

The Dark Forest

Private CRDTs as Global Database

or: "How to Power a New Internet"

github.com/wnfs-wg
github.com/fission-codes



BROOKLYN ZELENKA



@expede
Cofounder & CTO
Fission Codes



BROOKLYN ZELENKA



@expede
Cofounder & CTO
Fission Codes



BROOKLYN ZELENKA



@expede
Cofounder & CTO
Fission Codes

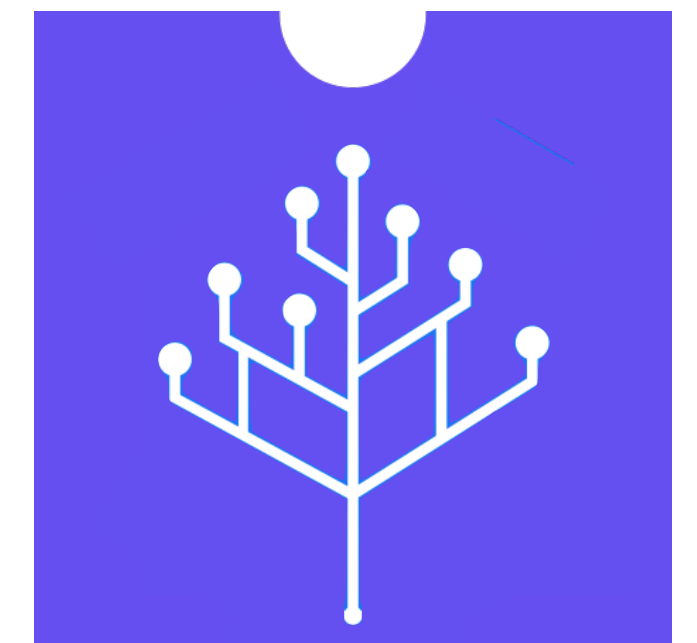


✨ **WNFS** ✨

Dialog

CAR Pool

IPVM



The Situation



The Situation 🤔

1. Services In 2022

The Situation 🤨

1. Services In 2022

Alice

Bob

Carol

The Situation 🤨


1. Services In 2022

Alice

Bob

Carol

Music Service



The Situation 🤨

1. Services In 2022

Alice


Bob

Carol

Alice's Music

Bob's Music

Music Service



Carol's Music

The Situation 🤔

1. Services In 2022

Alice

Bob

Carol

Alice's Music

Bob's Music

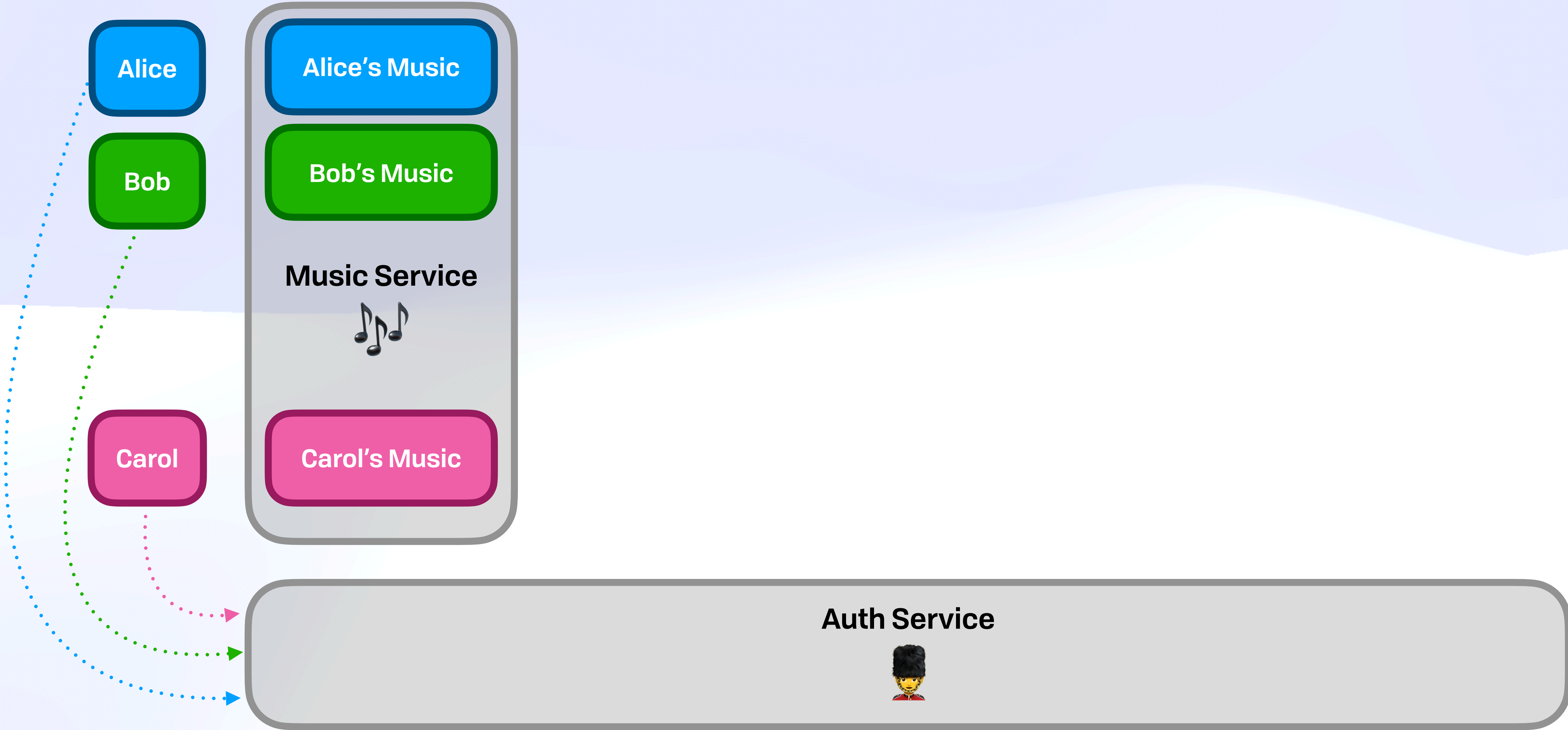
Music Service
🎵

Carol's Music

Auth Service
👤

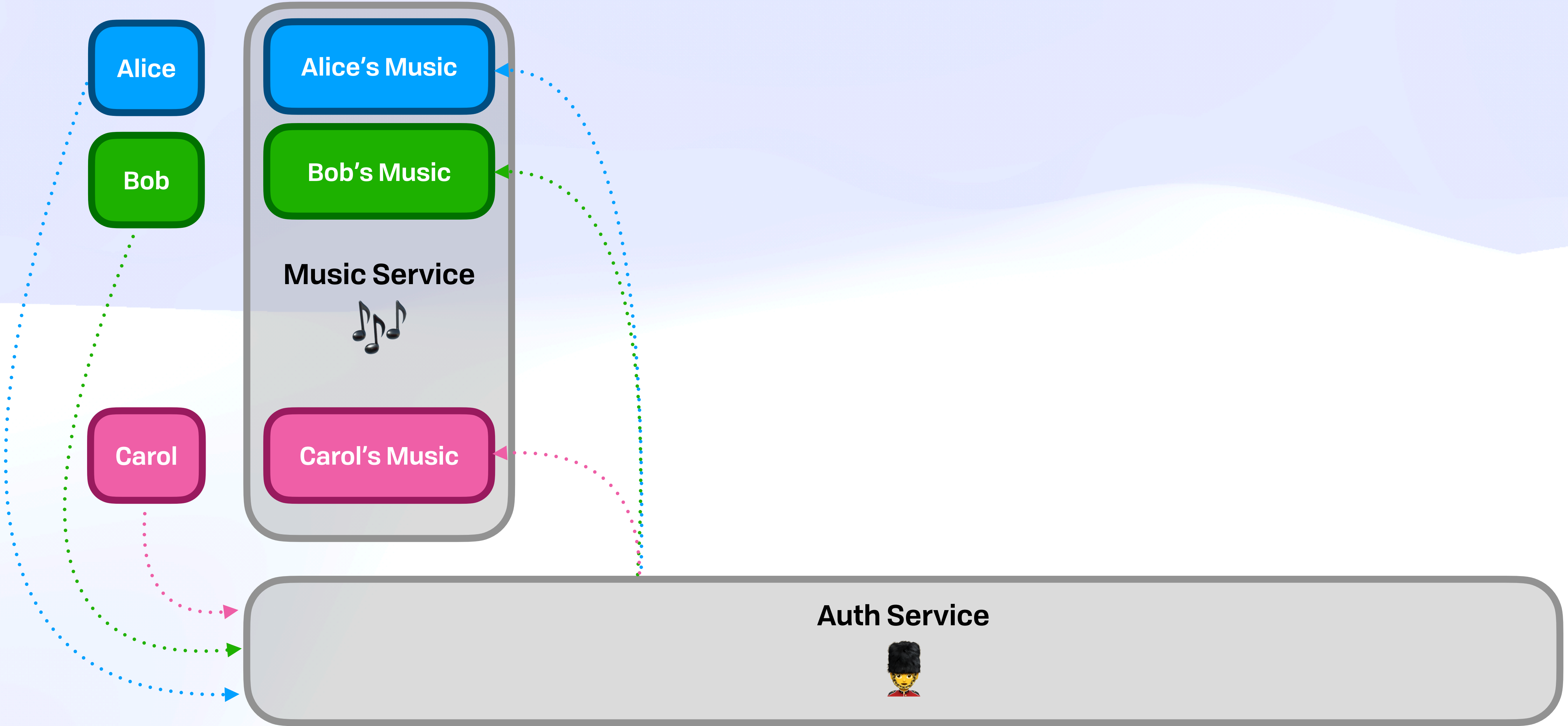
The Situation 🤔

1. Services In 2022



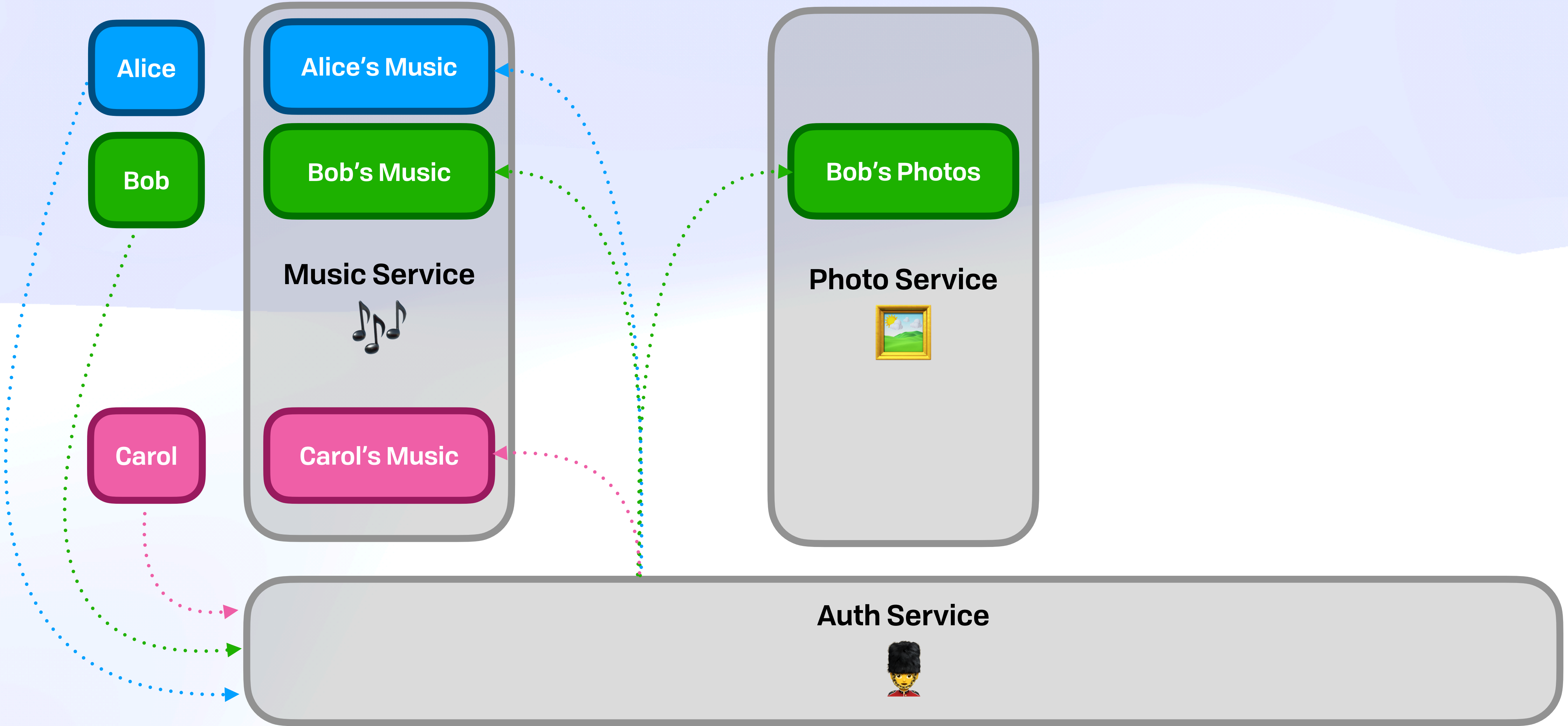
The Situation 🤔

1. Services In 2022



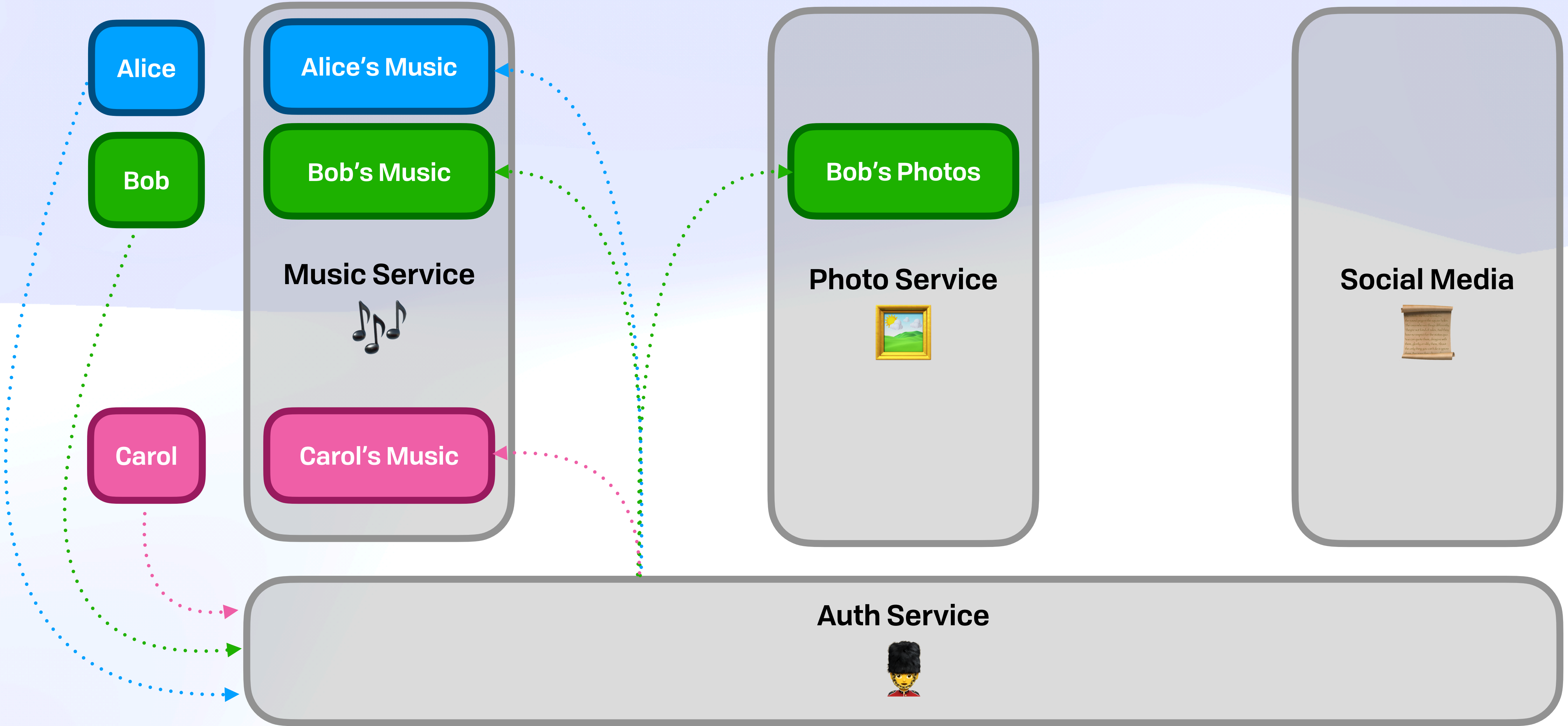
The Situation 🤔

1. Services In 2022



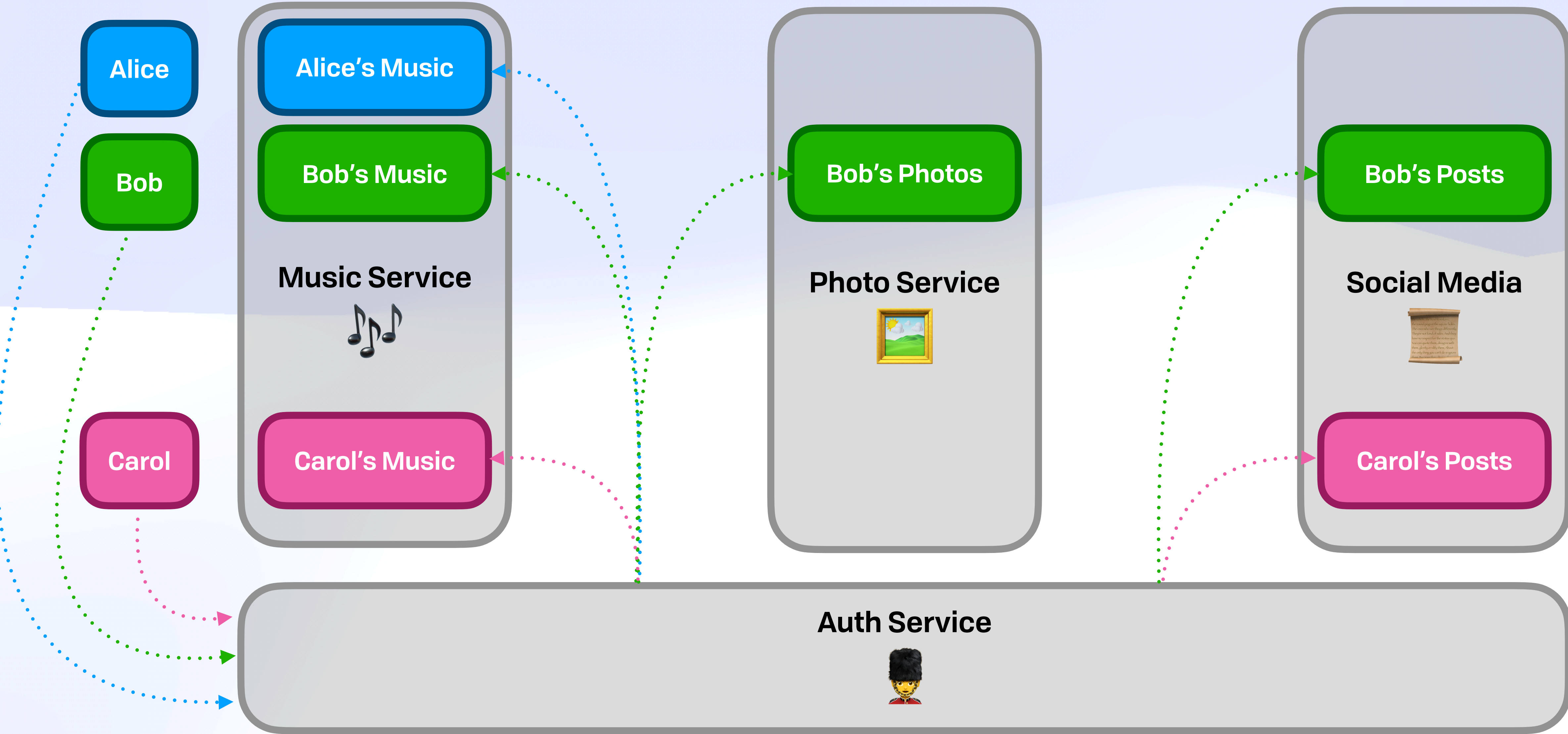
The Situation 🤔

1. Services In 2022



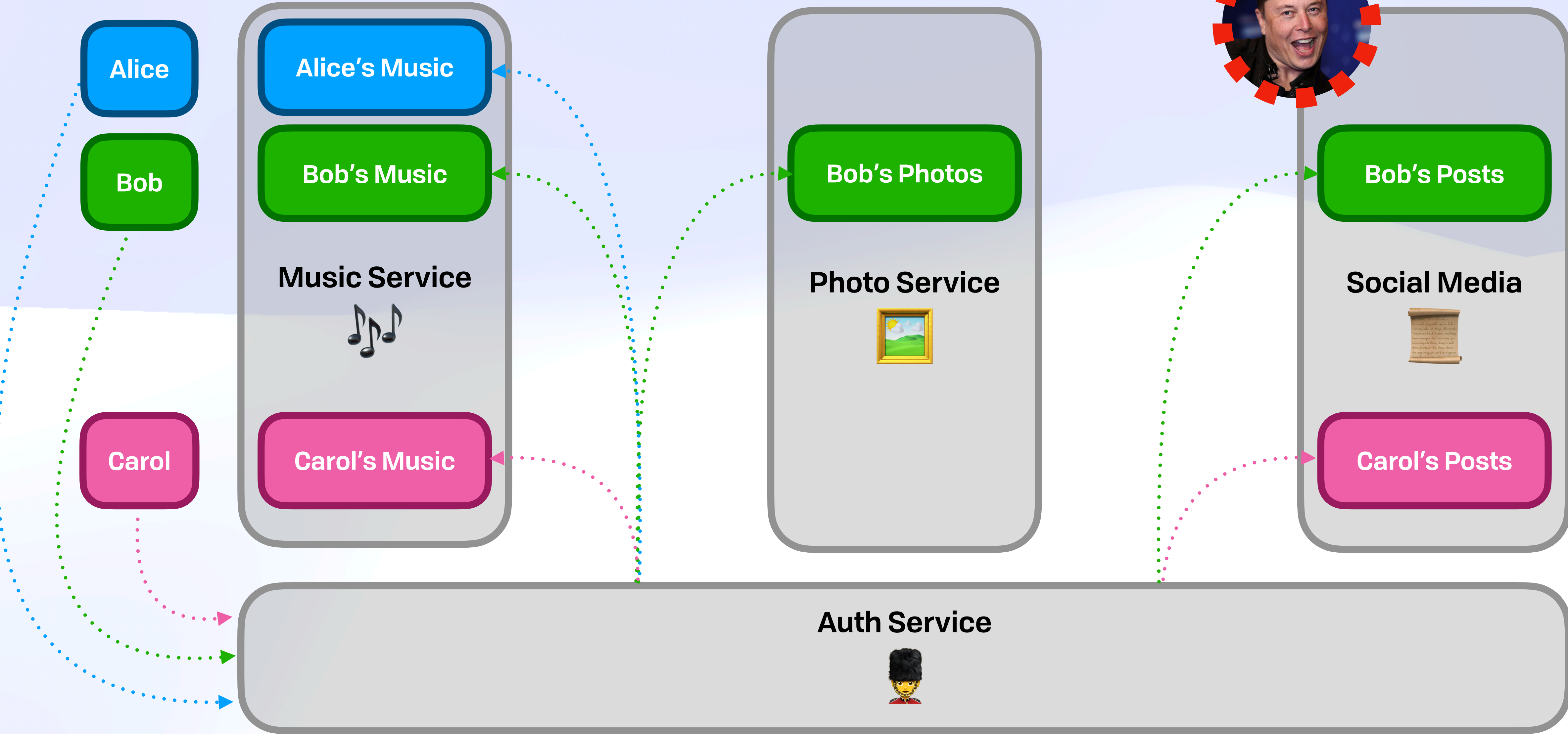
The Situation 🤔

1. Services In 2022



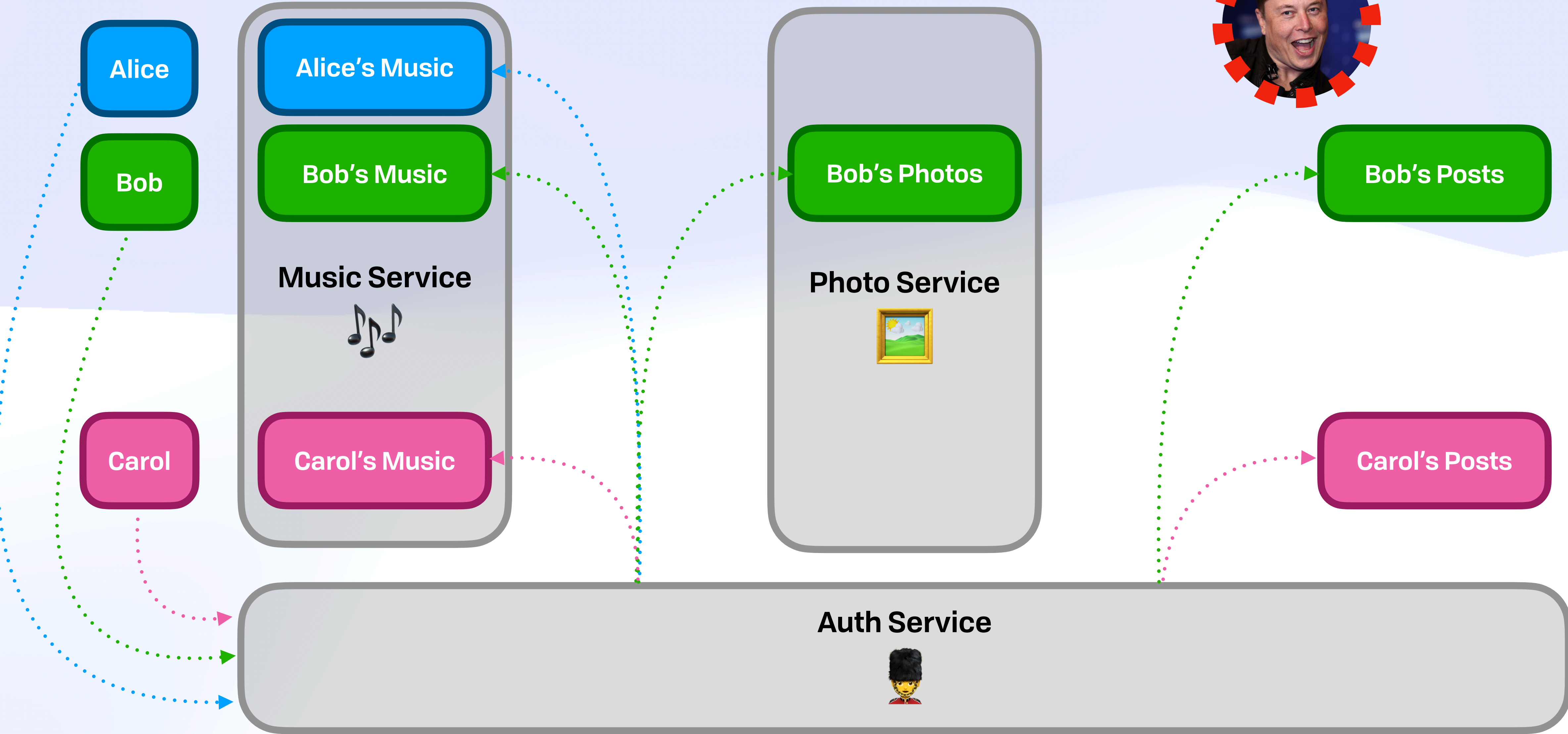
The Situation 🤨

1. Services In 2022



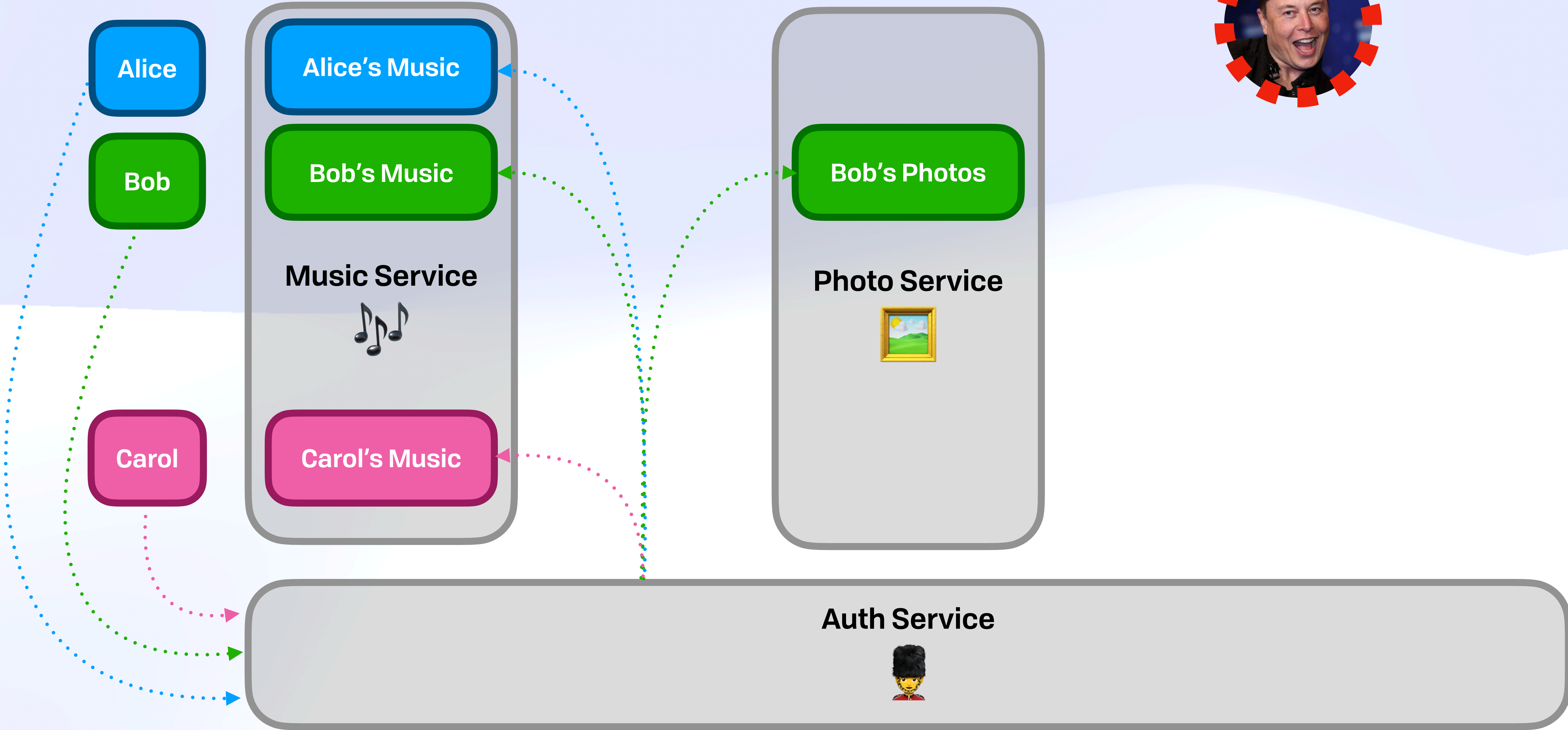
The Situation 🤨

1. Services In 2022



The Situation 🤨

1. Services In 2022



The Situation 🤨

2. Too Much In The Way

The Situation 🤔

2. Too Much In The Way

Users 🧑🏿 🧑🏻 🧑🏼 🧑🏽

Developer 🧑🏿💻

The Situation 🤔

2. Too Much In The Way

Users 🧑🧑🧑🧑

Browser 🖥️

REST / JSON-RPC / GraphQL 🔄

Server 🏠

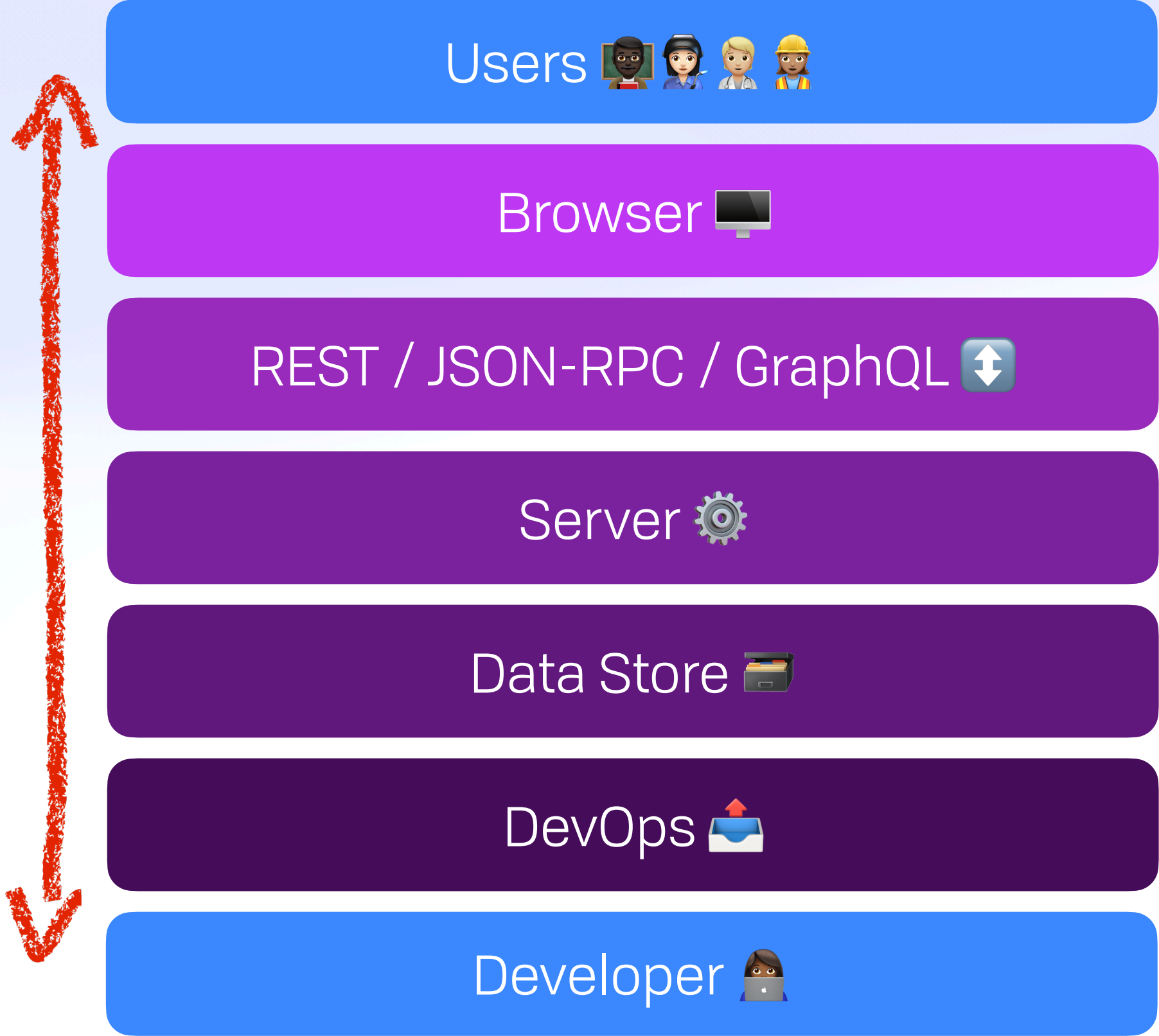
Data Store 🗄️

DevOps 📦

Developer 🧑💻

The Situation 🤔

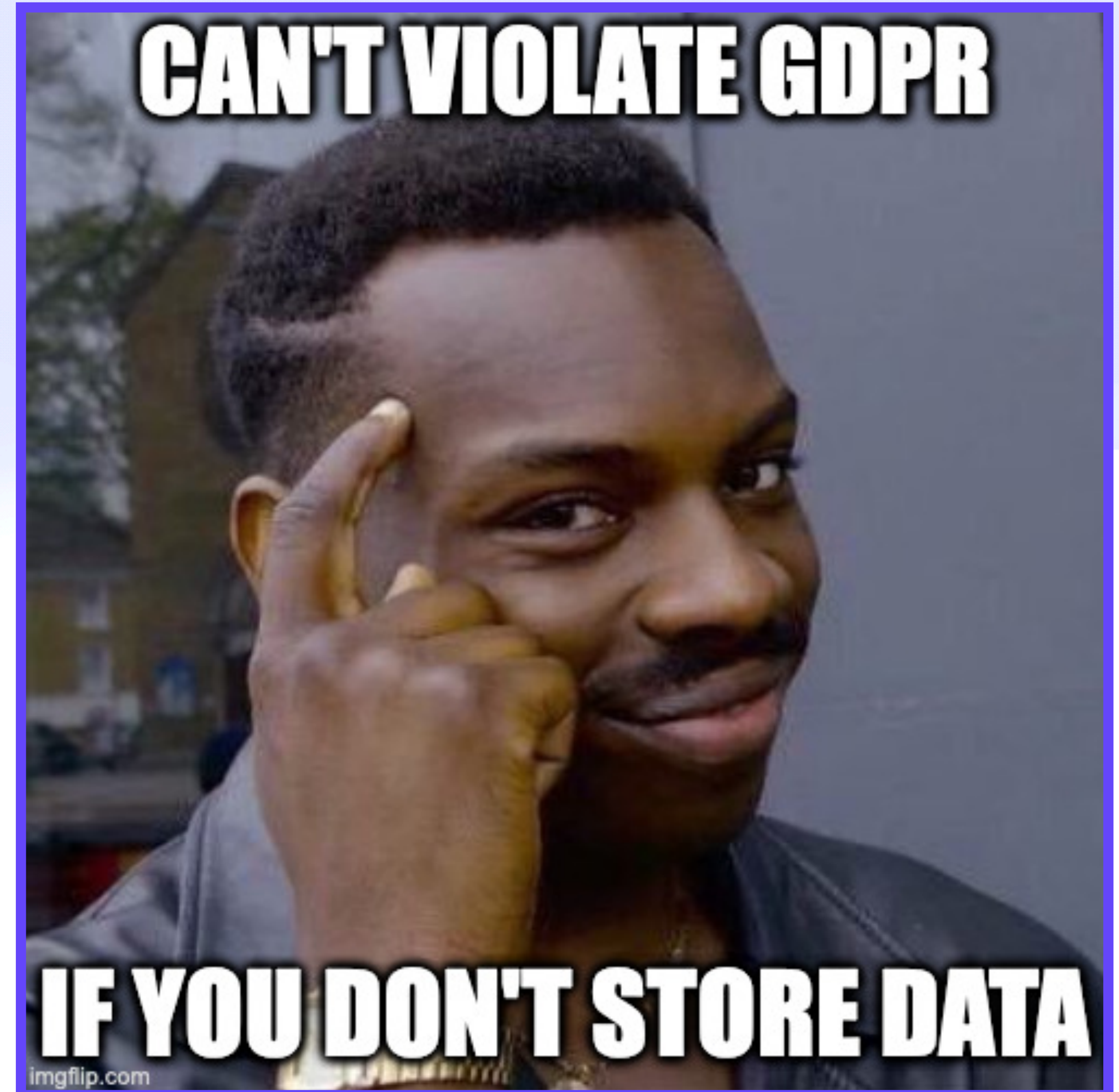
2. Too Much In The Way



The Situation 🤔

3. Toxic Data

- **2005:** Credit card info in DB
- **2015:** Personal info in DB
- **2025:** Nothing in DB



What If...

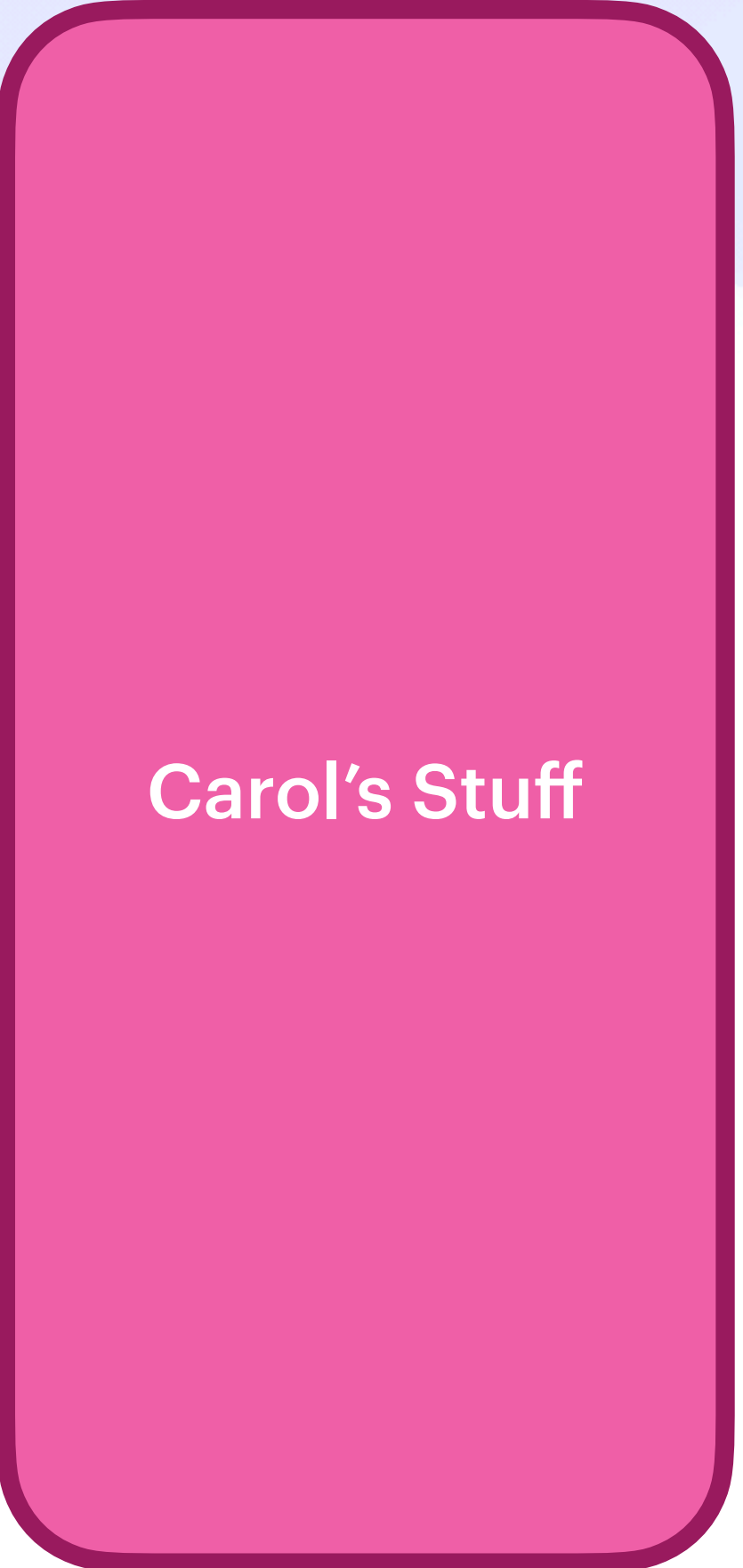
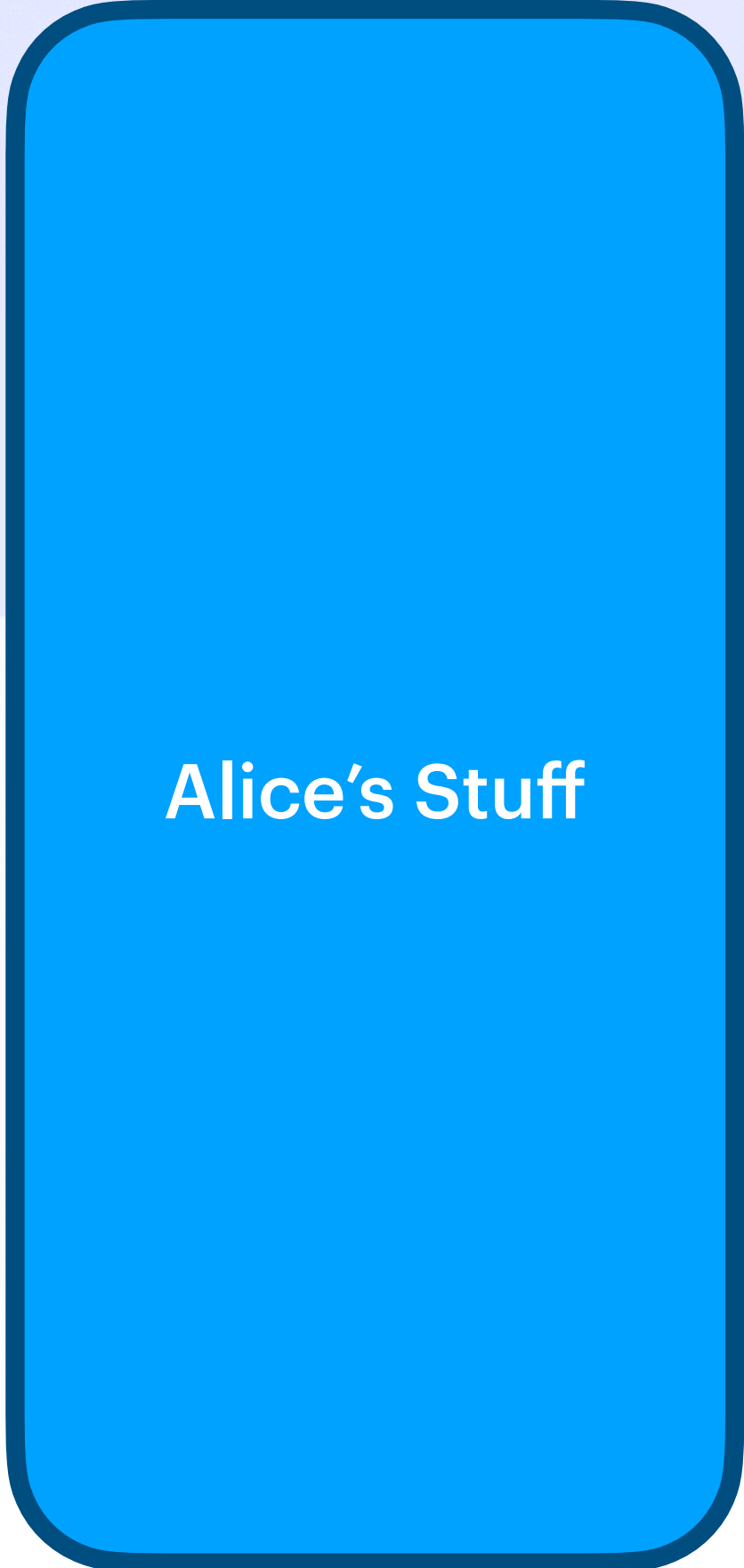


What If... 🤔

Networked Data, Not Apps

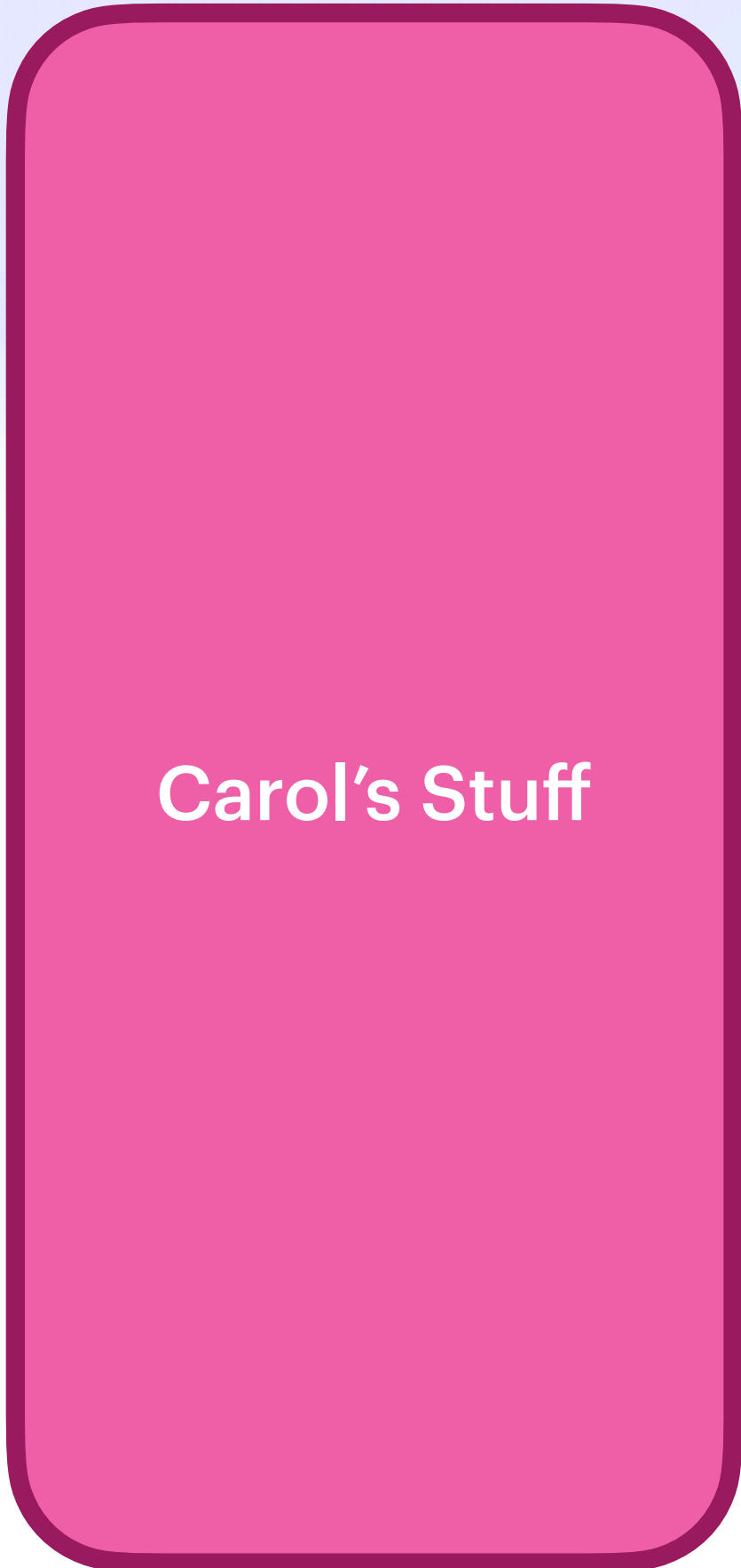
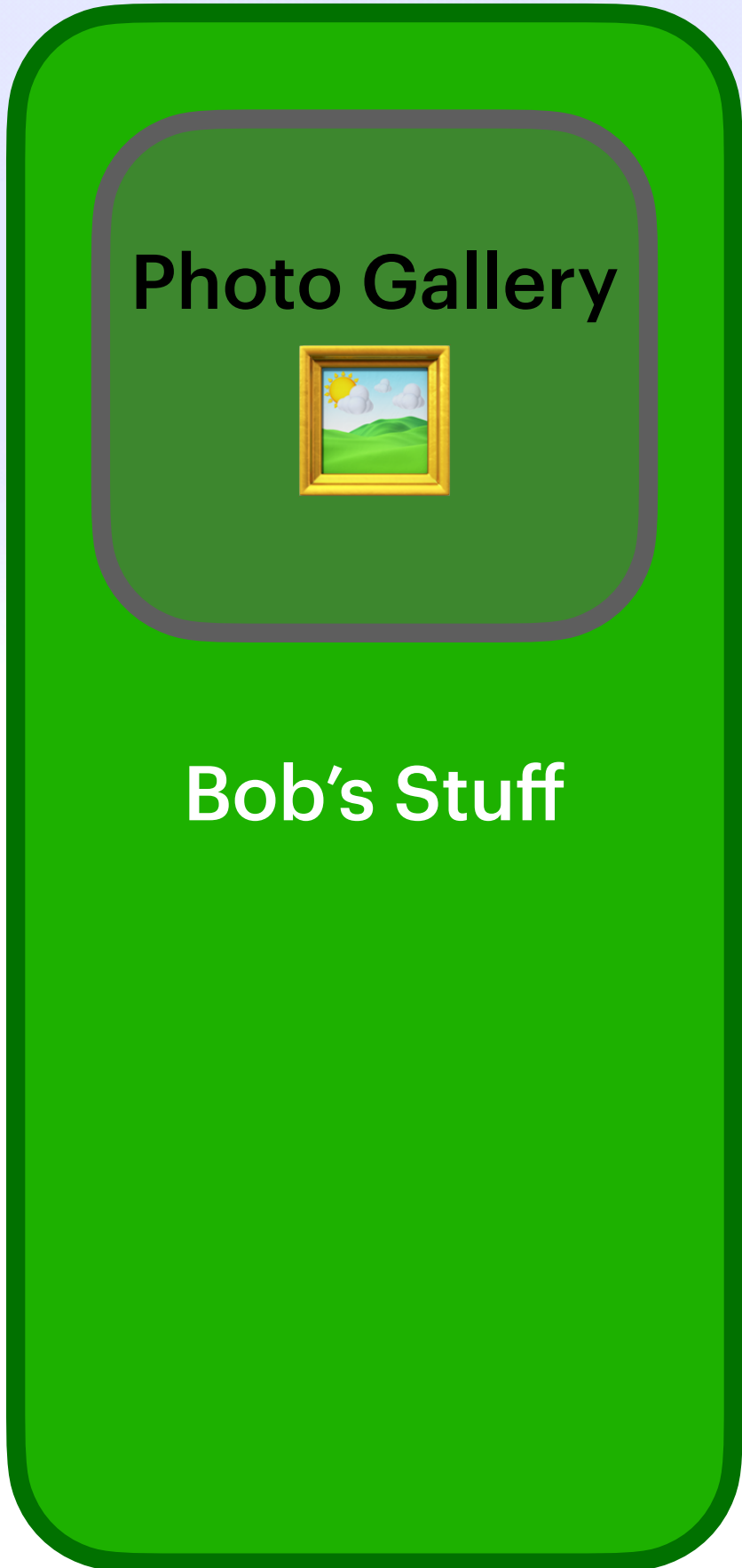
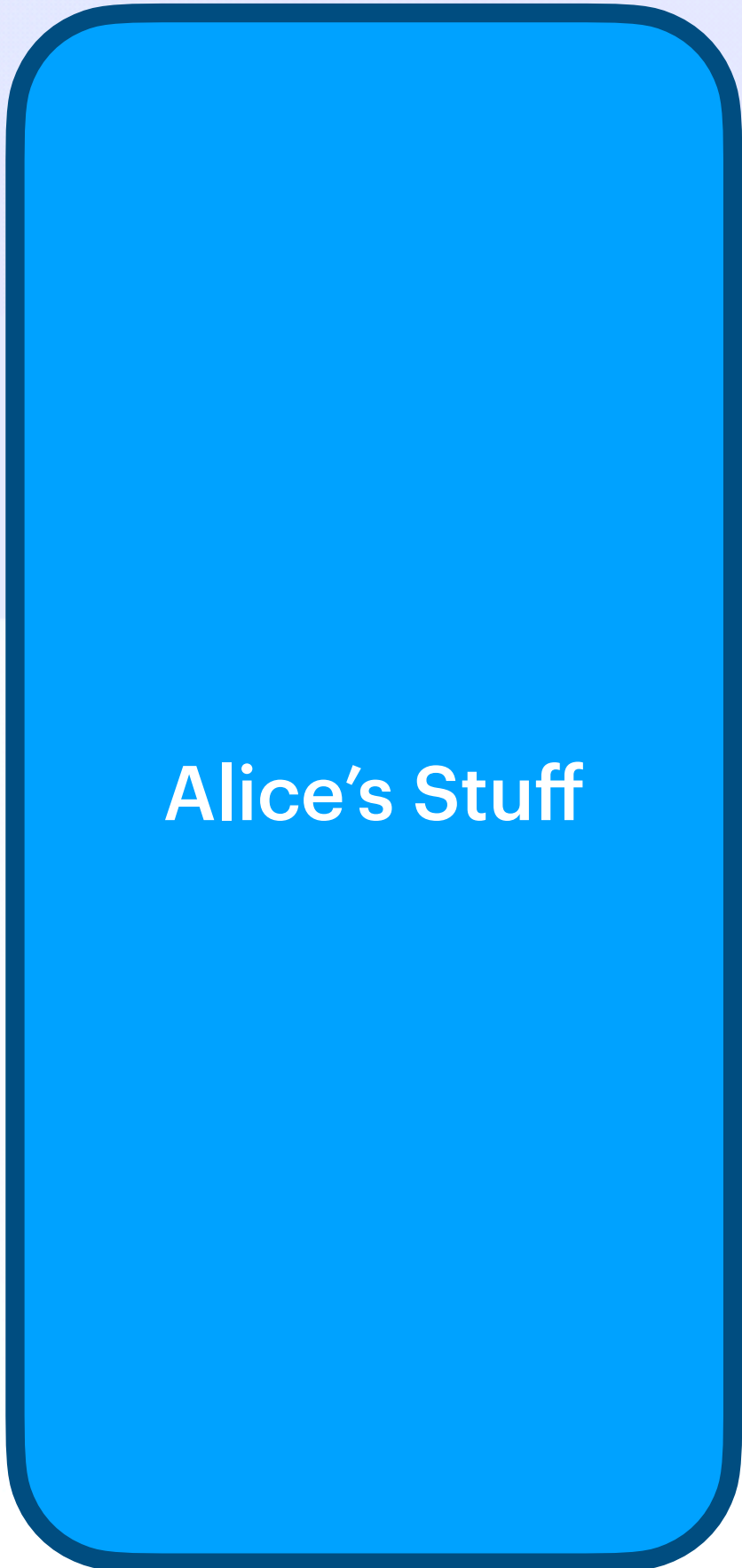
What If... 🤔

Networked Data, Not Apps



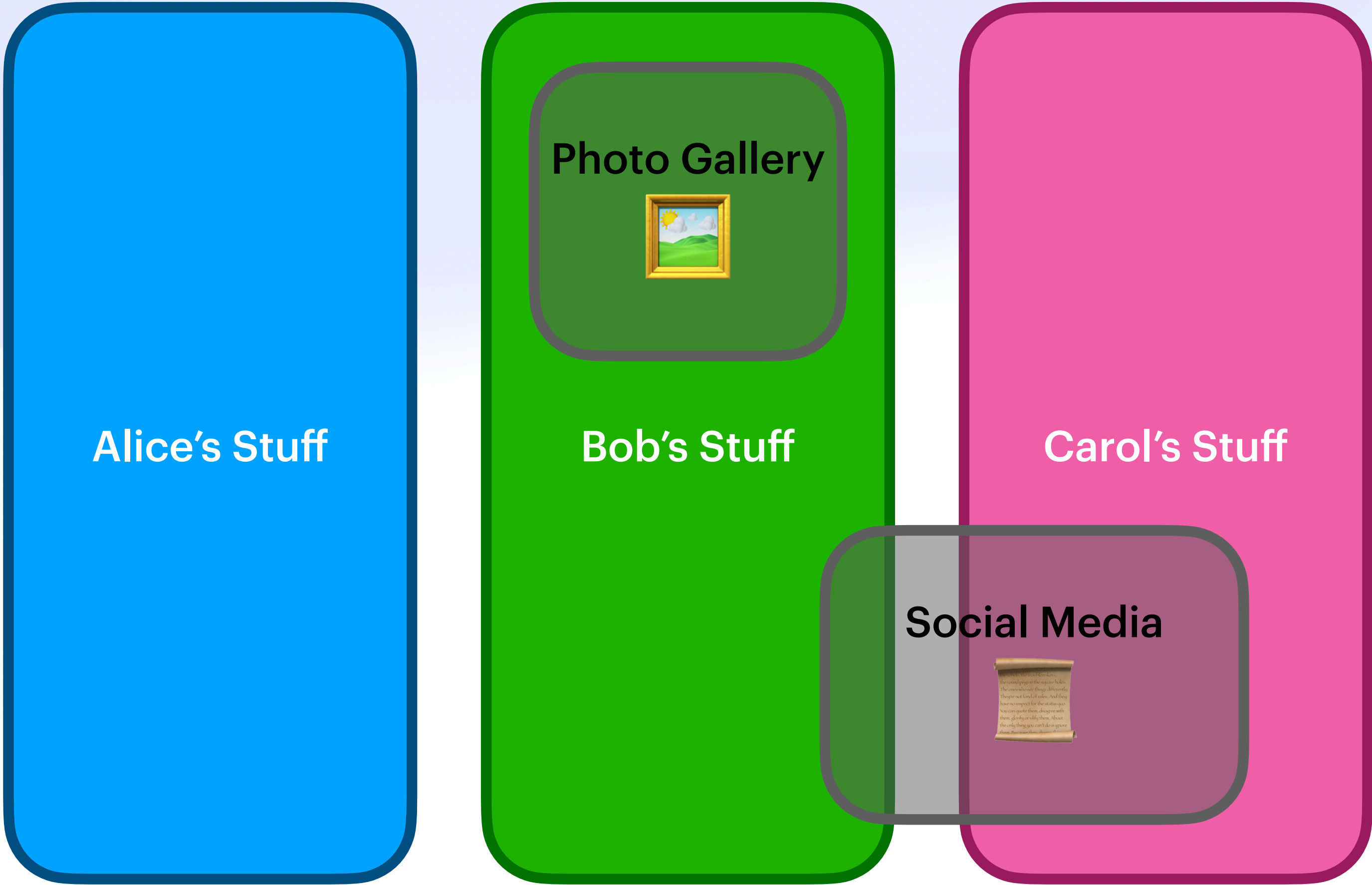
What If... 🤔

Networked Data, Not Apps



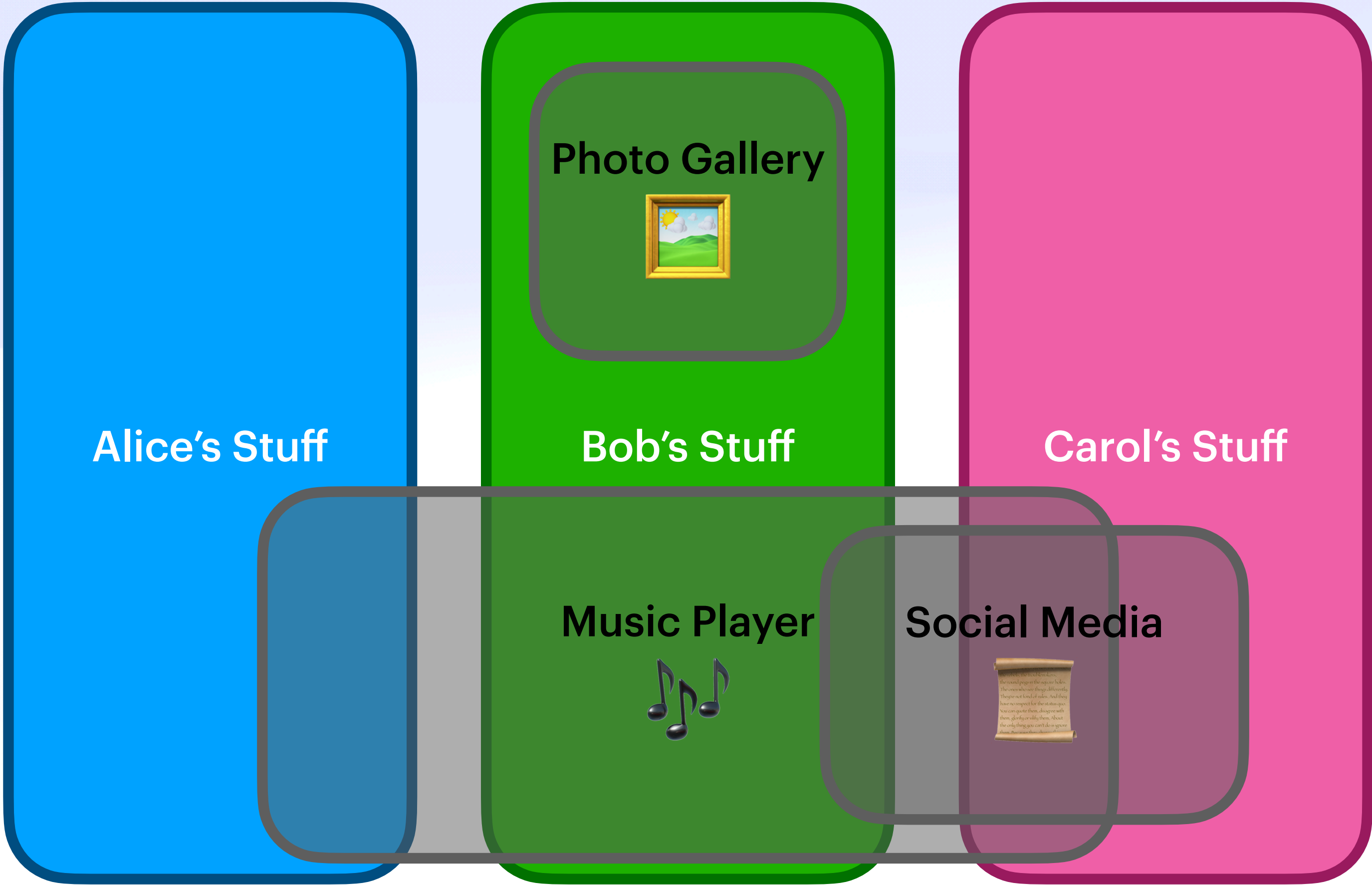
What If... 🤔

Networked Data, Not Apps



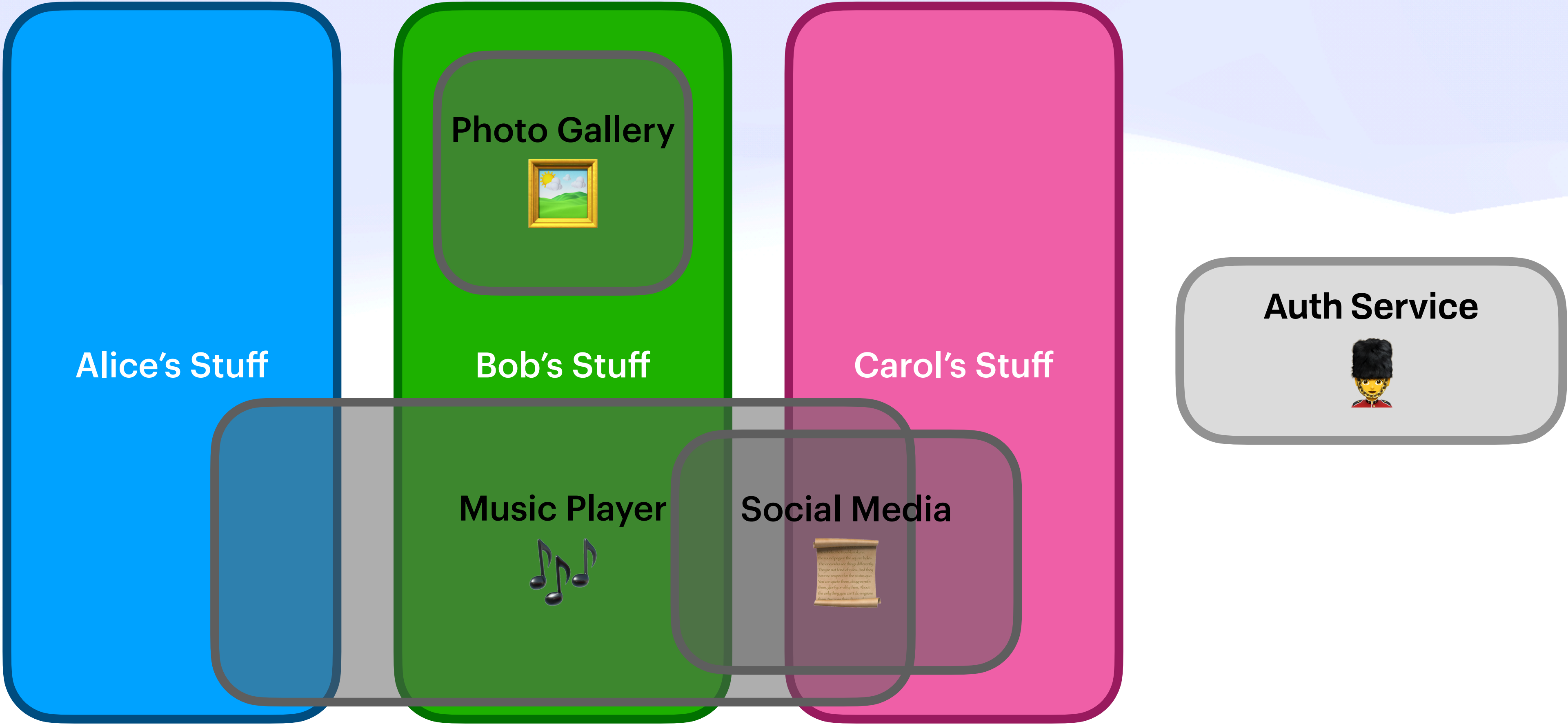
What If... 🤔

Networked Data, Not Apps



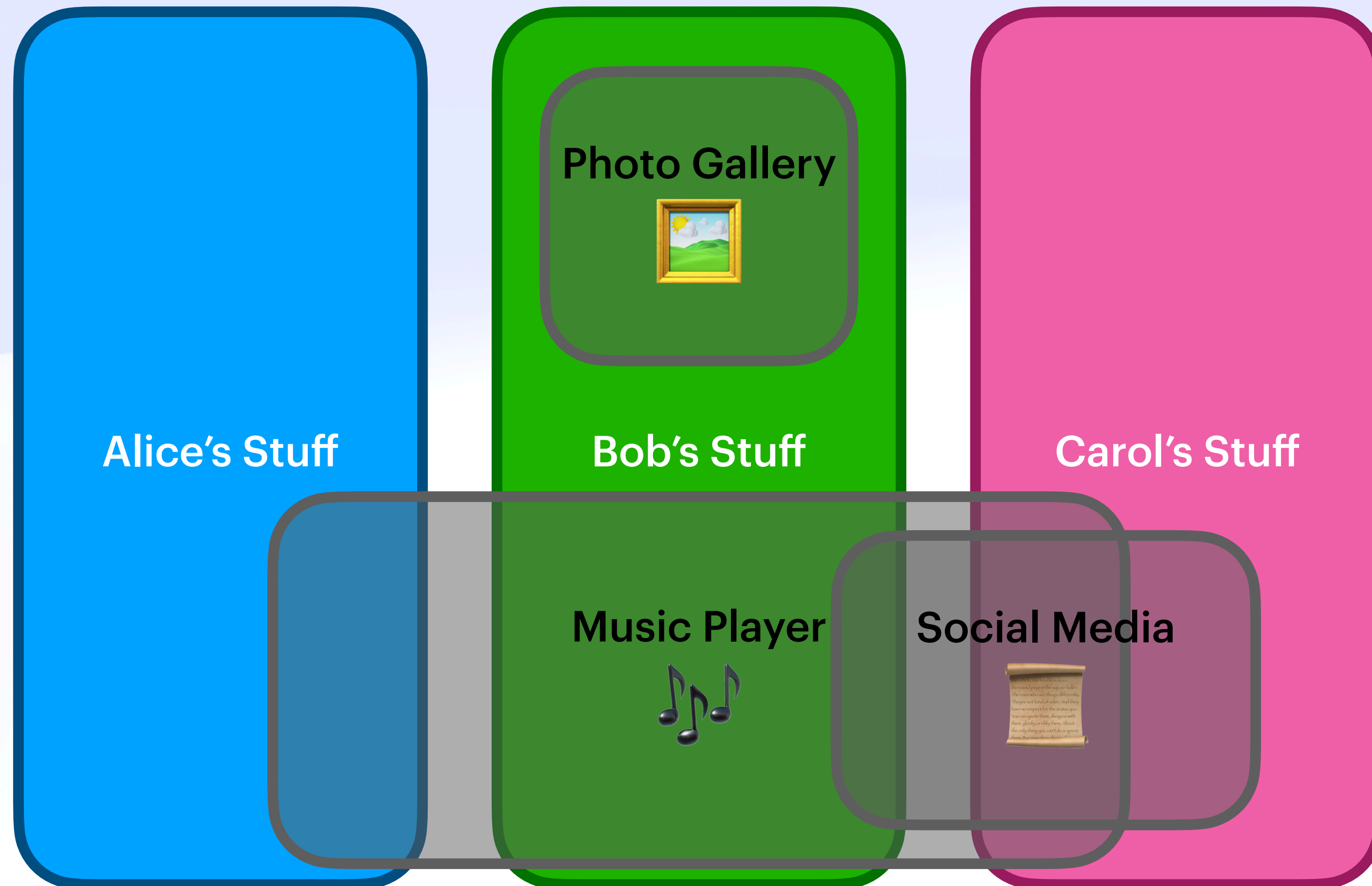
What If... 🤔

Networked Data, Not Apps



What If... 🤔

Networked Data, Not Apps



What If... 🤔

Open Protocols Everywhere

What If... 🤔

Open Protocols Everywhere

- HTTP being open lead to **tons** of innovation

Networking 

What If... 🤔

Open Protocols Everywhere

- HTTP being open lead to **tons** of innovation
- Open protocols for...

