Interplanetary Virtual Machine

🤹‍♀️ Content Addressed Compute for an Open World 🤝

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

Sometimes I think
the only universal
in the computing field is
the ***fetch-execute-cycle***.

Alan Perlis, Epigrams on Programming #44

IPVM

Brooklyn Zelenka @expede

github.com/expede

IPVM

Brooklyn Zelenka @expede

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

github.com/expede

IPVM

Brooklyn Zelenka @expede

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

github.com/expede

IPVM

Greatest Hits 🪩🕺

IPVM

Greatest Hits 🪩🕺

How we got here

What is an "IPVM" anyway?

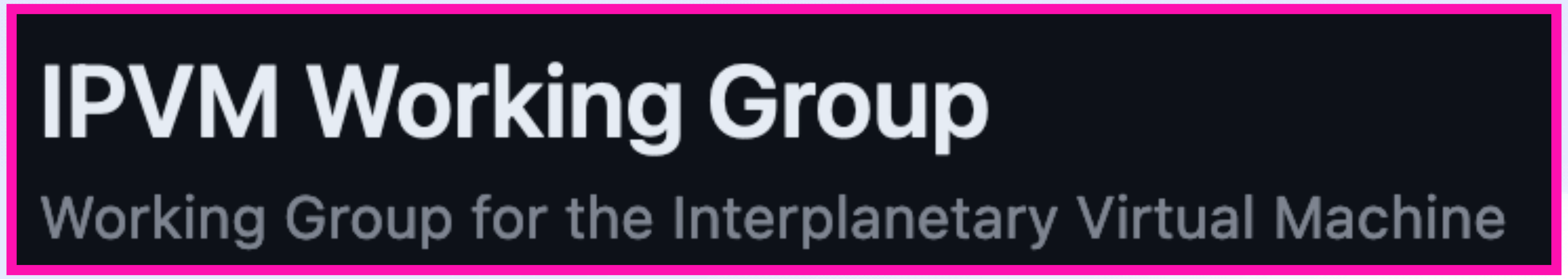
What we've learned

How to get involved

IPVM

🤝 Brought To You By...

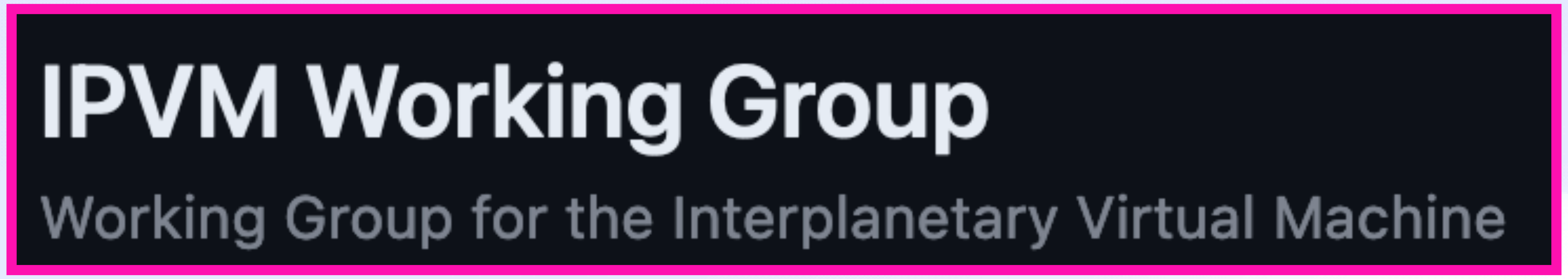
🤝 Brought To You By...



IPVM Working Group

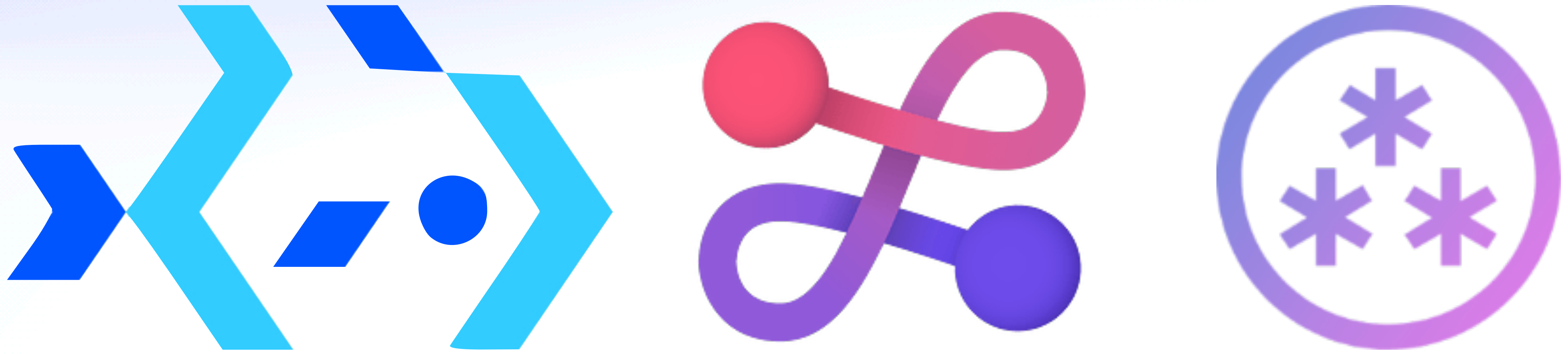
Working Group for the Interplanetary Virtual Machine

🤝 Brought To You By...

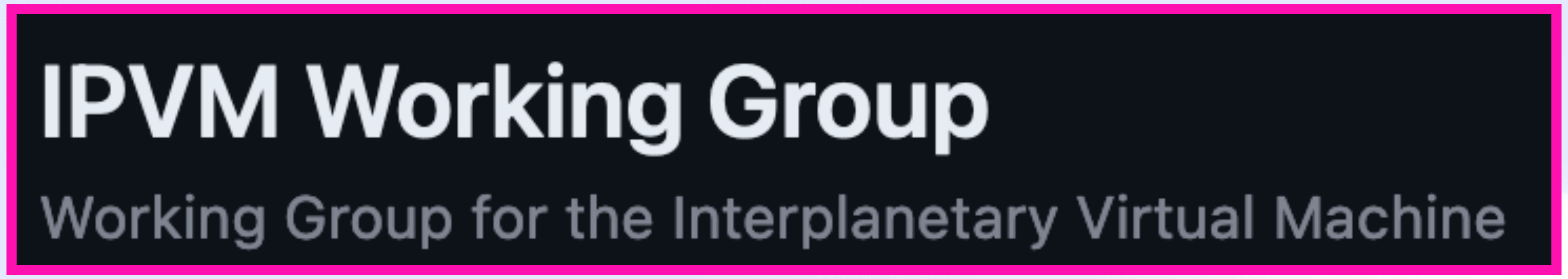


IPVM Working Group

Working Group for the Interplanetary Virtual Machine



🤝 Brought To You By...



IPVM Working Group

Working Group for the Interplanetary Virtual Machine

IPVM

Timeline

Timeline

2023

IPVM

Timeline

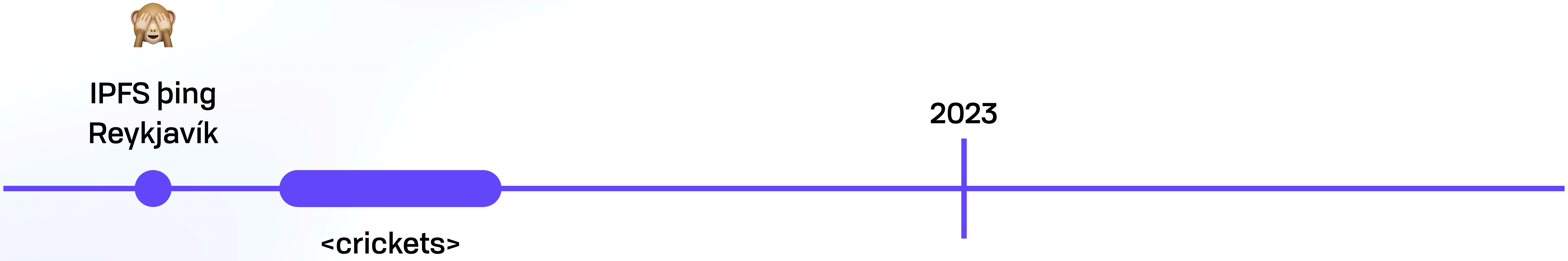
🙈

IPFS þing
Reykjavík

2023

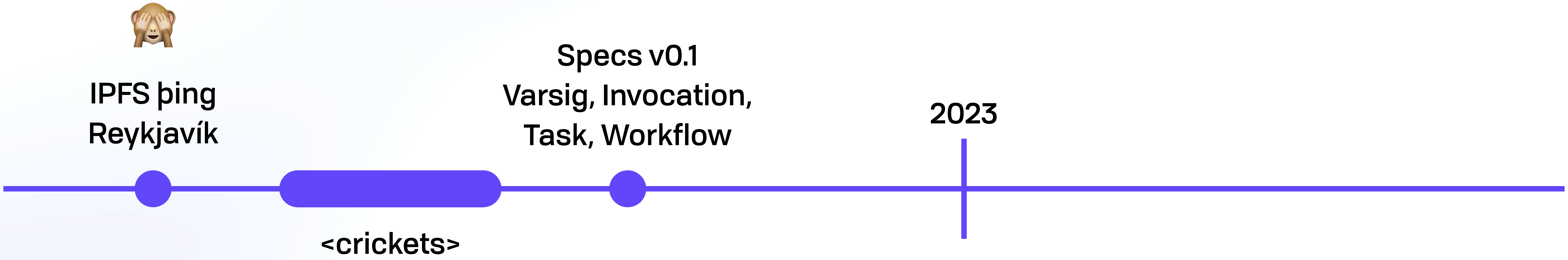
IPVM

Timeline



IPVM

Timeline



🙈

IPFS þing
Reykjavík

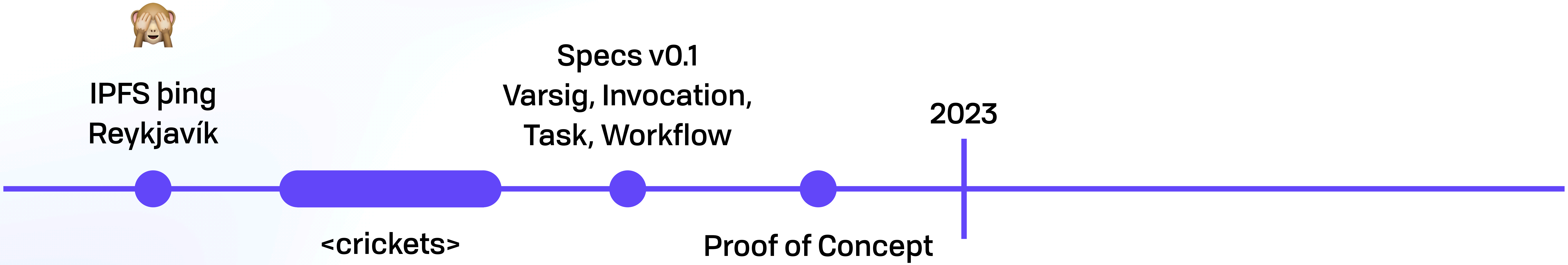
<crickets>

Specs v0.1
Varsig, Invocation,
Task, Workflow

2023

IPVM

Timeline



🙈

IPFS þing
Reykjavík

<crickets>

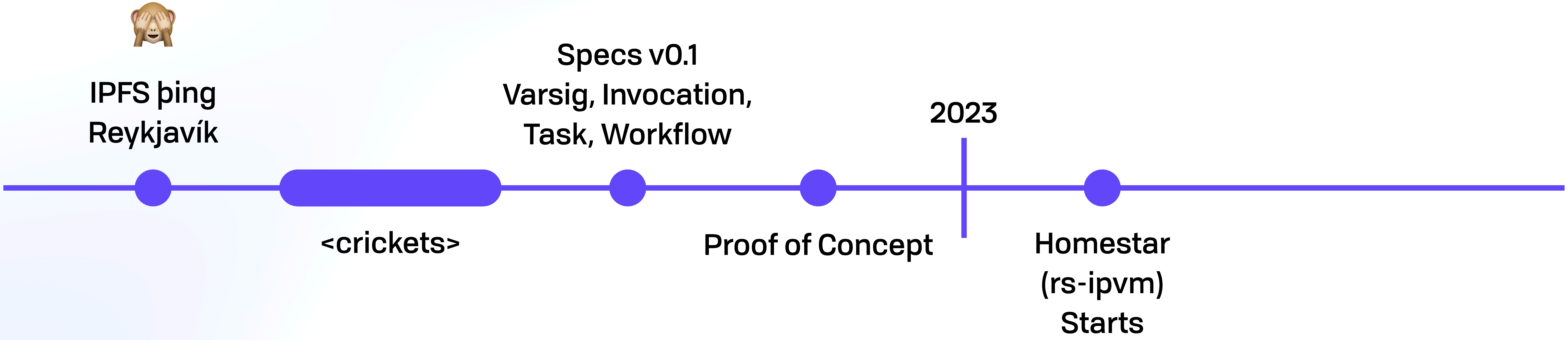
Specs v0.1
Varsig, Invocation,
Task, Workflow

Proof of Concept

2023

IPVM

Timeline



🙈

IPFS þing
Reykjavík

<crickets>

Specs v0.1
Varsig, Invocation,
Task, Workflow

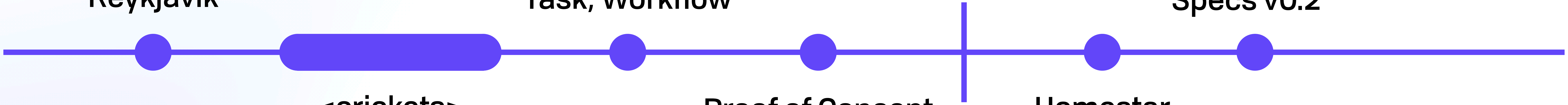
Proof of Concept

2023

Homestar
(rs-ipvm)
Starts

IPVM

Timeline



🙈

IPFS þing
Reykjavík

<crickets>

Specs v0.1
Varsig, Invocation,
Task, Workflow

Proof of Concept

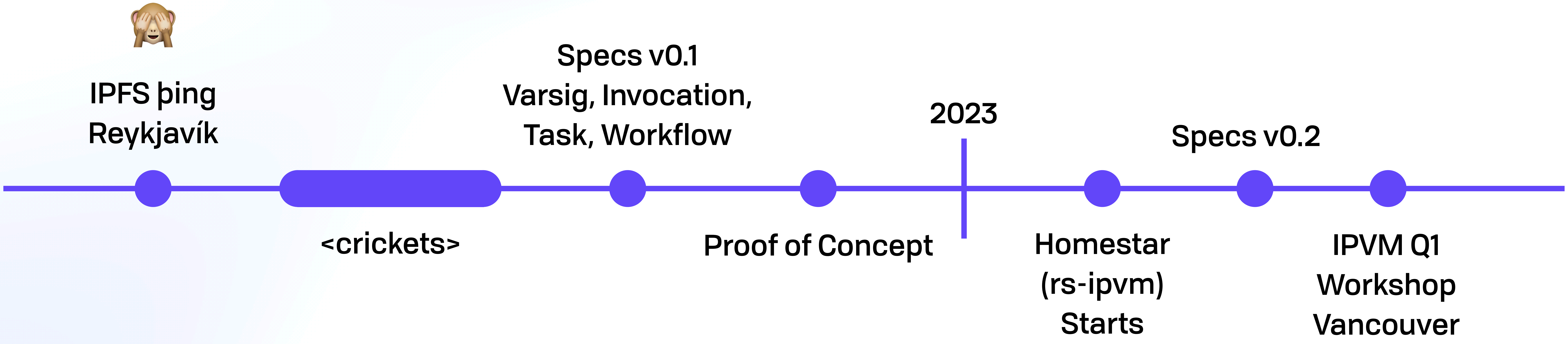
2023

Homestar
(rs-ipvm)
Starts

Specs v0.2

IPVM

Timeline



🙈

IPFS þing
Reykjavík

<crickets>

Specs v0.1
Varsig, Invocation,
Task, Workflow

Proof of Concept

2023

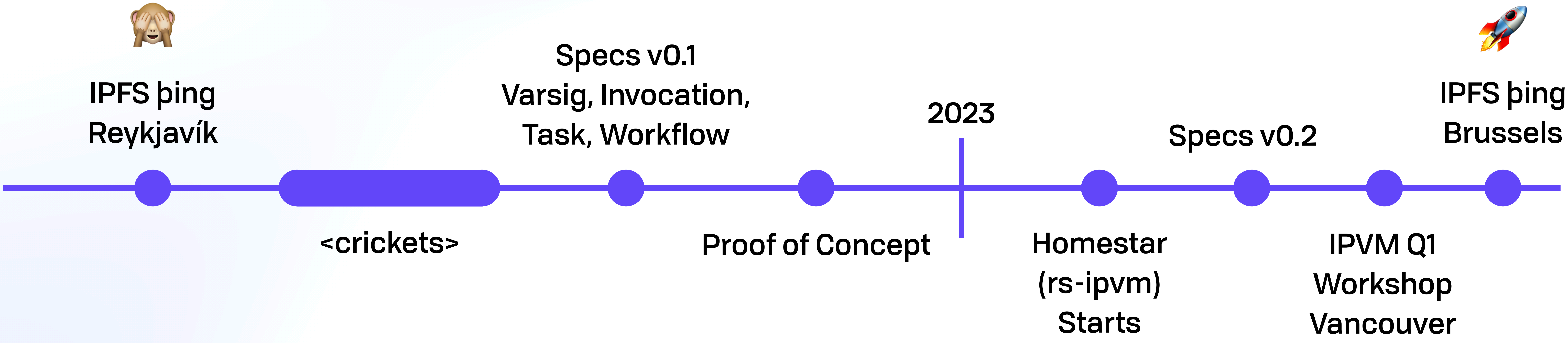
Homestar
(rs-ipvm)
Starts

Specs v0.2

IPVM Q1
Workshop
Vancouver

IPVM

Timeline



🙈

IPFS þing
Reykjavík

<crickets>

Specs v0.1
Varsig, Invocation,
Task, Workflow

Proof of Concept

2023

Homestar
(rs-ipvm)
Starts

Specs v0.2

IPVM Q1
Workshop
Vancouver

🚀

IPFS þing
Brussels

What Is An IPVM?

🤔

What is an IPVM 🤔

The HTTP of Compute 🤩

What is an IPVM 🤔

The HTTP of Compute 🤩

- Compute — like data — should be a **ubiquitous** commodity
- End users & IPFS teams can **depend** on having compute around
- Fully **consistent** functionality between clients
- **Replace** (e.g.) AWS Lambda with an **open protocol & nodes**

