Session                  |                   |
|              | 4 | Transport                | Segment, Datagram |
| Media layers | 3 | Network                  | Packet            |
|              | 2 | Data link                | Frame             |
|              | 1 | Physical                 | Bit, Symbol       |

[en.wikipedia.org/wiki/OSI\\_model](https://en.wikipedia.org/wiki/OSI_model)

**– The Web Foundation, History of the Web**



A New Hope

# *Back to Our Roots*

**1. Decentralisation**

**2. Non-discrimination**

**3. Bottom-up Design**

**4. Universality**

**5. Consensus**

| Layer        |   | Protocol data unit (PDU) |             |
|--------------|---|--------------------------|-------------|
| Host layers  | 7 | Application              | Data        |
|              | 6 | Presentation             |             |
|              | 5 | Session                  |             |
|              | 4 | Transport                |             |
| Media layers | 3 | Network                  | Packet      |
|              | 2 | Data link                | Frame       |
|              | 1 | Physical                 | Bit, Symbol |

[en.wikipedia.org/wiki/OSI\\_model](https://en.wikipedia.org/wiki/OSI_model)

**– The Web Foundation, History of the Web**

A New Hope

# *Back to Our Roots*

**1. Decentralisation**

**2. Non-discrimination**

**3. Bottom-up Design**

**4. Universality**

**5. Consensus**

| Layer        |   | Protocol data unit (PDU) |                   |
|--------------|---|--------------------------|-------------------|
| Host layers  | 7 | Application              | Data              |
|              | 6 | Presentation             |                   |
|              | 5 | Session                  |                   |
|              | 4 | Transport                | Segment, Datagram |
| Media layers | 3 | Network                  | Packet            |
|              | 2 | Data link                | Frame             |
|              | 1 | Physical                 | Bit, Symbol       |

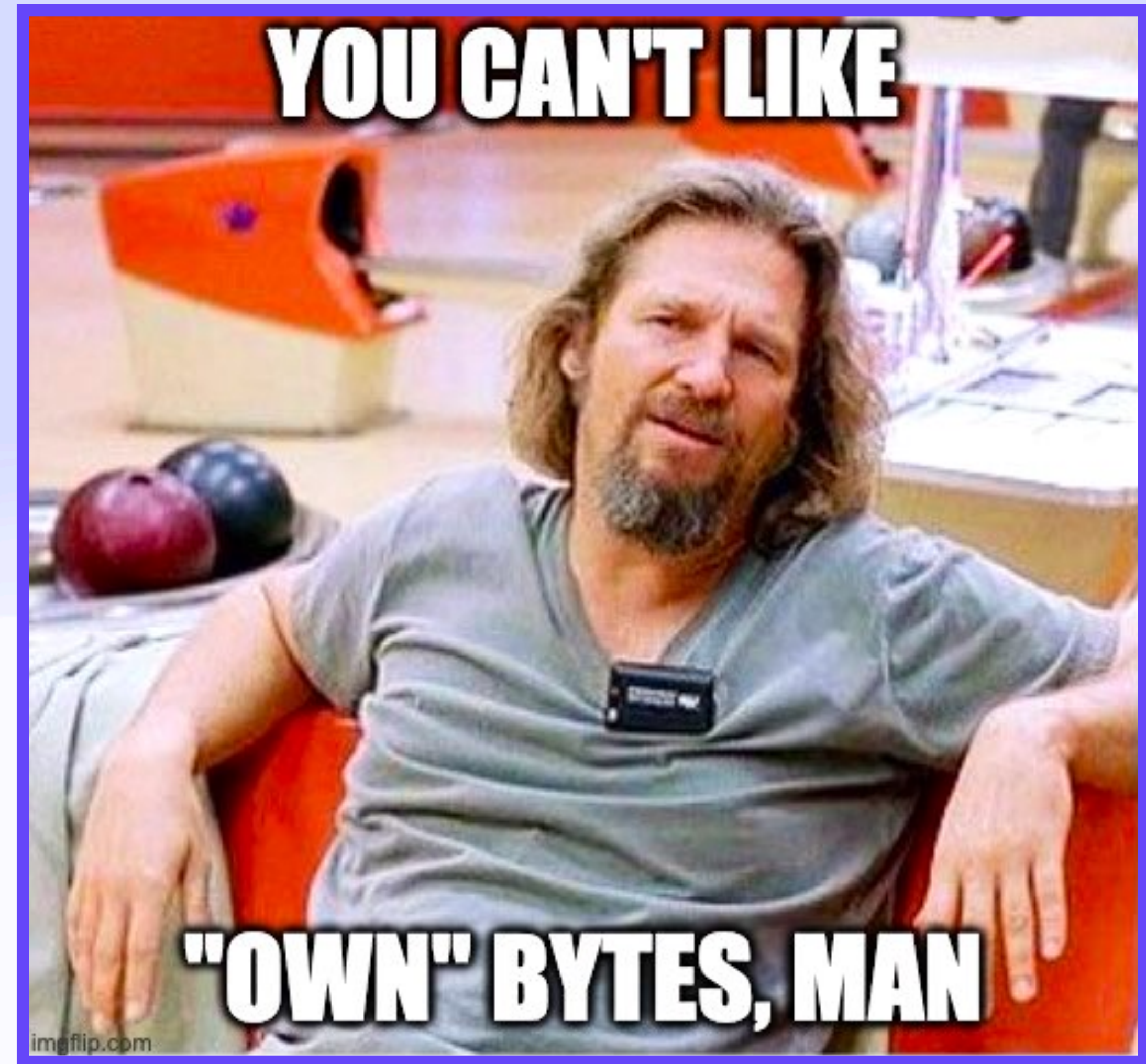
[en.wikipedia.org/wiki/OSI\\_model](https://en.wikipedia.org/wiki/OSI_model)

– **The Web Foundation**, History of the Web

A New Hope

# *User Agency*

- **Entry:** Empower users to participate
- **Exit:** Option to move or leave
- **Safety:** Control access to *your* data
- **Serve:** Provide capacity to others



Welcome to the Jungle

# *Distributed Systems*



Welcome to the Jungle

# *Systems*



Welcome to the Jungle

# *Disorderly Systems*



# Disorderly Systems



# Disorderly Systems

Much of the ***pain*** in traditional distributed programming comes from this mismatch: programmers are expected to ***bridge from an ordered programming model into a disordered reality*** that executes their code.

— The Bloom Language Website



# Disorderly Systems



# Disorderly Systems



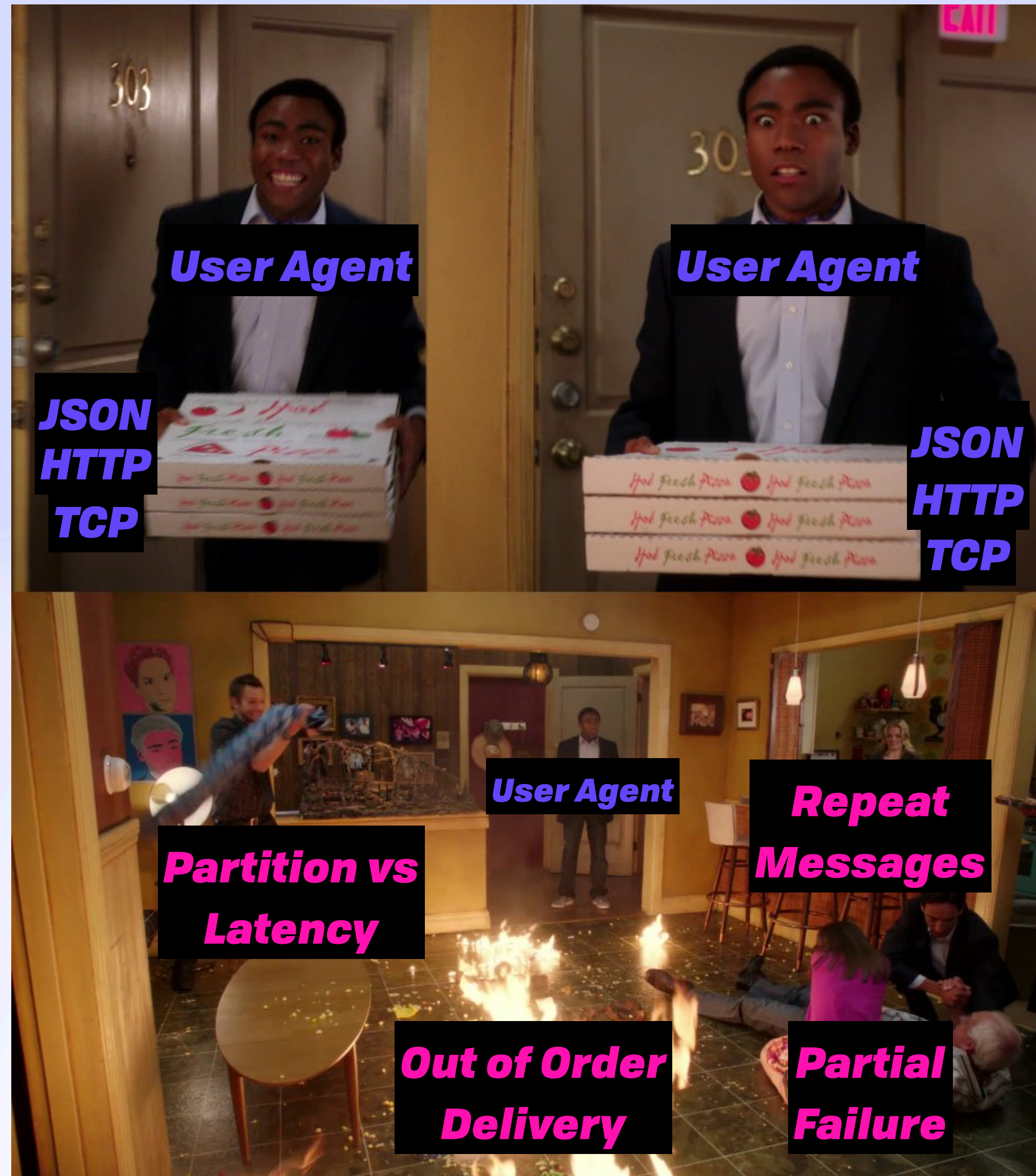
# Disorderly Systems



# Disorderly Systems

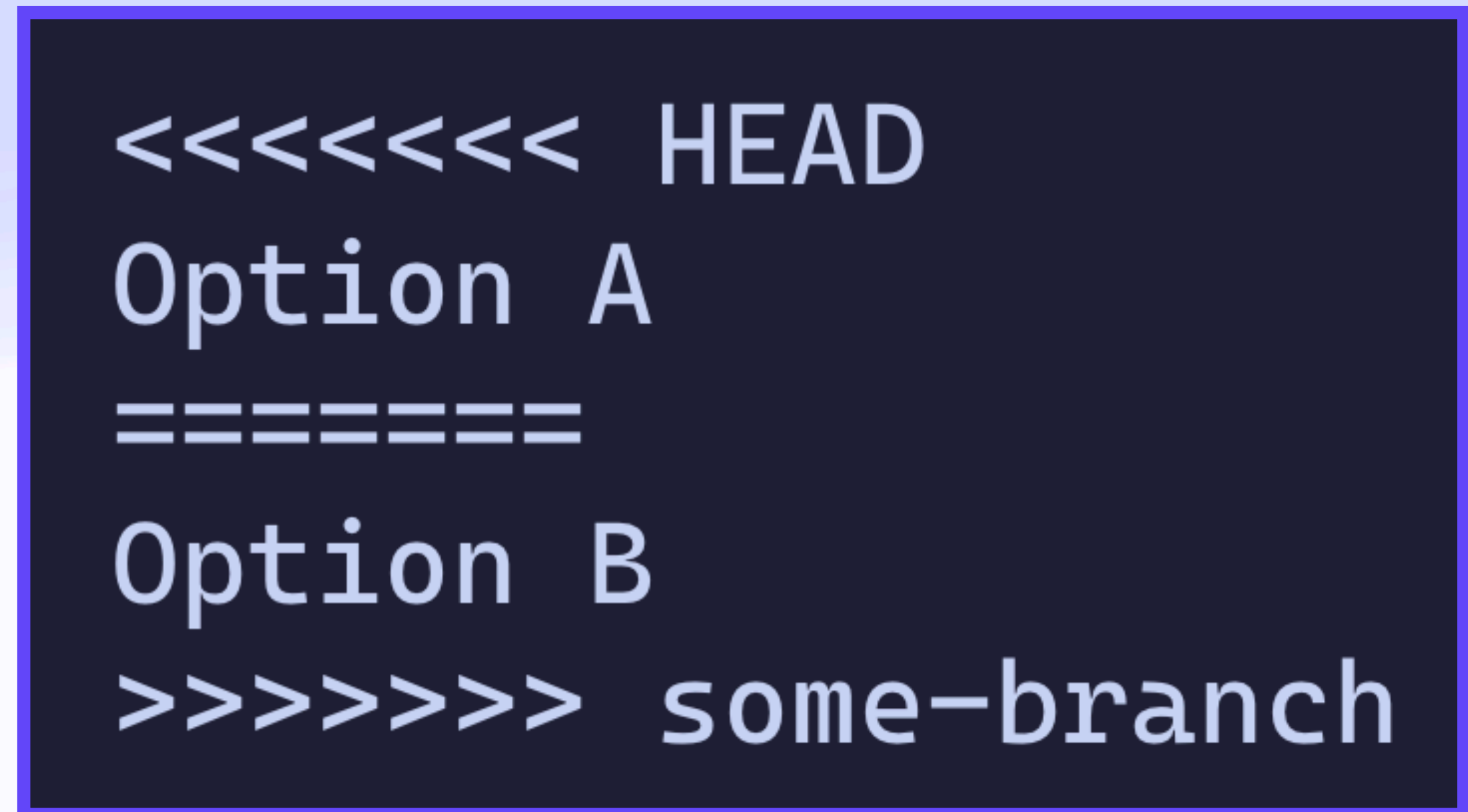
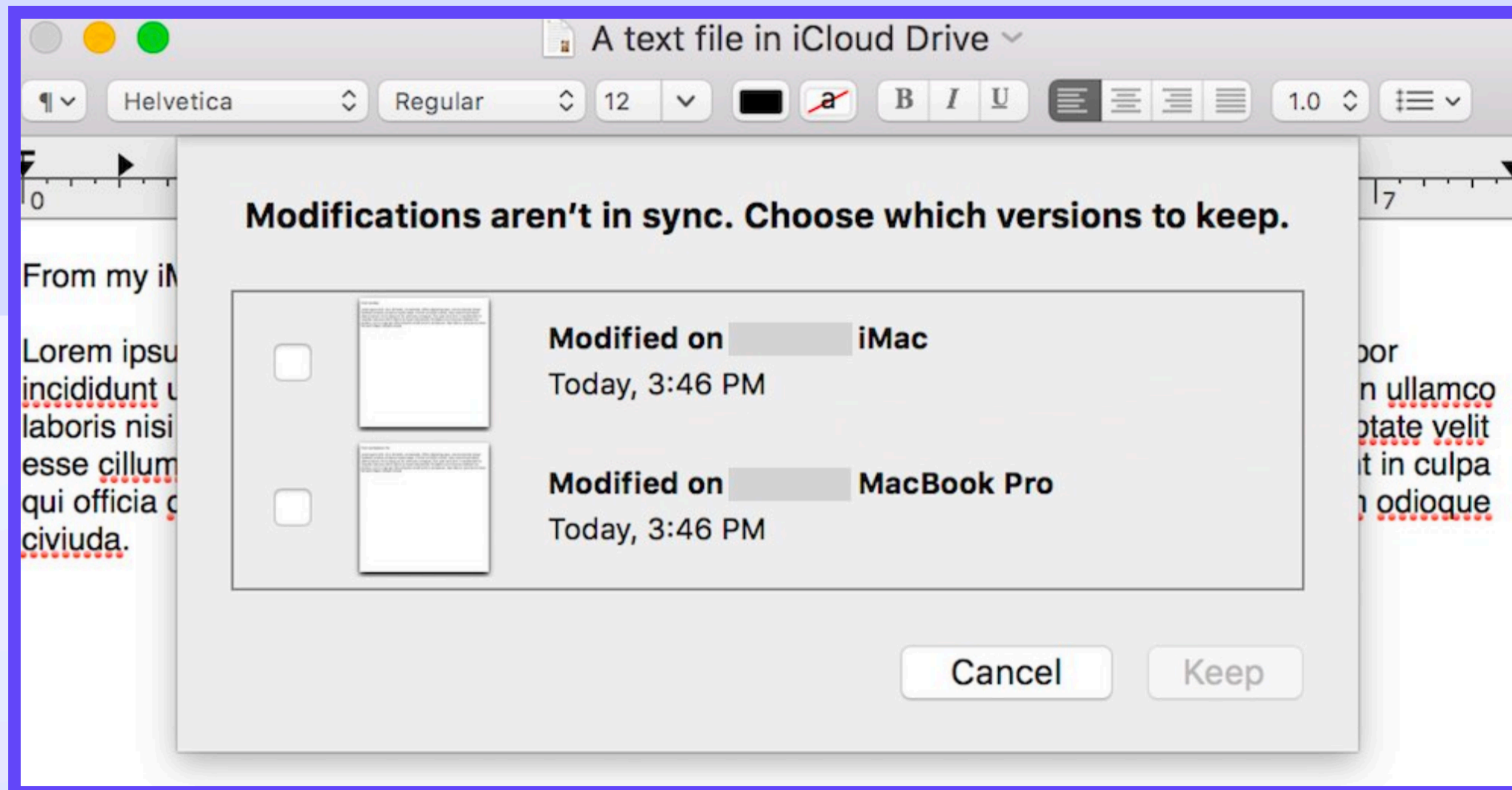


# Disorderly Systems



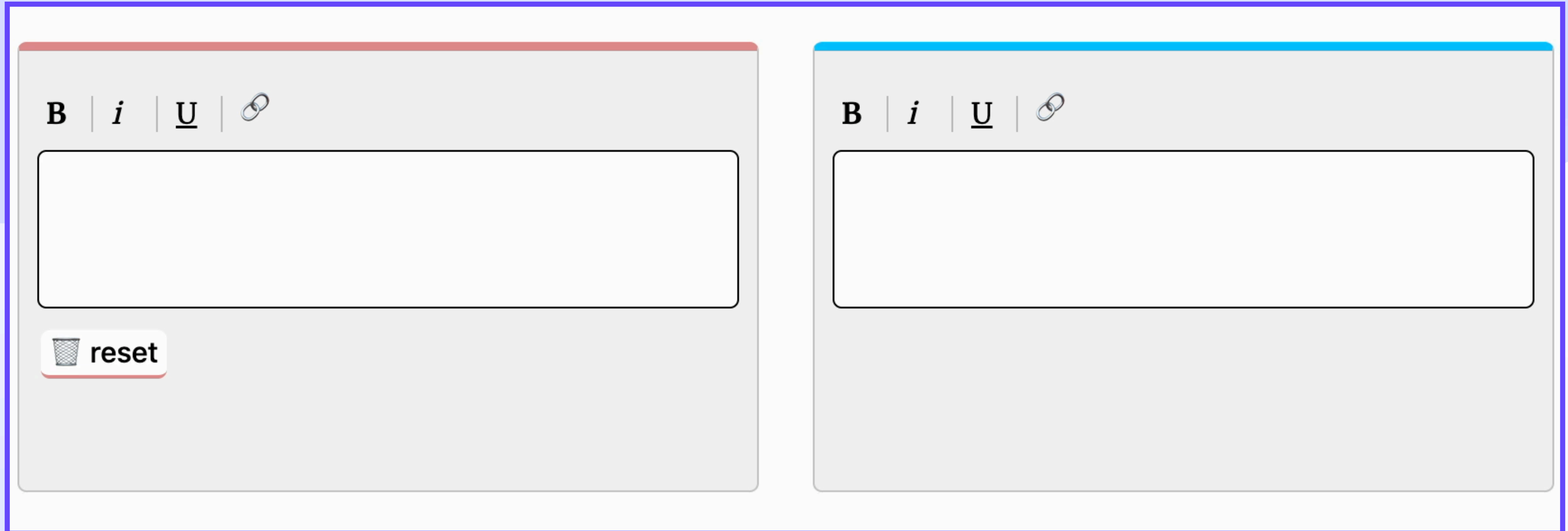
# Disorderly Systems

# Objectivist Ergonomics



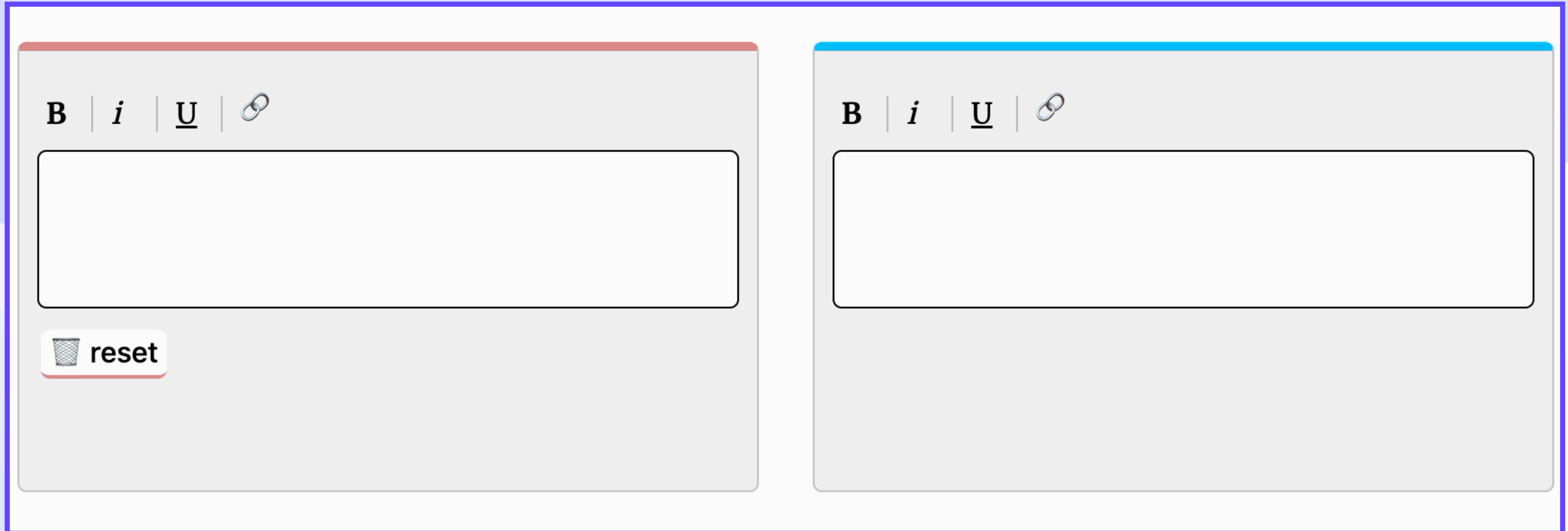
# Disorderly Systems

## *Enabling Tech: CRDTs*



# Disorderly Systems

## *Enabling Tech: CRDTs*





Disorderly Systems

**Keeping CALM** 🧘

A problem has a **consistent, coordination-free** distributed implementation if and only if it is **monotonic**.

**The CALM Theorem**

Disorderly Systems

***Monotonicity***



Disorderly Systems

***Monotonicity***

$\max(a, b)$

Disorderly Systems

***Monotonicity***

$\max(a, b)$



# Disorderly Systems

## *Monotonicity*

$\max(a, b)$

$\max(1, \underline{2})$



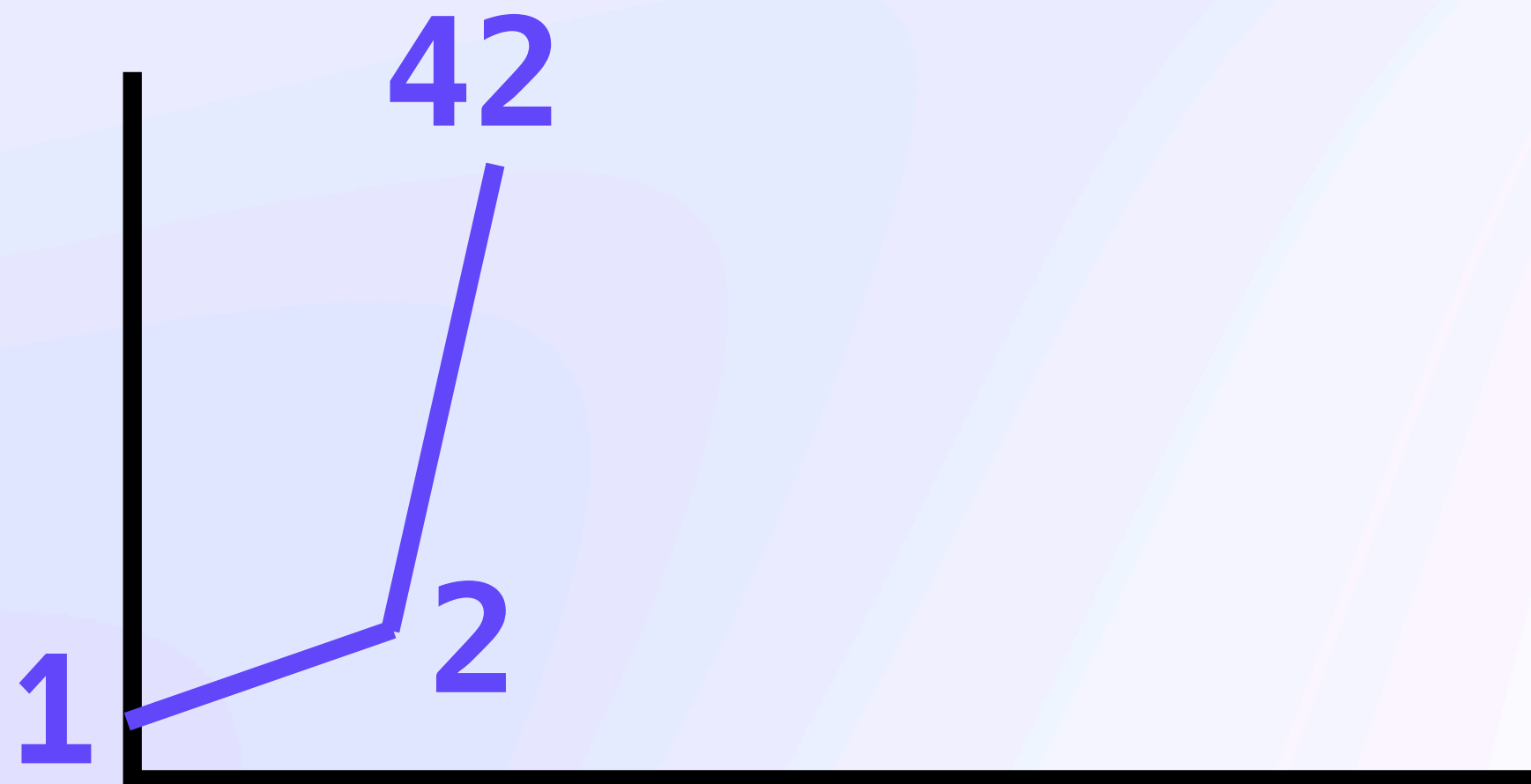
# Disorderly Systems

## *Monotonicity*

$\max(a, b)$

$\max(1, \underline{2})$

$\max(2, \underline{42})$



# Disorderly Systems

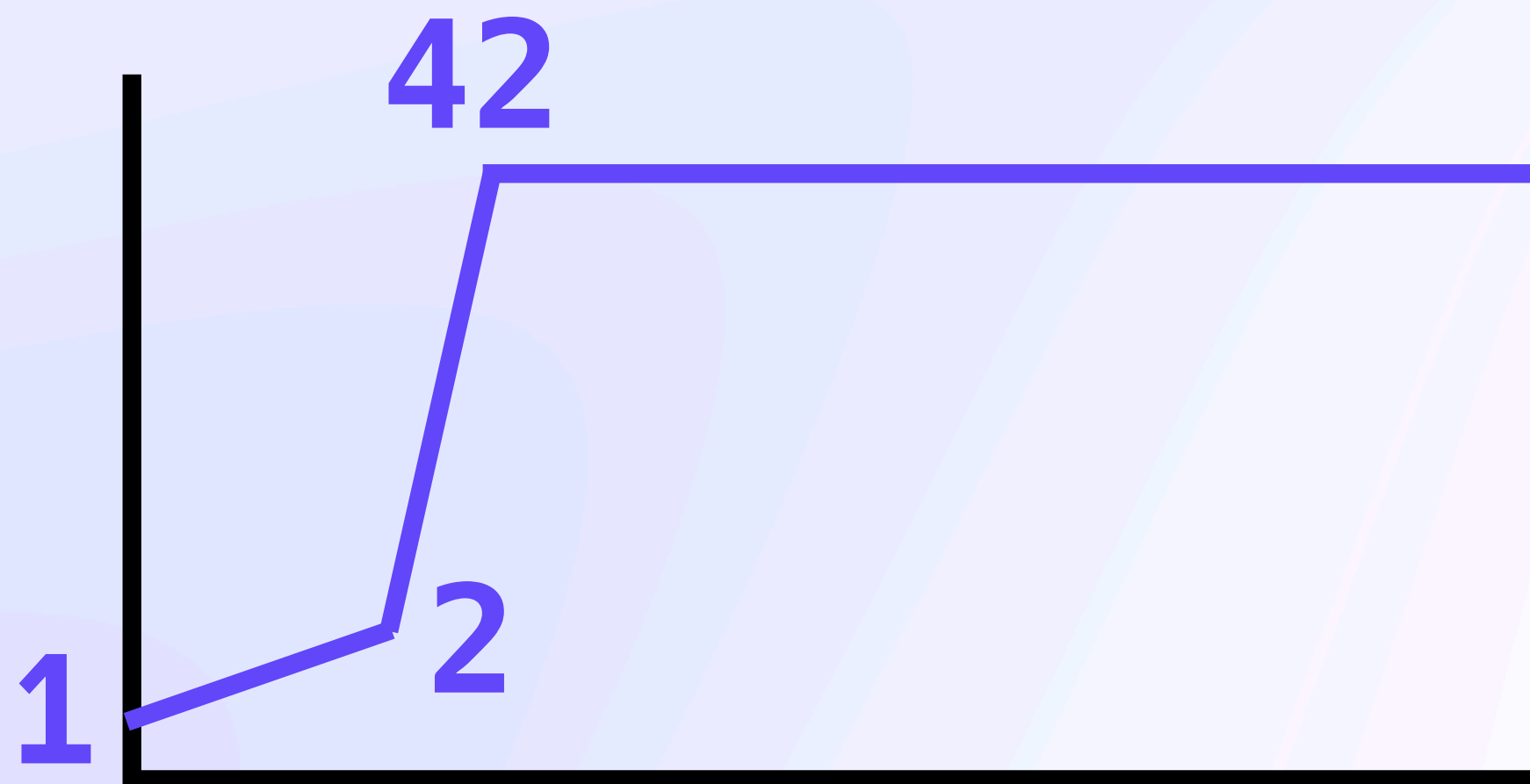
## *Monotonicity*

$\max(a, b)$

$\max(1, \underline{2})$

$\max(2, \underline{42})$

$\max(\underline{42}, 10)$



# Disorderly Systems

## *Monotonicity*

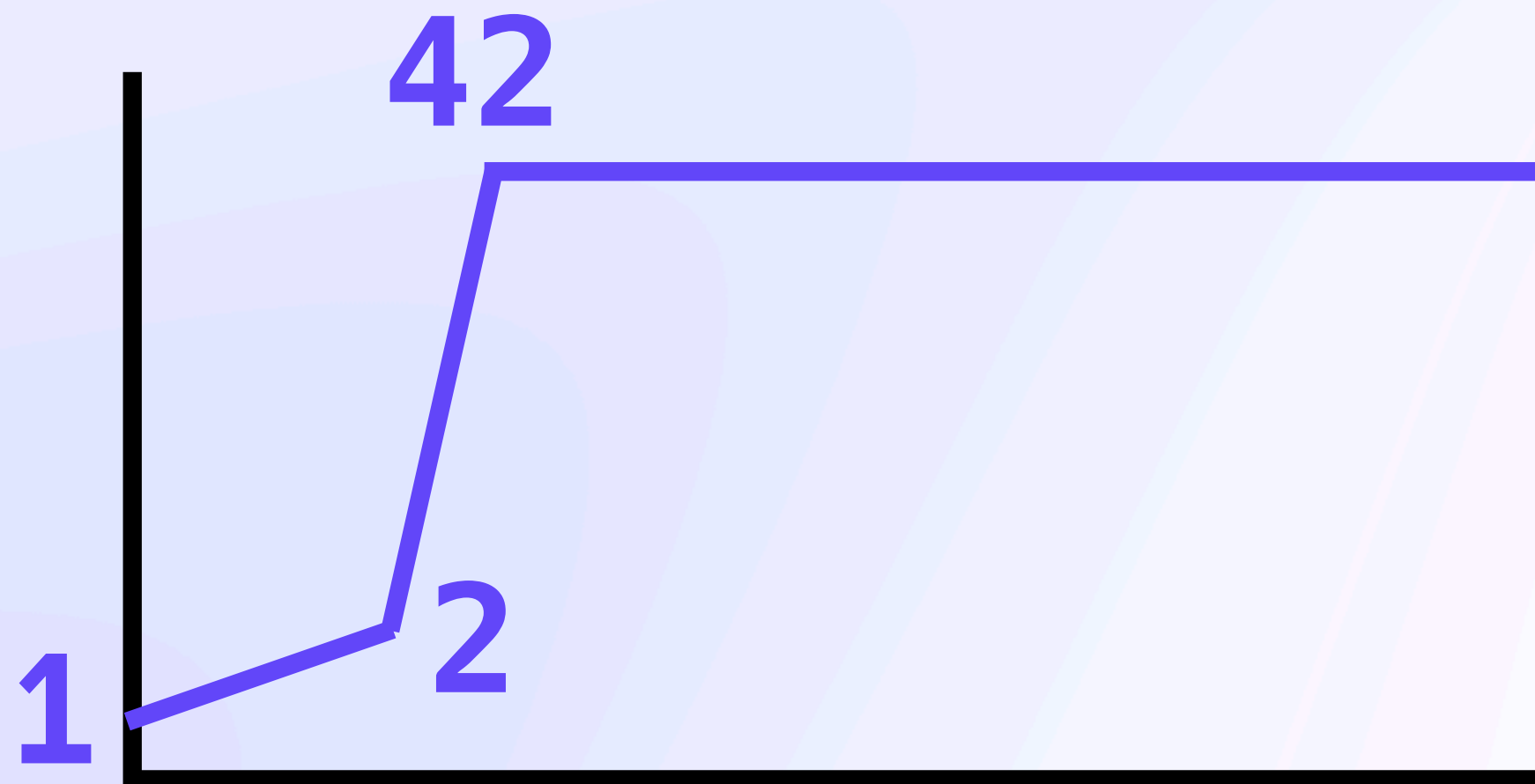
$\max(a, b)$

$\max(1, \underline{2})$

$\max(2, \underline{42})$

$\max(\underline{42}, 10)$

$\max(11, \underline{42})$





# Disorderly Systems

## *Monotonicity*

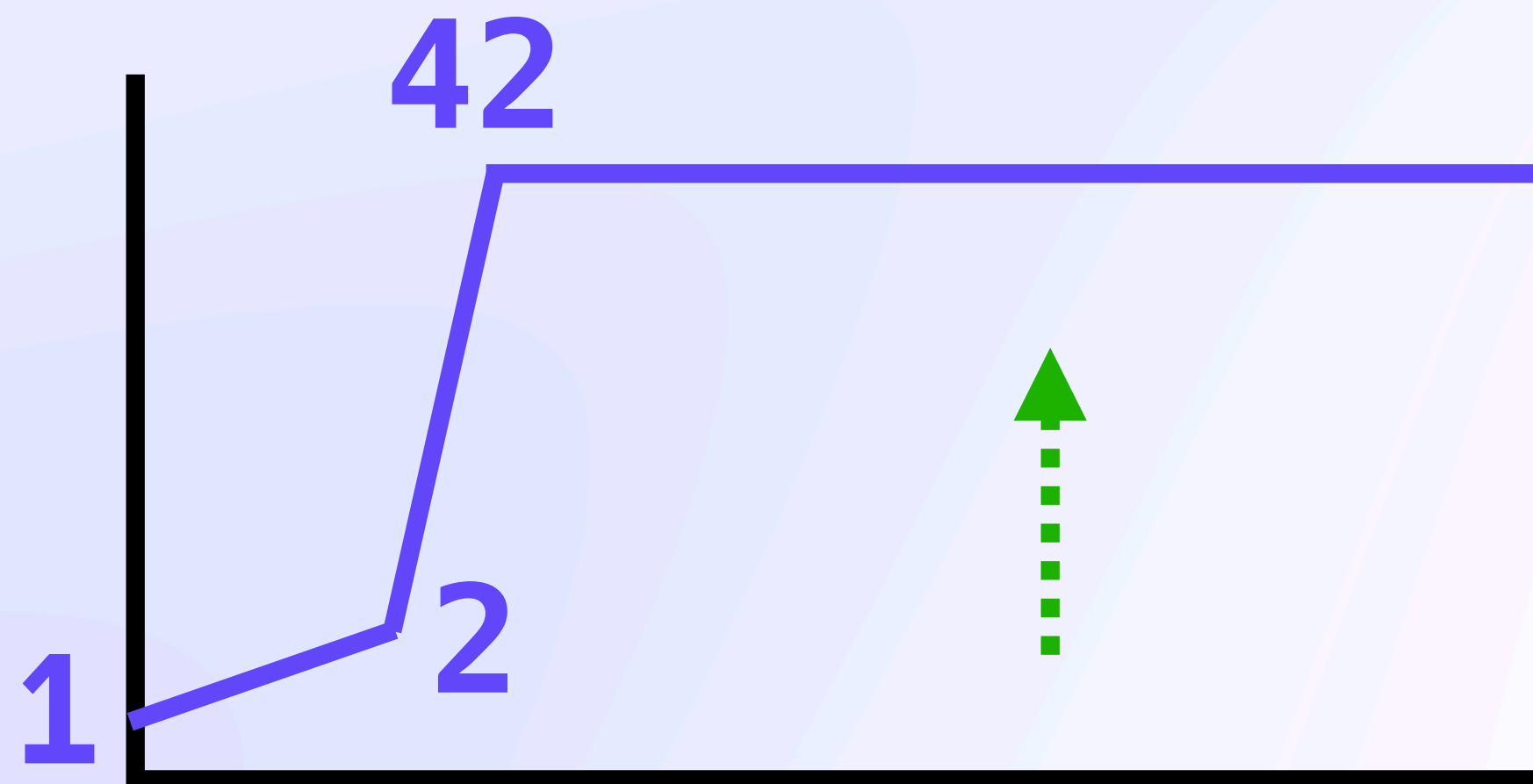
$\max(a, b)$

$\max(1, \underline{2})$

$\max(2, \underline{42})$

$\max(\underline{42}, 10)$

$\max(11, \underline{42})$



# Disorderly Systems

## *Monotonicity*

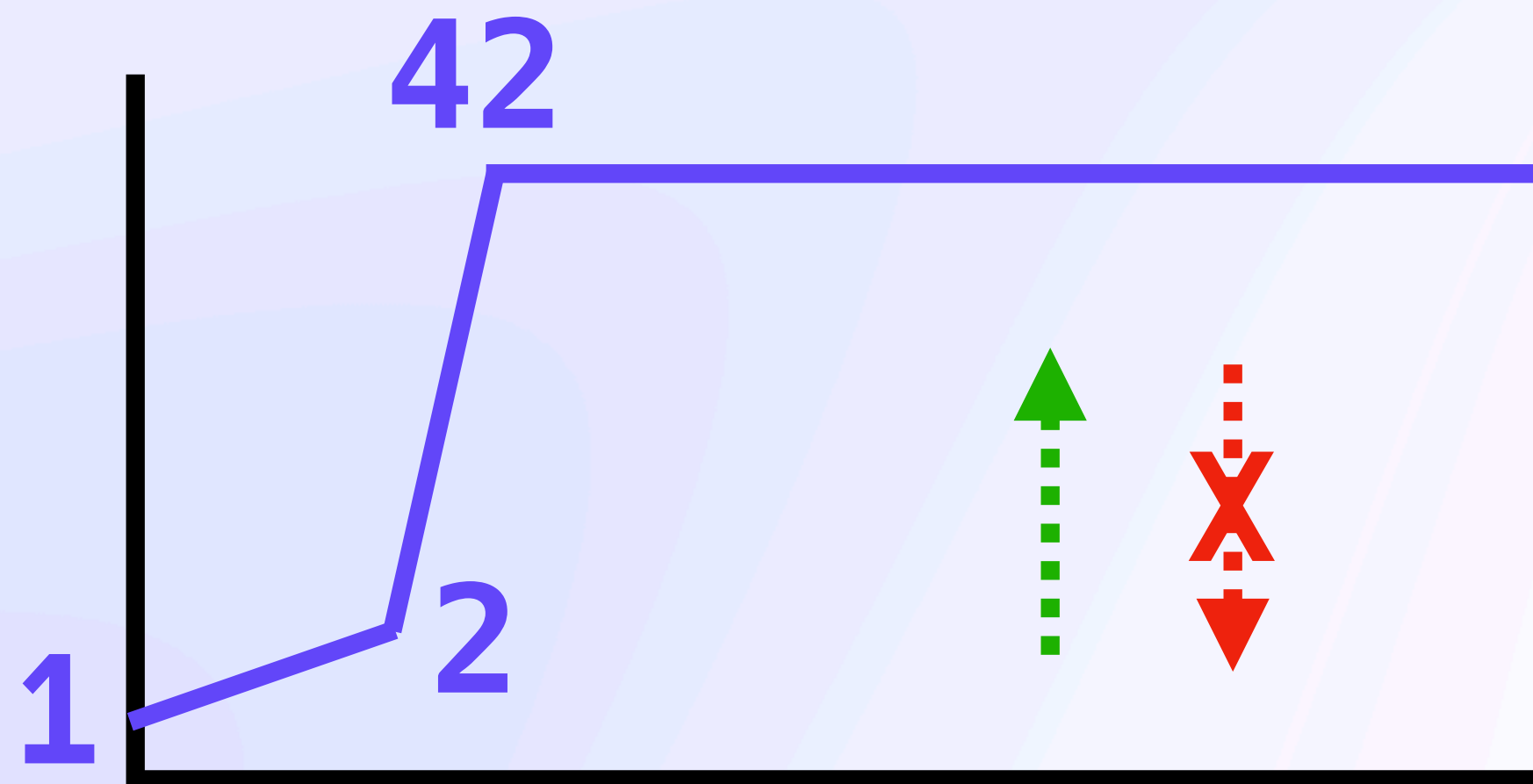
$\max(a, b)$

$\max(1, \underline{2})$

$\max(2, \underline{42})$

$\max(\underline{42}, 10)$

$\max(11, \underline{42})$



# Disorderly Systems

## *Monotonicity*

`max(a, b)`

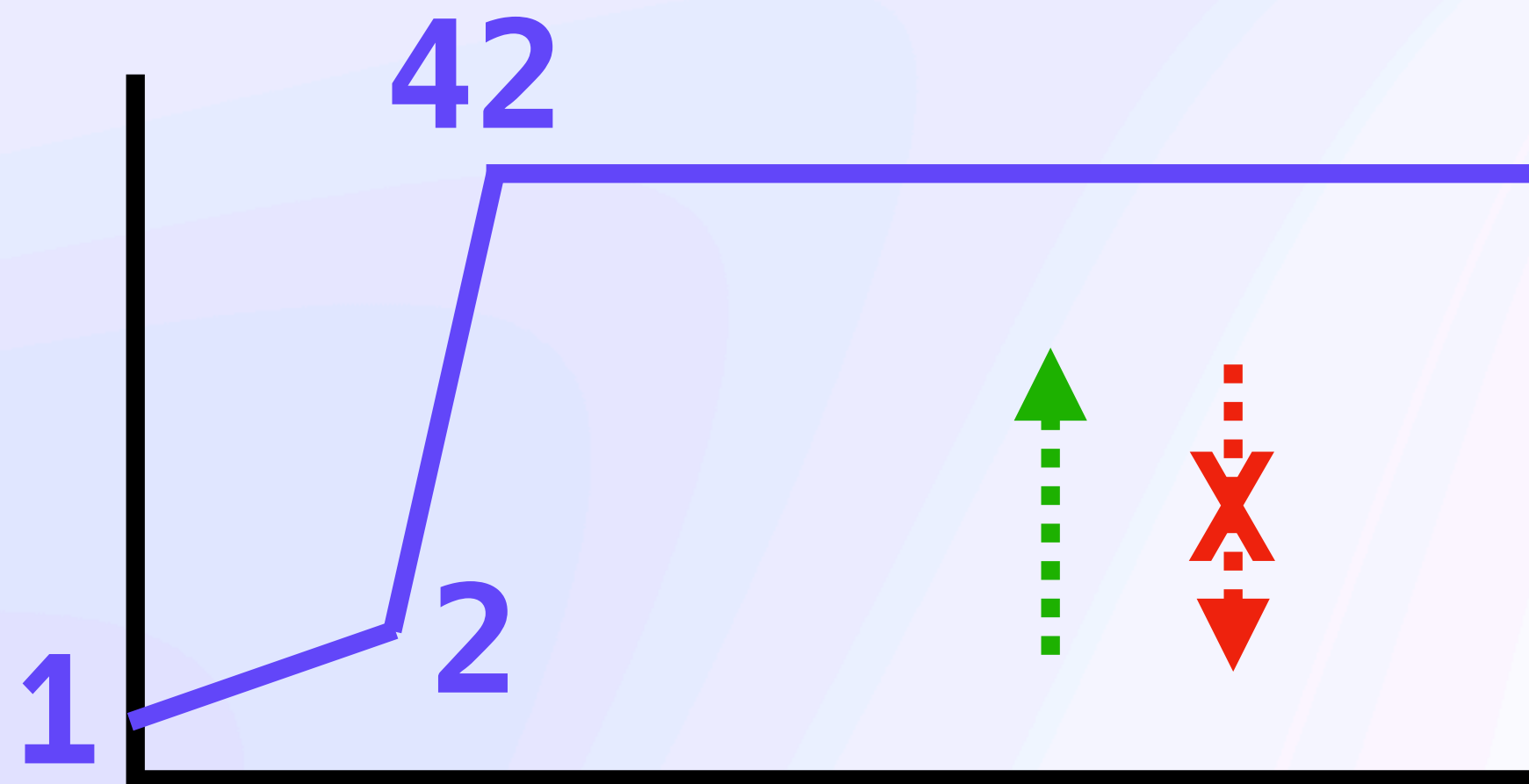
`max(1, 2)`

`max(2, 42)`

`max(42, 10)`

`max(11, 42)`

`set.add(item)`



# Disorderly Systems

## *Monotonicity*

$\max(a, b)$

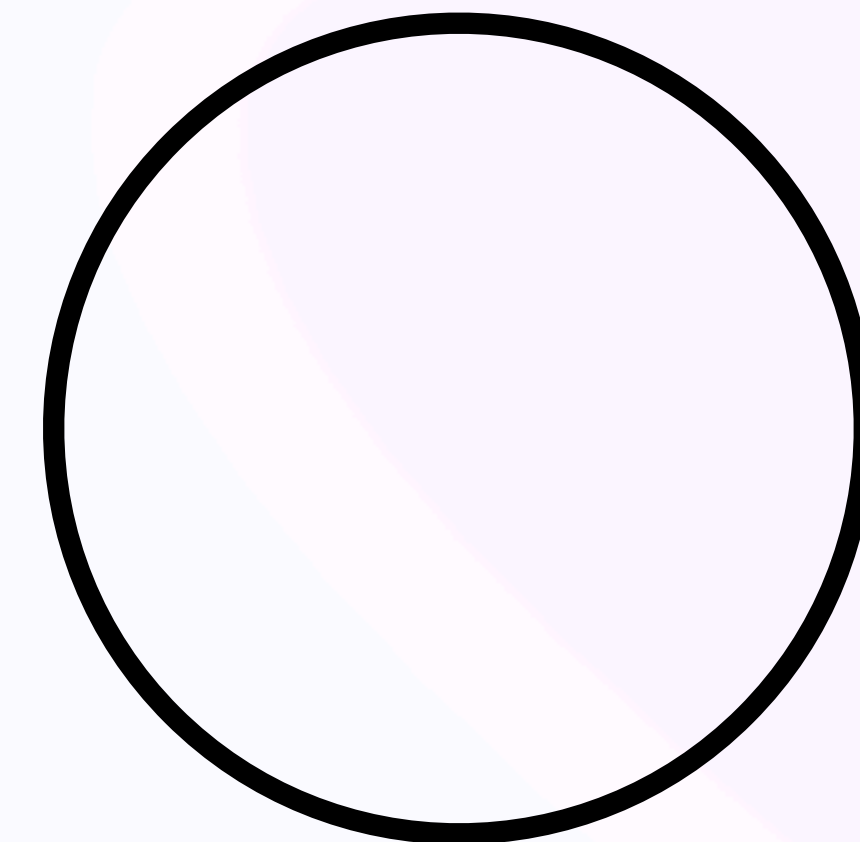
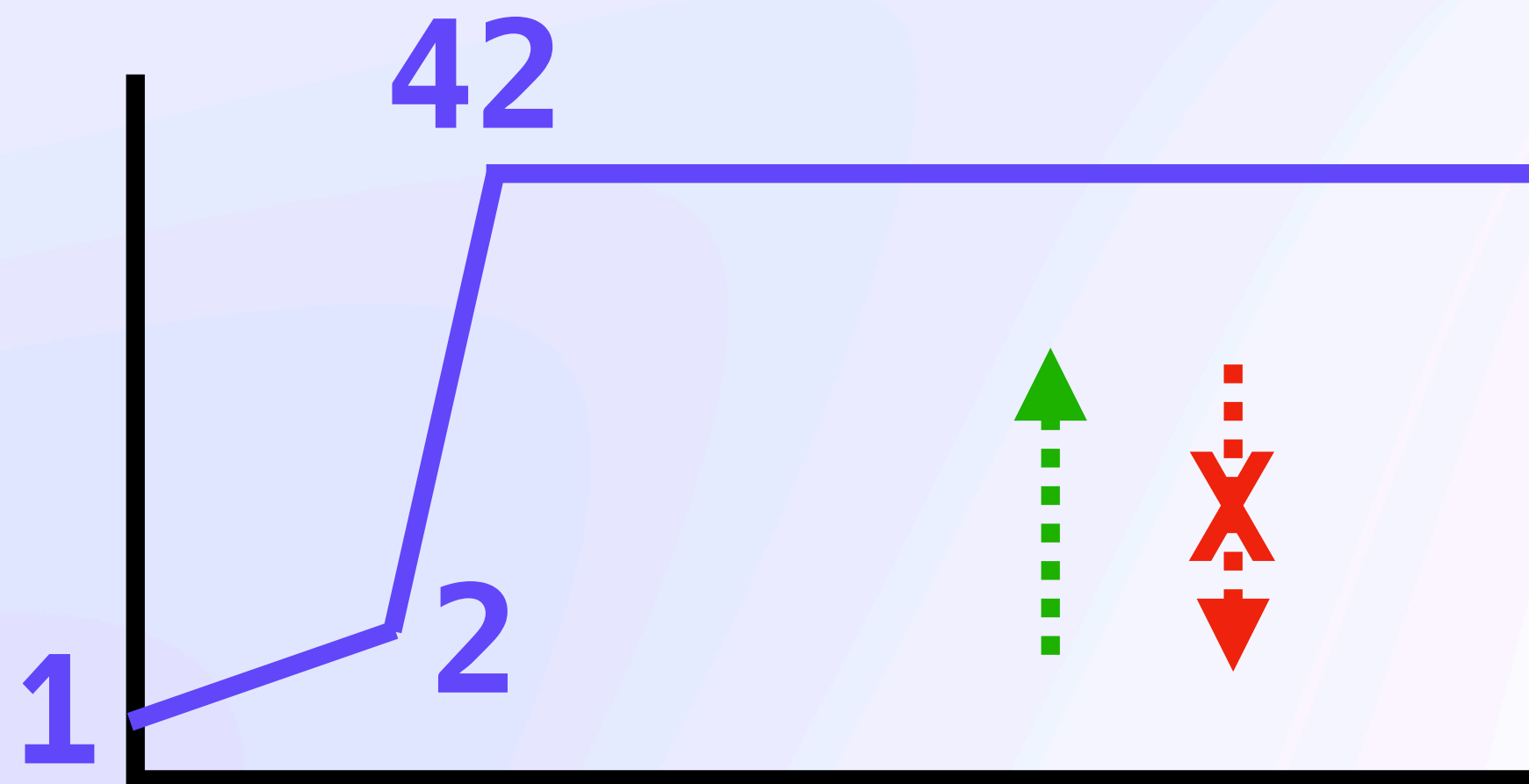
$\max(1, \underline{2})$

