



Better Together



IPVM For an Open World



github.com/ipvm-wg
fission.codes

IPVM: Better Together



IPVM: Better Together

IPVM: Better Together

Jesper, I have this idea in which we'll connect all of the worlds Erlang systems to each other, imagine if **every process could talk to every other process**, world-wide!

– **Joe Armstrong**, email to Jesper L. Andersen

IPVM: Better Together


Brooklyn Zelenka, @expede



github.com/expede

IPVM: Better Together

Brooklyn Zelenka, @expede


- ◆ Cofounder & CTO at Fission 
 - ◆ discord.gg/fissioncodes
 - ◆ @fission@plnetwork.xyz
- ◆ IPVM Spec Wrangler — github.com/ipvm-wg



github.com/expede

IPVM: Better Together

Brooklyn Zelenka, @expede

- ◆ Cofounder & CTO at Fission 
 - ◆ discord.gg/fissioncodes
 - ◆ @fission@plnetwork.xyz
- ◆ IPVM Spec Wrangler — github.com/ipvm-wg



github.com/expede

IPVM: Better Together

 ***Brought To You By...***

IPVM: Better Together

 ***Brought To You By...***

IPVM Working Group

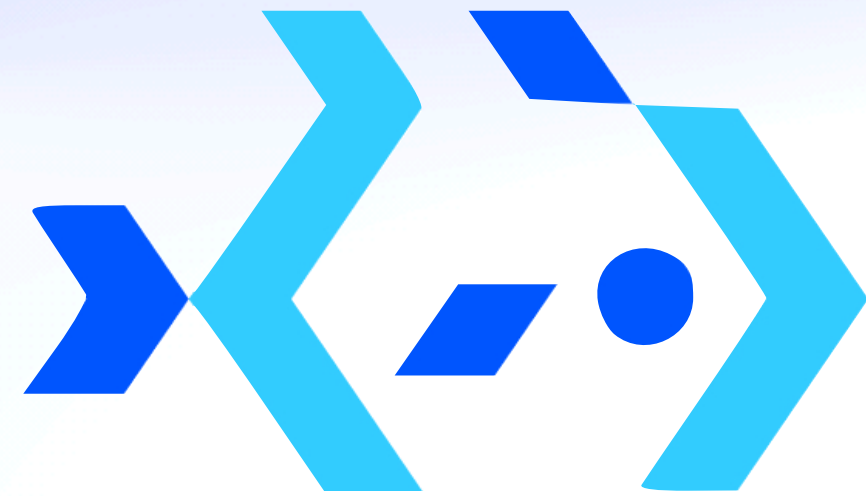
Working Group for the Interplanetary Virtual Machine

IPVM: Better Together

 ***Brought To You By...***

IPVM Working Group

Working Group for the Interplanetary Virtual Machine

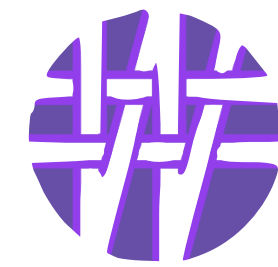
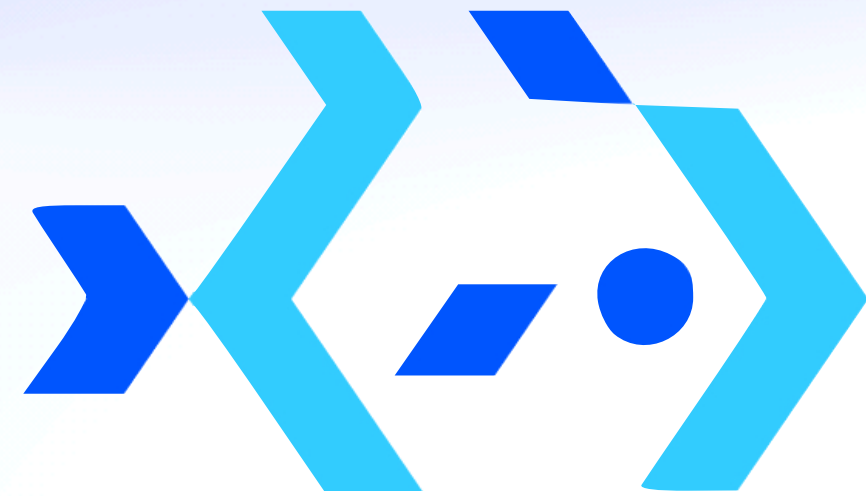


IPVM: Better Together

 ***Brought To You By...***

IPVM Working Group

Working Group for the Interplanetary Virtual Machine

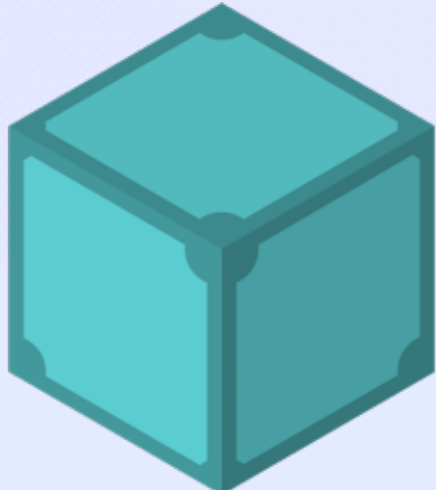


IPVM: Better Together

Consistency & Distribution

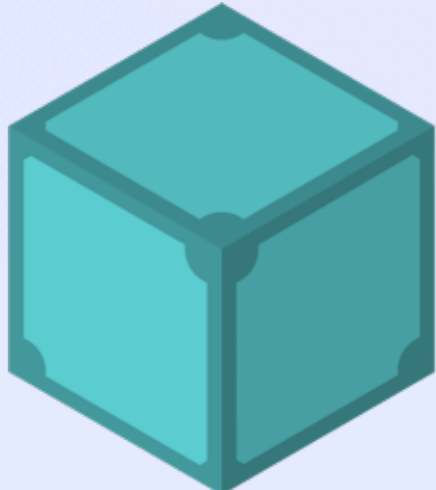
IPVM: Better Together

Consistency & Distribution



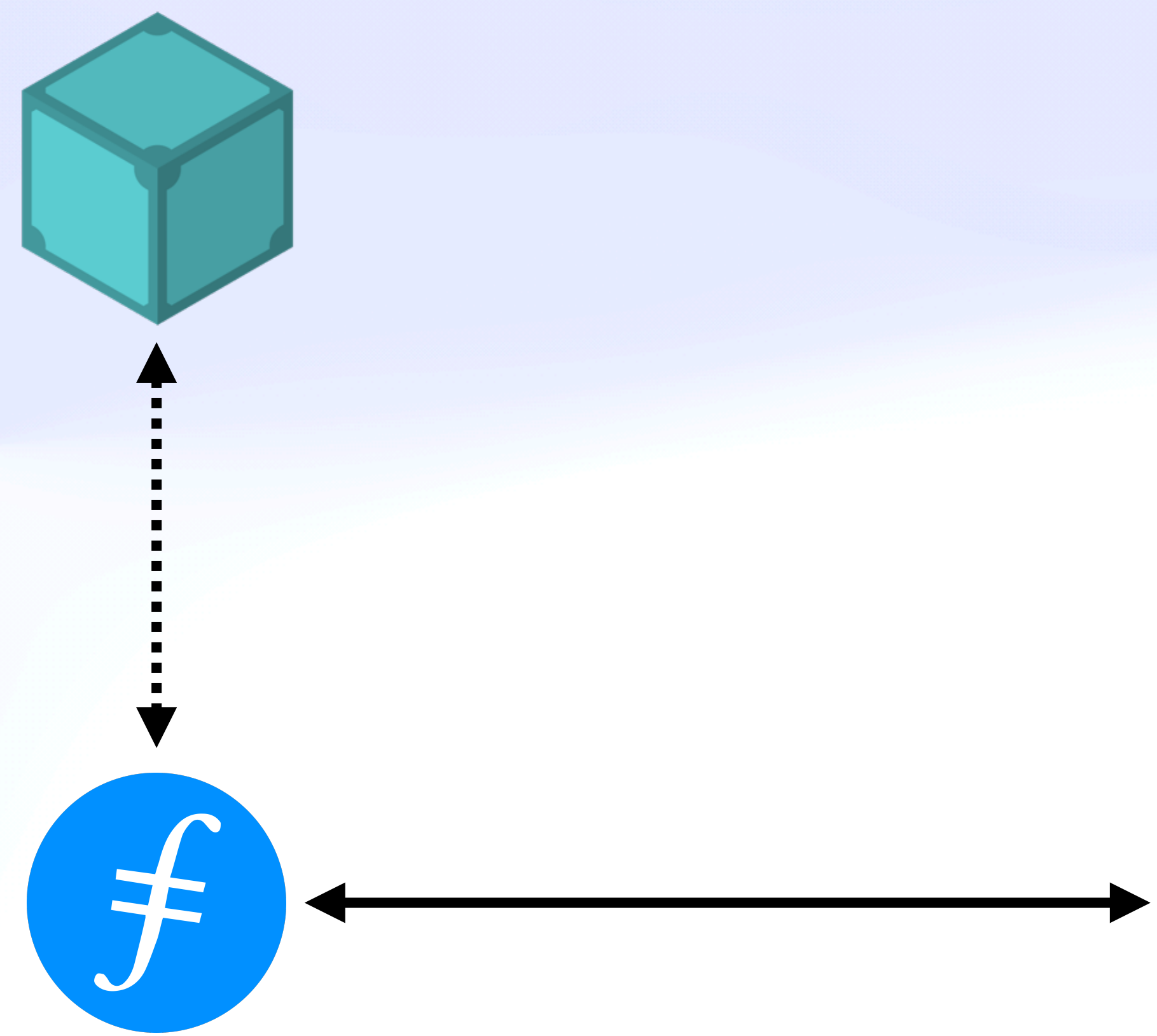
IPVM: Better Together

Consistency & Distribution



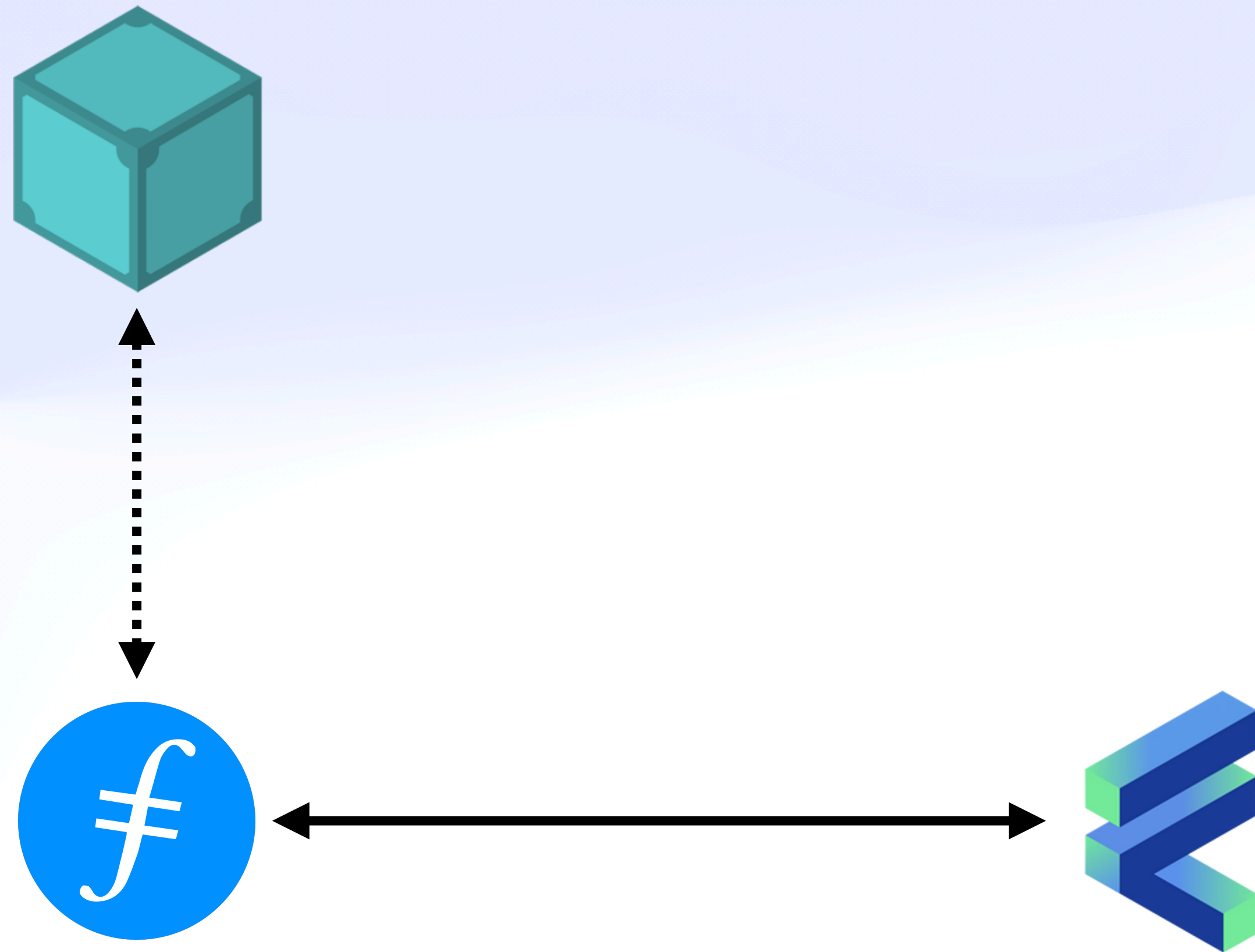
IPVM: Better Together

Consistency & Distribution



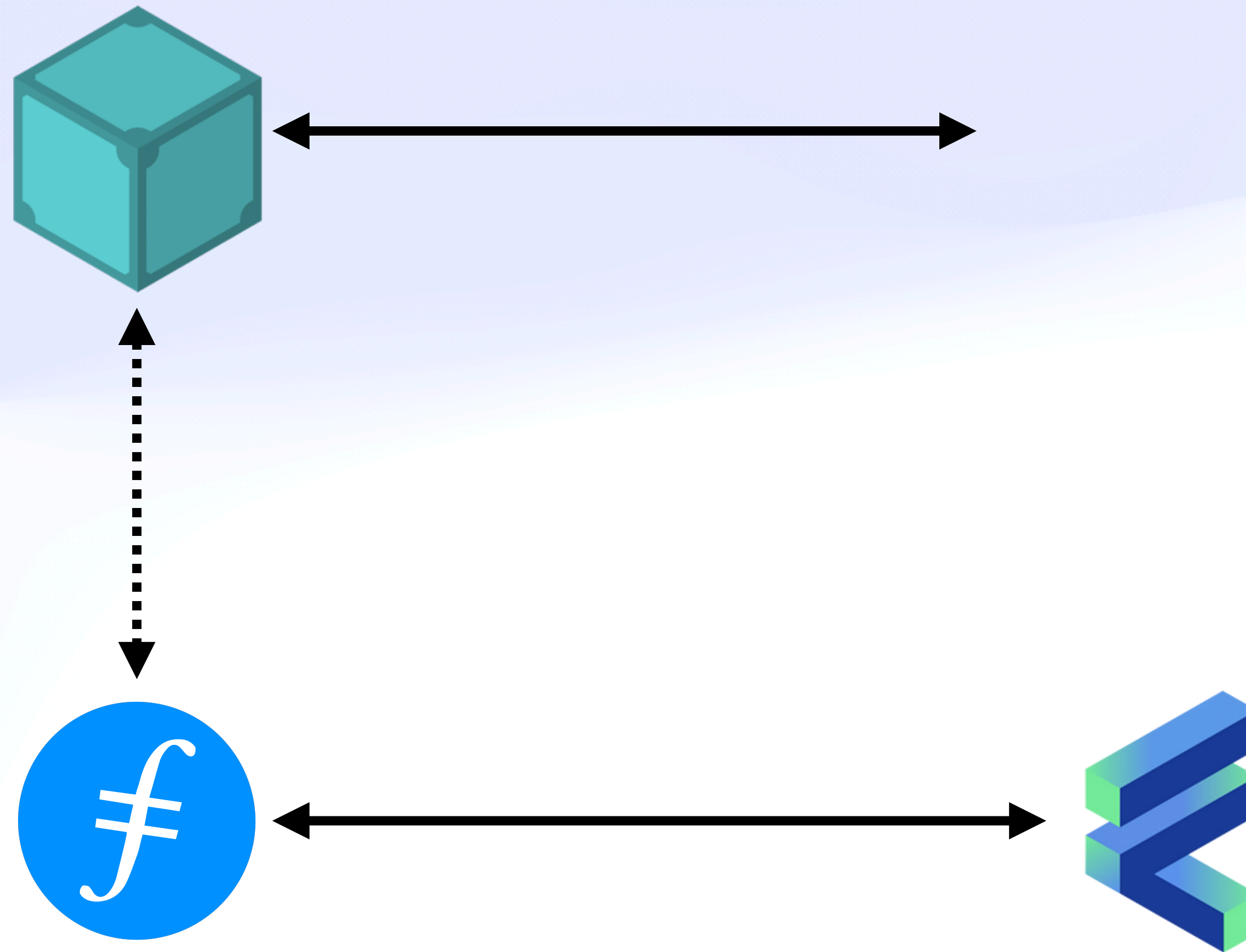
IPVM: Better Together

Consistency & Distribution



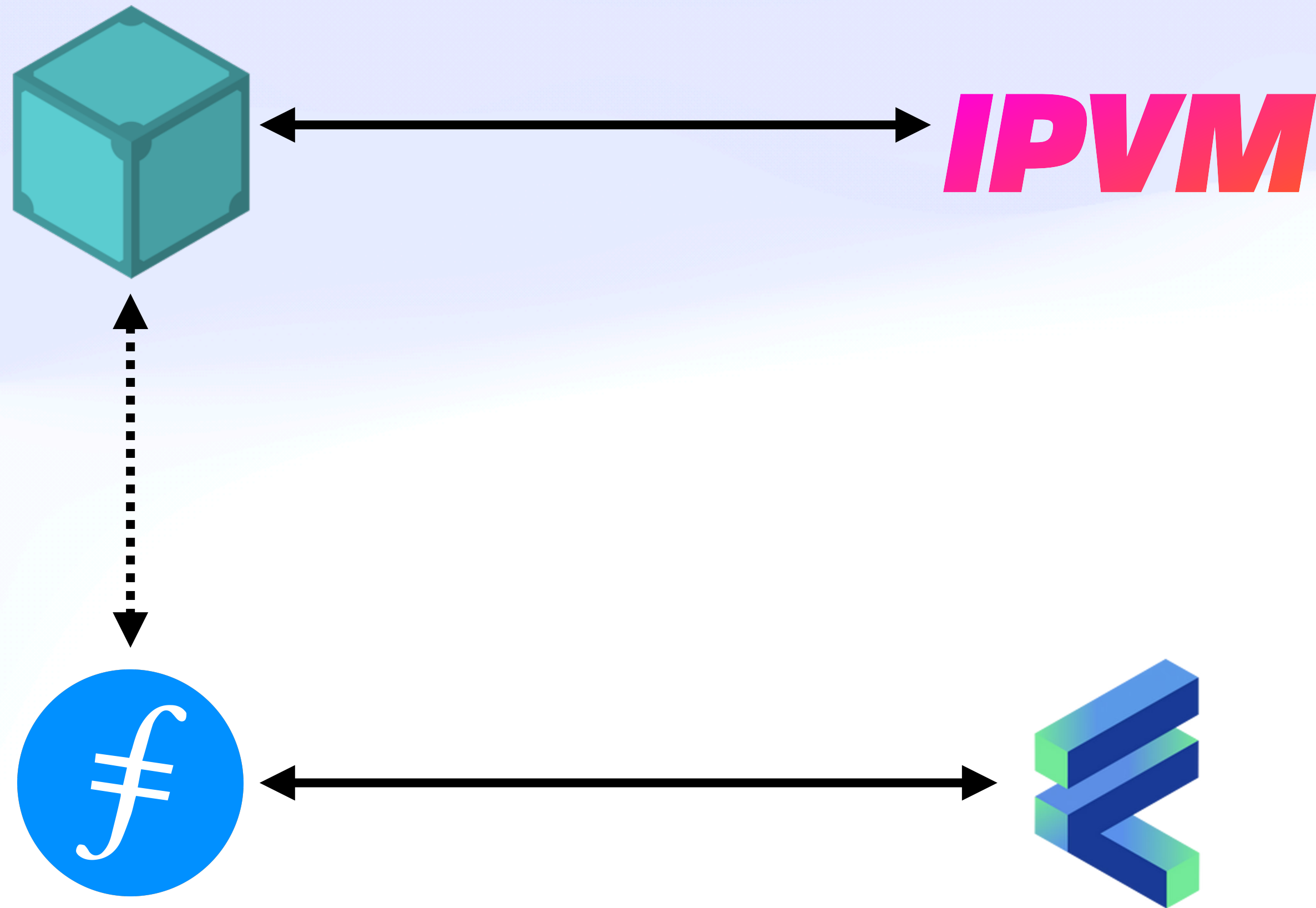
IPVM: Better Together

Consistency & Distribution



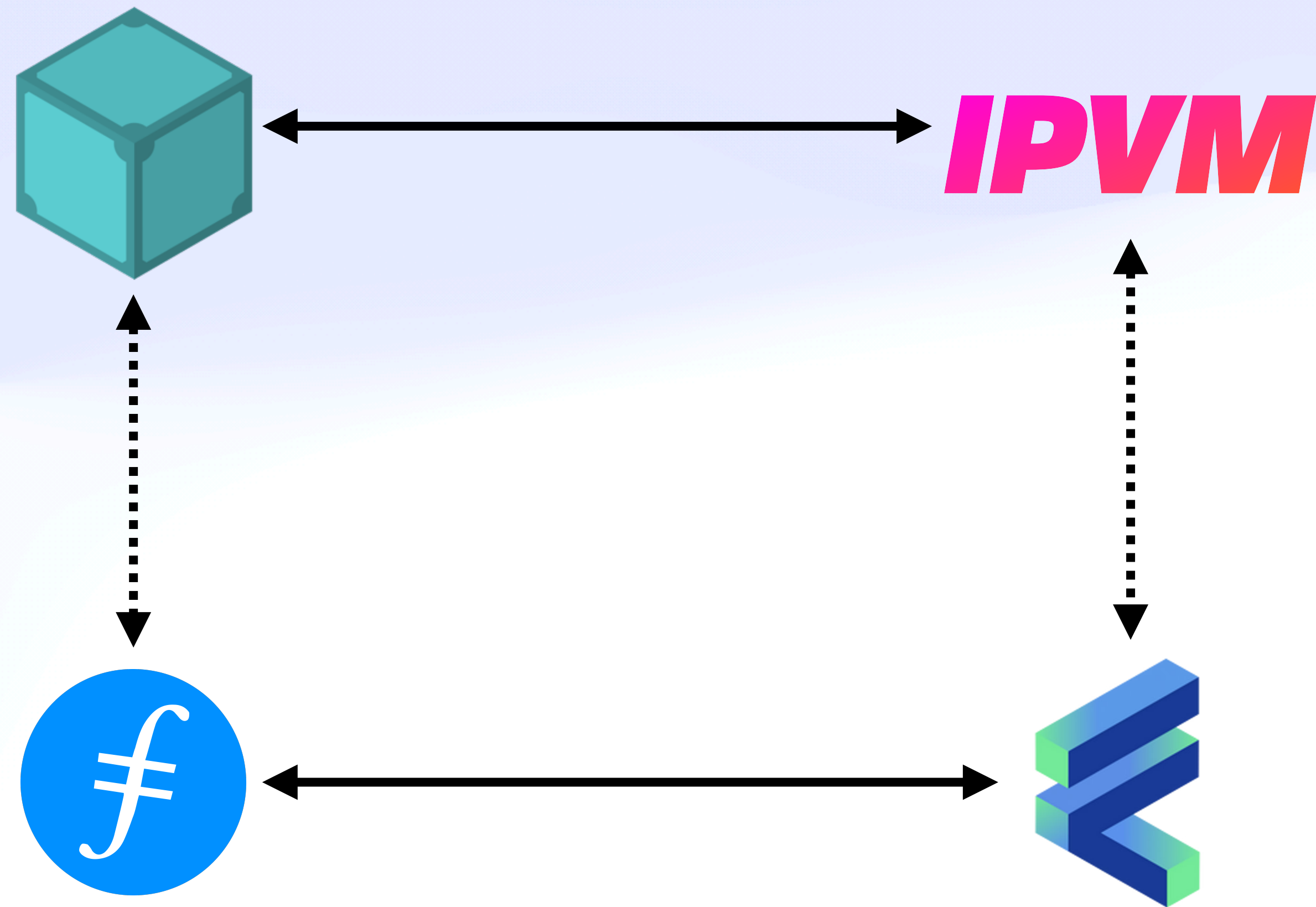
IPVM: Better Together

Consistency & Distribution



IPVM: Better Together

Consistency & Distribution



IPVM: Better Together

Everything, Everywhere, All At Once

IPVM: Better Together

Everything, Everywhere, All At Once

Nothing less than connecting
all of the world's users & services.

The "HTTP of Compute":
open, interoperable, & everywhere.

Must be ***substantially*** better
than the status quo.



A person with long dark hair, wearing a green quilted jacket and dark pants, is walking away from the camera down a dirt path in a forest. The path is lined with large, gnarled tree trunks and fallen logs. The background shows a dense forest of bare trees under a hazy sky. The overall mood is contemplative and slightly somber.

—We Can Do Better Than —

The Status Quo

The Status Quo

Consequences 

The Status Quo

Consequences

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

The Status Quo

Consequences 🍂

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



The Status Quo

Consequences 🍂

- Single source of truth ("**the**" database)
- Server-centric
 - "Front vs back end developers"
 - DevOps, Docker, k8s
 - How to train enough engineers?

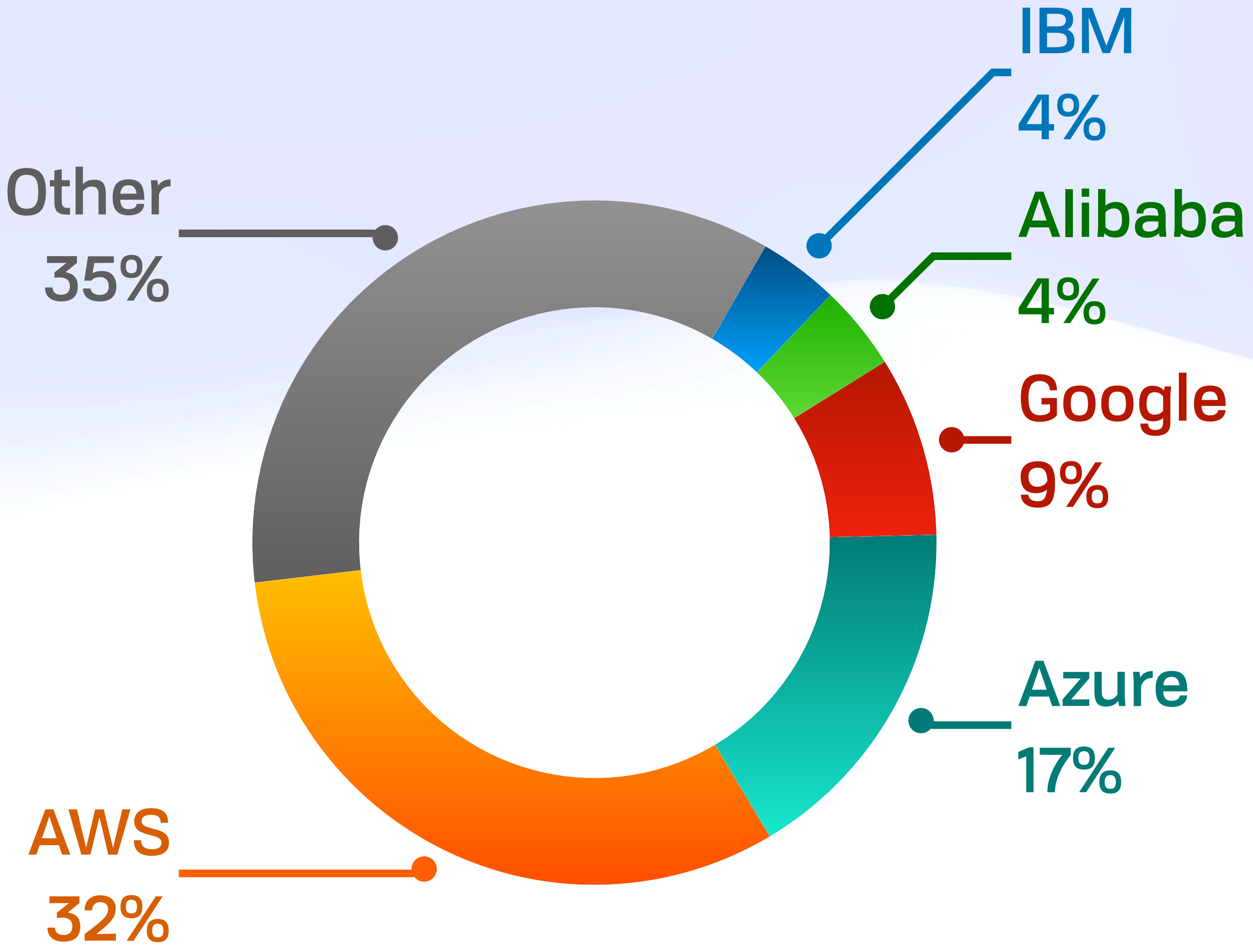


The Status Quo

Rolling Weight

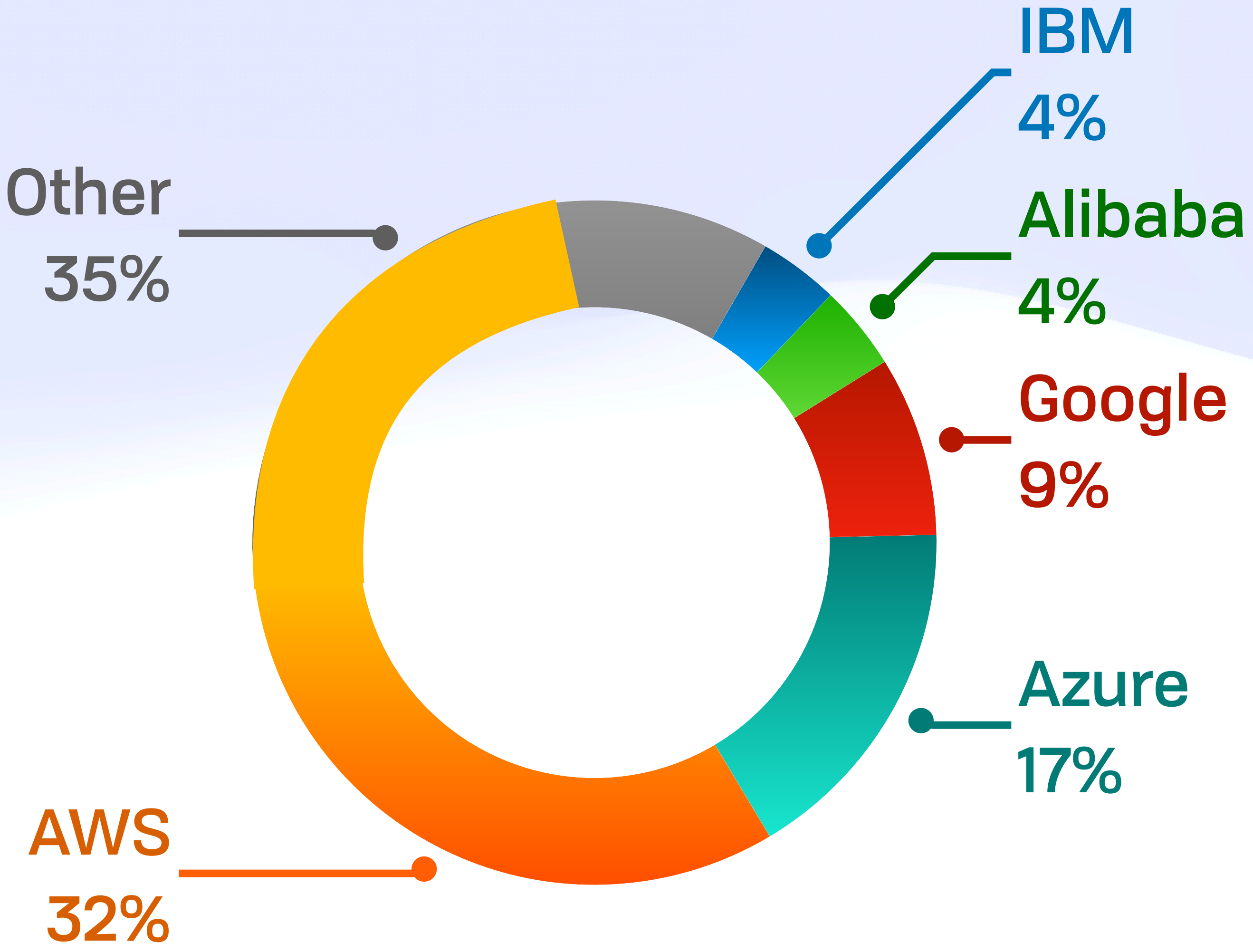
The Status Quo

Rolling Weight



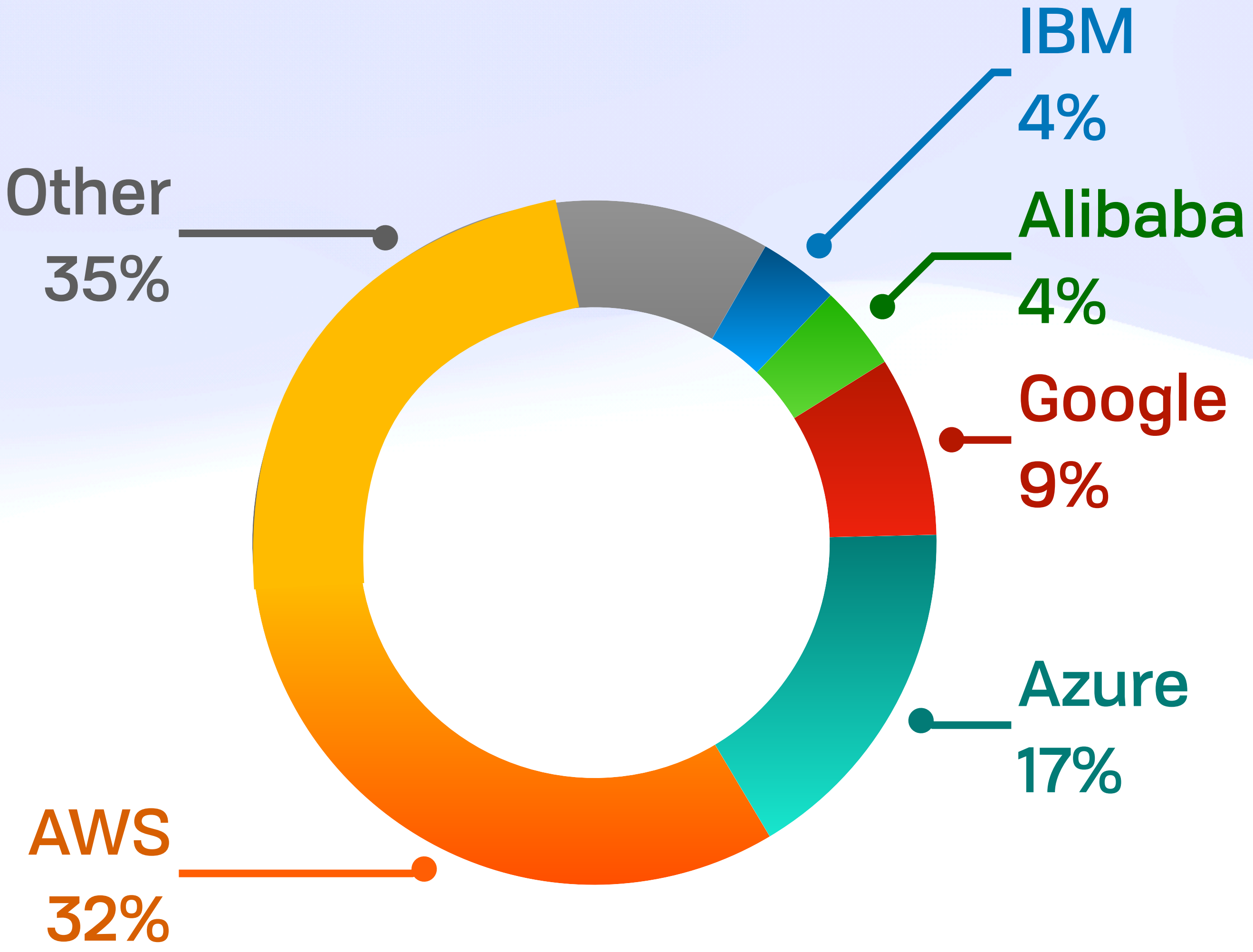
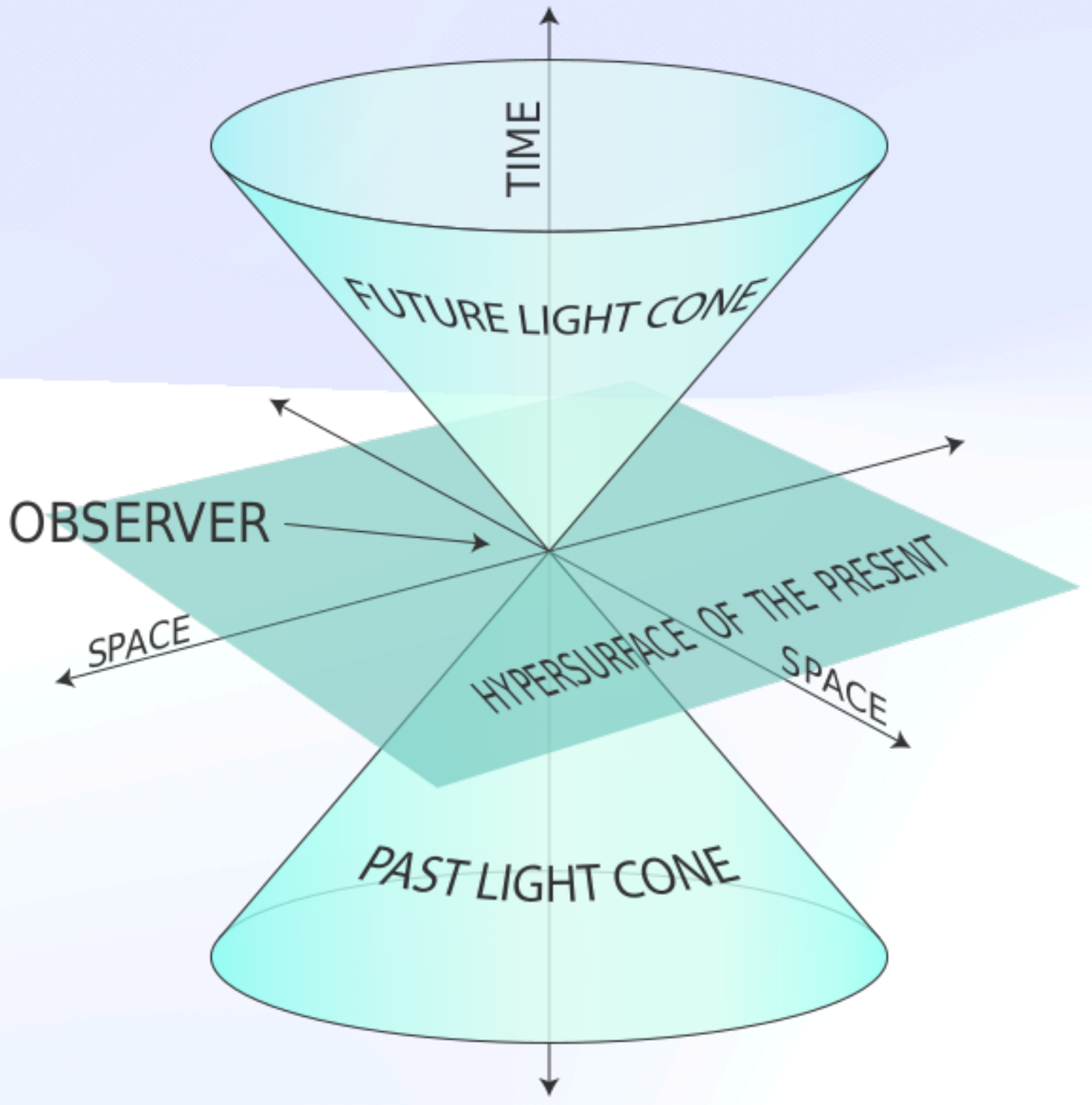
The Status Quo

Rolling Weight



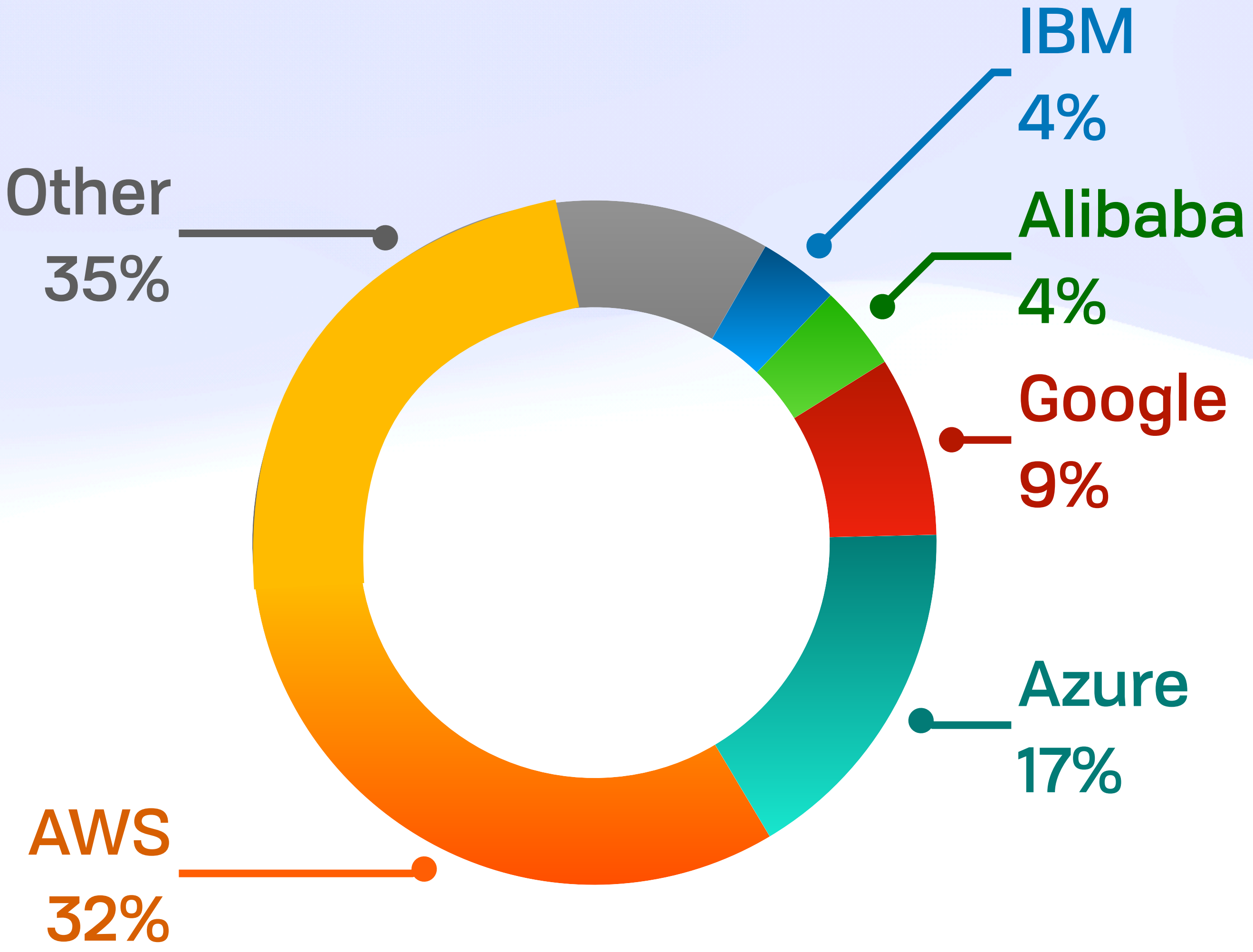
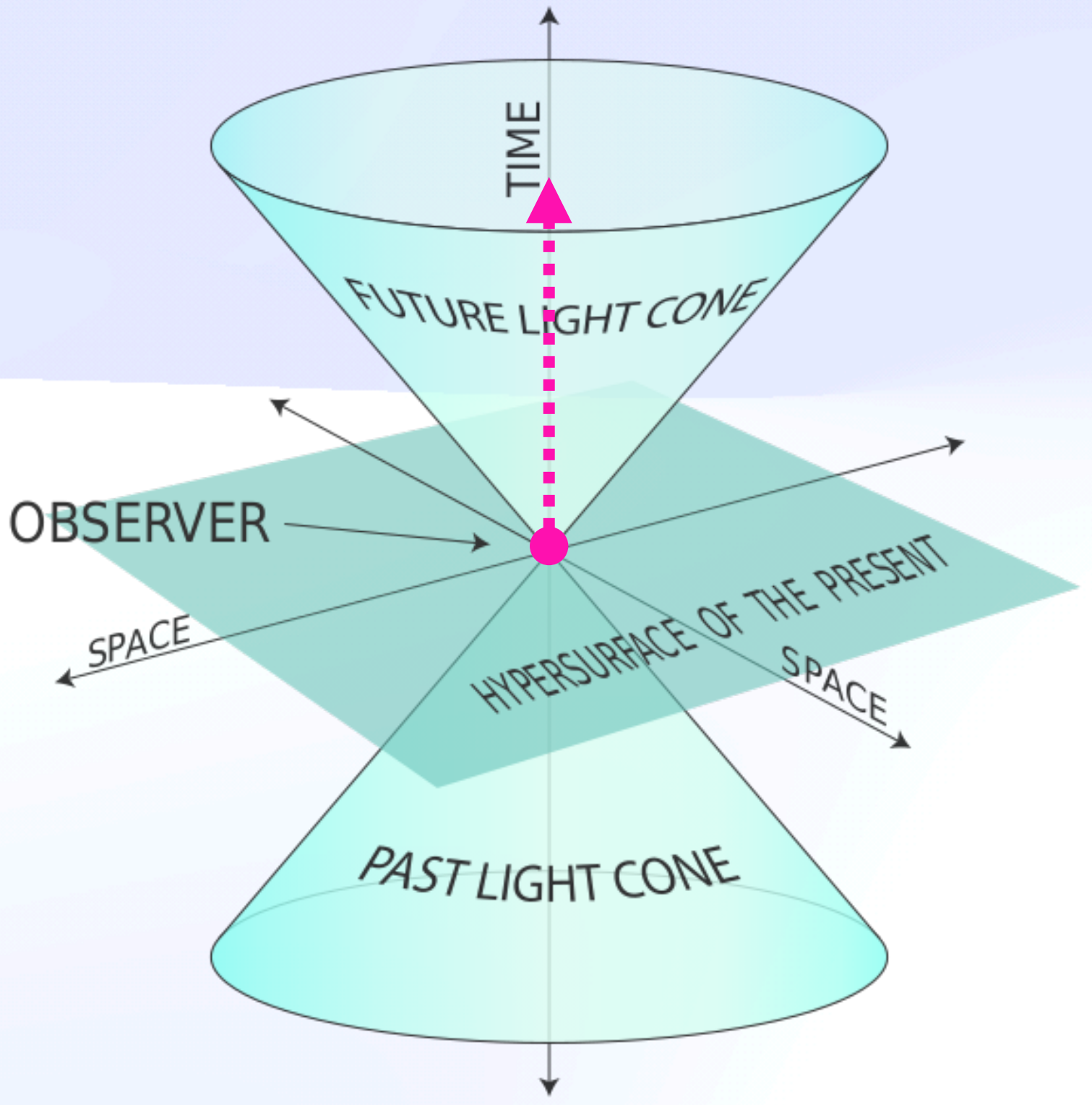
The Status Quo