Networking 

What If... 🤔

Open Protocols Everywhere

- HTTP being open lead to **tons** of innovation
- Open protocols for...
 - **Auth:** interop without pre-negotiation

Auth 

Networking 

What If... 🤔

Open Protocols Everywhere

- HTTP being open lead to **tons** of innovation
- Open protocols for...
 - **Auth:** interop without pre-negotiation
 - **Data:** ubiquitous "dumb" storage

Data 

Auth 

Networking 

What If... 🤔

Open Protocols Everywhere

- HTTP being open lead to **tons** of innovation
- Open protocols for...
 - **Auth:** interop without pre-negotiation
 - **Data:** ubiquitous "dumb" storage
 - **Compute:** local & remote lambdas

Compute 

Data 

Auth 

Networking 

What If... 🤔

Open Protocols Everywhere

- HTTP being open lead to **tons** of innovation
- Open protocols for...
 - **Auth:** interop without pre-negotiation
 - **Data:** ubiquitous "dumb" storage
 - **Compute:** local & remote lambdas

Compute 

Data 

Auth 

Networking 

What If... 🤔

Three Techniques

What If... 🤔

Three Techniques

Global Primary Keys 🌍

Portable Private Data 💾

BFT Concurrency 🔁

What If... 🤔

Three Techniques

Global Primary Keys 🌍

Portable Private Data 💾

BFT Concurrency 🔁

Content Addressing

What If... 🤔

Three Techniques

Global Primary Keys 🌍

Content Addressing

Portable Private Data 💾

Cryptrees / Dark Forest

BFT Concurrency 🔁

What If... 🤔

Three Techniques

Global Primary Keys 🌍

Content Addressing

Portable Private Data 💾

Cryptrees / Dark Forest

BFT Concurrency 🔁

Hash-Linked CRDTs

Content Addressing

Truly Global Links



Content Addressing

Content Addressing

The limitation of **local knowledge**
is the **fundamental fact**
about the setting in which we work,
and it is **a very powerful limitation**

– Nancy Lynch, A Hundred Impossibility Proofs for Distributed Computing

Content Addressing 

Addressing Stack

Content Addressing 

Addressing Stack

Physical Location 

```
send(42.123.45.6, path) = content1
```

```
send(42.123.45.6, path) = content2
```

Content Addressing 

Addressing Stack

Virtual Address 

{DNS: IP}

Physical Location 

`send(42.123.45.6, path) = content1`

`send(42.123.45.6, path) = content2`

Content Addressing 

Addressing Stack

Content ID 

```
{hash(content): content}
```

Virtual Address 

```
{DNS: IP}
```

Physical Location 

```
send(42.123.45.6, path) = content1
```

```
send(42.123.45.6, path) = content2
```

Content Addressing 

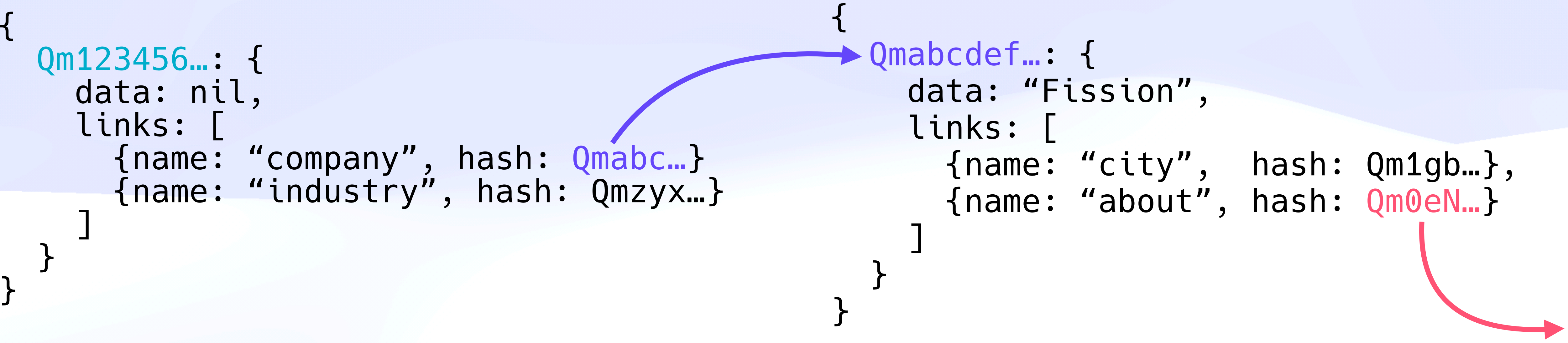
Hash-Based Relationships

Content Addressing

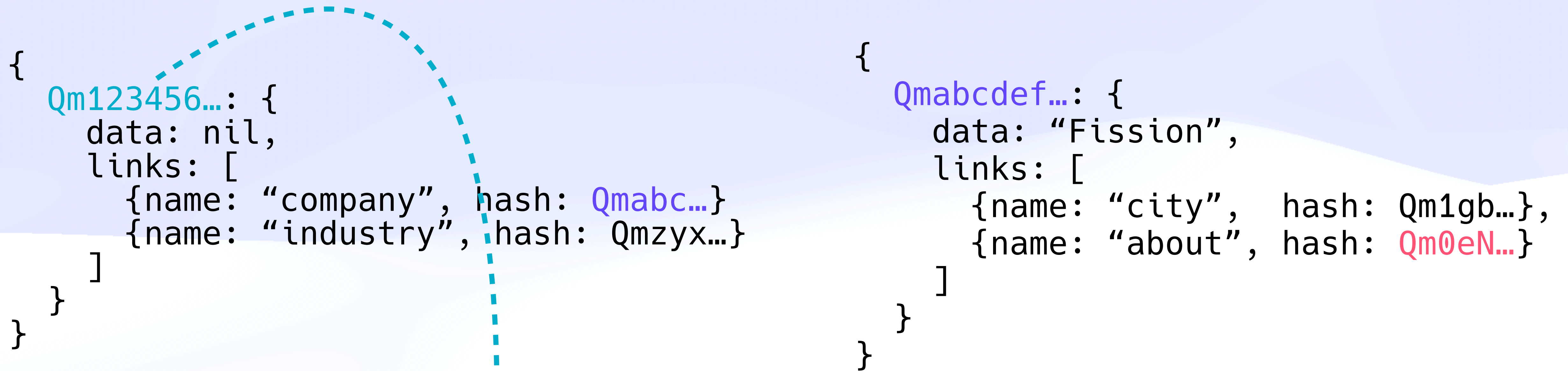
Hash-Based Relationships

```
{
  Qm123456...: {
    data: nil,
    links: [
      {name: "company", hash: Qmabc...}
      {name: "industry", hash: Qmzyx...}
    ]
  }
}
```

Hash-Based Relationships



Hash-Based Relationships



Qm123456.../company/about/ceo
=> "Boris Mann"

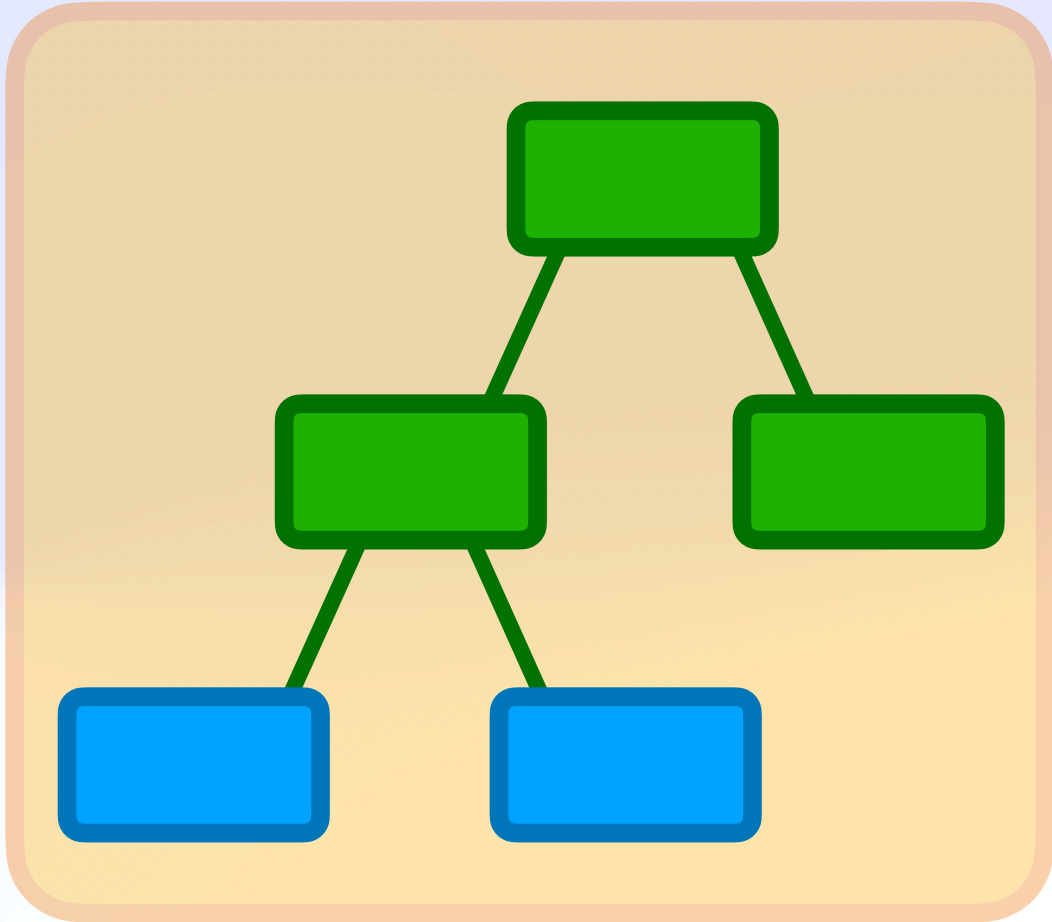
Content Addressing 

Hard & Soft Links

Content Addressing 🌐

Hard & Soft Links

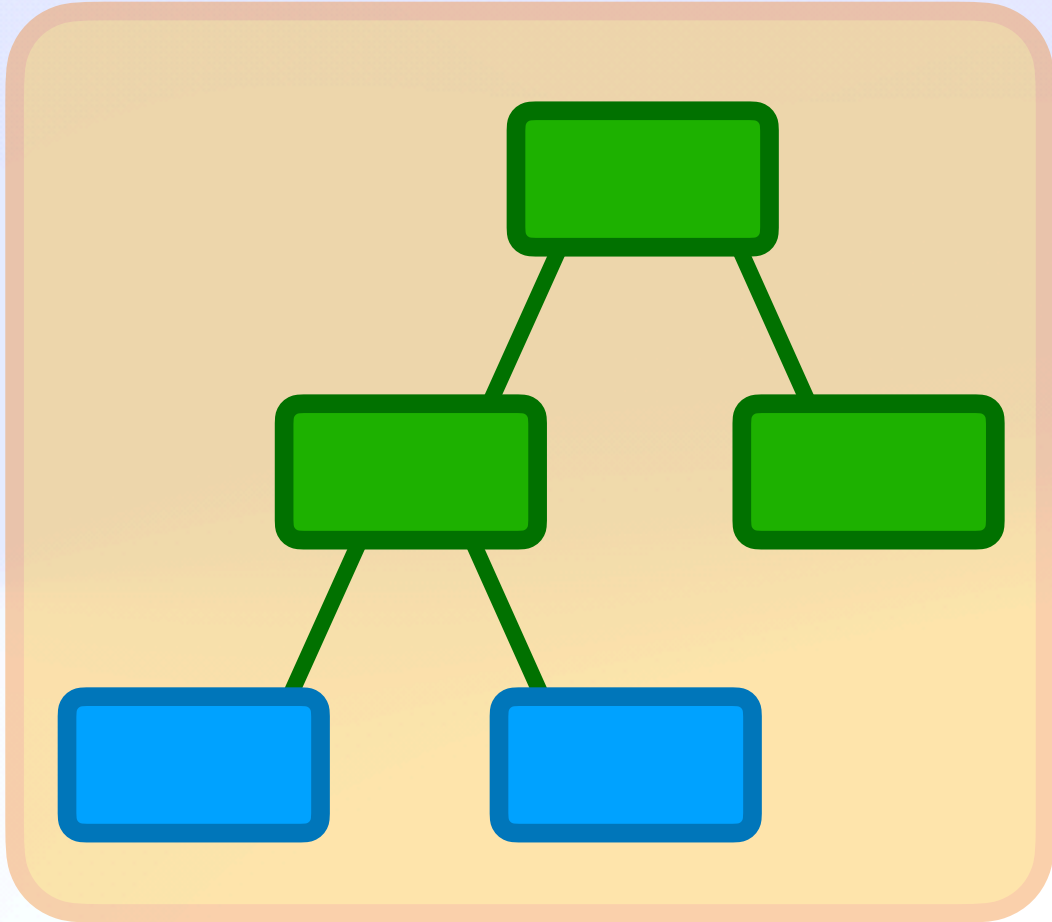
alice.fission.name



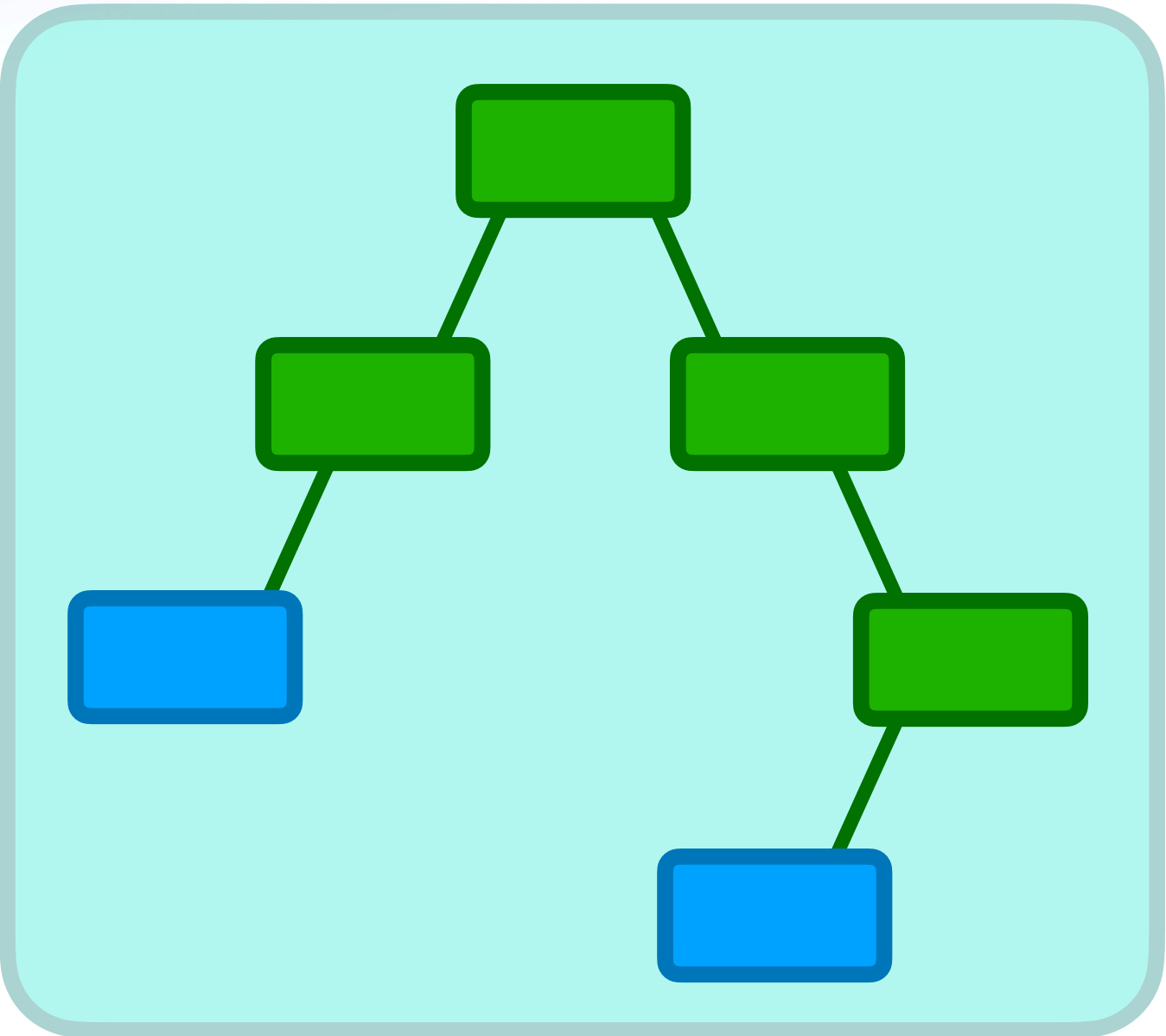
Content Addressing 🌐

Hard & Soft Links

alice.fission.name



bob.fission.name



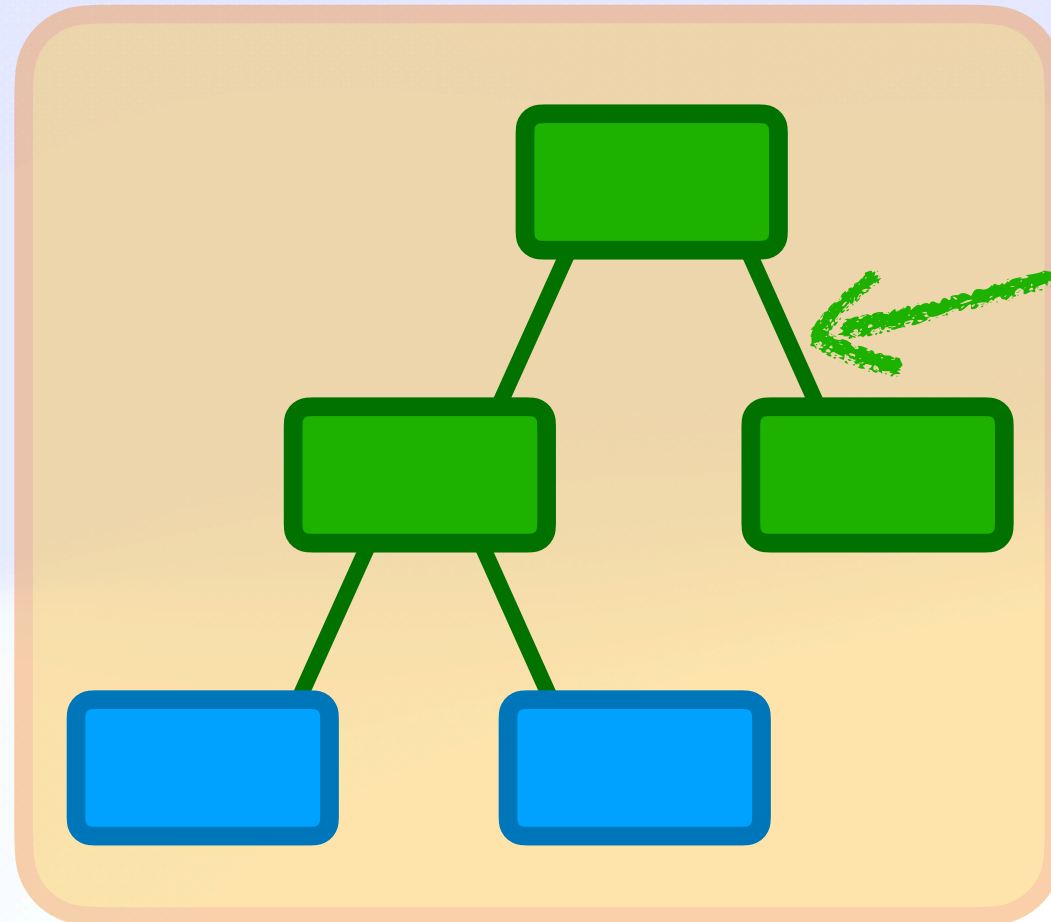
Content Addressing

Hard & Soft Links

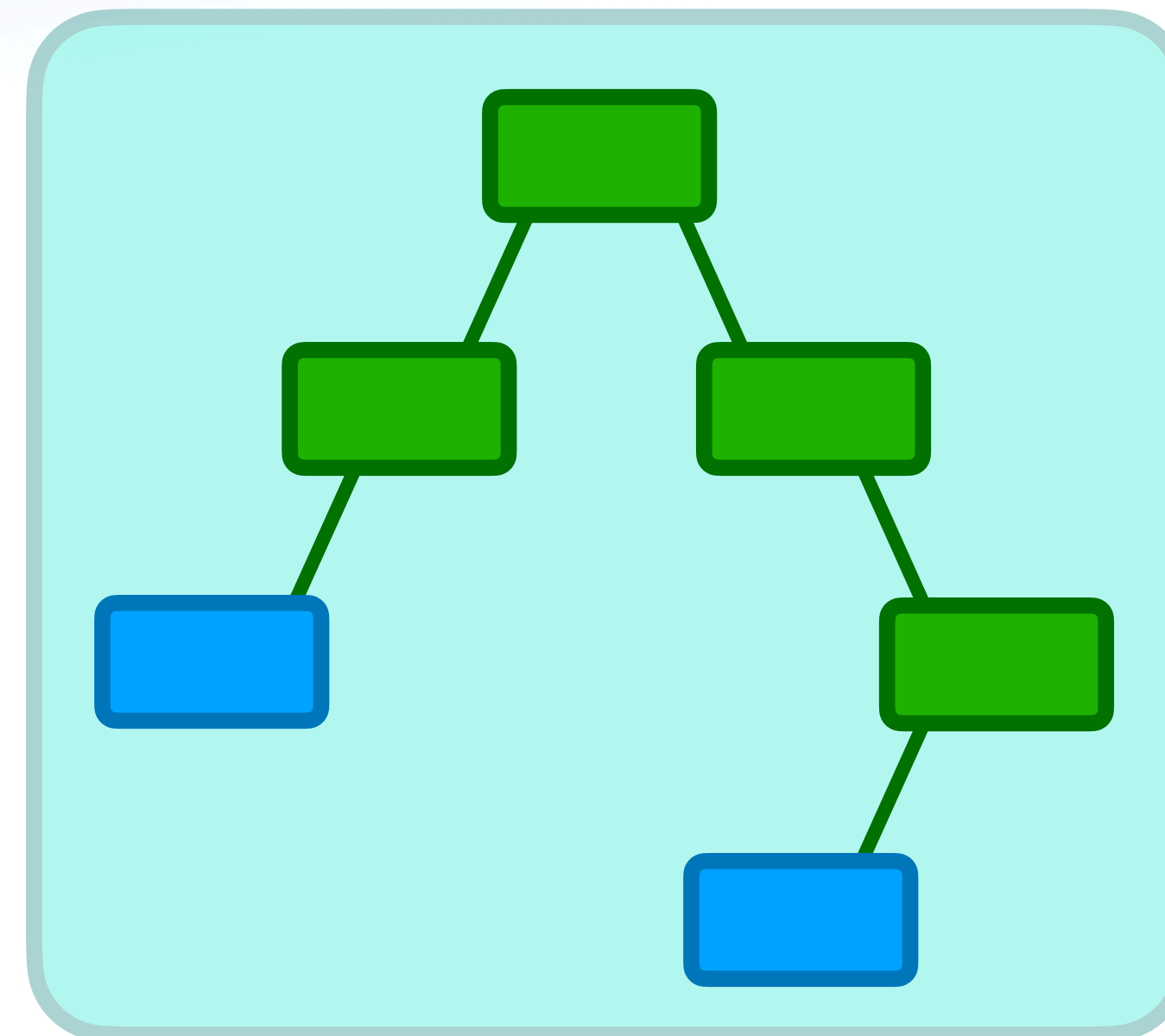
Hard Links

- ◆ New for the web!
- ◆ Direct reference
- ◆ 2 pointers ~ deduplicate

alice.fission.name



bob.fission.name

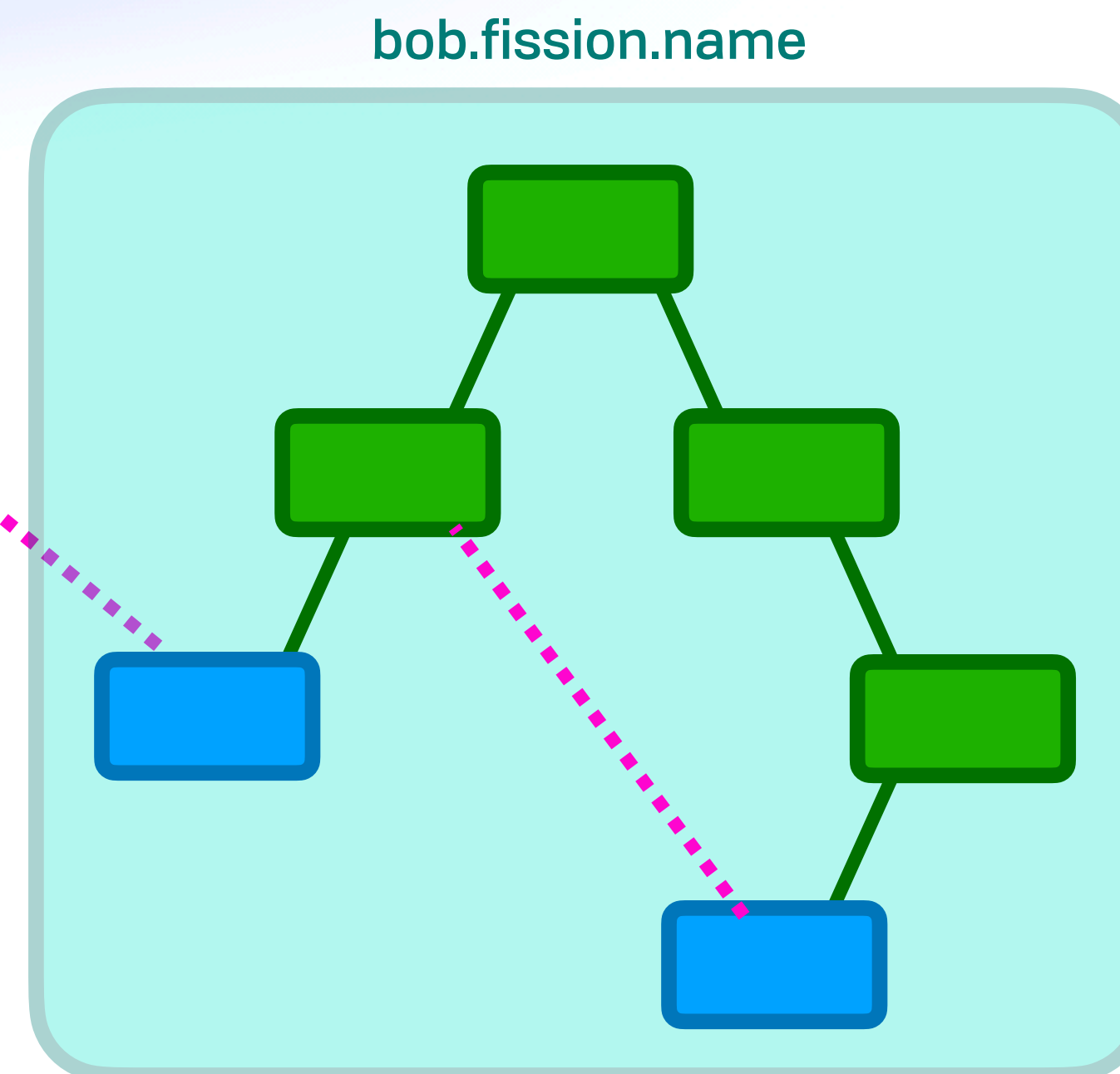
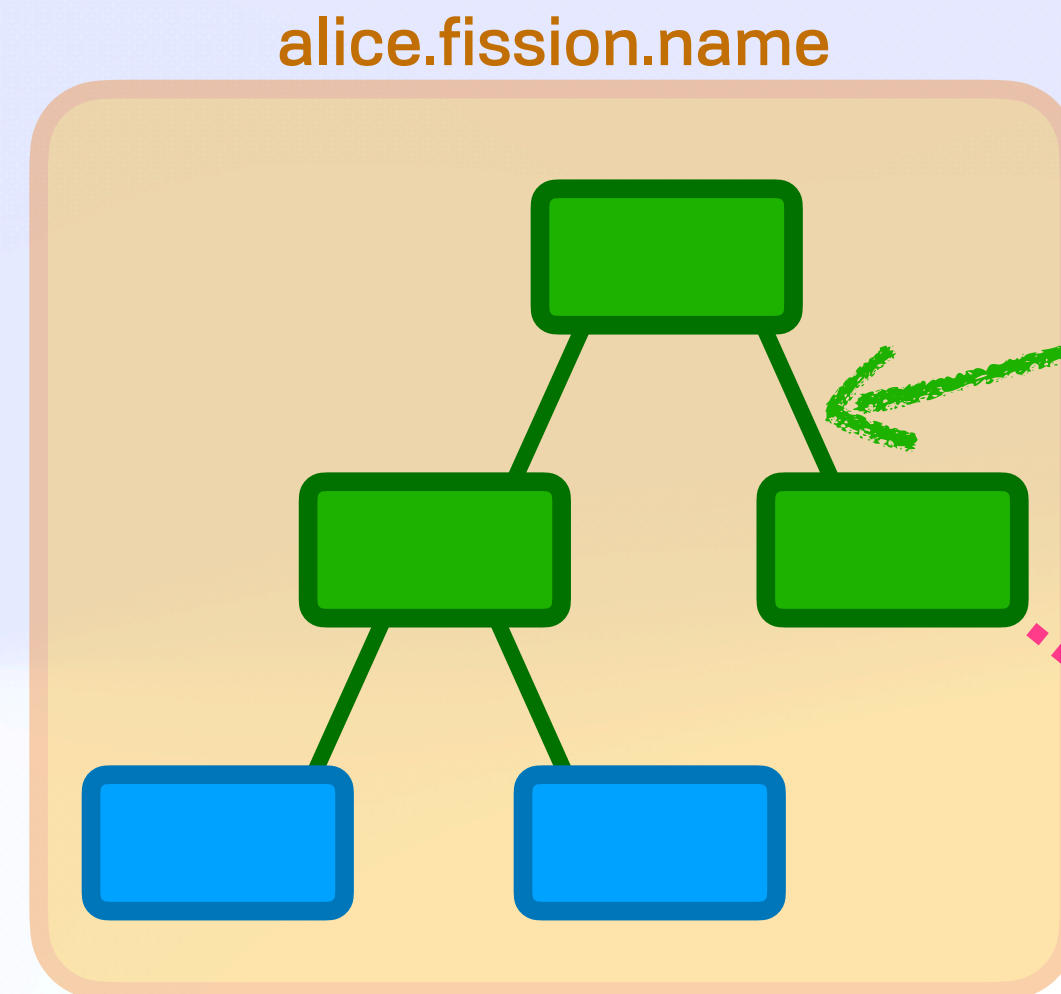


Content Addressing

Hard & Soft Links

Hard Links

- ◆ New for the web!
- ◆ Direct reference
- ◆ 2 pointers ~ deduplicate



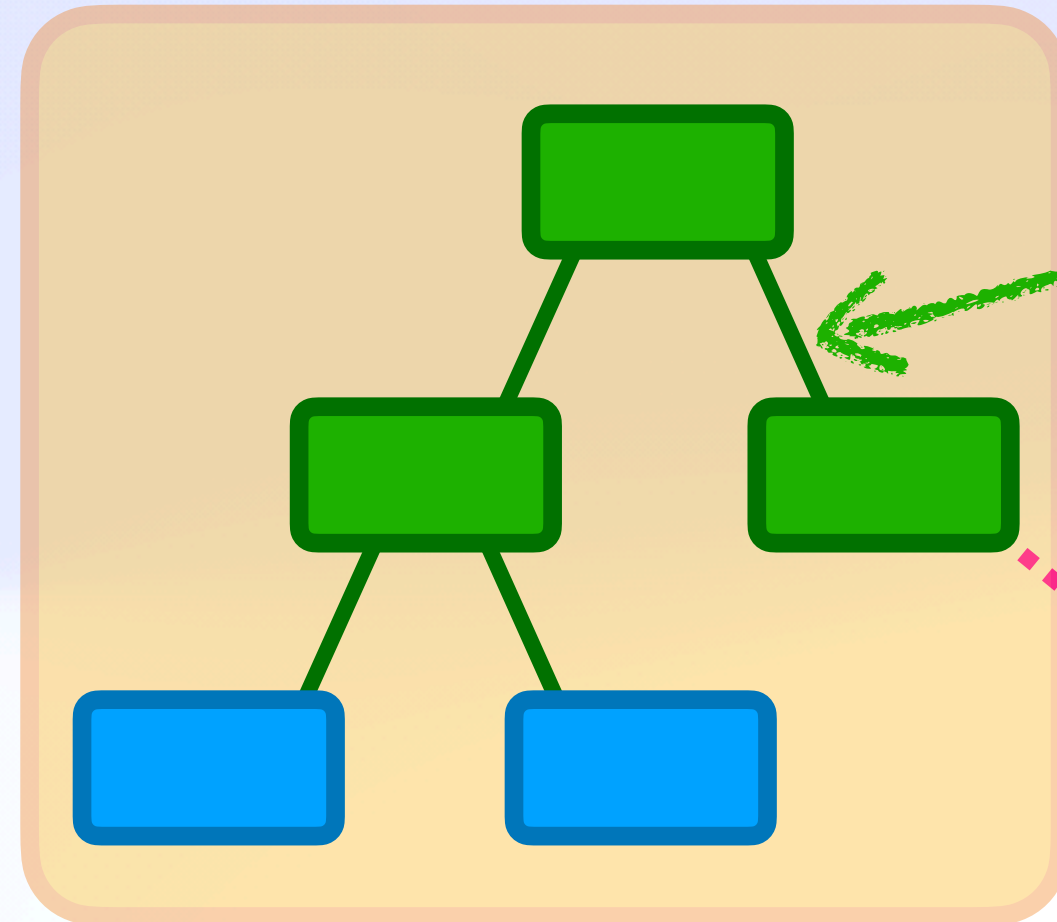
Content Addressing 🌐

Hard & Soft Links

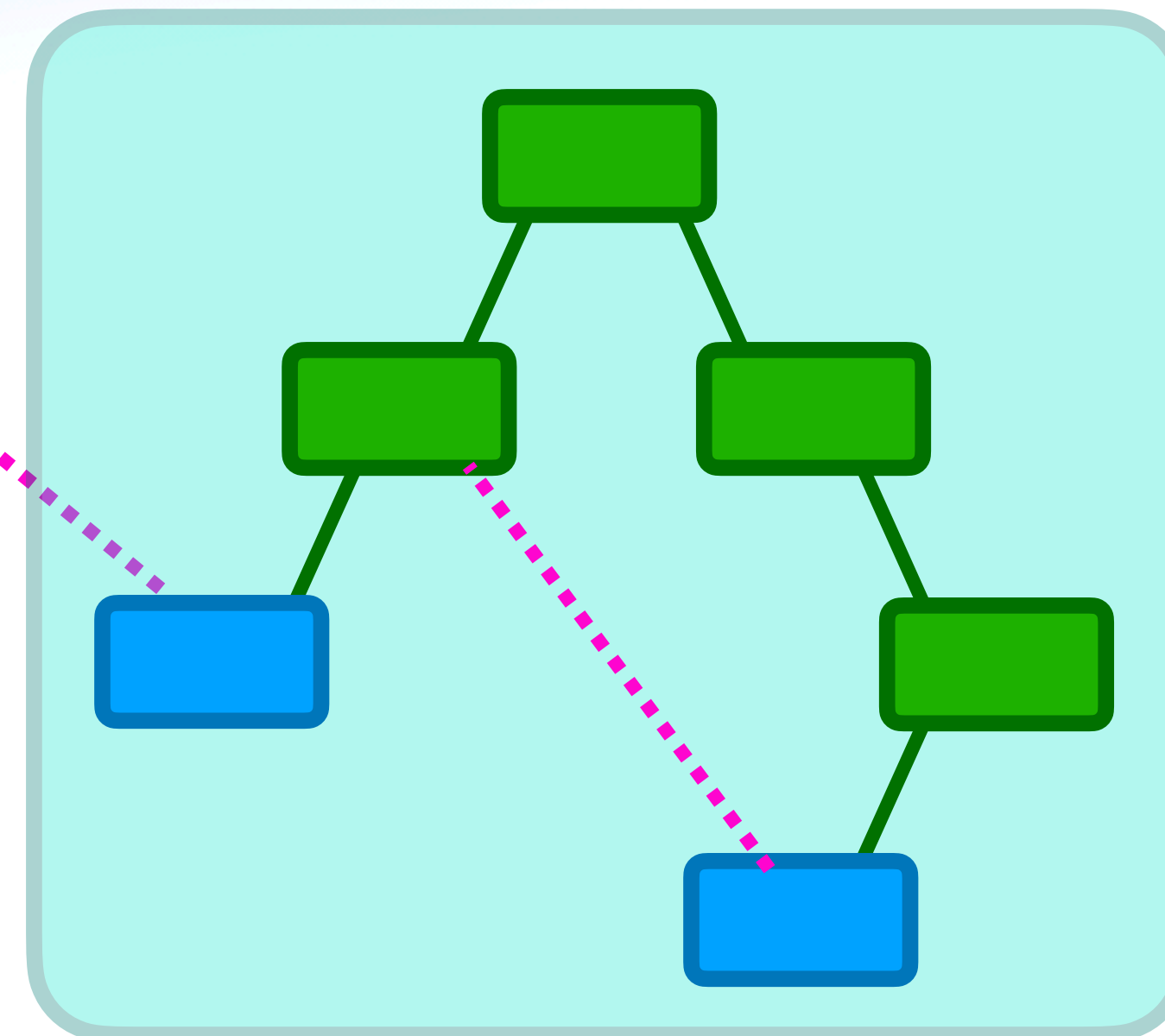
Hard Links

- ◆ New for the web!
- ◆ Direct reference
- ◆ 2 pointers ~ deduplicate

alice.fission.name



bob.fission.name



Soft Links

- ◆ Symlink or web link (URL)
- ◆ Pointer ~ latest
- ◆ Latest may break

Content Addressing

Content Addressed Storage Is Simple

```
class ContentAddressStore {
  store: {[cid: string]: ArrayBuffer}

  constructor() {
    this.store = {}
  }

  fetch(key: string): ArrayBuffer {
    return this.store[key]
  }

  async put(value: ArrayBuffer): Promise<string> {
    const digest = await crypto.subtle.digest('SHA-256', value) // Hash
    const key = btoa(String.fromCharCode(...new Uint8Array(digest))) // base64

    this.store[key] = value
    return key
  }
}
```


Content Addressing 🌐

Content Addressed Storage Is Simple

```
class ContentAddressStore {
  store: {[cid: string]: ArrayBuffer}

  constructor() {
    this.store = {}
  }

  fetch(key: string): ArrayBuffer {
    return this.store[key]
  }

  async put(value: ArrayBuffer): Promise<string> {
    const digest = await crypto.subtle.digest('SHA-256', value) // Hash
    const key = btoa(String.fromCharCode(...new Uint8Array(digest))) // base64

    this.store[key] = value
    return key
  }
}
```

Content Addressing

Content Addressed Storage Is Simple

```
class ContentAddressStore {
  store: {[cid: string]: ArrayBuffer}

  constructor() {
    this.store = {}
  }

  fetch(key: string): ArrayBuffer {
    return this.store[key]
  }

  async put(value: ArrayBuffer): Promise<string> {
    const digest = await crypto.subtle.digest('SHA-256', value) // Hash
    const key = btoa(String.fromCharCode(...new Uint8Array(digest))) // base64

    this.store[key] = value
    return key
  }
}
```


Content Addressing 🌐

Content Addressed Storage Is Simple

```
class ContentAddressStore {
  store: {[cid: string]: ArrayBuffer}

  constructor() {
    this.store = {}
  }

  fetch(key: string): ArrayBuffer {
    return this.store[key]
  }

  async put(value: ArrayBuffer): Promise<string> {
    const digest = await crypto.subtle.digest('SHA-256', value) // Hash
    const key = btoa(String.fromCharCode(...new Uint8Array(digest))) // base64

    this.store[key] = value
    return key
  }
}
```

Content Addressing

Content Addressed Storage Is Simple

```
class ContentAddressStore {
  store: {[cid: string]: ArrayBuffer}

  constructor() {
    this.store = {}
  }

  fetch(key: string): ArrayBuffer {
    return this.store[key]
  }

  async put(value: ArrayBuffer): Promise<string> {
    const digest = await crypto.subtle.digest('SHA-256', value) // Hash
    const key = btoa(String.fromCharCode(...new Uint8Array(digest))) // base64
    this.store[key] = value
    return key
  }
}
```


Content Addressing

Content Addressed Storage Is Simple

```
class ContentAddressStore {
  store: {[cid: string]: ArrayBuffer}

  constructor() {
    this.store = {}
  }

  fetch(key: string): ArrayBuffer {
    return this.store[key]
  }

  async put(value: ArrayBuffer): Promise<string> {
    const digest = await crypto.subtle.digest('SHA-256', value) // Hash
    const key = btoa(String.fromCharCode(...new Uint8Array(digest))) // base64

    this.store[key] = value
    return key
  }
}
```

Content Addressing 🌐

Content Addressed Storage Is Simple

```
class ContentAddressStore {
  store: {[cid: string]: ArrayBuffer}

  constructor() {
    this.store = {}
  }

  fetch(key: string): ArrayBuffer {
    return this.store[key]
  }

  async put(value: ArrayBuffer): Promise<string> {
    const digest = await crypto.subtle.digest('SHA-256', value) // Hash
    const key = btoa(String.fromCharCode(...new Uint8Array(digest))) // base64

    this.store[key] = value
    return key
  }
}
```


Content Addressing

Content Addressed Storage Is Simple

```
class ContentAddressStore {
  store: {[cid: string]: ArrayBuffer}

  constructor() {
    this.store = {}
  }

  fetch(key: string): ArrayBuffer {
    return this.store[key]
  }

  async put(value: ArrayBuffer): Promise<string> {
    const digest = await crypto.subtle.digest('SHA-256', value) // Hash
    const key = btoa(String.fromCharCode(...new Uint8Array(digest))) // base64

    this.store[key] = value
    return key
  }
}
```

Content Addressing

Upshot

- Names that leak no additional info
- A consistent way to reference data
- Self-verifying data for low/no-trust environment
- Hard links on the web

Private Data Substrate

Distributed Storage When The Pipes Are Broken

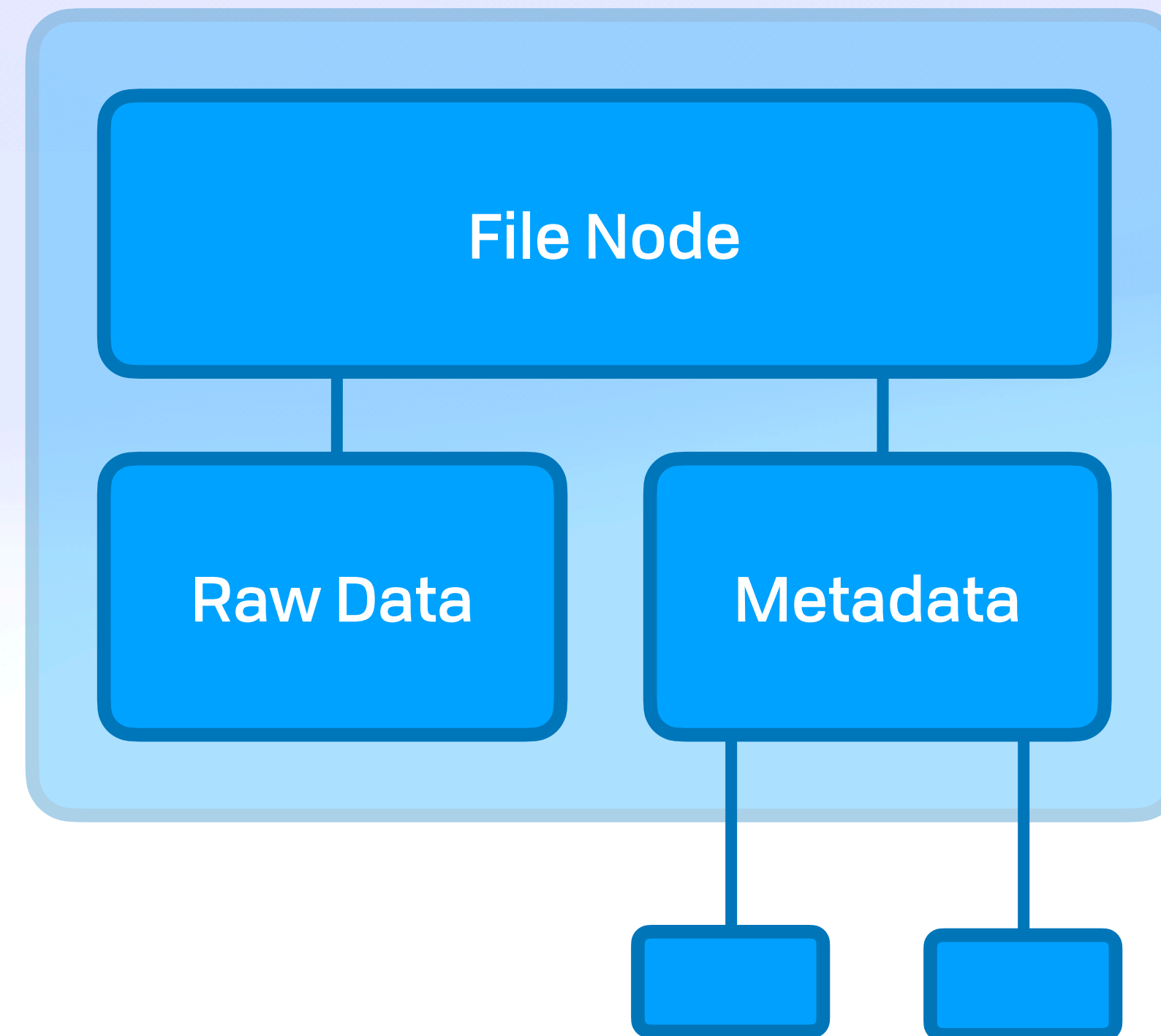


Private Data Substrate 

Virtual Files & Directories

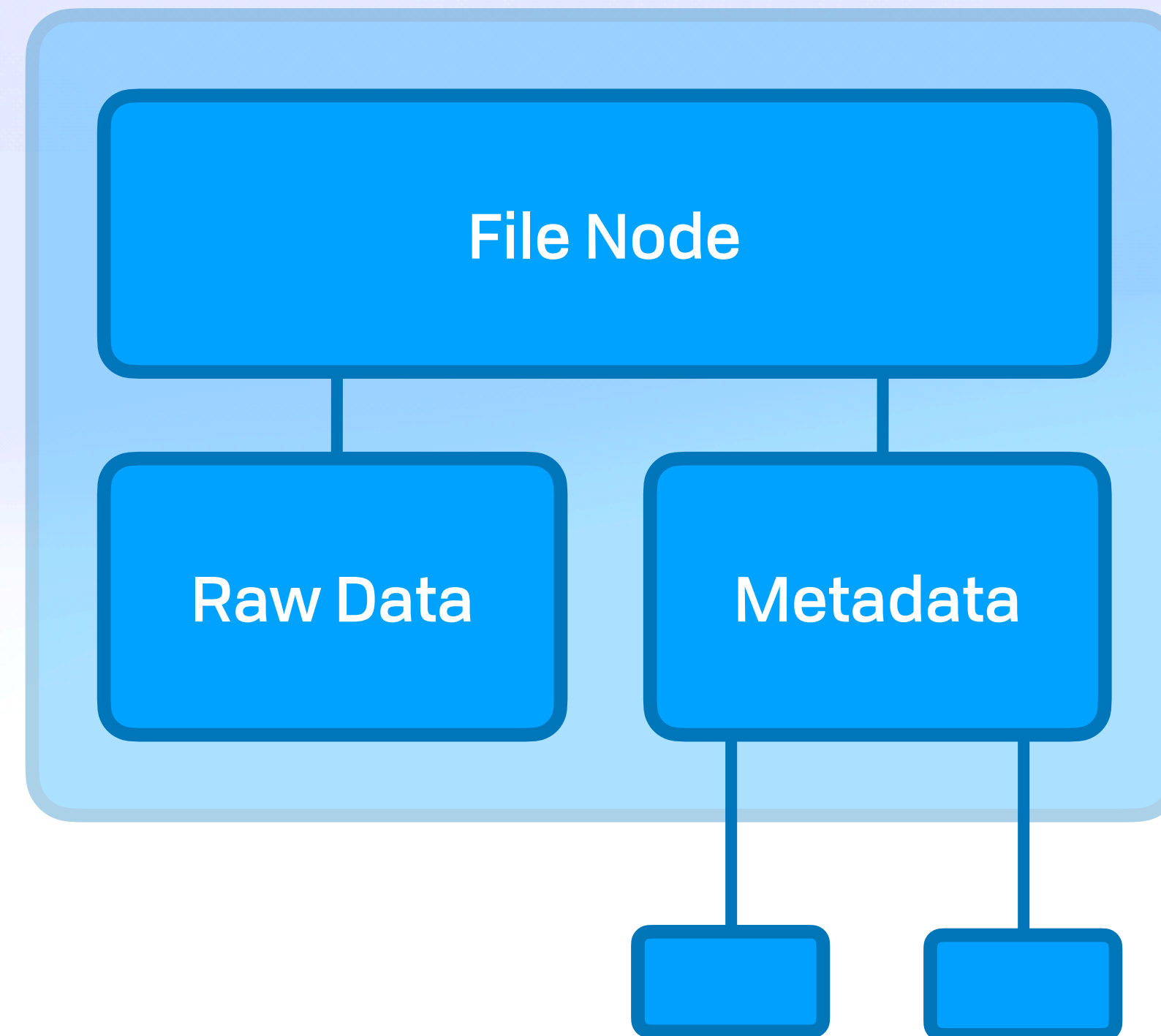
Private Data Substrate 

Virtual Files & Directories



Private Data Substrate 

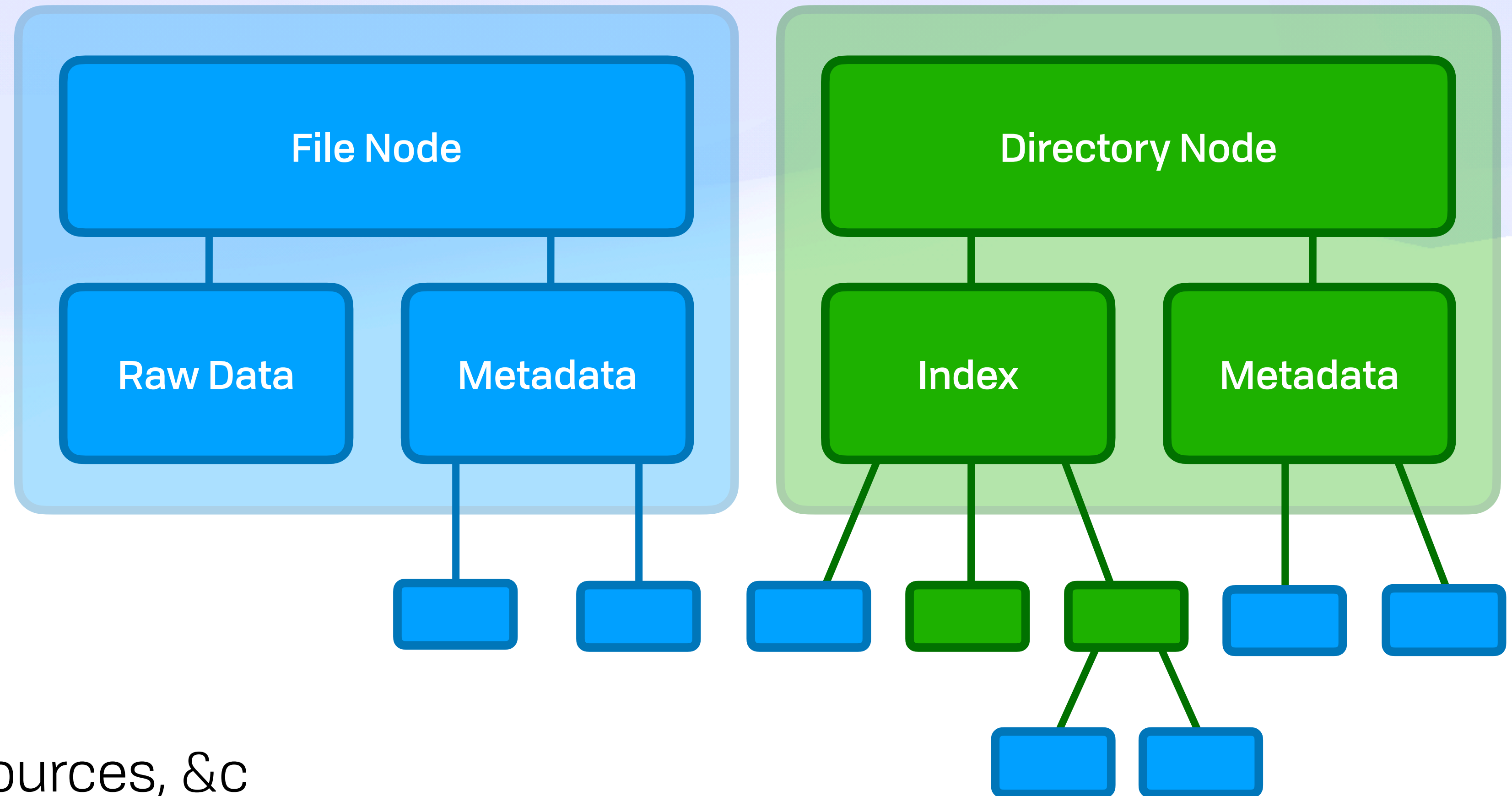
Virtual Files & Directories



- ◆ Virtual Node
 - ◆ Consistent interface
- ◆ Arbitrary metadata
 - ◆ Tags, creators, MIME, sources, &c

Virtual Files & Directories

- ◆ Virtual Node
 - ◆ Consistent interface
- ◆ Arbitrary metadata
 - ◆ Tags, creators, MIME, sources, &c

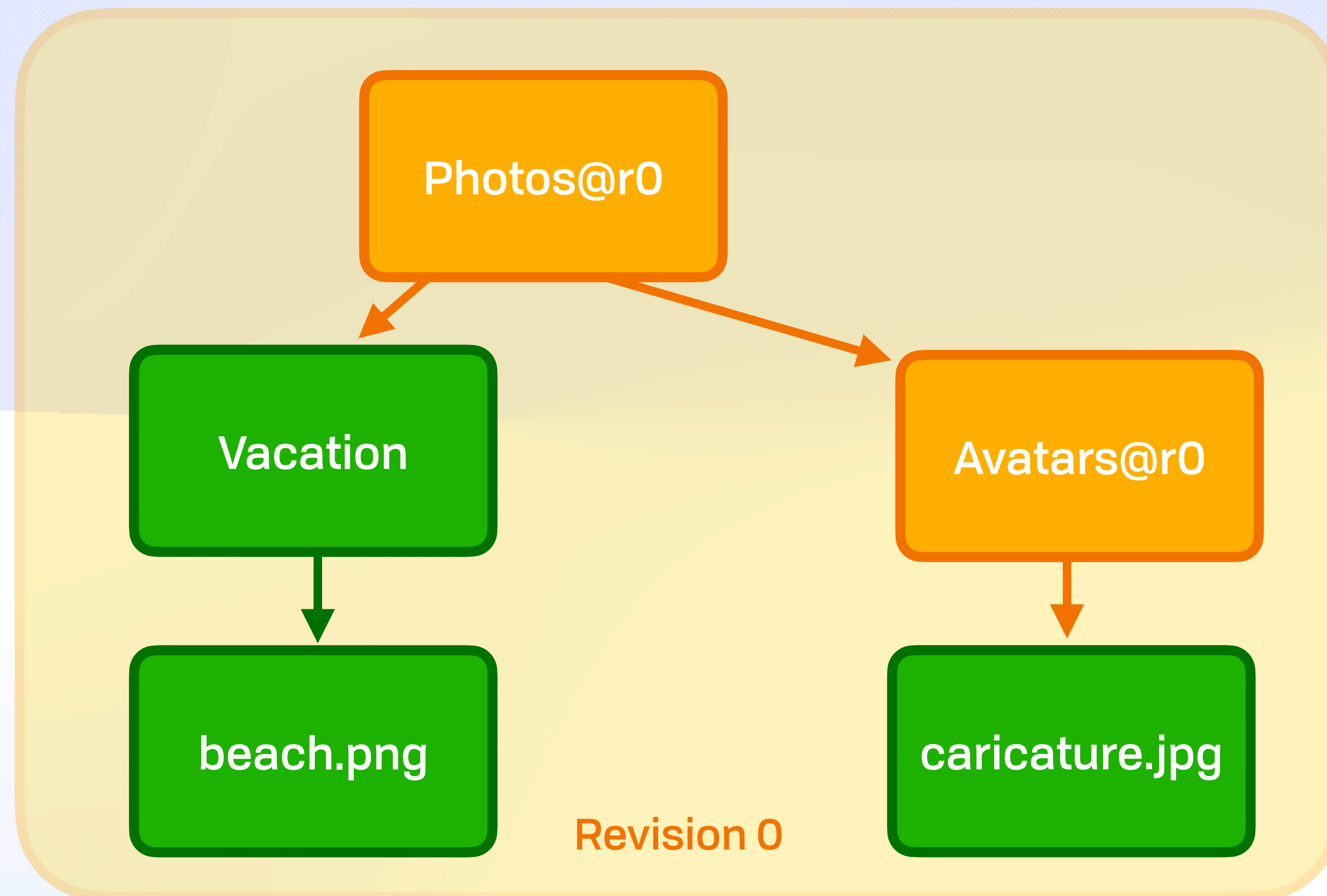


Private Data Substrate 

Persistent Versioning

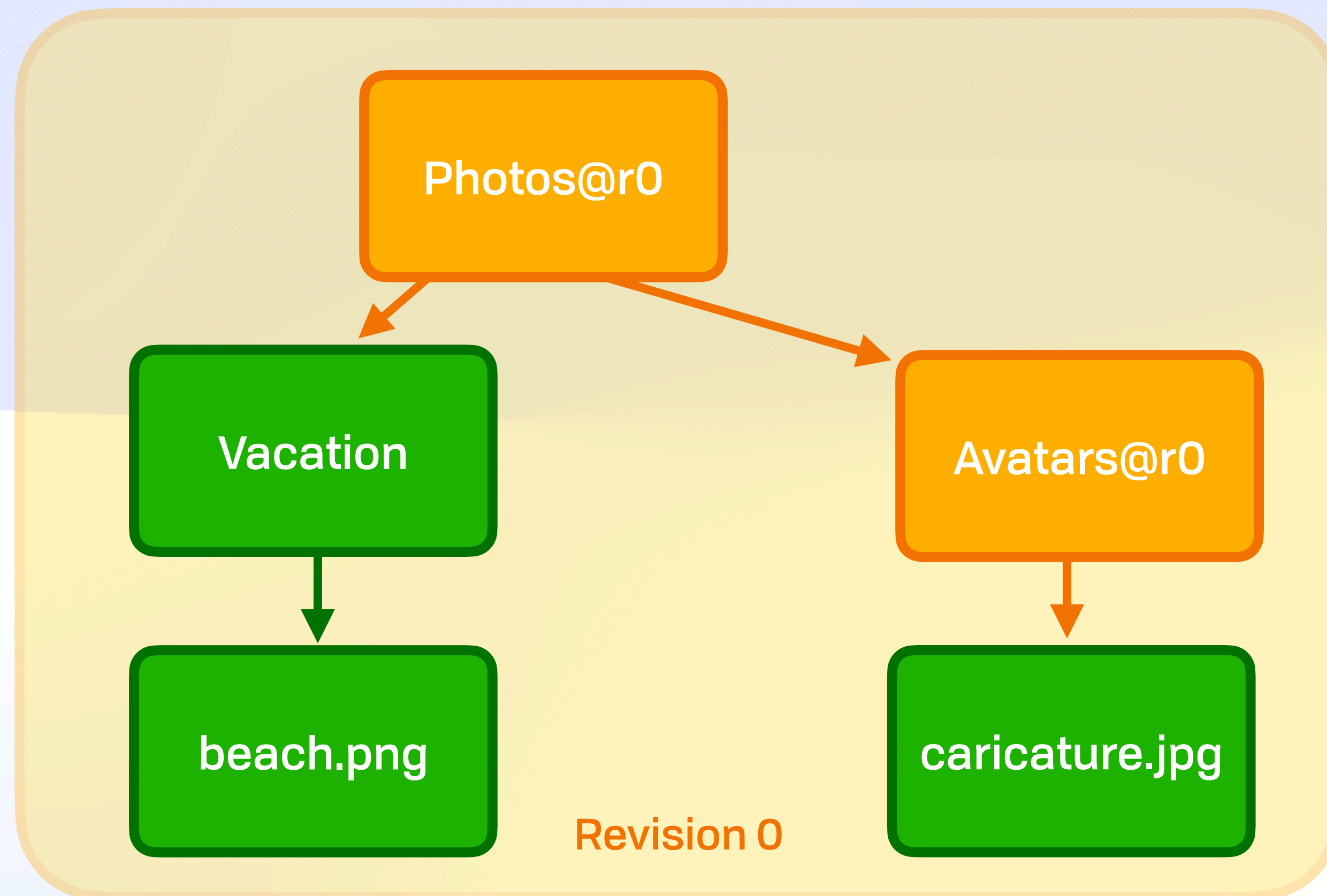
Private Data Substrate 

Persistent Versioning

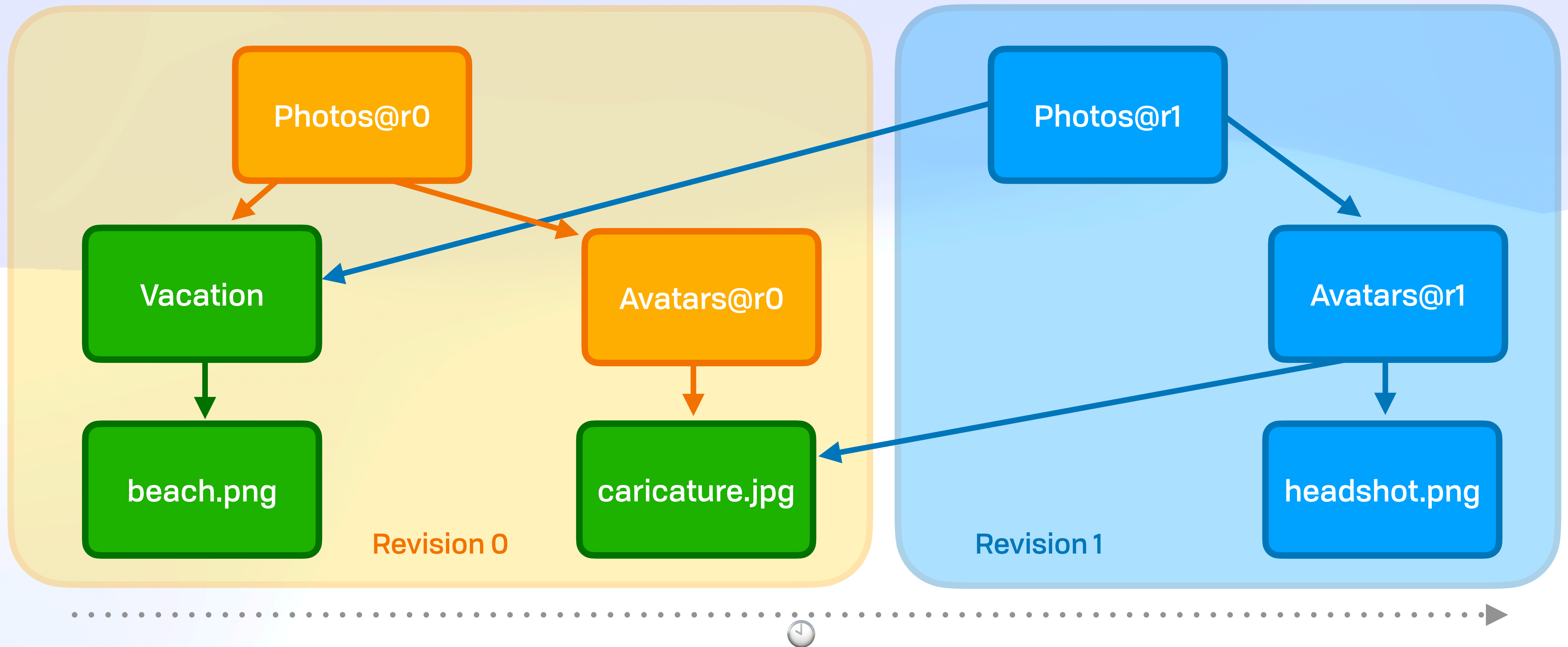


Private Data Substrate 

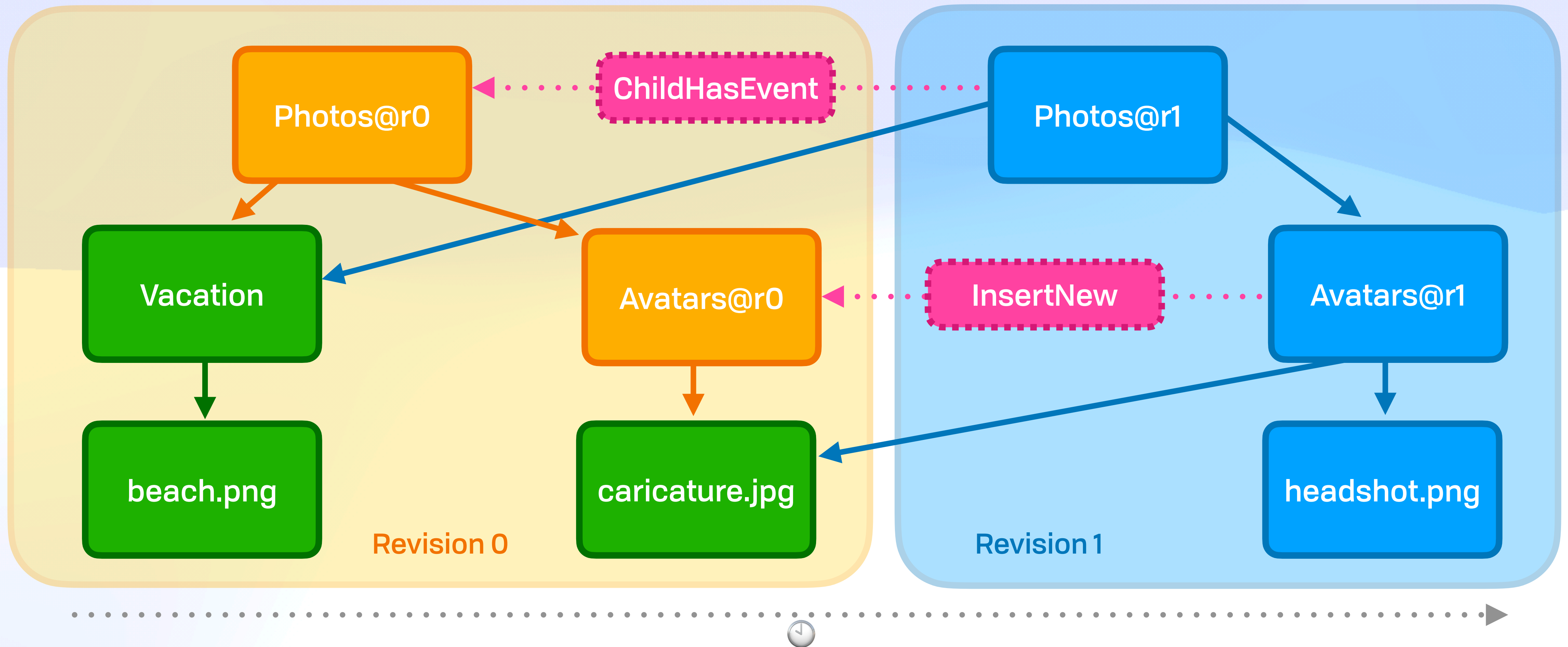
Persistent Versioning



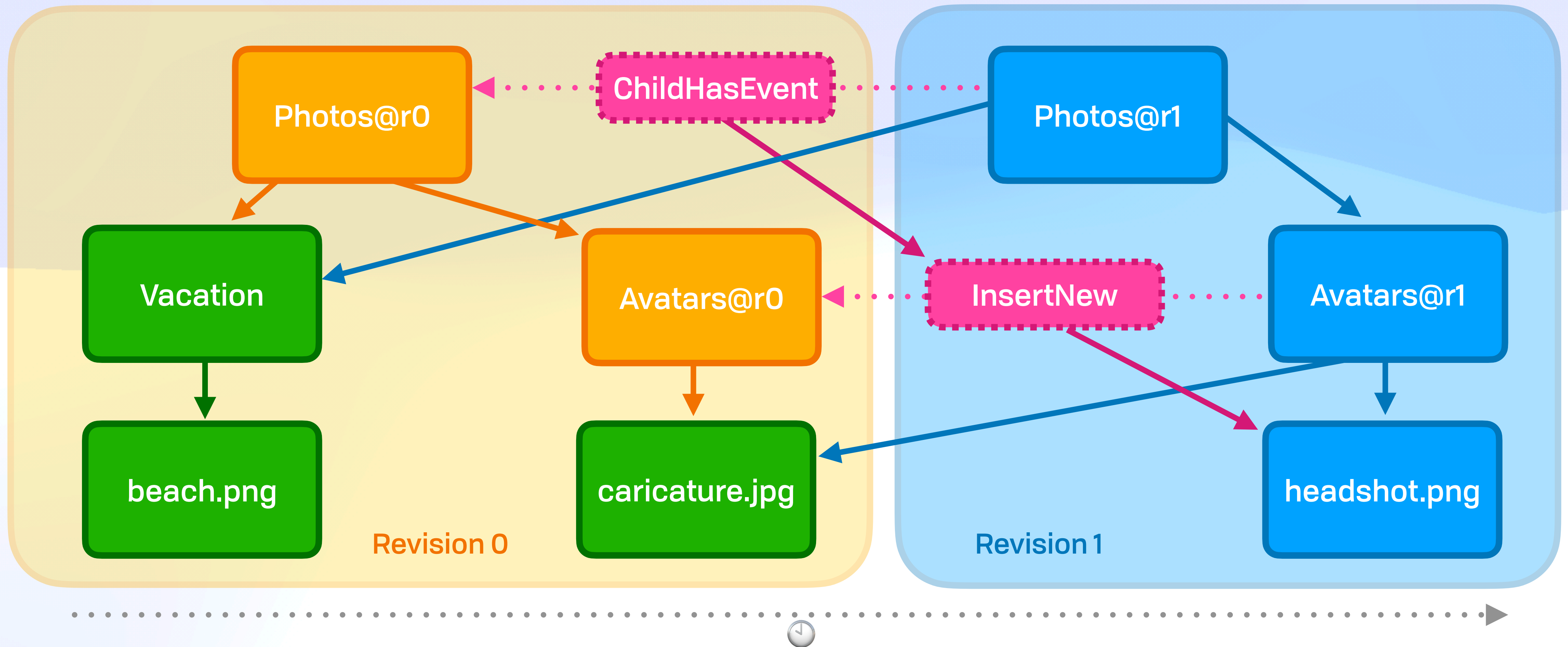
Persistent Versioning




Persistent Versioning



Persistent Versioning



Private Data Substrate 


It's All Just Maps

```
{
  kernel: {
    ty: FILE,
    history: [
      {cid: "bafkreidrgwjljxy6s7o5uvrifxnwefggi7chmye3pn6wyisv2n4b3uordi", event: UPDATE},
      {cid: "bafkreicushgsax35ihrwo6smk4qmjuwzbf7qzqg6o1qxyaisuxtgpvxrim", event: REPLACE}
    ]
  },
  metadata: {
    utime: 1661837091,
    author: "did:key:StEZpzSMtTt9k2vszgvCwF4fLQQSyA15W5AQ4z3AR6Bx4eFJ5crJFbuGxKmbma4"
  },
  content: "HwRZys6pMUjvyS4pYKSS4Et33jDB+kwvCuMJu4GeNCan1eAmhPykHFR+gc2D+Qa/..."
}
```