$\max(2, \underline{42})$

$\max(\underline{42}, 10)$

$\max(11, \underline{42})$

`set.add(item)`



# Disorderly Systems

## *Monotonicity*

$\max(a, b)$

$\max(1, \underline{2})$

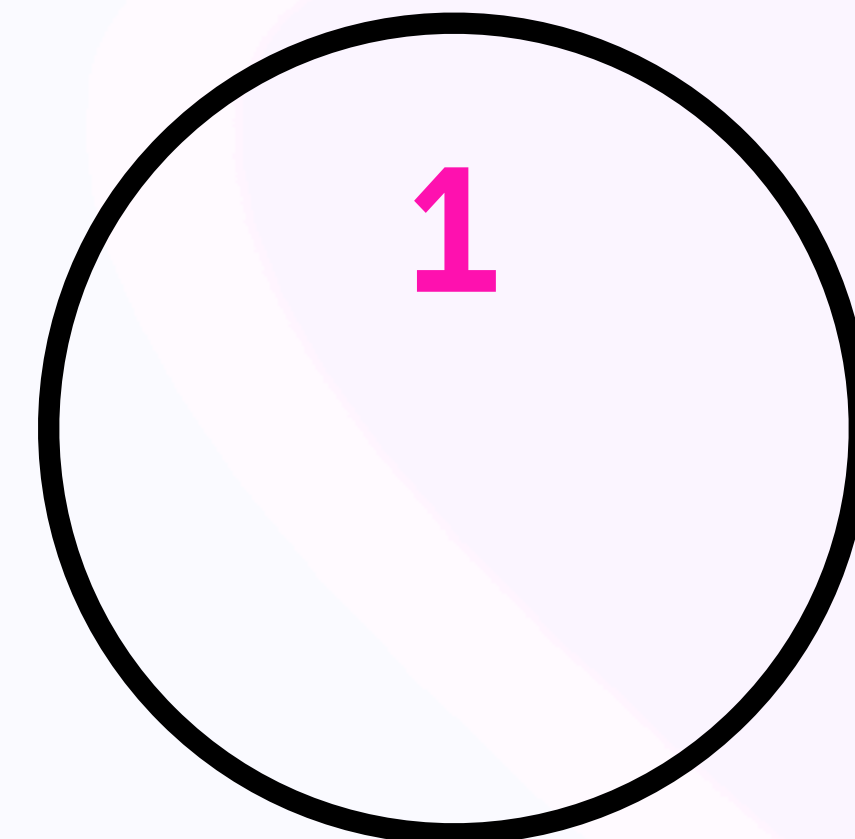
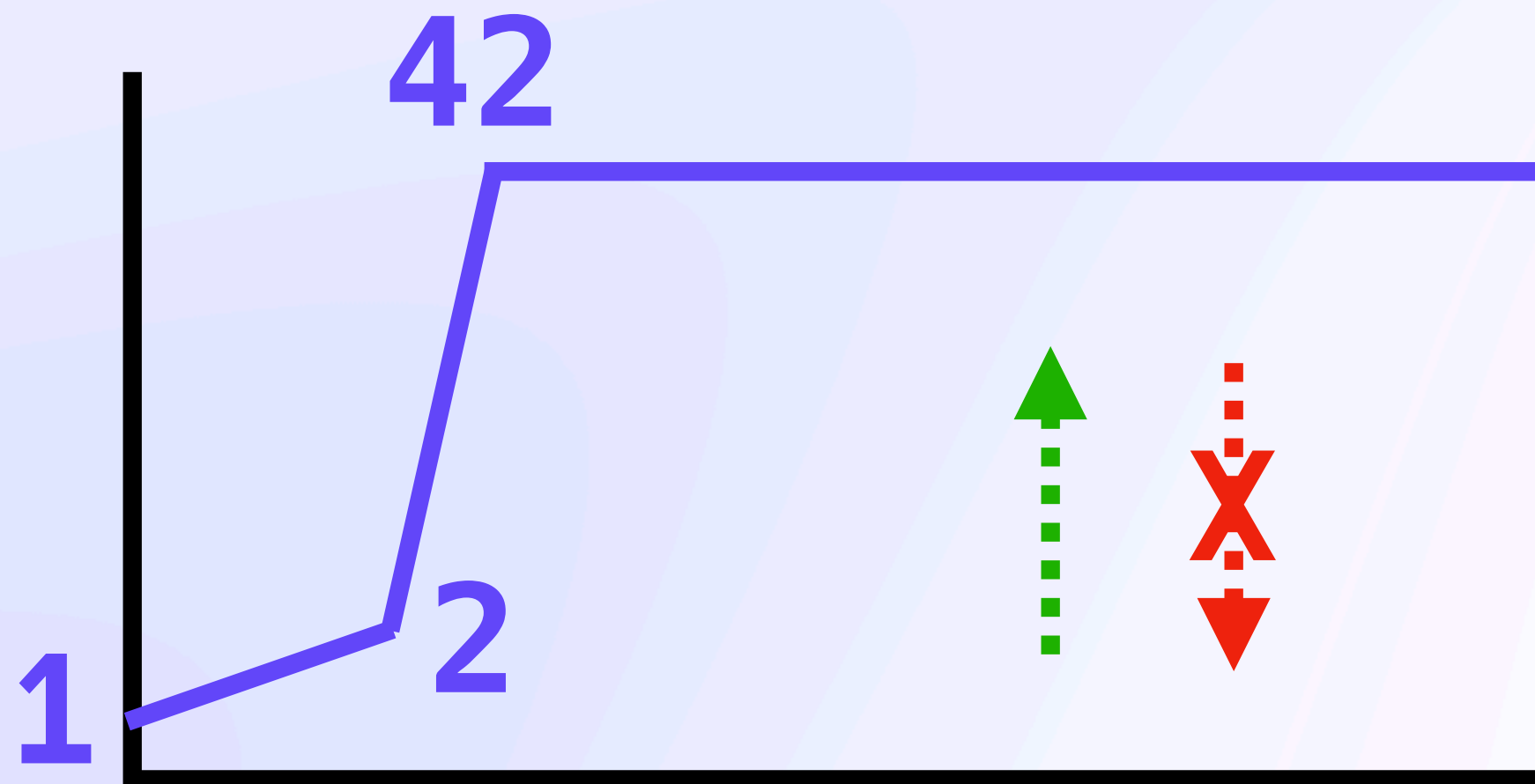
$\max(2, \underline{42})$

$\max(\underline{42}, 10)$

$\max(11, \underline{42})$

`set.add(item)`

`{}.add(1)`



# Disorderly Systems

## *Monotonicity*

`max(a, b)`

`max(1, 2)`

`max(2, 42)`

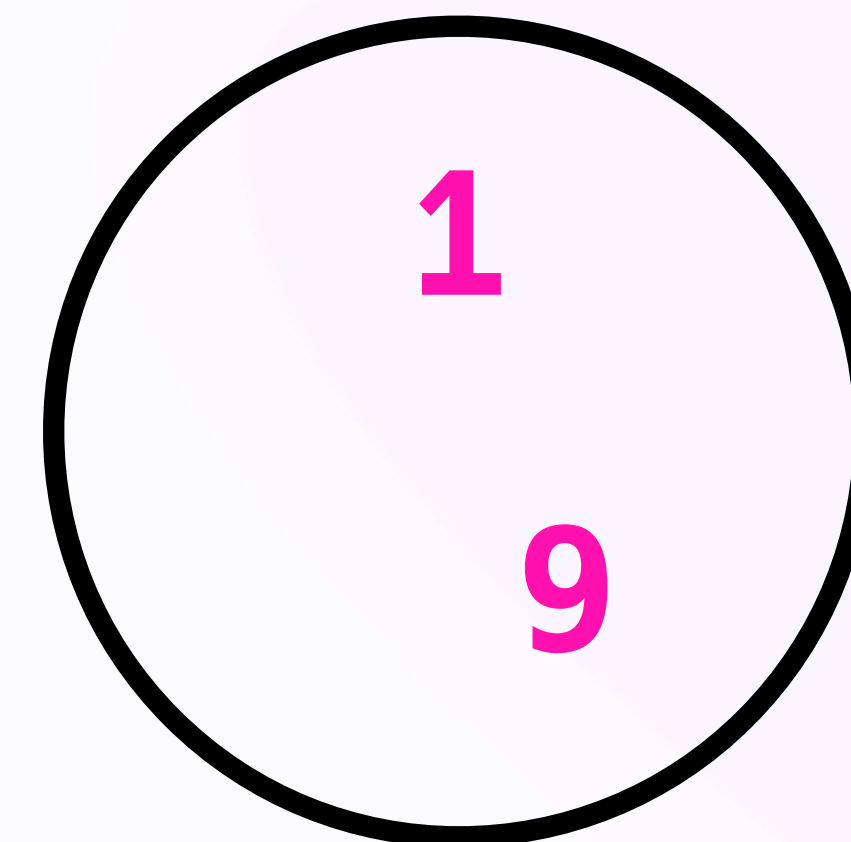
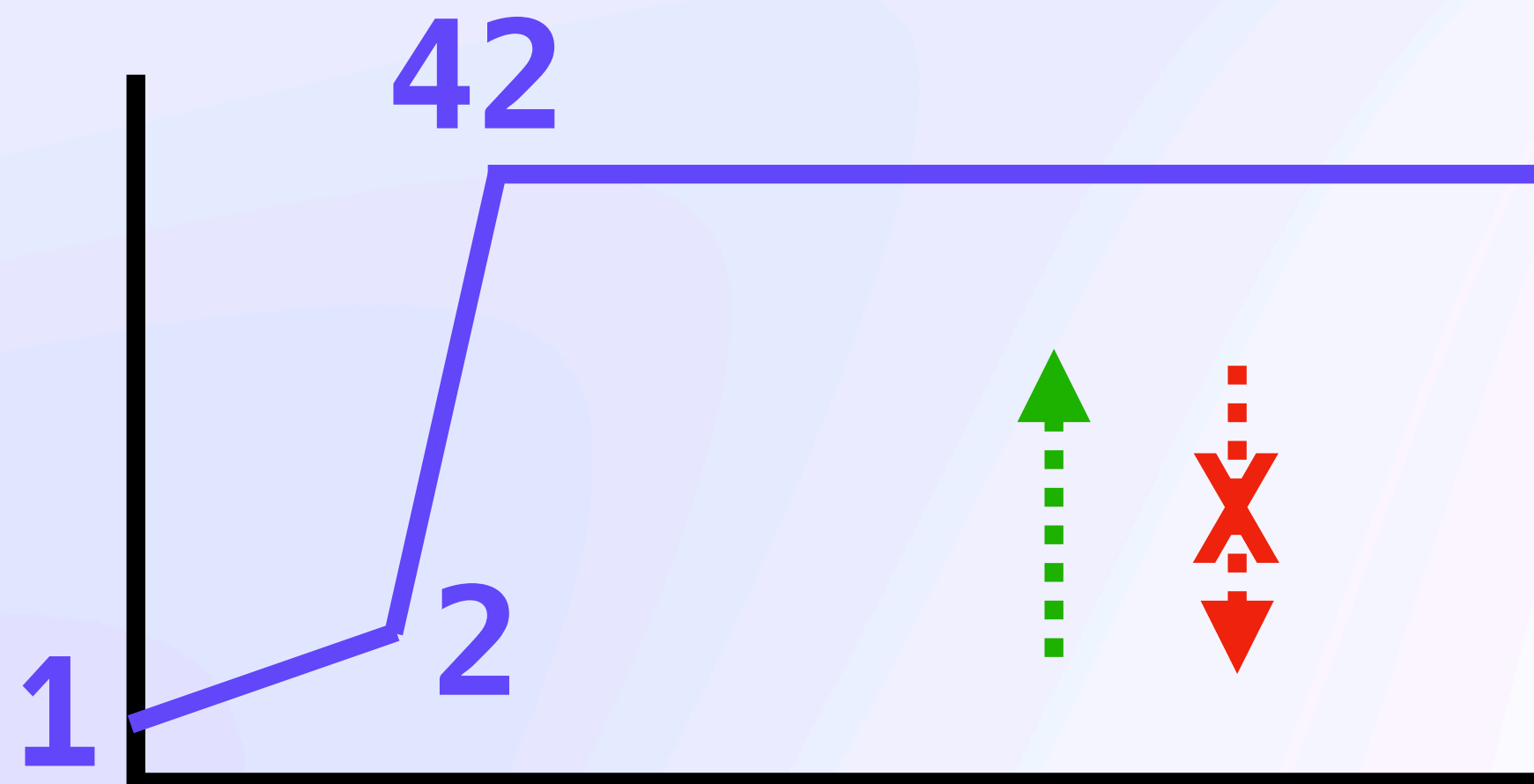
`max(42, 10)`

`max(11, 42)`

`set.add(item)`

`{}.add(1)`

`{1}.add(9)`



# Disorderly Systems

## *Monotonicity*

$\max(a, b)$

$\max(1, \underline{2})$

$\max(2, \underline{42})$

$\max(\underline{42}, 10)$

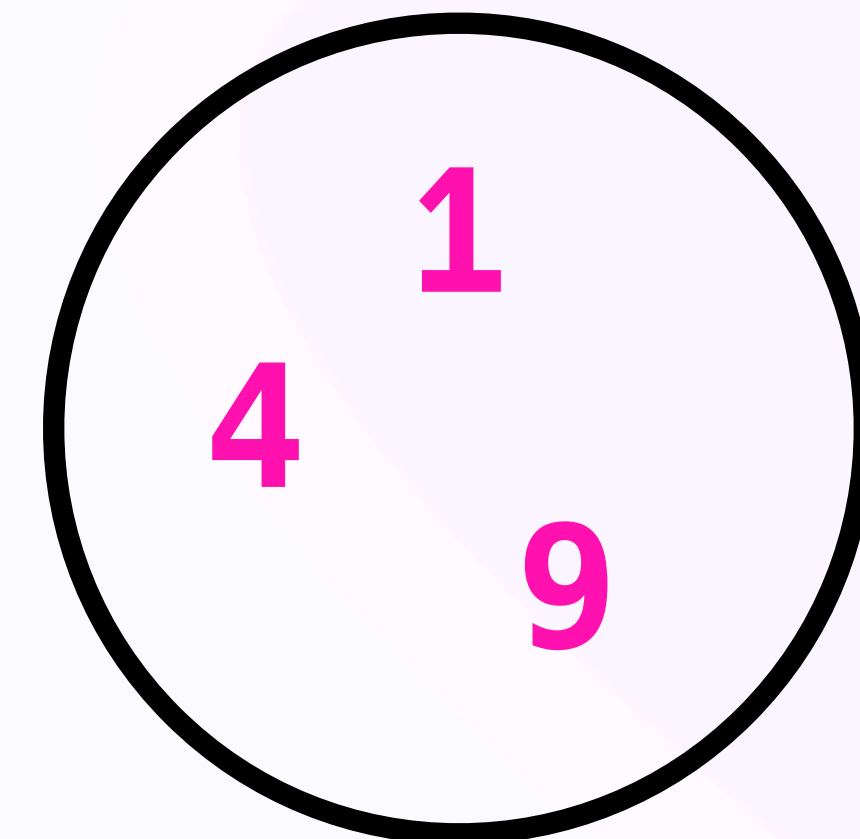
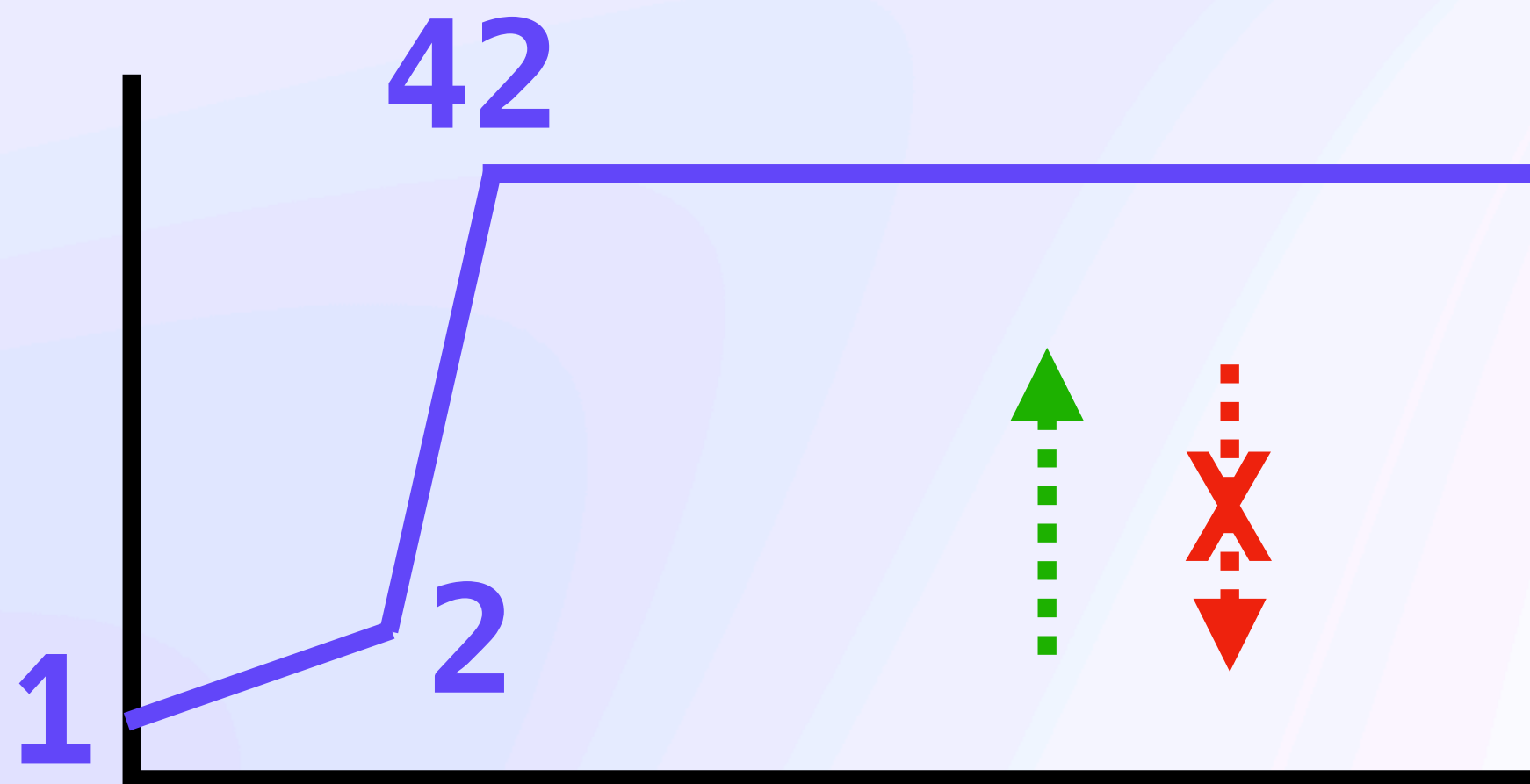
$\max(11, \underline{42})$

`set.add(item)`

`{}.add(1)`

`{1}.add(9)`

`{1, 9}.add(4)`



# Disorderly Systems

## *Monotonicity*

$\max(a, b)$

$\max(1, \underline{2})$

$\max(2, \underline{42})$

$\max(\underline{42}, 10)$

$\max(11, \underline{42})$

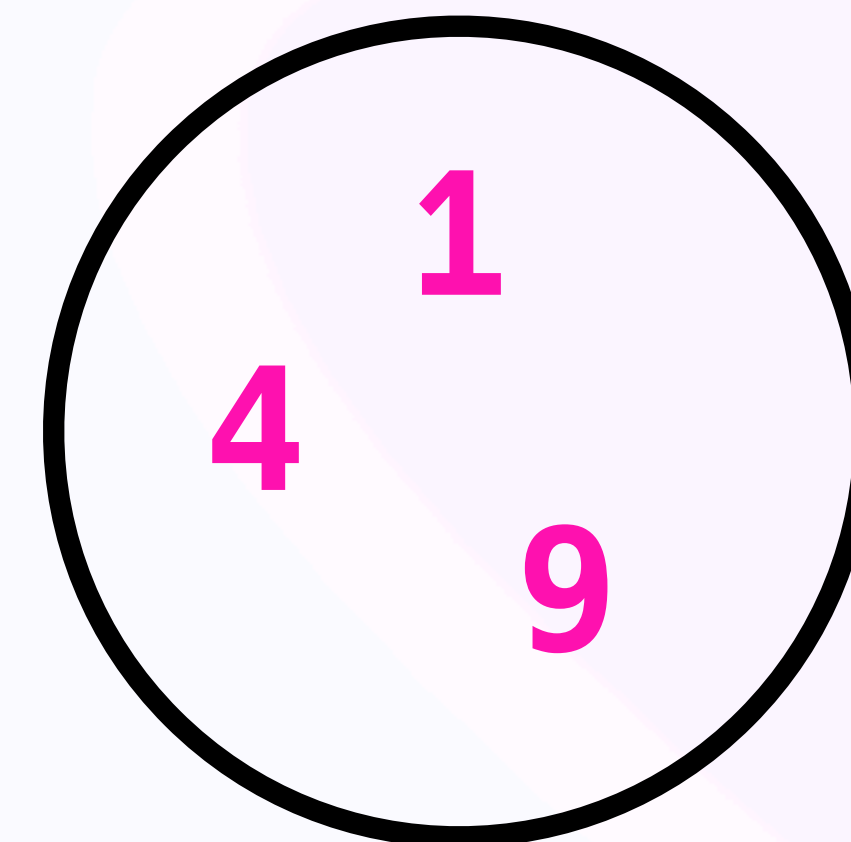
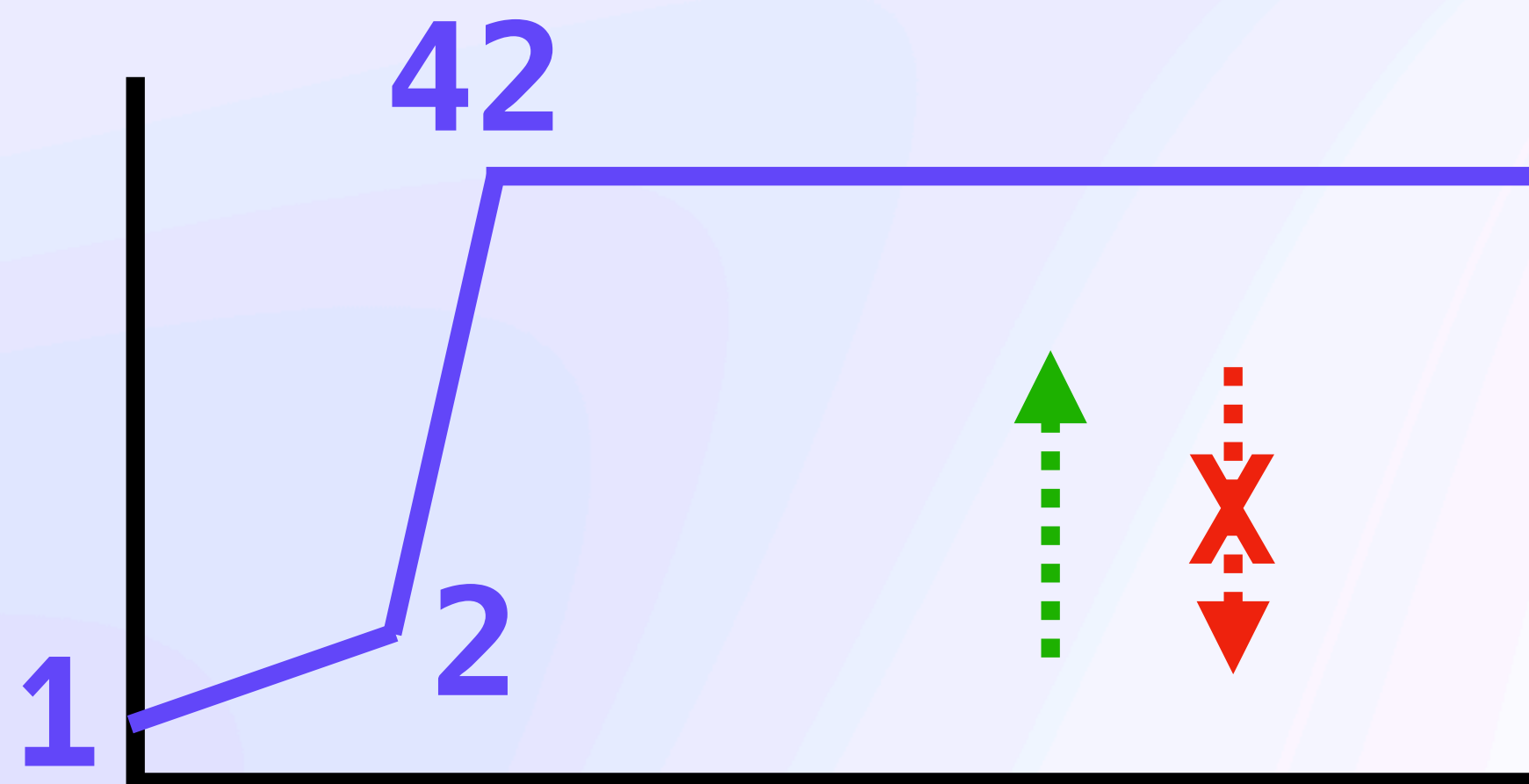
`set.add(item)`

`{}.add(1)`

`{1}.add(9)`

`{1, 9}.add(4)`

`{1, 4, 9}.add(9)`





# Disorderly Systems

## *Monotonicity*

$\max(a, b)$

$\max(1, \underline{2})$

$\max(2, \underline{42})$

$\max(\underline{42}, 10)$

$\max(11, \underline{42})$

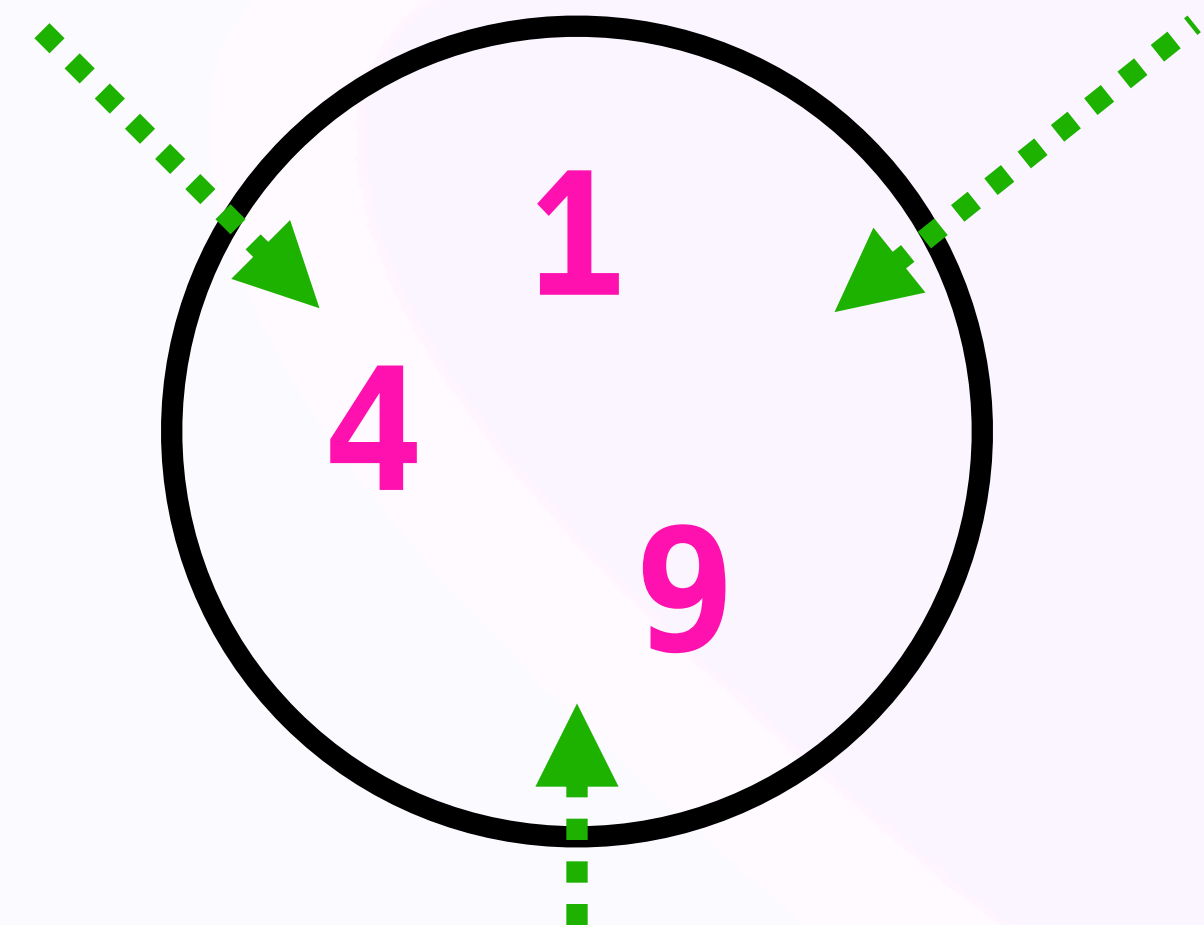
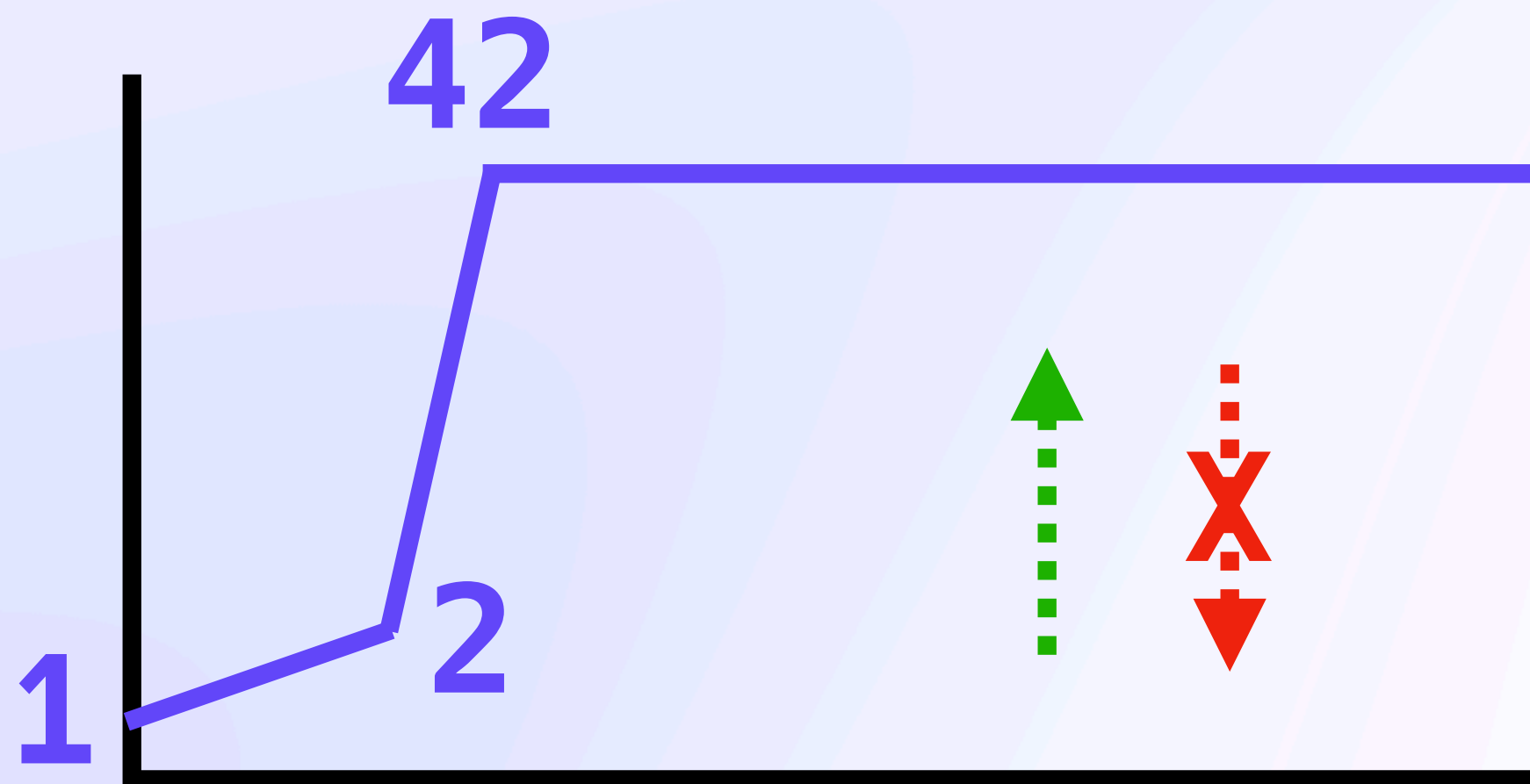
`set.add(item)`

`{}.add(1)`

`{1}.add(9)`

`{1, 9}.add(4)`

`{1, 4, 9}.add(9)`



# Disorderly Systems

## *Monotonicity*

$\max(a, b)$

$\max(1, \underline{2})$

$\max(2, \underline{42})$

$\max(\underline{42}, 10)$

$\max(11, \underline{42})$

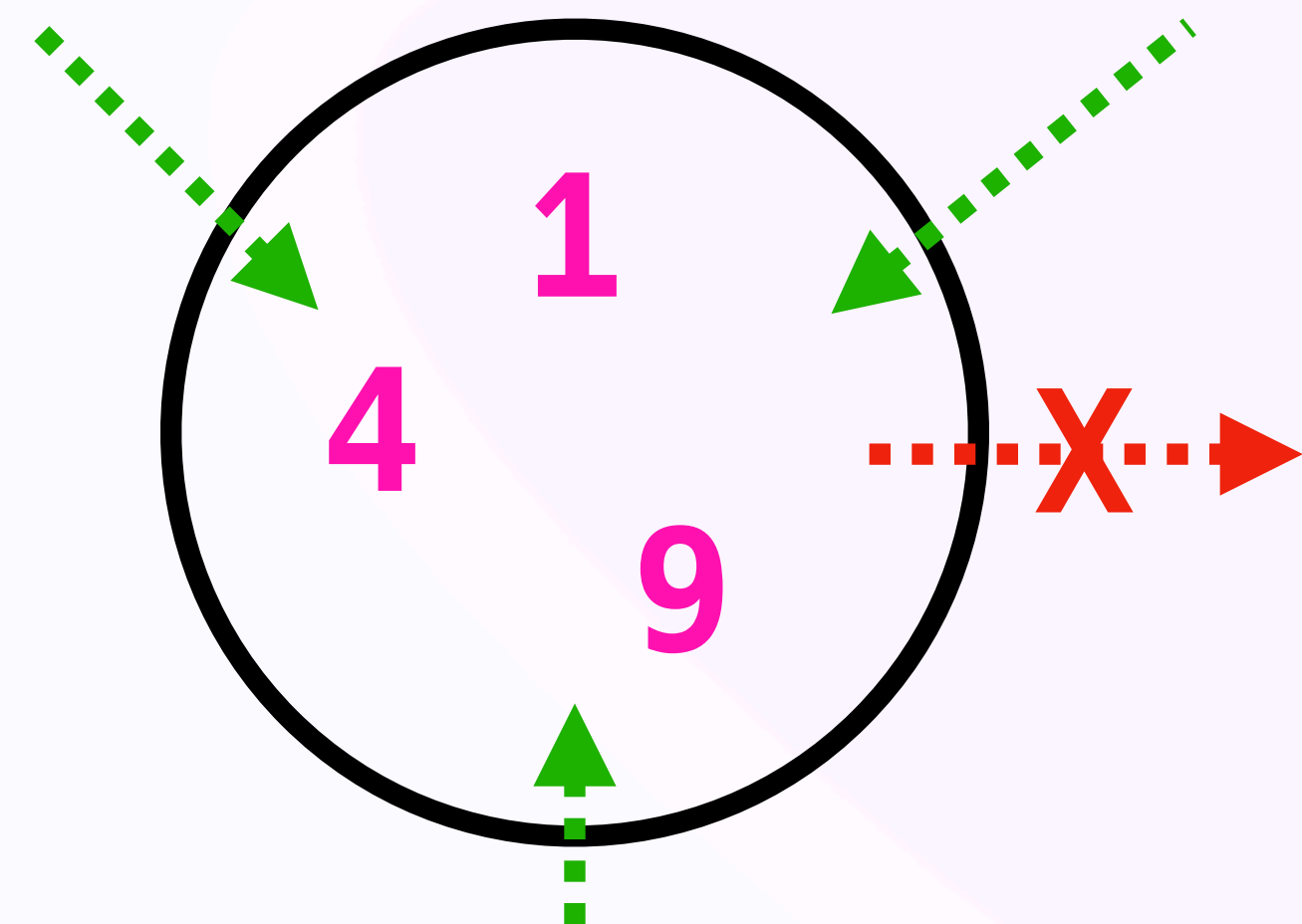
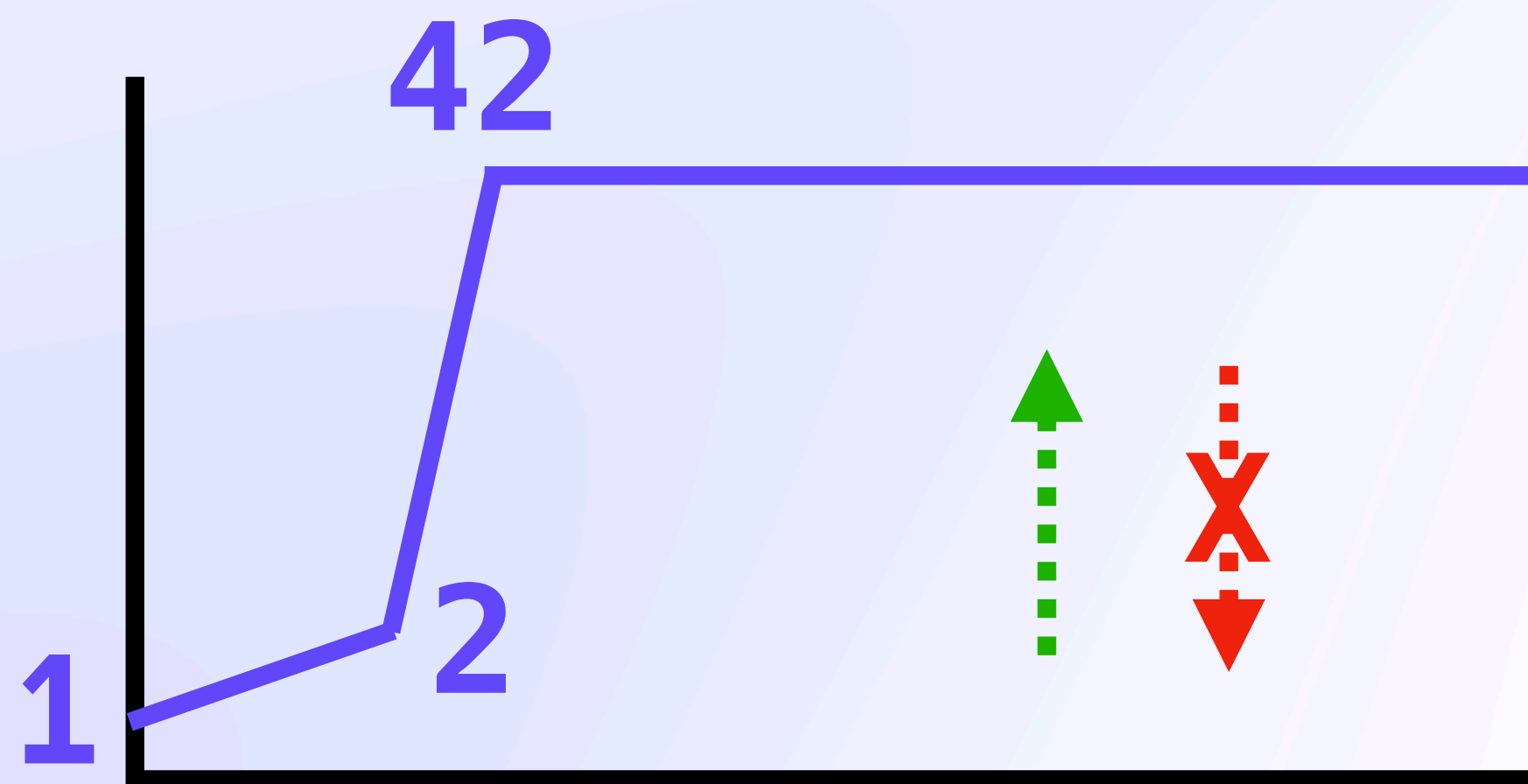
`set.add(item)`

`{}.add(1)`

`{1}.add(9)`

`{1, 9}.add(4)`

`{1, 4, 9}.add(9)`



Disorderly Systems

***Unafraid of Change*** 🤨

Disorderly Systems

***Unafraid of Change*** 🤨



**Automerge**

Disorderly Systems

***Unafraid of Change*** 🤔

```
automerge.change(doc, tx => {
```



**Automerge**

Disorderly Systems

***Unafraid of Change*** 🤔

```
automerge.change(doc, tx => {  
  automerge.splice(tx, ["text"], 0, 0, "Hello ")  
})
```



**Automerge**

Disorderly Systems

***Unafraid of Change*** 🤔

```
automerge.change(doc, tx => {  
  automerge.splice(tx, ["text"], 0, 0, "Hello ")  
  tx.counter.increment(20)  
})
```



**Automerge**

## Disorderly Systems

# *Unafraid of Change* 🤔

```
automerge.change(doc, tx => {  
  automerge.splice(tx, ["text"], 0, 0, "Hello ")  
  tx.counter.increment(20)  
  tx.map.key = "new value"  
  tx.map.nested_map.key = "new nested value"  
})
```



**Automerge**



## Disorderly Systems

# *Unafraid of Change* 🤔

```
automerge.change(doc, tx => {  
  automerge.splice(tx, ["text"], 0, 0, "Hello ")  
  tx.counter.increment(20)  
  tx.map.key = "new value"  
  tx.map.nested_map.key = "new nested value"  
  tx.list[0] = "A"  
  tx.list.insertAt(0, "Z")  
  tx.list[4].nested = "MAP"  
  tx.list[5][0] = "NESTED LIST"  
})
```



**Automerge**

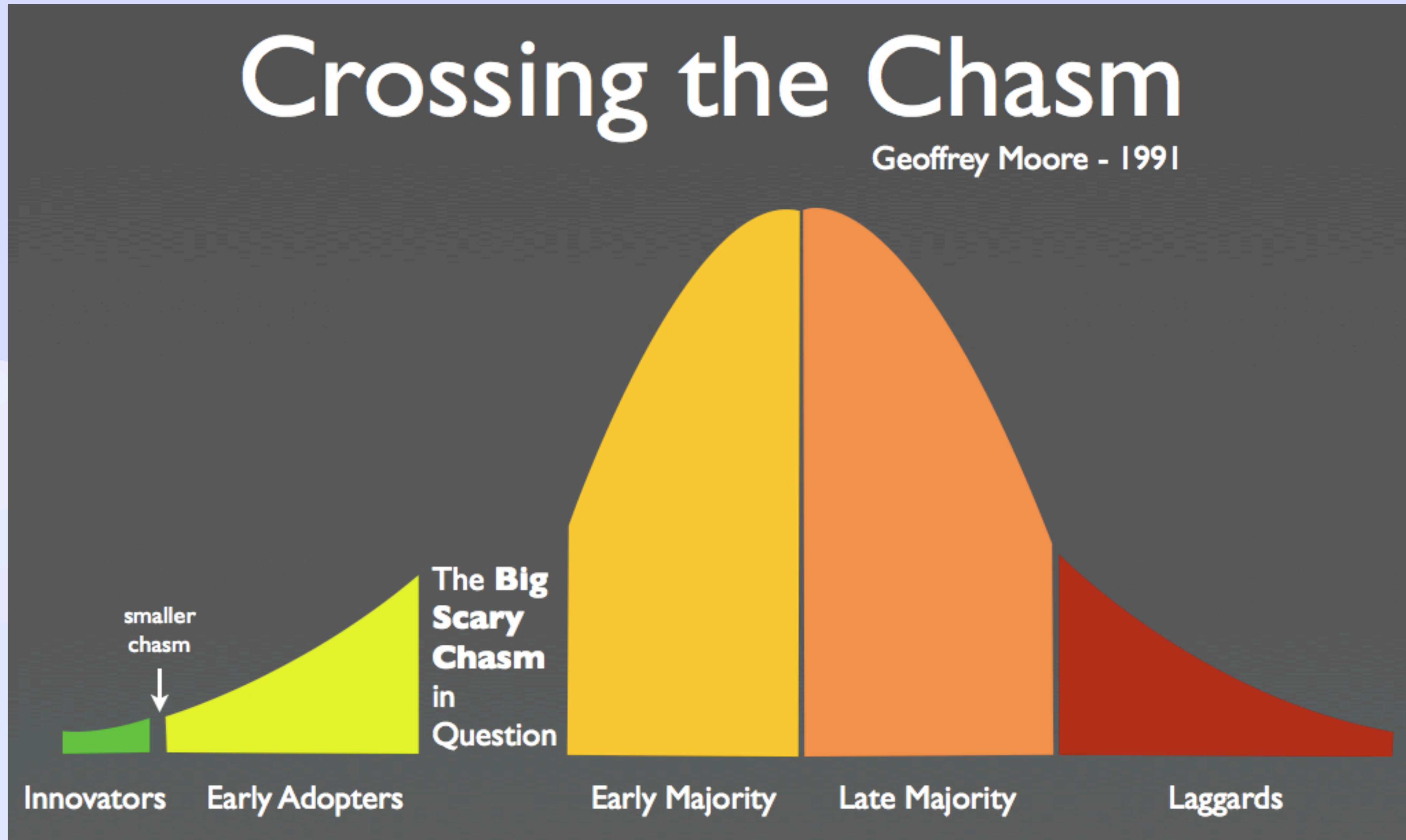
Fixing The Leaky Pipes... Statically

# *Access Control*



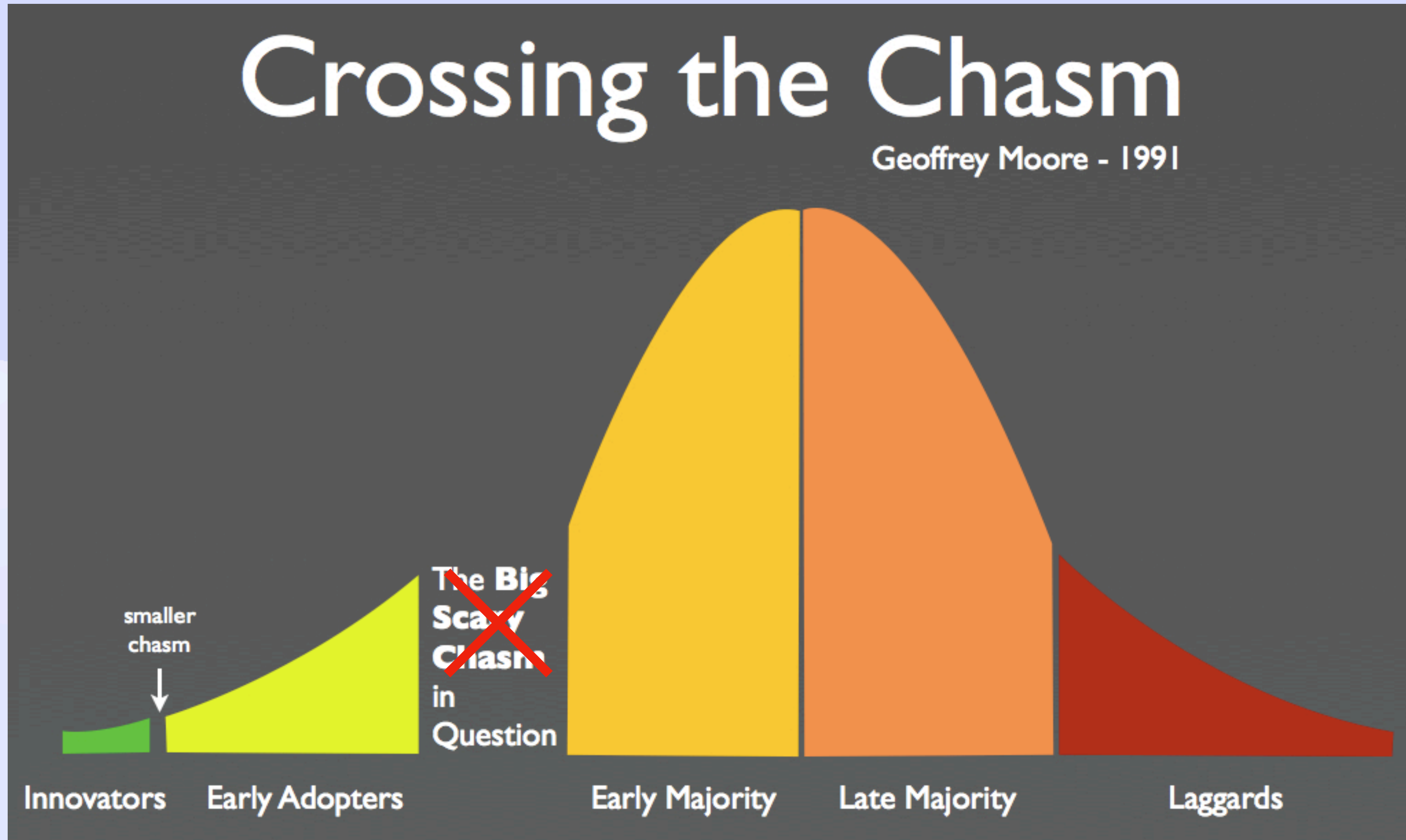
Access Control

# *Next Stage of Evolution*



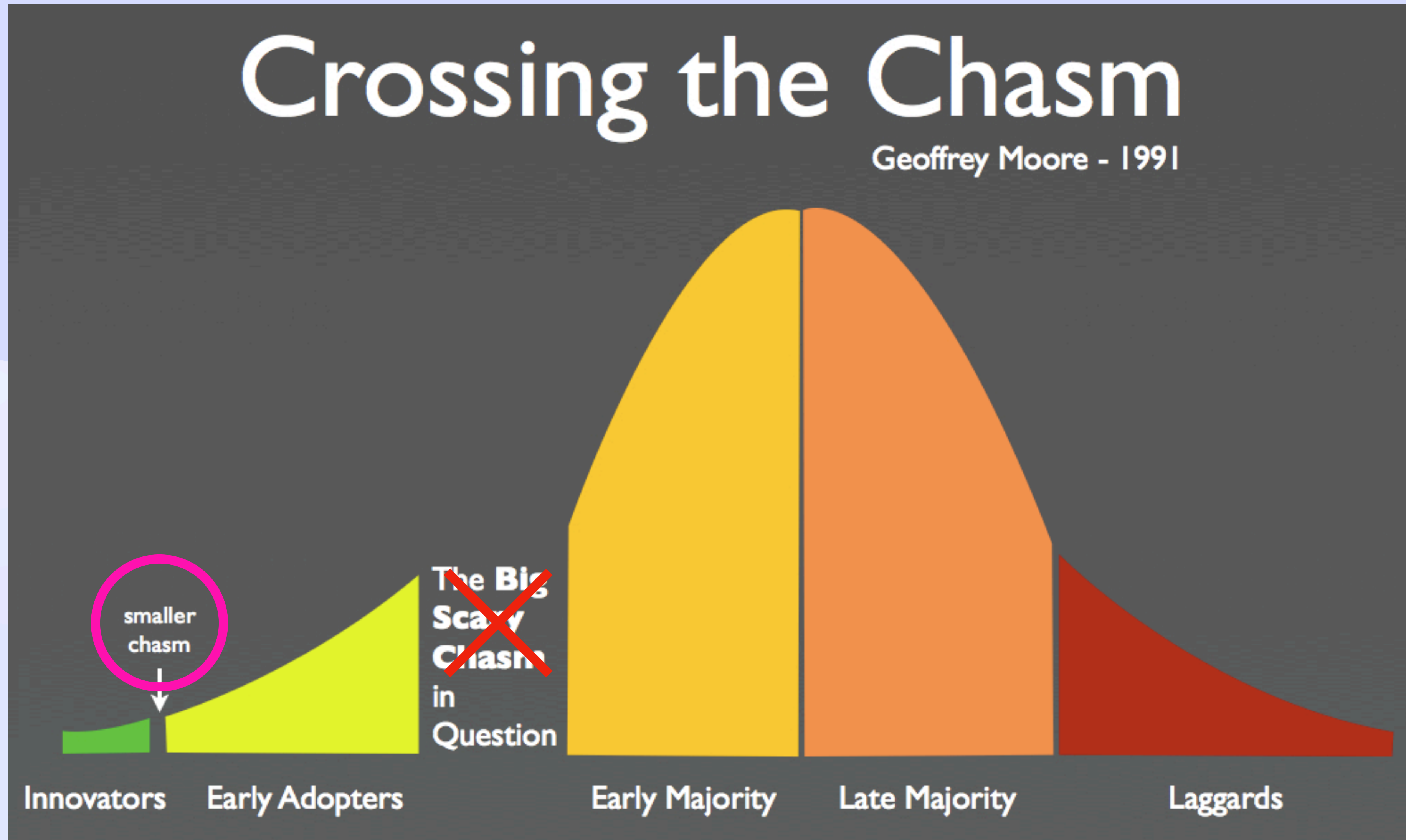
Access Control

# *Next Stage of Evolution*



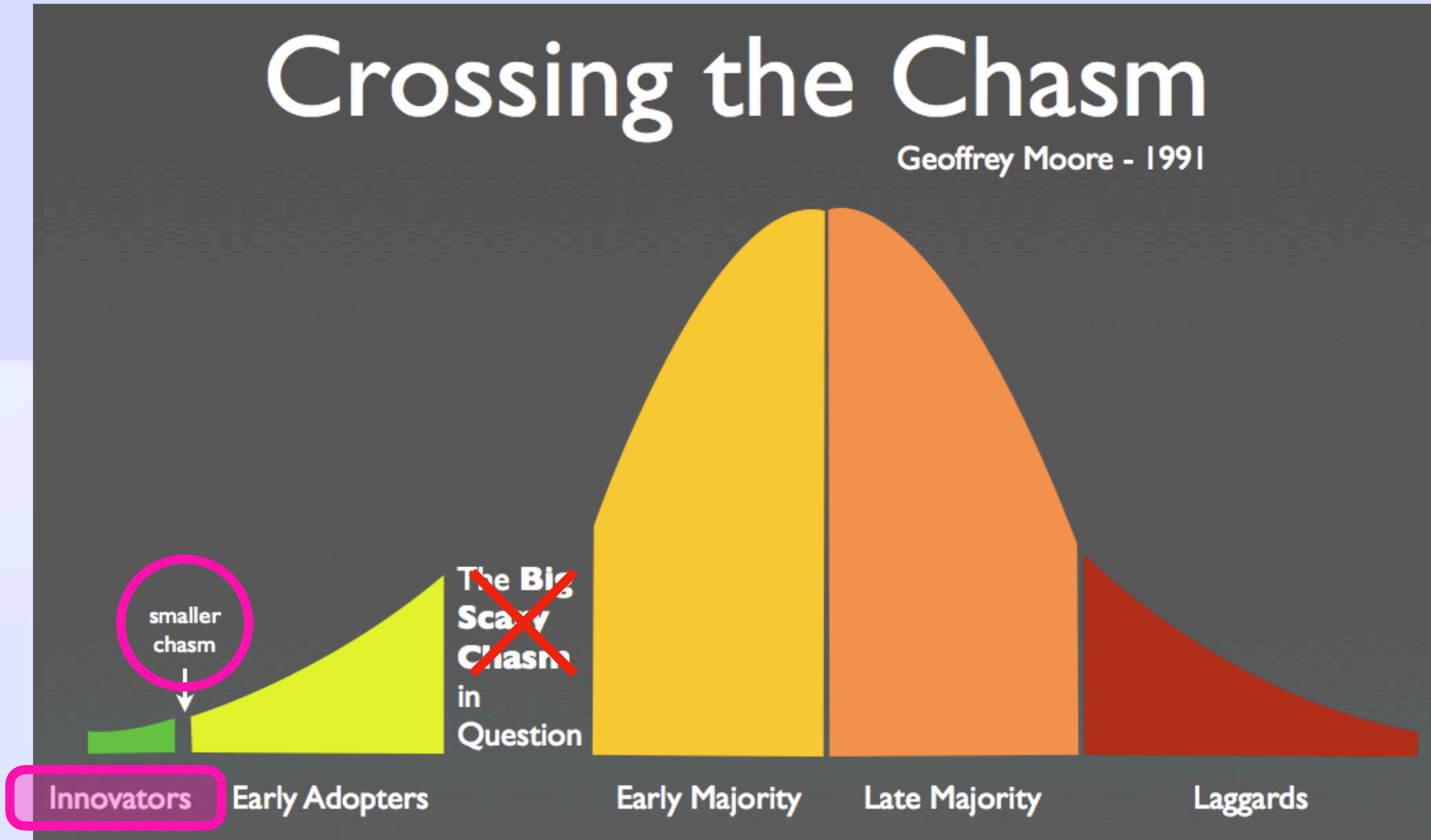
Access Control

# *Next Stage of Evolution*



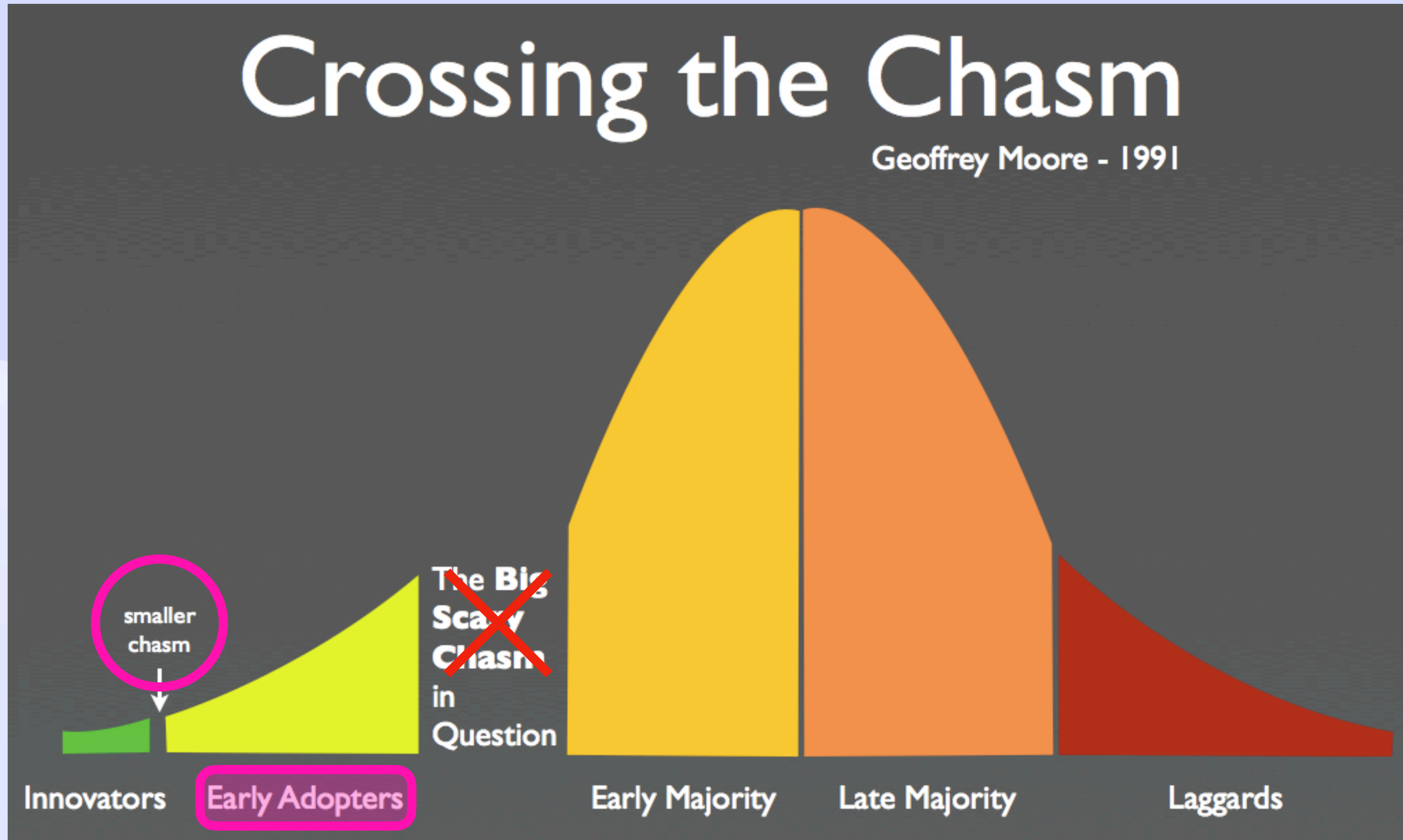
# Access Control

## *Next Stage of Evolution*



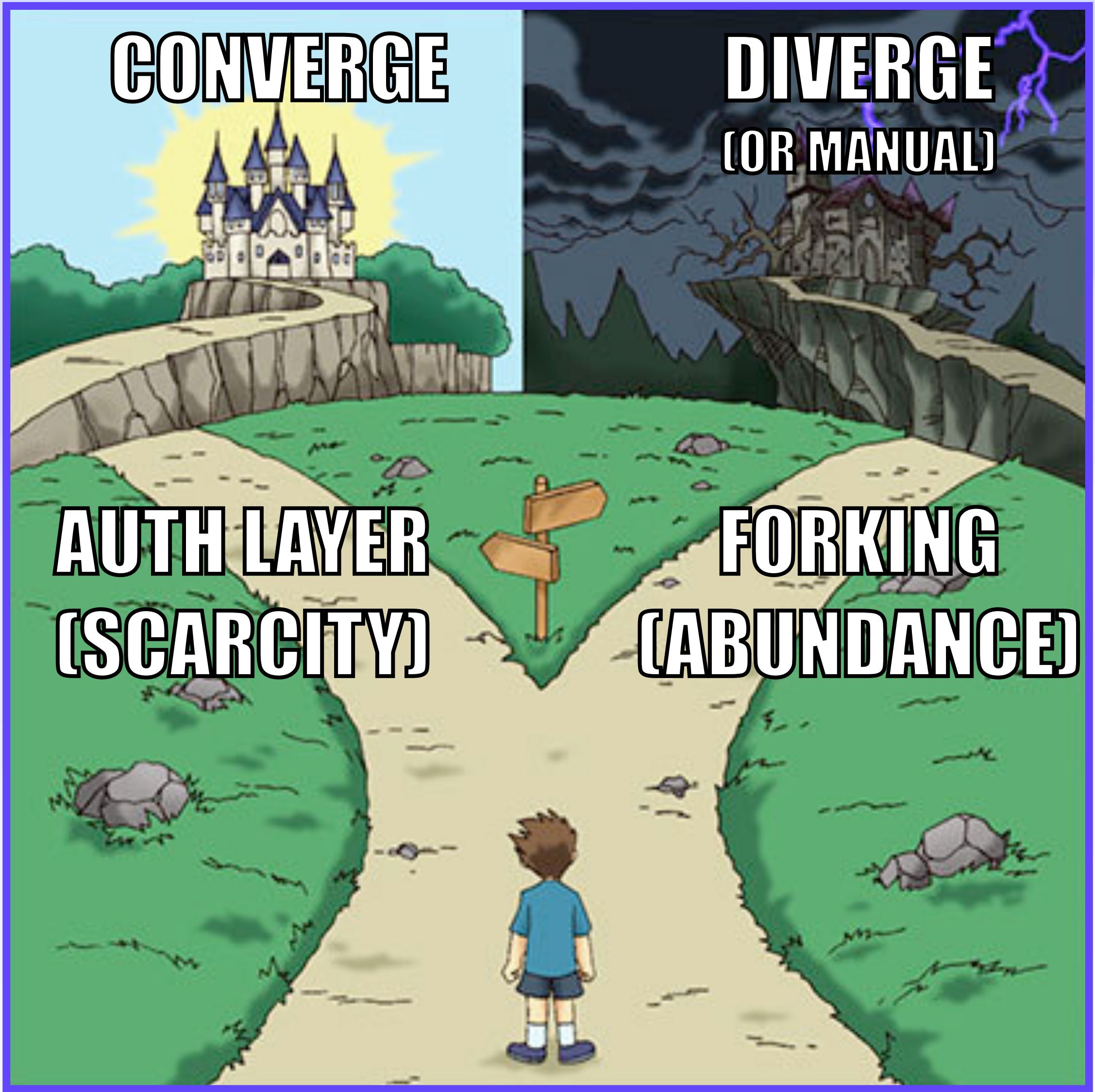
# Access Control

## *Next Stage of Evolution*



Access Control

# *Automatic Merging With Scarcity*





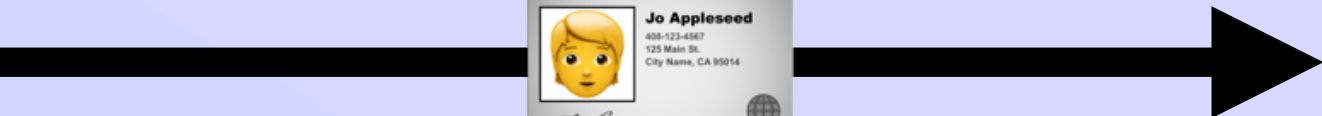
Access Control

# *Cloud Auth Flow*



# Access Control

# Cloud Auth Flow



A purple rounded rectangle containing authentication options. At the top left is a circular logo with 'OAUTH' and a large 'A'. To its right is a white button with an Apple logo and the text 'Sign In with Apple'. In the center is a yellow hand icon and a yellow emoji of a soldier in a red uniform with a black bearskin hat. At the bottom is a blue button with a Google 'G' logo and the text 'Sign in with Google'.