Rolling Weight



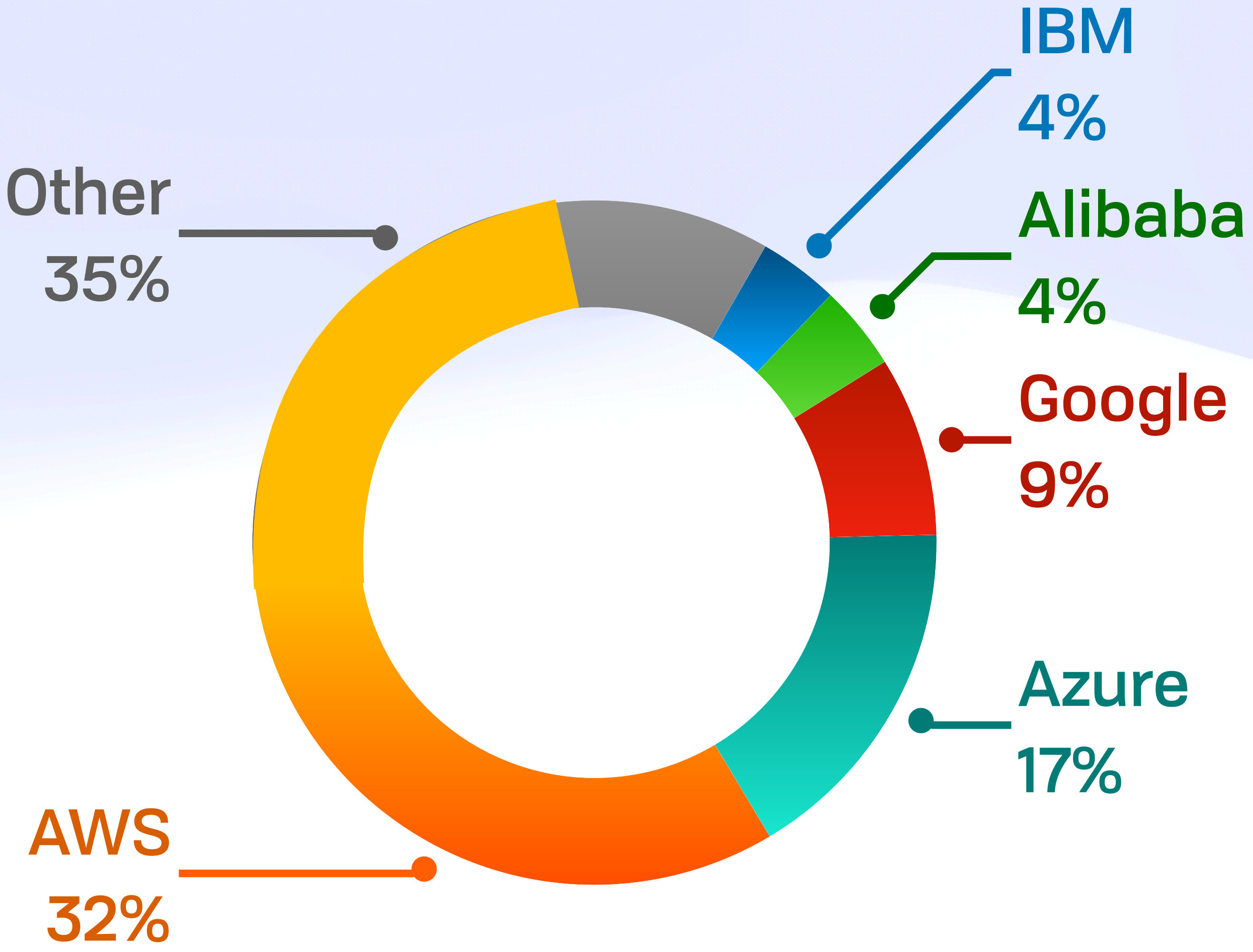
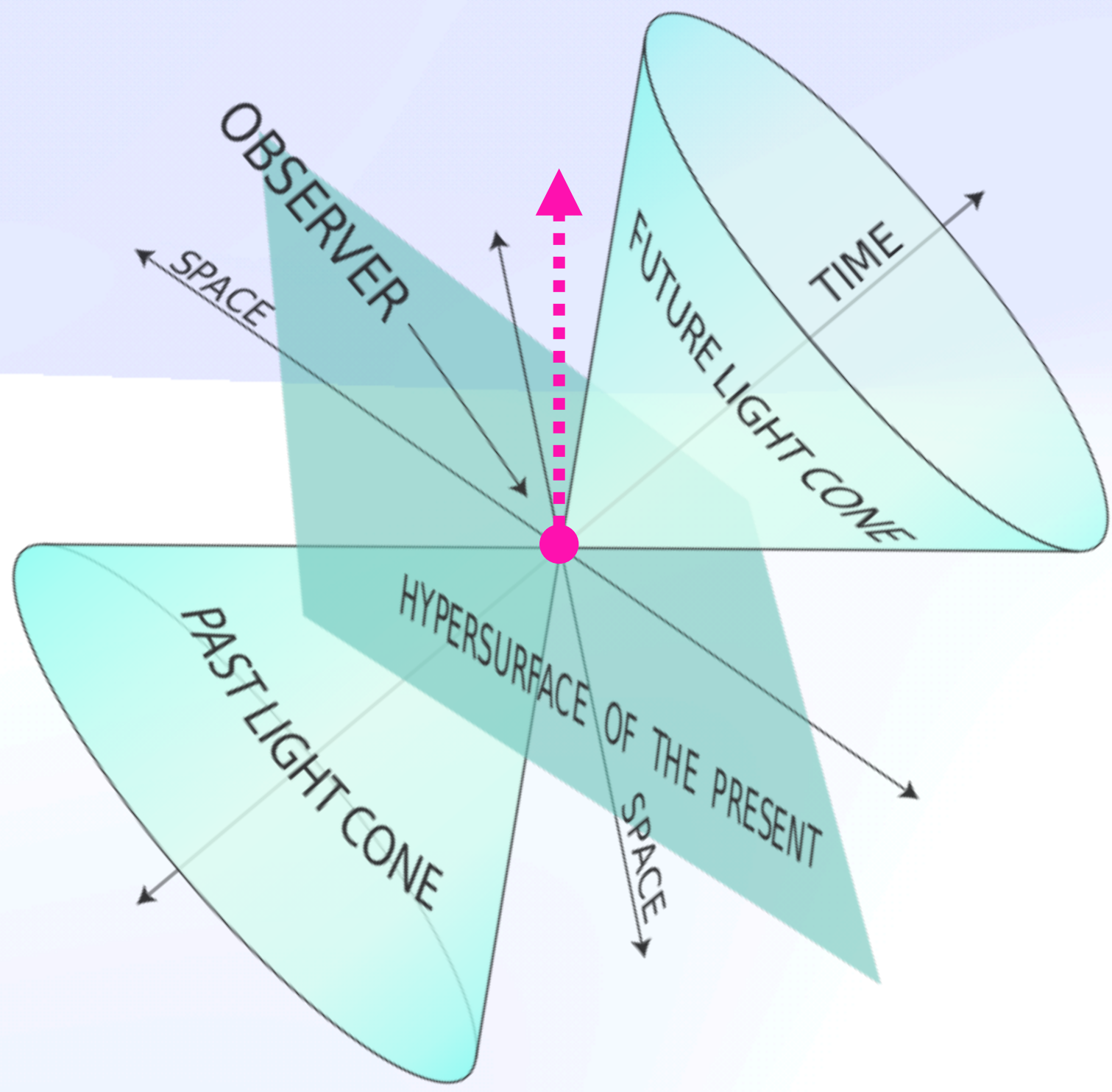
The Status Quo

Rolling Weight



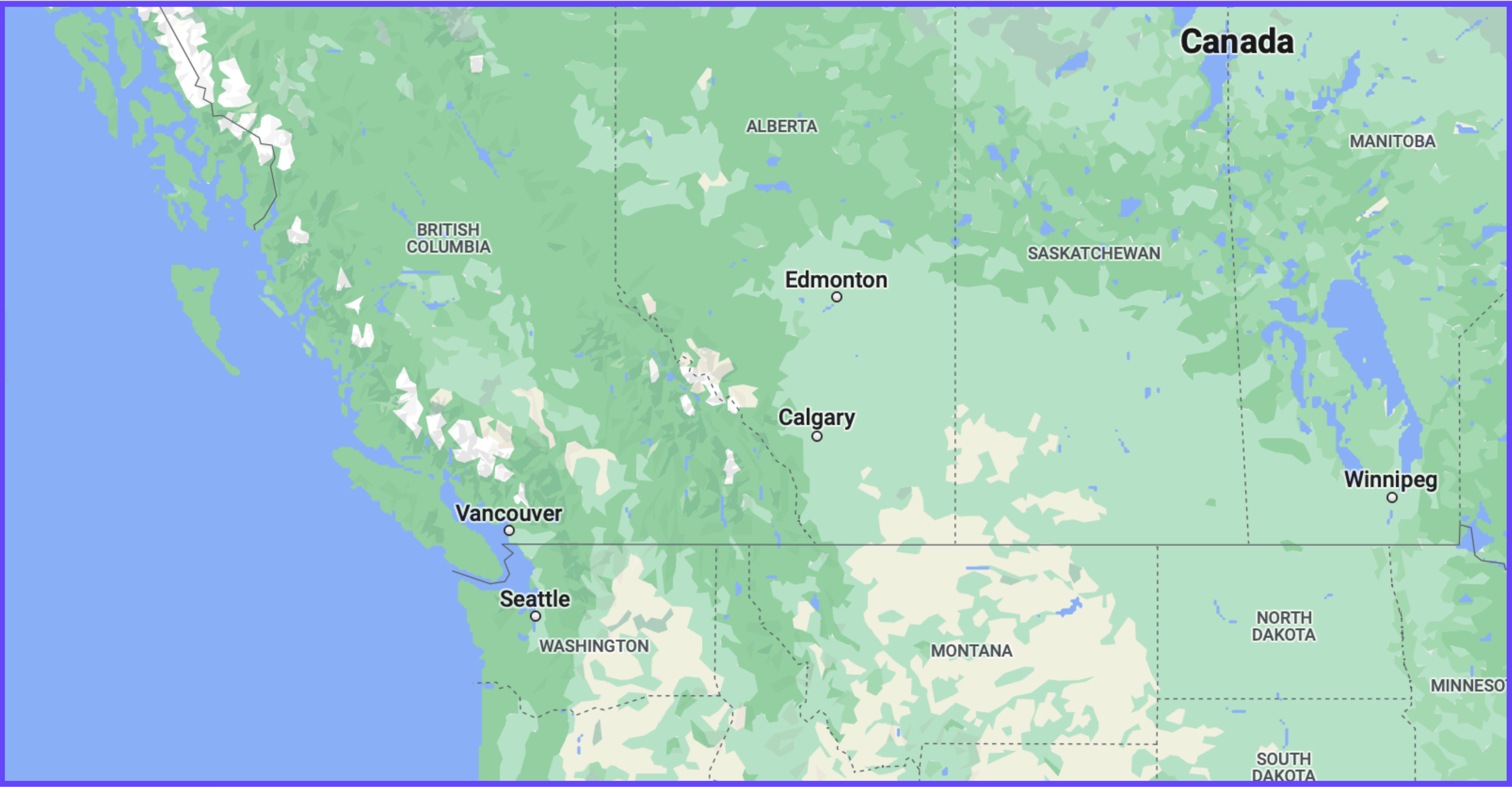
The Status Quo

Rolling Weight



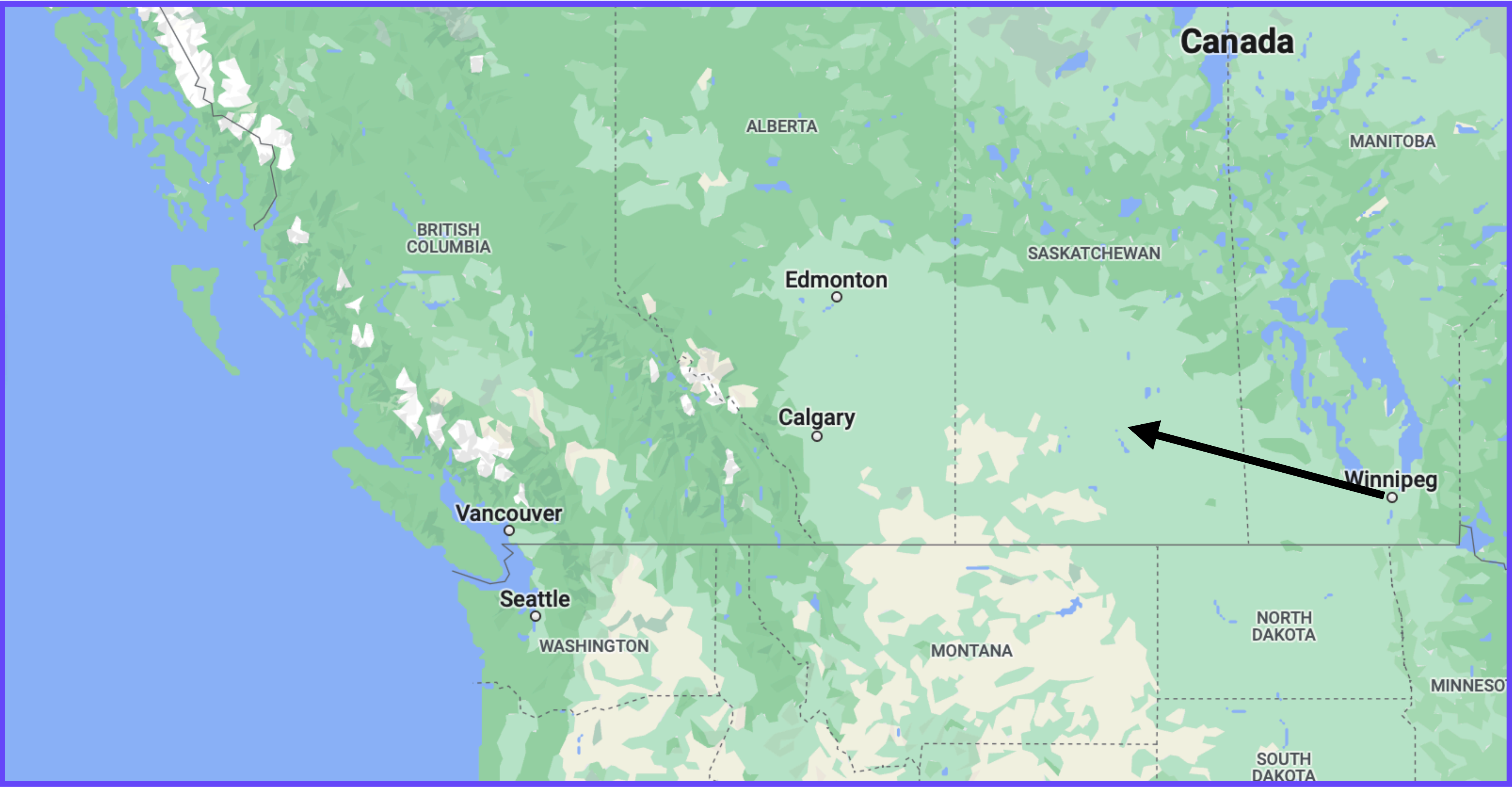
The Status Quo

Sending a "Direct" Message



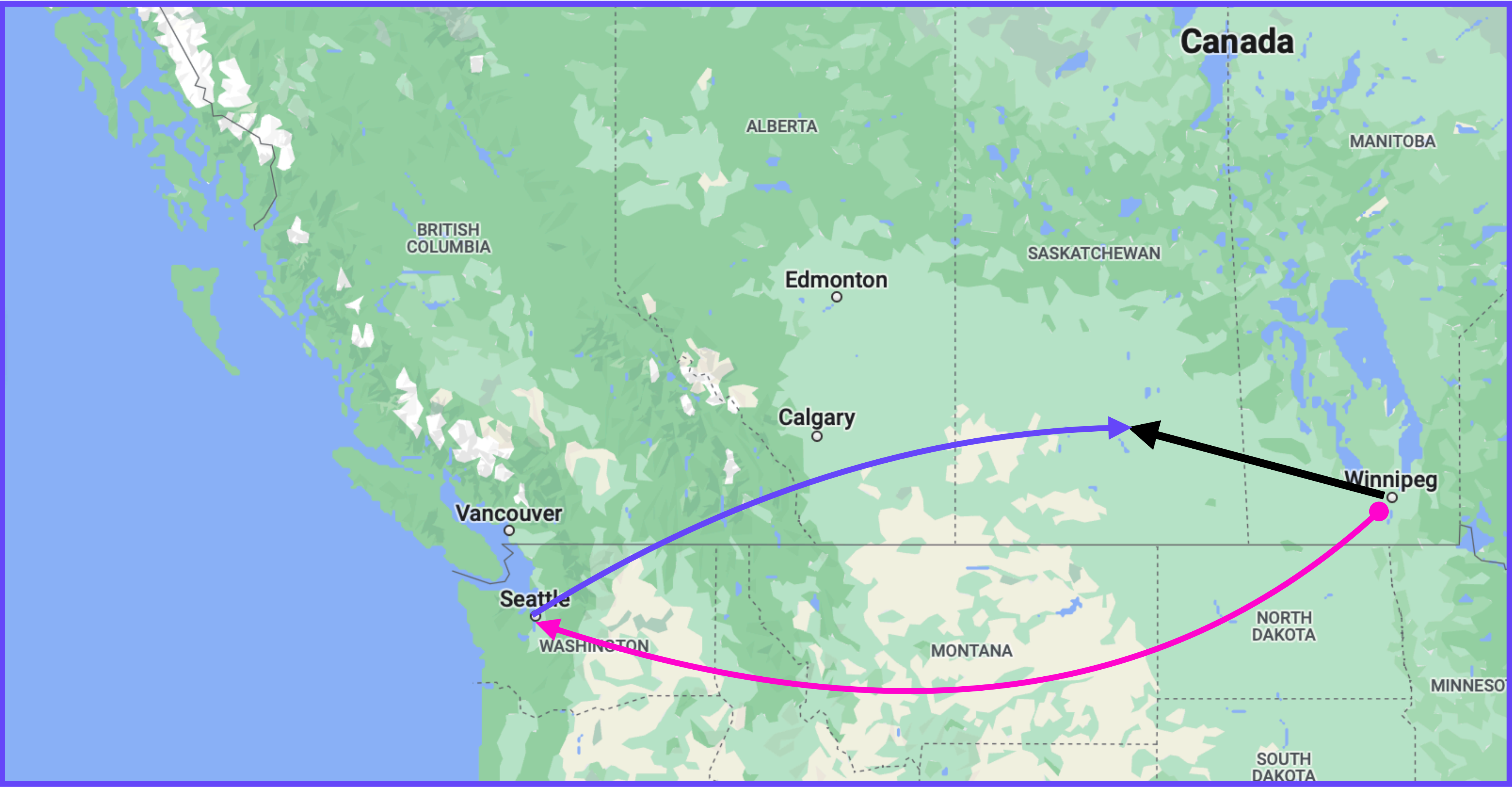
The Status Quo

Sending a "Direct" Message



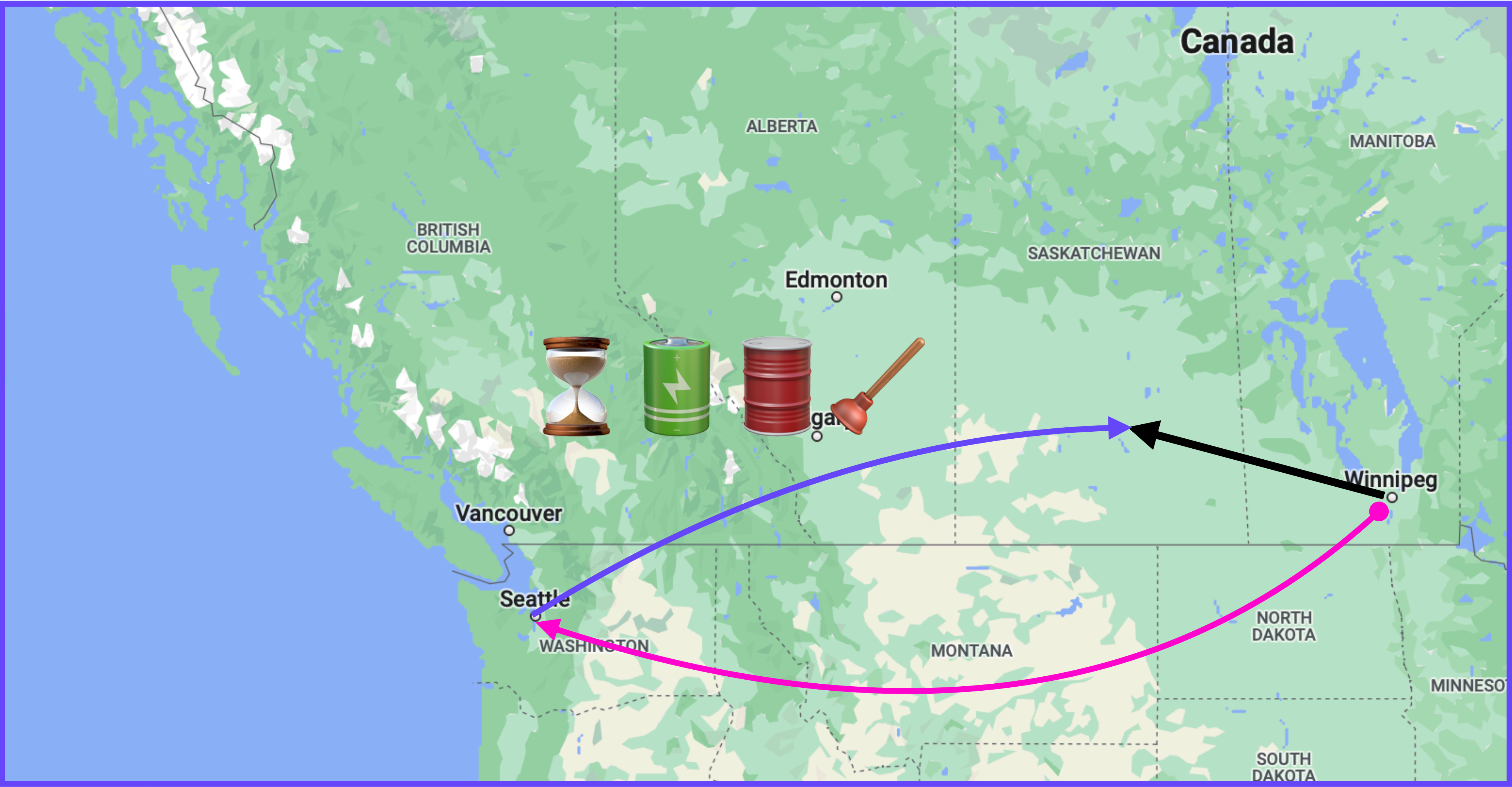
The Status Quo

Sending a "Direct" Message



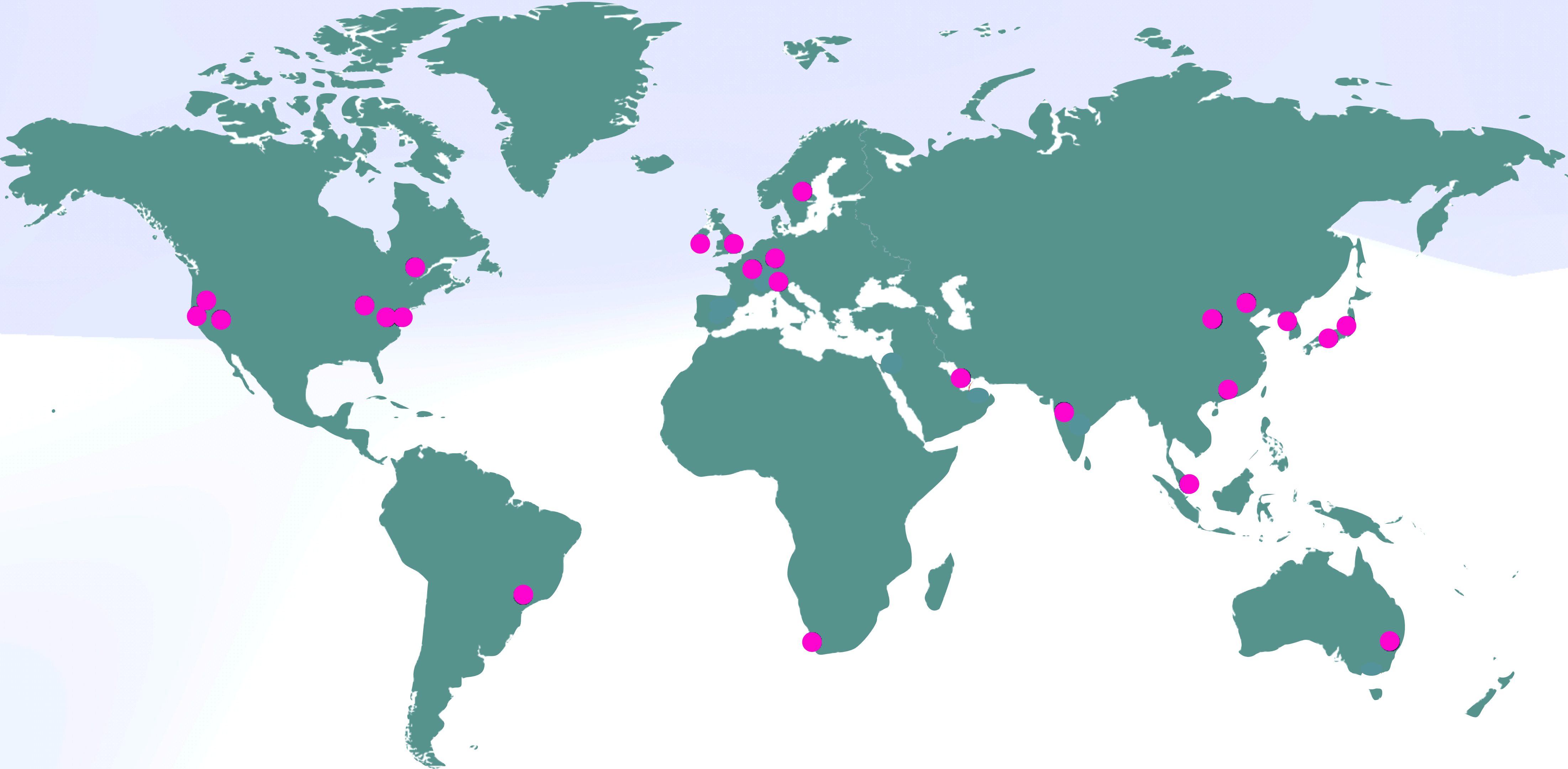
The Status Quo

Sending a "Direct" Message



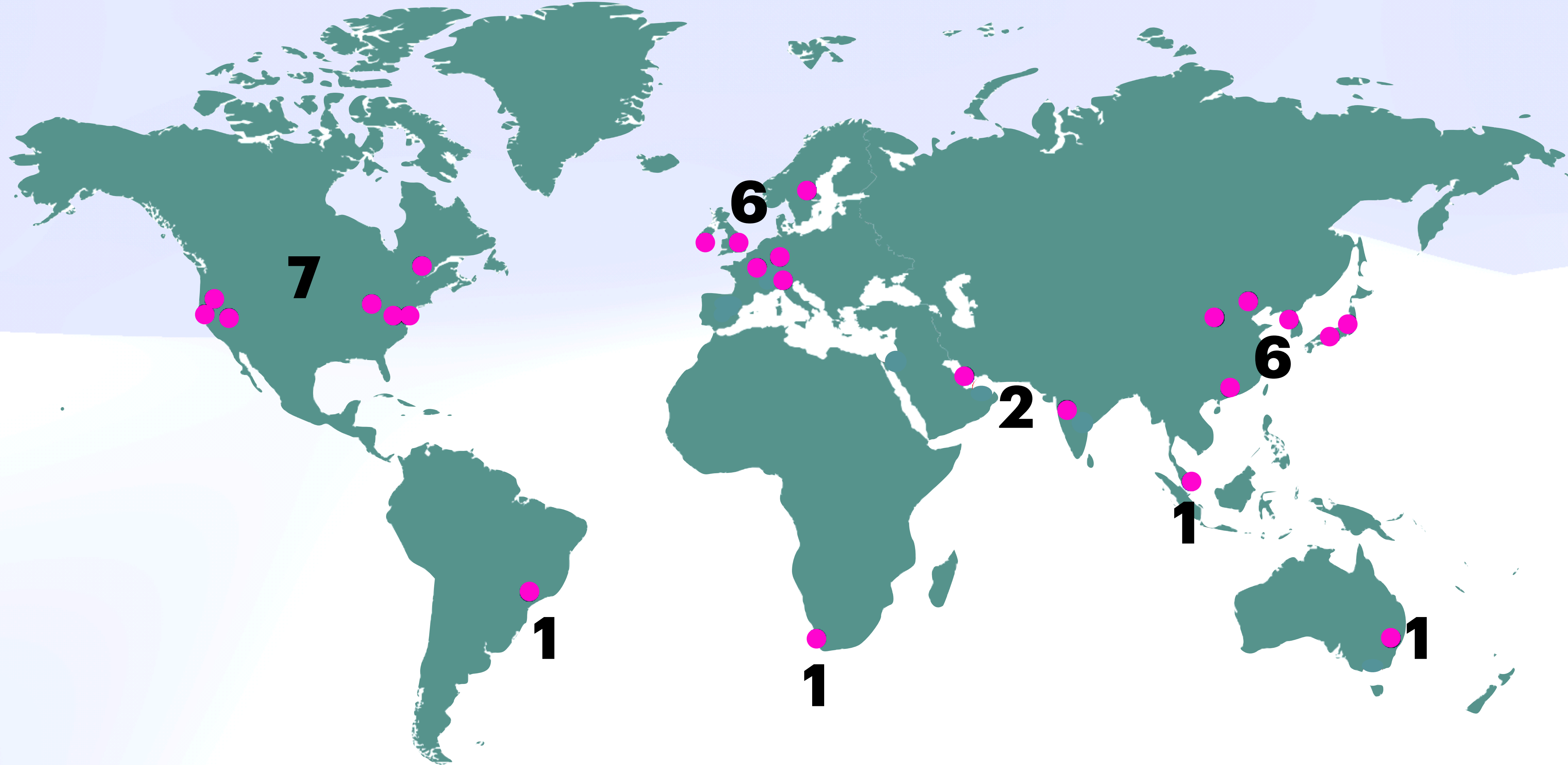
The Status Quo

Users vs Cloud Infra



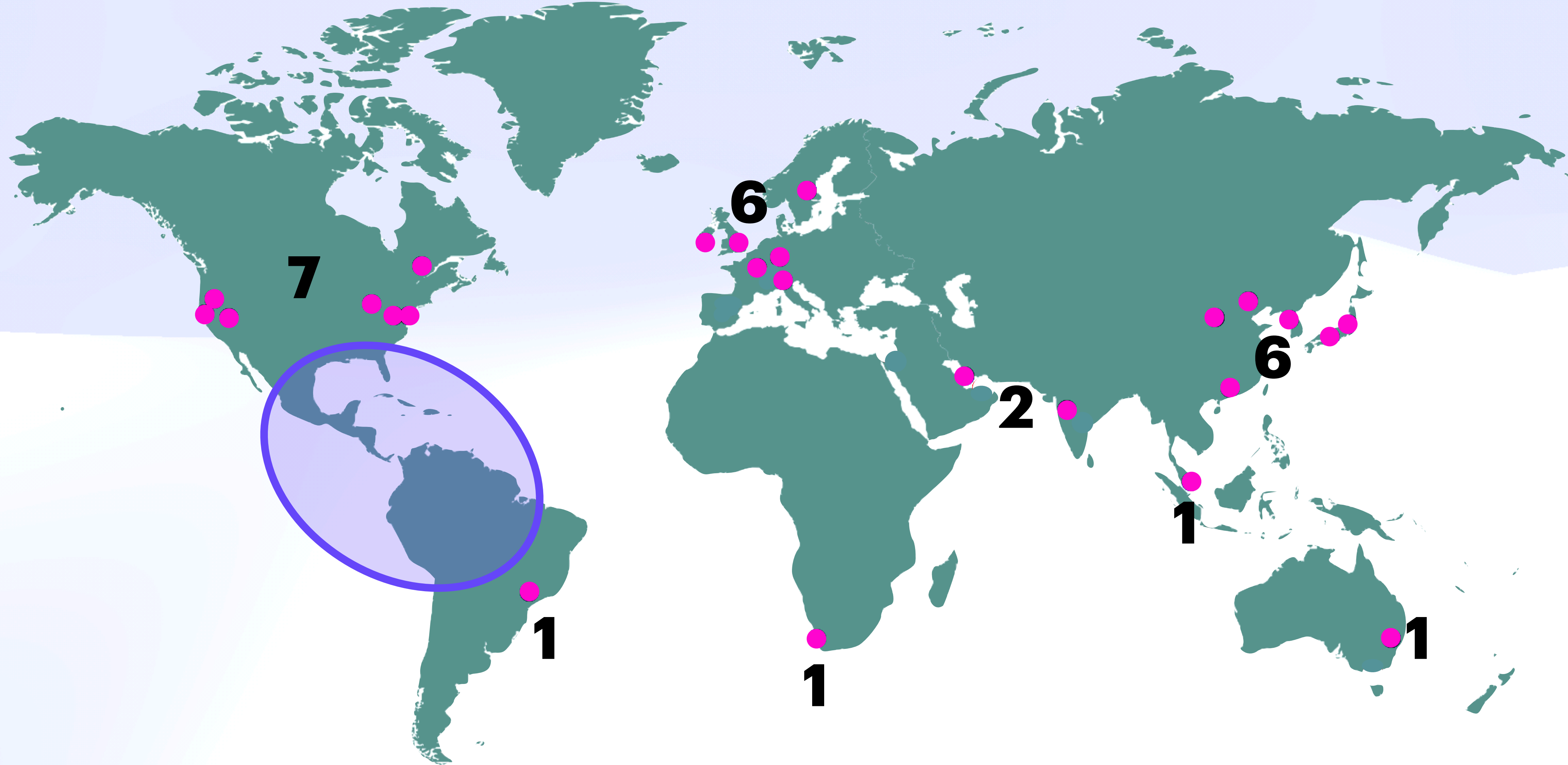
The Status Quo

Users vs Cloud Infra



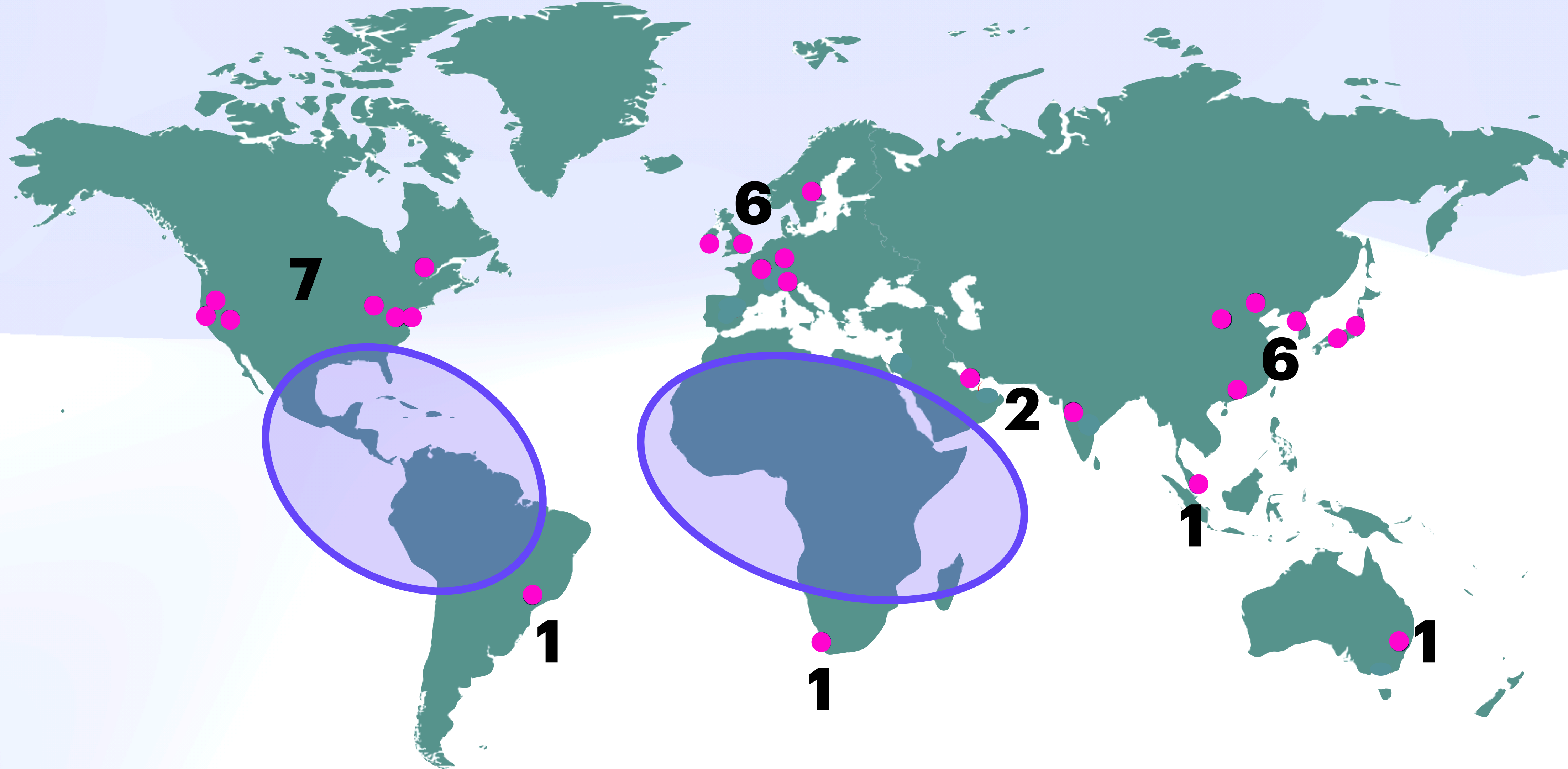
The Status Quo

Users vs Cloud Infra



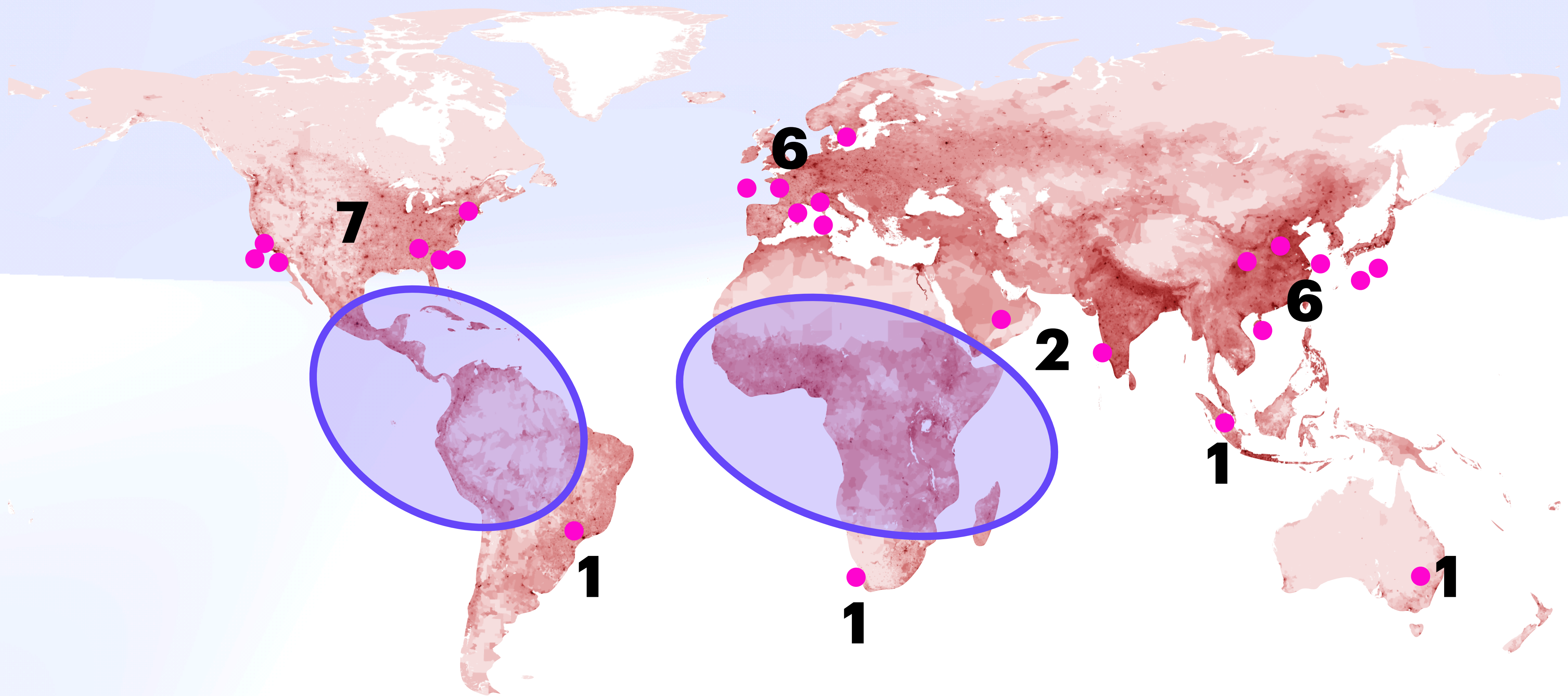
The Status Quo

Users vs Cloud Infra



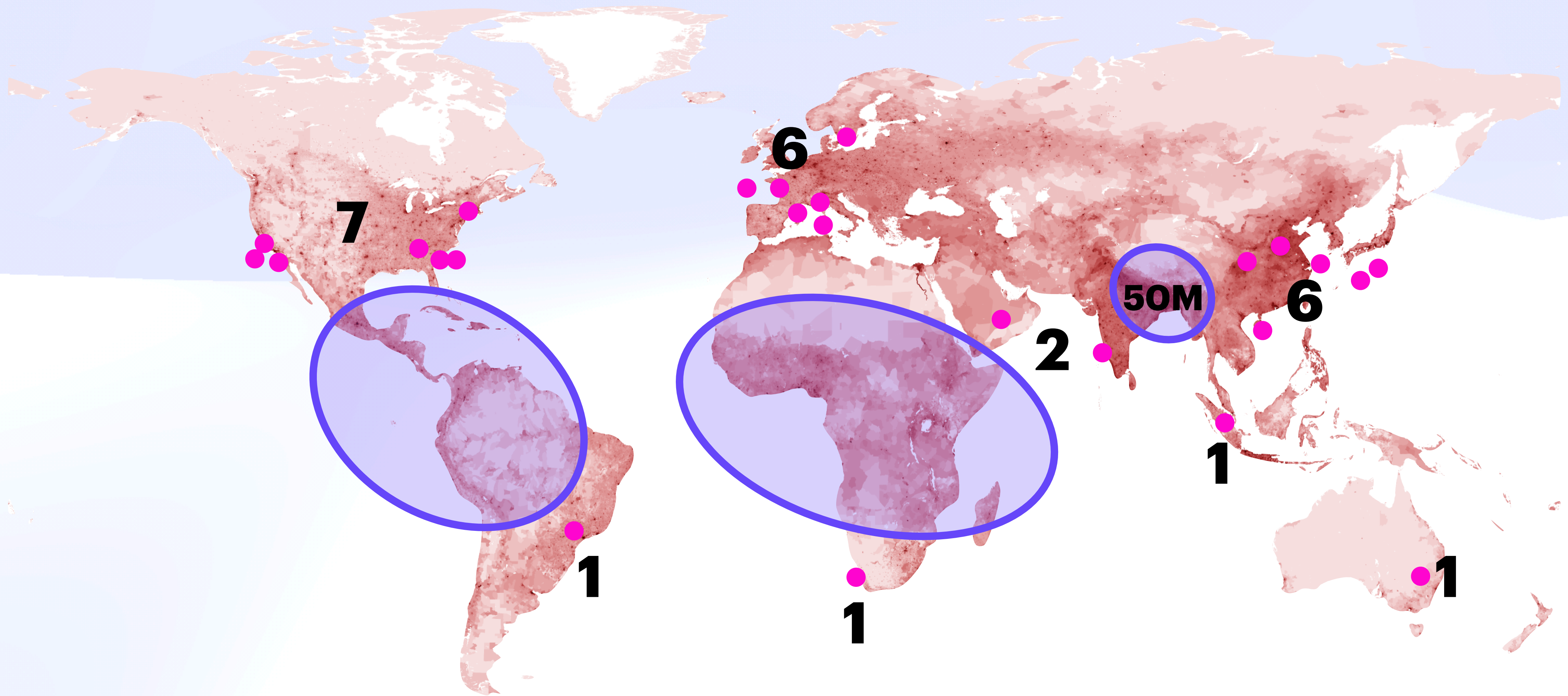
The Status Quo

Users vs Cloud Infra



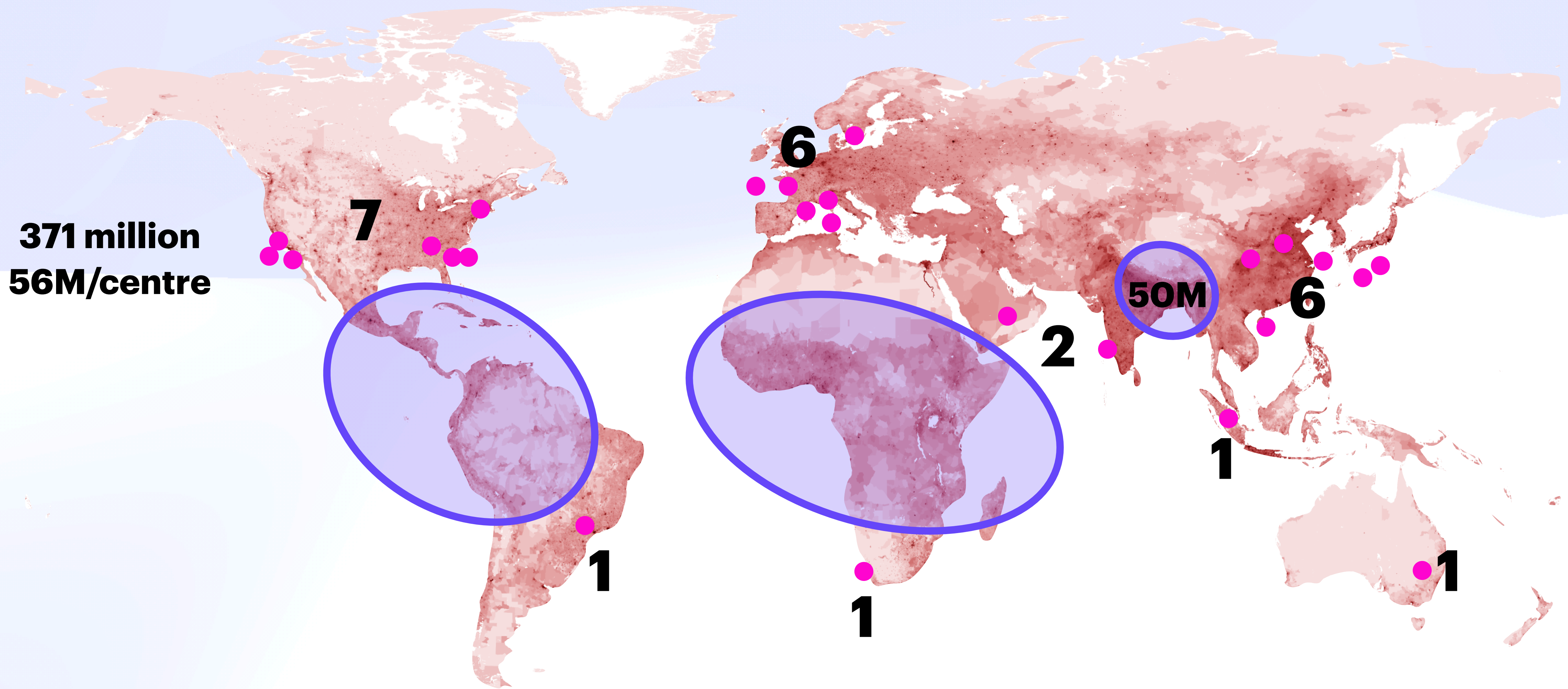
The Status Quo

Users vs Cloud Infra



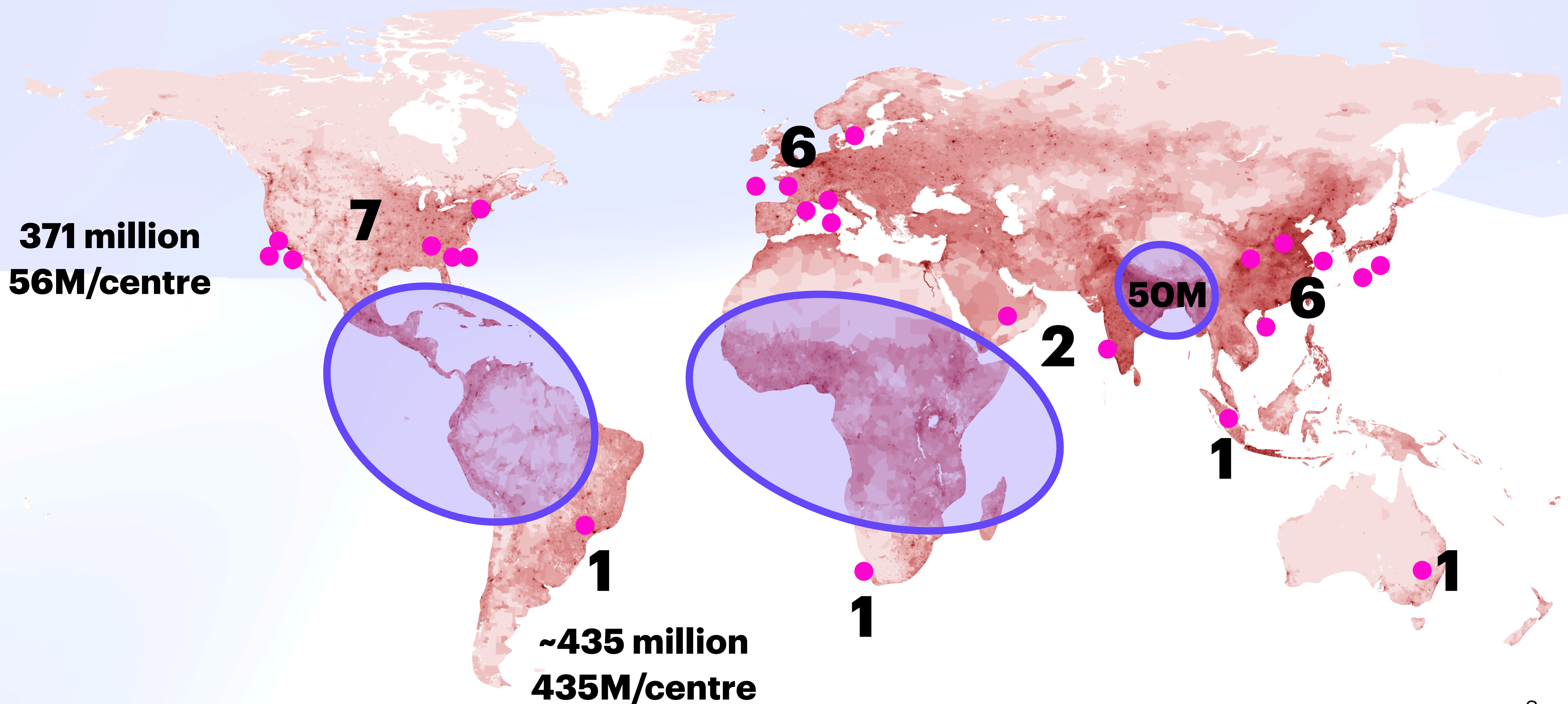
The Status Quo

Users vs Cloud Infra



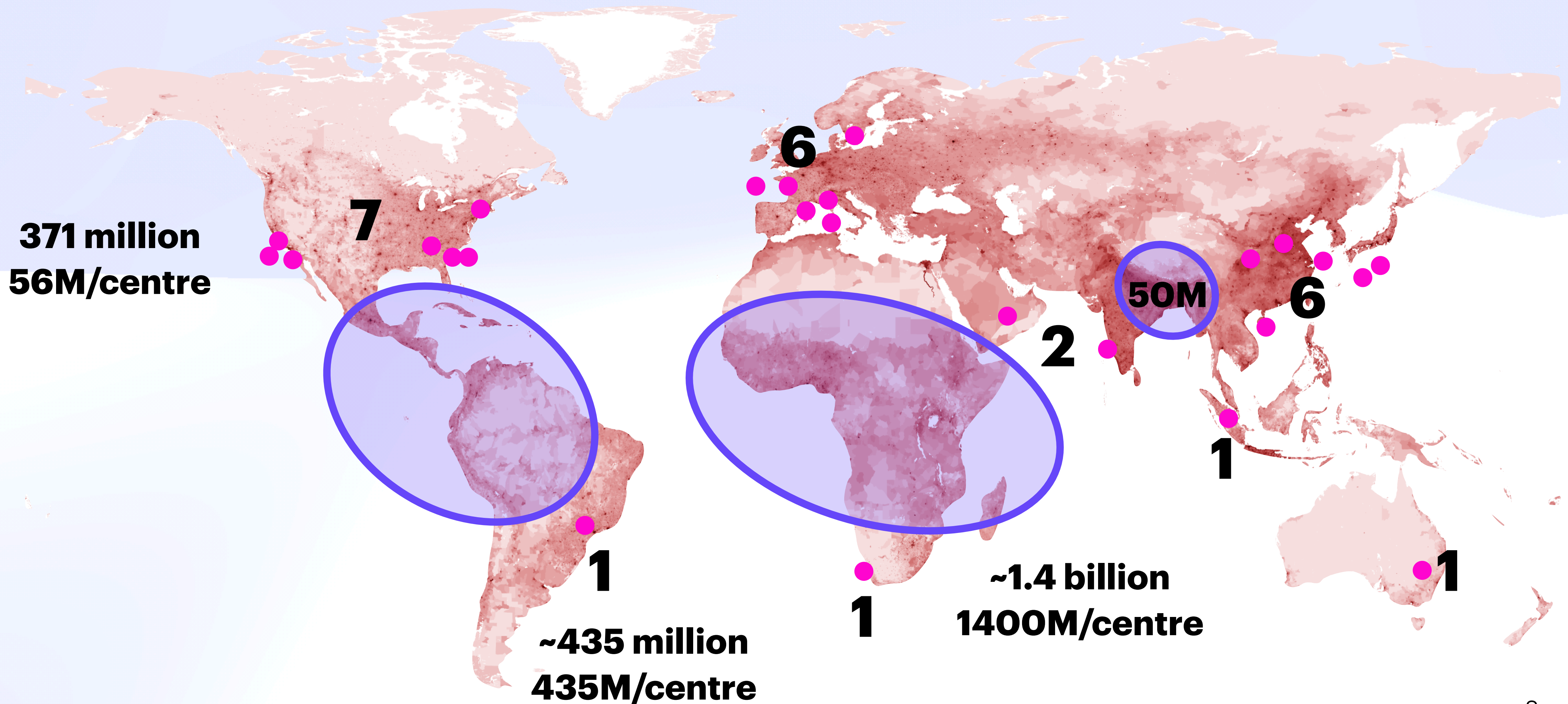
The Status Quo

Users vs Cloud Infra



The Status Quo

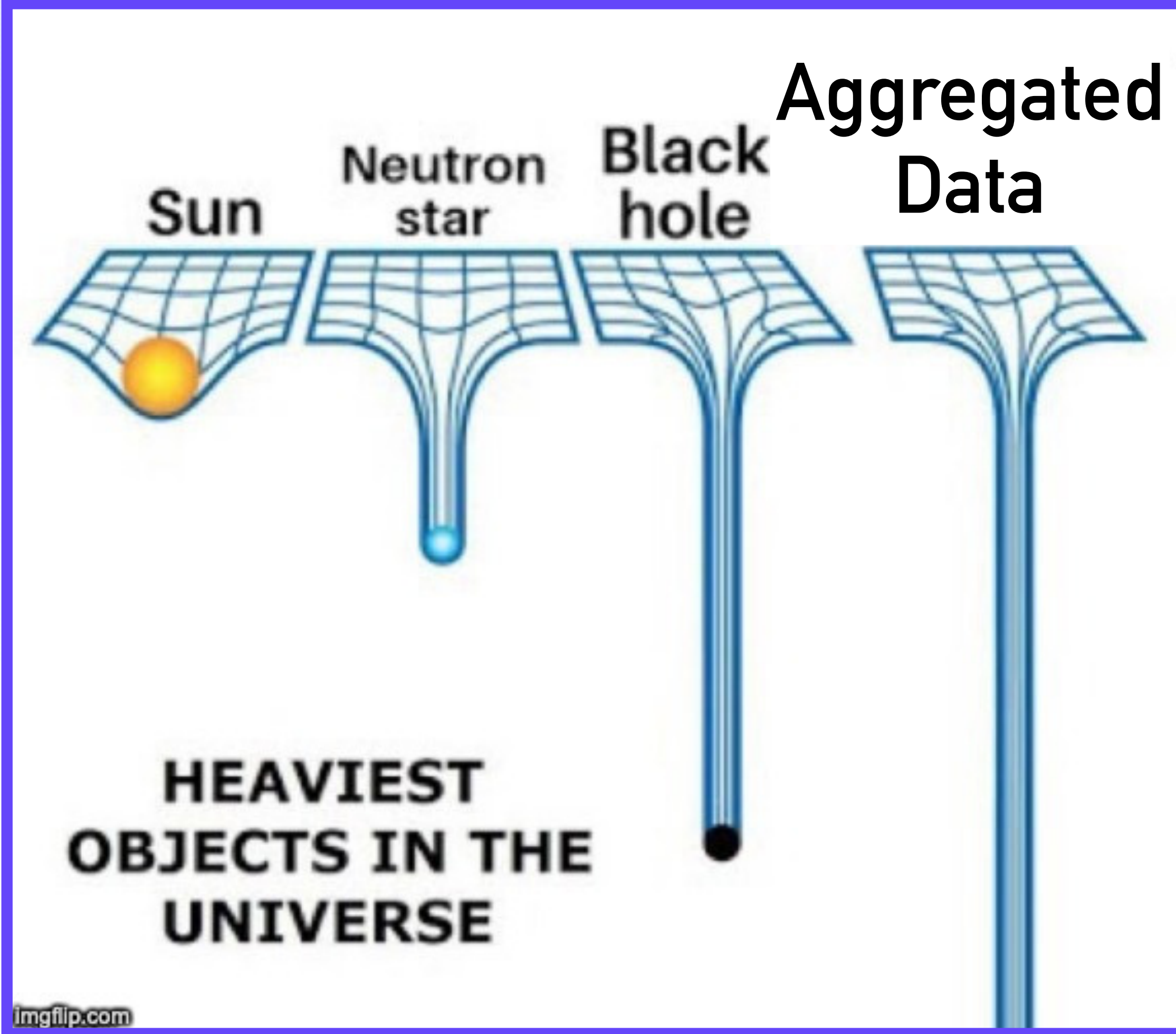
Users vs Cloud Infra



Source: AWS

The Status Quo

Event Horizon



The Status Quo

The Status Quo

As we continue to increase the number of globally connected devices, we **must embrace a design that considers every single member in the system as the primary site** for the data that it is generates.