UCAN Decentralize Auth

Everywhere

UCAN Decentralize Auth *Everywhere*

Far Edge
Commons
Cloud & Edge

UCAN Decentralize Auth *Everywhere*

Far Edge

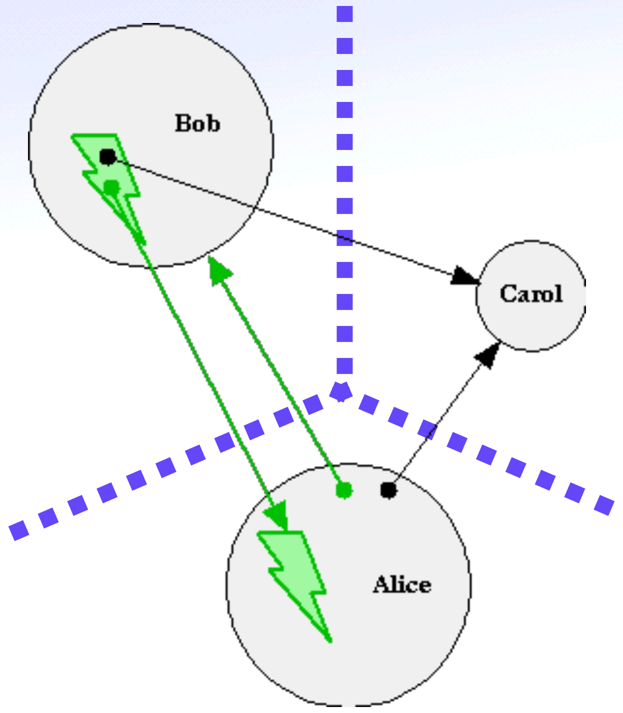
Commons

Cloud & Edge

UCAN Decentralize Auth
Everywhere

Commons

Far Edge



Cloud & Edge

What is an IPVM 🤔

Permissionless Interop

What is an IPVM 🤔

Permissionless Interop

🕵️‍♂️

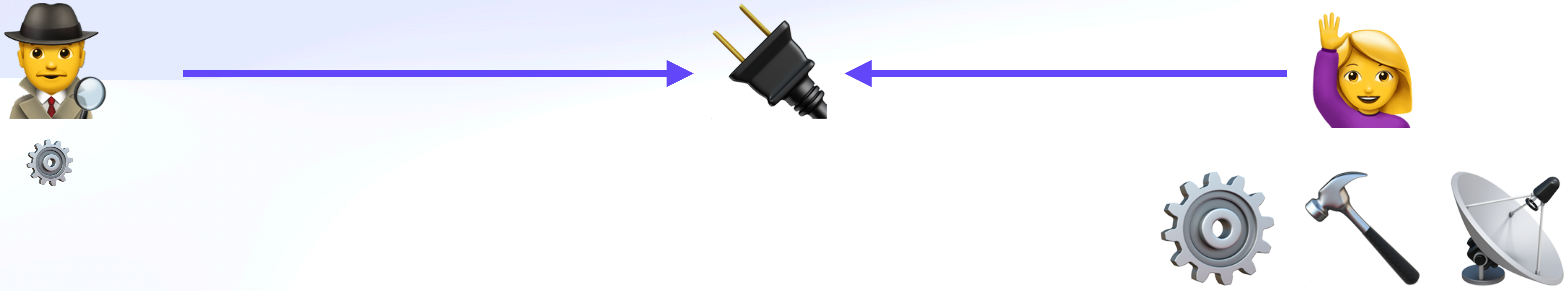
⚙️

What is an IPVM 🤔

Permissionless Interop

What is an IPVM 🤔

Permissionless Interop

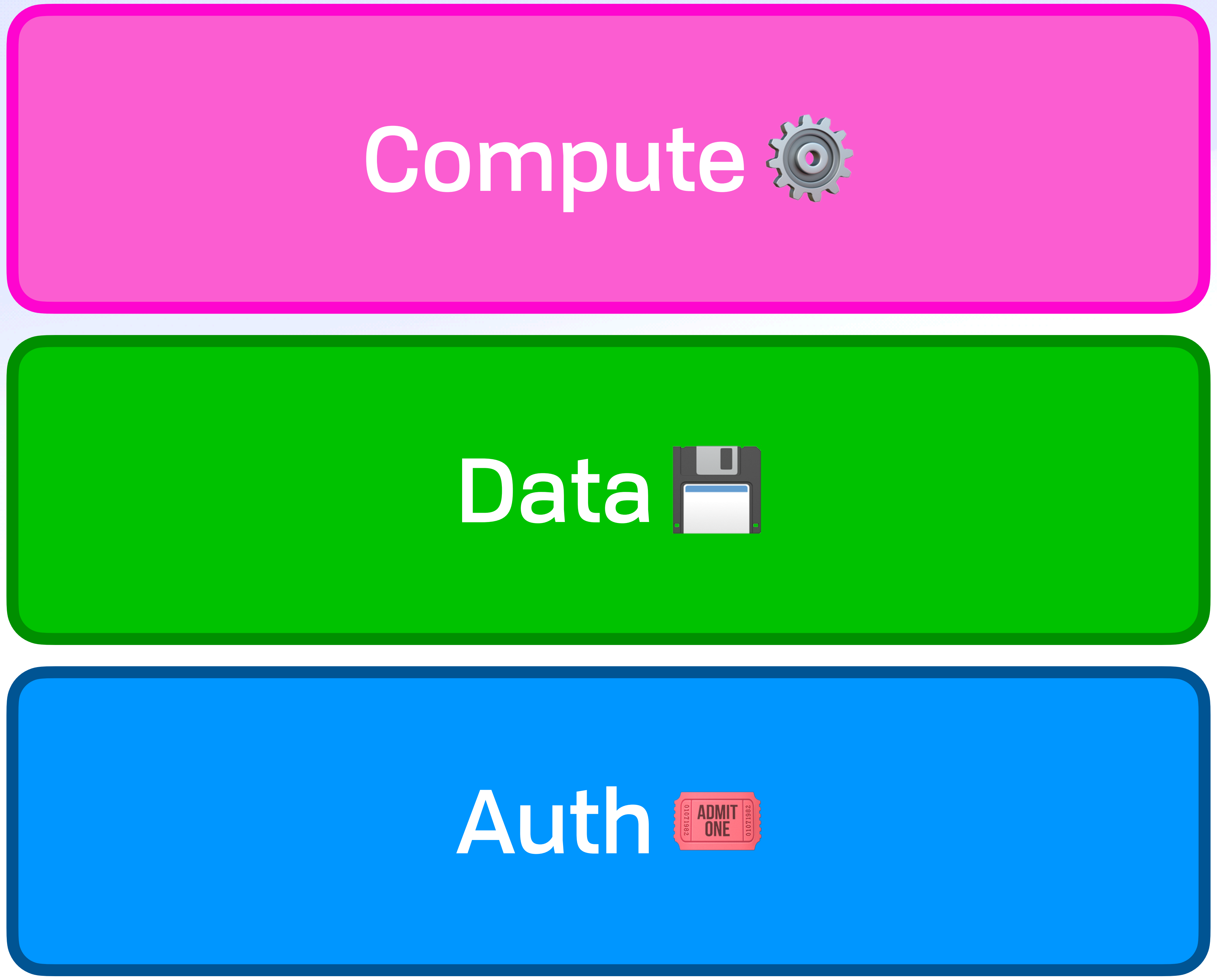


What is an IPVM 🤔

With Their Powers Combined

What is an IPVM 🤔

With Their Powers Combined



Compute ⚙️

Data 💾

Auth 🎟️

What is an IPVM 🤔

With Their Powers Combined

Compute ⚙️

Data 💾

Auth 🎟️

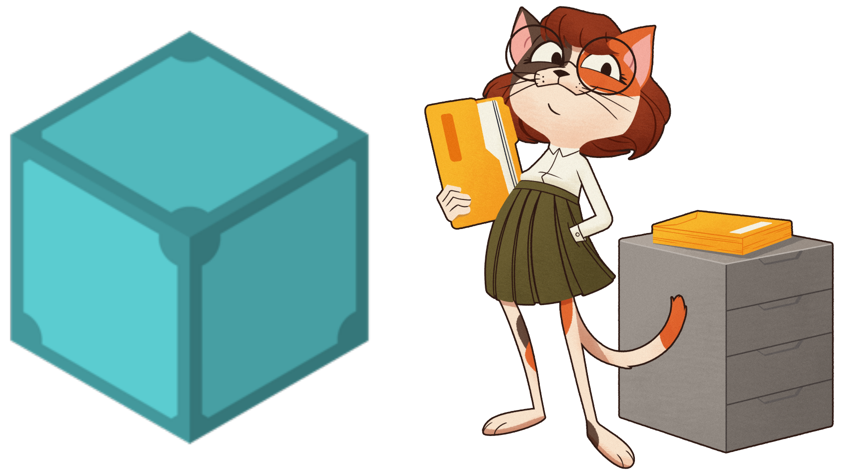
What is an IPVM 🤔

With Their Powers Combined

Compute ⚙️

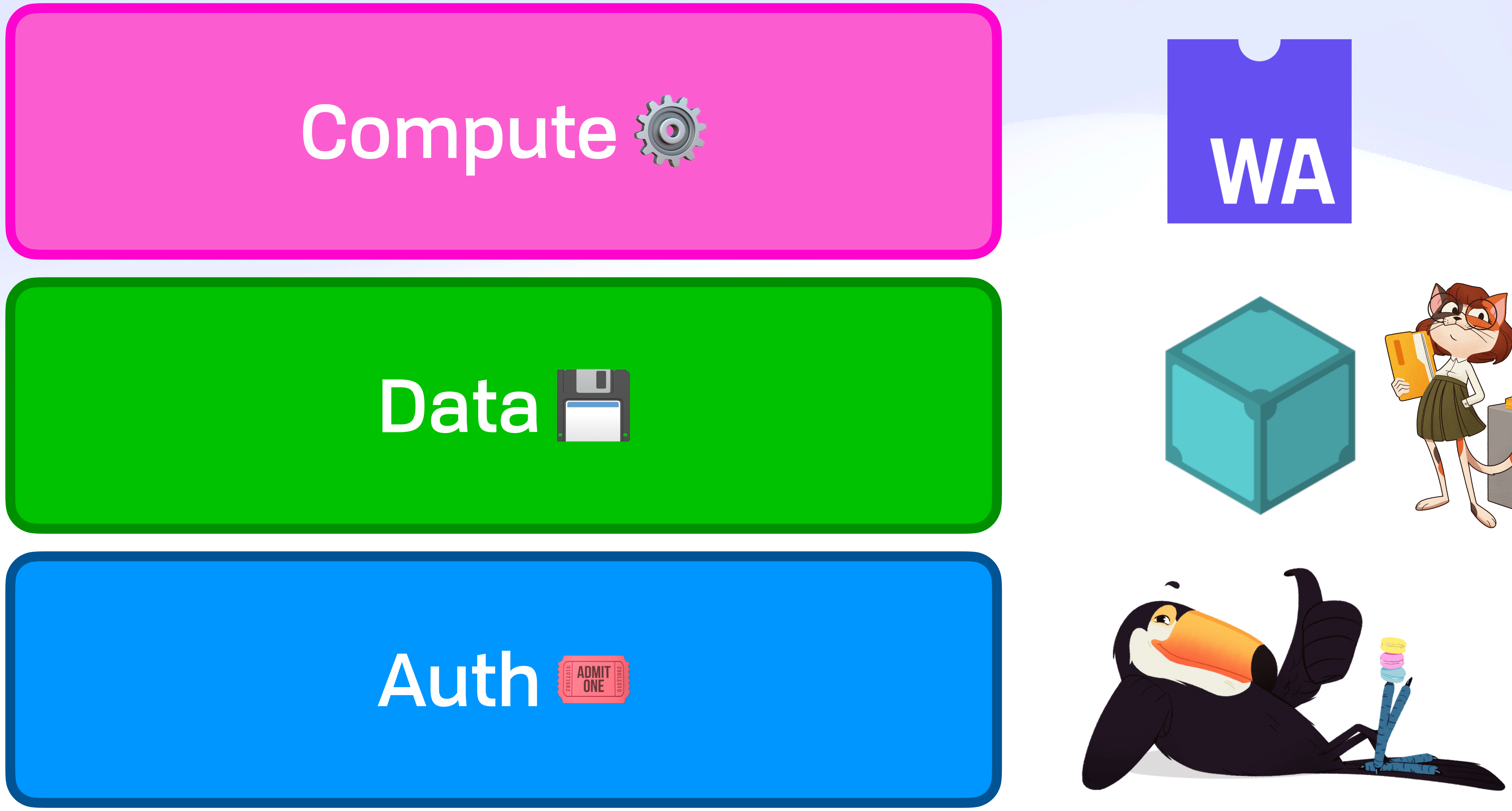
Data 💾

Auth 🎟️



What is an IPVM 🤔

With Their Powers Combined



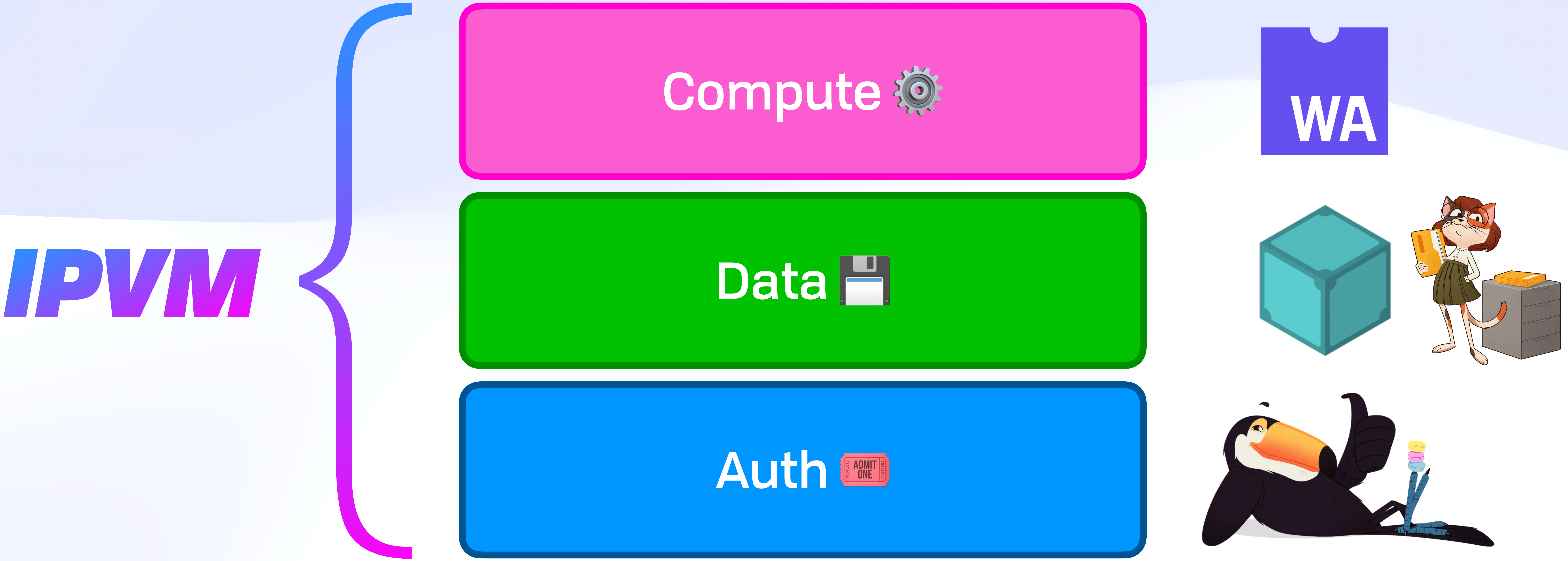
Compute ⚙️

Data 💾

Auth 🎟️

What is an IPVM 🤔

With Their Powers Combined



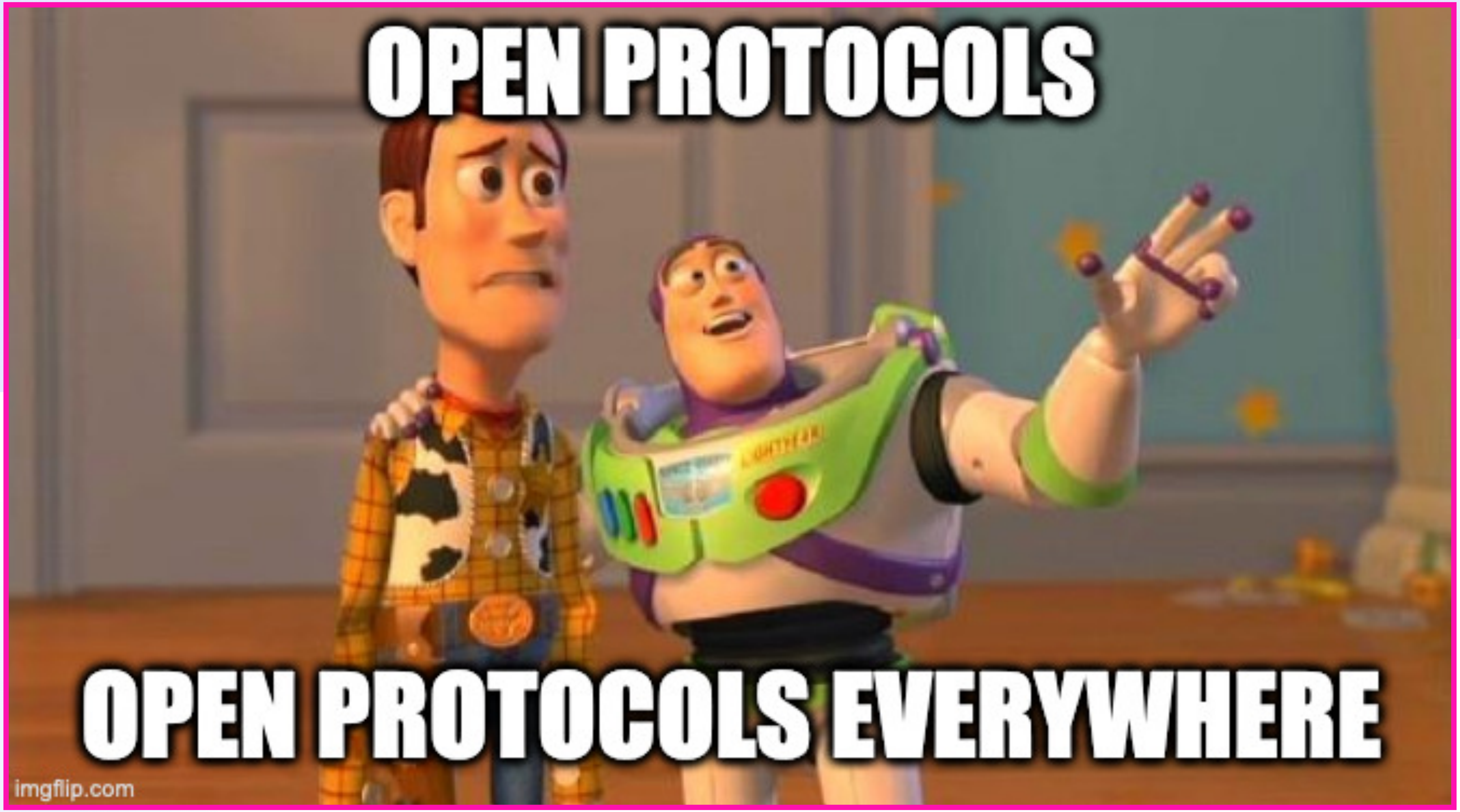
IPVM

Compute ⚙️

Data 💾

Auth 🎟️

What is an IPVM 🤔



What is an IPVM 🤔

Reusable Spec Stack

What is an IPVM 🤔

Reusable Spec Stack

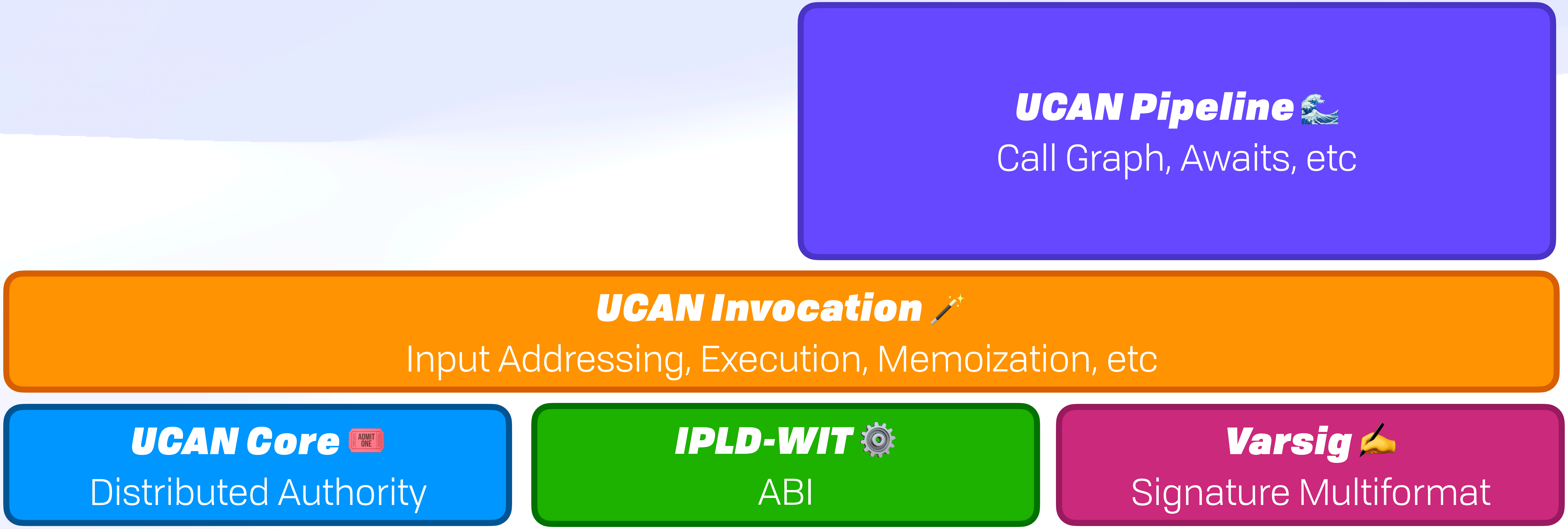
UCAN Core 🎟️
Distributed Authority

IPLD-WIT ⚙️
ABI

Varsig ✍️
Signature Multiformat

What is an IPVM 🤔

Reusable Spec Stack



UCAN Pipeline 🌊

Call Graph, Awaits, etc

UCAN Invocation 🪄

Input Addressing, Execution, Memoization, etc

UCAN Core 🎟️

Distributed Authority

IPLD-WIT ⚙️

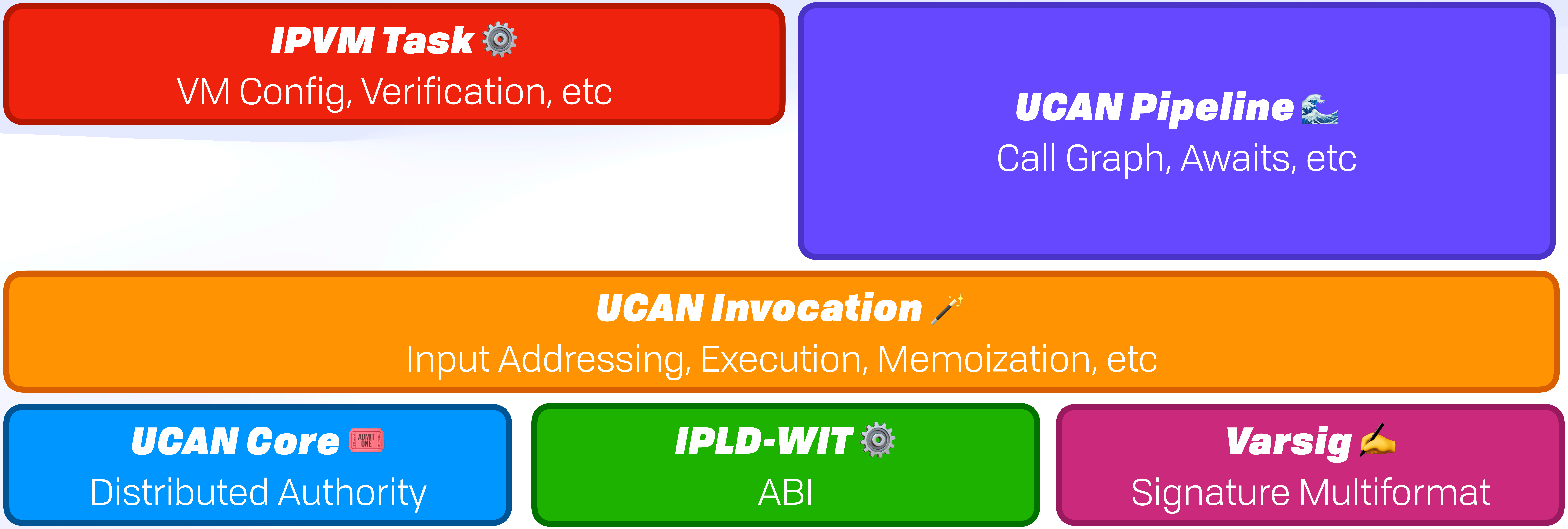
ABI

Varsig ✍️

Signature Multiformat

What is an IPVM 🤔

Reusable Spec Stack



IPVM Task ⚙️
VM Config, Verification, etc

UCAN Pipeline 🌊
Call Graph, Awaits, etc

UCAN Invocation 🪄
Input Addressing, Execution, Memoization, etc

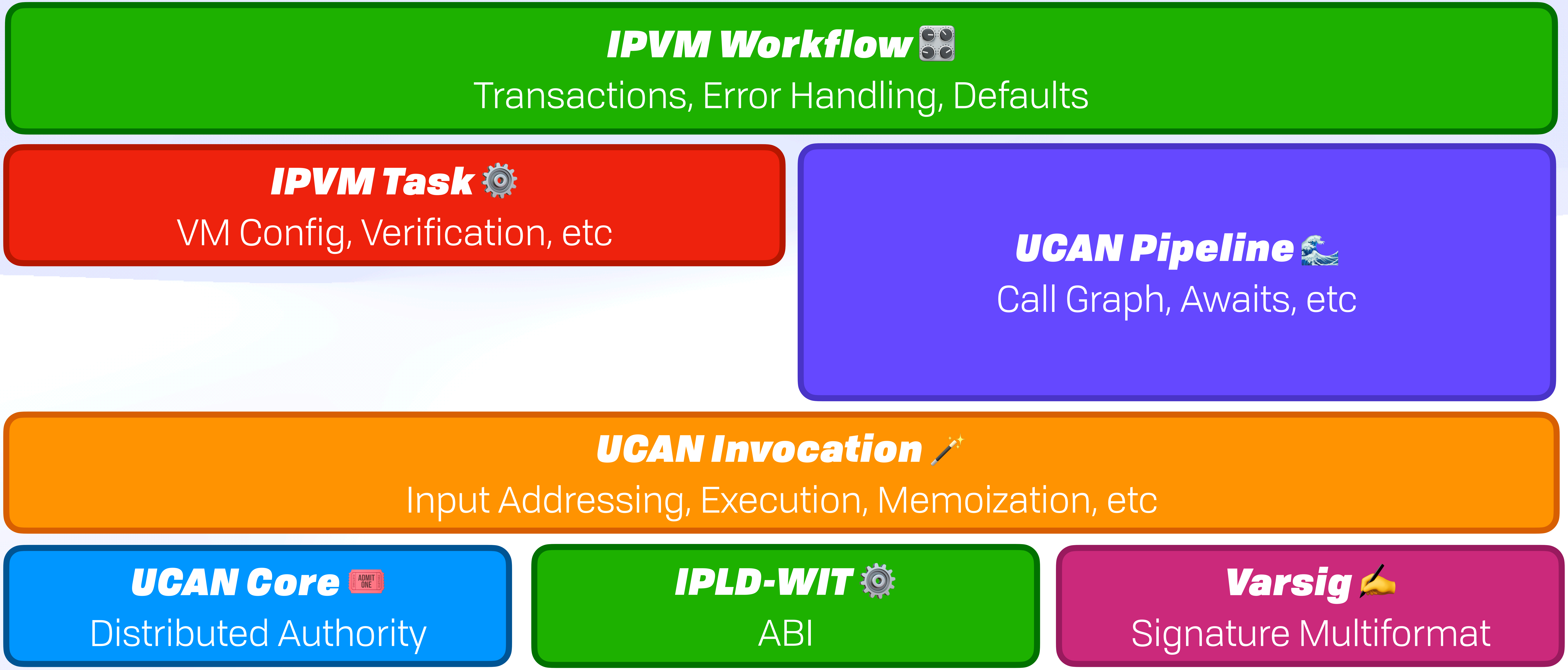
UCAN Core 🎟️
Distributed Authority

IPLD-WIT ⚙️
ABI

Varsig ✍️
Signature Multiformat

What is an IPVM 🤔

Reusable Spec Stack



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

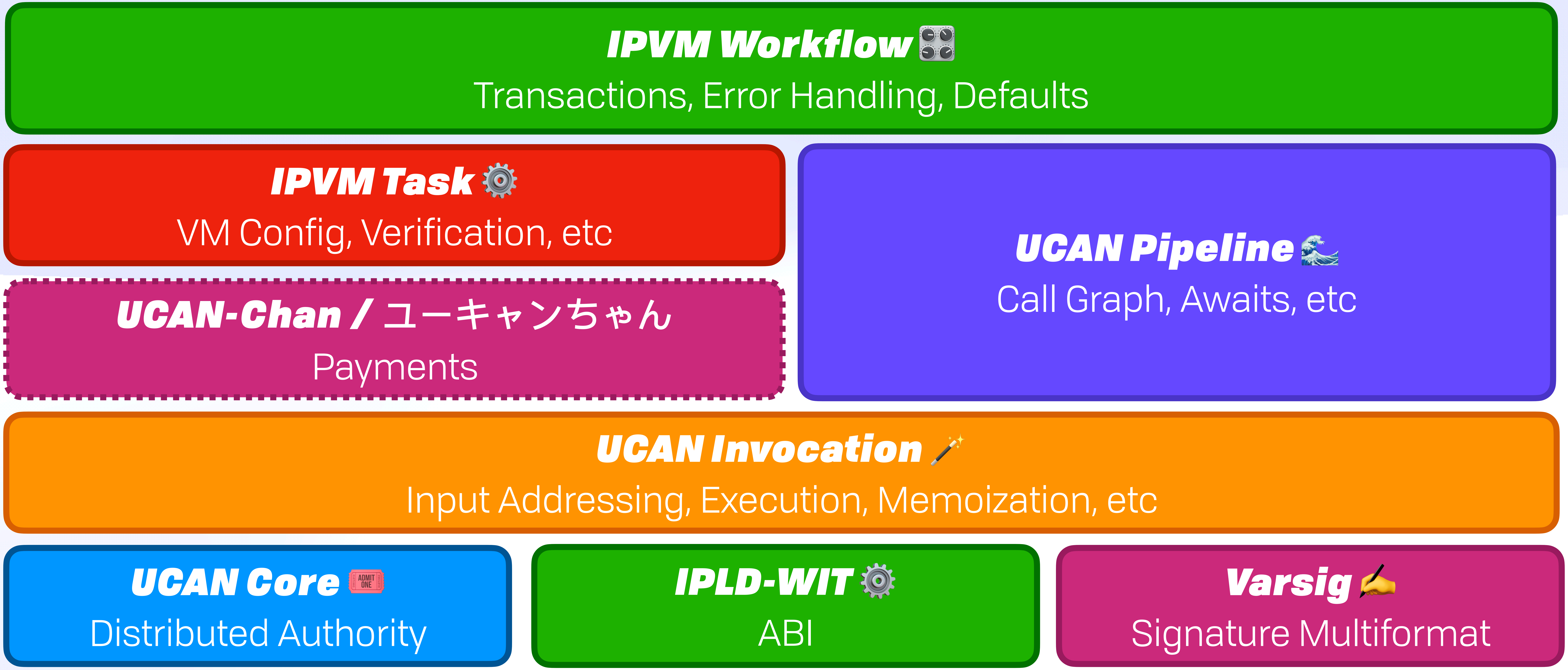
UCAN Core 🎟️
Distributed Authority

IPLD-WIT ⚙️
ABI

Varsig ✍️
Signature Multiformat

What is an IPVM 🤔

Reusable Spec Stack



IPVM Workflow 🎛️

Transactions, Error Handling, Defaults

IPVM Task ⚙️

VM Config, Verification, etc

UCAN-Chan / ユーキャンちゃん

Payments

UCAN Pipeline 🌊

Call Graph, Awaits, etc

UCAN Invocation 🪄

Input Addressing, Execution, Memoization, etc

UCAN Core 🎟️

Distributed Authority

IPLD-WIT ⚙️

ABI

Varsig ✍️

Signature Multiformat

What is an IPVM 🤔

The Friends You Made Along the Way

What is an IPVM 🤔

The Friends You Made Along the Way

What is an IPVM 🤔

The Friends You Made Along the Way

Invocation-as-IPLD

UCAN Invocation Spec

Invocation-as-IPLD

Reference vs Dispatch 🔑🚗

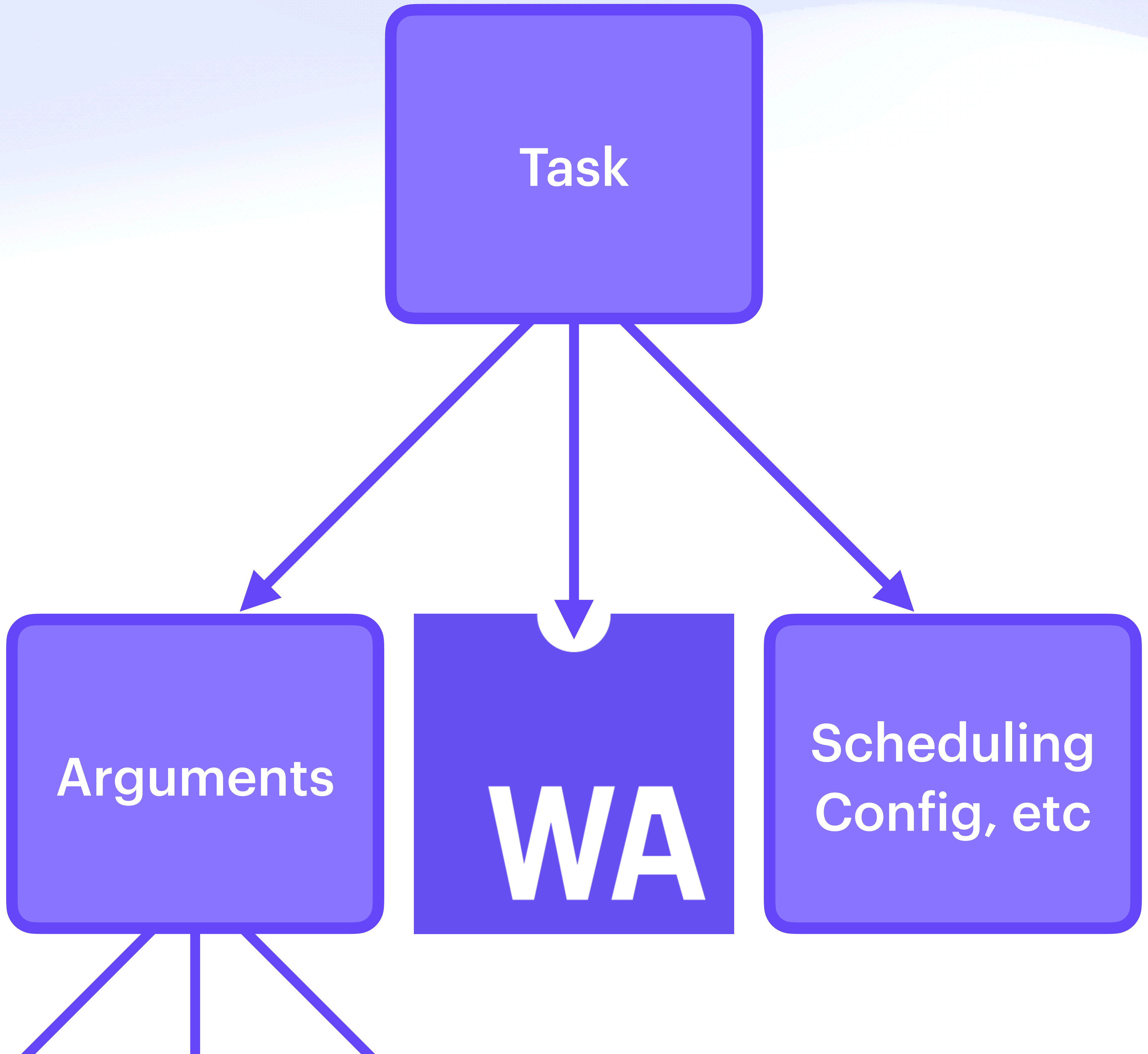
Invocation-as-IPLD

Reference vs Dispatch 🔑🚗

Arguments
WA

Invocation-as-IPLD

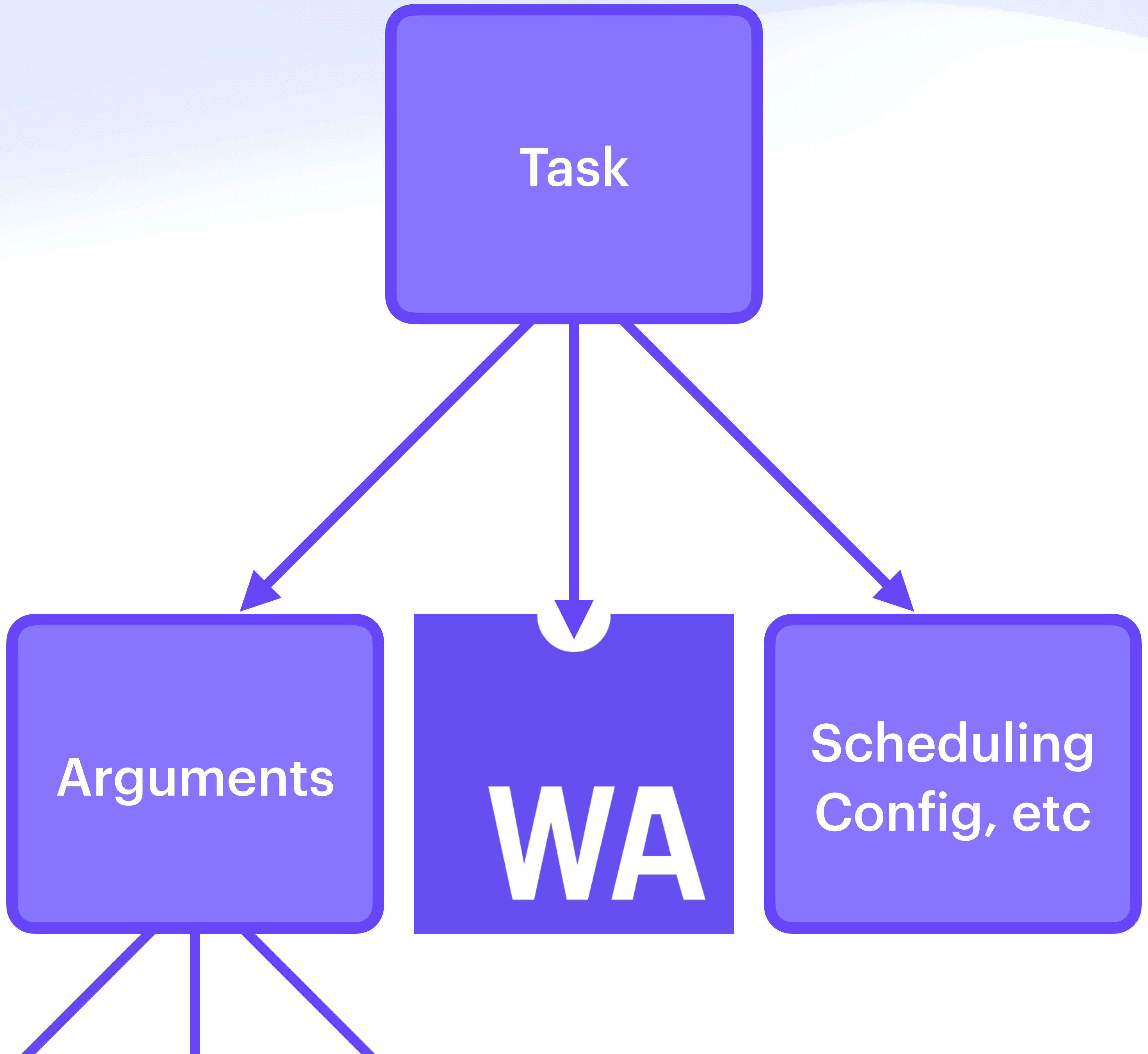
Reference vs Dispatch 🔑🚗



Invocation-as-IPLD

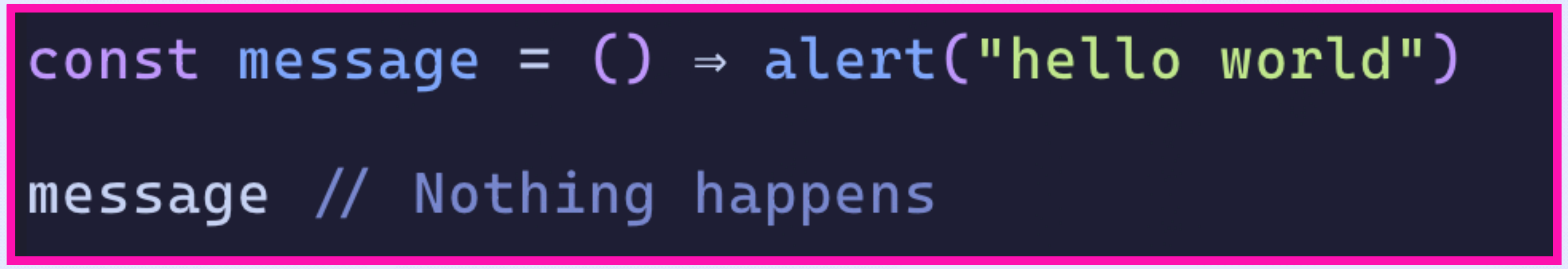
Reference vs Dispatch 🔑🚗

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

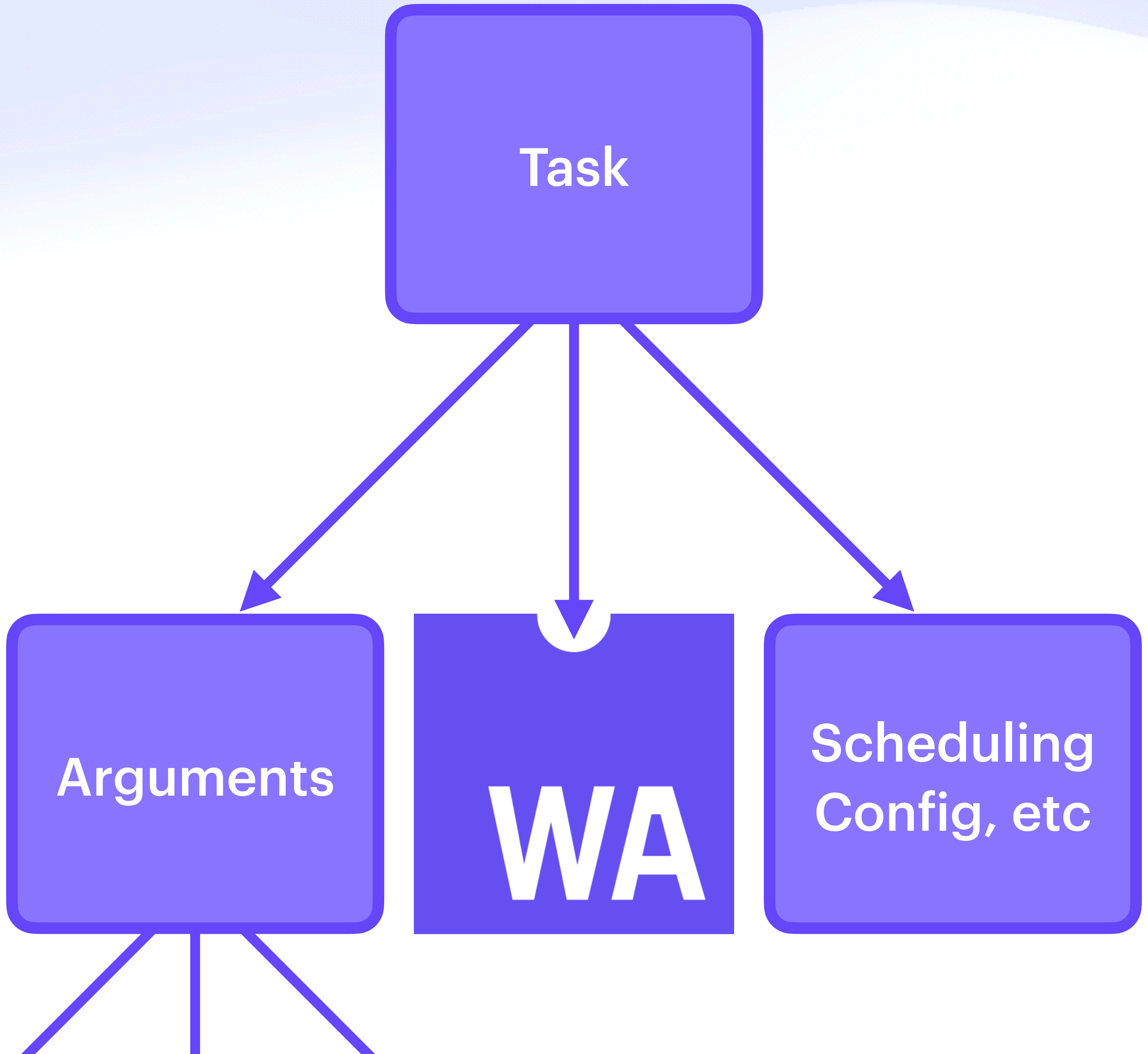


Invocation-as-IPLD

Reference vs Dispatch 🔑🚗

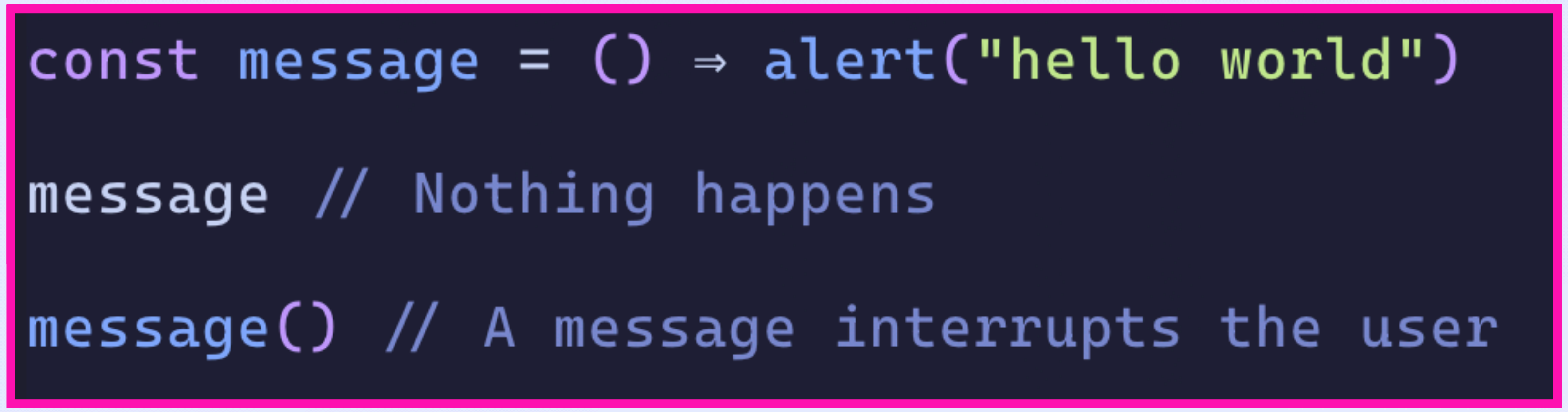


```
const message = () ⇒ alert("hello world")

message // Nothing happens
```

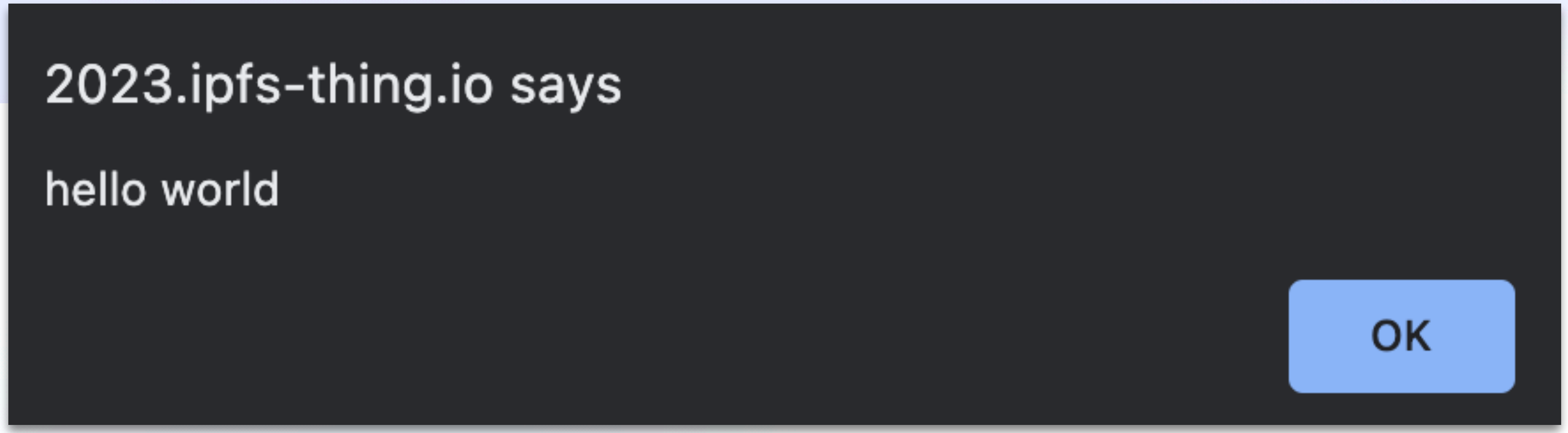


Invocation-as-IPLD

Reference vs Dispatch 🔑🚗

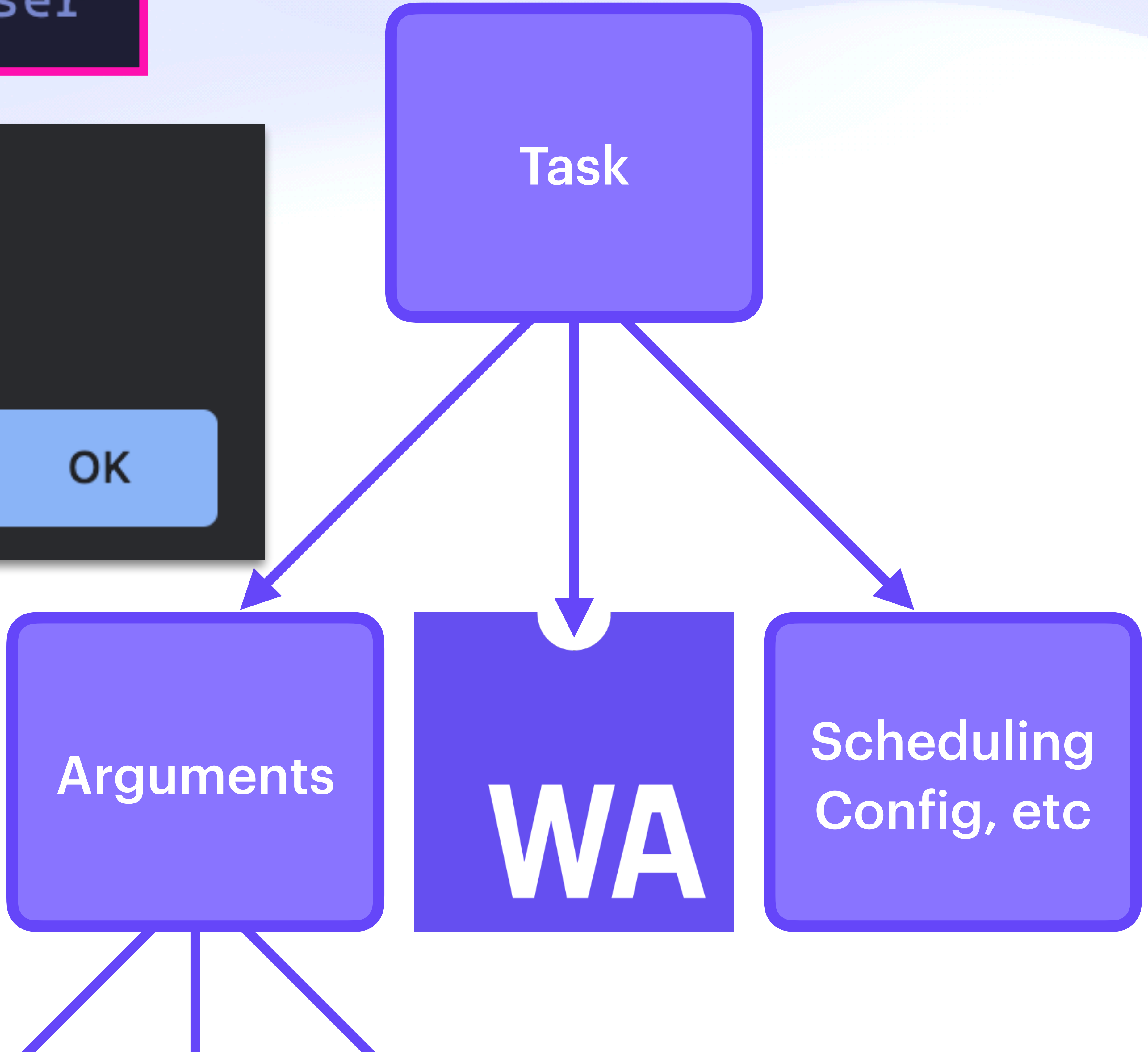


```
const message = () ⇒ alert("hello world")

message // Nothing happens

message() // A message interrupts the user
```



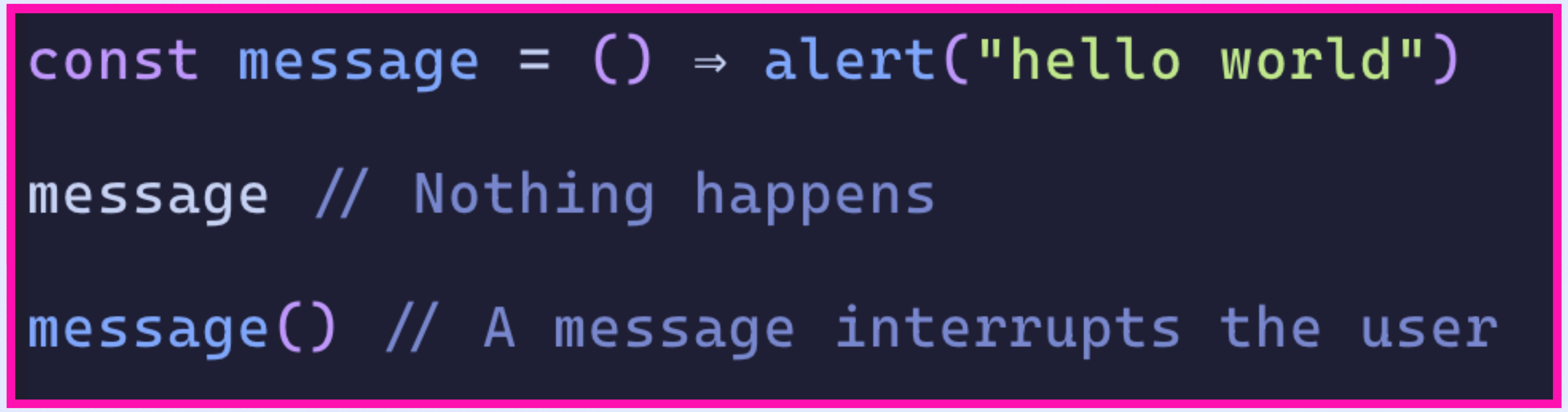
2023.ipfs-thing.io says

hello world

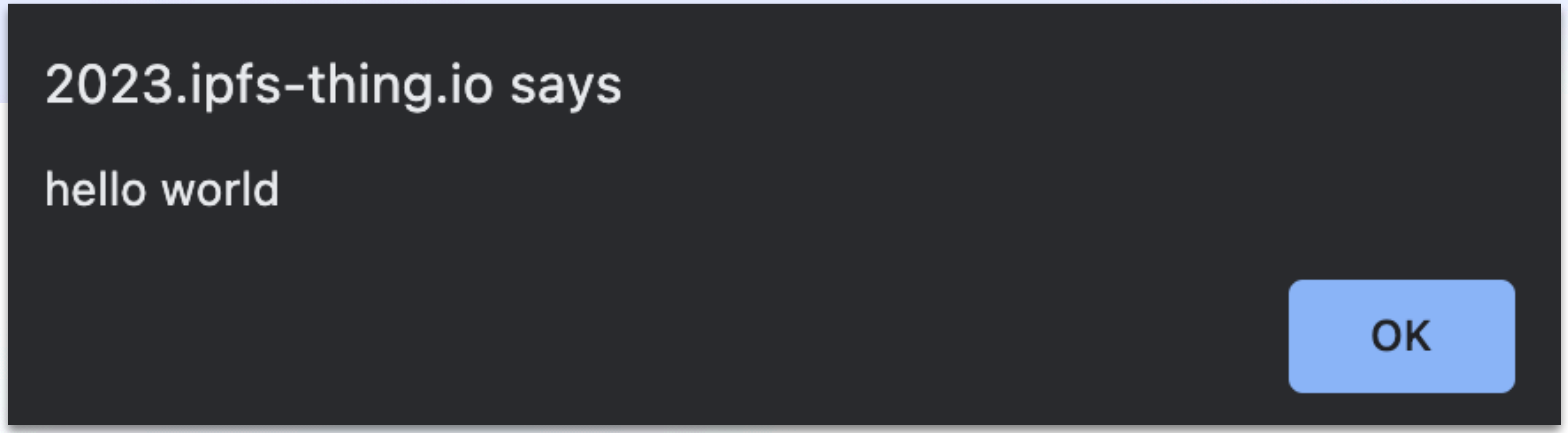


Invocation-as-IPLD

Reference vs Dispatch 🔑🚗

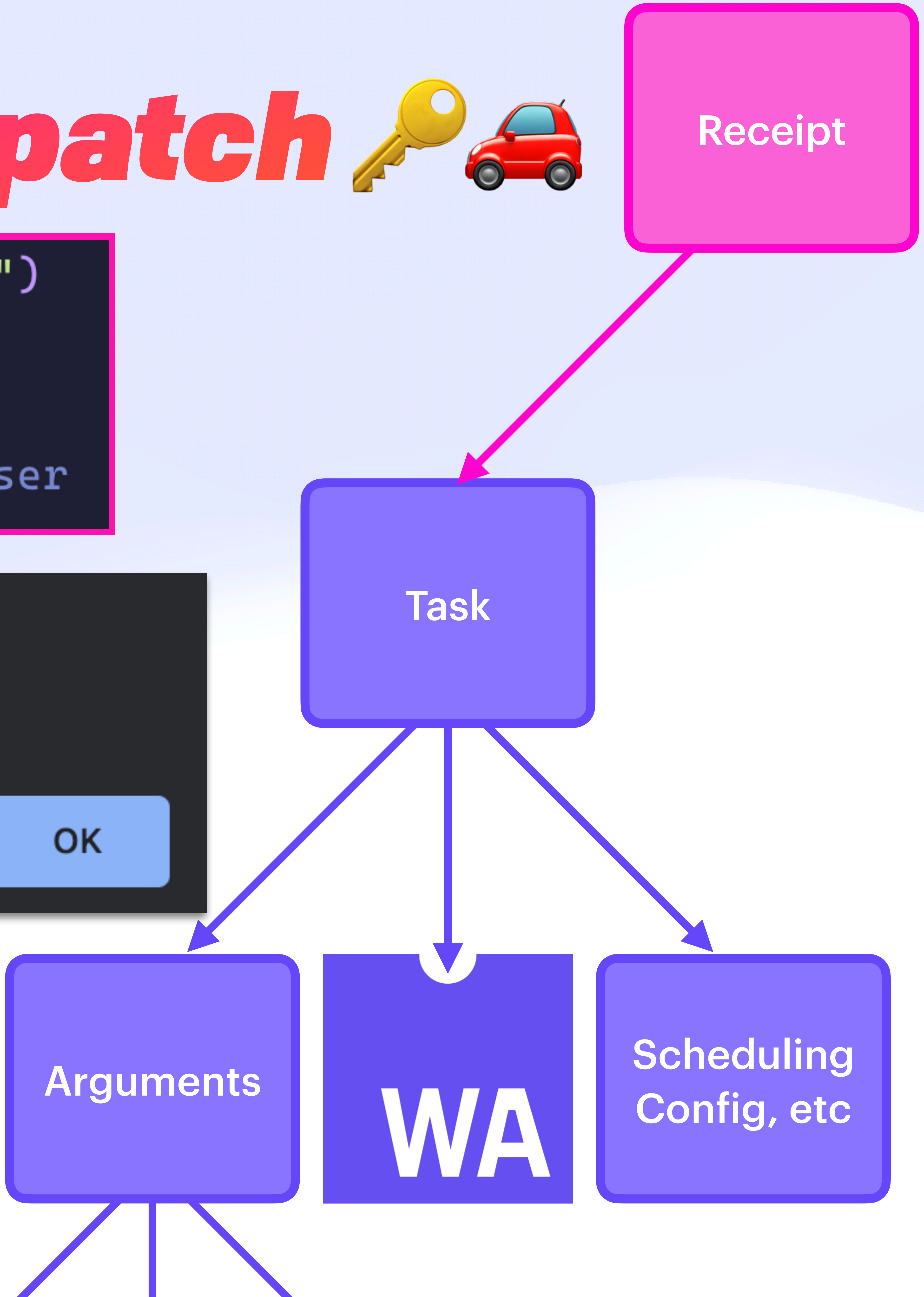


```
const message = () ⇒ alert("hello world")

message // Nothing happens

message() // A message interrupts the user
```



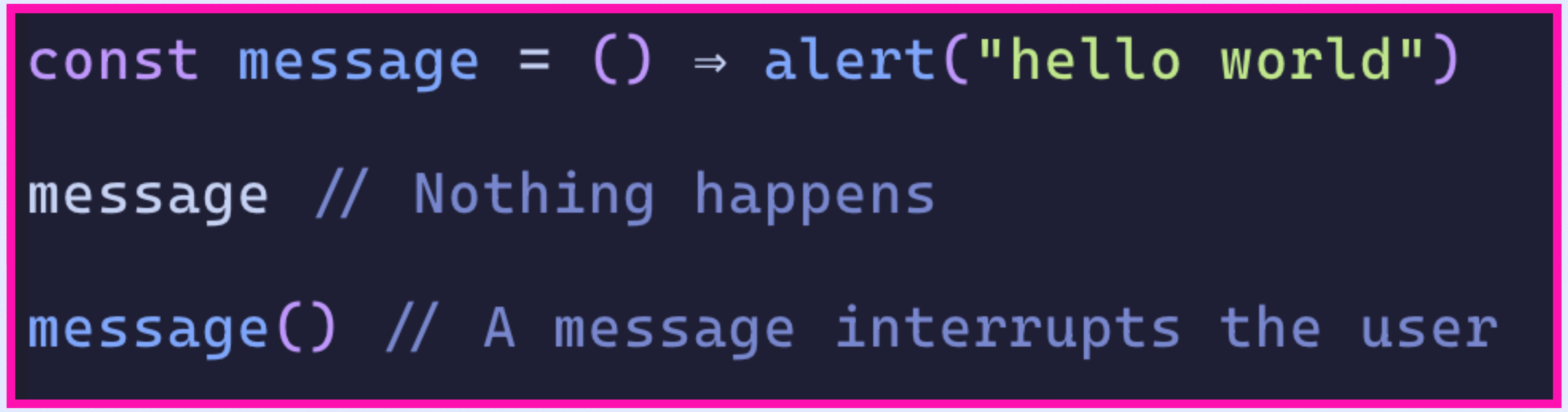
2023.ipfs-thing.io says

hello world

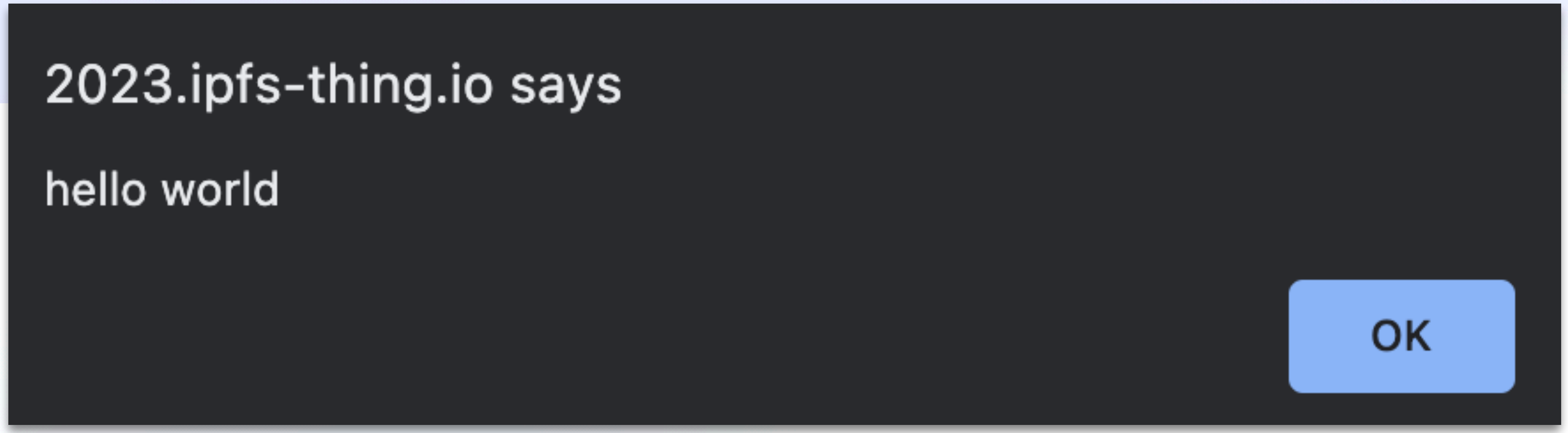


Invocation-as-IPLD

Reference vs Dispatch 🔑🚗

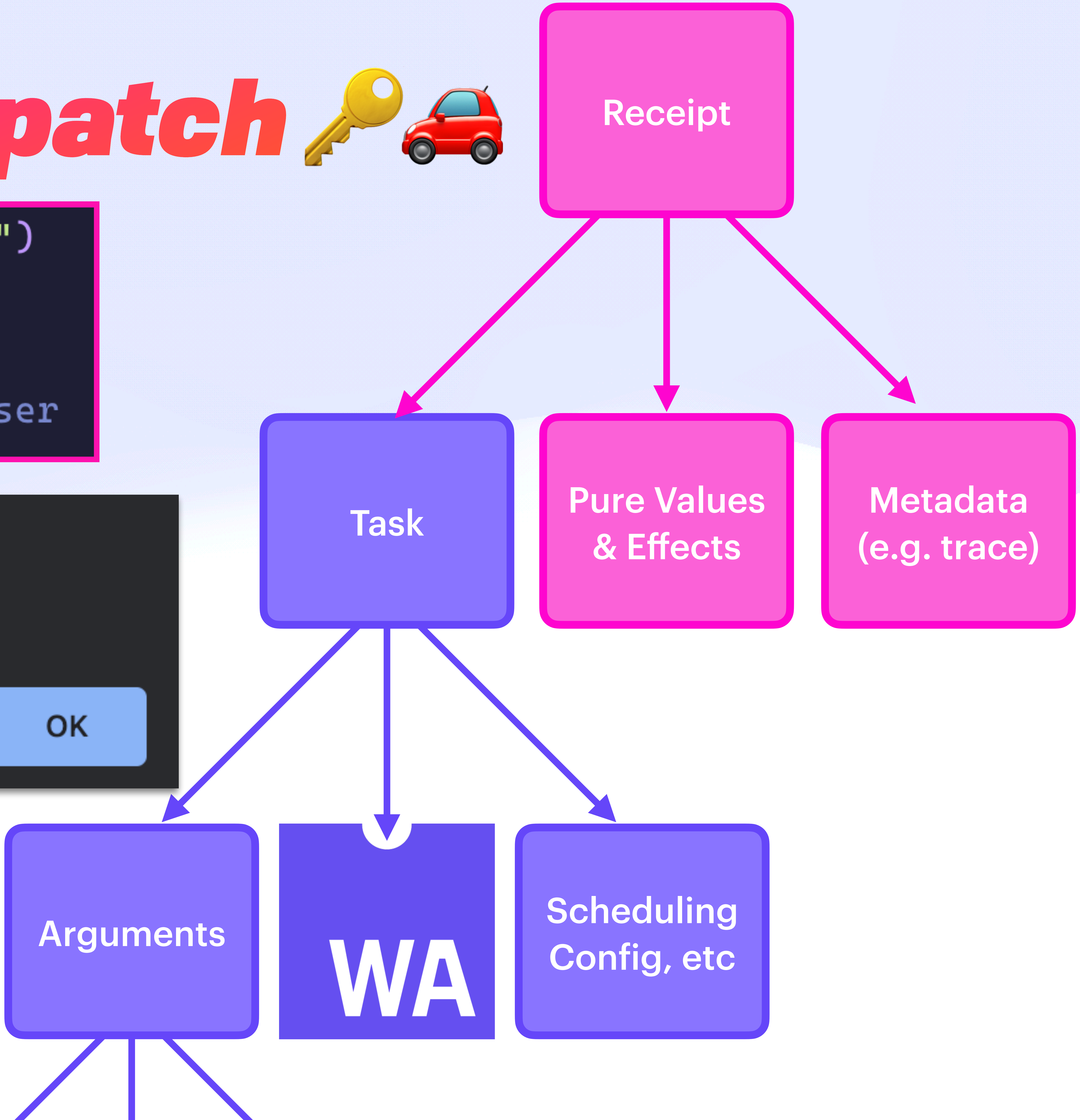


```
const message = () => alert("hello world")

message // Nothing happens

message() // A message interrupts the user
```