# Access Control

# Cloud Auth Flow



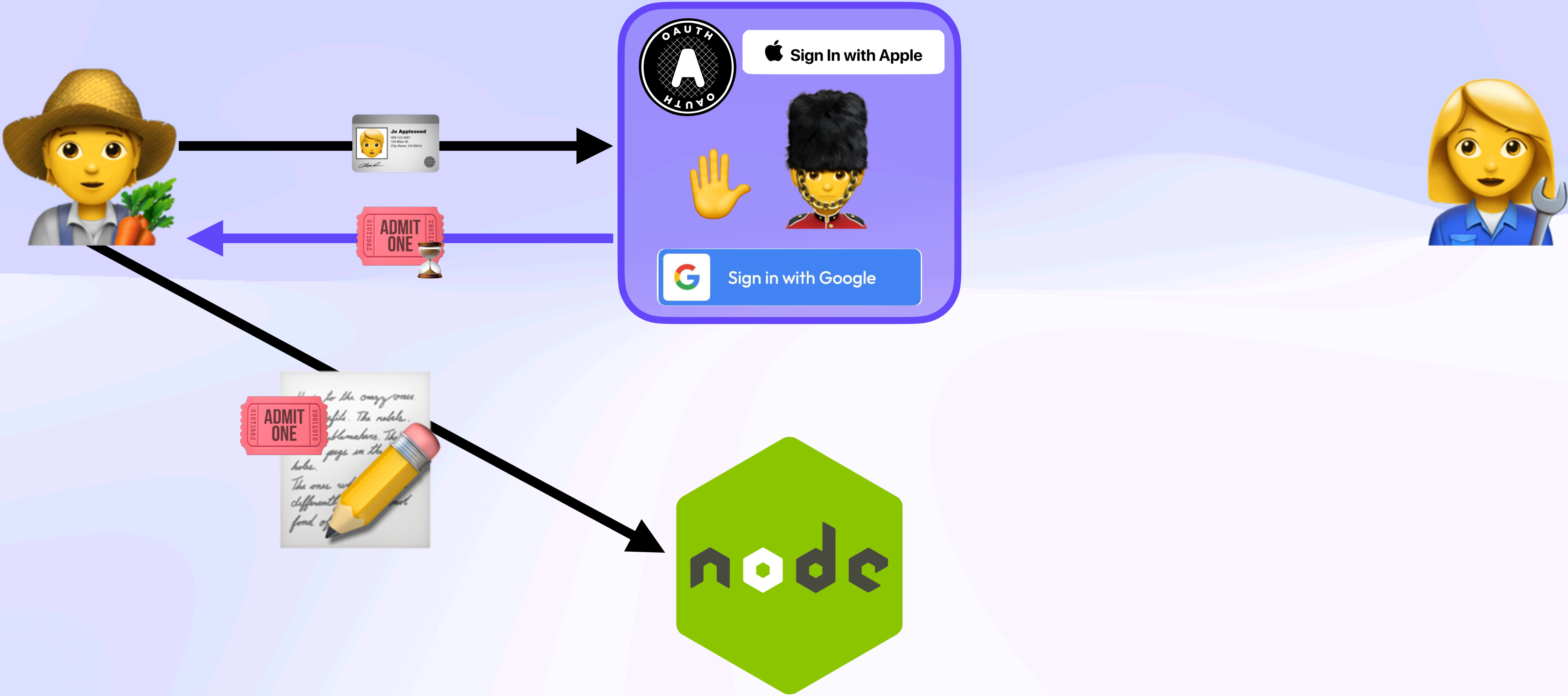
# Access Control

# Cloud Auth Flow



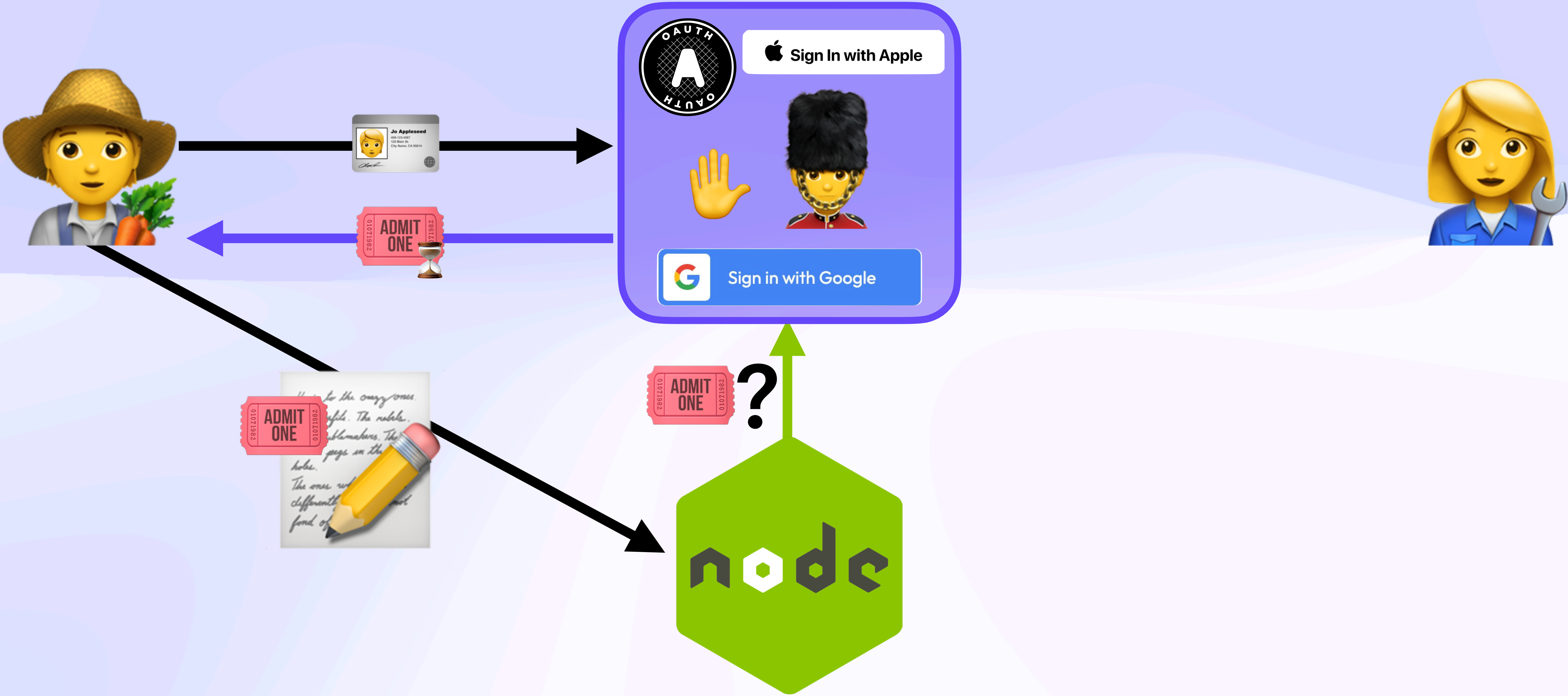
# Access Control

# Cloud Auth Flow



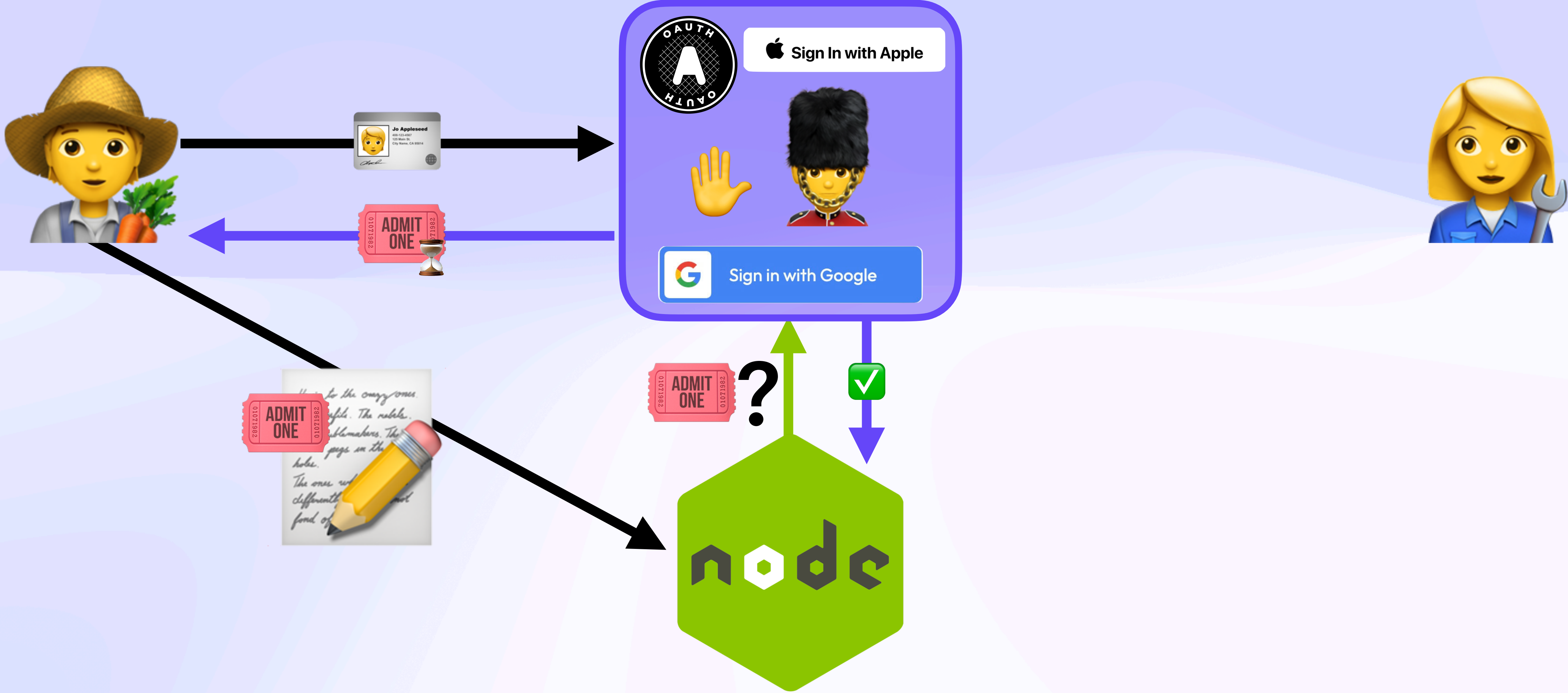
# Access Control

# Cloud Auth Flow



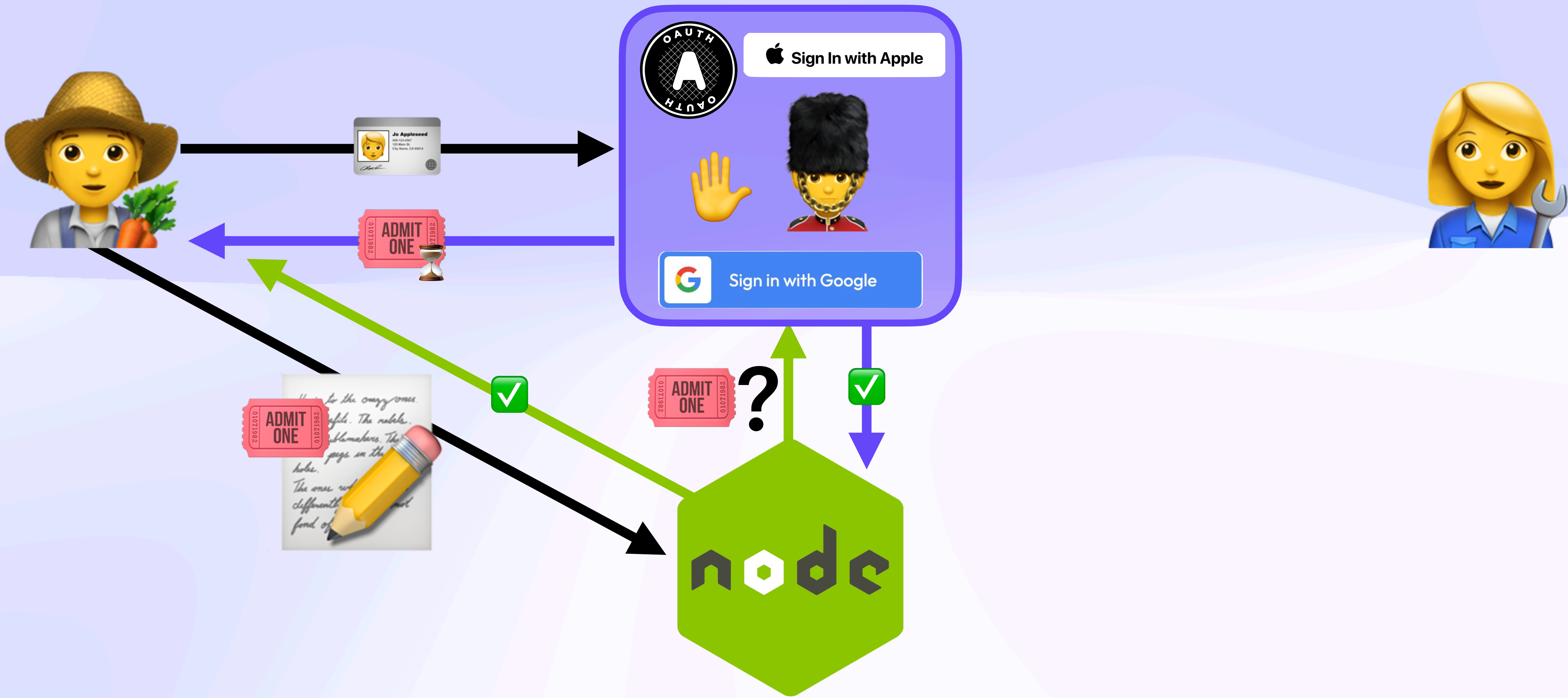
# Access Control

# Cloud Auth Flow



# Access Control

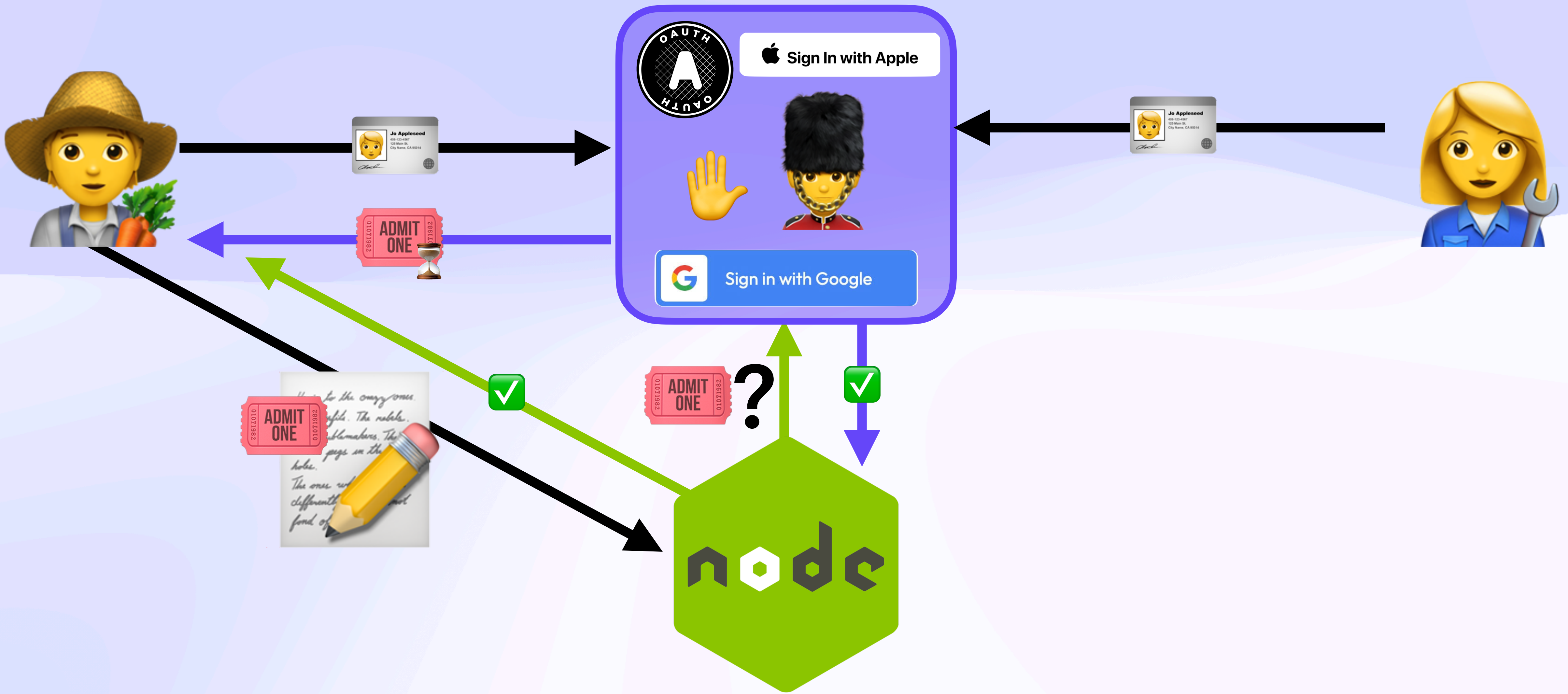
# Cloud Auth Flow





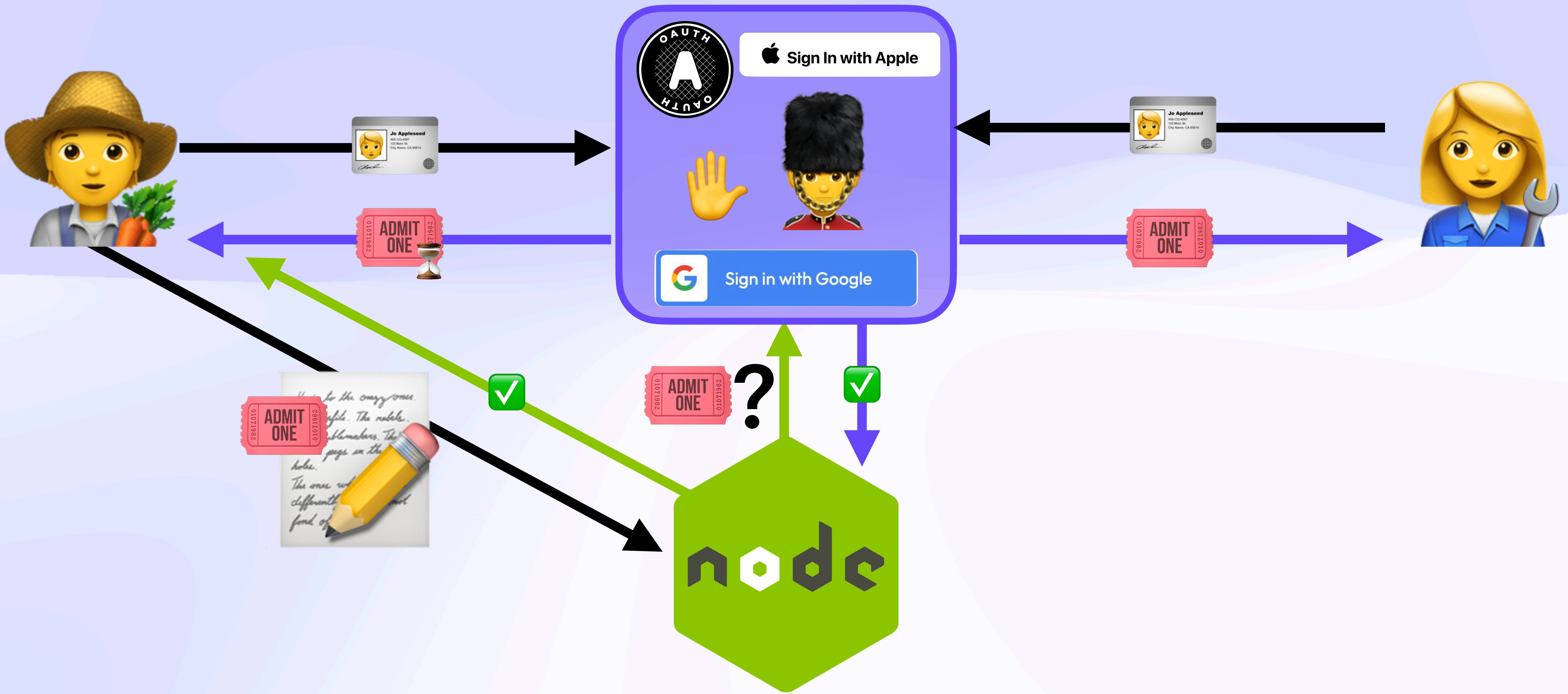
# Access Control

# Cloud Auth Flow



# Access Control

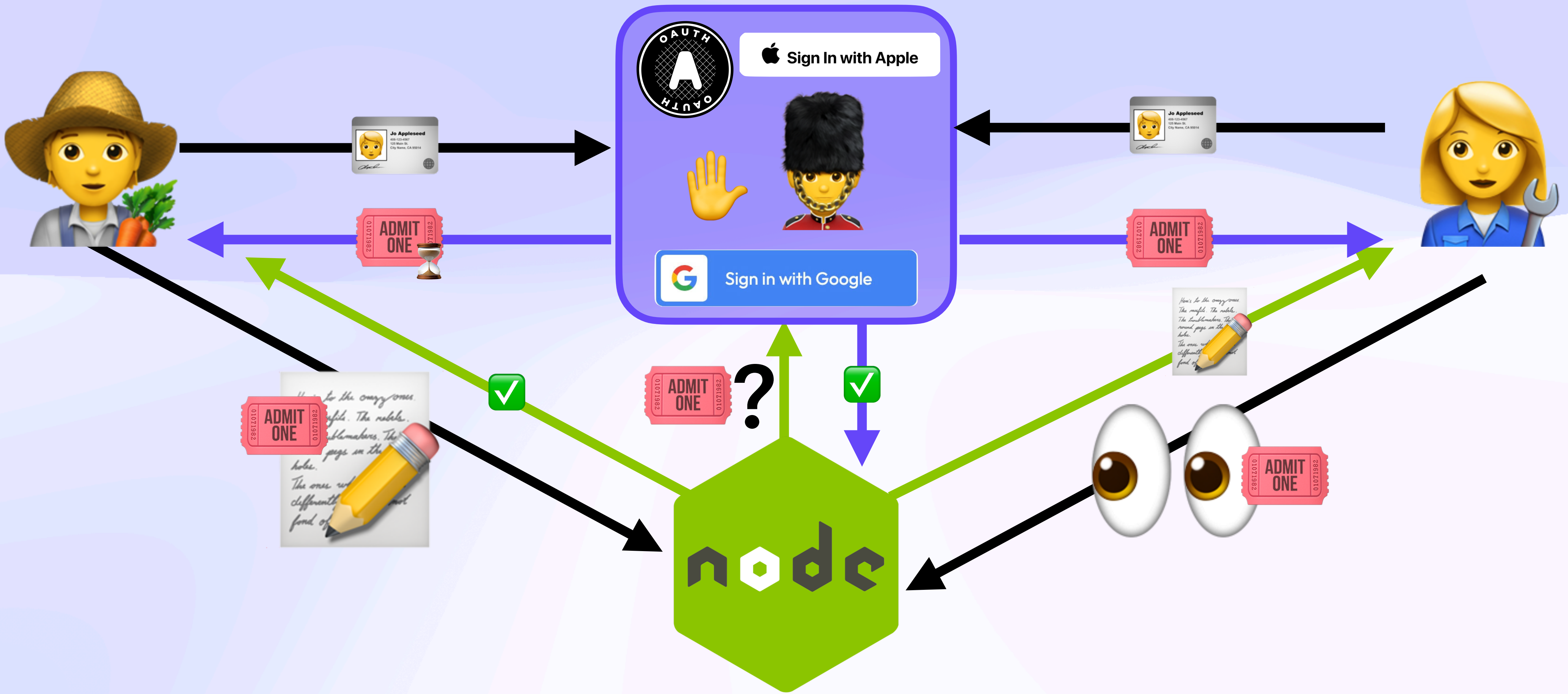
# Cloud Auth Flow





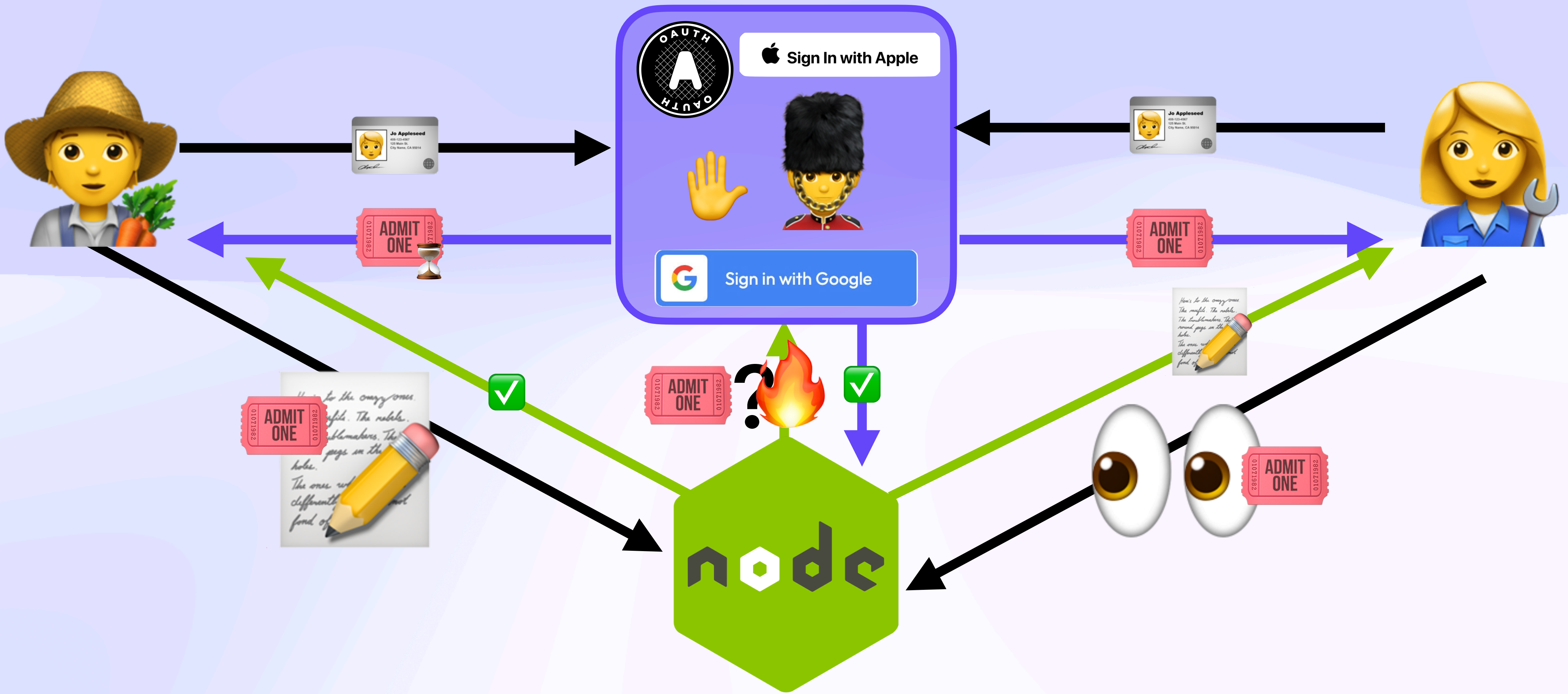
# Access Control

# Cloud Auth Flow



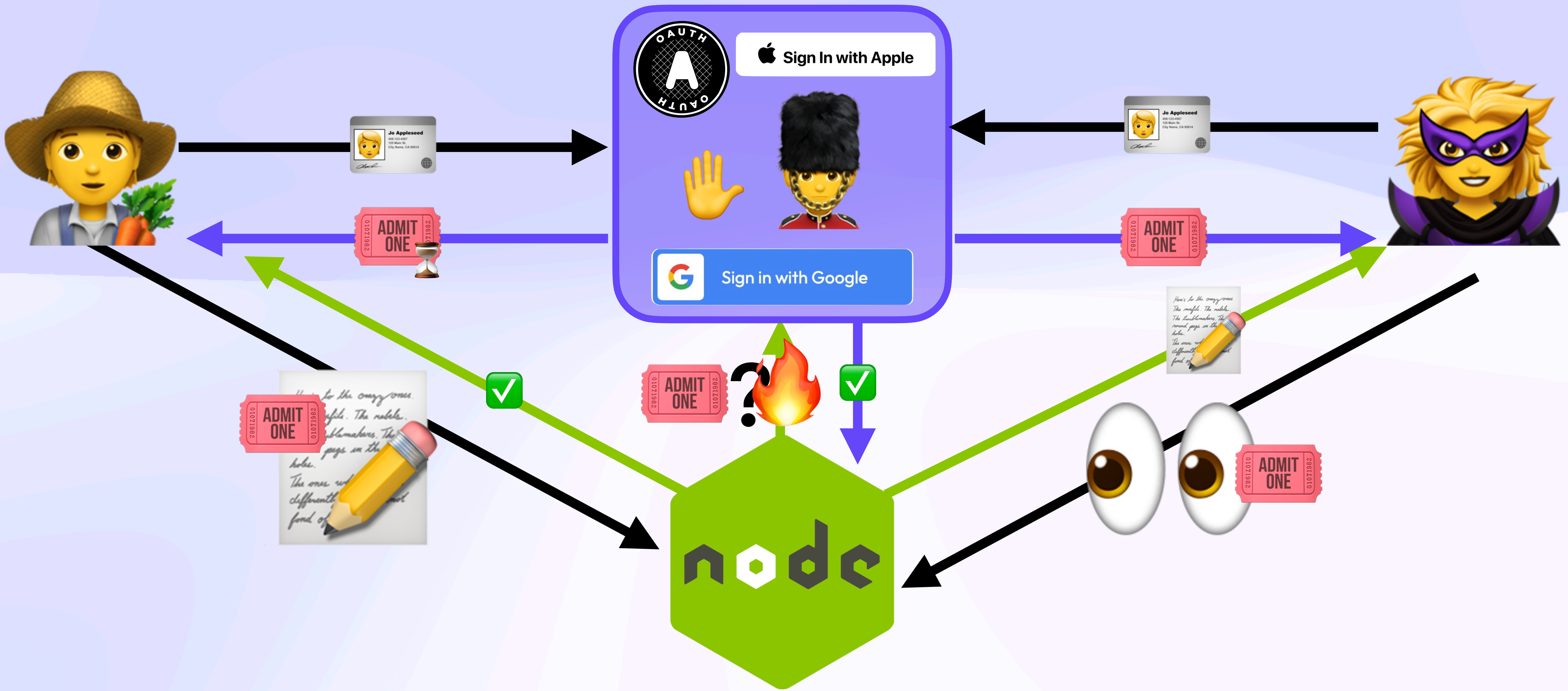
# Access Control

# Cloud Auth Flow



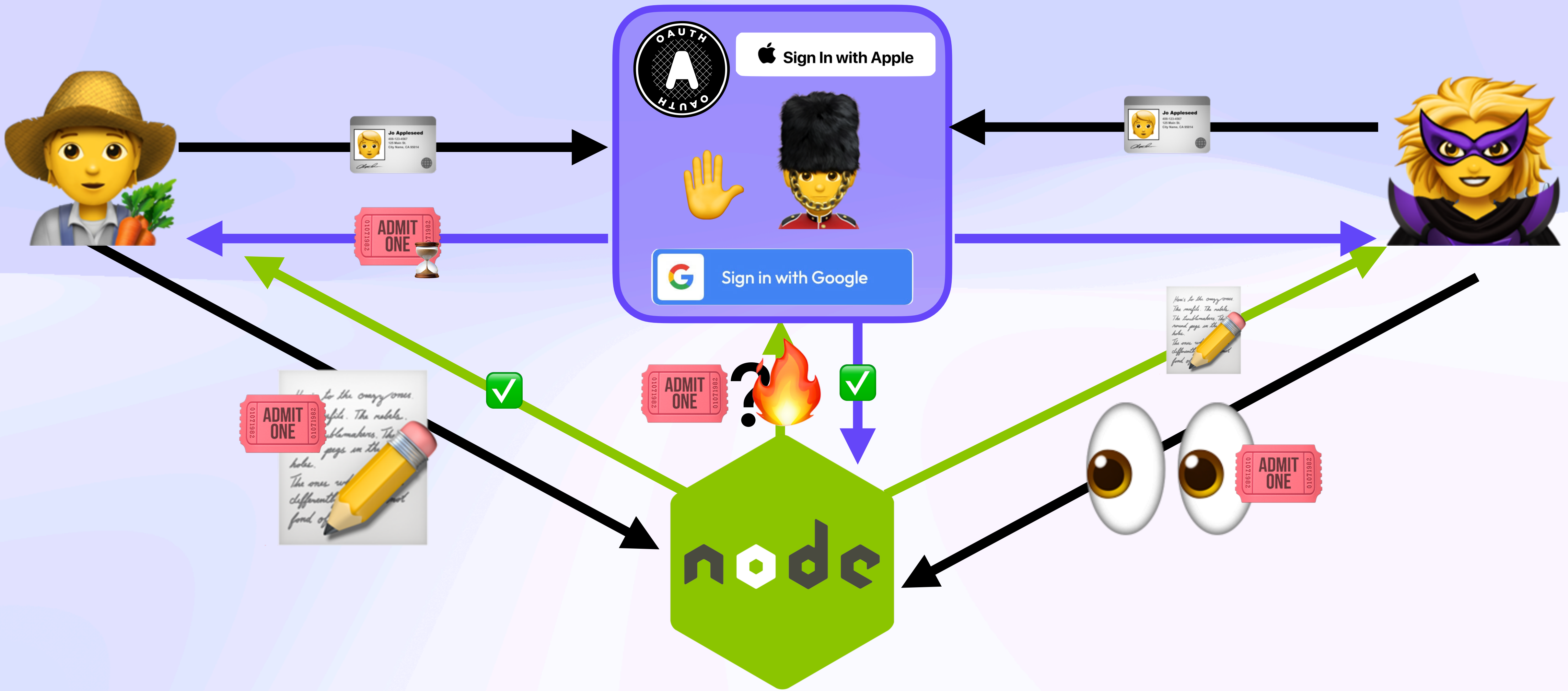
# Access Control

# Cloud Auth Flow



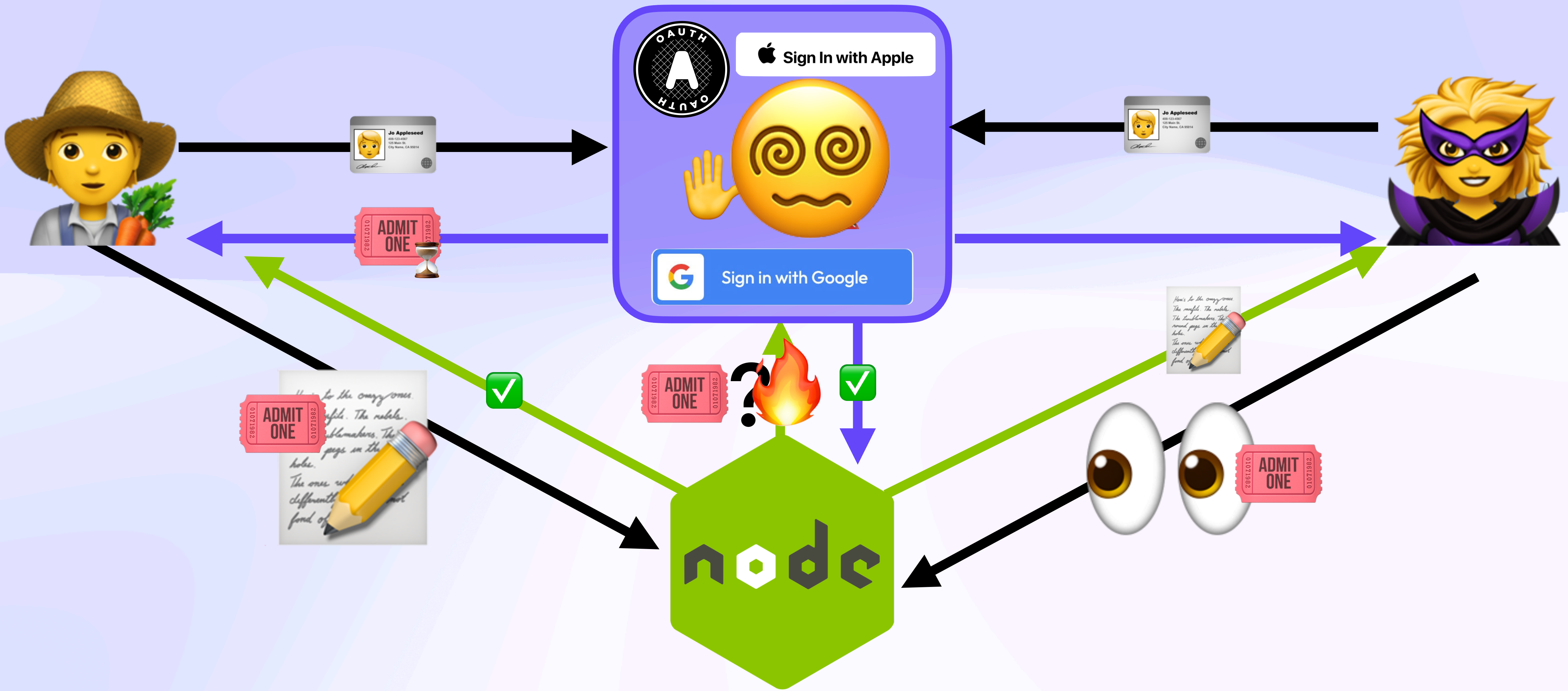
# Access Control

# Cloud Auth Flow



# Access Control

# Cloud Auth Flow





Access Control

***Cloud: Auth-as-Place***



# Access Control

# Cloud: Auth-as-Place



A purple rounded rectangle representing an authentication interface. It features a circular 'A' logo with 'OAUTH' text, a 'Sign In with Apple' button with the Apple logo, a hand icon, a soldier emoji, and a 'Sign in with Google' button with the Google logo.

# Access Control

# Cloud: Auth-as-Place

"Over Here"



"Over There"



# Access Control

# Cloud: Auth-as-Place

"Over Here"



"Over There"

A purple rounded rectangle representing an authentication interface. It contains:

- A circular logo with 'A' and 'AUTH'.
- A button: "Sign In with Apple" with the Apple logo.
- A hand icon and a soldier emoji.
- A button: "Sign in with Google" with the Google logo.

Access Control

***Local-First: In Pictures***



# Access Control

# *Local-First: In Pictures*



# Access Control

# Local-First: In Pictures



Access Control

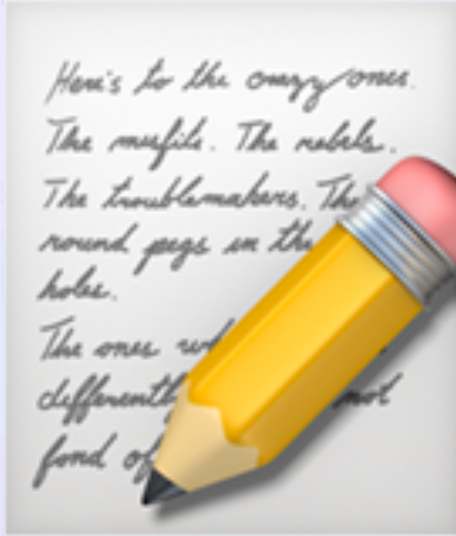
***Auth as Data: "Auth Must Travel with Data"***





# Access Control

# ***Auth as Data: "Auth Must Travel with Data"***



# Access Control

## ***Auth as Data: "Auth Must Travel with Data"***



# Access Control

## ***Auth as Data: "Auth Must Travel with Data"***



# Access Control

# ***Auth as Data: "Auth Must Travel with Data"***



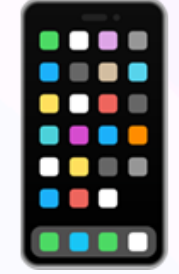
# Access Control

## ***Auth as Data: "Auth Must Travel with Data"***

"Auth Here"  
"Data Here"



"Auth There"  
"Data There"



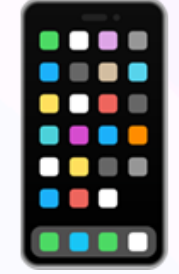
# Access Control

## ***Auth as Data: "Auth Must Travel with Data"***

"Auth Here"  
"Data Here"



"Auth There"  
"Data There"



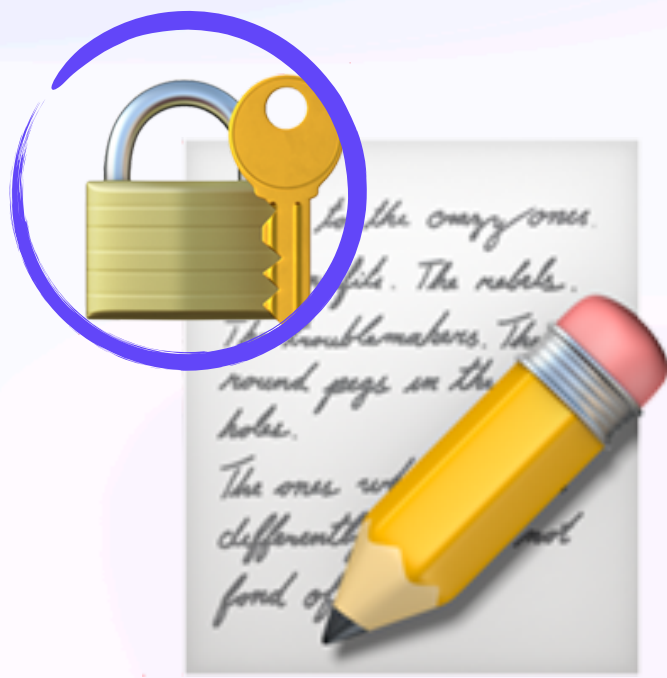
# Access Control

## ***Auth as Data: "Auth Must Travel with Data"***

"Auth Here"  
"Data Here"



"Auth There"  
"Data There"



# Access Control



Access Control

***Cloud***

***Local-First***

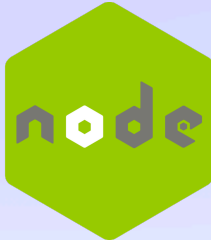
# Access Control


*Cloud*

Auth 



Compute 



Data 




*Local-First*

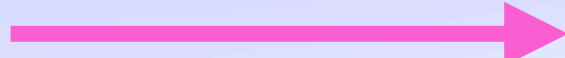
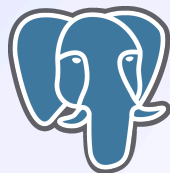
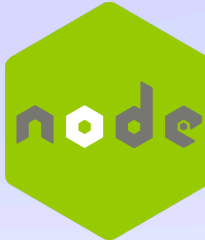
# Access Control

## Cloud

Auth 


Compute 

Data 



## Local-First

Compute 


Data 

# Access Control


## Cloud


Auth 

Compute 

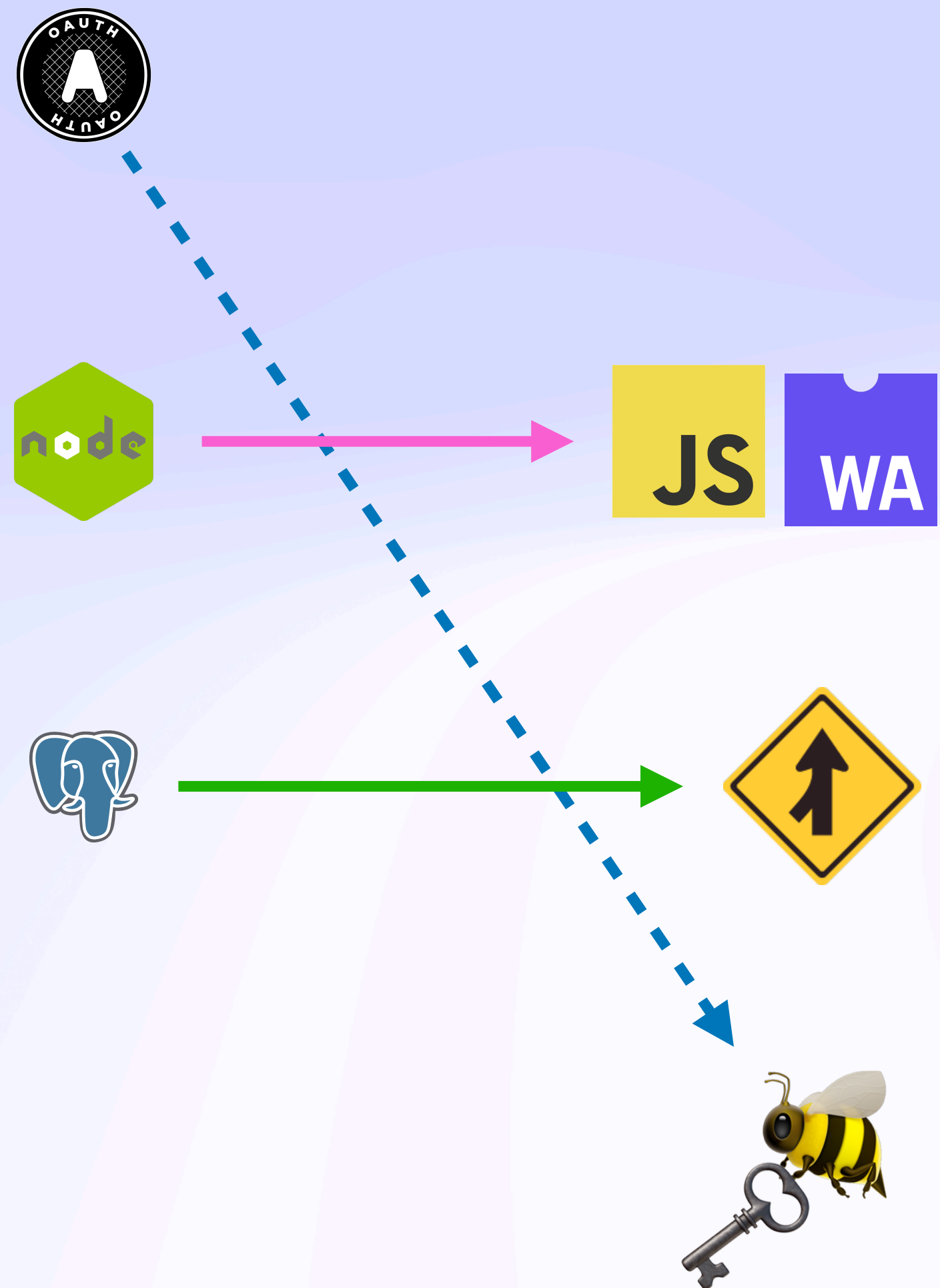
Data 

## Local-First

Compute 

Data 

Auth 



# Access Control



## Access Control

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

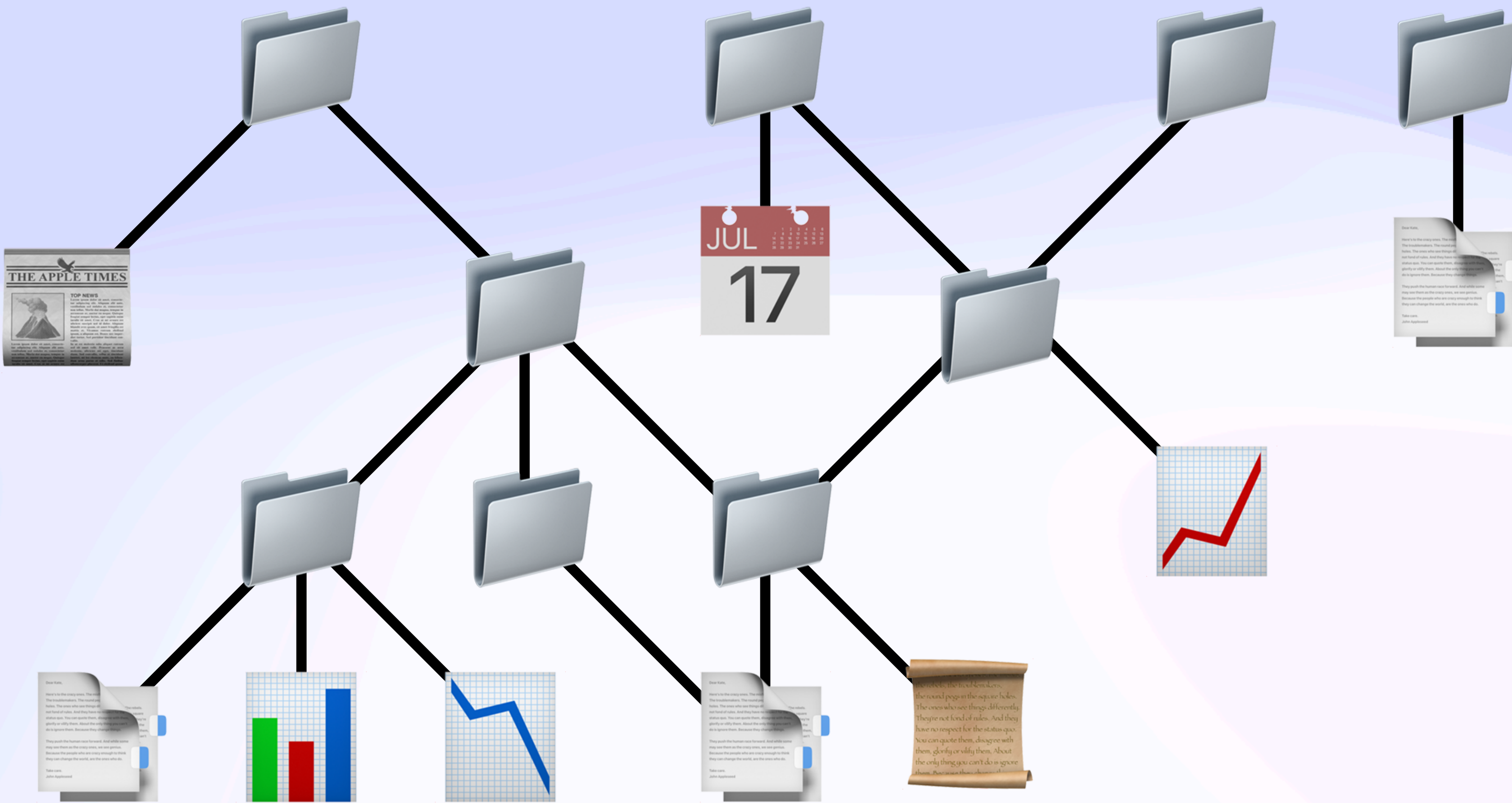
Lea Kissner, Google's Global Lead of Privacy Technologies