It is **completely impractical** that we can look at a **single, or a small number, of globally distributed data centers** as the **primary site for all global information** that we desire to perform computations with.

– **Meiklejohn**, A Certain Tendency Of The Database Community

The Status Quo





A photograph of a tunnel with tracks leading to a bright light at the end. The tunnel is dark and narrow, with the tracks curving slightly to the right. The light at the end is bright and yellowish, creating a strong contrast with the dark interior. The tunnel walls are made of rough, textured material, possibly concrete or stone.

— The Times They Are A-Changin' —

Signs of a Way Out

Signs of a Way Out

Back to Our Roots

Signs of a Way Out

Back to Our Roots

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

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

Signs of a Way Out

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

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

Signs of a Way Out

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

– The Web Foundation, History of the Web

Signs of a Way Out

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

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

Signs of a Way Out

Everything, Everywhere

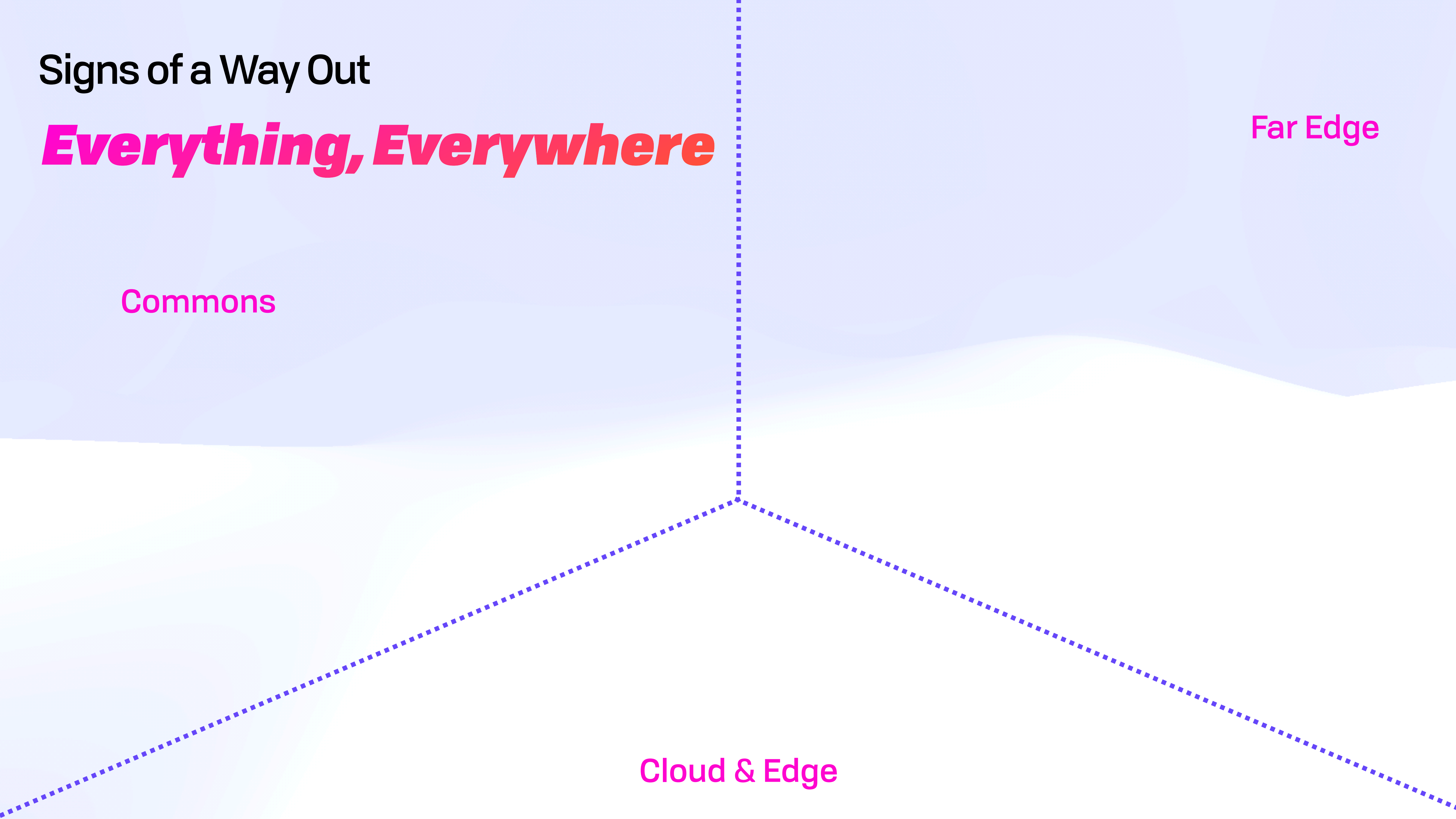
Signs of a Way Out

Everything, Everywhere

Far Edge

Commons

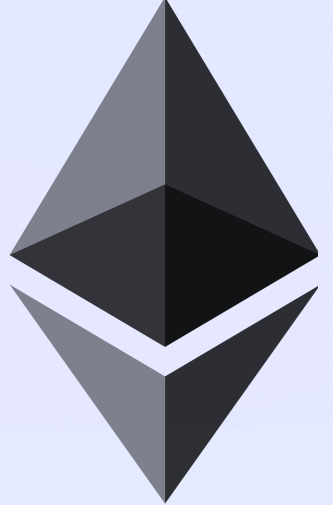
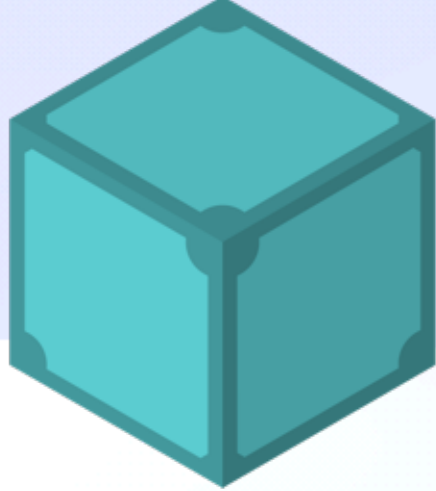
Cloud & Edge



Signs of a Way Out

Everything, Everywhere

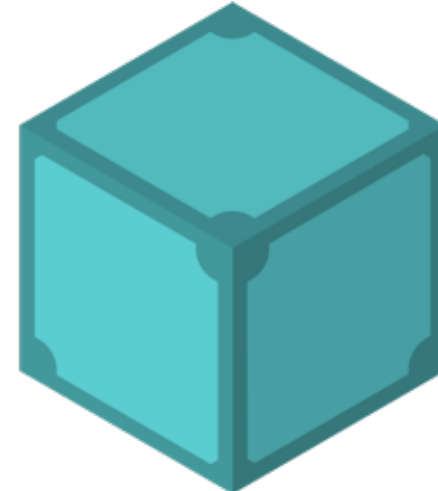
Commons



Cloud & Edge



Far Edge

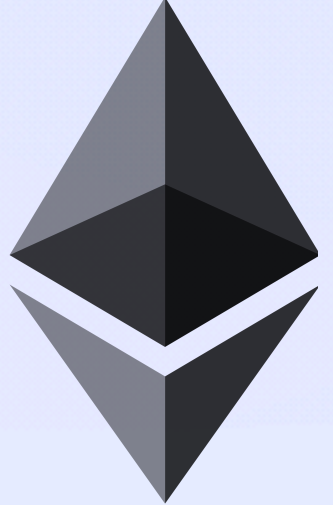
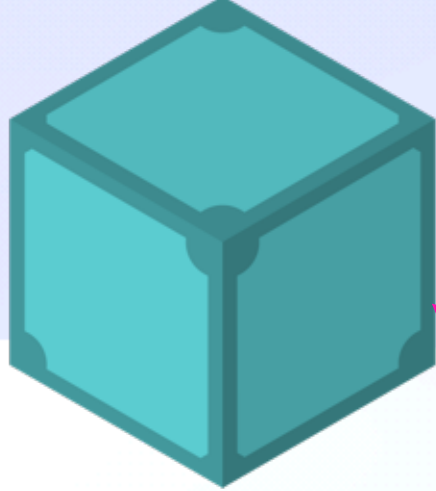


Signs of a Way Out

Everything, Everywhere

Commons

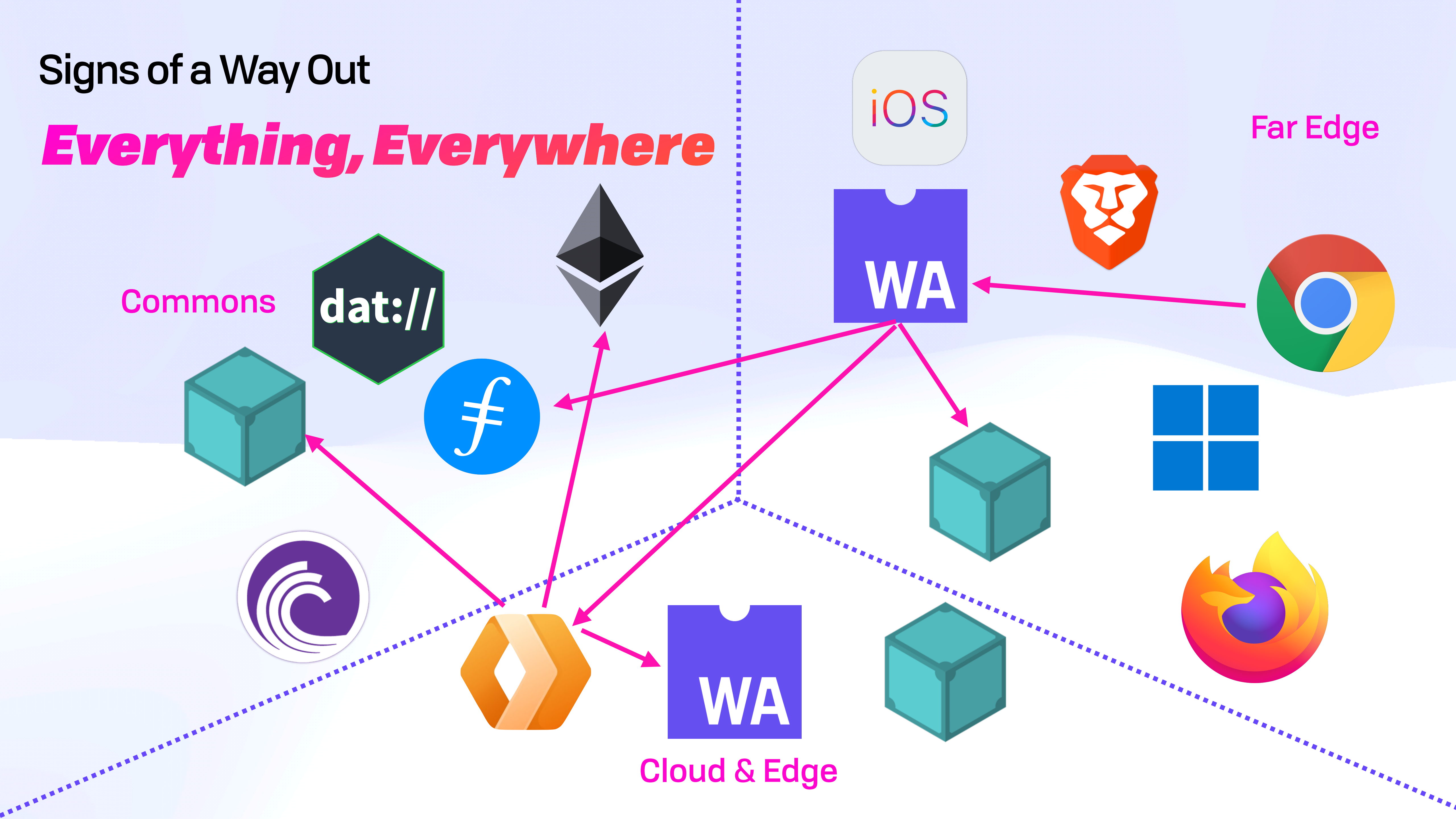
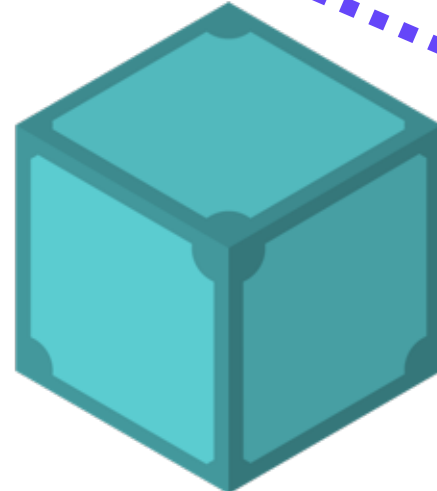
dat://



Cloud & Edge



Far Edge



Signs of a Way Out

Dependency Stack

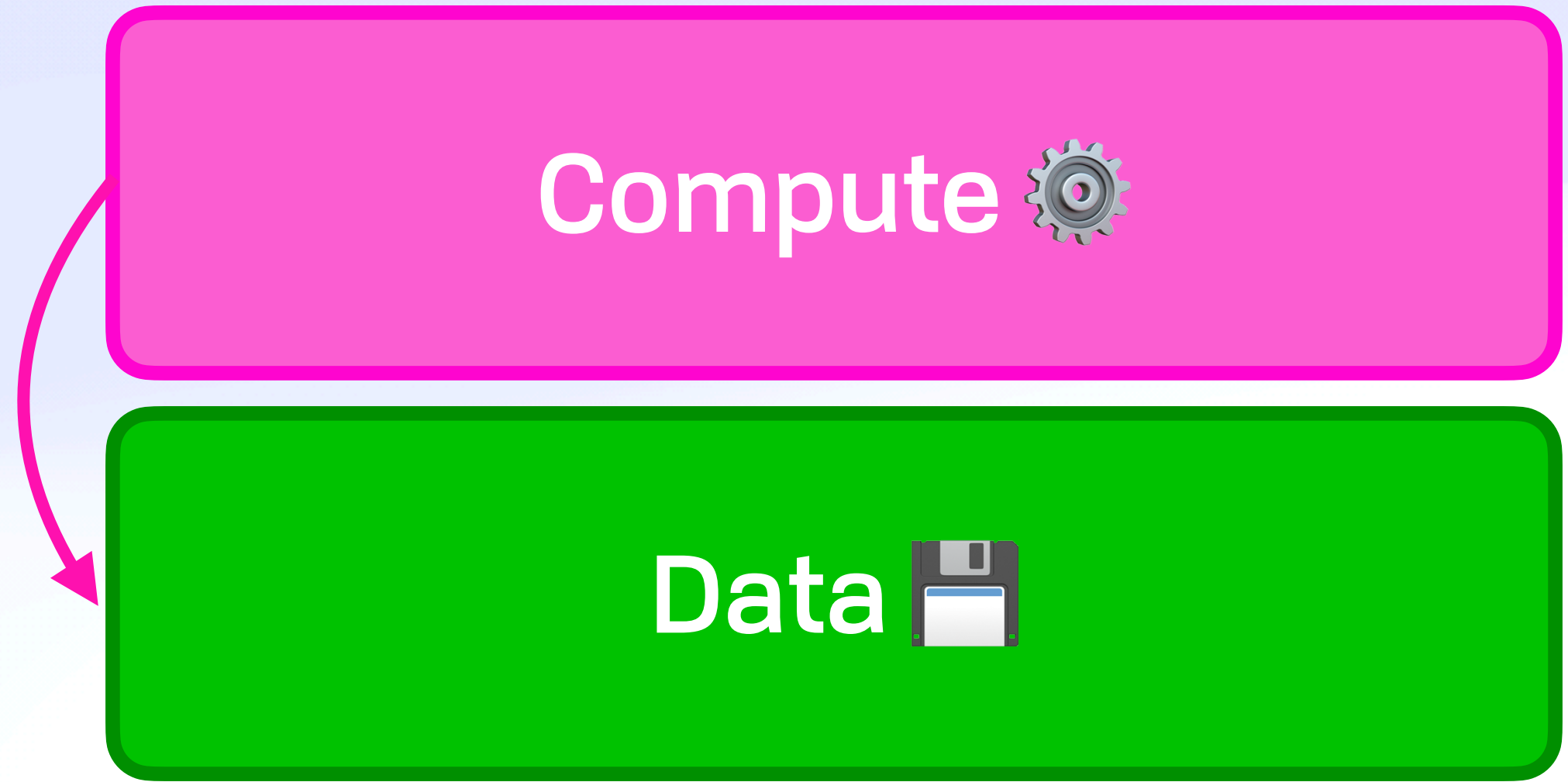
Signs of a Way Out

Dependency Stack

Compute 

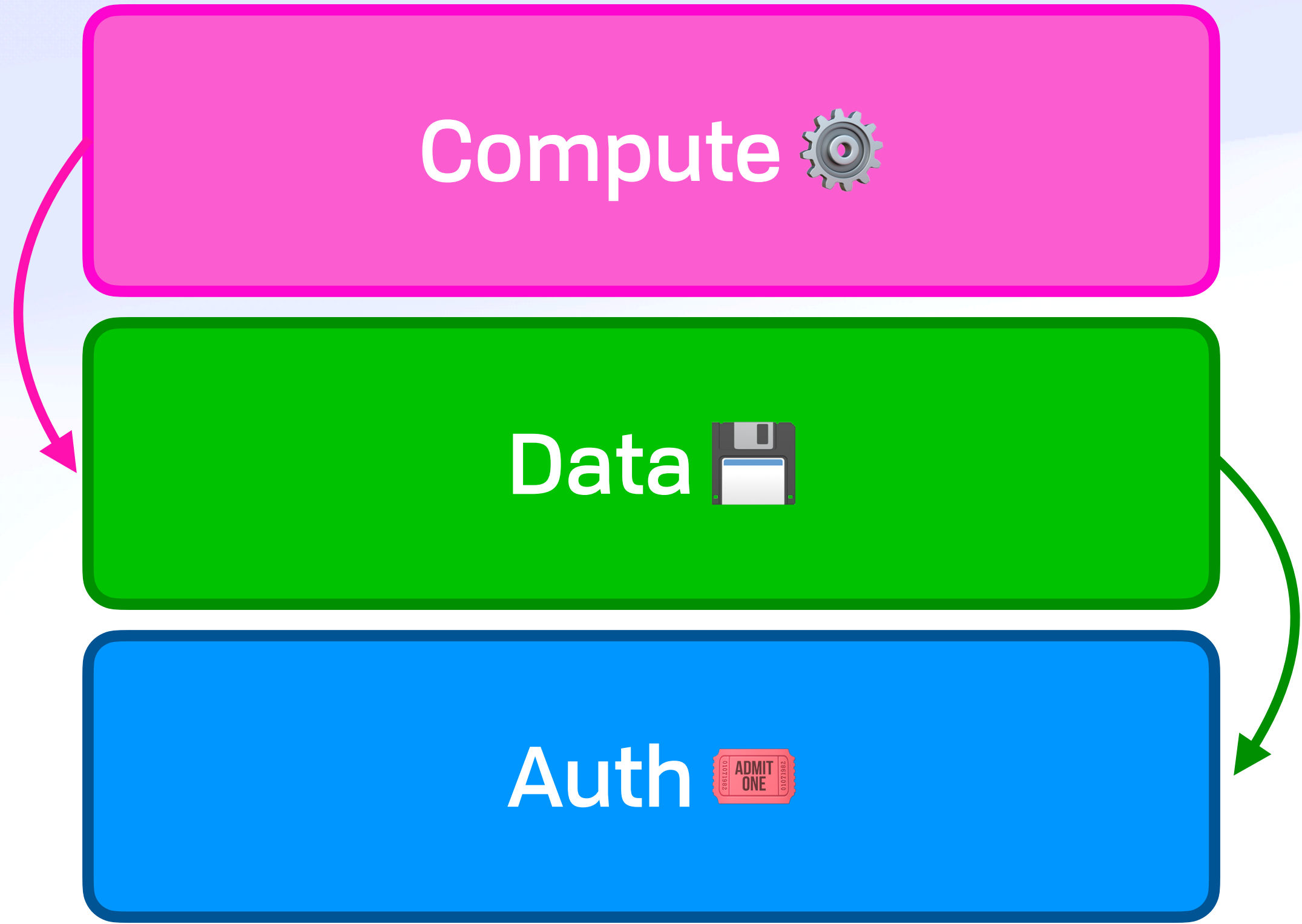
Signs of a Way Out

Dependency Stack



Signs of a Way Out

Dependency Stack



Signs of a Way Out

ACL Redux

Signs of a Way Out

ACL Redux



Signs of a Way Out

ACL Redux



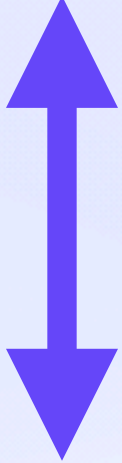
Signs of a Way Out

ACL Redux



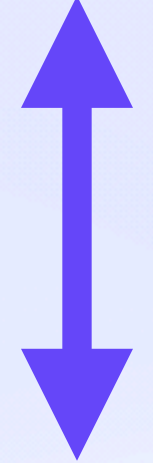
Signs of a Way Out

ACL Redux



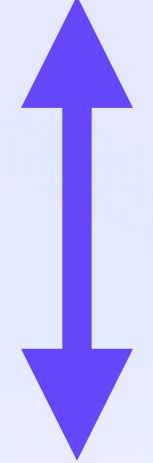
Signs of a Way Out

ACL Redux



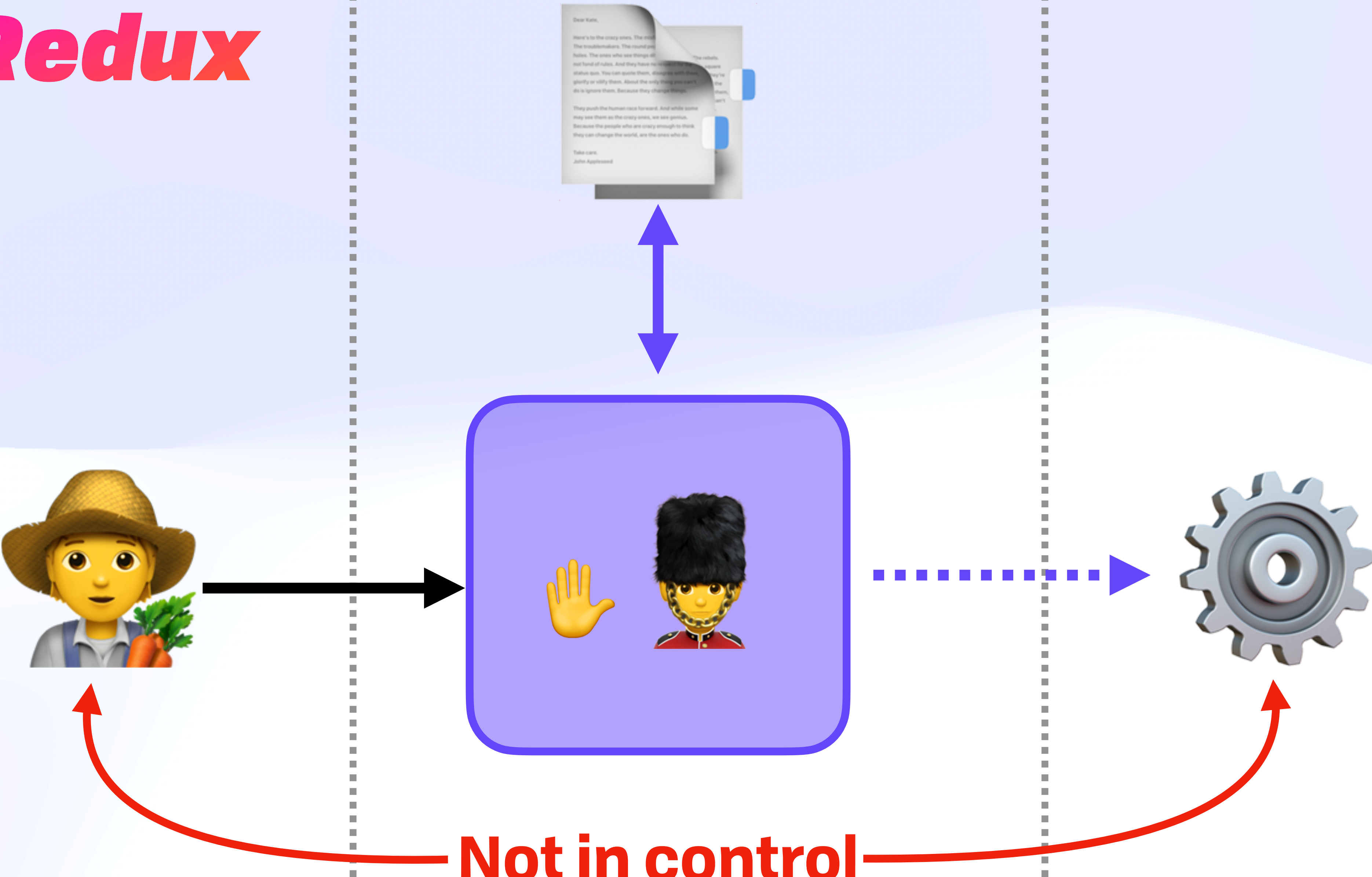
Signs of a Way Out

ACL Redux



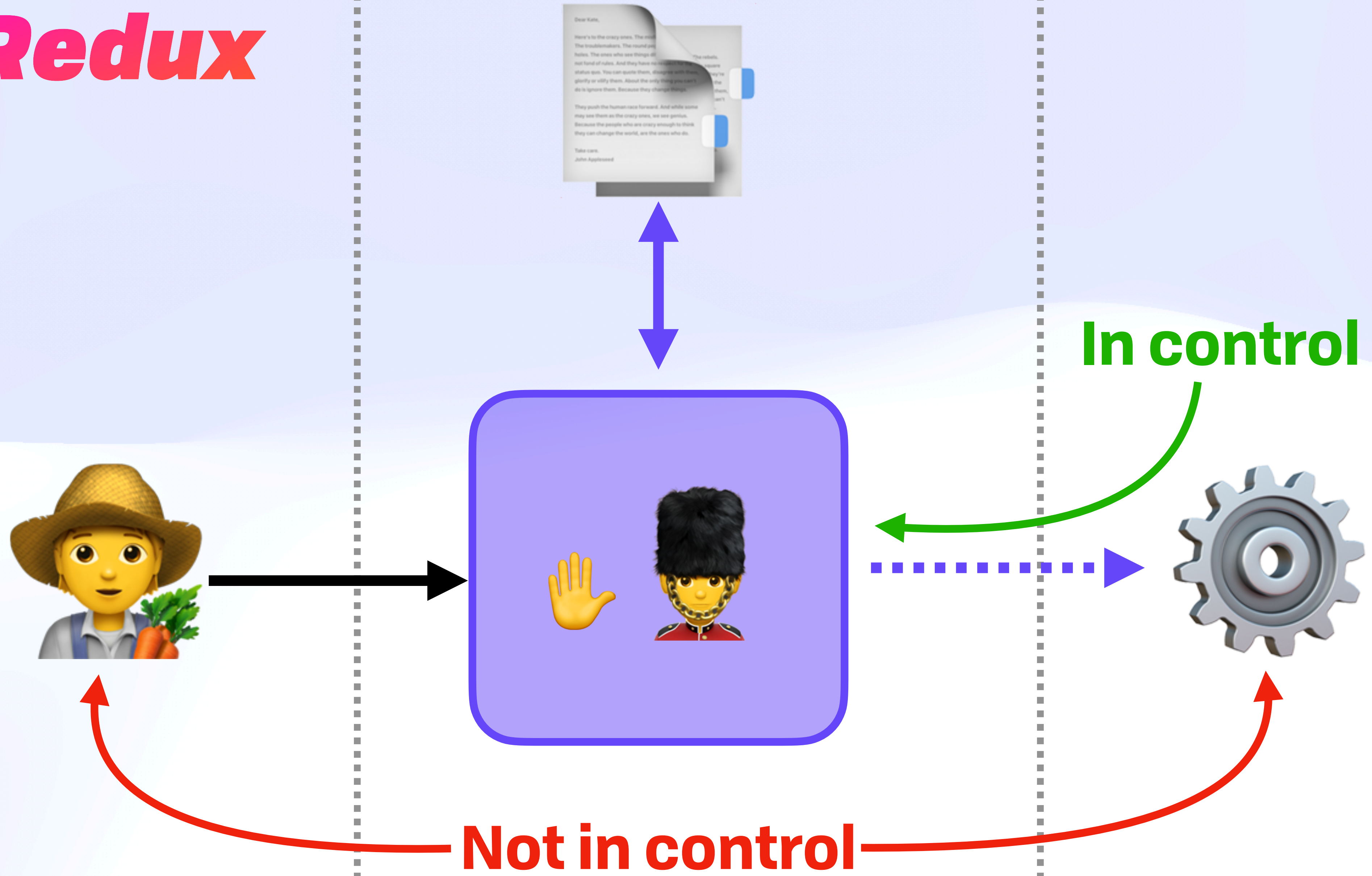
Signs of a Way Out

ACL Redux



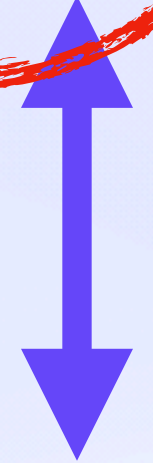
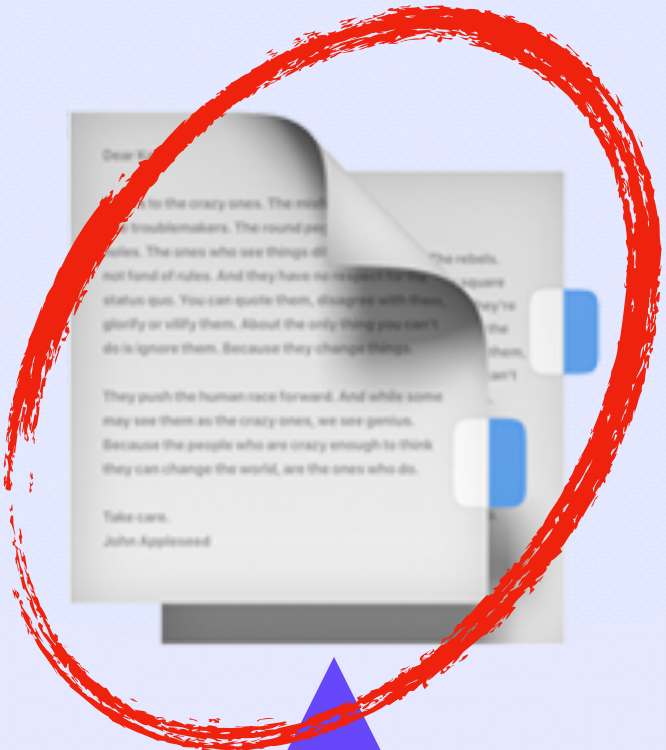
Signs of a Way Out

ACL Redux



Signs of a Way Out

ACL Redux



In control



Not in control

Signs of a Way Out

Capabilities

Signs of a Way Out

Capabilities



Signs of a Way Out

Capabilities



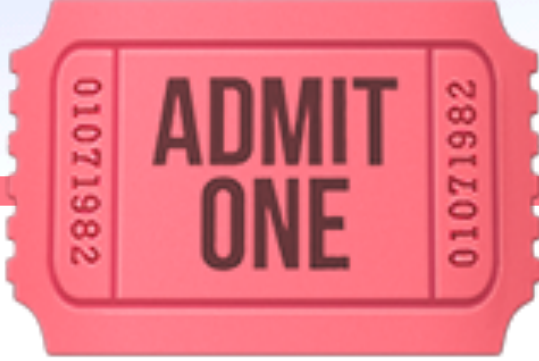
Signs of a Way Out

Capabilities



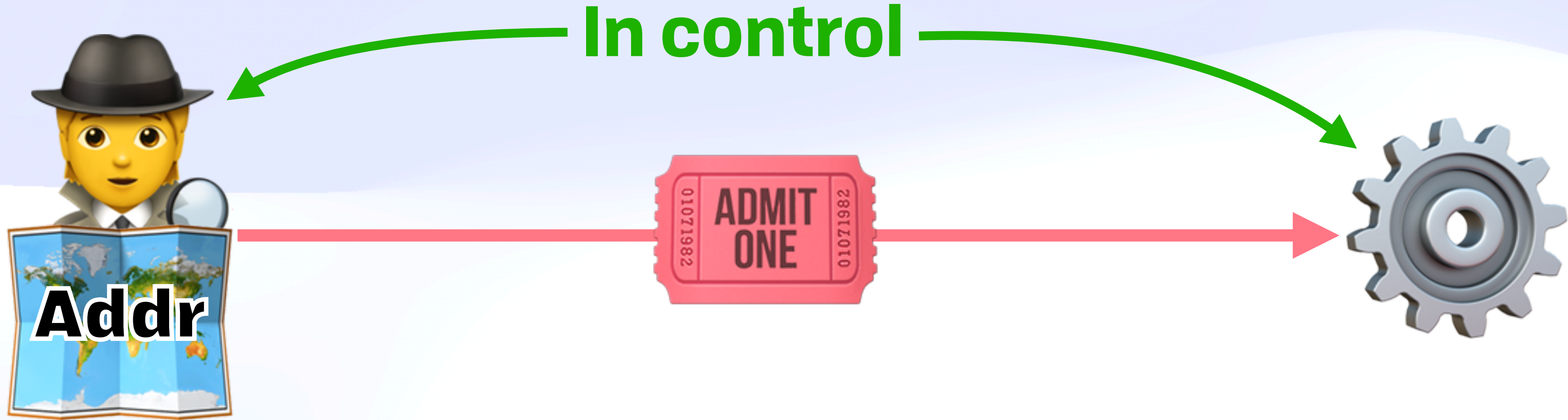
Signs of a Way Out

Capabilities



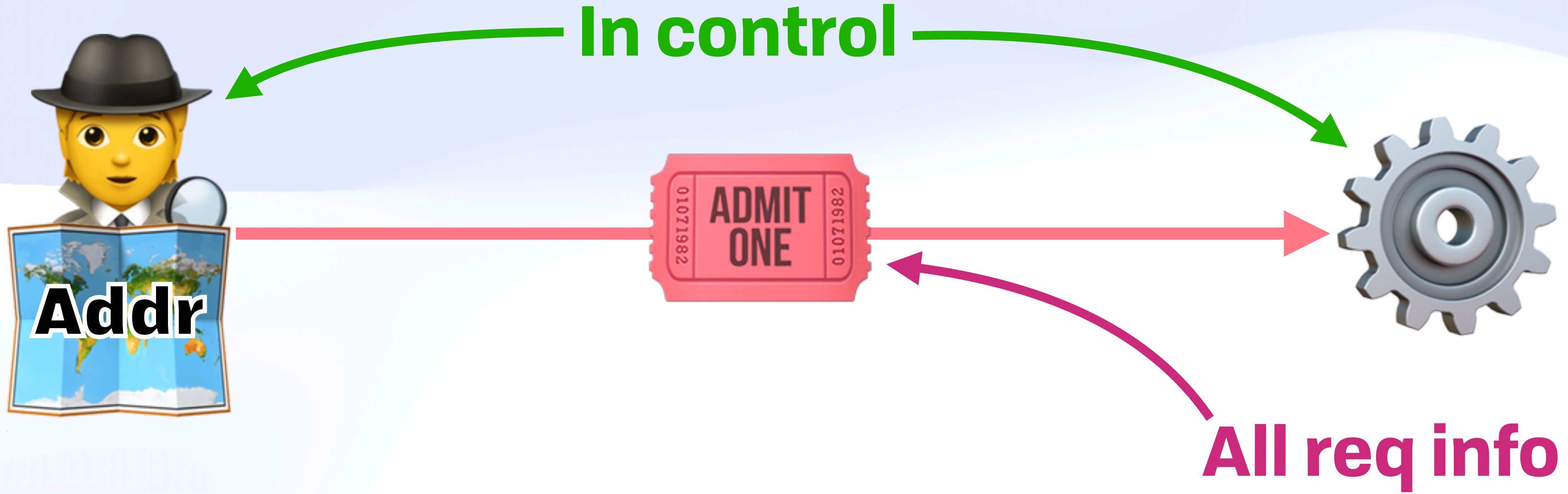
Signs of a Way Out

Capabilities



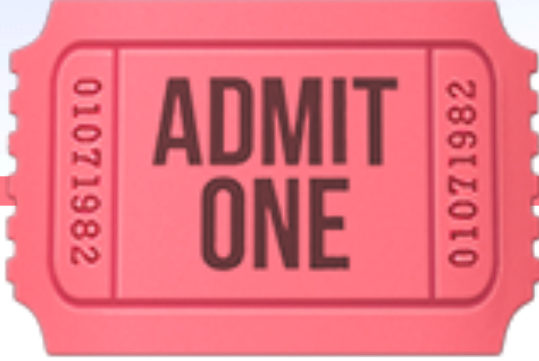
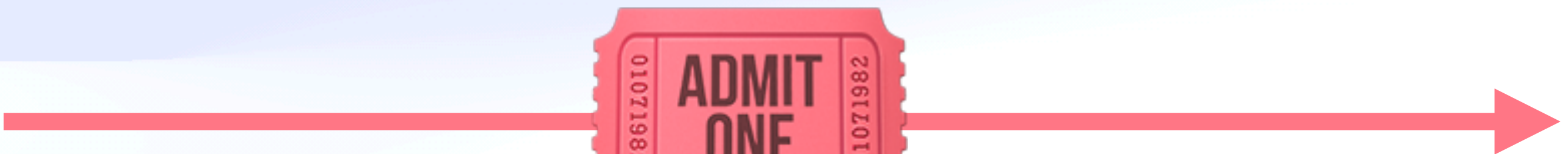
Signs of a Way Out

Capabilities



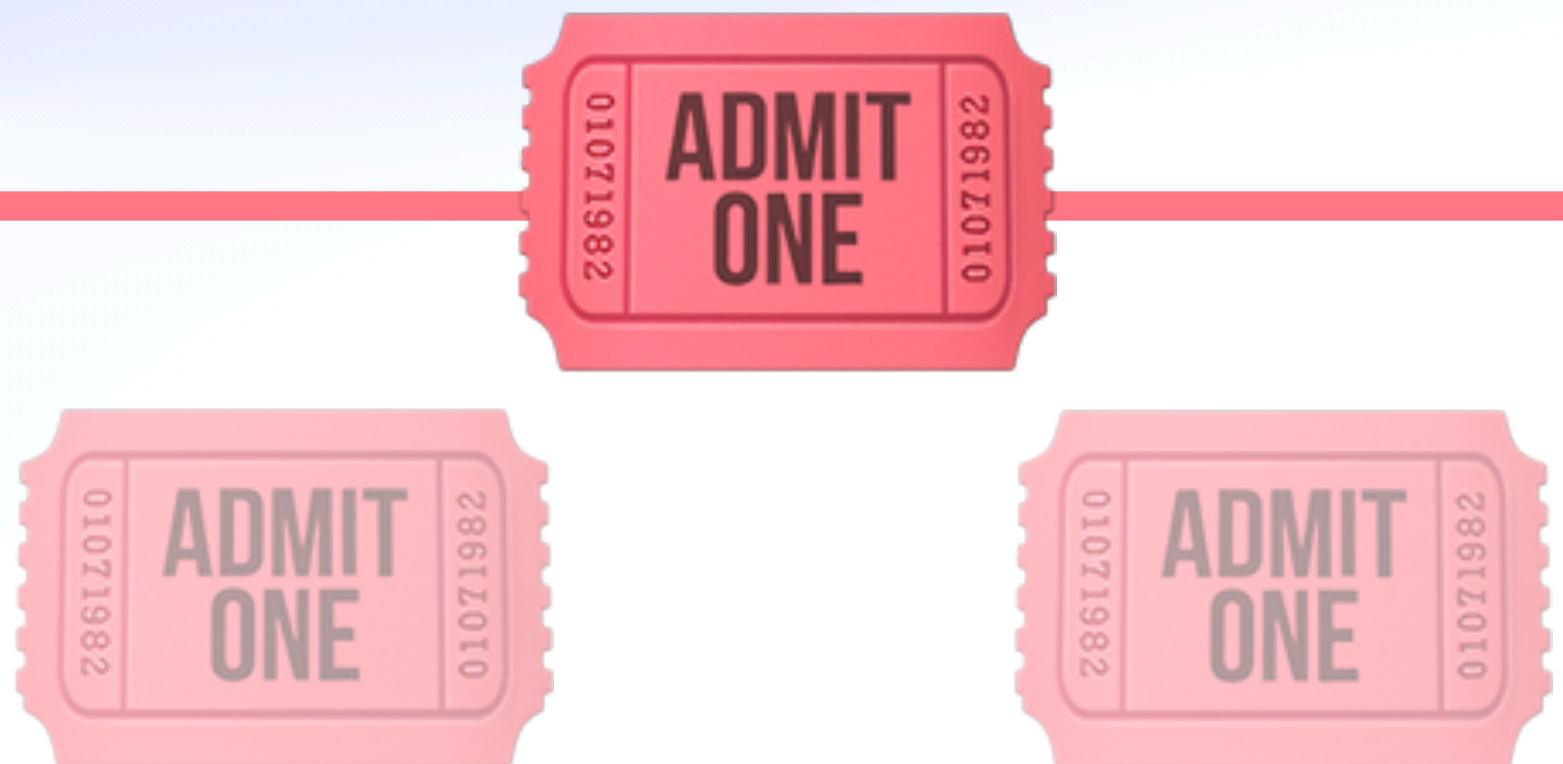
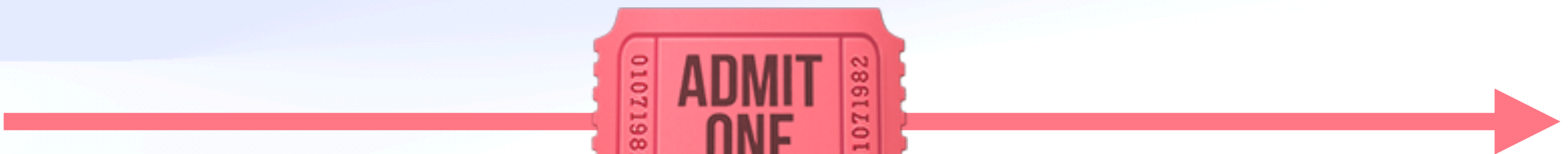
Signs of a Way Out

Capabilities



Signs of a Way Out

Capabilities



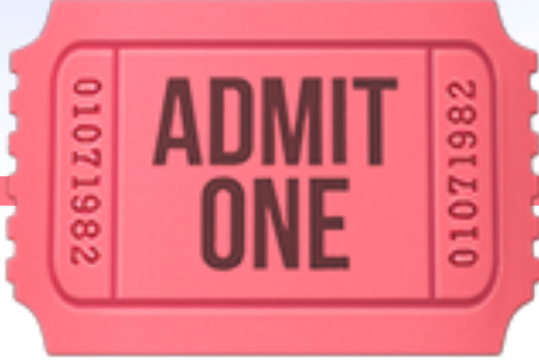
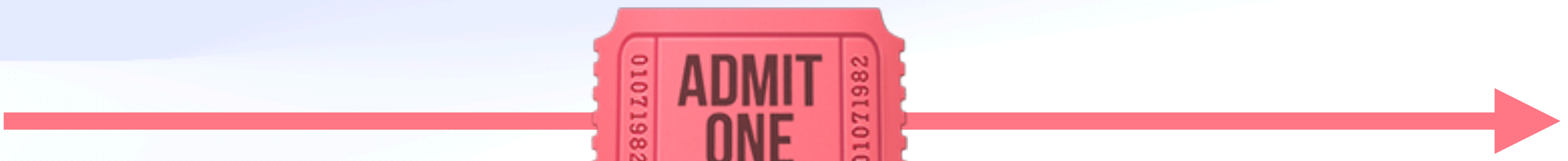
Signs of a Way Out