Private Data Substrate

It's All Just Maps

```
{
  kernel: {
    ty: FILE,
    history: [
      {cid: "bafkreidrgwjljxy6s7o5uvrifxnwefggi7chmye3pn6wyisv2n4b3uordi", event: UPDATE},
      {cid: "bafkreicushgsax35ihrwo6smk4qmjuwzbf7qzqg6o1qxyaisuxtgpvxrim", event: REPLACE}
    ]
  },
  metadata: {
    utime: 1661837091,
    author: "did:key:StEZpzSMtTt9k2vszgvCwF4fLQQSyA15W5AQ4z3AR6Bx4eFJ5crJFbuGxKmbma4"
  },
  content: "HwRZys6pMUjvyS4pYKSS4Et33jDB+kwvCuMJu4GeNCan1eAmhPykHFR+gc2D+Qa/..."
}
```


Private Data Substrate 

It's All Just Maps

```
{
  kernel: {
    ty: FILE,
    history: [
      {cid: "bafkreidrgwjljxy6s7o5uvrifxnwefggi7chmye3pn6wyisv2n4b3uordi", event: UPDATE},
      {cid: "bafkreicushgsax35ihrwo6smk4qmjuwzbf7qzqg6o1qxyaisuxtgpvxrim", event: REPLACE}
    ]
  },
  metadata: {
    utime: 1661837091,
    author: "did:key:StEZpzSMtTt9k2vszgvCwF4fLQQSyA15W5AQ4z3AR6Bx4eFJ5crJFbuGxKmbma4"
  },
  content: "HwRZys6pMUjvyS4pYKSS4Et33jDB+kwvCuMJu4GeNCan1eAmhPykHFR+gc2D+Qa/..."
}
```


Private Data Substrate

It's All Just Maps

```
{
  kernel: {
    ty: FILE,
    history: [
      {cid: "bafkreidrgwj\ljxy6s7o5uvrifxnwefggi7chmye3pn6wyisv2n4b3uordi", event: UPDATE},
      {cid: "bafkreicushgsax35ihrwo6smk4qmjuwzbf7qzqg6o\lqxyaisuxtgpvxrim", event: REPLACE}
    ]
  },
  metadata: {
    utime: 1661837091,
    author: "did:key:StEZpzSMtTt9k2vszgvCwF4fLQQSyA15W5AQ4z3AR6Bx4eFJ5crJFbuGxKmbma4"
  },
  content: "HwRZys6pMUjvyS4pYKSS4Et33jDB+kwvCuMJu4GeNCan1eAmhPykHFR+gc2D+Qa/..."
}
```


Private Data Substrate

It's All Just Maps

```
{
  kernel: {
    ty: FILE,
    history: [
      {cid: "bafkreidrgwjljxy6s7o5uvrifxnwefggi7chmye3pn6wyisv2n4b3uordi", event: UPDATE},
      {cid: "bafkreicushgsax35ihrwo6smk4qmjuwzbf7qzqg6o1qxyaisuxtgpixrim", event: REPLACE}
    ]
  },
  metadata: {
    utime: 1661837091,
    author: "did:key:StEZpzSMtTt9k2vszgvCwF4fLQQSyA15W5AQ4z3AR6Bx4eFJ5crJFbuGxKmbma4"
  },
  content: "HwRZys6pMUjvyS4pYKSS4Et33jDB+kvvCuMJu4GeNCan1eAmhPykHFR+gc2D+Qa/..."
}
```


Private Data Substrate

It's All Just Maps

```
{
  kernel: {
    ty: FILE,
    history: [
      {cid: "bafkreidrgwjljxy6s7o5uvrifxnwefggi7chmye3pn6wyisv2n4b3uordi", event: UPDATE},
      {cid: "bafkreicushgsax35ihrwo6smk4qmjuwzbf7qzqg6o1qxyaisuxtgpvxrim", event: REPLACE}
    ]
  },
  metadata: {
    utime: 1661837091,
    author: "did:key:StEZpzSMtTt9k2vszgvCwF4fLQQSyA15W5AQ4z3AR6Bx4eFJ5crJFbuGxKmbma4"
  },
  content: "HwRZys6pMUjvyS4pYKSS4Et33jDB+kwvCuMJu4GeNCan1eAmhPykHFR+gc2D+Qa/..."
}
```


Private Data Substrate

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

Dr. Lea Kissner, Google's Global Lead of Privacy Technologies

Private Data Substrate

SubtleCrypto (WebCrypto)

Web Crypto API

The **Web Crypto API** is an interface allowing a script to use cryptographic primitives in order to build systems using cryptography.

Note: This feature is available in [Web Workers](#)

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	K Br
		3.1-6.1					3.2-6.1									
4-36		7-10.1	2-33	10-23			7-10.3	4-5.4								
37-103	12-103	11-15.5	34-103	24-89	6-10		11-15.5	6.2-17.0		12-12.1		2.1-4.4.4				
104	104	15.6	104	90	11	104	15.6	18.0	all	64	12.12	104	101	10.4	7.12	
105-107		16.0-TP	105-106				16.0									

Source: caniuse.com

Private Data Substrate 

Non-Extractable Keys

Private Data Substrate 

Non-Extractable Keys



Private Data Substrate 

Non-Extractable Keys



Private Data Substrate 

Non-Extractable Keys



Private Data Substrate 

Non-Extractable Keys



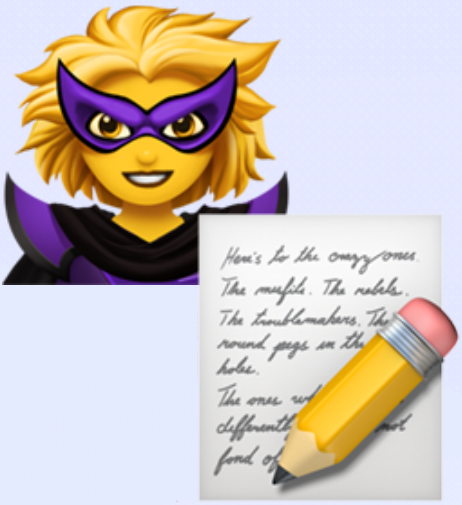
Private Data Substrate 

Non-Extractable Keys



Private Data Substrate 

Non-Extractable Keys



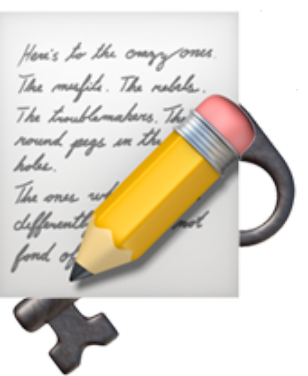
Private Data Substrate 

Non-Extractable Keys



Private Data Substrate 

Non-Extractable Keys



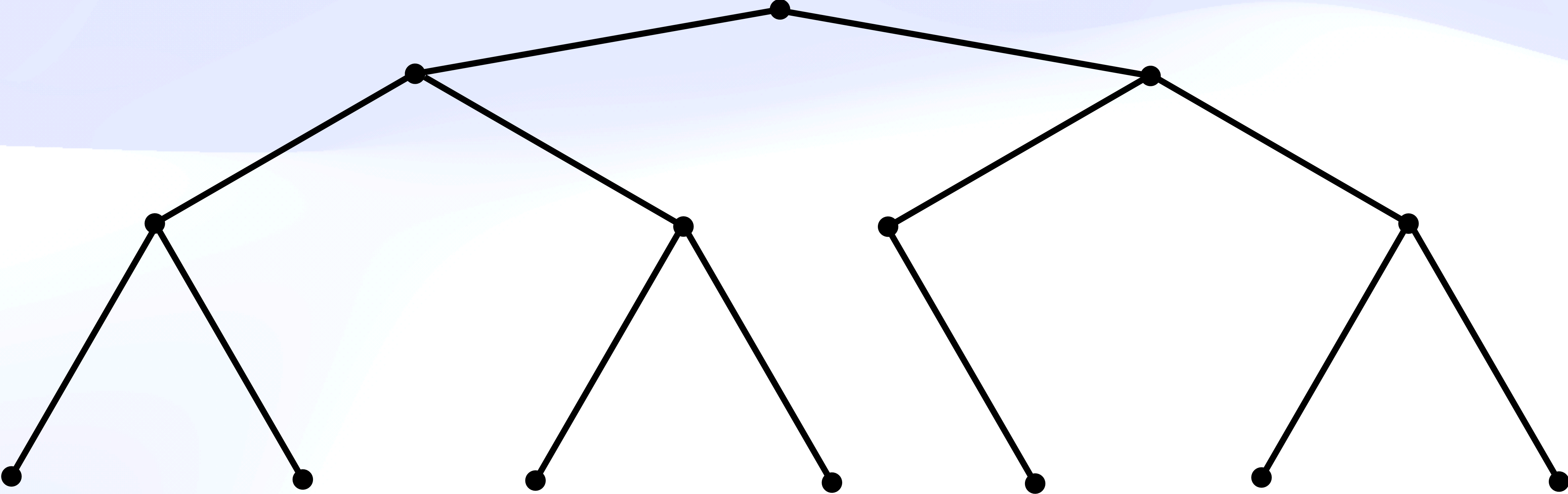
Private Data Substrate 

Non-Extractable Keys



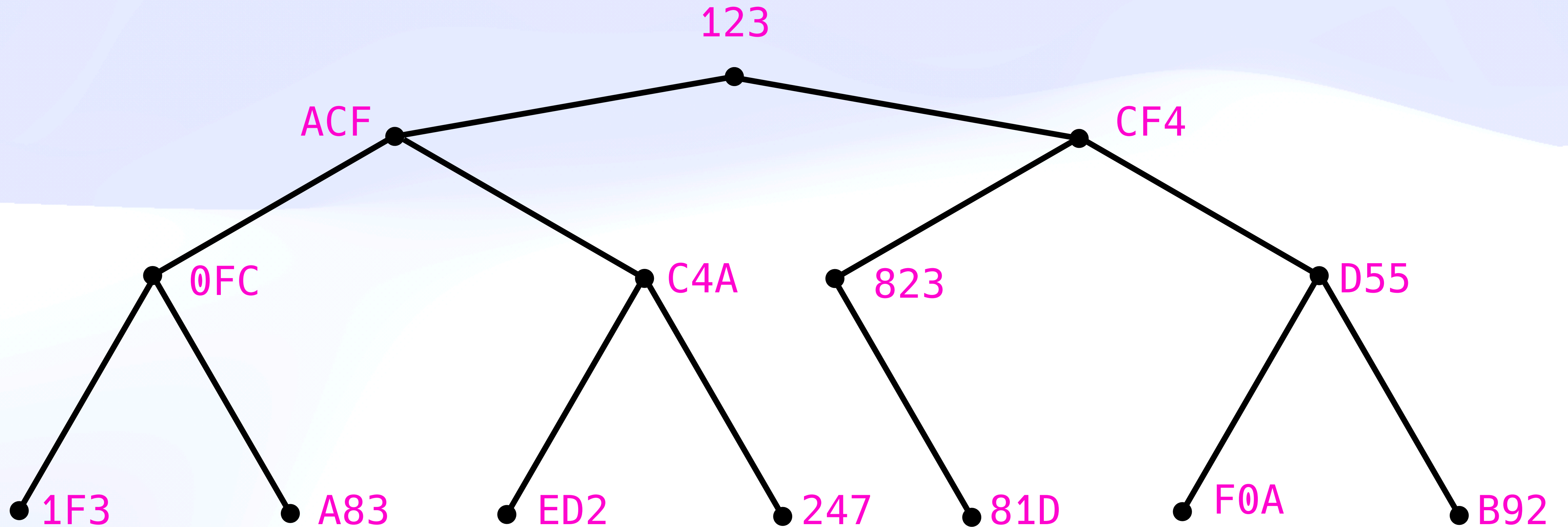
Private Data Substrate 


Self-Authenticating & Self-Authorizing



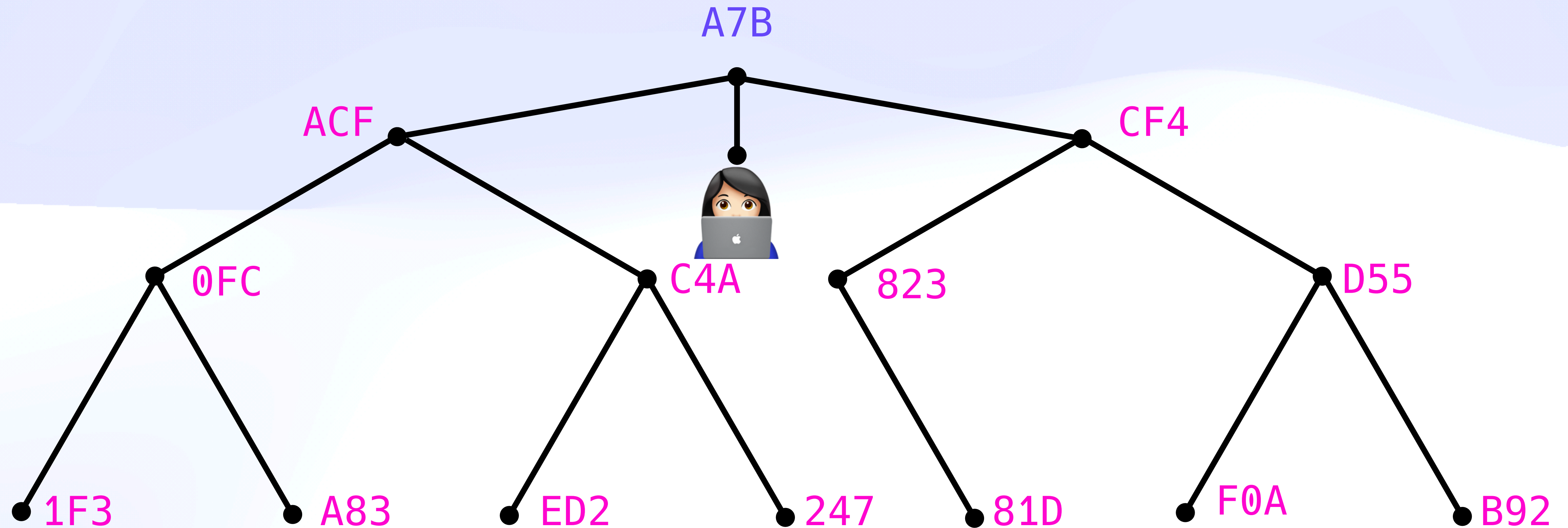
Private Data Substrate 

Self-Authenticating & Self-Authorizing



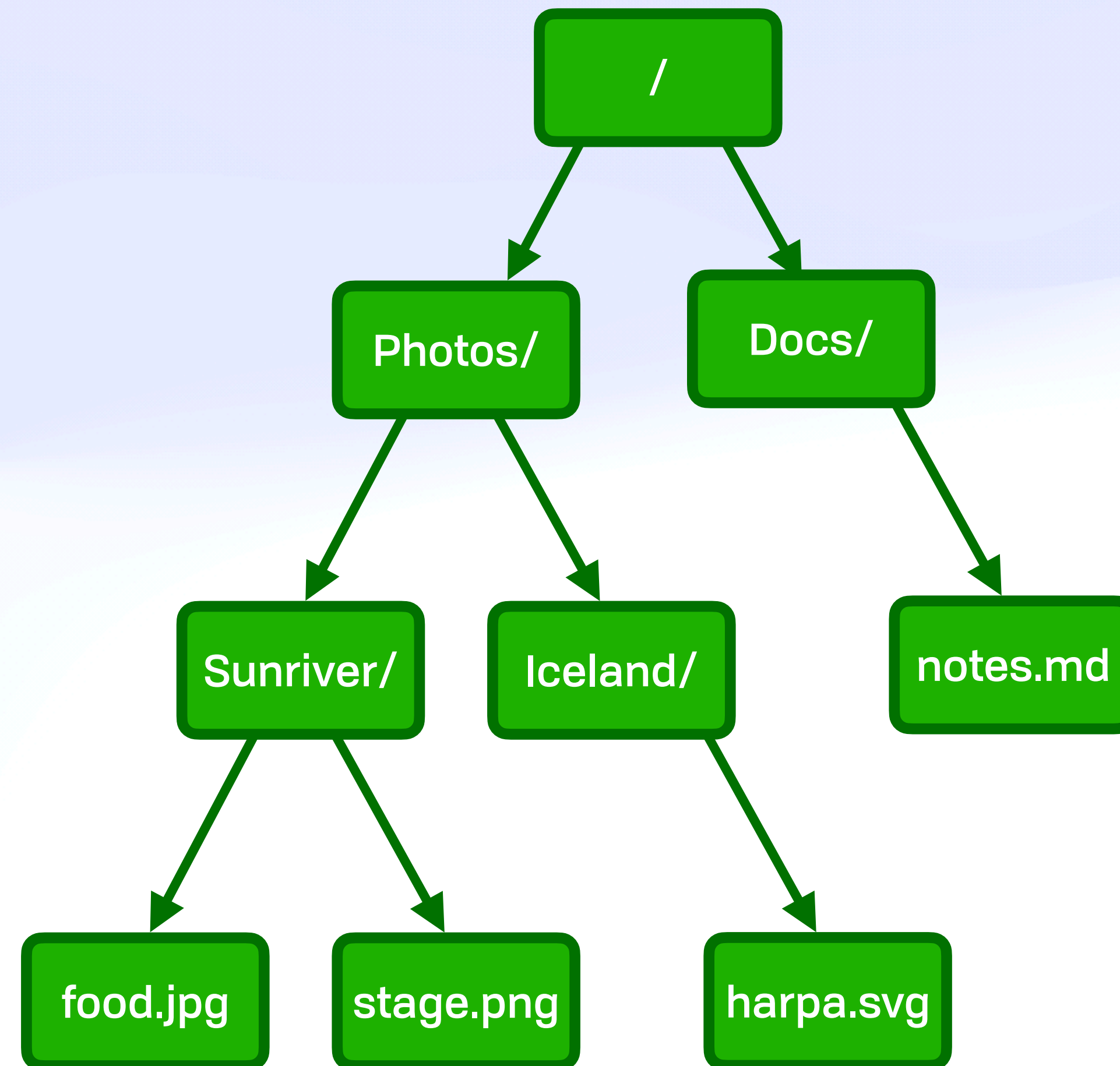
Private Data Substrate 

Self-Authenticating & Self-Authorizing



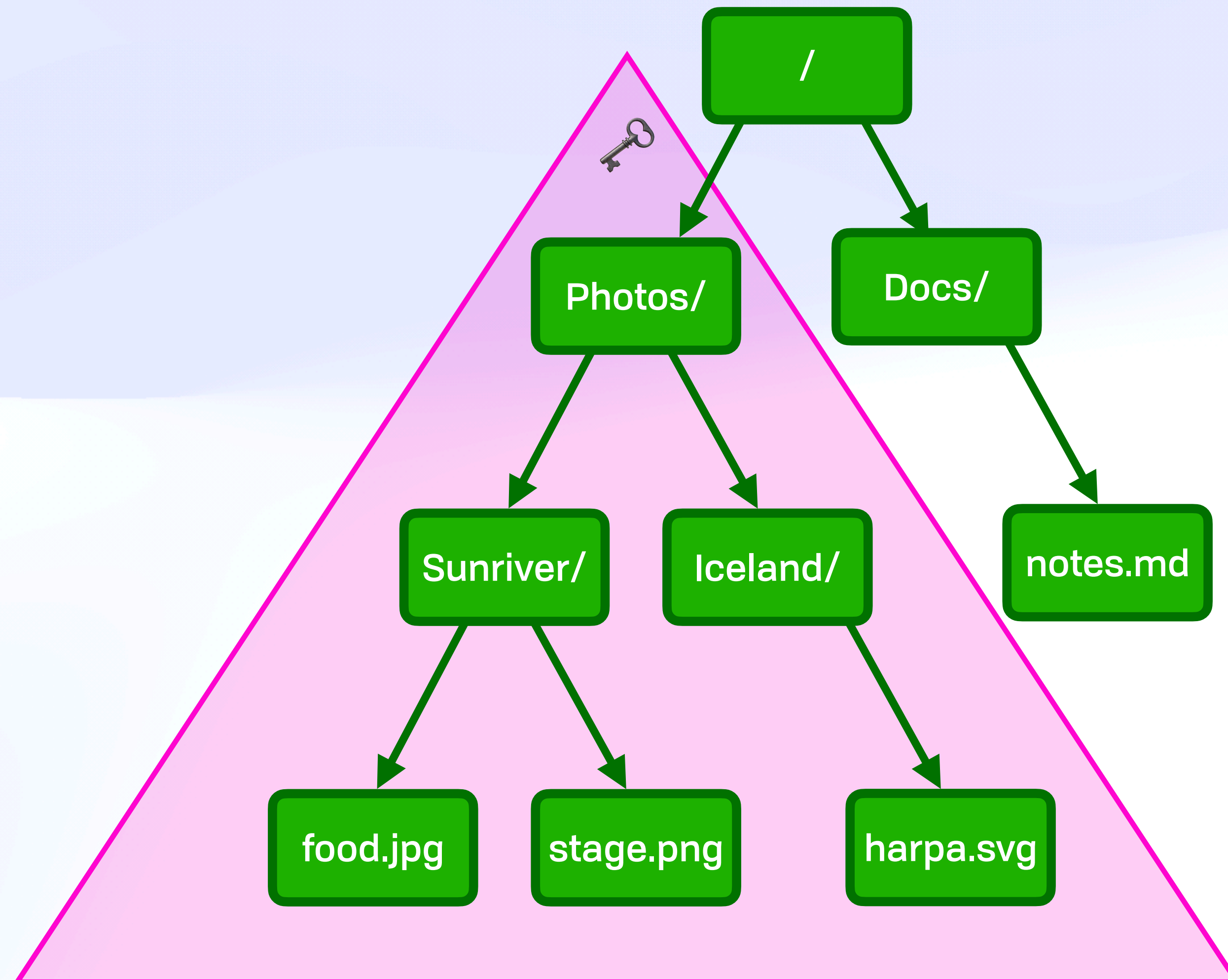
Private Data Substrate 

Offline Access Control



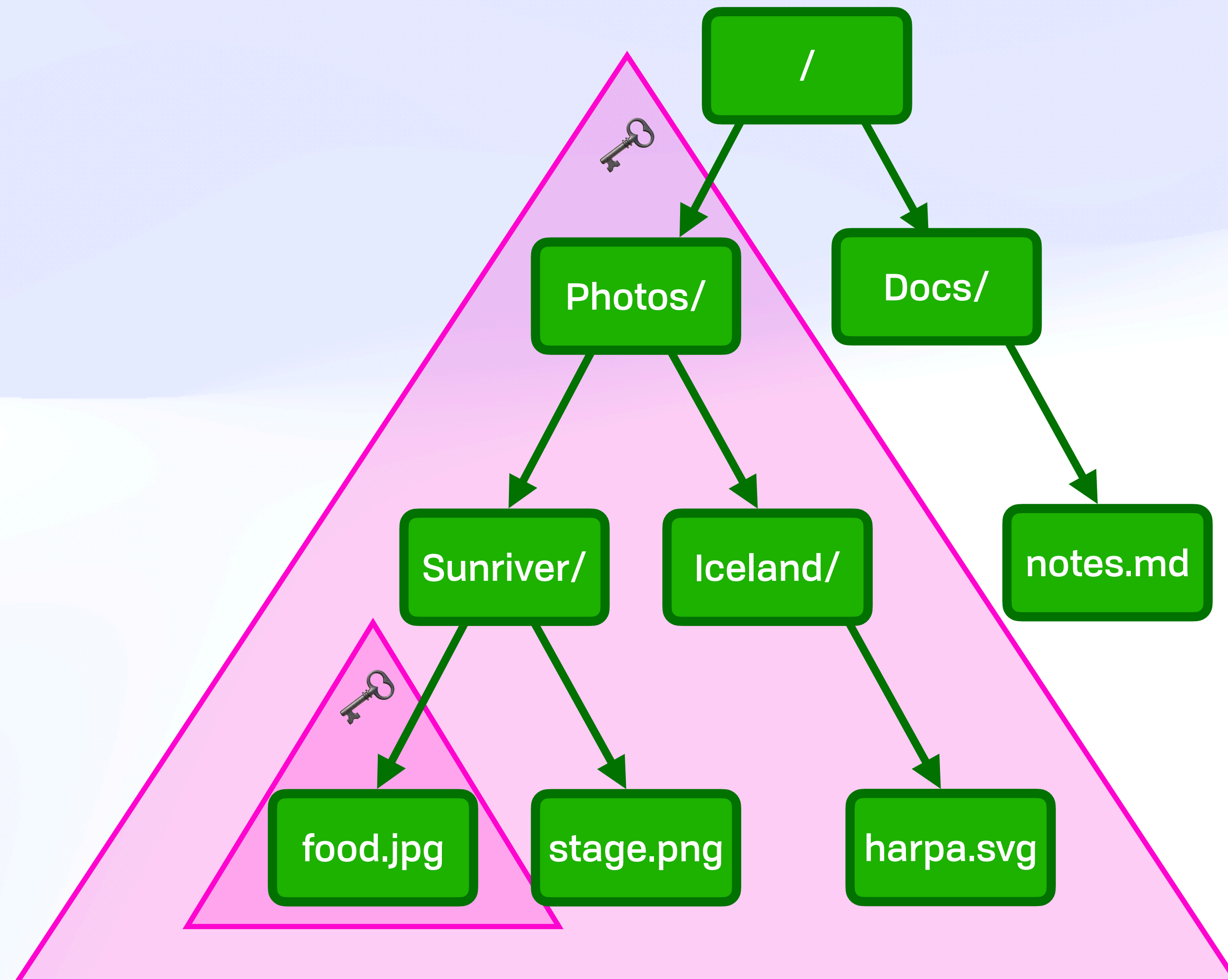
Private Data Substrate 

Offline Access Control



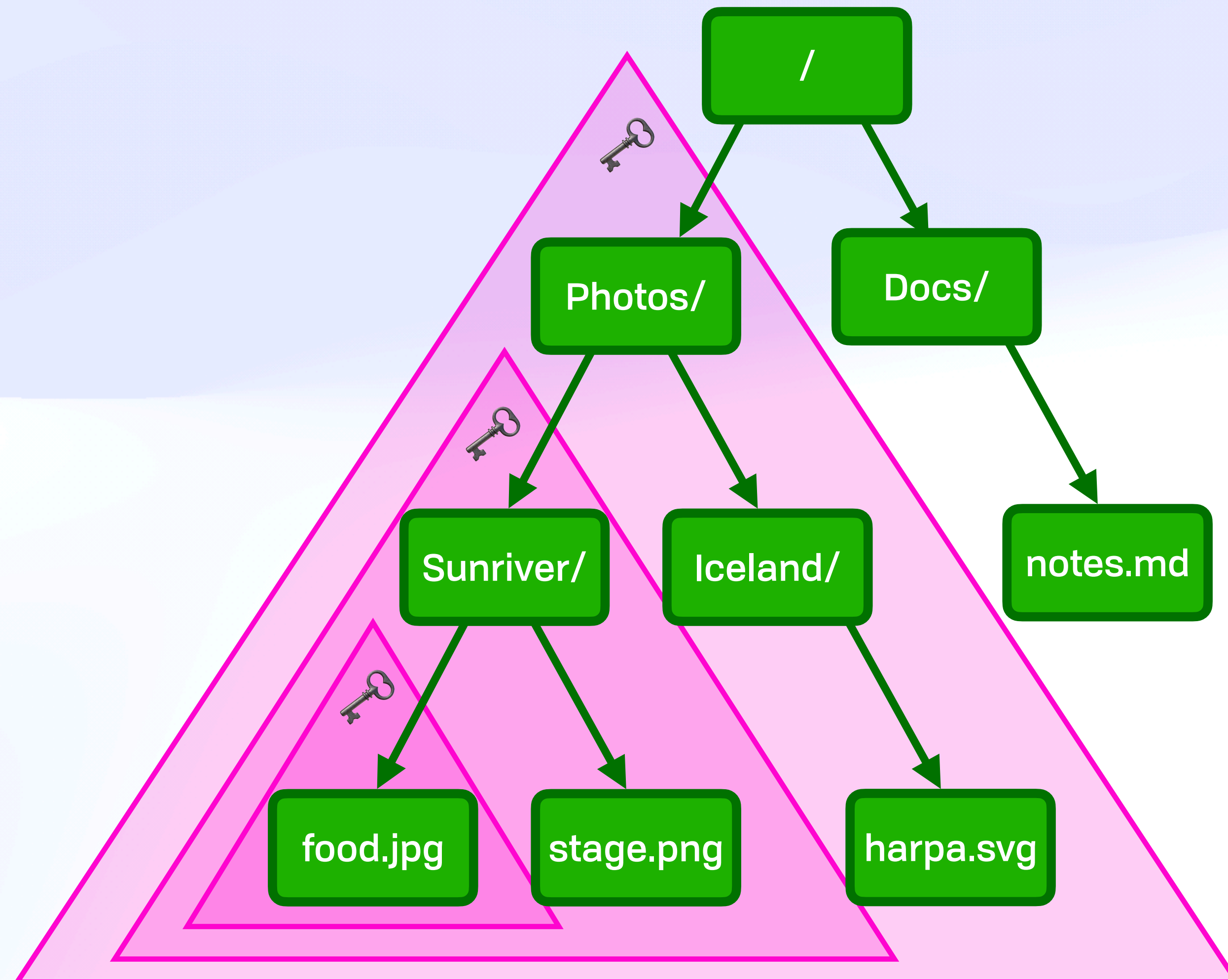
Private Data Substrate 

Offline Access Control



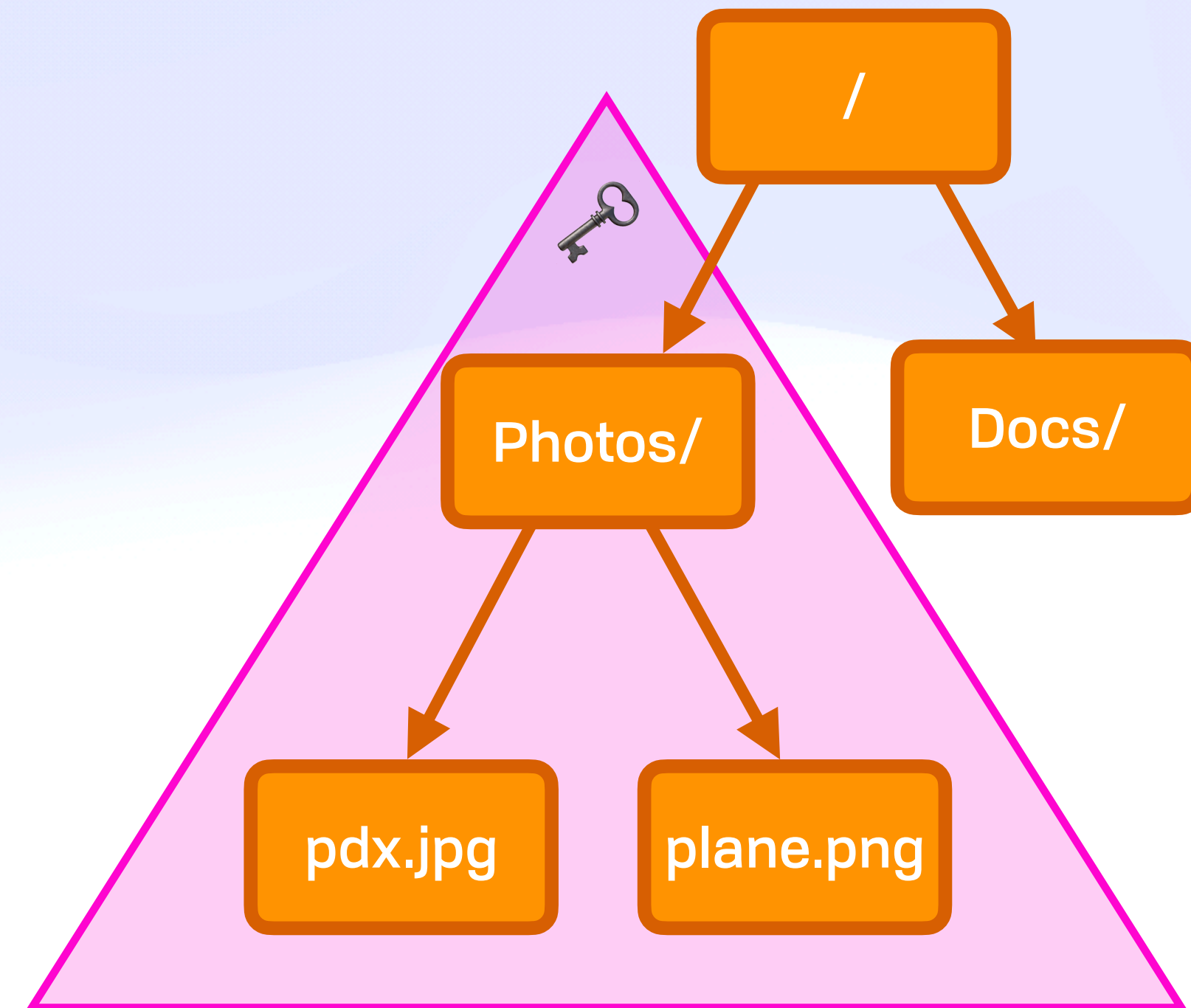
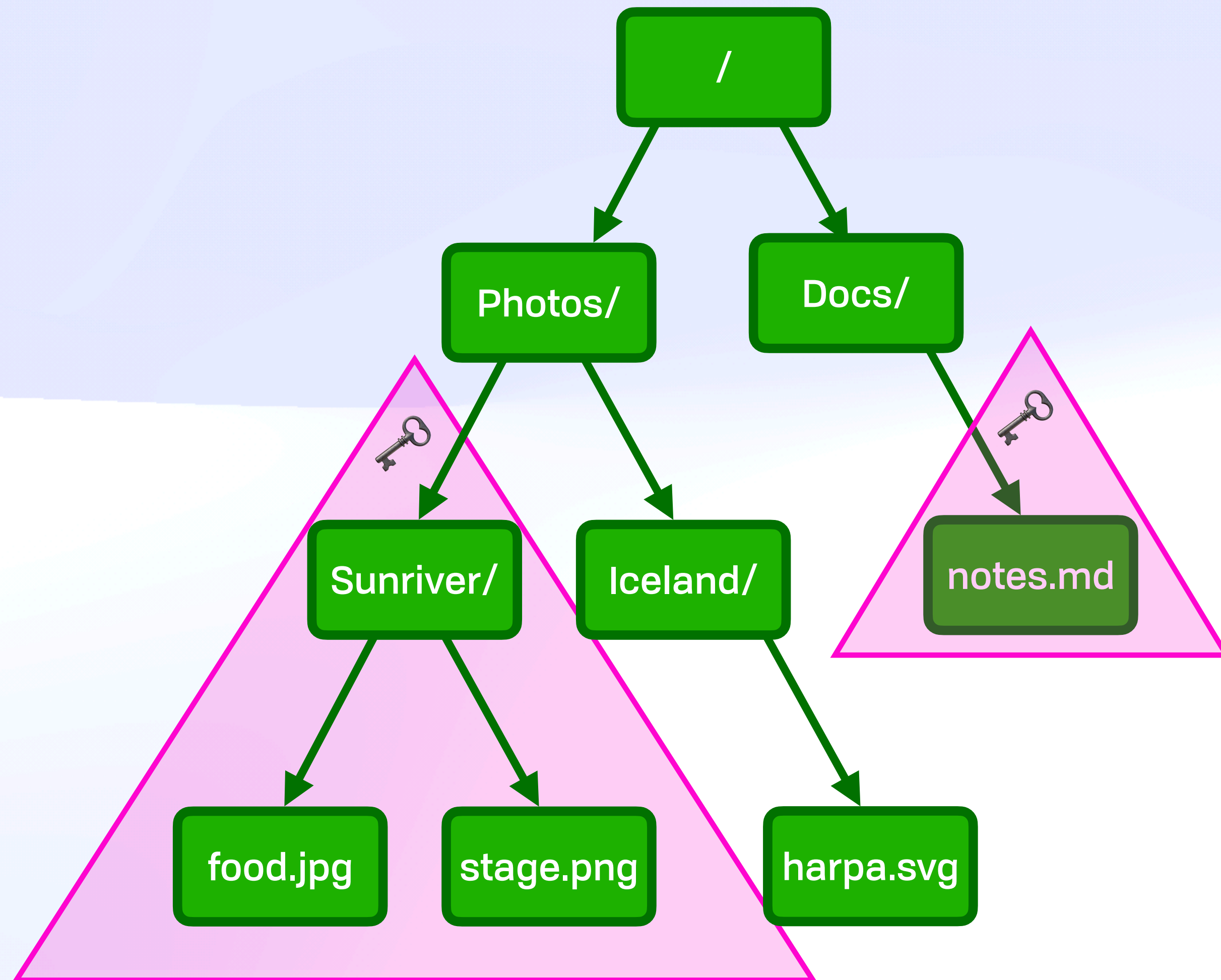
Private Data Substrate 