Access Control

***Universal Read Control***

# Access Control

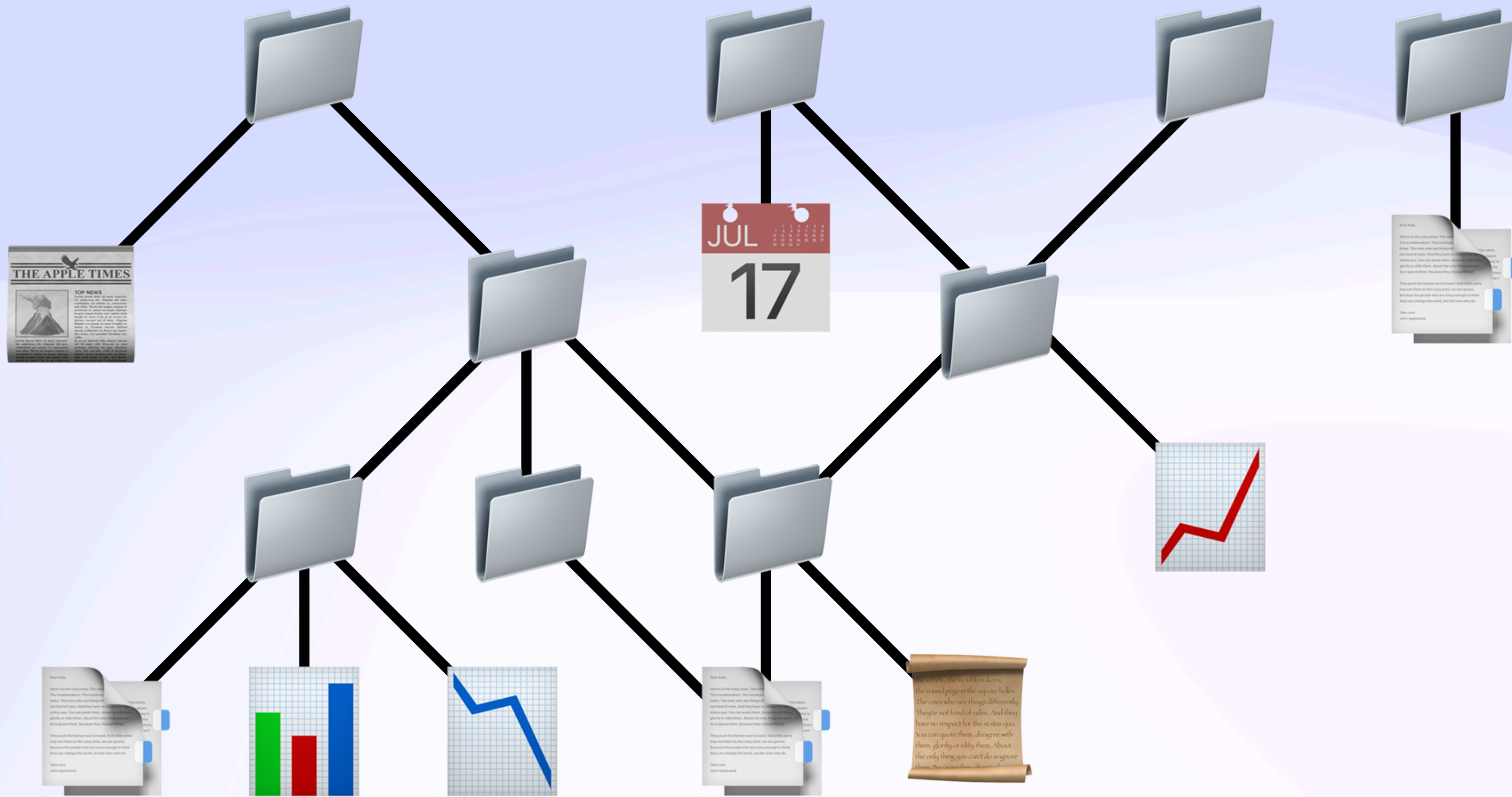
# *Universal Read Control*





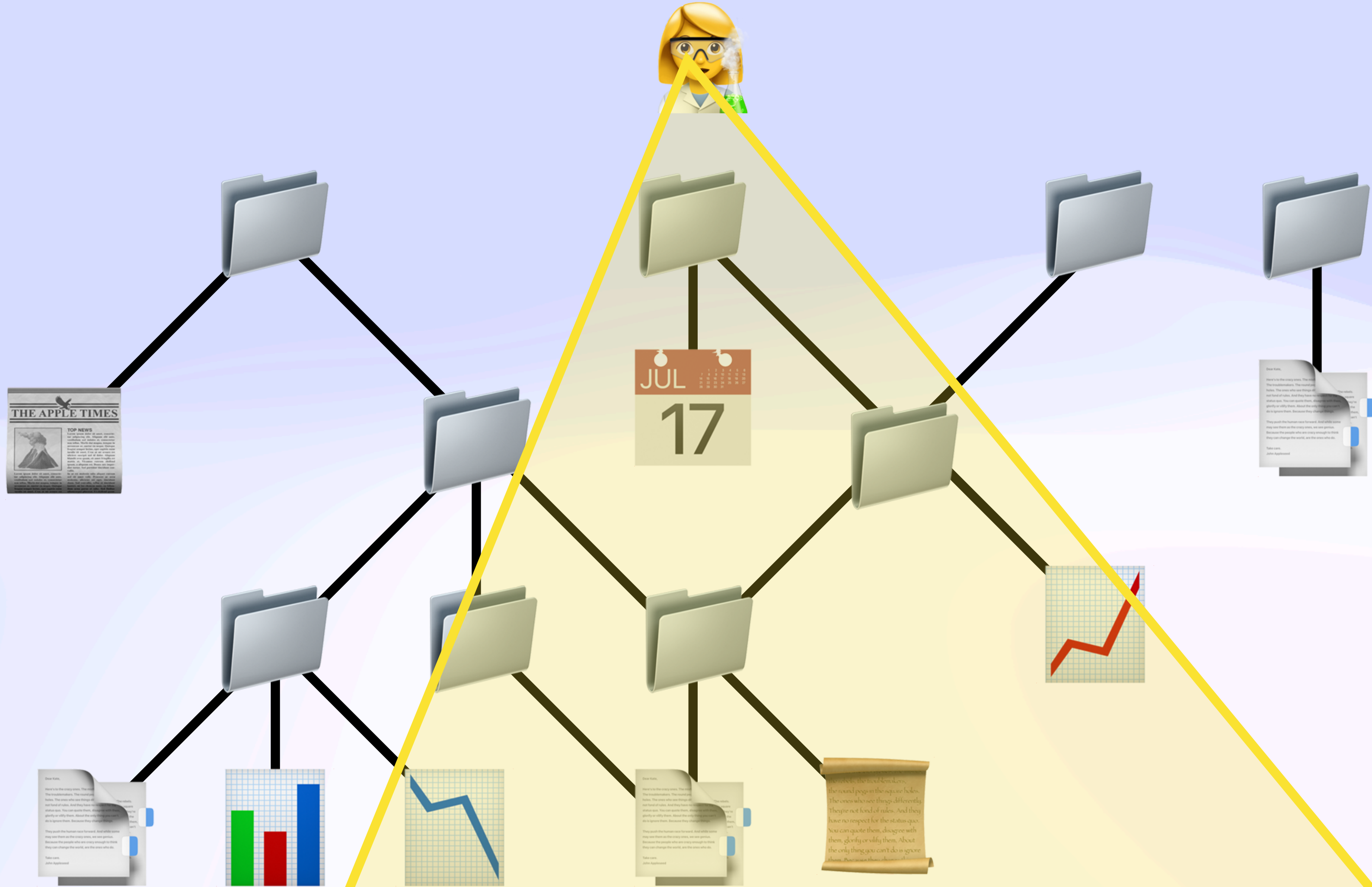
# Access Control

# Universal Read Control



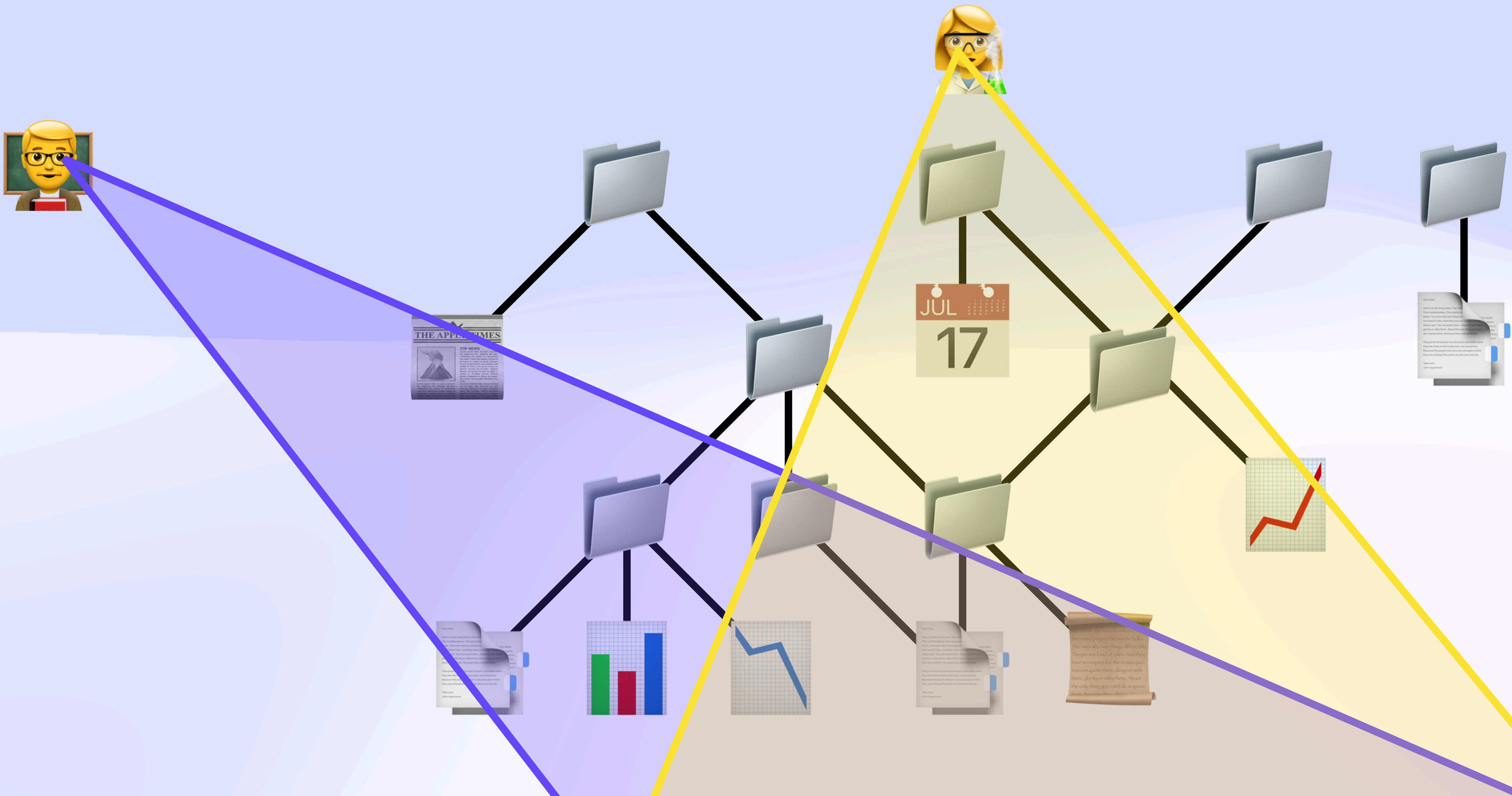
# Access Control

# Universal Read Control



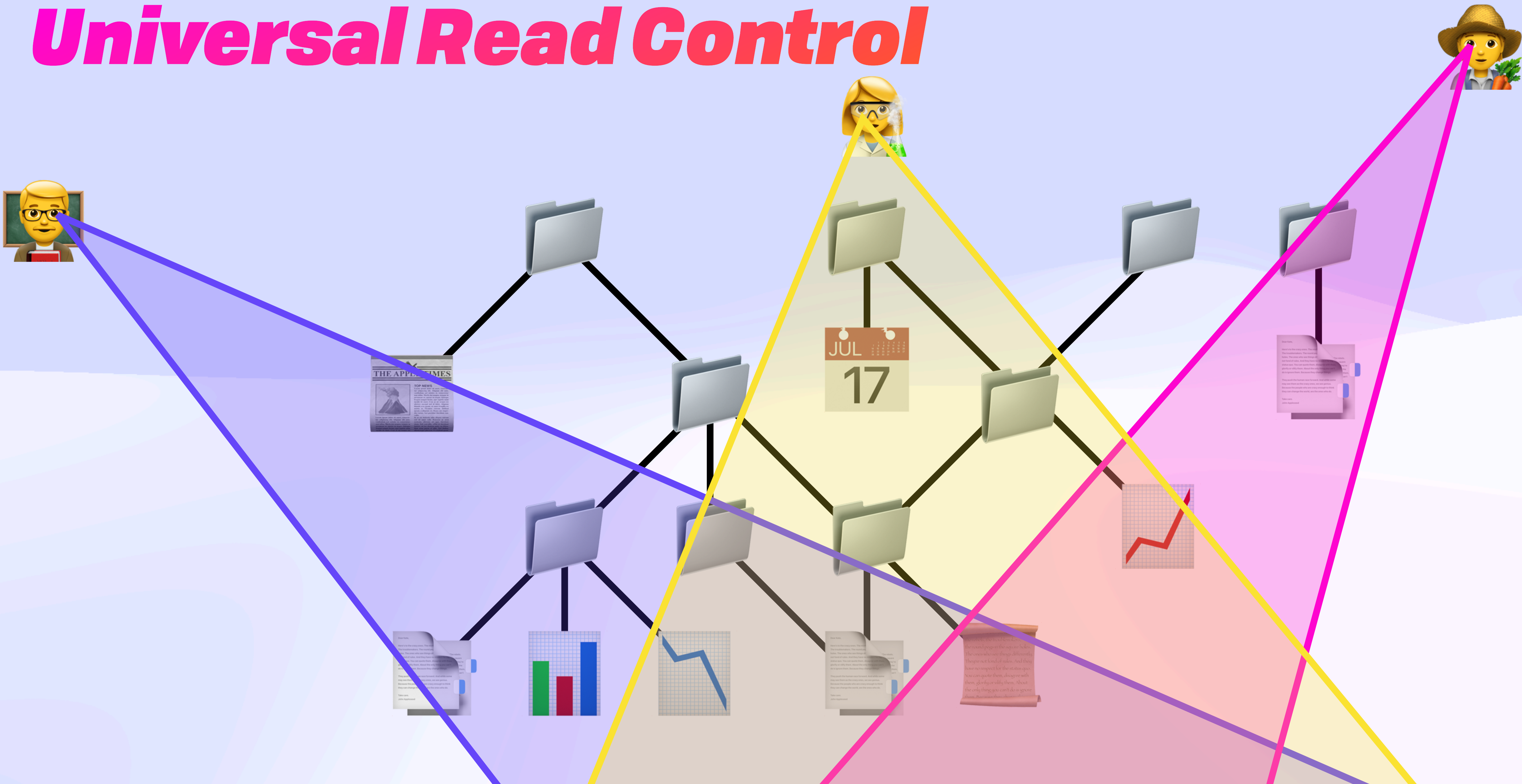
# Access Control

# Universal Read Control



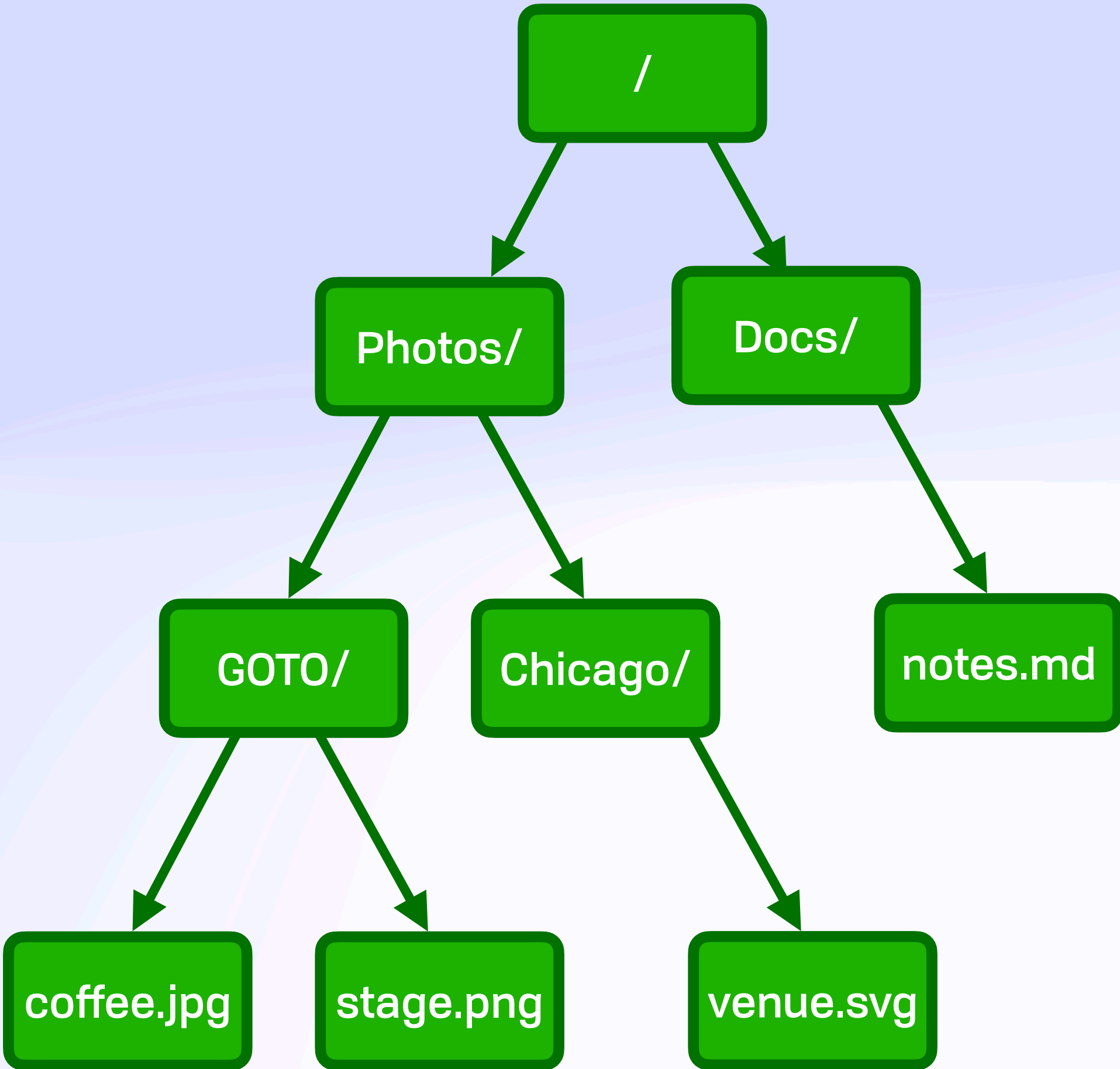
# Access Control

# Universal Read Control



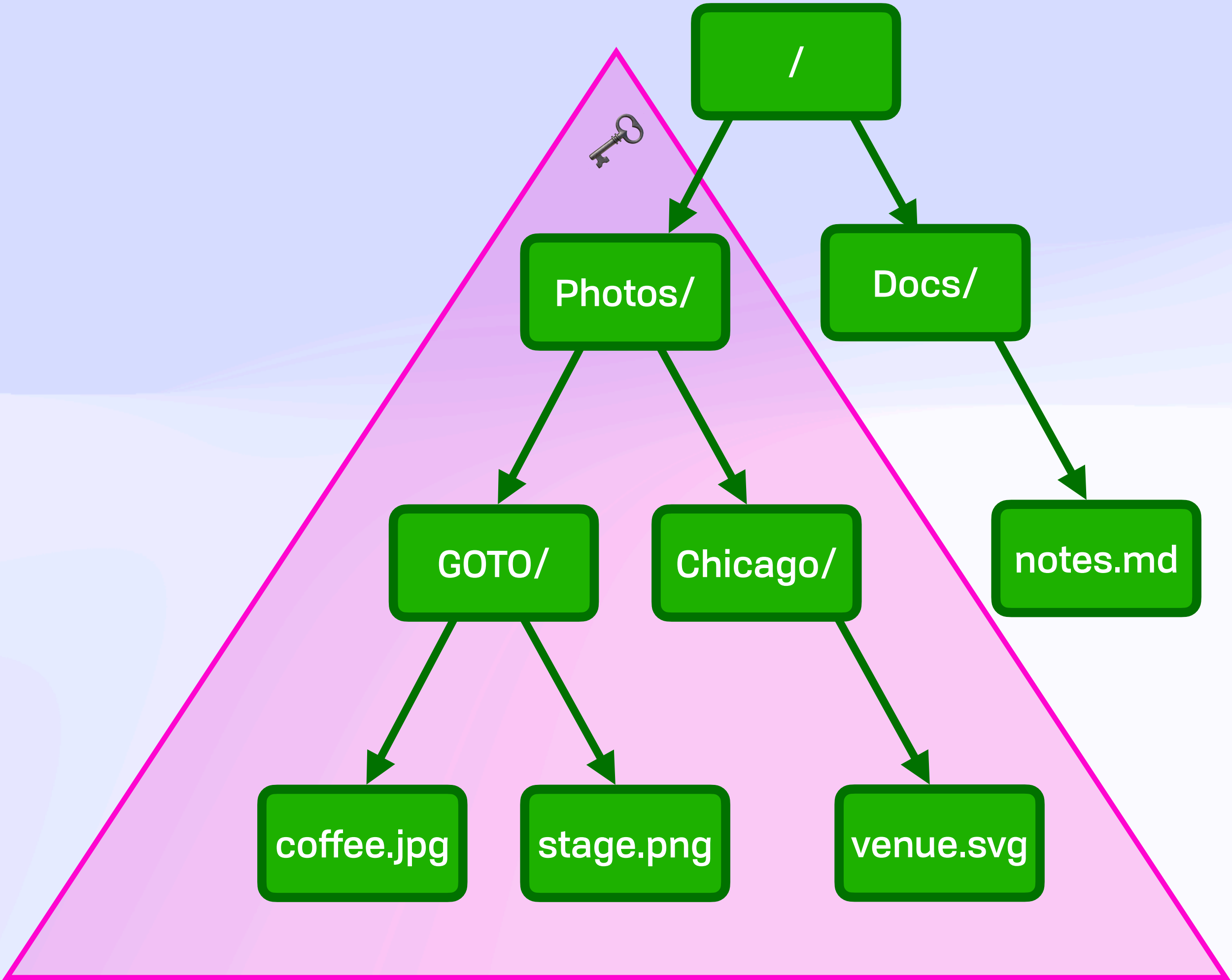
# Access Control

# *Offline Read Control*



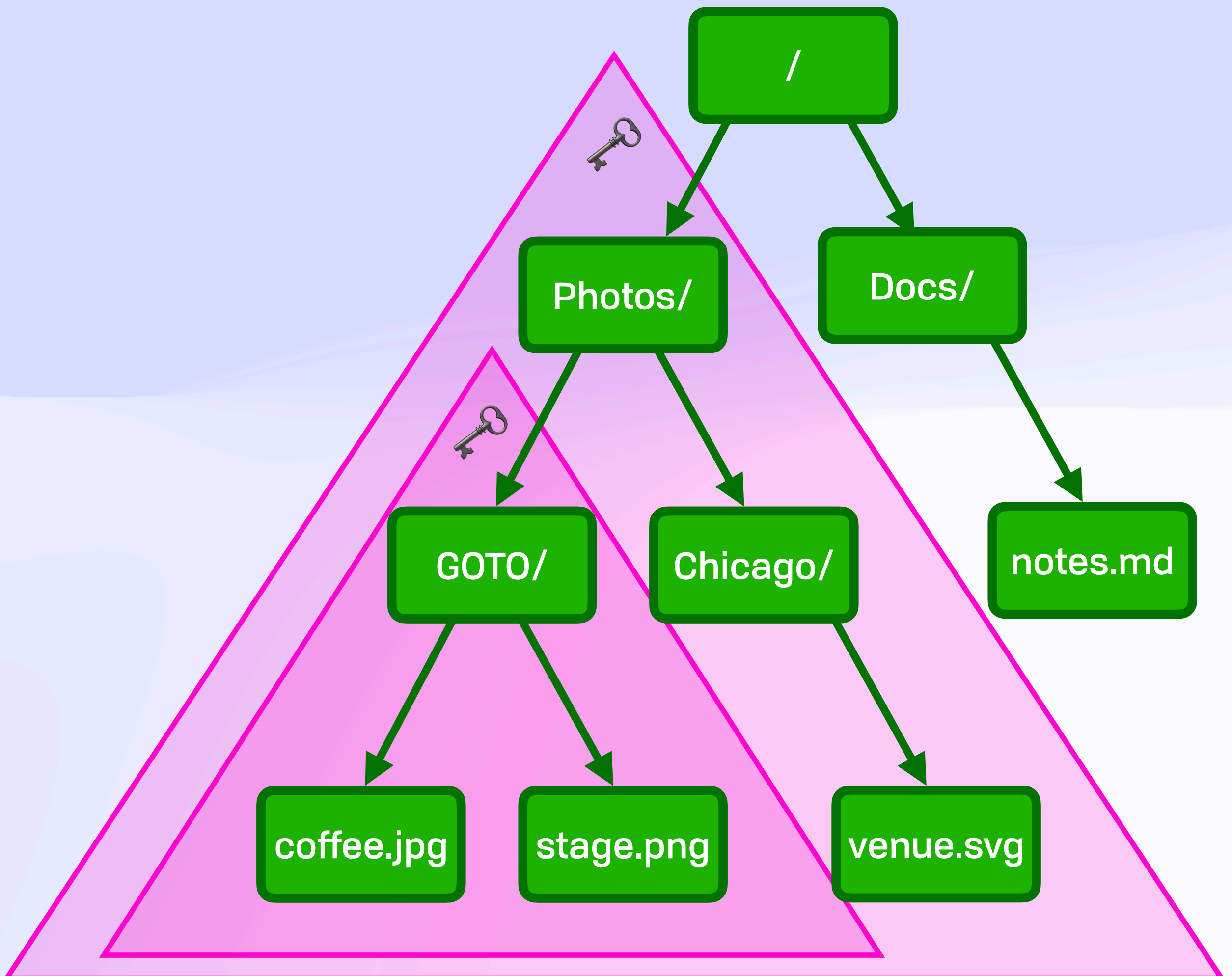
# Access Control

# *Offline Read Control*



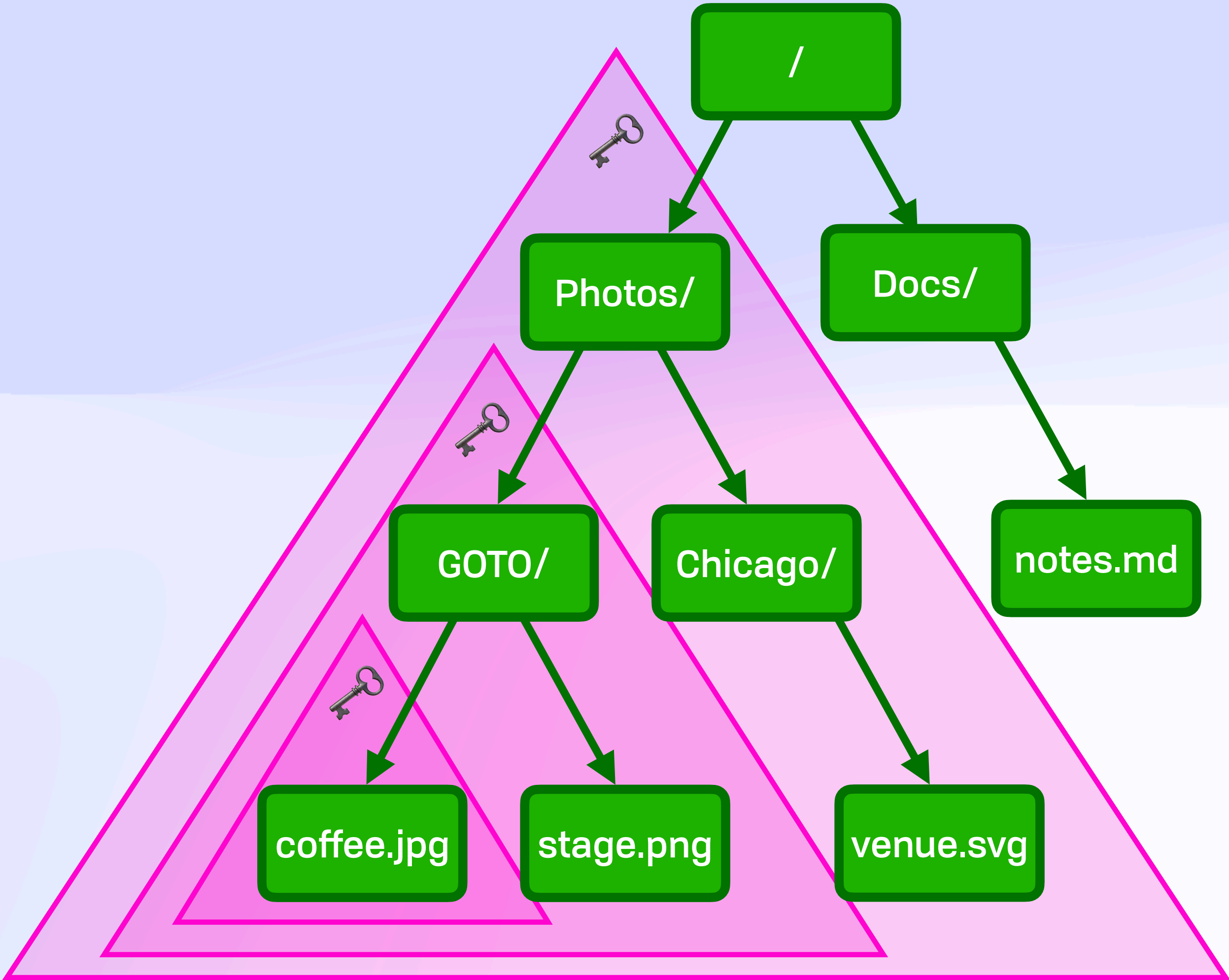
# Access Control

# *Offline Read Control*



# Access Control

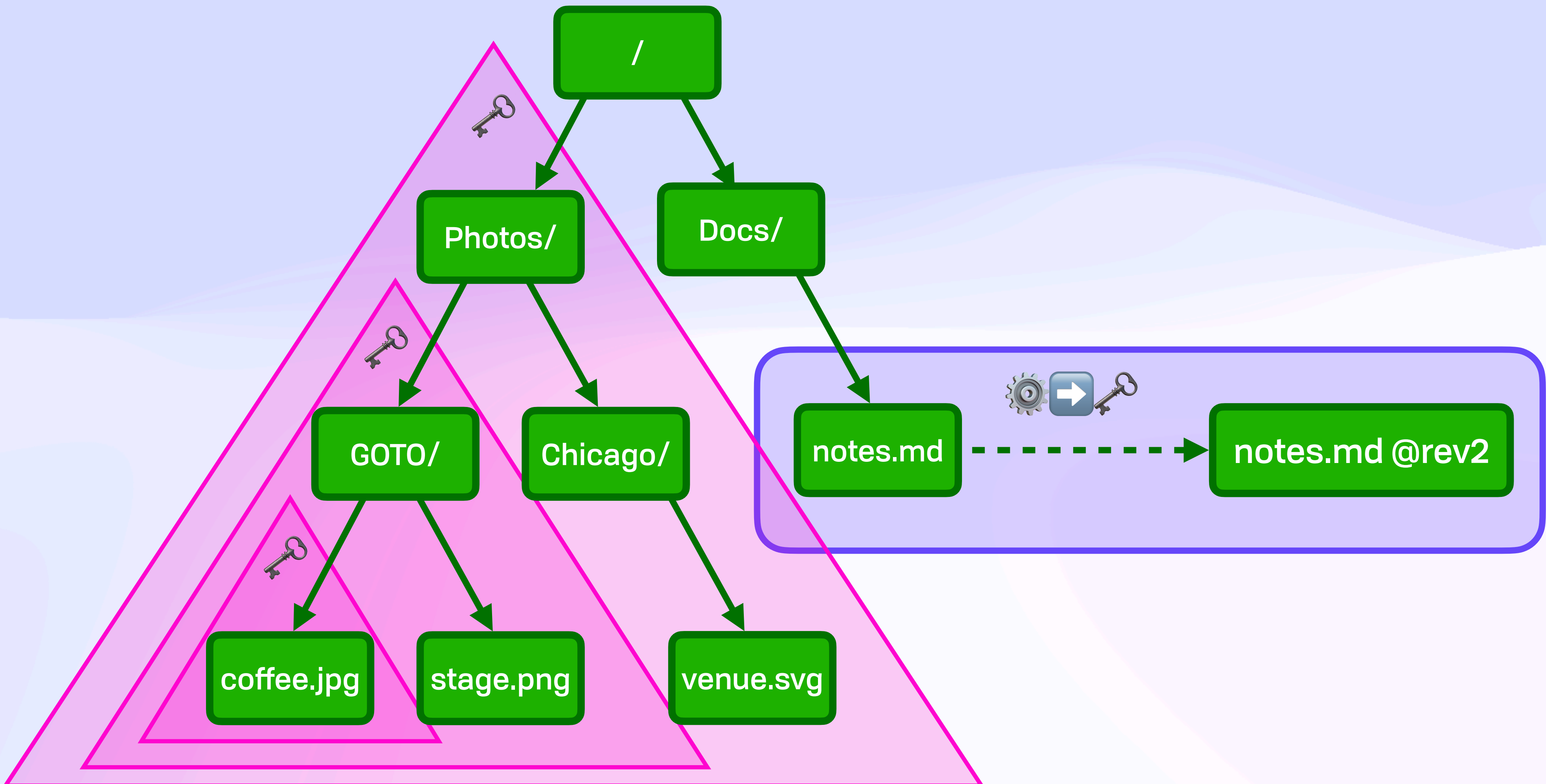
# *Offline Read Control*





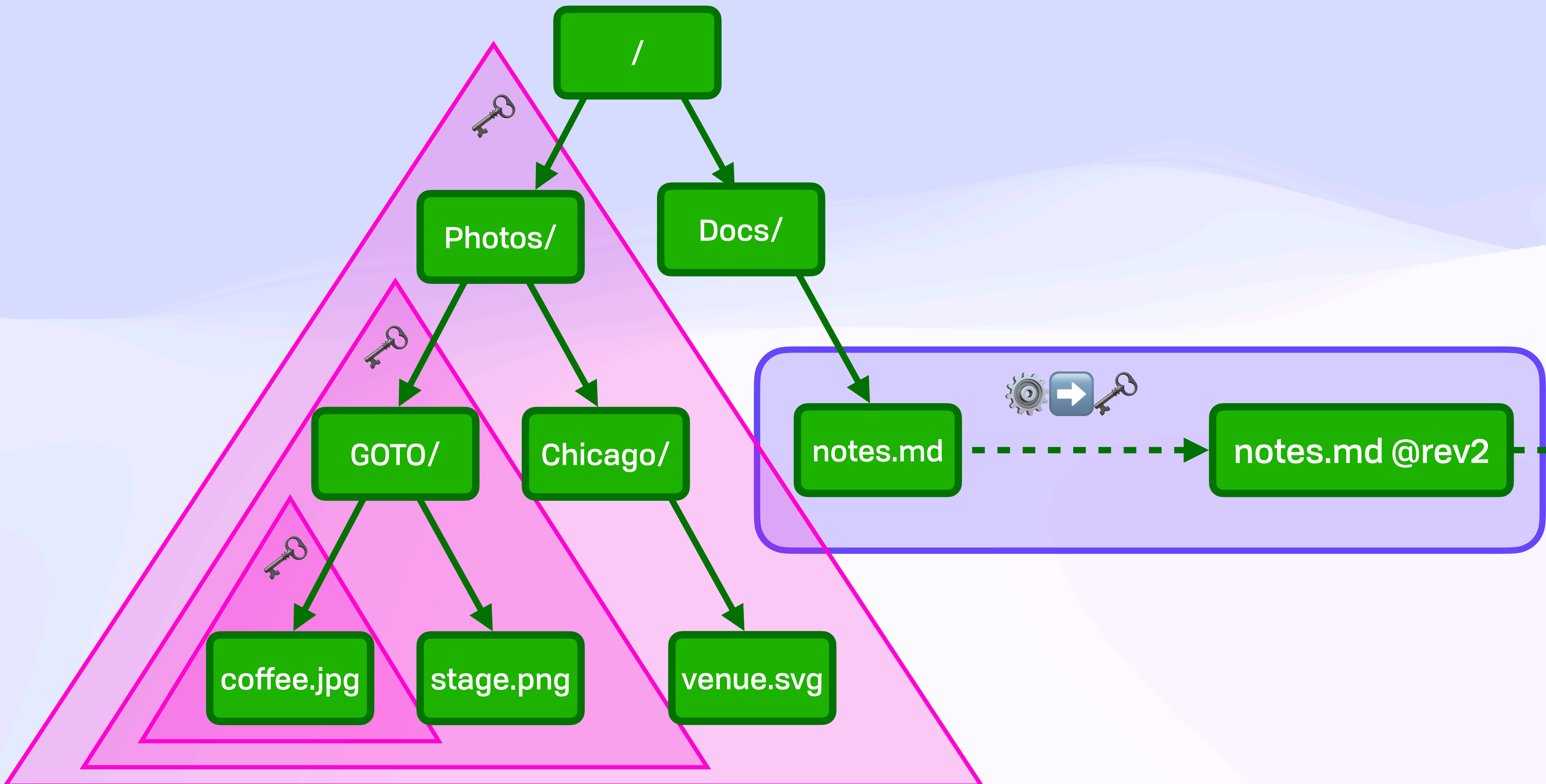
# Access Control

# Offline Read Control



# Access Control

# Offline Read Control

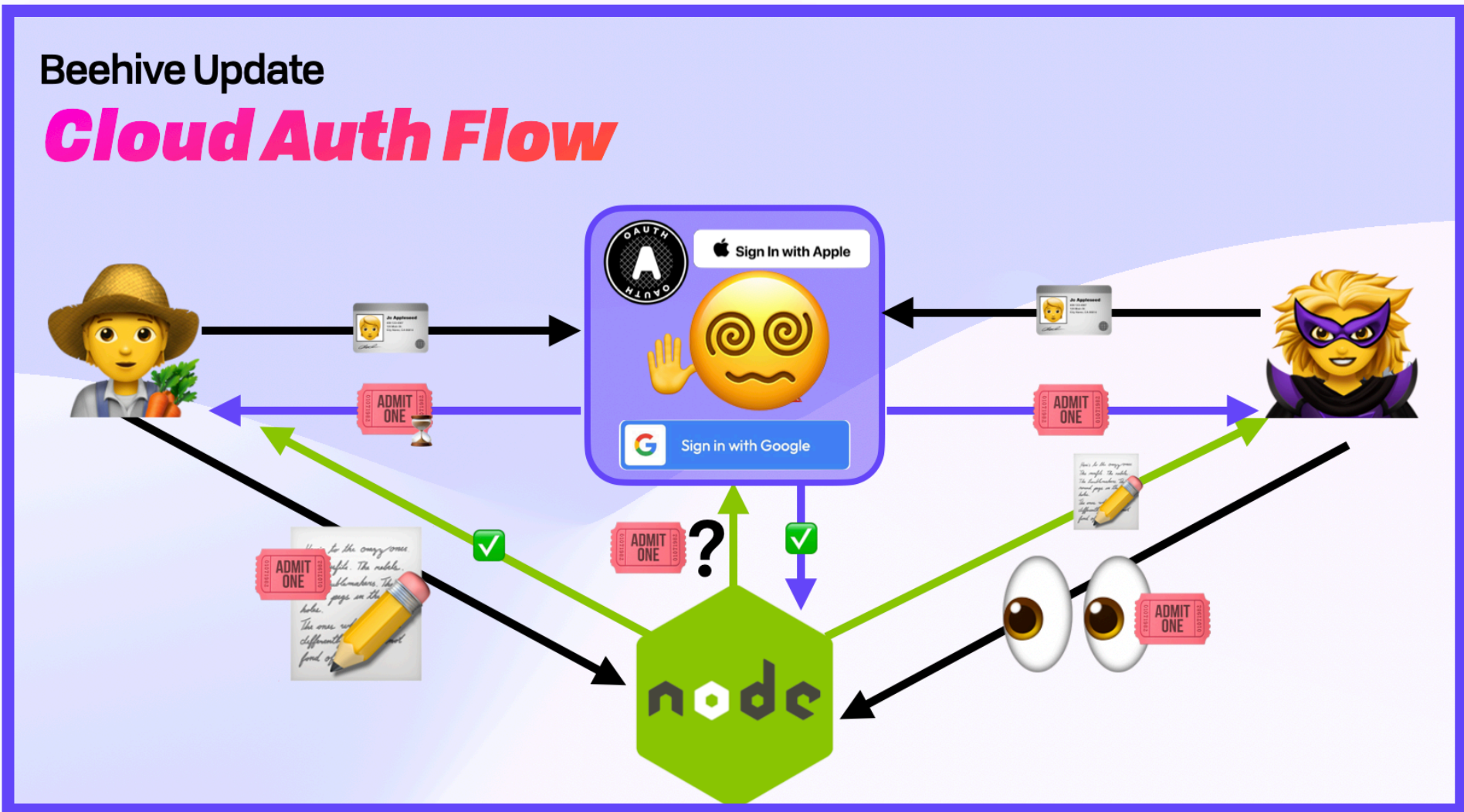


Access Control

# ***Self-Authenticating Changes***

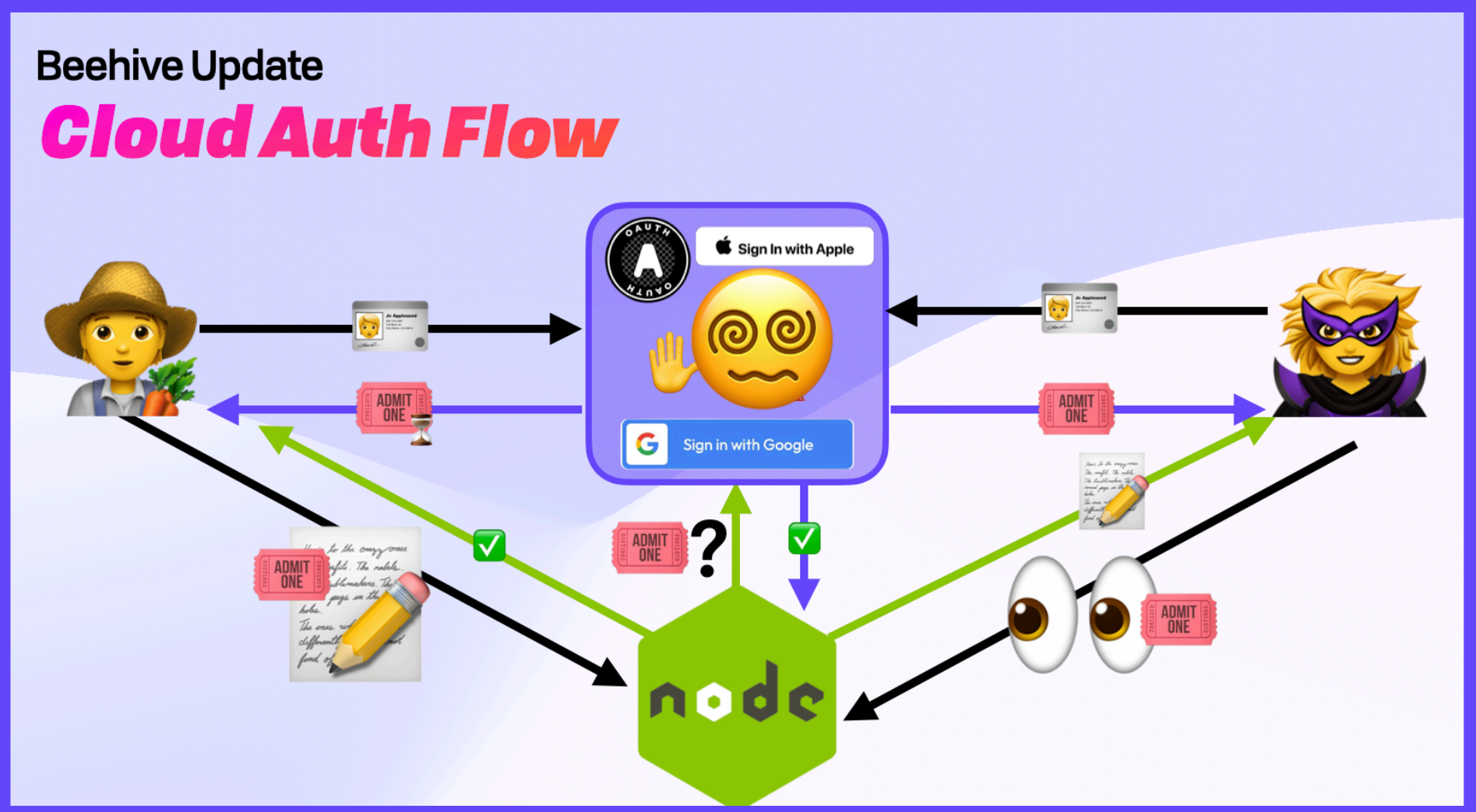
# Access Control

# Self-Authenticating Changes



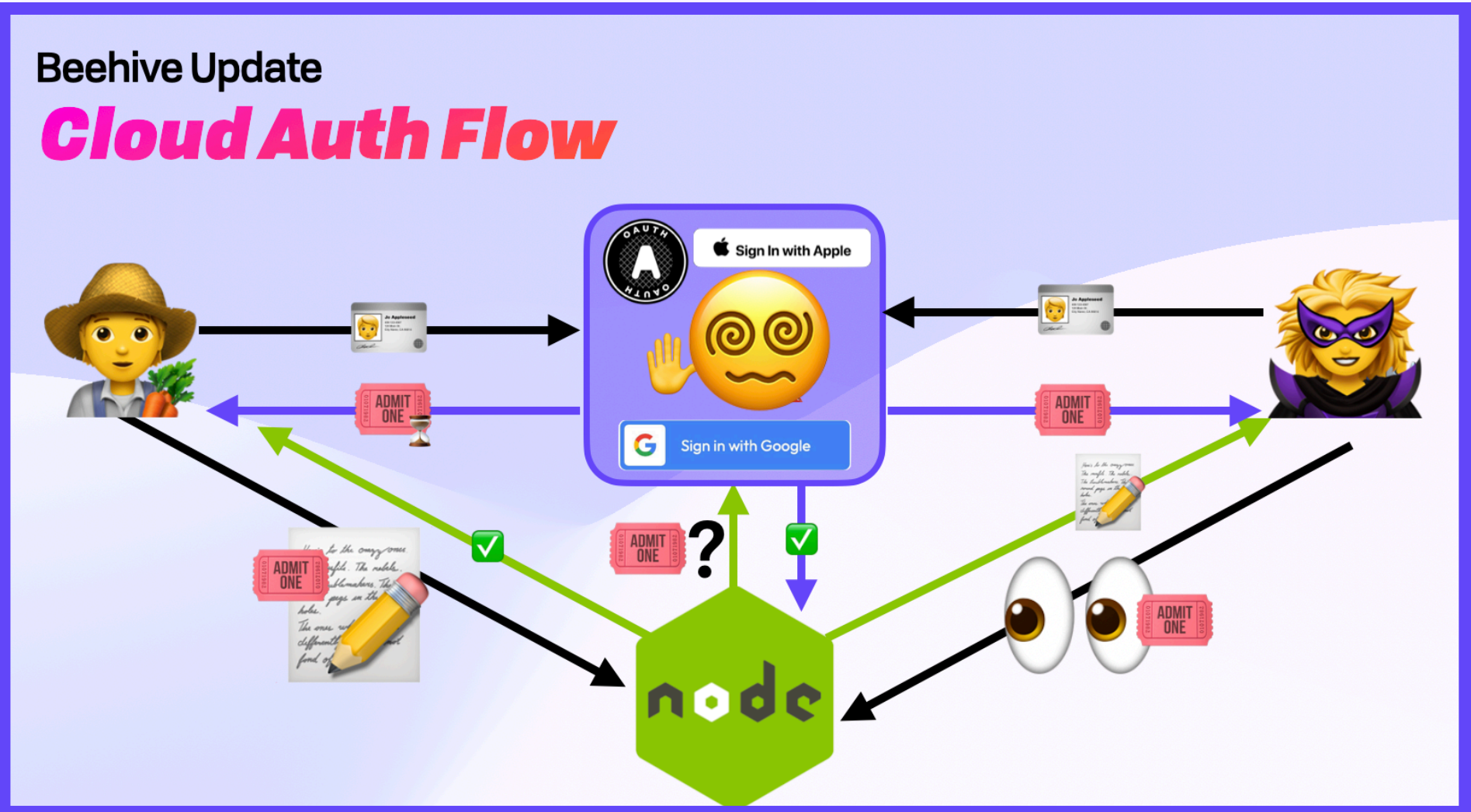
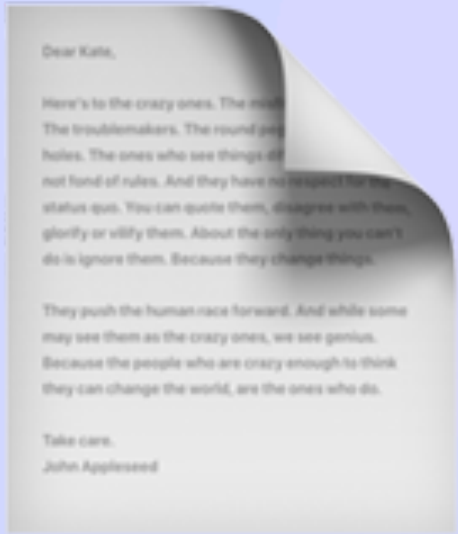
# Access Control