Capabilities



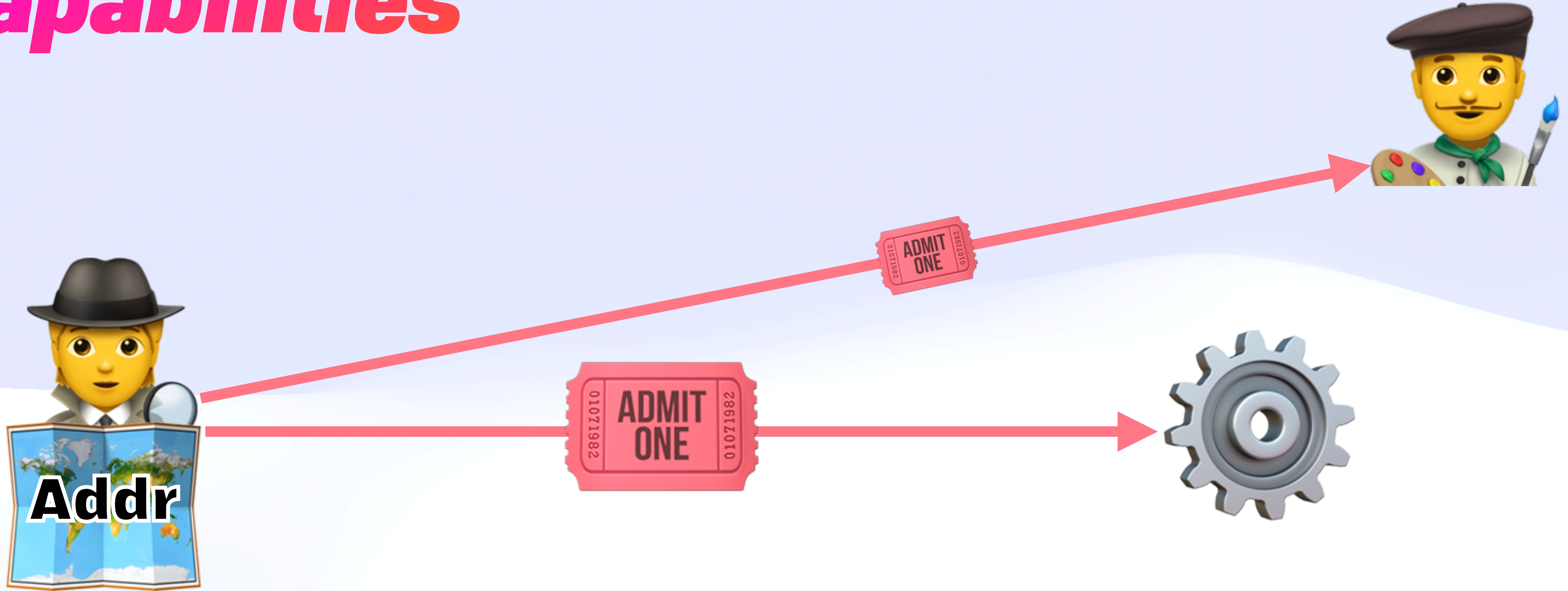
Signs of a Way Out

Capabilities



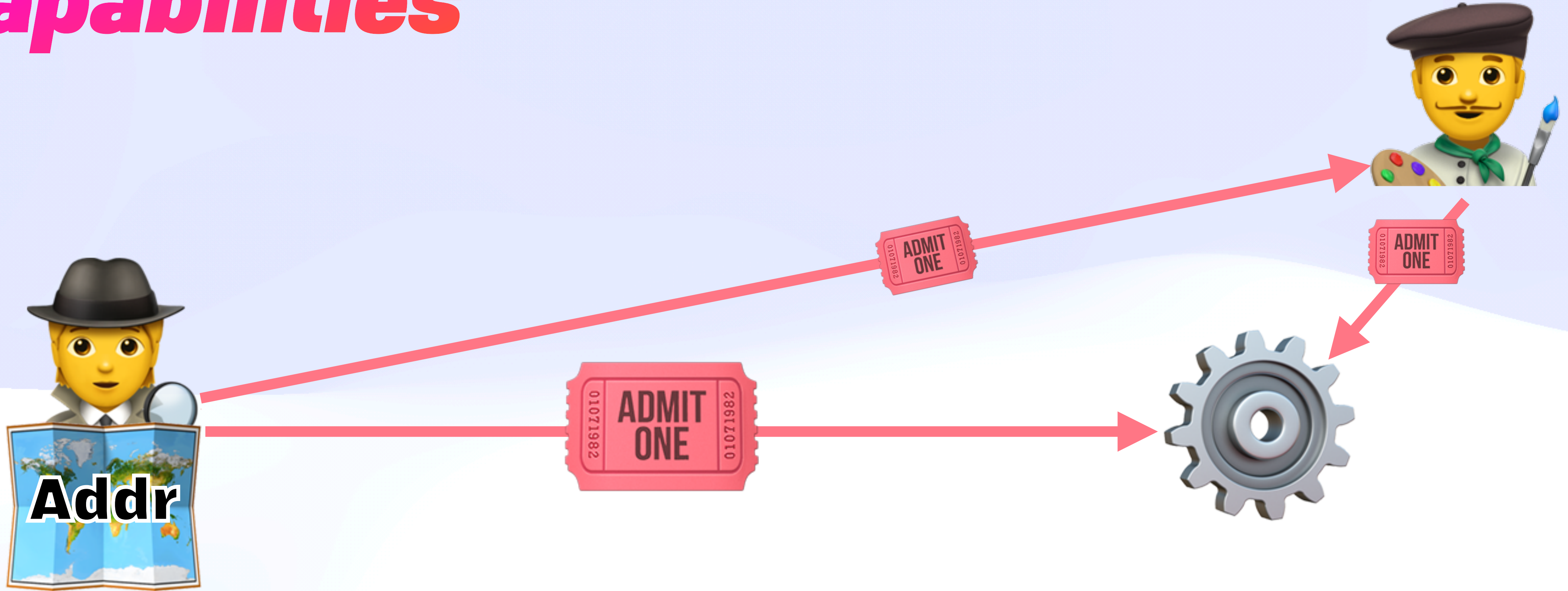
Signs of a Way Out

Capabilities



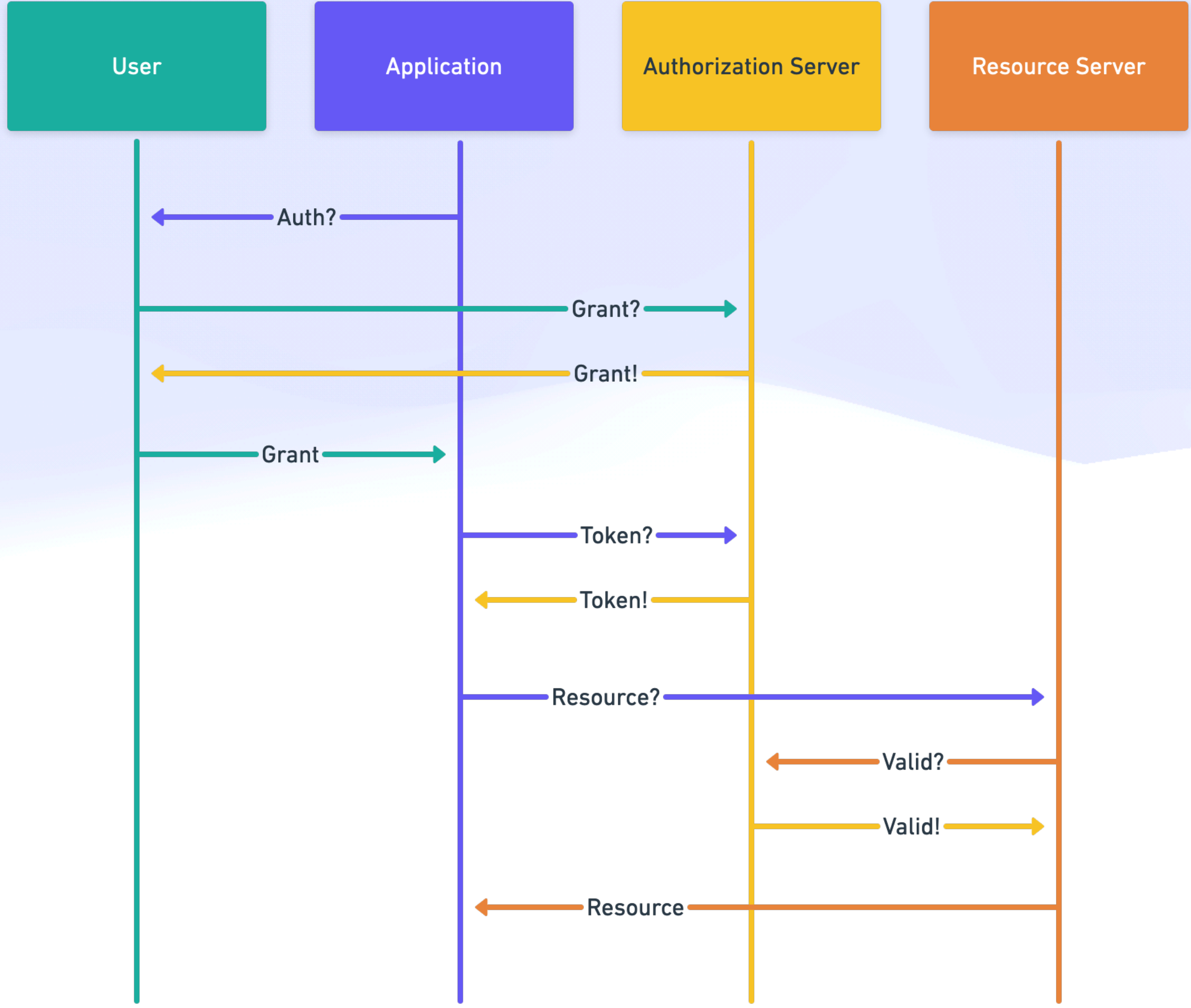
Signs of a Way Out

Capabilities



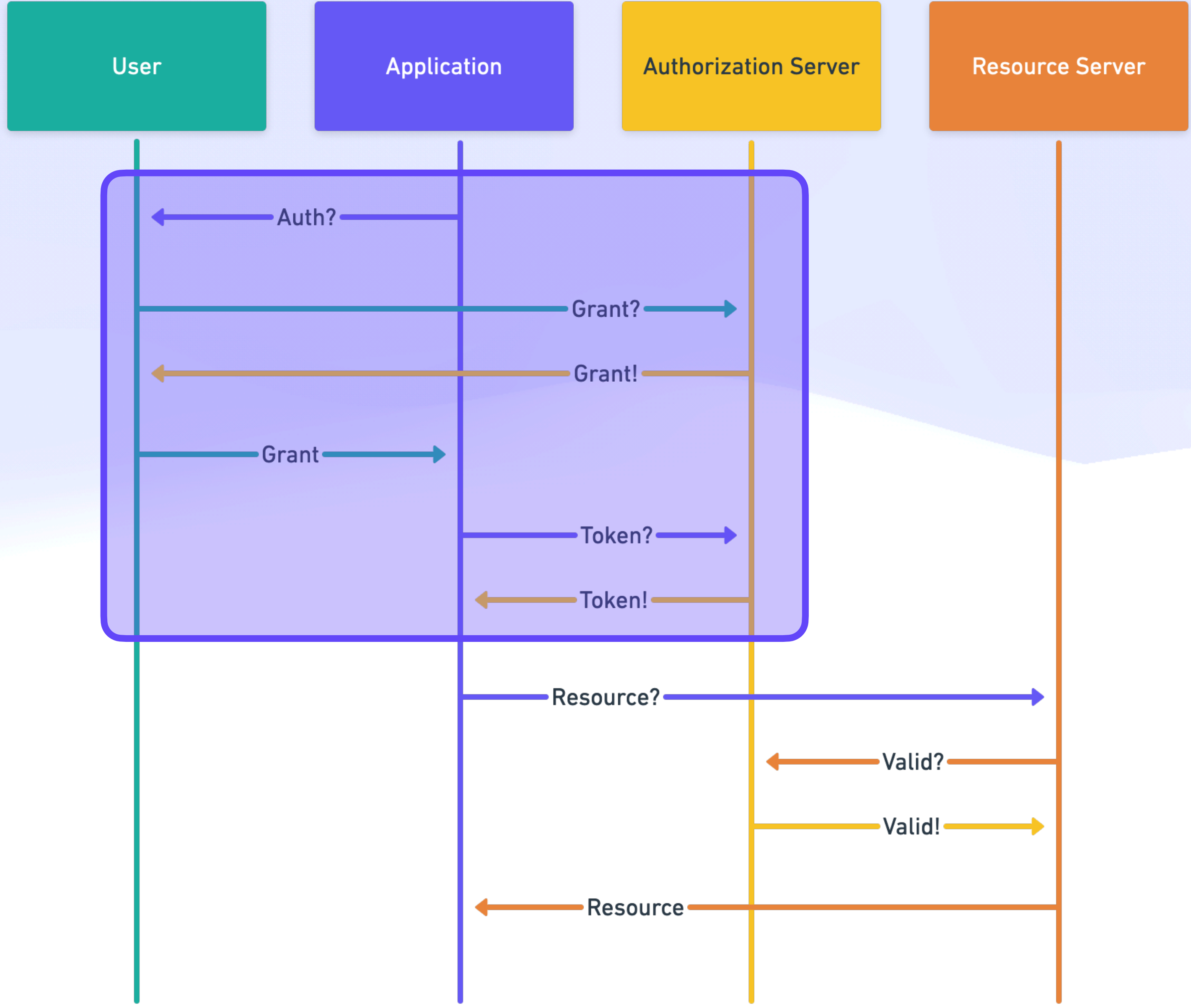
Signs of a Way Out

OAuth Sequence



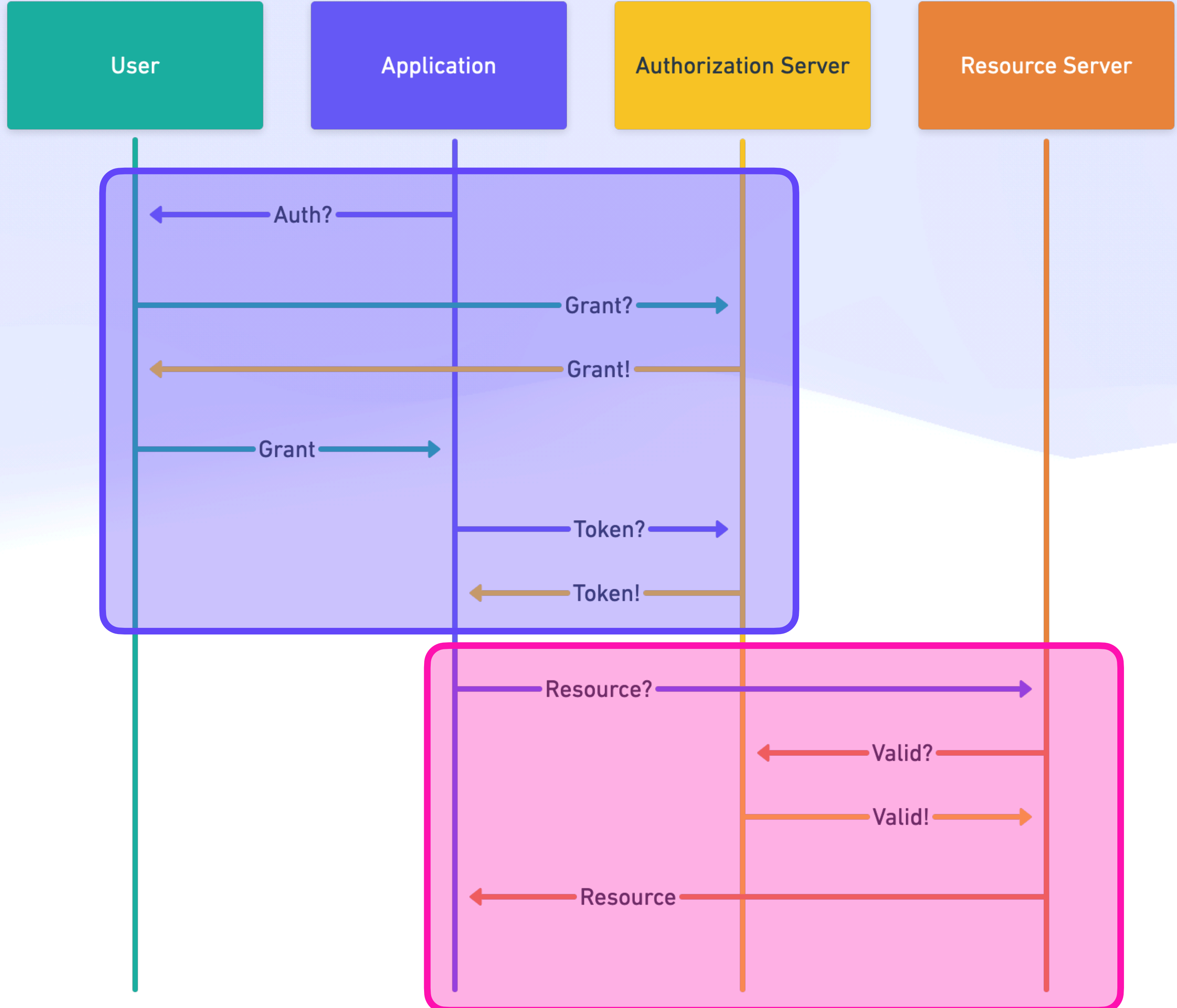
Signs of a Way Out

OAuth Sequence



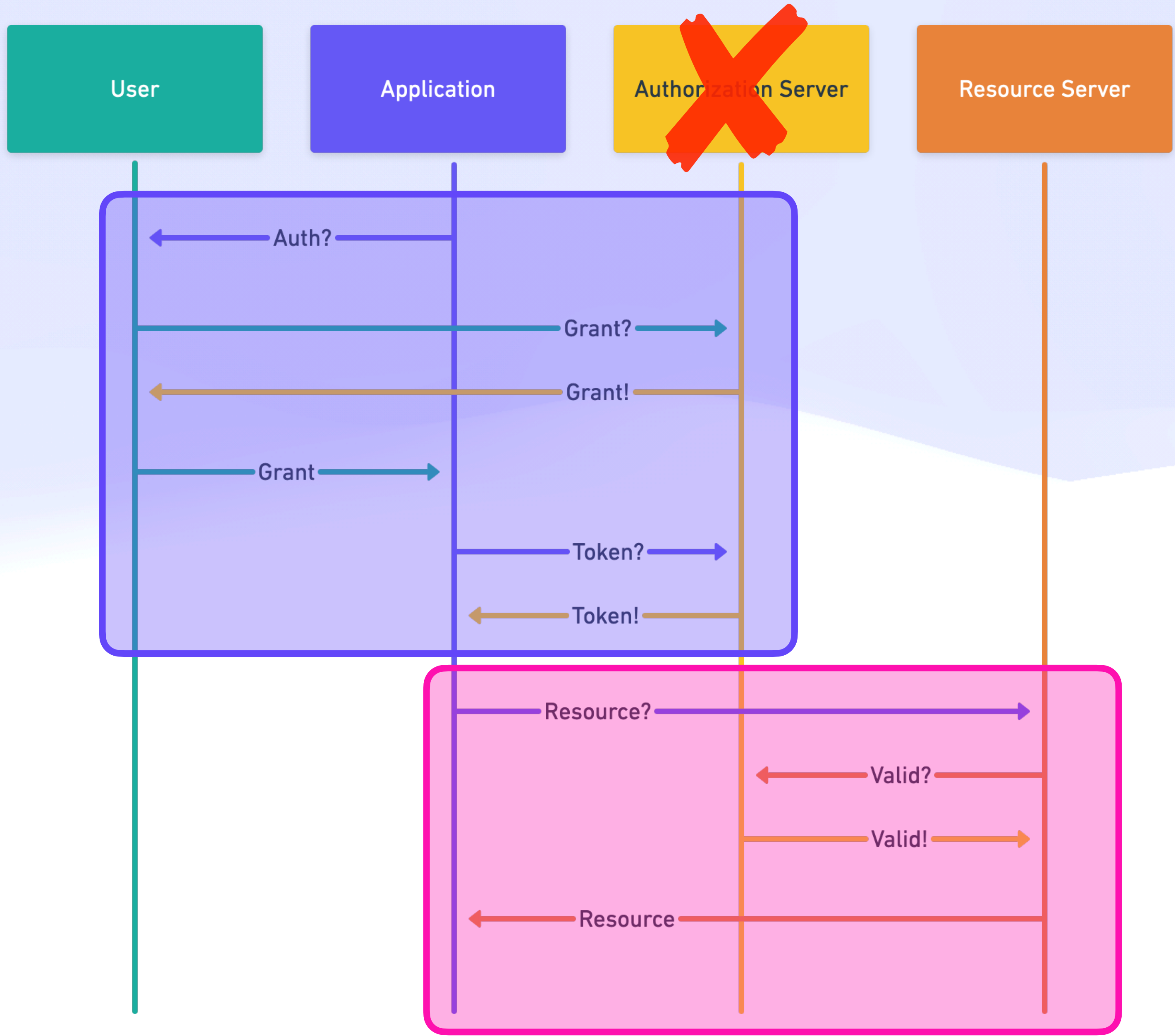
Signs of a Way Out

OAuth Sequence



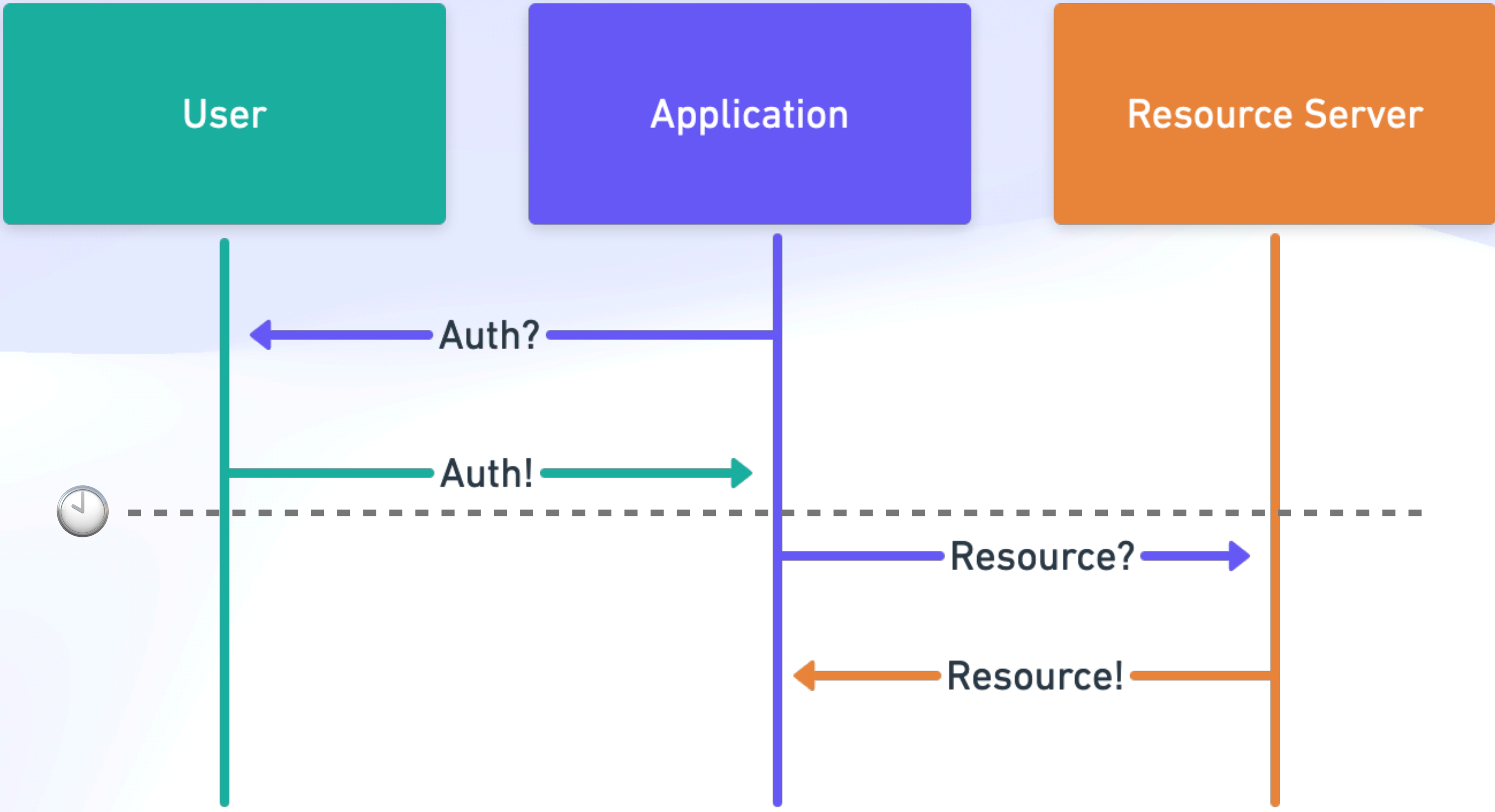
Signs of a Way Out

OAuth Sequence



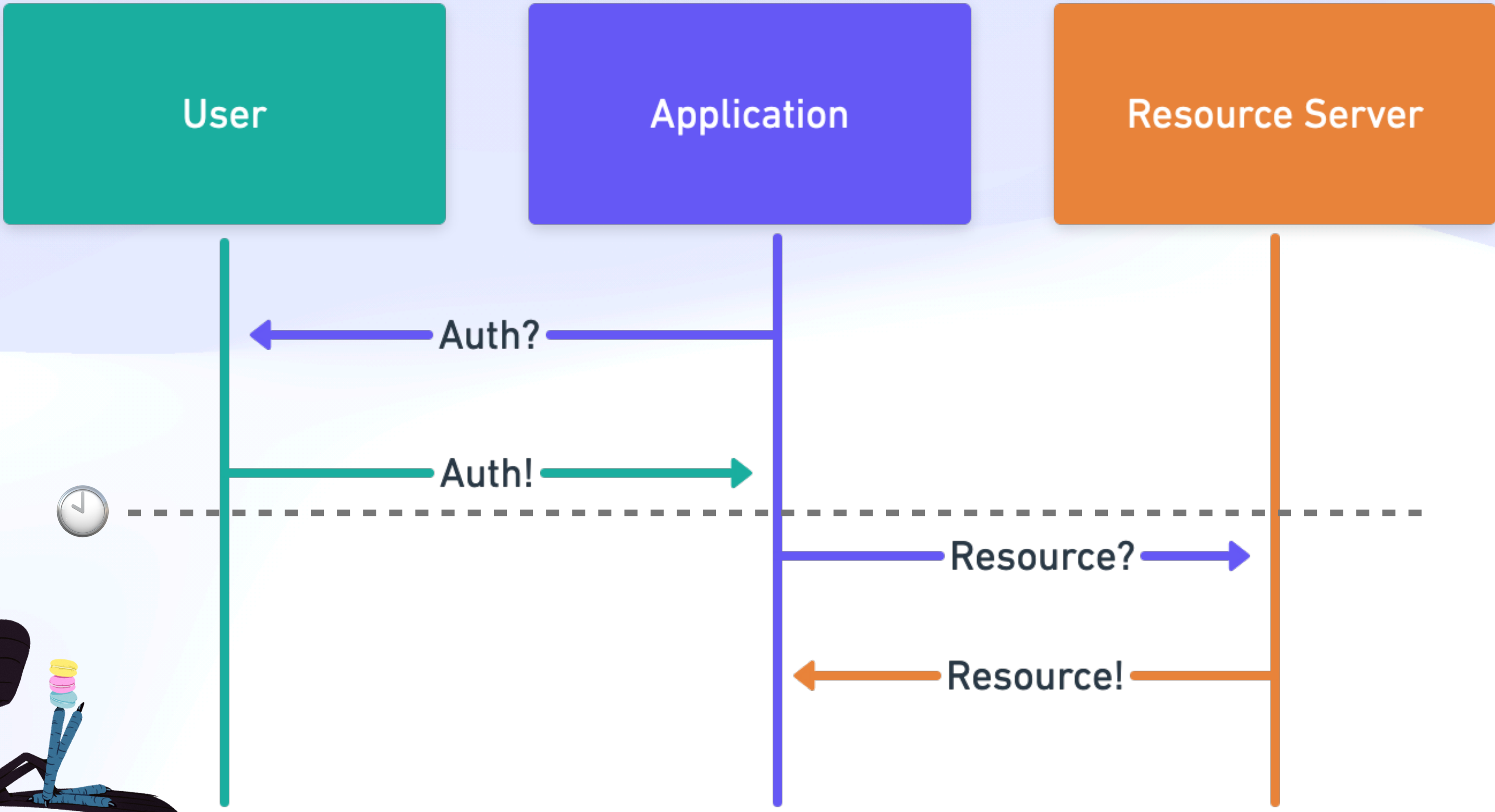
Signs of a Way Out

UCAN Sequence



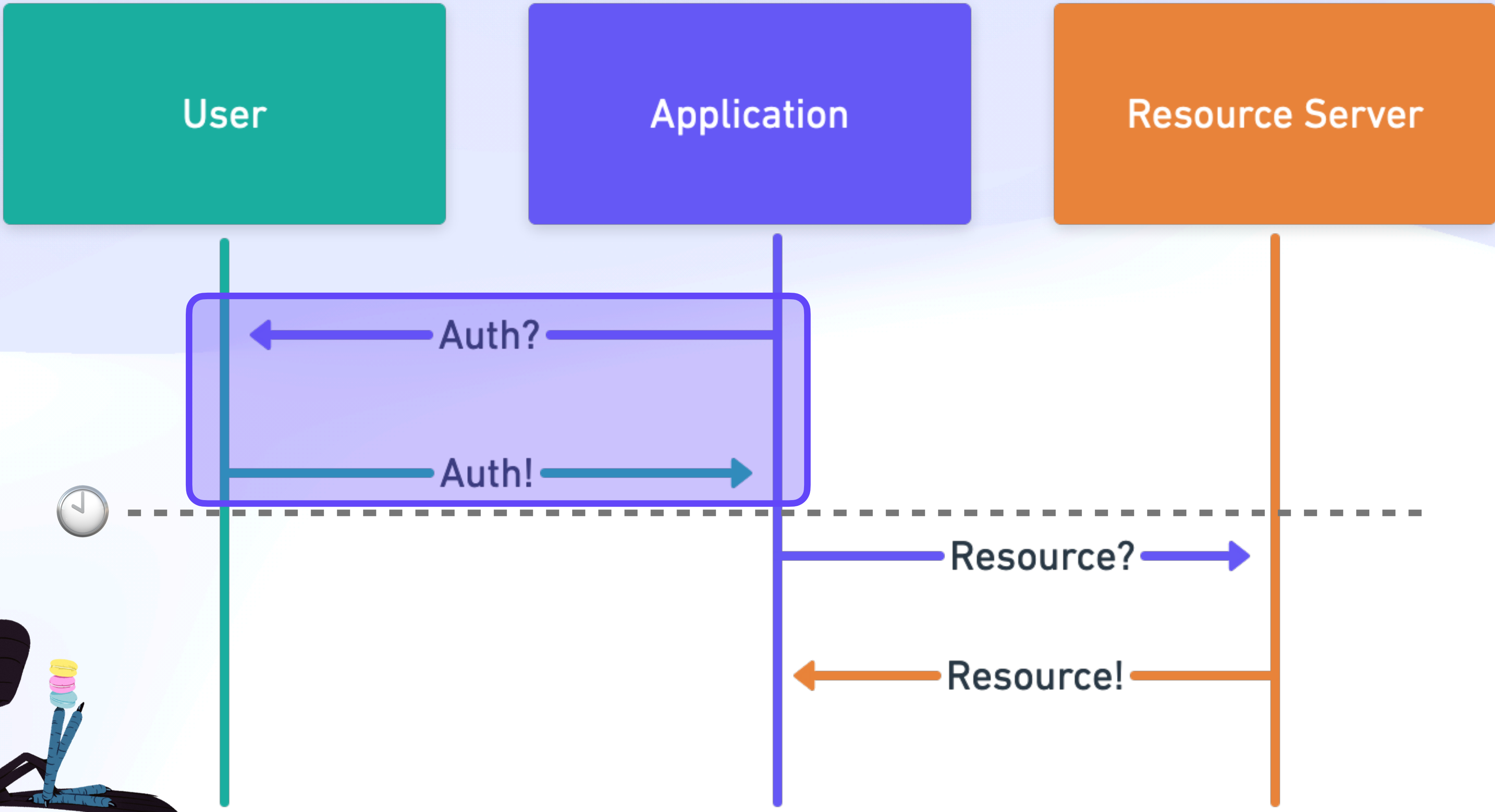
Signs of a Way Out

UCAN Sequence



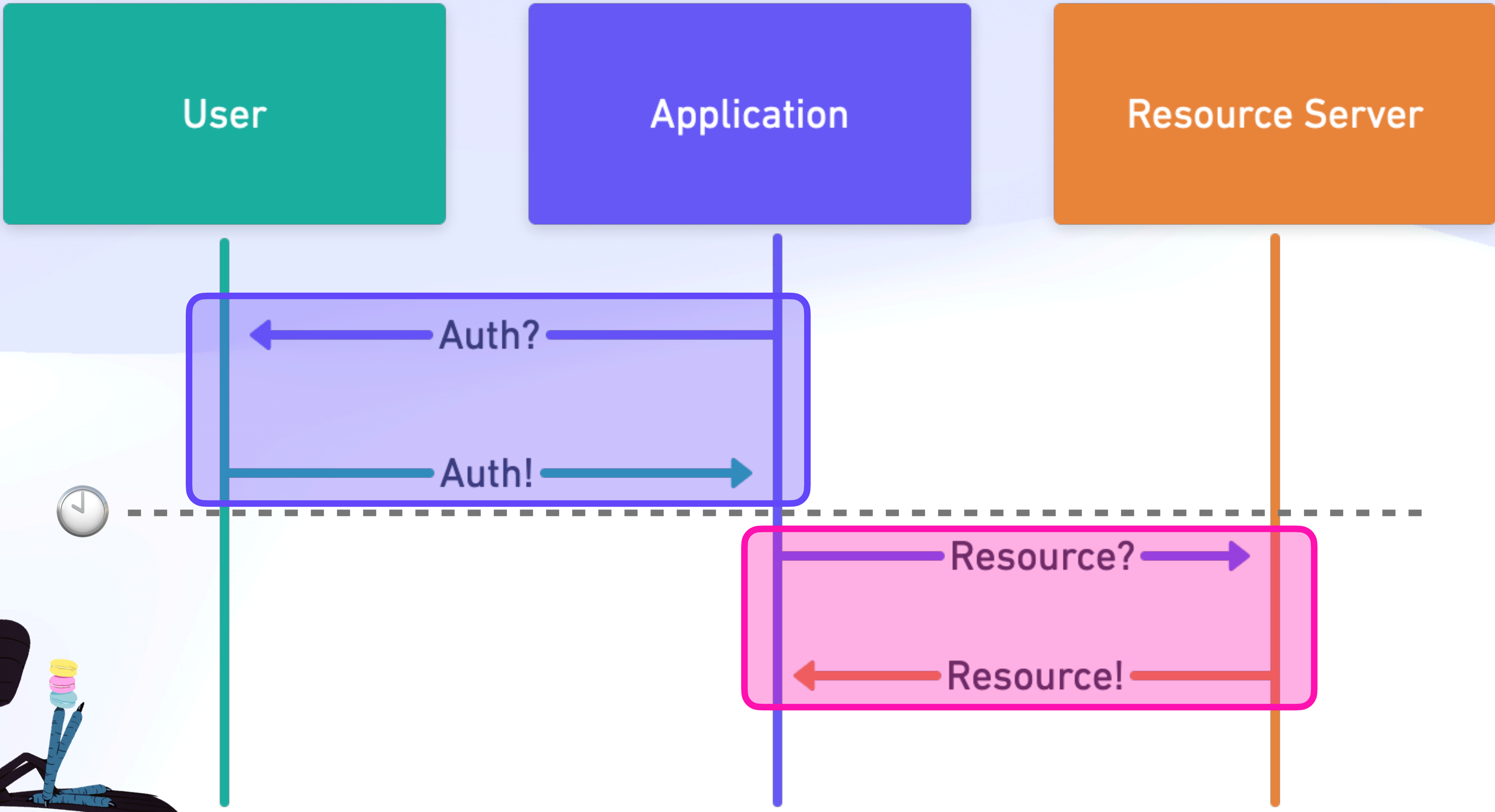
Signs of a Way Out

UCAN Sequence



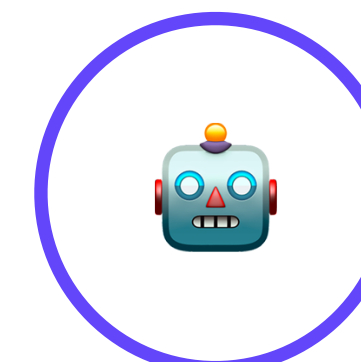
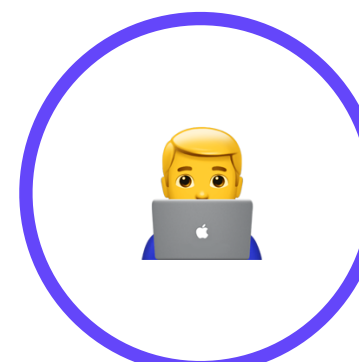
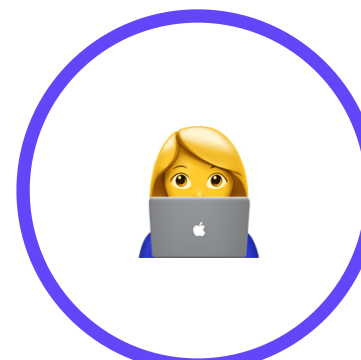
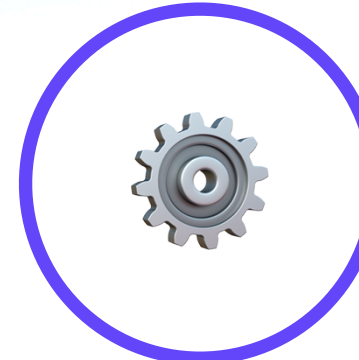
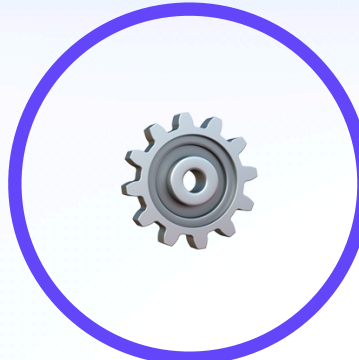
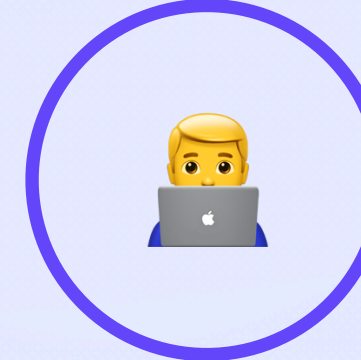
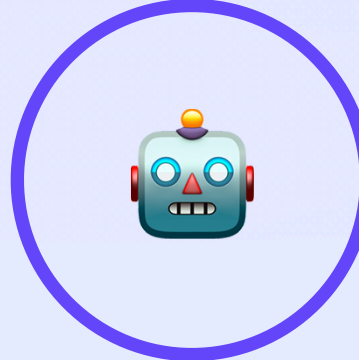
Signs of a Way Out

UCAN Sequence



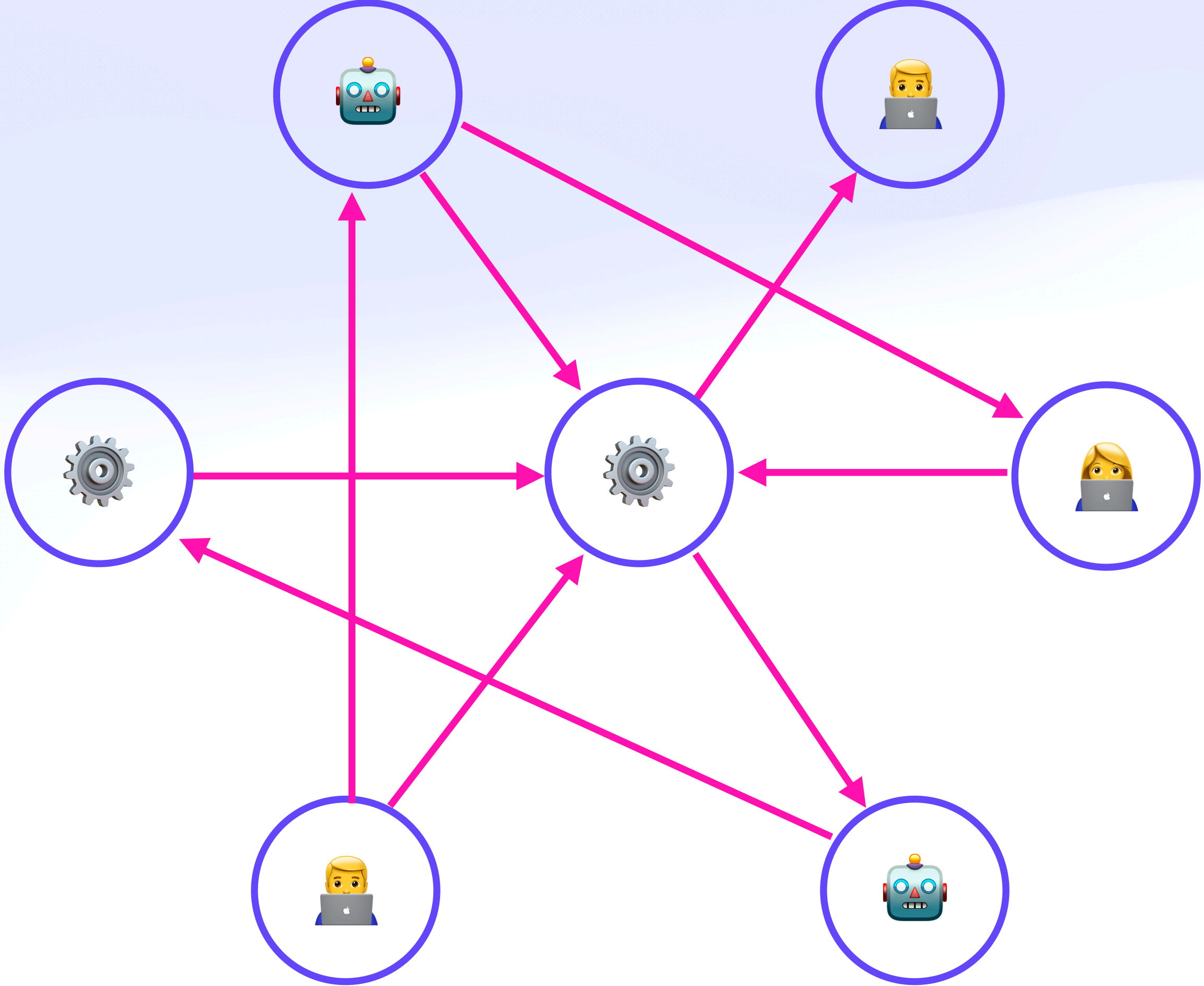
Signs of a Way Out

Disorderly Programming



Signs of a Way Out

Disorderly Programming

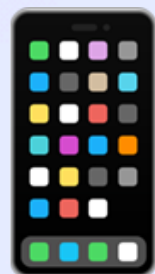


Signs of a Way Out

Don't Make Me Think

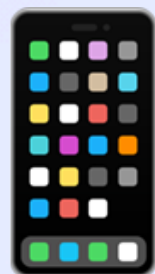
Signs of a Way Out

Don't Make Me Think



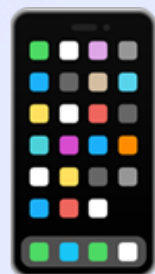
Signs of a Way Out

Don't Make Me Think



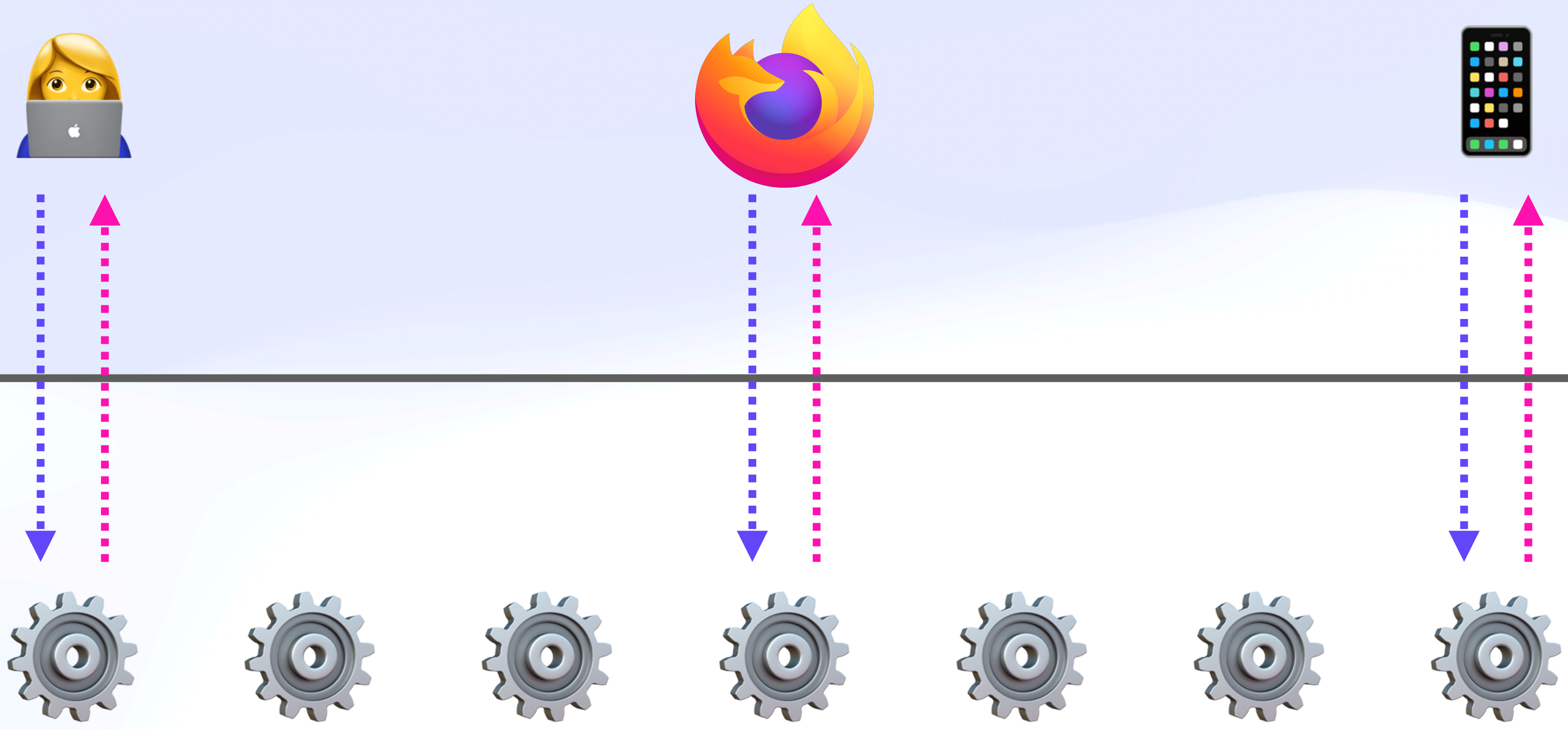
Signs of a Way Out

Don't Make Me Think



Signs of a Way Out

Don't Make Me Think





A computer monitor is shown from a three-quarter perspective, with a large, dynamic splash of multi-colored liquid (pink, orange, teal, and yellow) erupting from its screen. The splash is highly detailed with various droplets and splatters, creating a sense of motion and energy. The background is a solid, light teal color.

Compute Substrate

— Schedule. Execute. Verify. Reuse —

Compute Substrate

Wasm Eats the World

Compute Substrate

Wasm Eats the World



Compute Substrate

Wasm Eats the World



WA

Bring your own language

Runs everywhere

Deterministic mode

Compute Substrate


With Their Powers Combined!

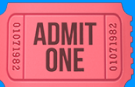


Compute Substrate

With Their Powers Combined!

Compute 


Data 


Auth 



Compute Substrate

With Their Powers Combined!

Compute 

Data 

Auth 


Everywhere!




Compute Substrate

With Their Powers Combined!

Compute 

Data 

Auth 


Everywhere!



Compute Substrate

With Their Powers Combined!

Compute 

Data 

Auth 

Everywhere!

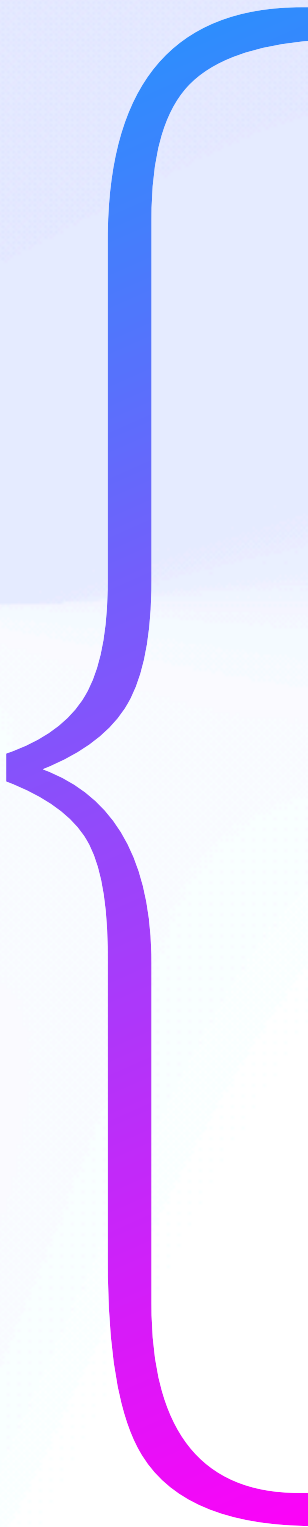


Compute Substrate


With Their Powers Combined!


Everywhere!