Offline Access Control



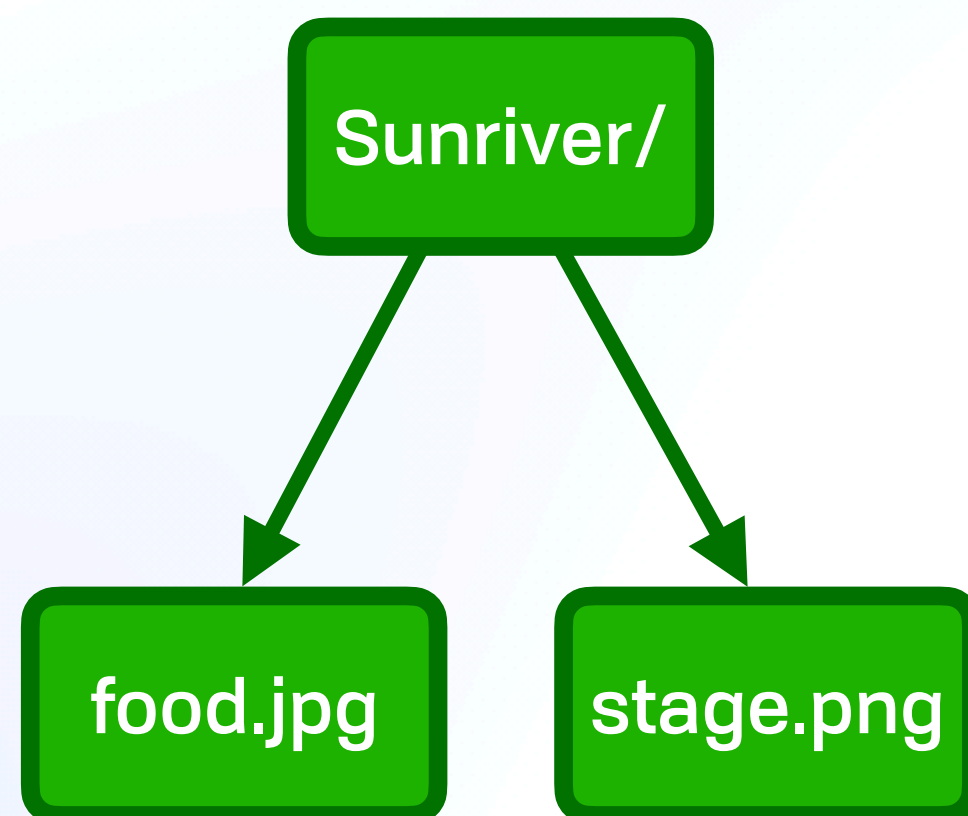
Private Data Substrate 

Offline Access Control

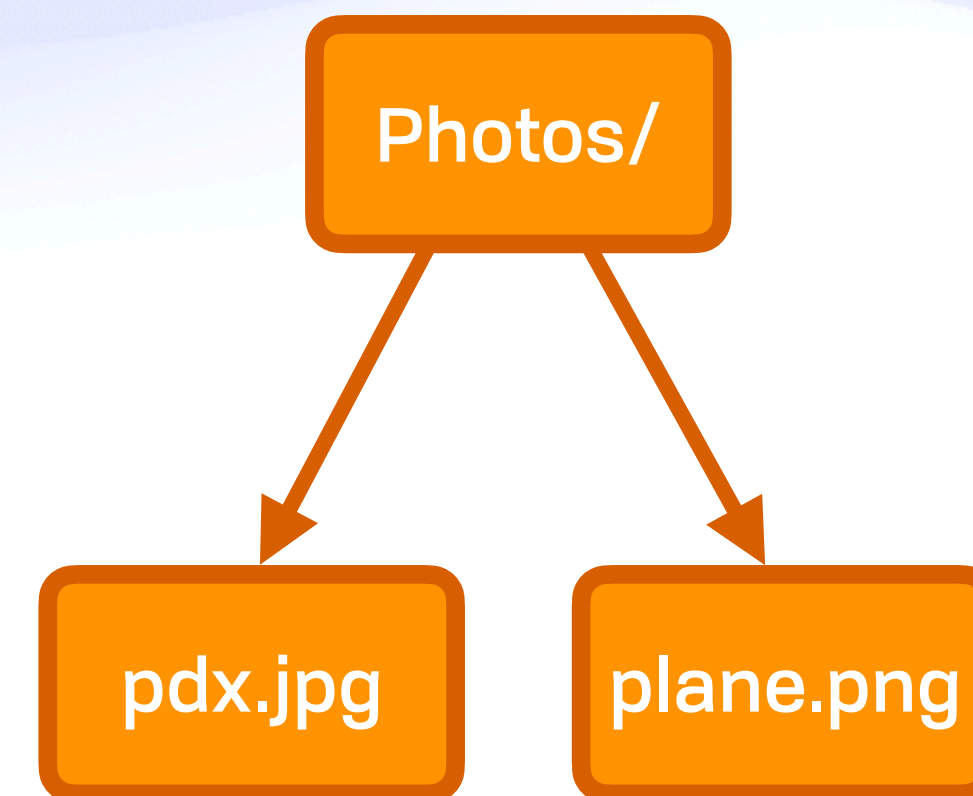


Private Data Substrate 

Offline Access Control

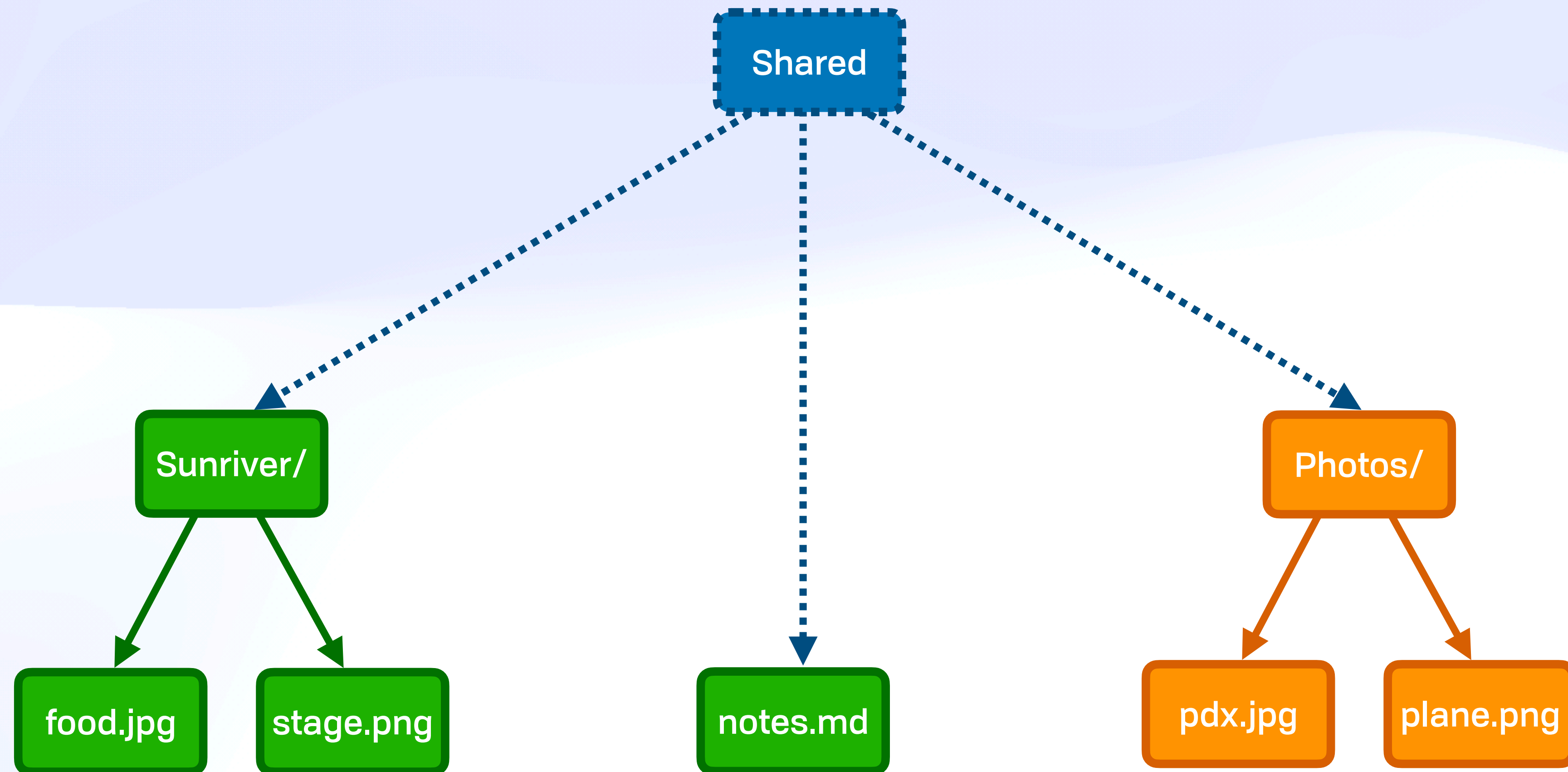


notes.md



Private Data Substrate 

Offline Access Control



Private Data Substrate 

Private Nodes

Private Data Substrate 

Private Nodes

Binary

Encrypted Node 

Encrypted Node 

Encrypted Node 

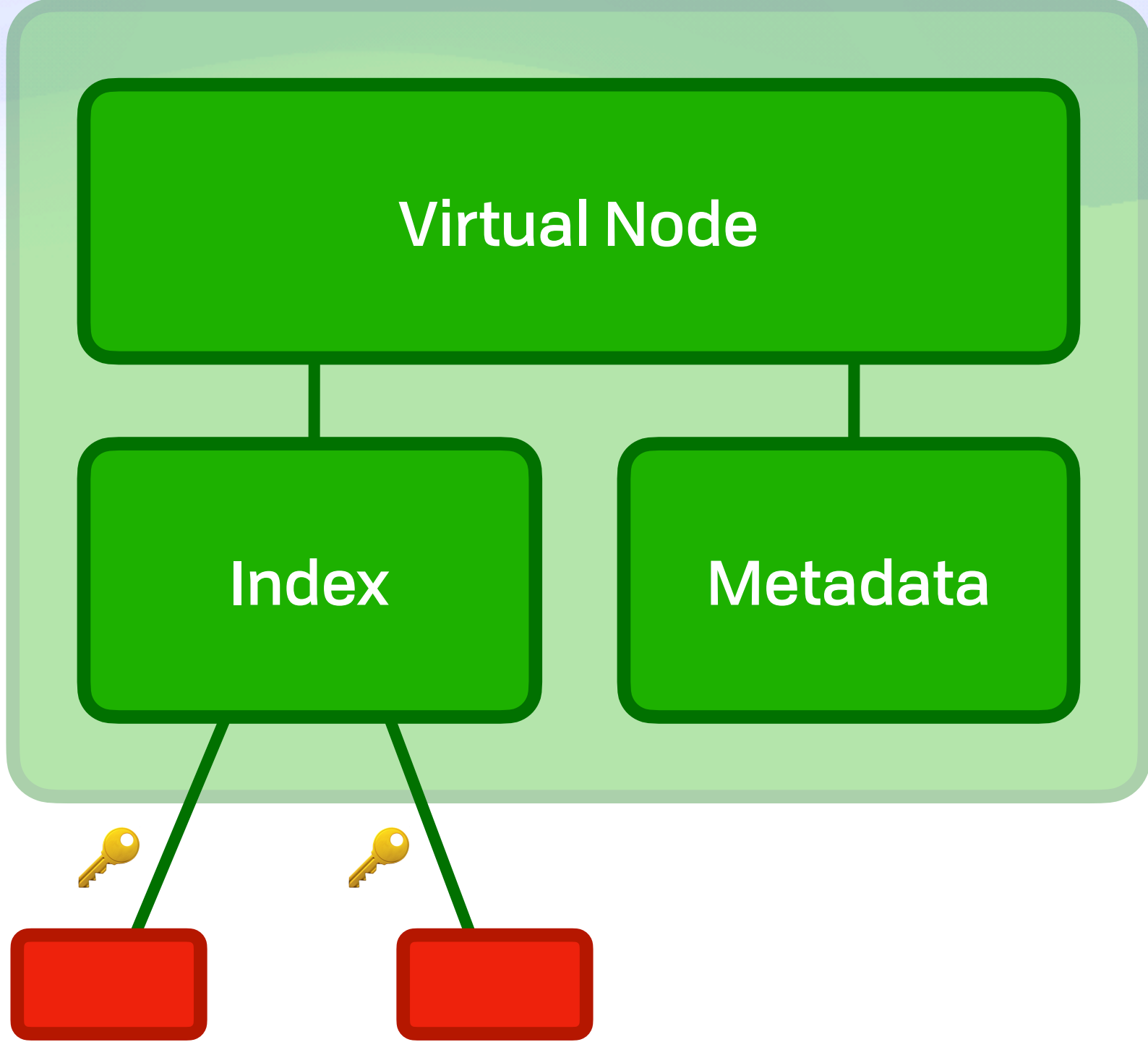
Private Data Substrate

Private Nodes

Binary

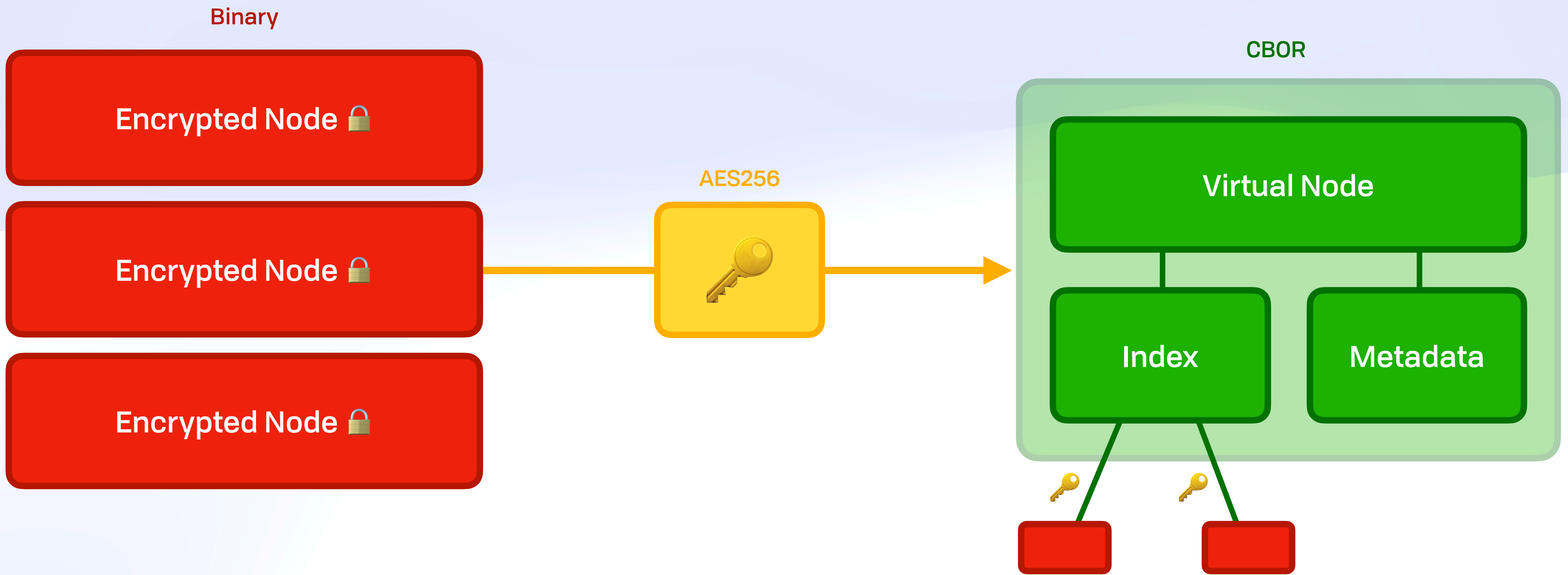


CBOR



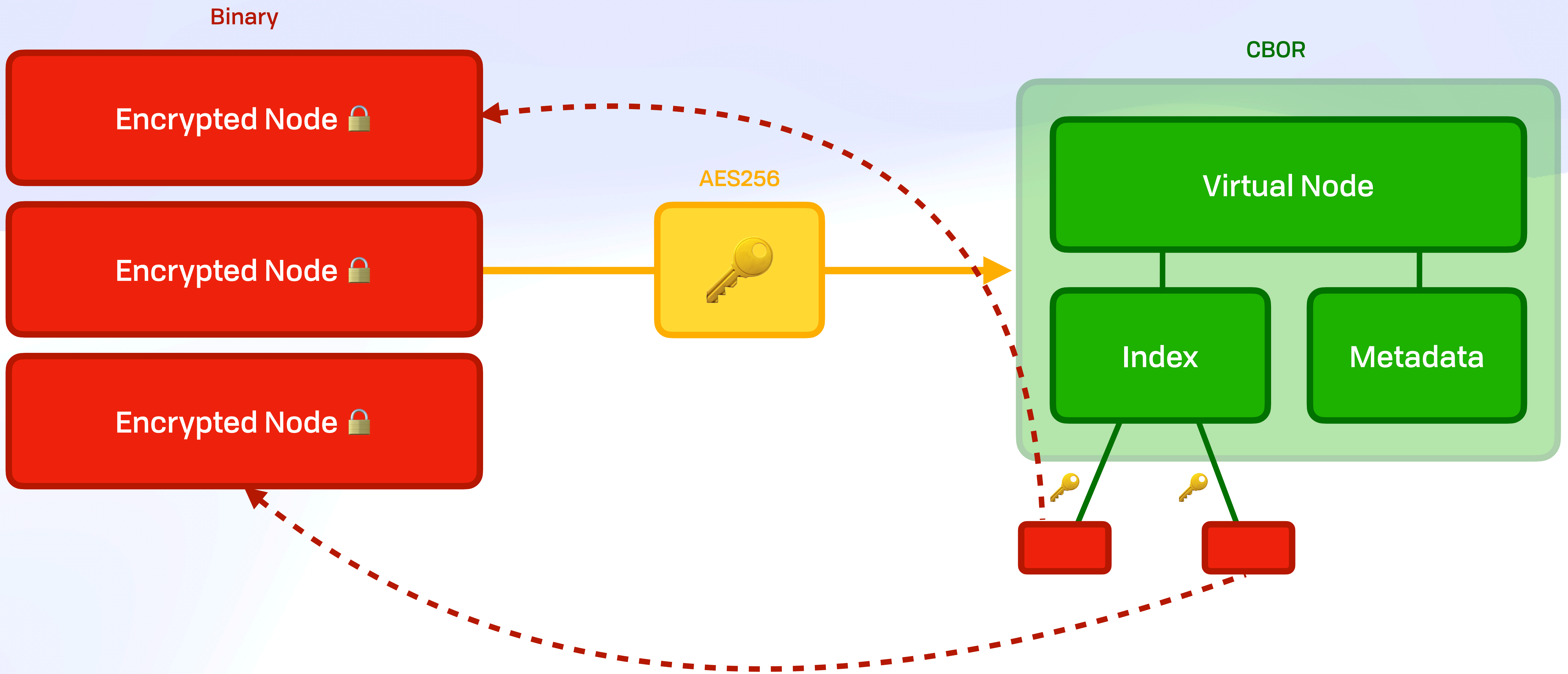
Private Data Substrate

Private Nodes



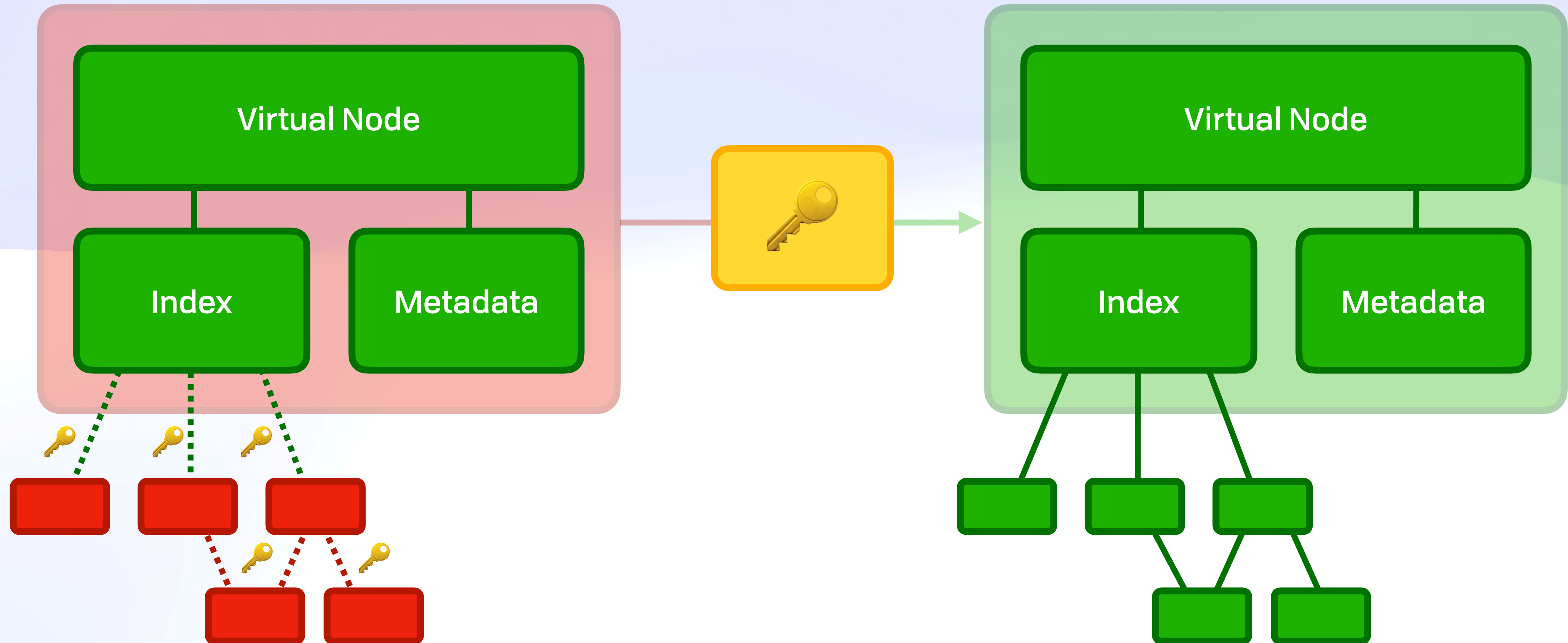
Private Data Substrate

Private Nodes



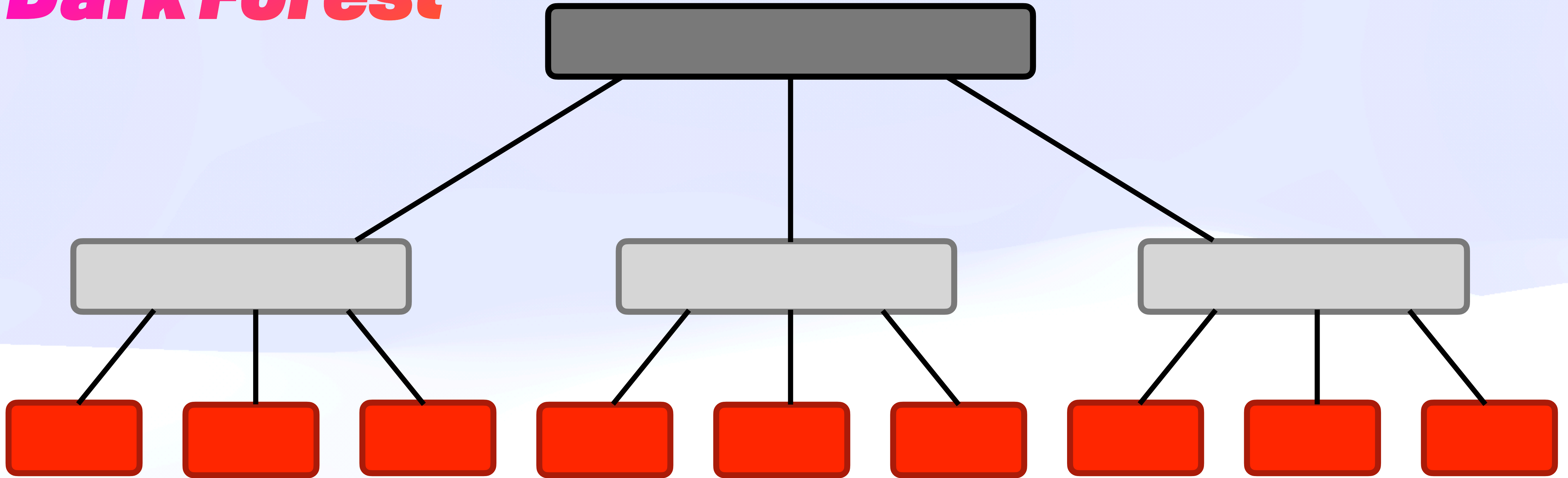
Private Data Substrate 

Reconstruction



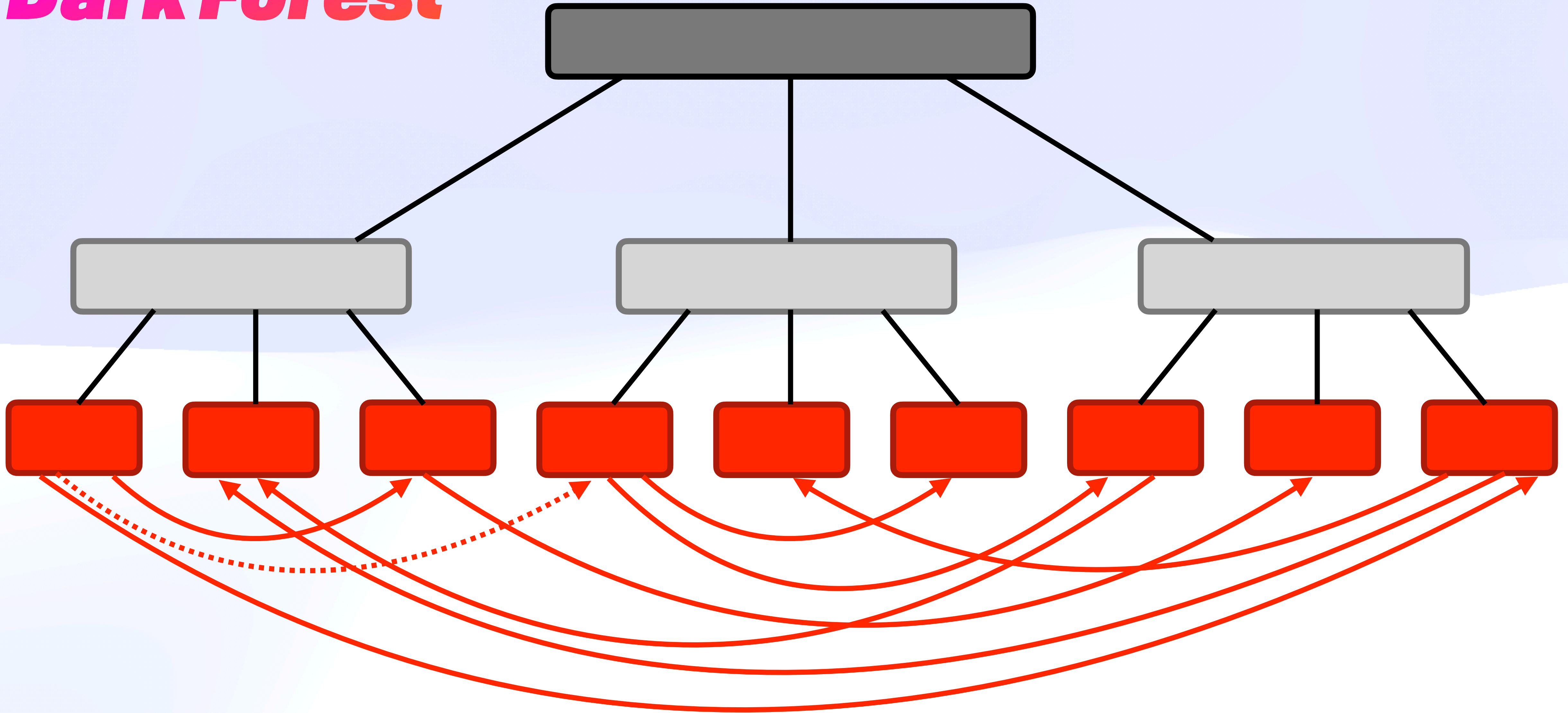
Private Data Substrate 

Dark Forest



Private Data Substrate 

Dark Forest



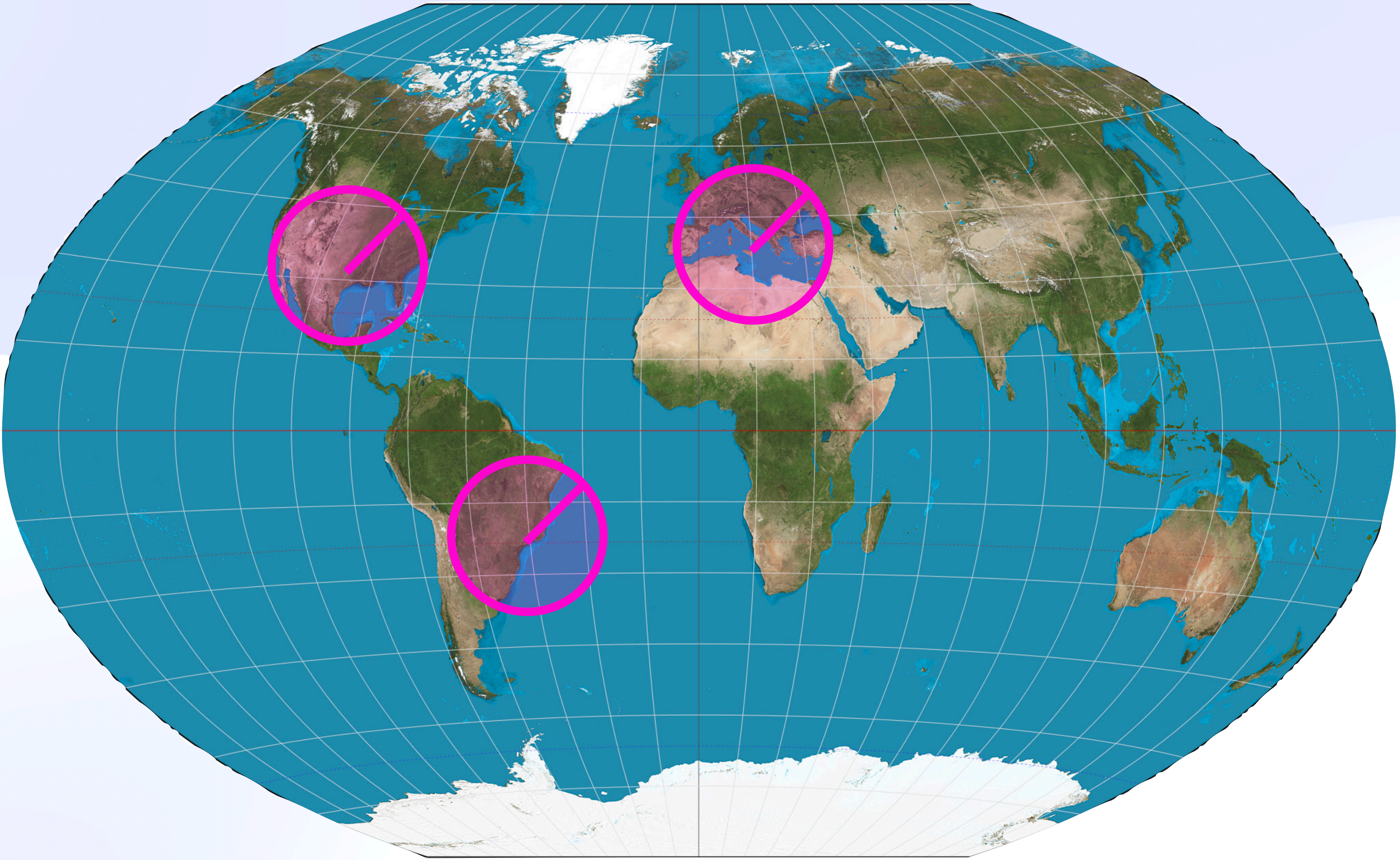
Fault-Safe Concurrency

Safely Merge Unreliable Updates



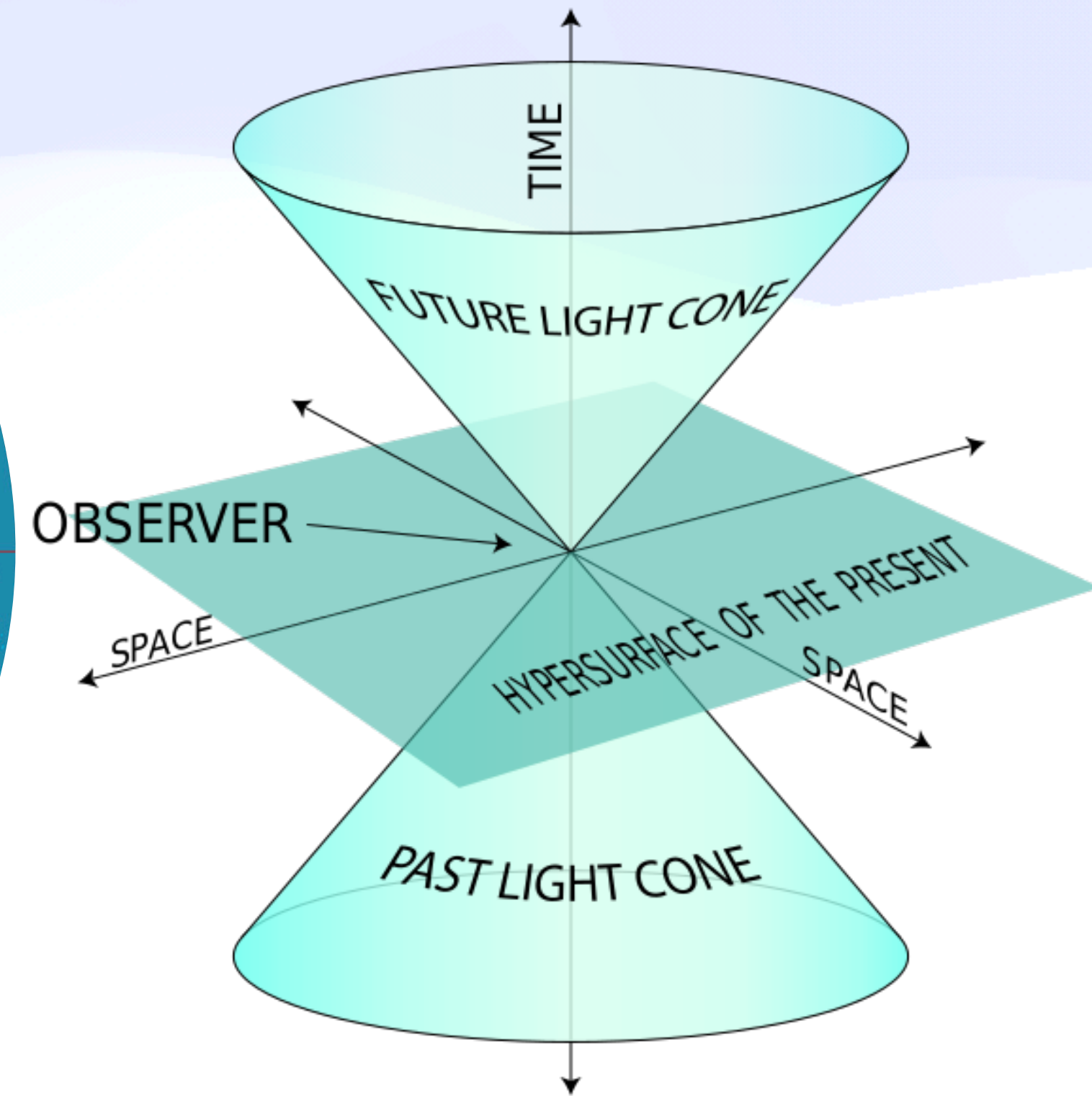
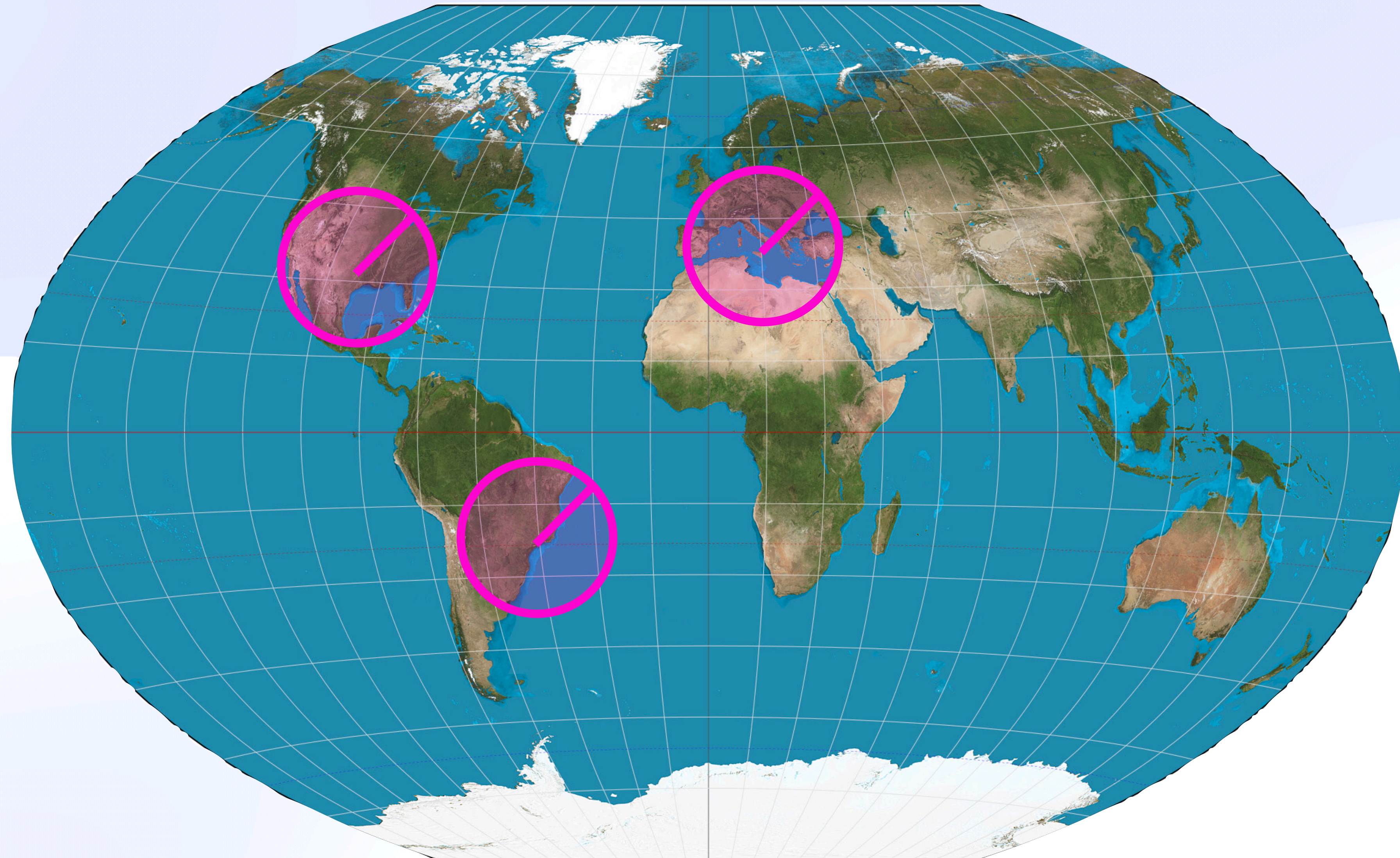
Fault-Safe Concurrency 

Causal Islands



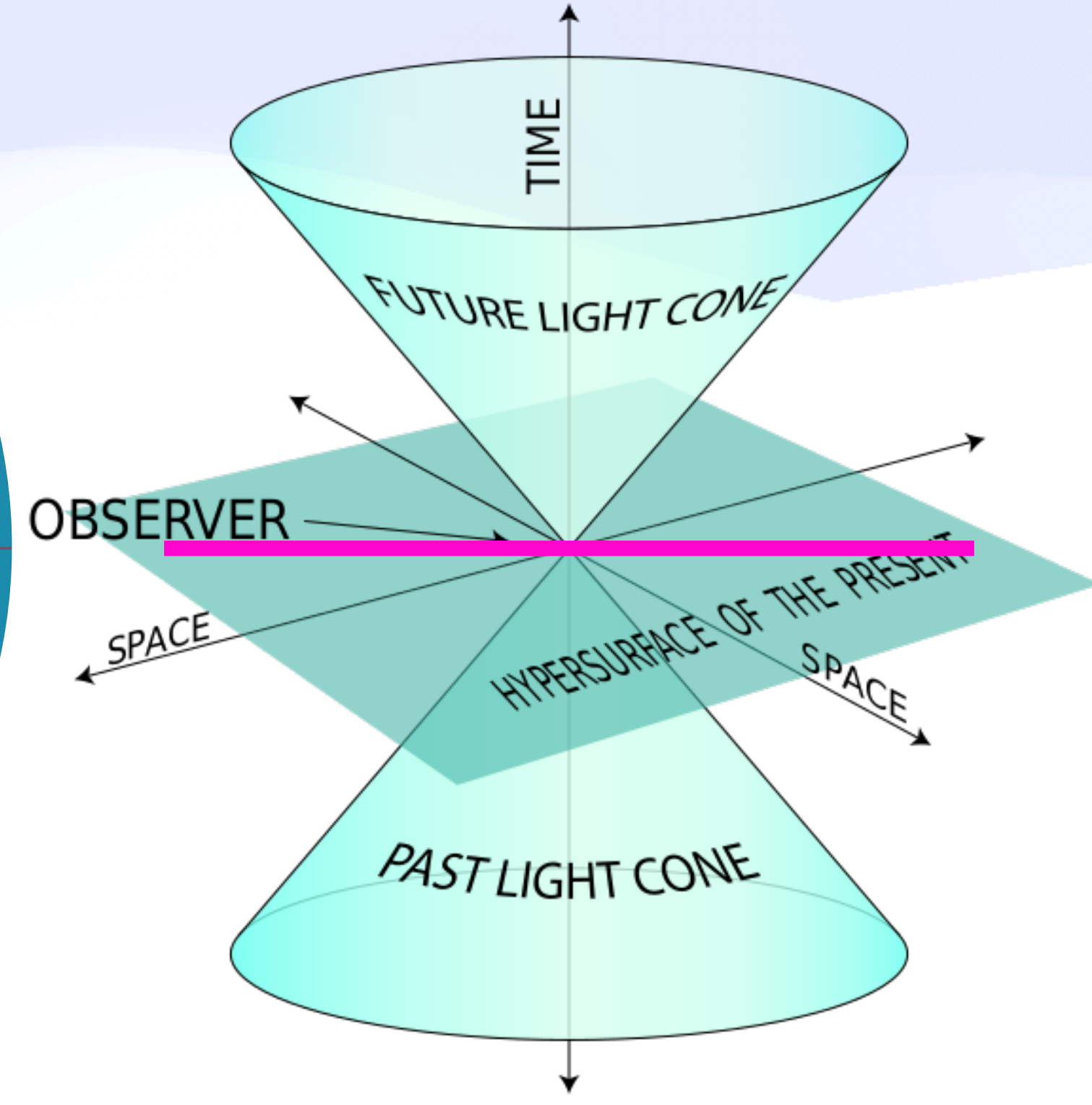
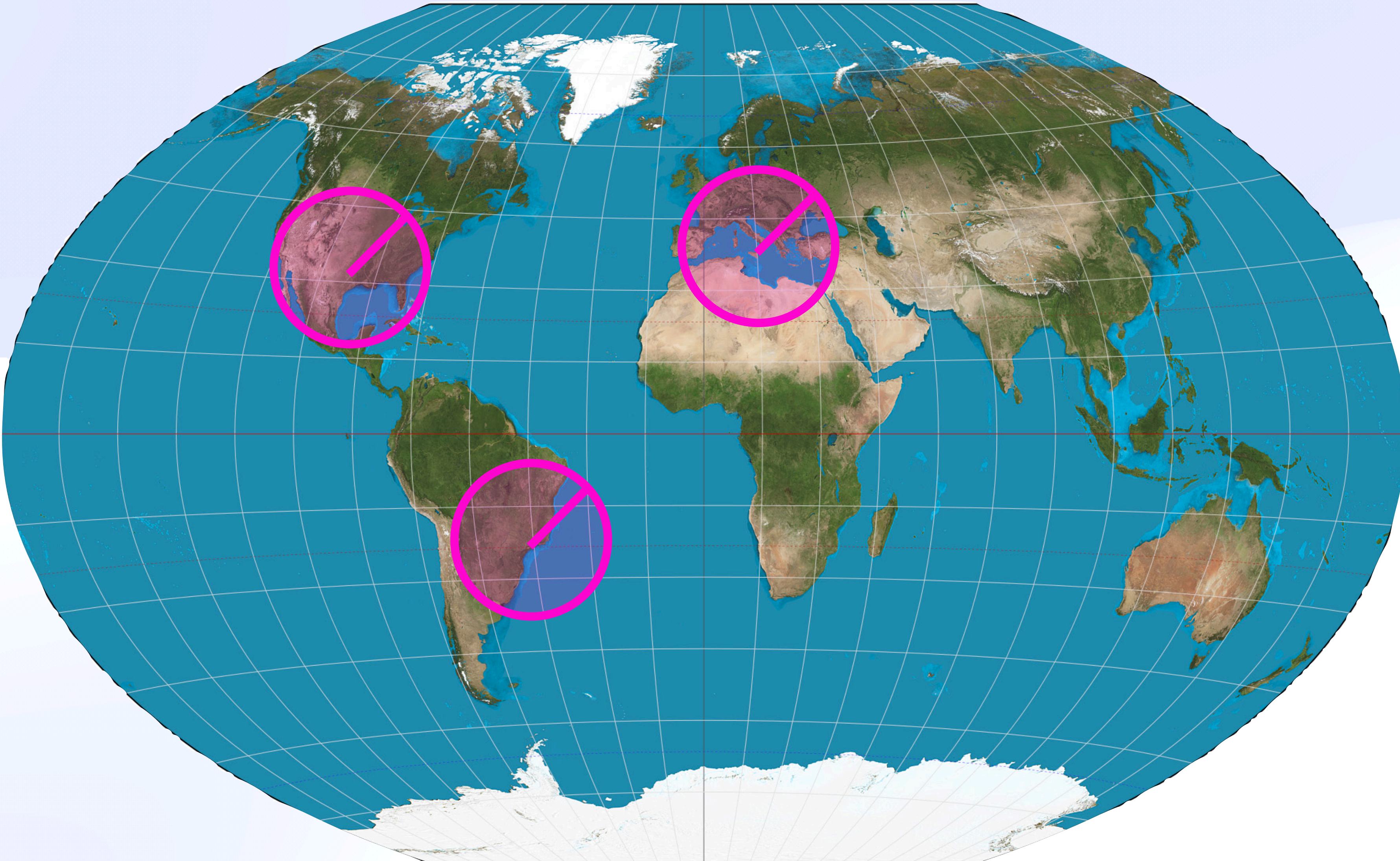
Fault-Safe Concurrency 

Causal Islands



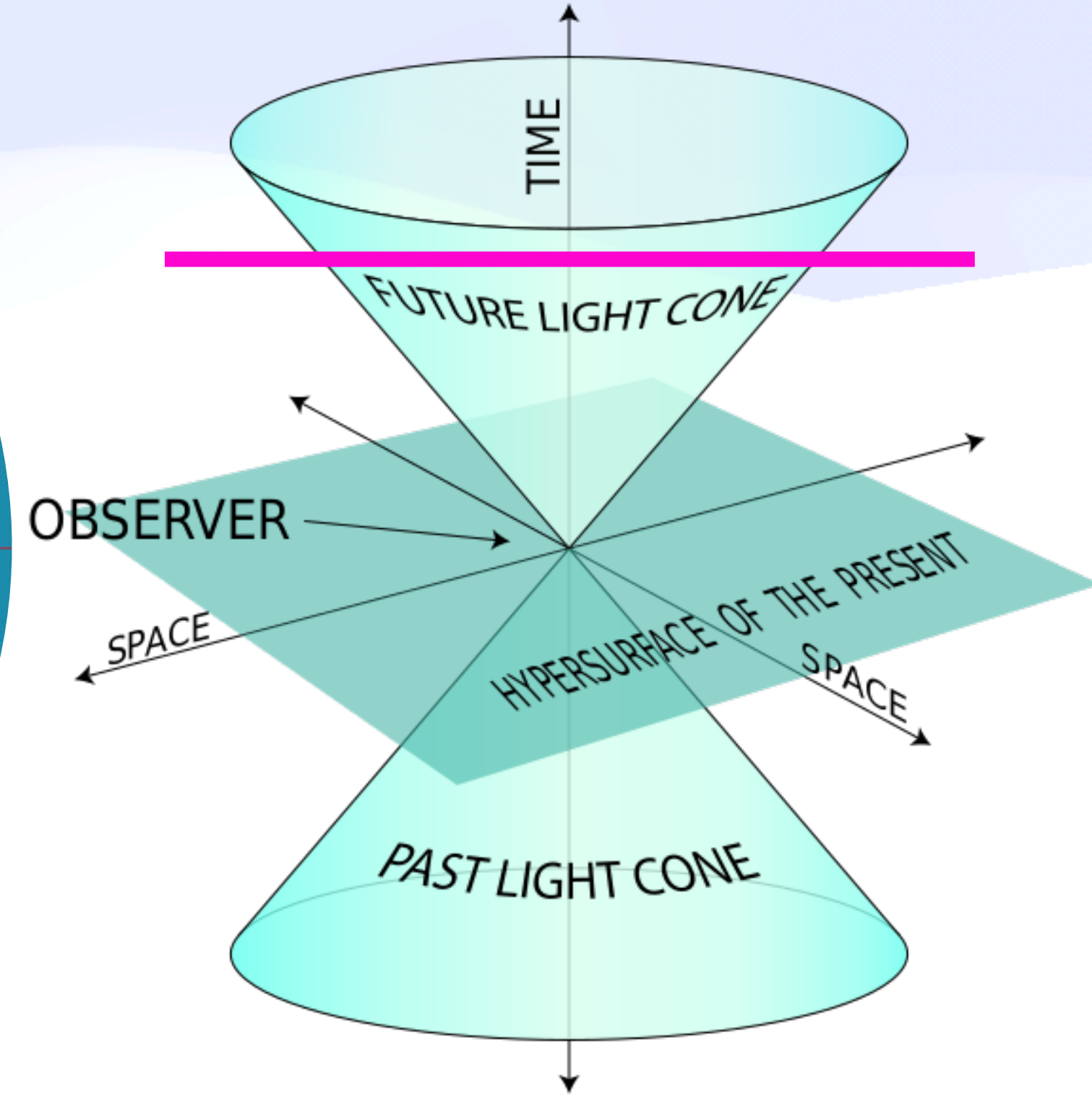
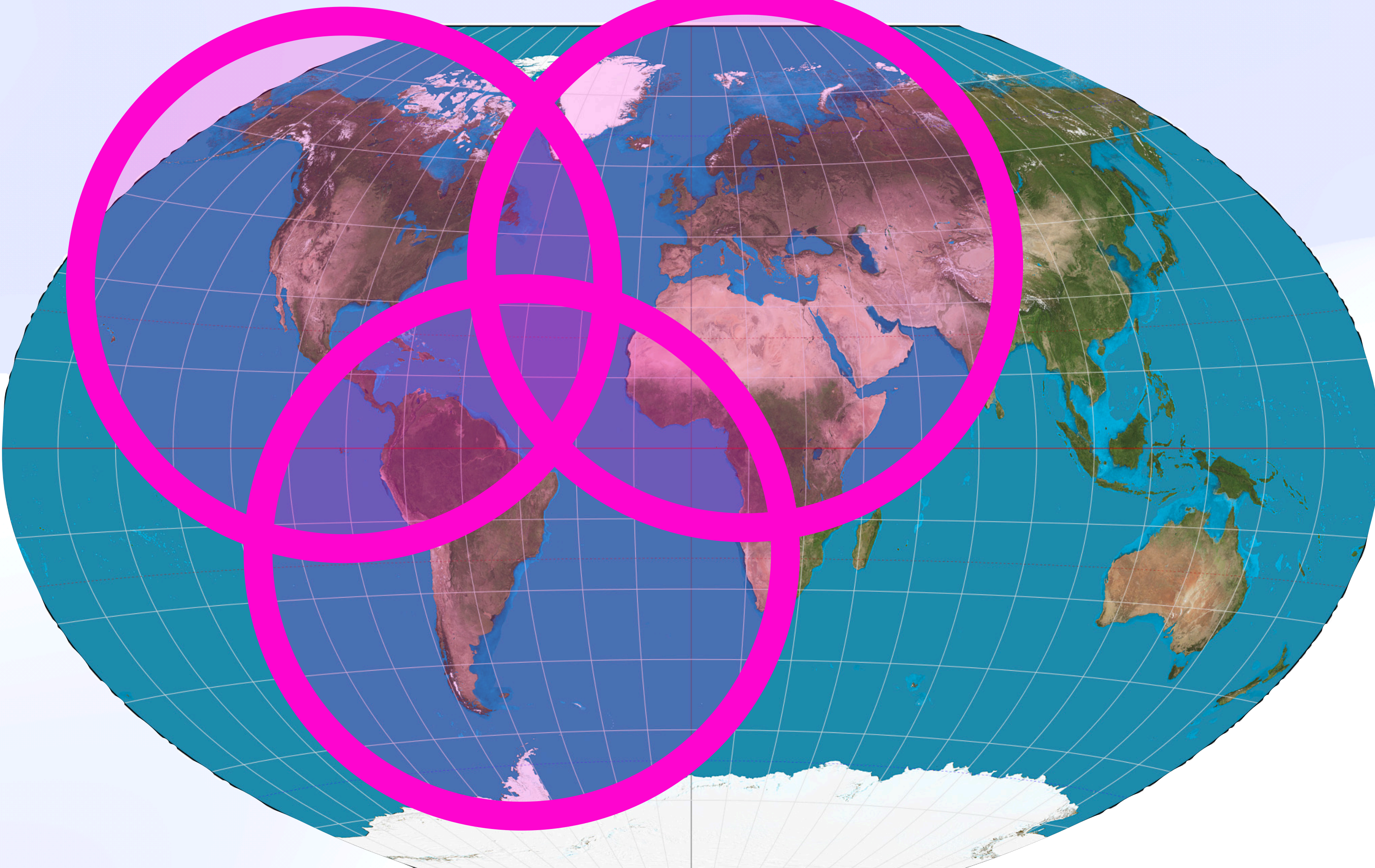
Fault-Safe Concurrency

Causal Islands



Fault-Safe Concurrency 

Causal Islands

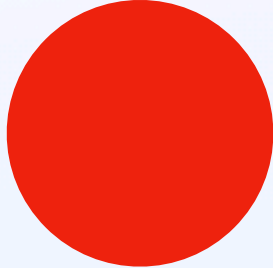
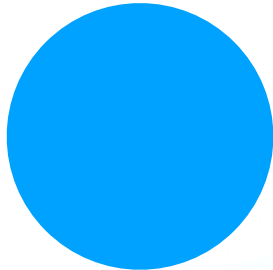


Fault-Safe Concurrency 

Gossiping Out of Order

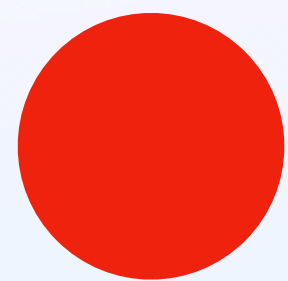
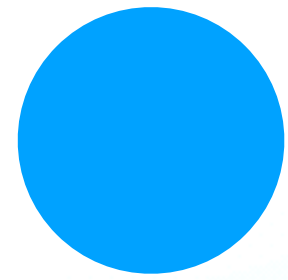
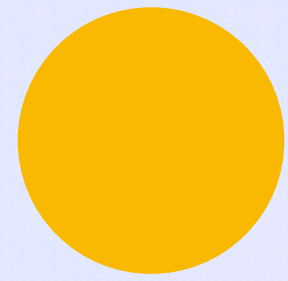
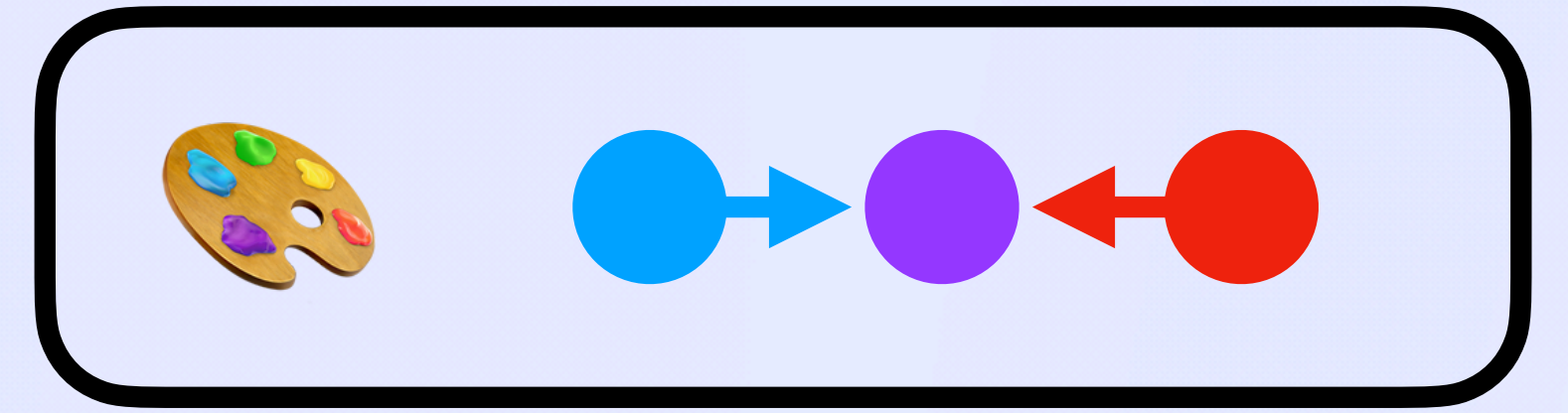
Fault-Safe Concurrency 

Gossiping Out of Order



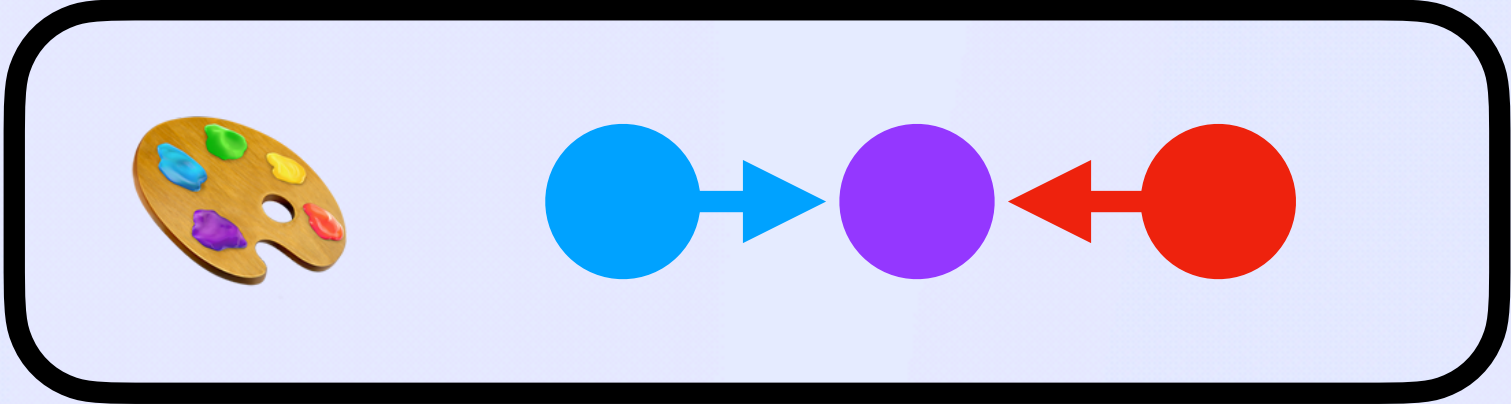
Fault-Safe Concurrency 

Gossiping Out of Order



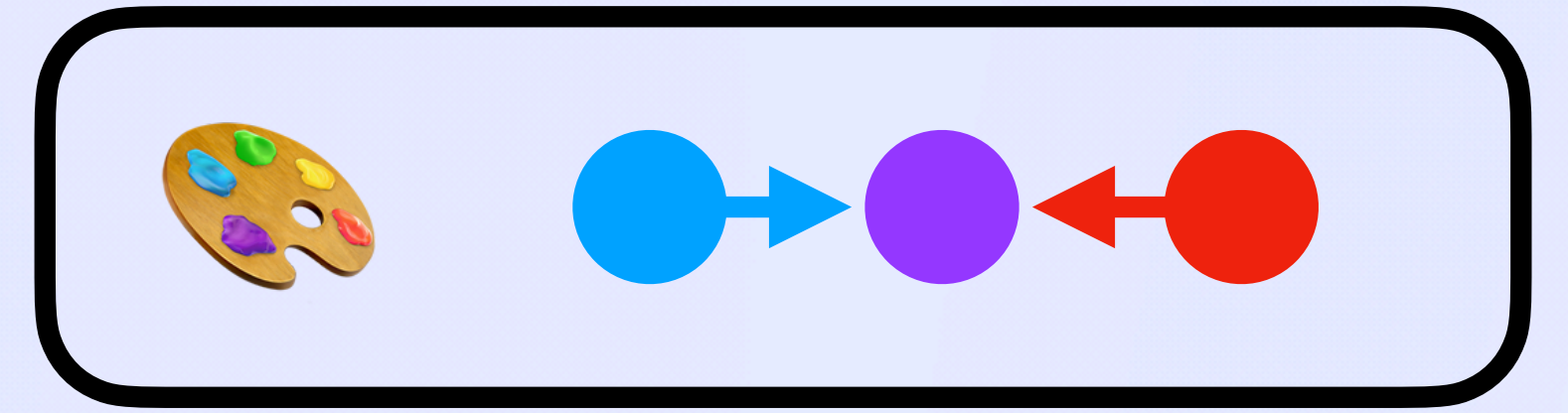
Fault-Safe Concurrency 

Gossiping Out of Order



Fault-Safe Concurrency 

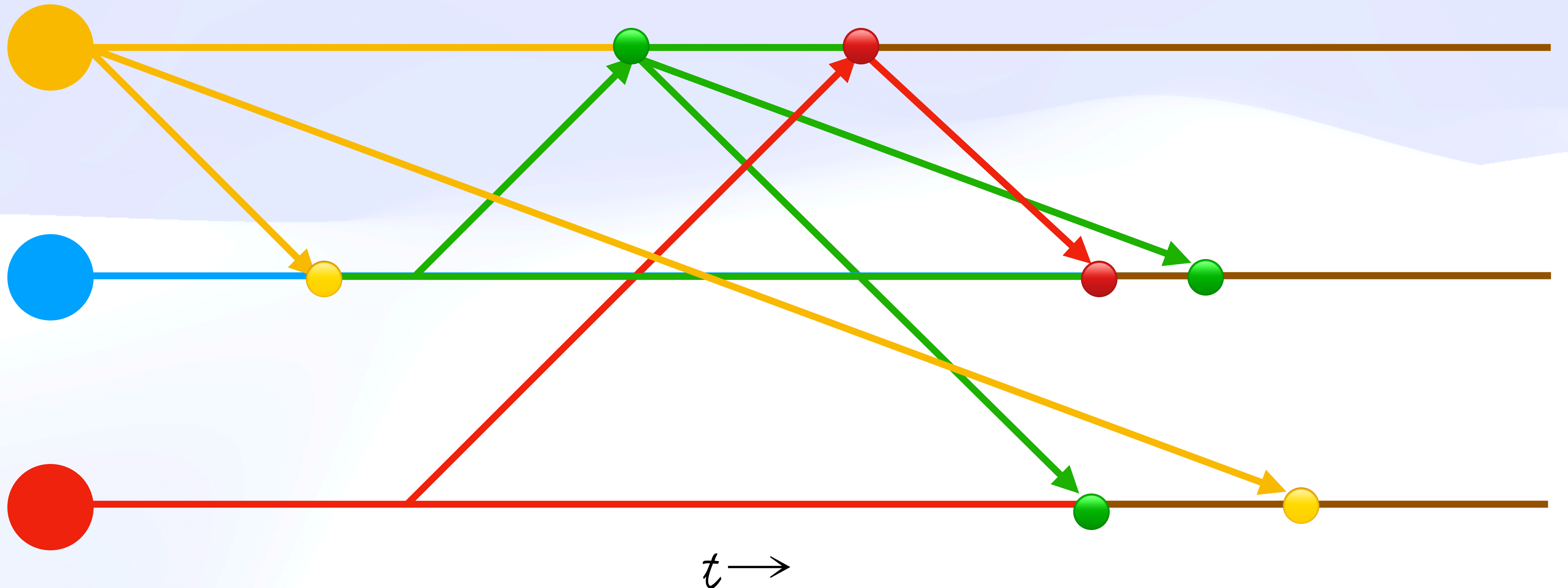
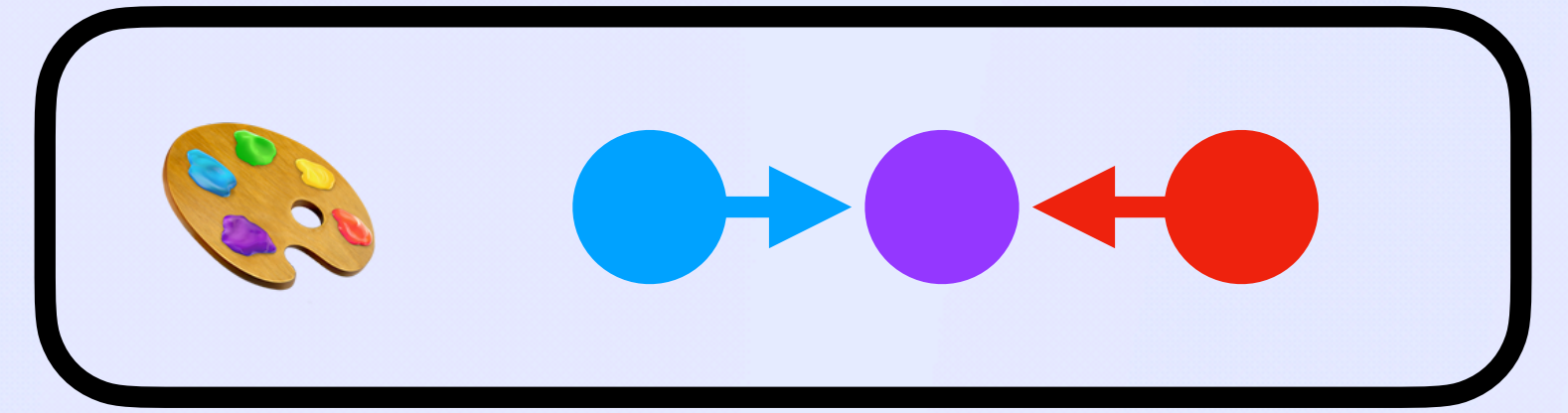
Gossiping Out of Order



$t \rightarrow$

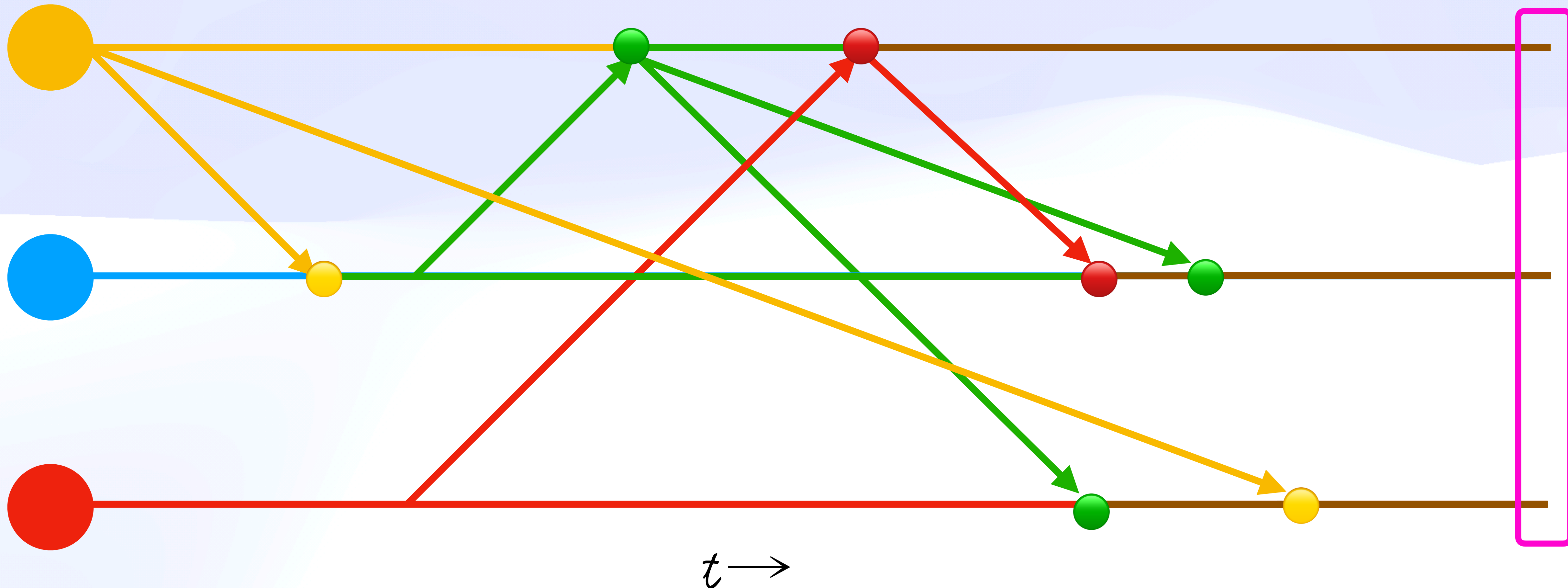
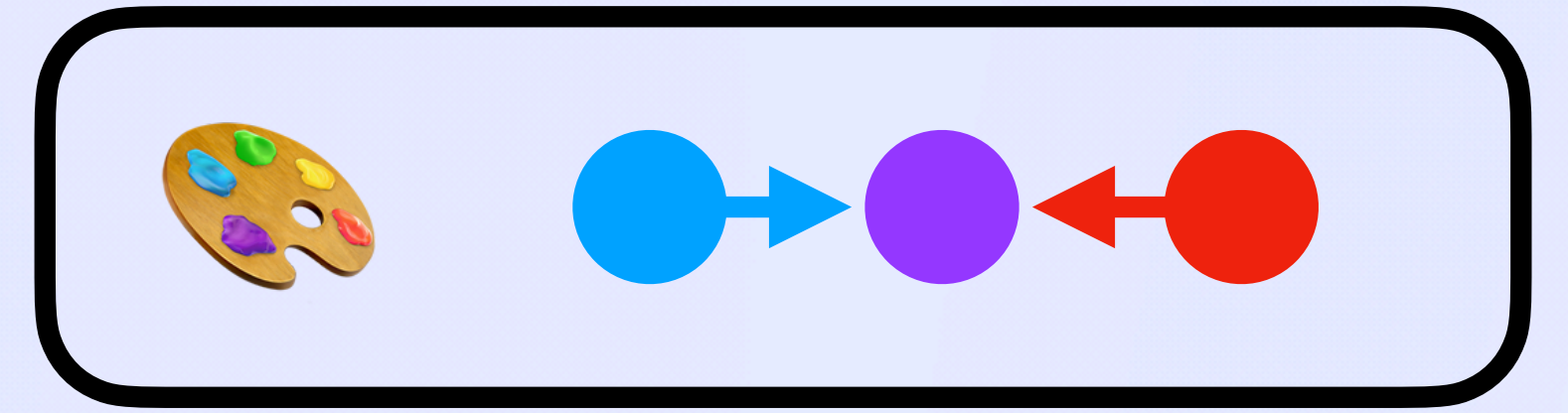
Fault-Safe Concurrency 

Gossiping Out of Order



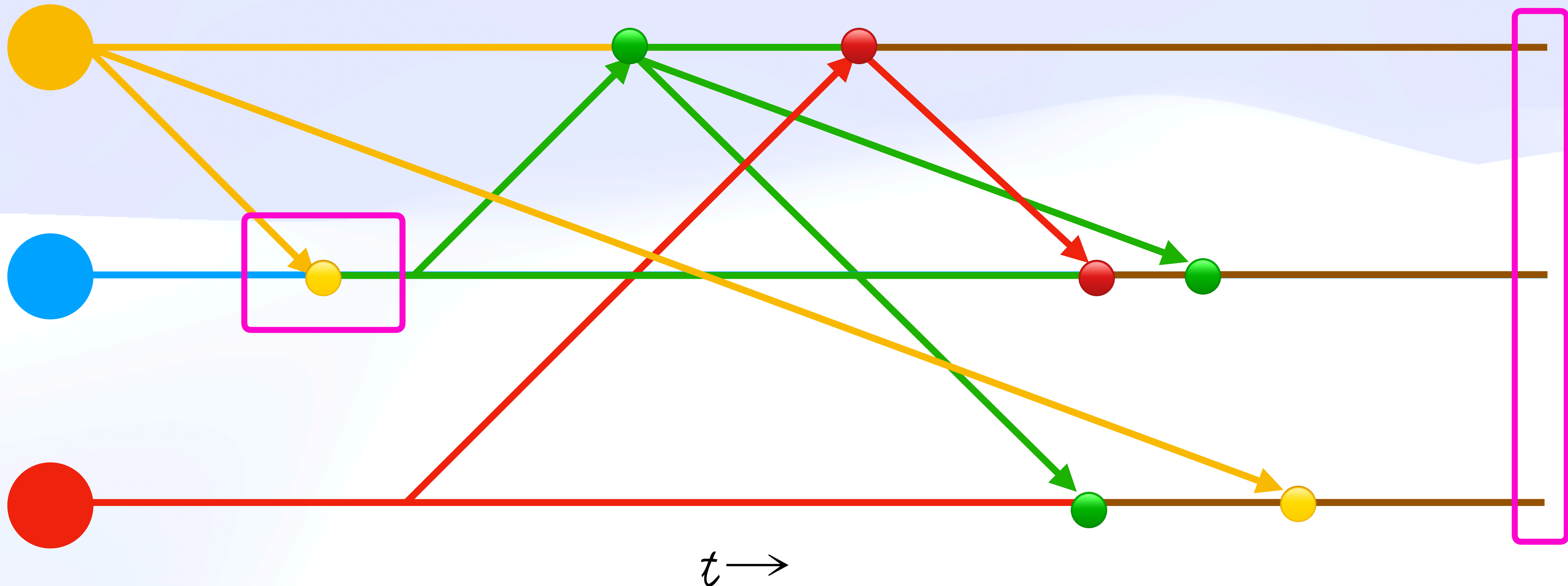
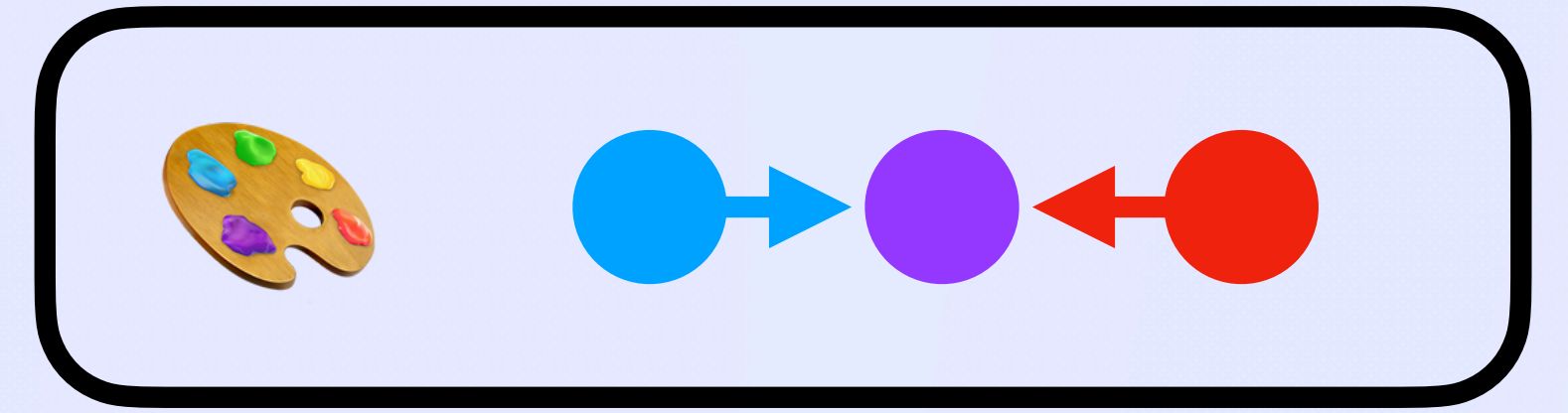
Fault-Safe Concurrency 

Gossiping Out of Order



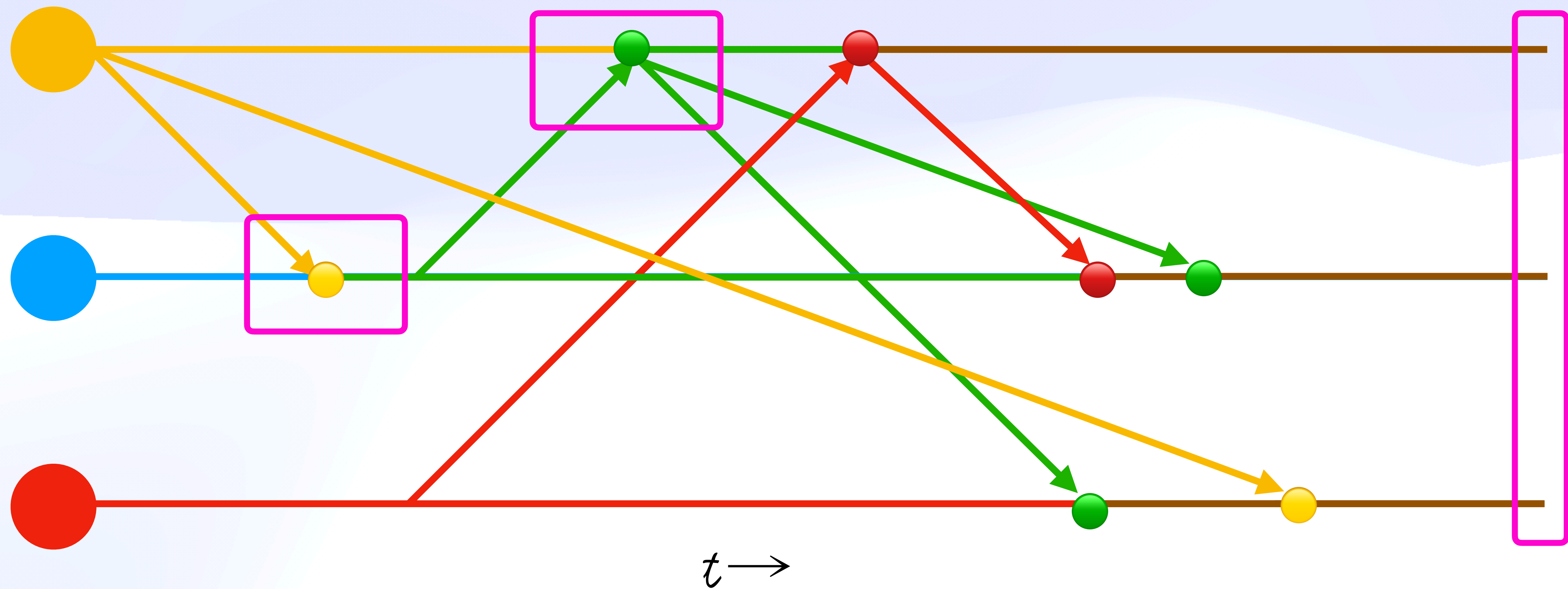
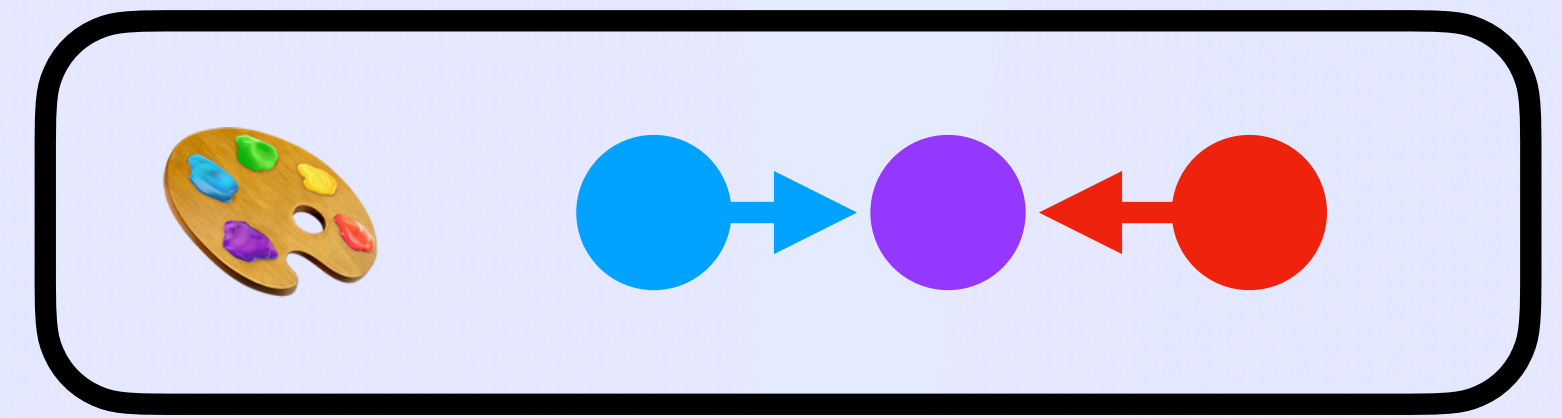
Fault-Safe Concurrency 

Gossiping Out of Order



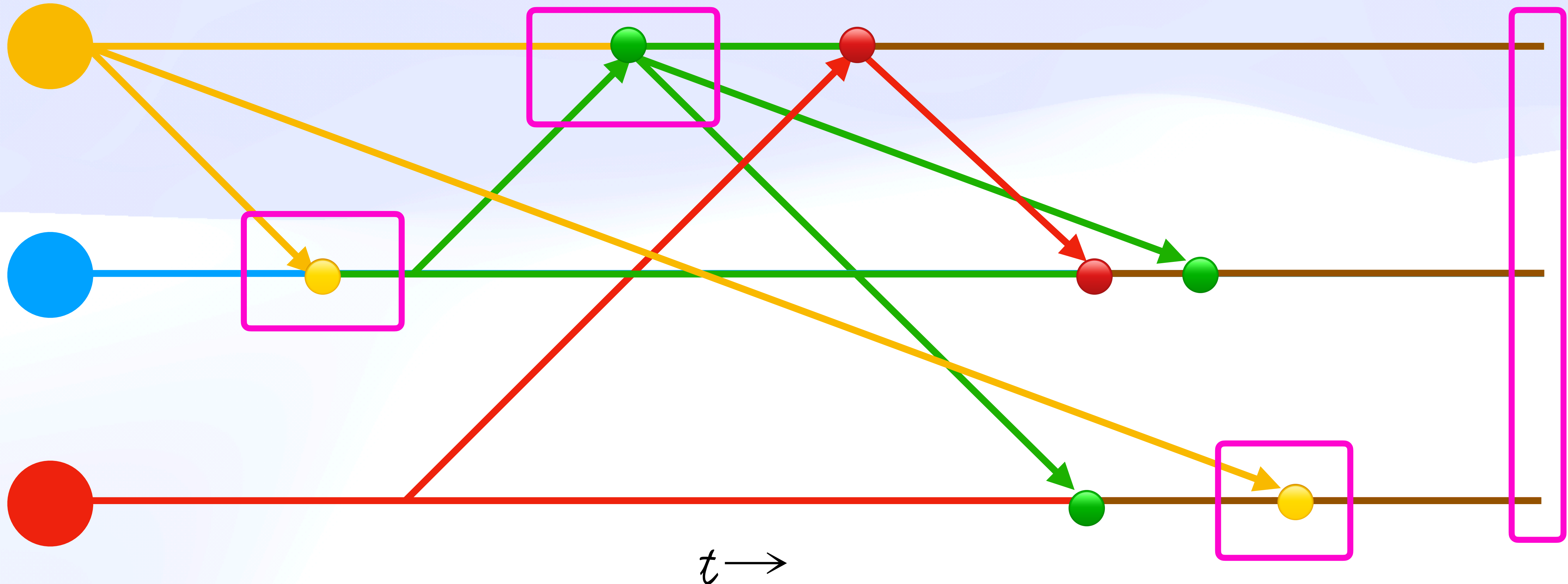
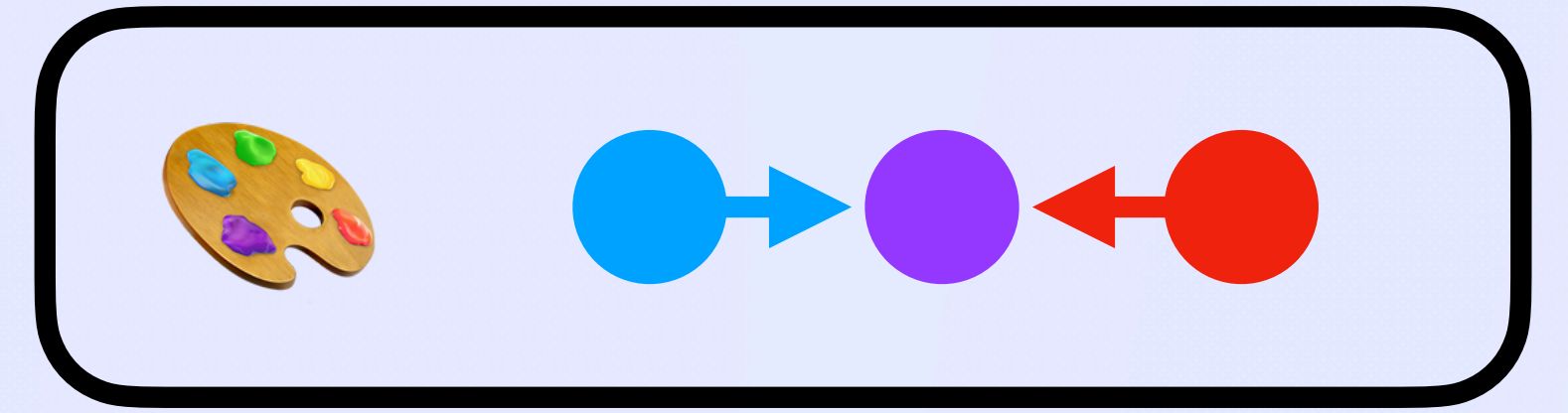
Fault-Safe Concurrency 

Gossiping Out of Order



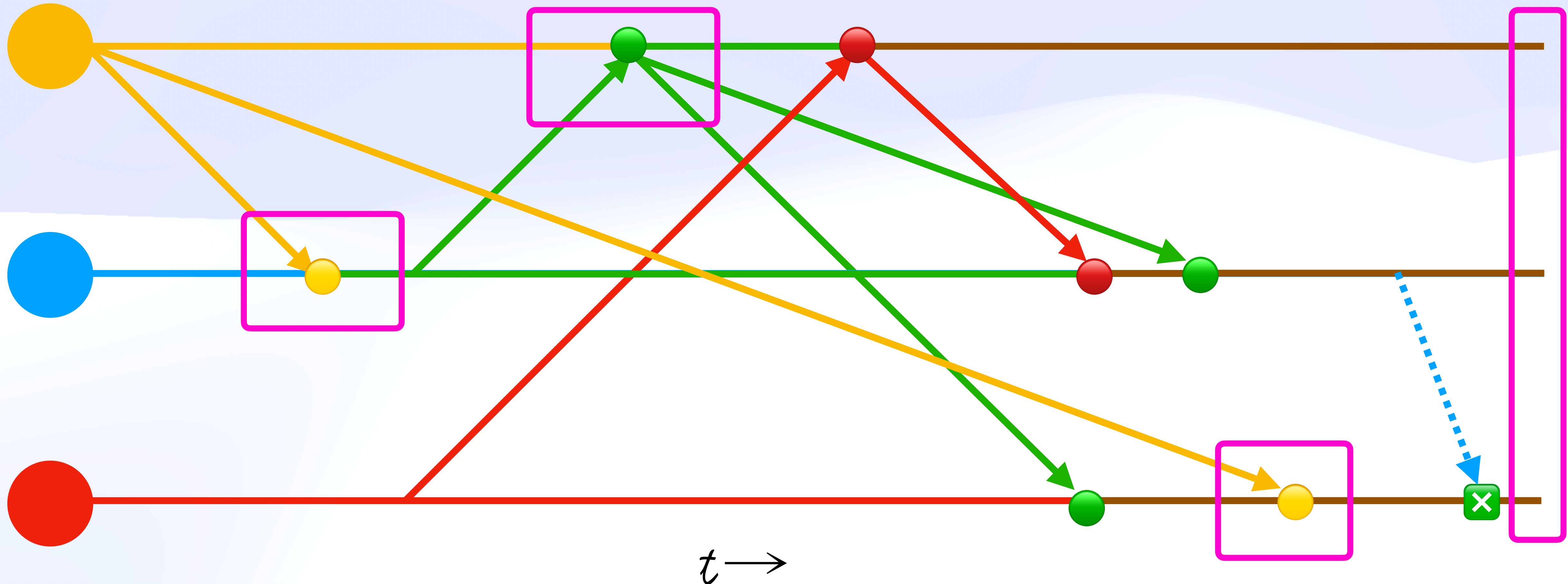
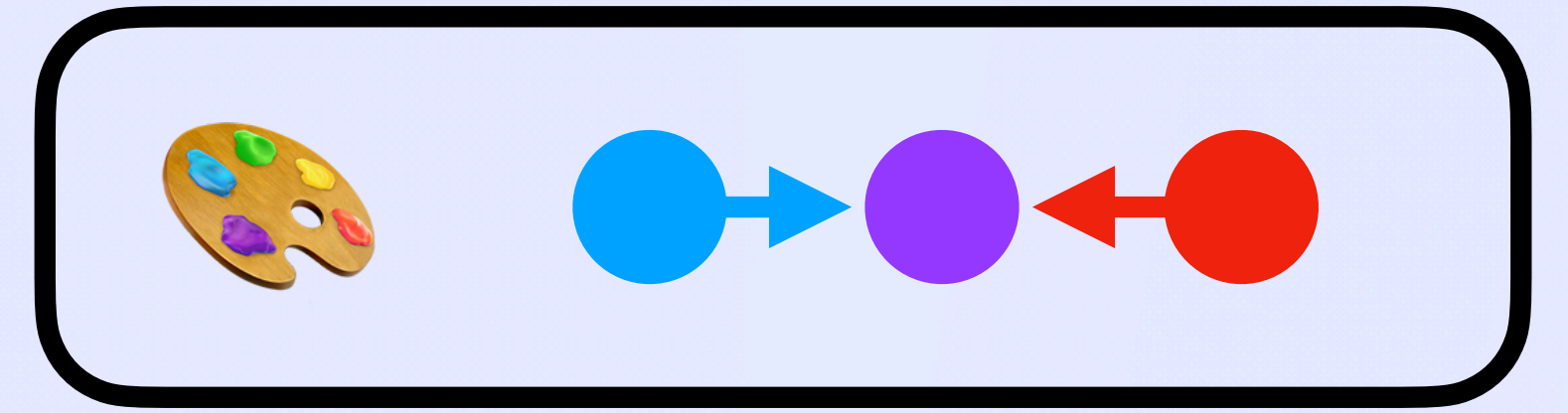
Fault-Safe Concurrency 