2023.ipfs-thing.io says

hello world

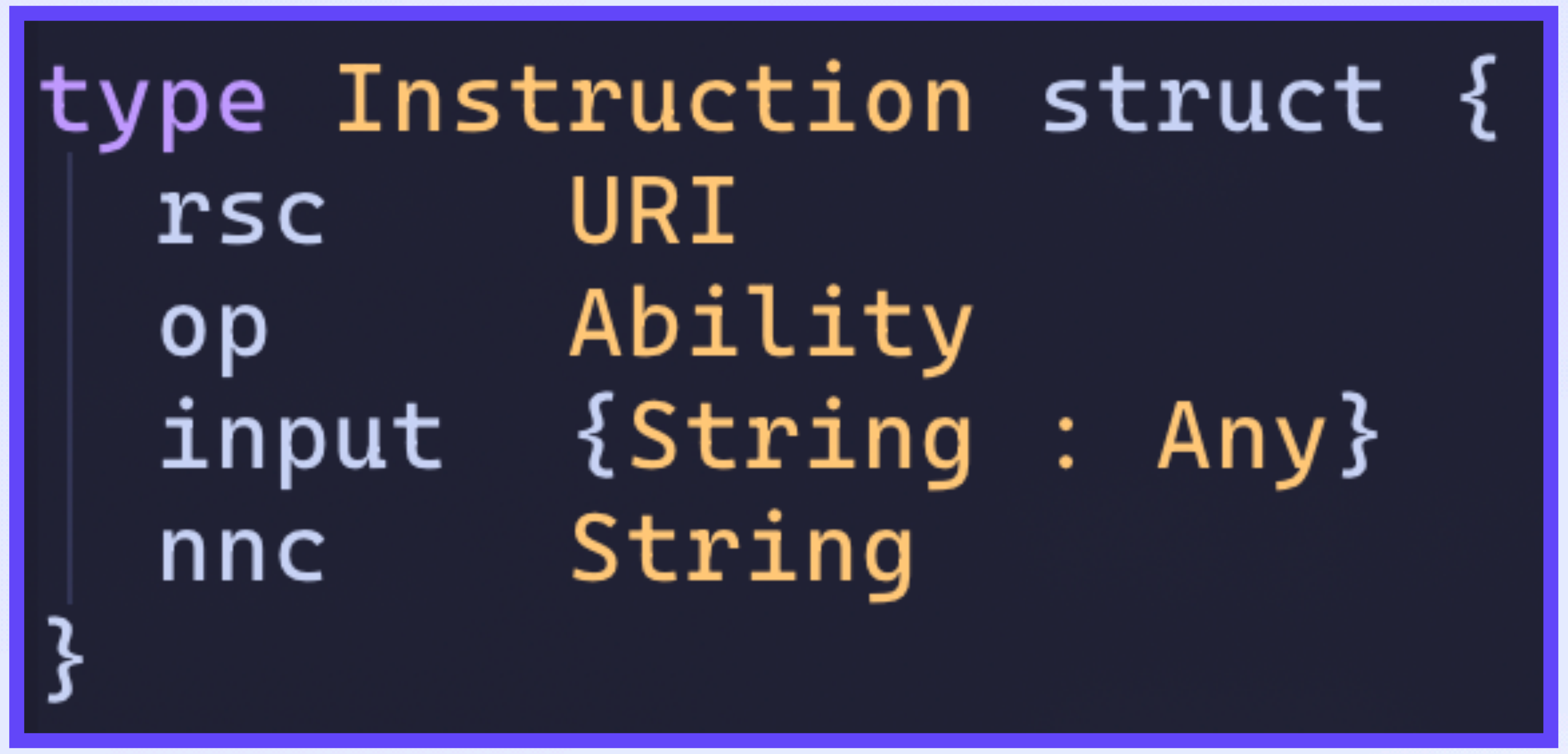


Invocation-as-IPLD

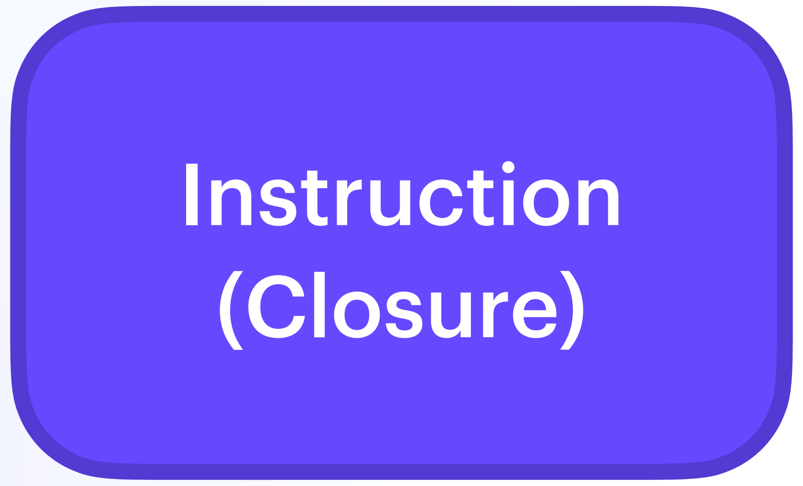
IPLD Schema

Invocation-as-IPLD

IPLD Schema



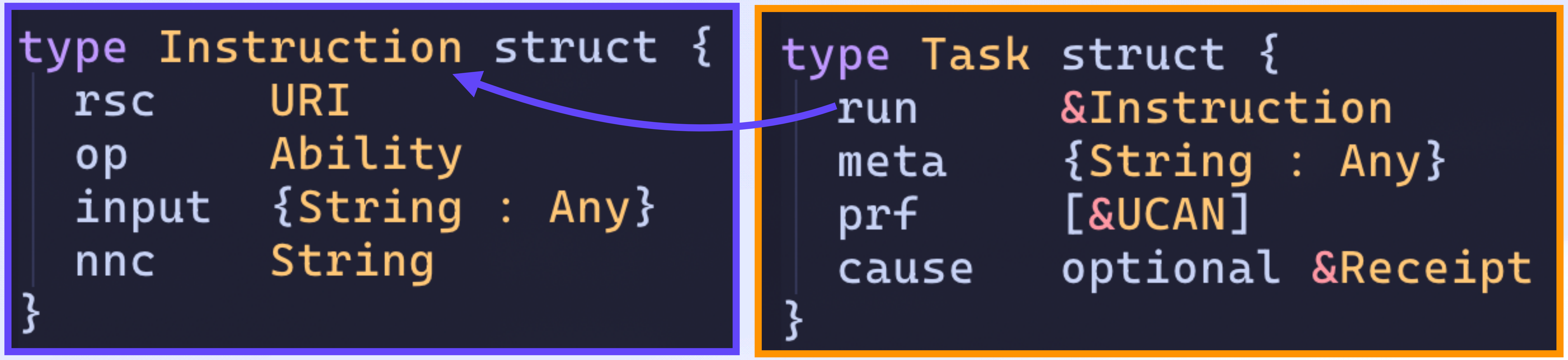
```
type Instruction struct {
  rsc     URI
  op      Ability
  input   {String : Any}
  nnc     String
}
```



Instruction
(Closure)

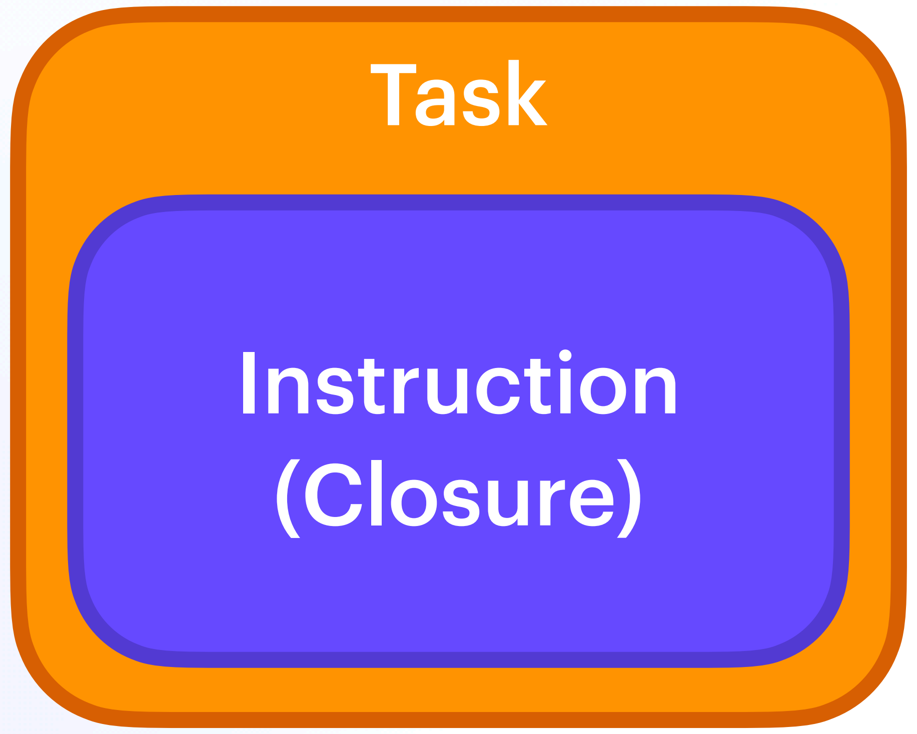
Invocation-as-IPLD

IPLD Schema



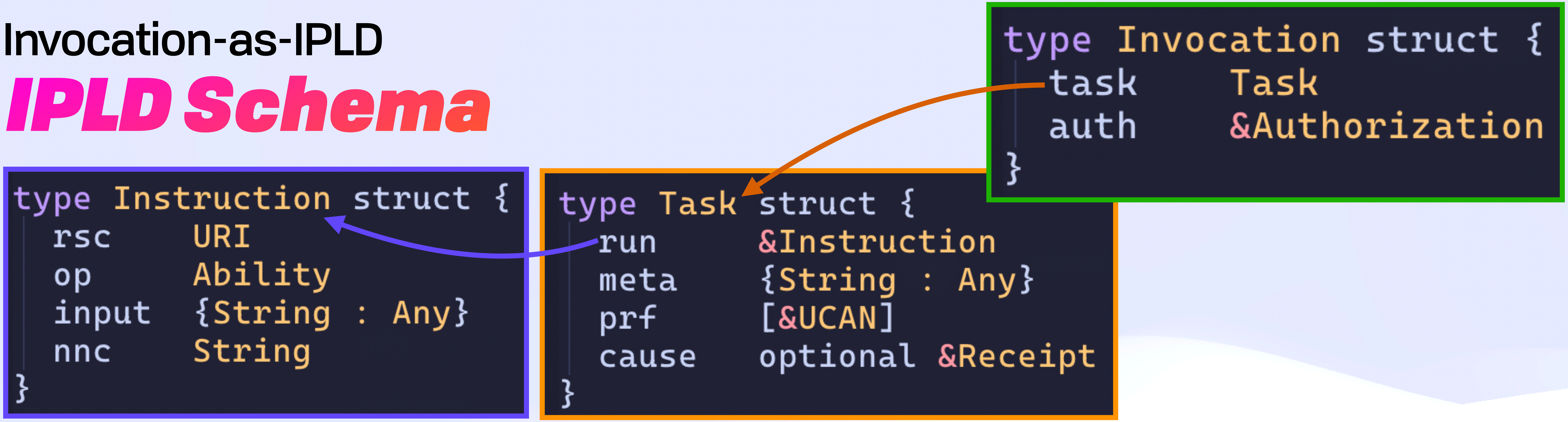
```
type Instruction struct {
  rsc     URI
  op      Ability
  input   {String : Any}
  nnc     String
}
```

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



Invocation-as-IPLD

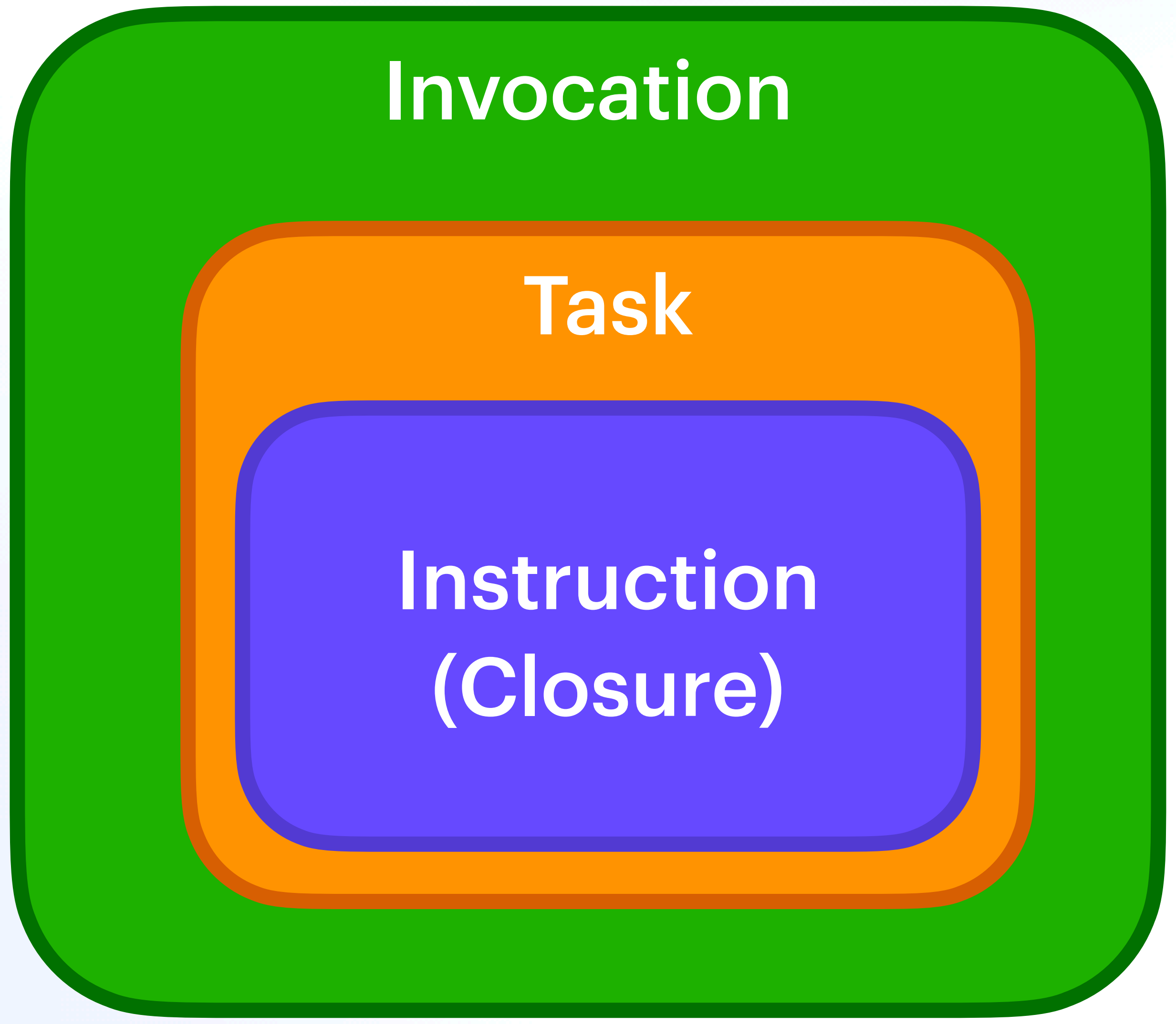
IPLD Schema



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

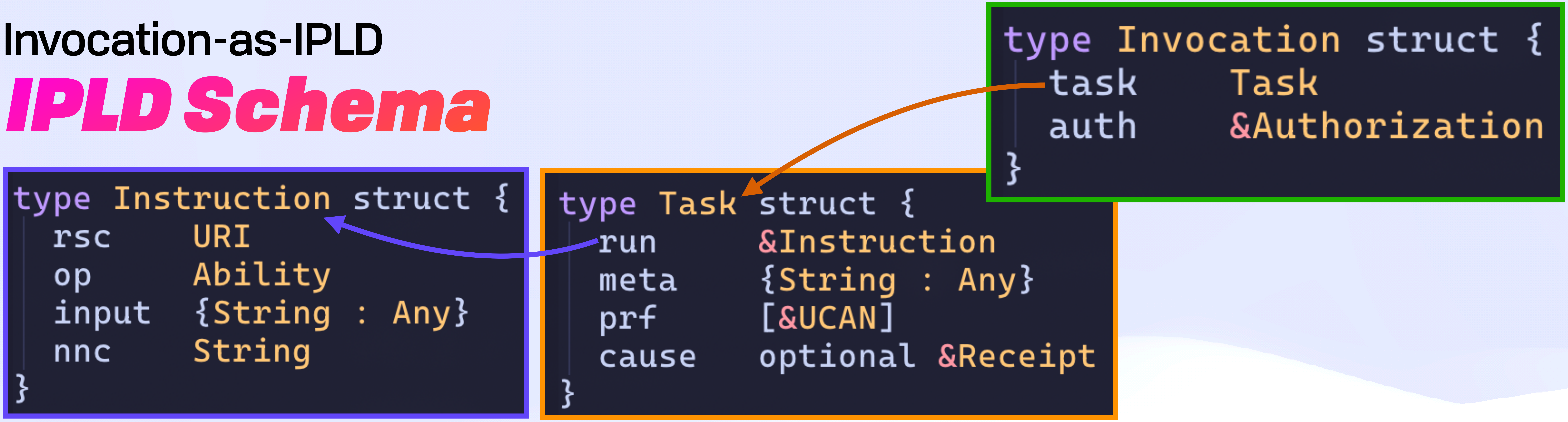
```
type Task struct {
  run    &Instruction
  meta   {String : Any}
  prf    [&UCAN]
  cause  optional &Receipt
}
```

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



Invocation-as-IPLD

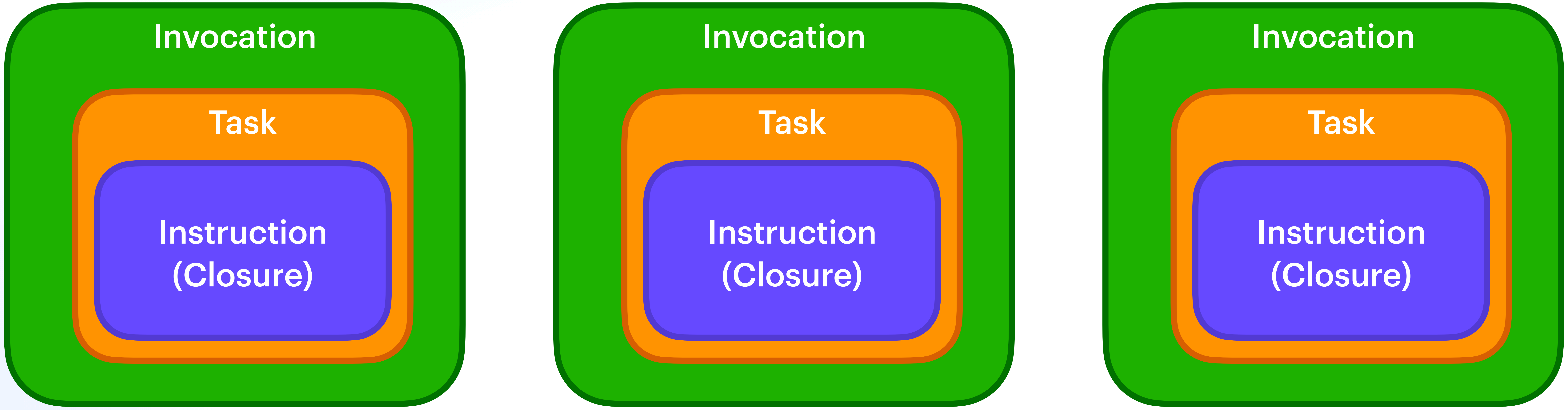
IPLD Schema



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

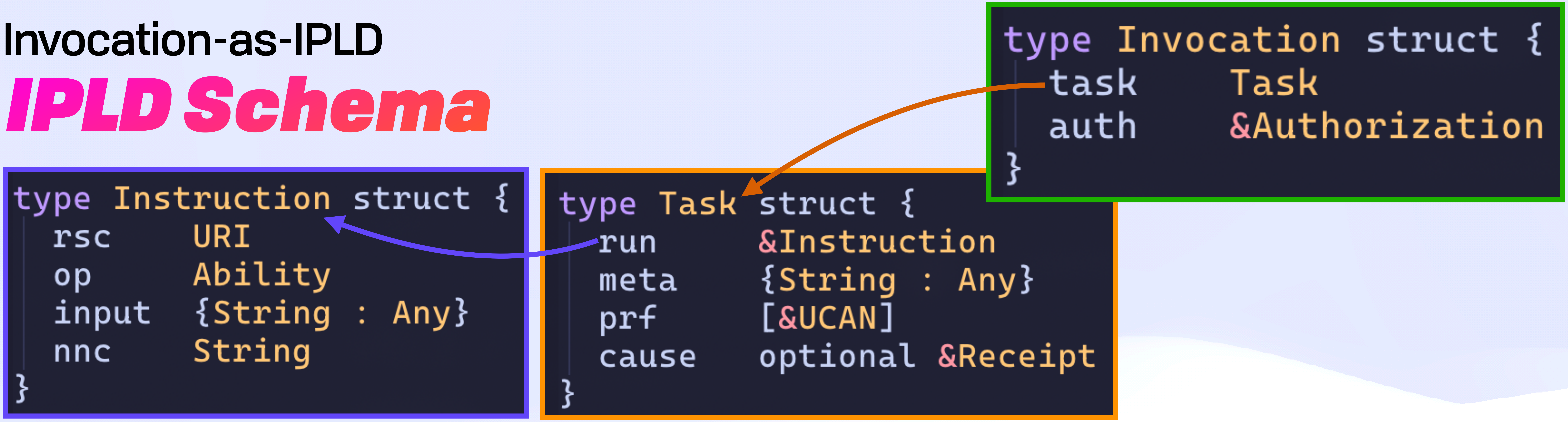
```
type Task struct {
  run    &Instruction
  meta   {String : Any}
  prf    [&UCAN]
  cause  optional &Receipt
}
```

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



Invocation-as-IPLD

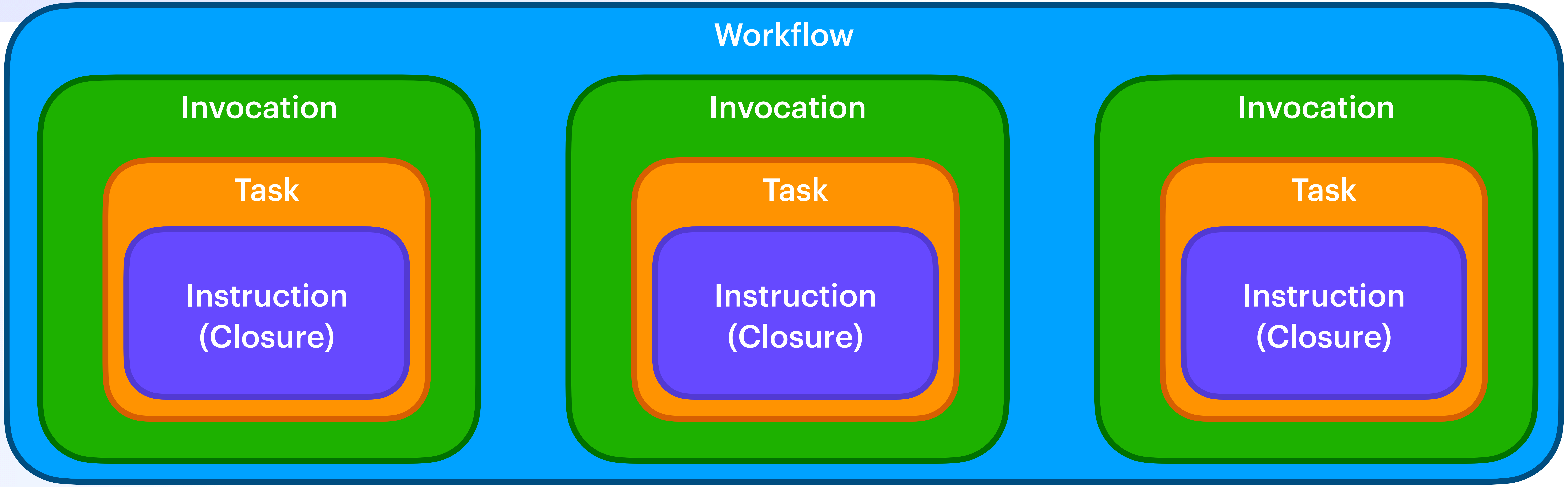
IPLD Schema



```
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
}
```

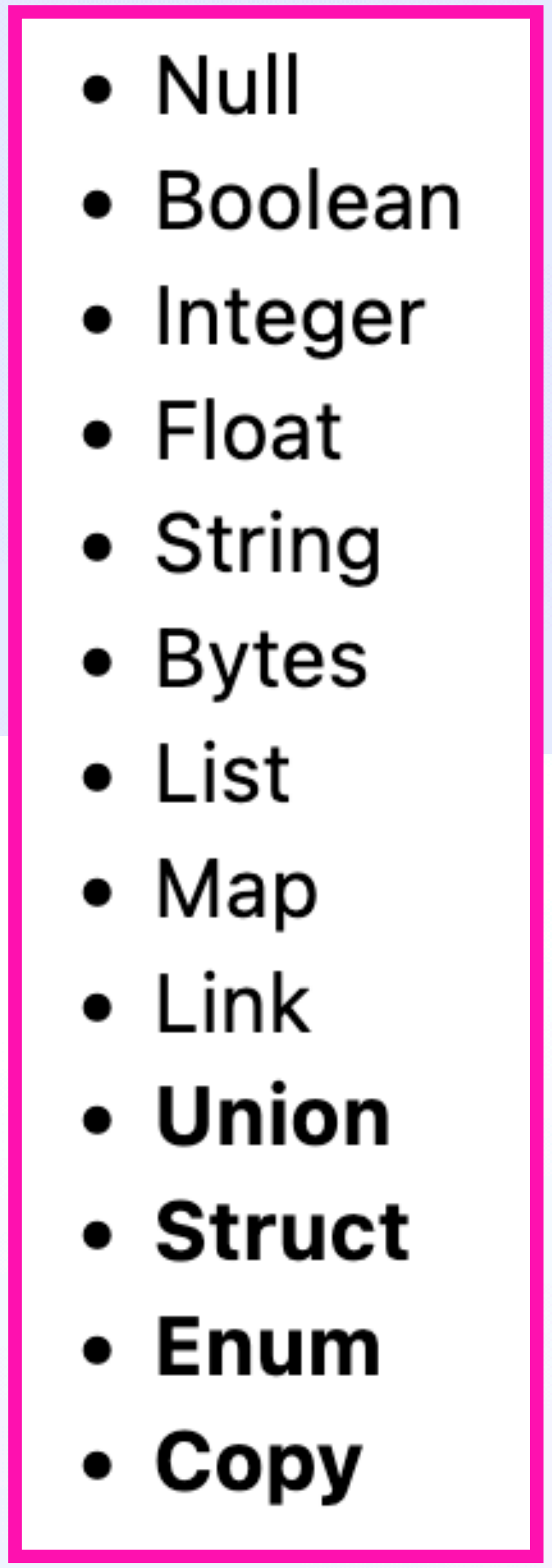


Invocation-as-IPLD

Matching Impedance

Invocation-as-IPLD

Matching Impedance

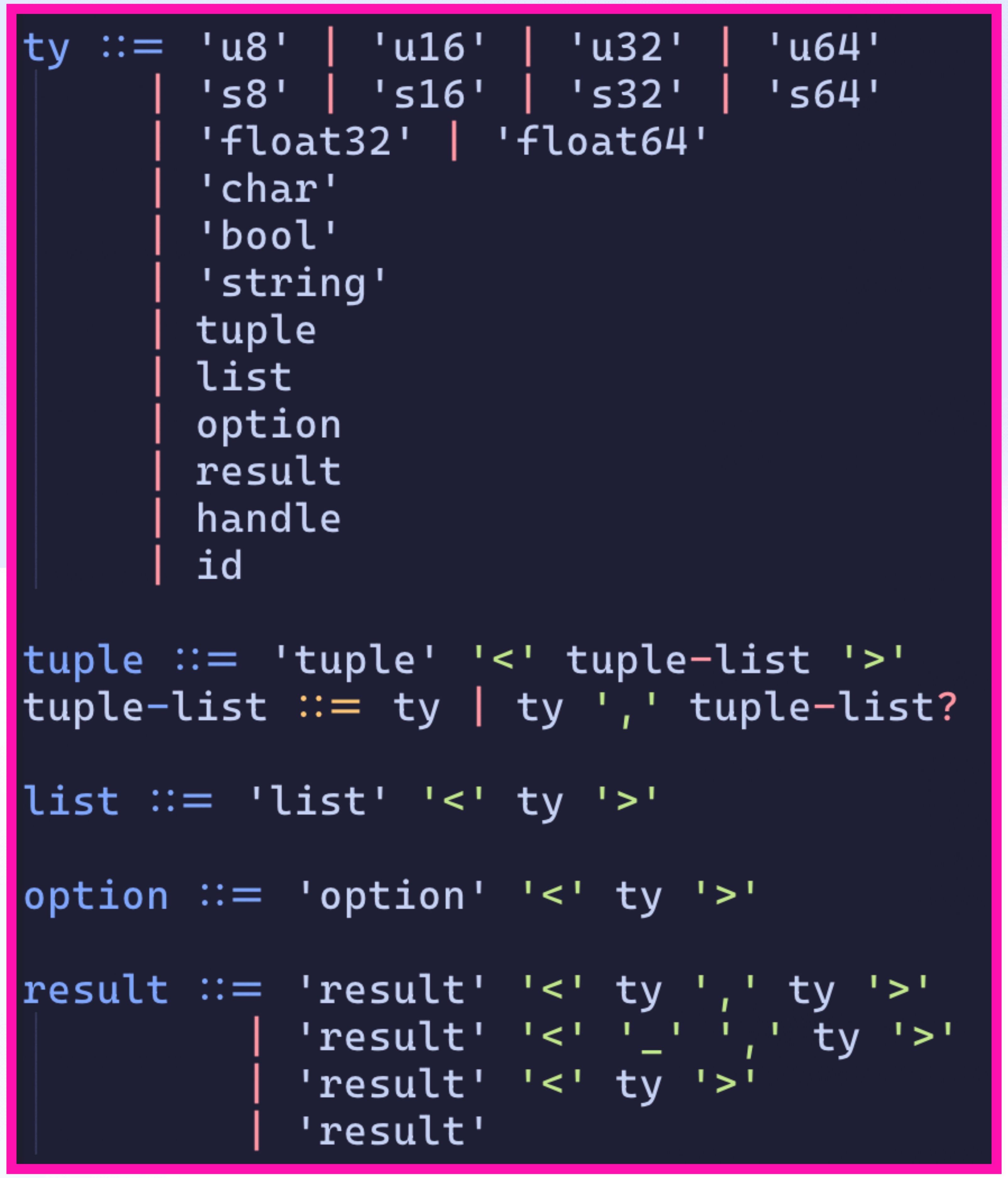


- Null
- Boolean
- Integer
- Float
- String
- Bytes
- List
- Map
- Link
- **Union**
- **Struct**
- **Enum**
- **Copy**

Invocation-as-IPLD

Matching Impedance

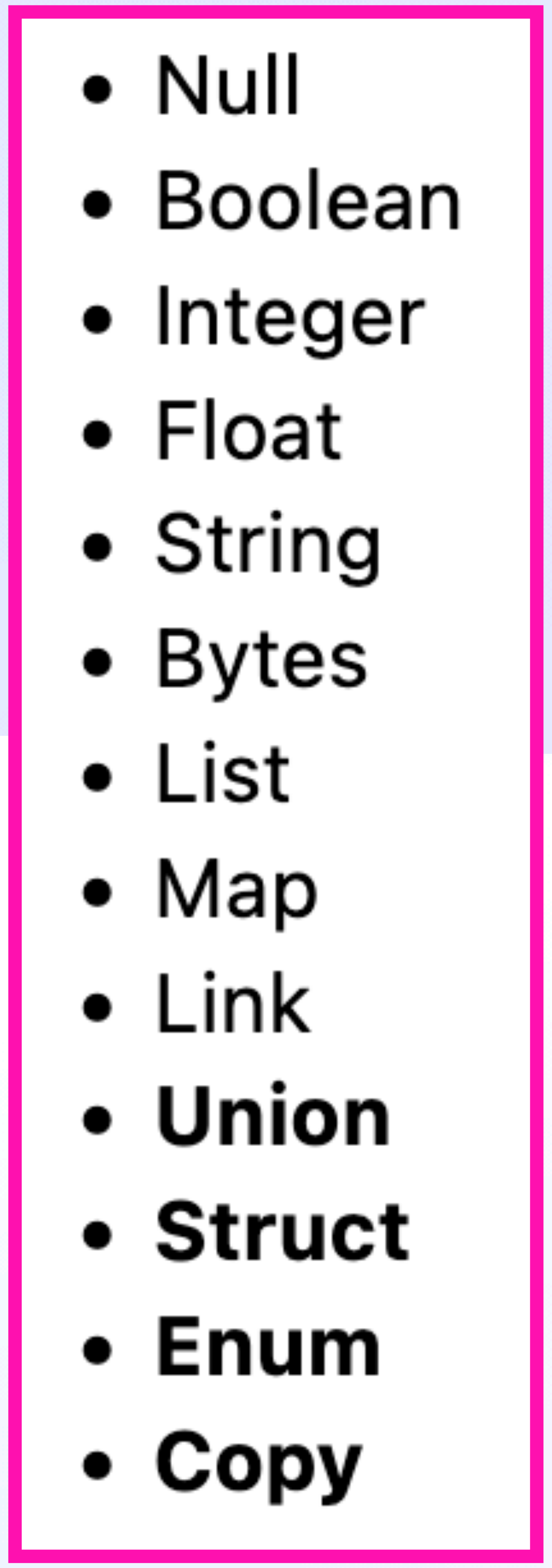
- Null
- Boolean
- Integer
- Float
- String
- Bytes
- List
- Map
- Link
- **Union**
- **Struct**
- **Enum**
- **Copy**



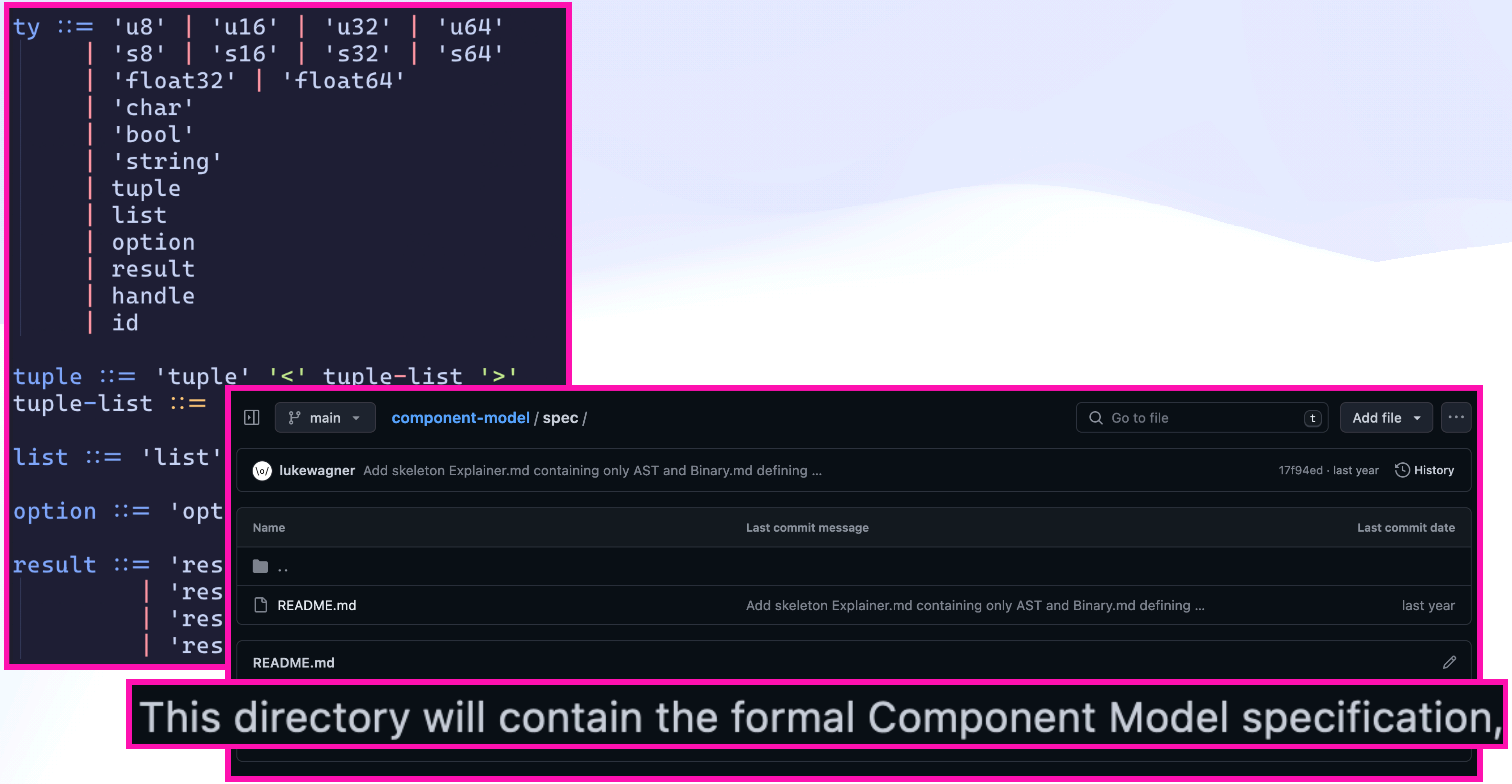
```
ty ::= 'u8' | 'u16' | 'u32' | 'u64'
     | 's8' | 's16' | 's32' | 's64'
     | 'float32' | 'float64'
     | 'char'
     | 'bool'
     | 'string'
     | tuple
     | list
     | option
     | result
     | handle
     | id

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'
```

Invocation-as-IPLD

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
```

lukewagner Add skeleton Explainer.md containing only AST and Binary.md defining ... 17f94ed · last year

README.md Add skeleton Explainer.md containing only AST and Binary.md defining ... last year

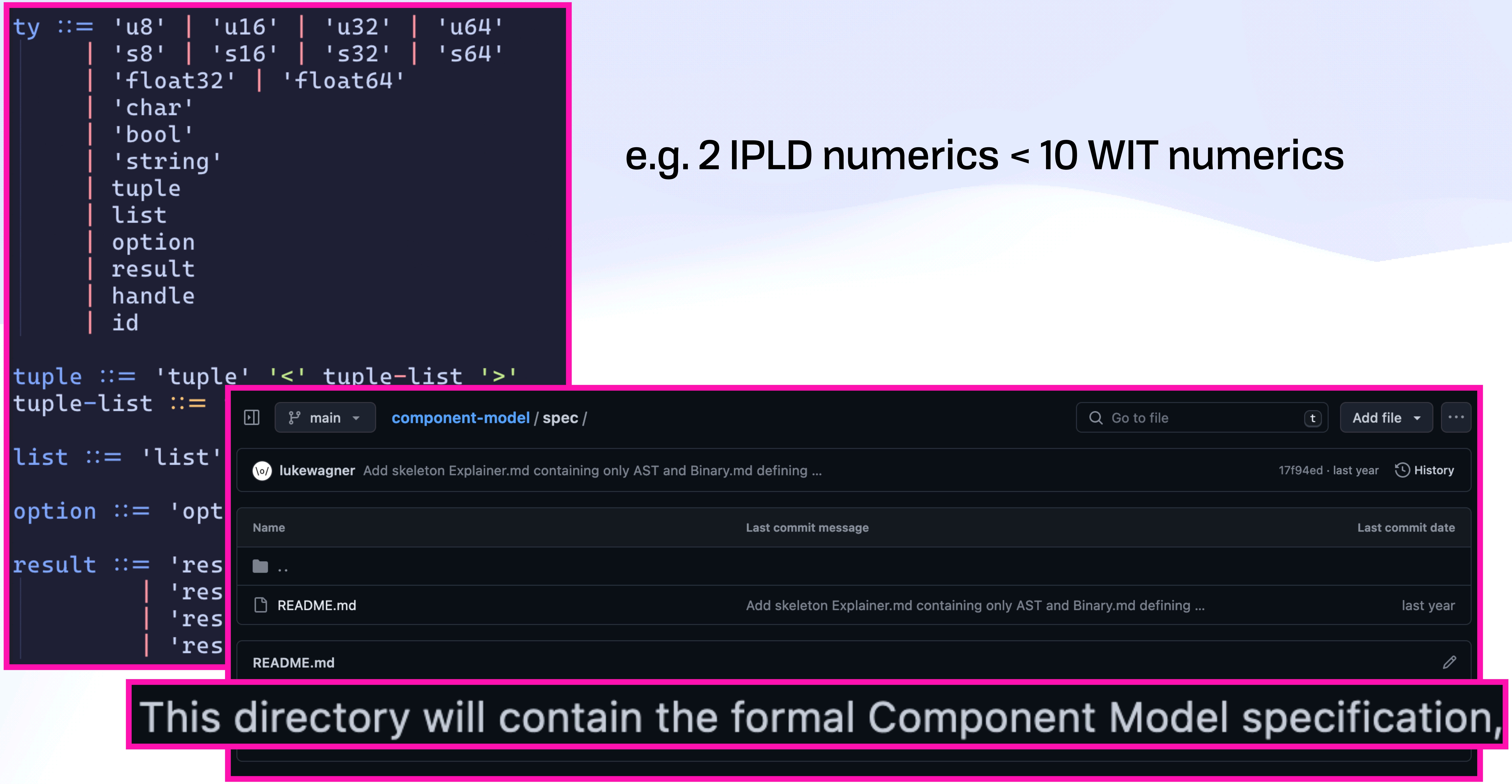
README.md

This directory will contain the formal Component Model specification,

Invocation-as-IPLD

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
```

lukewagner Add skeleton Explainer.md containing only AST and Binary.md defining ...