IPVM



Compute 

Data 

Auth 



Compute Substrate

Code-as-Data

Compute Substrate

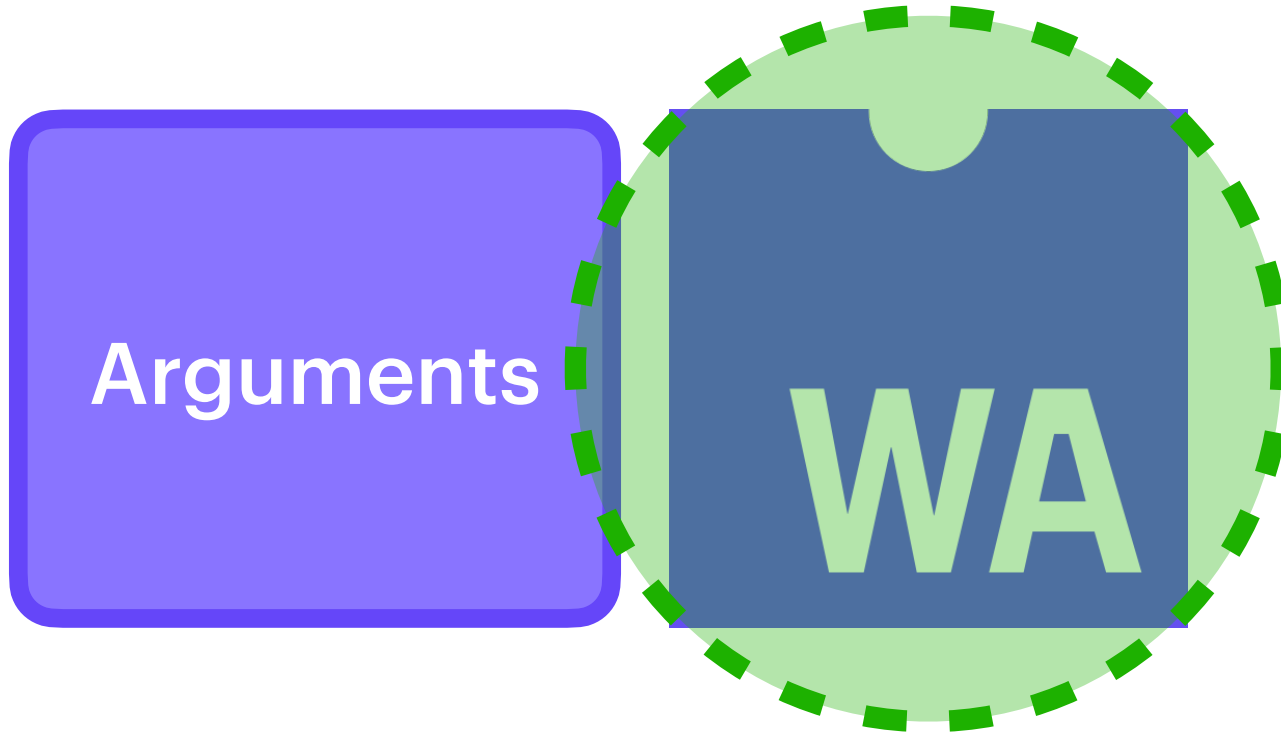
Code-as-Data

Arguments

WA

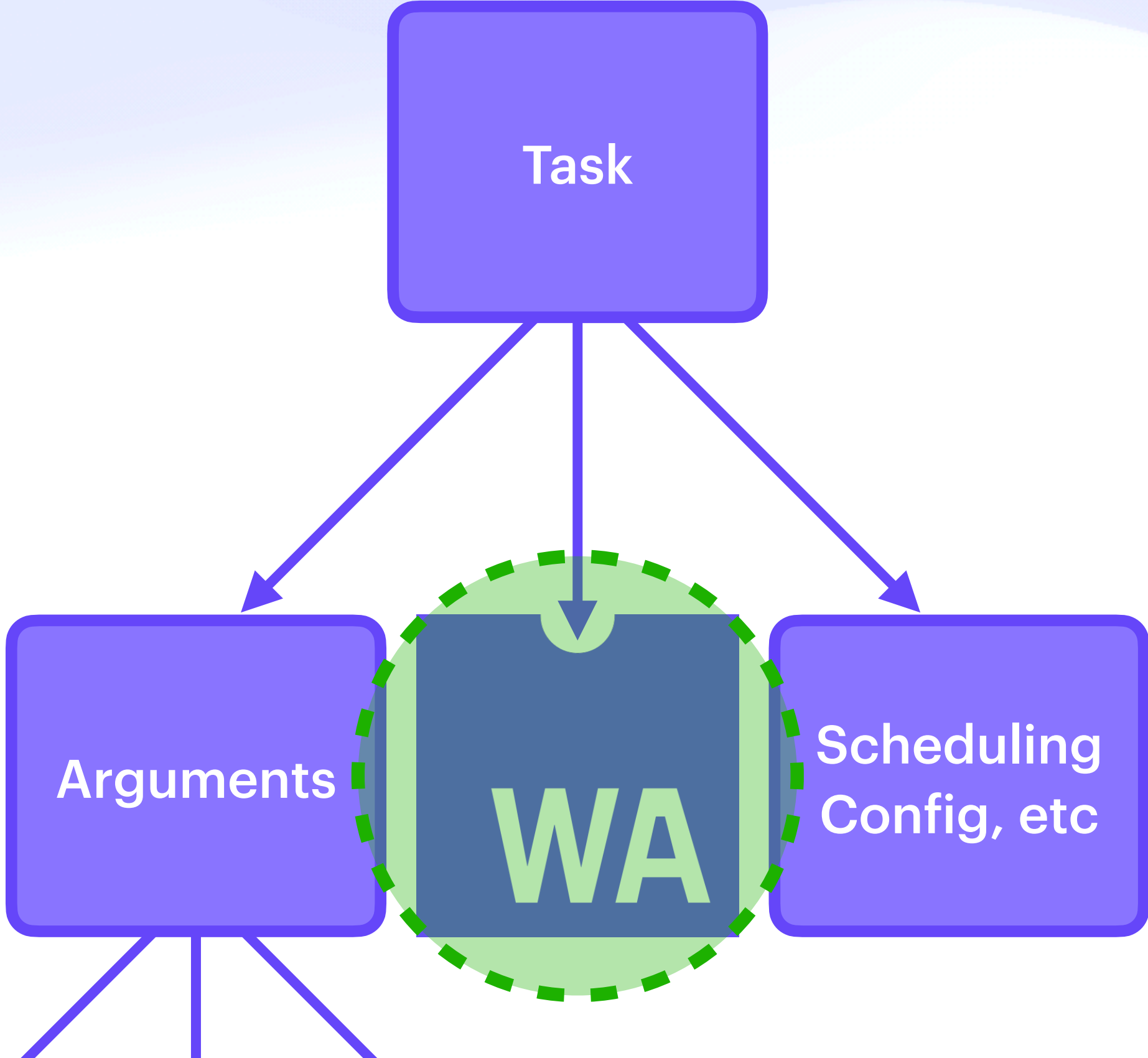
Compute Substrate

Code-as-Data



Compute Substrate

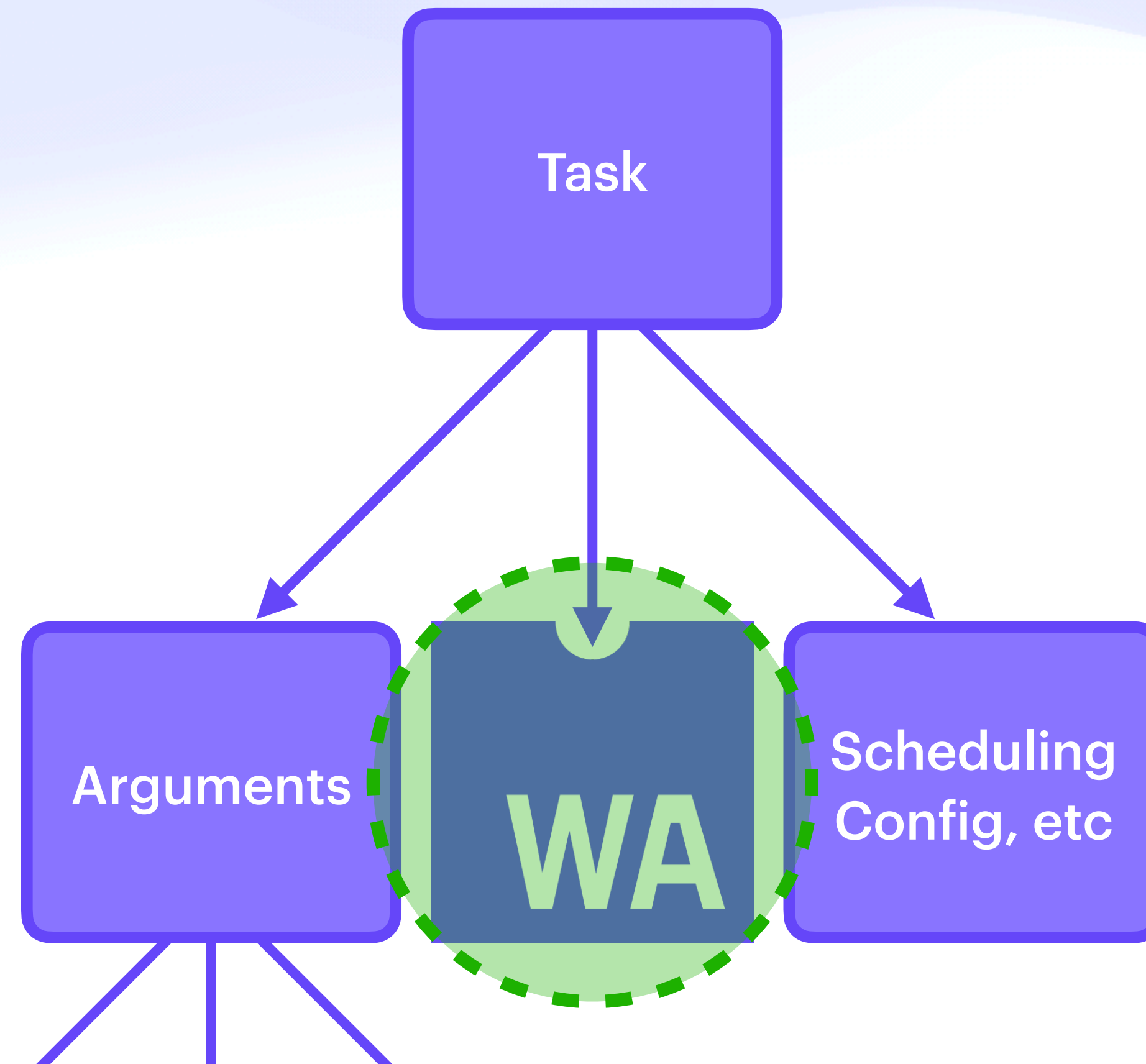
Code-as-Data



Compute Substrate

Code-as-Data

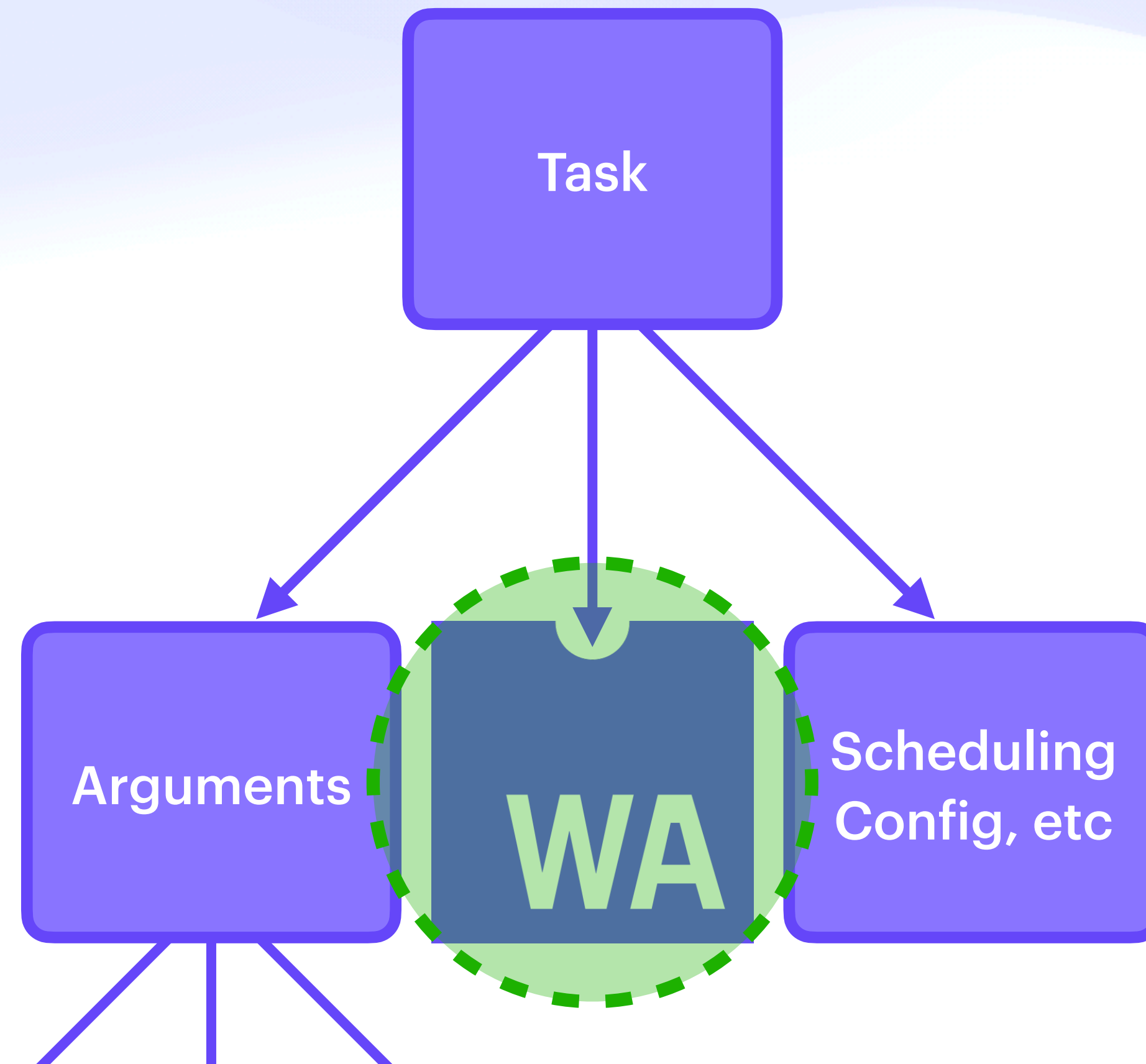
```
const message = () => alert("hello world")
```



Compute Substrate

Code-as-Data

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



Compute Substrate

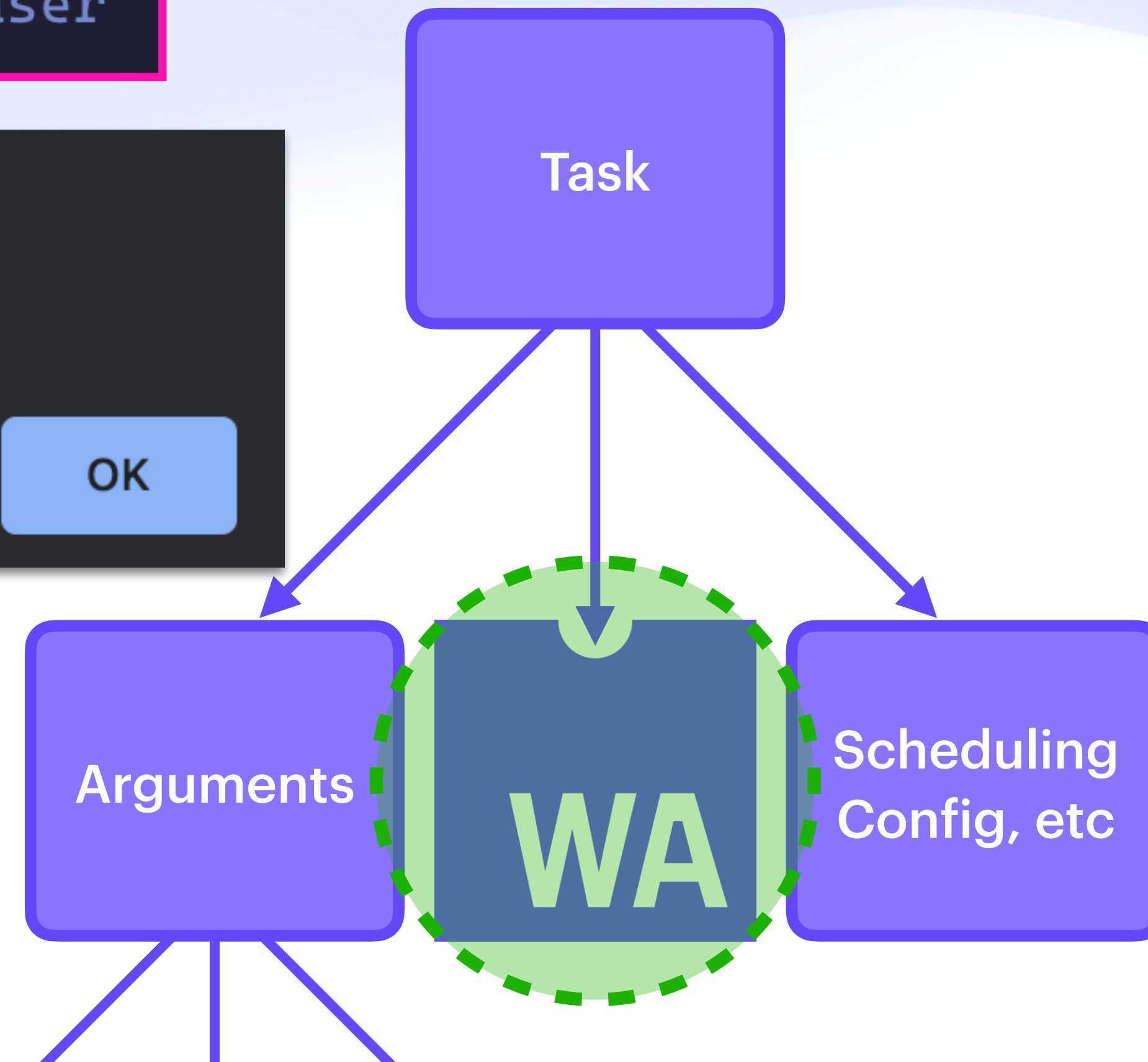
Code-as-Data

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

2023.ipfs-thing.io says

hello world

OK



Compute Substrate

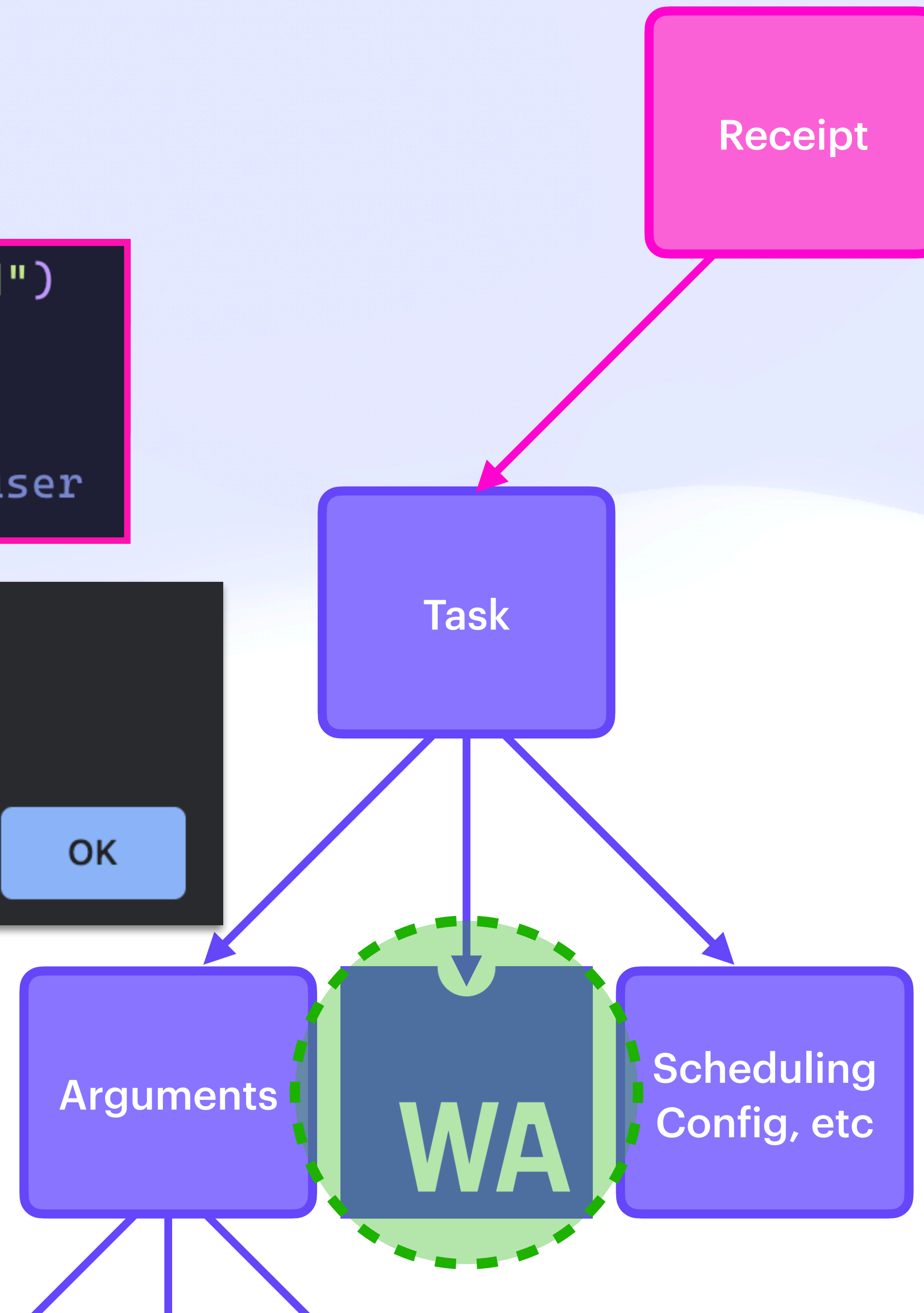
Code-as-Data

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

2023.ipfs-thing.io says

hello world

OK



Compute Substrate

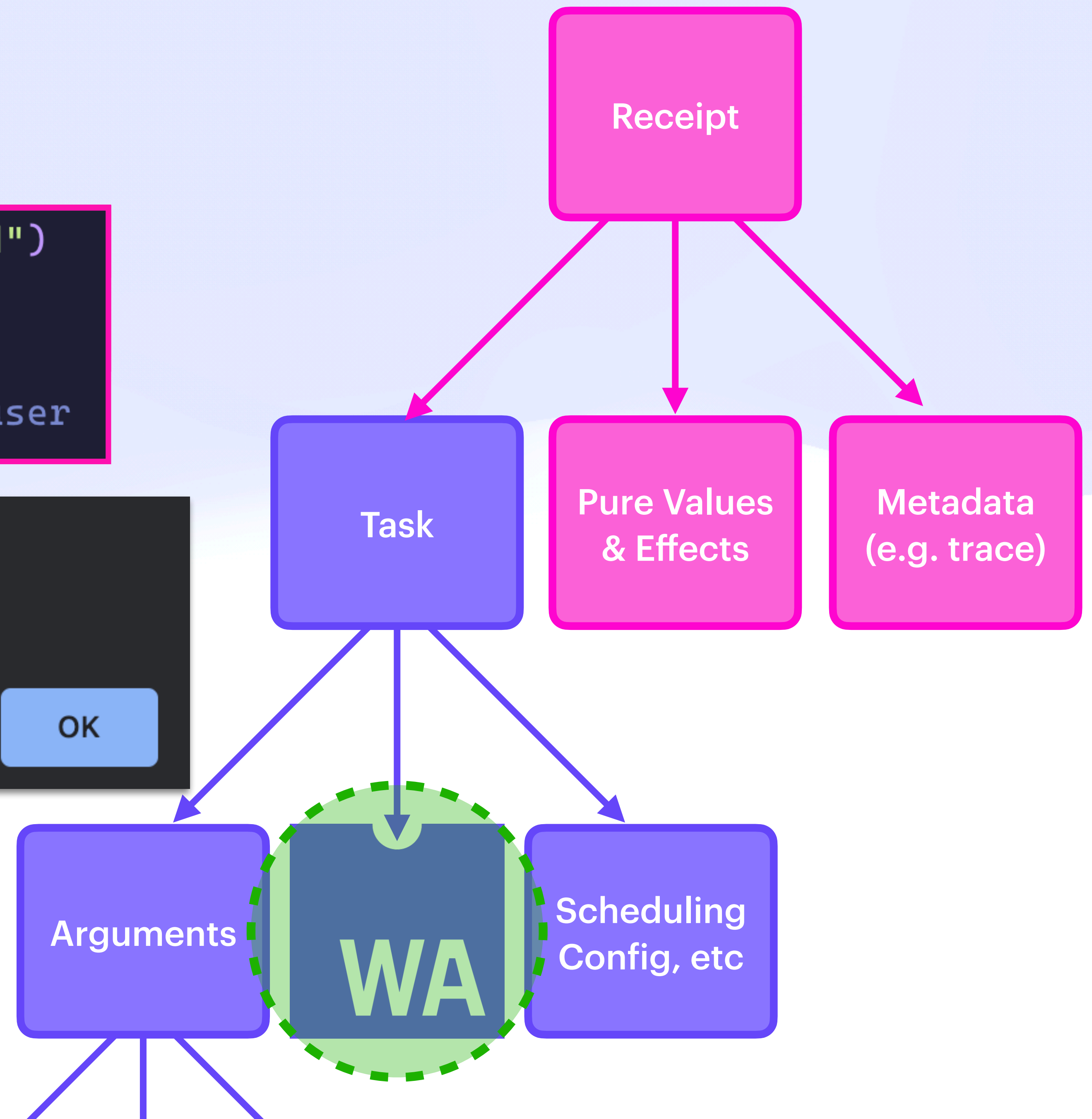
Code-as-Data

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

2023.ipfs-thing.io says

hello world

OK



Compute Substrate

Input Addressing

Compute Substrate

Input Addressing



Compute Substrate

Input Addressing



Compute Substrate

Input Addressing

hash(🦞)



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

Compute Substrate

Input Addressing

hash(🦞)



```
hash({  
  rsc: "dns:example.com"  
  op: "crud/update"  
  input: {foo: "bar"}  
})
```



Compute Substrate

IPLD Schema

Compute Substrate

IPLD Schema

```
type Instruction struct {  
  rsc      URI  
  op       Ability  
  input    {String : Any}  
  nnc      String  
}
```

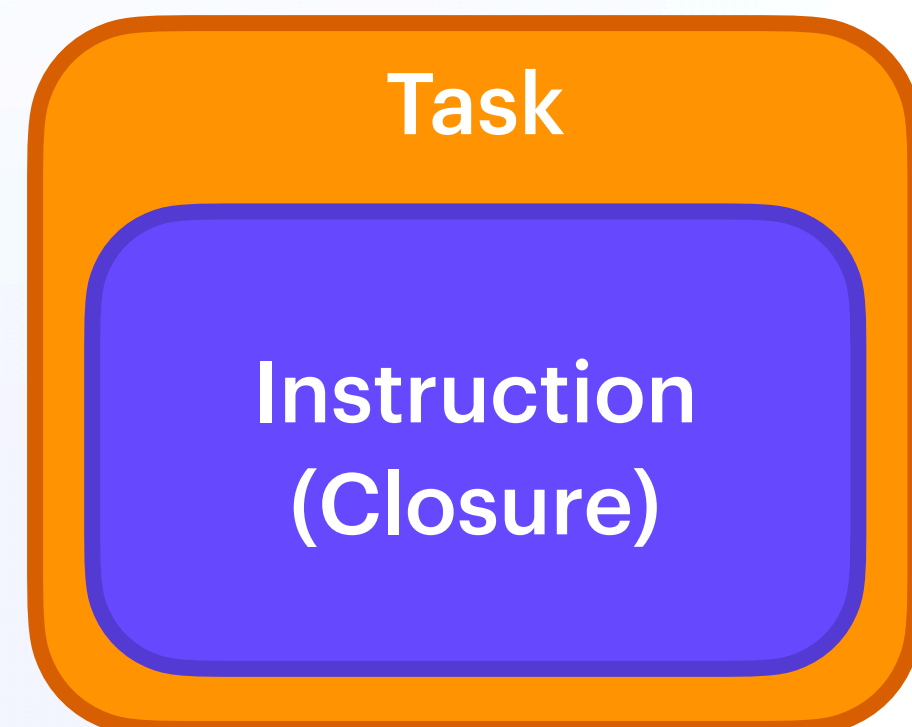
Instruction
(Closure)

Compute Substrate

IPLD Schema

```
type Instruction struct {  
  rsc      URI  
  op       Ability  
  input    {String : Any}  
  nnc      String  
}
```

```
type Task struct {  
  run      &Instruction  
  meta     {String : Any}  
  prf      [&UCAN]  
  cause    optional &Receipt  
}
```



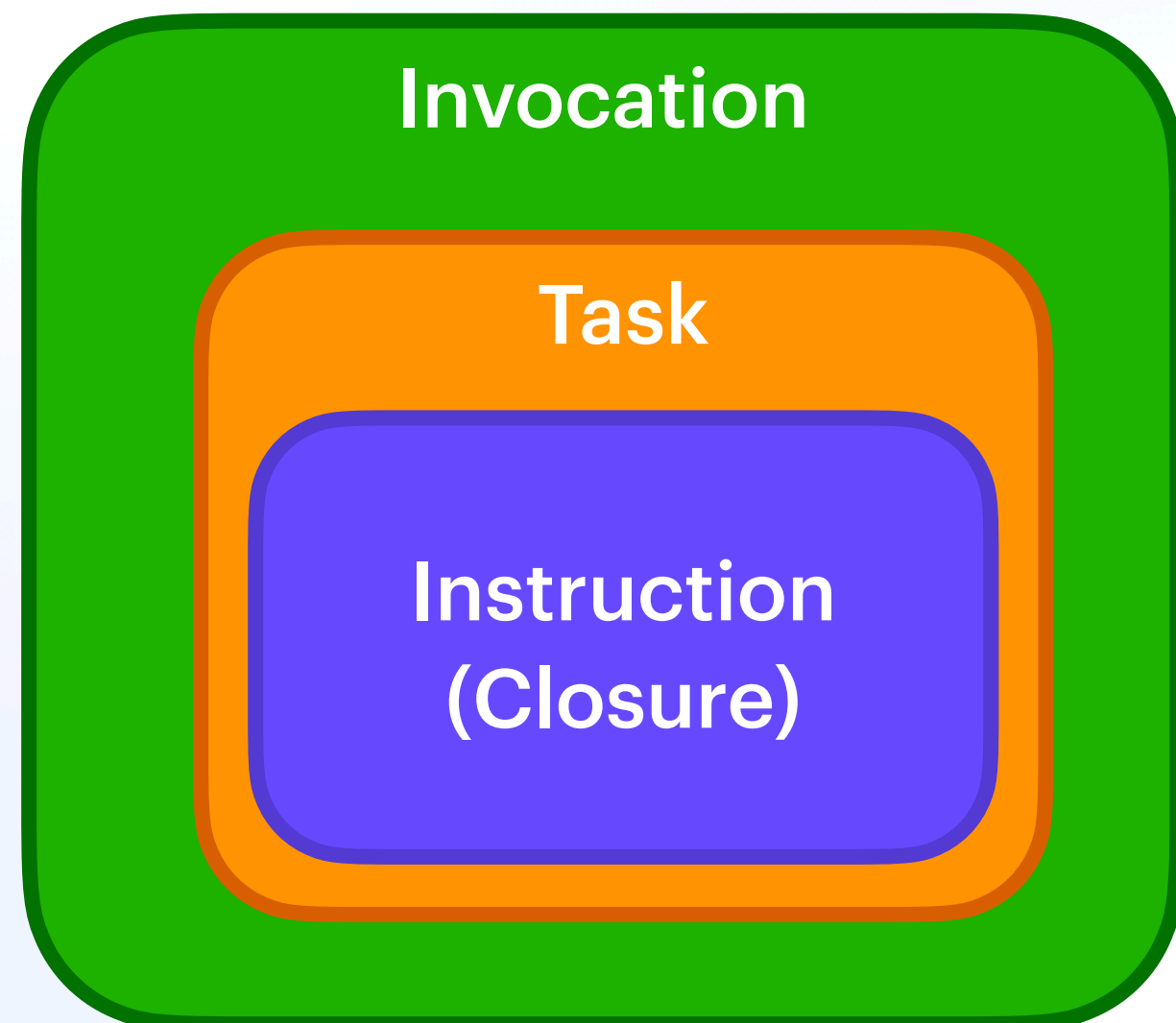
Compute Substrate

IPLD Schema

```
type Instruction struct {  
  rsc      URI  
  op       Ability  
  input    {String : Any}  
  nnc      String  
}
```

```
type Task struct {  
  run      &Instruction  
  meta     {String : Any}  
  prf      [&UCAN]  
  cause    optional &Receipt  
}
```

```
type Invocation struct {  
  task     Task  
  auth     &Authorization  
}
```



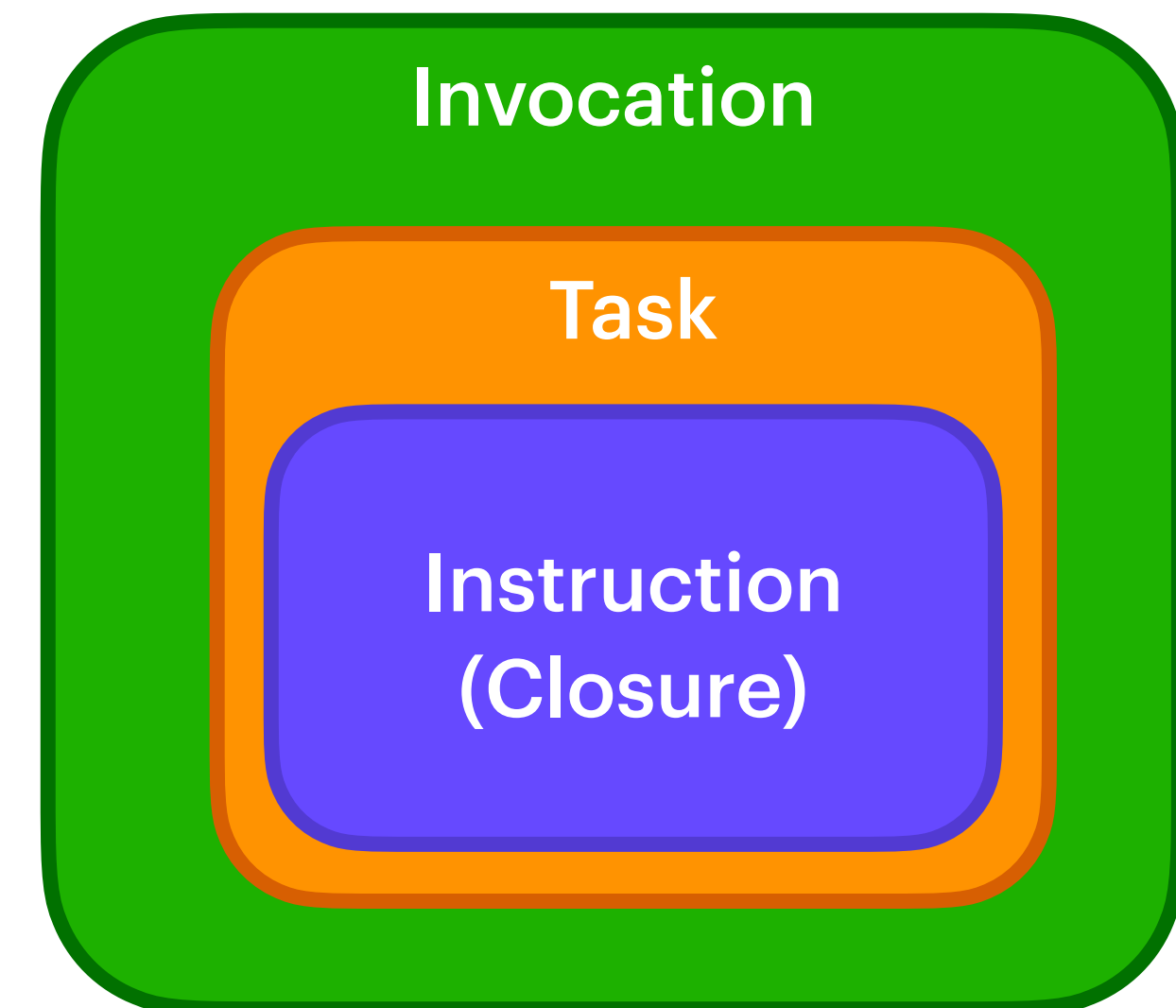
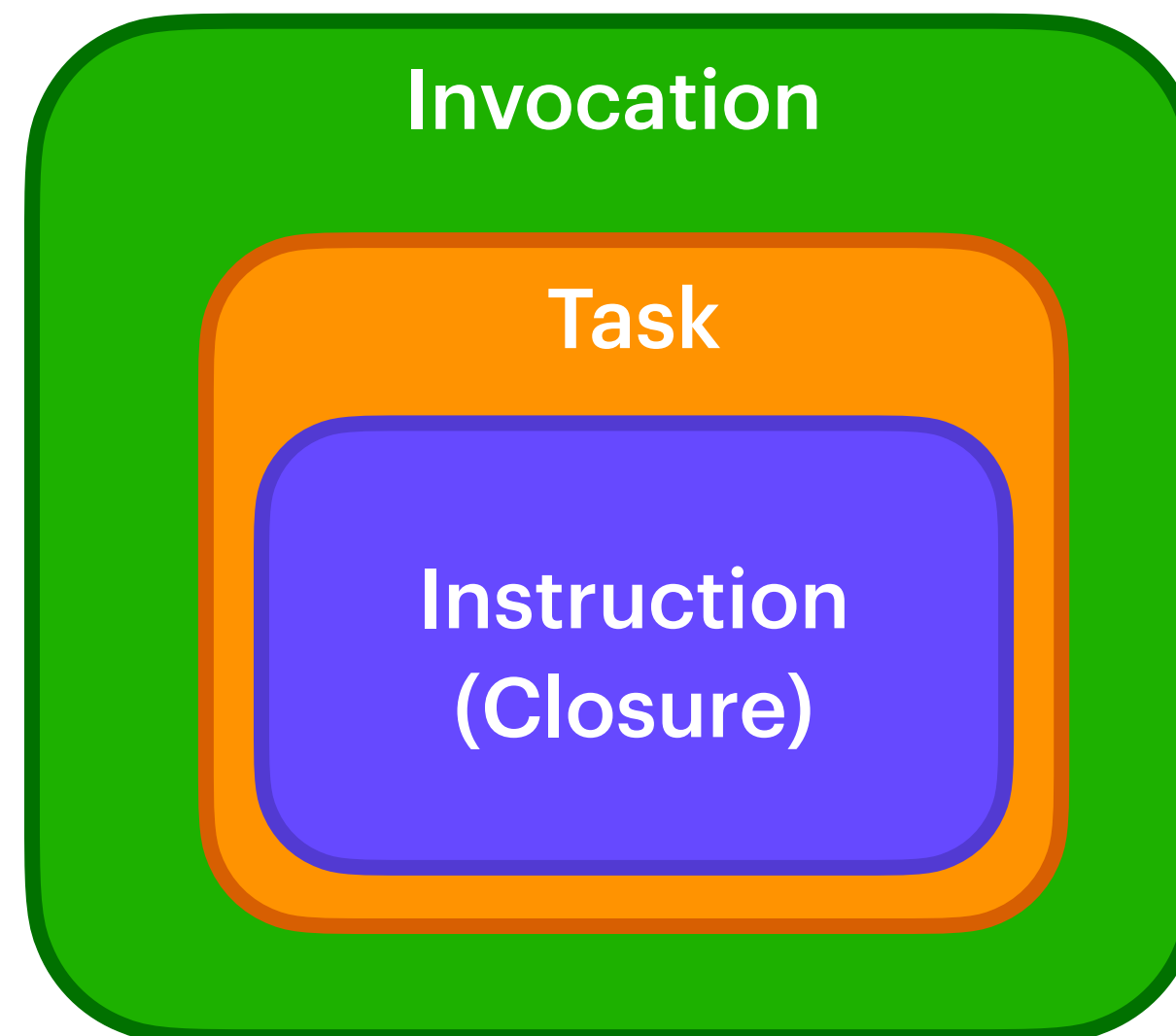
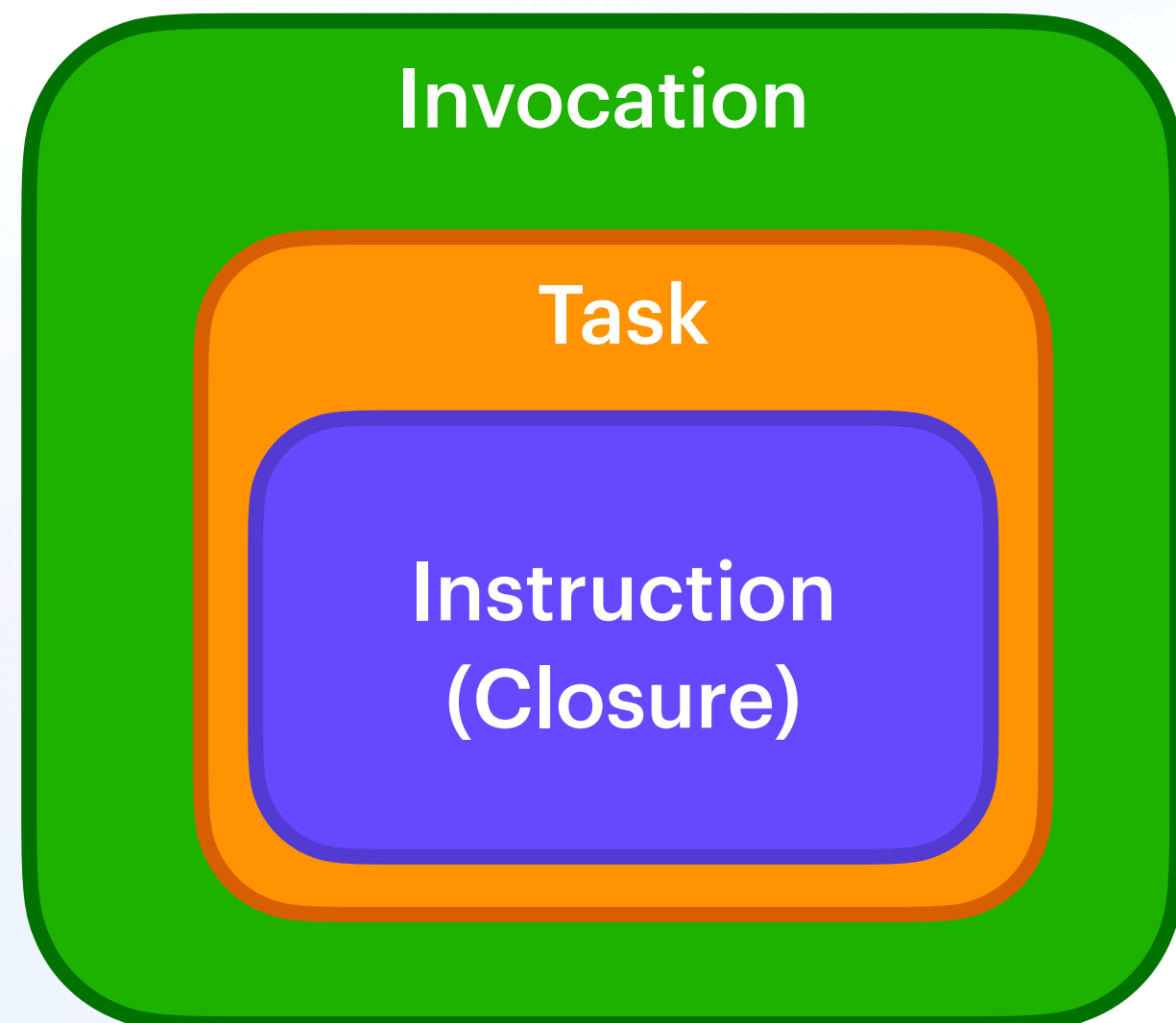
Compute Substrate

IPLD Schema

```
type Instruction struct {  
  rsc      URI  
  op       Ability  
  input    {String : Any}  
  nnc      String  
}
```

```
type Task struct {  
  run      &Instruction  
  meta     {String : Any}  
  prf      [&UCAN]  
  cause    optional &Receipt  
}
```

```
type Invocation struct {  
  task     Task  
  auth     &Authorization  
}
```



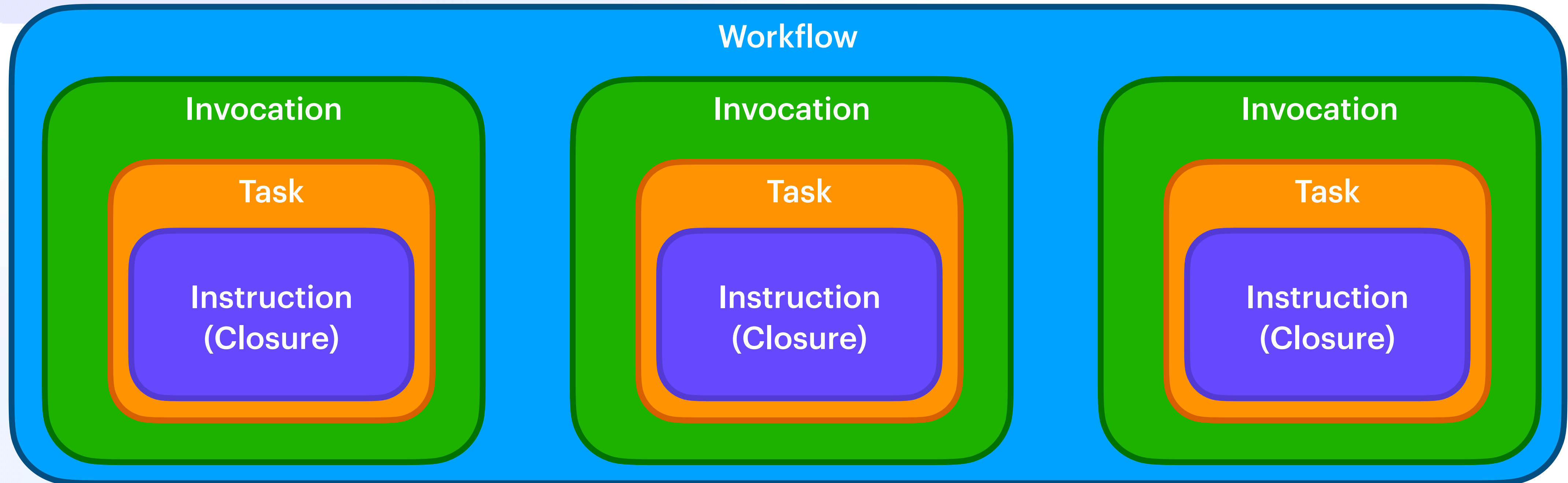
Compute Substrate

IPLD Schema

```
type Instruction struct {  
  rsc      URI  
  op       Ability  
  input    {String : Any}  
  nnc      String  
}
```

```
type Task struct {  
  run      &Instruction  
  meta     {String : Any}  
  prf      [&UCAN]  
  cause    optional &Receipt  
}
```

```
type Invocation struct {  
  task     Task  
  auth     &Authorization  
}
```



Compute Substrate

Matching Impedance

Compute Substrate

Matching Impedance

- Null
- Boolean
- Integer
- Float
- String
- Bytes
- List
- Map
- Link
- **Union**
- **Struct**
- **Enum**
- **Copy**

Compute Substrate

Matching Impedance

- Null
- Boolean
- Integer
- Float
- String
- Bytes
- List
- Map
- Link
- **Union**
- **Struct**
- **Enum**
- **Copy**

```
ty ::= 'u8' | 'u16' | 'u32' | 'u64'
     's8' | 's16' | 's32' | 's64'
     'float32' | 'float64'
     'char'
     'bool'
     'string'
     tuple
     list
     option
     result
     handle
     id

tuple ::= 'tuple' '<' tuple-list '>'
tuple-list ::= ty | ty ',' tuple-list?

list ::= 'list' '<' ty '>'

option ::= 'option' '<' ty '>'

result ::= 'result' '<' ty ' ' ty '>'
         'result' '<' '_' ' ' ty '>'
         'result' '<' ty '>'
         'result'
```

Compute Substrate

Matching Impedance

- Null
- Boolean
- Integer
- Float
- String
- Bytes
- List
- Map
- Link
- **Union**
- **Struct**
- **Enum**
- **Copy**

```
ty ::= 'u8' | 'u16' | 'u32' | 'u64'  
    's8' | 's16' | 's32' | 's64'  
    'float32' | 'float64'  
    'char'  
    'bool'  
    'string'  
    tuple  
    list  
    option  
    result  
    handle  
    id
```

```
tuple ::= 'tuple' '<' tuple-list '>'  
tuple-list ::=
```

```
list ::= 'list'
```

```
option ::= 'opt'
```

```
result ::= 'res'  
         'res'  
         'res'  
         'res'
```

The screenshot shows a GitHub repository page for 'component-model / spec'. At the top, there's a search bar and 'Add file' button. Below that, a commit by 'lukewagner' is visible with the message 'Add skeleton Explainer.md containing only AST and Binary.md defining ...'. A table below lists files and their commit details:

| Name | Last commit message | Last commit date |
|-----------|--|------------------|
| .. | | |
| README.md | Add skeleton Explainer.md containing only AST and Binary.md defining ... | last year |

At the bottom, there's a 'README.md' file with an edit icon.