Gossiping Out of Order



Fault-Safe Concurrency 

Gossiping Out of Order



Fault-Safe Concurrency 

Entropy Isn't What It Used to Be

Fault-Safe Concurrency 

Properties Save the Day

Fault-Safe Concurrency 

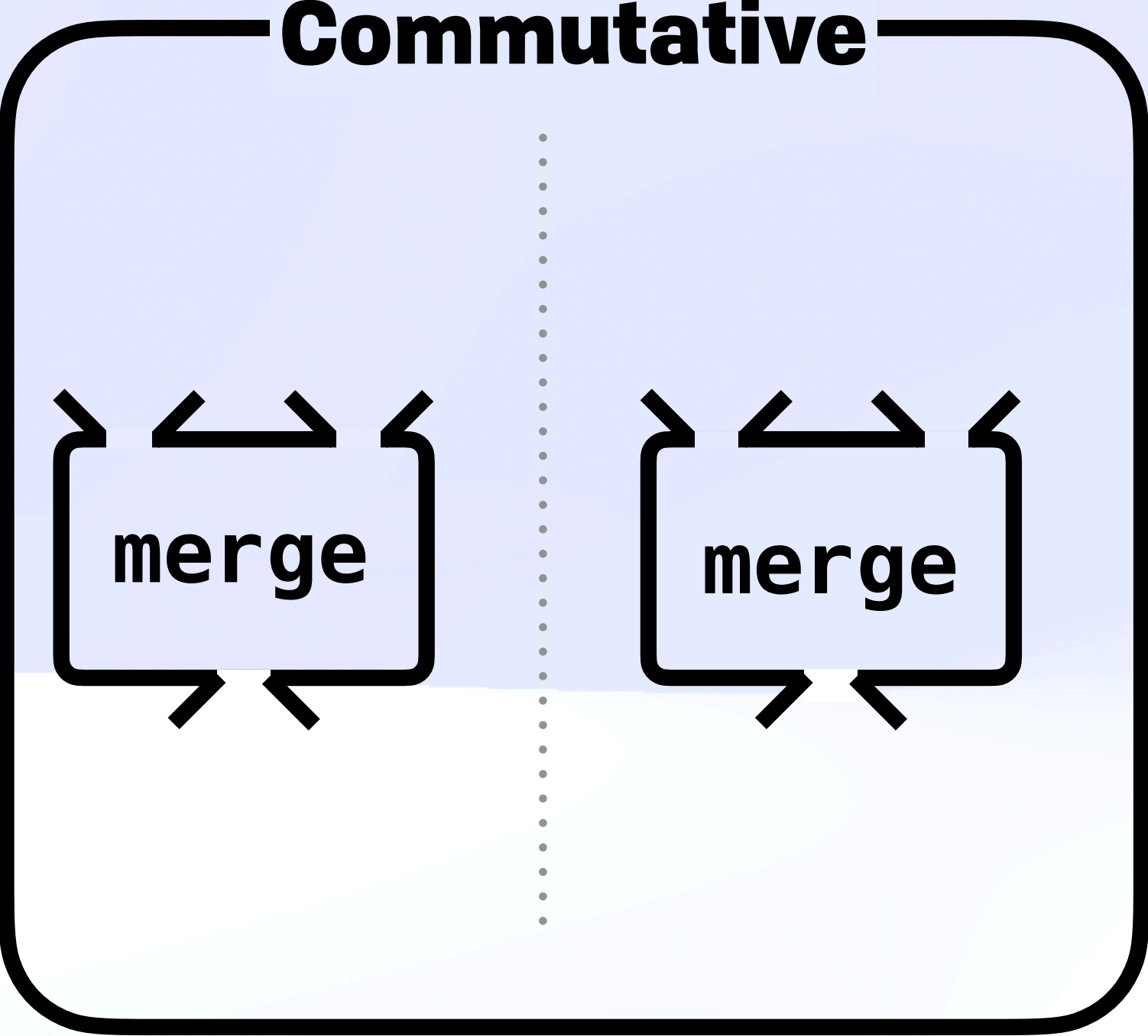
Properties Save the Day

Commutative



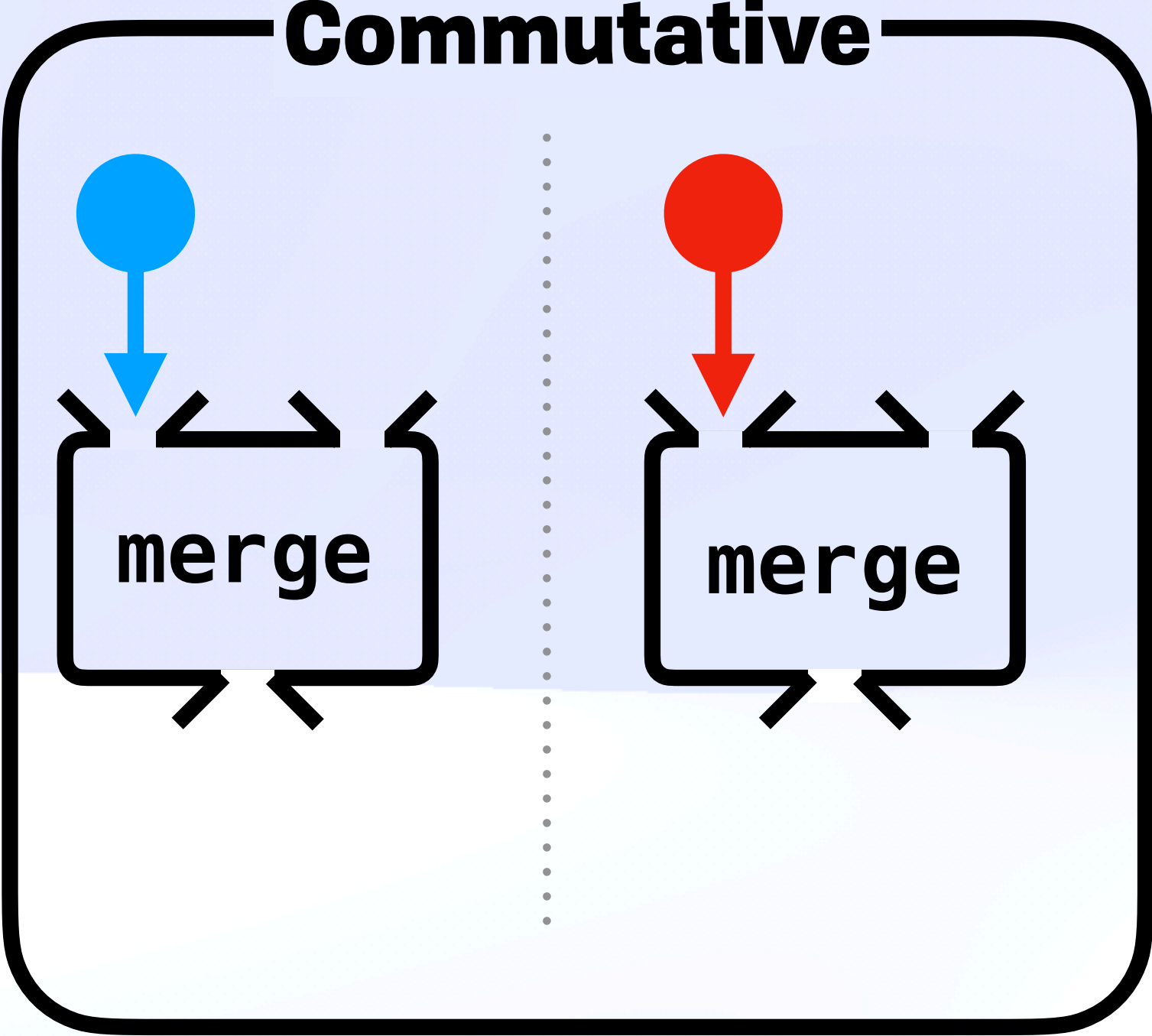
Fault-Safe Concurrency 

Properties Save the Day



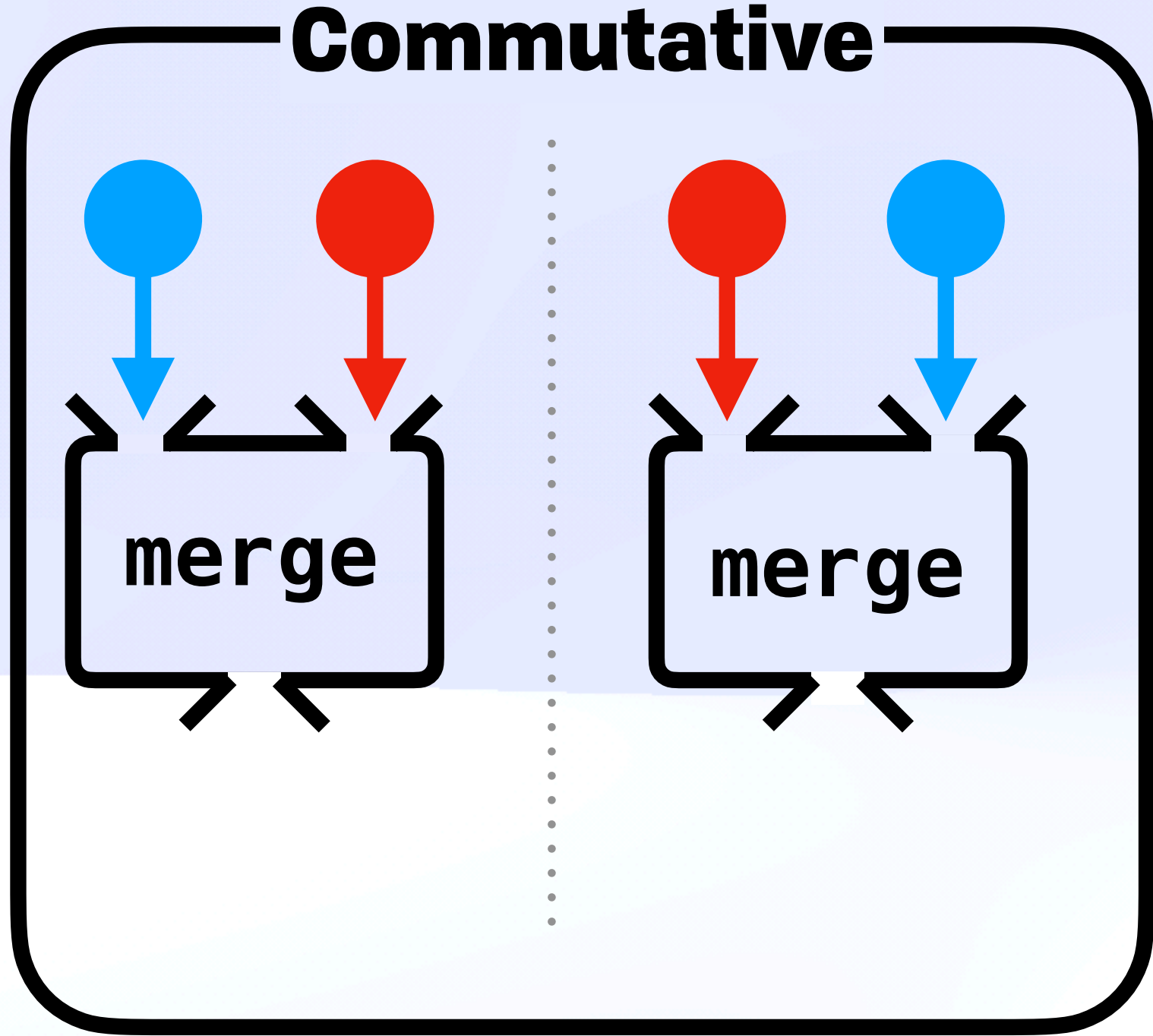
Fault-Safe Concurrency 

Properties Save the Day



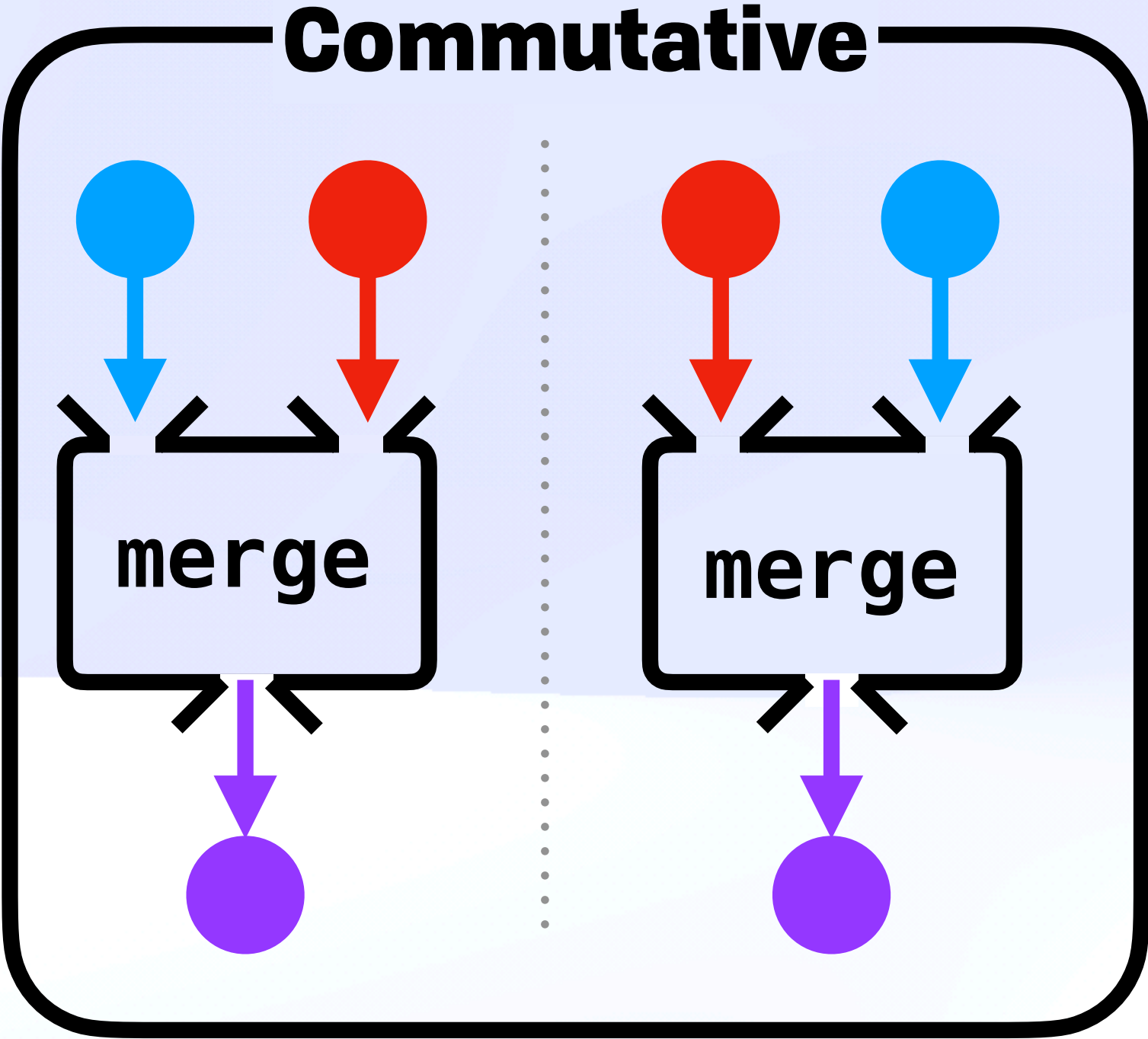
Fault-Safe Concurrency 

Properties Save the Day



Fault-Safe Concurrency 

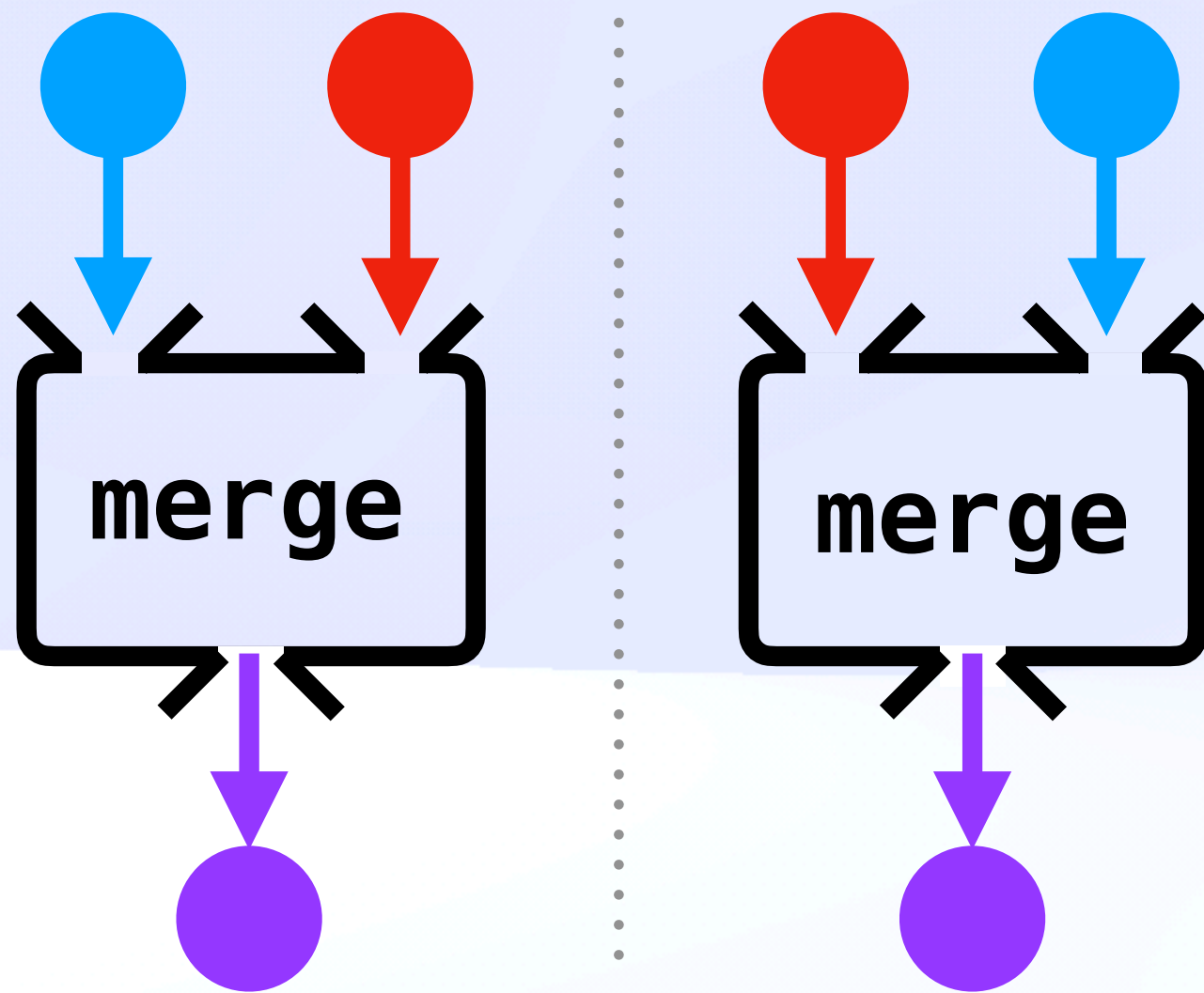
Properties Save the Day



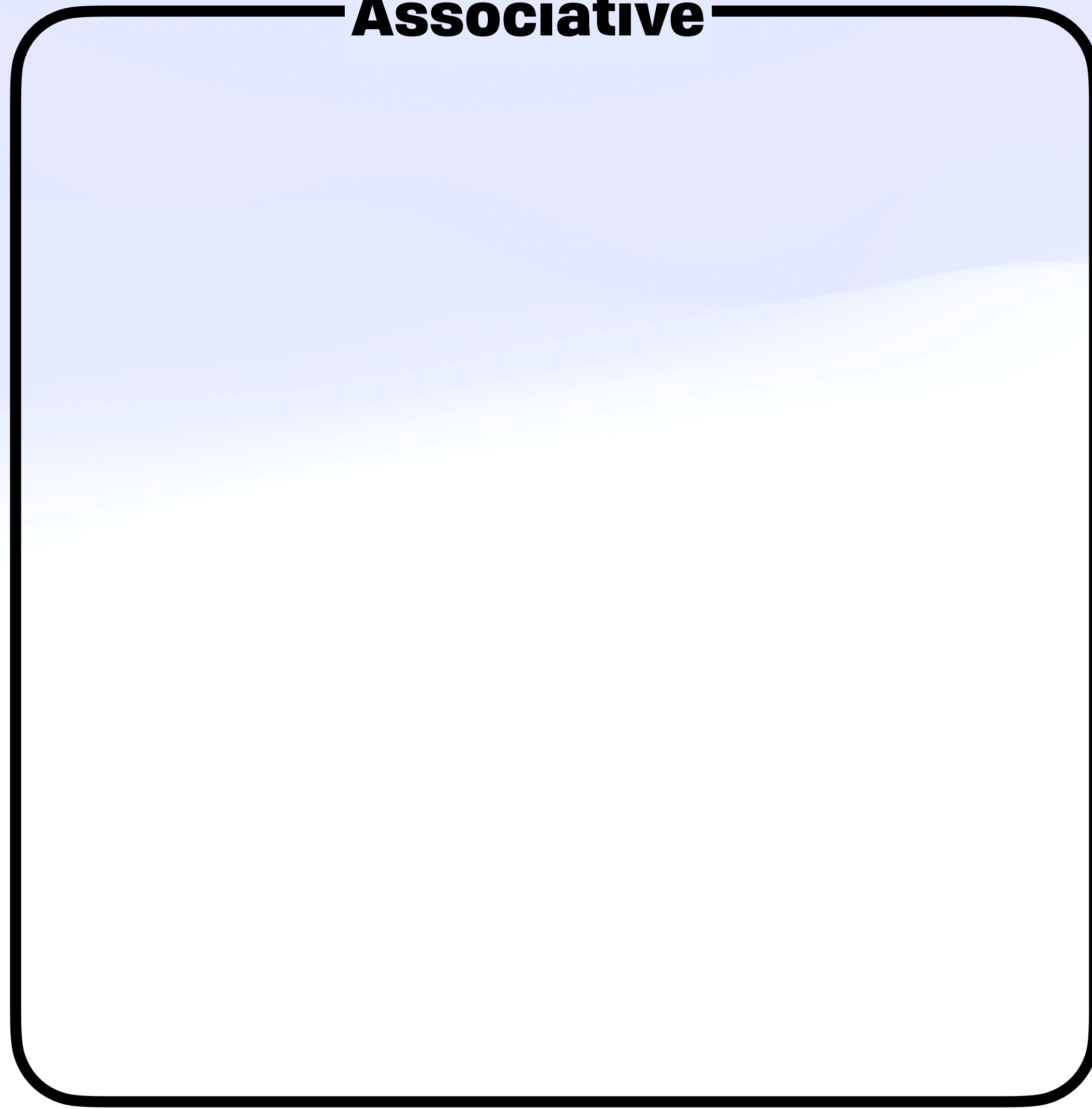
Fault-Safe Concurrency 

Properties Save the Day

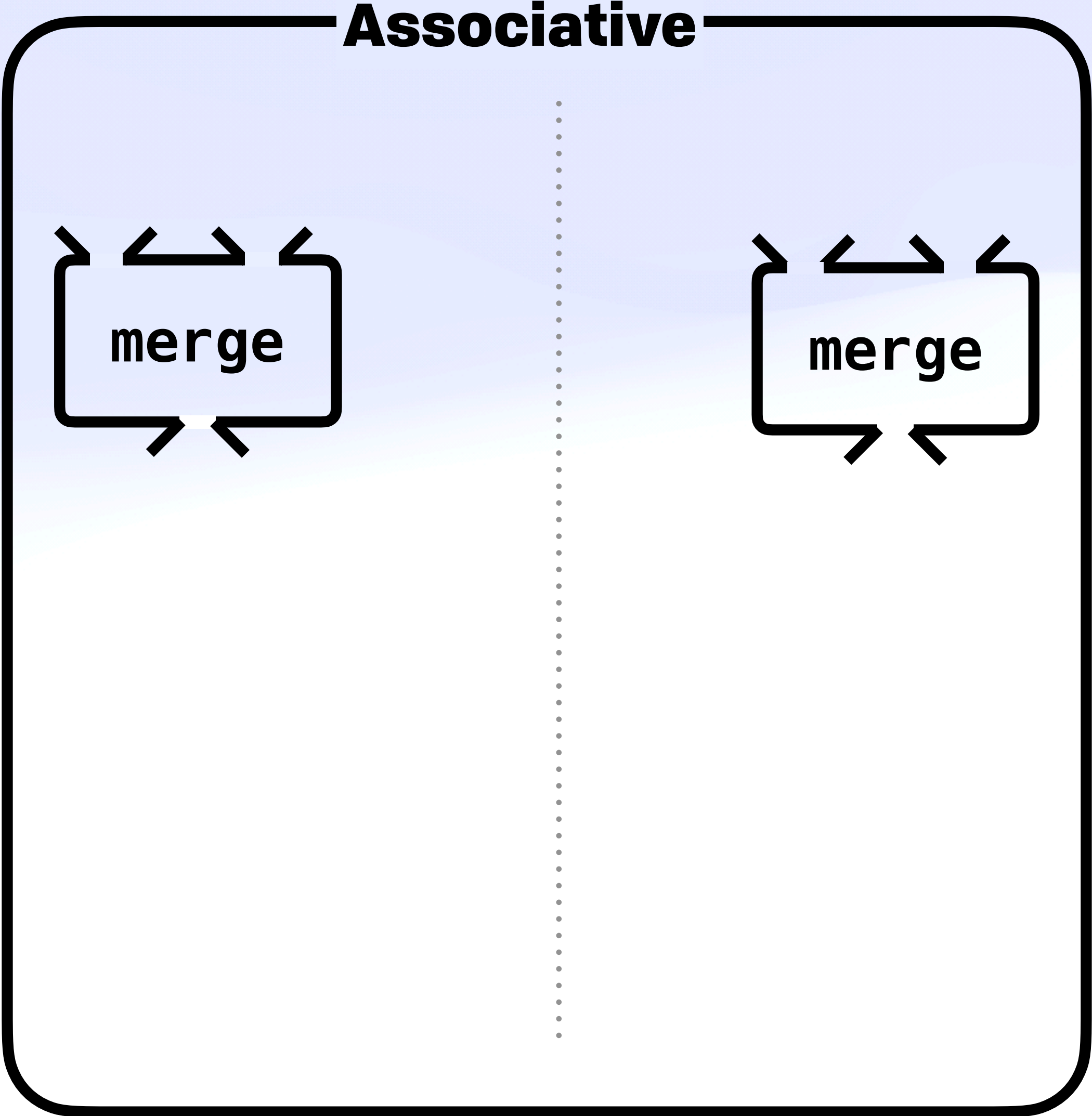
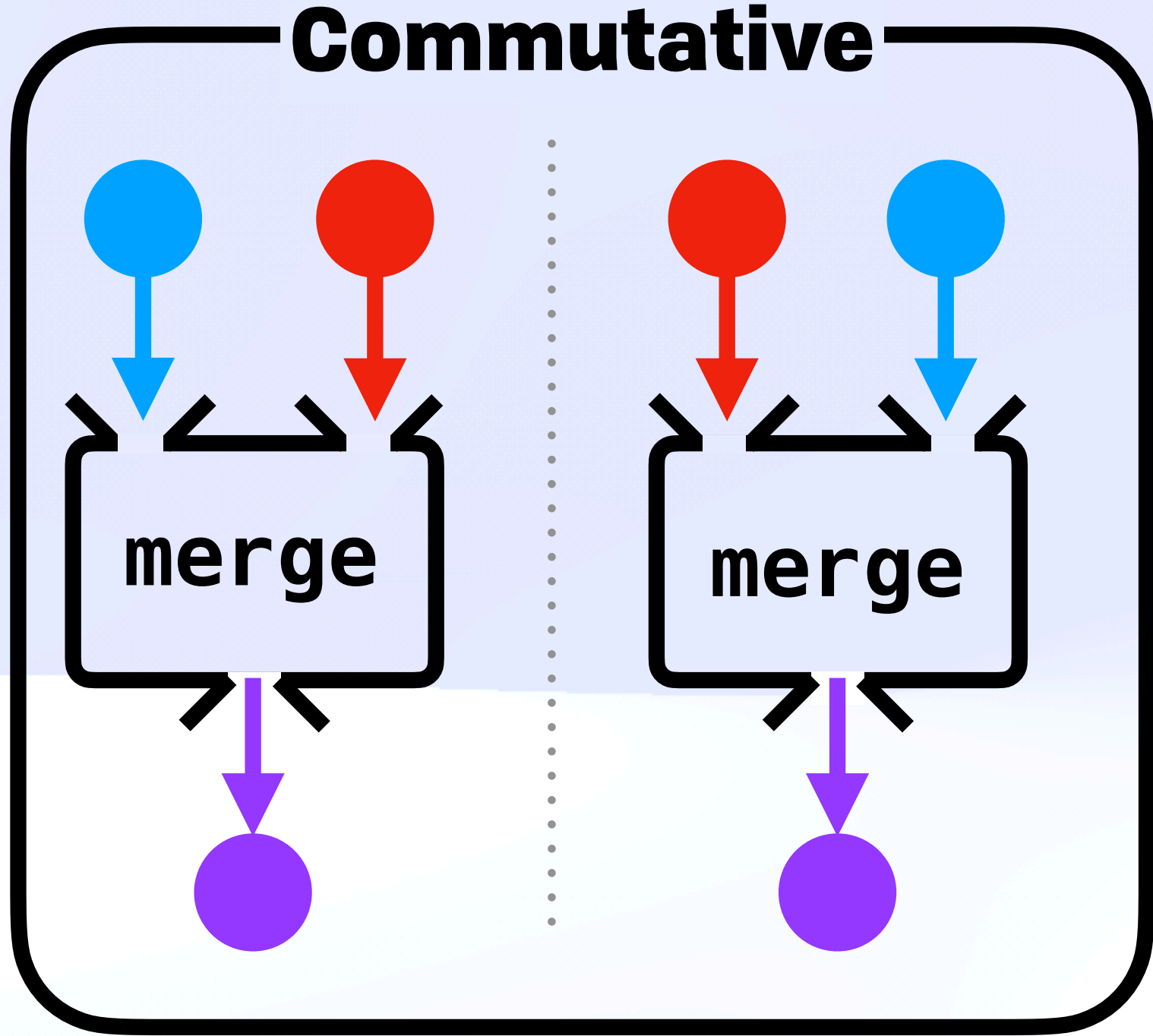
Commutative



Associative

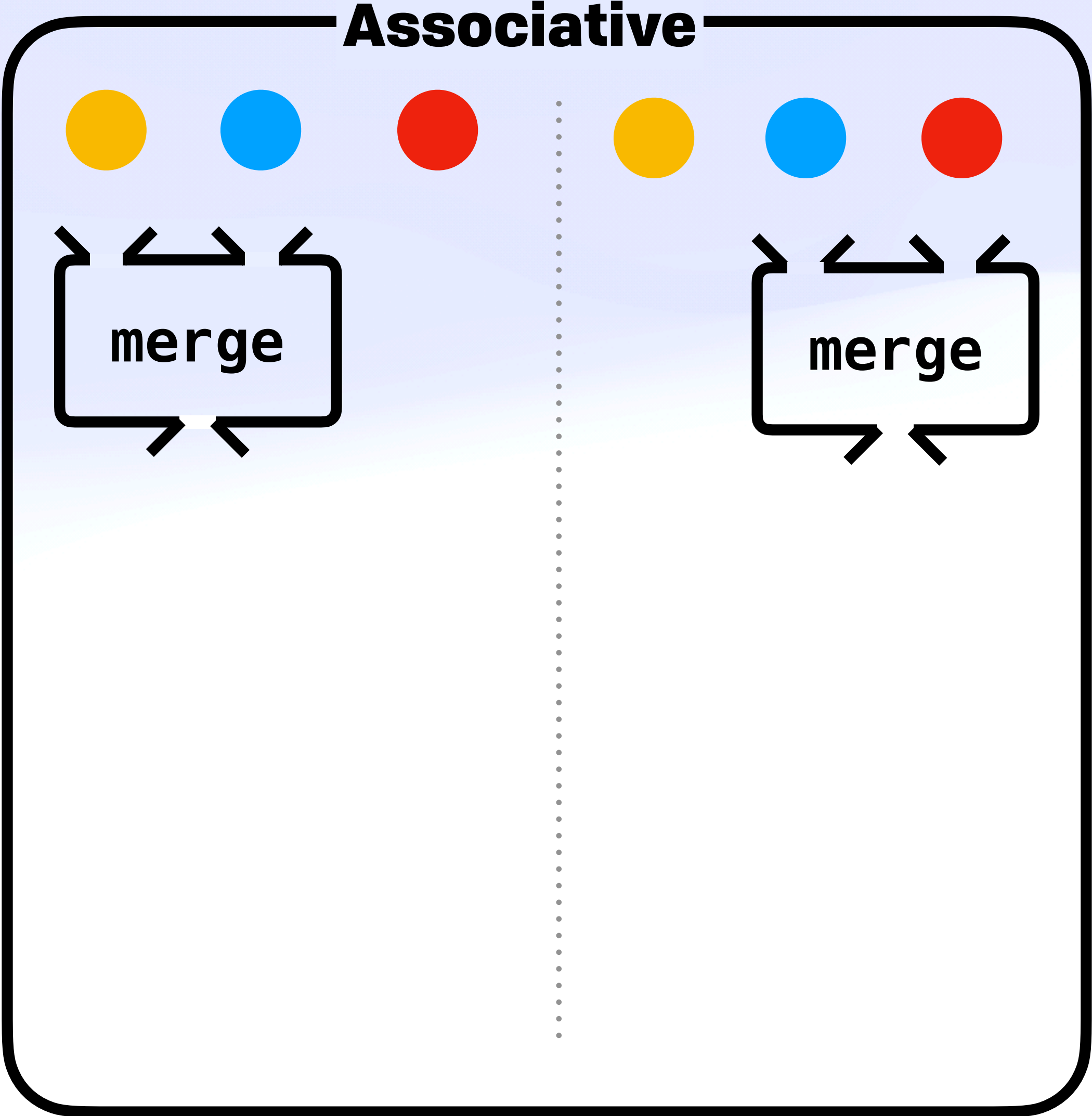
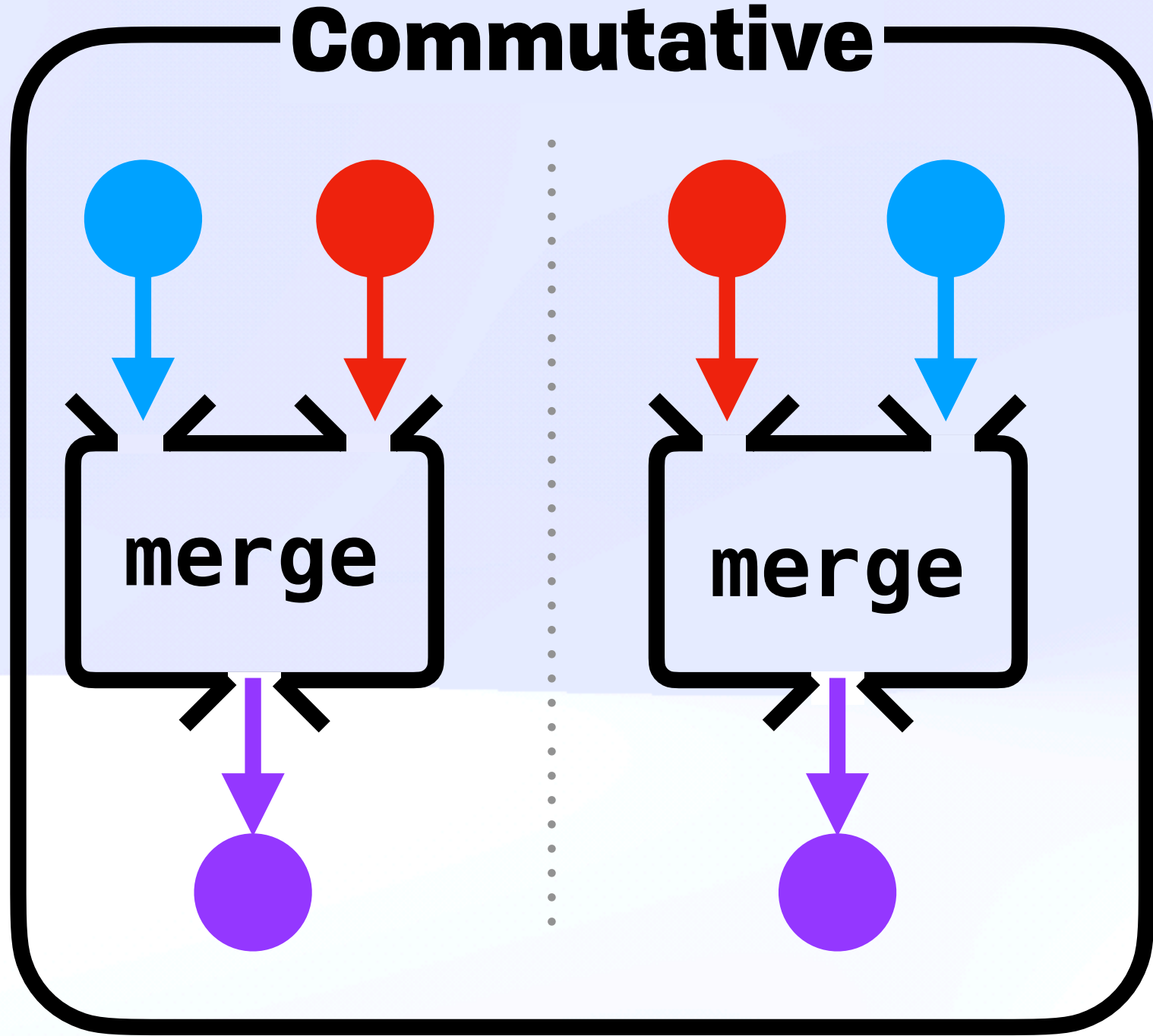


Properties Save the Day



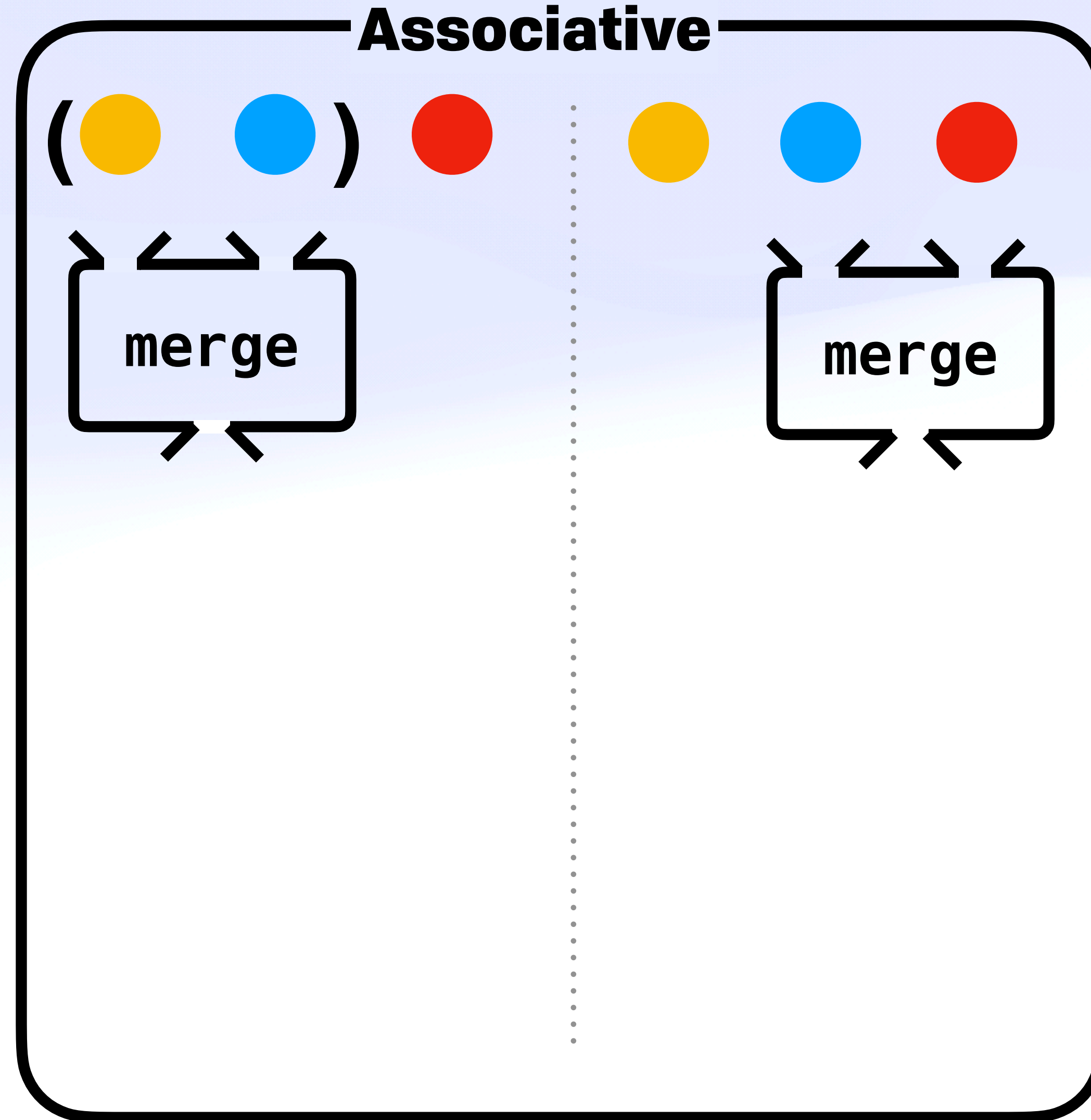
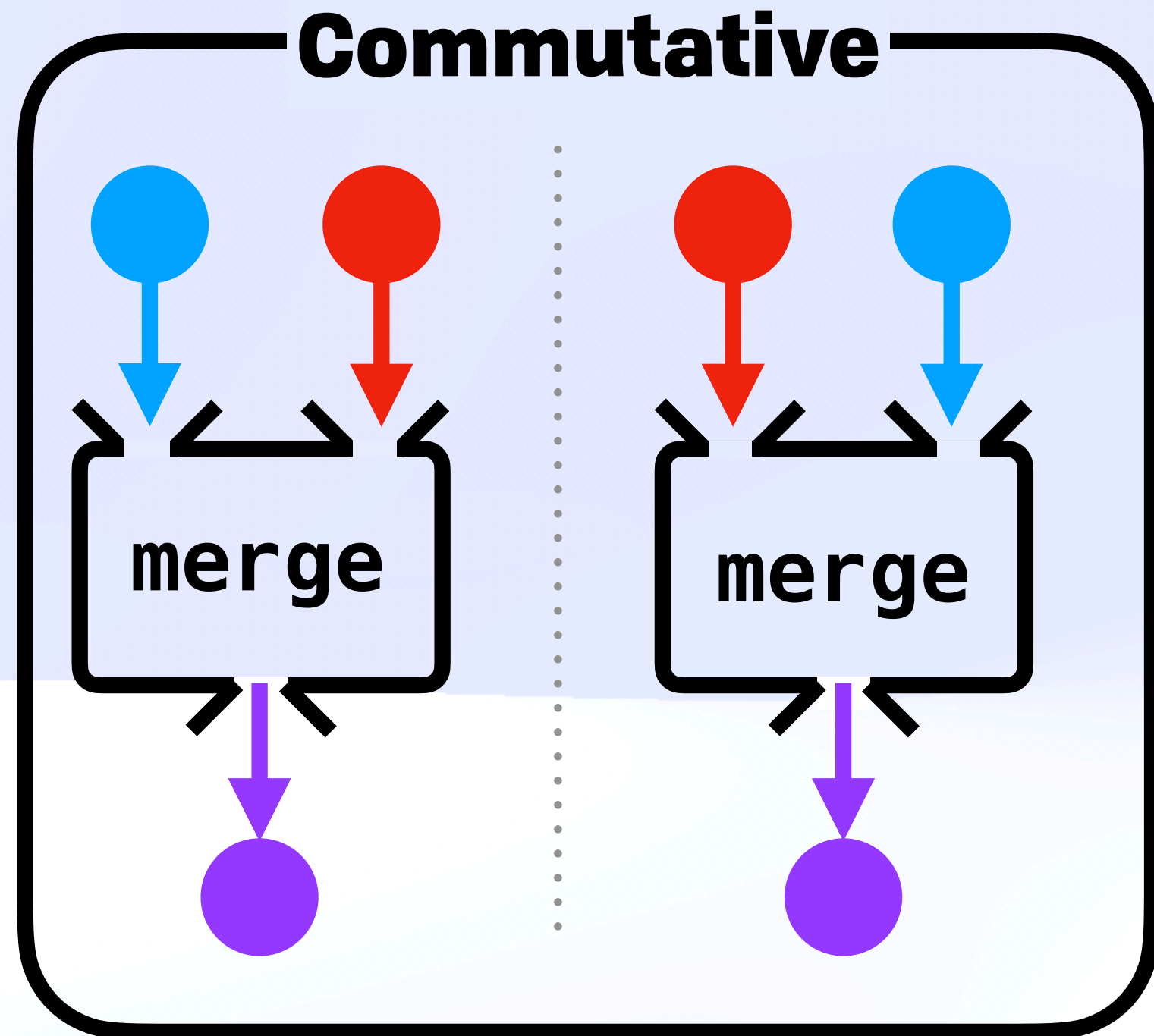
Fault-Safe Concurrency

Properties Save the Day



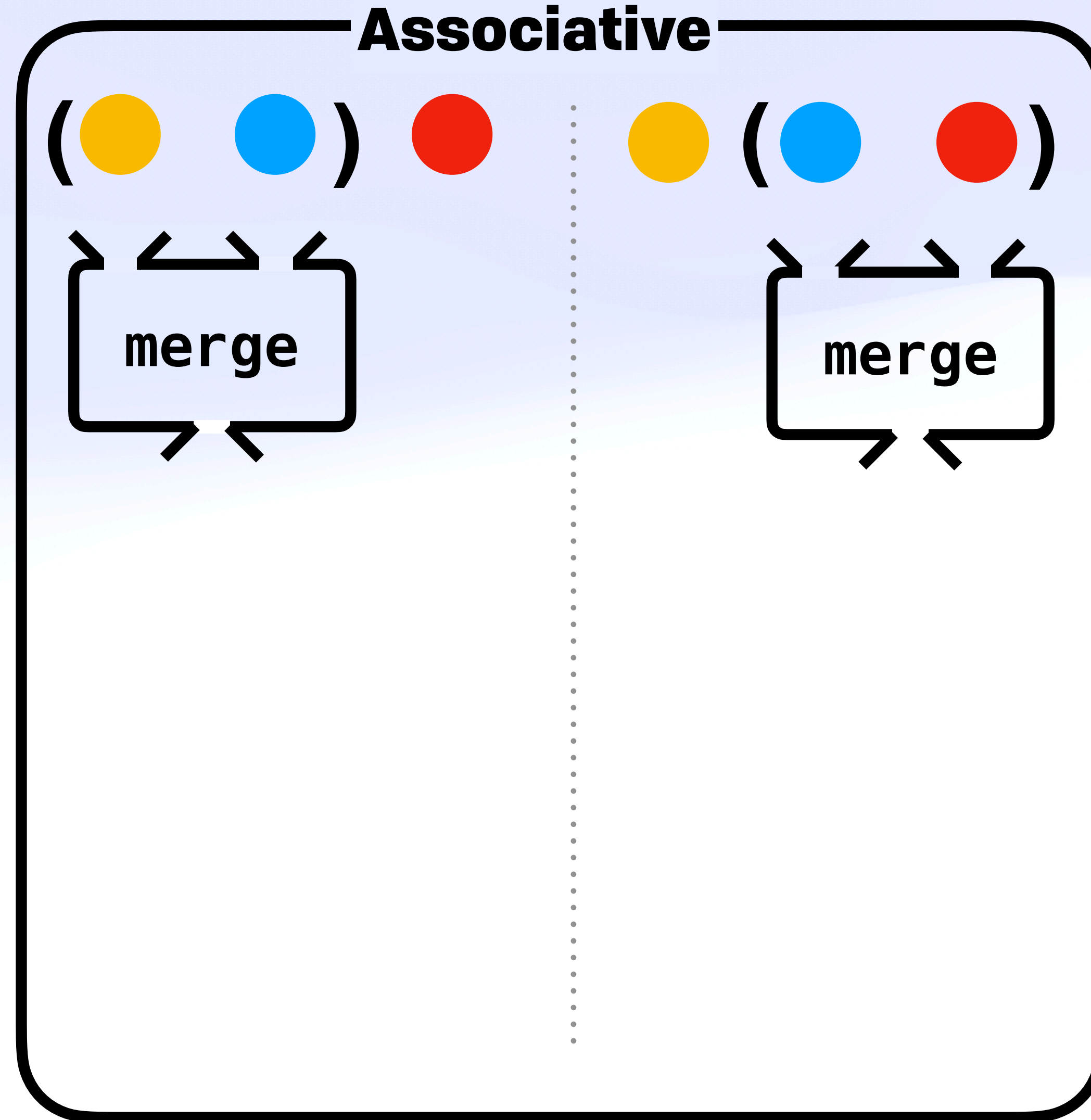
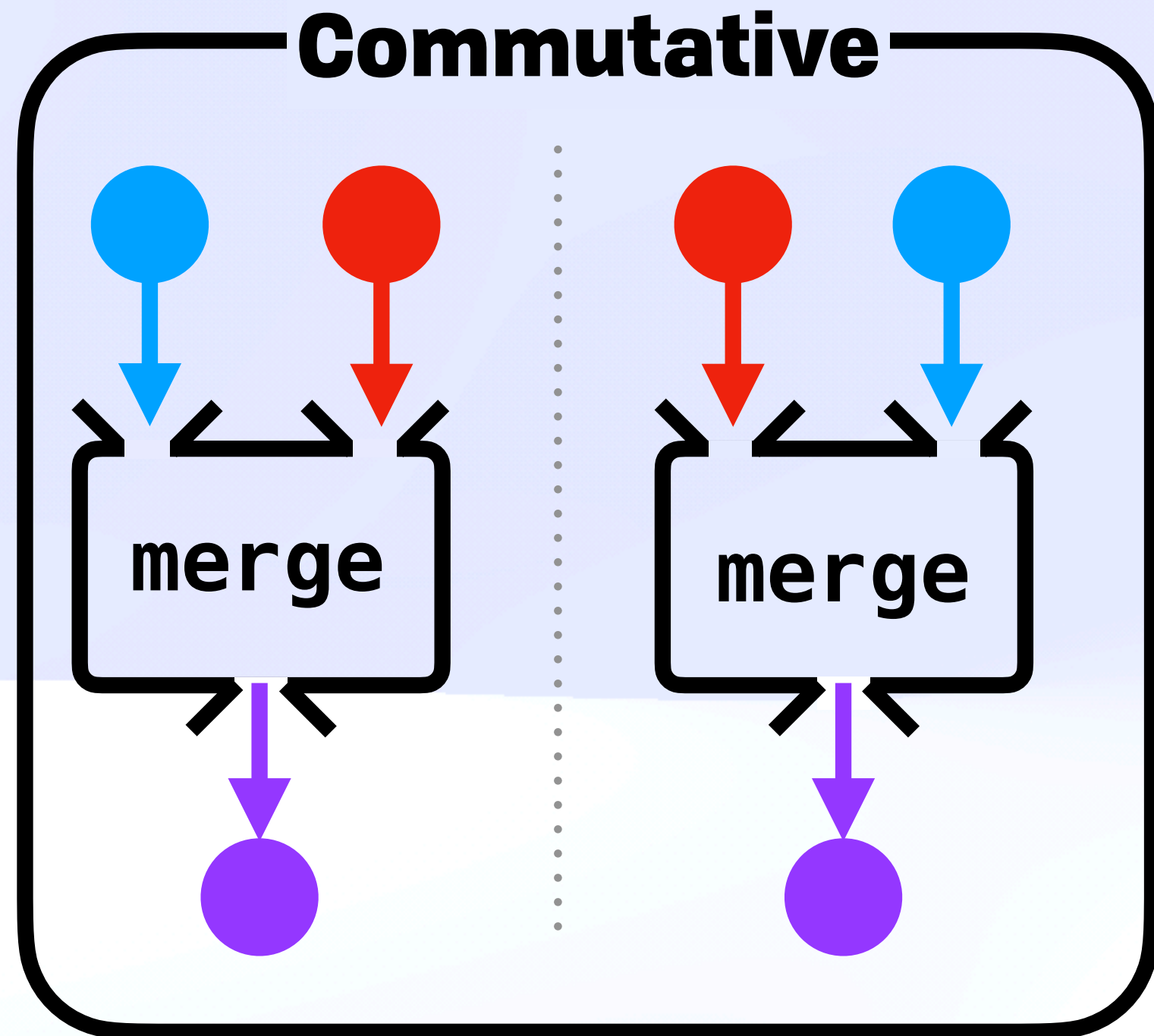
Fault-Safe Concurrency

Properties Save the Day



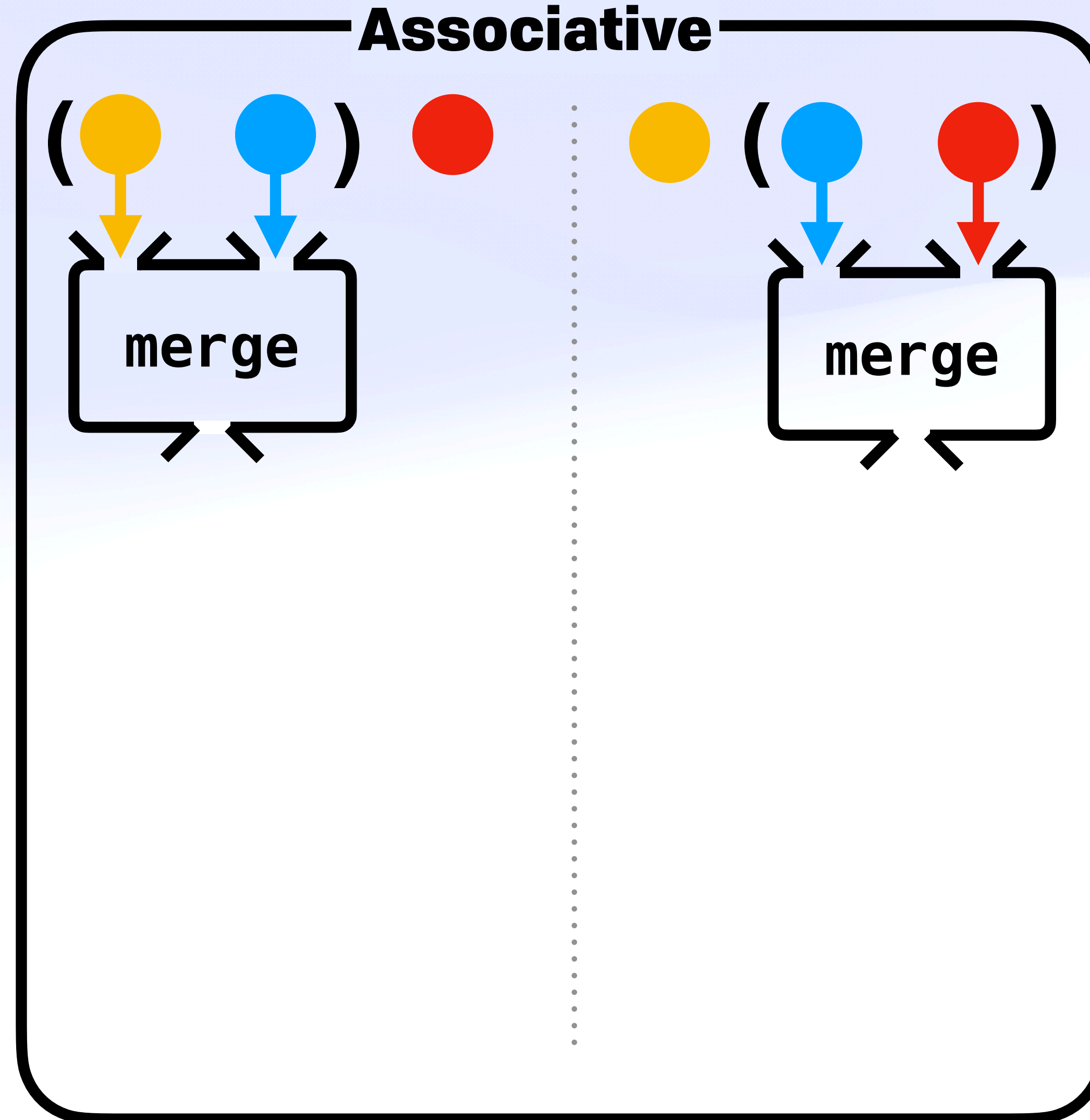
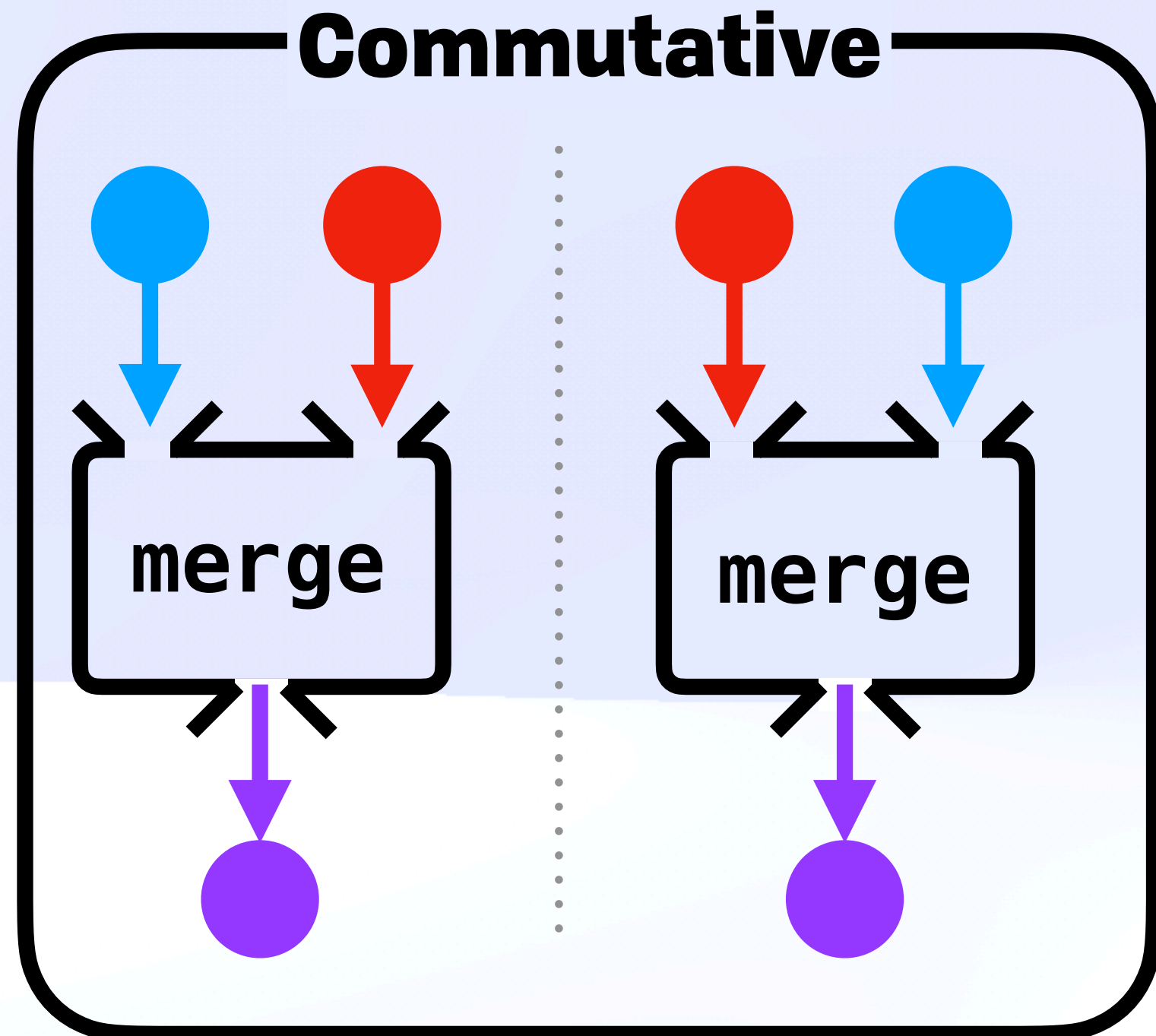
Fault-Safe Concurrency

Properties Save the Day



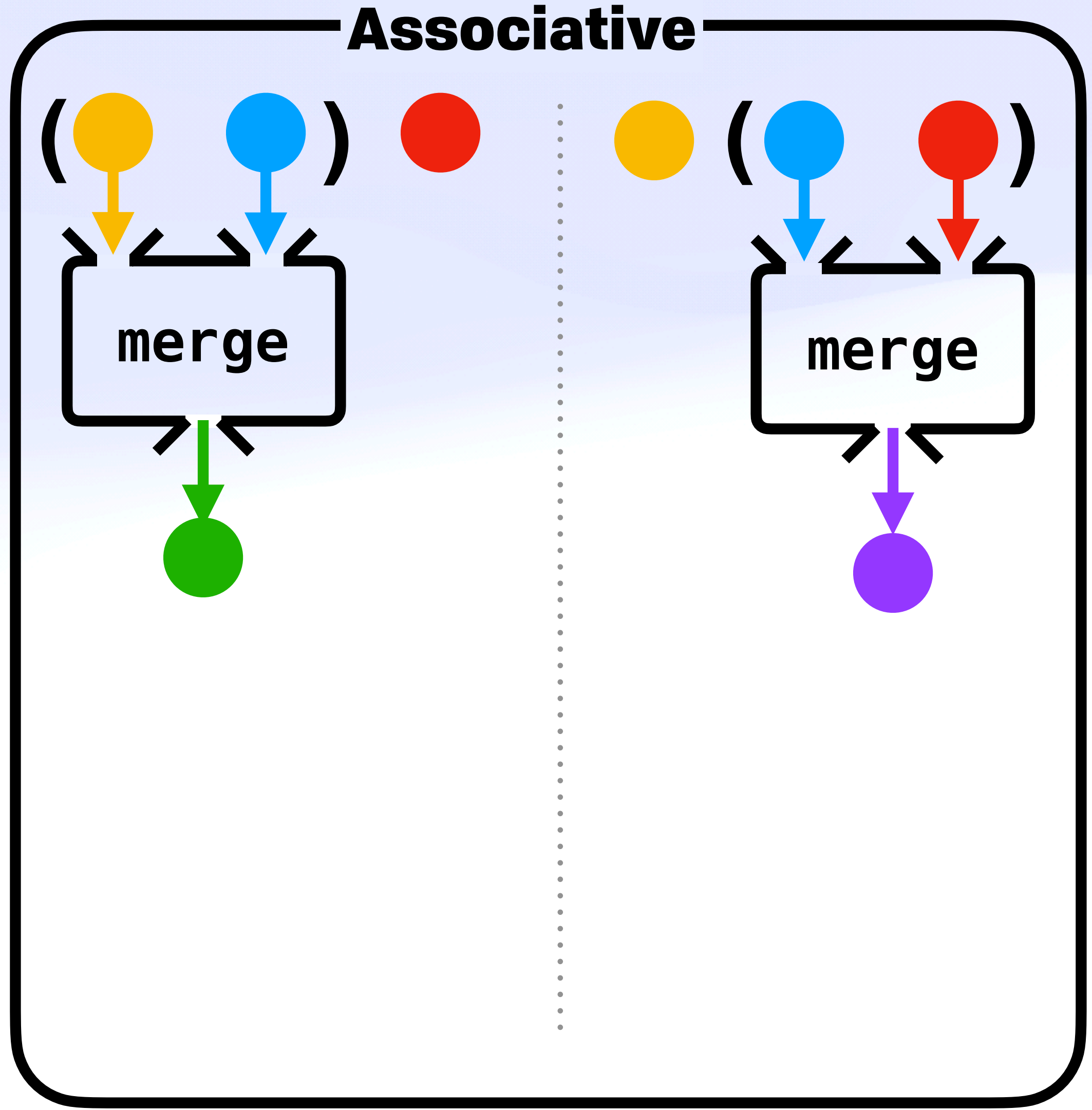
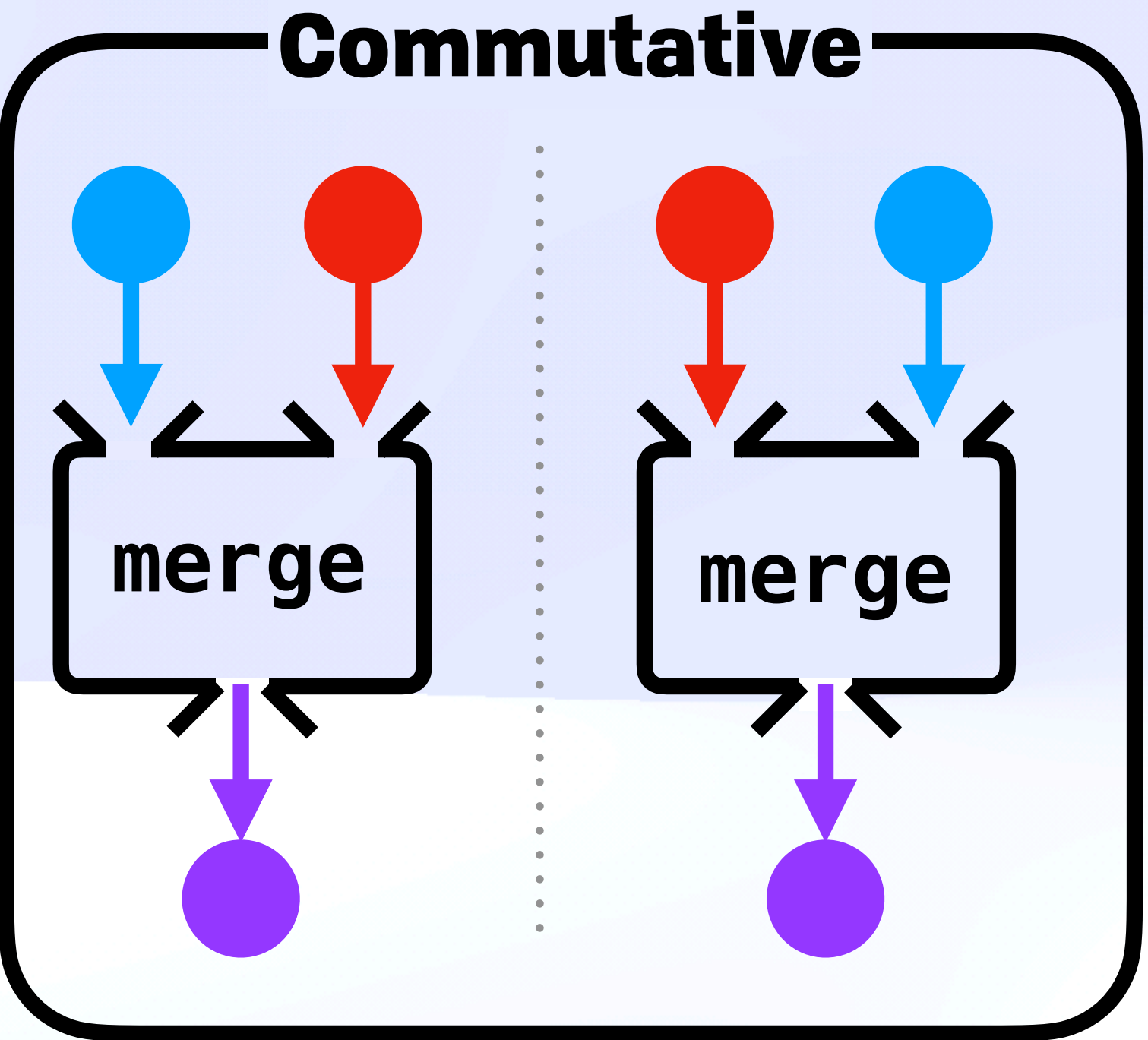
Fault-Safe Concurrency