# Self-Authenticating Changes



# Access Control

# Self-Authenticating Changes

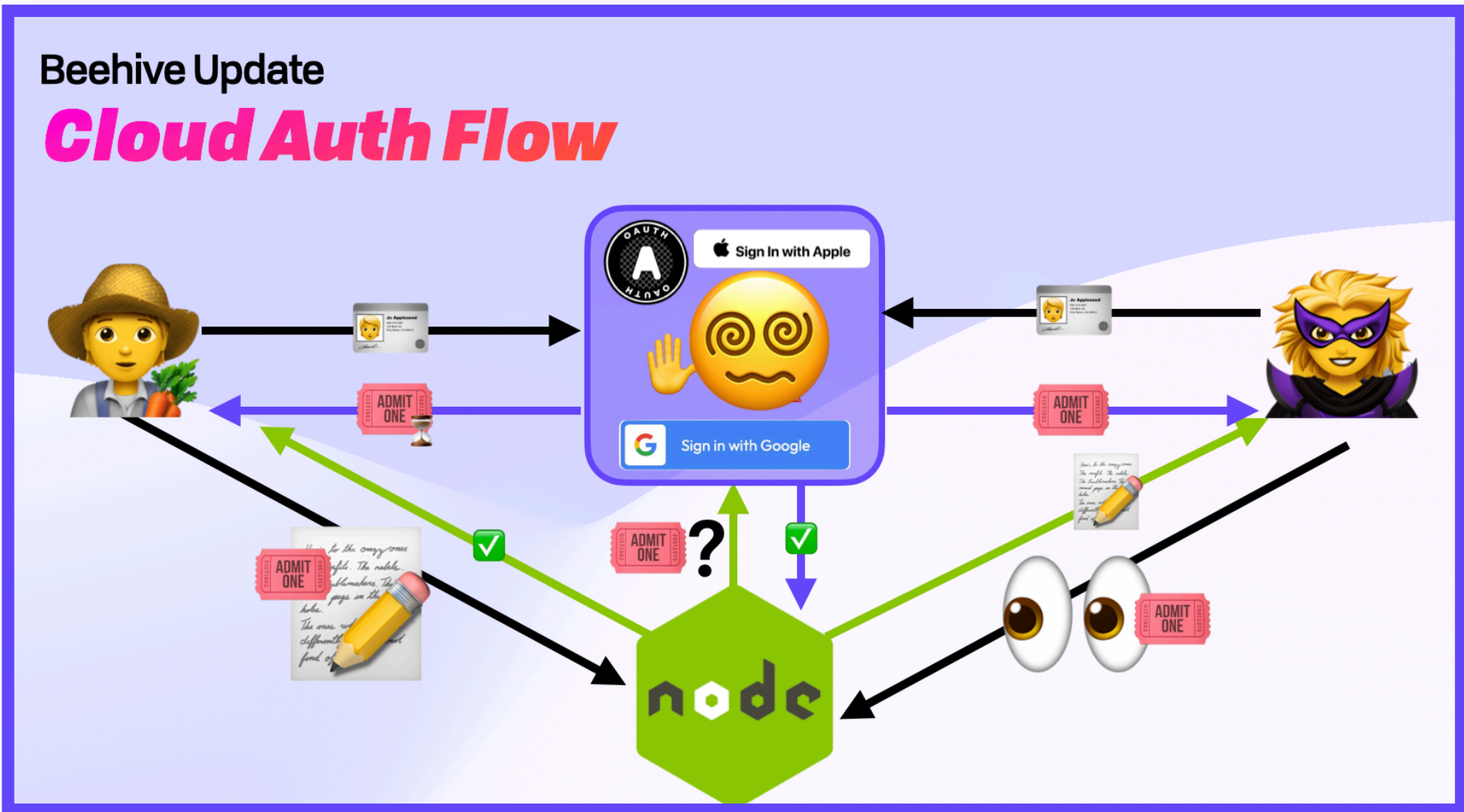


# Access Control

# Self-Authenticating Changes



Dear Kate,  
Here's to the crazy ones. The misfits. The troublemakers. The round pegs in a square hole. The ones who see things that not fond of rules. And they have no sense of 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

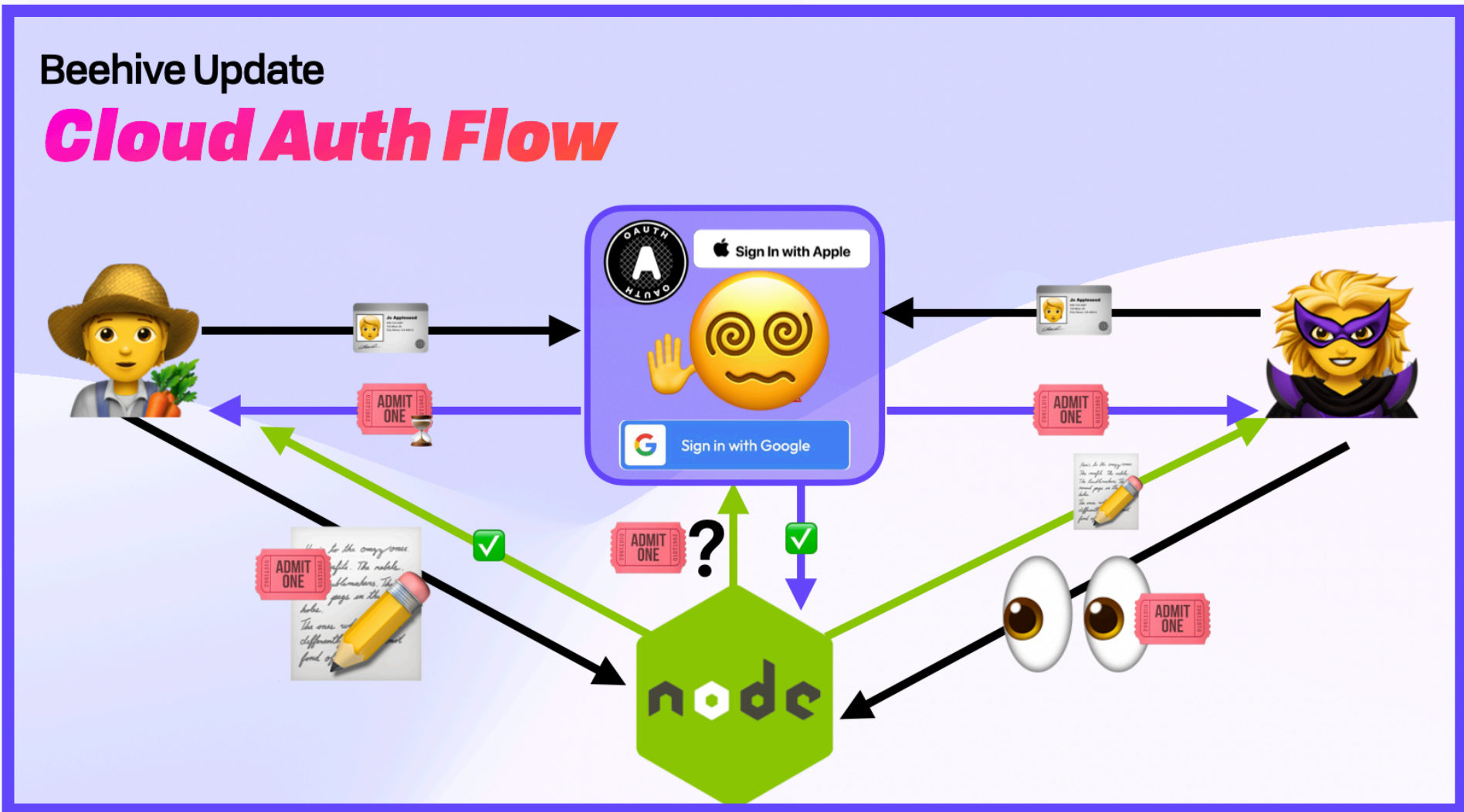


# Access Control

# Self-Authenticating Changes



Dear Kate,  
Here's to the crazy ones. The misfits. The troublemakers. The round pegs in a square hole. The ones who see things that not fond of rules. And they have no sense of 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



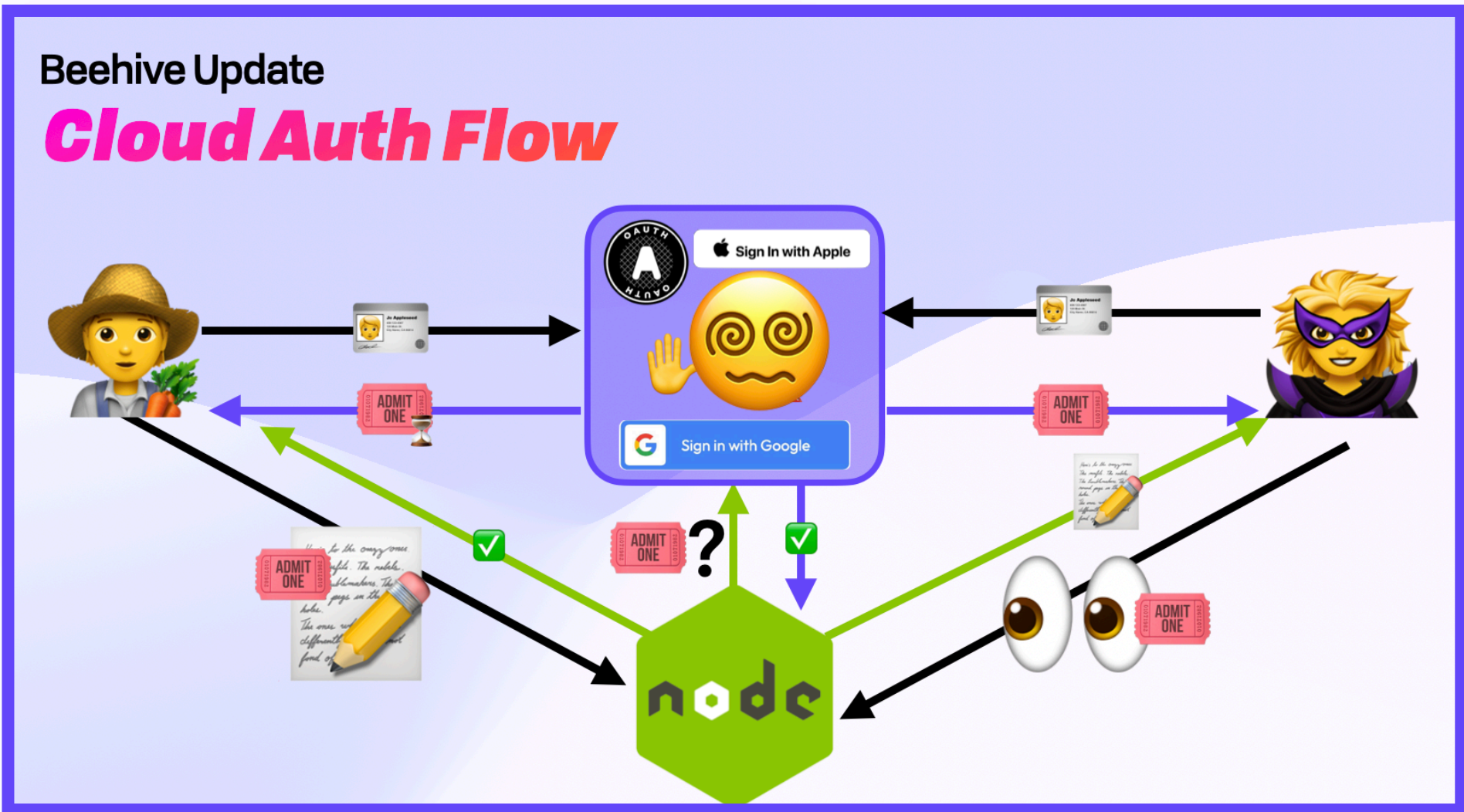
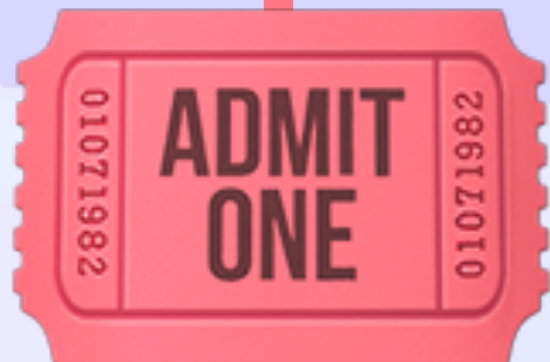


# Access Control

# Self-Authenticating Changes

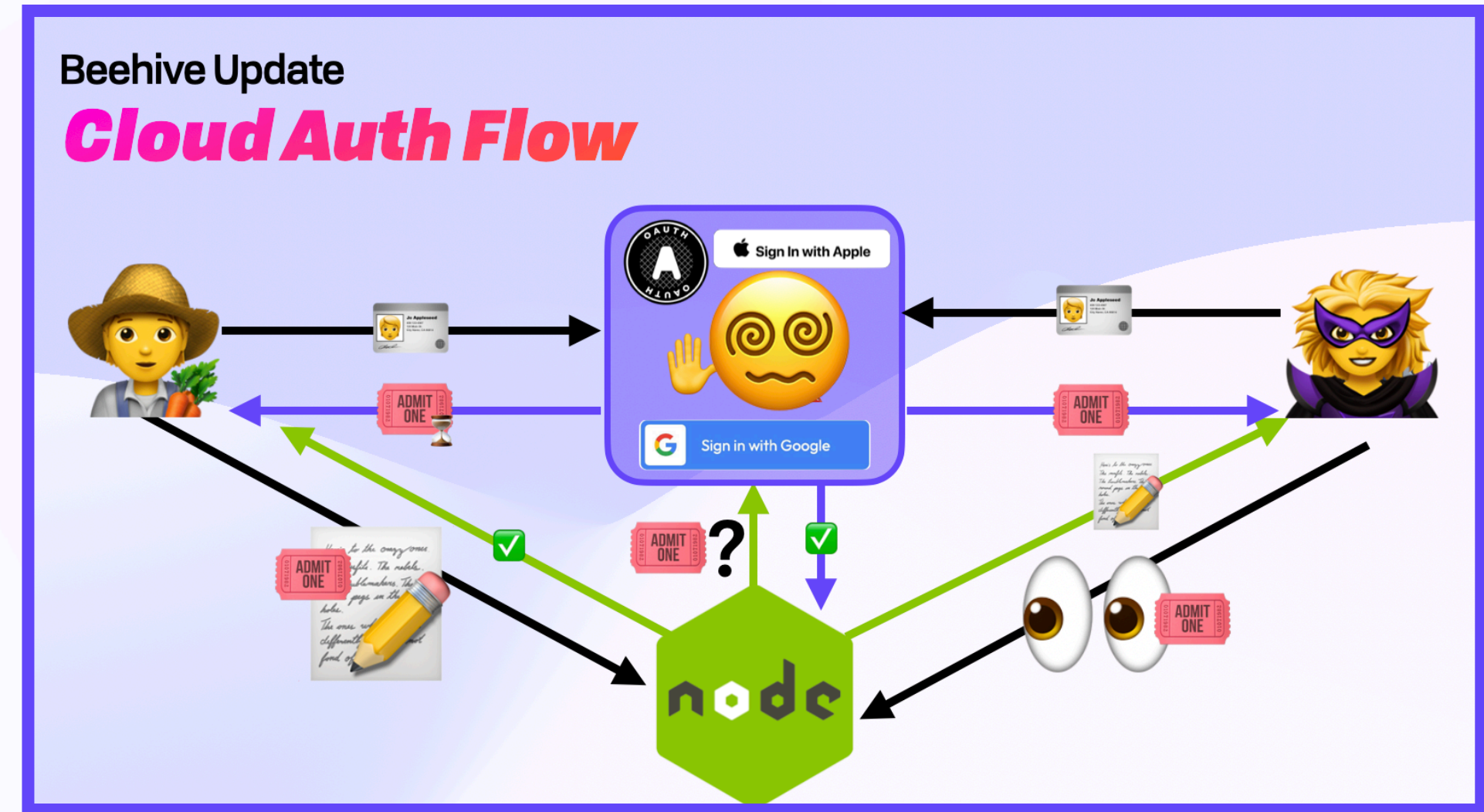
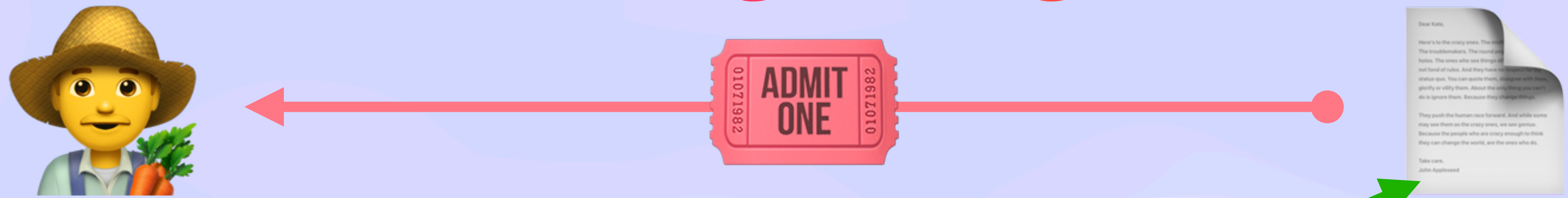


Dear Kate,  
Here's to the crazy ones. The misfits. The troublemakers. The round pegs in a square hole. The ones who see things that not fond of rules. And they have no sense of status quo. You can quote them, disagree with them, glorify or vilify them. About the only thing you can't do is ignore them. 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



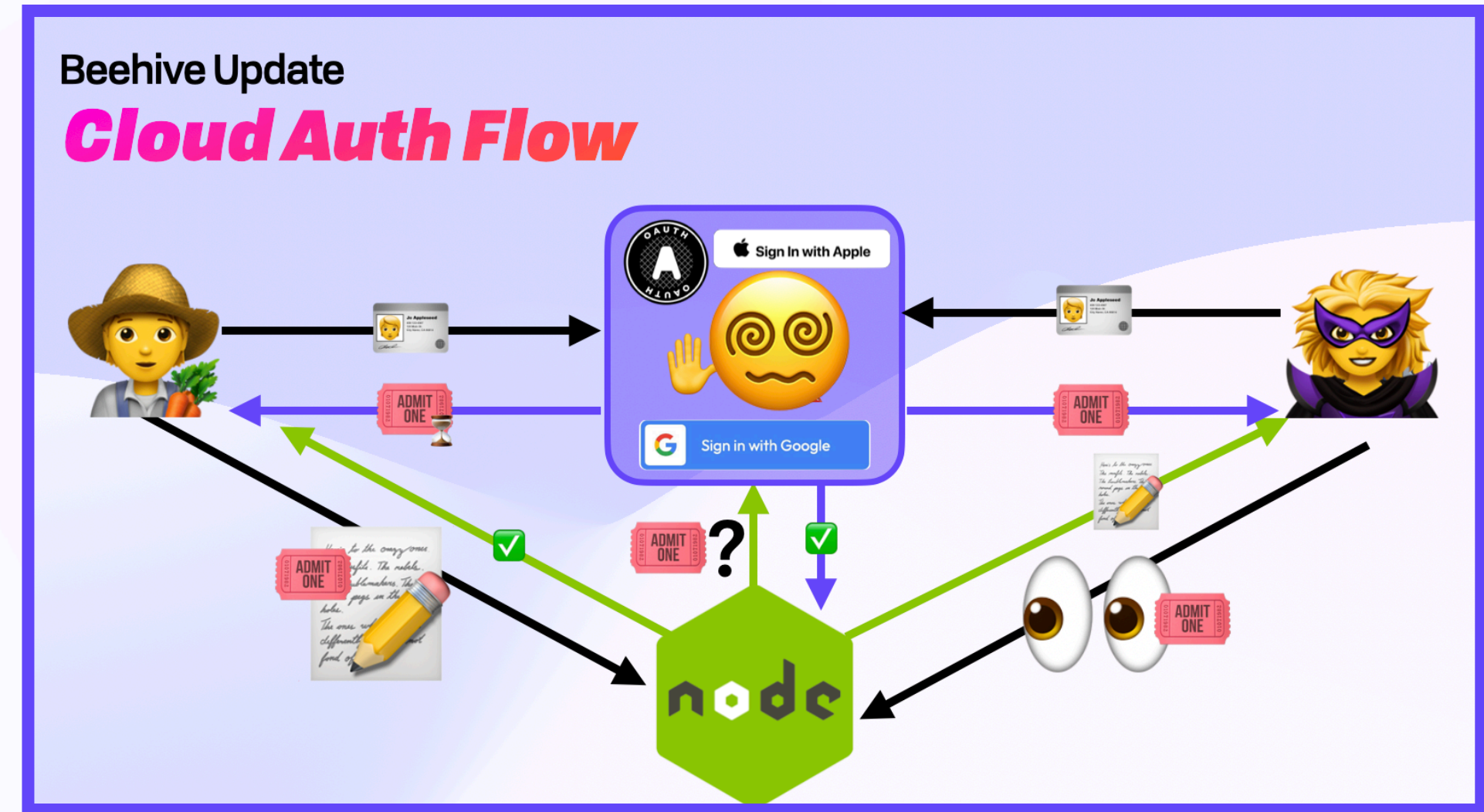
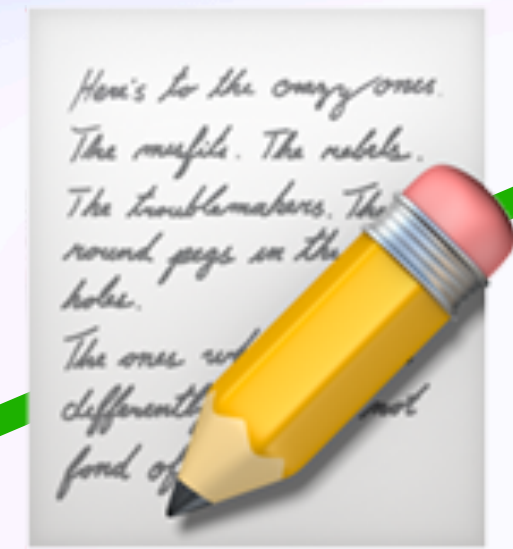
# Access Control

# Self-Authenticating Changes



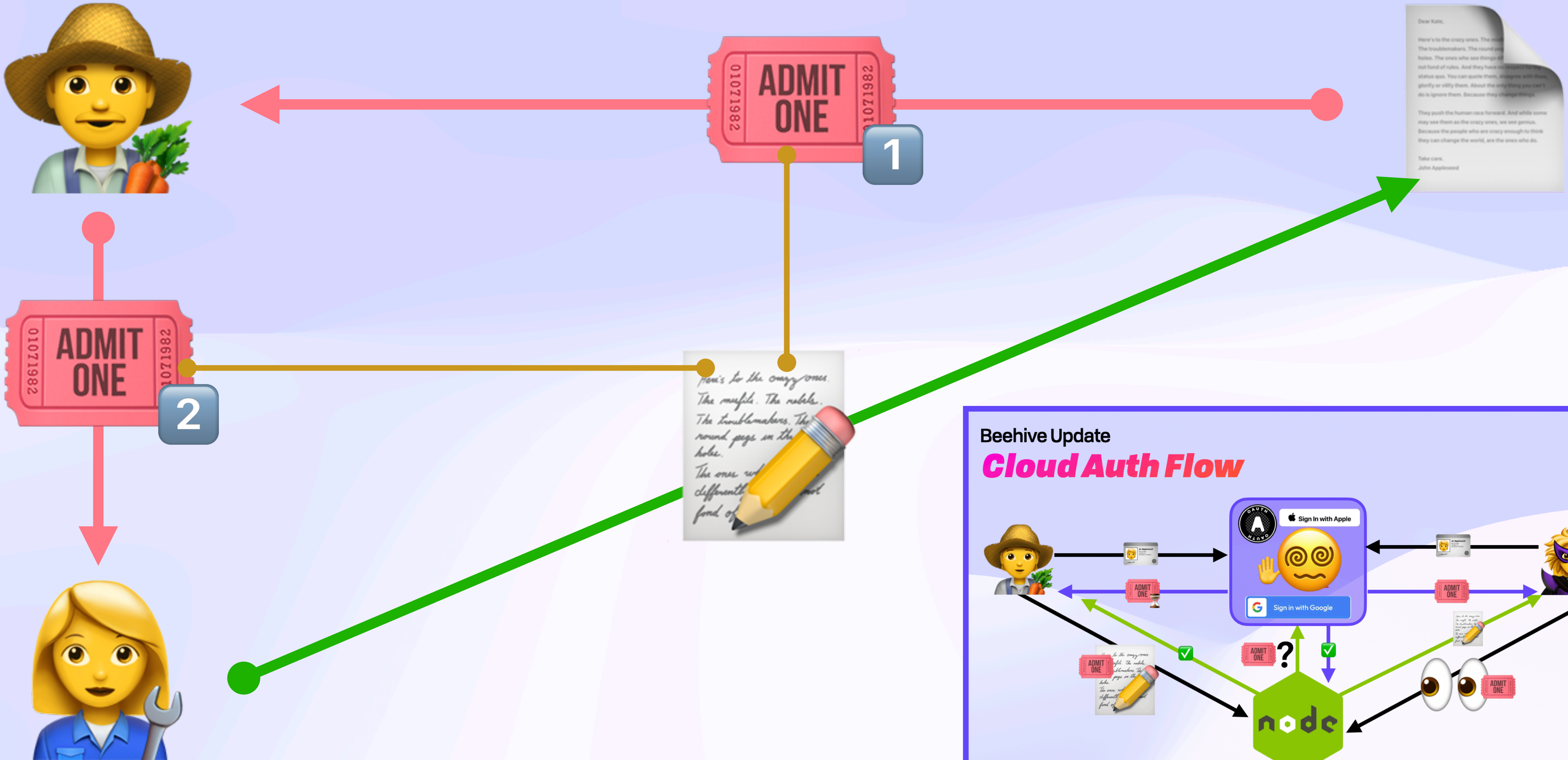
# Access Control

# Self-Authenticating Changes



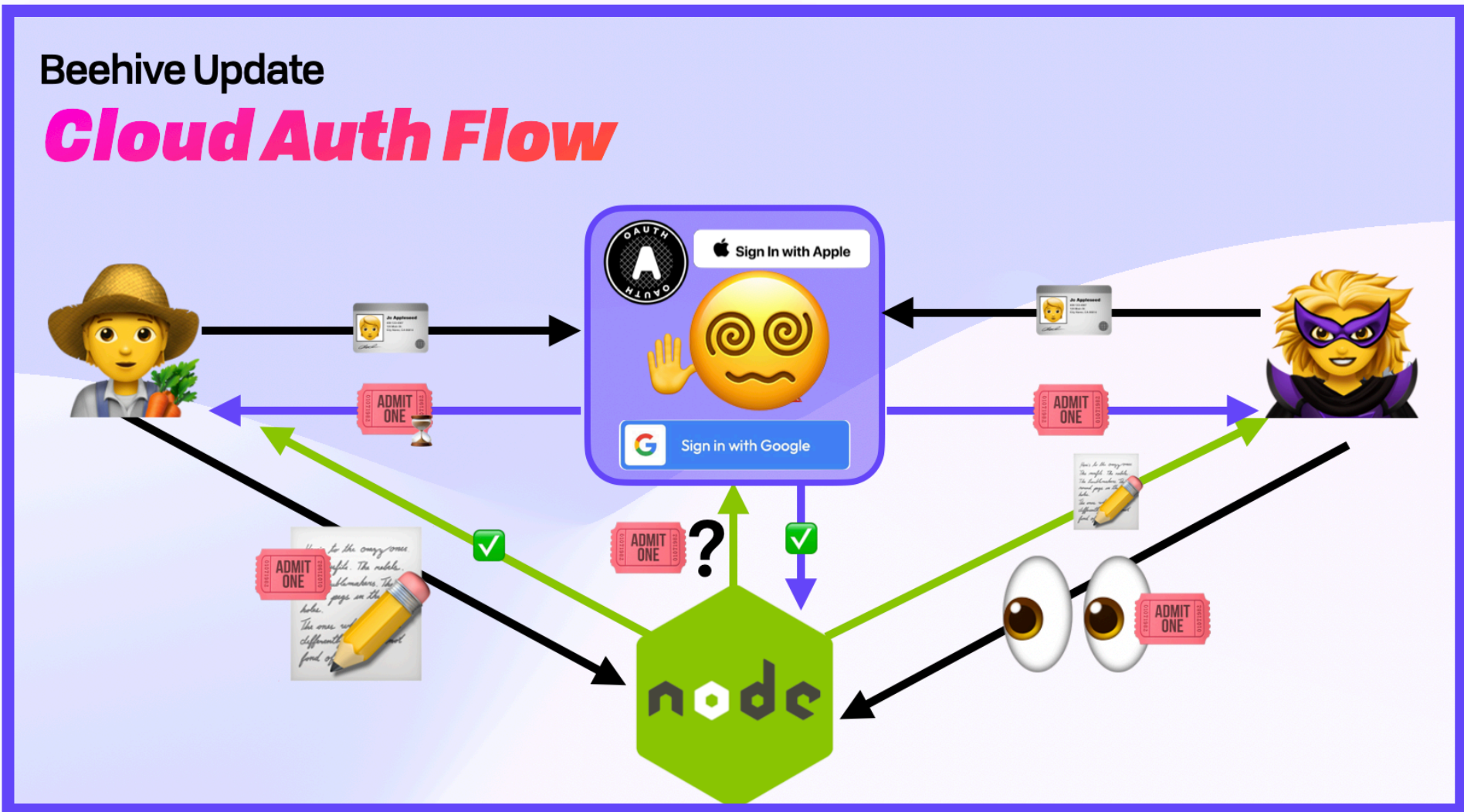
# Access Control

# Self-Authenticating Changes



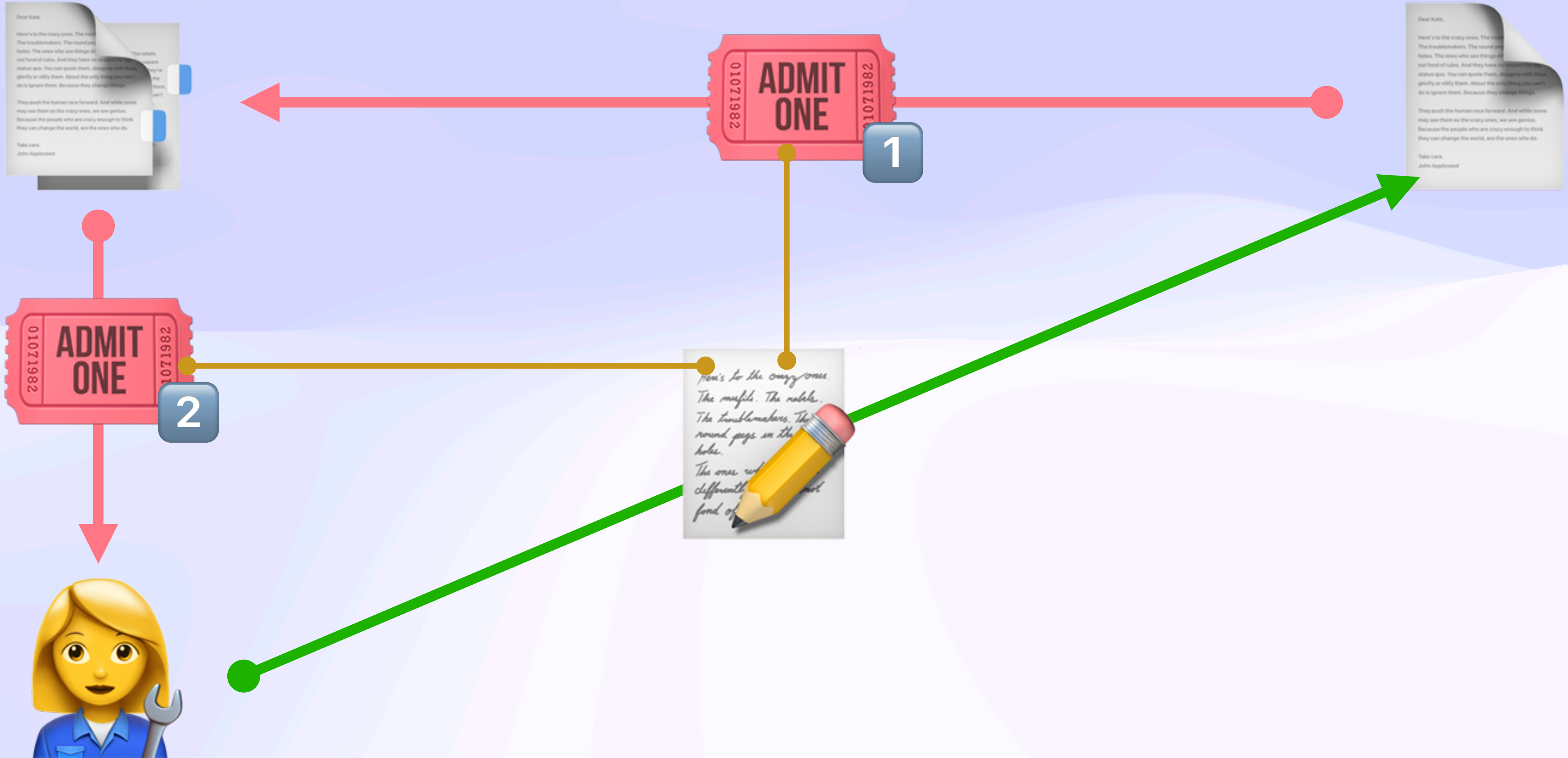
Dear Kate,  
Here's to the crazy ones. The misfits. The troublemakers. The round pegs in the square holes. The ones who see things that not fond of rules. And they have no sense of status quo. You can quote them, disagree with them, glorify or vilify them. About the only thing you can't do is ignore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

Here's to the crazy ones.  
The misfits. The rebels.  
The troublemakers. The round pegs in the square holes.  
The ones who see things differently. The ones who are fond of



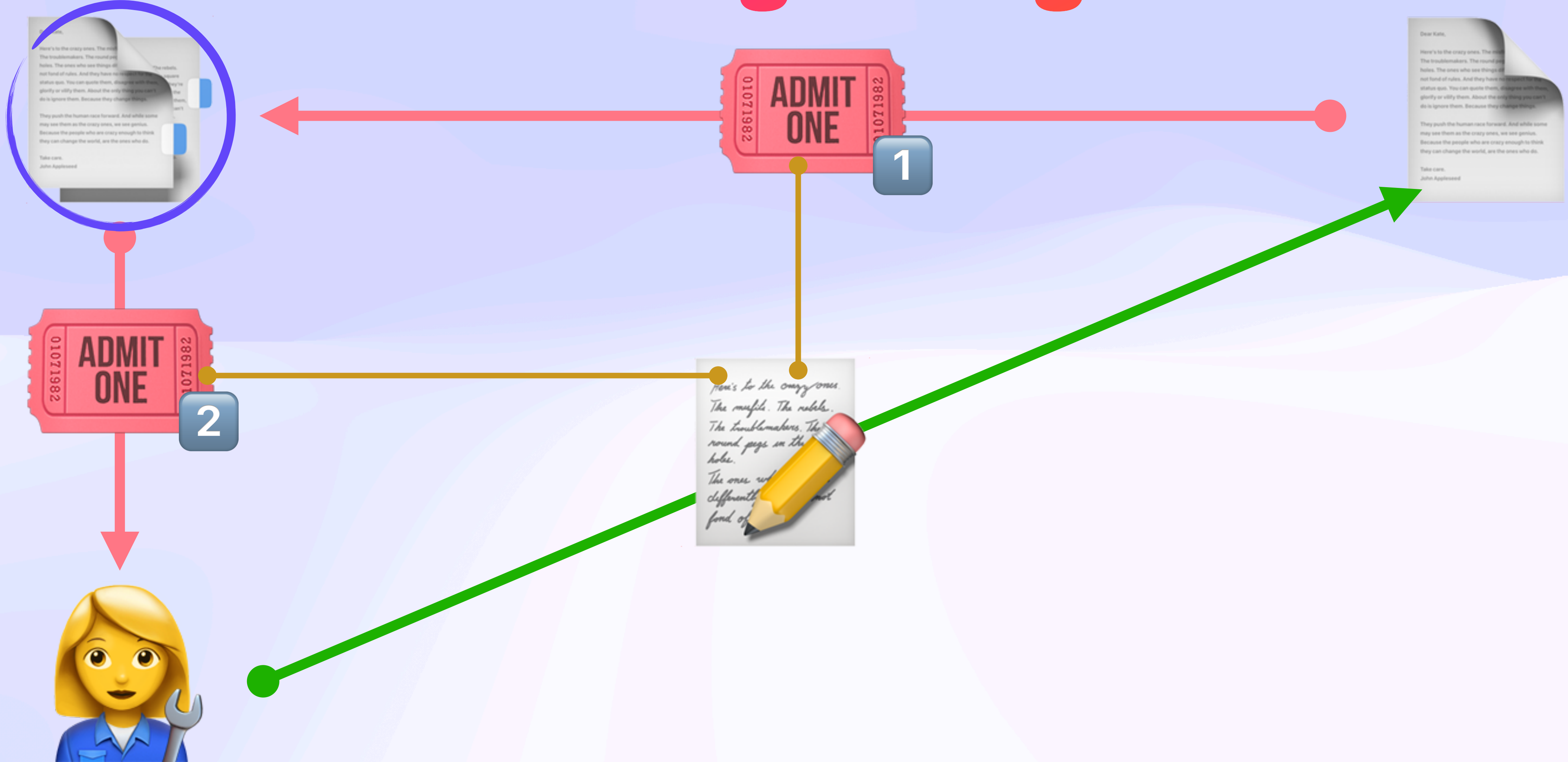
# Access Control

# Self-Authenticating Changes



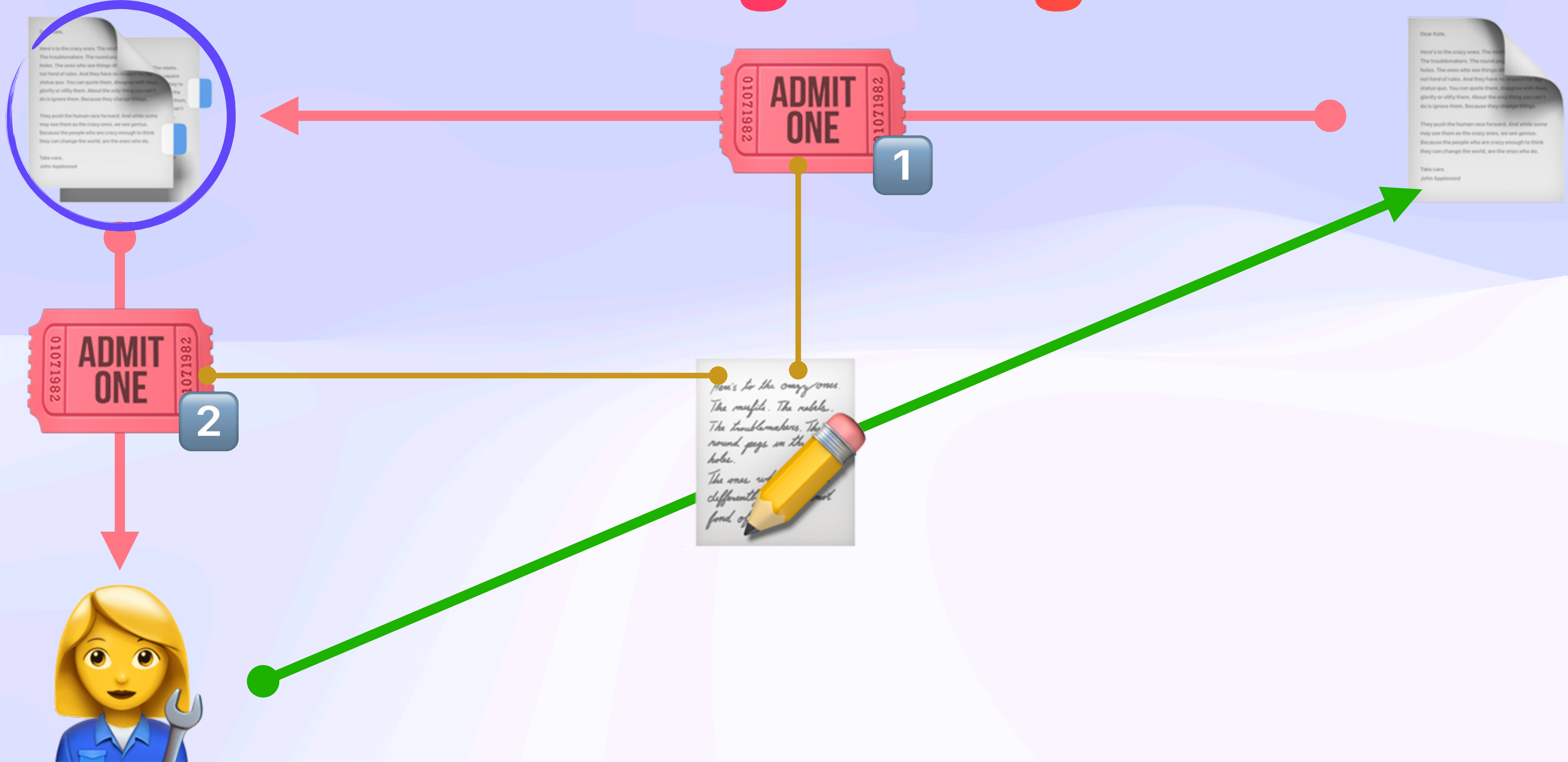
# Access Control

# Self-Authenticating Changes



# Access Control

# Self-Authenticating Changes



# Access Control

**JWT++**

```
{
  "aud": "did:key:zStEZpzSMtTt9k2vszgvCwF4fLQSyA15W5AQ4z3AR6Bx4eFJ5crJFbuGxKmbma4",
  "iss": "did:key:z5C4fuP2DDJChhMBCwAkpYUMuJZdNWWH5NeYjUyY8btYfzDh3aHwT5picHr9Ttjq",

  "nbf": 1611204719,
  "exp": 1611300000,

  "fct": [
    {
      "sha256": "B94D27B9934D3E08A52E52D7DA7DABFAC484EFE37A5380EE9088F7ACE2EFCDE9",
      "msg": "hello world"
    }
  ]

  "att": [
    {
      "wnfs": "boris.fission.name/public/photos/",
      "cap": "OVERWRITE"
    },
    {
      "email": "boris@fission.codes",
      "cap": "SEND"
    }
  ],

  "prf": [
    "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsInVhdCI6IjE6IjAuMS4wIn0.eyJhdWQiOiJkaWQ6a2V5Omp"
  ]
}
```



# Access Control

**JWT++**

```
{
  "aud": "did:key:zStEZpzSMtTt9k2vszgvCwF4fL00SvA15W5A04z3AR6Bx4eFJ5crJFbuGxKmbma4",
  "iss": "did:key:z5C4fuP2DDJChhMBCwAkpYUMuJZdNWWH5NeYjUyY8btYfzDh3aHwT5picHr9Ttjq",
  "nbf": 1611204719,
  "exp": 1611300000,
  "fct": [
    {
      "sha256": "B94D27B9934D3E08A52E52D7DA7DABFAC484EFE37A5380EE9088F7ACE2EFCDE9",
      "msg": "hello world"
    }
  ]
  "att": [
    {
      "wnfs": "boris.fission.name/public/photos/",
      "cap": "OVERWRITE"
    },
    {
      "email": "boris@fission.codes",
      "cap": "SEND"
    }
  ],
  "prf": [
    "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsInVhdiiI6IjAuMS4wIn0.eyJhdWQiOiJkaWQ6a2V50np"
  ]
}
```

# Access Control

**JWT++**

```
{
  "aud": "did:key:zStEZpzSMtTt9k2vszgvCwF4fL00SvA15W5A04z3AR6Bx4eFJ5crJFbuGxKmbma4",
  "iss": "did:key:z5C4fuP2DDJChhMBCwAkpYUMuJZdNWWH5NeYjUyY8btYfzDh3aHwT5picHr9Ttjq",

  "nbf": 1611204719,
  "exp": 1611300000,

  "fct": [
    {
      "sha256": "B94D27B9934D3E08A52E52D7DA7DABFAC484EFE37A5380EE9088F7ACE2EFCDE9",
      "msg": "hello world"
    }
  ]

  "att": [
    {
      "wnfs": "boris.fission.name/public/photos/"
      "cap": "OVERWRITE"
    },
    {
      "email": "boris@fission.codes",
      "cap": "SEND"
    }
  ],

  "prf": [
    "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsInVhdDI6IjAuMS4wIn0.eyJhdWQiOiJkaWQ6a2V50np"
  ]
}
```

# Access Control

**JWT++**

```
{
  "aud": "did:key:zStEZpzSMtTt9k2vszgvCwF4fL00SvA15W5A04z3AR6Bx4eFJ5crJFbuGxKmbma4",
  "iss": "did:key:z5C4fuP2DDJChhMBCwAkpYUMuJZdNWWH5NeYjUyY8btYfzDh3aHwT5picHr9Ttjq",

  "nbf": 1611204719,
  "exp": 1611300000,

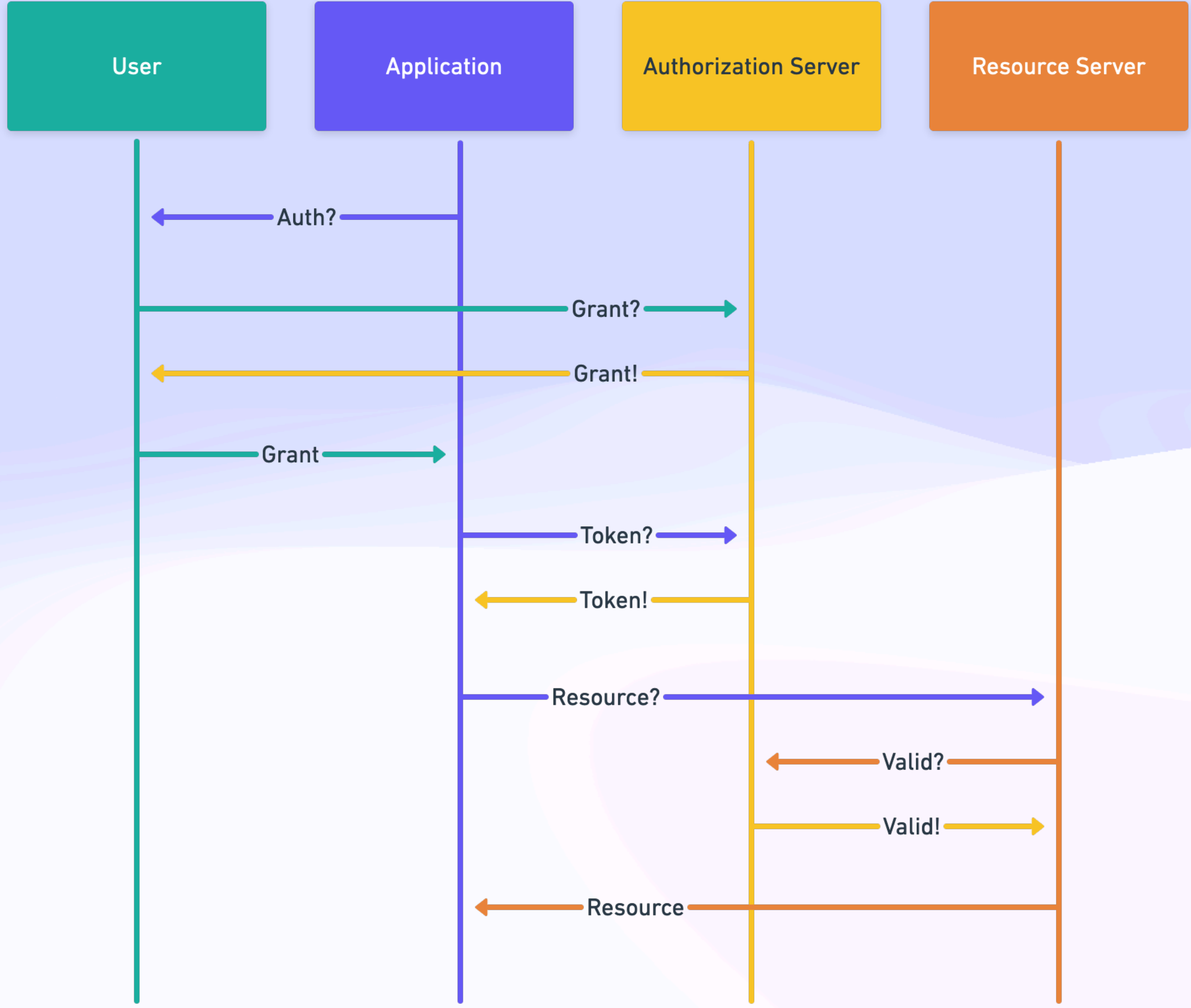
  "fct": [
    {
      "sha256": "B94D27B9934D3E08A52E52D7DA7DABFAC484EFE37A5380EE9088F7ACE2EFCDE9",
      "msg": "hello world"
    }
  ]

  "att": [
    {
      "wnfs": "boris.fission.name/public/photos/"
      "cap": "OVERWRITE"
    },
    {
      "email": "boris@fission.codes",
      "cap": "SEND"
    }
  ],

  "prf": [
    "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsInVhdDI6IjAuMS4wIn0.eyJhdWQiOiJkaWQ6a2V5OjE6IiwiaXNja3QiOiJkaWQ6a2V5OjE6IiwiaWF0IjoiMTYxMTIwNDcxOSJ9"
  ]
}
```