This directory will contain the formal Component Model specification,

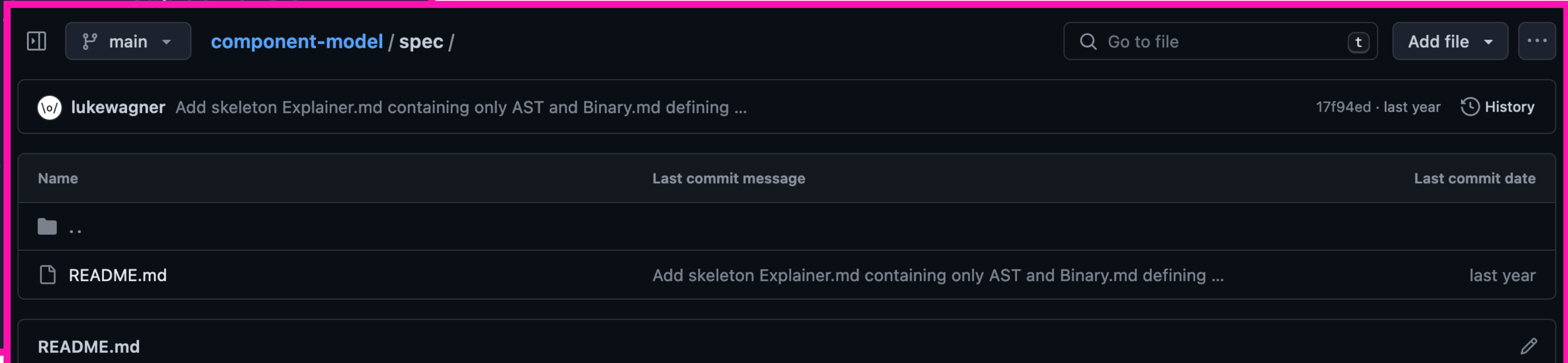
Compute Substrate

Matching Impedance

- Null
- Boolean
- Integer
- Float
- String
- Bytes
- List
- Map
- Link
- **Union**
- **Struct**
- **Enum**
- **Copy**

```
ty ::= 'u8' | 'u16' | 'u32' | 'u64'  
    's8' | 's16' | 's32' | 's64'  
    'float32' | 'float64'  
    'char'  
    'bool'  
    'string'  
    tuple  
    list  
    option  
    result  
    handle  
    id  
  
tuple ::= 'tuple' '<' tuple-list '>'  
tuple-list ::=  
  
list ::= 'list'  
  
option ::= 'opt'  
  
result ::= 'res'  
         'res'  
         'res'  
         'res'
```

e.g. 2 IPLD numerics < 10 WIT numerics



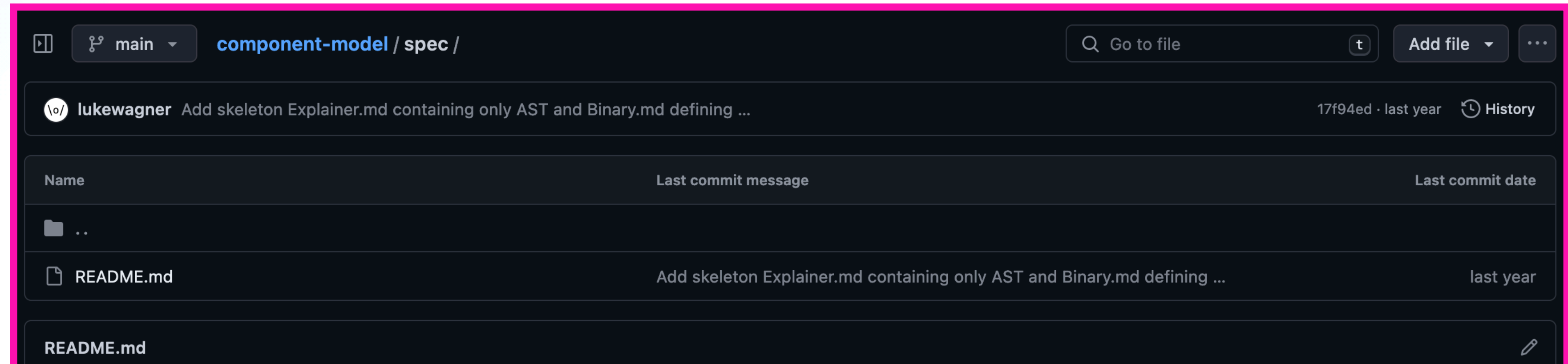
This directory will contain the formal Component Model specification,

Compute Substrate

Matching Impedance

- Null
- Boolean
- Integer
- Float
- String
- Bytes
- List
- Map
- Link
- **Union**
- **Struct**
- **Enum**
- **Copy**

e.g. 2 IPLD numerics < 10 WIT numerics



This directory will contain the formal Component Model specification,

Compute Substrate

Matching Impedance

```
{  
  "run": {  
    "rsc": "ipfs://bafkreigpbimktgowom47jv7frrt3xvnb7ati4upgguykyn2cuunt32l63ya",  
    "op": "wasm/run",  
    "input": {  
      "args": ["hello", "world"]  
    }  
  },  
  "meta": {  
    "limits": {  
      "fuel": 10000  
    },  
    "tags": ["demo", "wasm", "ucan", "ipvm"],  
    "author": "@expede@octodon.social"  
  }  
}
```