17f94ed · last year

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

README.md

This directory will contain the formal Component Model specification,

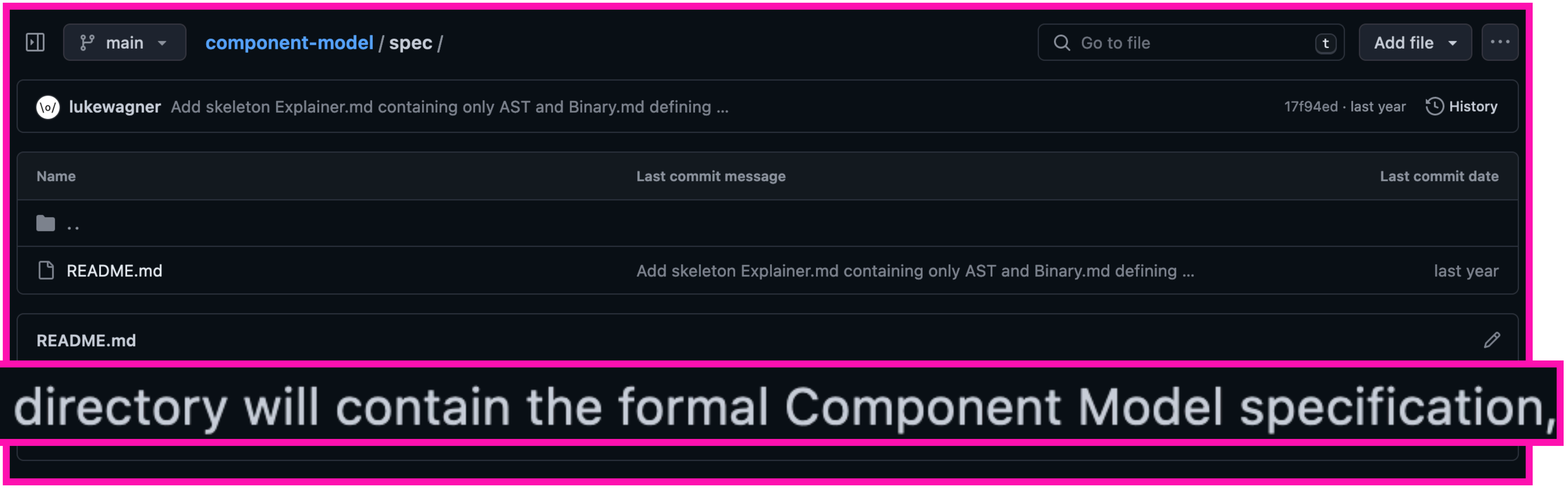
e.g. 2 IPLD numerics < 10 WIT numerics

Invocation-as-IPLD

Matching Impedance

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

e.g. 2 IPLD numerics < 10 WIT numerics



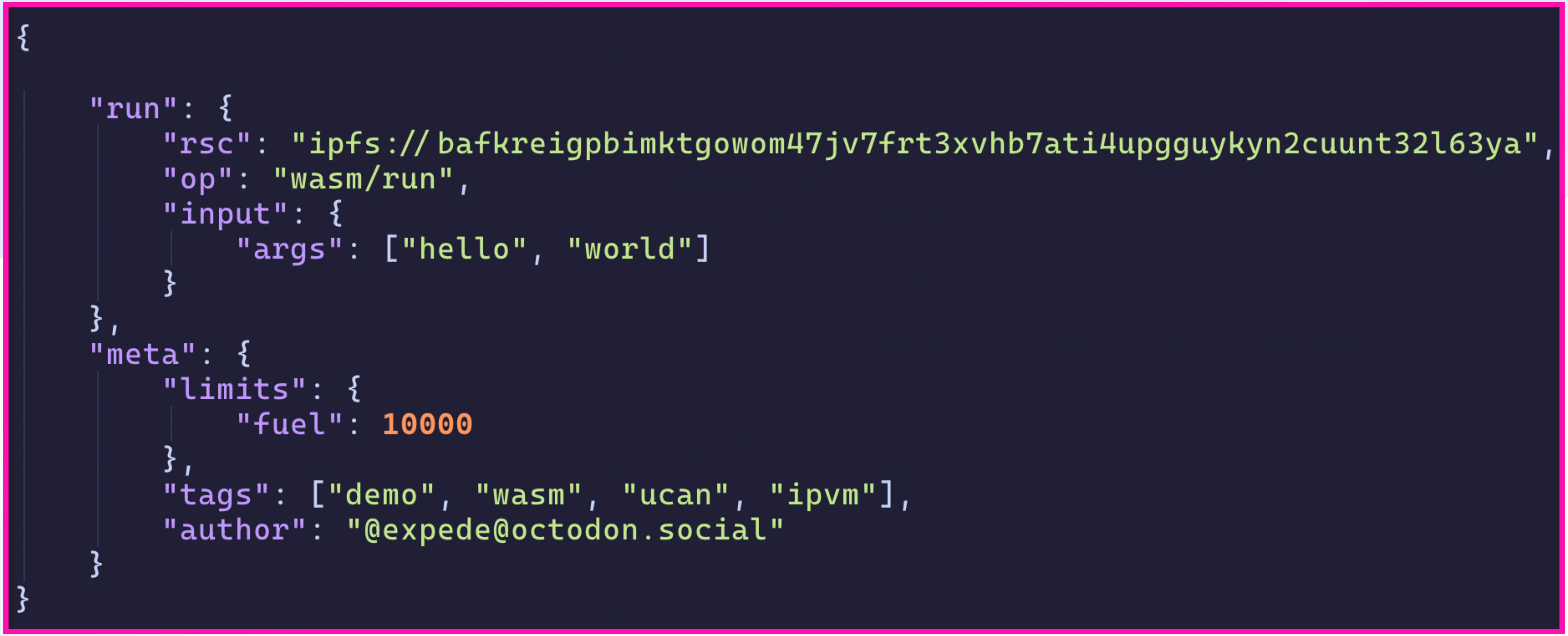
lukewagner Add skeleton Explainer.md containing only AST and Binary.md defining ... 17f94ed · last year

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

This directory will contain the formal Component Model specification,

Invocation-as-IPLD

Matching Impedance



```
{

    "run": {
        "rsc": "ipfs://bafkreigpbimktgowom47jv7frt3xvhb7ati4upgguykyn2cuunt32l63ya",
        "op": "wasm/run",
        "input": {
            "args": ["hello", "world"]
        }
    },
    "meta": {
        "limits": {
            "fuel": 10000
        },
        "tags": ["demo", "wasm", "ucan", "ipvm"],
        "author": "@expede@octodon.social"
    }
}
```

Dataflow & Pipelining

🚰

Dataflow & Pipelining 🚰

Dataflow & Pipelining 🚰

Their recommendation, which I feel was prescient, was that [dataflow] seemed to them ***more like a law of nature***, which is not patentable.

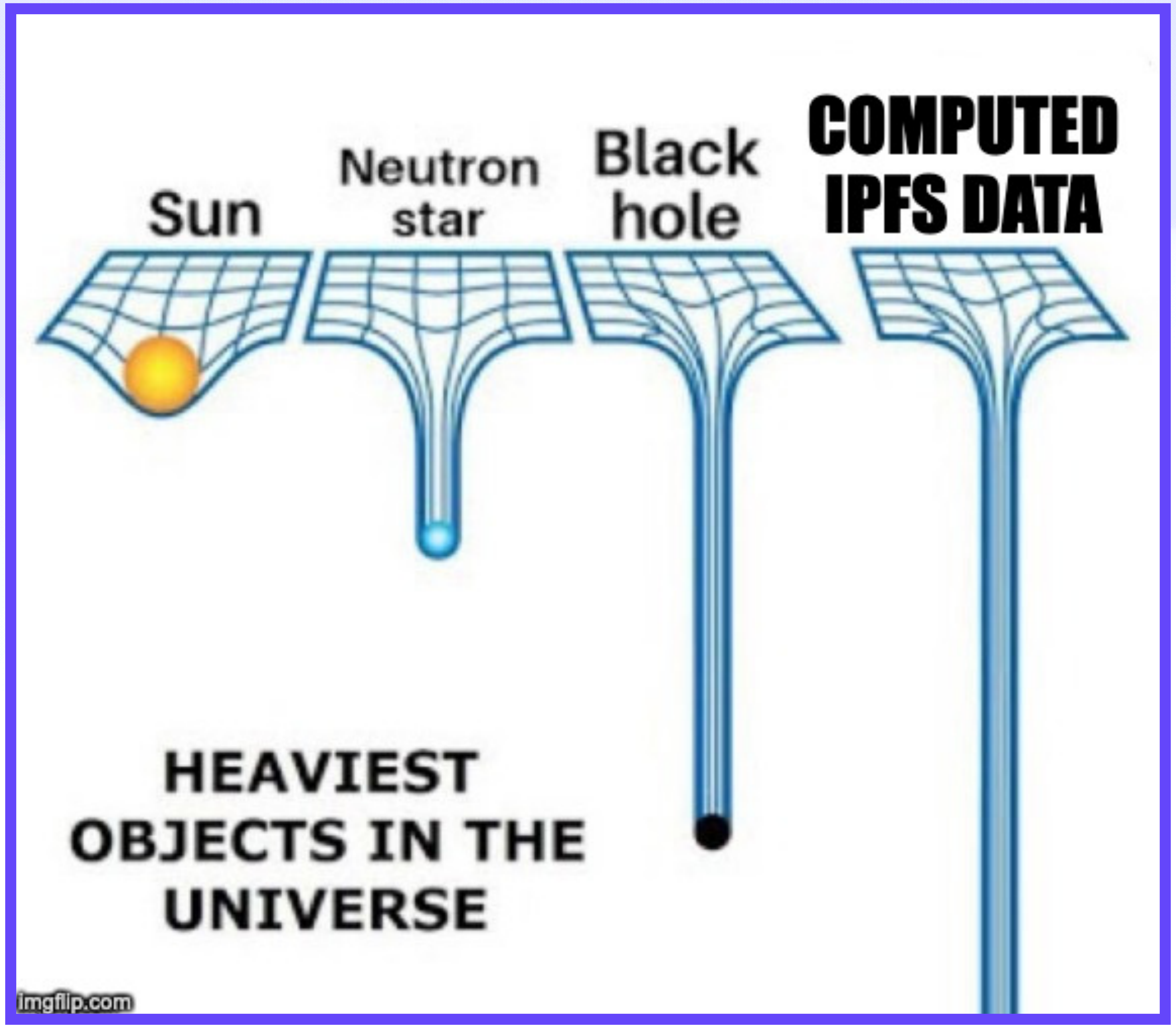
J. Paul Morrison, Flow-Based Programming

Dataflow & Pipelining 🚰

Solving for Data Gravity

Dataflow & Pipelining 🚰

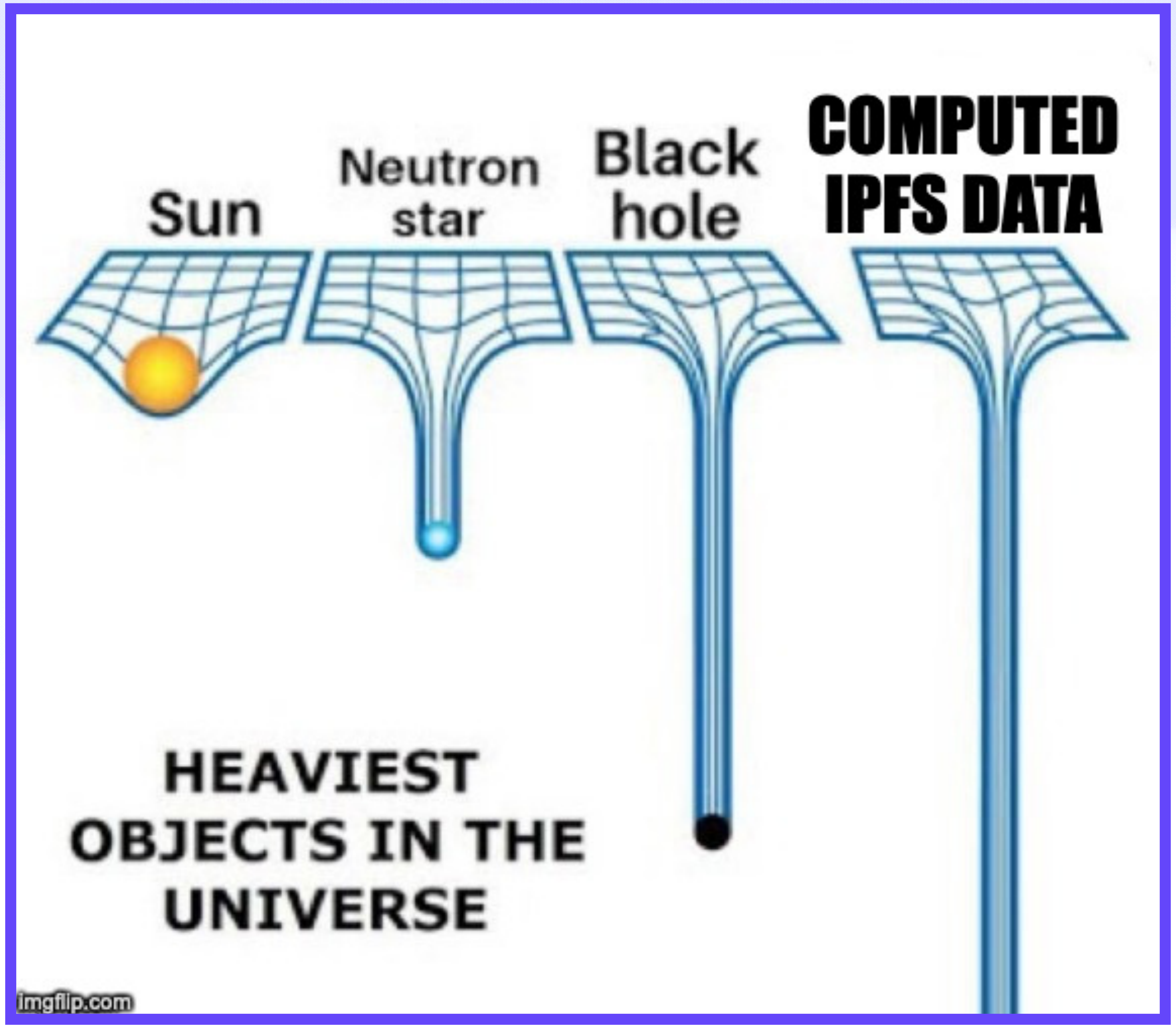
Solving for Data Gravity



Dataflow & Pipelining 🚰

Solving for Data Gravity

1. Fetch data
2. Compute on data
3. Output more data
4. GOTO step 1



Dataflow & Pipelining 🚰

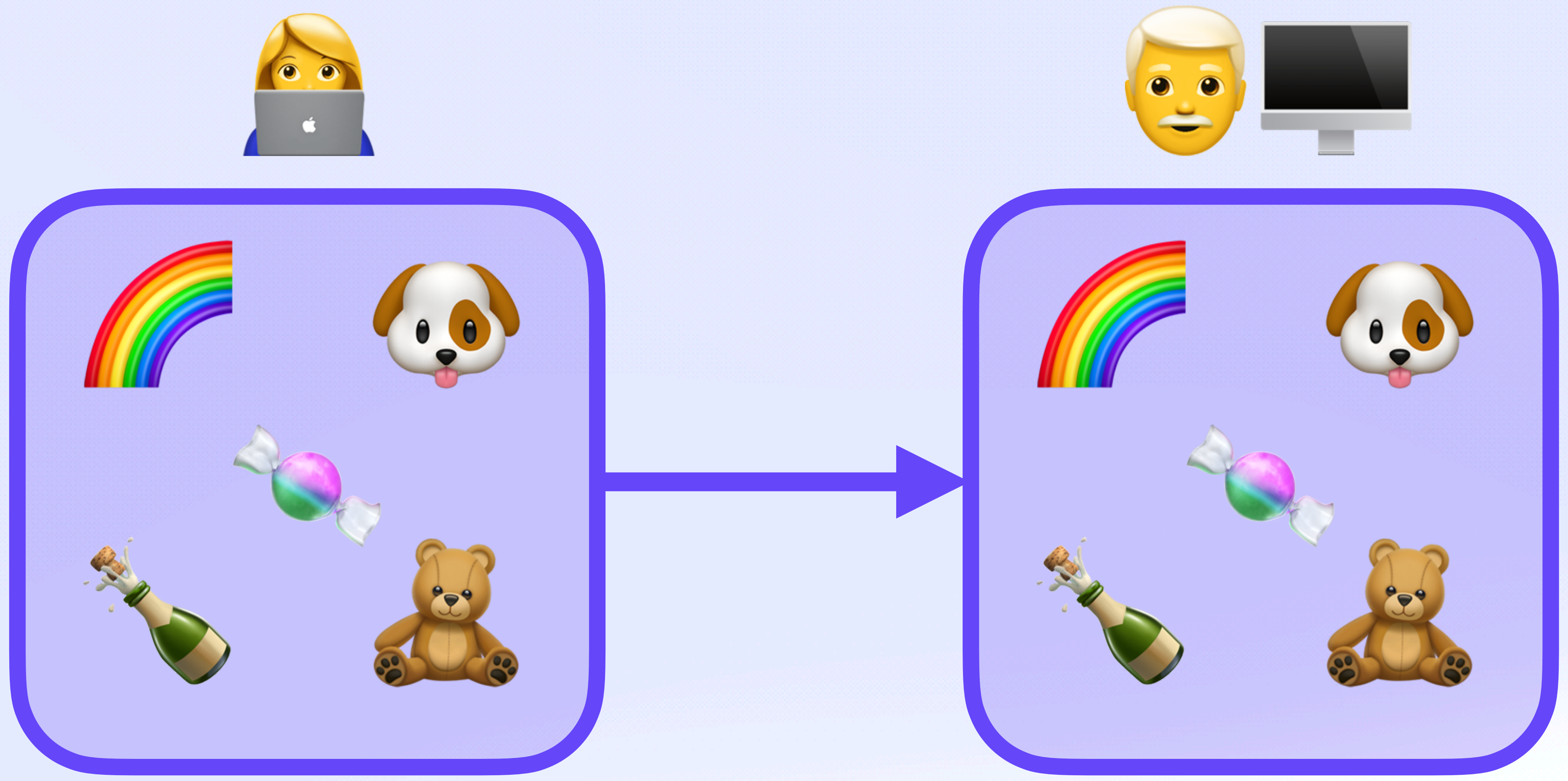
Transfer Authority

Dataflow & Pipelining 🚰

Transfer Authority

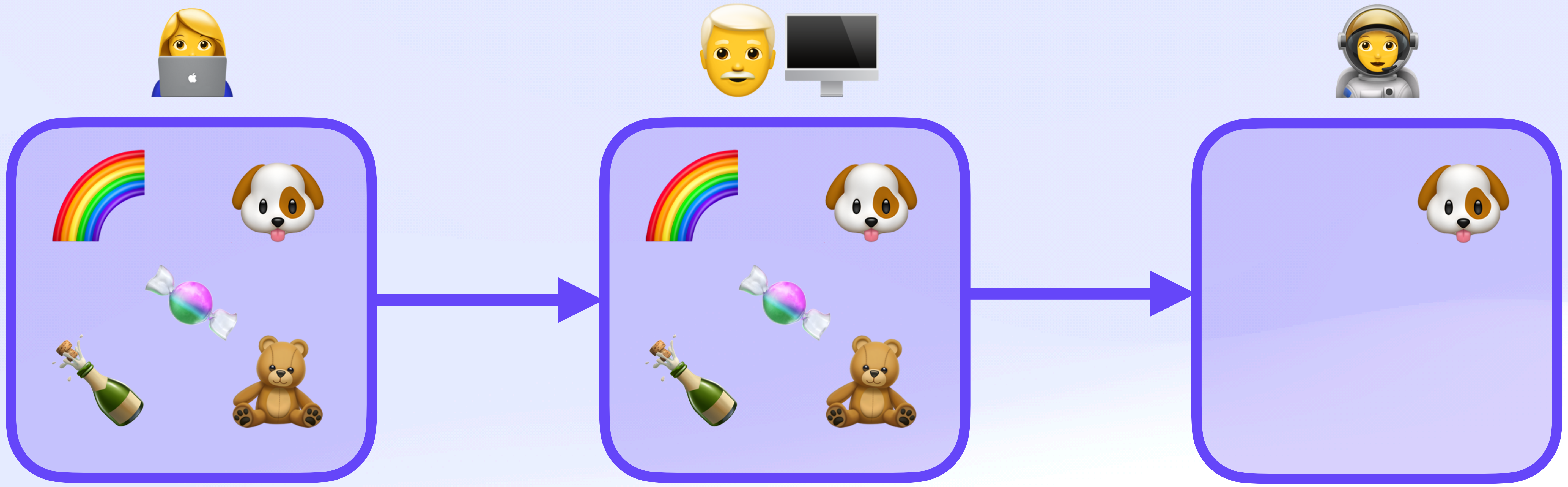
Dataflow & Pipelining 🚰

Transfer Authority



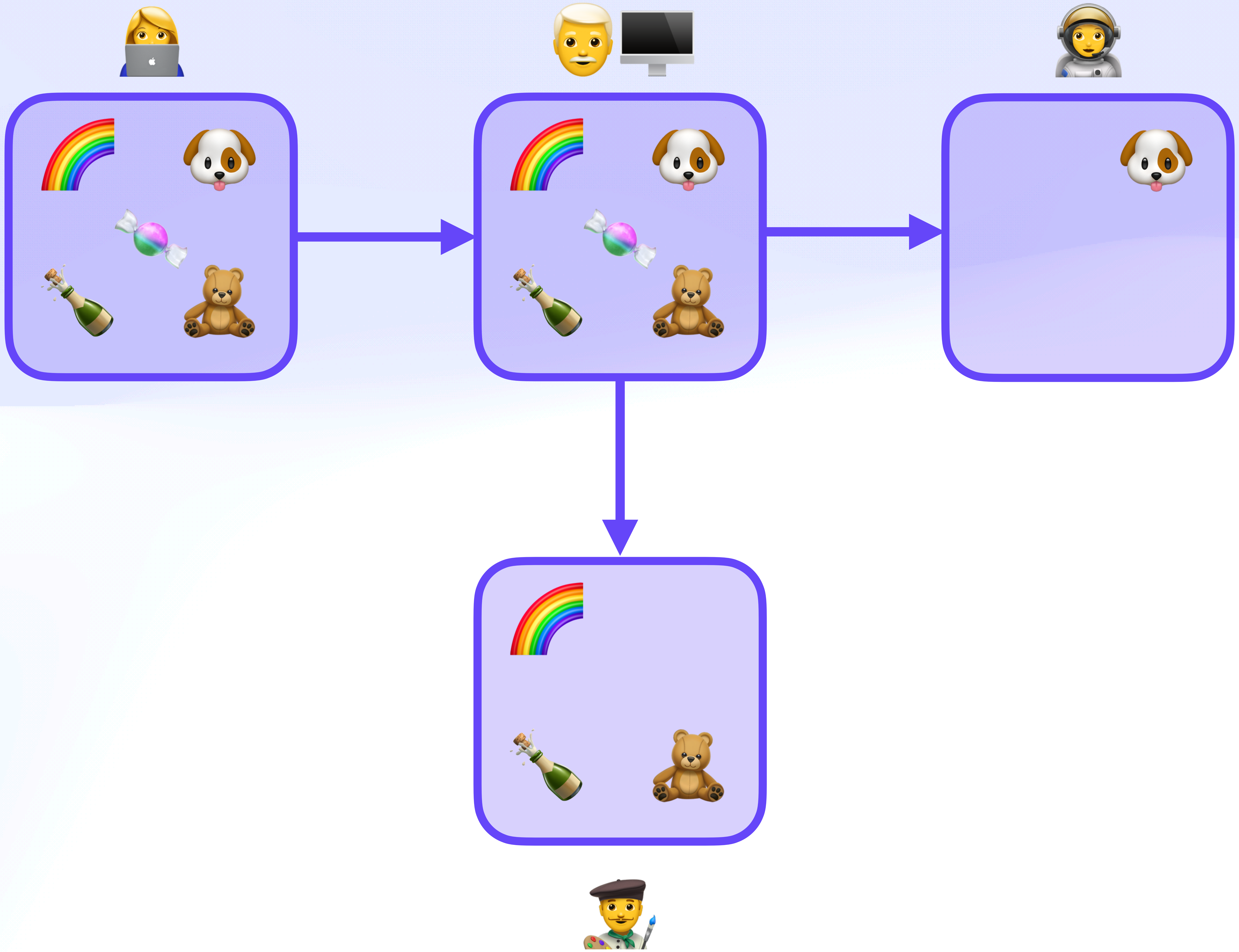
Dataflow & Pipelining 🚰

Transfer Authority



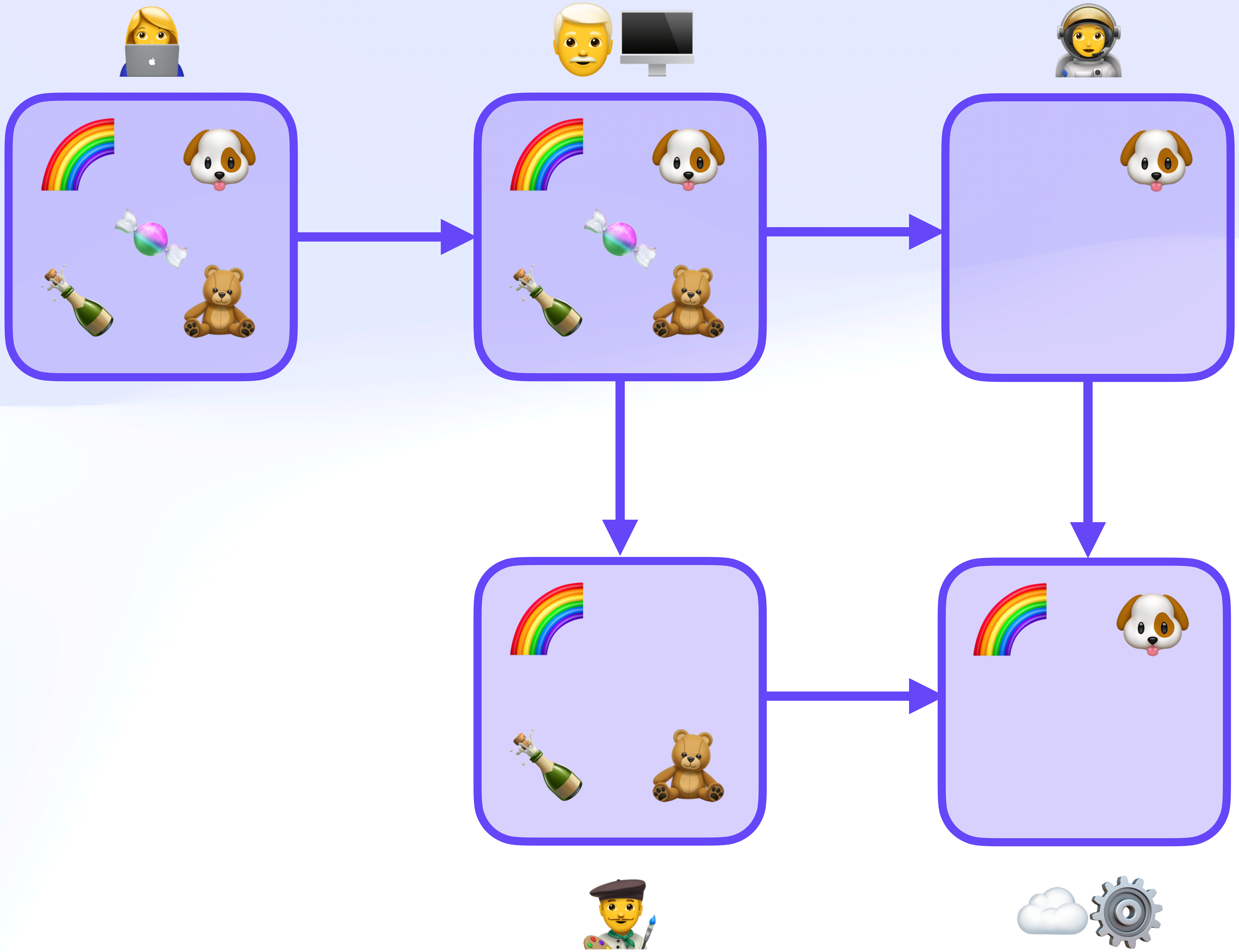
Dataflow & Pipelining 🚰

Transfer Authority



Dataflow & Pipelining 🚰

Transfer Authority



Dataflow & Pipelining 🚰

Distributed Invocation

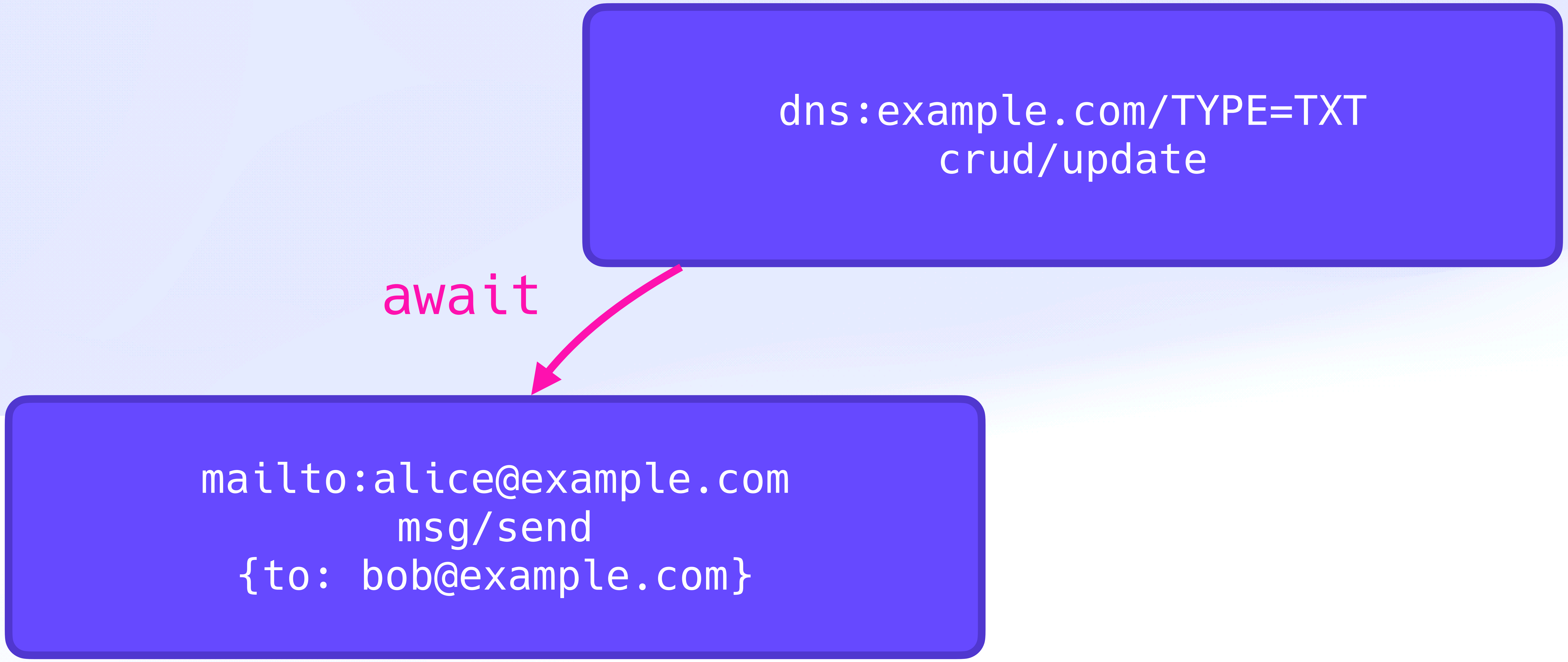
Dataflow & Pipelining 🚰

Distributed Invocation

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

Dataflow & Pipelining 🚰

Distributed Invocation



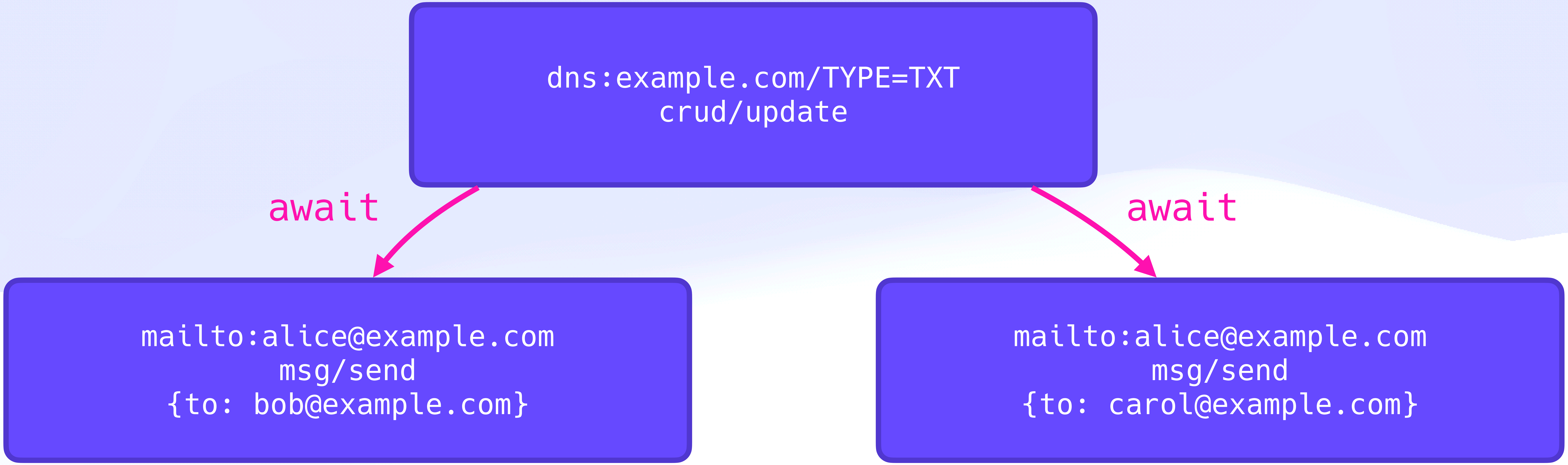
dns:example.com/TYPE=TXT
crud/update

await

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

Dataflow & Pipelining 🚰

Distributed Invocation



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

await

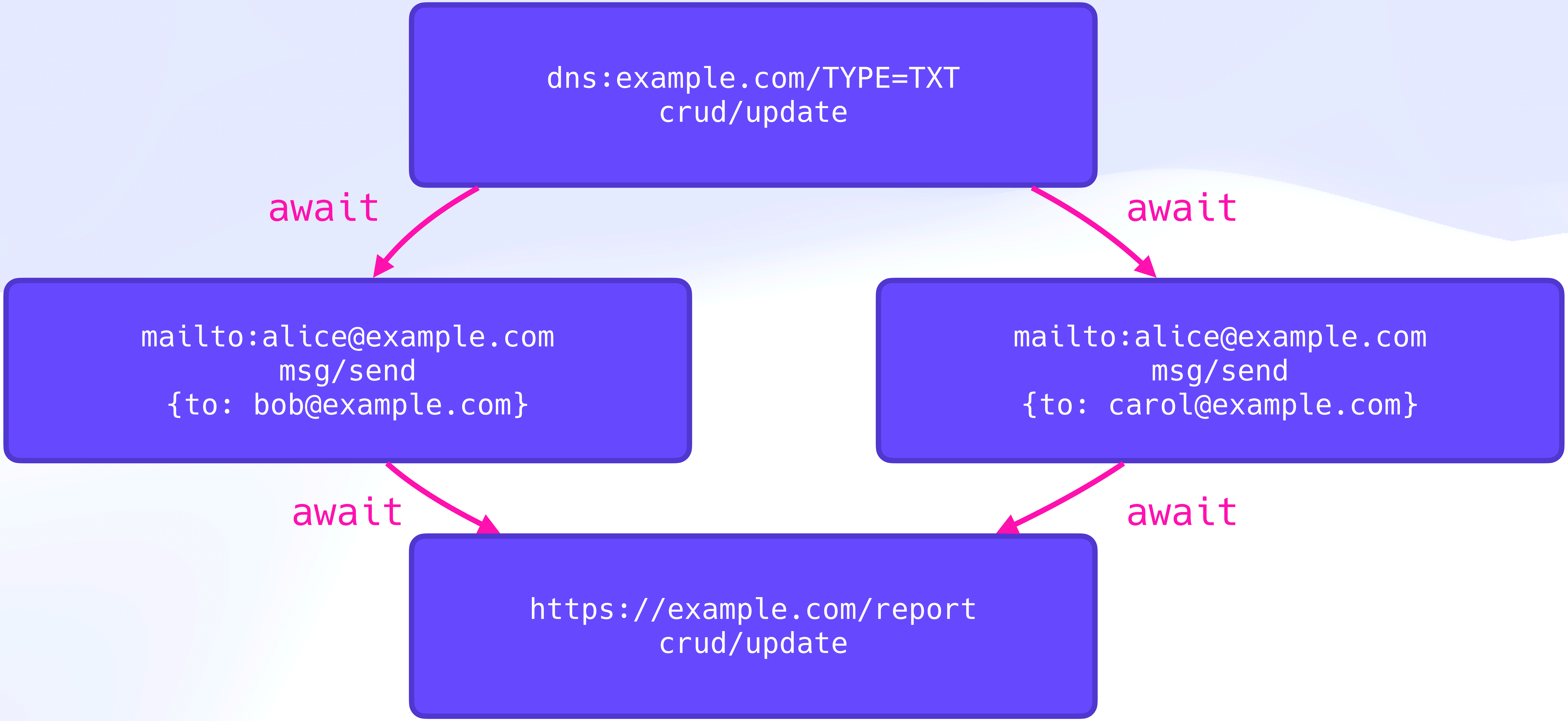
await

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

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

Dataflow & Pipelining 🚰

Distributed Invocation



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

await

await

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

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

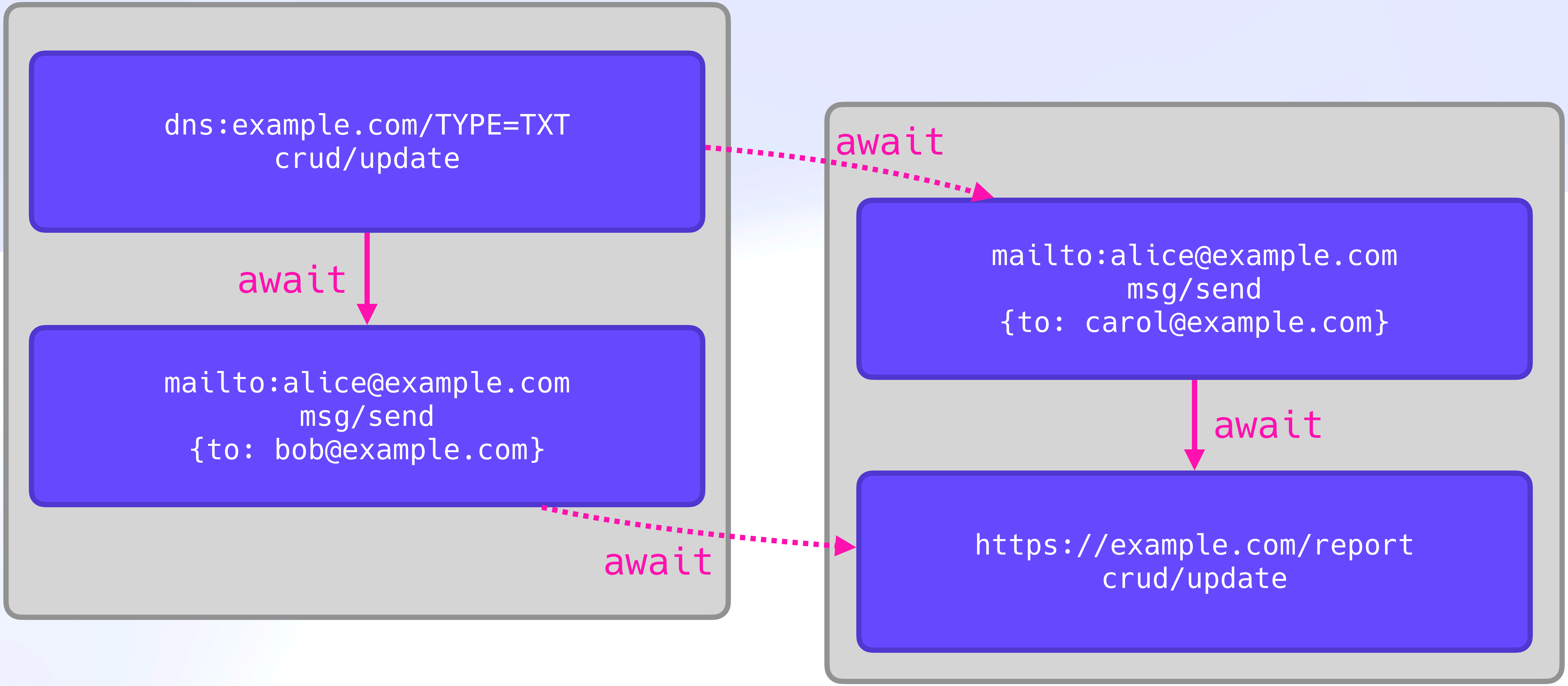
await

await

https://example.com/report
crud/update

Dataflow & Pipelining 🚰

Distributed Invocation



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

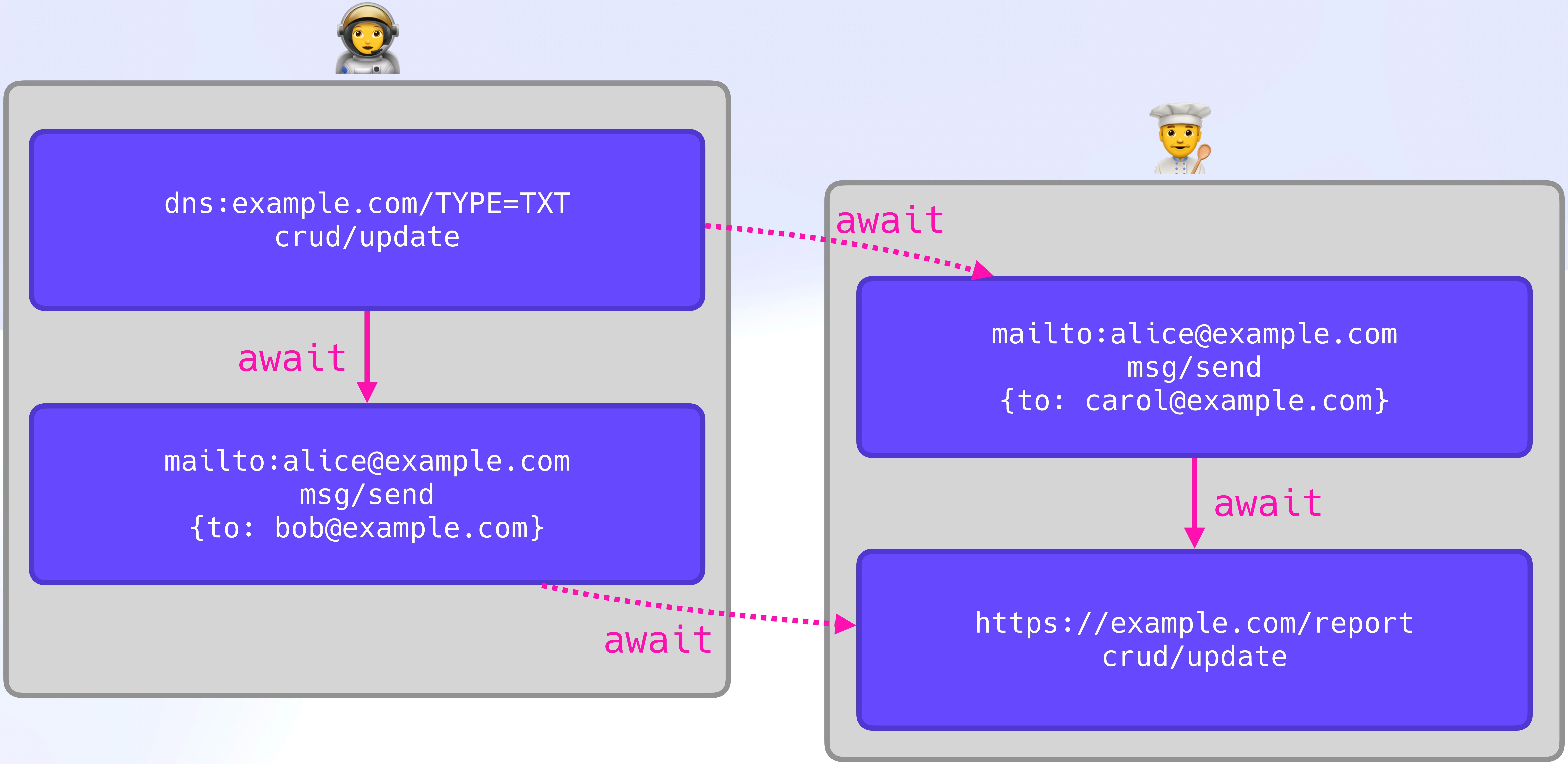
mailto:alice@example.com
msg/send
{to: bob@example.com}