Properties Save the Day



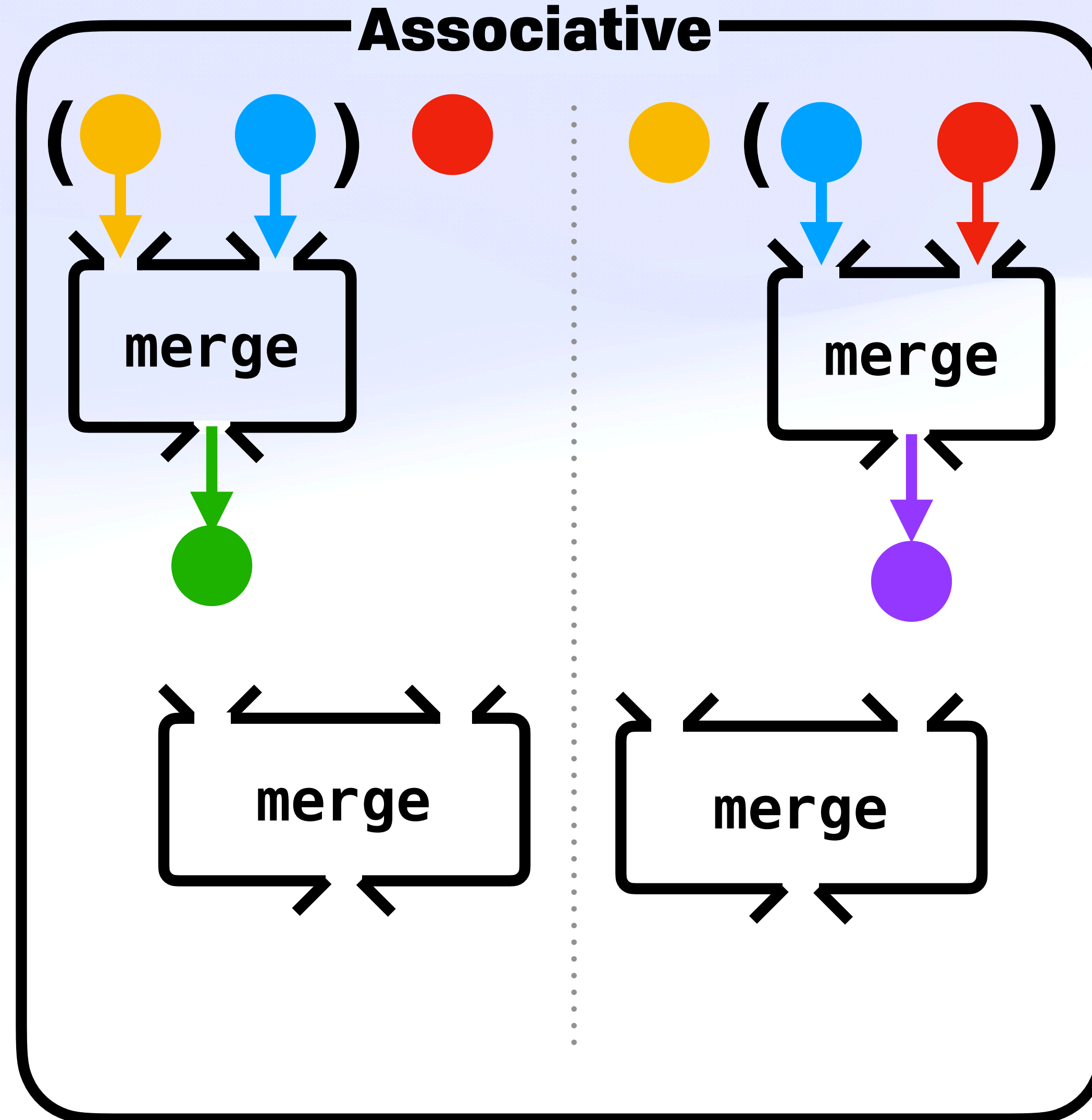
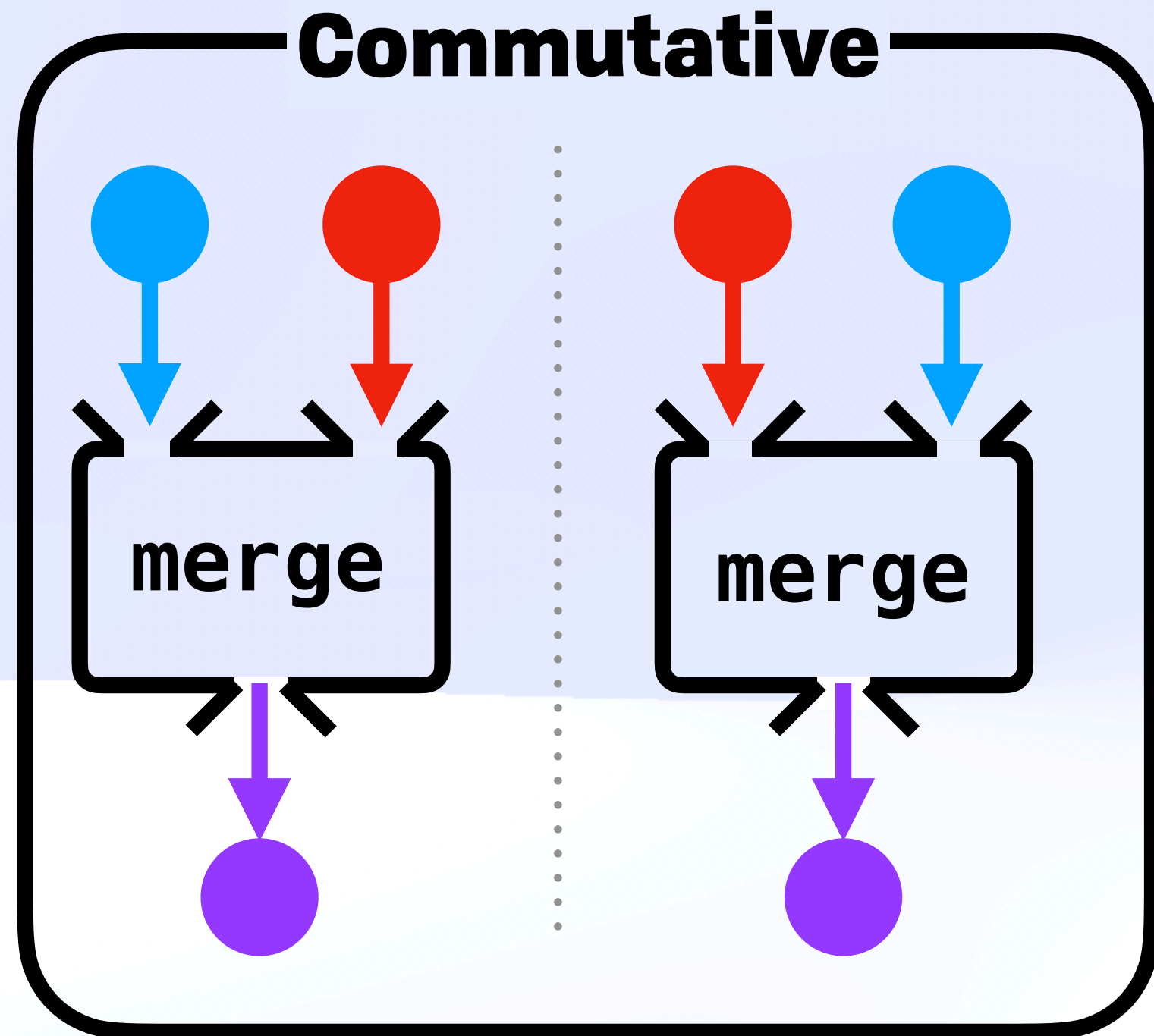
Fault-Safe Concurrency

Properties Save the Day



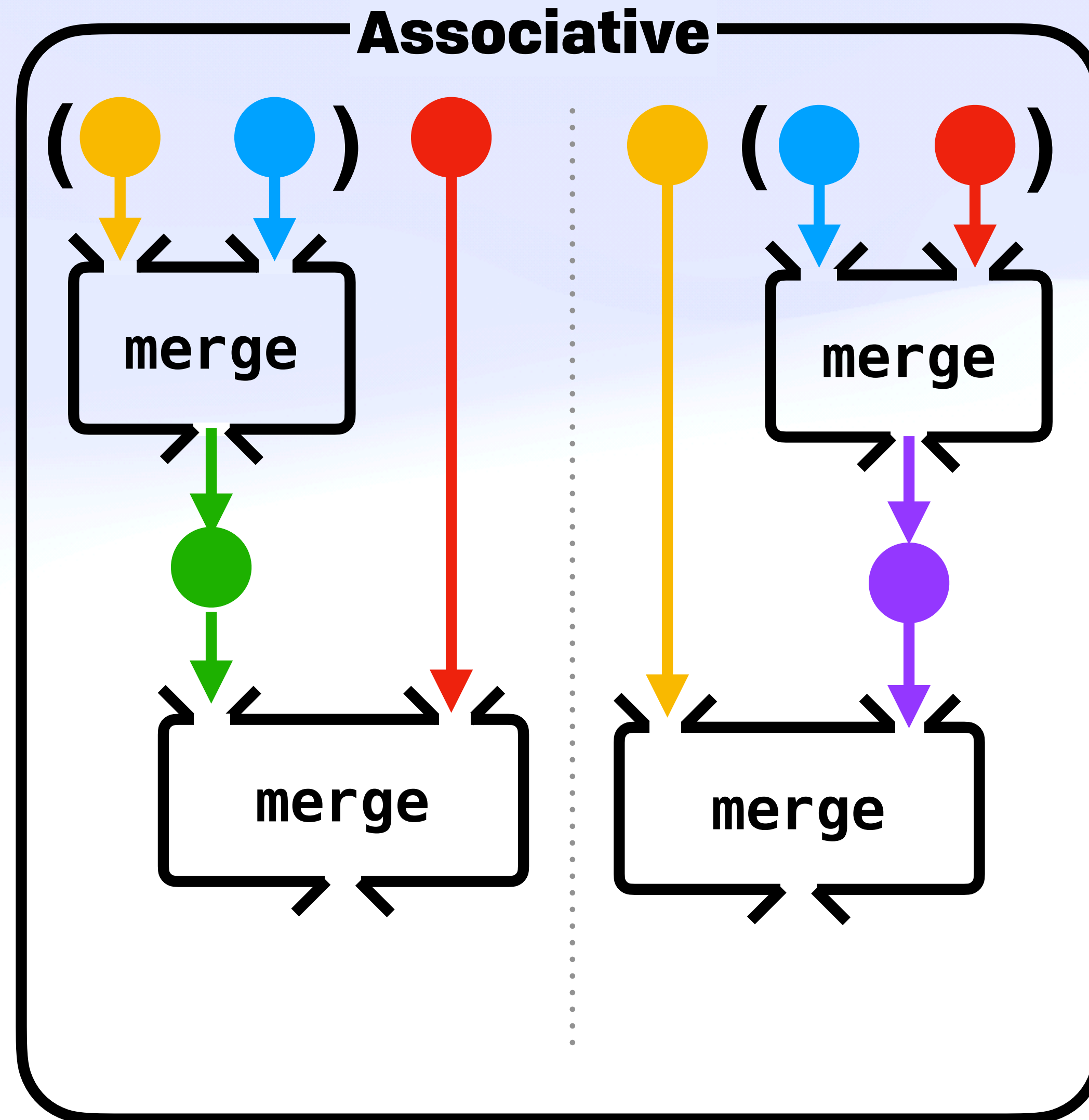
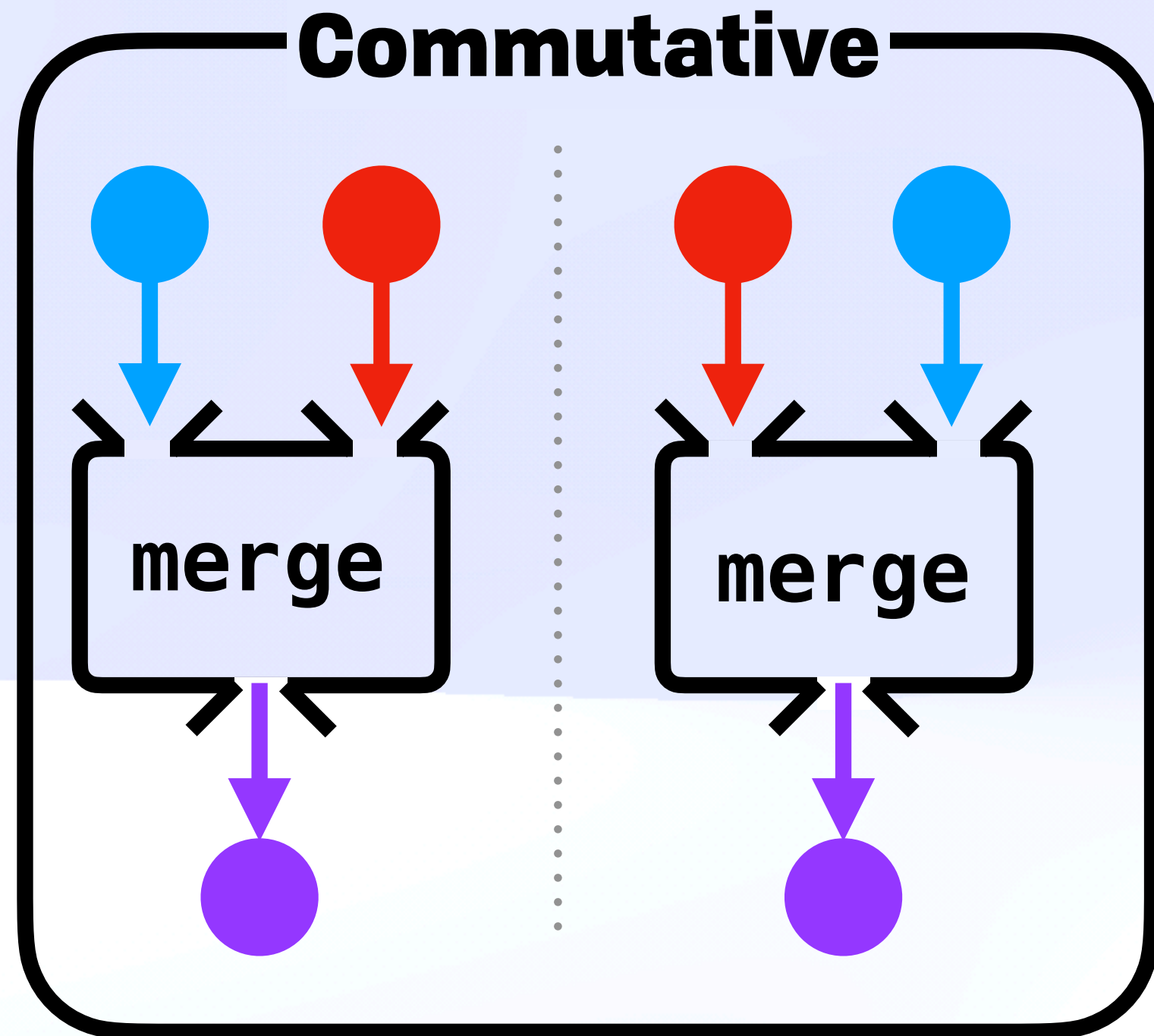
Fault-Safe Concurrency

Properties Save the Day



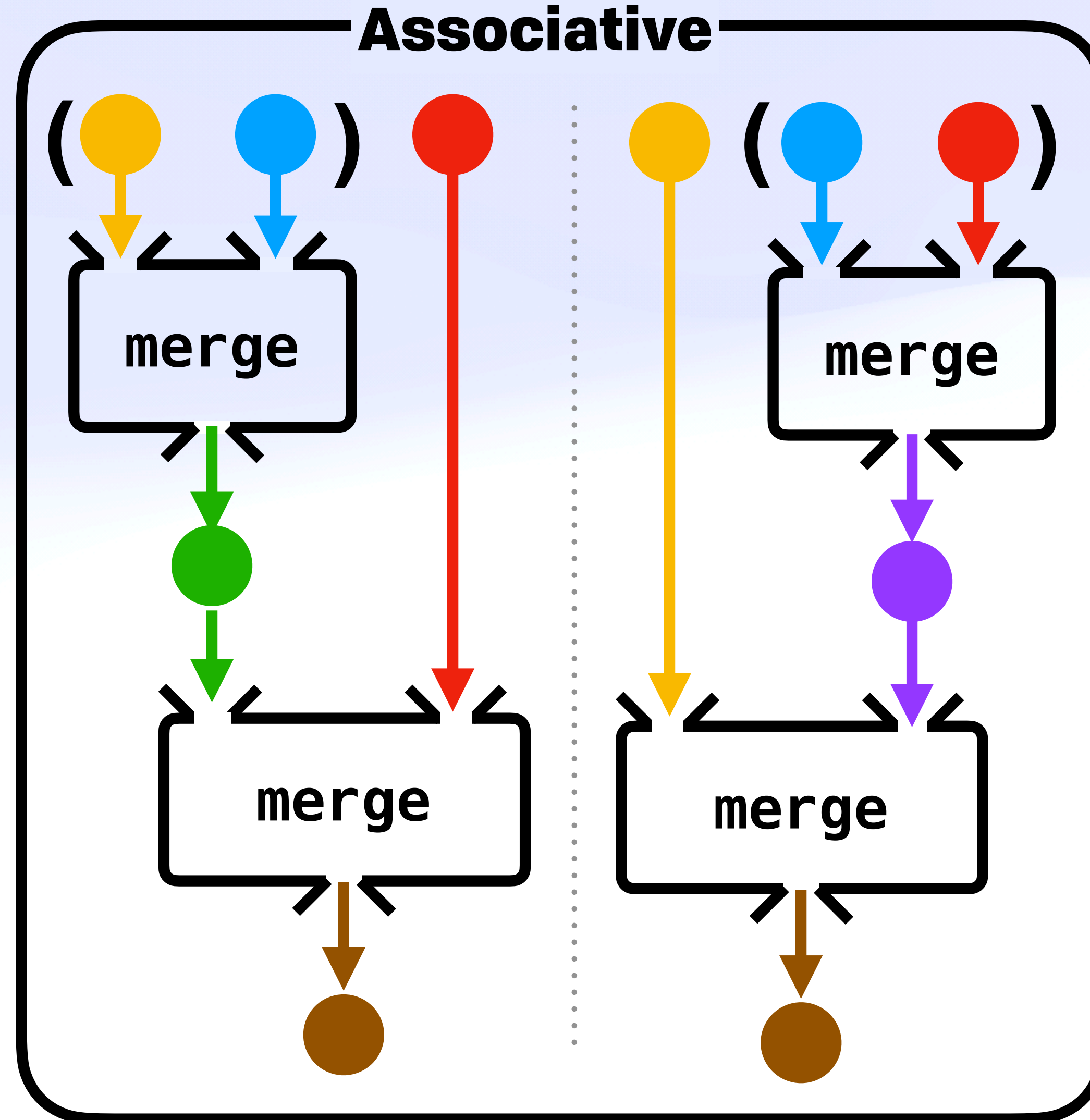
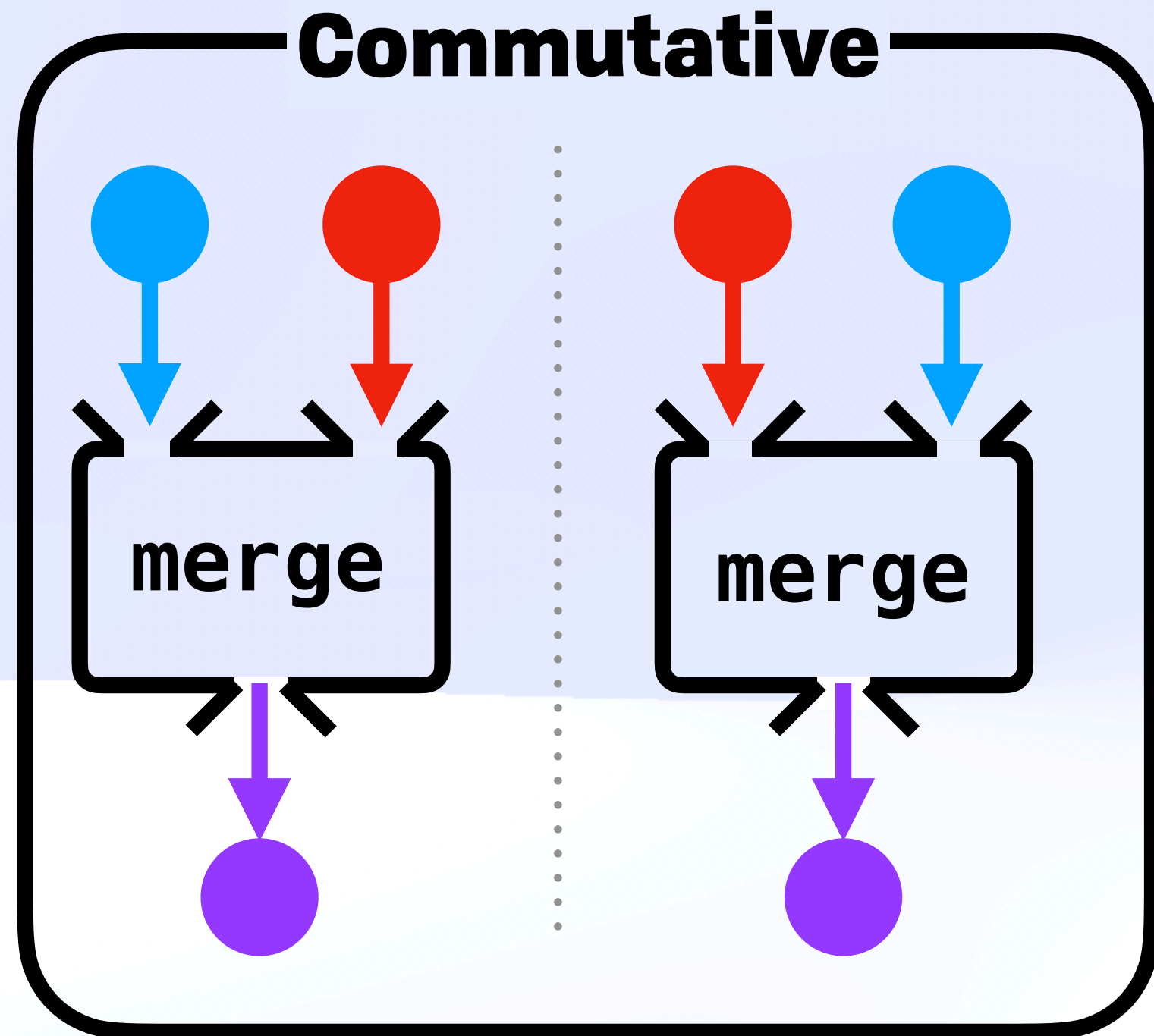
Fault-Safe Concurrency

Properties Save the Day



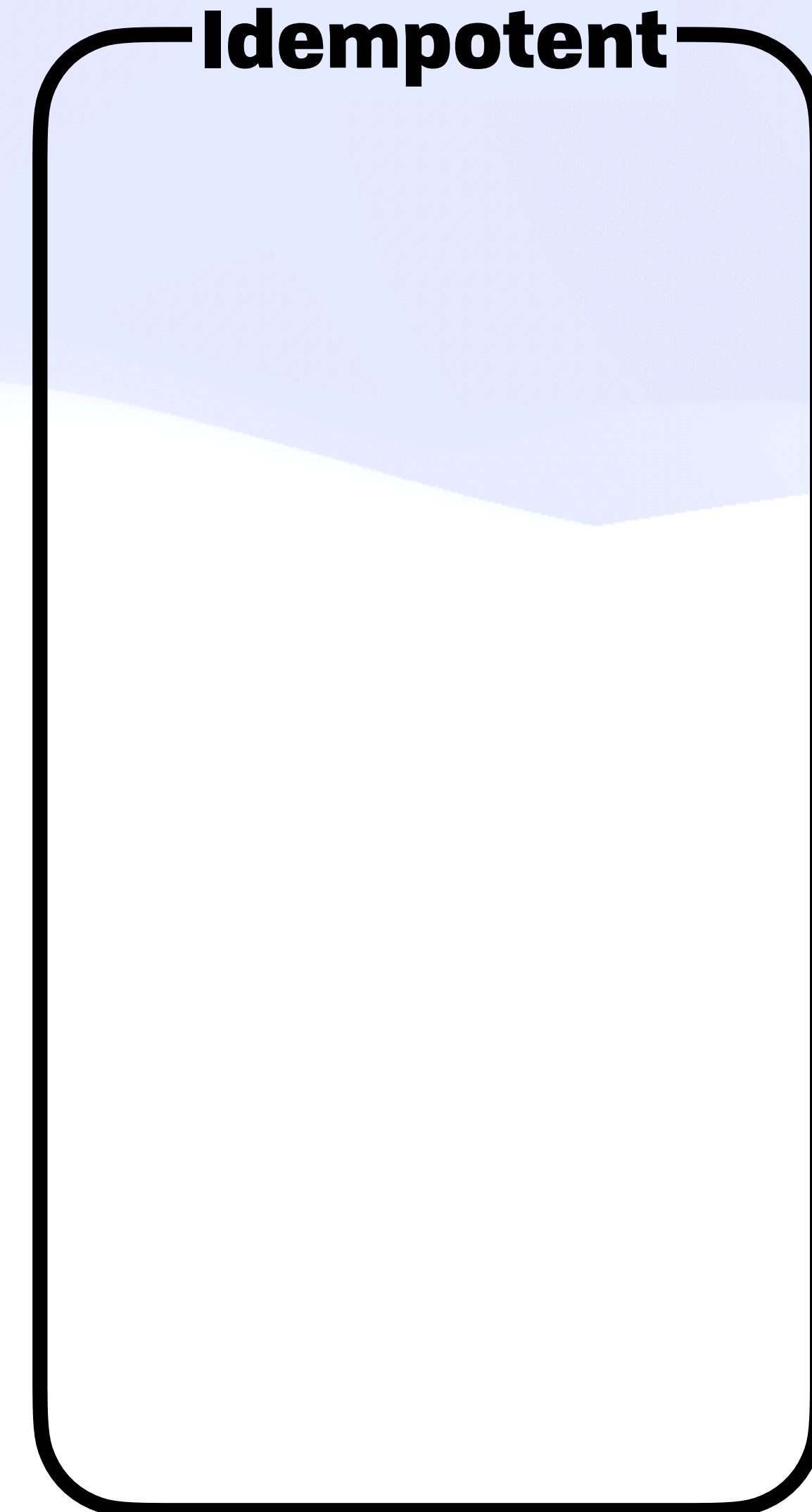
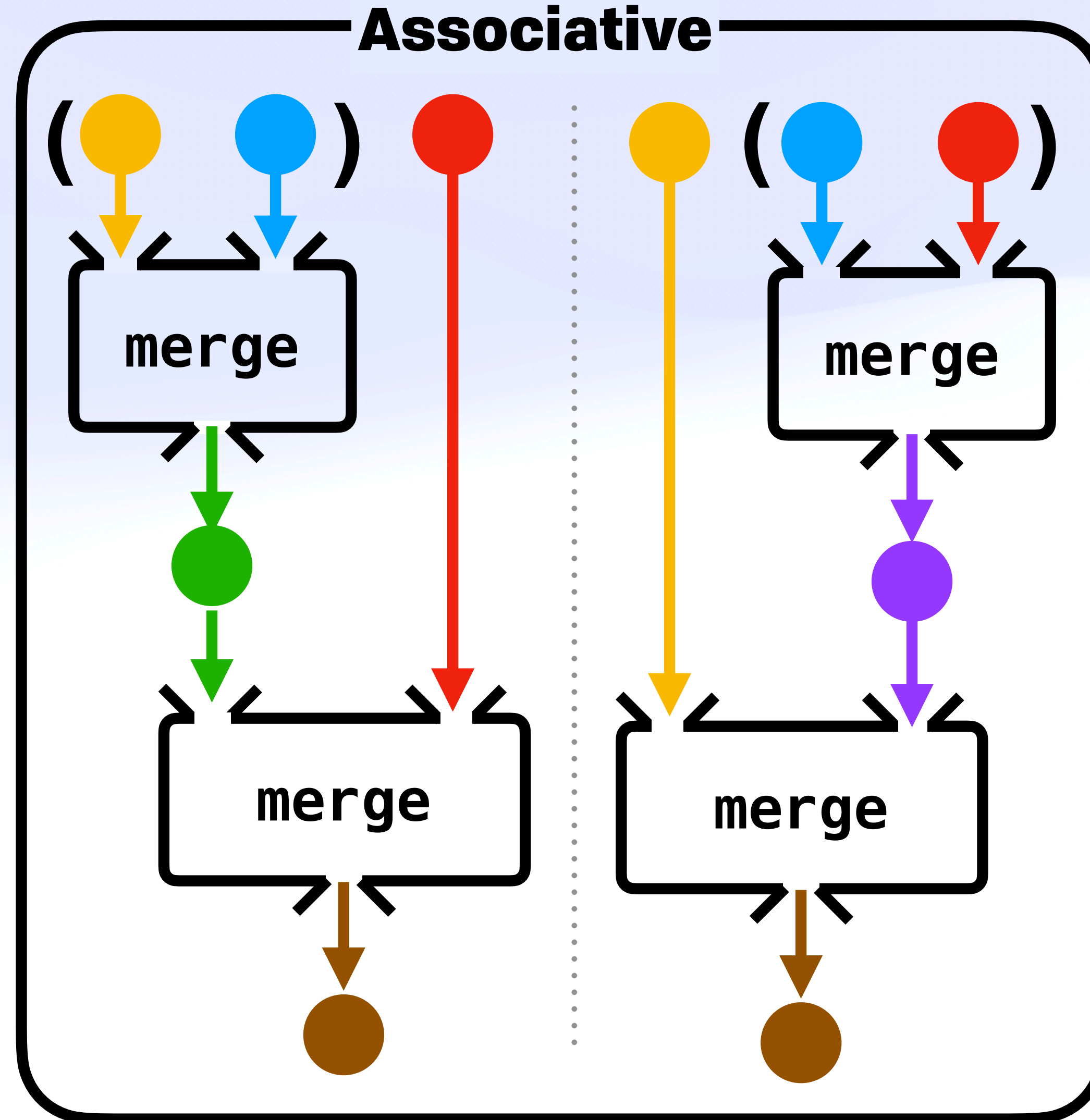
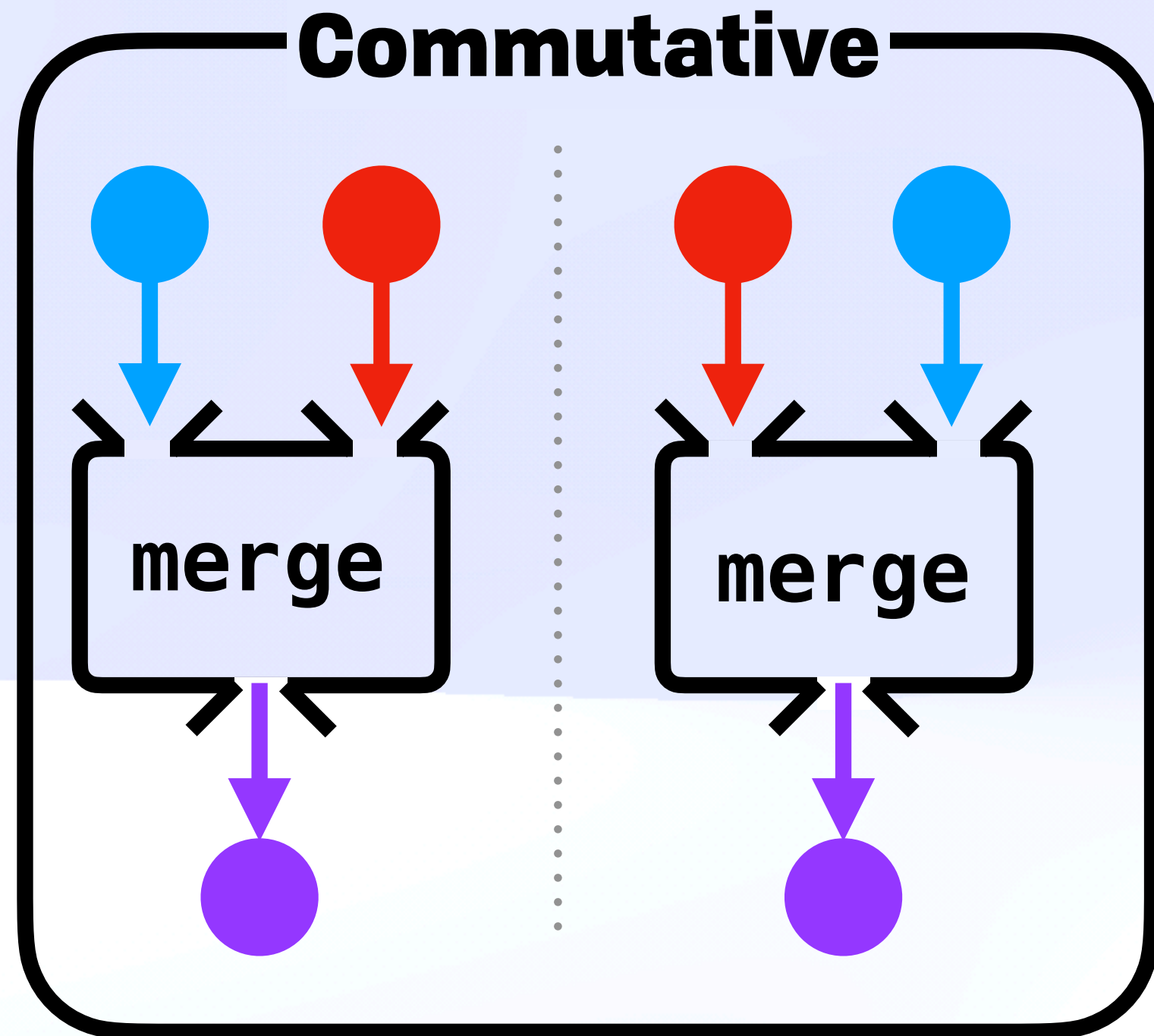
Fault-Safe Concurrency

Properties Save the Day



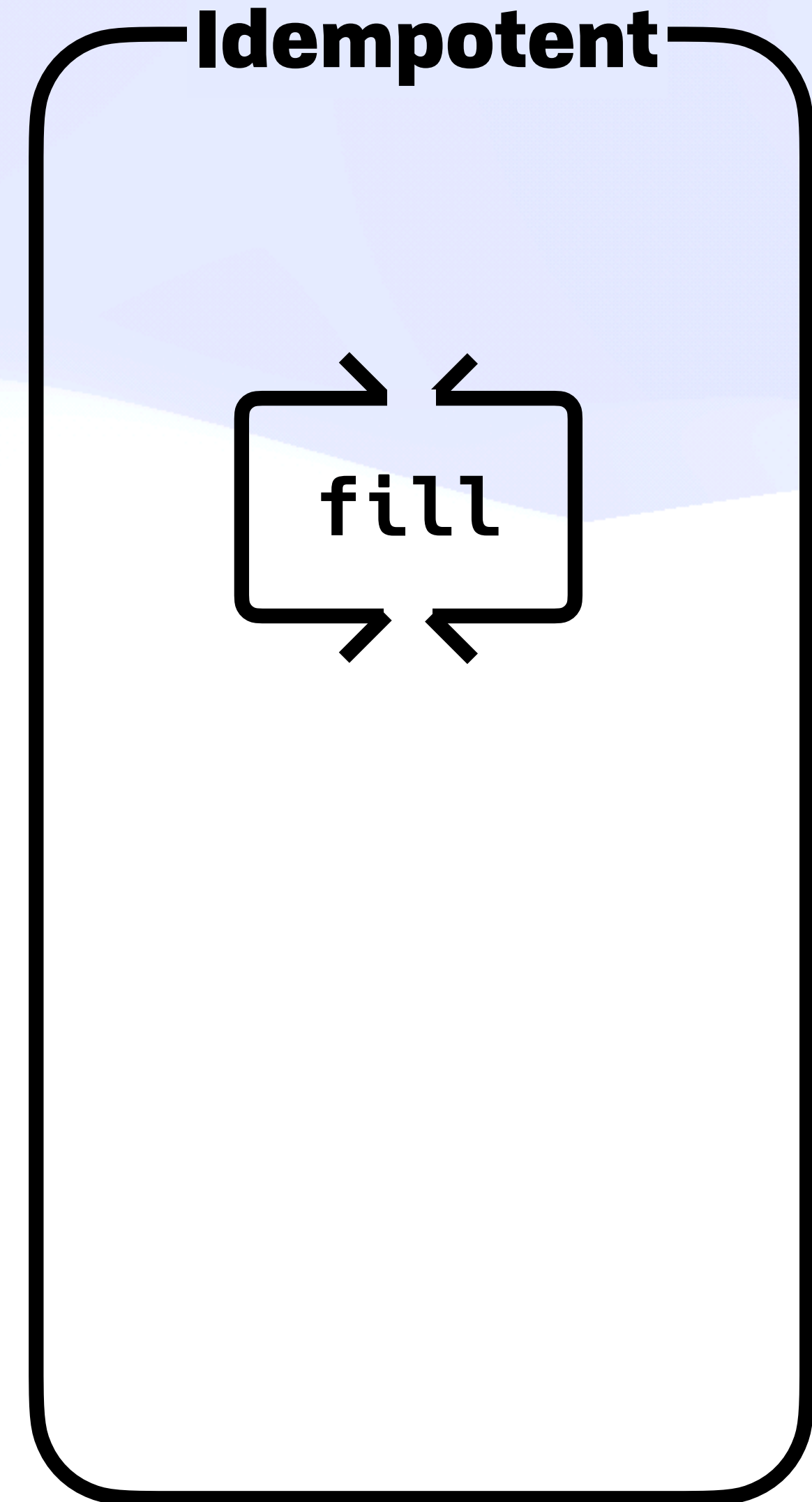
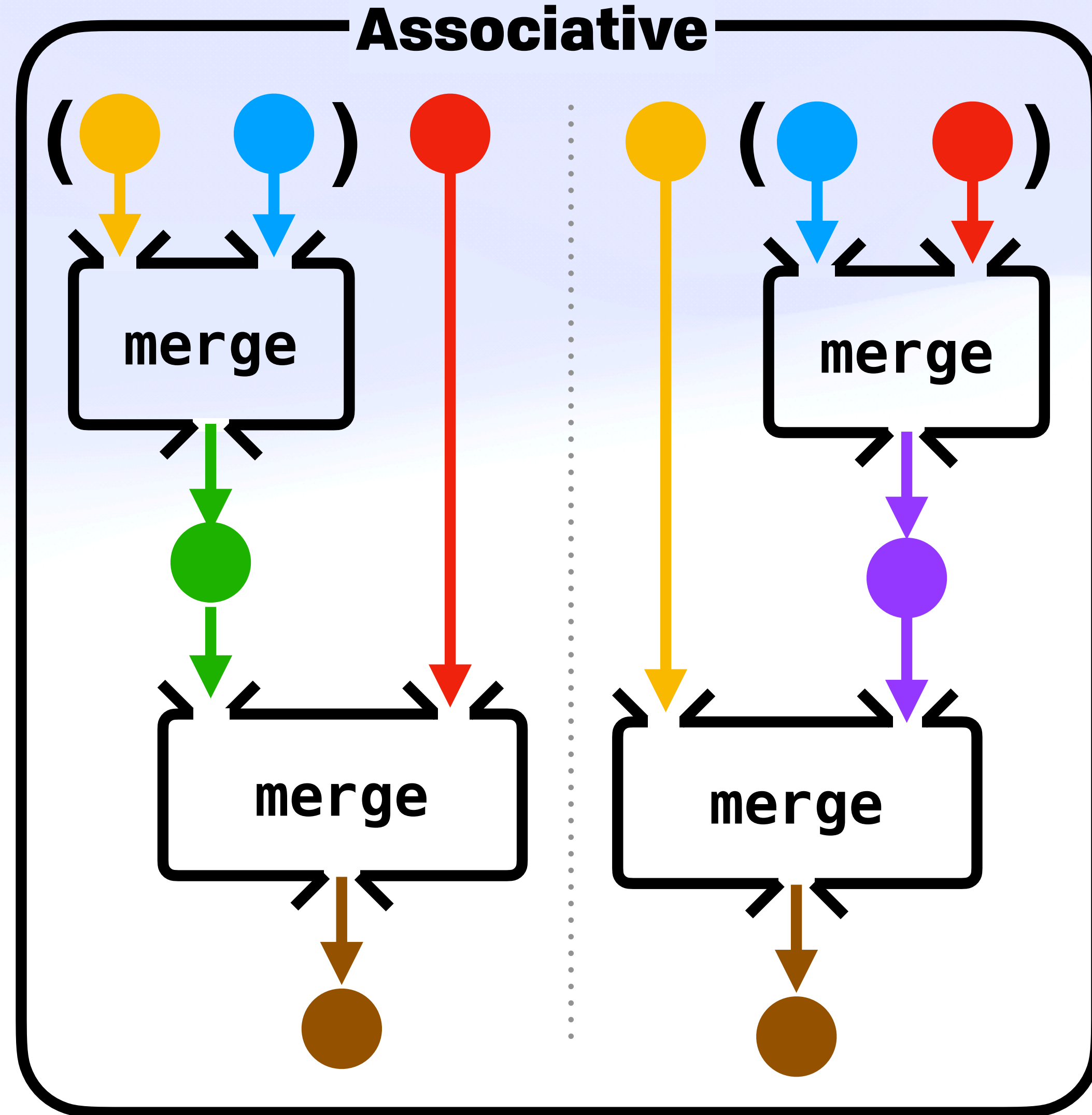
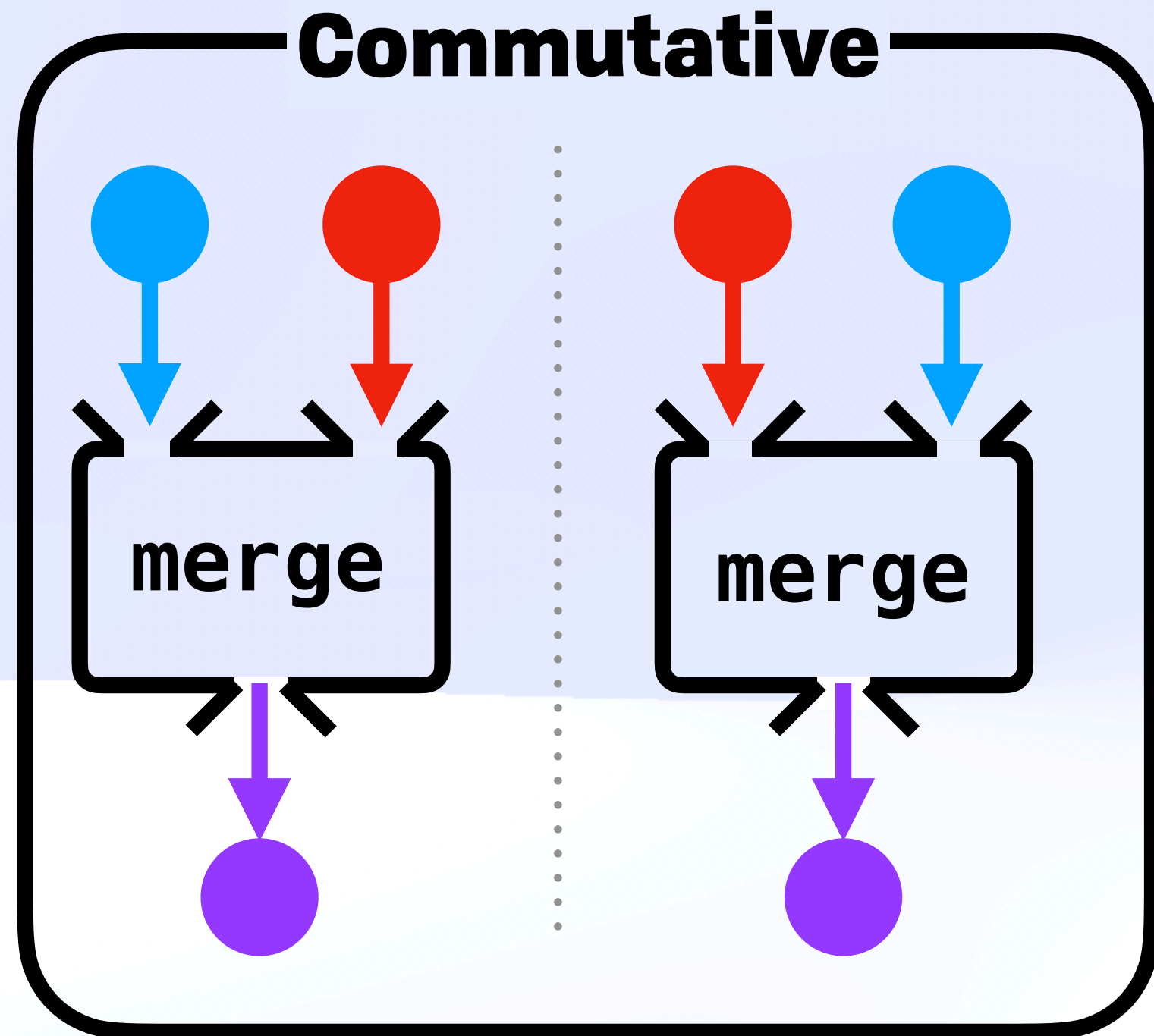
Fault-Safe Concurrency

Properties Save the Day



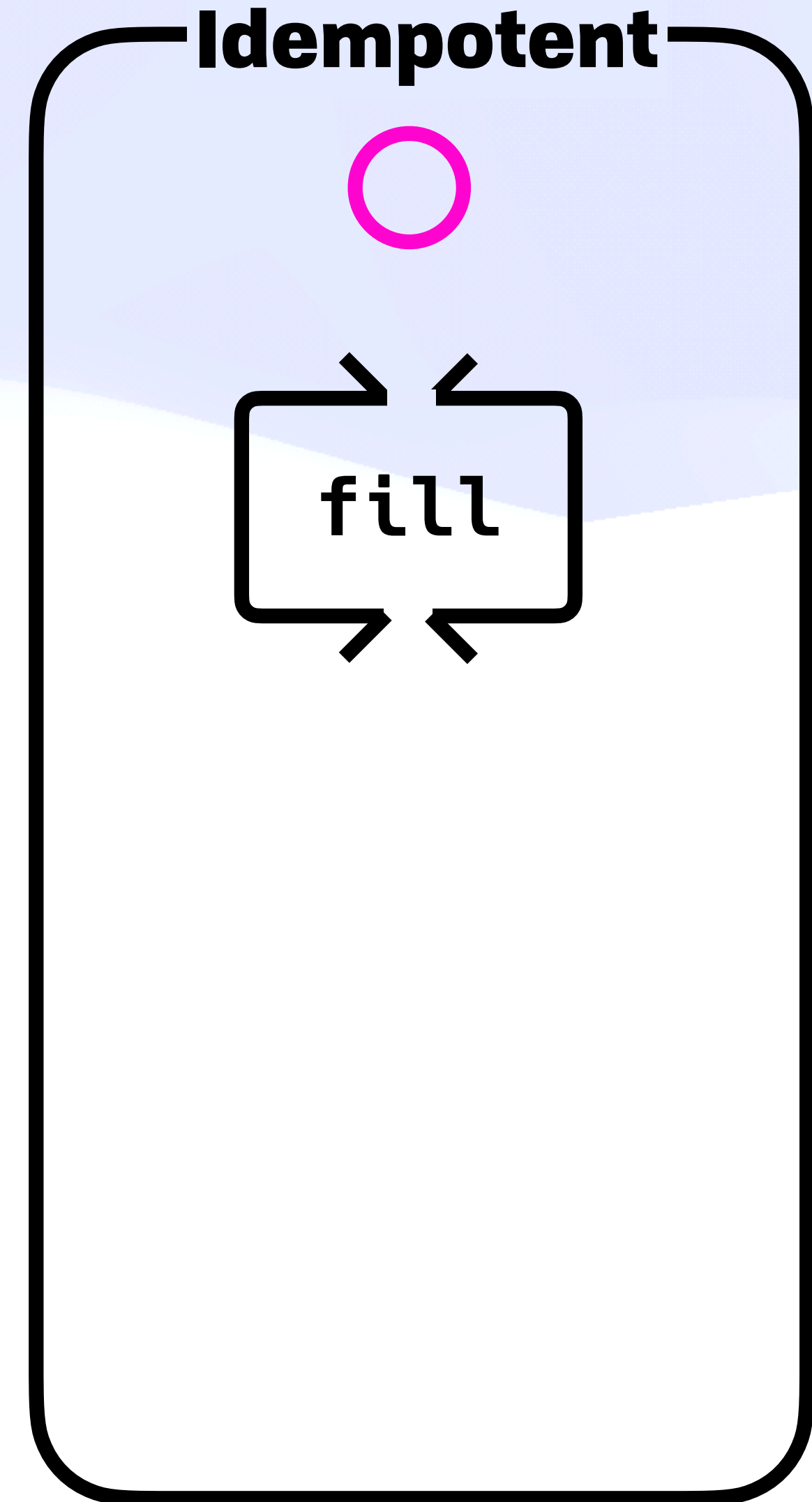
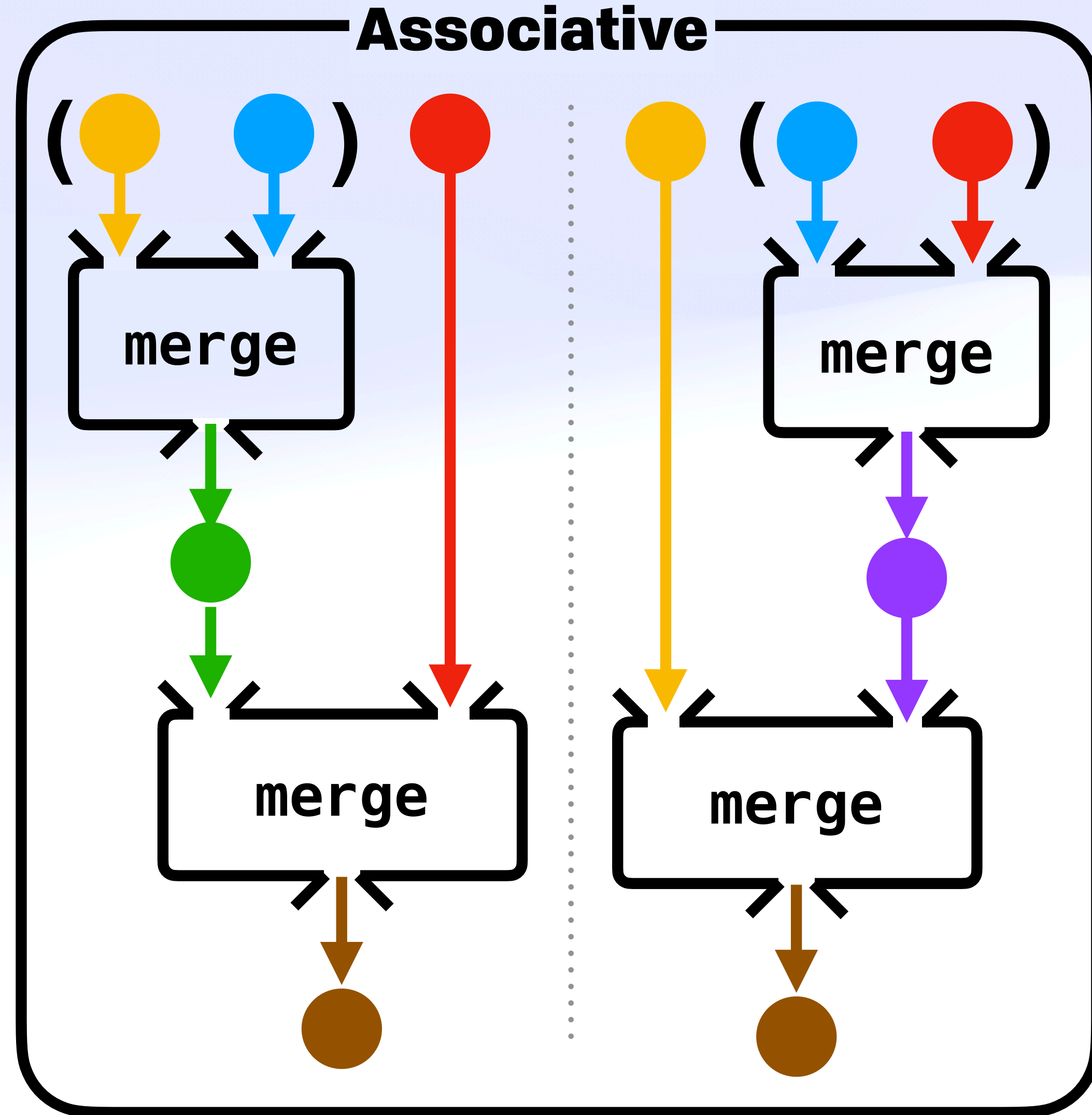
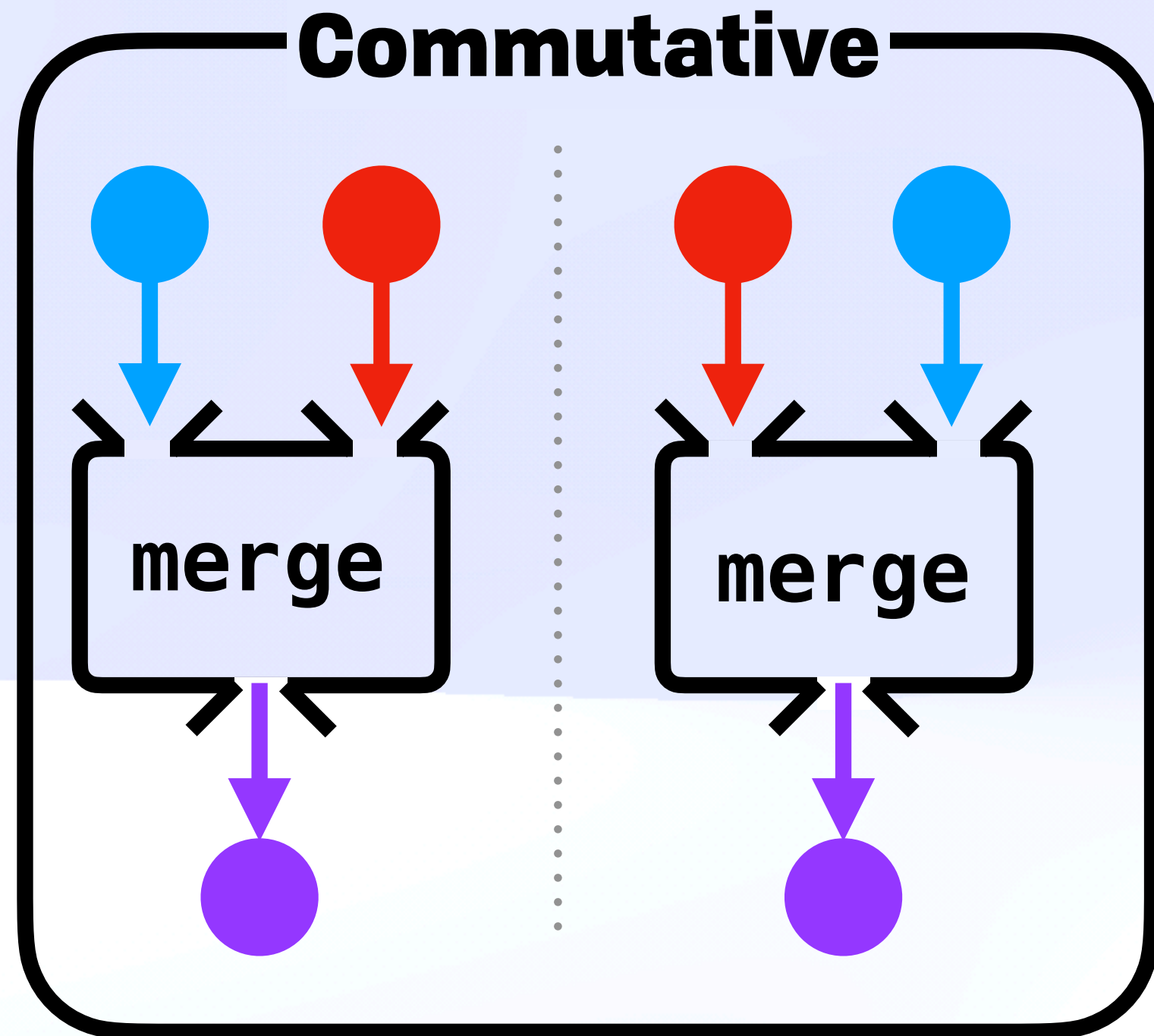
Fault-Safe Concurrency

Properties Save the Day



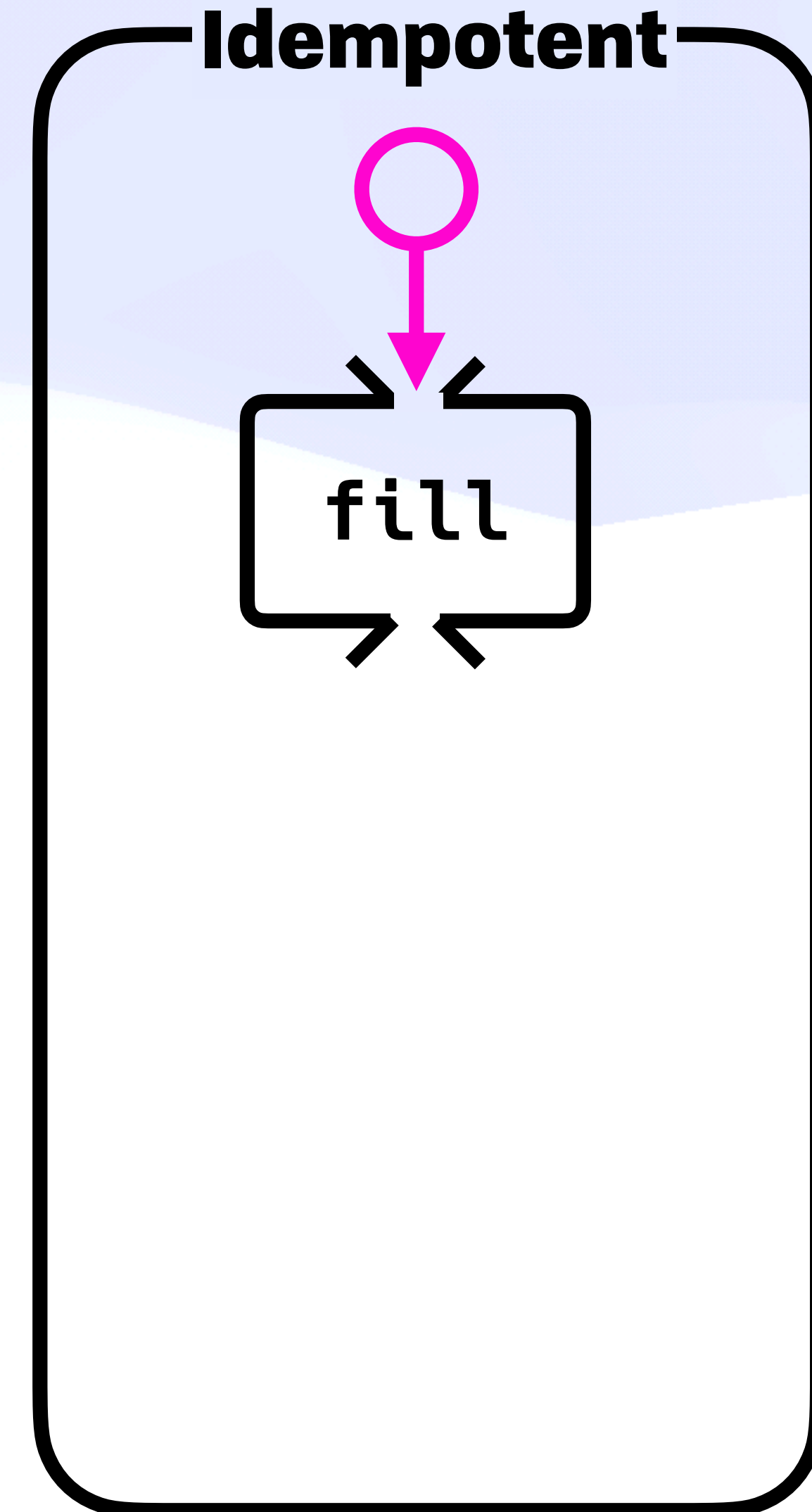
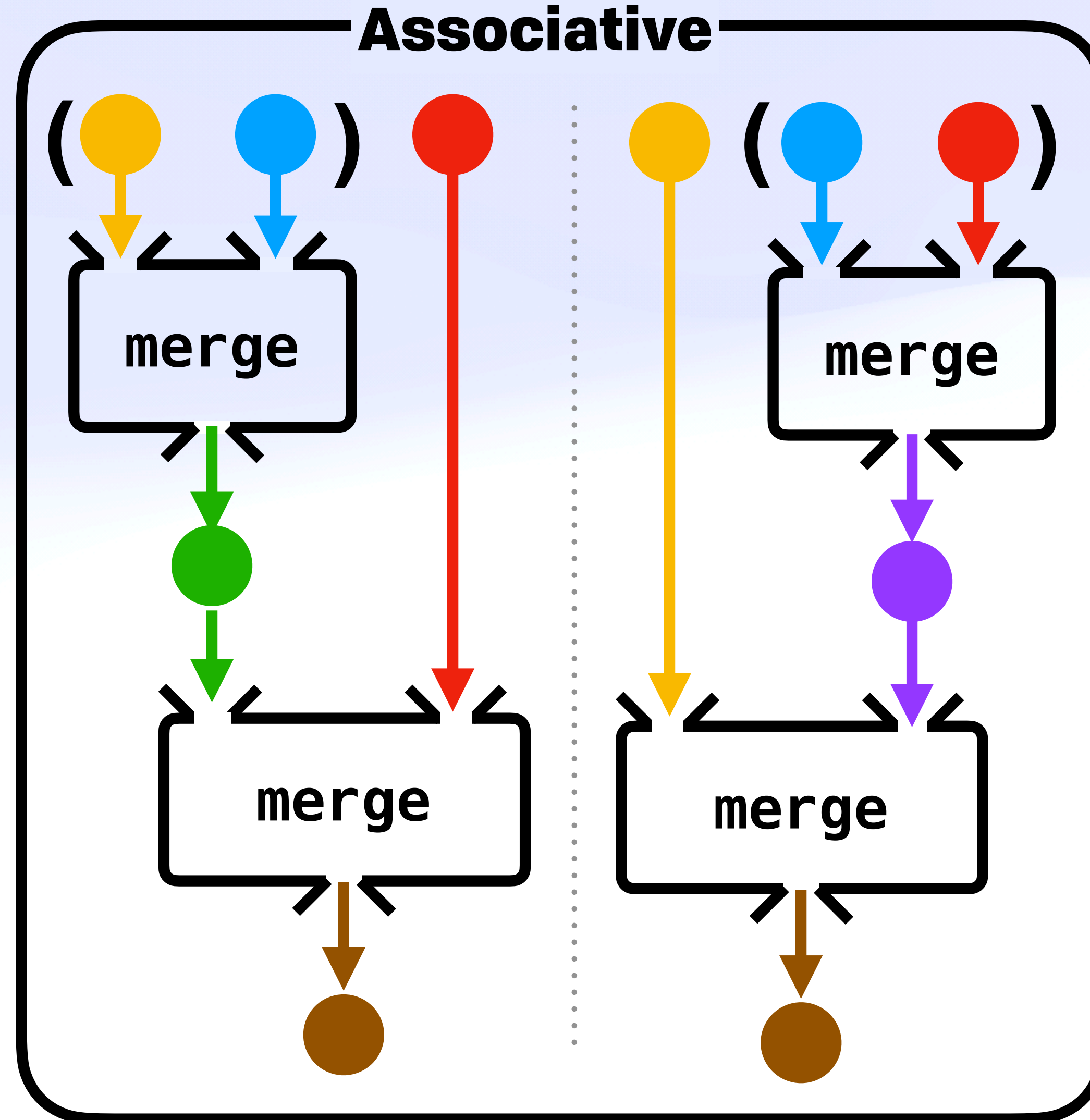
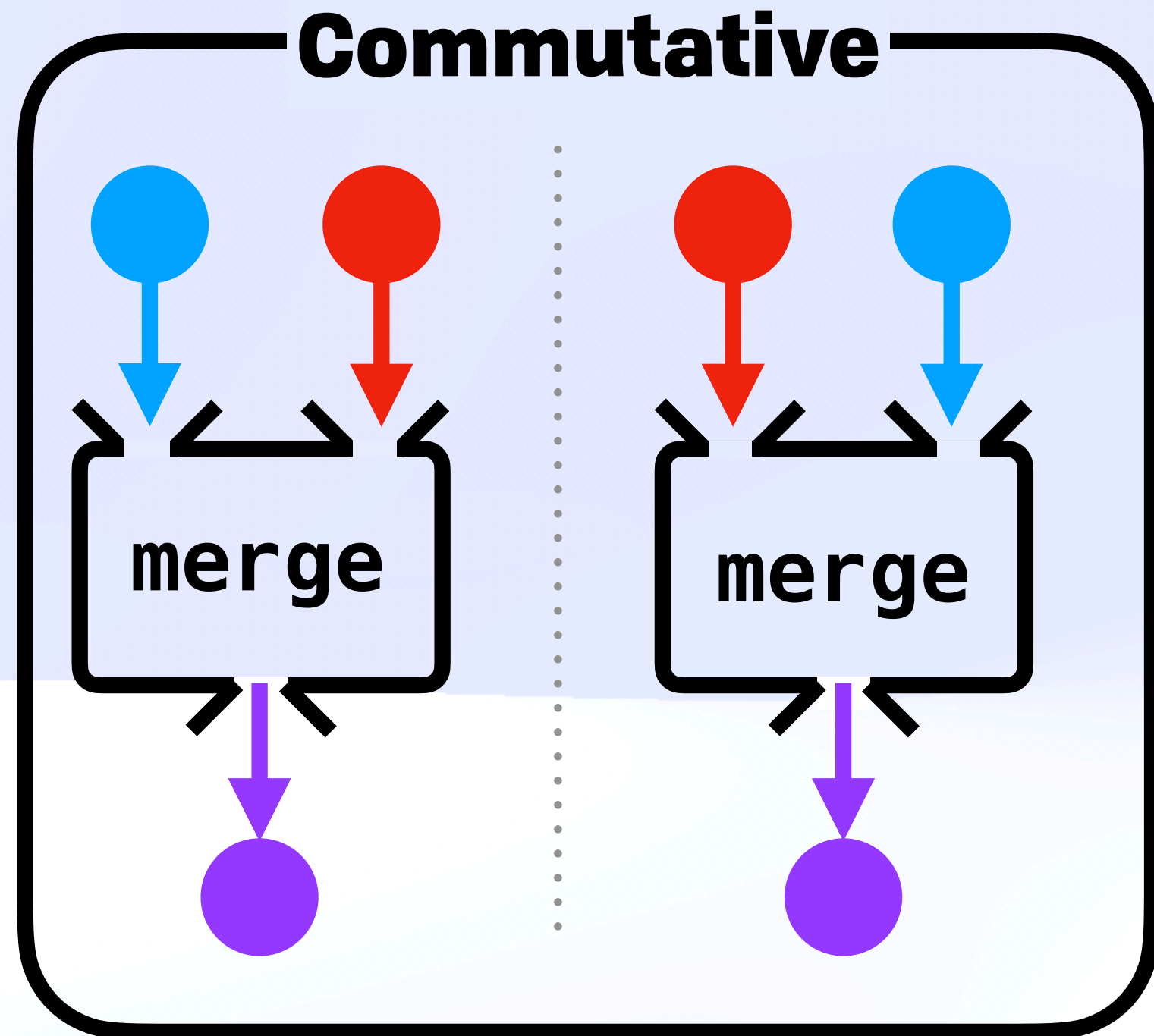
Fault-Safe Concurrency