Compute Substrate

Composition Without Imposition

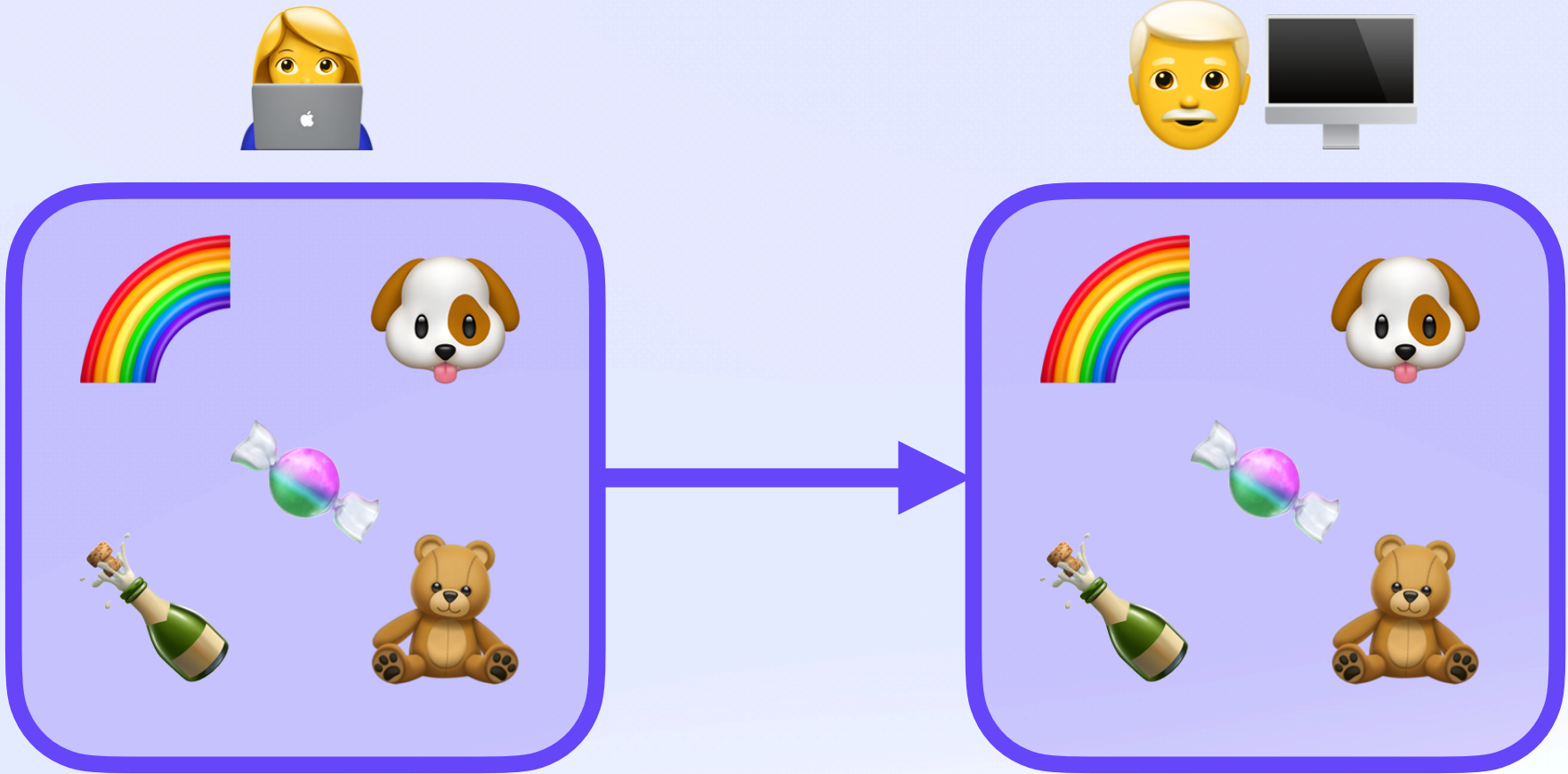
Compute Substrate

Composition Without Imposition



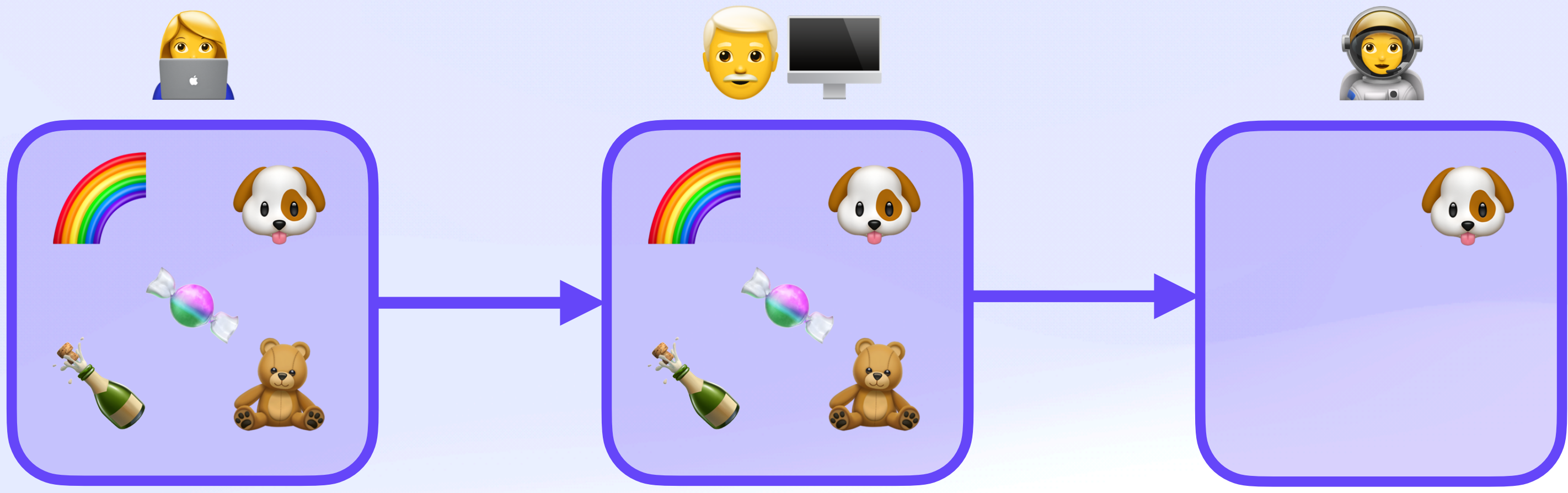
Compute Substrate

Composition Without Imposition



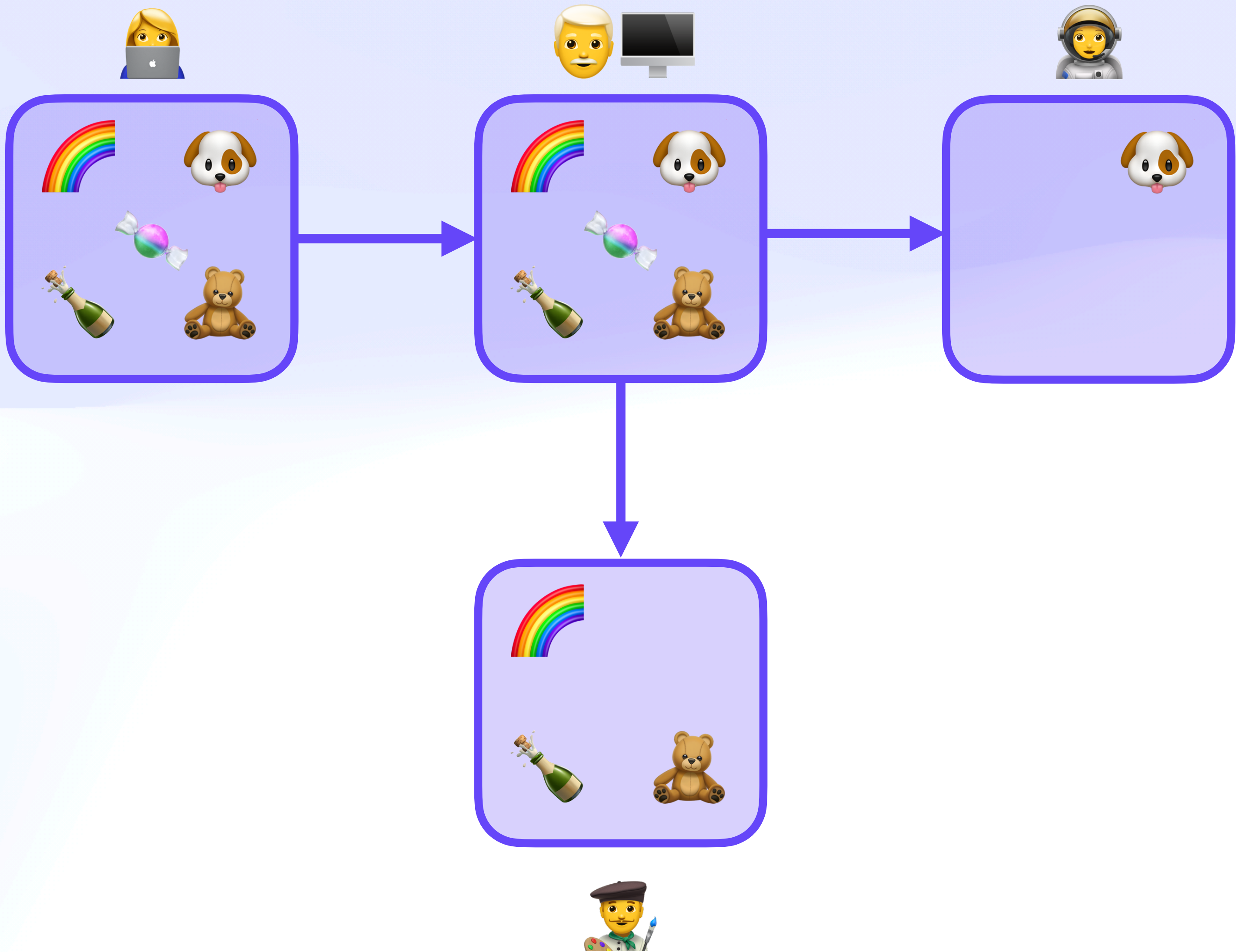
Compute Substrate

Composition Without Imposition



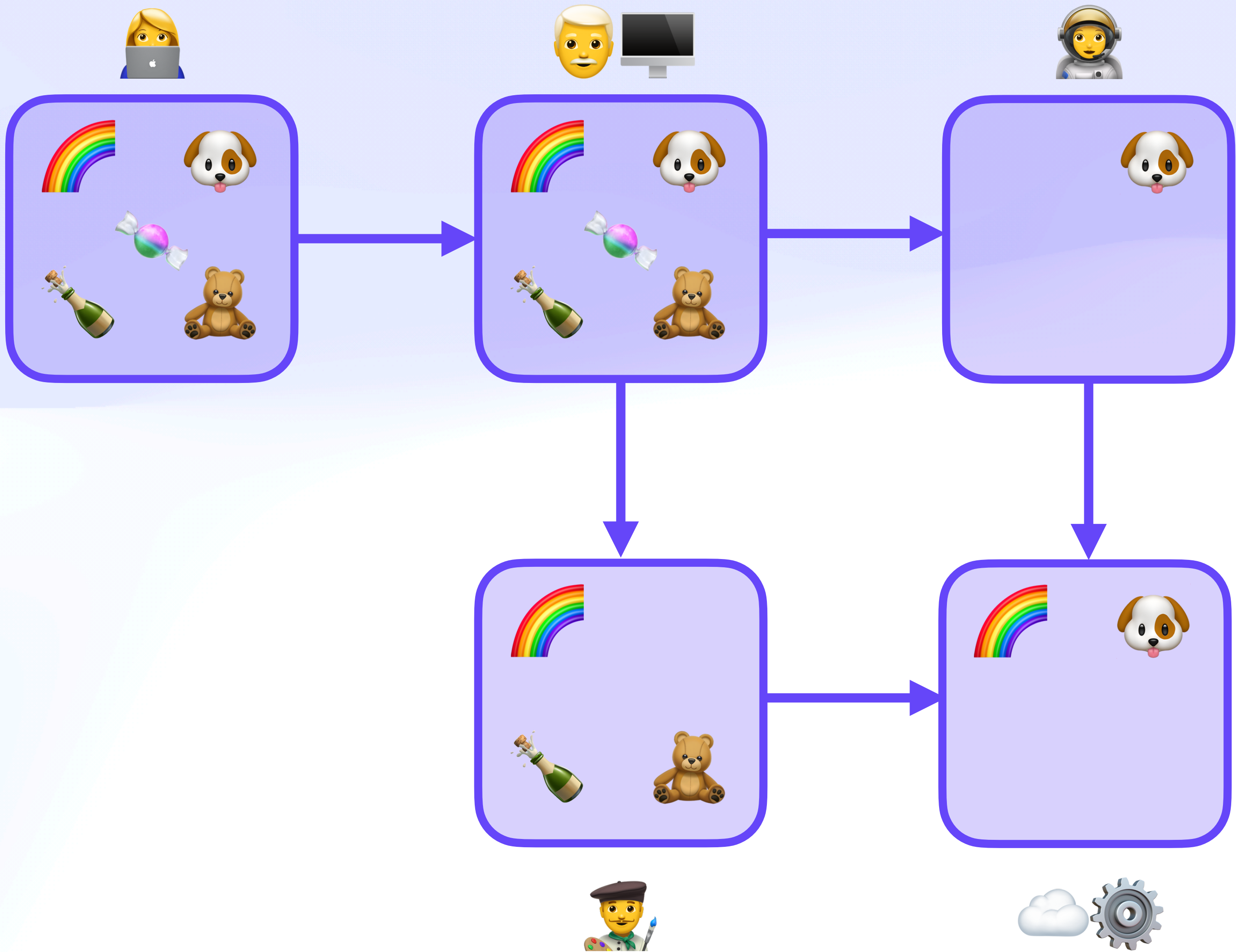
Compute Substrate

Composition Without Imposition



Compute Substrate

Composition Without Imposition



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
```

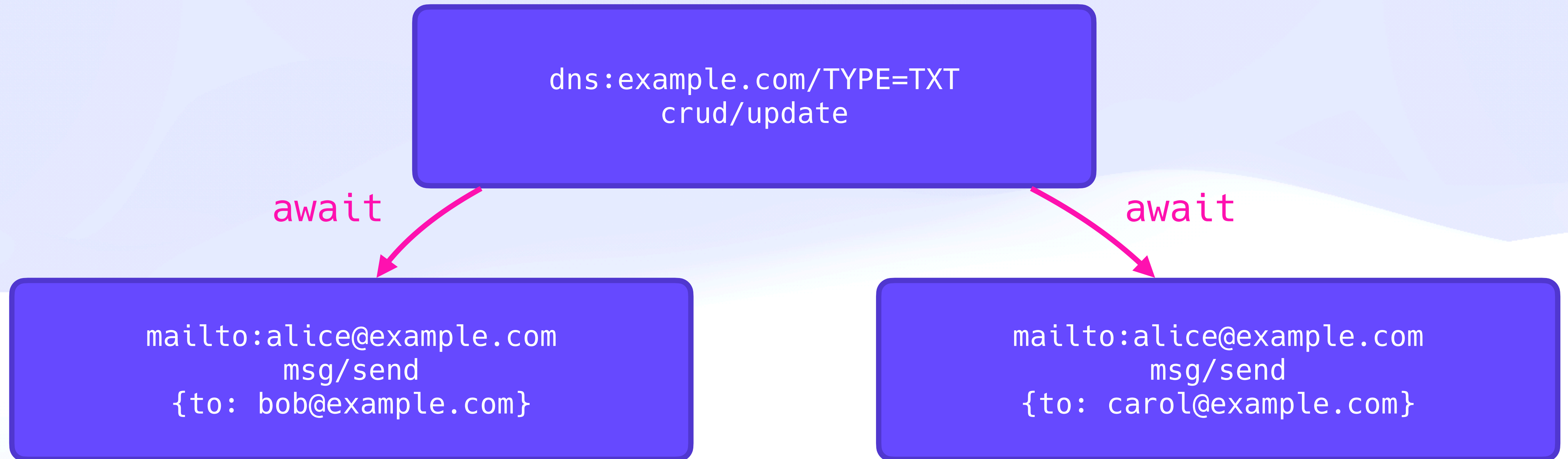
await



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

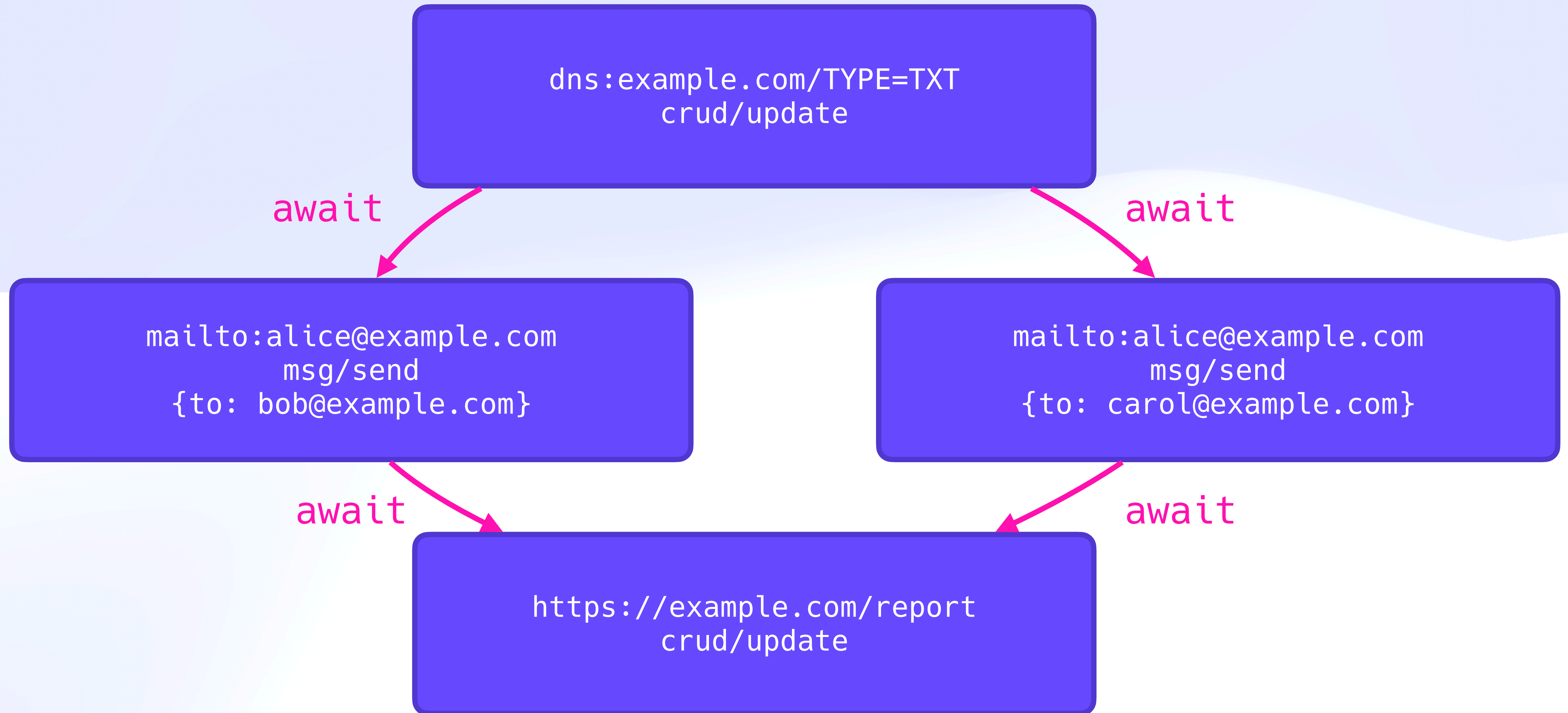
Compute Substrate

Distributed Invocation



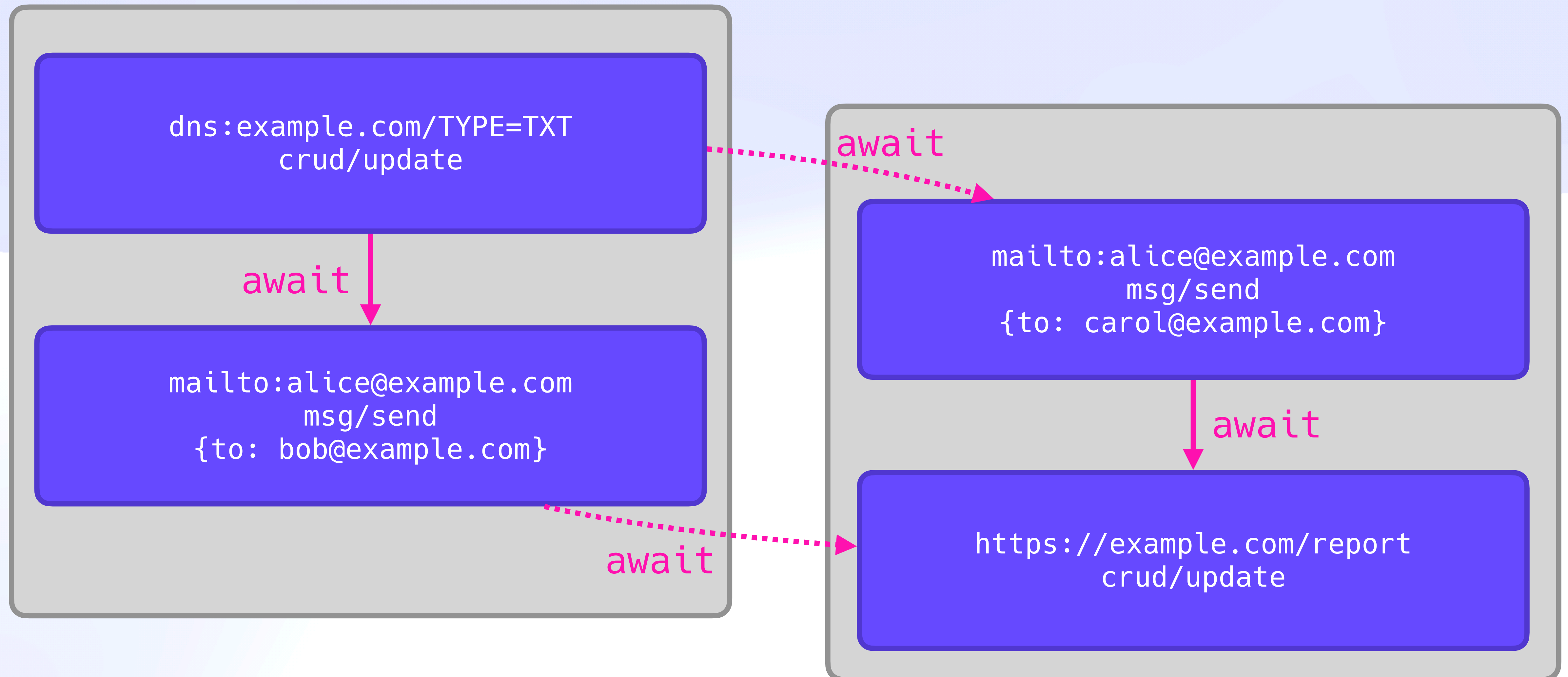
Compute Substrate

Distributed Invocation



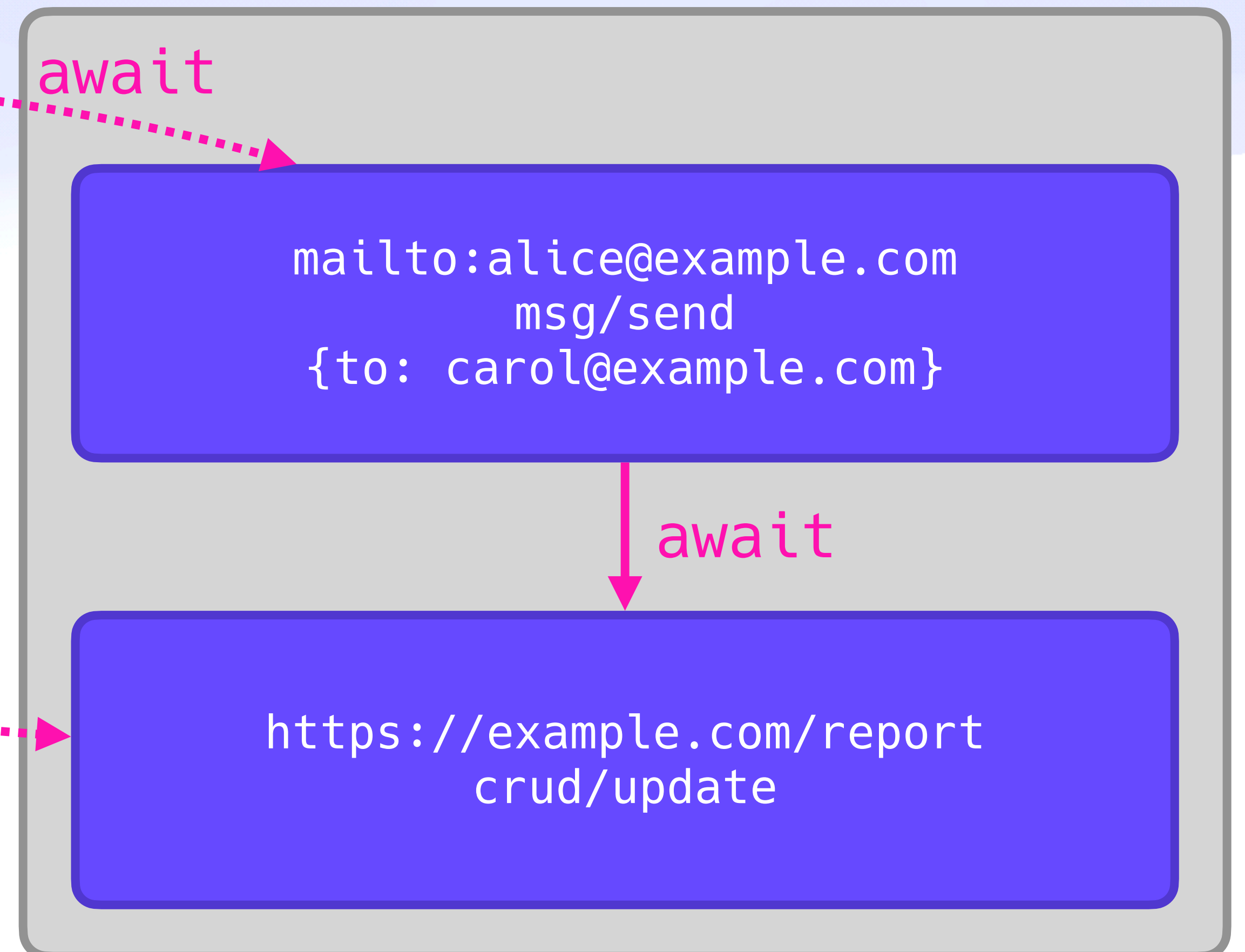
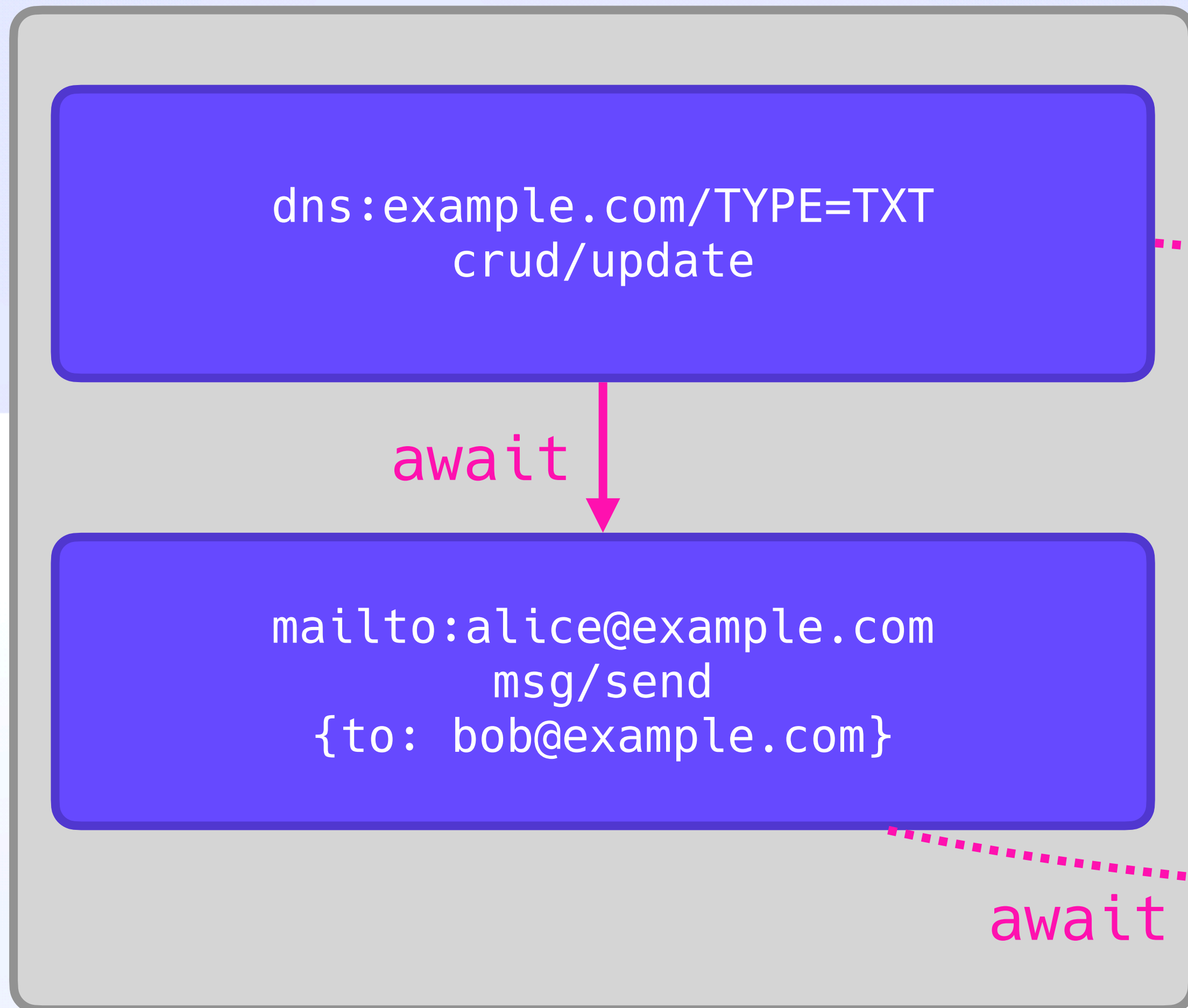
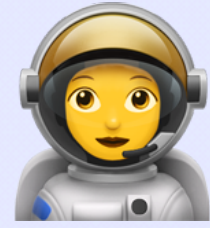
Compute Substrate

Distributed Invocation



Compute Substrate

Distributed Invocation

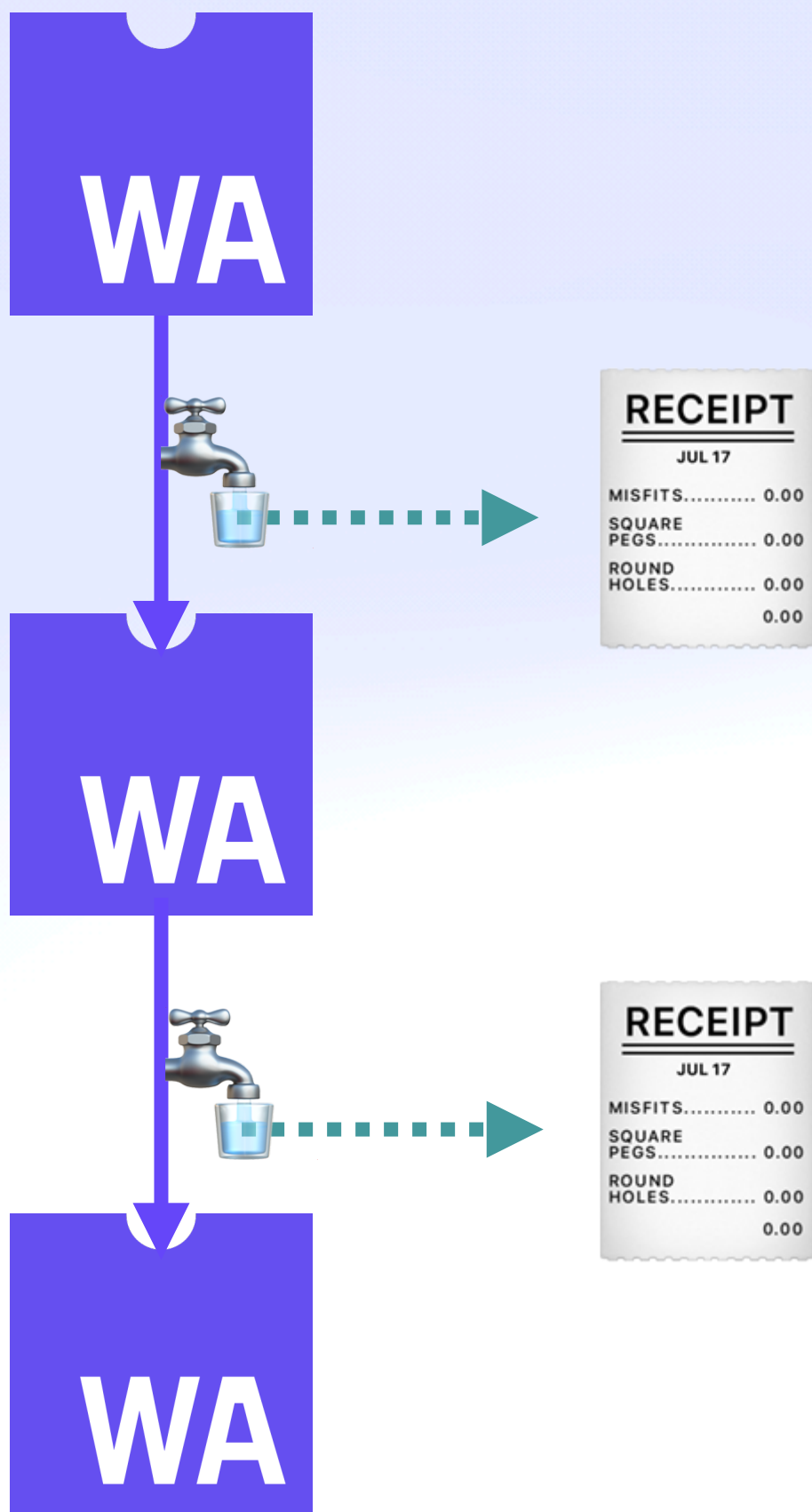


Compute Substrate

Cache, Suspend, Move, Verify

Compute Substrate

Cache, Suspend, Move, Verify

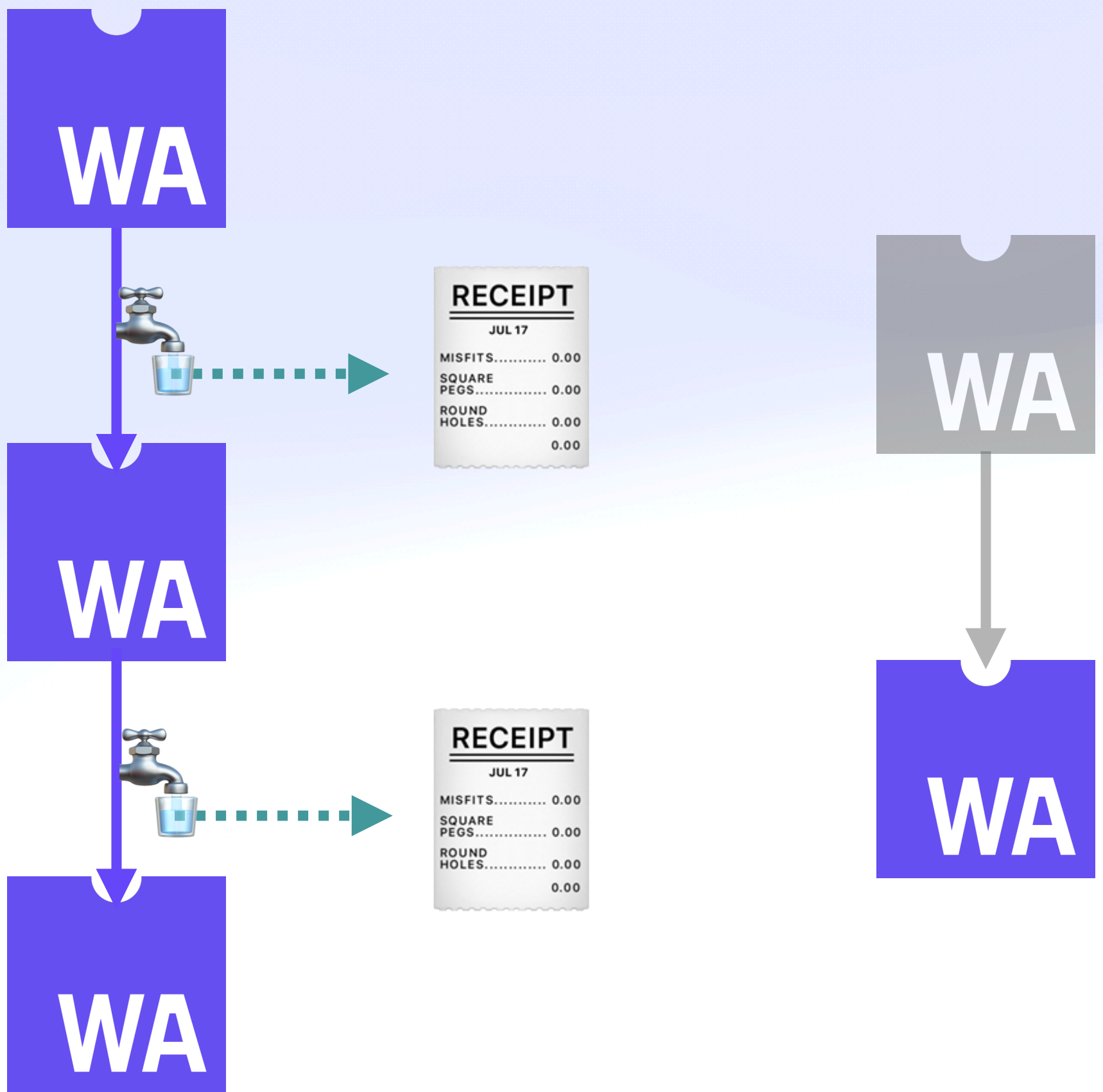


| RECEIPT | |
|------------------|------|
| JUL 17 | |
| MISFITS..... | 0.00 |
| SQUARE PEGS..... | 0.00 |
| ROUND HOLES..... | 0.00 |
| | 0.00 |

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

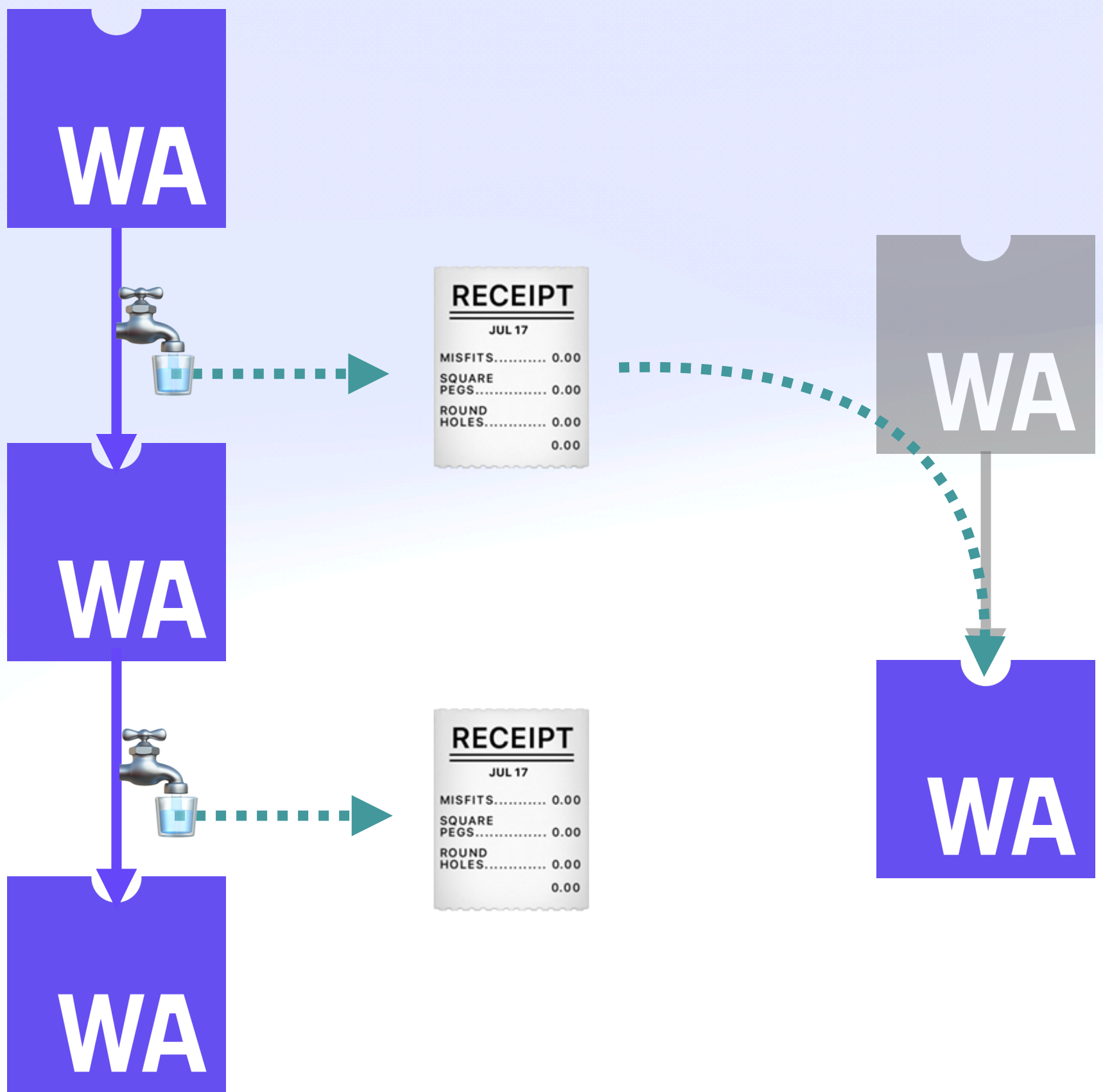
Compute Substrate

Cache, Suspend, Move, Verify



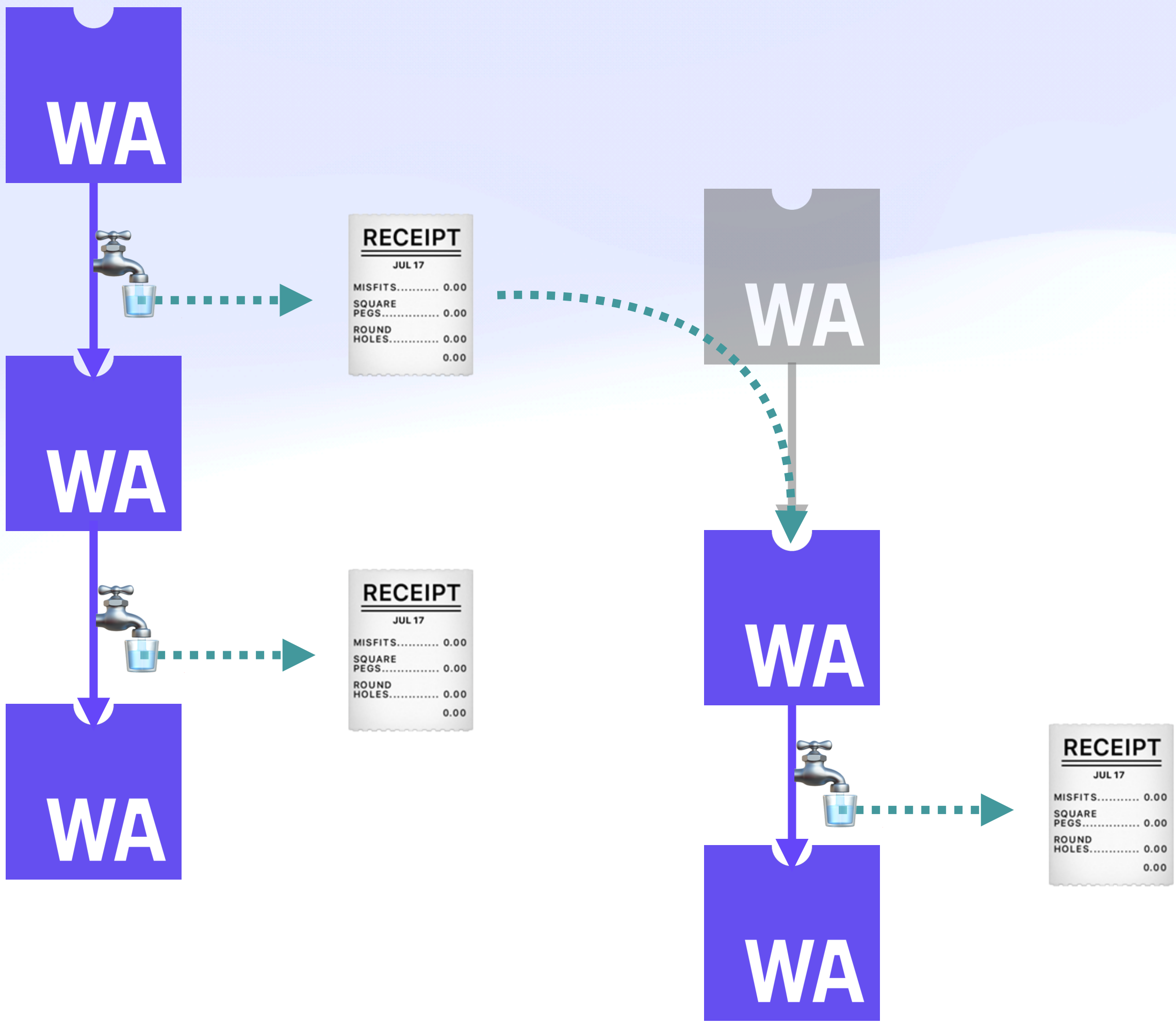
Compute Substrate

Cache, Suspend, Move, Verify



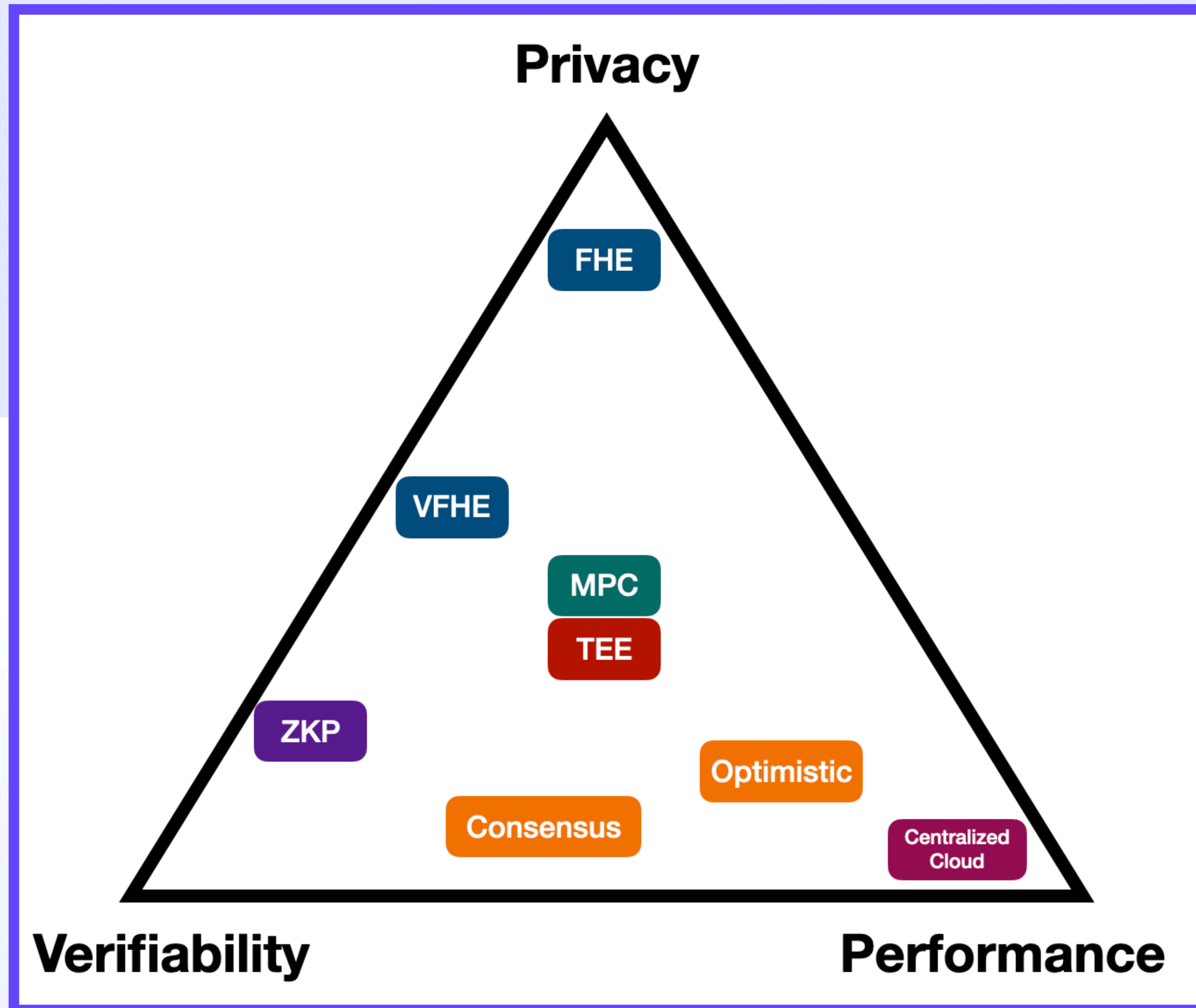
Compute Substrate

Cache, Suspend, Move, Verify



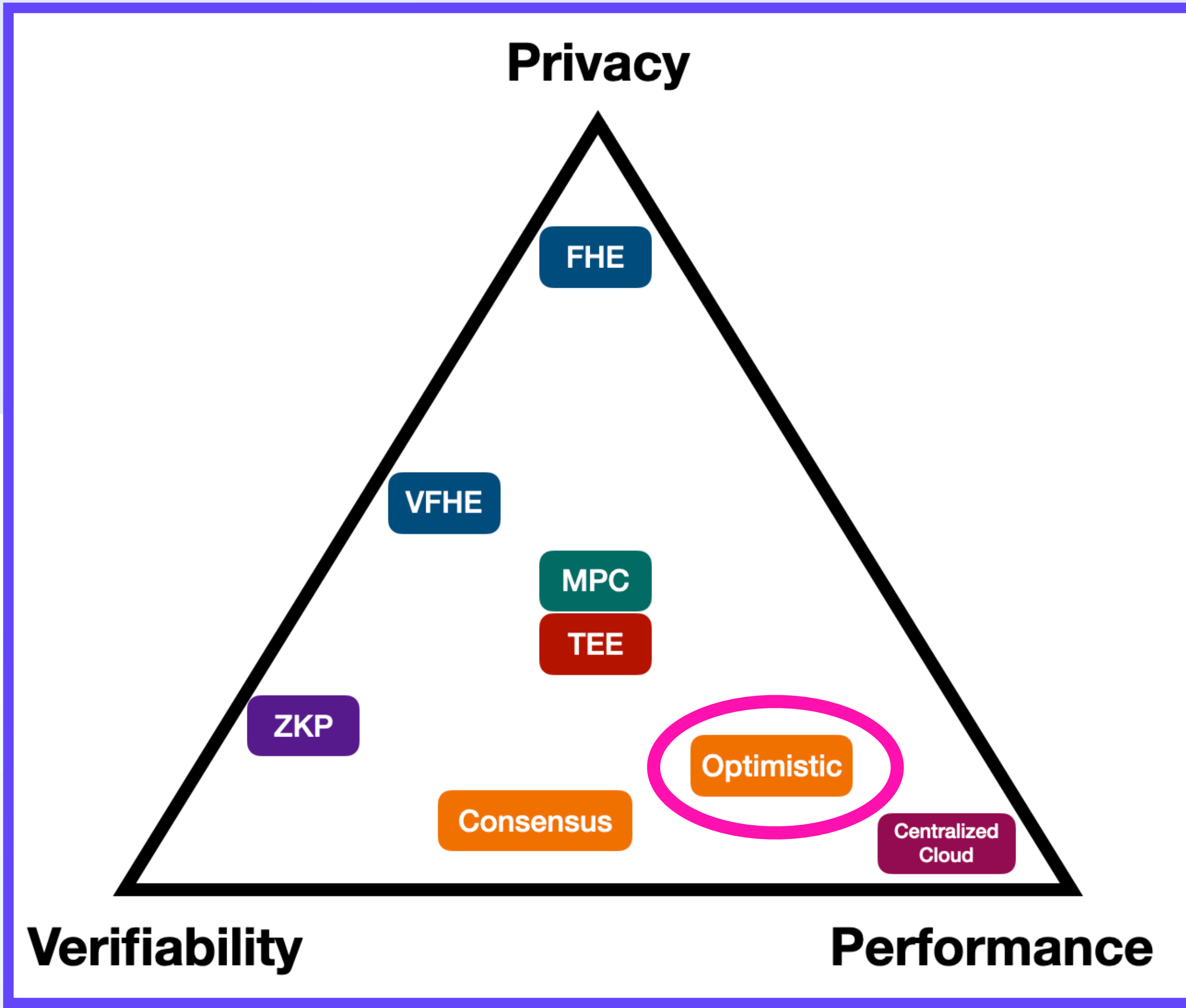
Compute Substrate

Determinism = Fully Verifiable Computation



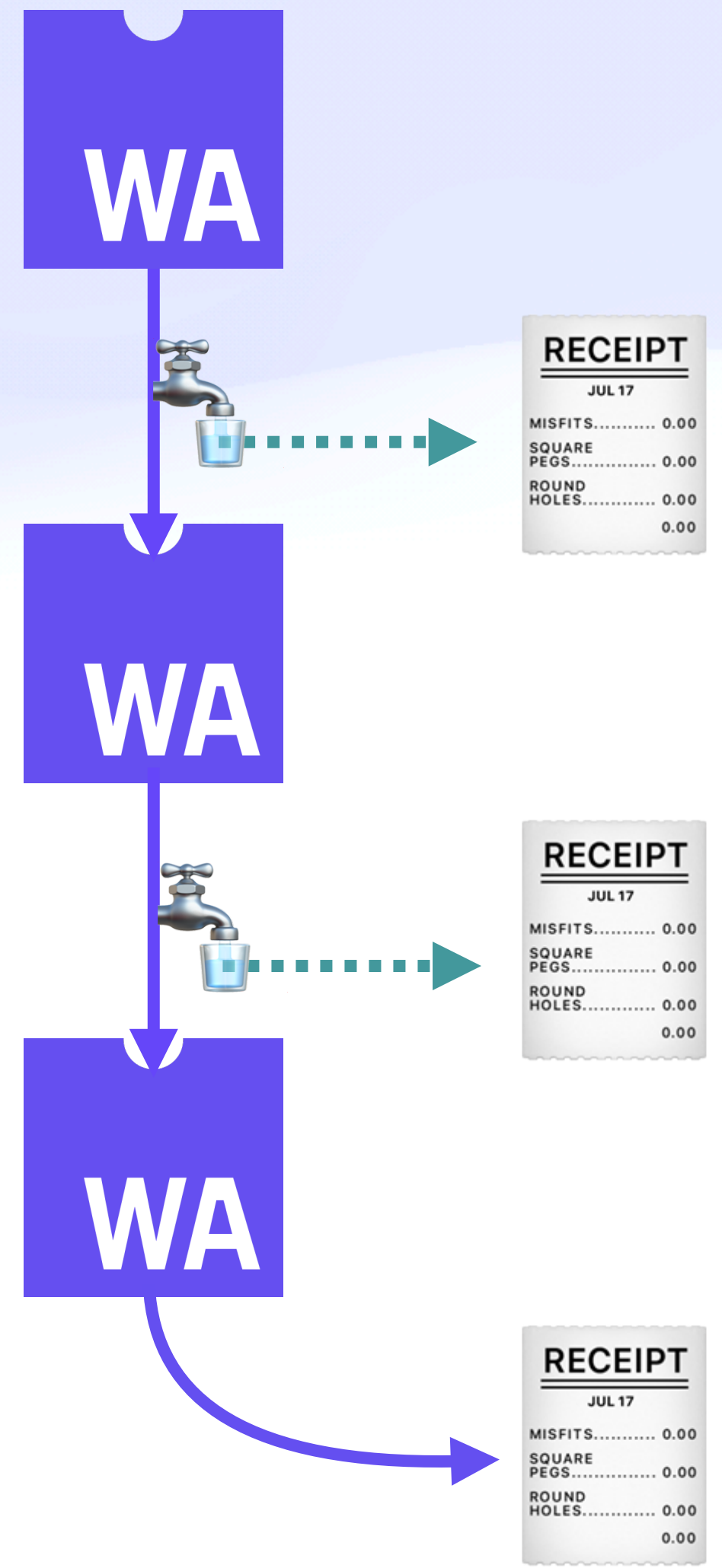
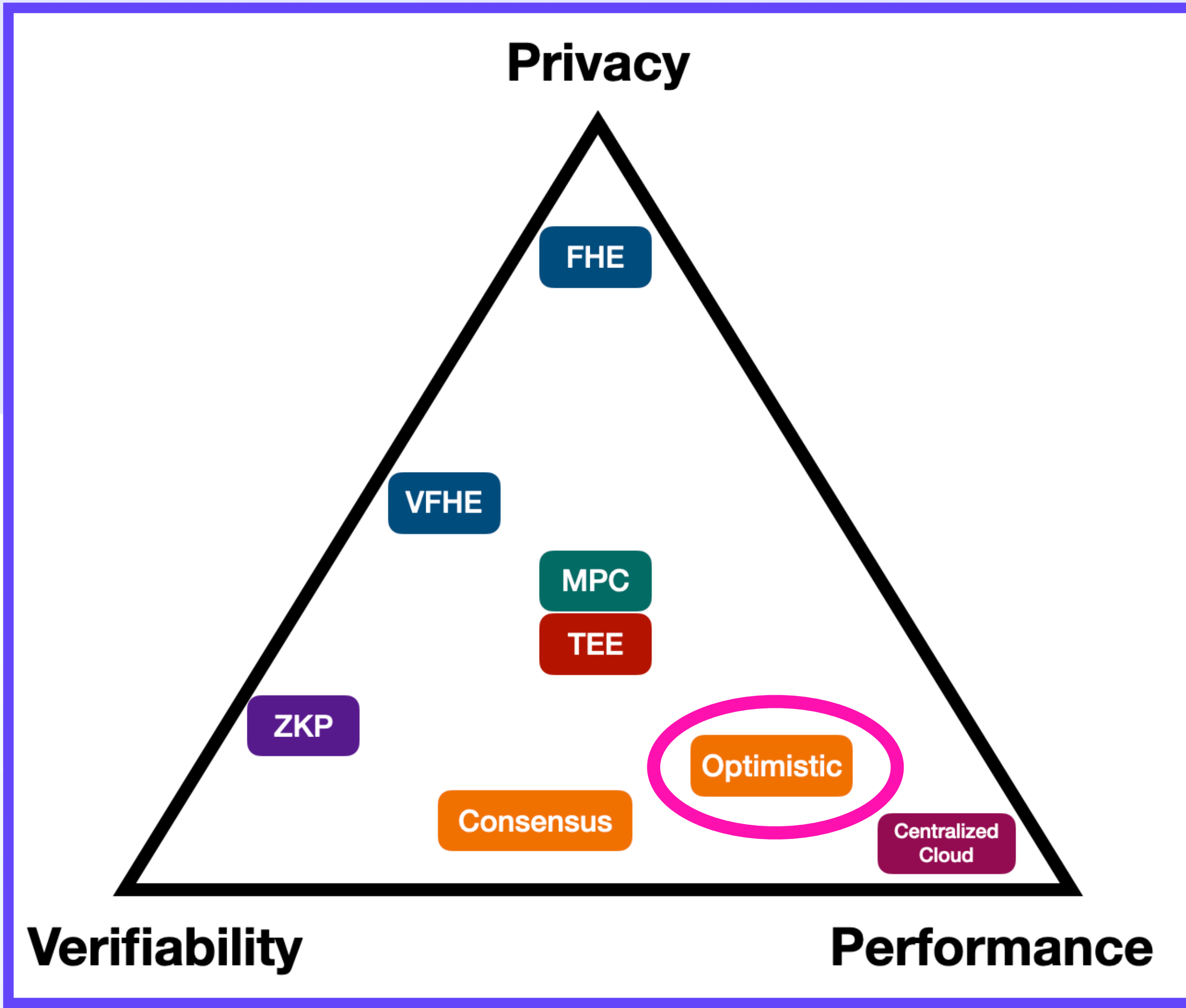
Compute Substrate

Determinism = Fully Verifiable Computation



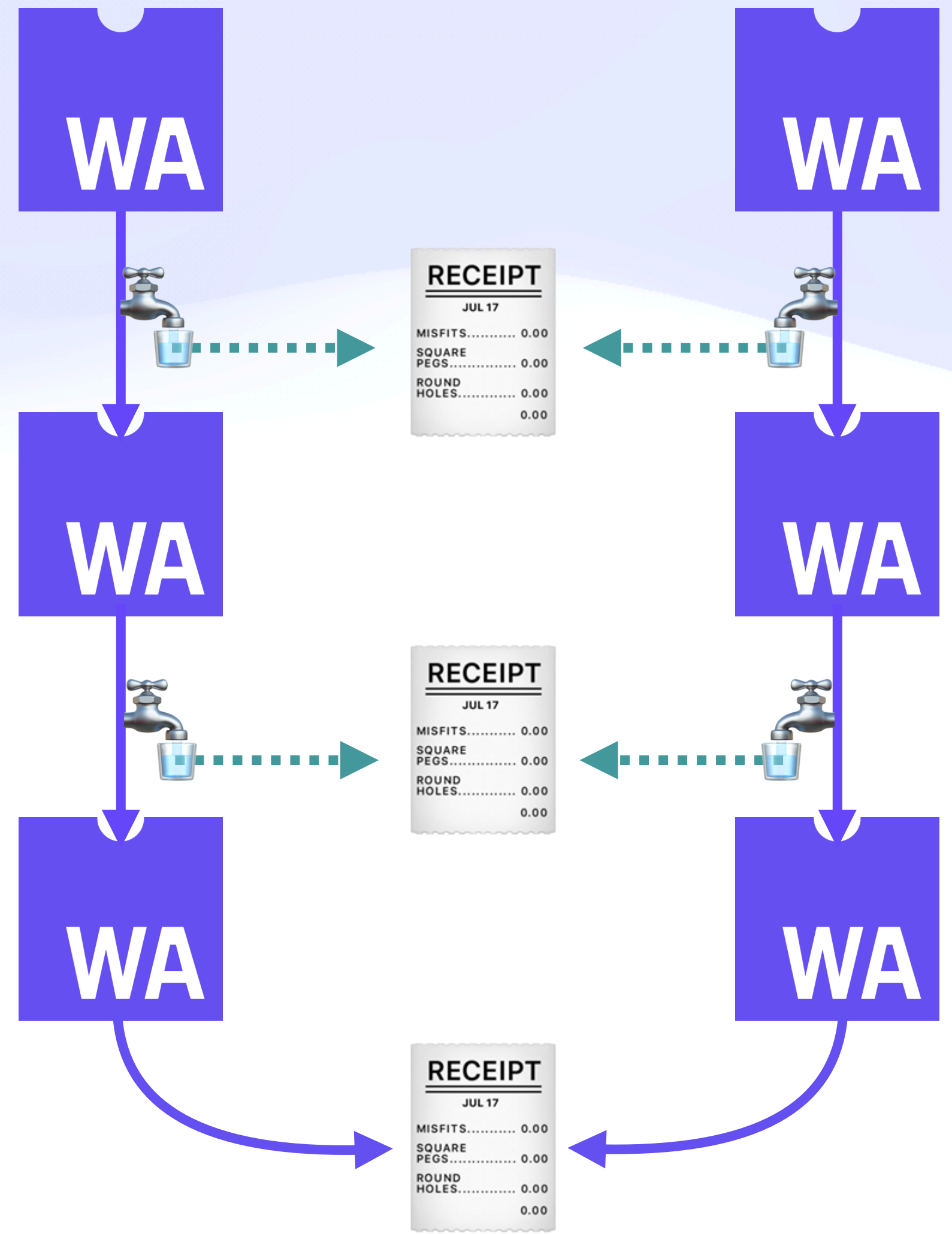
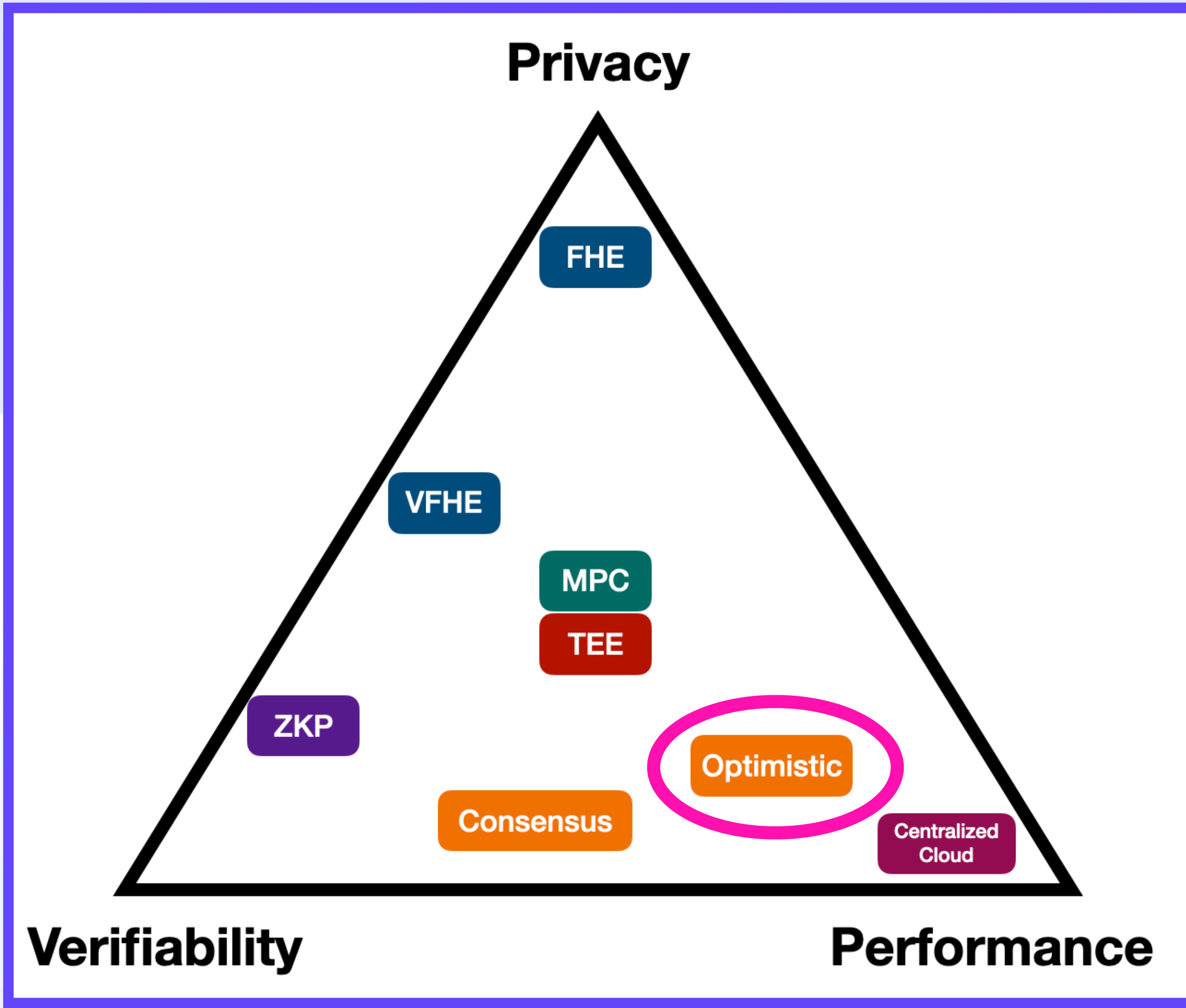
Compute Substrate

Determinism = Fully Verifiable Computation



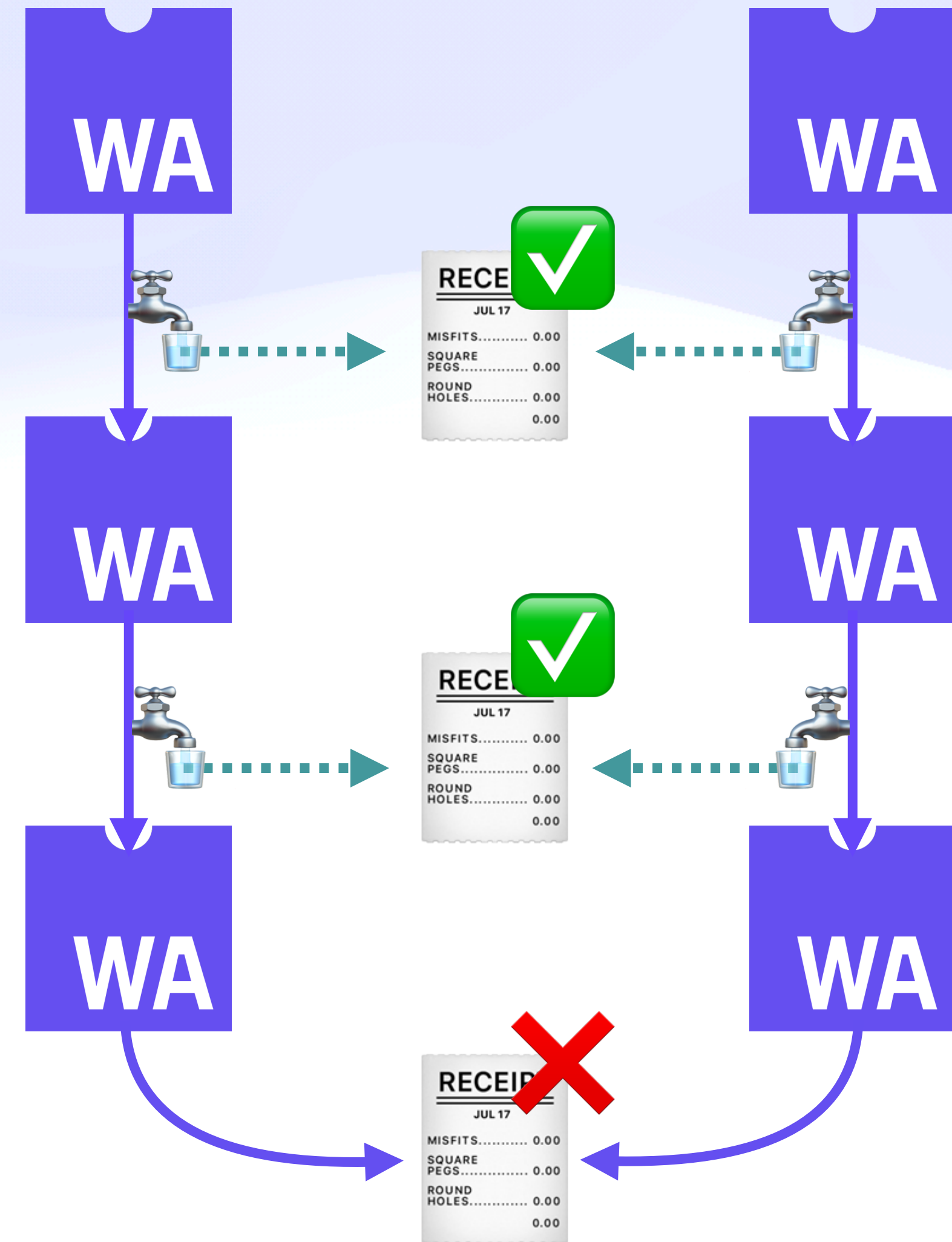
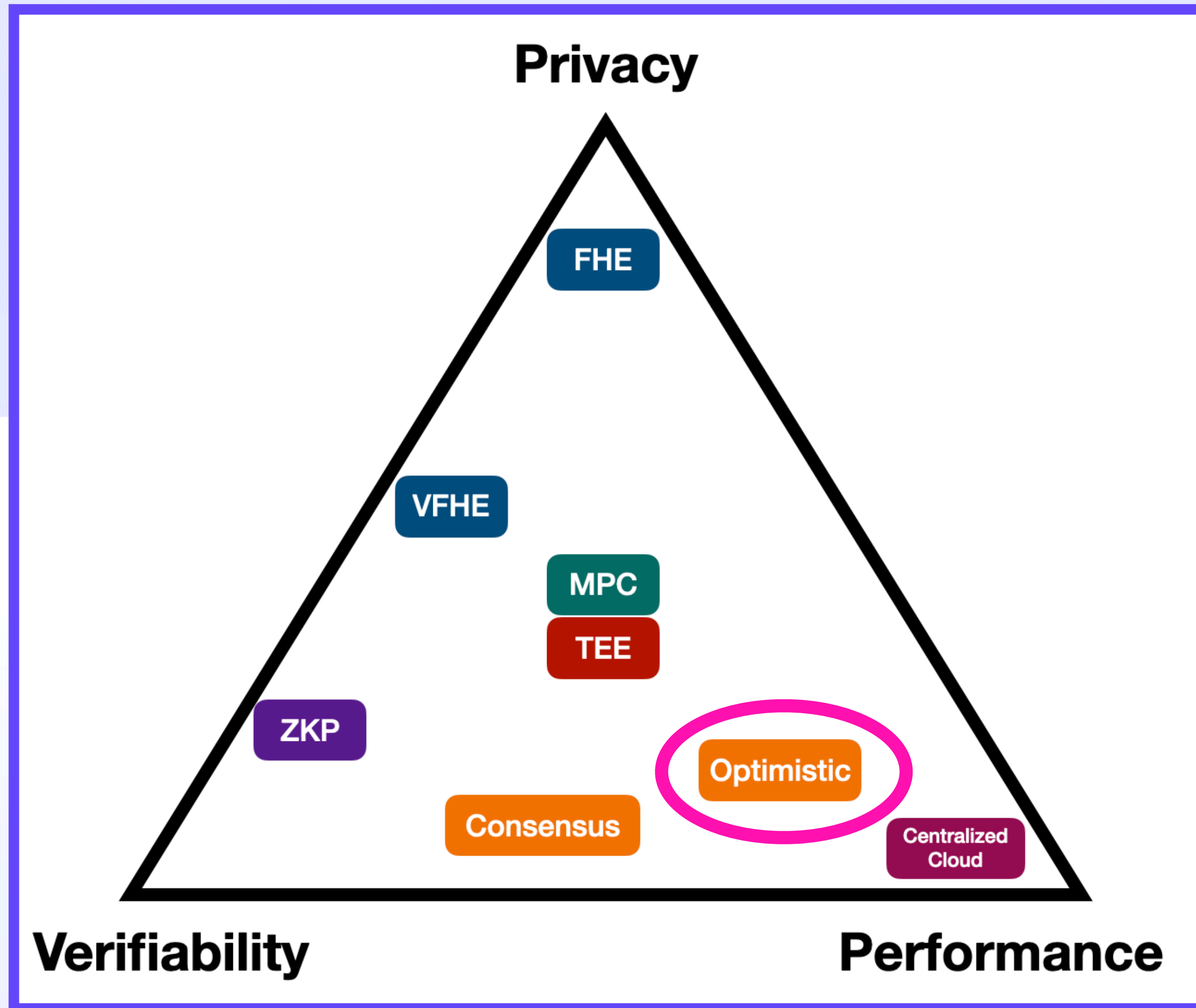
Compute Substrate

Determinism = Fully Verifiable Computation



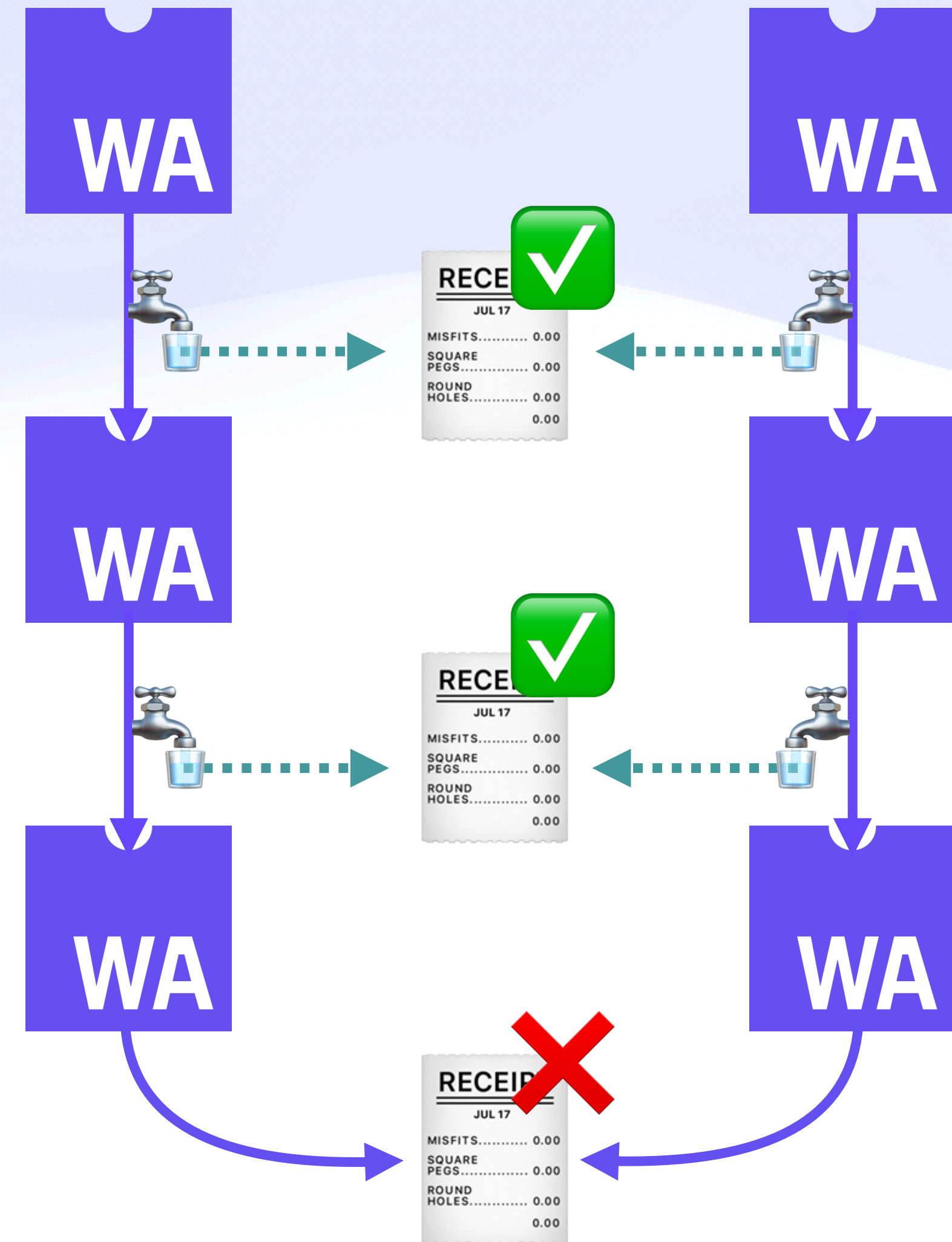
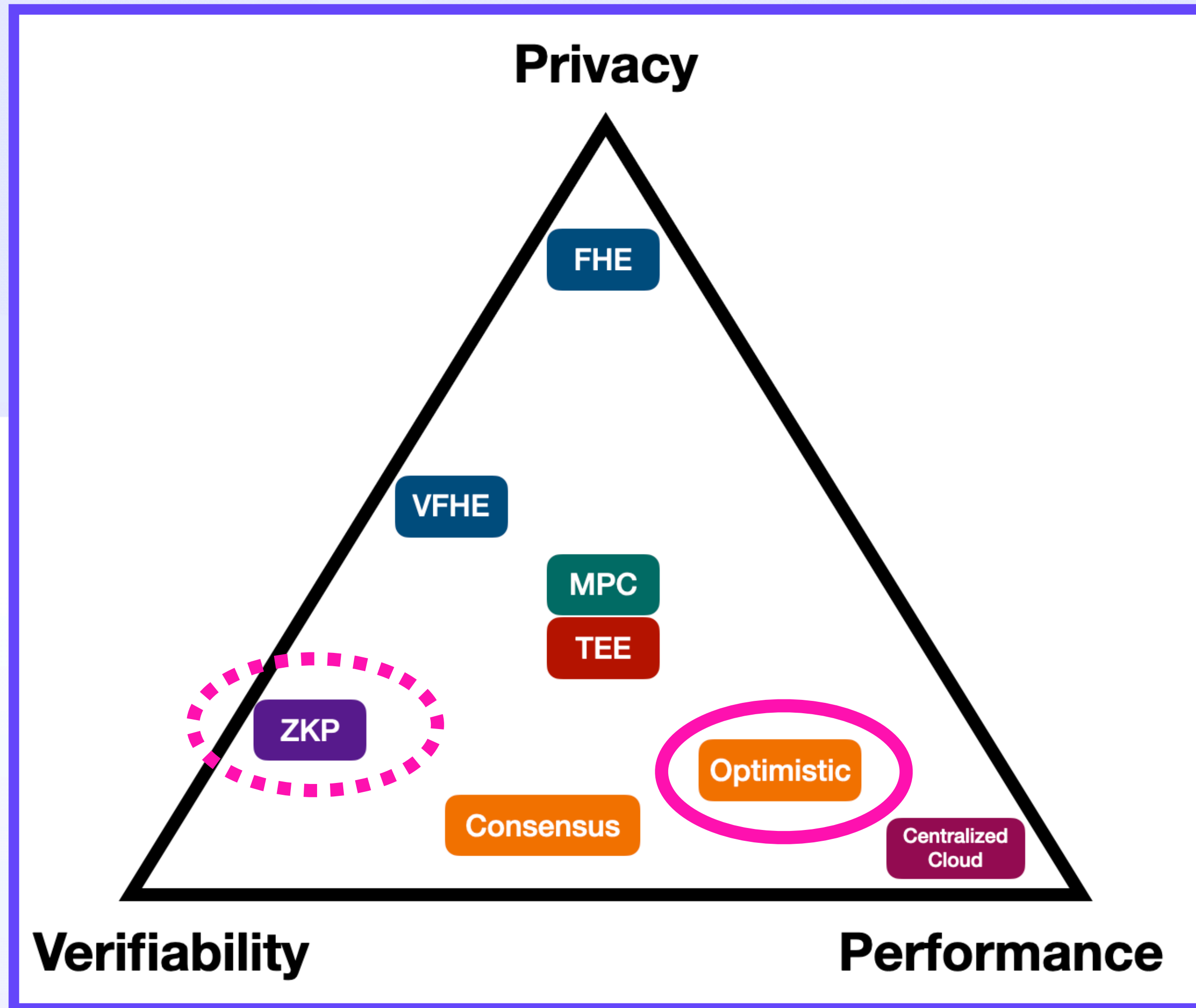
Compute Substrate

Determinism = Fully Verifiable Computation



Compute Substrate

Determinism = Fully Verifiable Computation



Compute Substrate

Surprise: Reverse Lookup For Free



Compute Substrate

Surprise: Reverse Lookup For Free



◆ **InID → Computed Result**

Compute Substrate

Surprise: Reverse Lookup For Free



- ◆ **InID → Computed Result**
- ◆ e.g. AI moderation classifier

Compute Substrate

Surprise: Reverse Lookup For Free



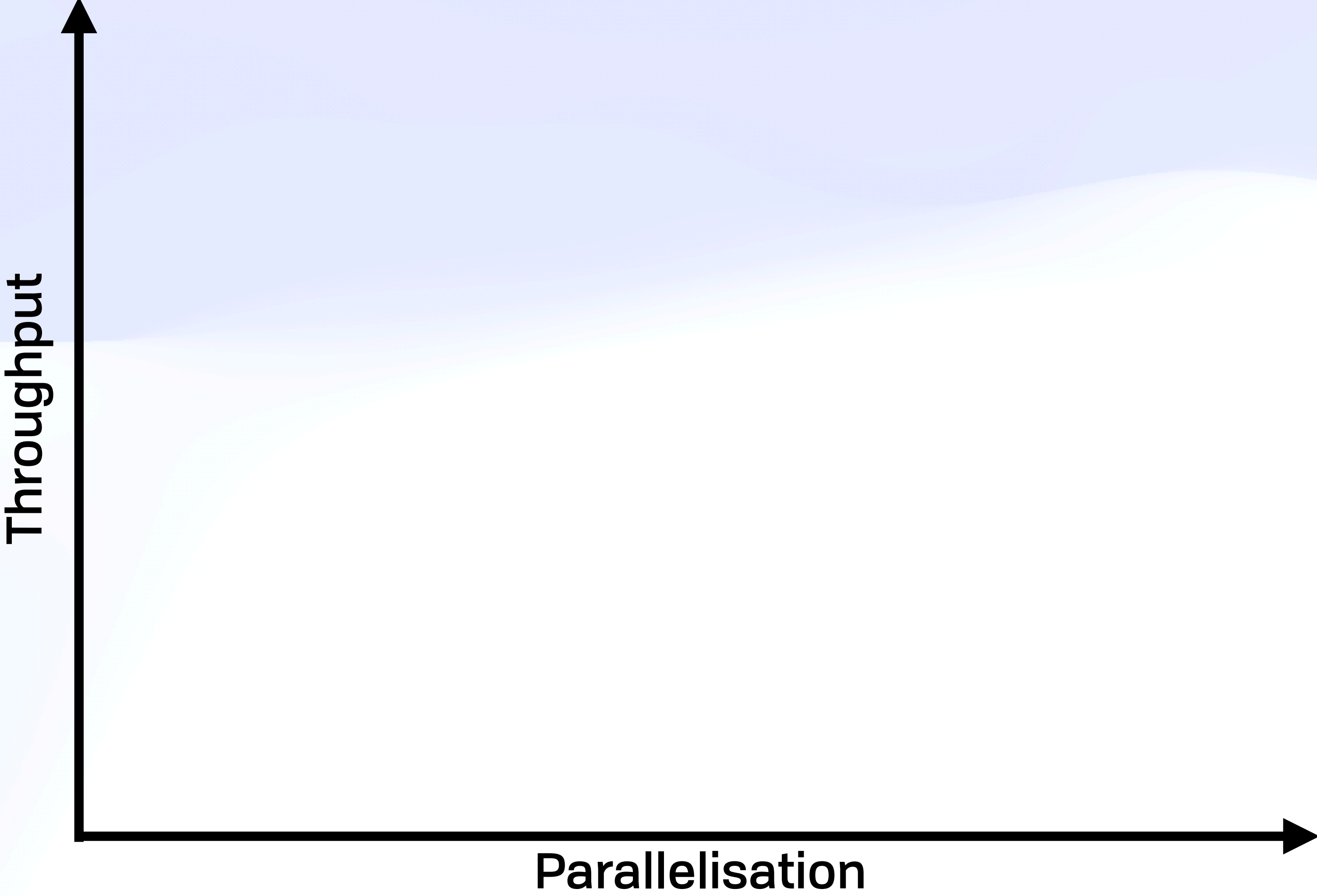
- ◆ **InID → Computed Result**
- ◆ e.g. AI moderation classifier
- ◆ e.g. Distributed token validation

Compute Substrate

With a Little Help From My Friends

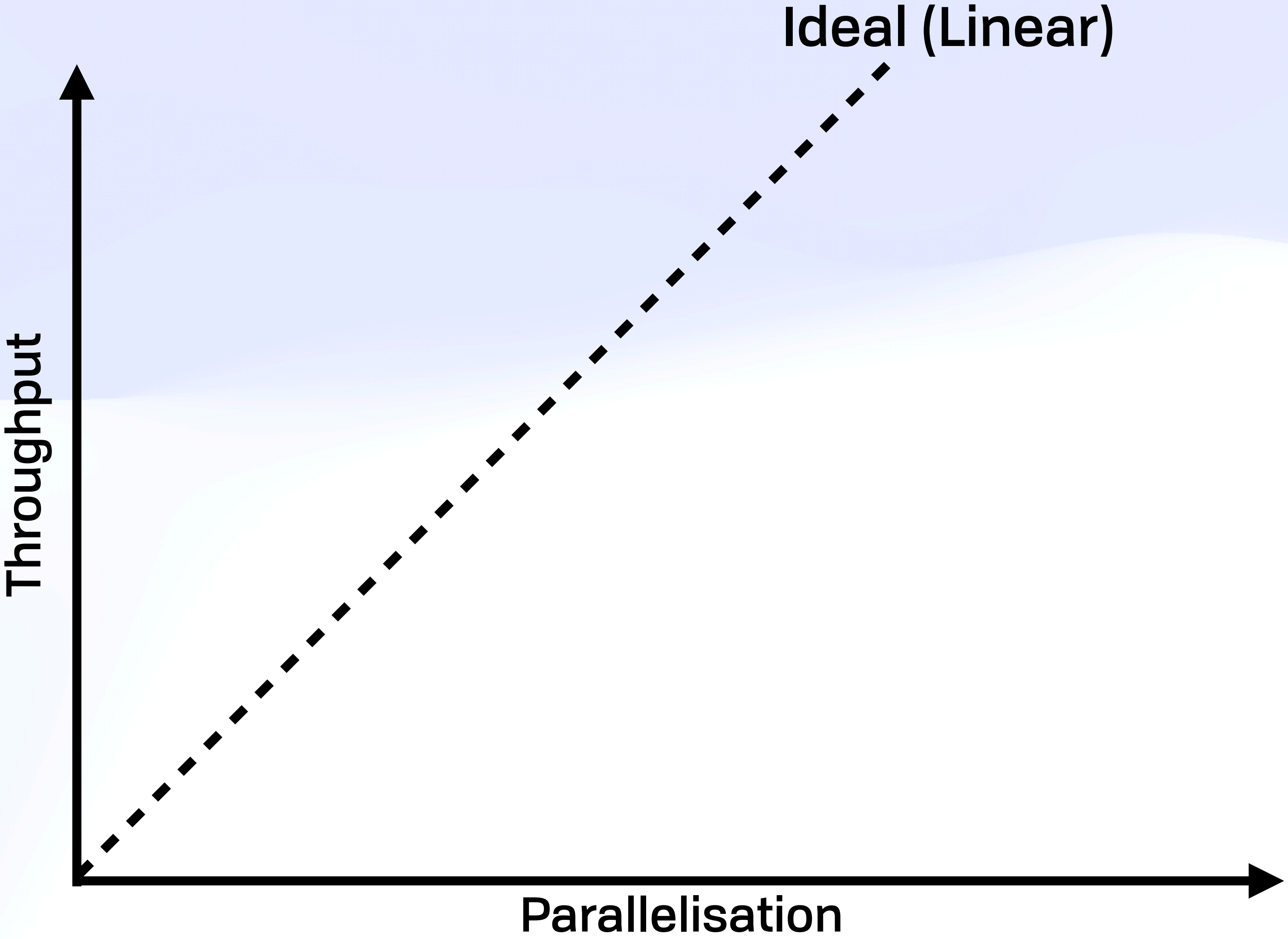
Compute Substrate

With a Little Help From My Friends



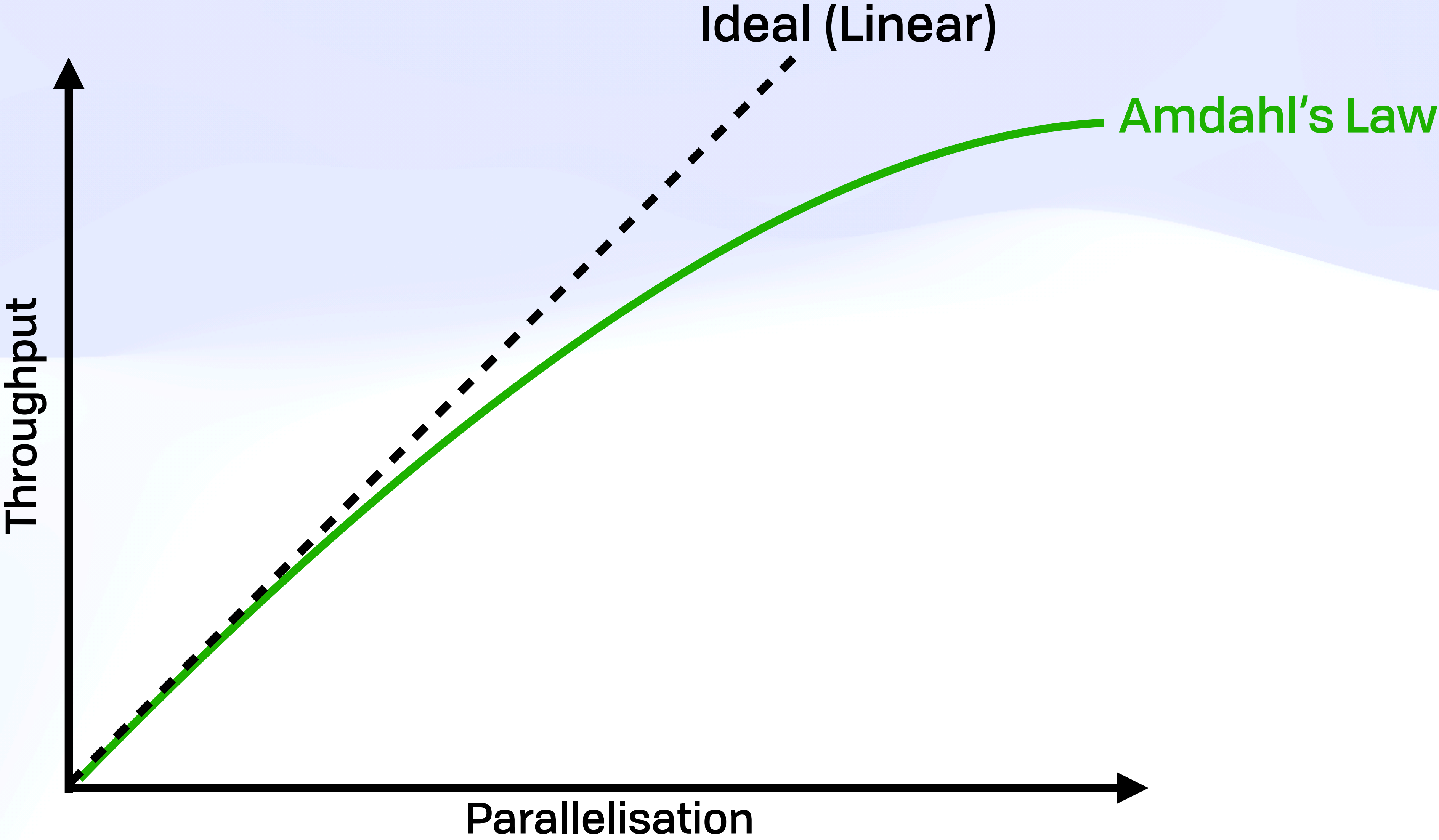
Compute Substrate

With a Little Help From My Friends



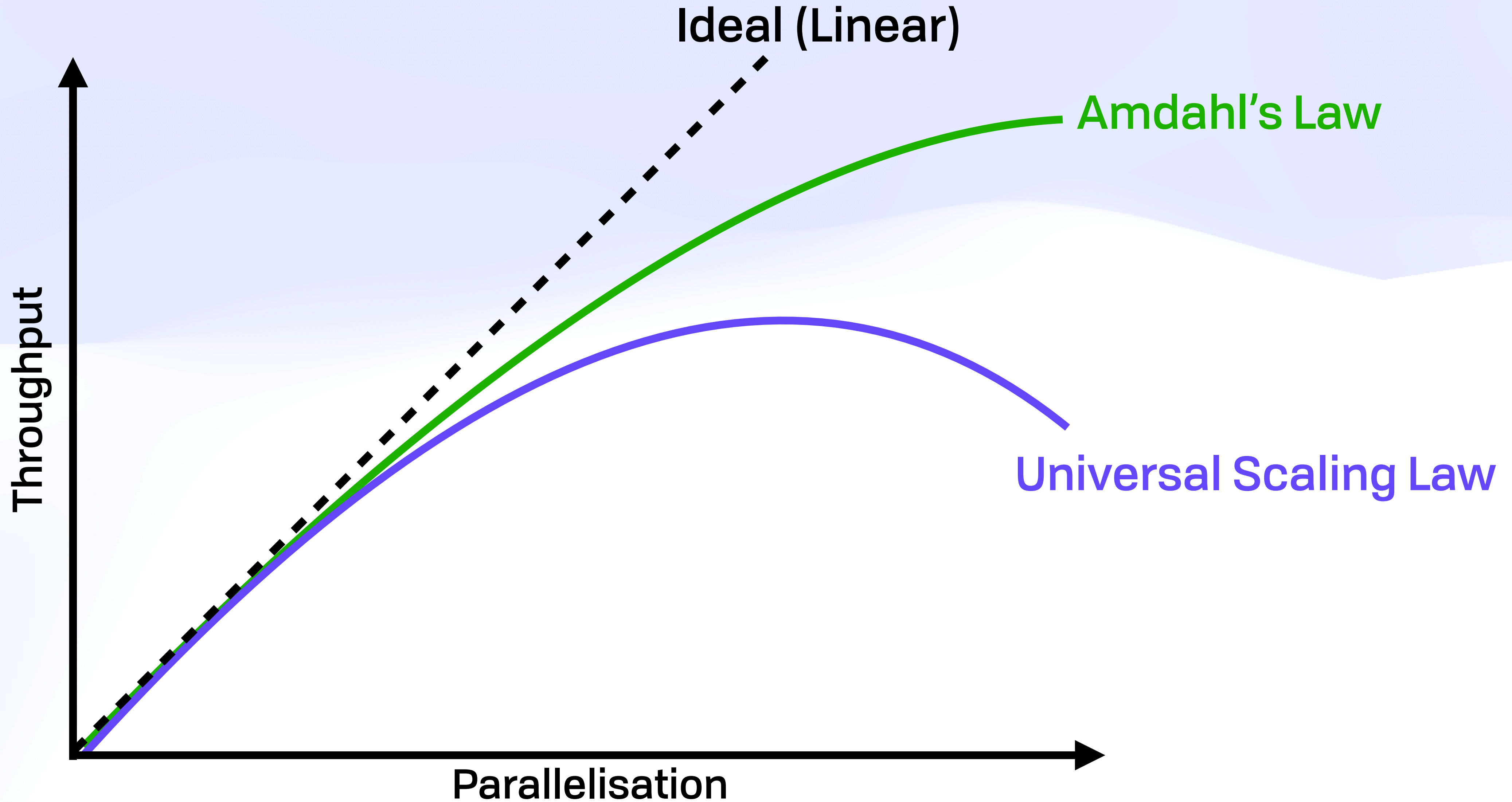
Compute Substrate

With a Little Help From My Friends



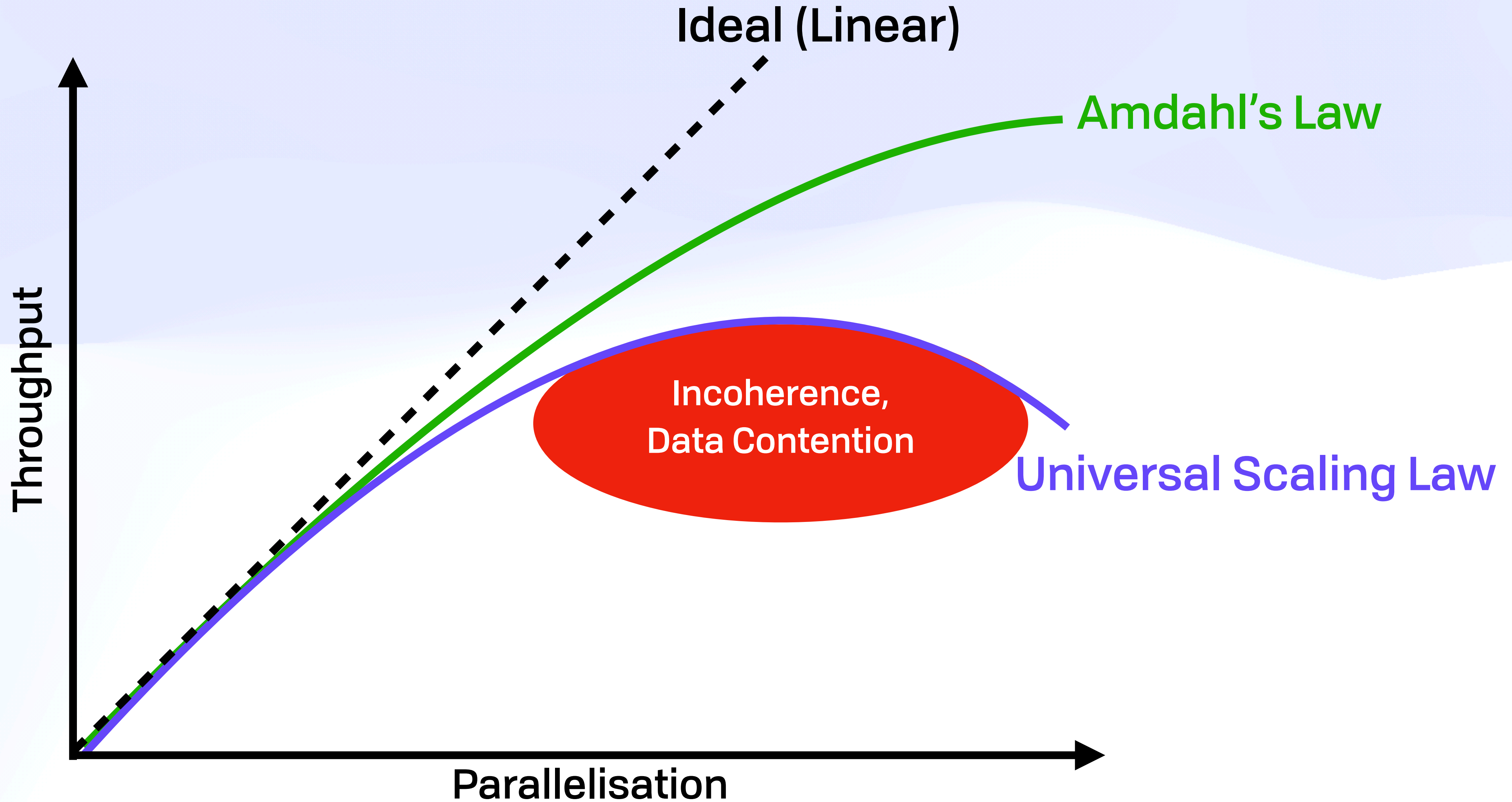
Compute Substrate

With a Little Help From My Friends



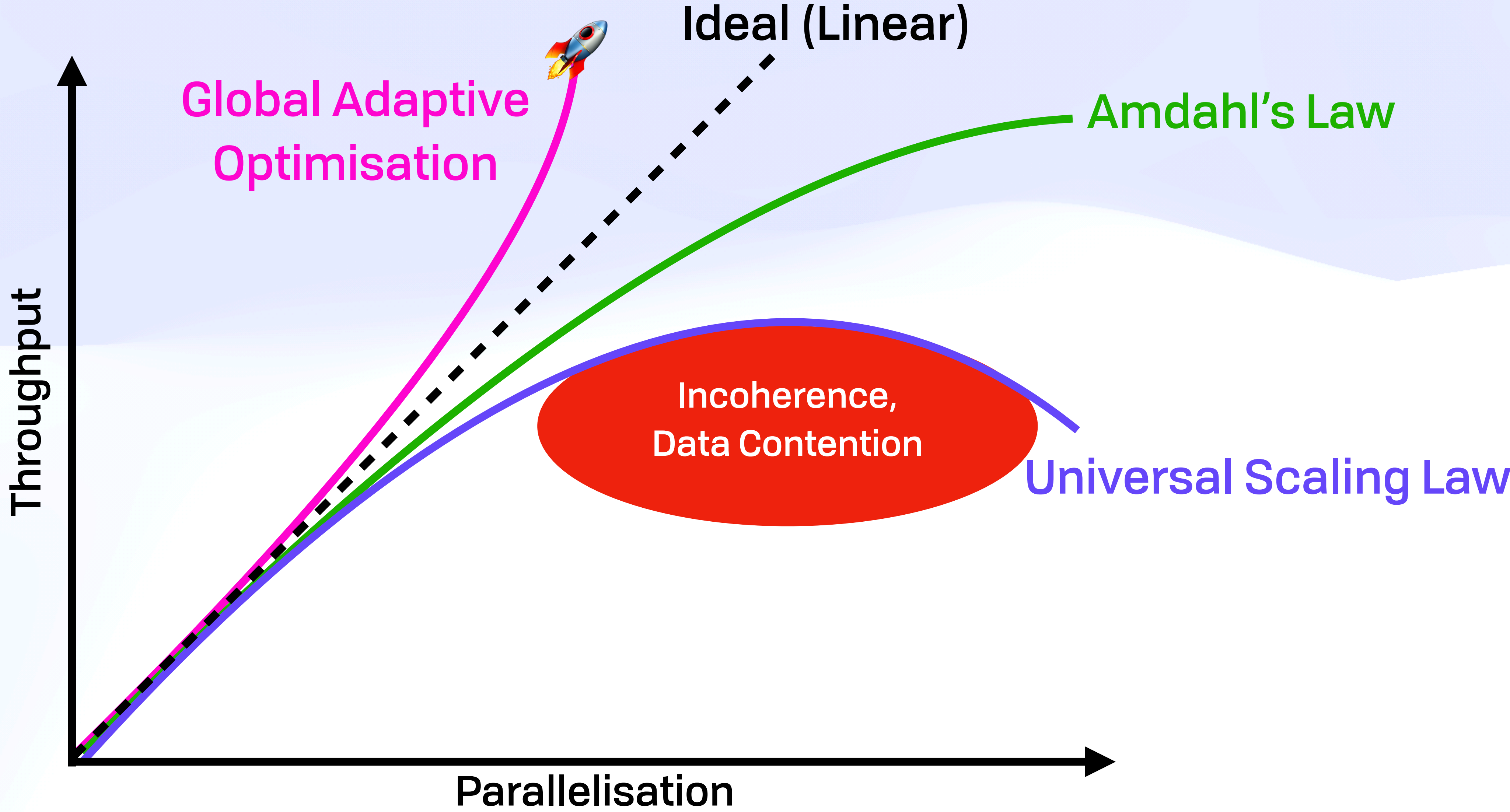
Compute Substrate

With a Little Help From My Friends



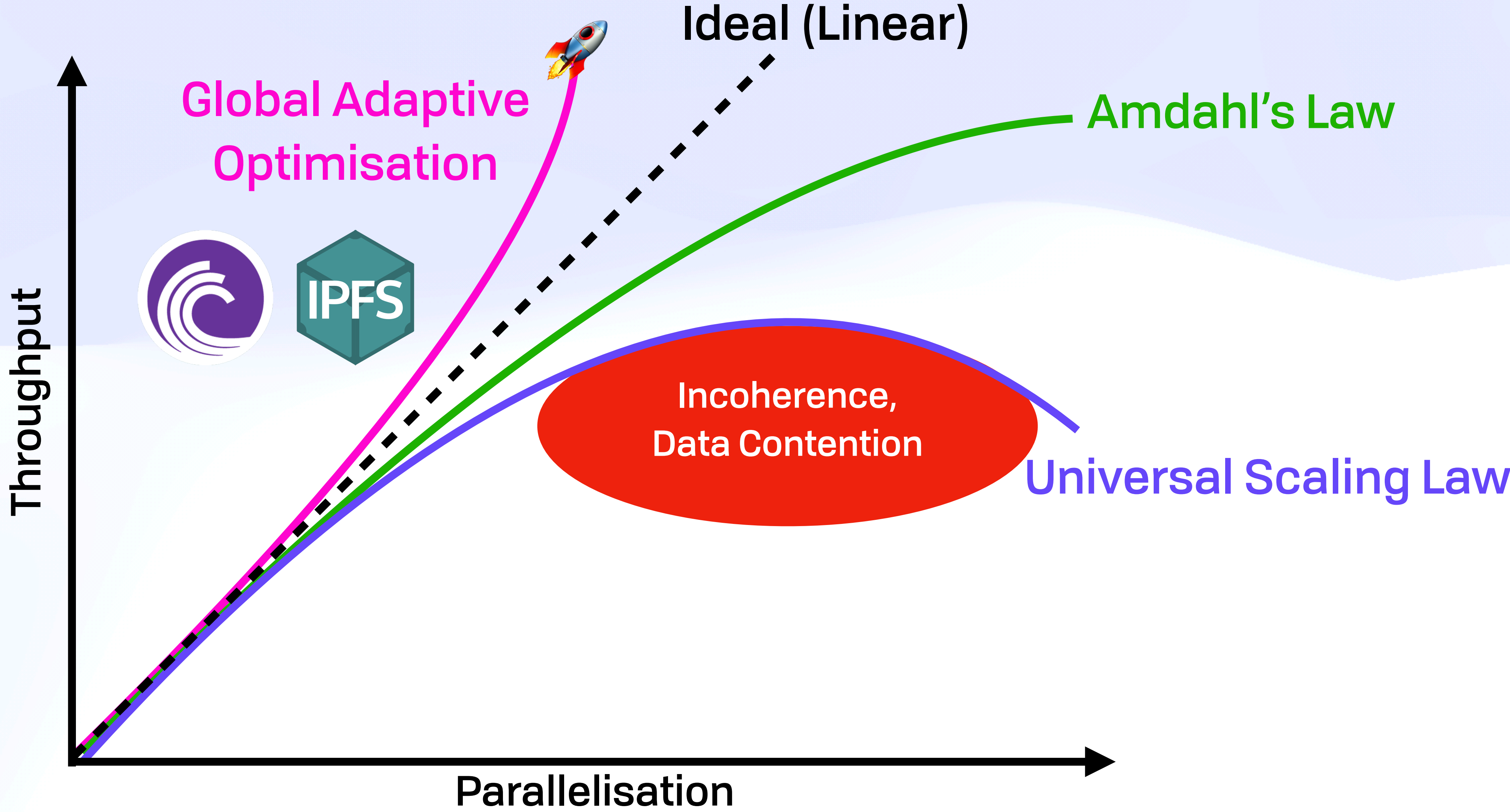
Compute Substrate

With a Little Help From My Friends



Compute Substrate

With a Little Help From My Friends





The image depicts two men dressed as construction workers, wearing orange hard hats and high-visibility yellow safety vests over grey tank tops. They are both shouting with their mouths wide open and their fists clenched in a celebratory gesture. The man on the left has a beard and glasses, while the man on the right has long hair, a beard, and glasses. They are surrounded by a vibrant, fiery atmosphere with streaks of green and purple light and a large, bright orange and yellow flame on the right side. The overall scene is one of intense joy and energy.