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

https://example.com/report
crud/update

Dataflow & Pipelining 🚰

Distributed Invocation



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

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

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

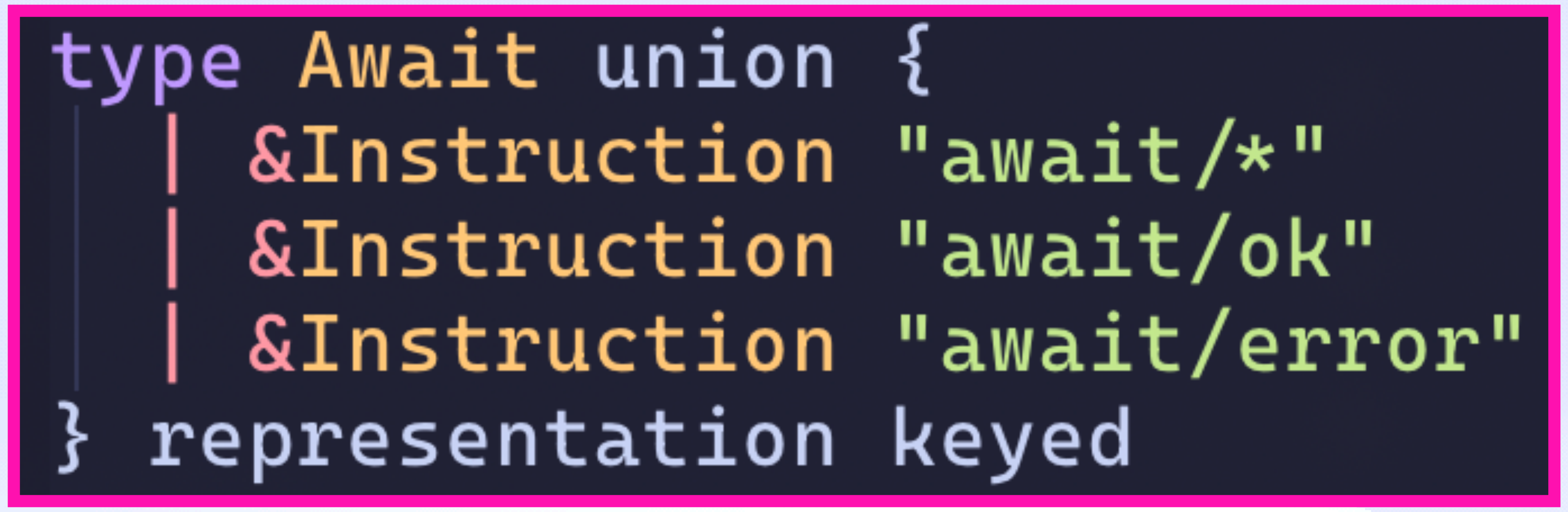
https://example.com/report
crud/update

Dataflow & Pipelining 🚰

Abstract Resolution Cycle

Dataflow & Pipelining 🚰

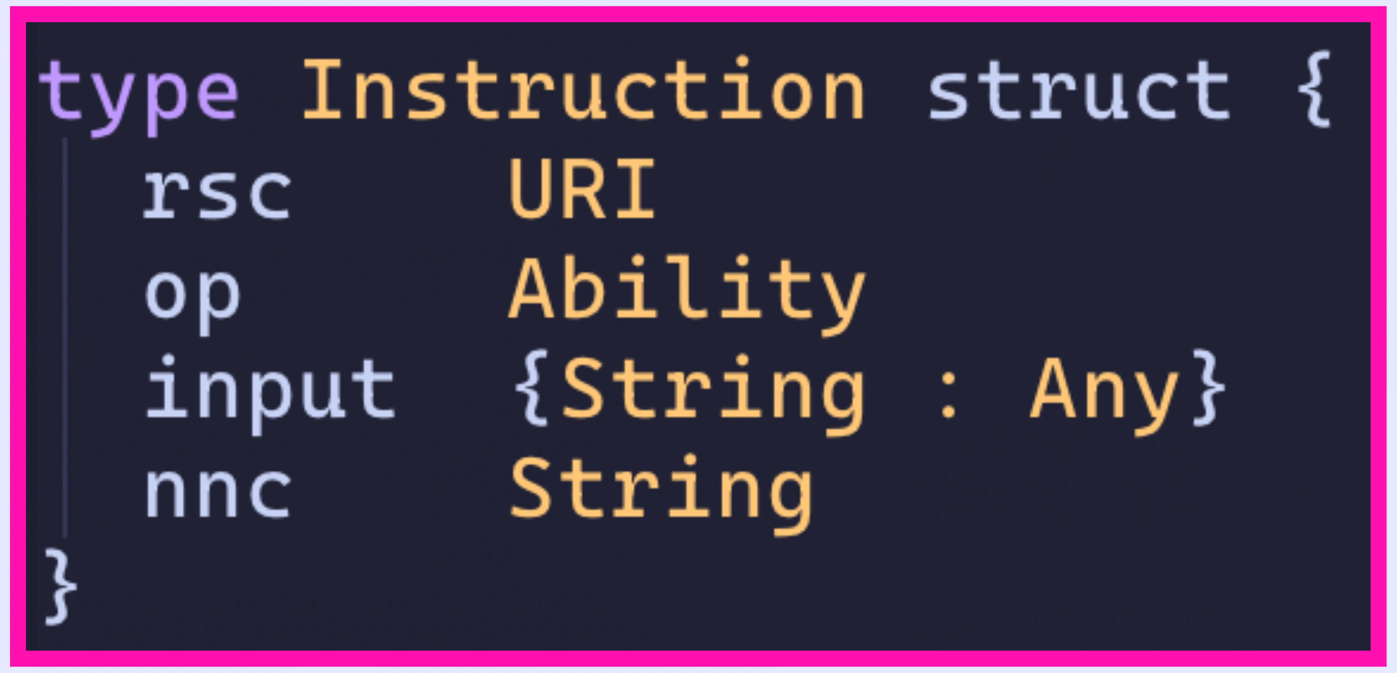
Abstract Resolution Cycle



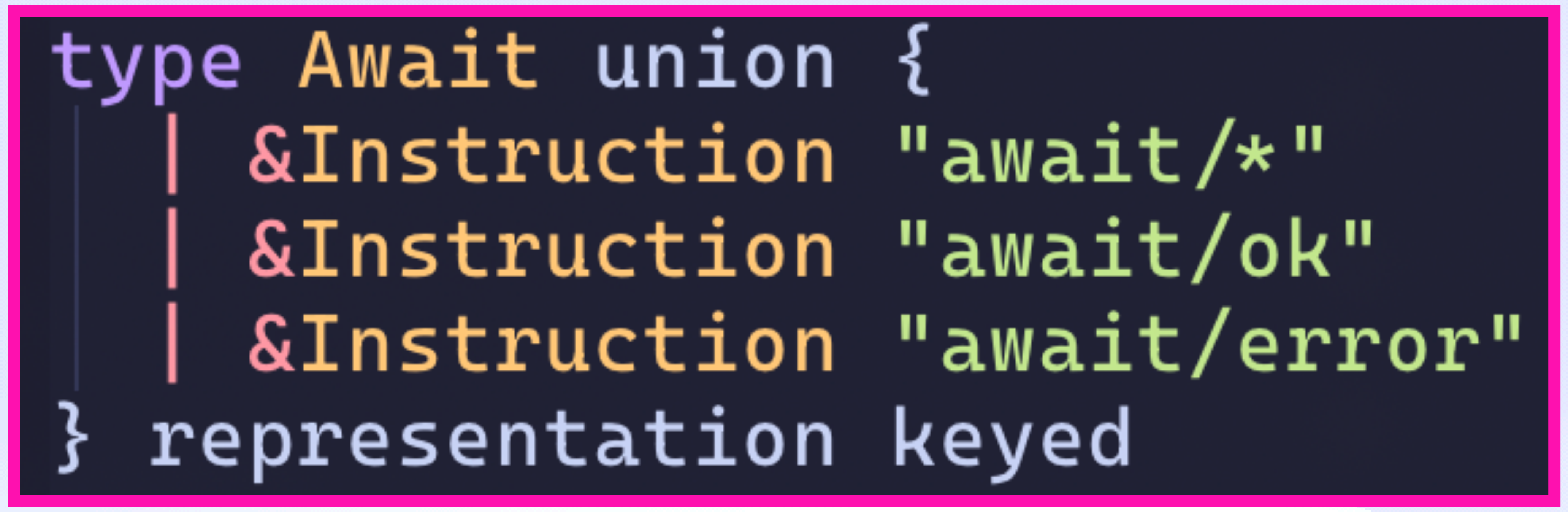
```
type Await union {
  | &Instruction "await/*"
  | &Instruction "await/ok"
  | &Instruction "await/error"
} representation keyed
```

Dataflow & Pipelining 🚰

Abstract Resolution Cycle



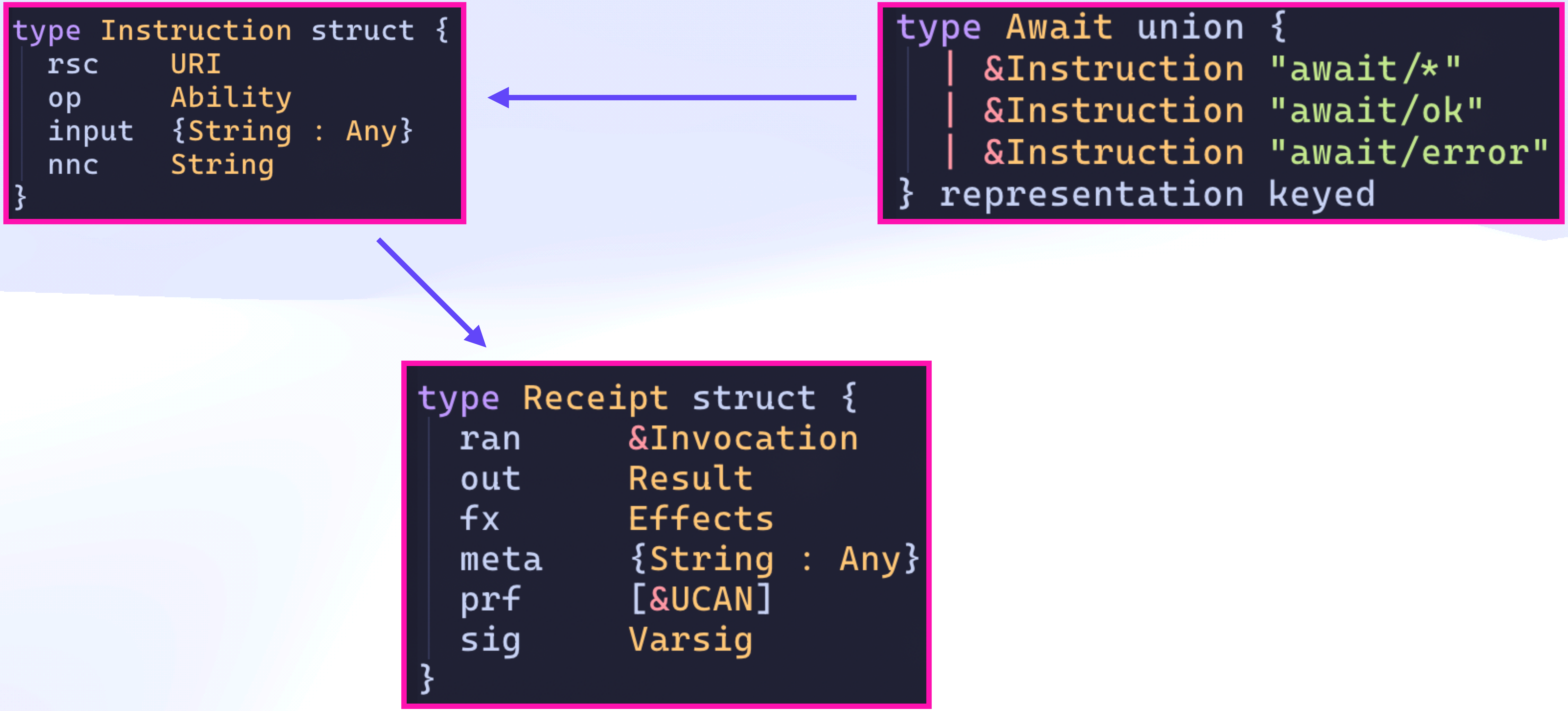
```
type Instruction struct {
  rsc     URI
  op      Ability
  input   {String : Any}
  nnc     String
}
```



```
type Await union {
  | &Instruction "await/*"
  | &Instruction "await/ok"
  | &Instruction "await/error"
} representation keyed
```

Dataflow & Pipelining 🚰

Abstract Resolution Cycle



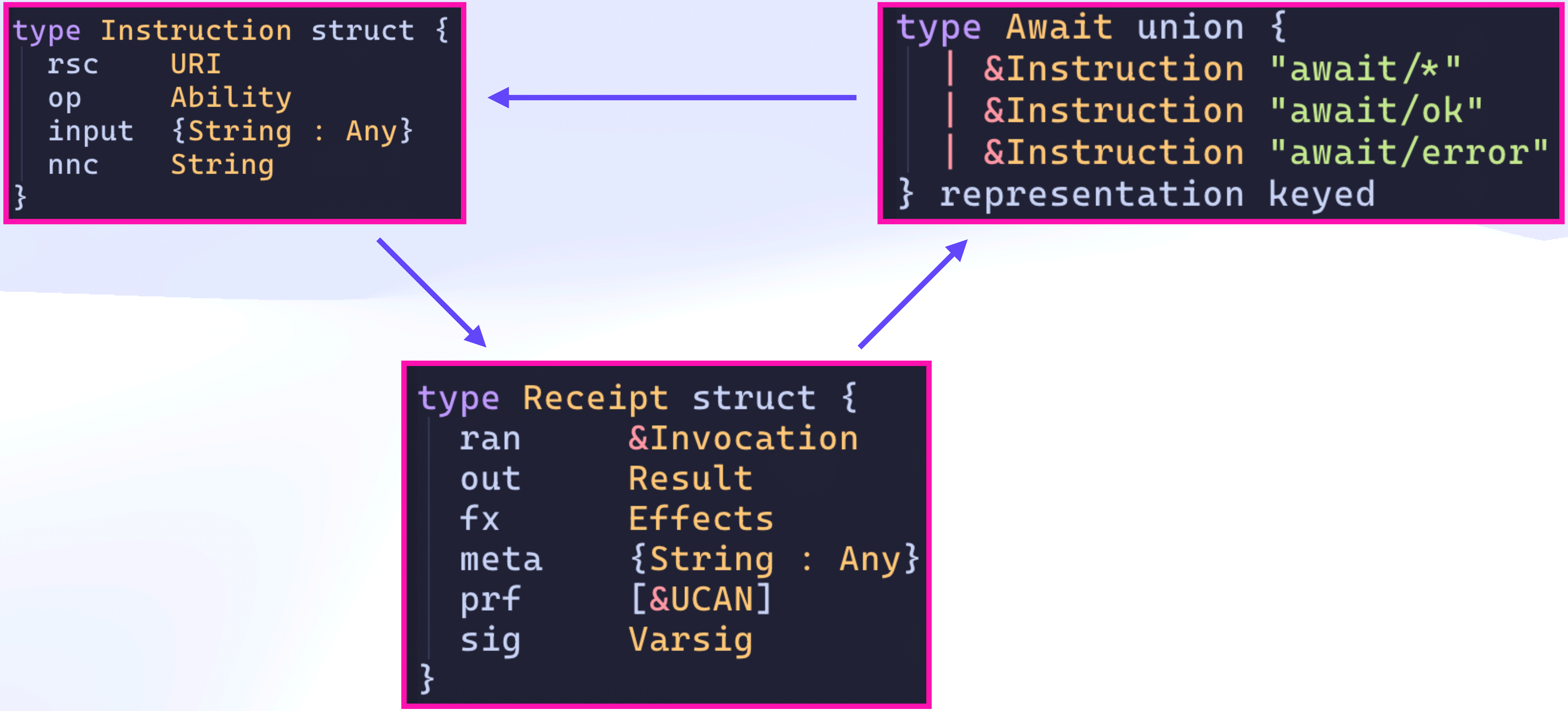
```
type Instruction struct {
  rsc     URI
  op      Ability
  input   {String : Any}
  nnc     String
}
```

```
type Await union {
  | &Instruction "await/*"
  | &Instruction "await/ok"
  | &Instruction "await/error"
} representation keyed
```

```
type Receipt struct {
  ran       &Invocation
  out       Result
  fx        Effects
  meta      {String : Any}
  prf       [&UCAN]
  sig       Varsig
}
```

Dataflow & Pipelining 🚰

Abstract Resolution Cycle



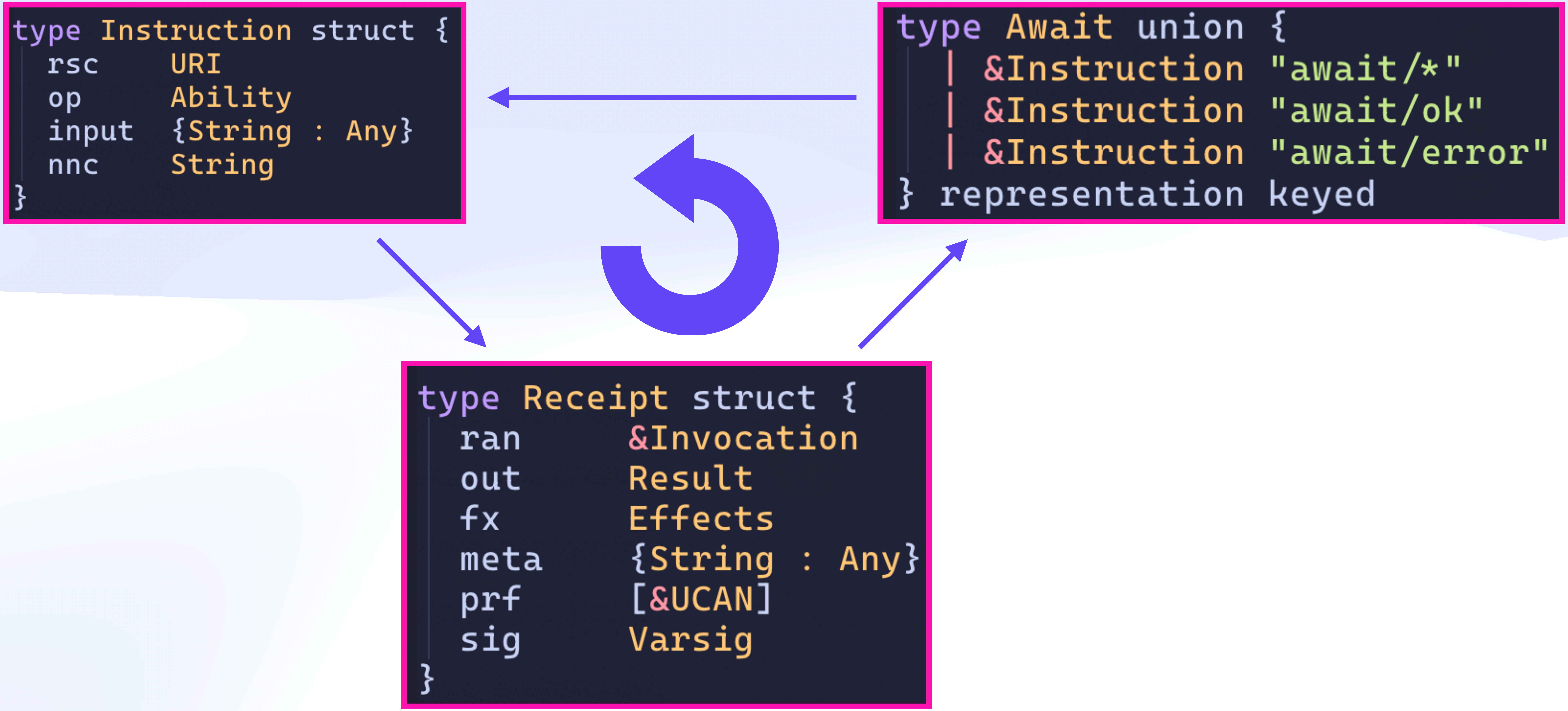
```
type Instruction struct {
  rsc    URI
  op     Ability
  input  {String : Any}
  nnc    String
}
```

```
type Await union {
  | &Instruction "await/*"
  | &Instruction "await/ok"
  | &Instruction "await/error"
} representation keyed
```

```
type Receipt struct {
  ran     &Invocation
  out     Result
  fx      Effects
  meta    {String : Any}
  prf     [&UCAN]
  sig     Varsig
}
```

Dataflow & Pipelining 🚰

Abstract Resolution Cycle



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

```
type Await union {
  | &Instruction "await/*"
  | &Instruction "await/ok"
  | &Instruction "await/error"
} representation keyed
```

```
type Receipt struct {
  ran       &Invocation
  out       Result
  fx        Effects
  meta      {String : Any}
  prf       [&UCAN]
  sig       Varsig
}
```

Dataflow & Pipelining 🚰

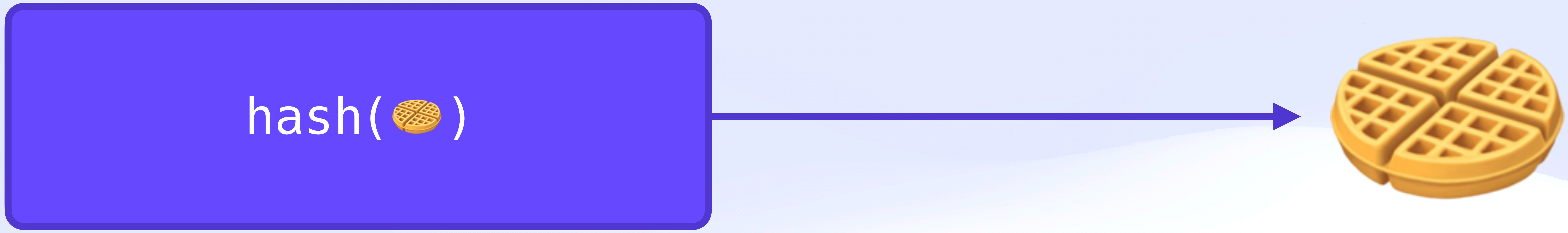
Input Addressing

Dataflow & Pipelining 🚰

Input Addressing

Dataflow & Pipelining 🚰

Input Addressing



hash(🧇)

Dataflow & Pipelining 🚰

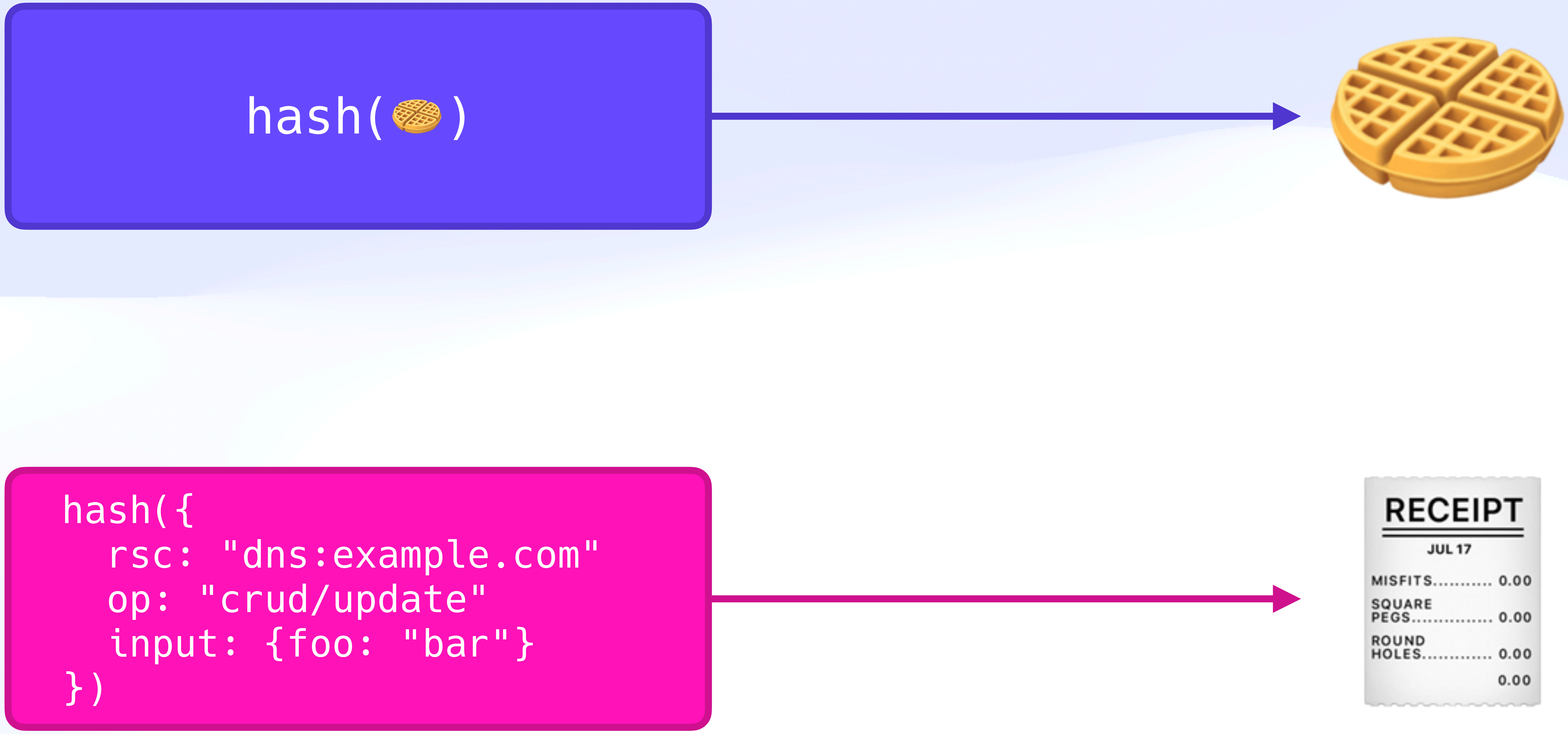
Input Addressing

hash(🧇)

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

Dataflow & Pipelining 🚰

Input Addressing

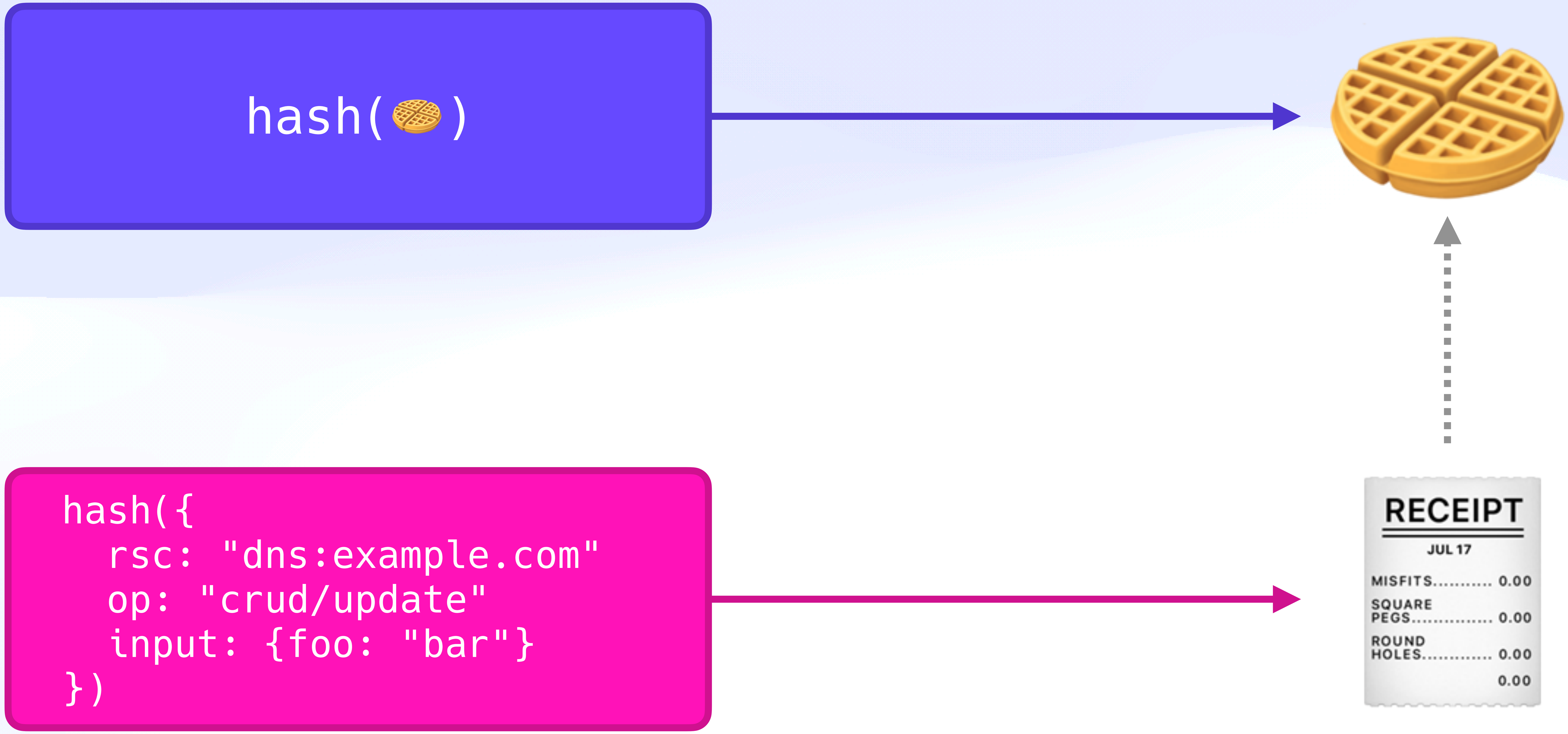


hash(🧇)

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

Dataflow & Pipelining 🚰

Input Addressing



hash(🧇)

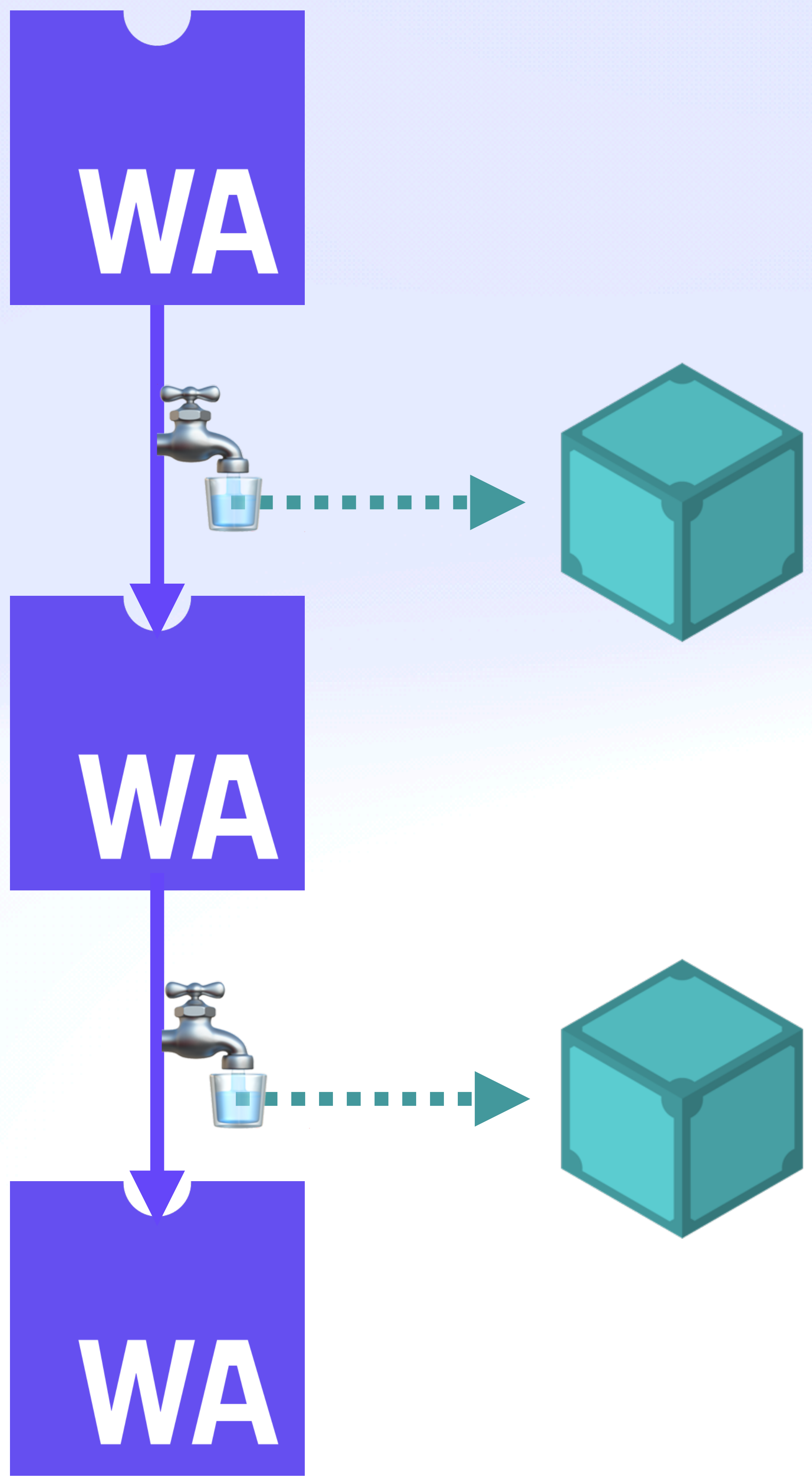
```
hash({
  rsc: "dns:example.com"
  op: "crud/update"
  input: {foo: "bar"}
})
```