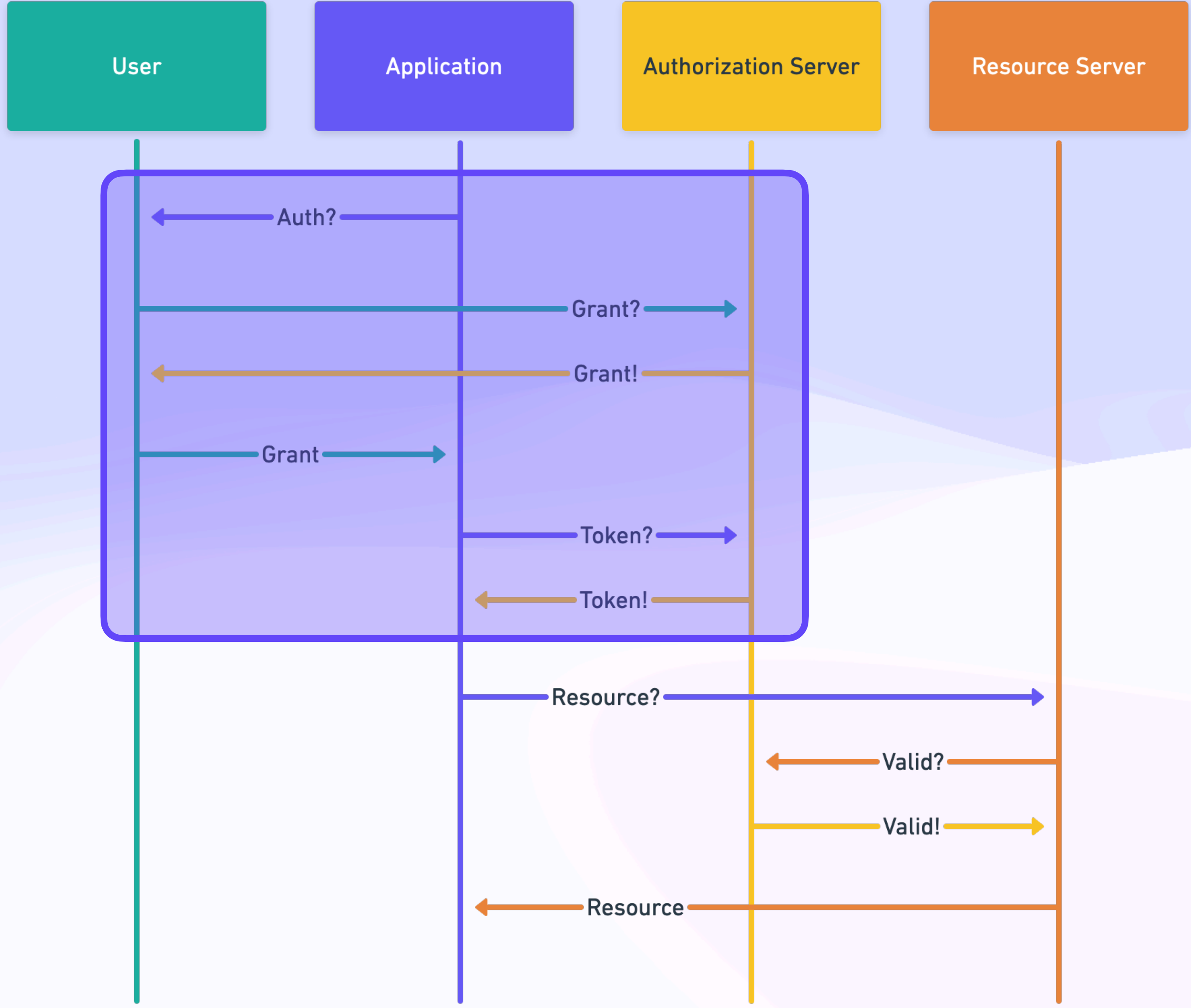
# Access Control

# *OAuth Sequence*



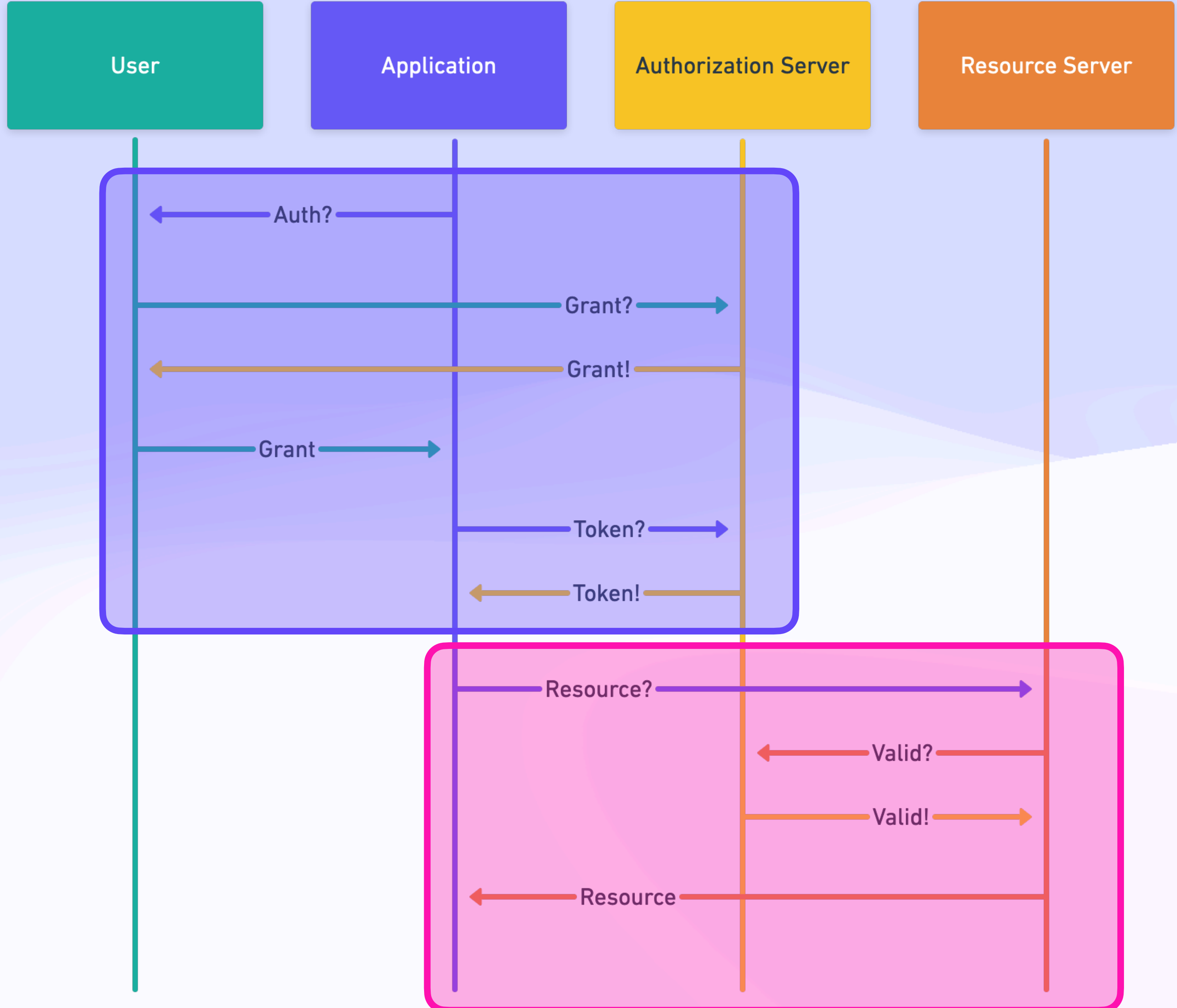
# Access Control

# *OAuth Sequence*



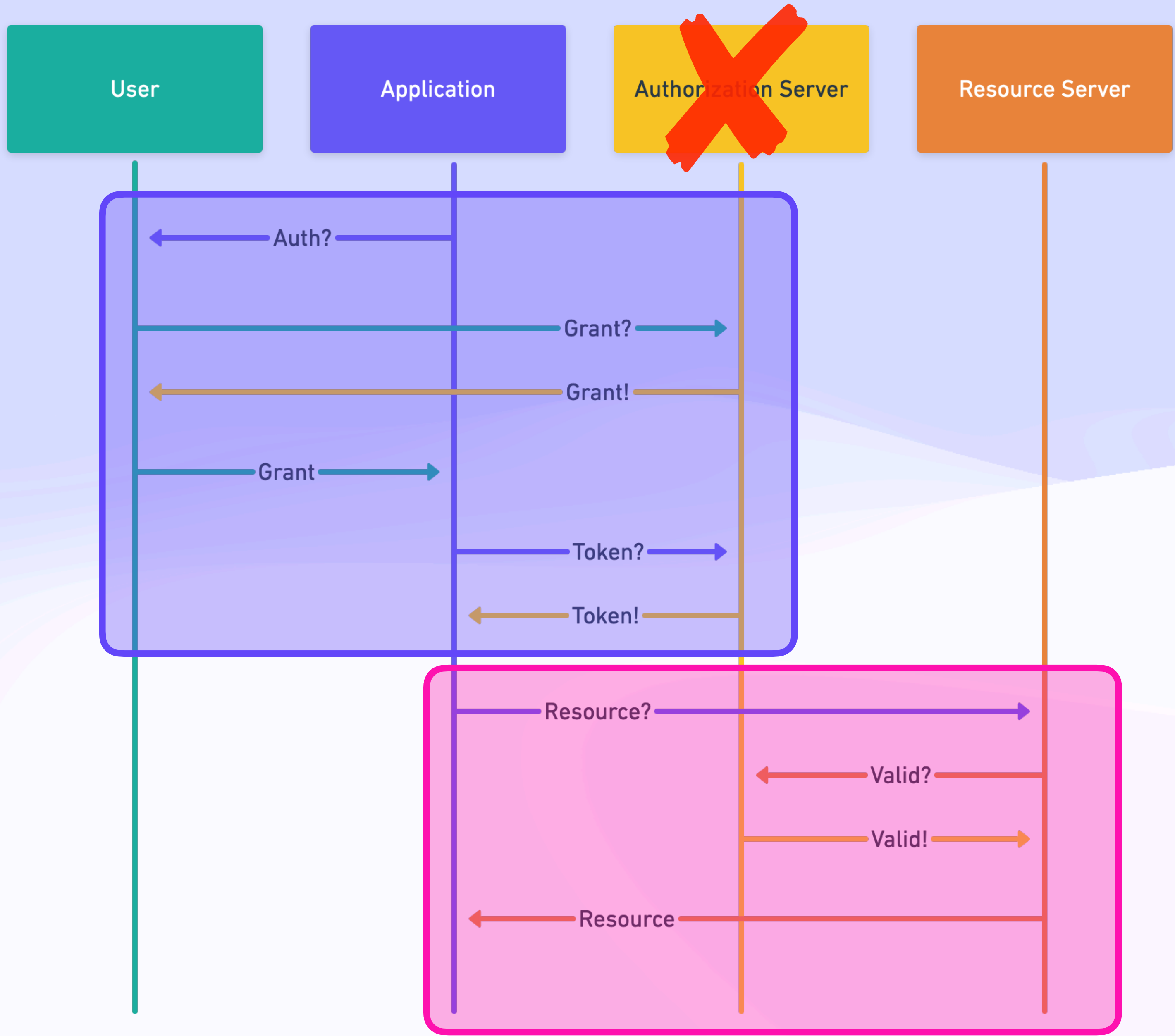
# Access Control

# *OAuth Sequence*



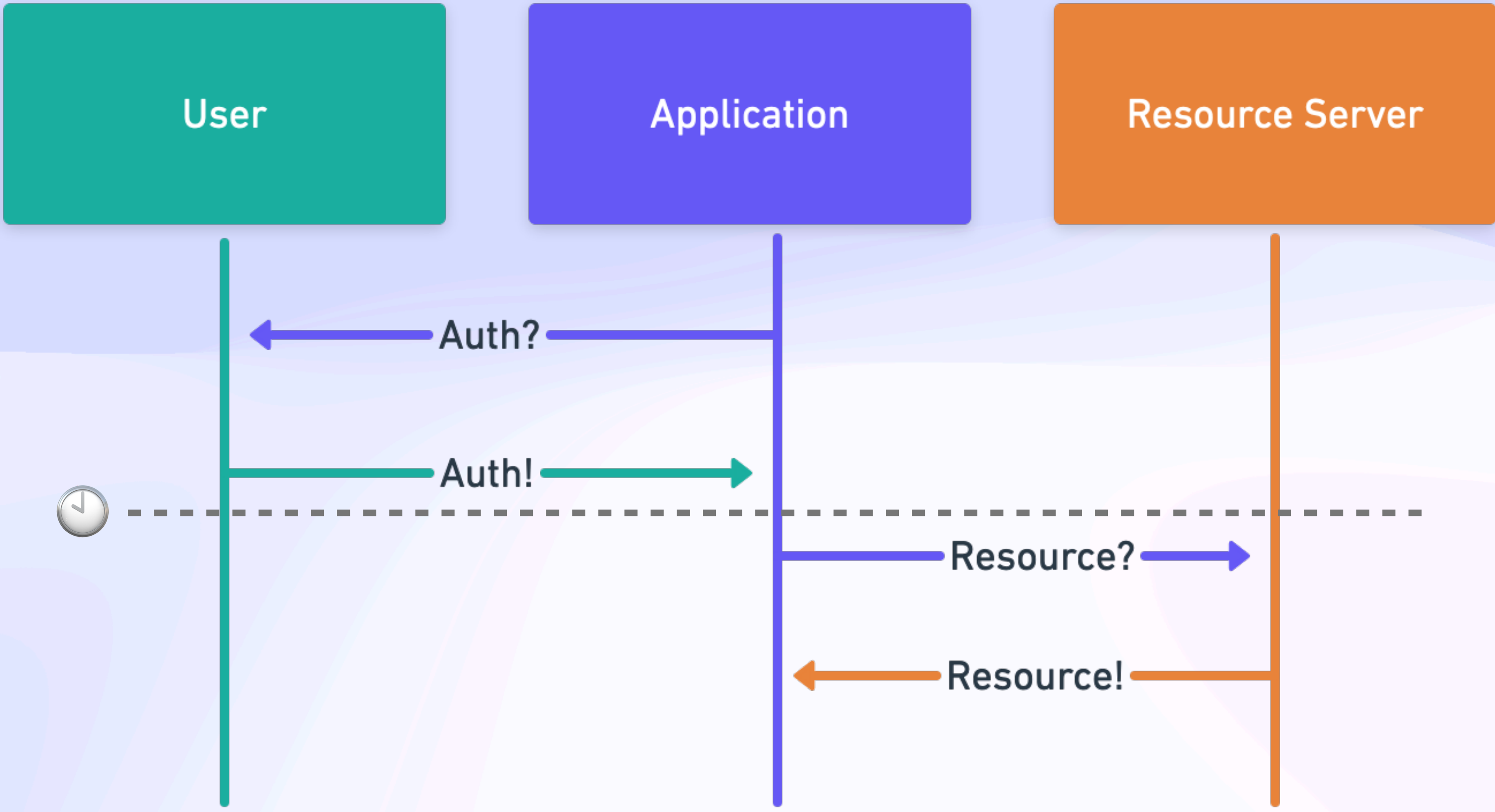
# Access Control

# *OAuth Sequence*



# Access Control

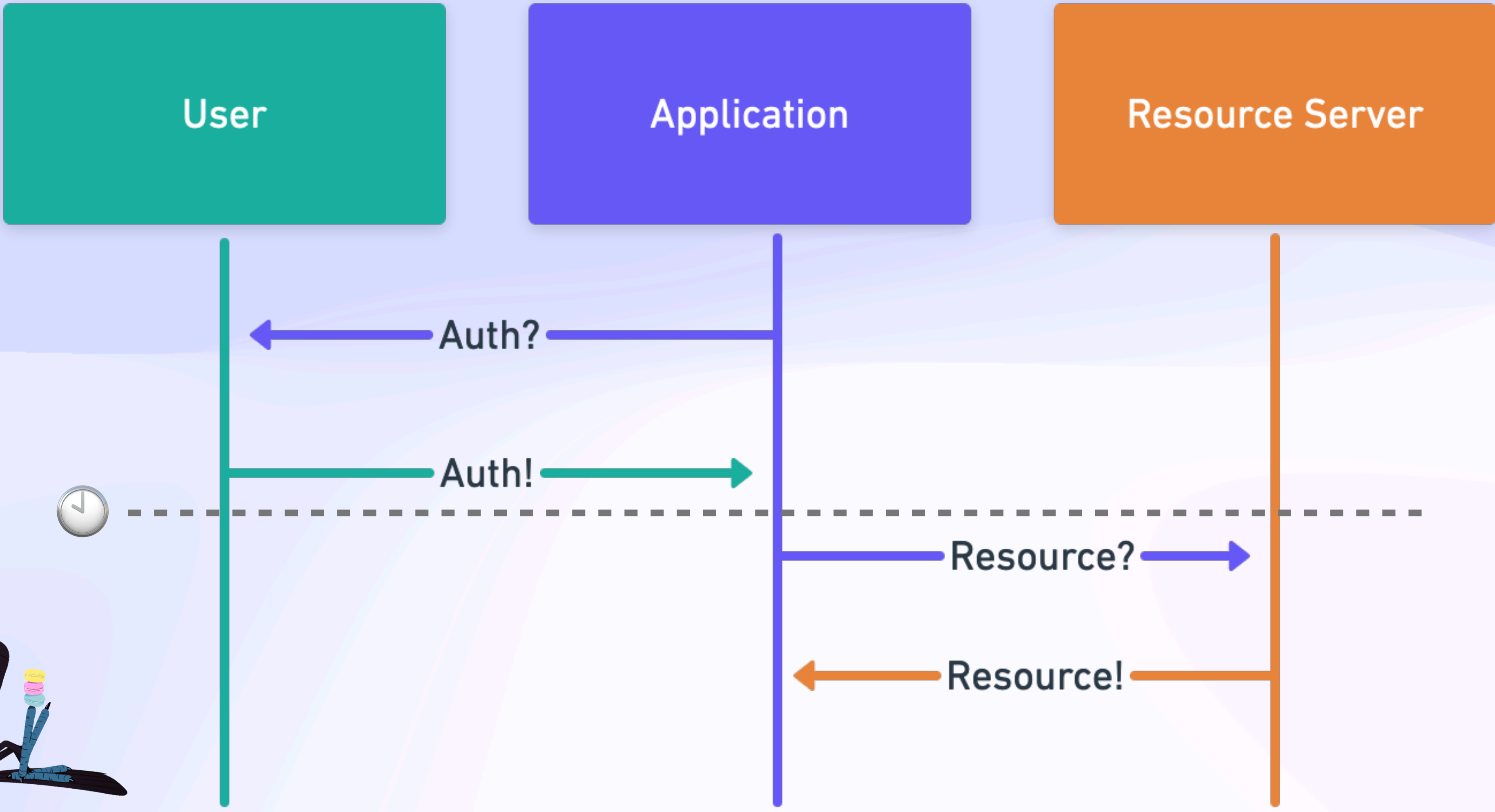
# *UCAN Sequence*





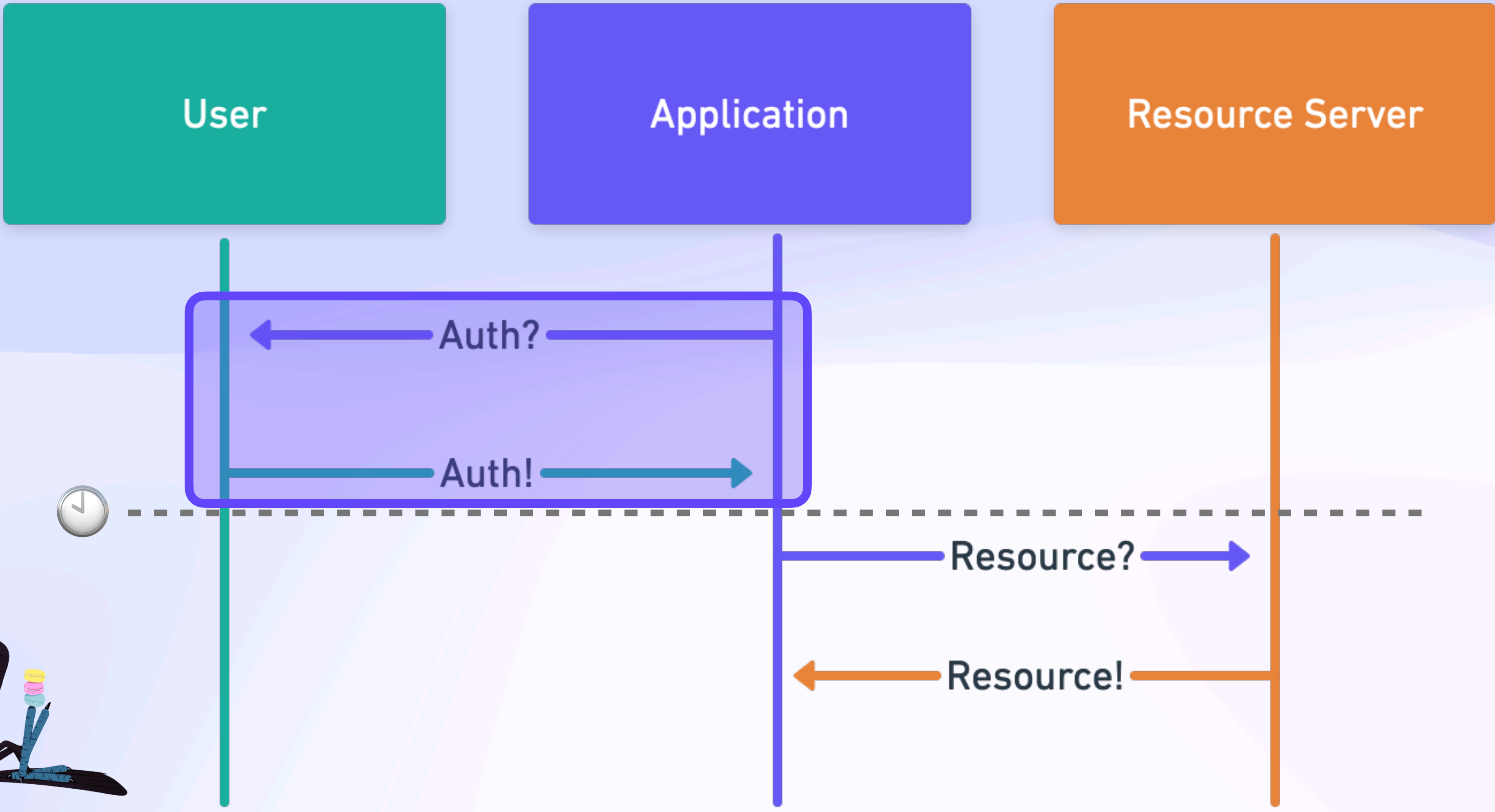
# Access Control

# UCAN Sequence



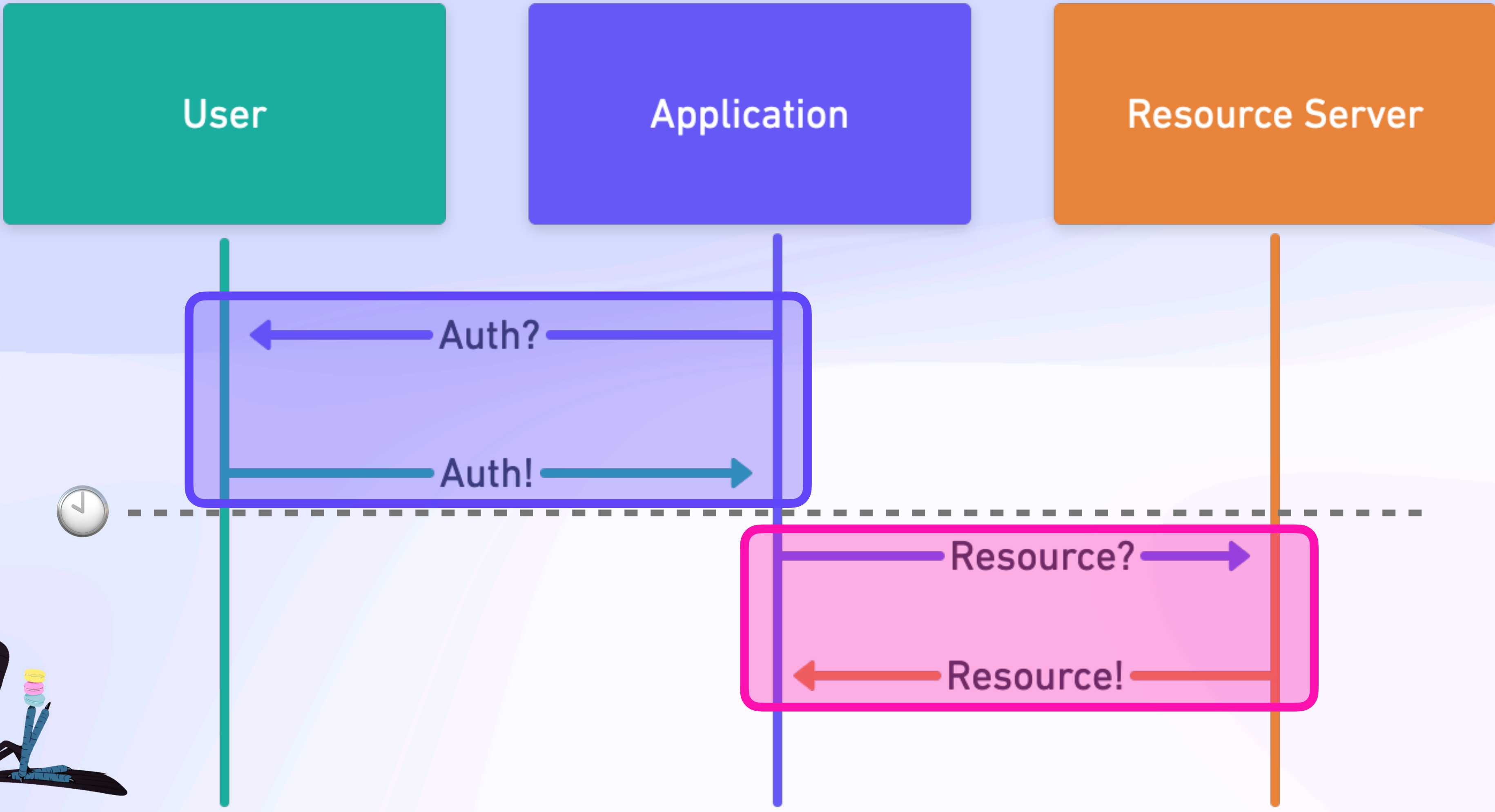
# Access Control

# *UCAN Sequence*



# Access Control

# UCAN Sequence



Goodbye Cloud, Hello Crowd

# *Universal Compute*



Compute Substrate

***Code-as-Data***



Compute Substrate

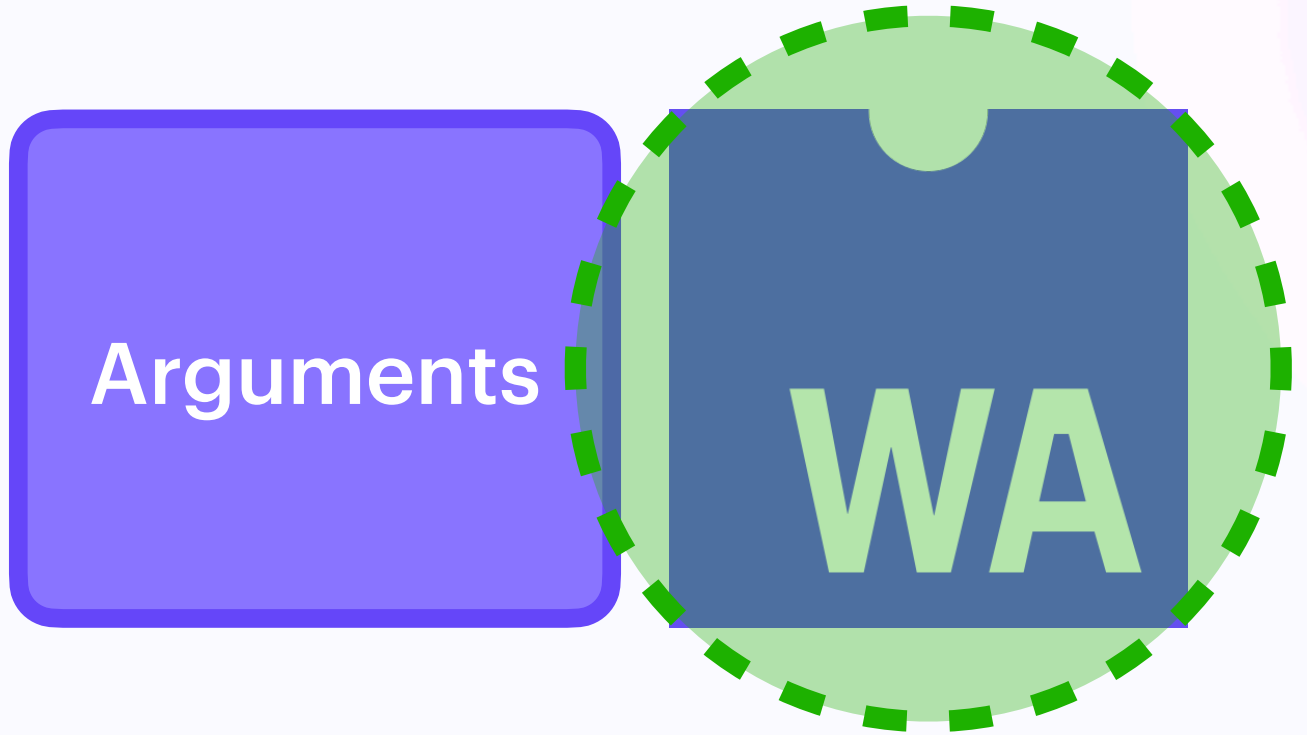
# ***Code-as-Data***

Arguments

WA

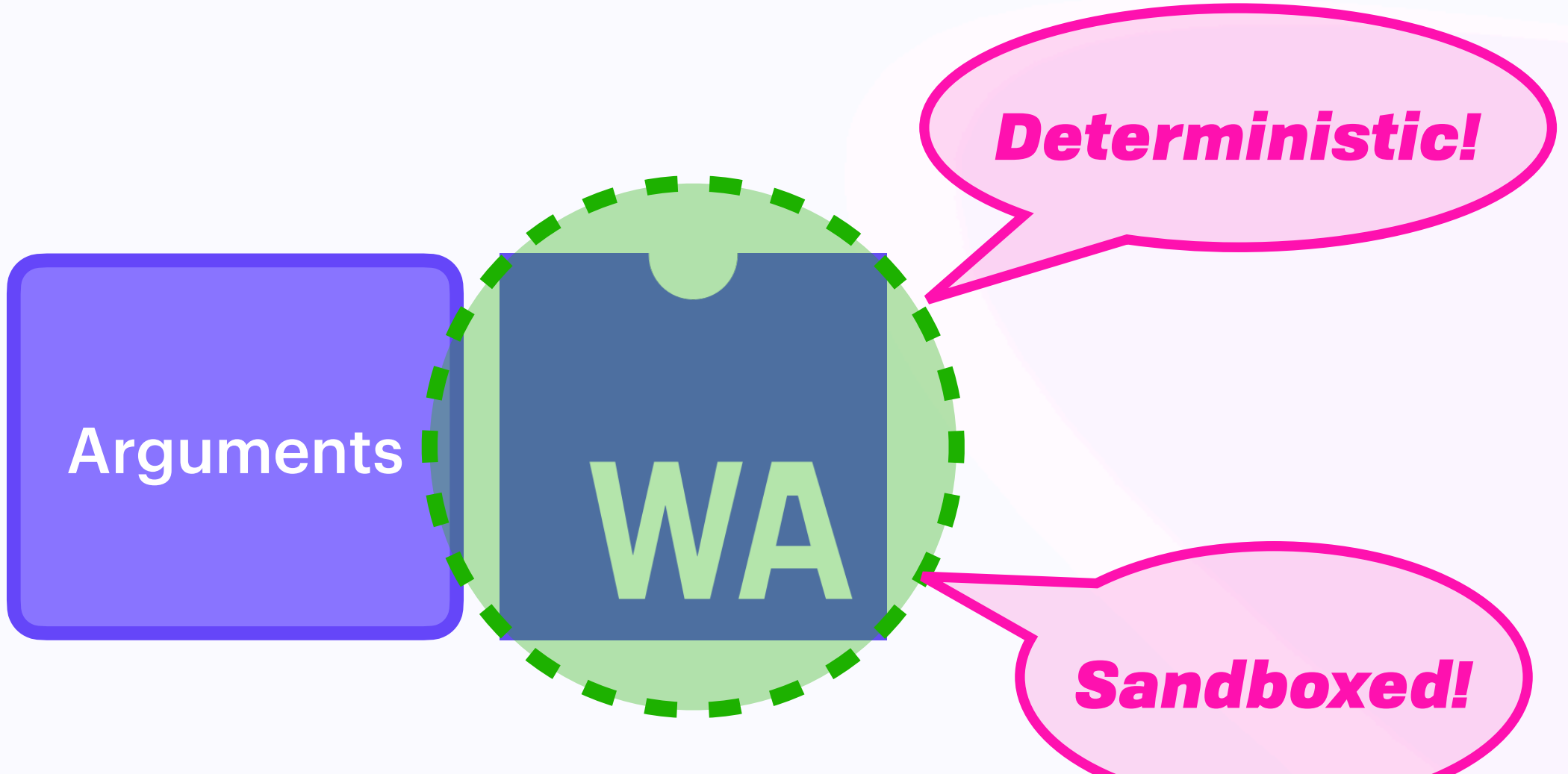
Compute Substrate

# *Code-as-Data*



Compute Substrate

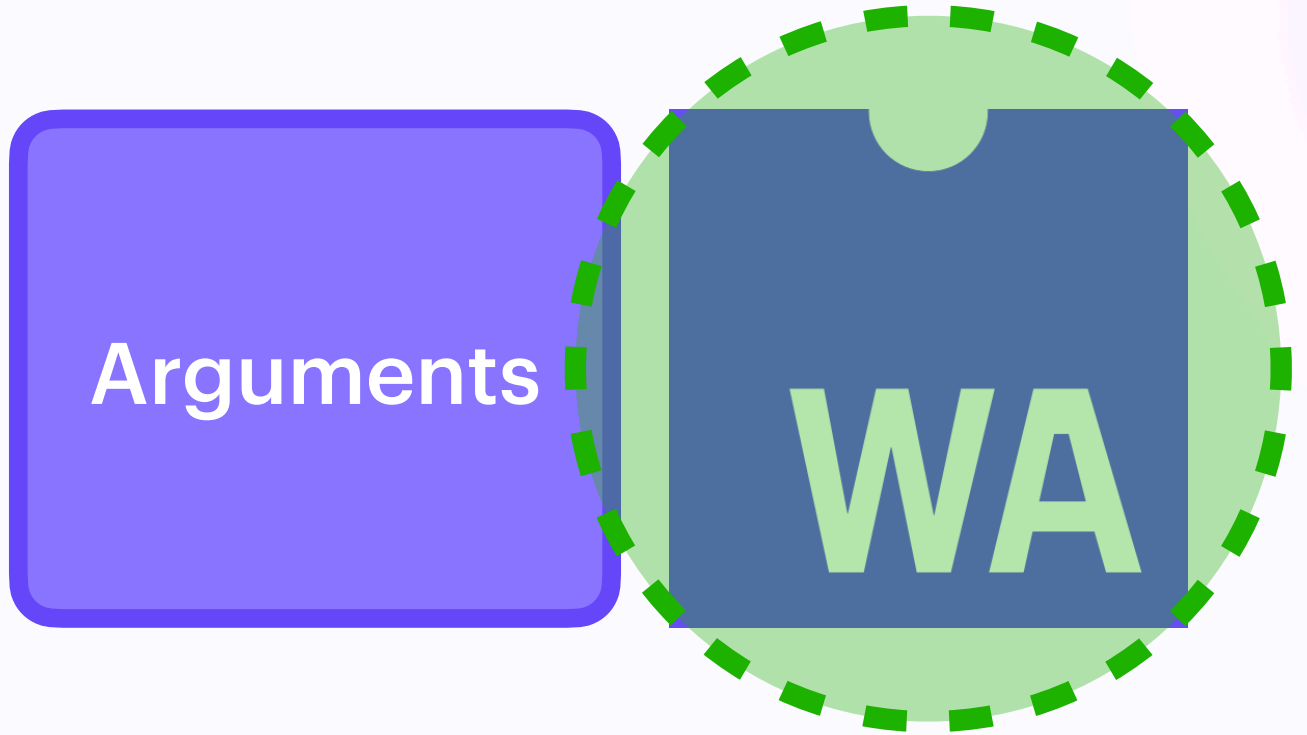
# ***Code-as-Data***





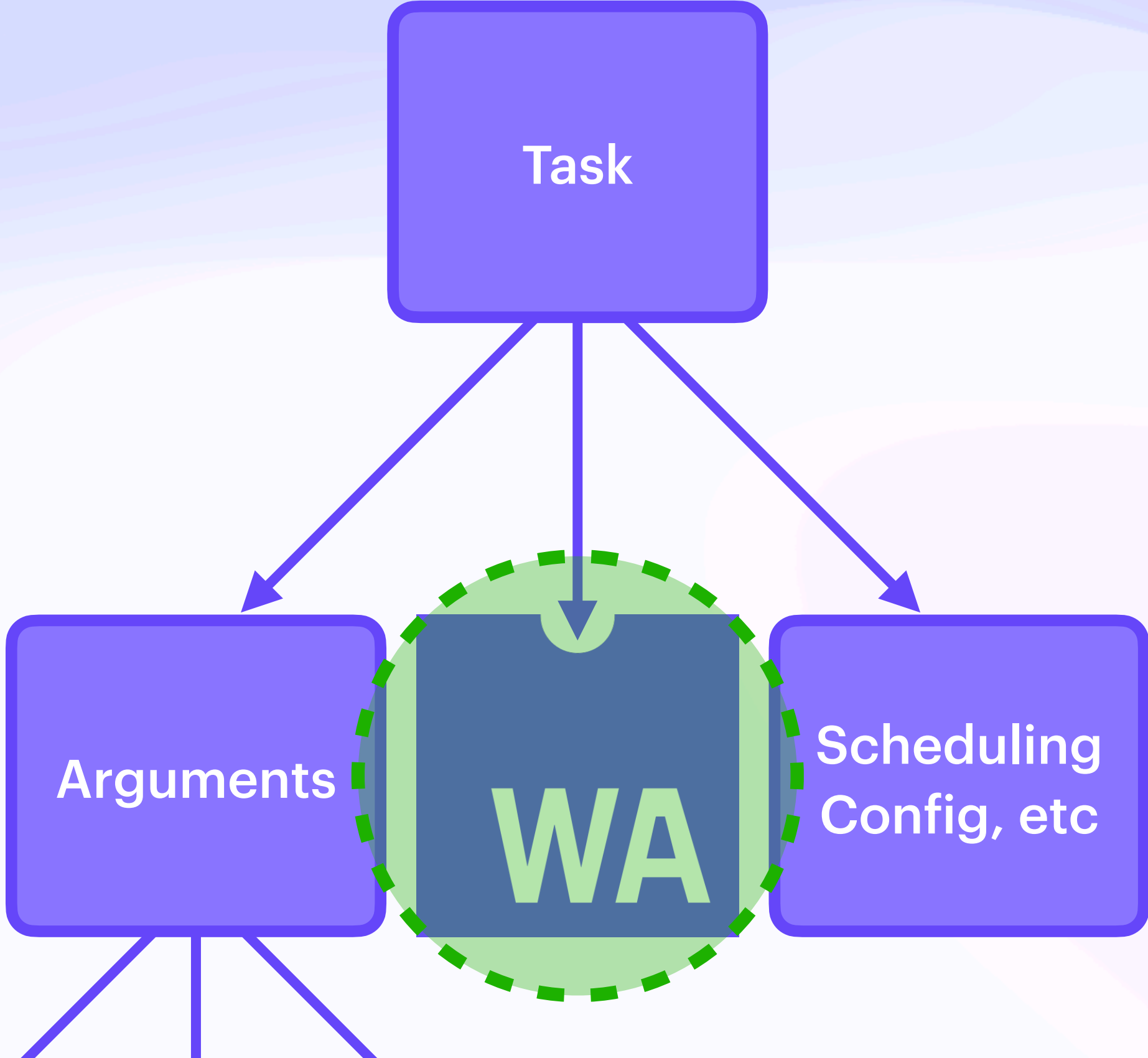
Compute Substrate

# *Code-as-Data*



Compute Substrate

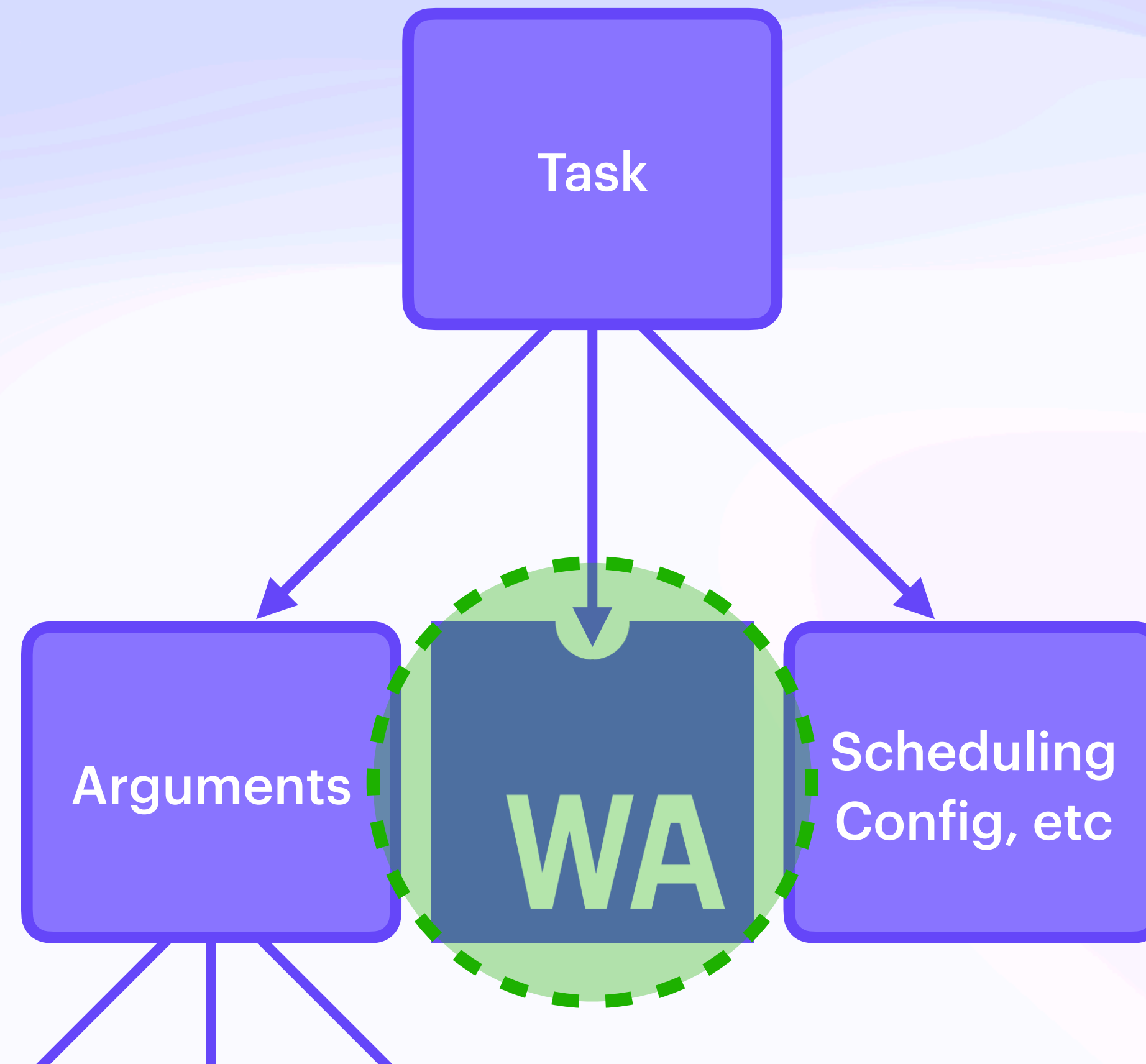
# *Code-as-Data*



# Compute Substrate

## *Code-as-Data*

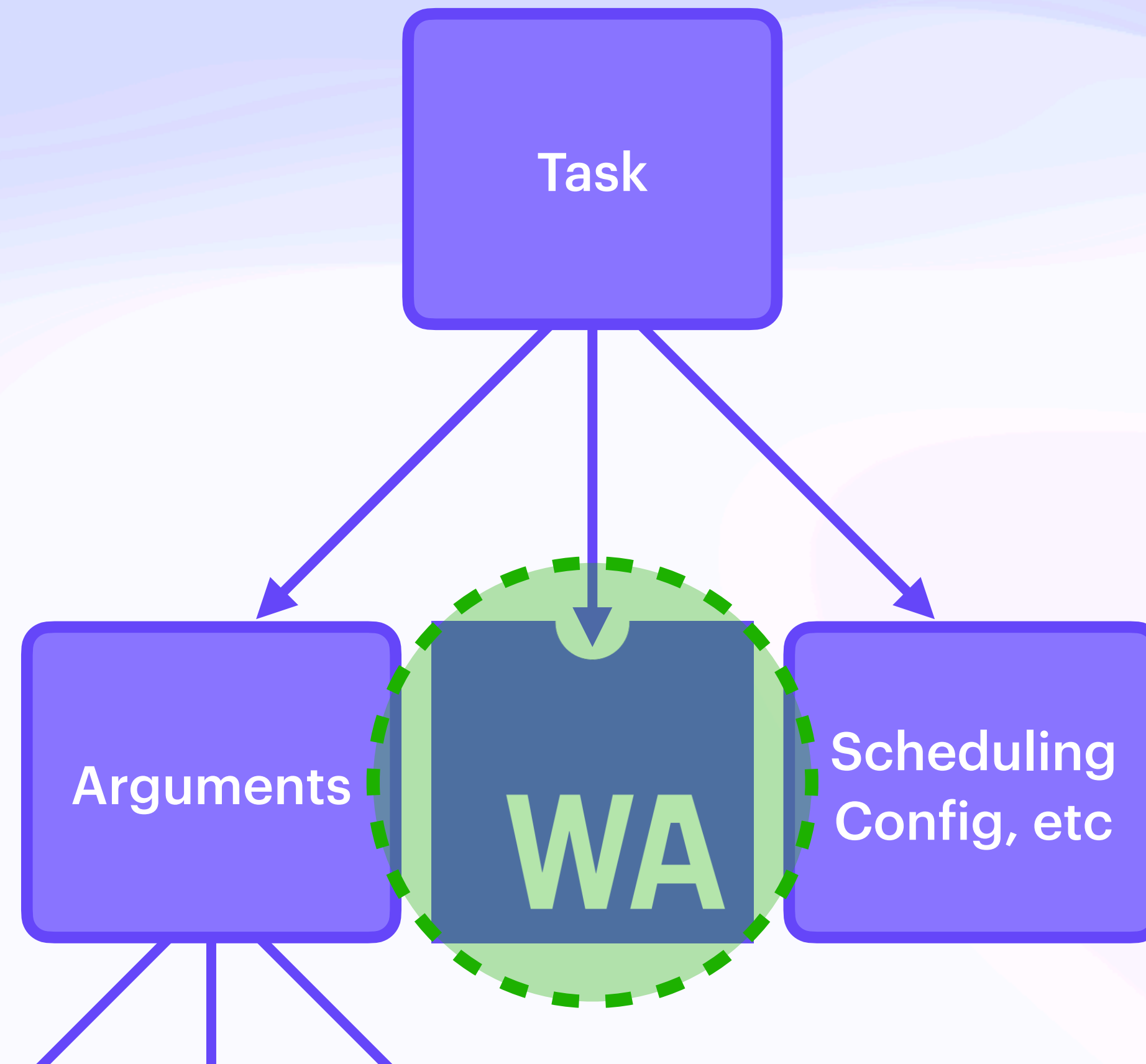
```
const message = () => alert("hello world")
```



# Compute Substrate

## *Code-as-Data*

```
const message = () => alert("hello world")  
message // Nothing happens
```



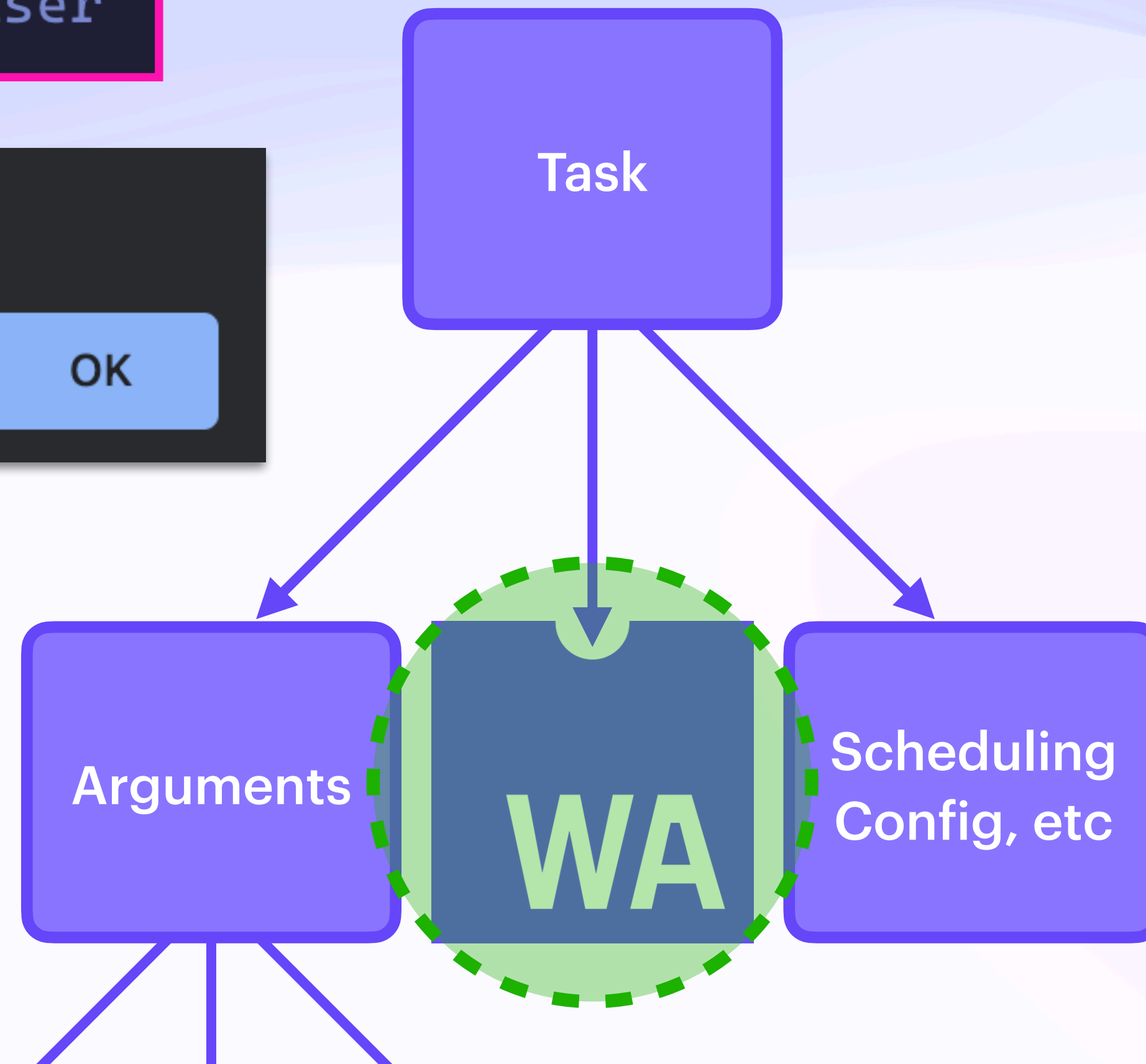
# Compute Substrate

## *Code-as-Data*

```
const message = () => alert("hello world")  
message // Nothing happens  
message() // A message interrupts the user
```

hello world

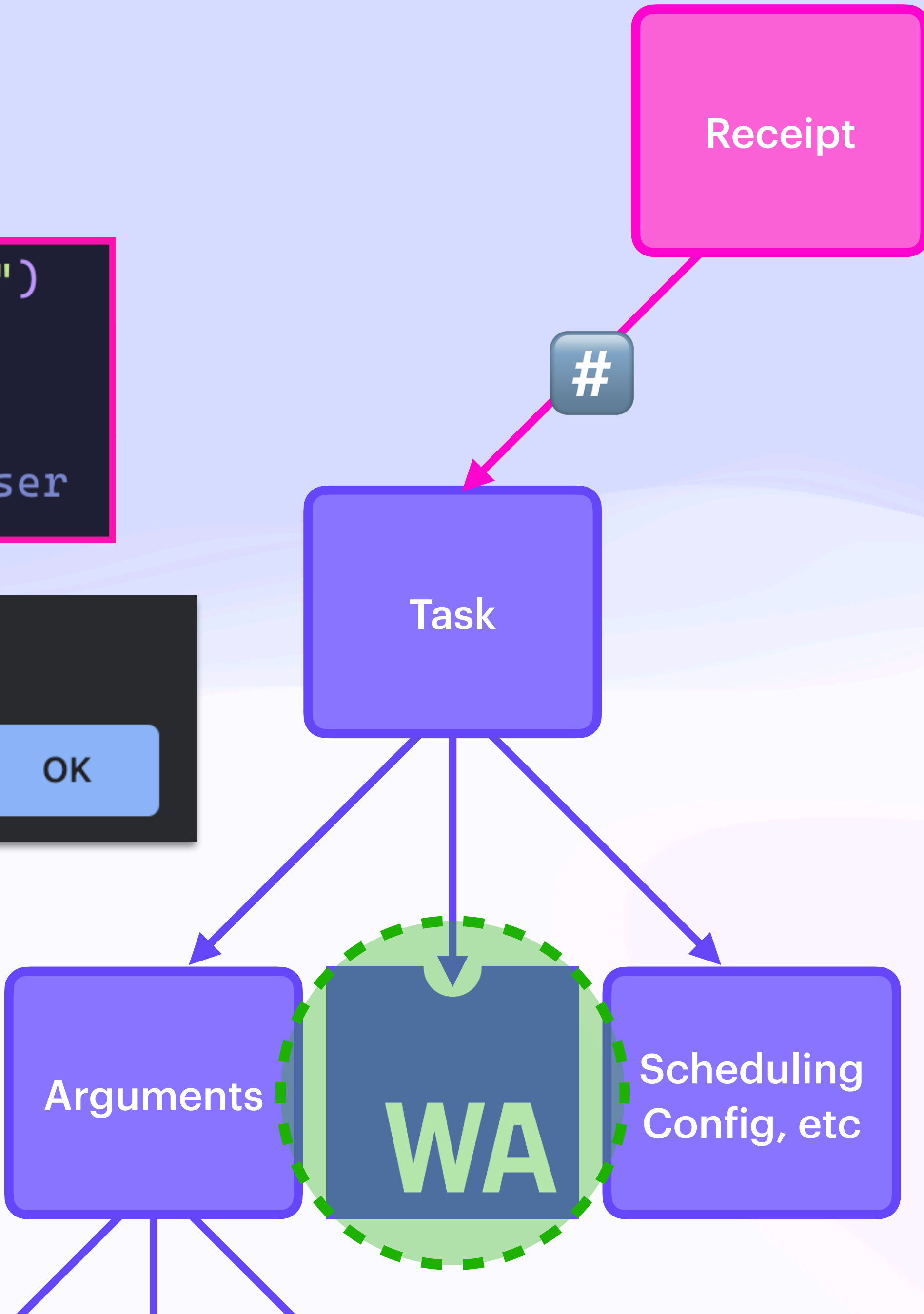
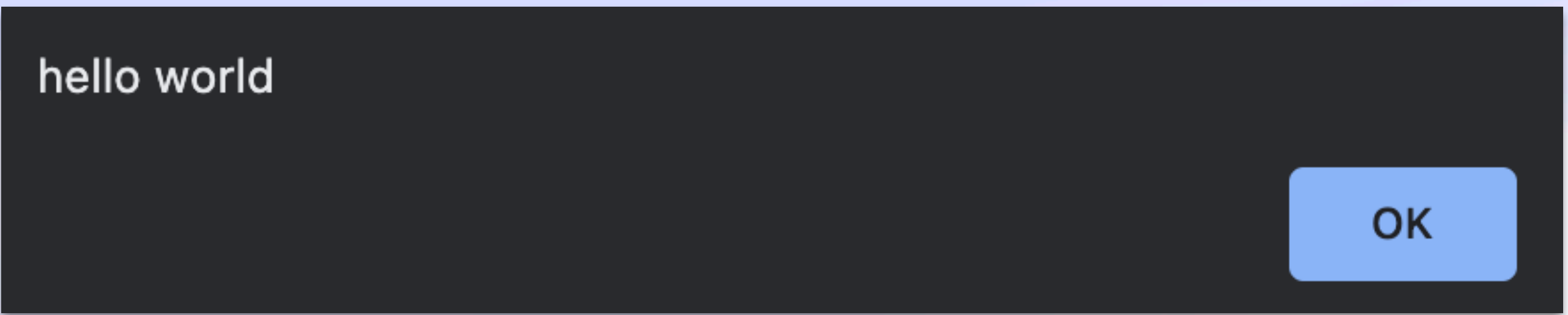
OK



# Compute Substrate

# Code-as-Data

```
const message = () => alert("hello world")  
message // Nothing happens  
message() // A message interrupts the user
```



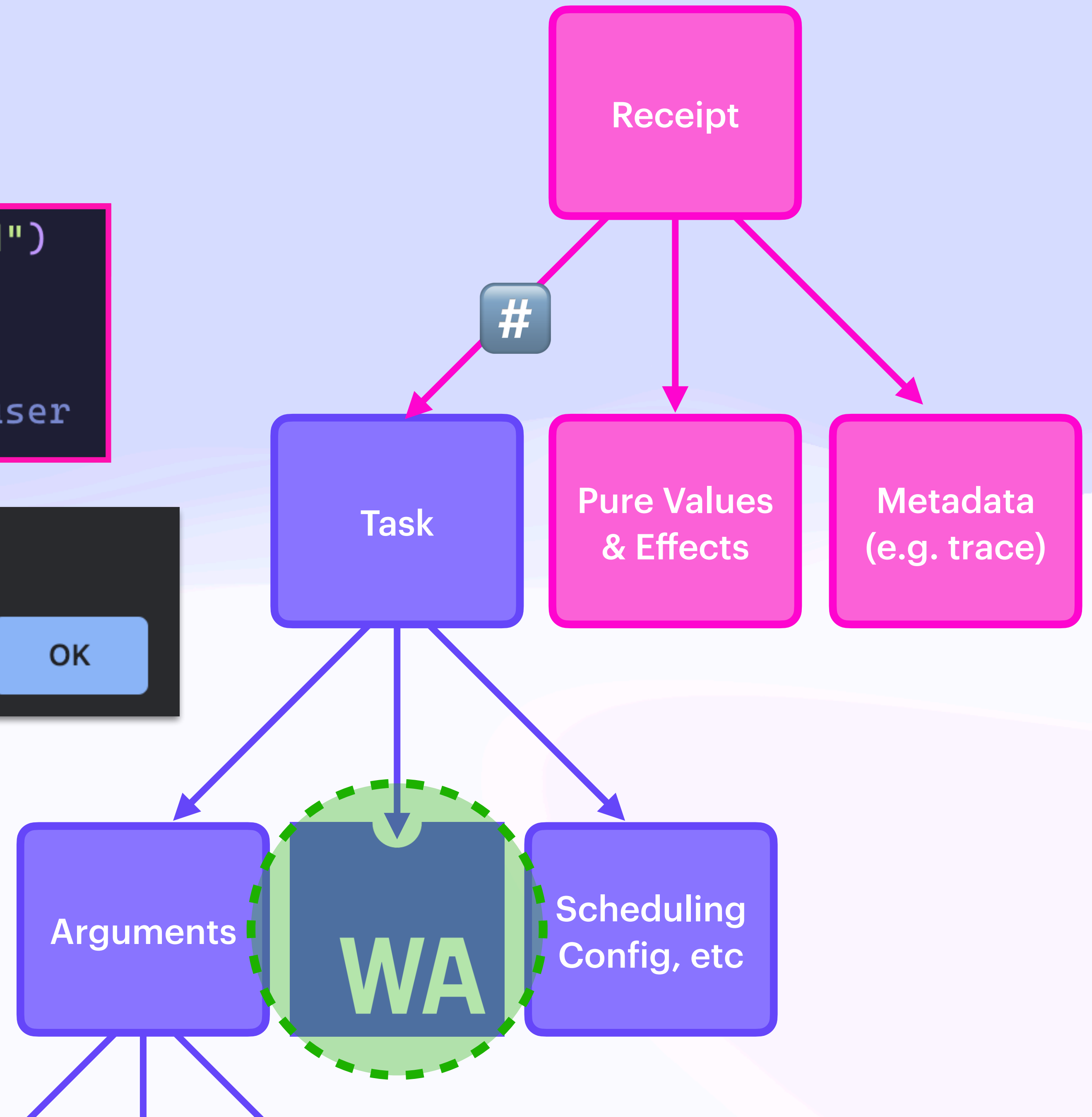
# Compute Substrate

## *Code-as-Data*

```
const message = () => alert("hello world")  
message // Nothing happens  
message() // A message interrupts the user
```

hello world

OK



Compute Substrate

***Distributed Invocation***





## Compute Substrate

# *Distributed Invocation*

```
{
  "uri": "ipfs://bafkreibmj5zo6x2g7kuzcqpsikr5q34rnzgbjkxk6rjf5ibu5szmx74hxy",
  "call": "wasm/run",
  "input": {
    "func": "add_one",
    "args": [42]
  }
}
```

# Compute Substrate

## *Distributed Invocation*

```
{
  "uri": "ipfs://bafkreibmj5zo6x2g7kuzcqpsikr5q34rnzgbjkxk6rjf5ibu5szmx74hxy",
  "call": "wasm/run",
  "input": {
    "func": "add_one",
    "args": [42]
  }
}
```

```
{
  "uri": "ipfs://bafkreibmj5zo6x2g7kuzcqpsikr5q34rnzgbjkxk6rjf5ibu5szmx74hxy",
  "call": "wasm/run",
  "input": {
    "func": "add_one",
    "args": [{"await/ok": {"/": "bafkreiauharffox63dv2iakndymassol3ryznr32tqii6ijw6ter3ksleu"}}]
  }
}
```