Properties Save the Day



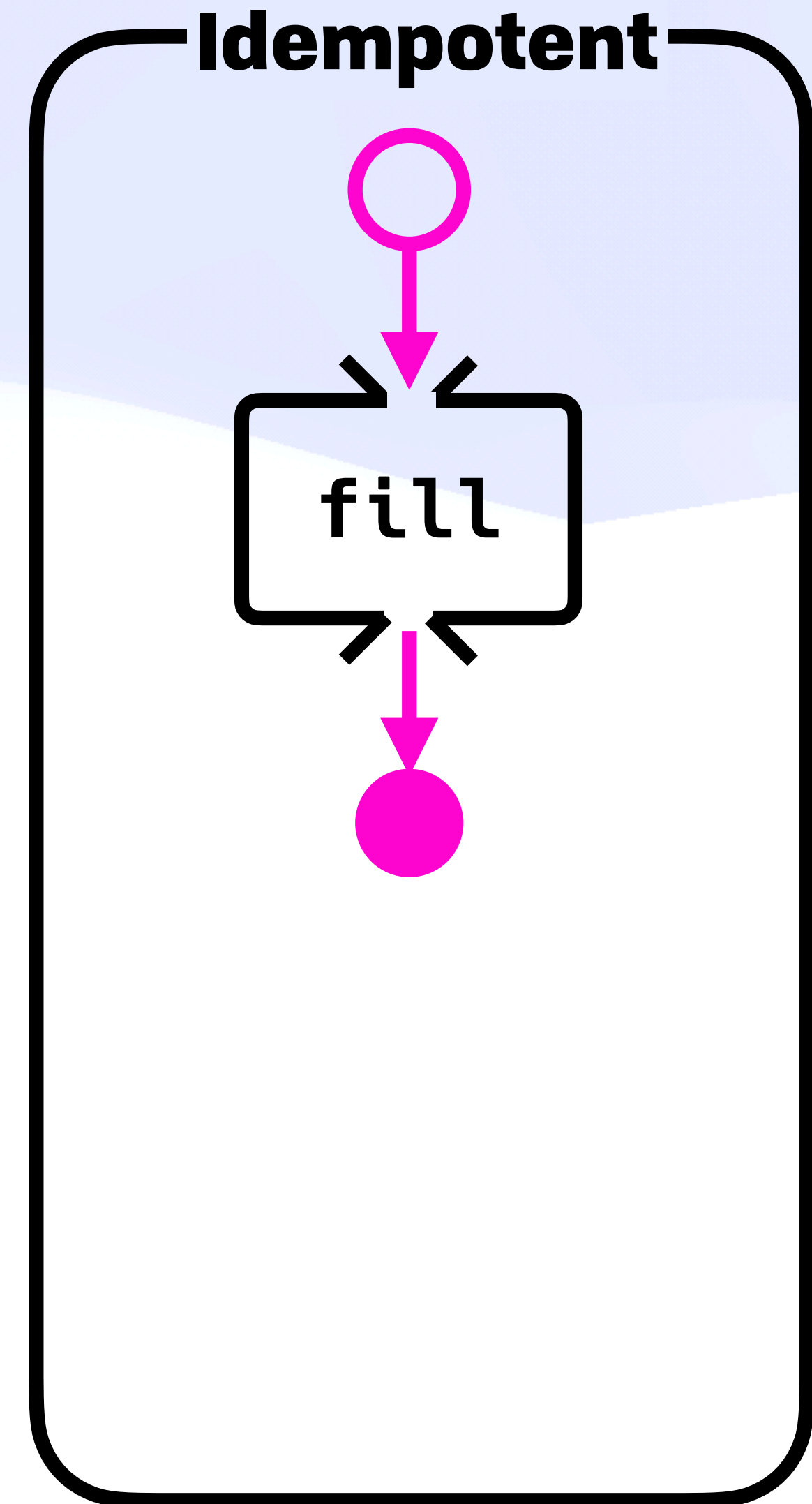
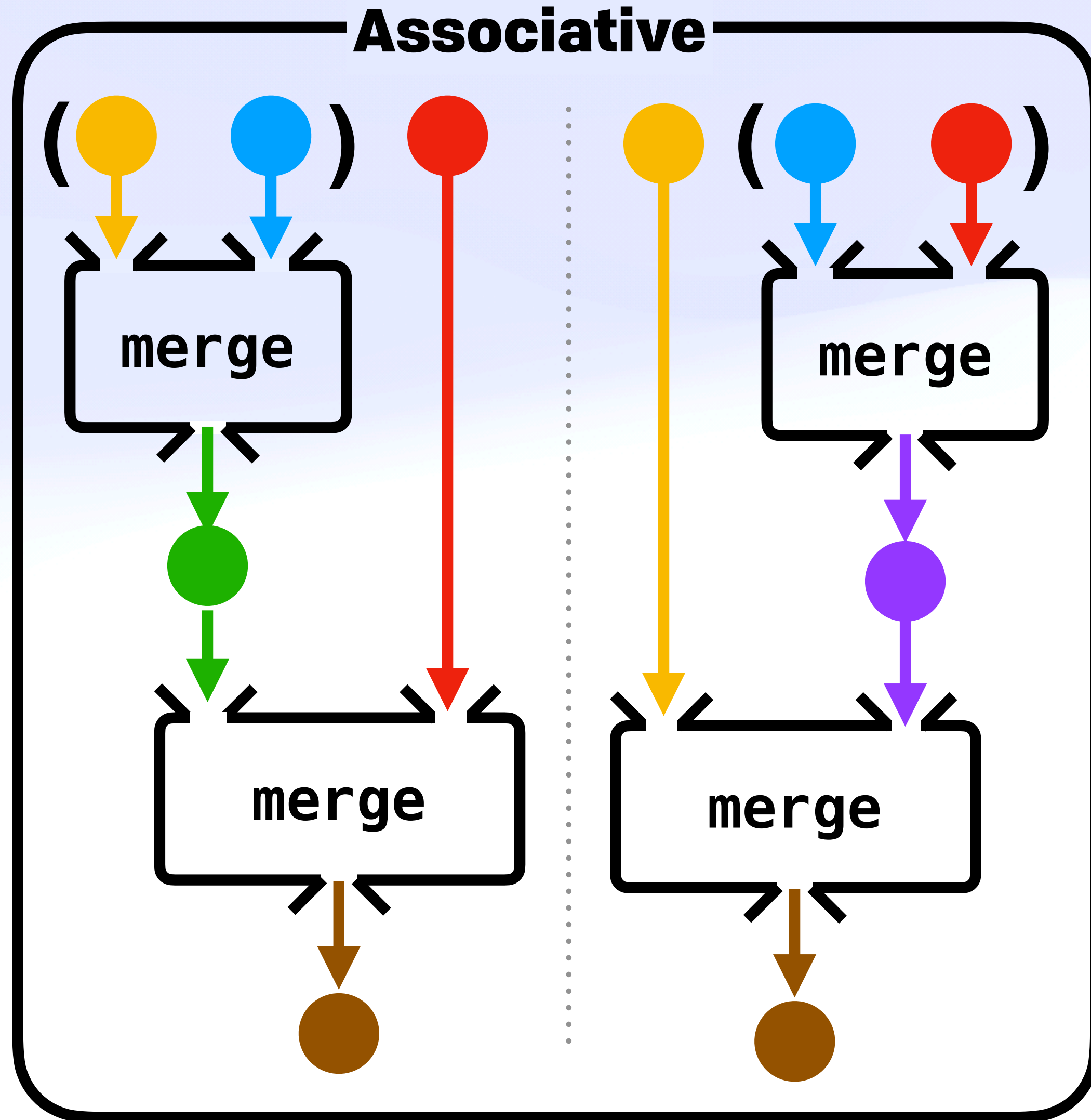
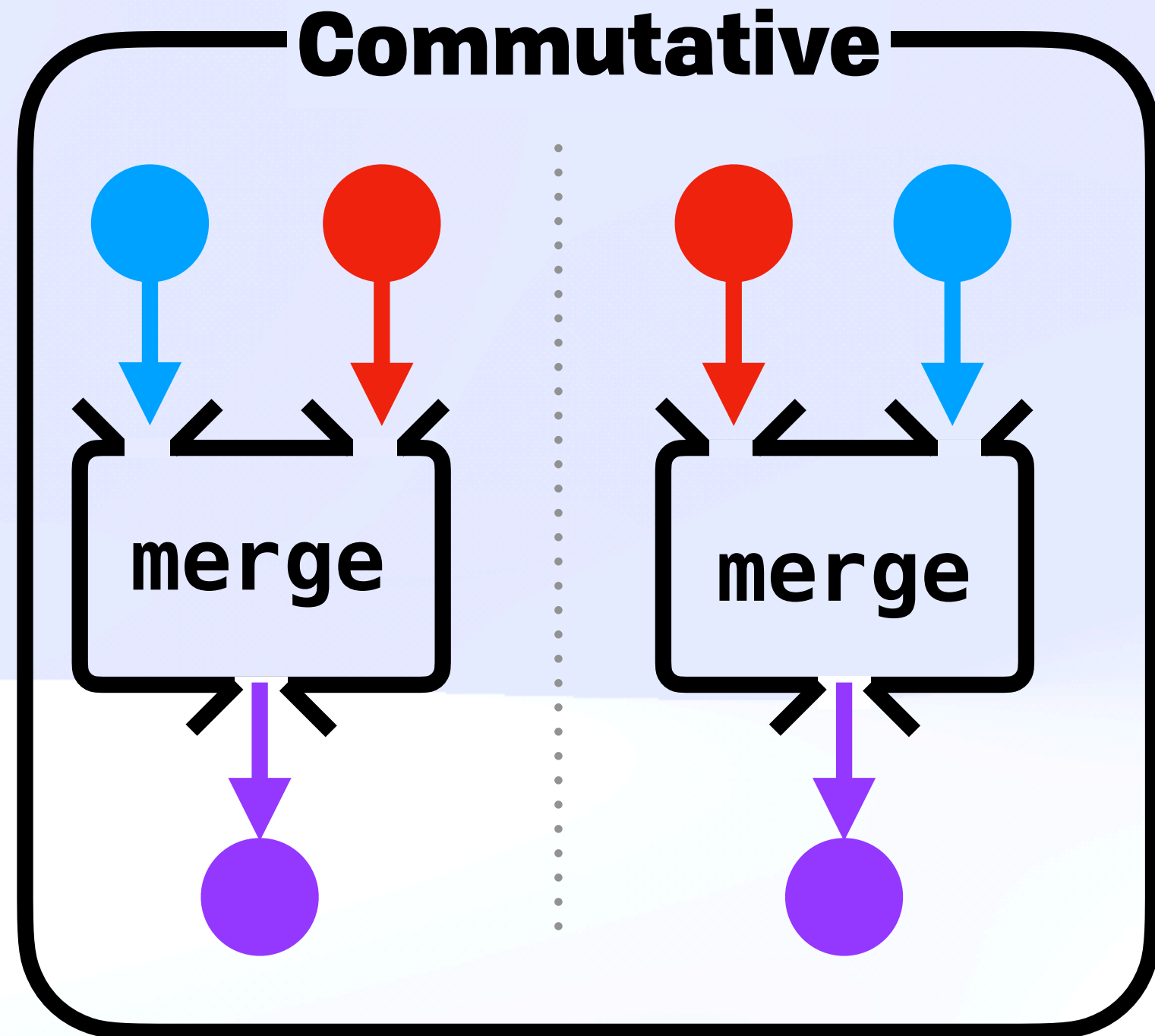
Fault-Safe Concurrency

Properties Save the Day



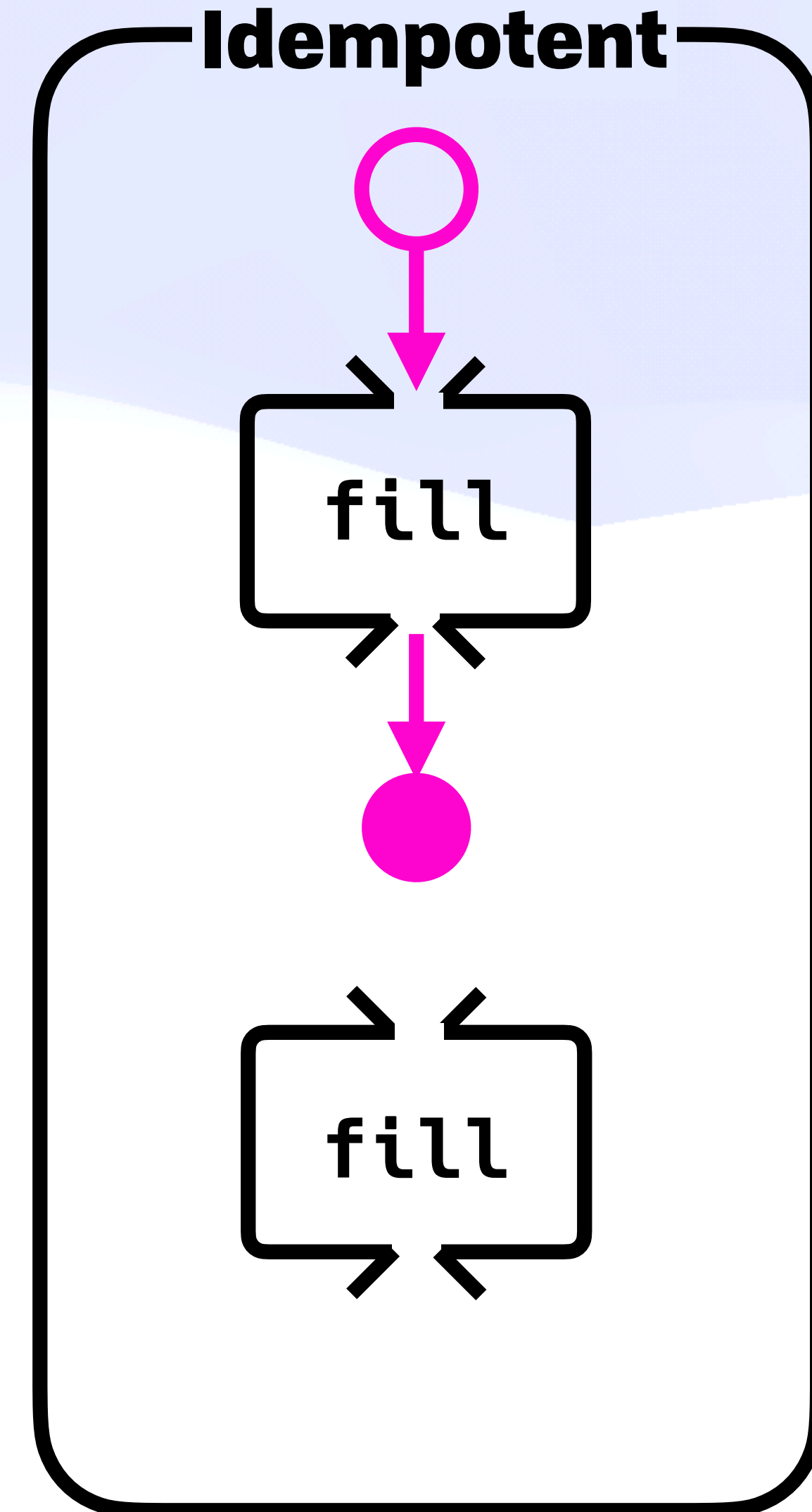
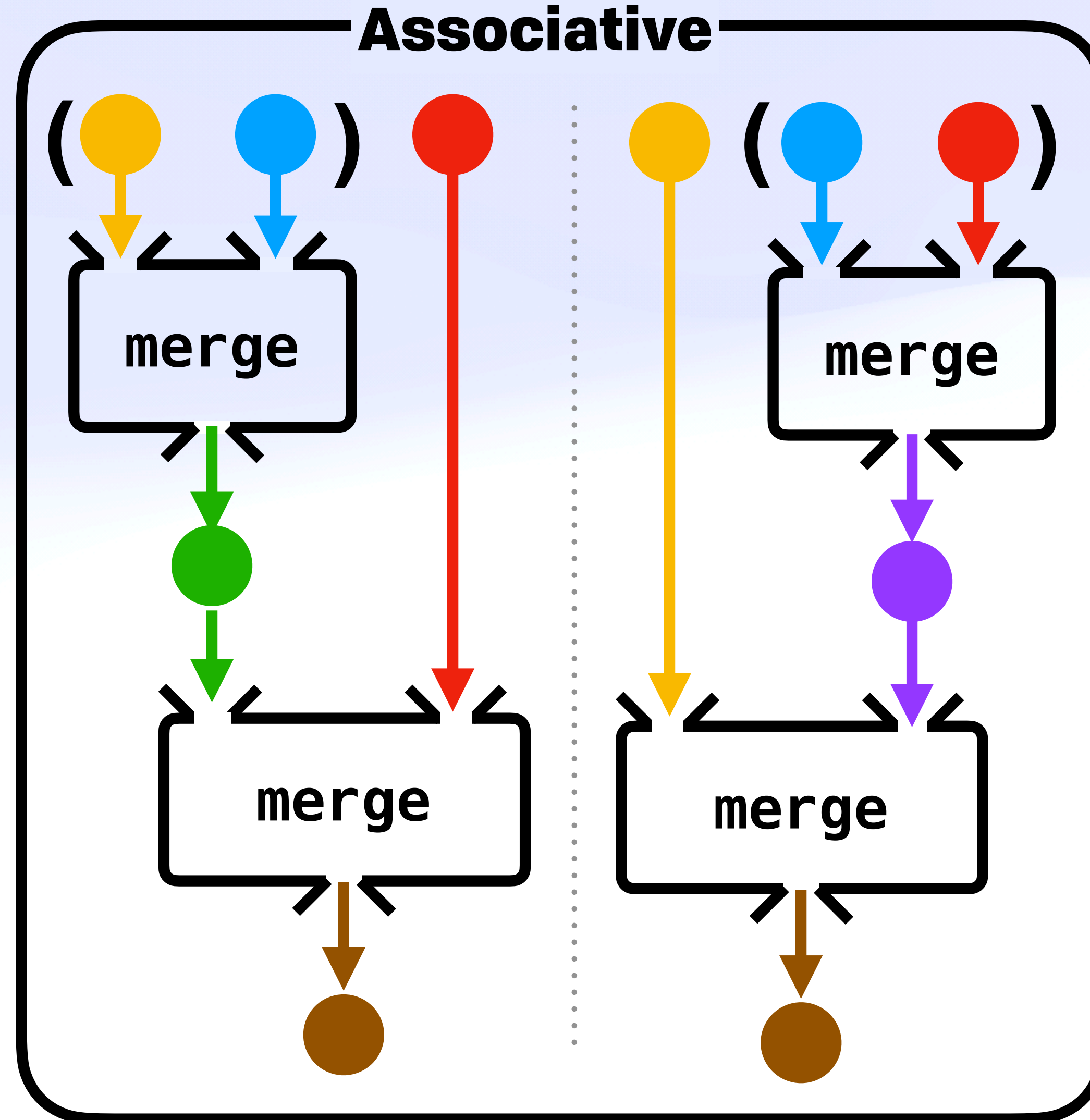
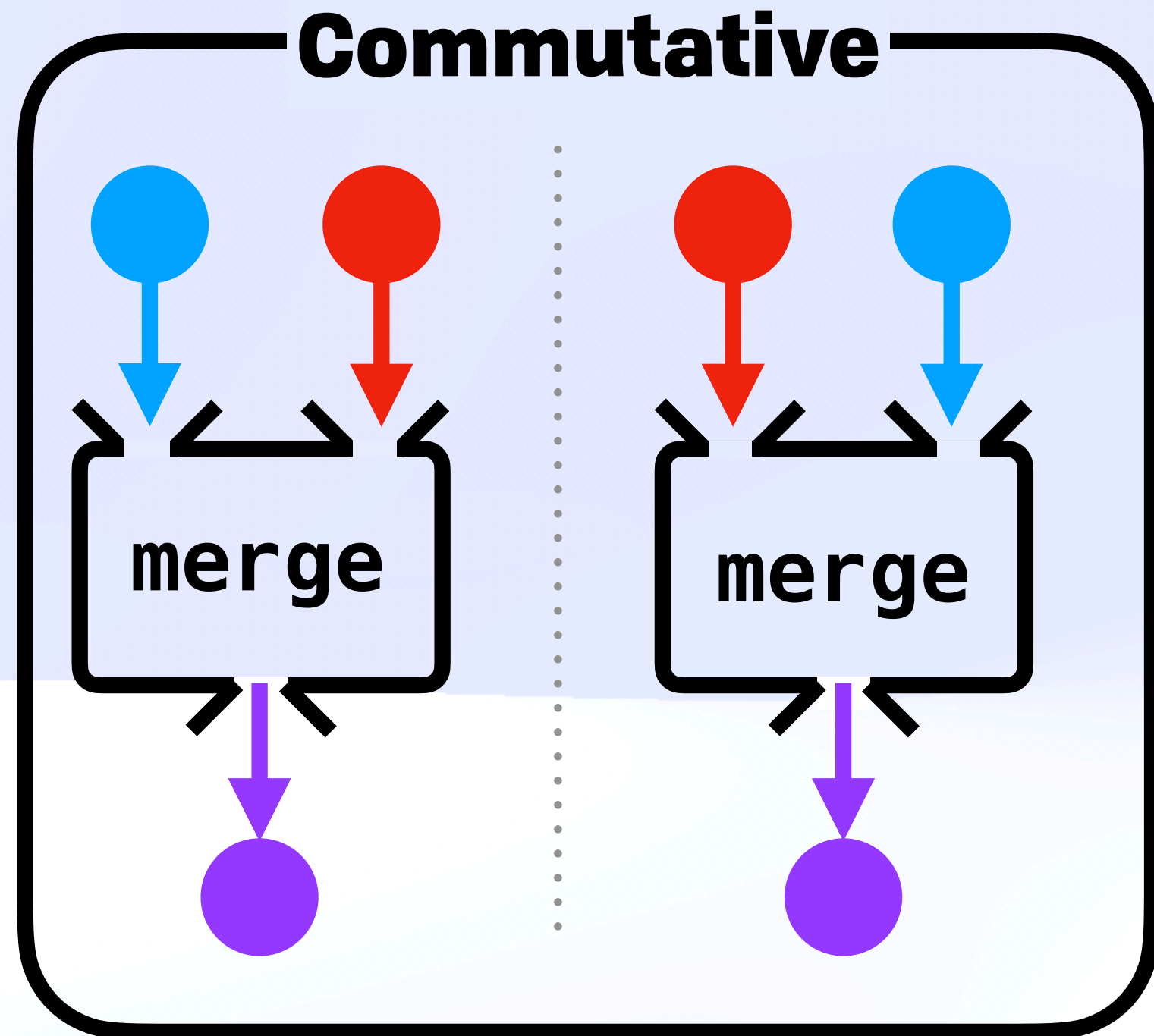
Fault-Safe Concurrency

Properties Save the Day



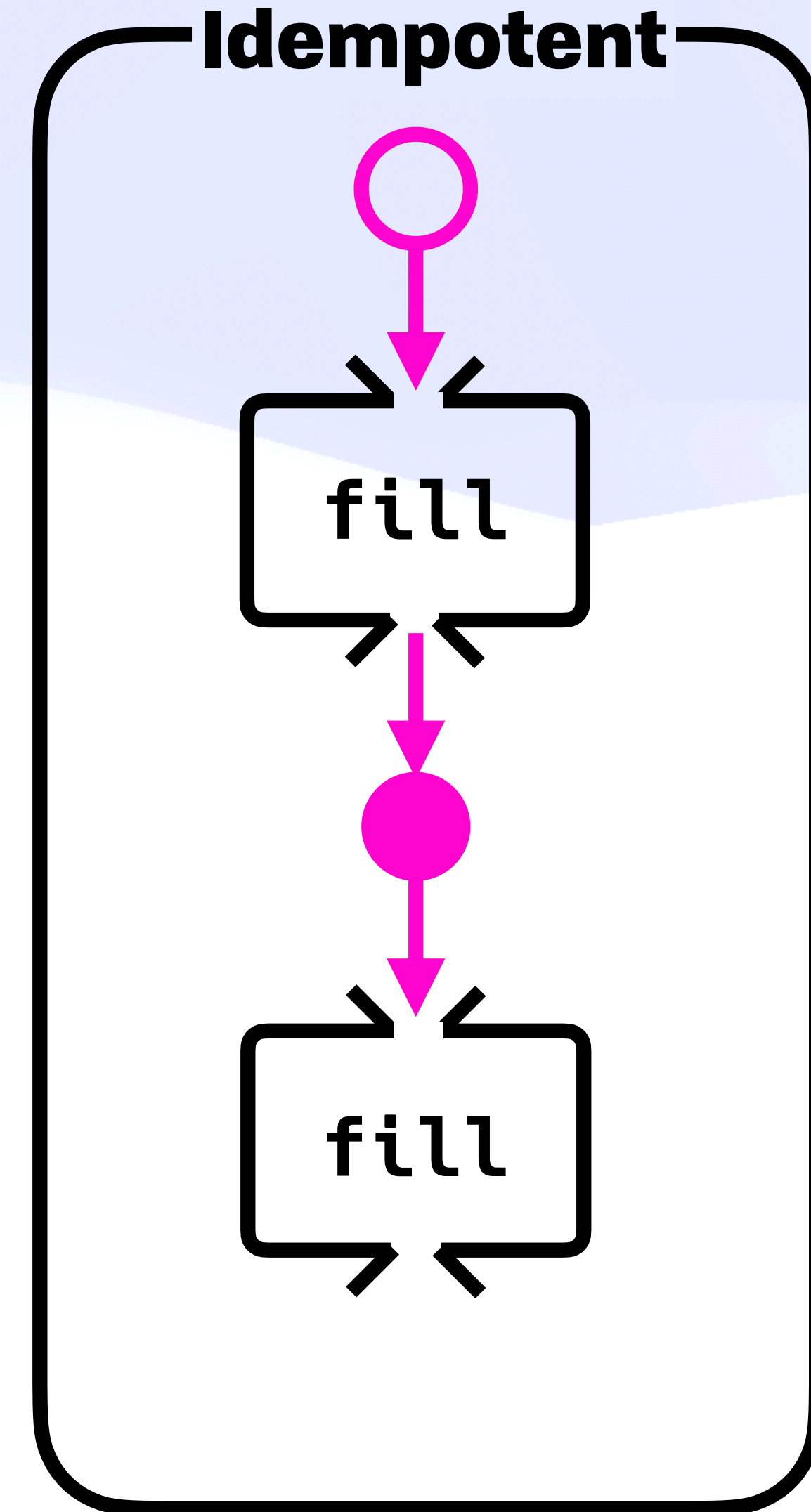
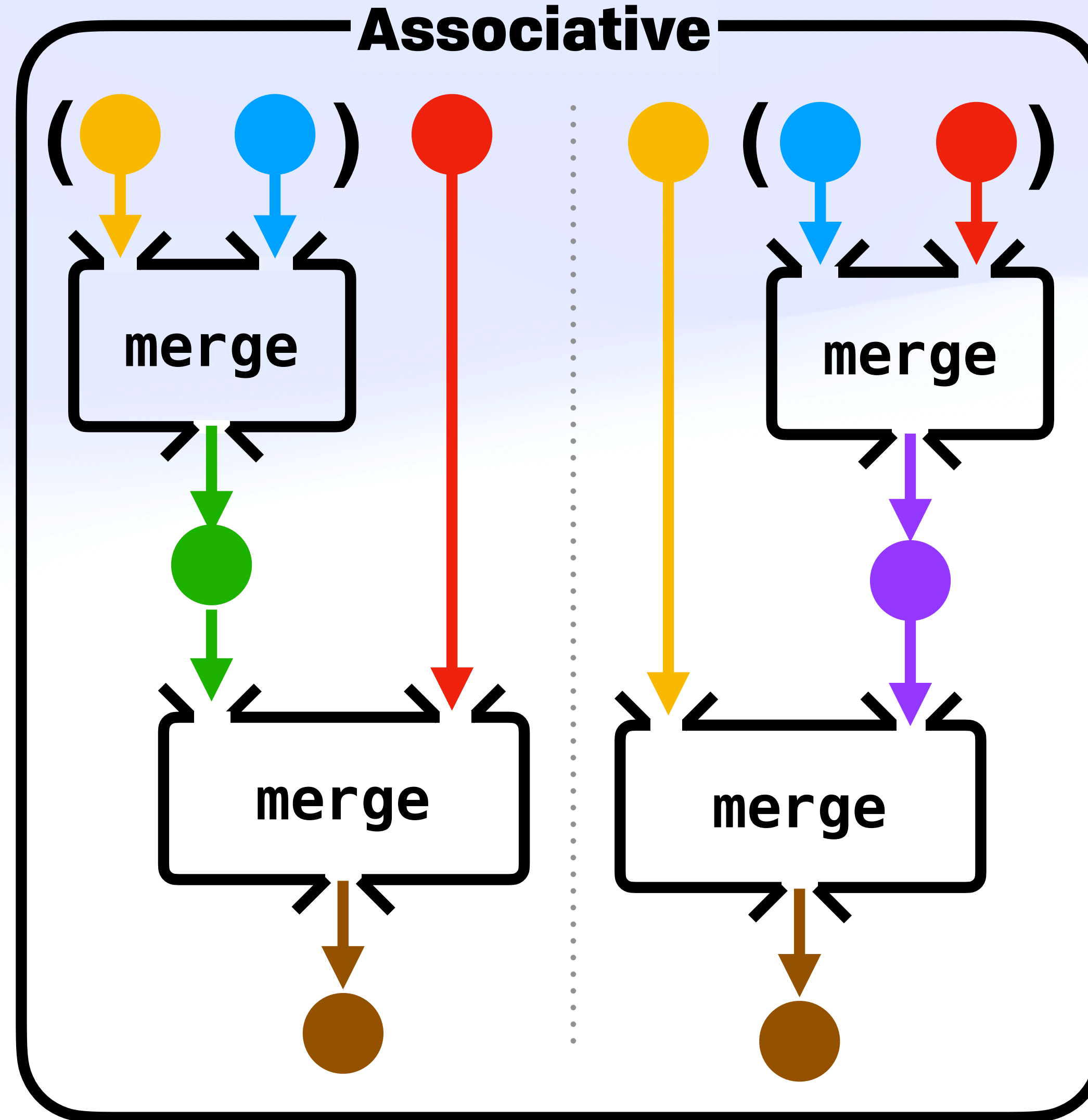
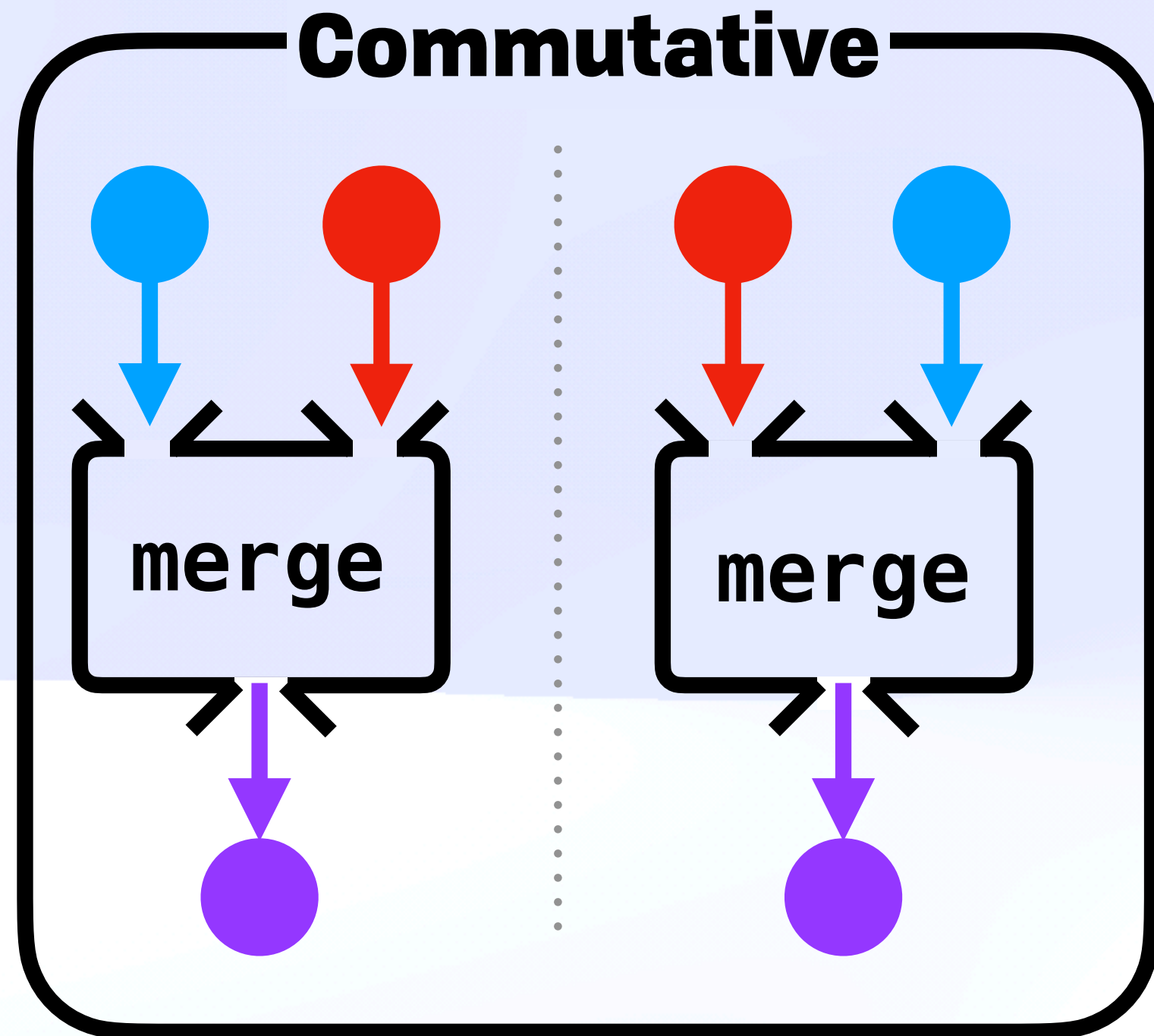
Fault-Safe Concurrency

Properties Save the Day



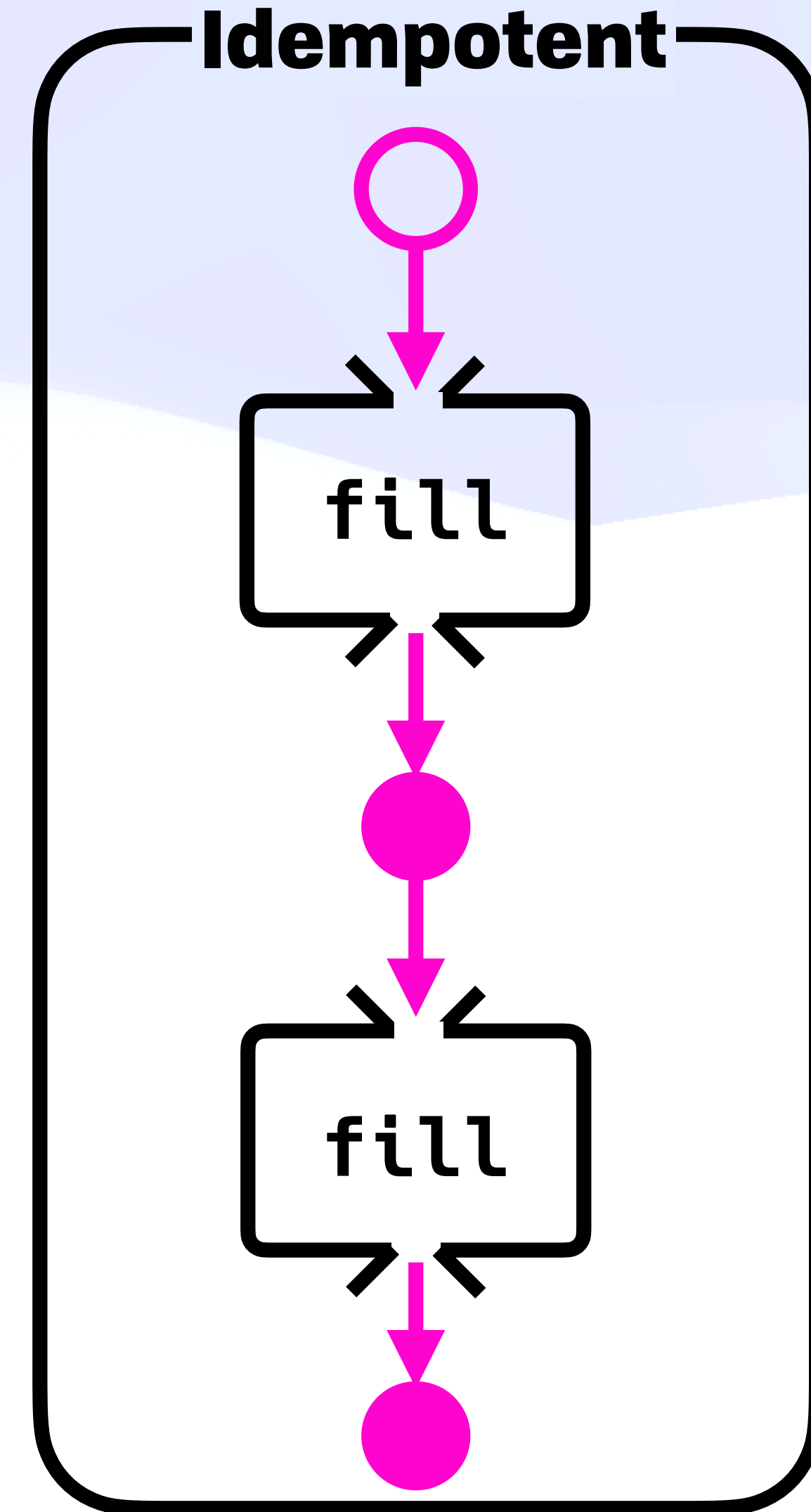
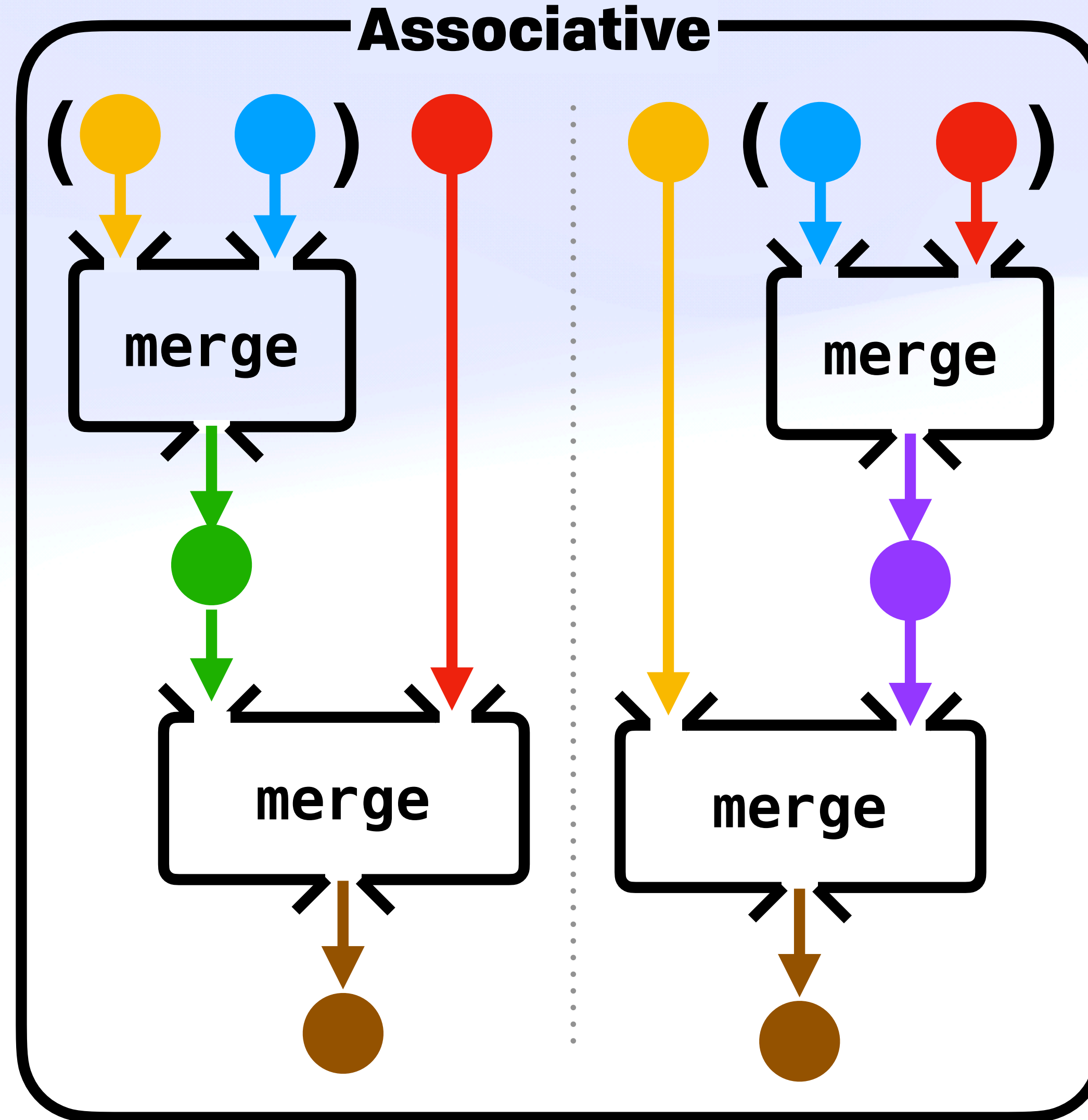
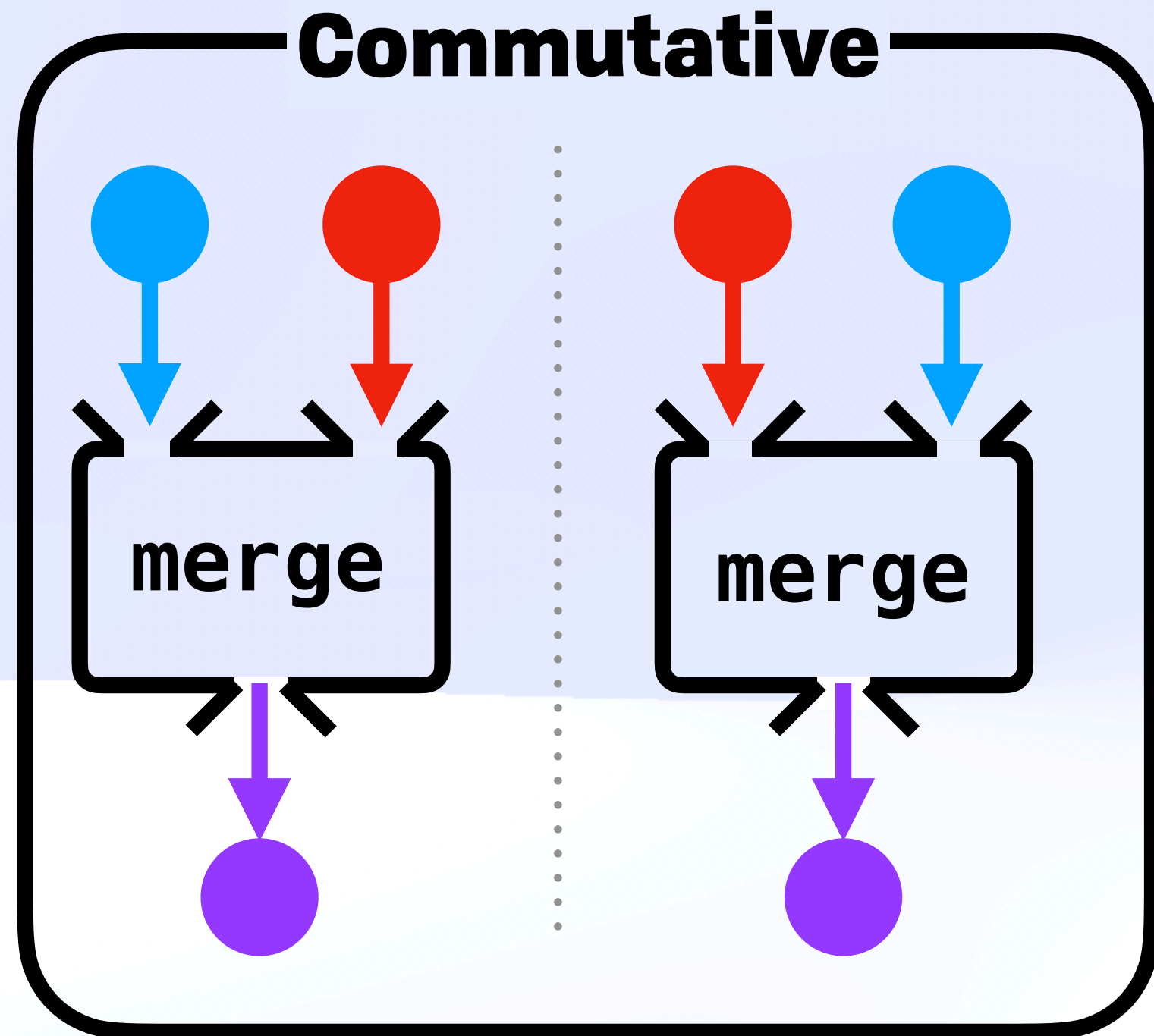
Fault-Safe Concurrency

Properties Save the Day



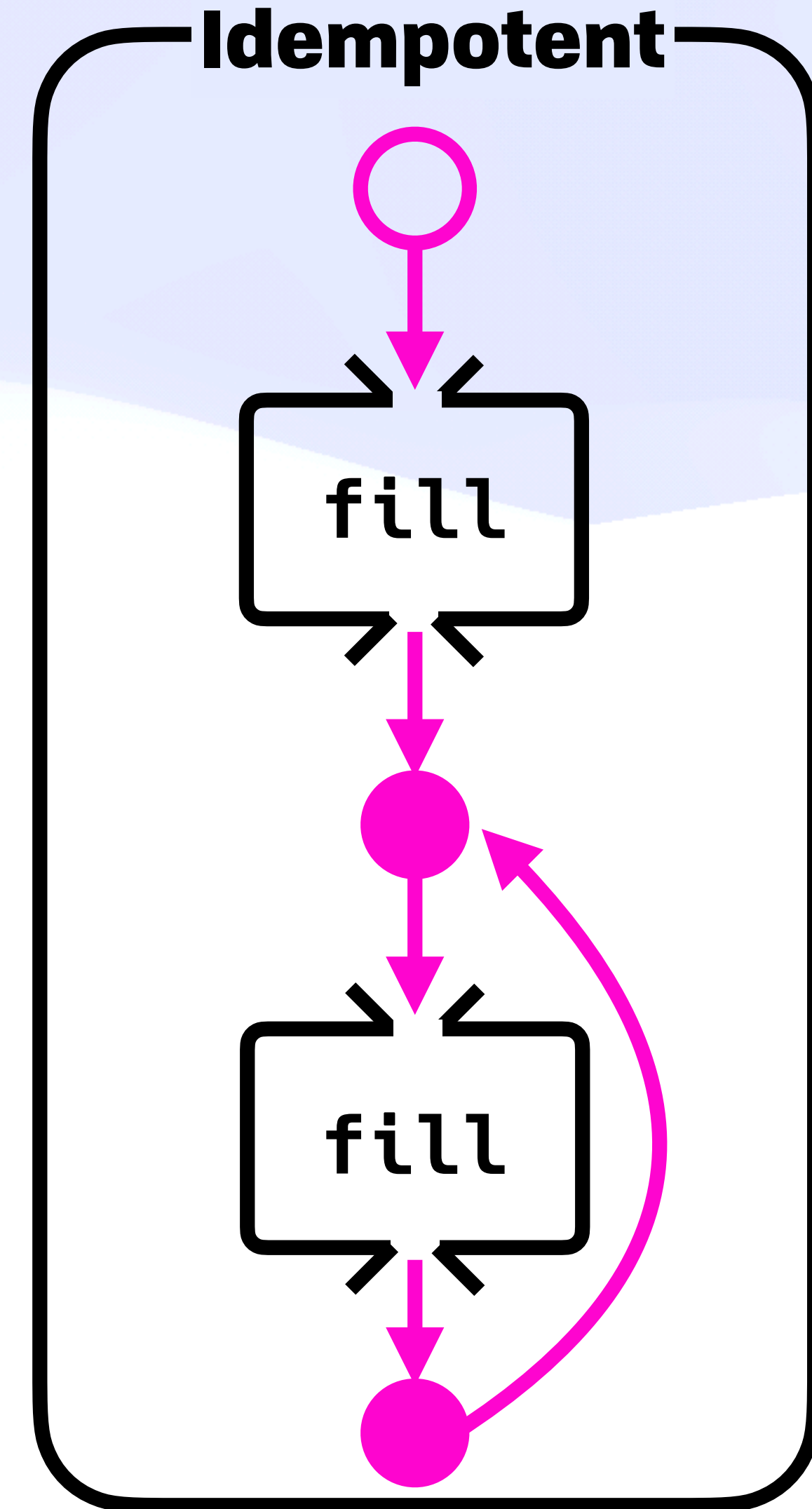
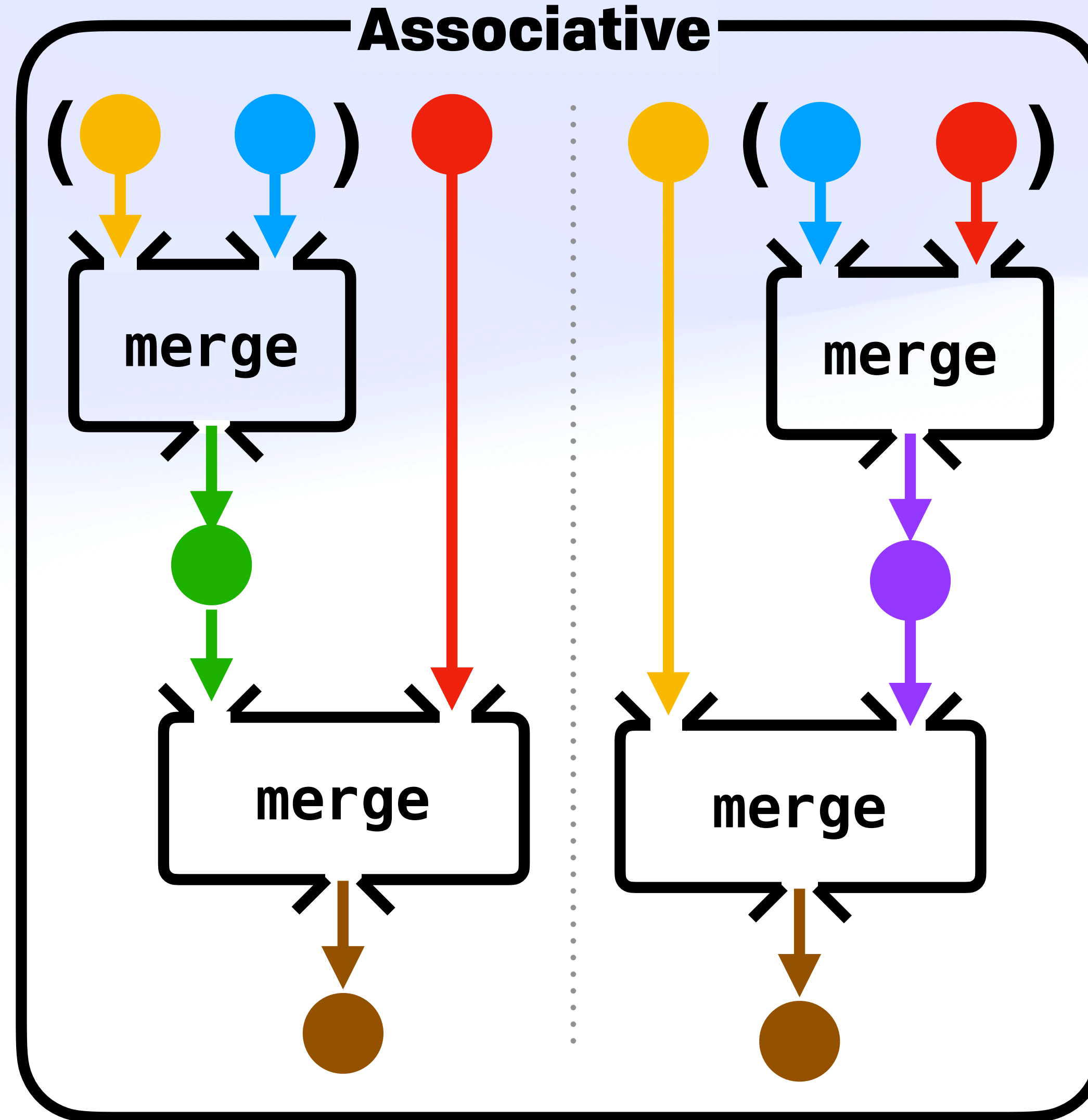
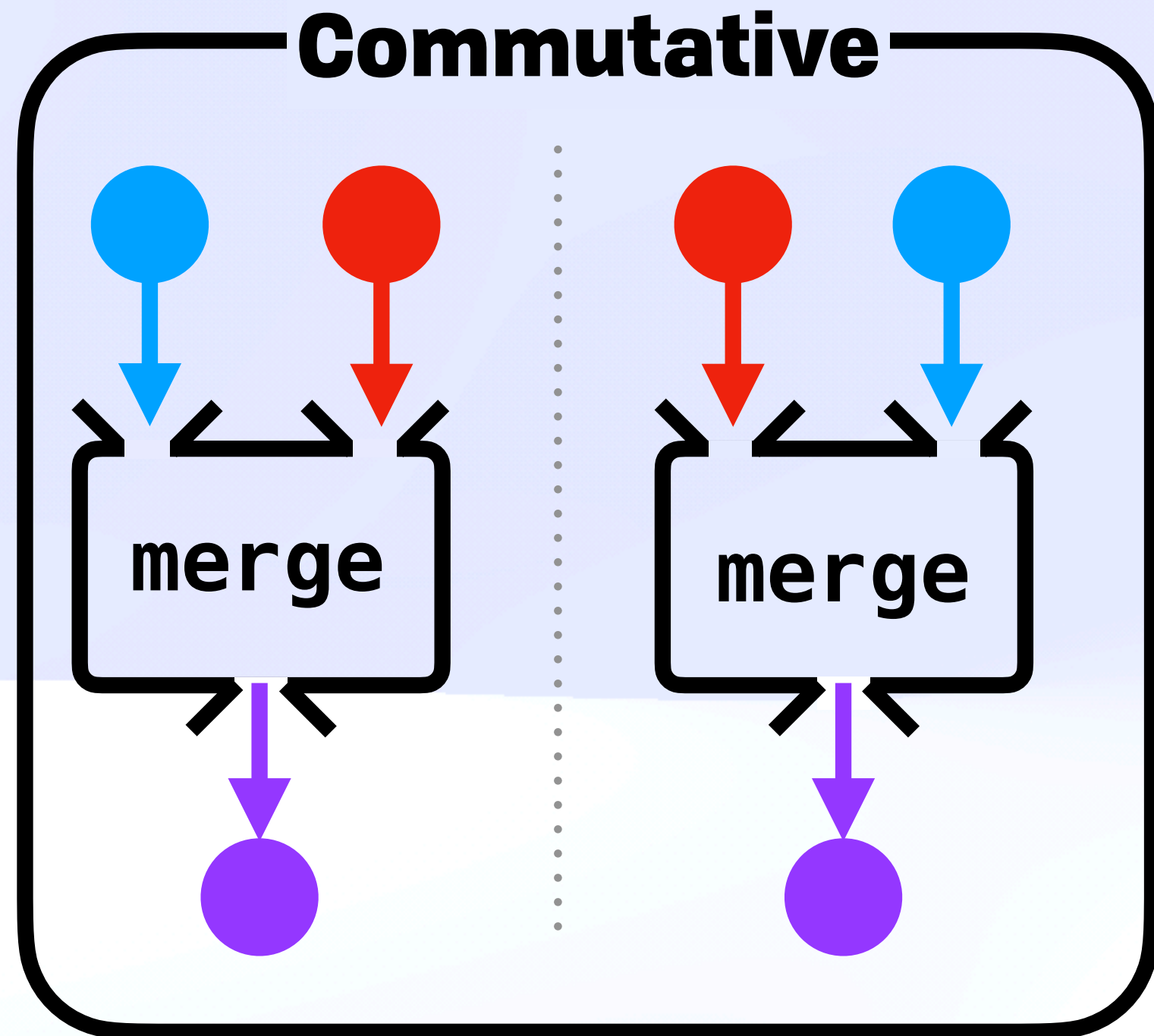
Fault-Safe Concurrency

Properties Save the Day



Fault-Safe Concurrency

Properties Save the Day



Fault-Safe Concurrency 

Temporal Confluence

Fault-Safe Concurrency 

Temporal Confluence

- ◆ BFT-CRDTs
- ◆ Persistent data structure
- ◆ Automatic file-level reconciliation
- ◆ Pluggable sub-file reconciliation (forthcoming)
- ◆ Basis of upcoming BFT Datalog "at scale" work 🤔

Fault-Safe Concurrency 


Temporal Confluence

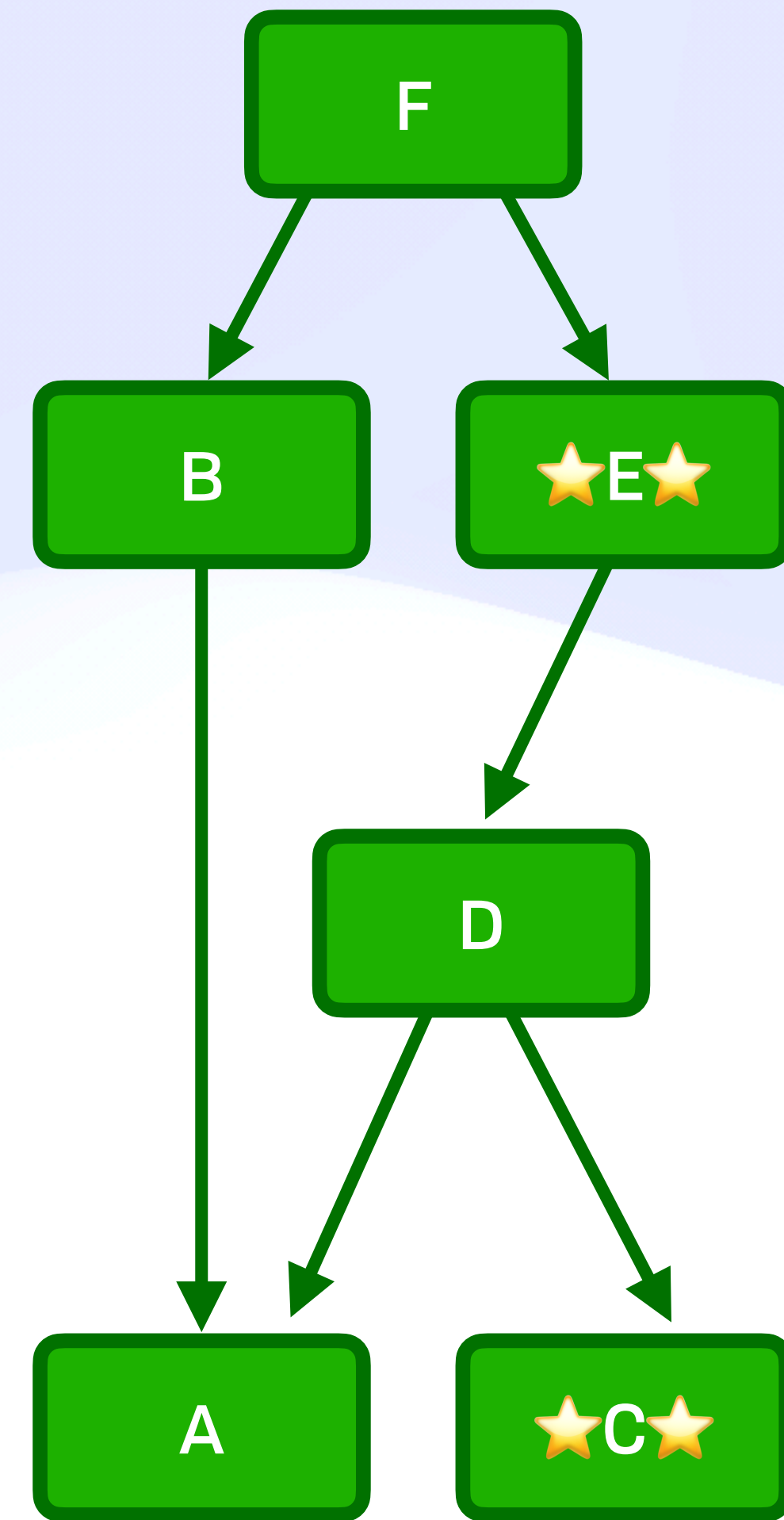
- ◆ BFT-CRDTs
- ◆ Persistent data structure
- ◆ Automatic file-level reconciliation
- ◆ Pluggable sub-file reconciliation (forthcoming)
- ◆ Basis of upcoming BFT Datalog "at scale" work 🤪

Single File History / "Causal Shadow"

Fault-Safe Concurrency

Temporal Confluence

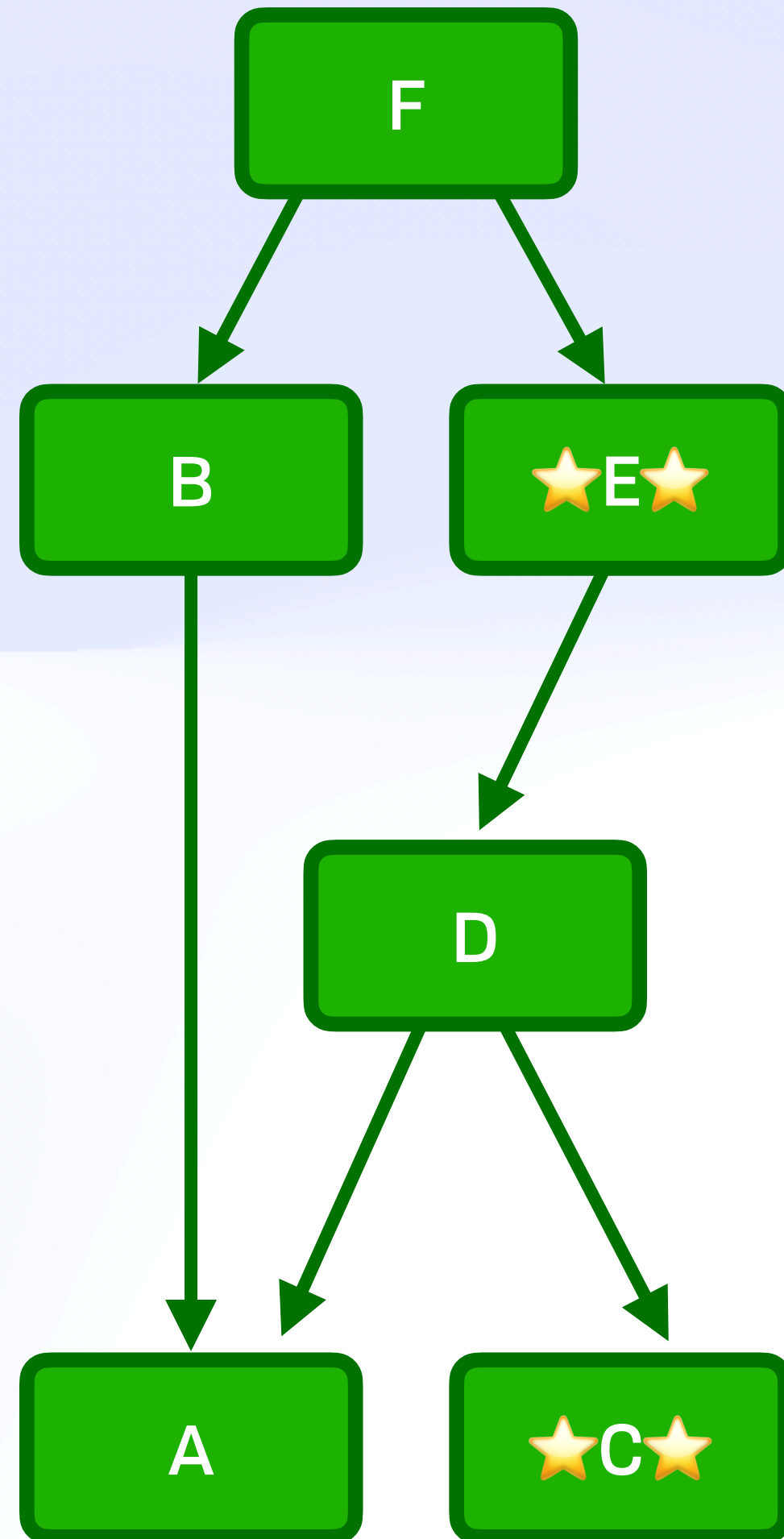
- ◆ BFT-CRDTs
- ◆ Persistent data structure
- ◆ Automatic file-level reconciliation
- ◆ Pluggable sub-file reconciliation (forthcoming)
- ◆ Basis of upcoming BFT Datalog "at scale" work 



Single File History / "Causal Shadow"

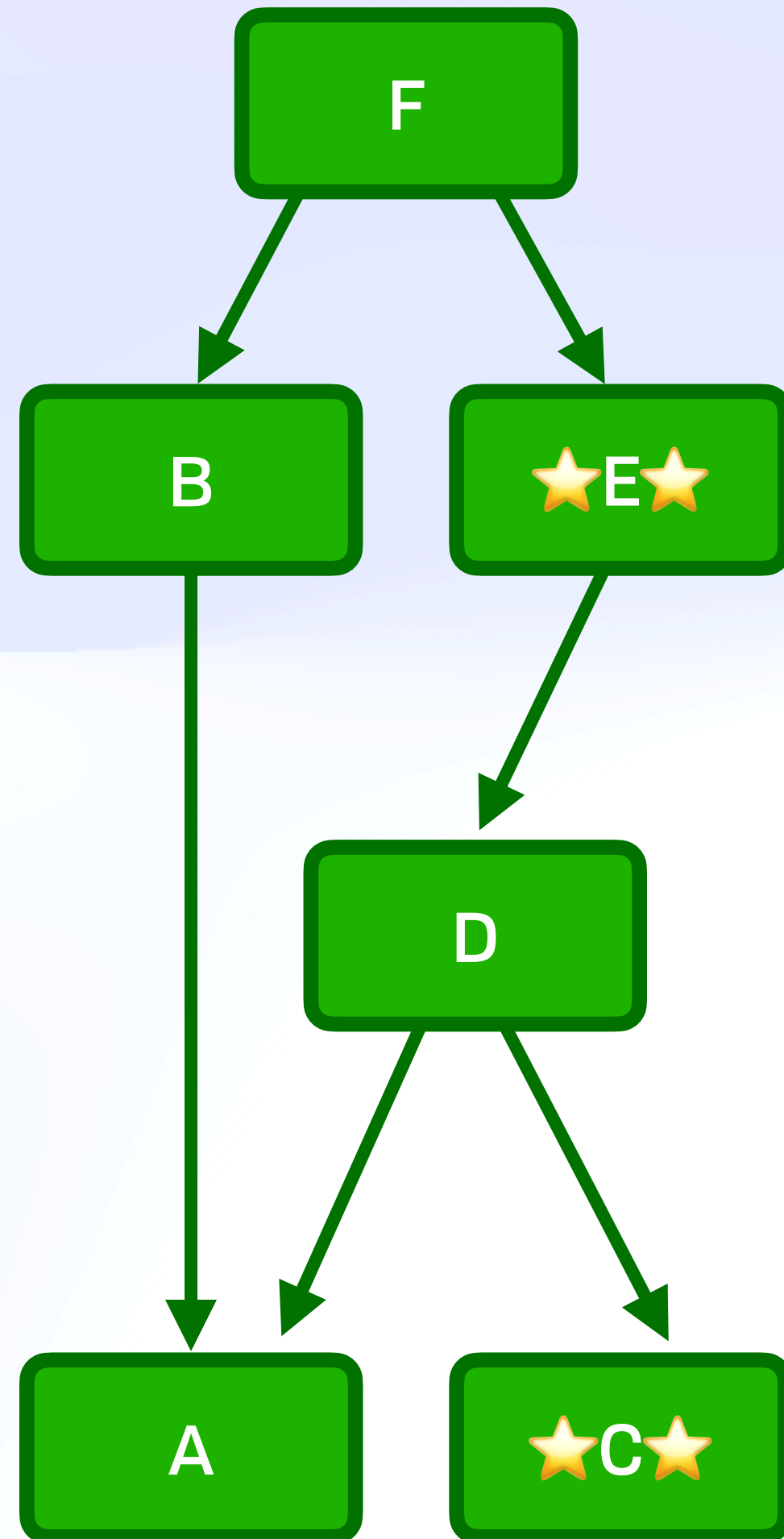
Fault-Safe Concurrency 

Multiplayer Docs (Incl. Encryption)