Dataflow & Pipelining 🚰

Cache Intermediate Results

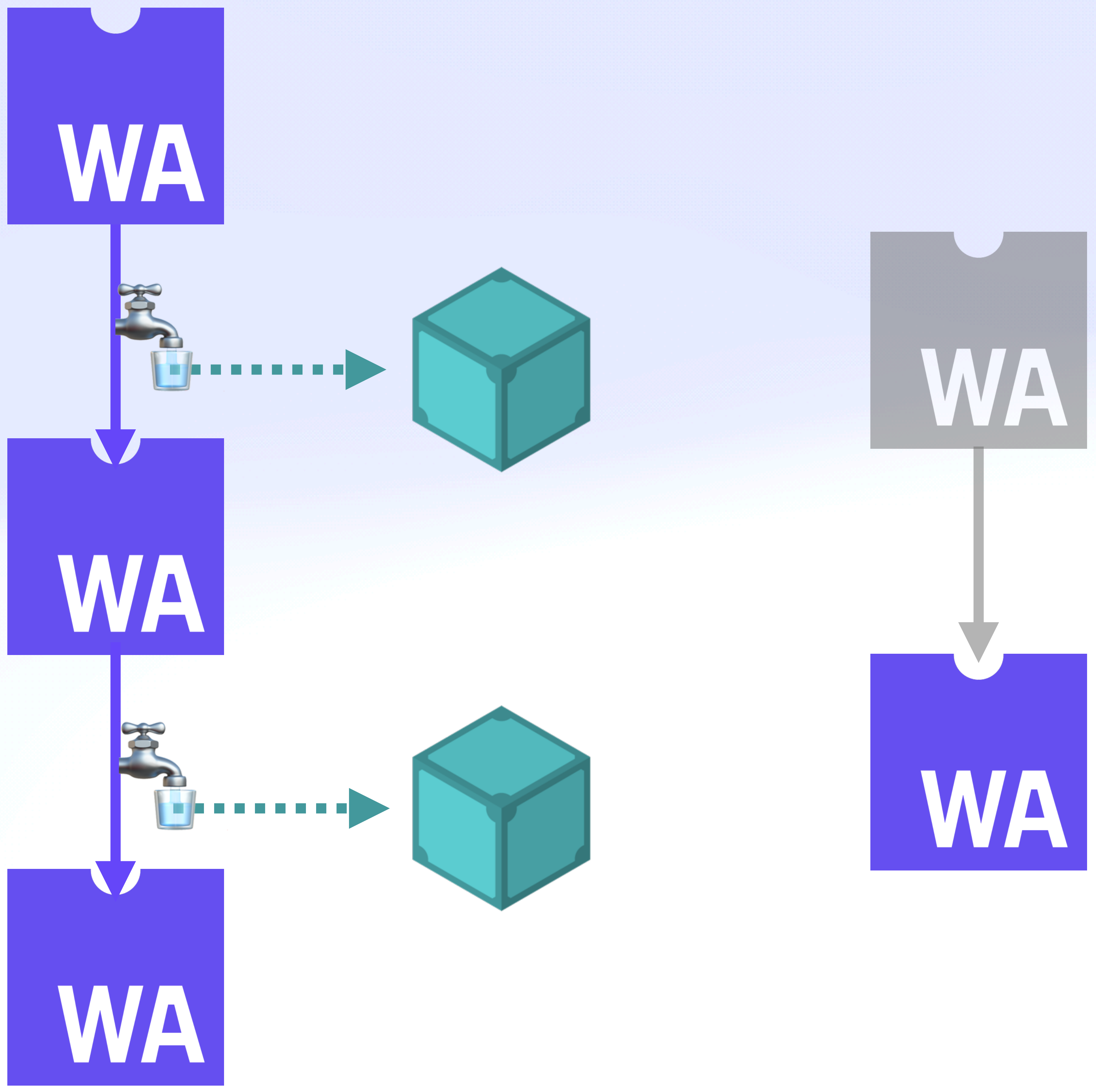
Dataflow & Pipelining 🚰

Cache Intermediate Results



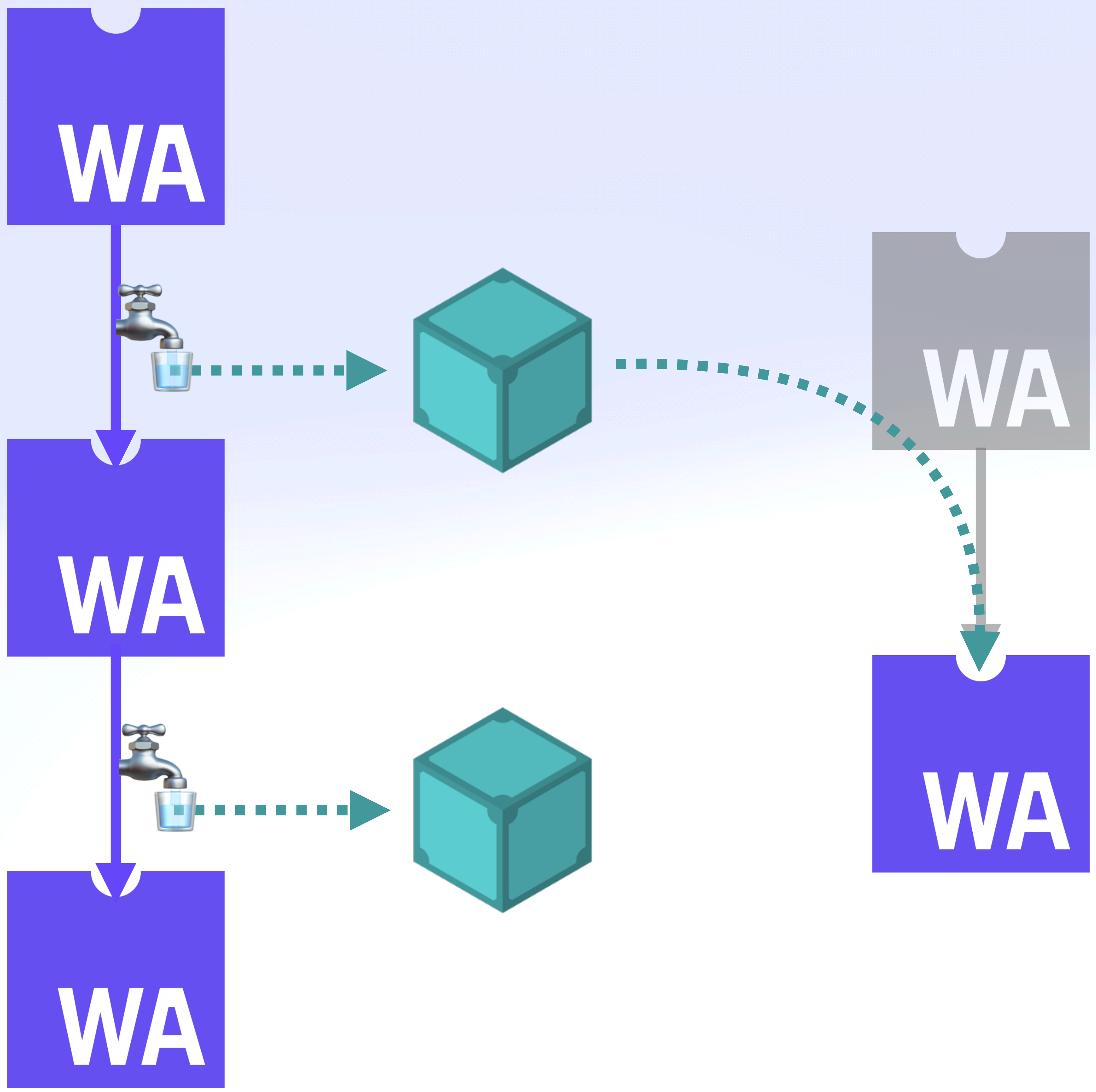
Dataflow & Pipelining 🚰

Cache Intermediate Results



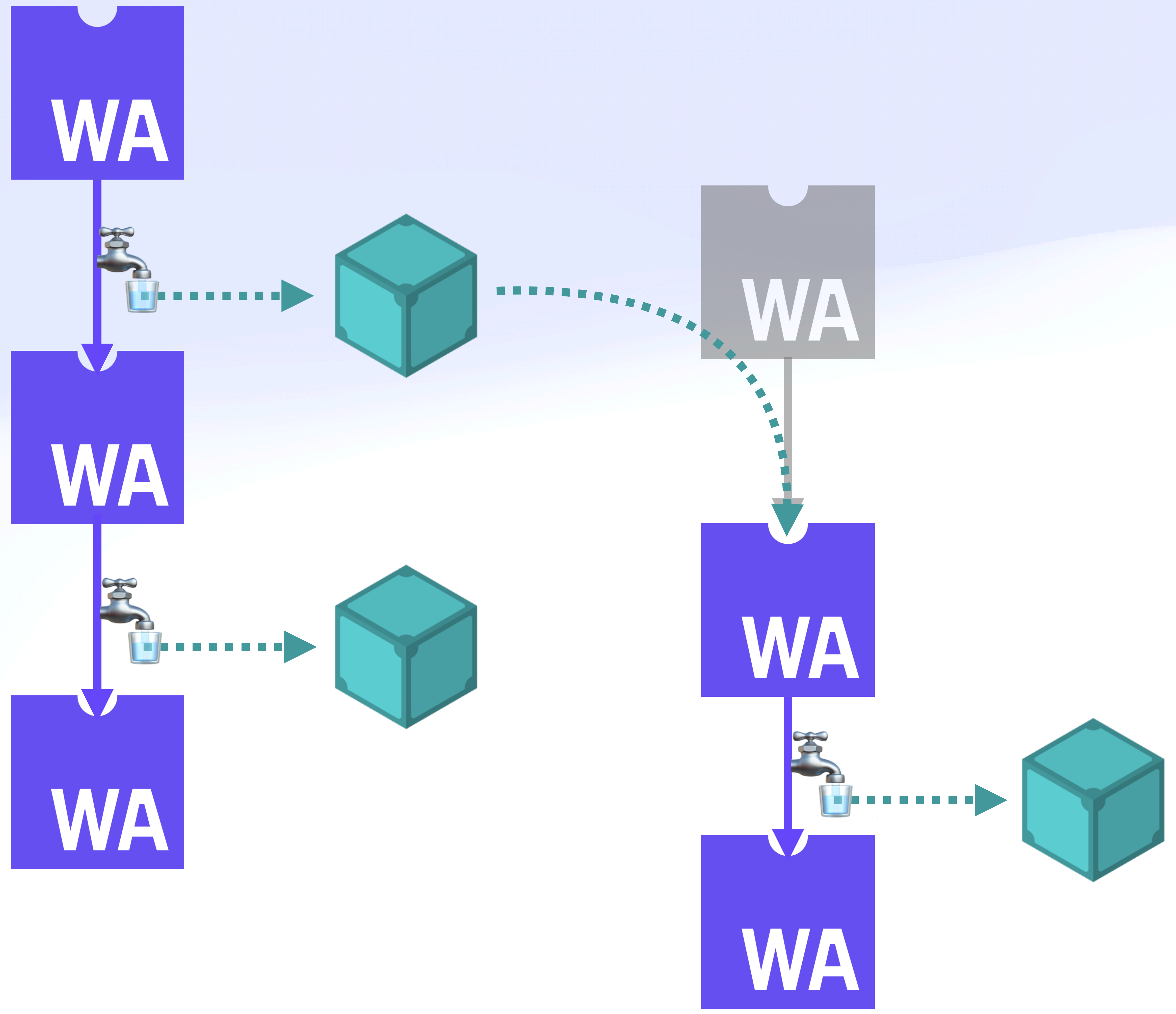
Dataflow & Pipelining 🚰

Cache Intermediate Results



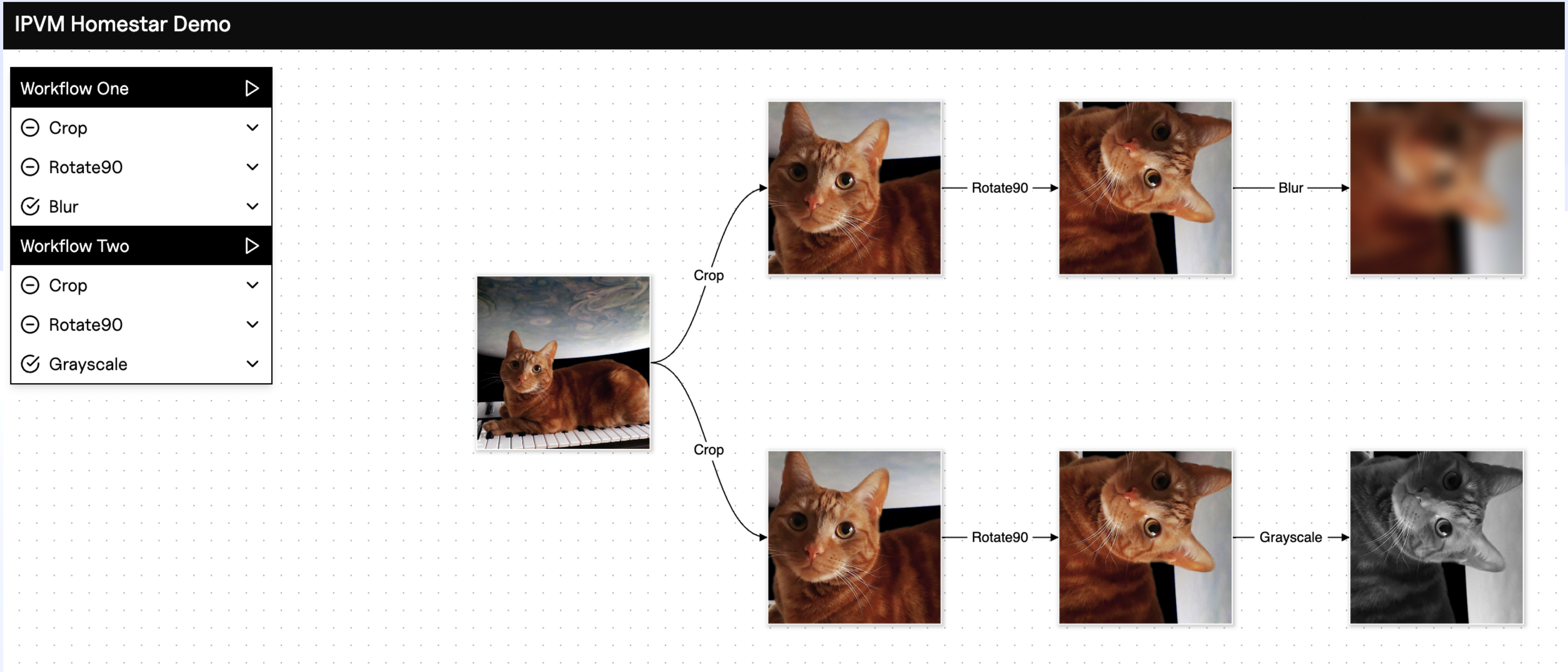
Dataflow & Pipelining 🚰

Cache Intermediate Results



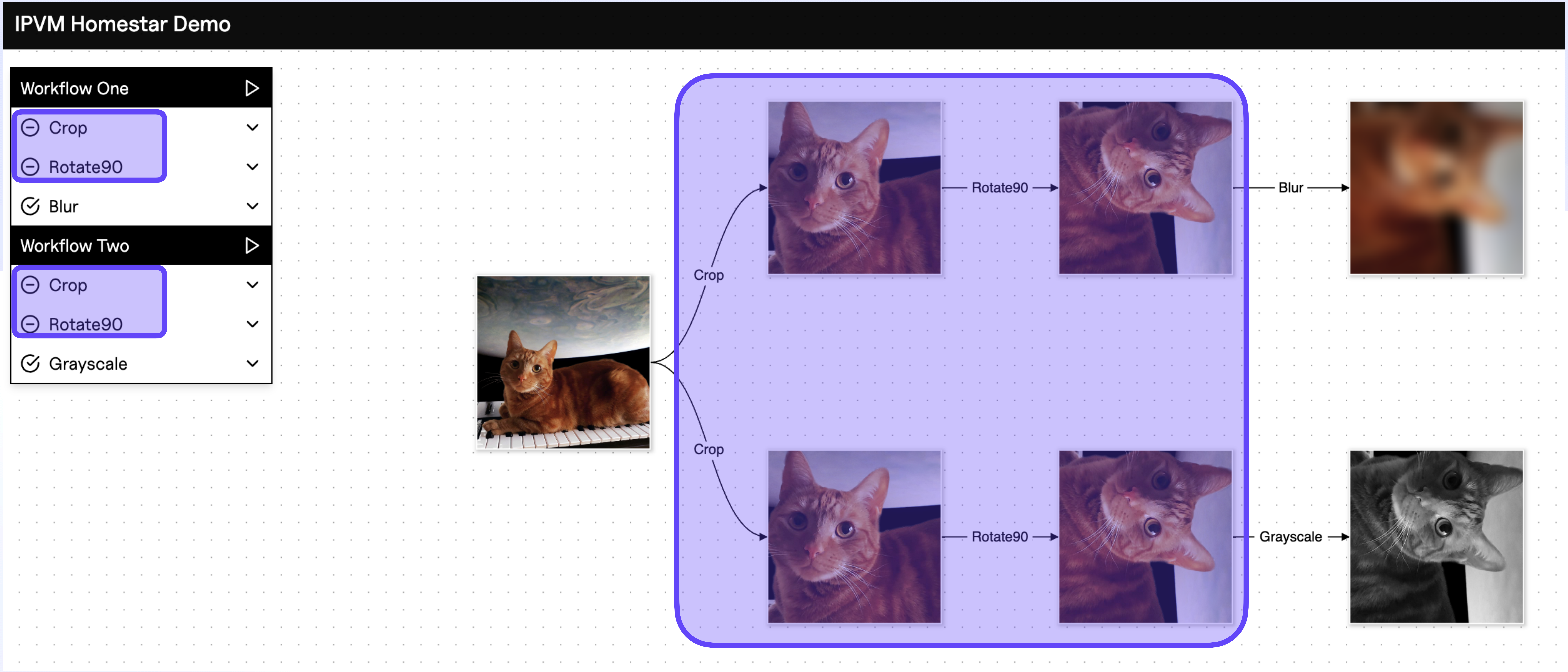
Dataflow & Pipelining 🚰

Reduce, Resume, Recycle ♻️



Dataflow & Pipelining 🚰

Reduce, Resume, Recycle ♻️



IPVM Homestar Demo

Workflow One ▷

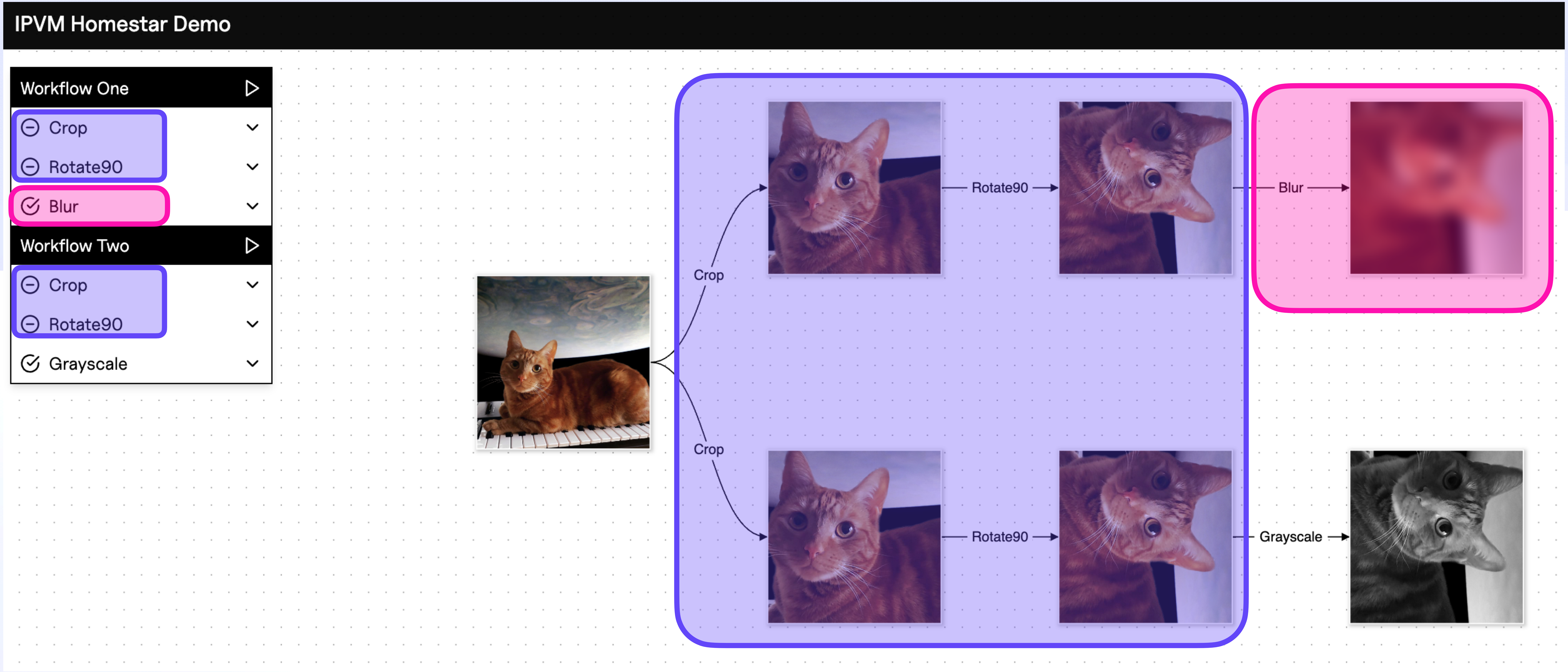
- ⊖ Crop
- ⊖ Rotate90
- ✓ Blur

Workflow Two ▷

- ⊖ Crop
- ⊖ Rotate90
- ✓ Grayscale

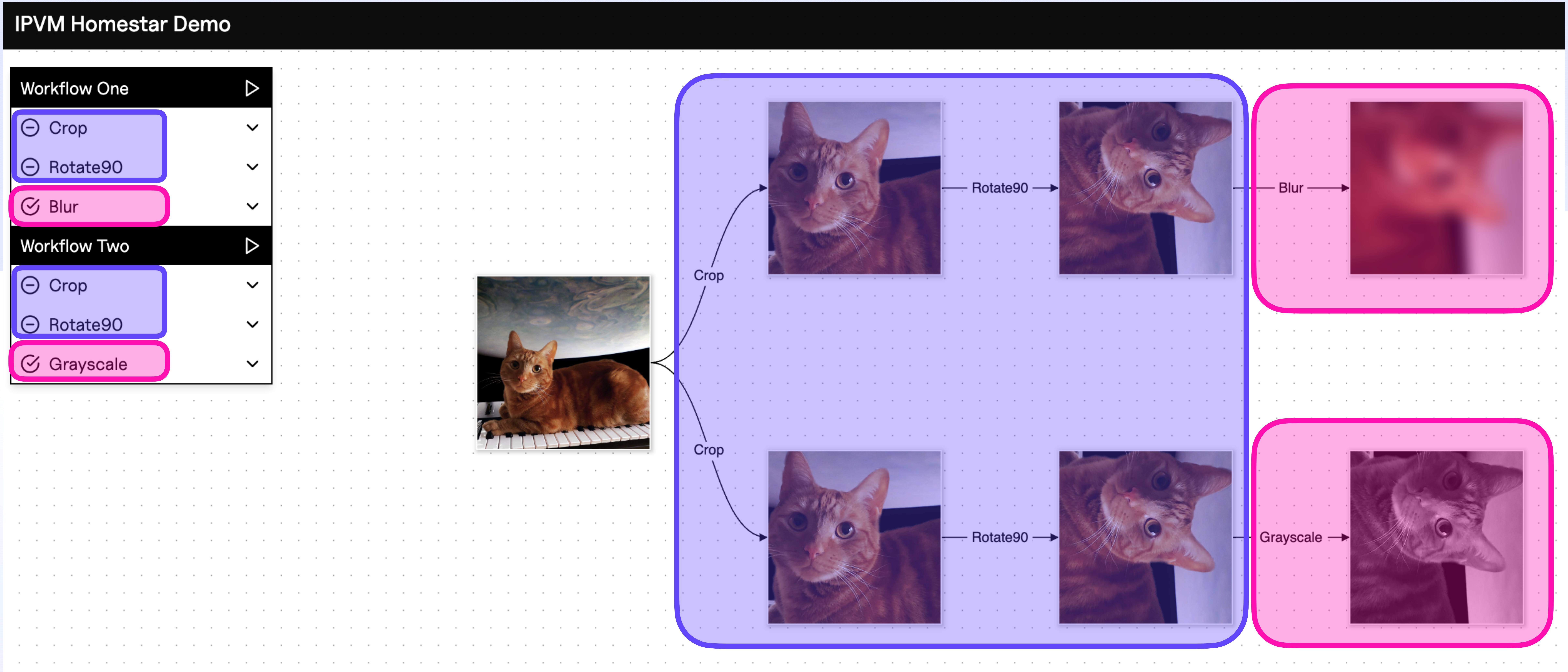
Dataflow & Pipelining 🚰

Reduce, Resume, Recycle ♻️


IPVM Homestar Demo

Dataflow & Pipelining 🚰

Reduce, Resume, Recycle ♻️



Decentralised Memoization

🧾

Decentralised Memoization 🧾

Decentralised Memoization 🧾

> [T]he ***speed of light*** is constant and
> ***New York is not getting any closer to Tokyo***.
> As hardware continues to improve,
> the ***latency barrier*** between distant
> machines will ***increasingly dominate***

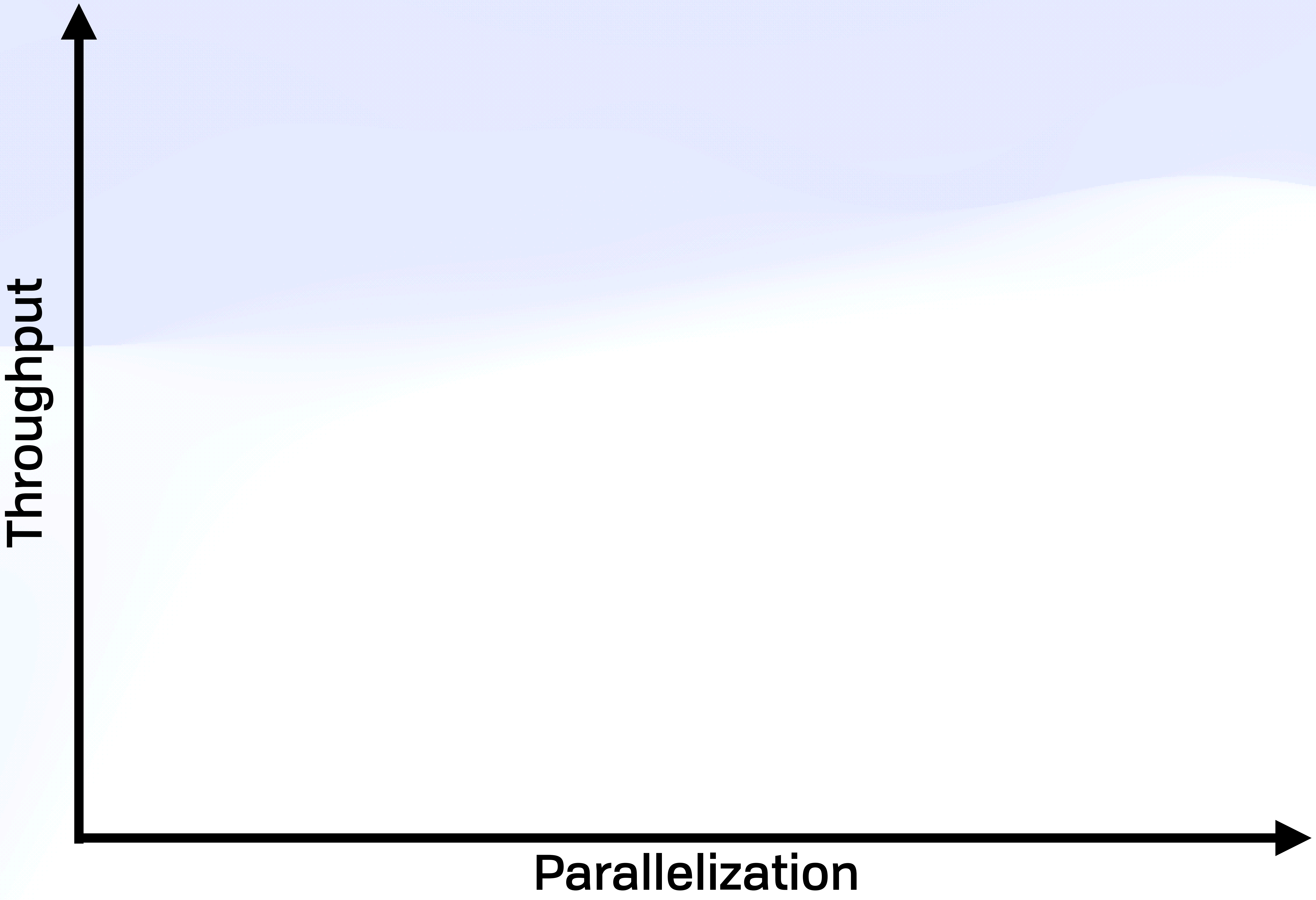
Mark Miller, Robust Composition

Decentralised Memoization 🧾

With a Little Scale From My Friends

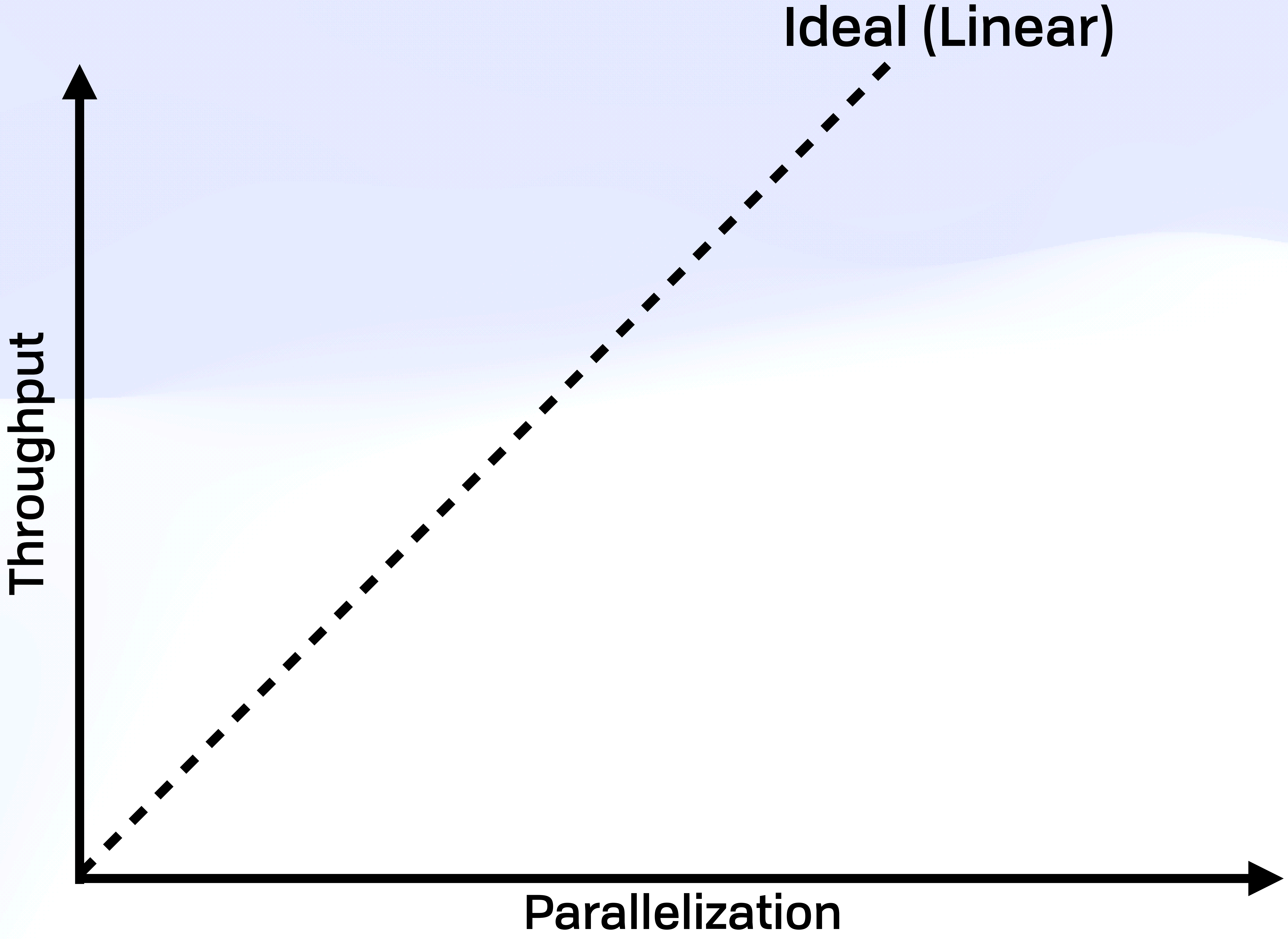
Decentralised Memoization 🧾

With a Little Scale From My Friends



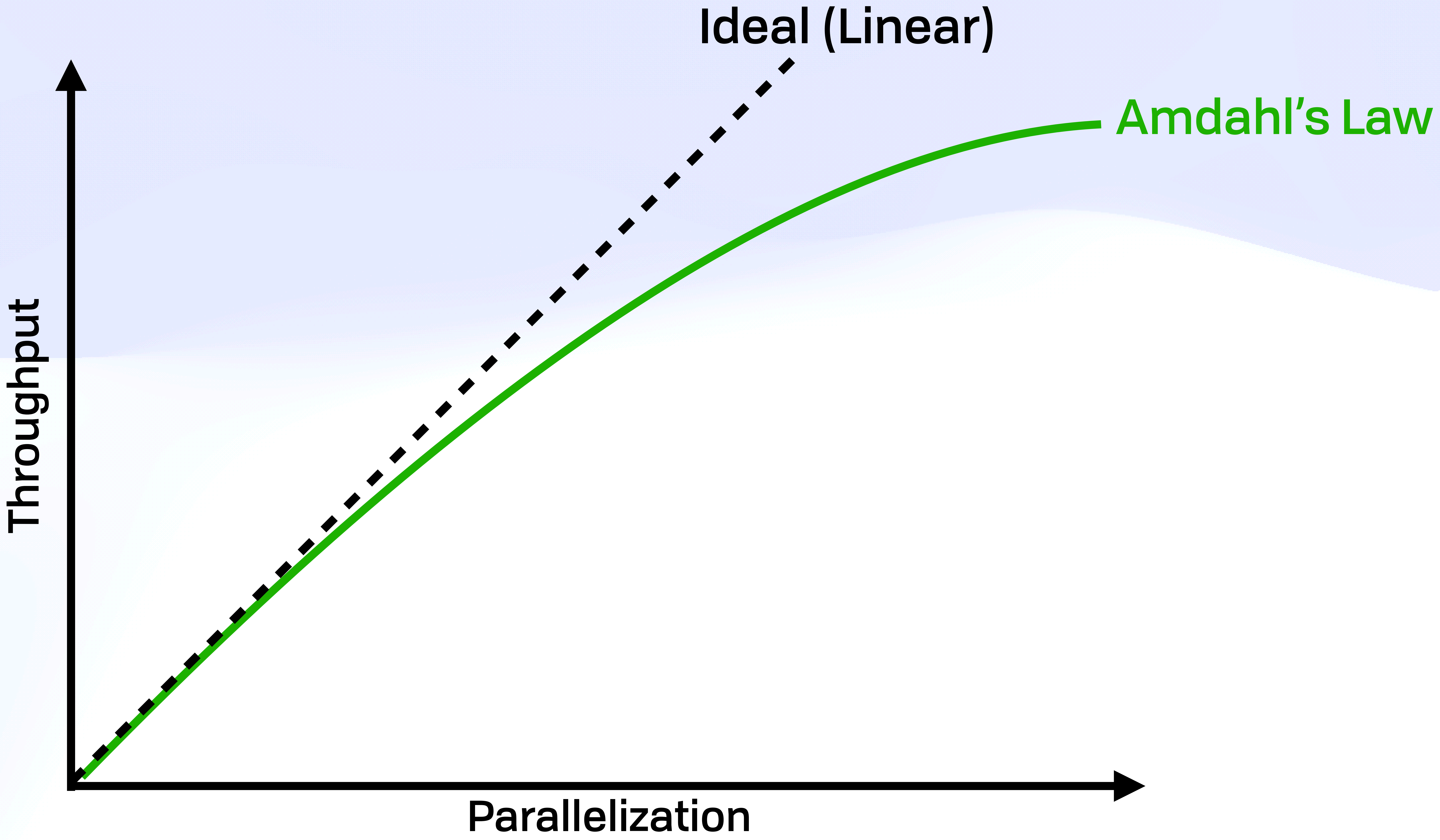
Decentralised Memoization 🧾

With a Little Scale From My Friends



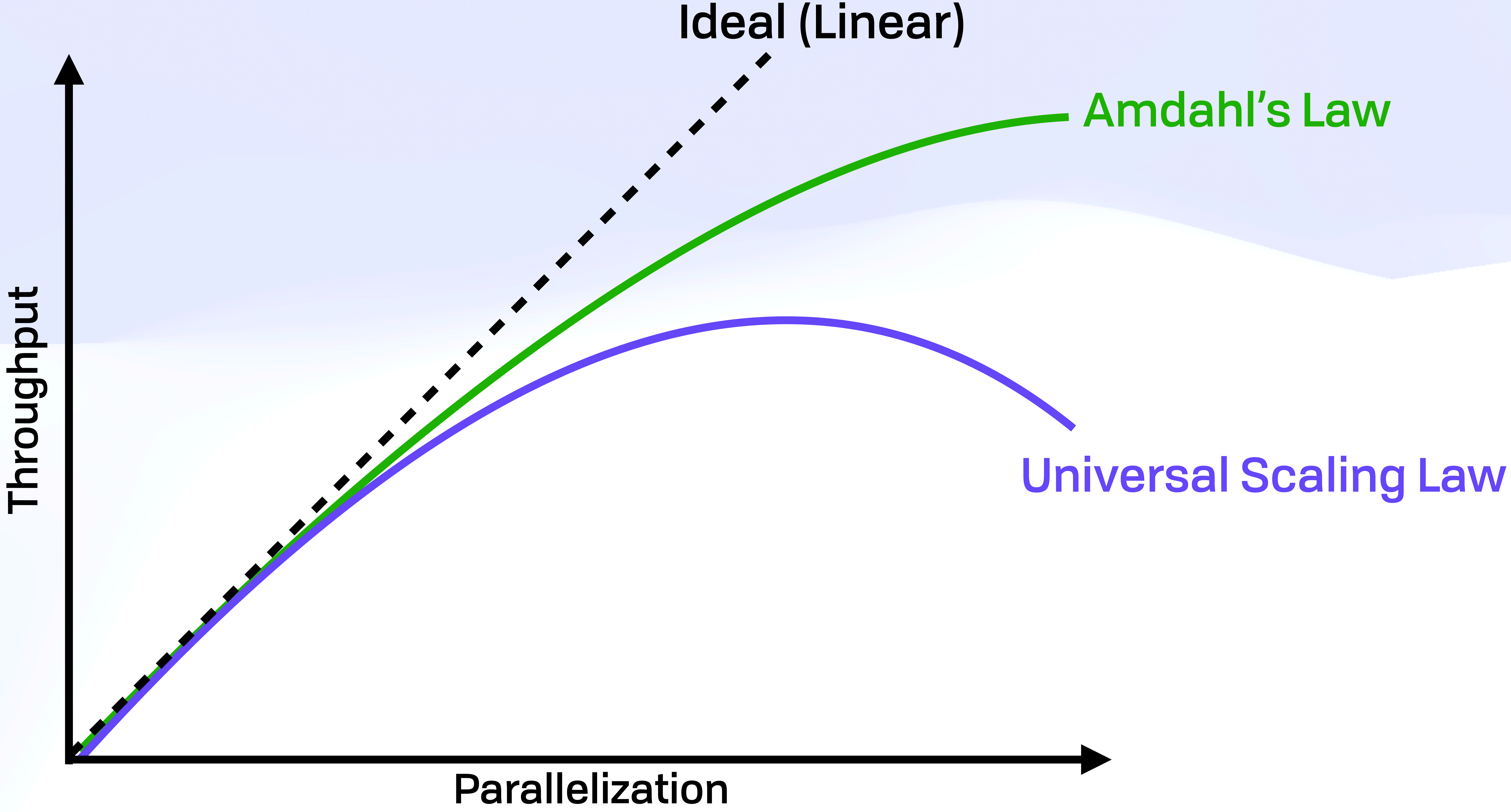
Decentralised Memoization 🧾

With a Little Scale From My Friends



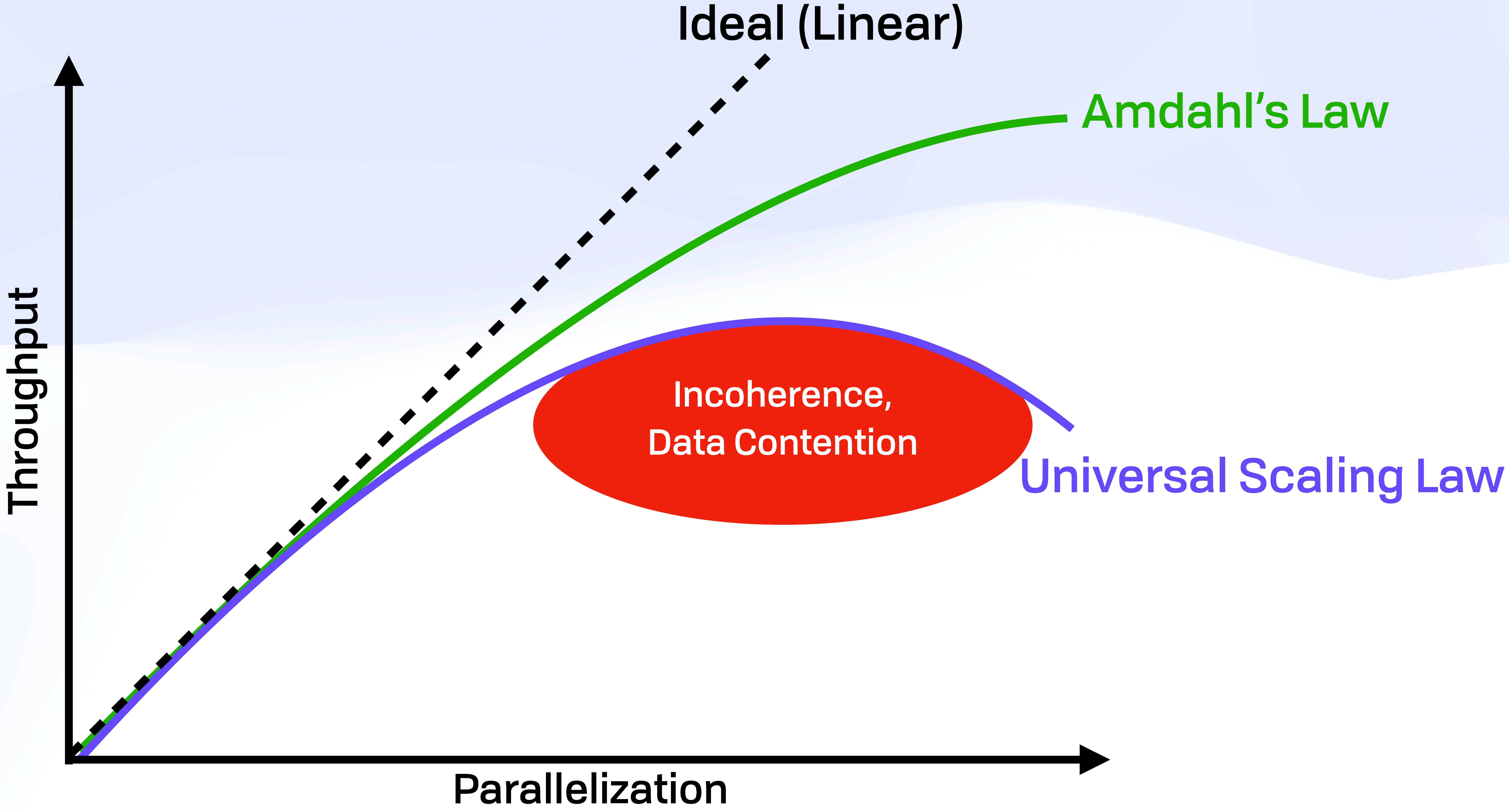
Decentralised Memoization 🧾

With a Little Scale From My Friends



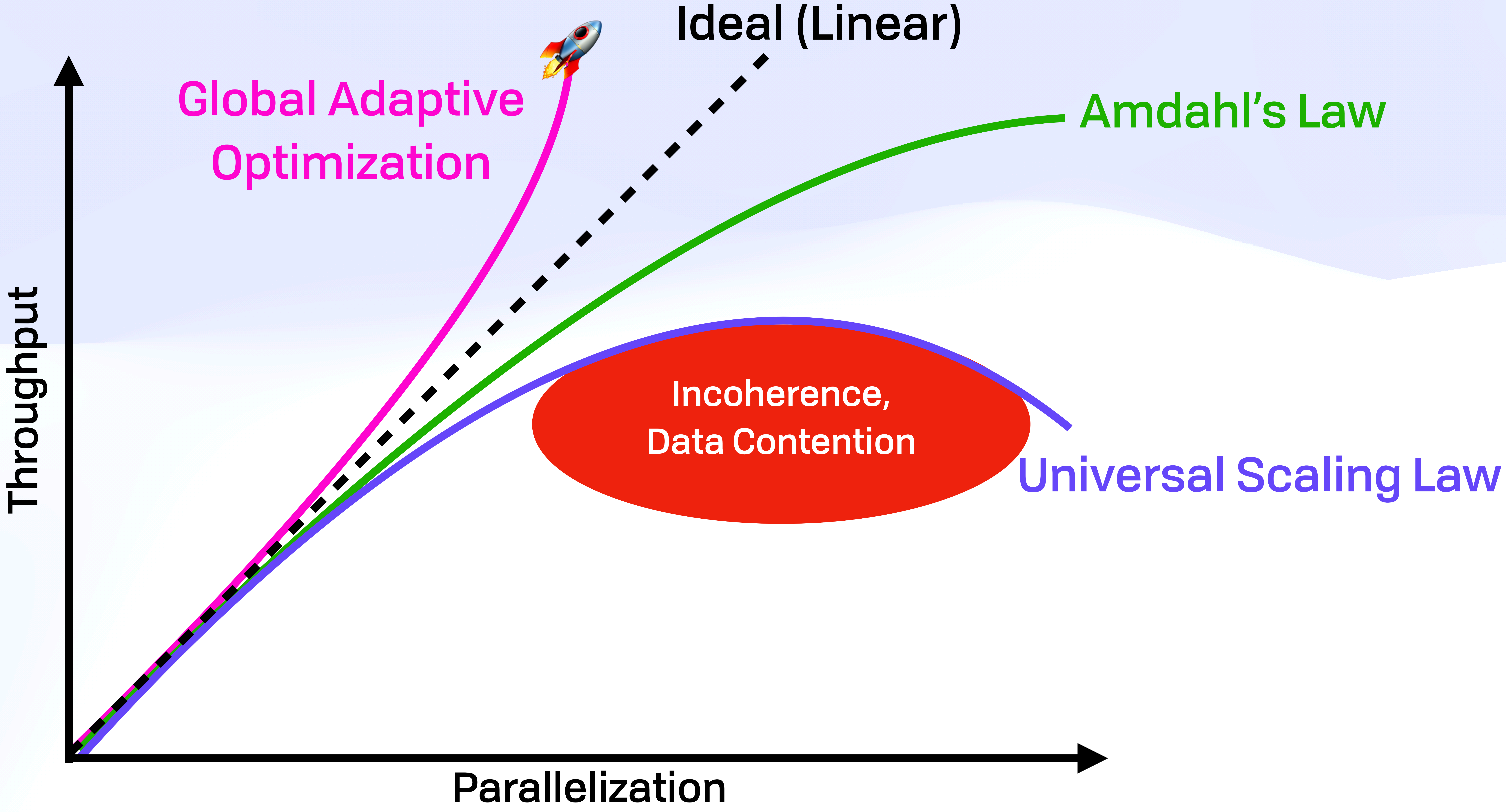
Decentralised Memoization 🧾

With a Little Scale From My Friends



Decentralised Memoization 🧾

With a Little Scale From My Friends



Decentralised Memoization 🧾

Surprise: Reverse Lookup For Free

Decentralised Memoization 🧾

Surprise: Reverse Lookup For Free

- CID → Computed Metadata

Decentralised Memoization 🧾

Surprise: Reverse Lookup For Free

- **CID → Computed Metadata**
- e.g. AI moderation classifier

Decentralised Memoization 🧾

Surprise: Reverse Lookup For Free

- **CID → Computed Metadata**
- e.g. AI moderation classifier
- e.g. Distributed token validation

The Safety Dance

🕺

The Safety Dance 🕺

The Safety Dance 🕺

"virtual resiliency",
analogous to virtual memory [...]
allows ***failure oblivious*** code to run in
a ***failure resistant*** manner

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

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 form the database community, which has shown how declarative database systems can ***completely isolate the developer from the possibility of failure***

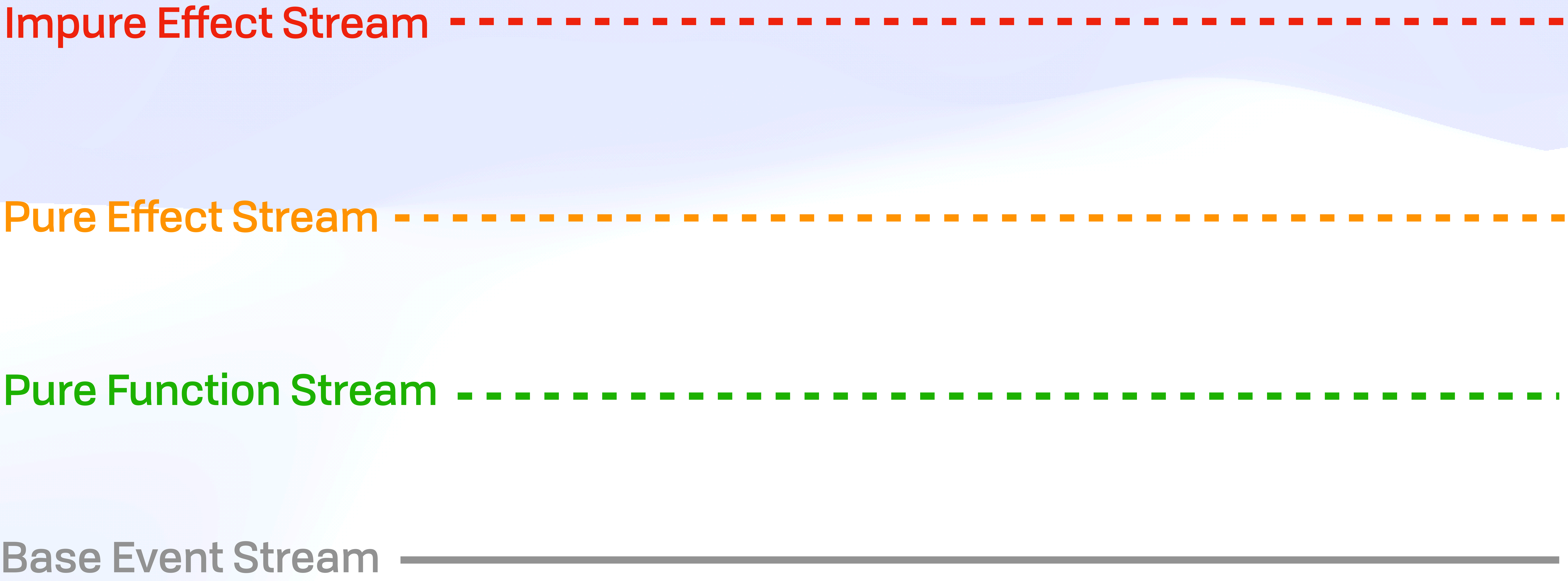
Goldstein et al, AMBROSIA: Providing Performant Virtual Resiliency for Distributed Applications