# Compute Substrate

## *Distributed Invocation*

```
{
  "uri": "ipfs://bafkreibmj5zo6x2g7kuzcqpsikr5q34rnzgbjkxk6rjf5ibu5szmx74hxy",
  "call": "wasm/run",
  "input": {
    "func": "add_one",
    "args": [42]
  }
}
```

```
{
  "uri": "ipfs://bafkreibmj5zo6x2g7kuzcqpsikr5q34rnzgbjkxk6rjf5ibu5szmx74hxy",
  "call": "wasm/run",
  "input": {
    "func": "add_one",
    "args": [{"await/ok"} {"":"/": "bafkreiauharffox63dv2iakndymassol3ryznr32tqii6ijw6ter3ksleu"}}]
  }
}
```

# Compute Substrate

## *Distributed Invocation*

```
{
  "uri": "ipfs://bafkreibmj5zo6x2g7kuzcqpsikr5q34rnzgbjkxk6rjf5ibu5szmx74hxy",
  "call": "wasm/run",
  "input": {
    "func": "add_one",
    "args": [42]
  }
}
```

```
{
  "uri": "ipfs://bafkreibmj5zo6x2g7kuzcqpsikr5q34rnzgbjkxk6rjf5ibu5szmx74hxy",
  "call": "wasm/run",
  "input": {
    "func": "add_one",
    "args": [{"await/ok"} {"":"/": "bafkreiauharffox63dv2iakndymassol3ryznr32tqii6ijw6ter3ksleu"}}]
  }
}
```

Compute Substrate

***Distributed Invocation***



Compute Substrate

# *Distributed Invocation*

```
dns:example.com/TYPE=TXT  
crud/update
```

# Compute Substrate

## *Distributed Invocation*

```
dns:example.com/TYPE=TXT  
crud/update
```

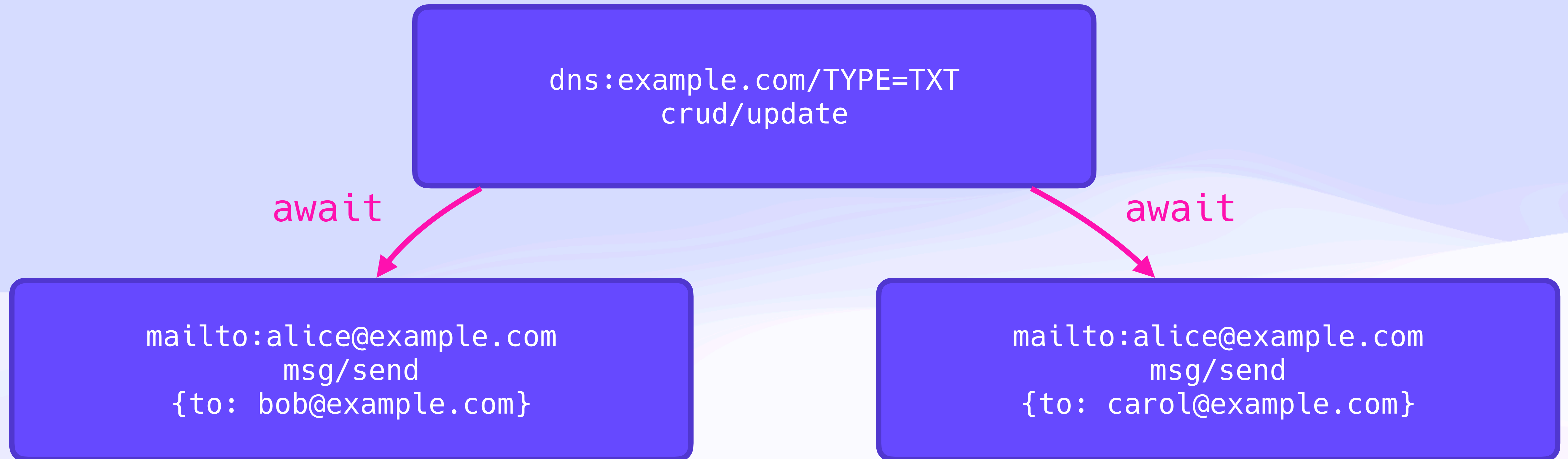
await



```
mailto:alice@example.com  
msg/send  
{to: bob@example.com}
```

# Compute Substrate

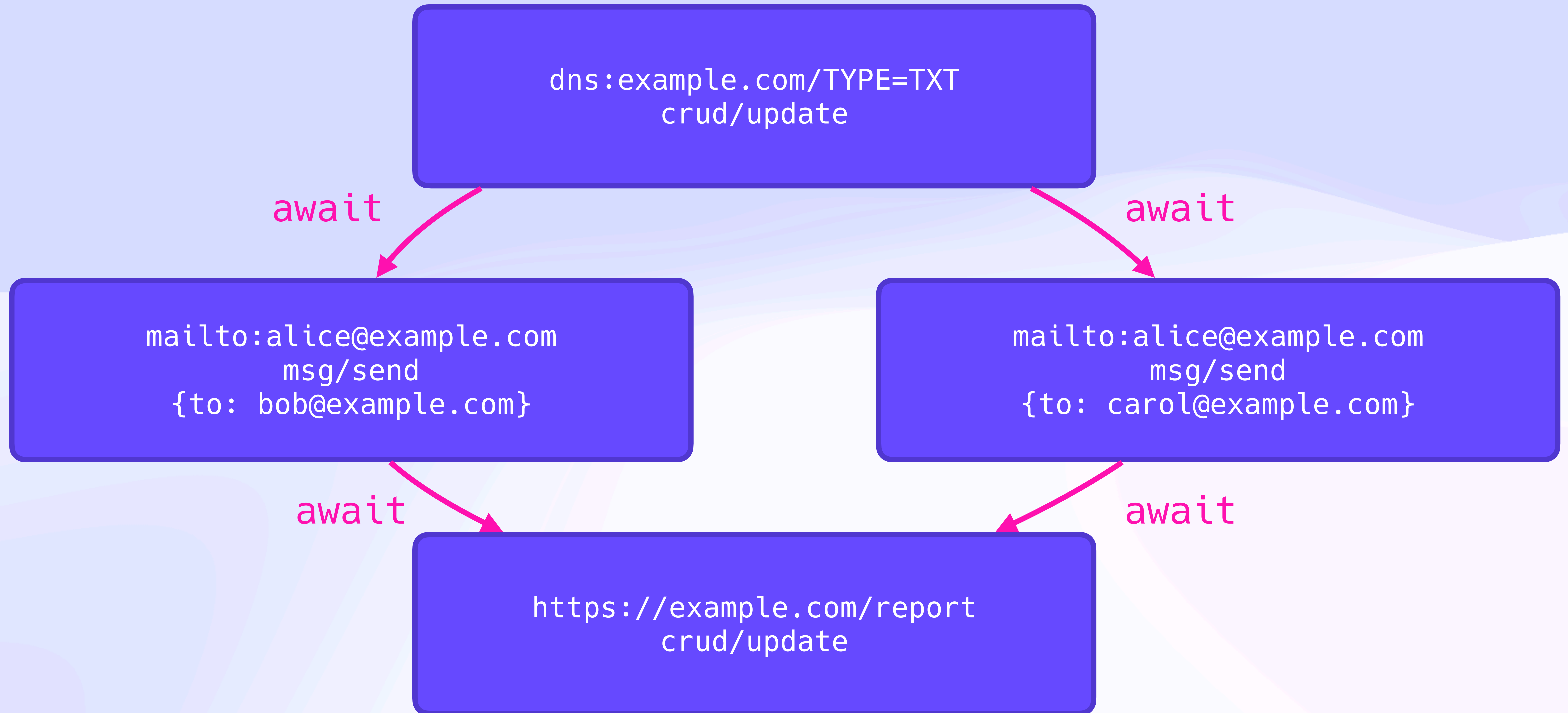
## *Distributed Invocation*





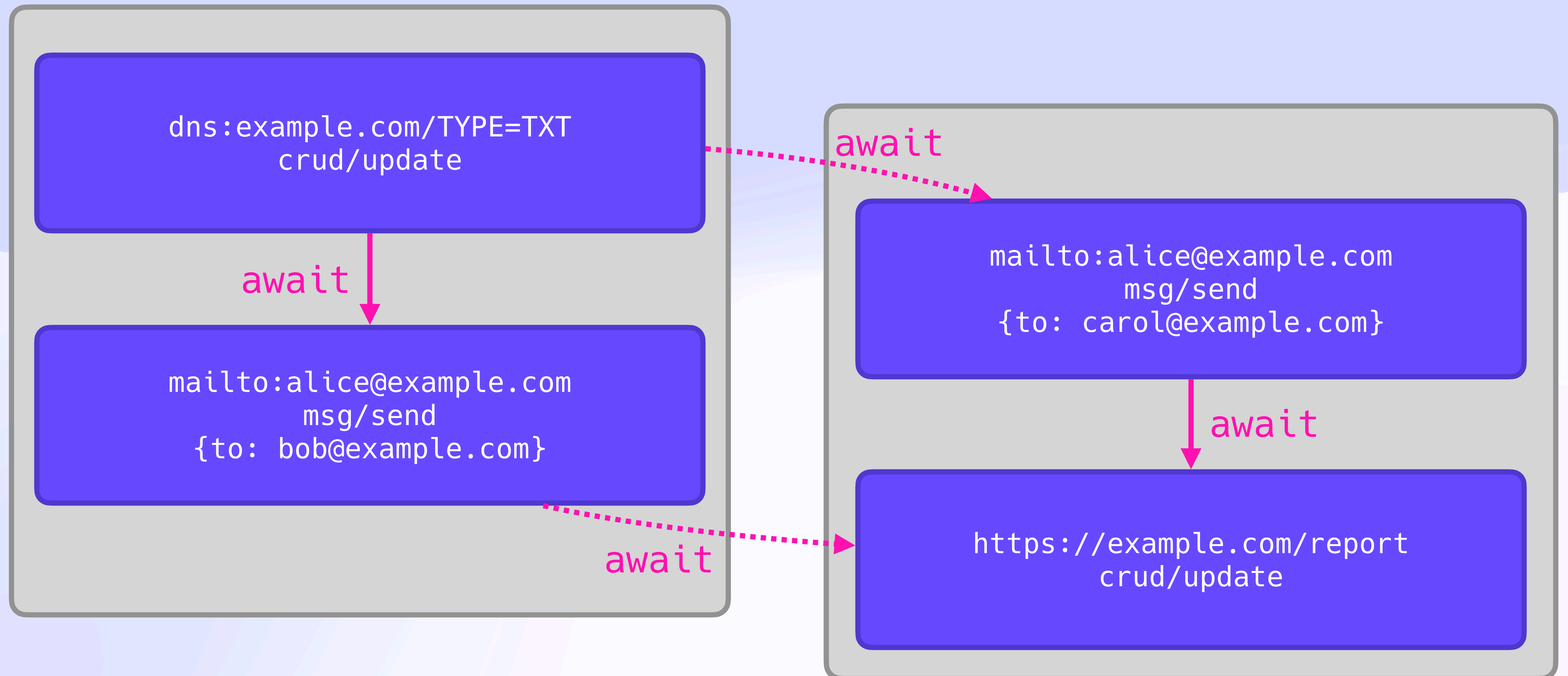
# Compute Substrate

## *Distributed Invocation*



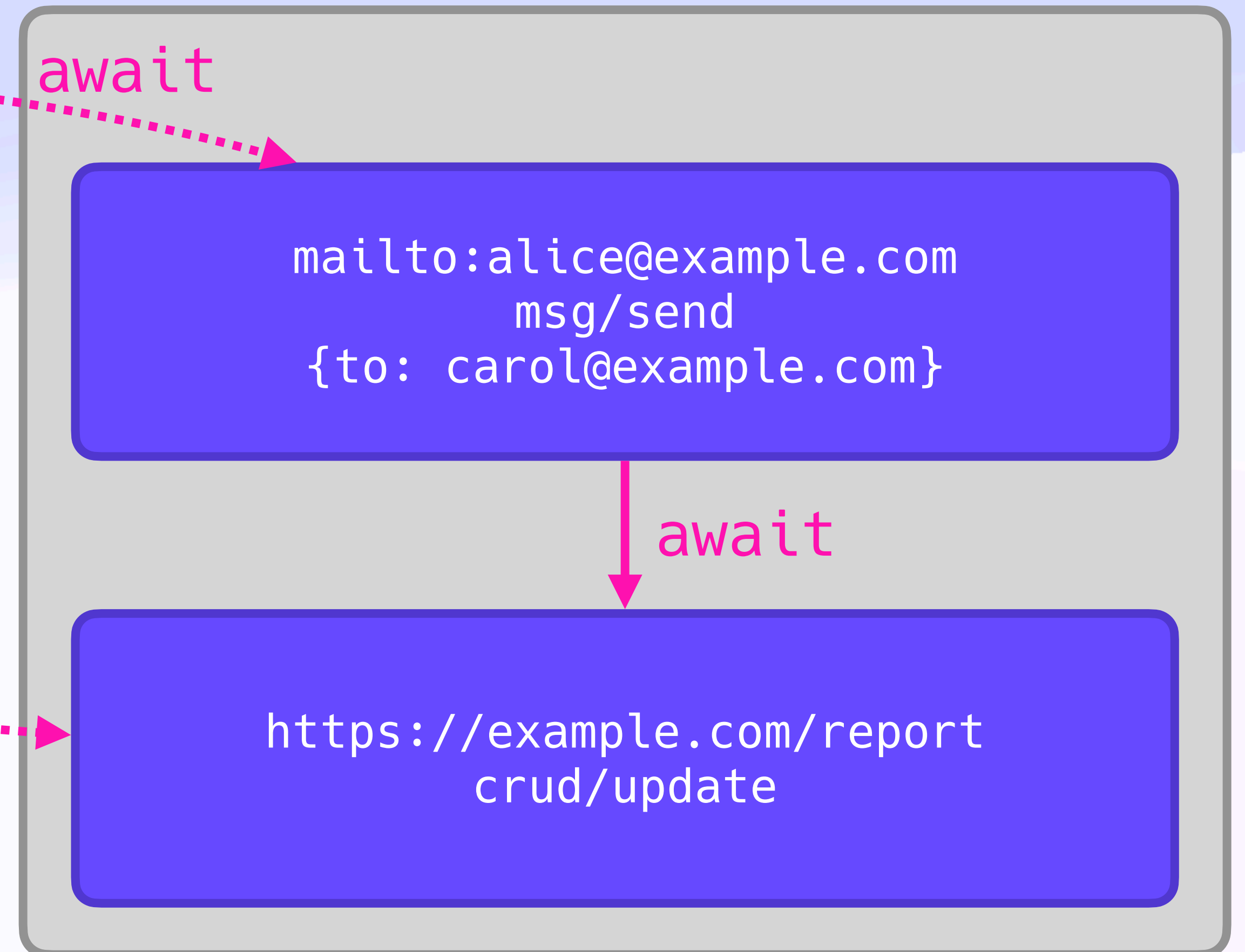
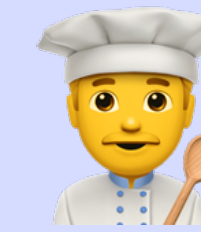
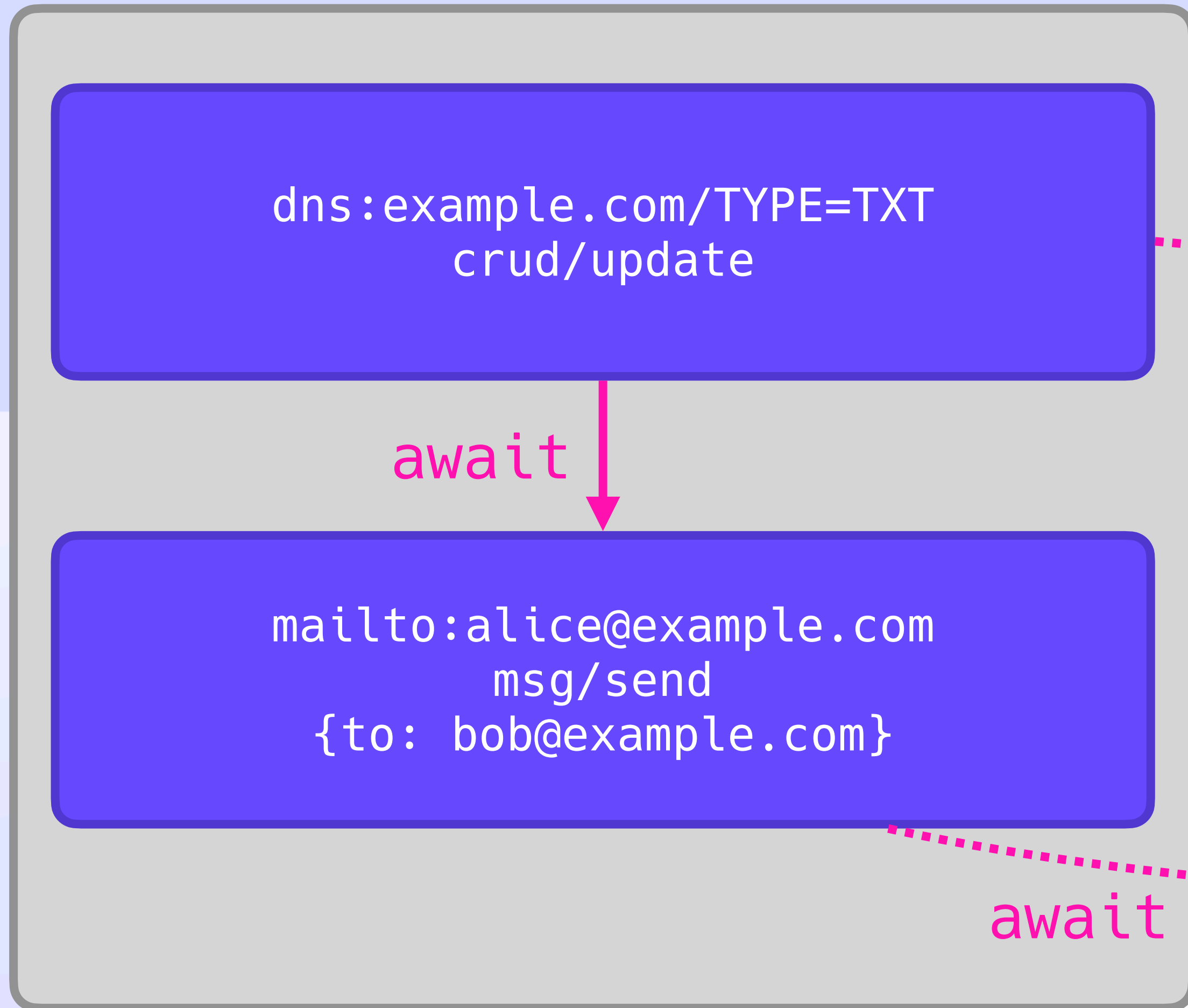
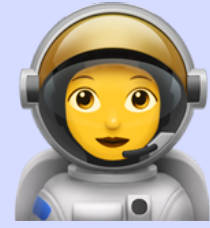
# Compute Substrate

## *Distributed Invocation*



# Compute Substrate

## *Distributed Invocation*

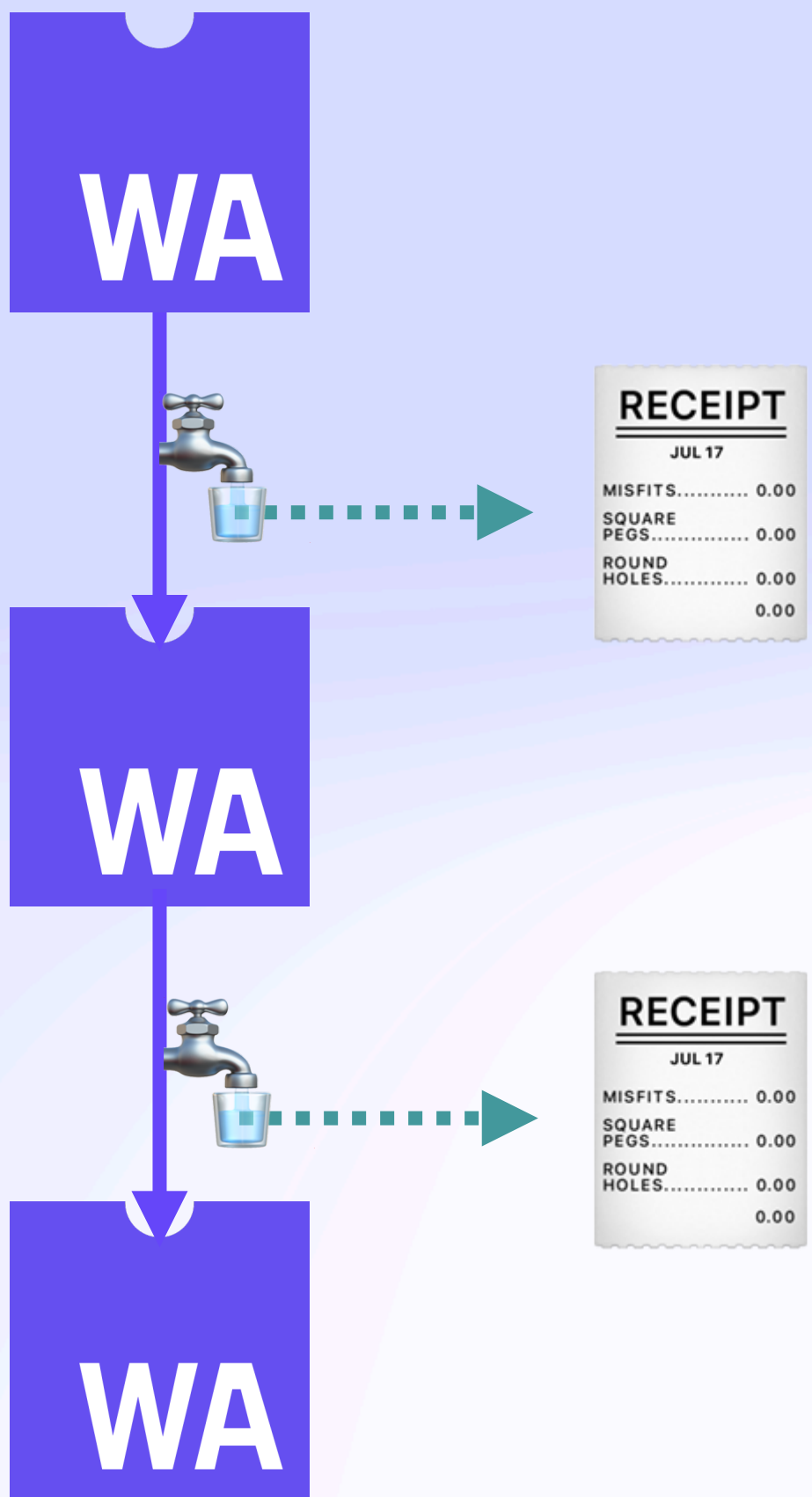


Compute Substrate

***Cache, Suspend, Move, Verify***

# Compute Substrate

# Cache, Suspend, Move, Verify

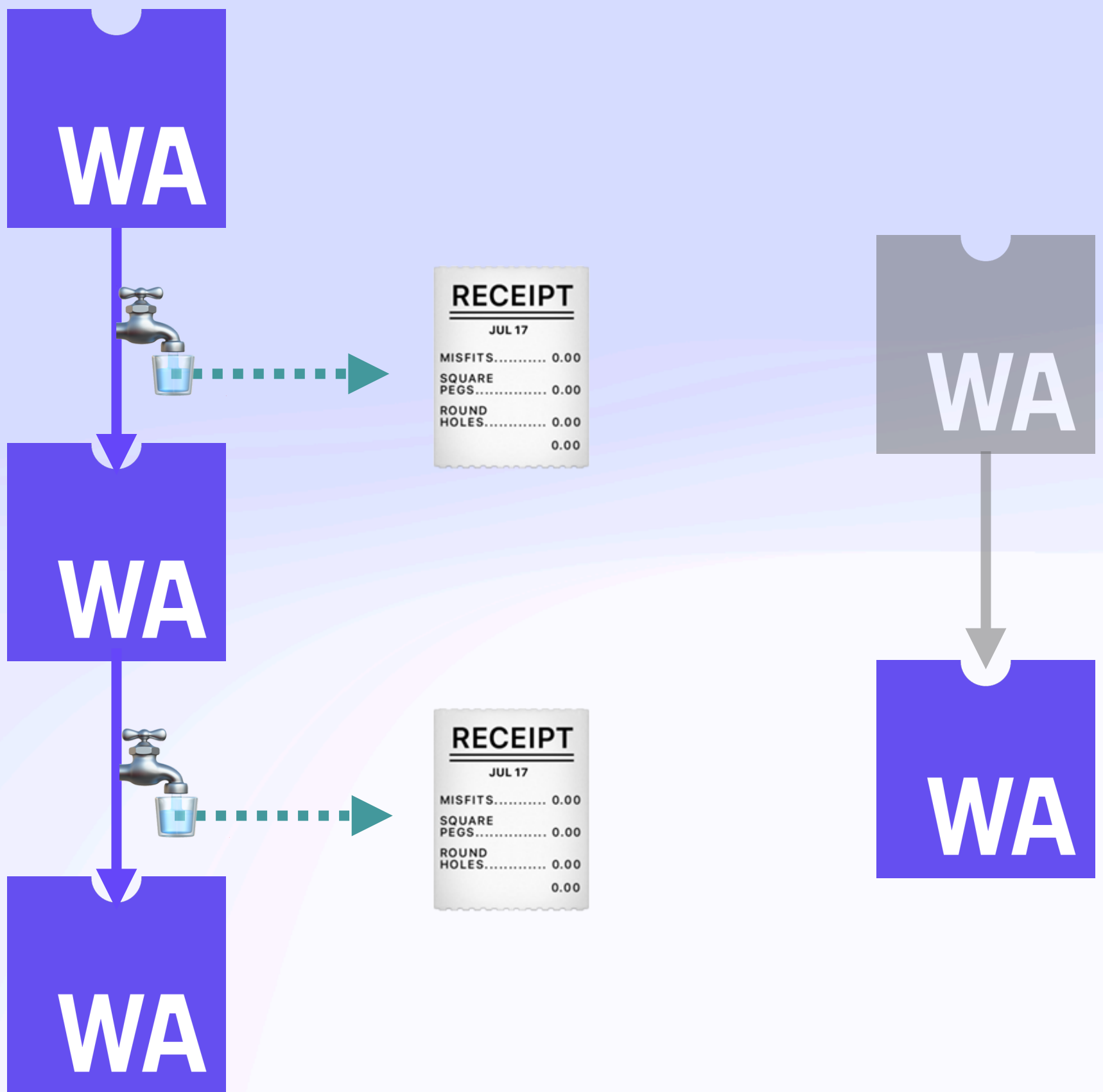


| RECEIPT          |      |
|------------------|------|
| JUL 17           |      |
| MISFITS.....     | 0.00 |
| SQUARE PEGS..... | 0.00 |
| ROUND HOLES..... | 0.00 |
|                  | 0.00 |

| RECEIPT          |      |
|------------------|------|
| JUL 17           |      |
| MISFITS.....     | 0.00 |
| SQUARE PEGS..... | 0.00 |
| ROUND HOLES..... | 0.00 |
|                  | 0.00 |