The Safety Dance

— Out of the Locality Tar Pit —

The Safety Dance 🕺

The Safety Dance

If their application can be cast as ***pure data processing***, they benefit from the past 40-50 years of work from the database community, [and] ***completely isolate the developer from the possibility of failure***

— **Goldstein et al**, AMBROSIA: Providing Performant Virtual Resiliency for Distributed Applications

The Safety Dance 

Virtual Resiliency

The Safety Dance 🕺

Virtual Resiliency

Mutable 🦋

Idempotent 🔄

Deterministic 📅
JUL 17


The Safety Dance 

Virtual Resiliency

Query A

Query B

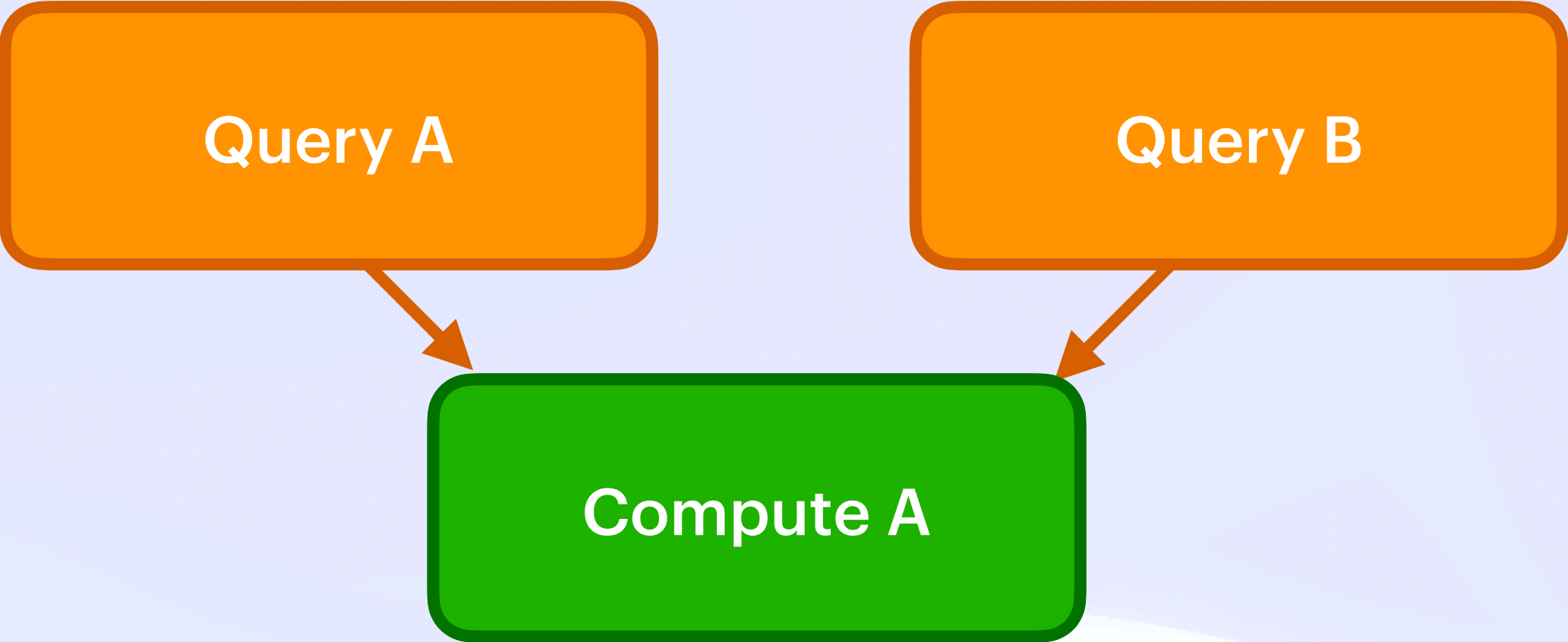
Mutable 

Idempotent 

Deterministic 

The Safety Dance

Virtual Resiliency



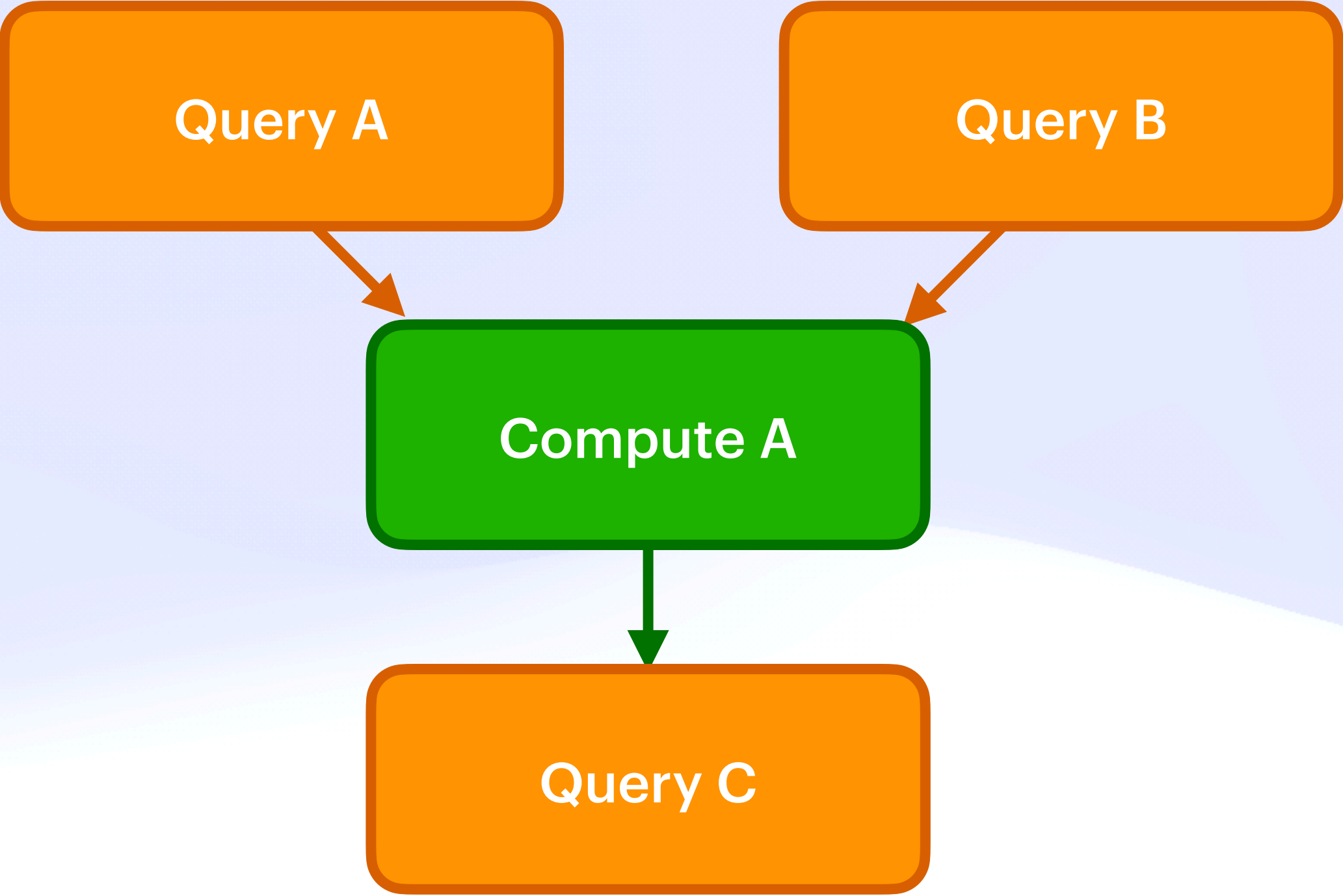
Mutable 

Idempotent 

Deterministic 

The Safety Dance 🧑

Virtual Resiliency



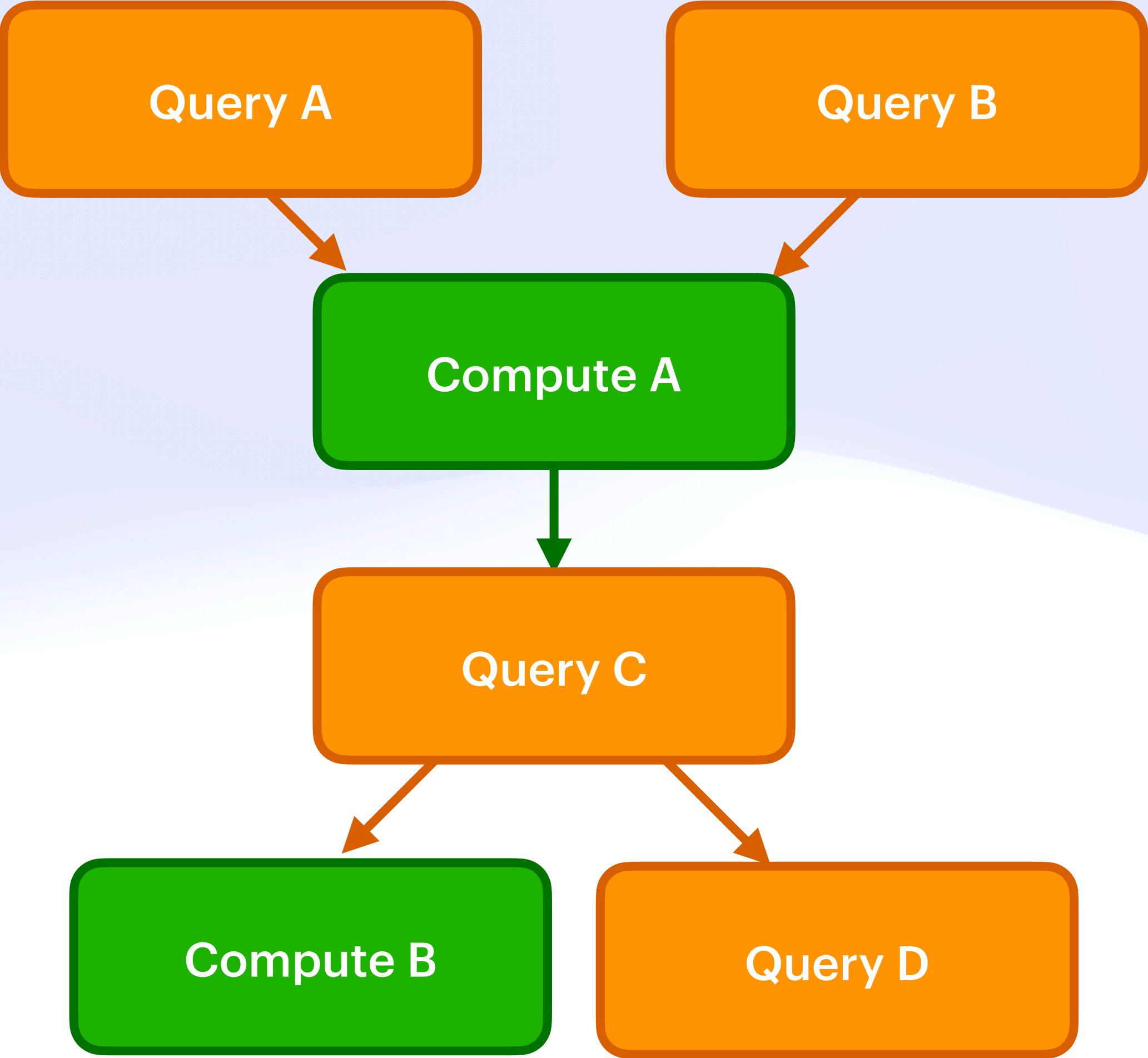
Mutable 🦋

Idempotent 🔄

Deterministic 📅 17

The Safety Dance 🕺

Virtual Resiliency



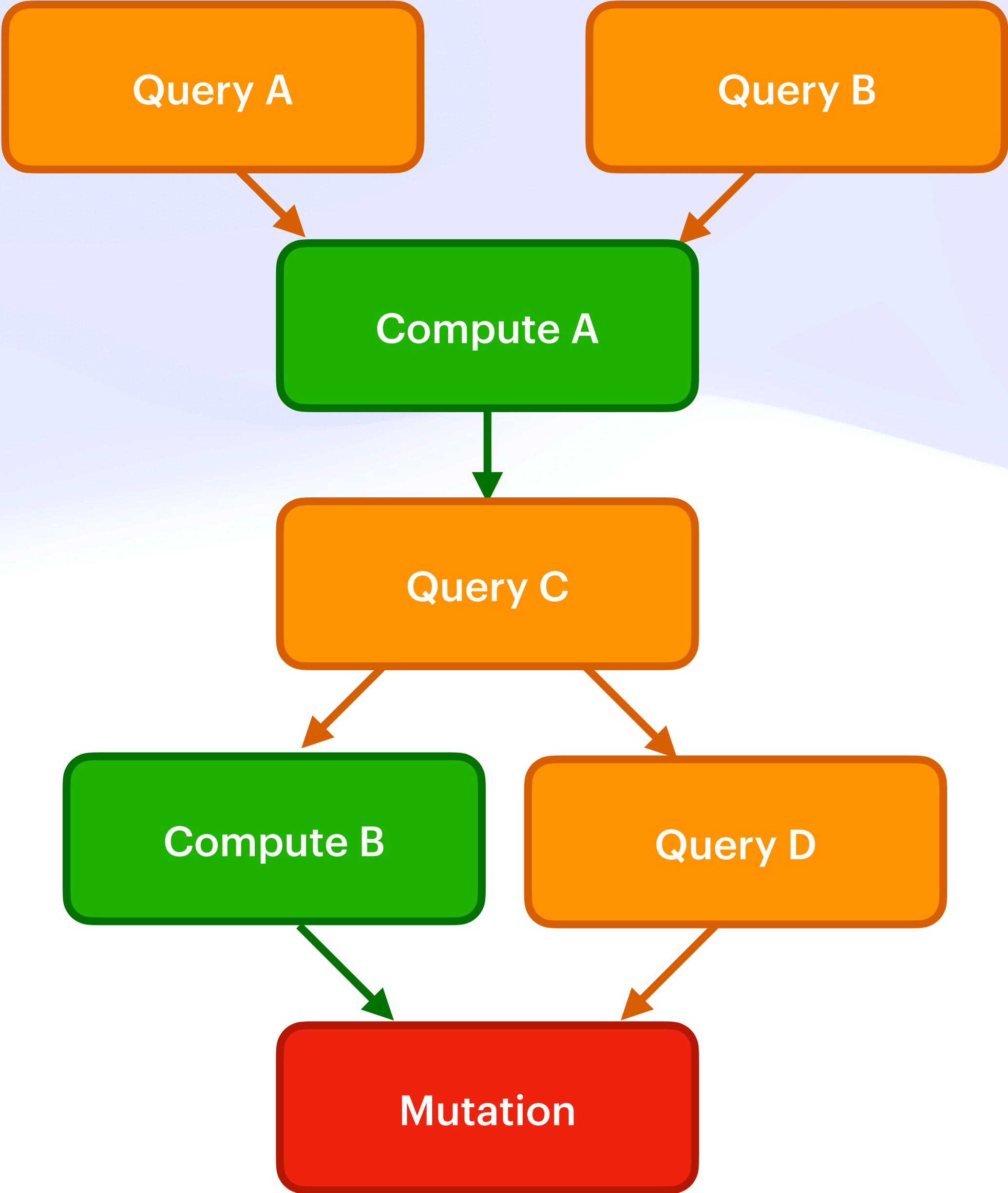
Mutable 🦋

Idempotent 🔄

Deterministic 📅 17

The Safety Dance 🧑

Virtual Resiliency



Mutable 🦋

Idempotent 🔄

Deterministic 📅 17

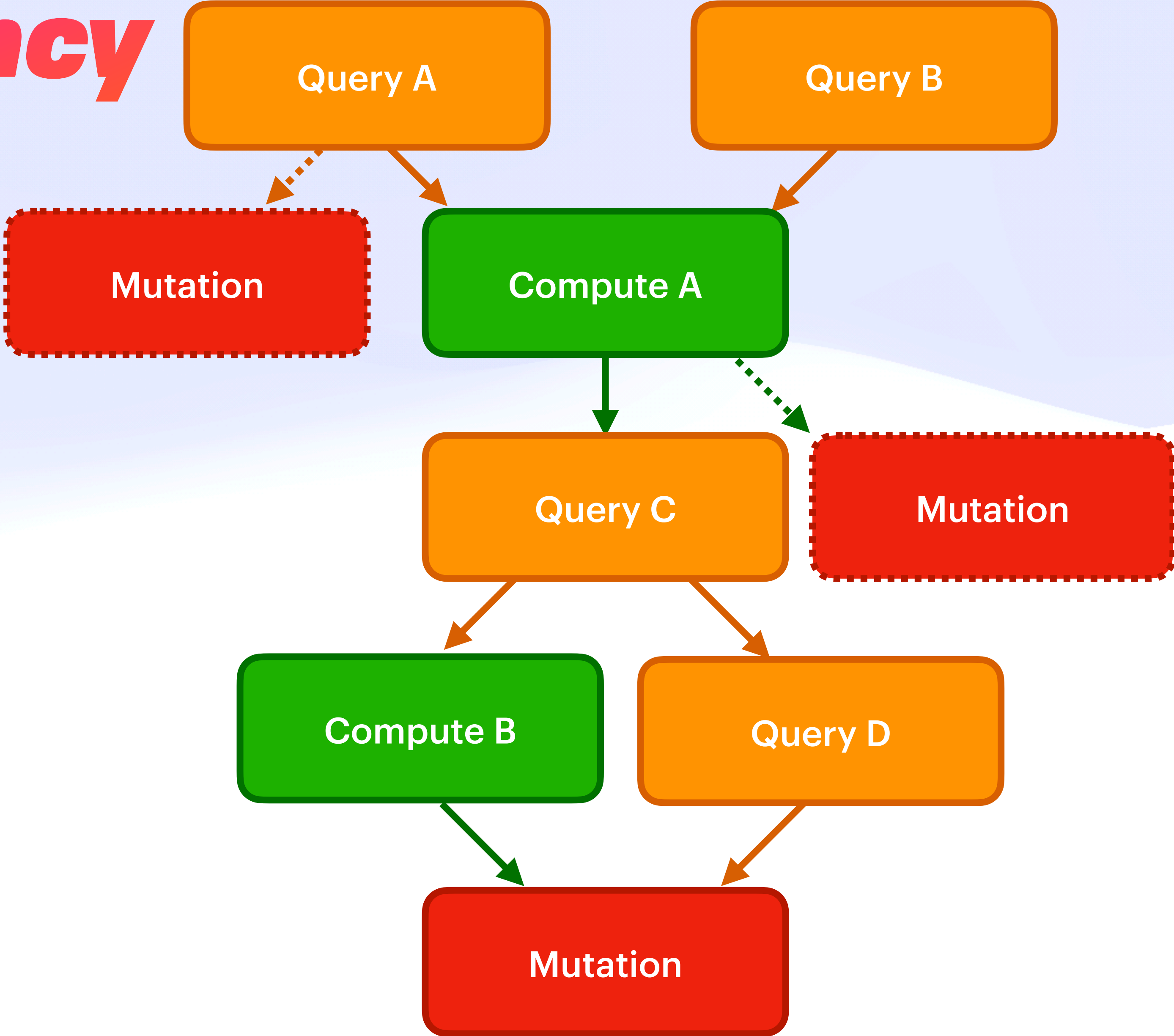
The Safety Dance 🧑

Virtual Resiliency

Mutable 🦋

Idempotent 🔄

Deterministic 📅 17



The Safety Dance 

Simplified Safe Layout

The Safety Dance 

Simplified Safe Layout

Queries

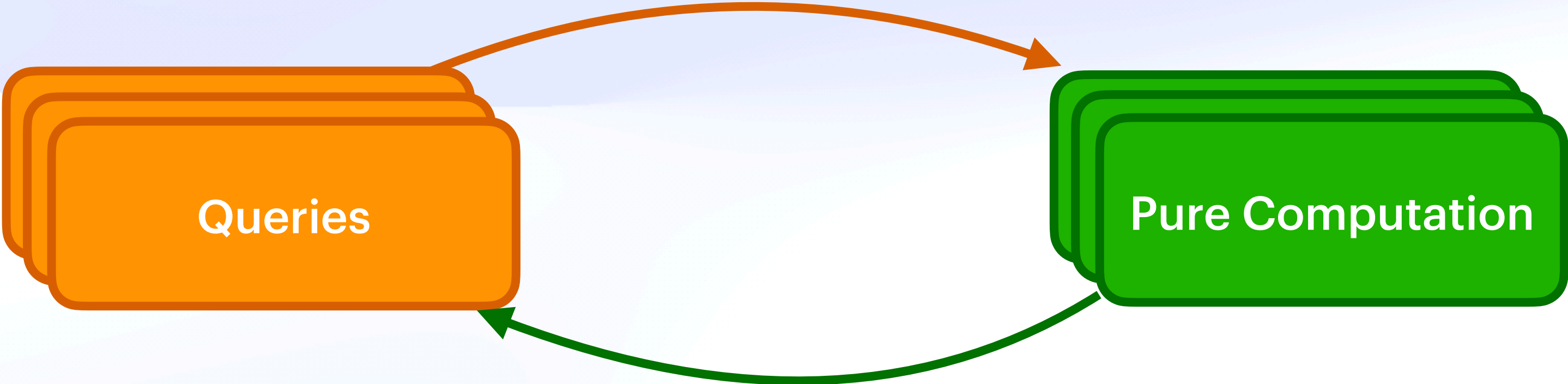
The Safety Dance 🕺

Simplified Safe Layout



The Safety Dance 🕺

Simplified Safe Layout



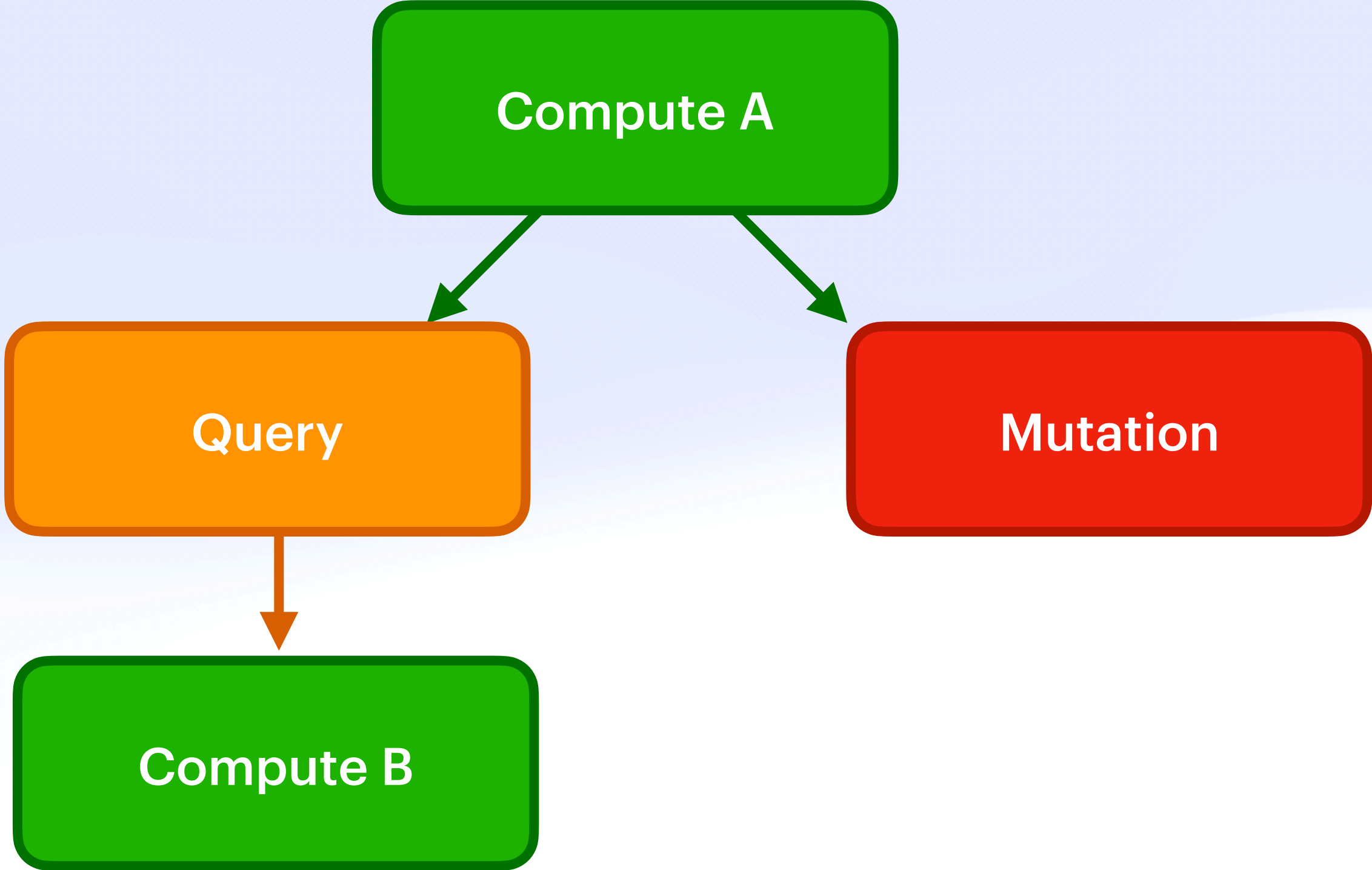
The Safety Dance 🧑

Simplified Safe Layout



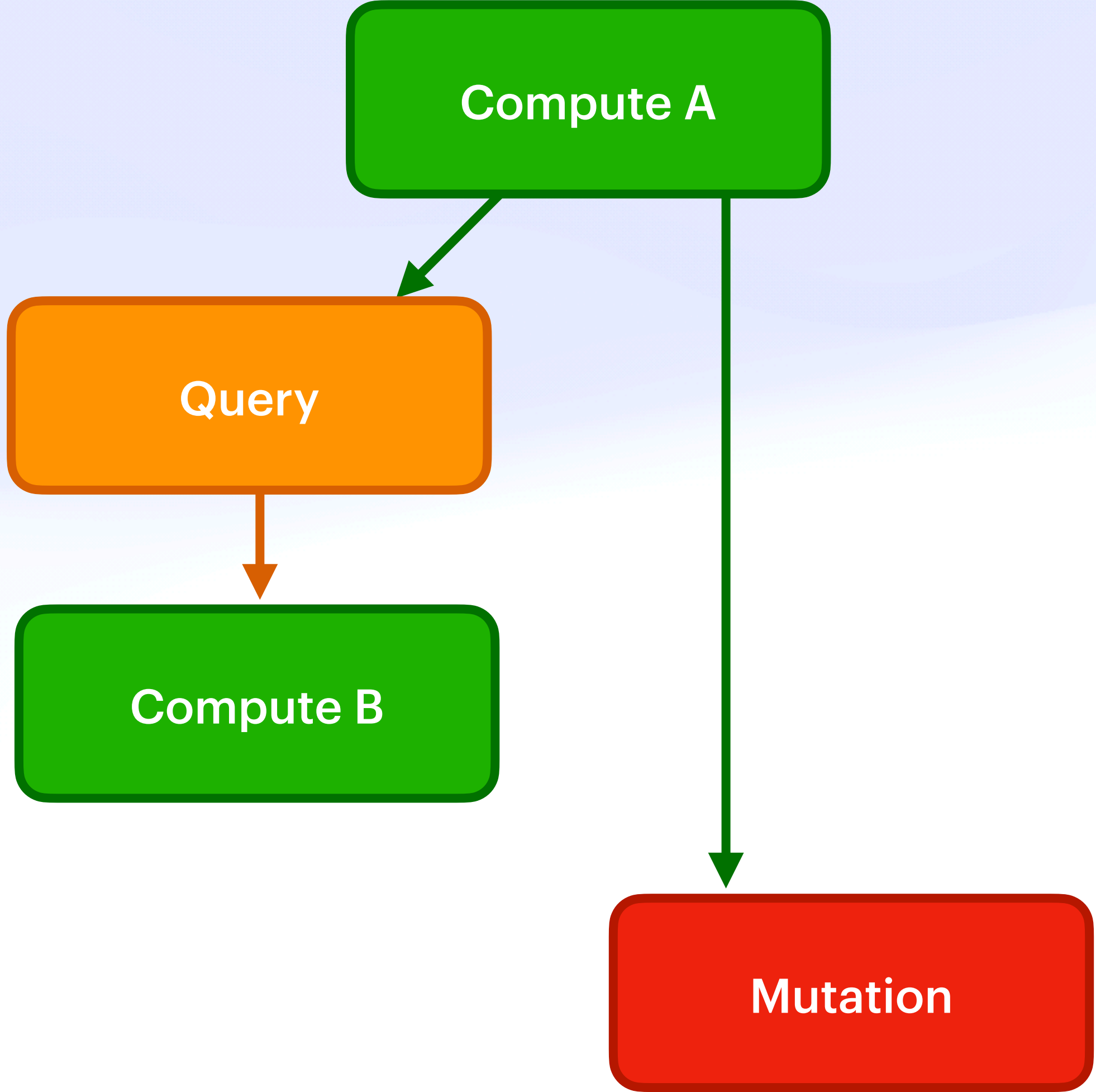
The Safety Dance 🕺

Simple Example



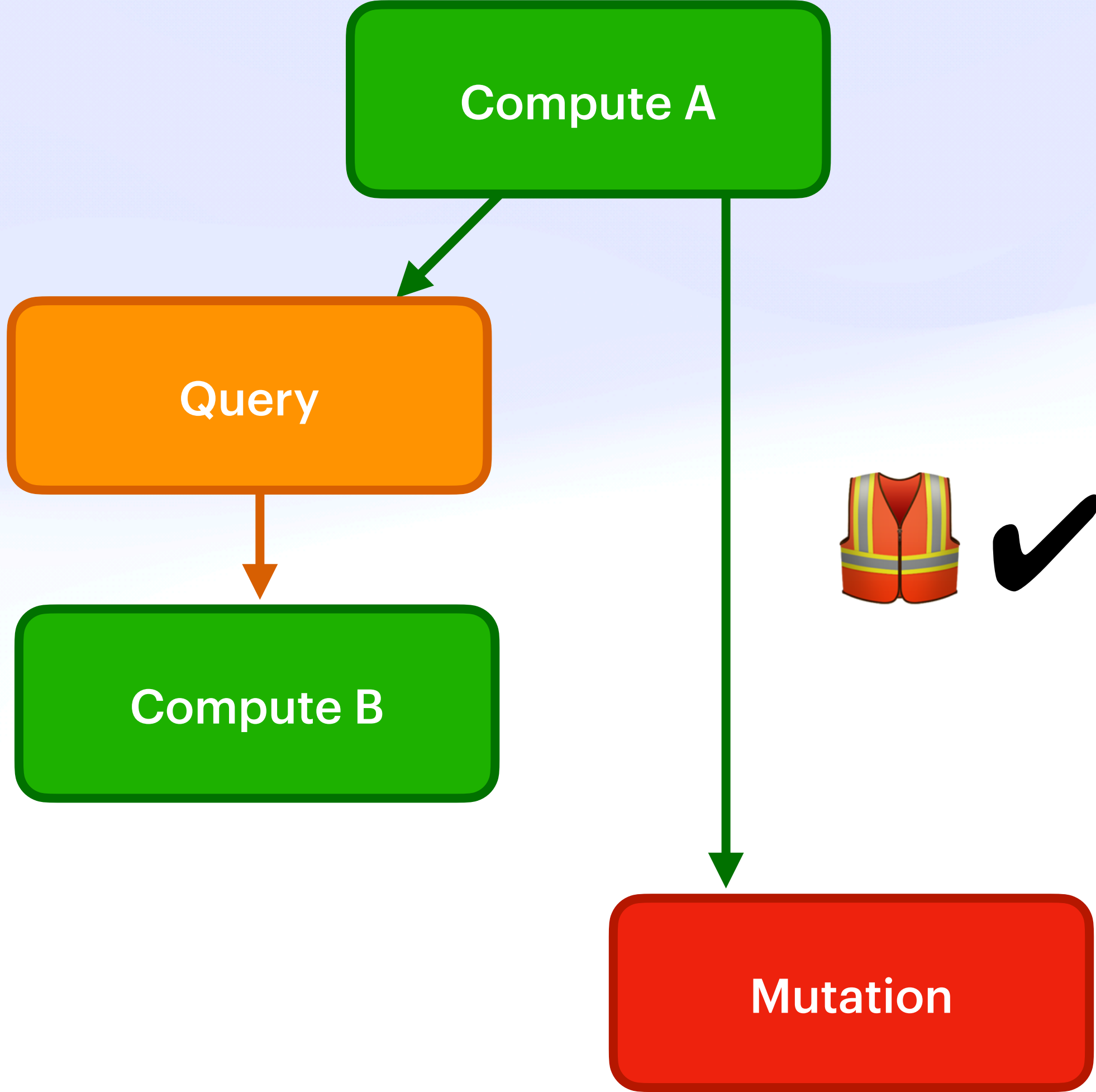
The Safety Dance 🕺

Simple Example



The Safety Dance 

Simple Example

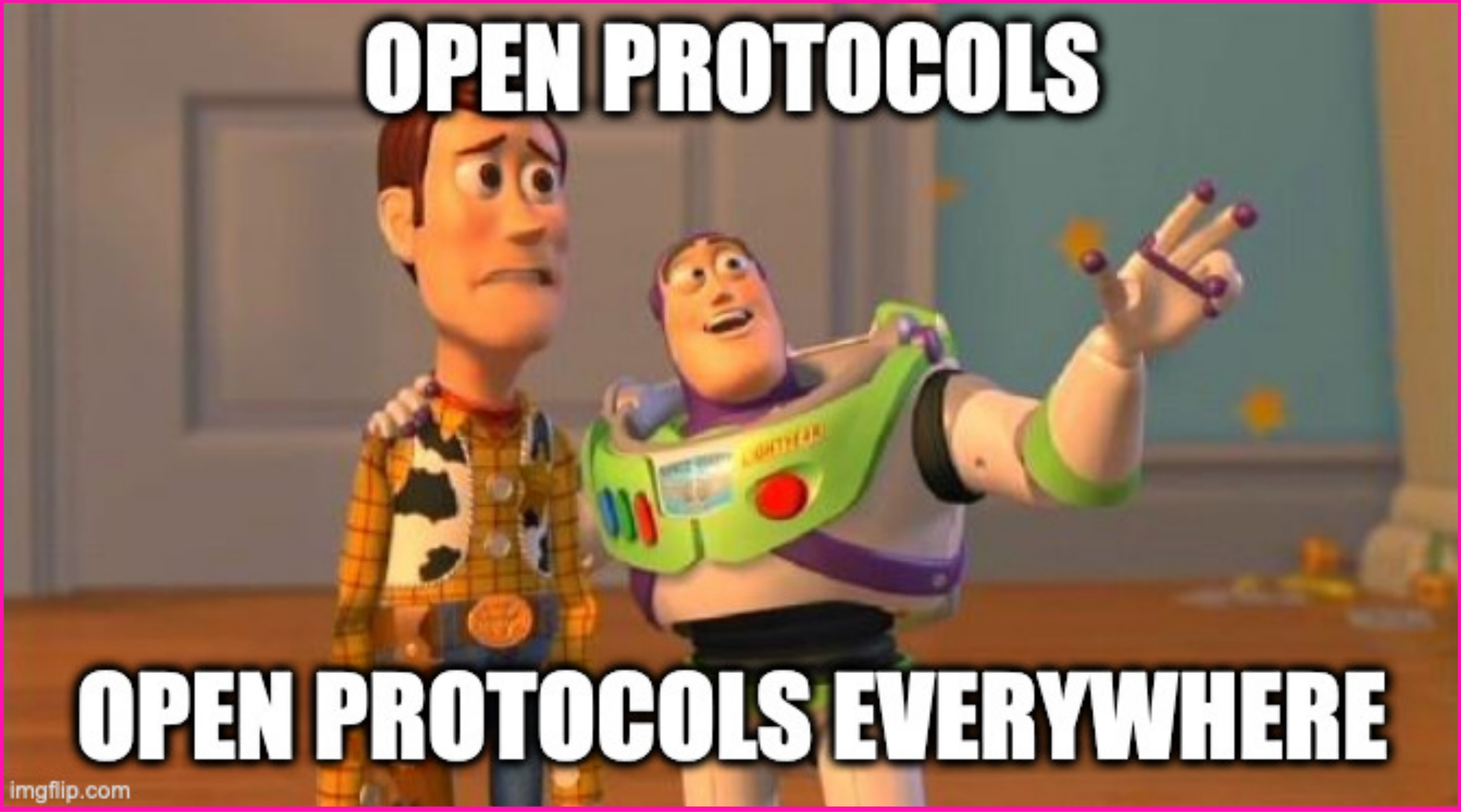




A large, vibrant red gift box is the central focus, wrapped in a glossy red paper. A wide, gold-colored ribbon is wrapped around it, forming a large, elegant bow on the top left. The box is positioned against a light, pale blue background. At the bottom of the frame, a diverse crowd of people is gathered, their heads and shoulders visible as they look up at the massive gift. The overall scene conveys a sense of anticipation and celebration.

Wrap Up

Wrap Up



OPEN PROTOCOLS EVERYWHERE

Wrap Up

Reusable/Remixable Specs

Wrap Up

Reusable/Remixable Specs

UCAN Core 

Distributed Authority

IPLD-WIT 

ABI

Varsig 

Signature Multiformat

Wrap Up

Reusable/Remixable Specs

UCAN Pipeline 🌊

Call Graph, Awaits, etc

UCAN Invocation ✨

Input Addressing, Execution, Memoization, etc

UCAN Core 📄

Distributed Authority

IPLD-WIT ⚙️

ABI

Varsig ✍️

Signature Multiformat

Wrap Up

Reusable/Remixable Specs

IPVM Task 

VM Config, Verification, etc

UCAN Pipeline 

Call Graph, Awaits, etc

UCAN Invocation 

Input Addressing, Execution, Memoization, etc

UCAN Core 

Distributed Authority

IPLD-WIT 

ABI

Varsig 

Signature Multiformat

Wrap Up

Reusable/Remixable Specs

IPVM Workflow 🎮

Transactions, Error Handling, Defaults

IPVM Task ⚙️

VM Config, Verification, etc

UCAN Pipeline 🌊

Call Graph, Awaits, etc

UCAN Invocation ✨

Input Addressing, Execution, Memoization, etc

UCAN Core 📄

Distributed Authority

IPLD-WIT ⚙️

ABI

Varsig ✍️

Signature Multiformat

Wrap Up

Reusable/Remixable Specs

IPVM Workflow 🎮

Transactions, Error Handling, Defaults

IPVM Task ⚙️

VM Config, Verification, etc

UCAN Pipeline 🌊

Call Graph, Awaits, etc

UCAN-Chan / ユーキャンちゃん

Consumable Channels

UCAN Invocation ✨

Input Addressing, Execution, Memoization, etc

UCAN Core 📄

Distributed Authority

IPLD-WIT ⚙️

ABI

Varsig ✍️

Signature Multiformat

github.com/ipvm-wg
lu.ma/ipvm



Thank You, Boston



brooklyn@fission.codes
discord.gg/fissioncodes
github.com/expede

github.com/ipvm-wg
lu.ma/ipvm



Thank You, Boston



brooklyn@fission.codes
discord.gg/fissioncodes
github.com/expede



github.com/ipvm-wg
lu.ma/ipvm



Thank You, Boston



brooklyn@fission.codes
discord.gg/fissioncodes
github.com/expede