The Safety Dance 🕺

Non-Monotonicity

The Safety Dance 🕺

Non-Monotonicity



Impure Effect Stream

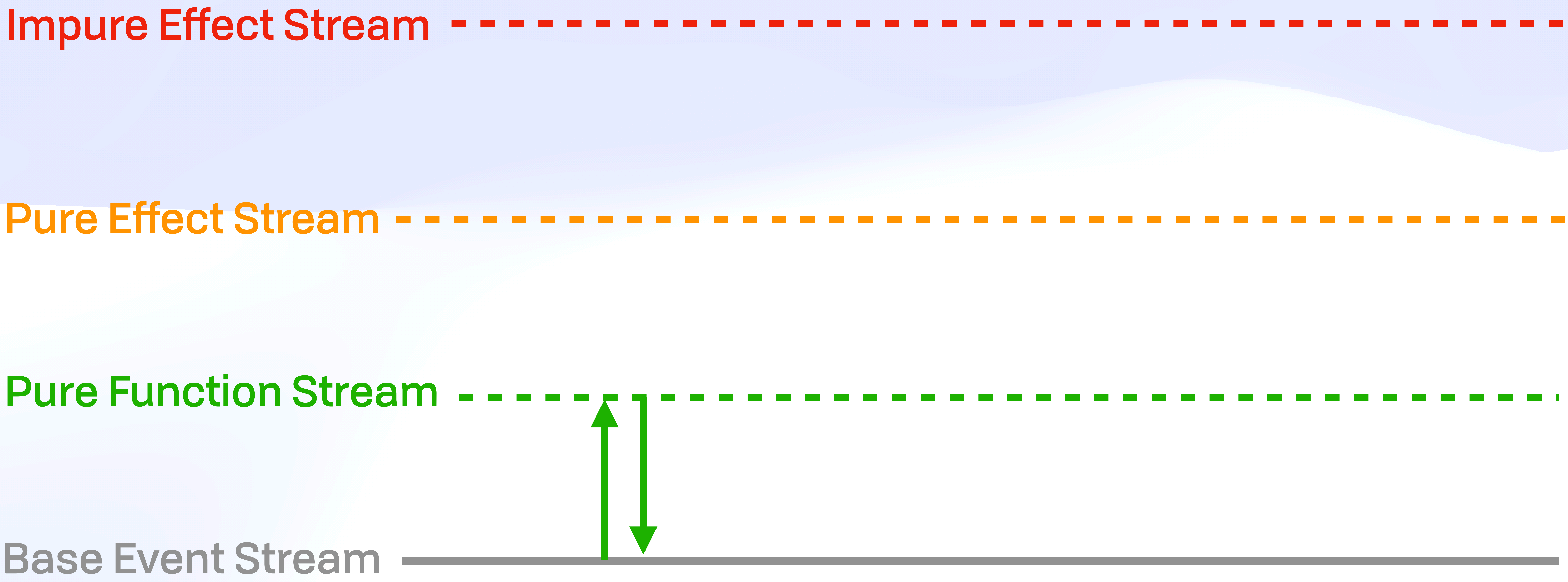
Pure Effect Stream

Pure Function Stream

Base Event Stream

The Safety Dance 🕺

Non-Monotonicity



Impure Effect Stream

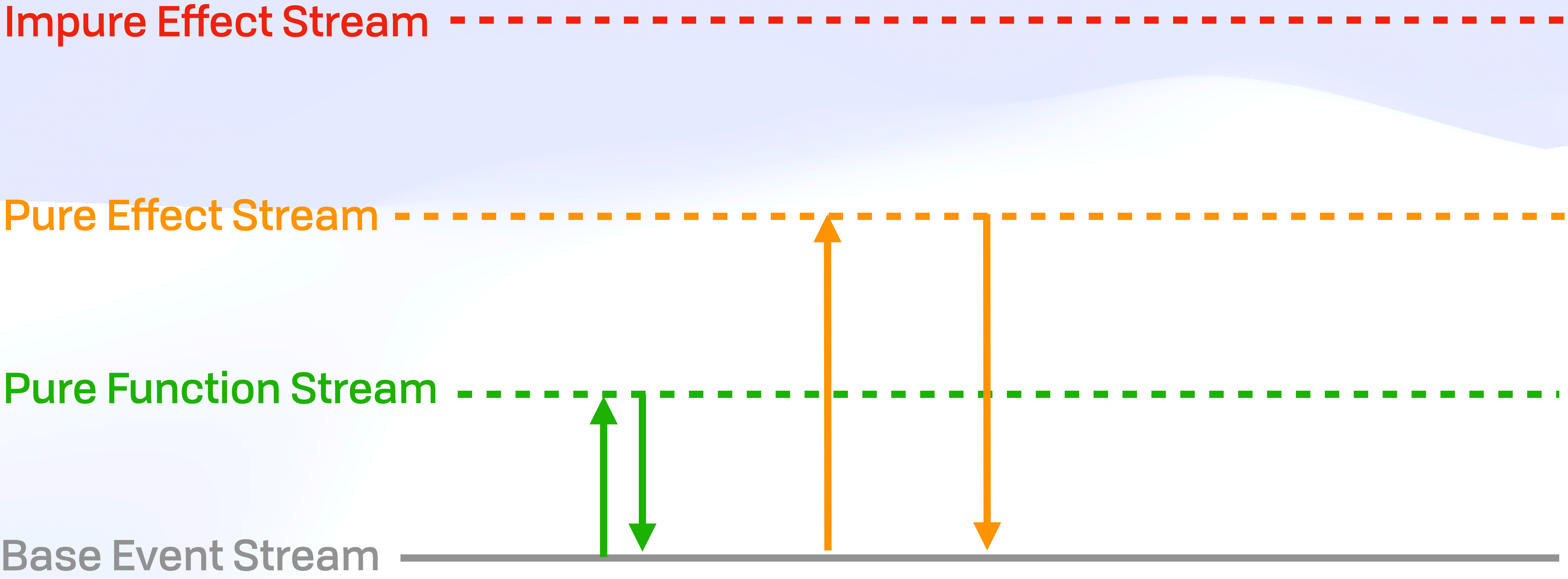
Pure Effect Stream

Pure Function Stream

Base Event Stream

The Safety Dance 🕺

Non-Monotonicity


Impure Effect Stream

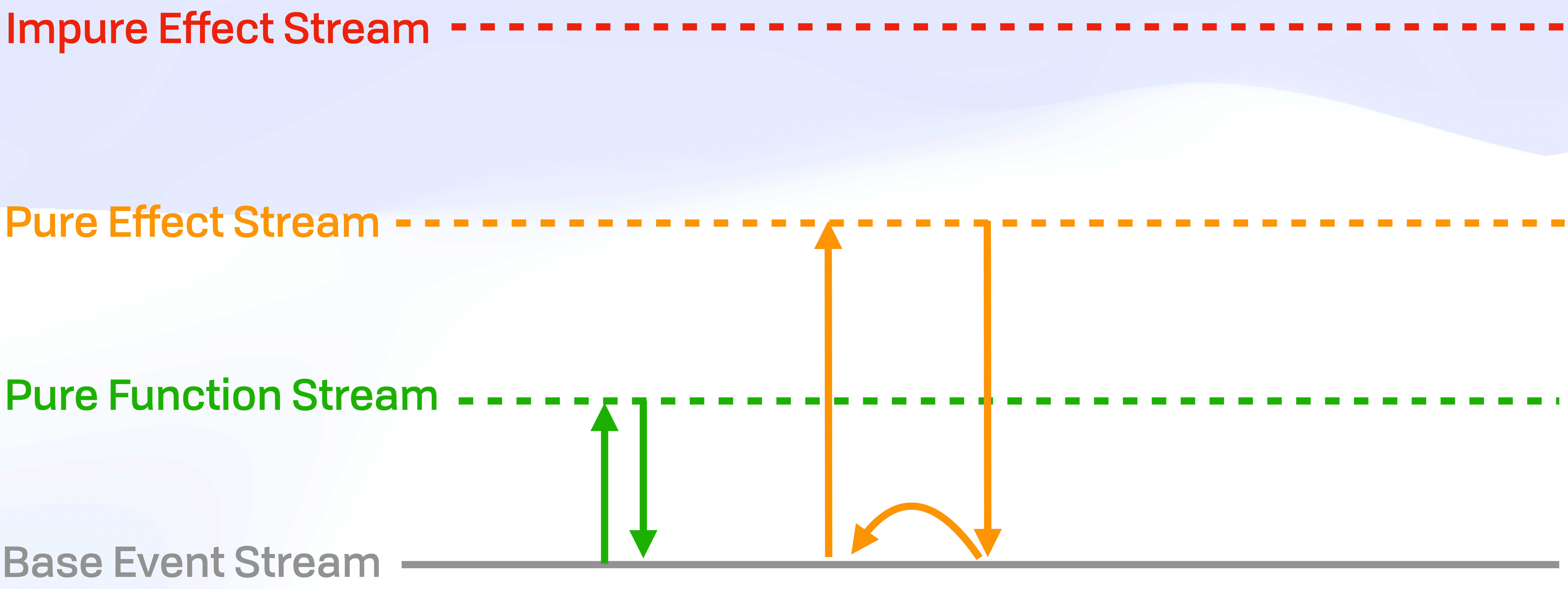
Pure Effect Stream

Pure Function Stream

Base Event Stream

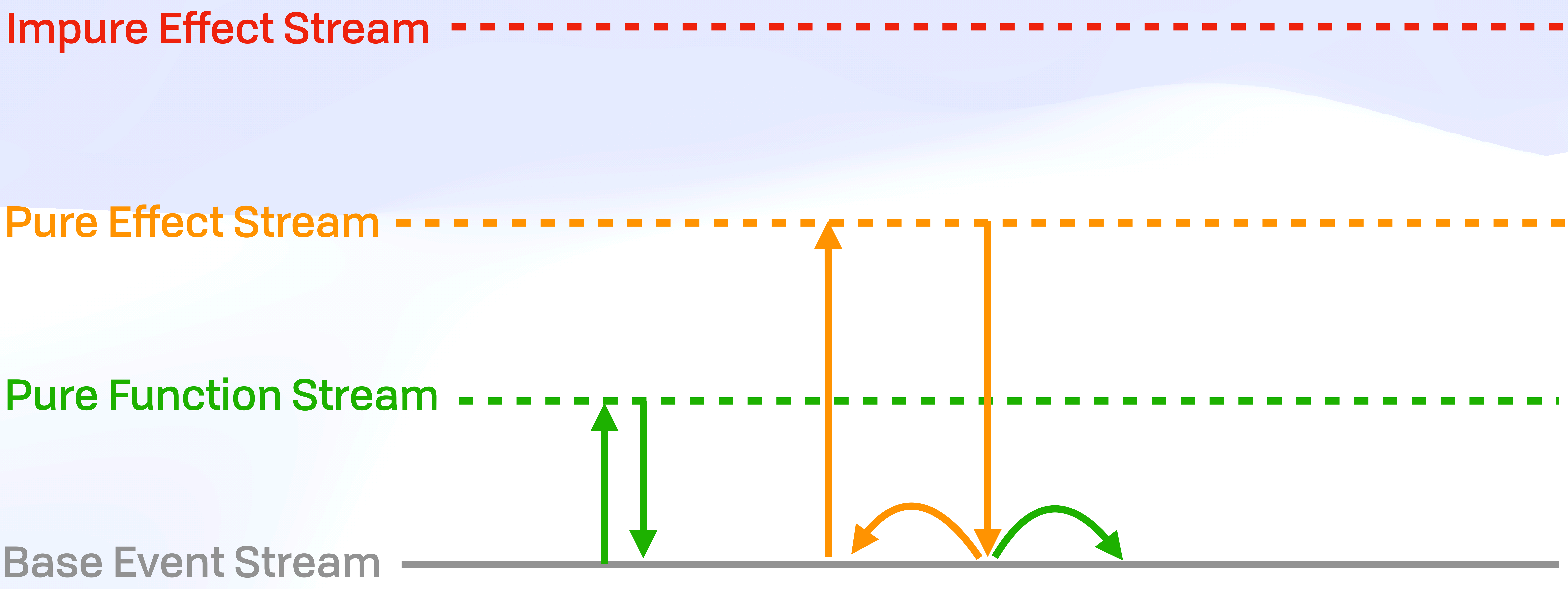
The Safety Dance 🕺

Non-Monotonicity



The Safety Dance 🕺

Non-Monotonicity



Impure Effect Stream

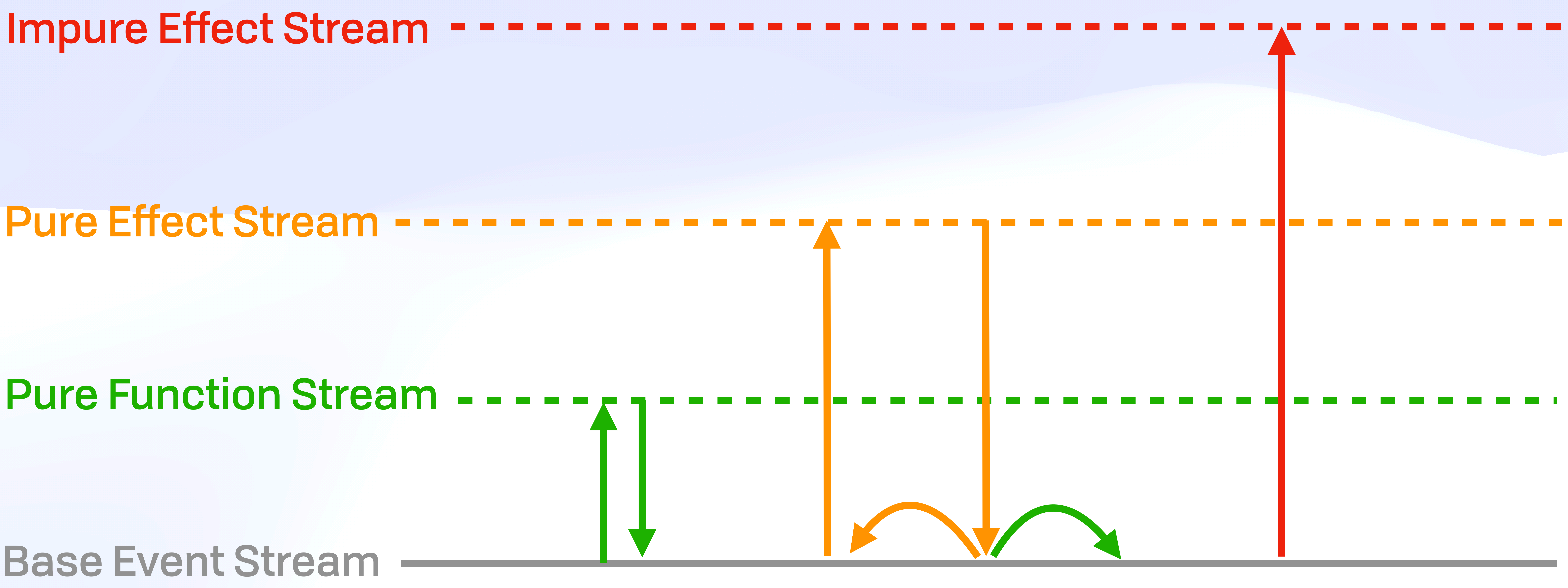
Pure Effect Stream

Pure Function Stream

Base Event Stream

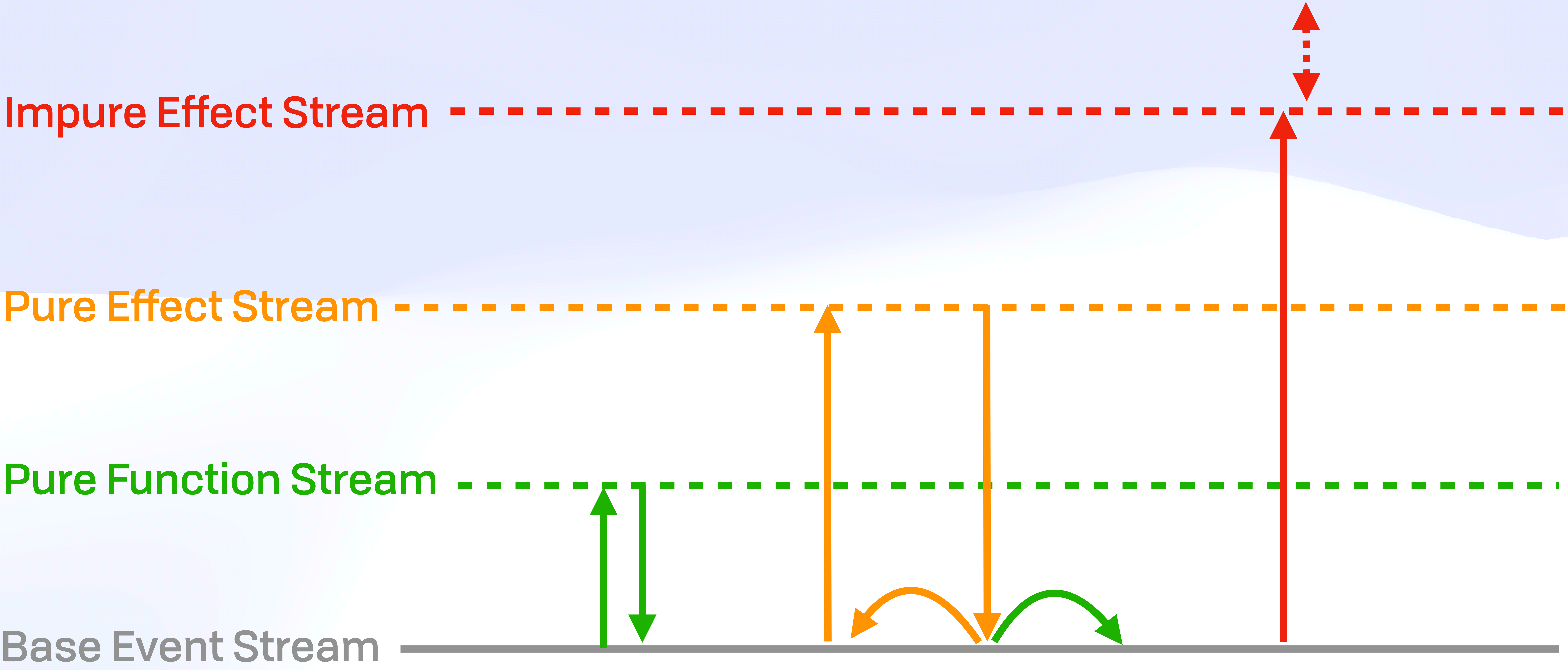
The Safety Dance 🕺

Non-Monotonicity



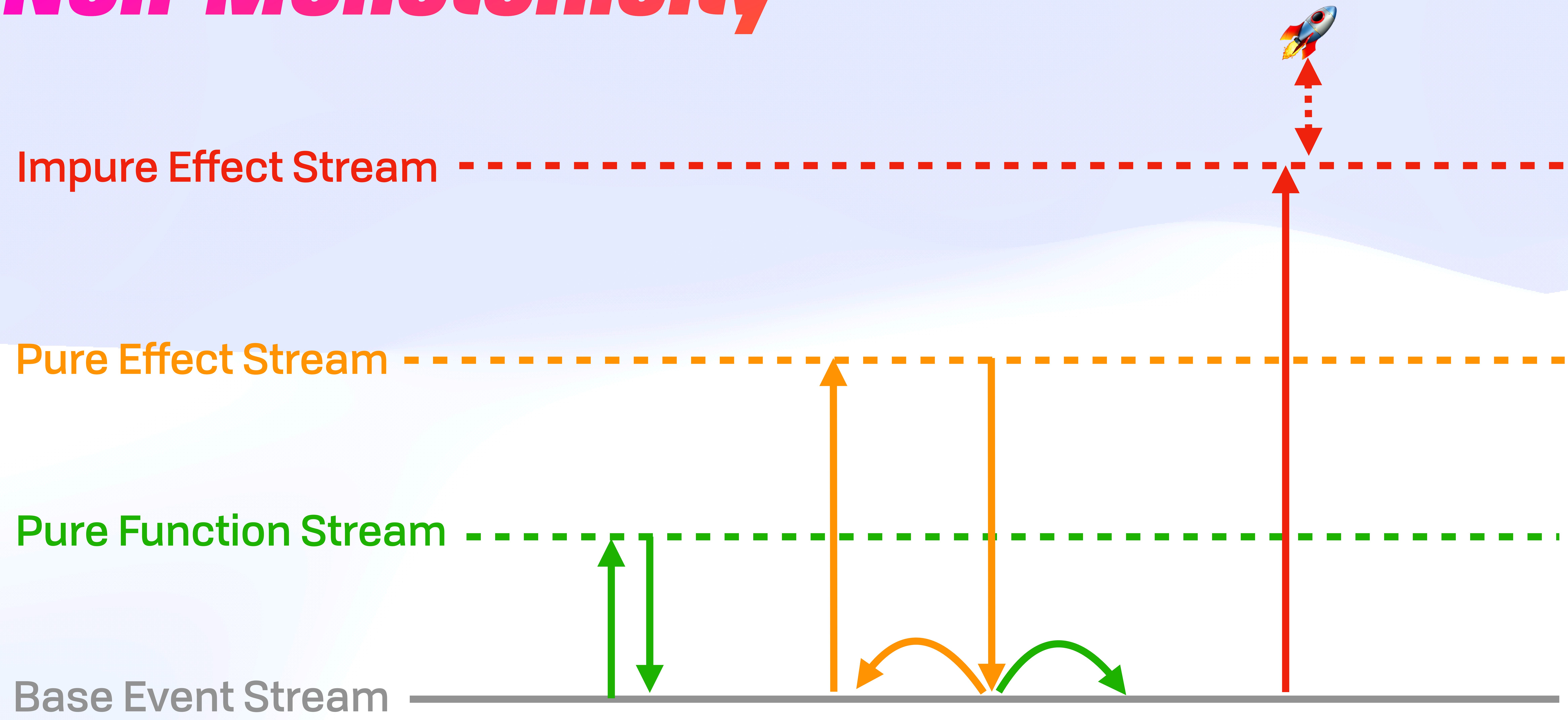
The Safety Dance 🕺

Non-Monotonicity



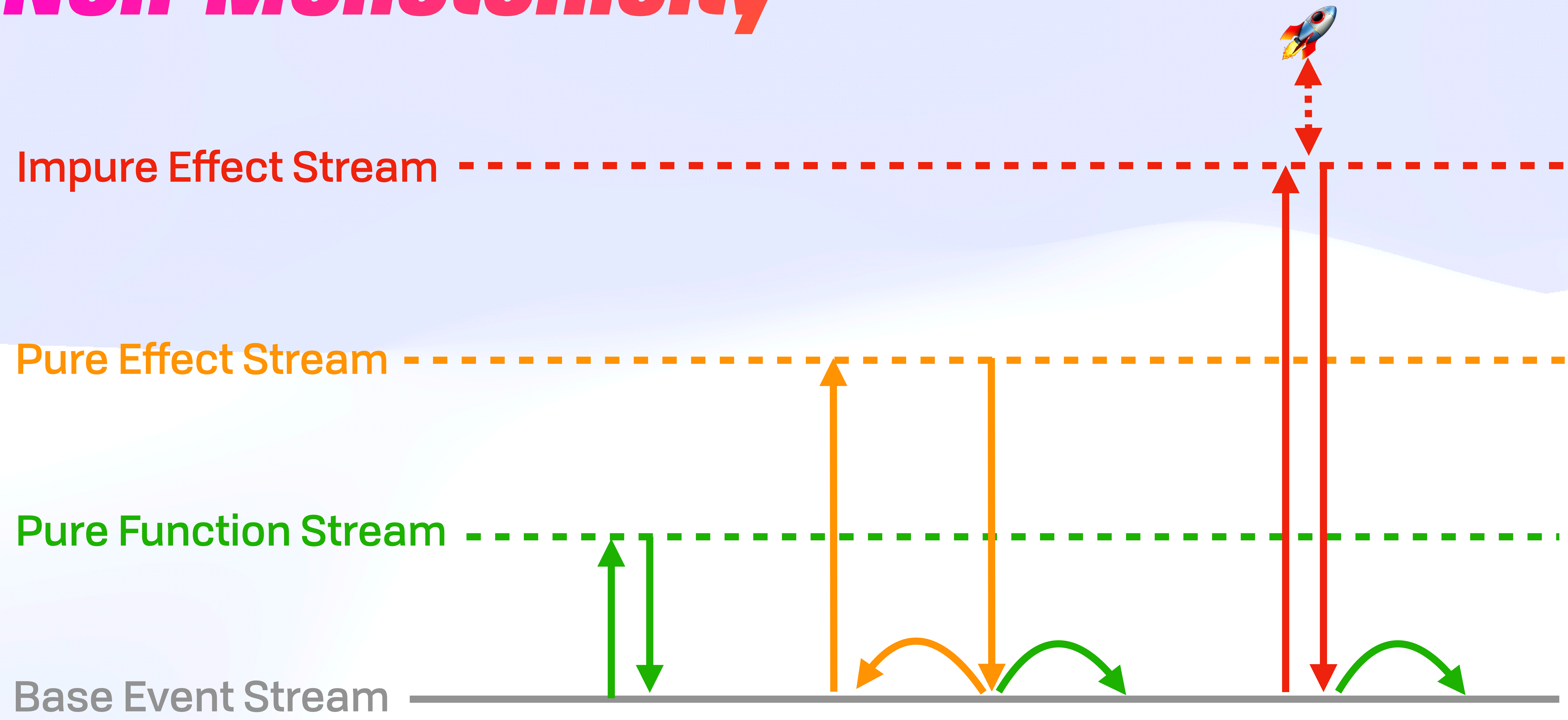
The Safety Dance 🕺

Non-Monotonicity



The Safety Dance 🕺

Non-Monotonicity



The Safety Dance 🕺

Virtual Resiliency

The Safety Dance 🕺

Virtual Resiliency

Mutation 🦋

Idempotent 🔂

Deterministic 📅

The Safety Dance 🕺

Virtual Resiliency

Query A

Query B

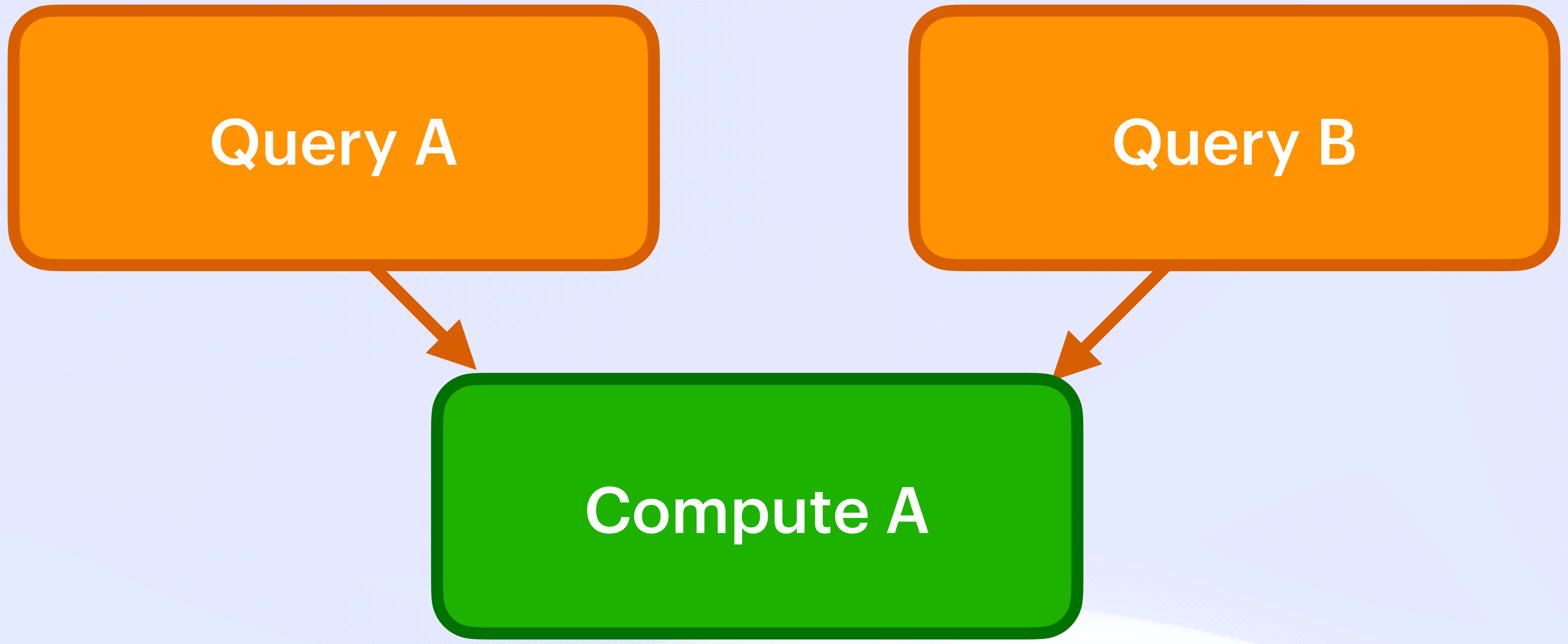
Mutation 🦋

Idempotent 🔂

Deterministic 📅

The Safety Dance 🕺

Virtual Resiliency



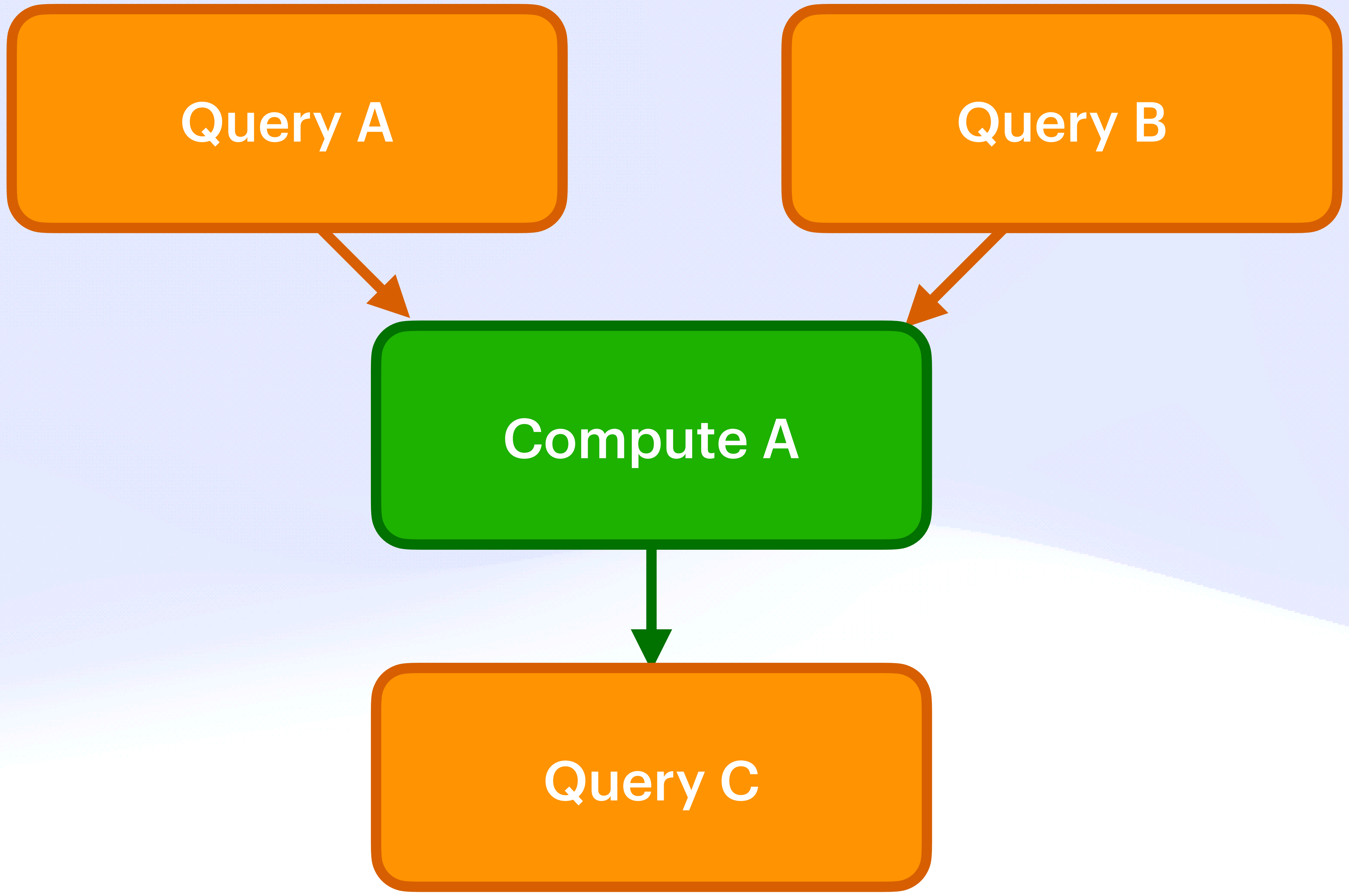
Mutation 🦋

Idempotent 🔂

Deterministic 📅

The Safety Dance 🕺

Virtual Resiliency



- Mutation 🦋
- Idempotent 🔂
- Deterministic 📅

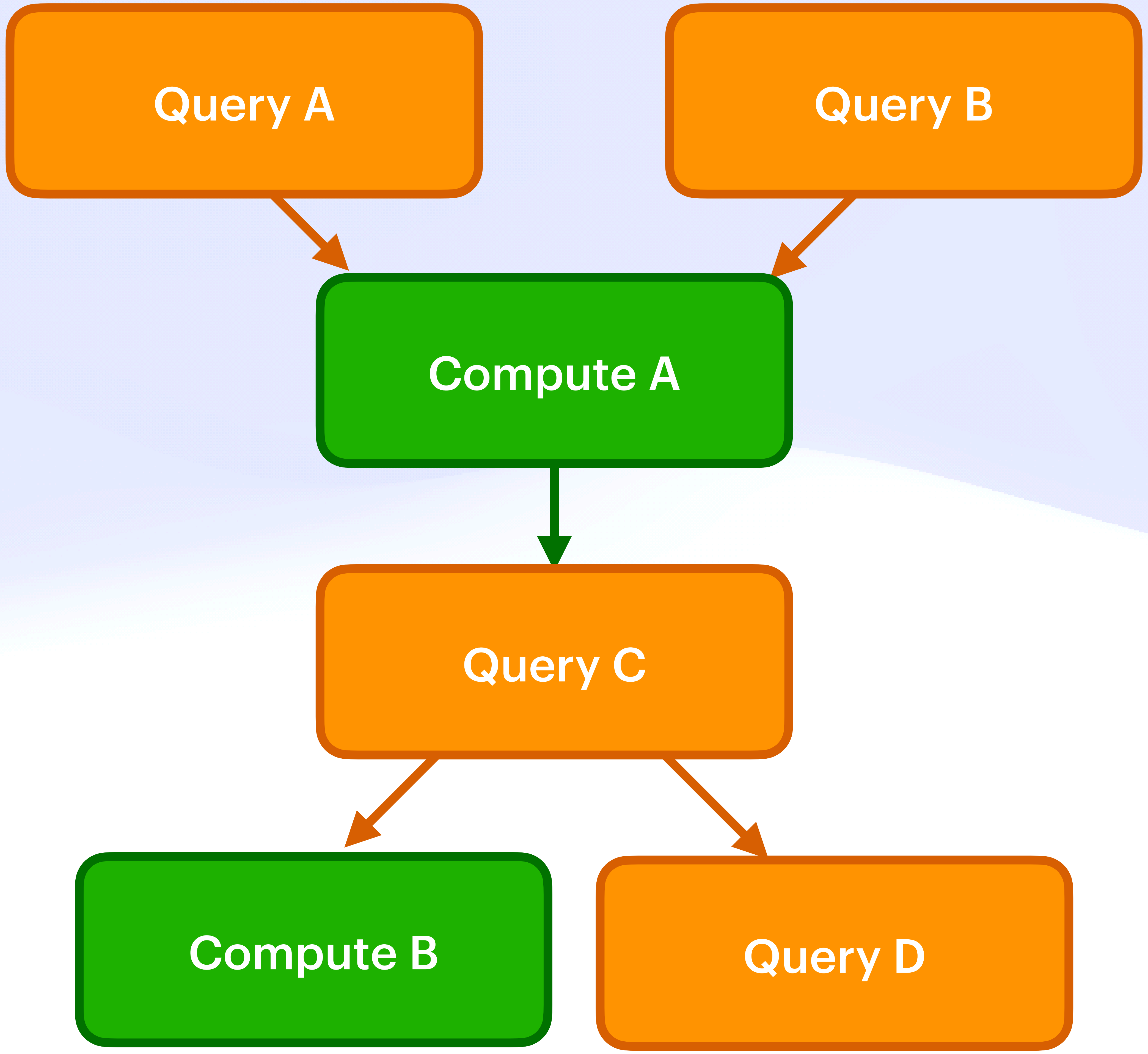
The Safety Dance 🕺

Virtual Resiliency

Mutation 🦋

Idempotent 🔂

Deterministic 📅



Query A

Query B

Compute A

Query C

Compute B

Query D

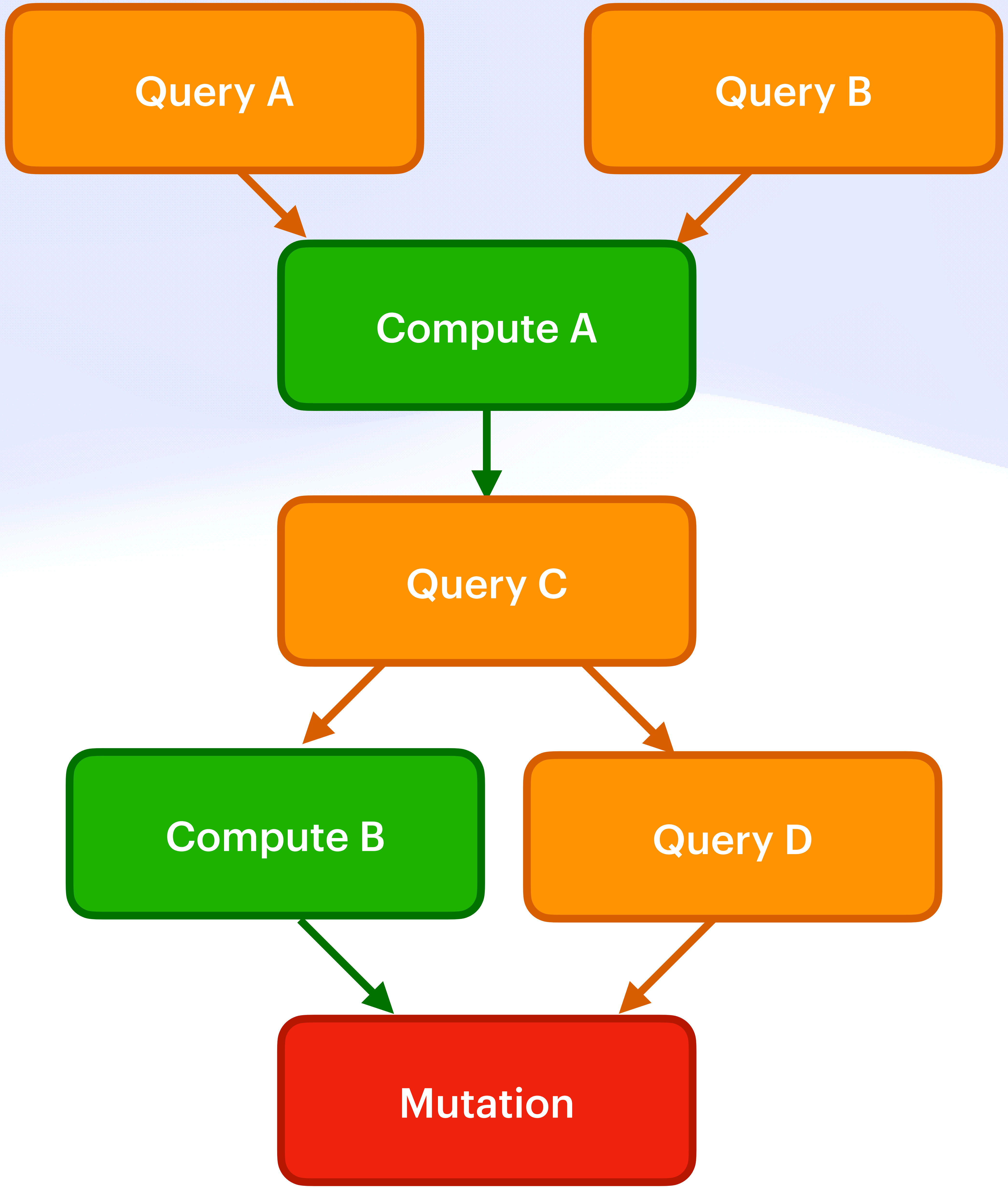
The Safety Dance 🕺

Virtual Resiliency

Mutation 🦋

Idempotent 🔂

Deterministic 📅



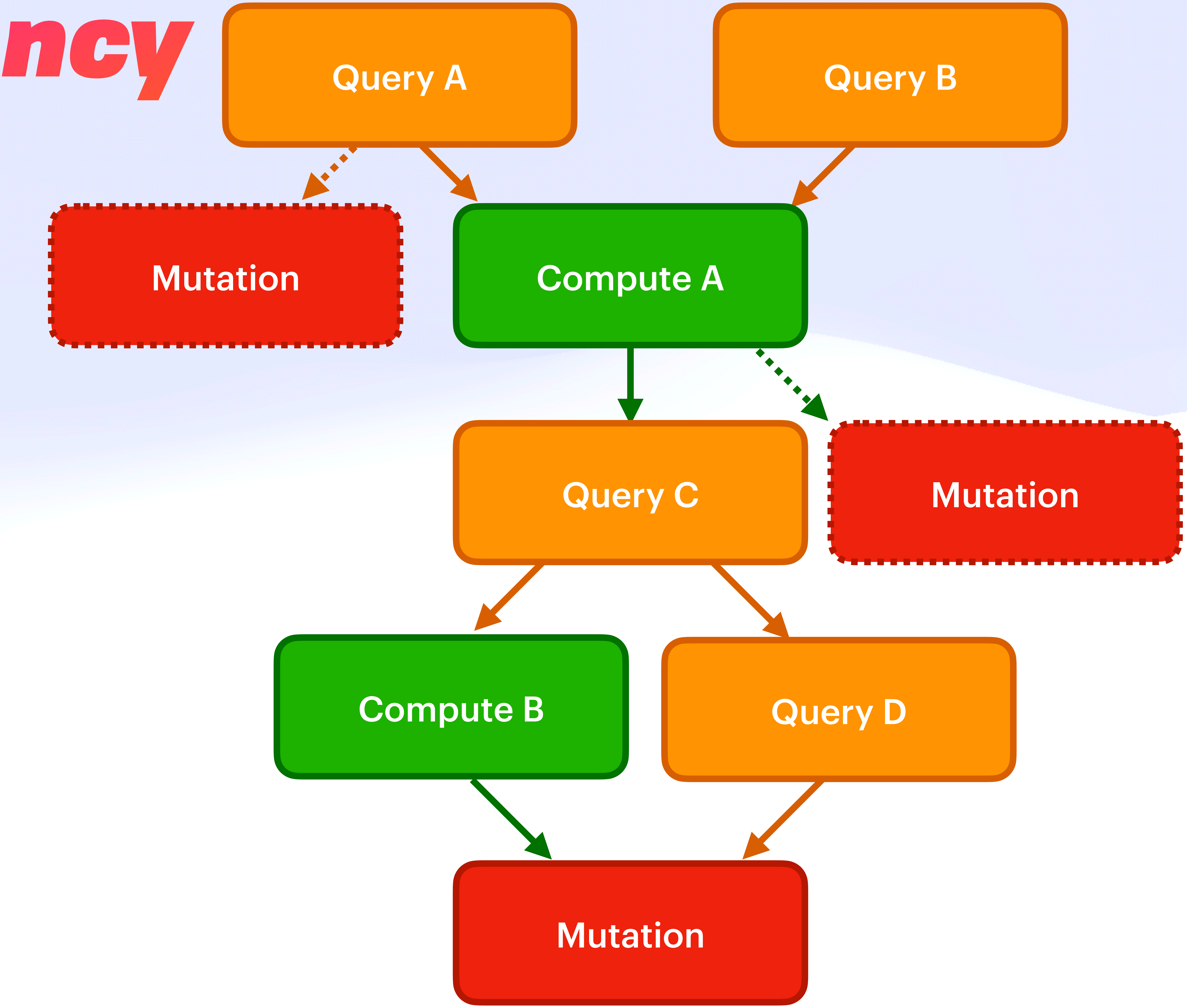
The Safety Dance 🕺

Virtual Resiliency

Mutation 🦋

Idempotent 🔂

Deterministic 📅



The Safety Dance 🕺

Simplified Safe Layout

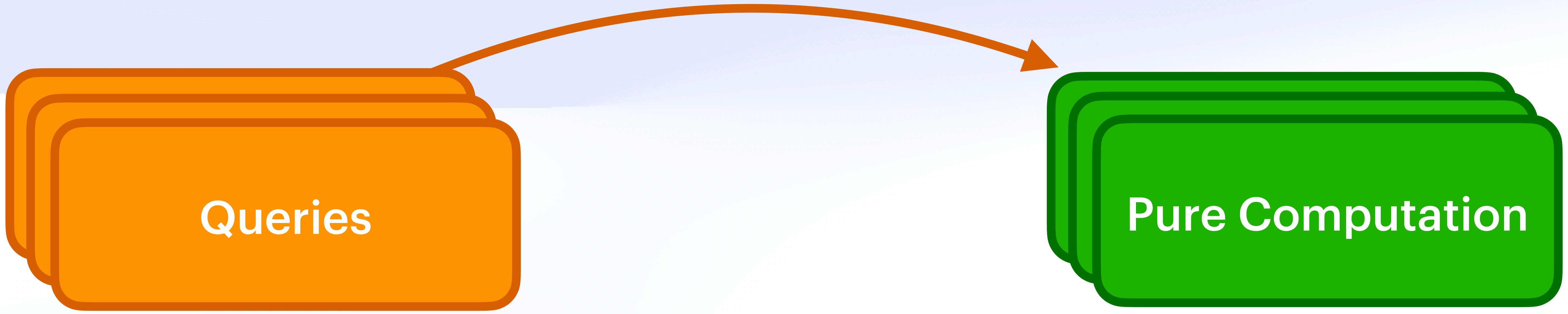
The Safety Dance 🕺

Simplified Safe Layout

Queries

The Safety Dance 🕺

Simplified Safe Layout

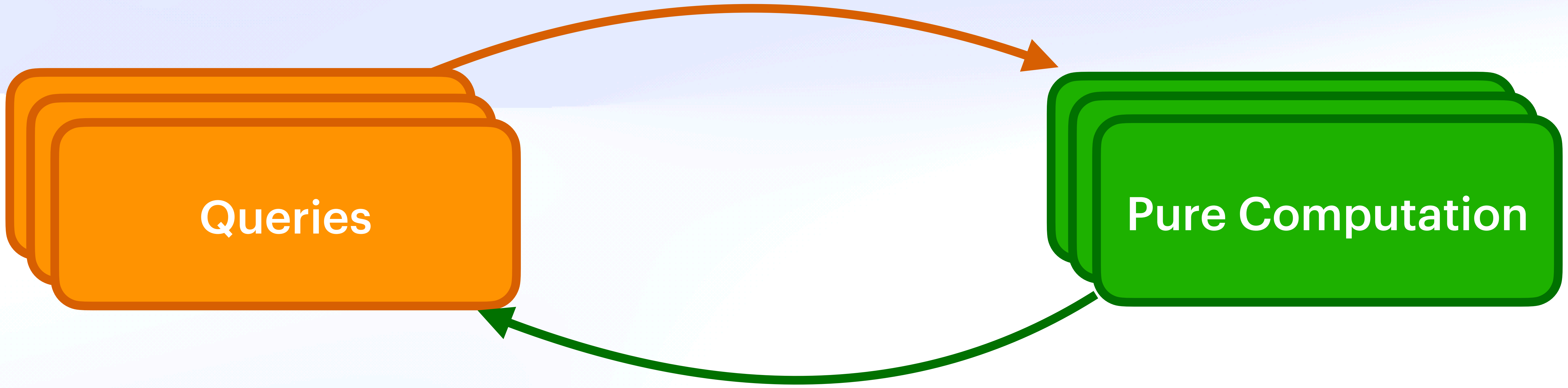


Queries

Pure Computation

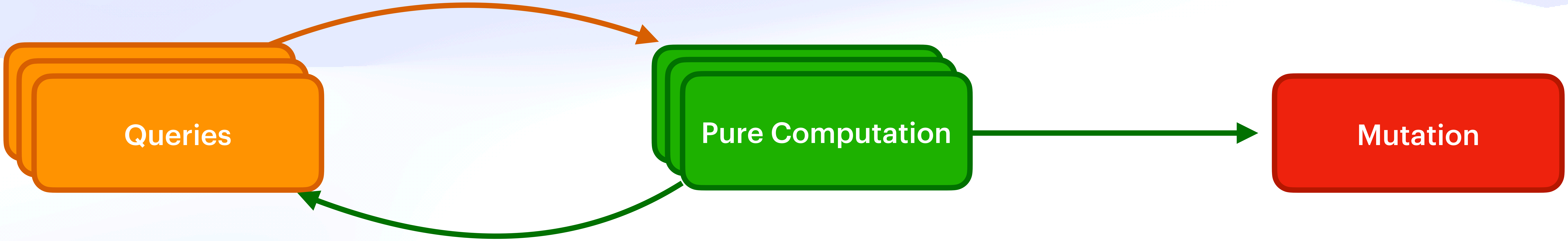
The Safety Dance 🕺

Simplified Safe Layout



The Safety Dance 🕺