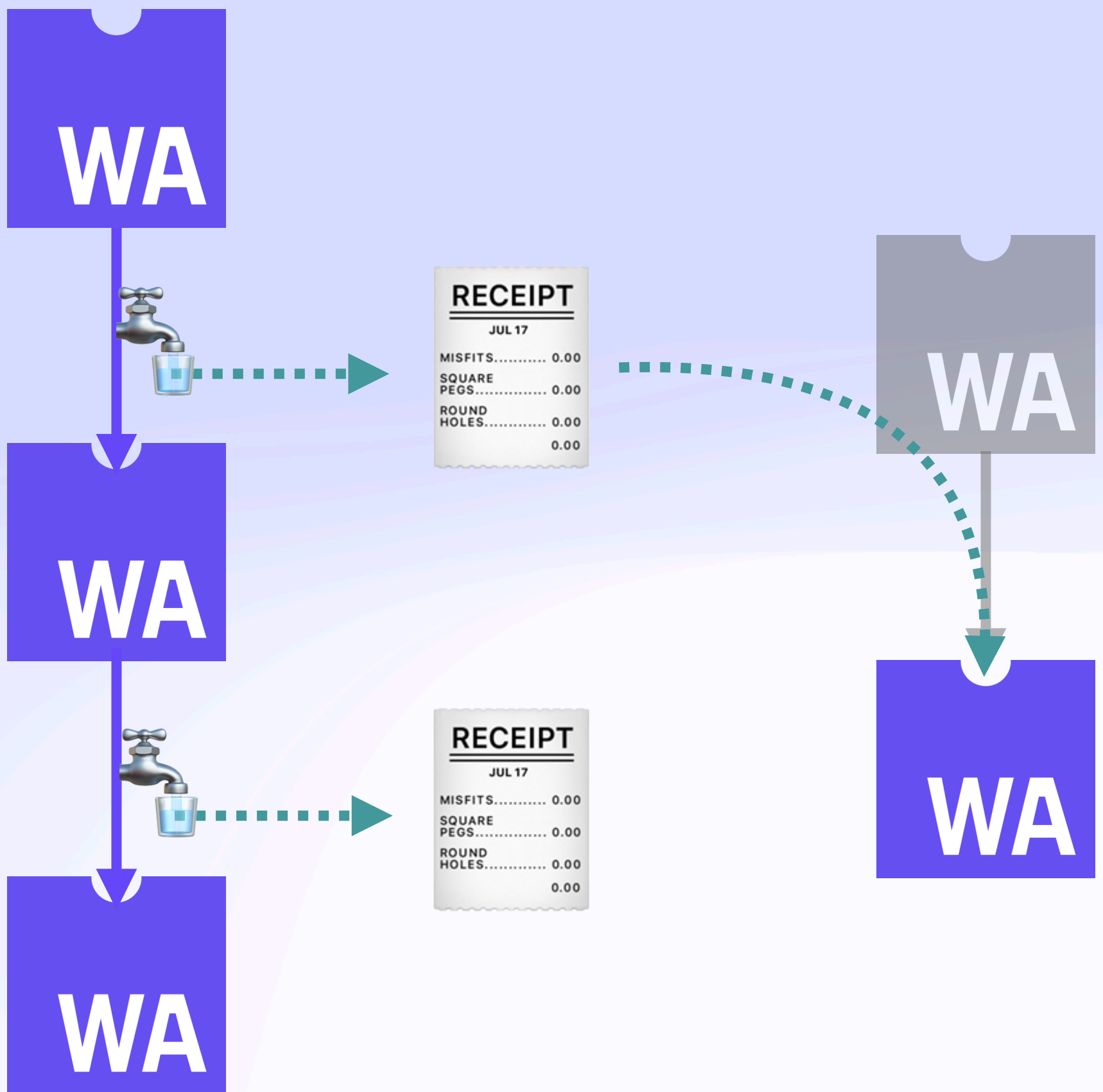
# Compute Substrate

# Cache, Suspend, Move, Verify



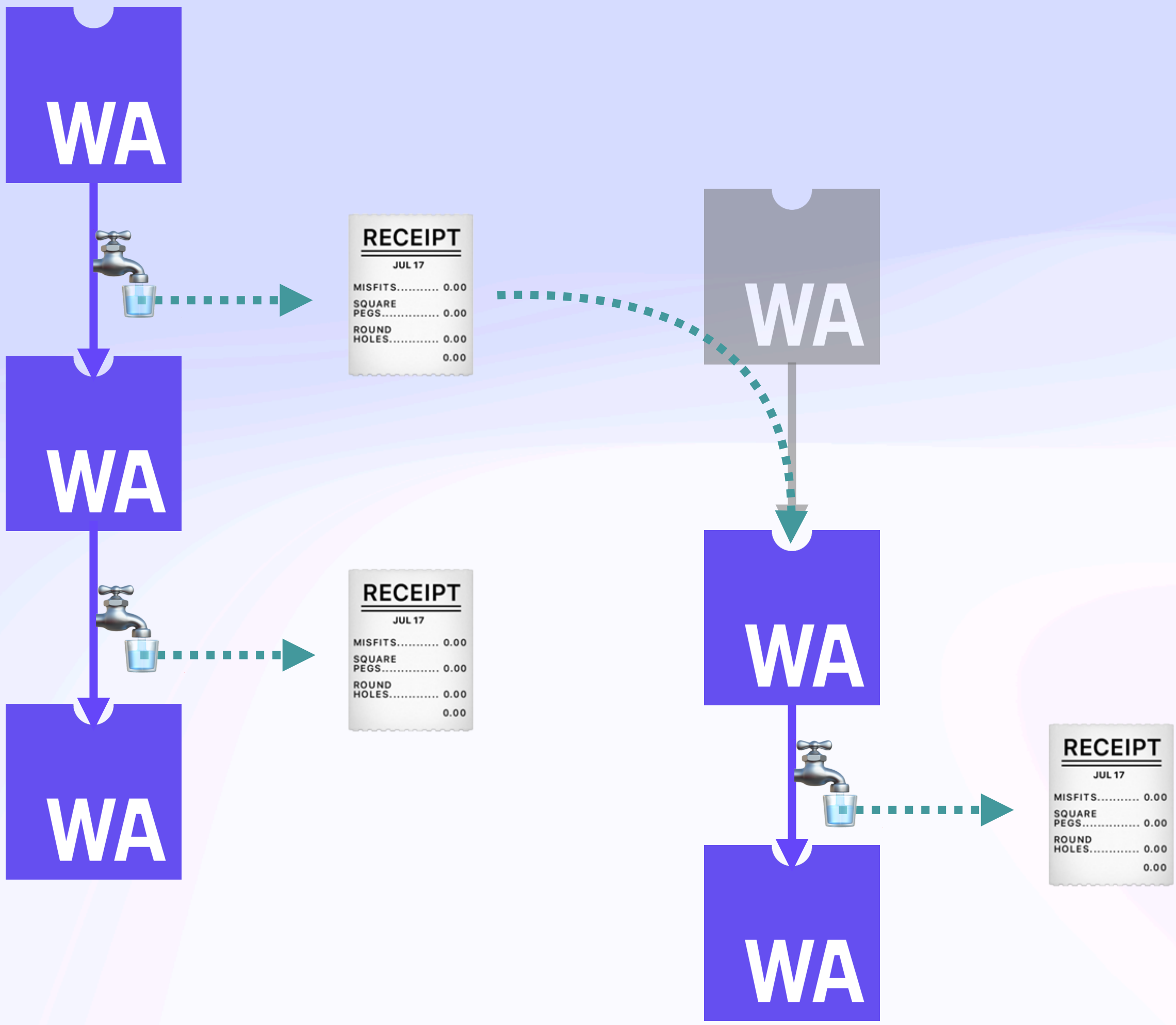
# Compute Substrate

# Cache, Suspend, Move, Verify



# Compute Substrate

# Cache, Suspend, Move, Verify





Compute Substrate

***Command! Query? Compute.***

Mutation Effect Stream - - - - -

Query Effect Stream - - - - -

Pure Function Stream - - - - -

Base Event Stream \_\_\_\_\_

$t \rightarrow$

Compute Substrate

***Command! Query? Compute.***

Mutation Effect Stream - - - - -

Query Effect Stream - - - - -

Pure Function Stream - - - - -

Base Event Stream \_\_\_\_\_



$t \rightarrow$

Compute Substrate

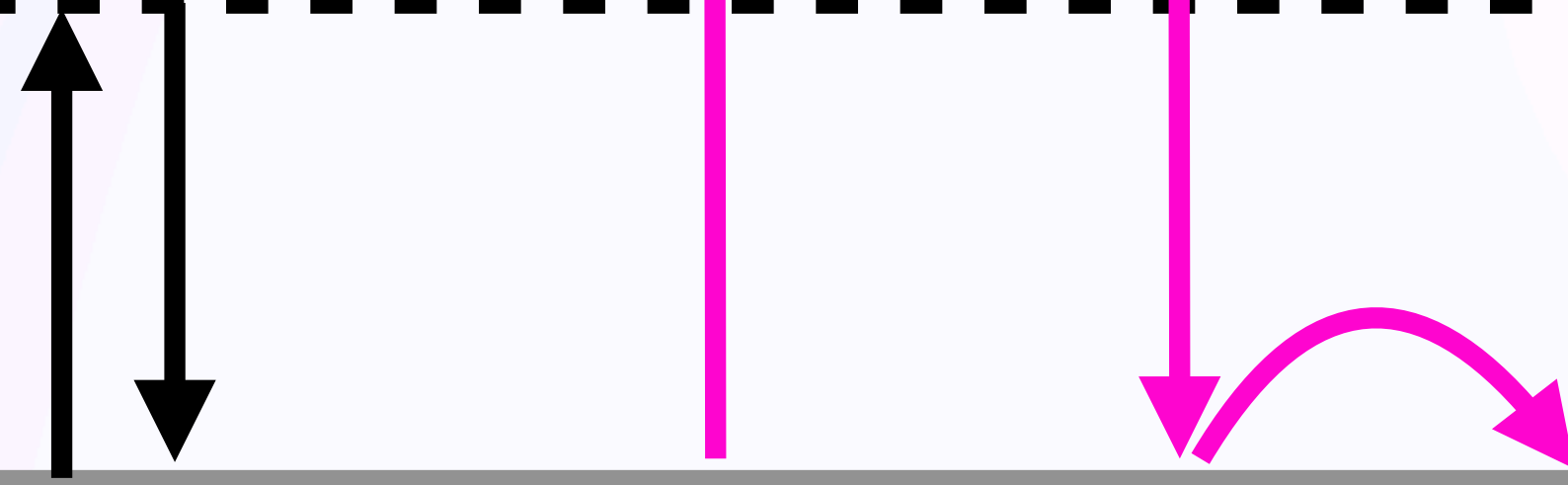
**Command! Query? Compute.**

Mutation Effect Stream

Query Effect Stream

Pure Function Stream

Base Event Stream



Compute Substrate

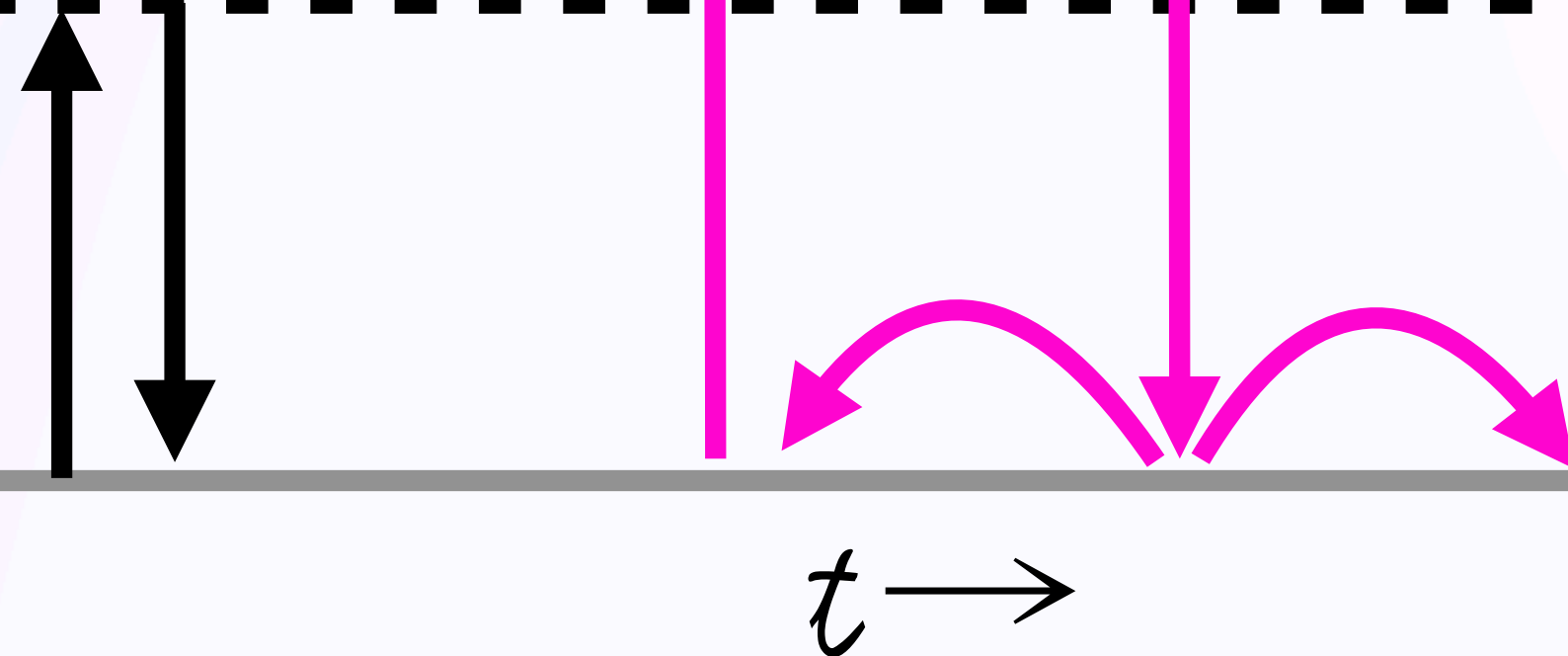
**Command! Query? Compute.**

Mutation Effect Stream

Query Effect Stream

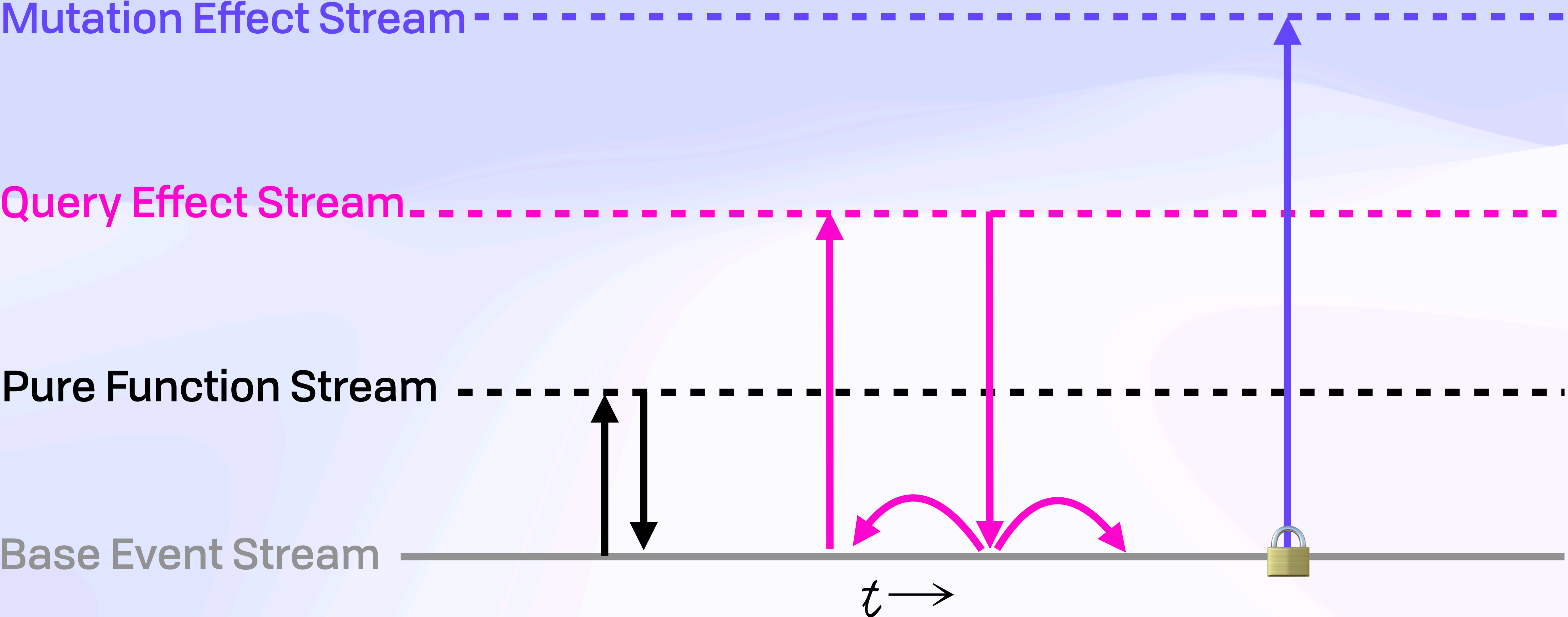
Pure Function Stream

Base Event Stream



# Compute Substrate

# *Command! Query? Compute.*



# Compute Substrate

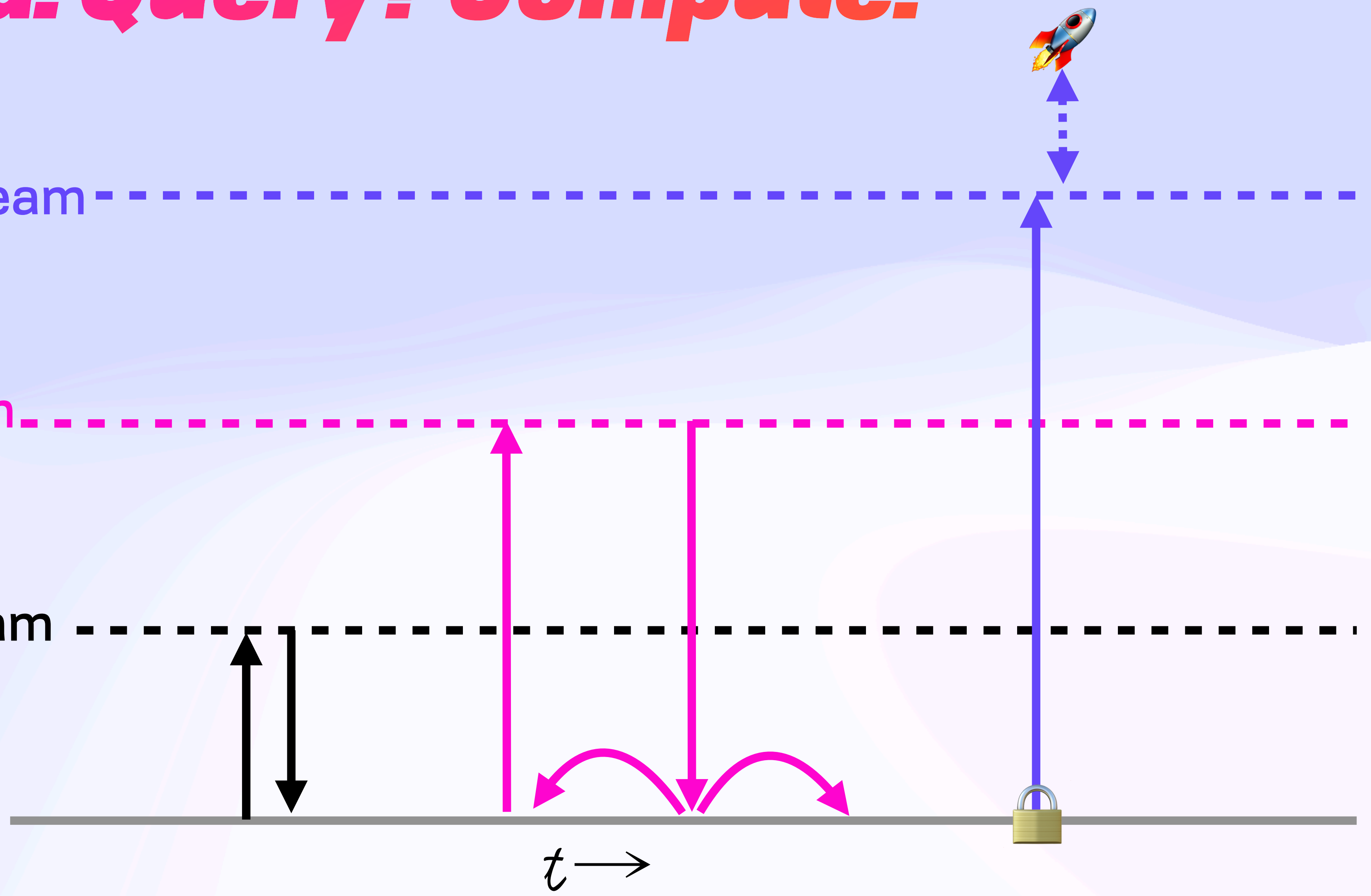
# *Command! Query? Compute.*

Mutation Effect Stream

Query Effect Stream

Pure Function Stream

Base Event Stream



Compute Substrate

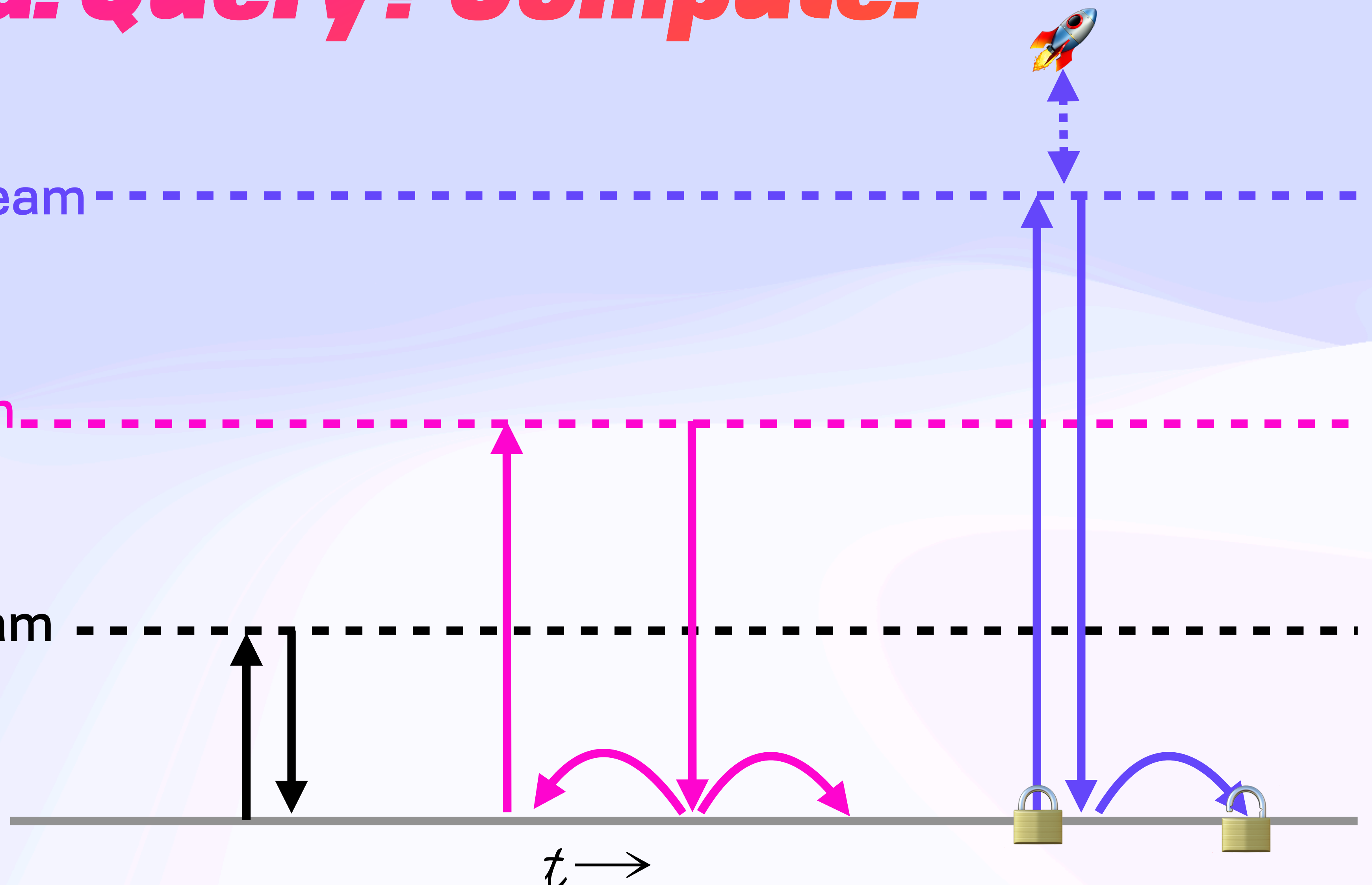
**Command! Query? Compute.**

Mutation Effect Stream

Query Effect Stream

Pure Function Stream

Base Event Stream



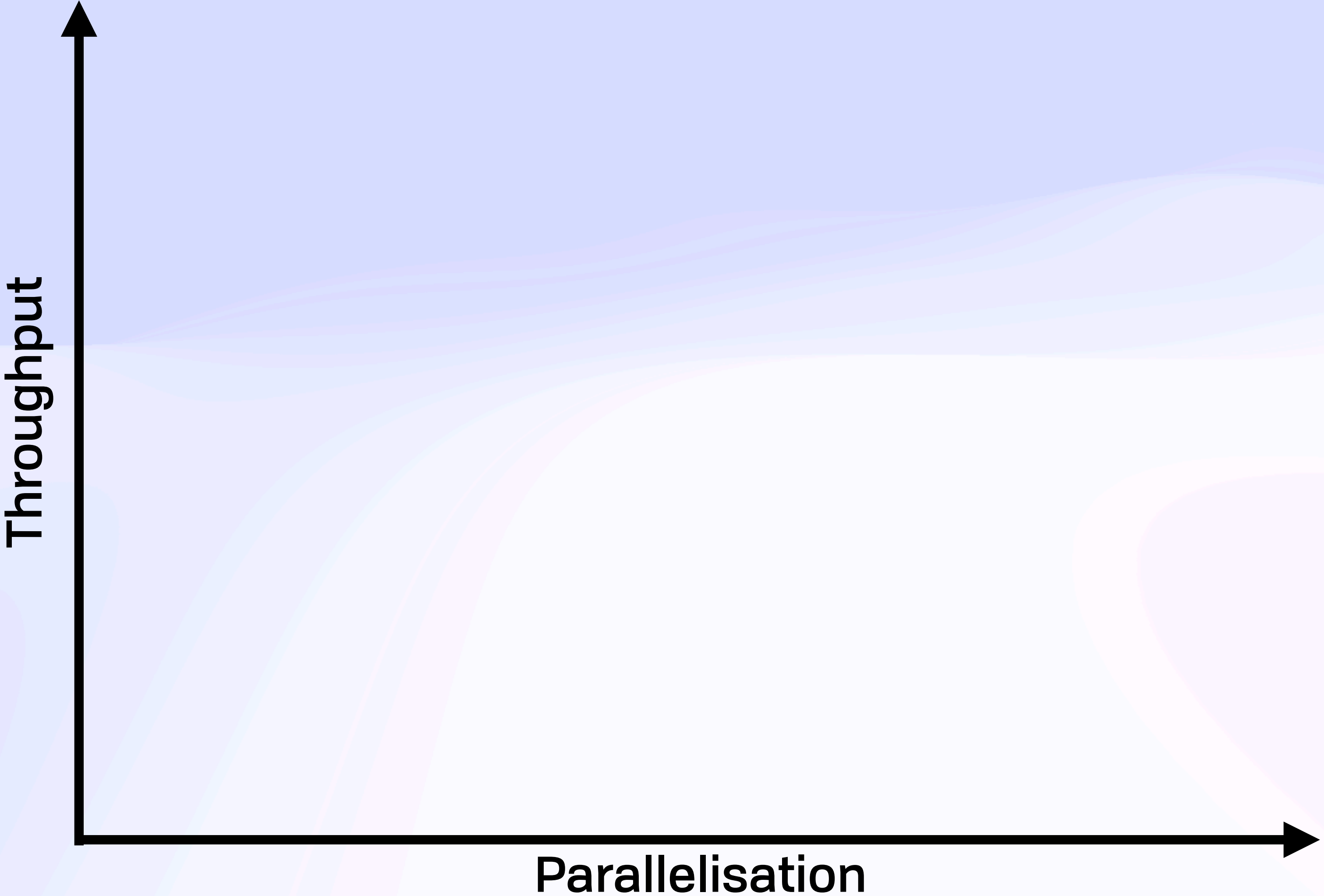
Compute Substrate

***With a Little Help From My Friends***



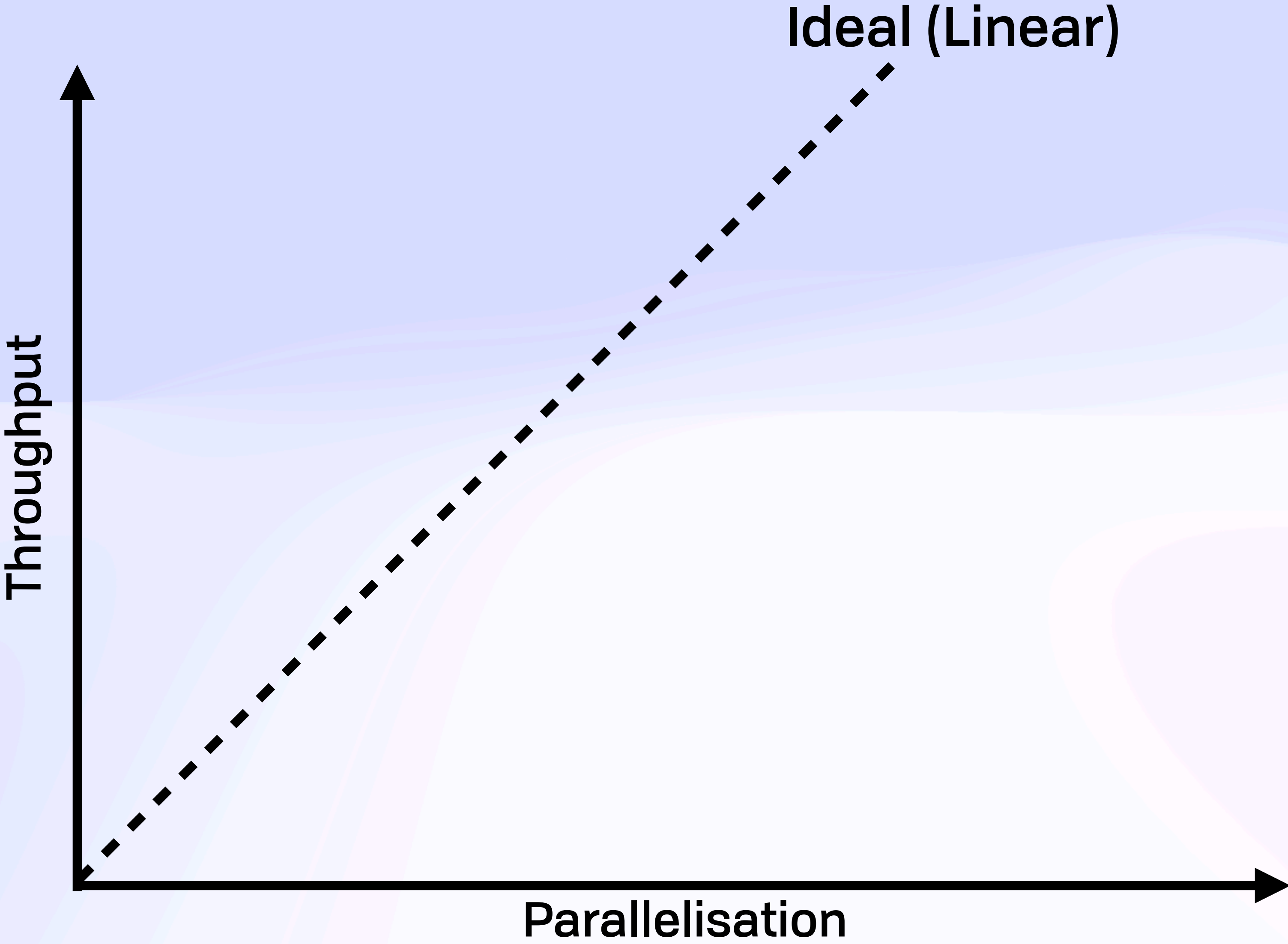
Compute Substrate

# *With a Little Help From My Friends*



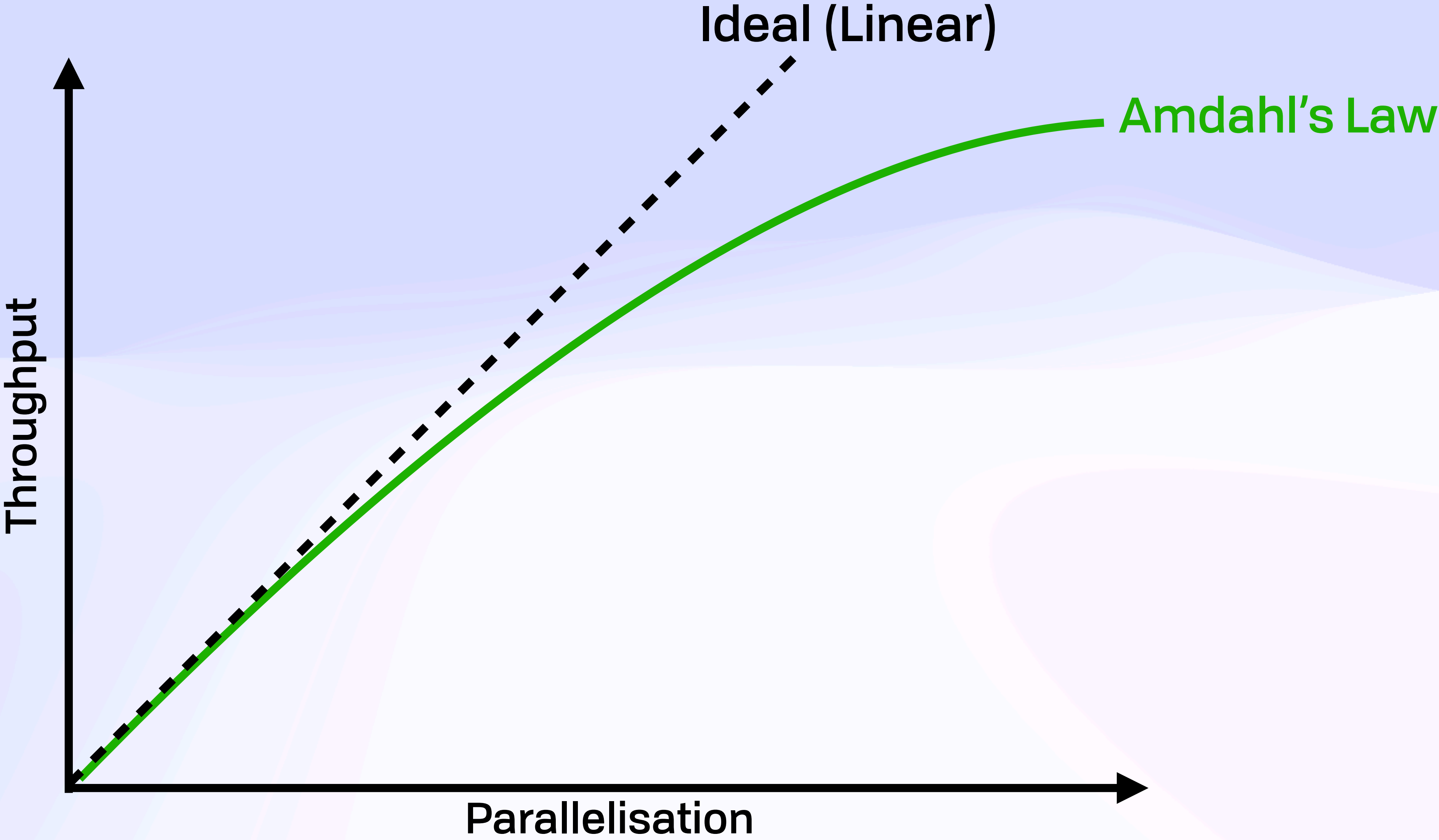
Compute Substrate

# *With a Little Help From My Friends*



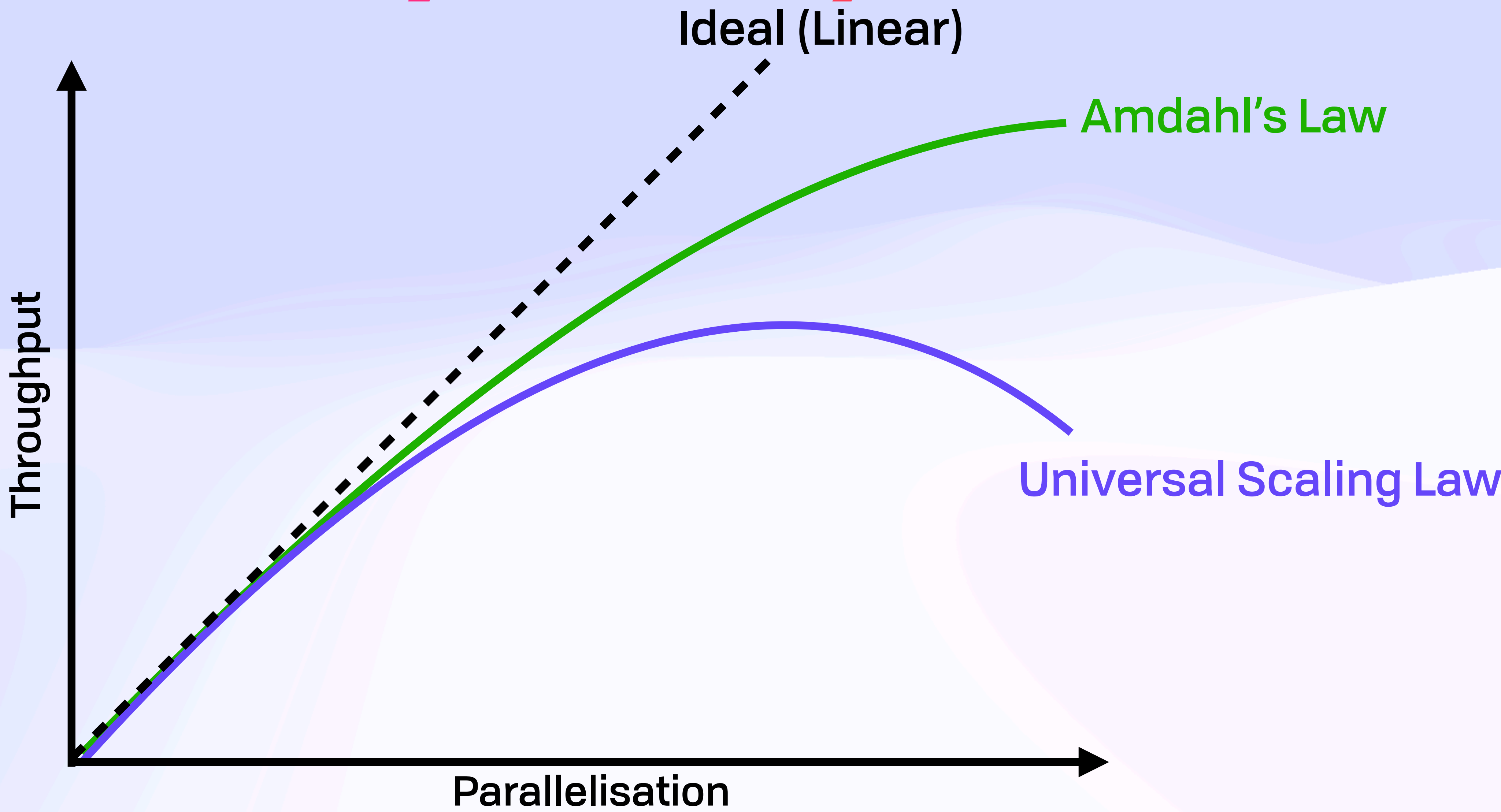
Compute Substrate

# *With a Little Help From My Friends*



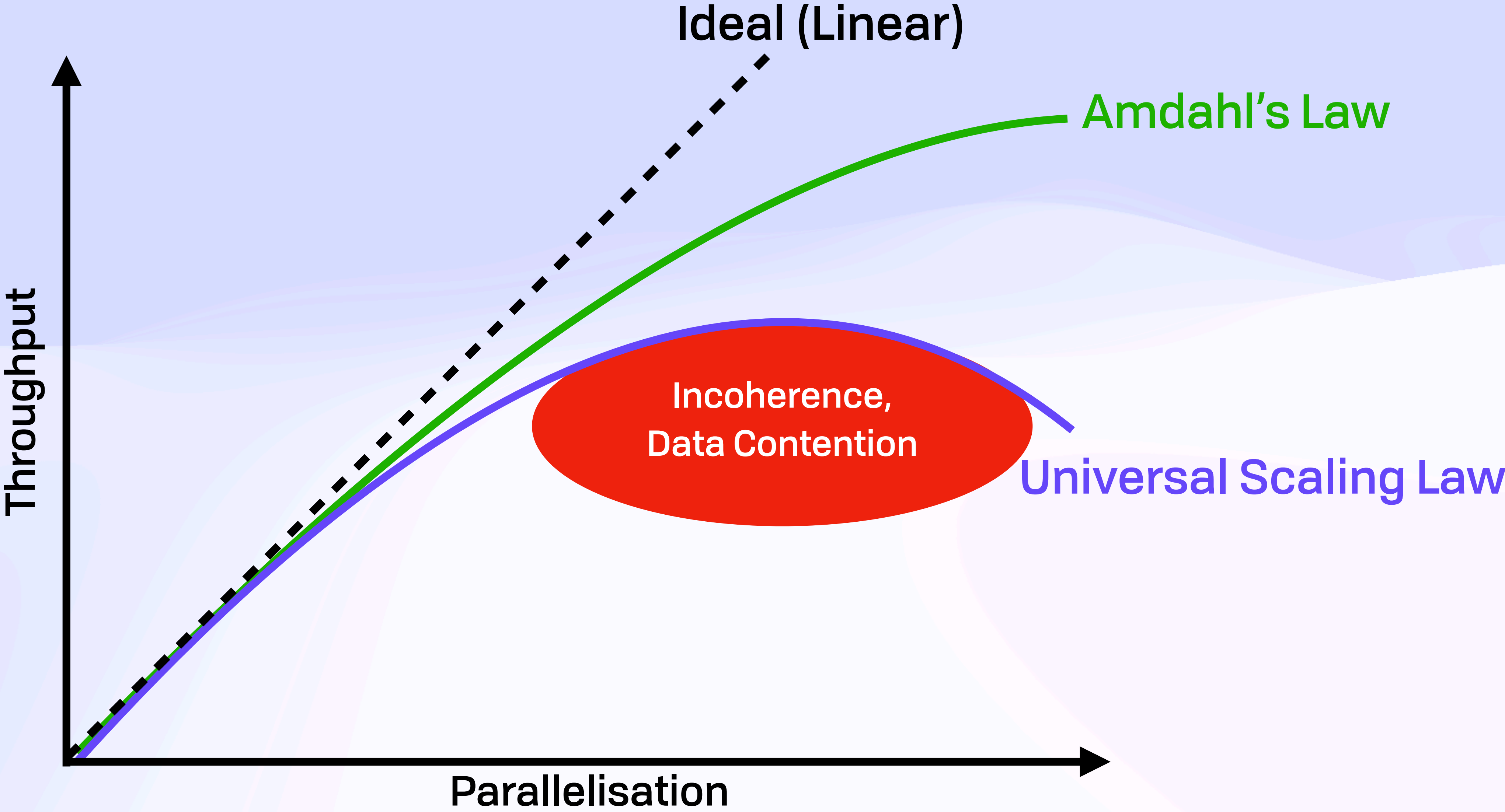
Compute Substrate

# *With a Little Help From My Friends*



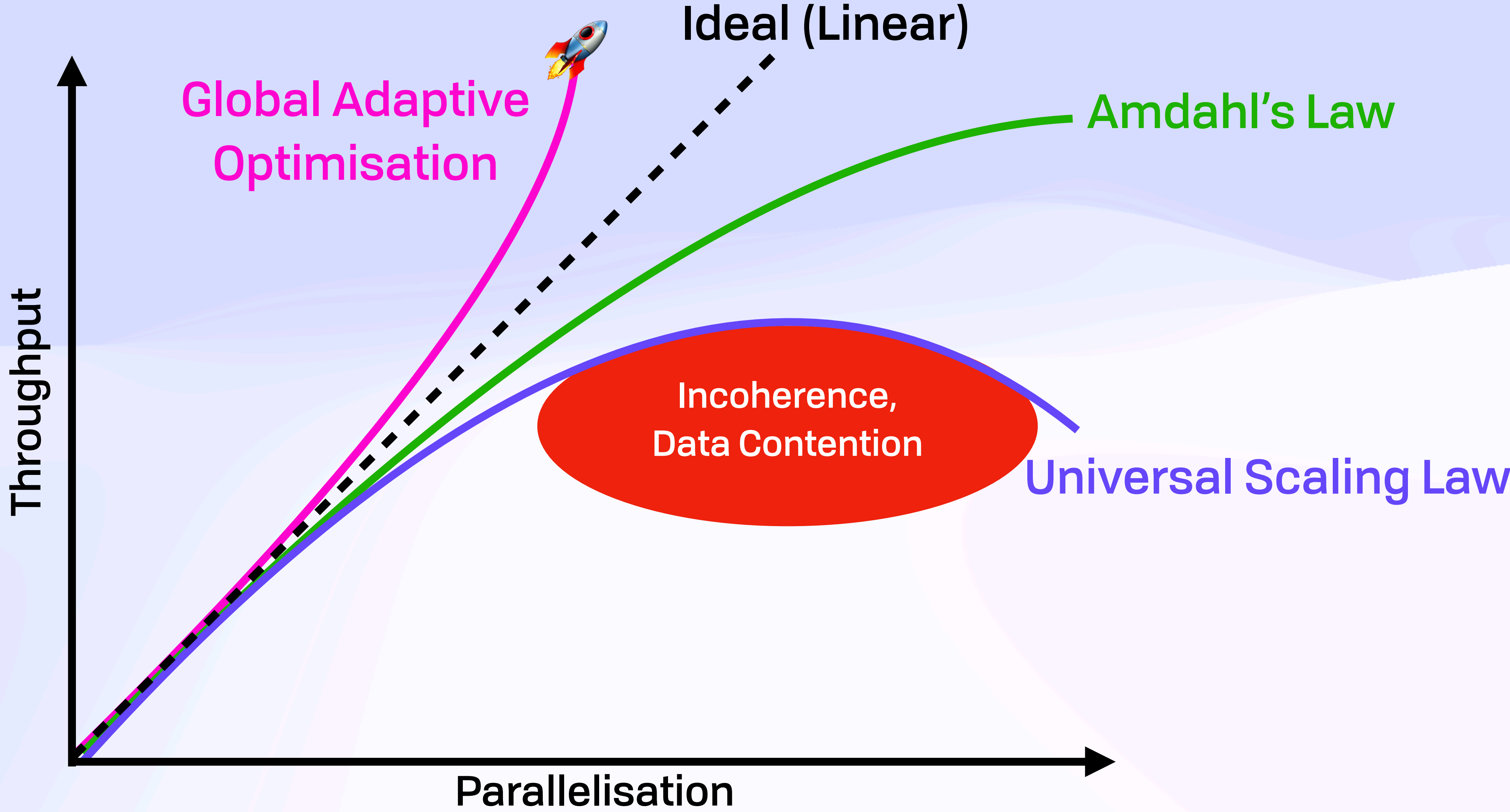
Compute Substrate

# *With a Little Help From My Friends*



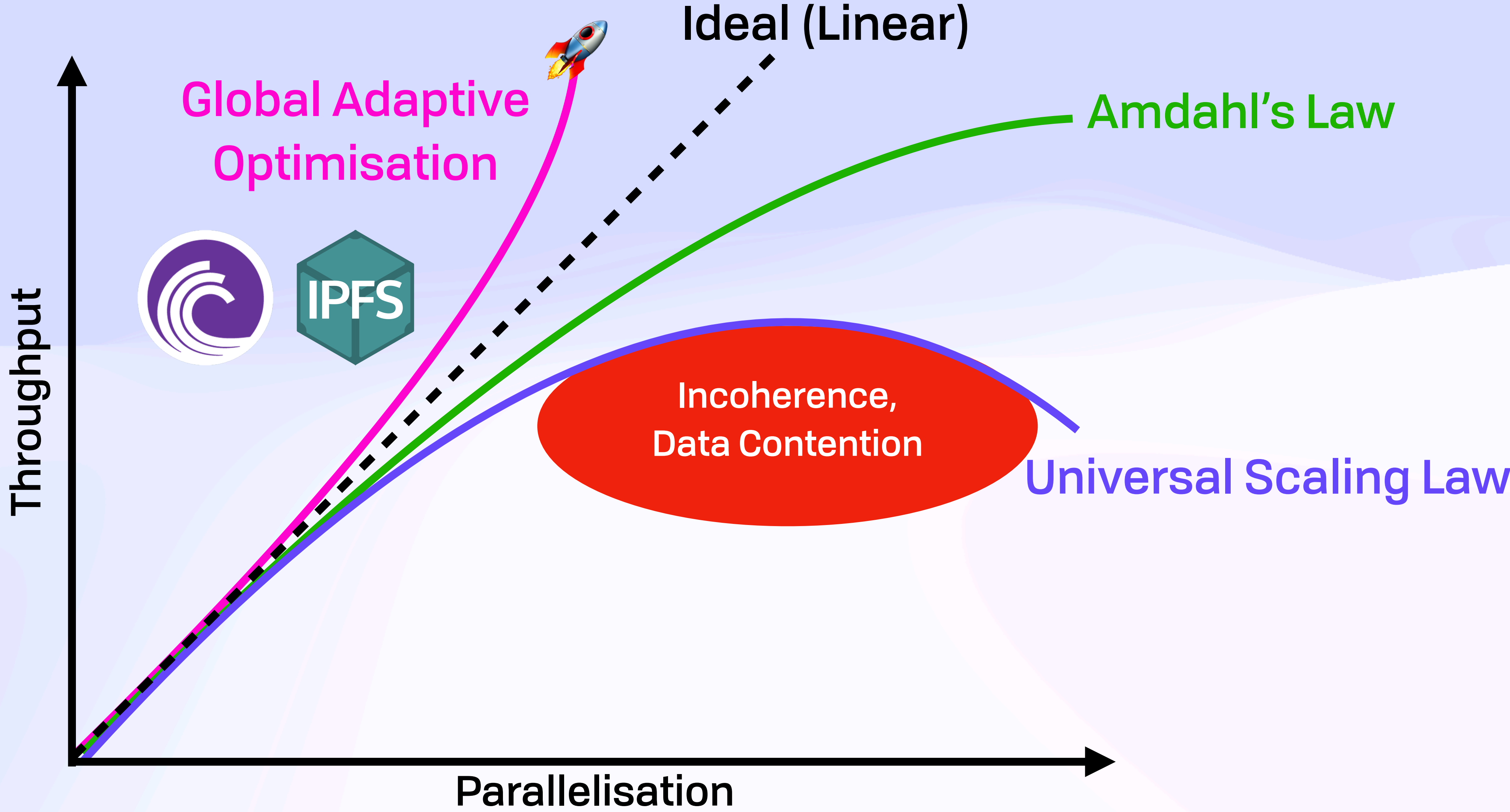
Compute Substrate

# *With a Little Help From My Friends*



# Compute Substrate

# *With a Little Help From My Friends*



Compute Substrate

# *Run Once, And Never Again*





Compute Substrate

# *Run Once, And Never Again*



◆ **Input Hash → Cached Output**

Compute Substrate

# *Run Once, And Never Again*



- ◆ **Input Hash → Cached Output**
- ◆ "Instant" AI

Compute Substrate

# *Run Once, And Never Again*



- ◆ **Input Hash → Cached Output**
  - ◆ "Instant" AI
    - ◆ e.g. moderation, tagging

Compute Substrate

# *Run Once, And Never Again*



- ◆ **Input Hash → Cached Output**
  - ◆ "Instant" AI
    - ◆ e.g. moderation, tagging
  - ◆ EigenTrust

Compute Substrate

# *Run Once, And Never Again*



- ◆ **Input Hash → Cached Output**
  - ◆ "Instant" AI
    - ◆ e.g. moderation, tagging
  - ◆ EigenTrust
  - ◆ Resizing, thumbs, cropping

Compute Substrate

# *Run Once, And Never Again*



- ◆ **Input Hash → Cached Output**
  - ◆ "Instant" AI
    - ◆ e.g. moderation, tagging
  - ◆ EigenTrust
  - ◆ Resizing, thumbs, cropping
  - ◆ Durable execution **everywhere**

Wrapping Up

***Where Do We Go From Here?***



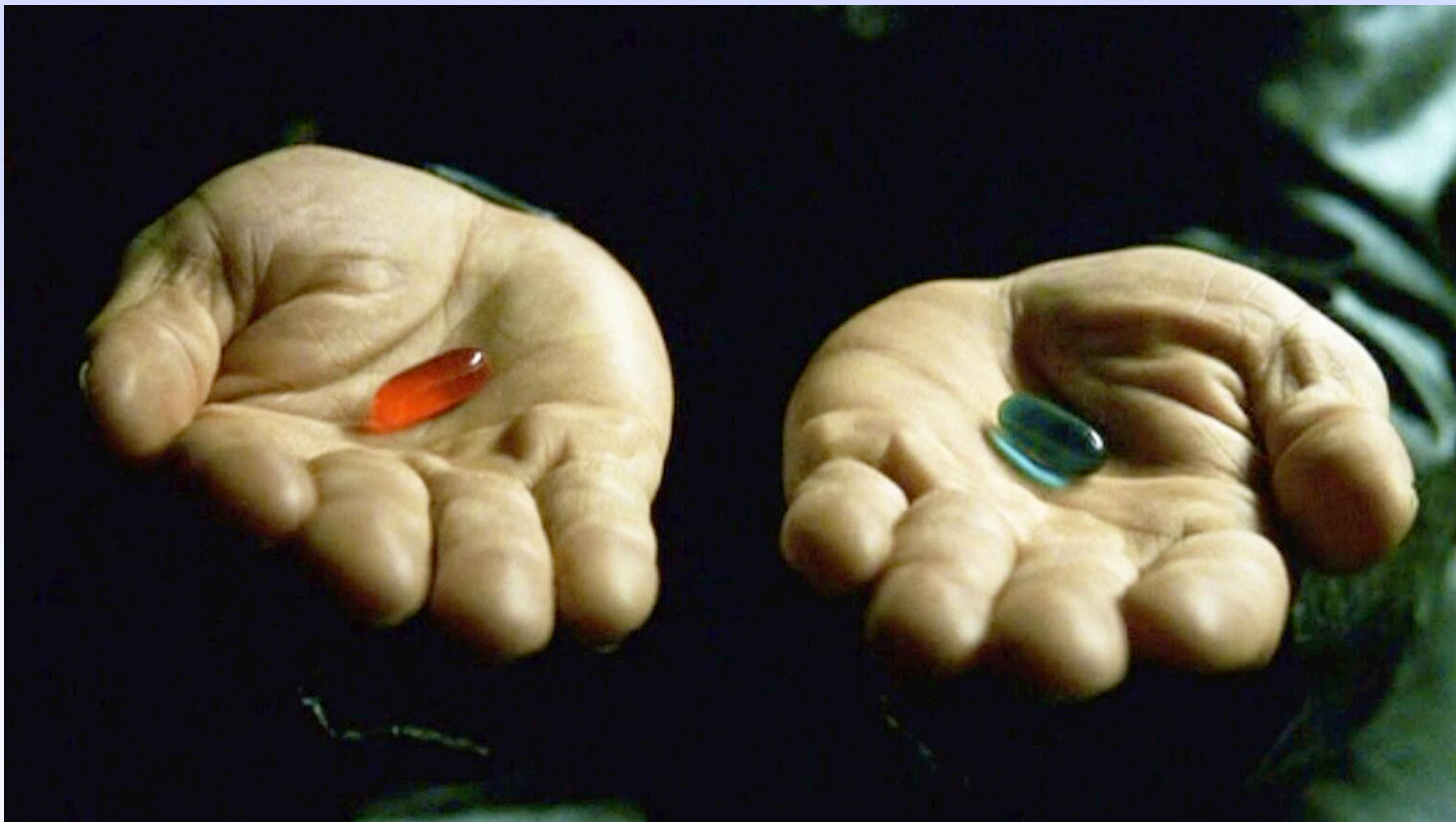
Where Do We Go From Here?

***There Is No ~~Spoon~~ Back End***



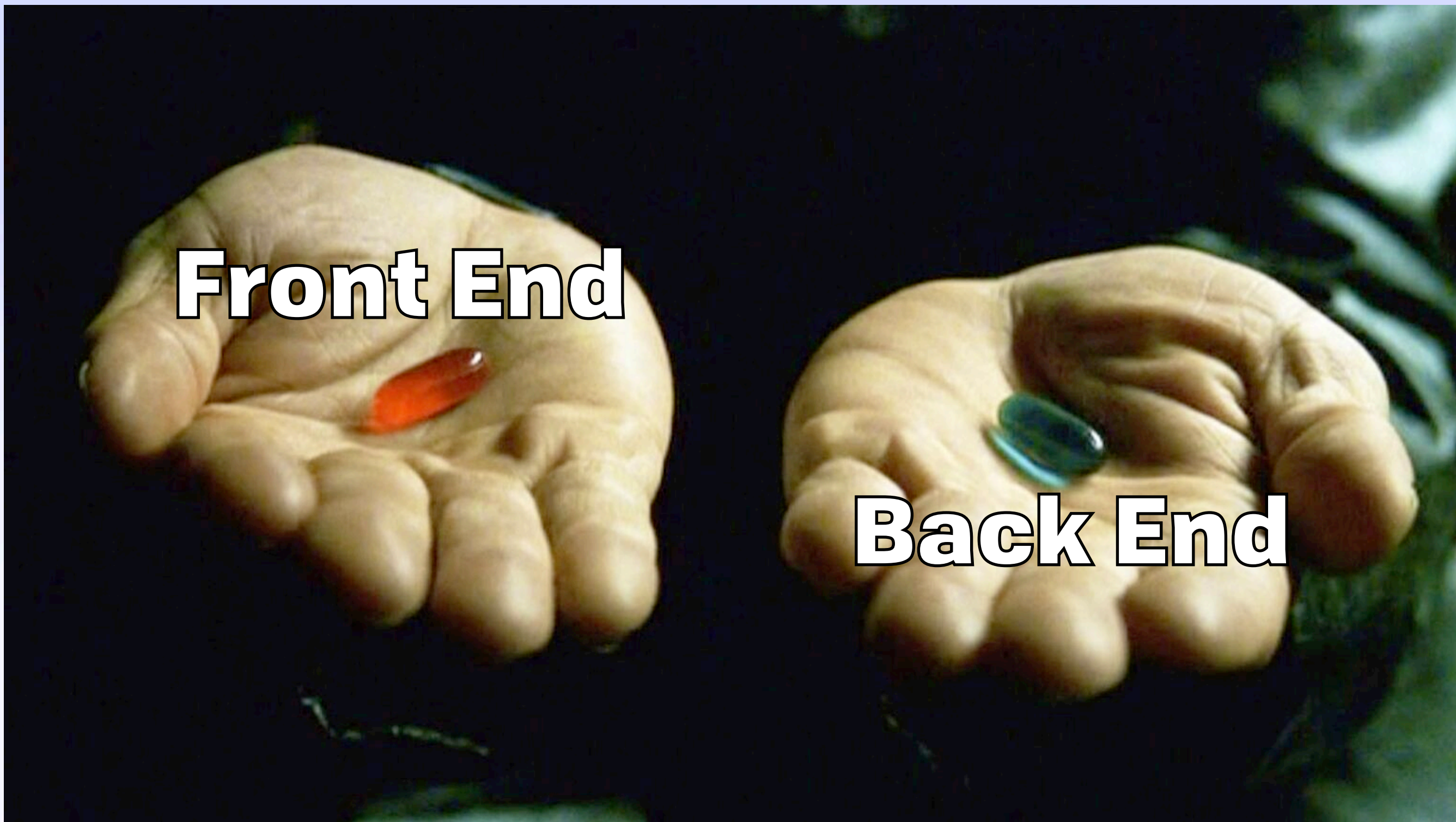
Where Do We Go From Here?

***There Is No ~~Spoon~~ Back End***



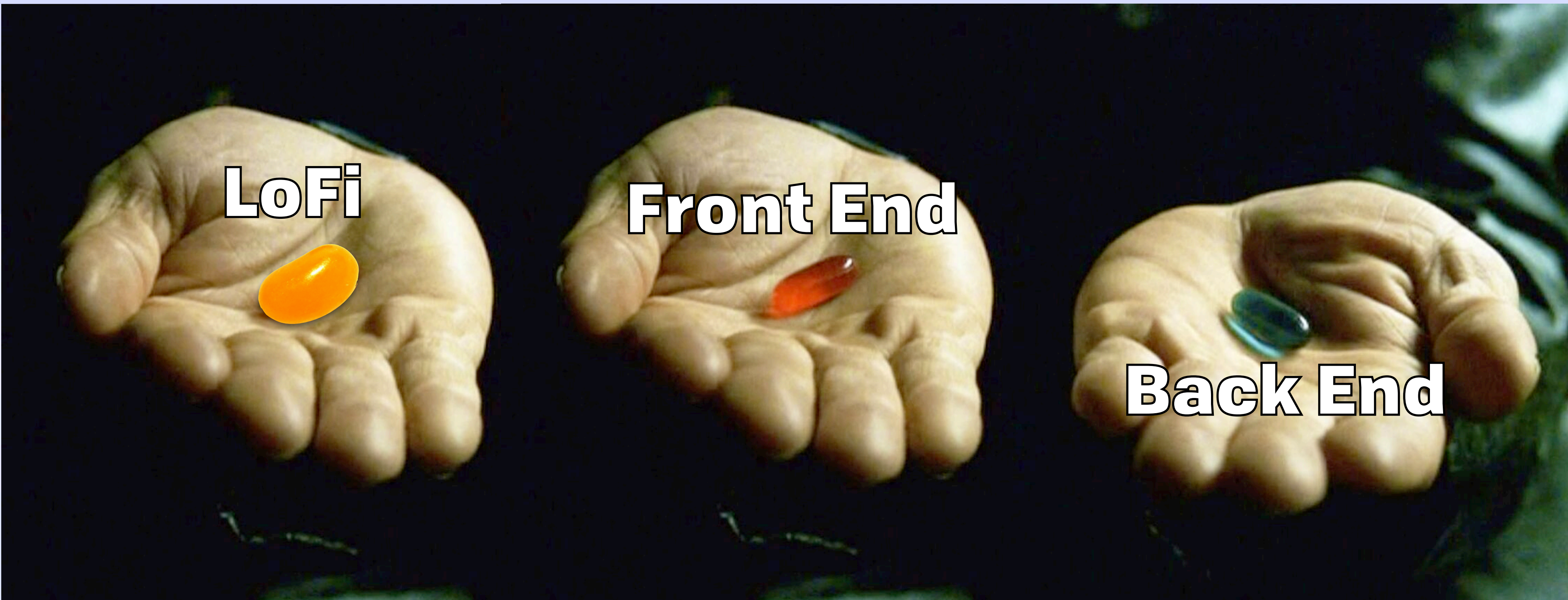
Where Do We Go From Here?

***There Is No ~~Spoon~~ Back End***



Where Do We Go From Here?

***There Is No ~~Spoon~~ Back End***



Where Do We Go From Here?

***New Constraints = New Design Space***

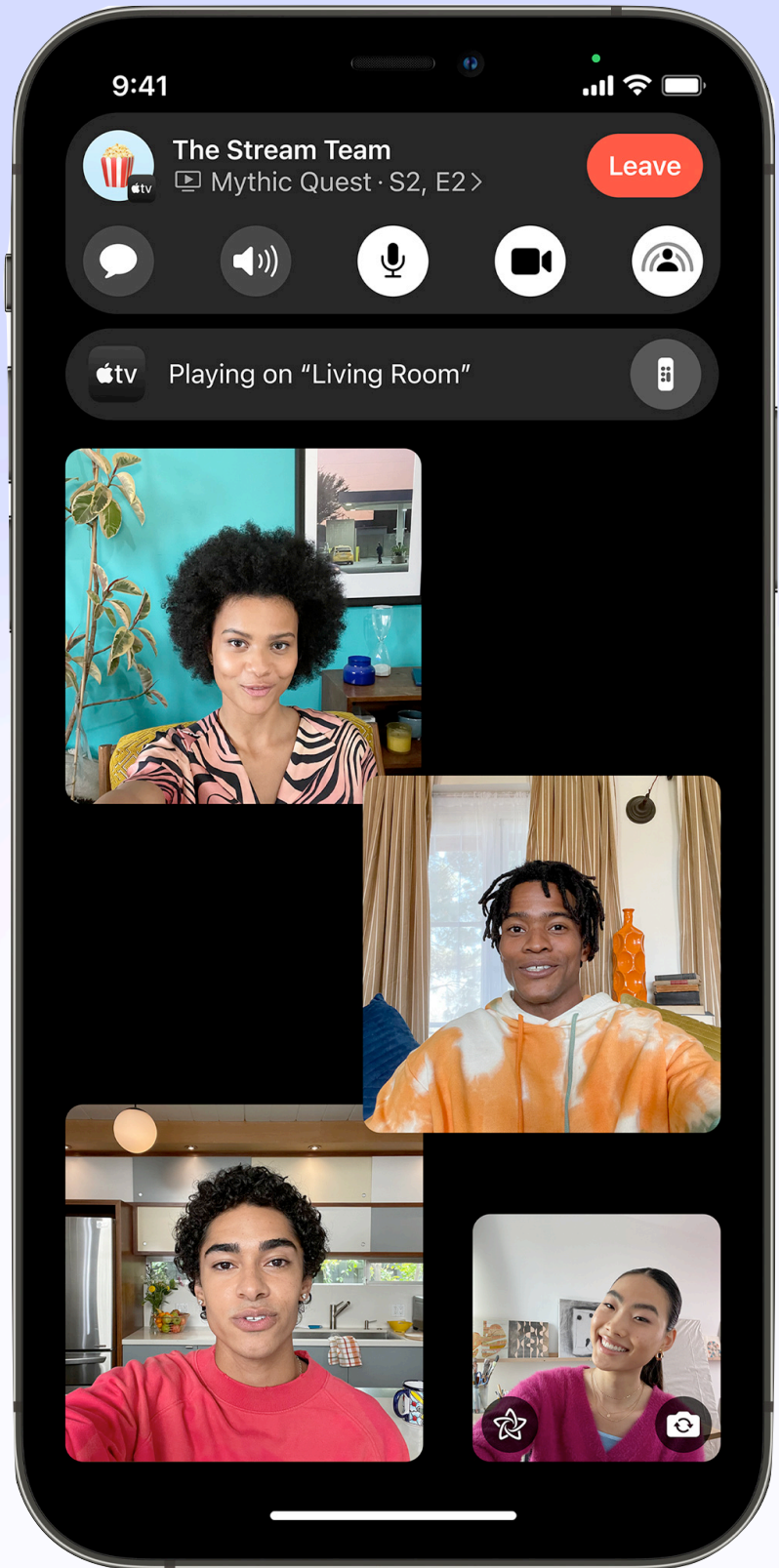
Where Do We Go From Here?

# *New Constraints = New Design Space*



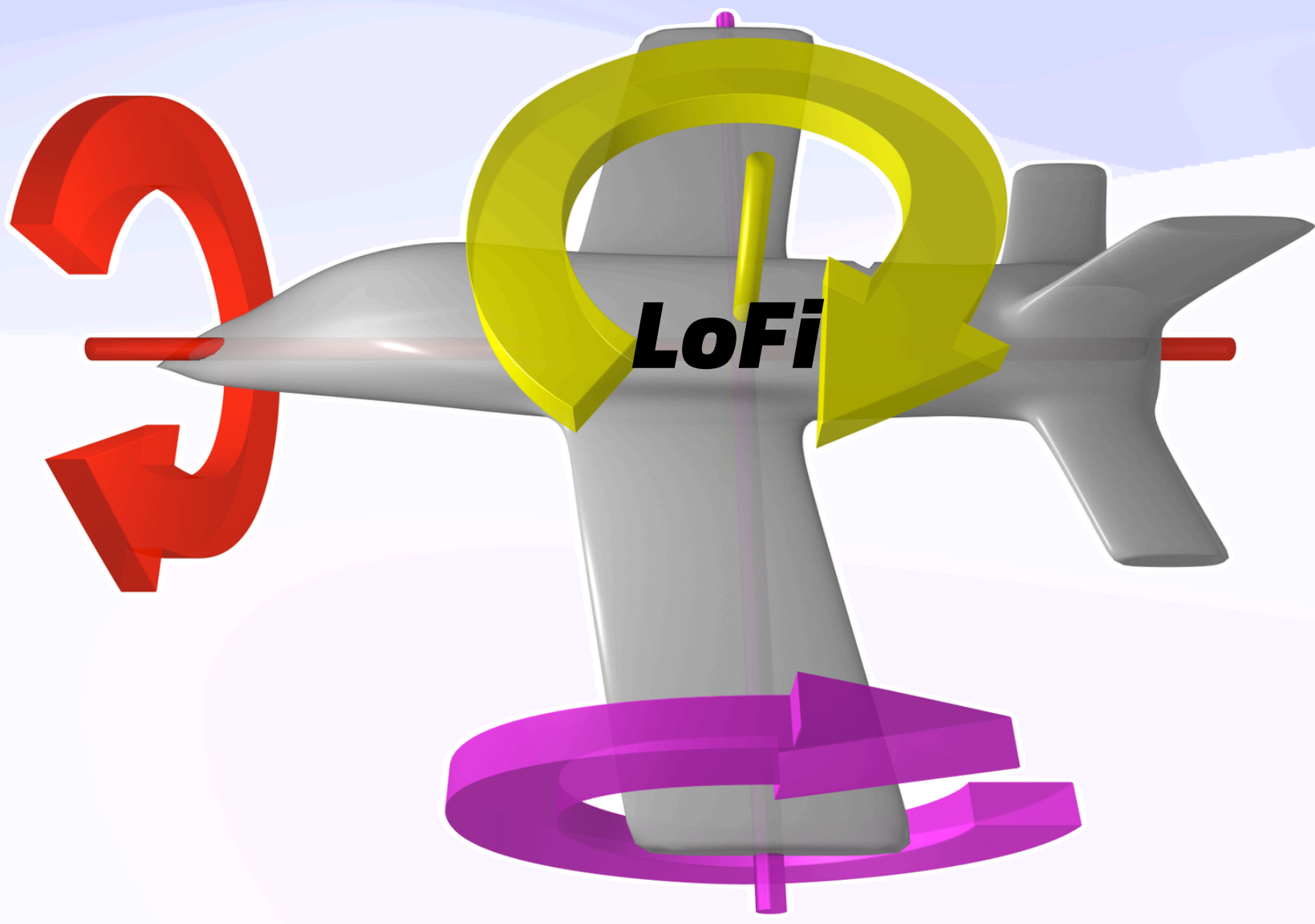
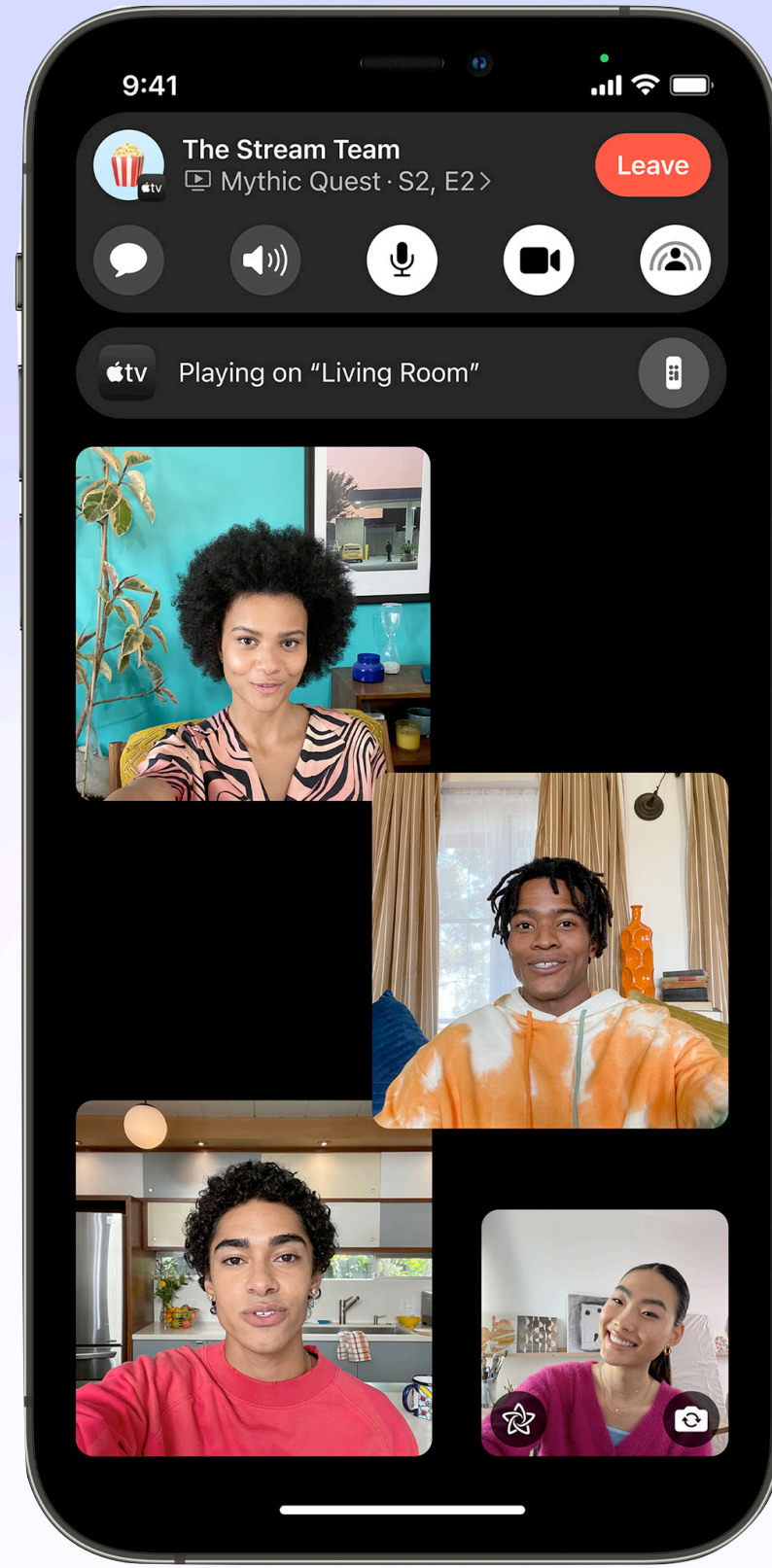
Where Do We Go From Here?

# *New Constraints = New Design Space*



Where Do We Go From Here?

# *New Constraints = New Design Space*



Where Do We Go From Here?  
***Not All or Nothing***

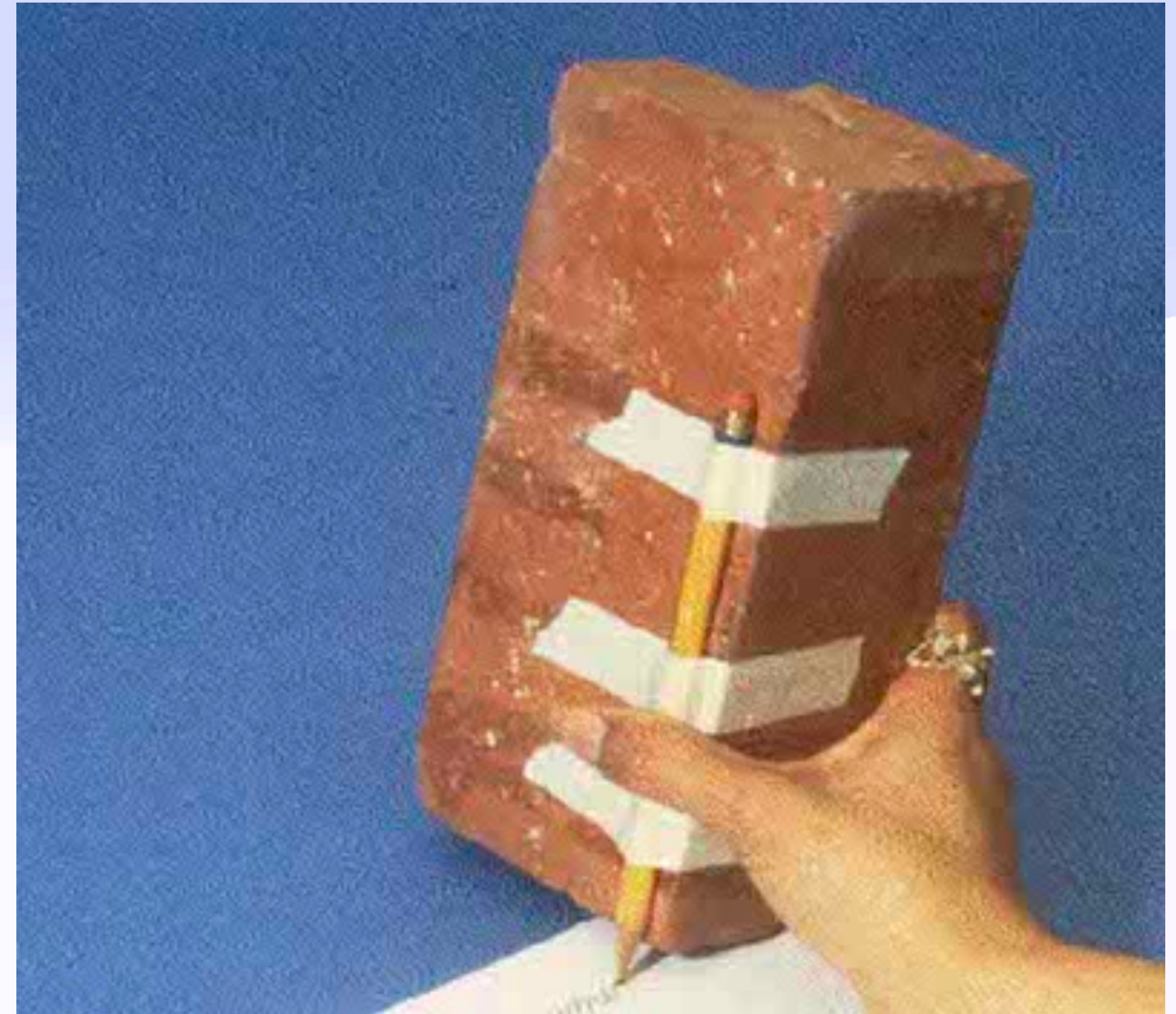




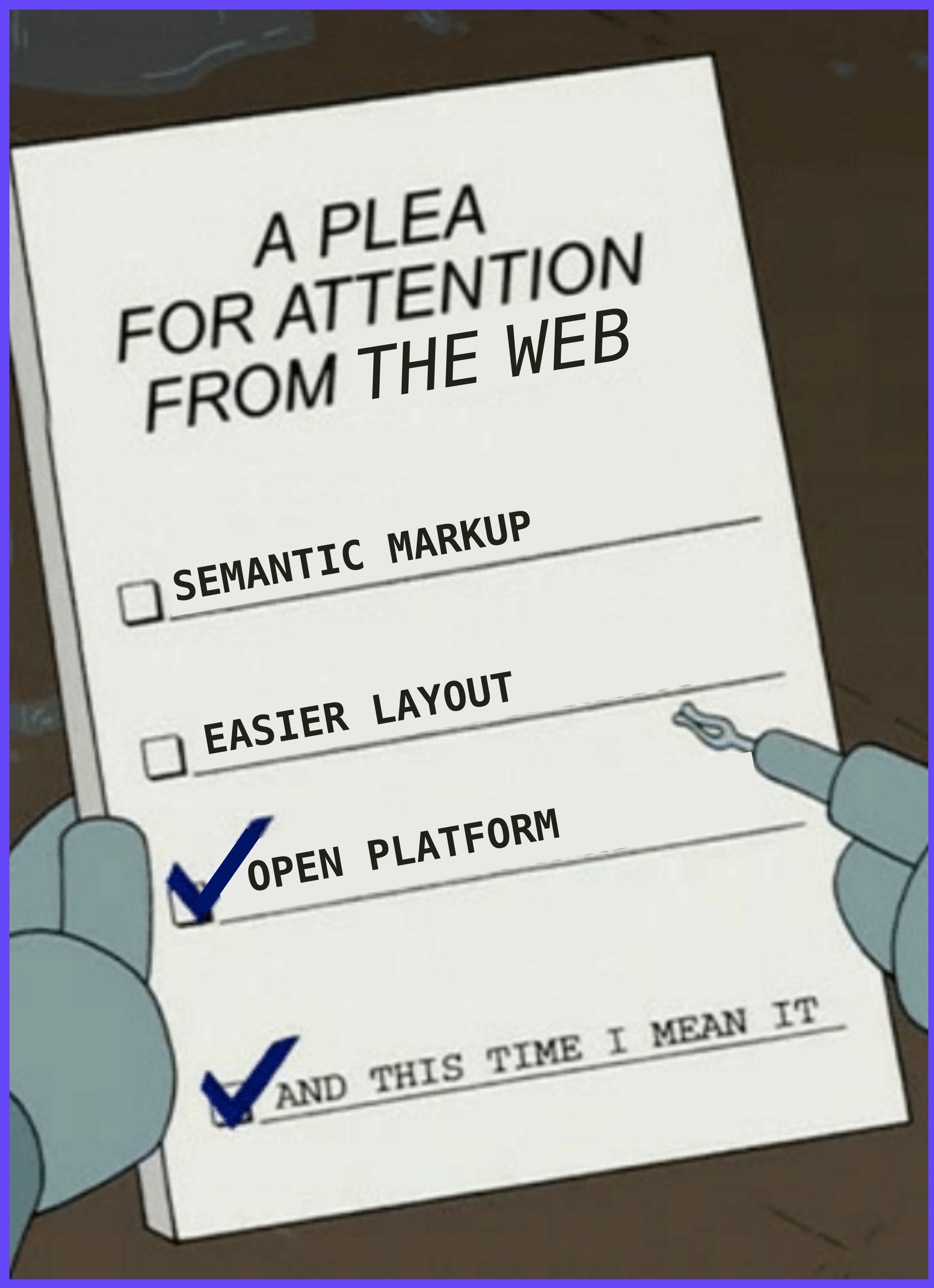
# Where Do We Go From Here?

## ***Not All or Nothing***

- ◆ LoFi is great at human sizes & speeds
- ◆ ...but harder to do global things
  - ◆ e.g. "Build the Bluesky firehose"



# A Roadmap Wrap Up



automerge.org



Automerge



inkandswitch.com/bee hive

github.com/ucan-wg



**Thank You, Chicago**



 @expede@types.pl

 bsky.app/profile/expede.wtf

 hello@brooklynzelenka.com