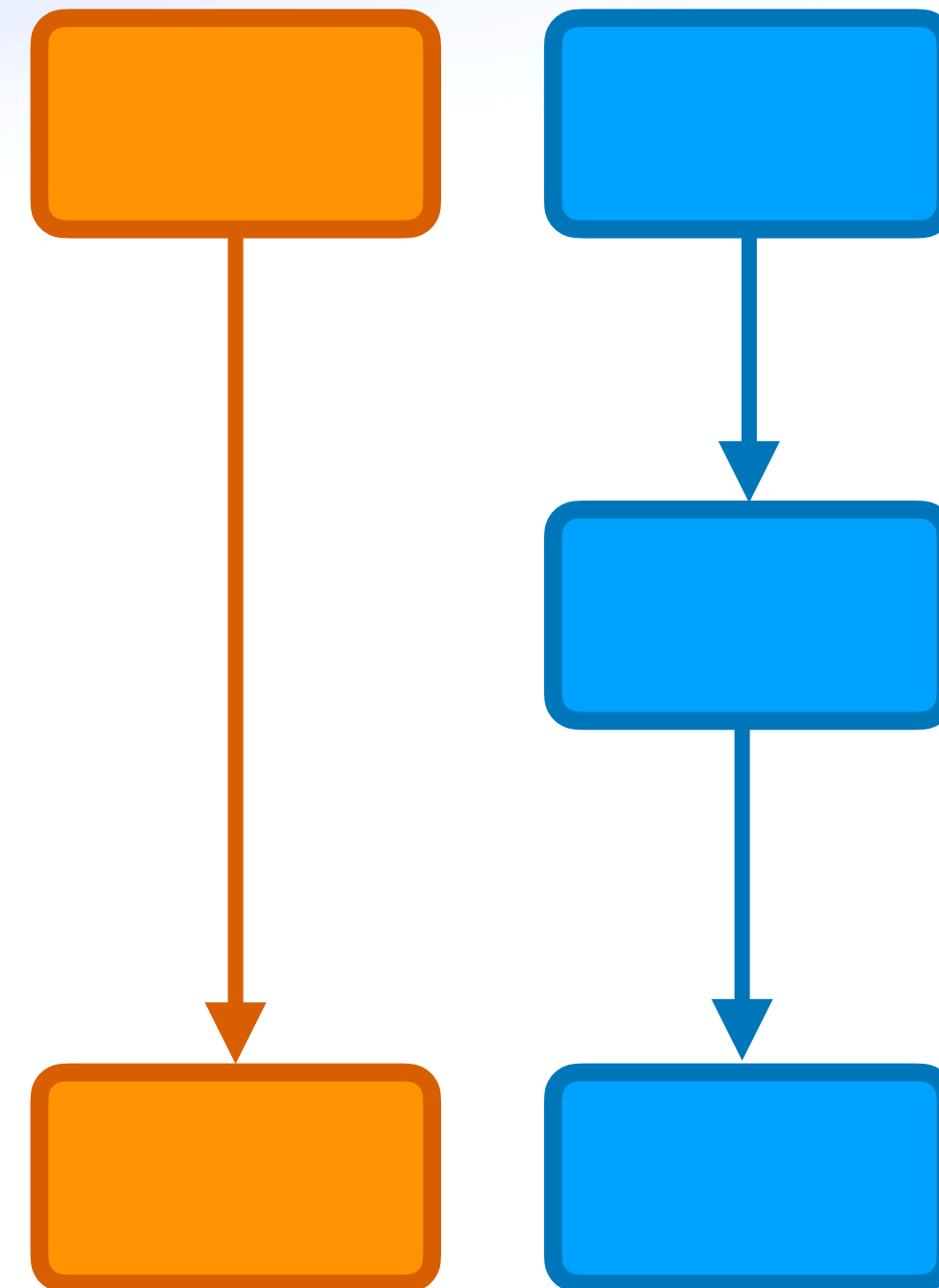
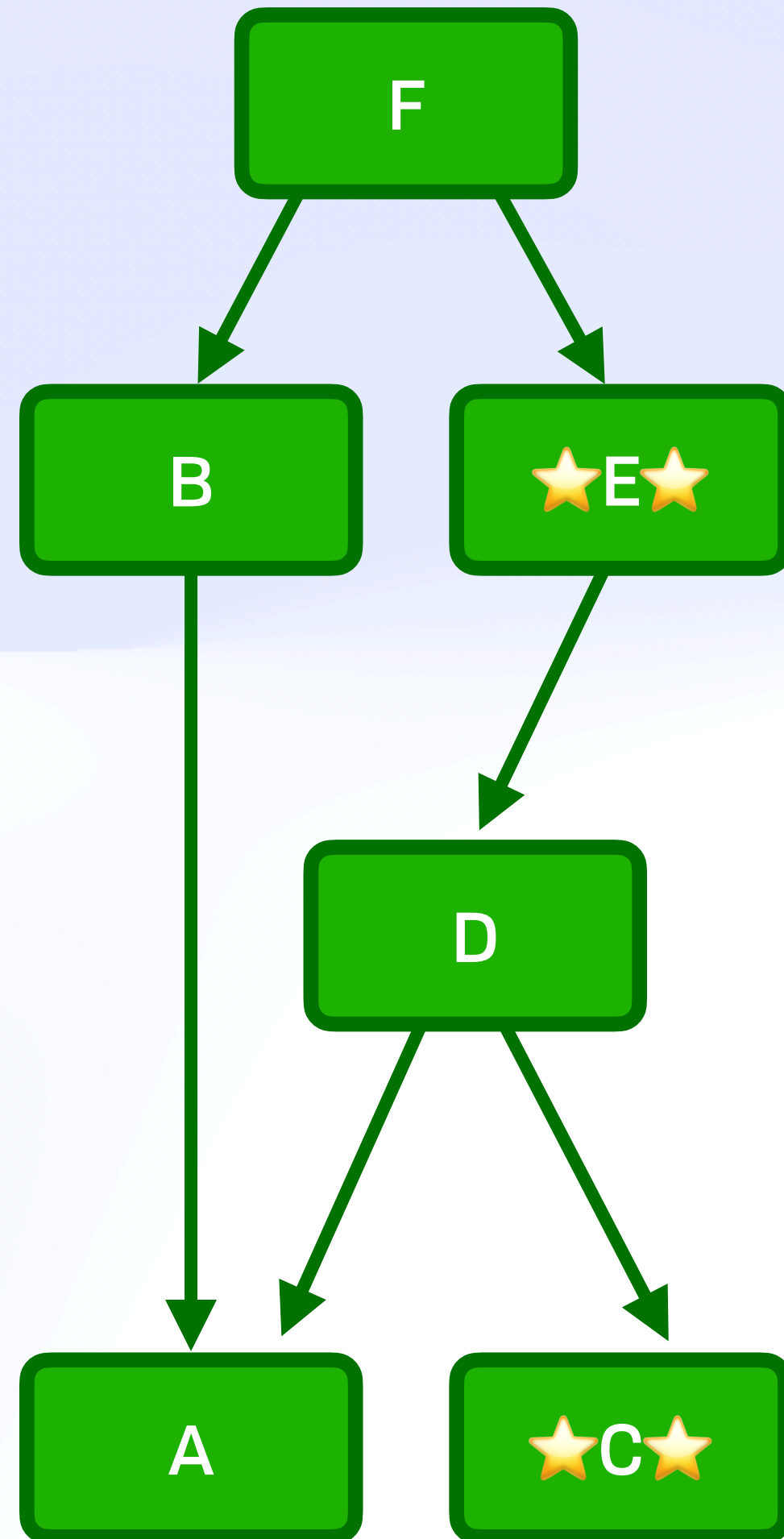
Fault-Safe Concurrency 

Multiplayer Docs (Incl. Encryption)



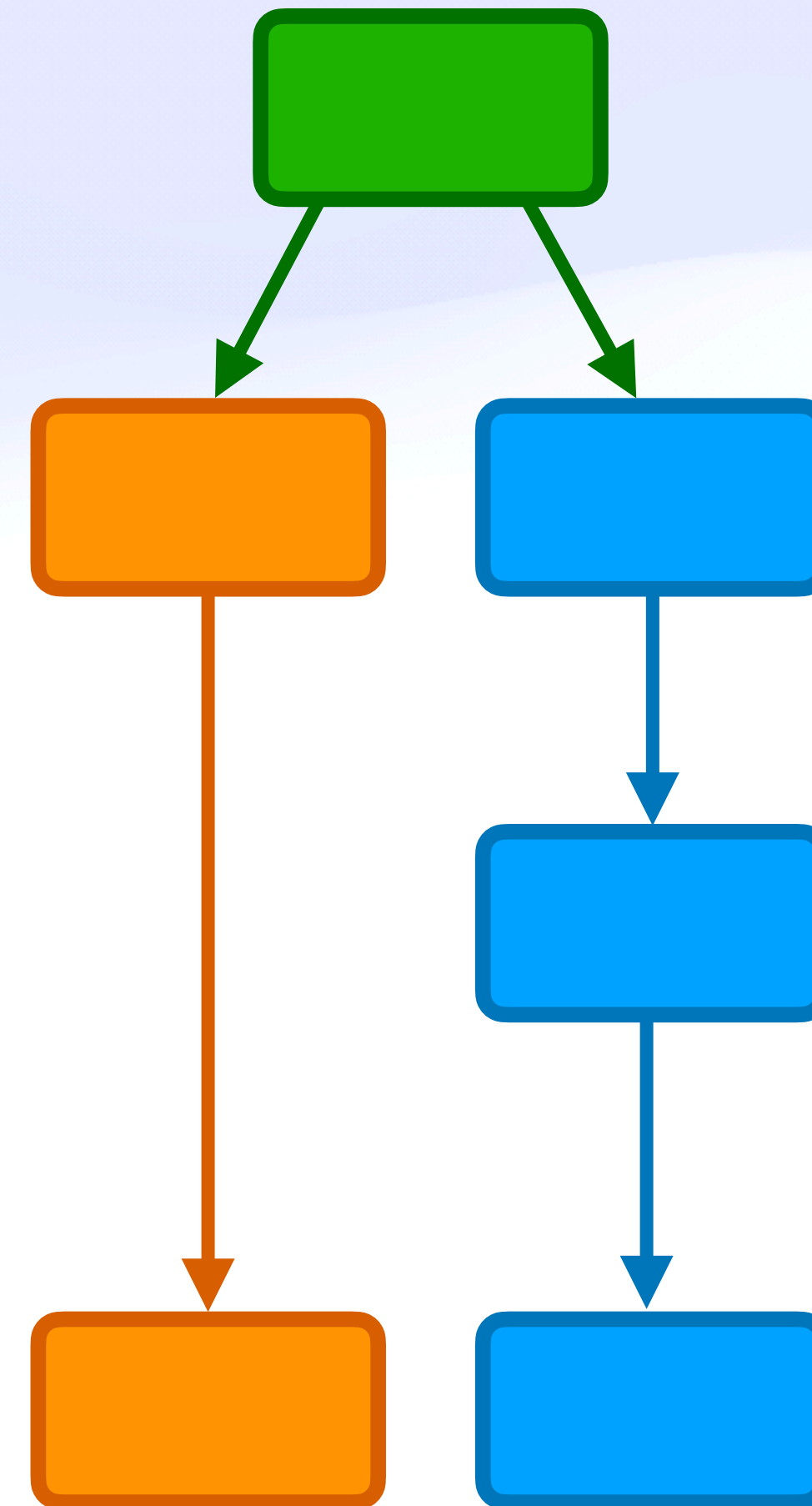
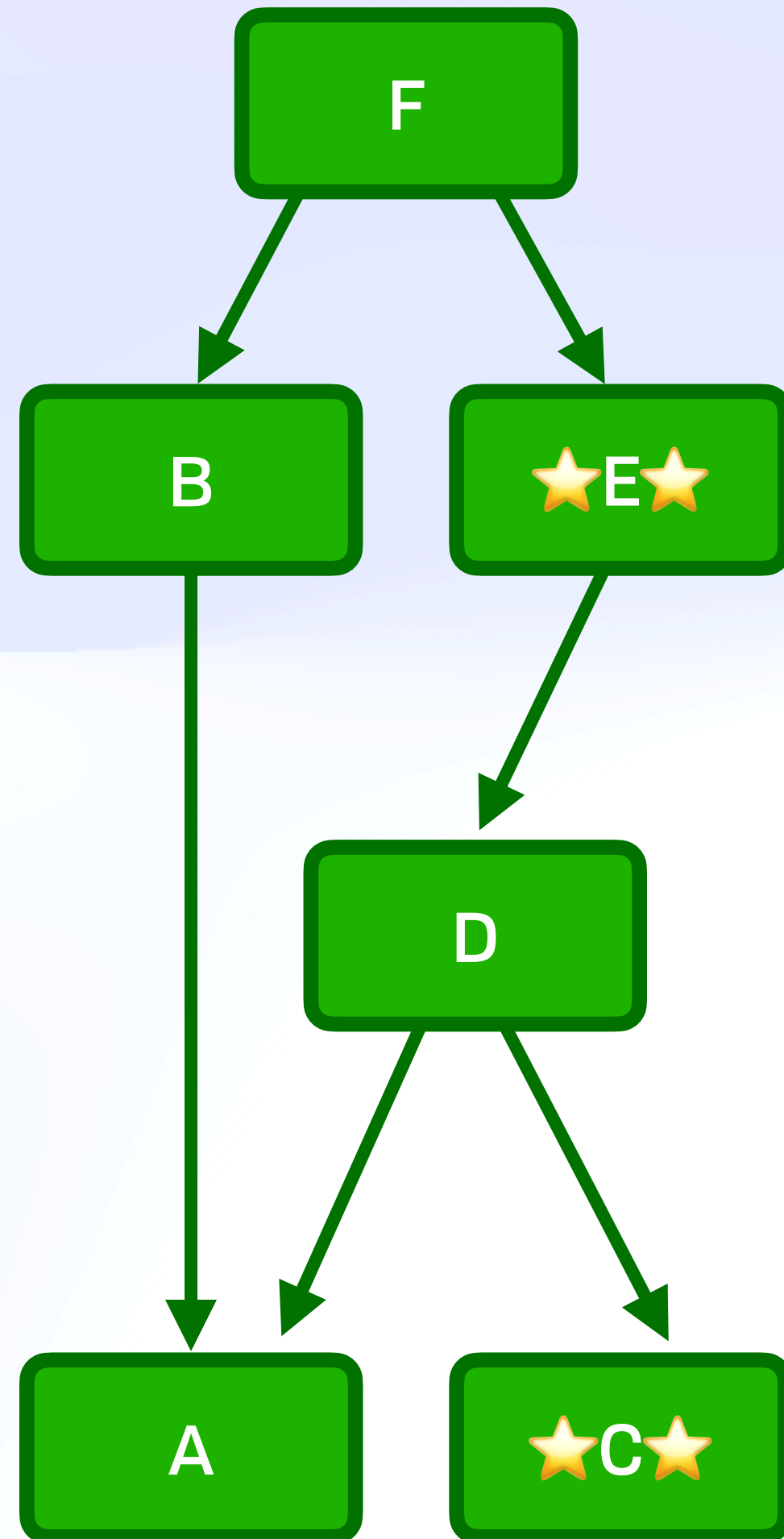
Fault-Safe Concurrency 

Multiplayer Docs (Incl. Encryption)



Fault-Safe Concurrency 

Multiplayer Docs (Incl. Encryption)



Fault-Safe Concurrency 

Alternate History, Same State

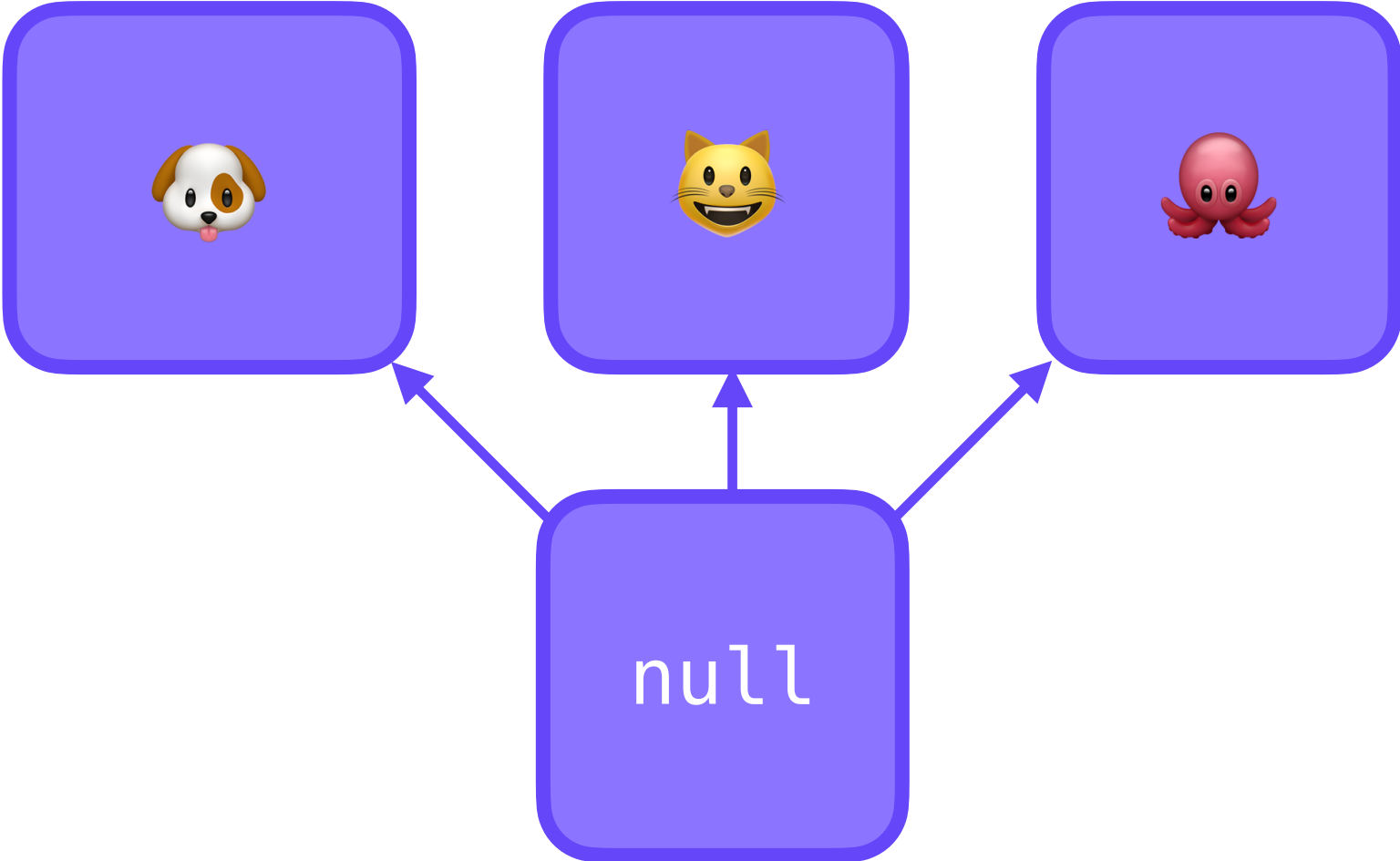
Fault-Safe Concurrency 

Alternate History, Same State

null

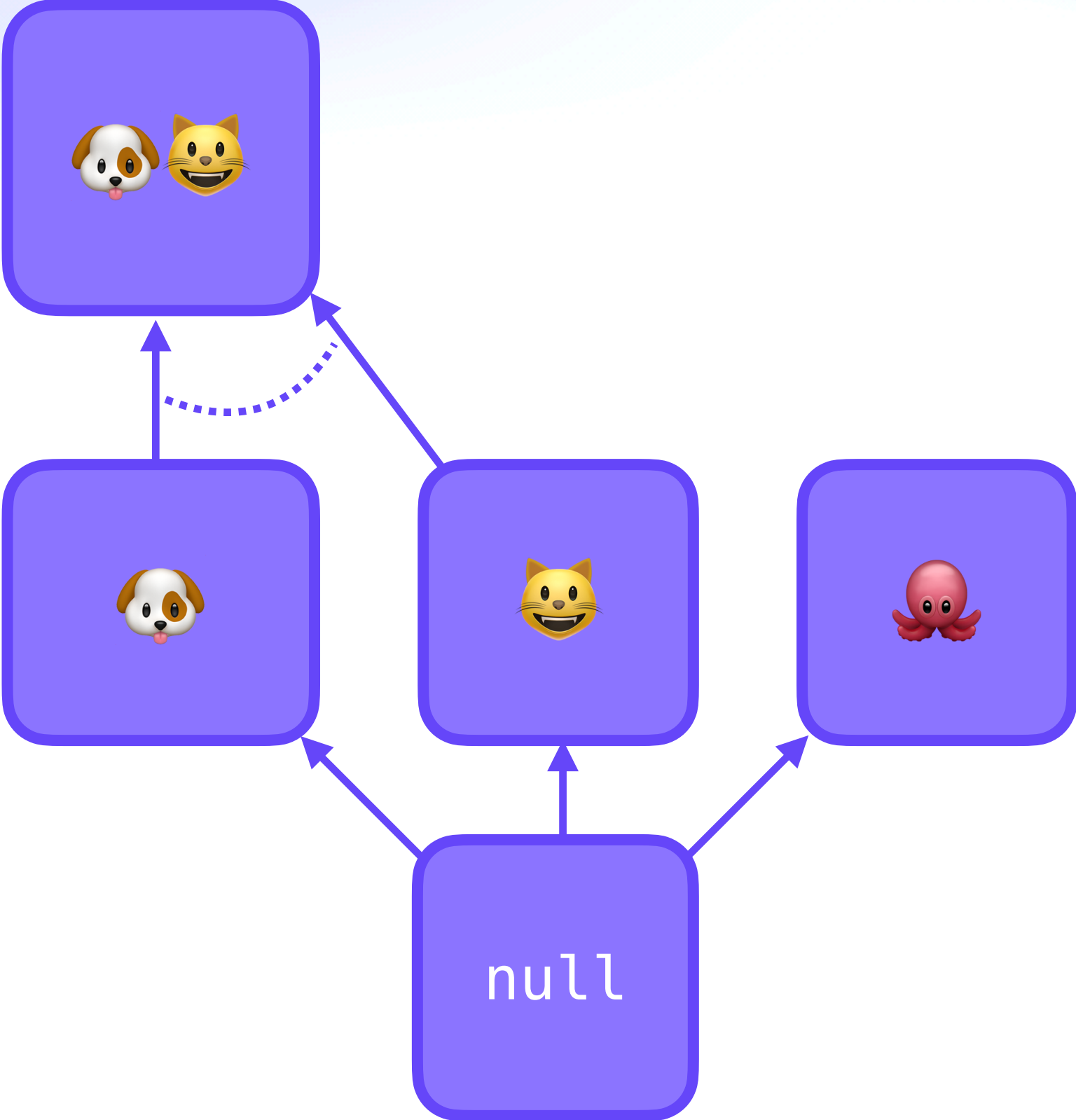
Fault-Safe Concurrency 

Alternate History, Same State



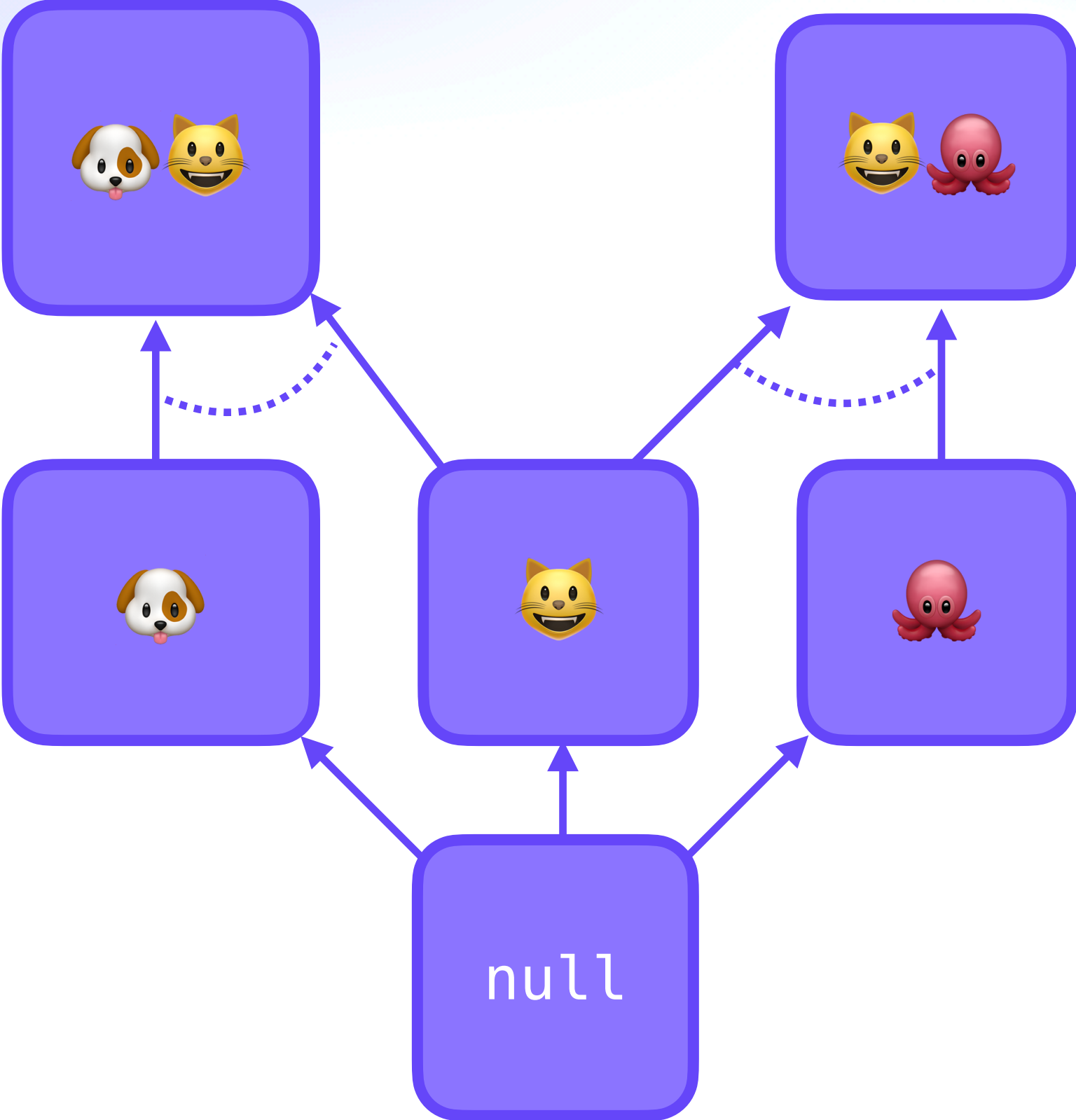
Fault-Safe Concurrency 

Alternate History, Same State



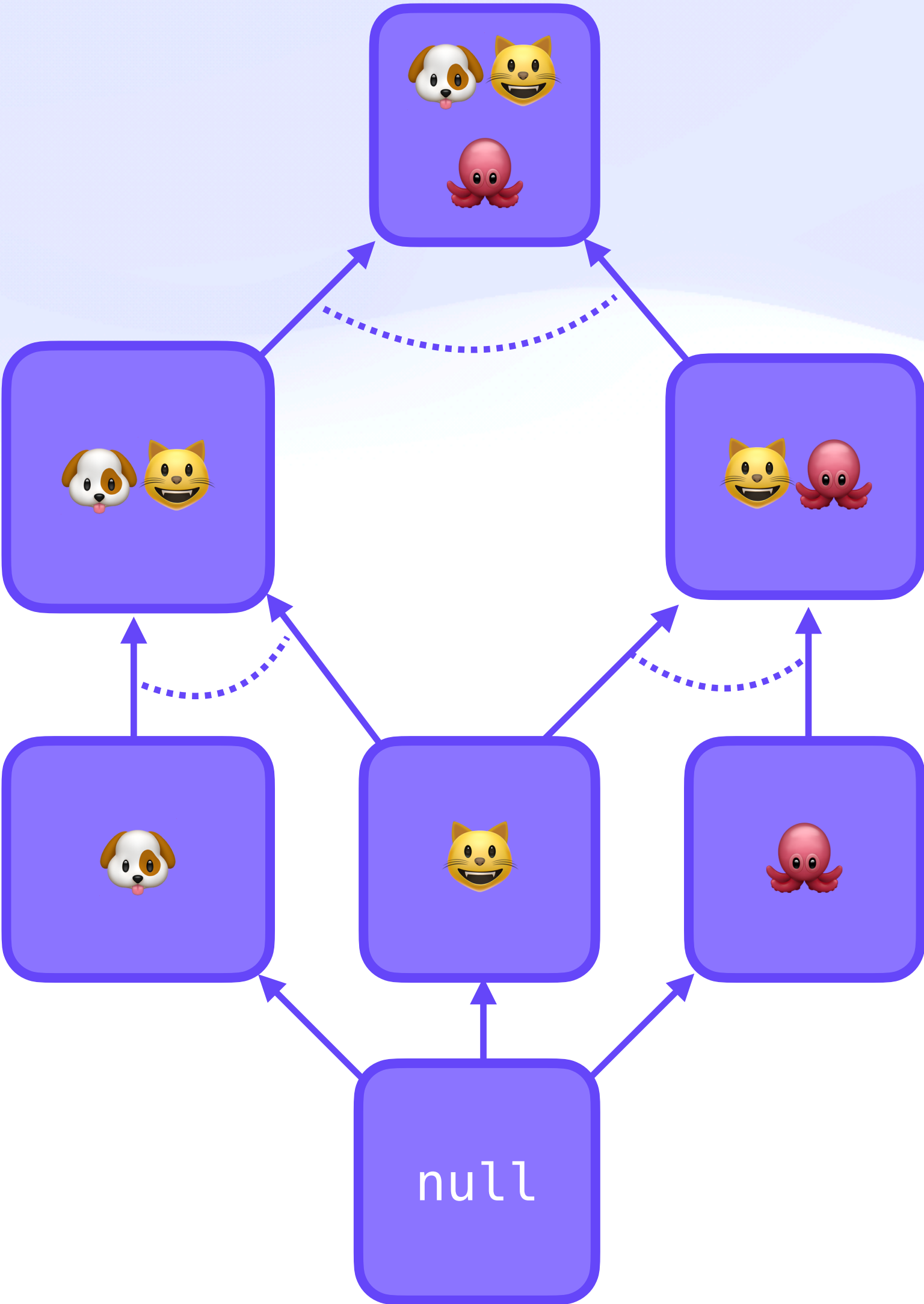
Fault-Safe Concurrency 

Alternate History, Same State



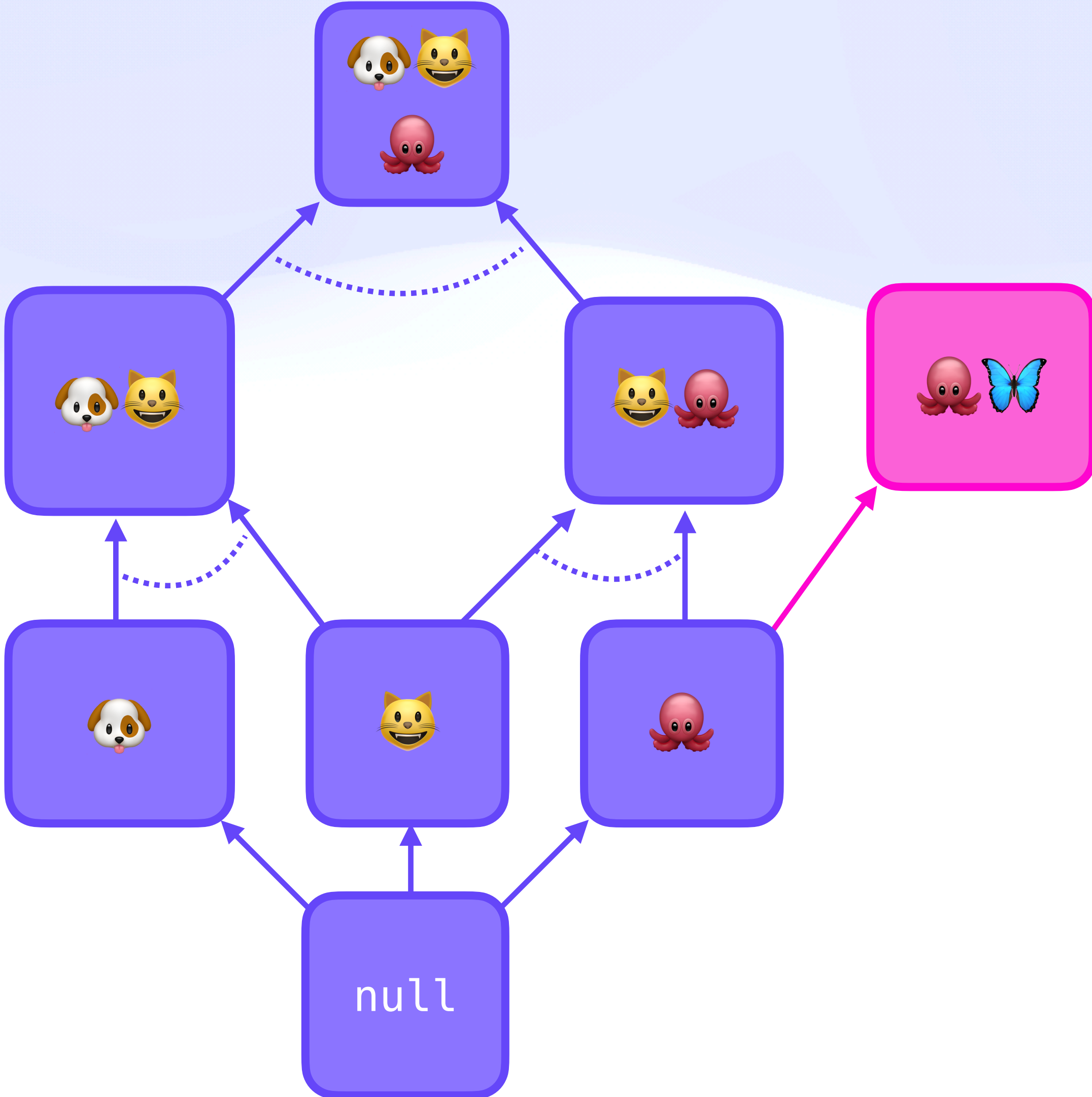
Fault-Safe Concurrency

Alternate History, Same State



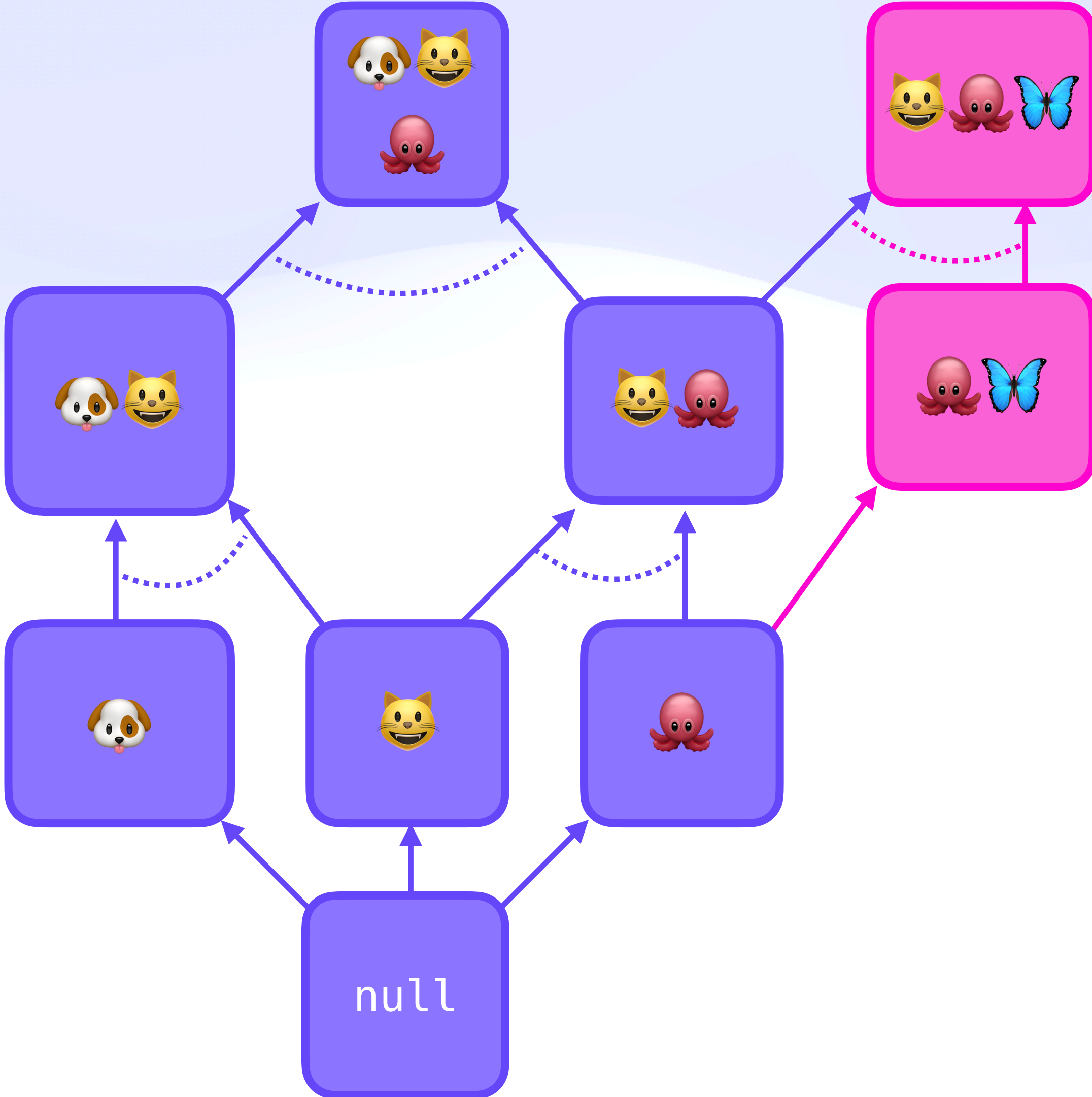
Fault-Safe Concurrency

Alternate History, Same State



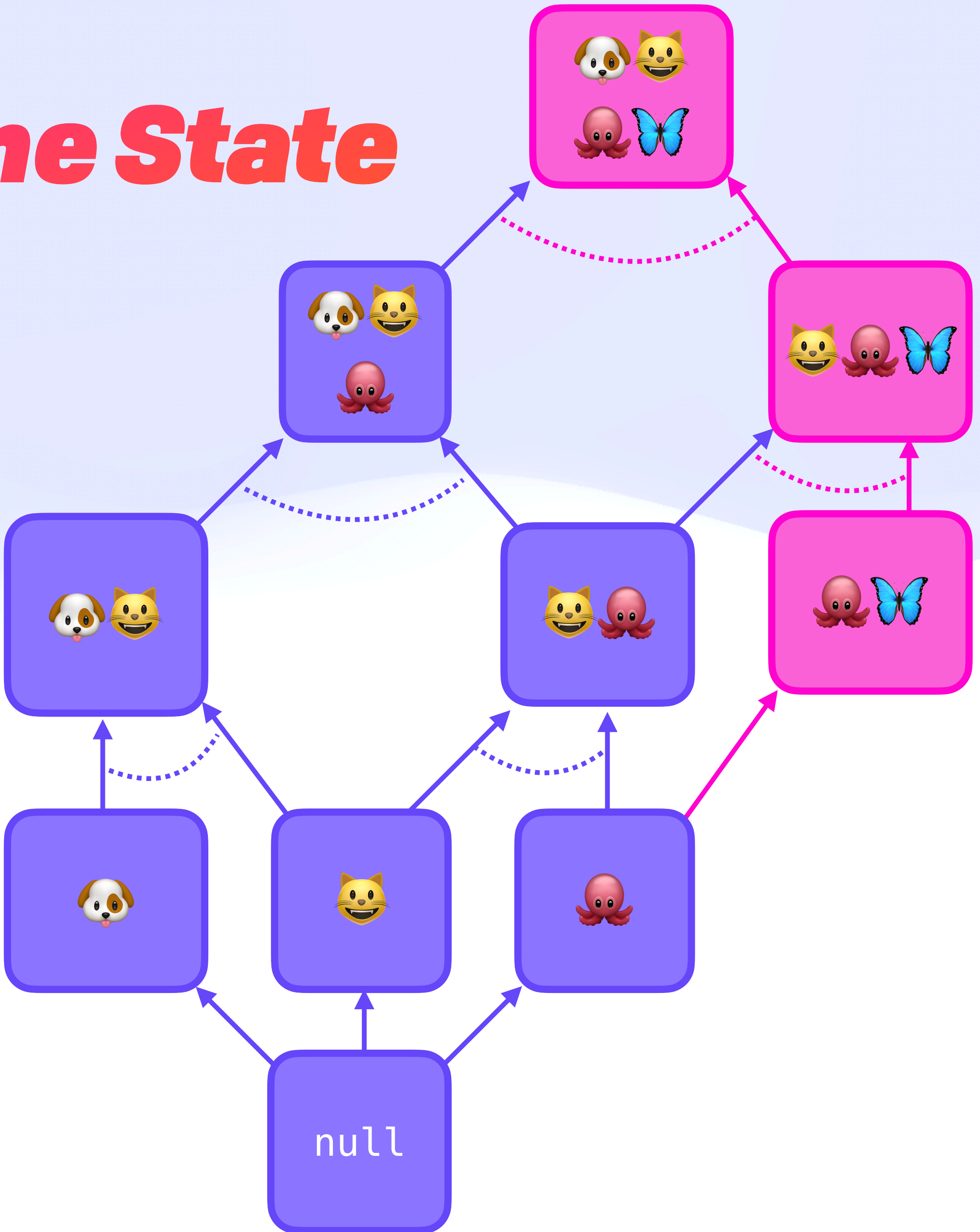
Fault-Safe Concurrency

Alternate History, Same State



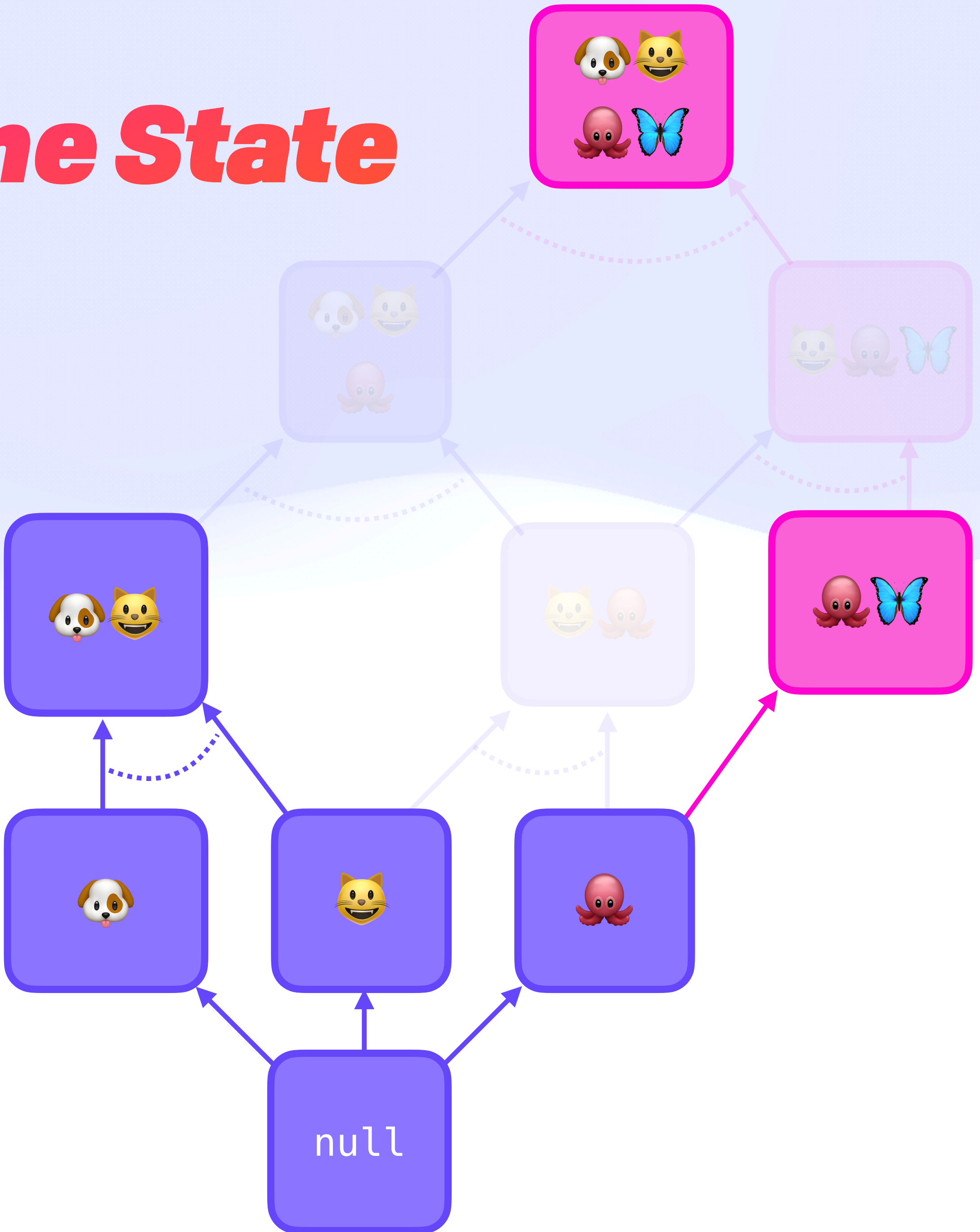
Fault-Safe Concurrency

Alternate History, Same State



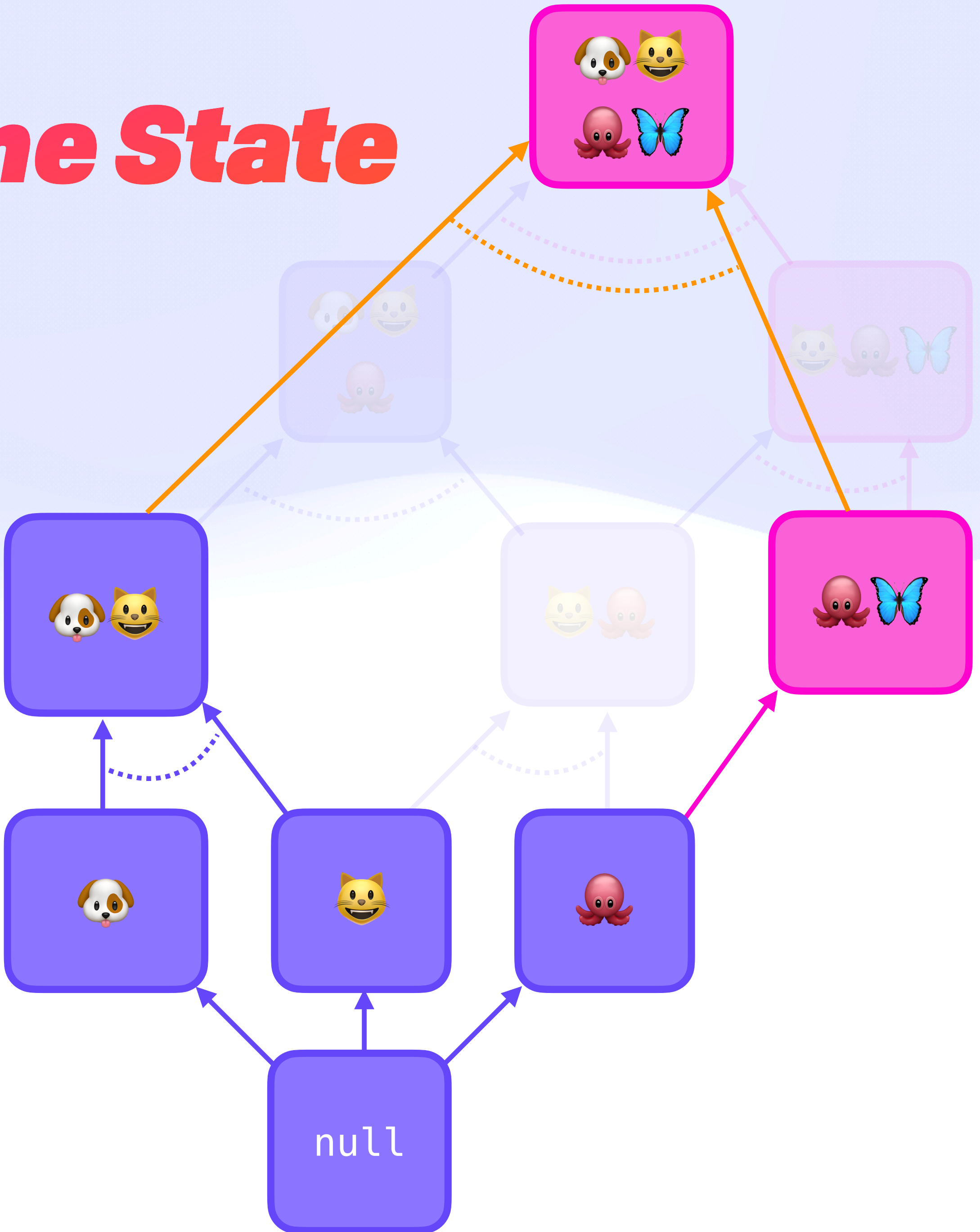
Fault-Safe Concurrency 

Alternate History, Same State



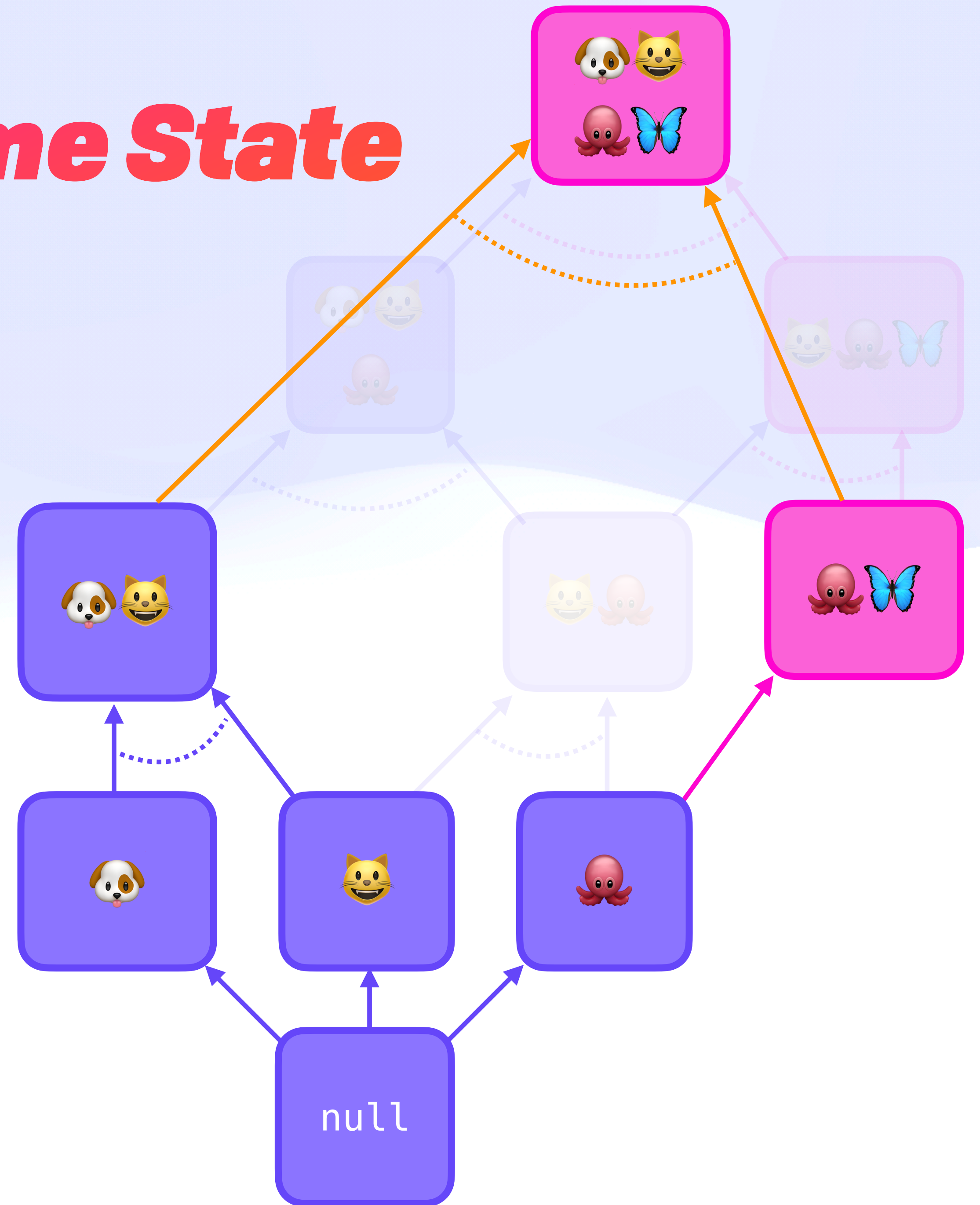
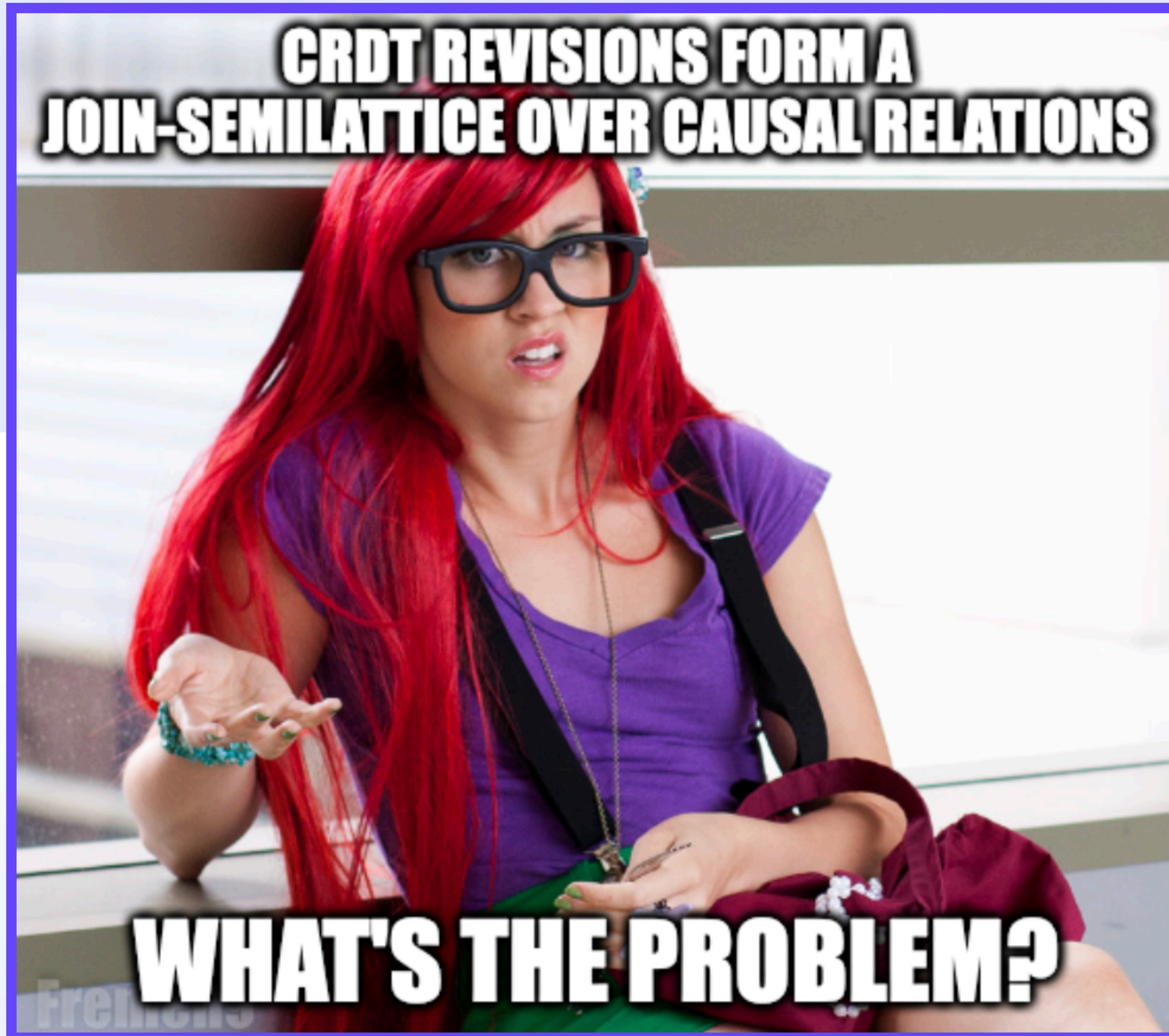
Fault-Safe Concurrency 

Alternate History, Same State



Fault-Safe Concurrency 

Alternate History, Same State



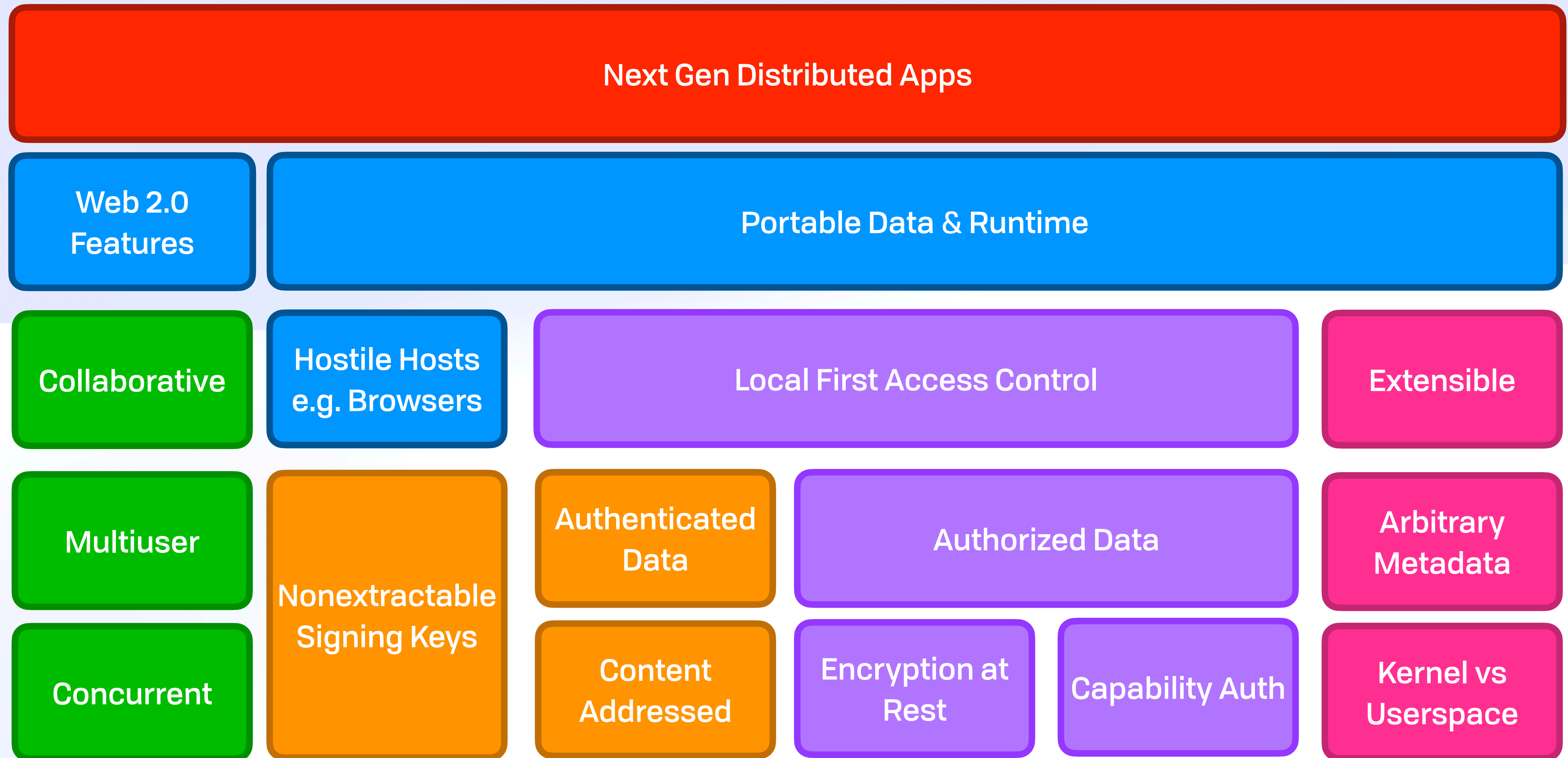
Wrap Up

Thoughts & Field Notes



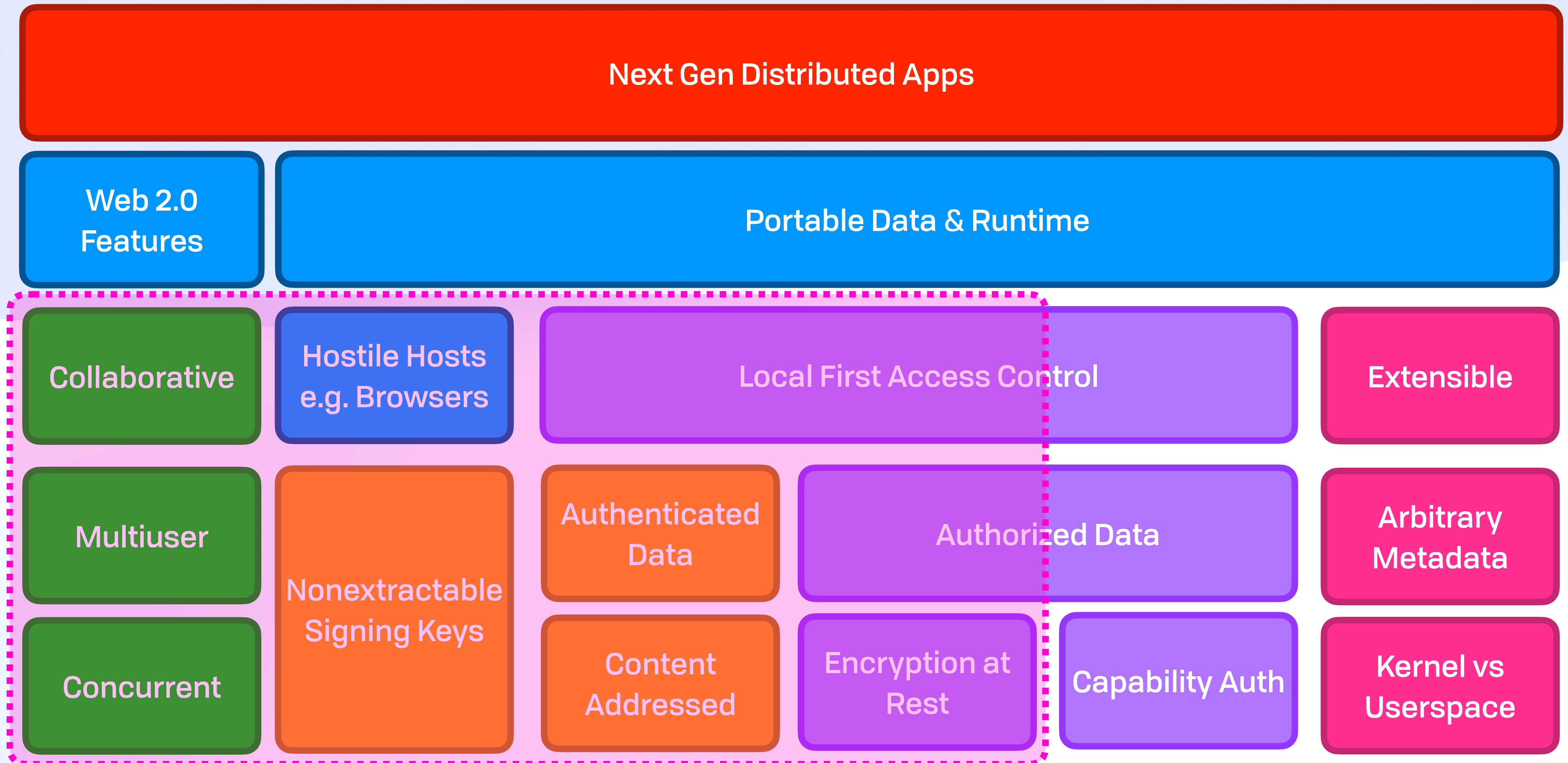
Wrap Up 📁

Requirements Diagram



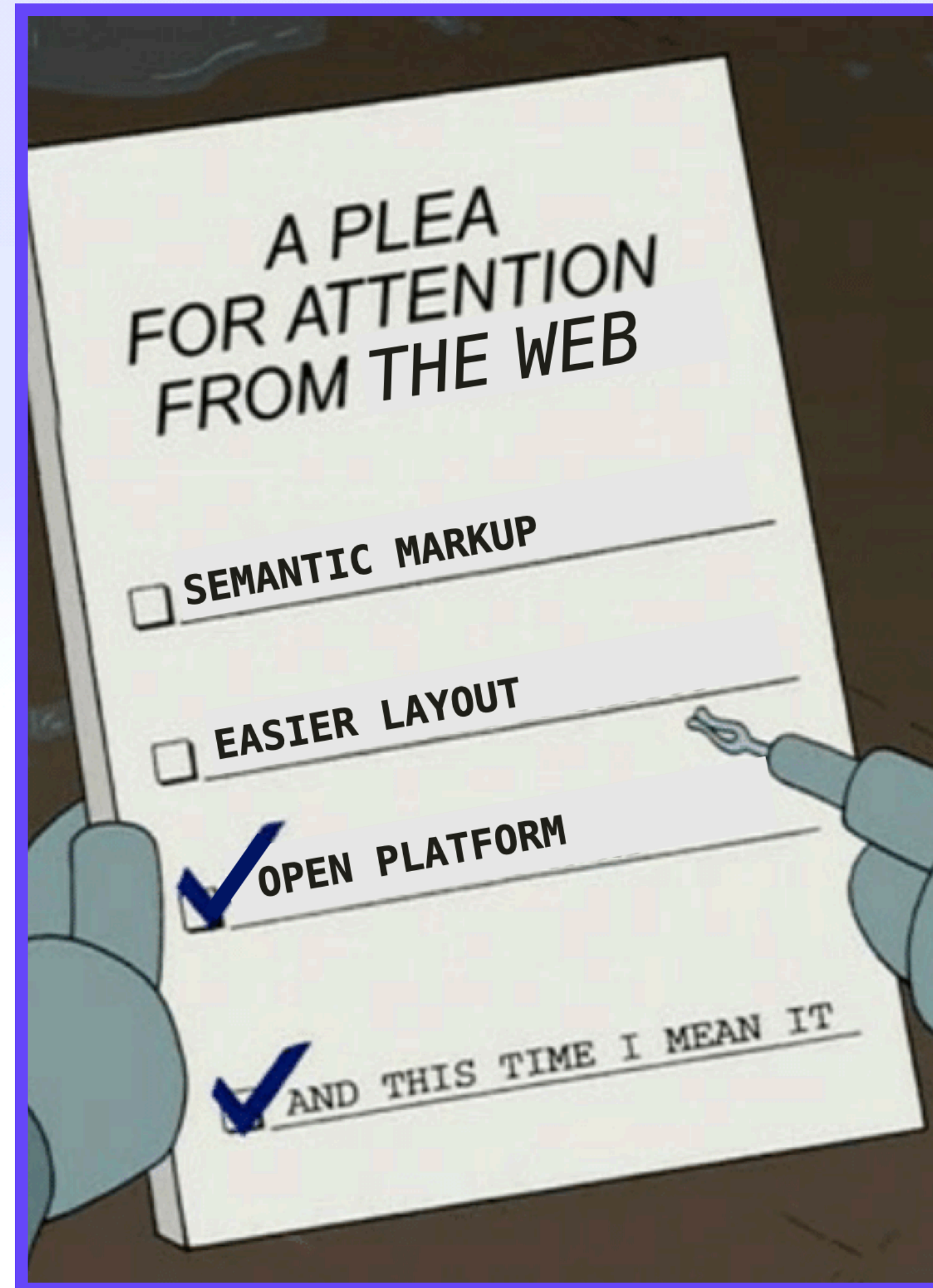
Wrap Up 📺

Requirements Diagram



Wrap Up 📦

Better Together



  **Thank You, CascadiaJS**  

github.com/ucan-wg

github.com/wnfs-wg

More in-depth at Strange Loop 2022 (or talk to us today)