Simplified Safe Layout



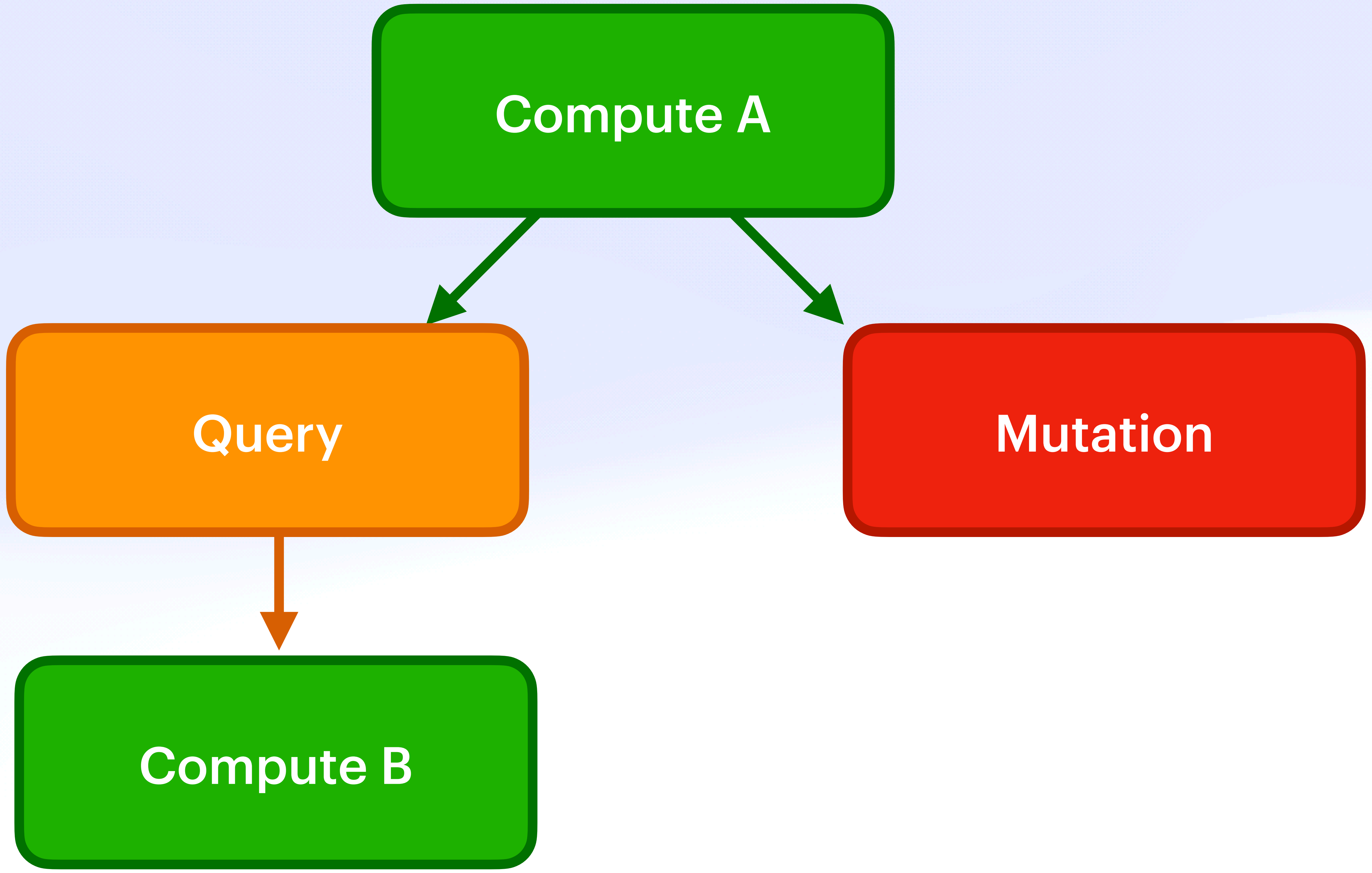
Queries

Pure Computation

Mutation

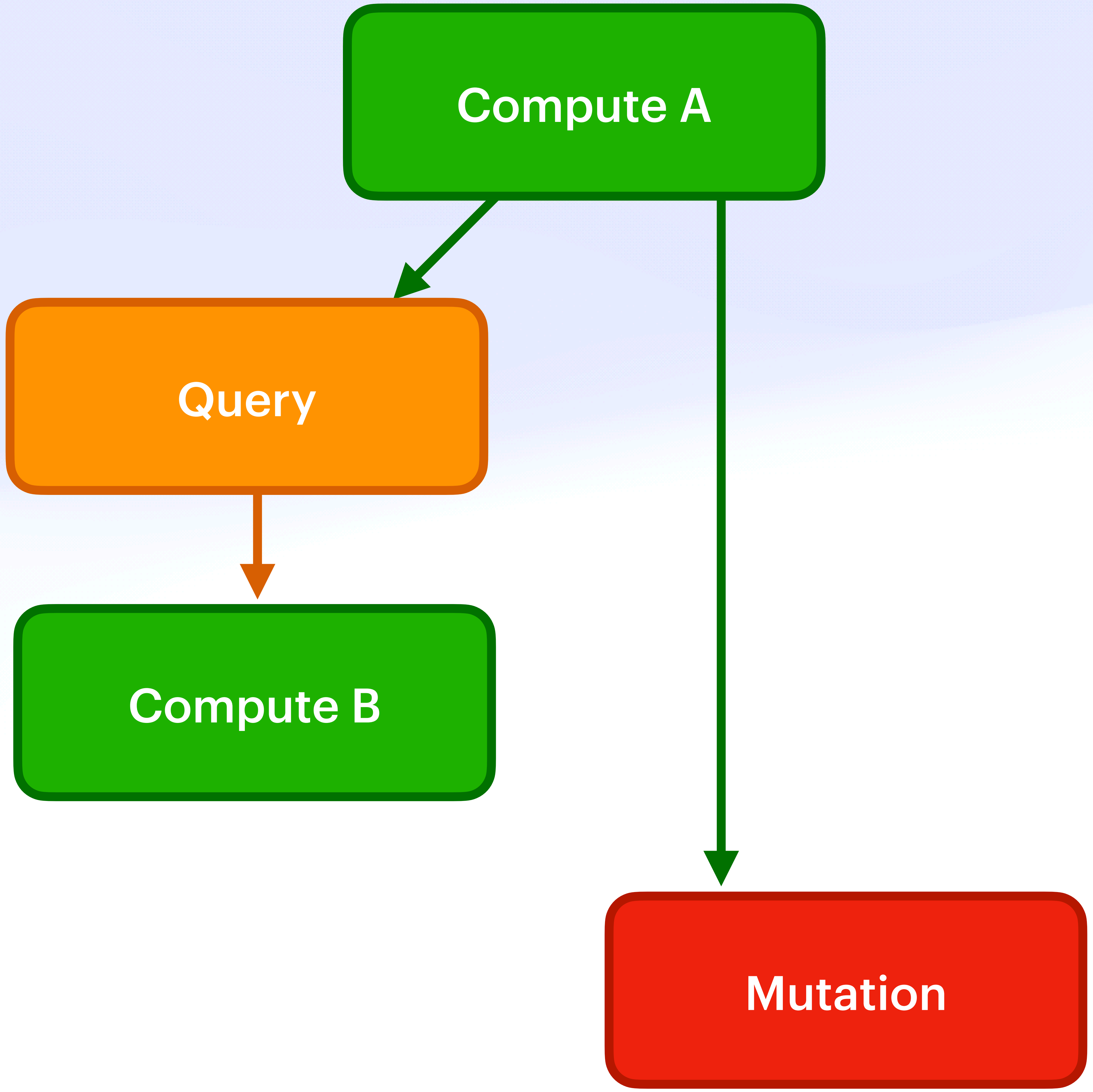
The Safety Dance 🕺

Simple Example



The Safety Dance 🕺

Simple Example



The Safety Dance 🕺

From CID to CHa 🍵

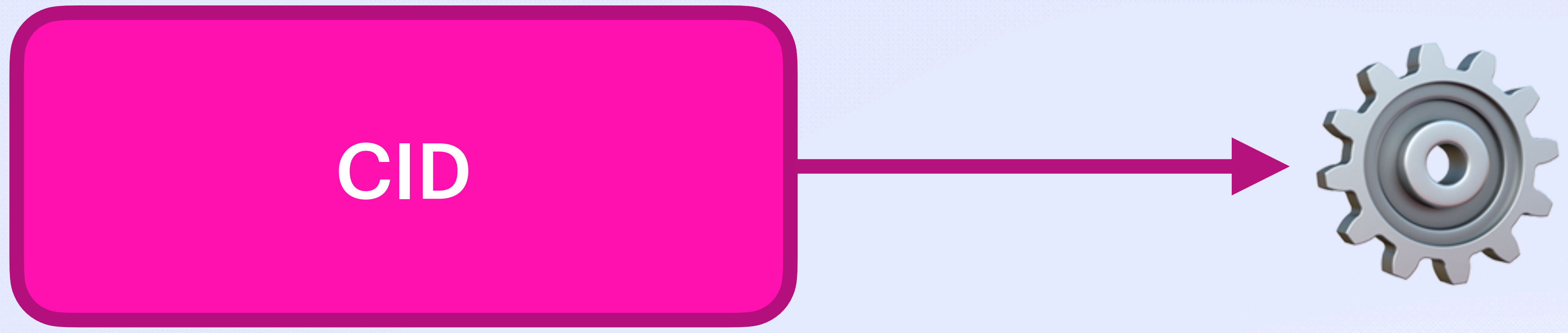
The Safety Dance 🕺

From CID to CHa 🍵

CID

The Safety Dance 🕺

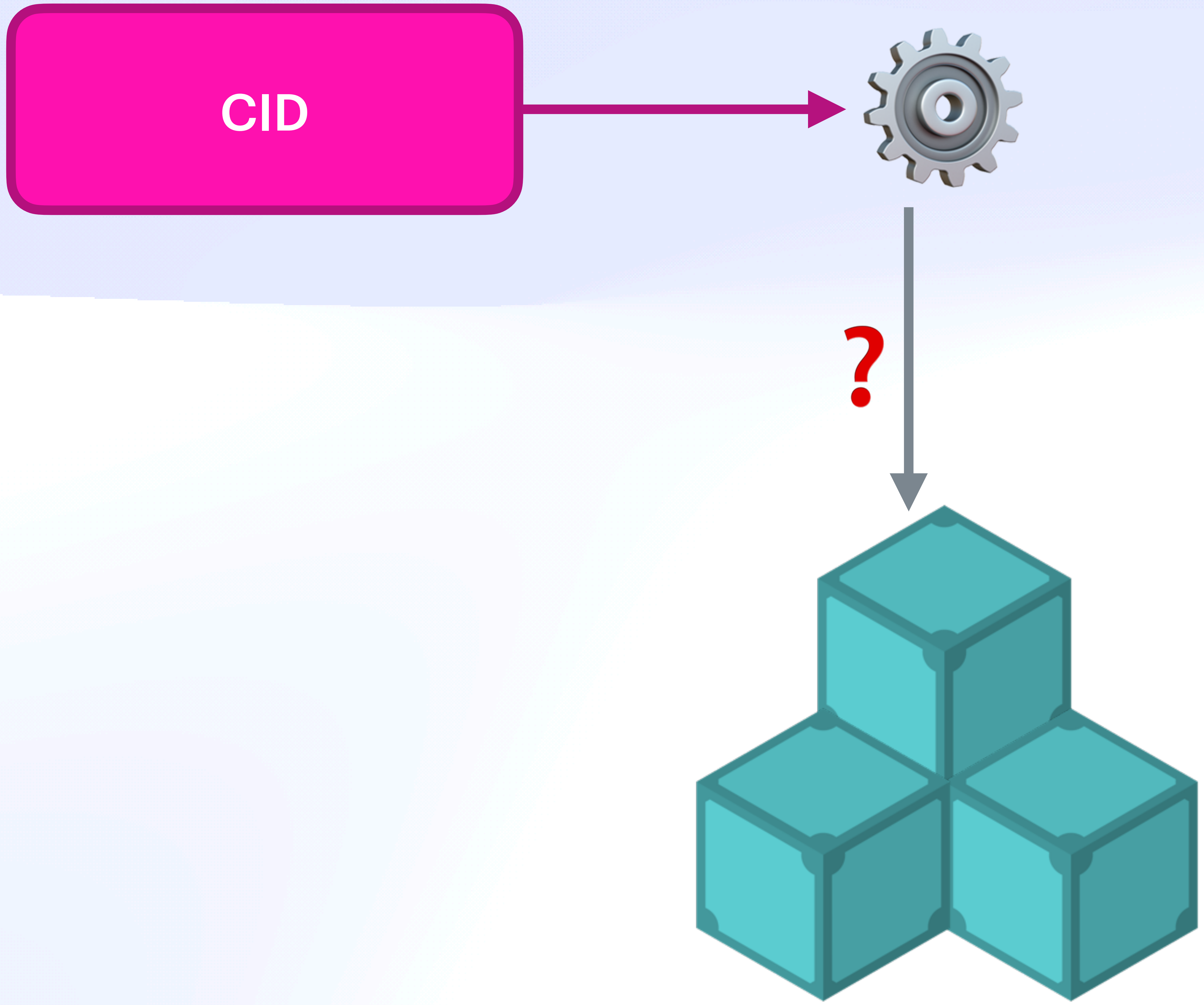
From CID to CHa 🍵



CID

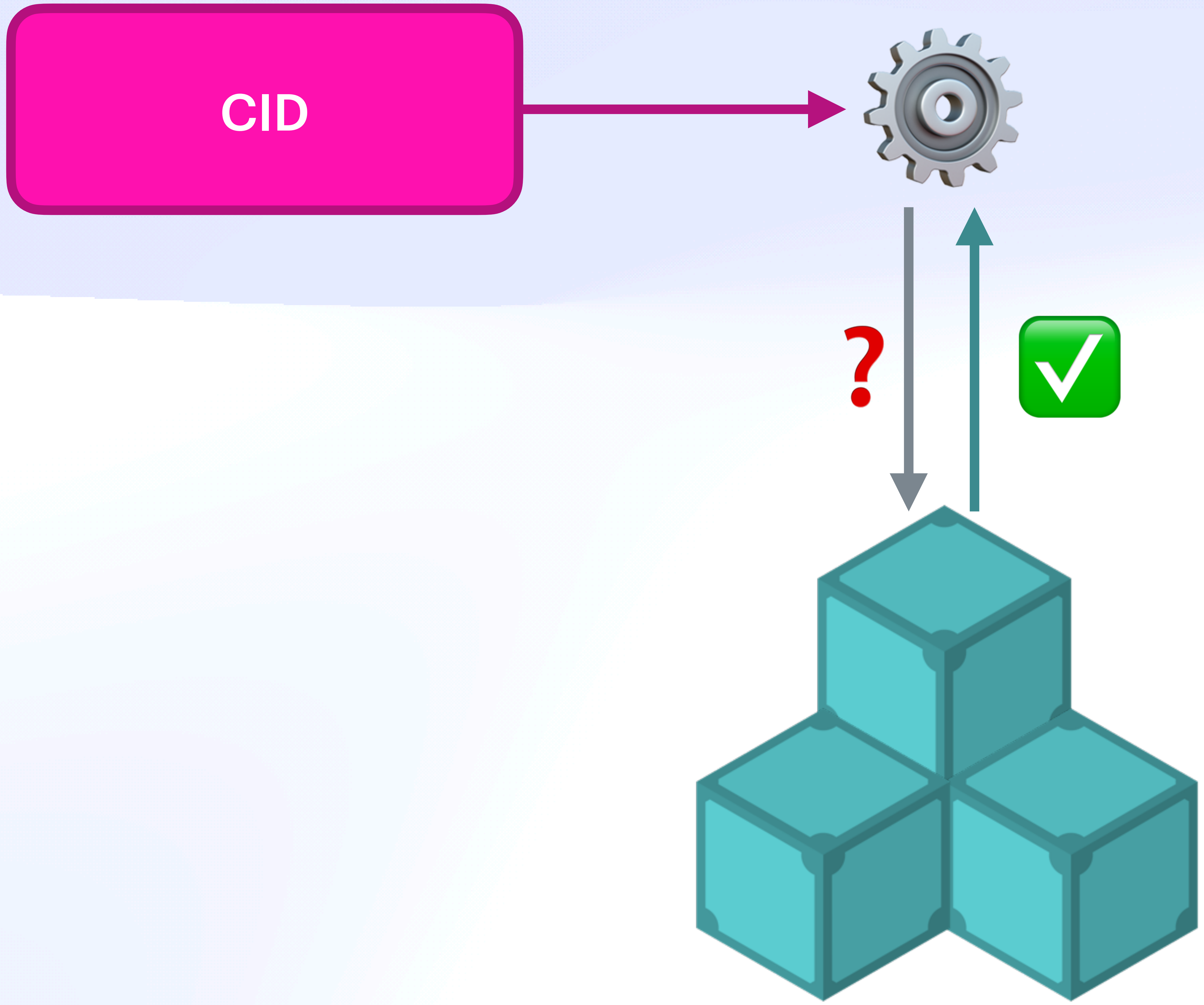
The Safety Dance 🕺

From CID to CHa 🍵



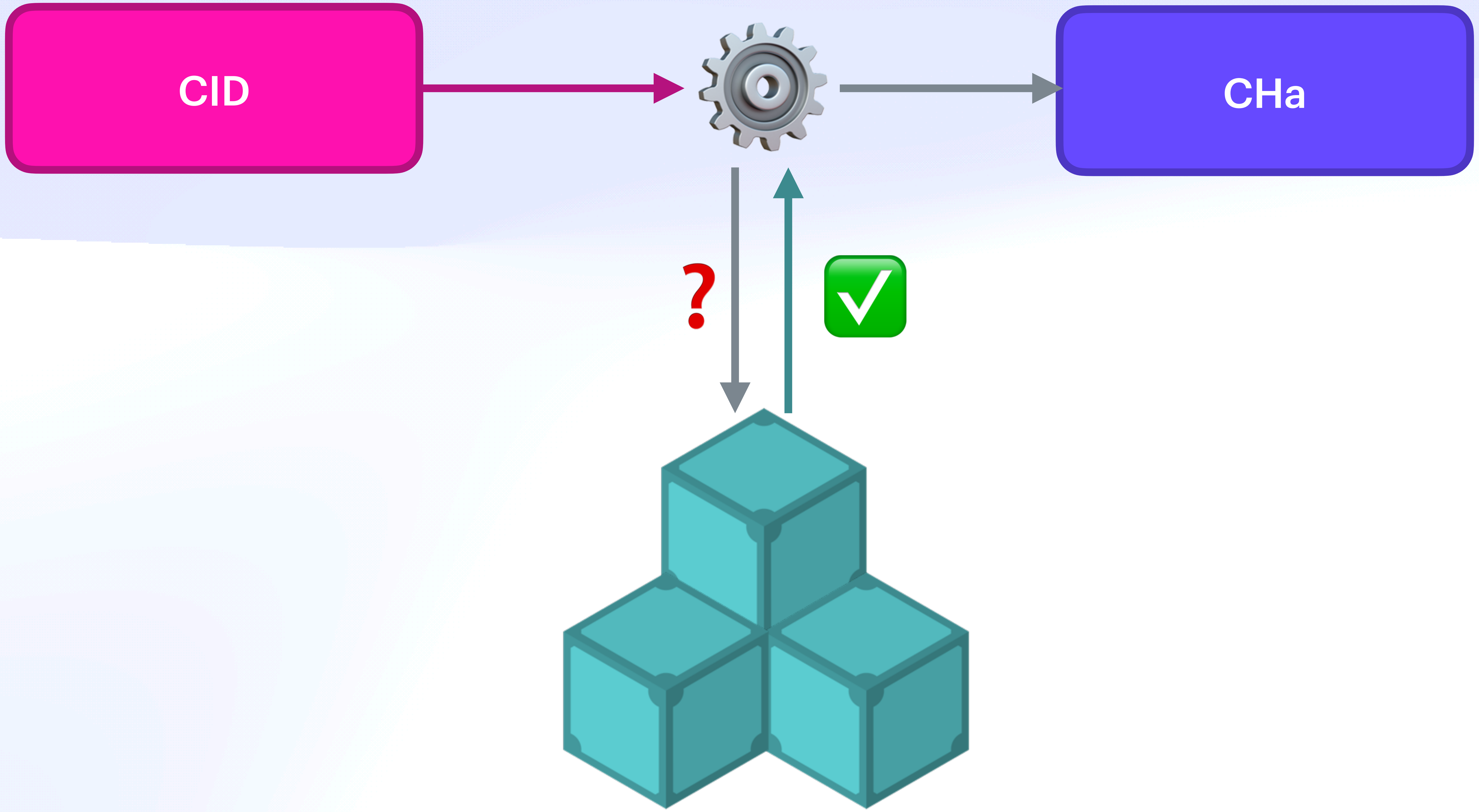
The Safety Dance 🕺

From CID to CHa 🍵



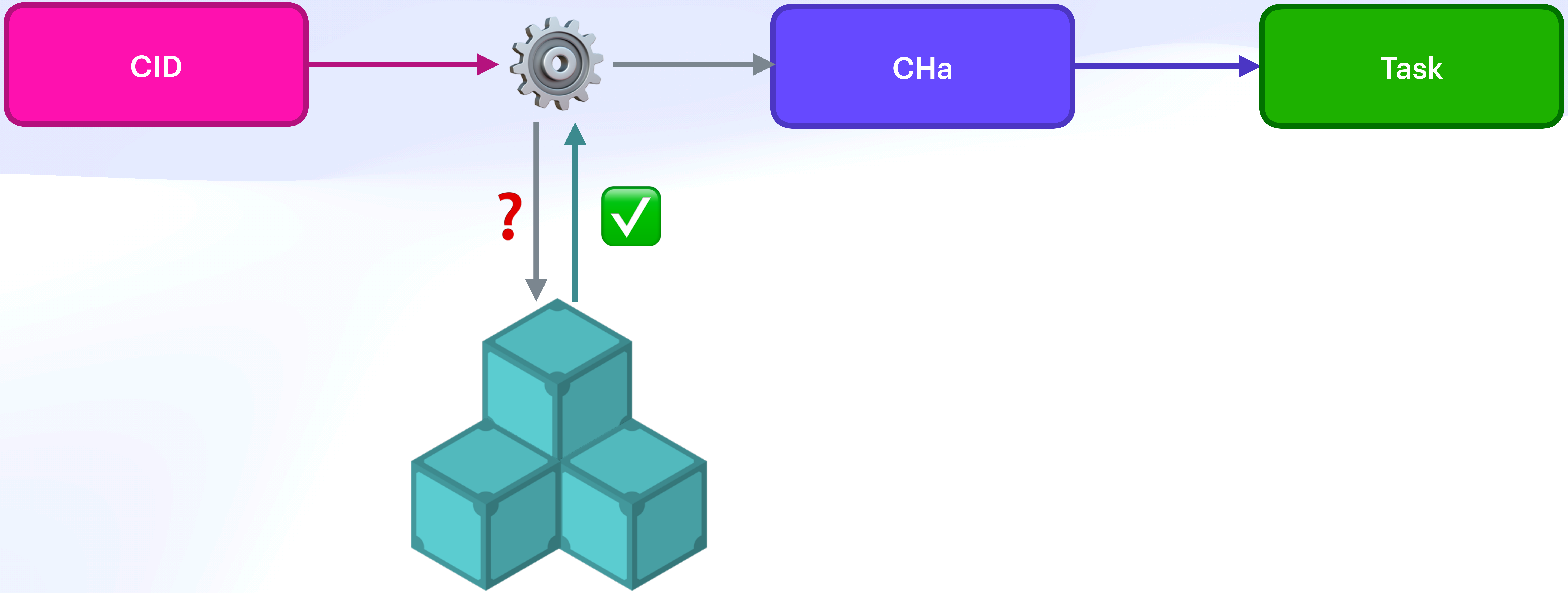
The Safety Dance 🕺

From CID to CHa 🍵



The Safety Dance 🕺

From CID to CHa 🍵

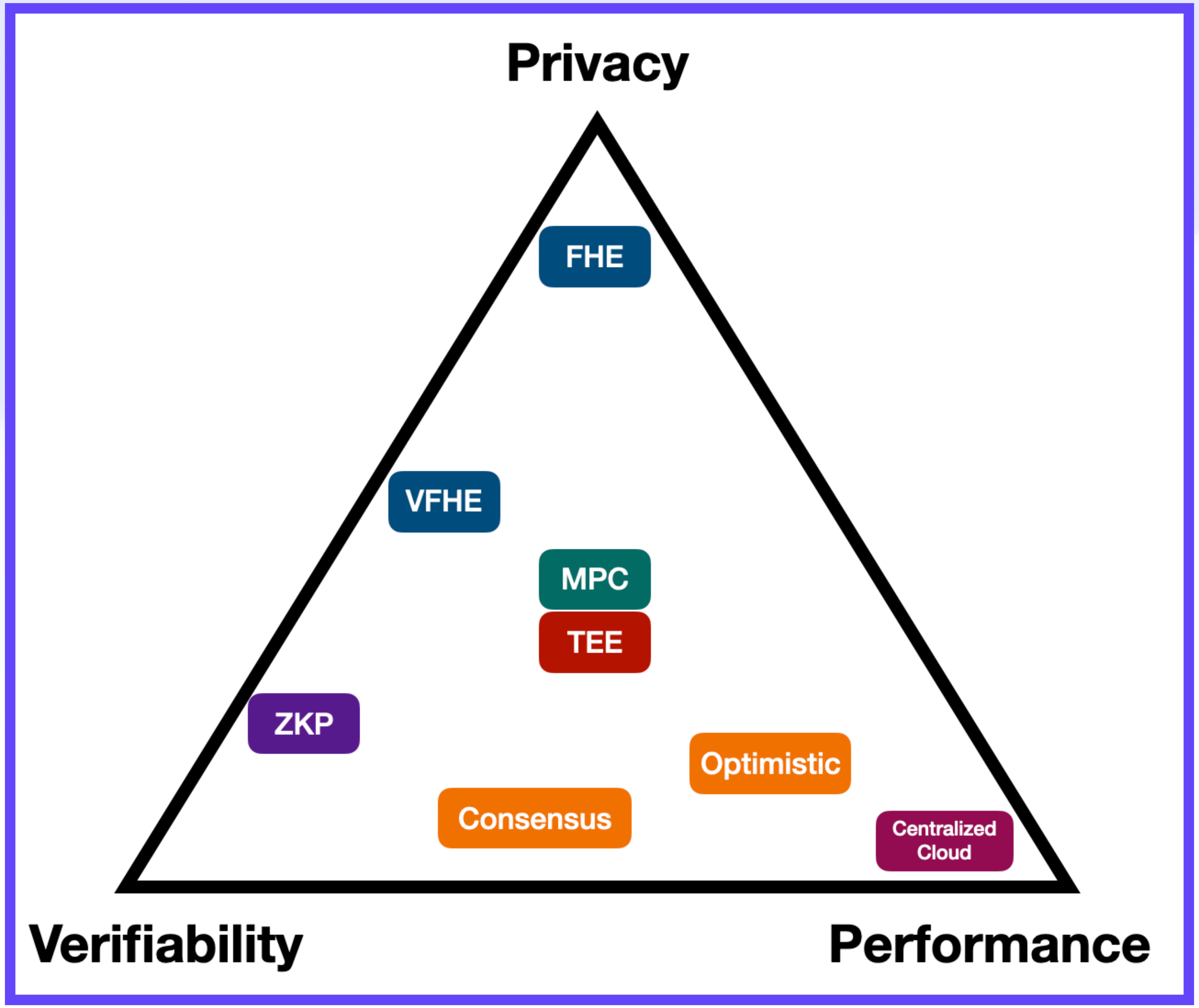


"Curated" Future & Todos

🔮

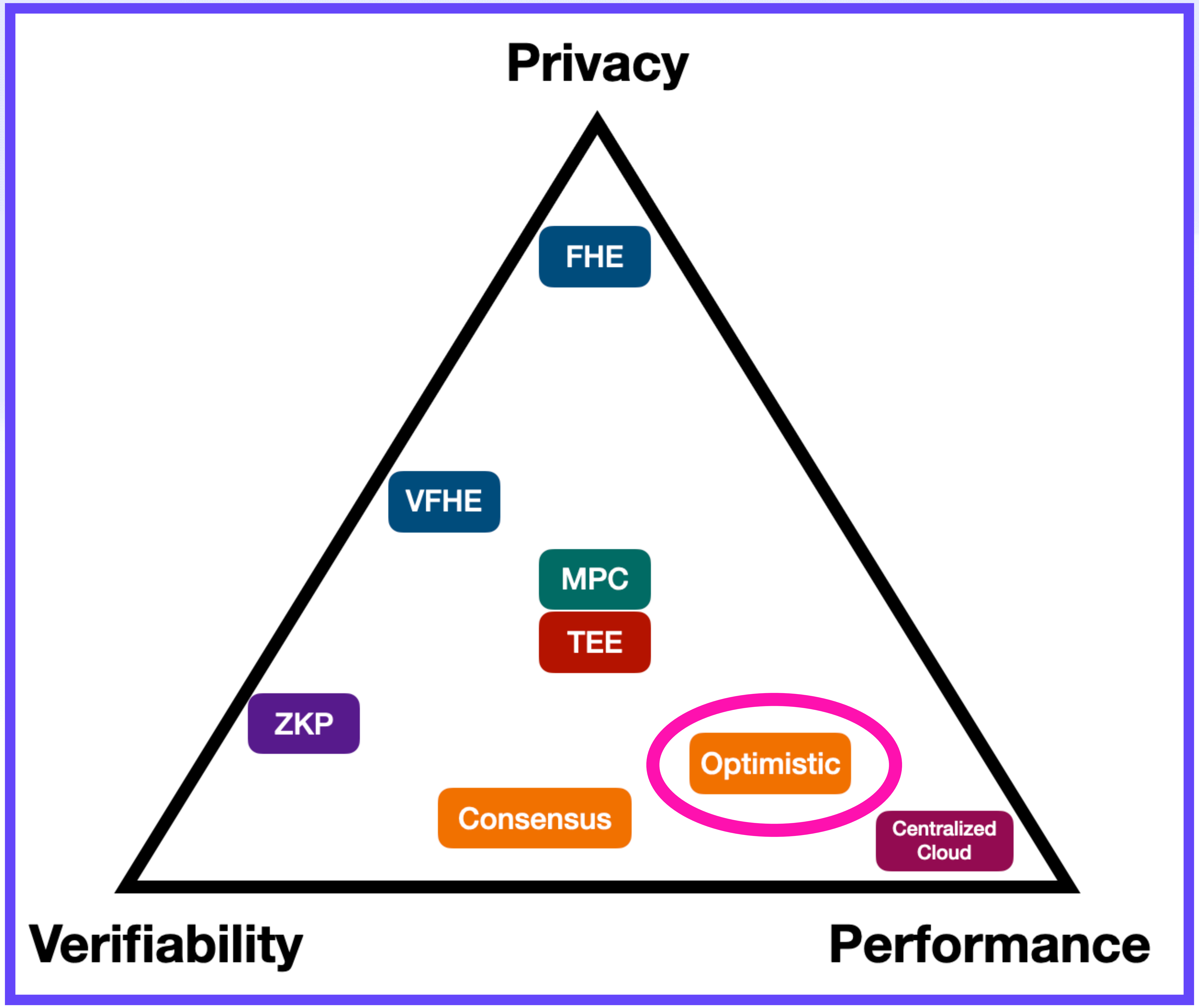
Requirements

On Deck: Optimistic Verification



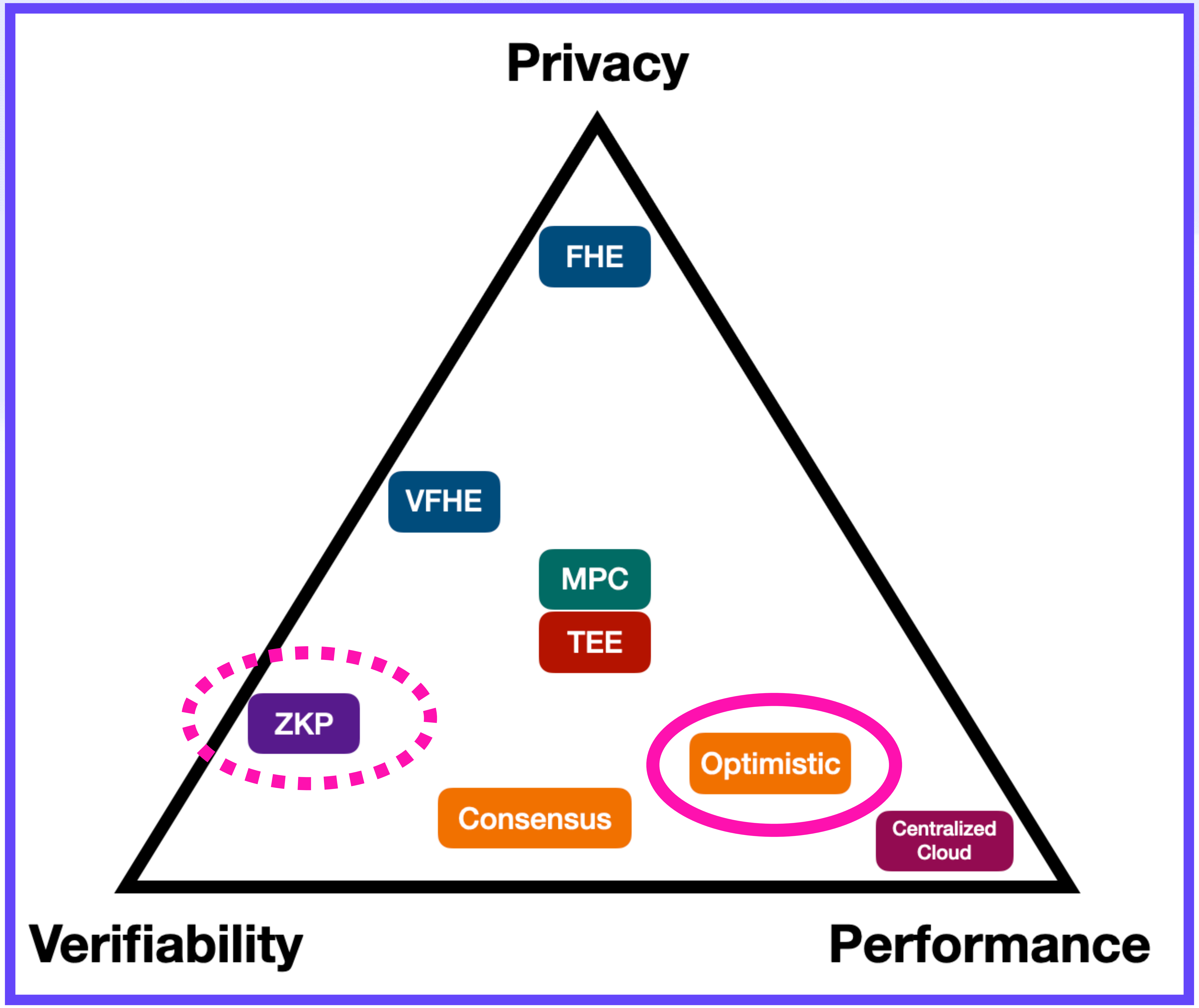
Requirements

On Deck: Optimistic Verification



Requirements

On Deck: Optimistic Verification



UCAN Decentralize Auth

"IPFS Run"

```
»  ipfs run bafkreigpbimktgowom47jv7frt3xvhb7ati4upgguykyn2cuunt32l63ya --args hello world
```

UCAN Decentralize Auth

Decentralised Wasm Repositories

UCAN Decentralize Auth

Decentralised Wasm Repositories

+ NNS

WA

Join Us

✊

Join Us ✊

Connect

Join Us ✊

Connect

- **Community**: github.com/ipvm-wg

Join Us ✊

Connect

- **Community**: github.com/ipvm-wg

Join Us ✊

Connect

- **Community**: github.com/ipvm-wg

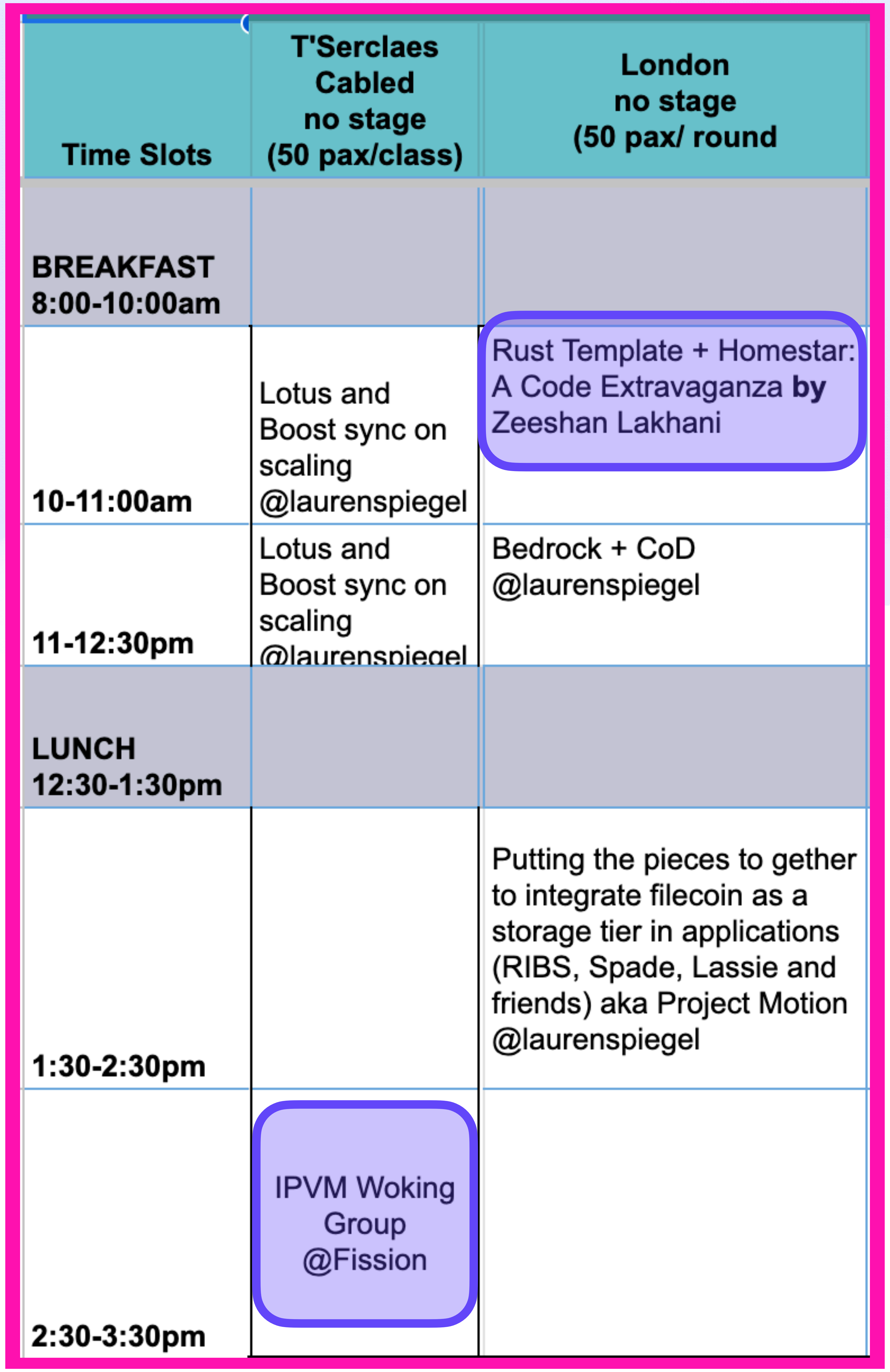
- **Calls**: lu.ma/ipvm

Join Us ✊

Connect

- **Community**: github.com/ipvm-wg

- **Calls**: lu.ma/ipvm



Time Slots	T'Serclaes Cabled no stage (50 pax/class)	London no stage (50 pax/ round
BREAKFAST 8:00-10:00am		
10-11:00am	Lotus and Boost sync on scaling @laurenspiegel	Rust Template + Homestar: A Code Extravaganza **by** Zeeshan Lakhani
11-12:30pm	Lotus and Boost sync on scaling @laurenspiegel	Bedrock + CoD @laurenspiegel
LUNCH 12:30-1:30pm		
1:30-2:30pm		Putting the pieces to gether to integrate filecoin as a storage tier in applications (RIBS, Spade, Lassie and friends) aka Project Motion @laurenspiegel
2:30-3:30pm	IPVM Woking Group @Fission	

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

🎉 *þank you, IPFS þing* 🇧🇪

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