



Web Native File System (WNFS)

Web Native File System

Brooklyn Zelenka, @expede



Web Native File System

Brooklyn Zelenka, @expede

- Cofounder/CTO at Fission
- PLT, VMs, FP, Distributed Systems
- Previously an Ethereum Core Dev
- FOSS — Witchcraft, Exceptional, &c



Web Native 

Web Native

Web Apps are Too Hard — Shrink the Development Cycle

Web Native

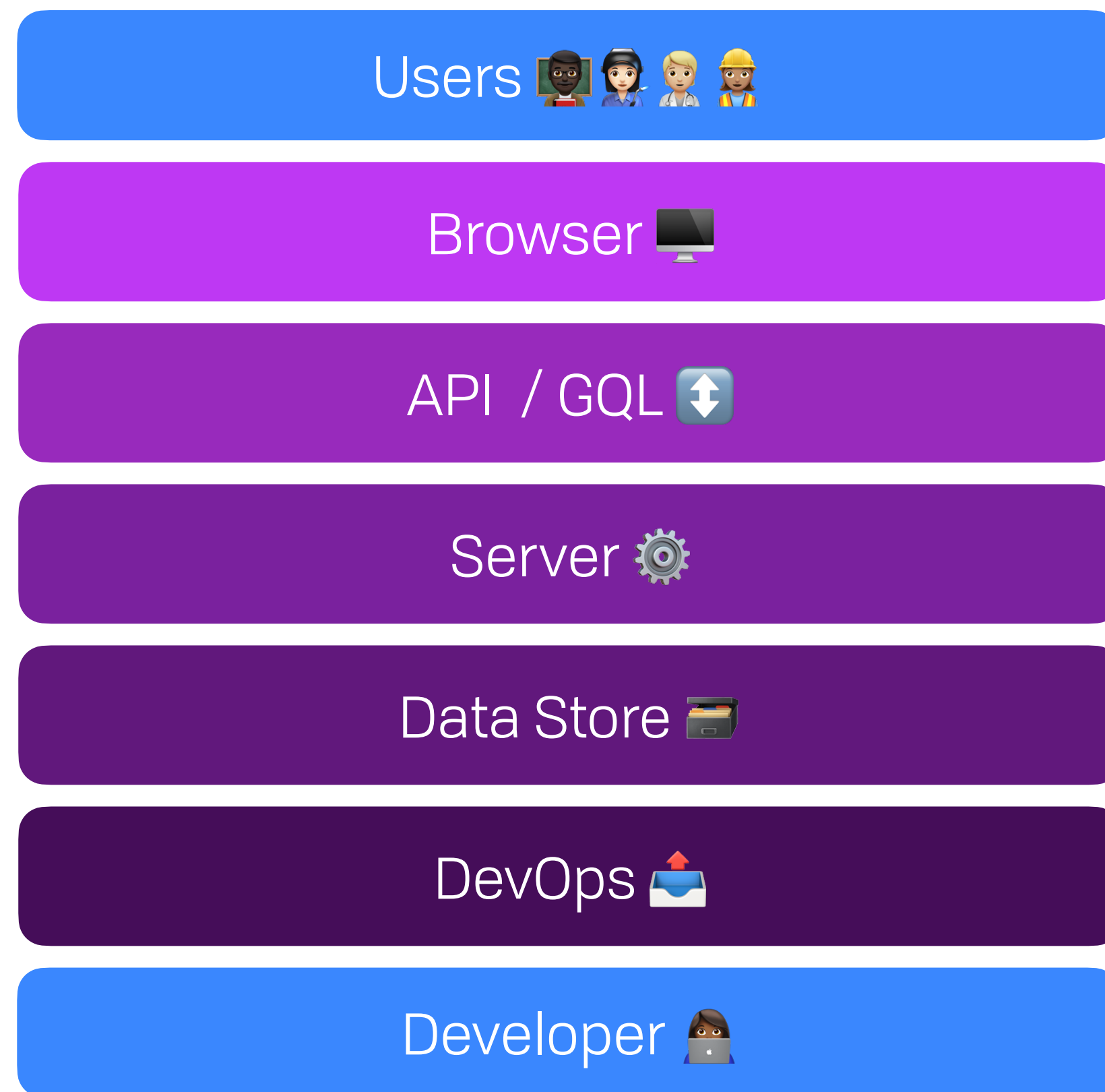
Web Apps are Too Hard — Shrink the Development Cycle

Users 

Developer 

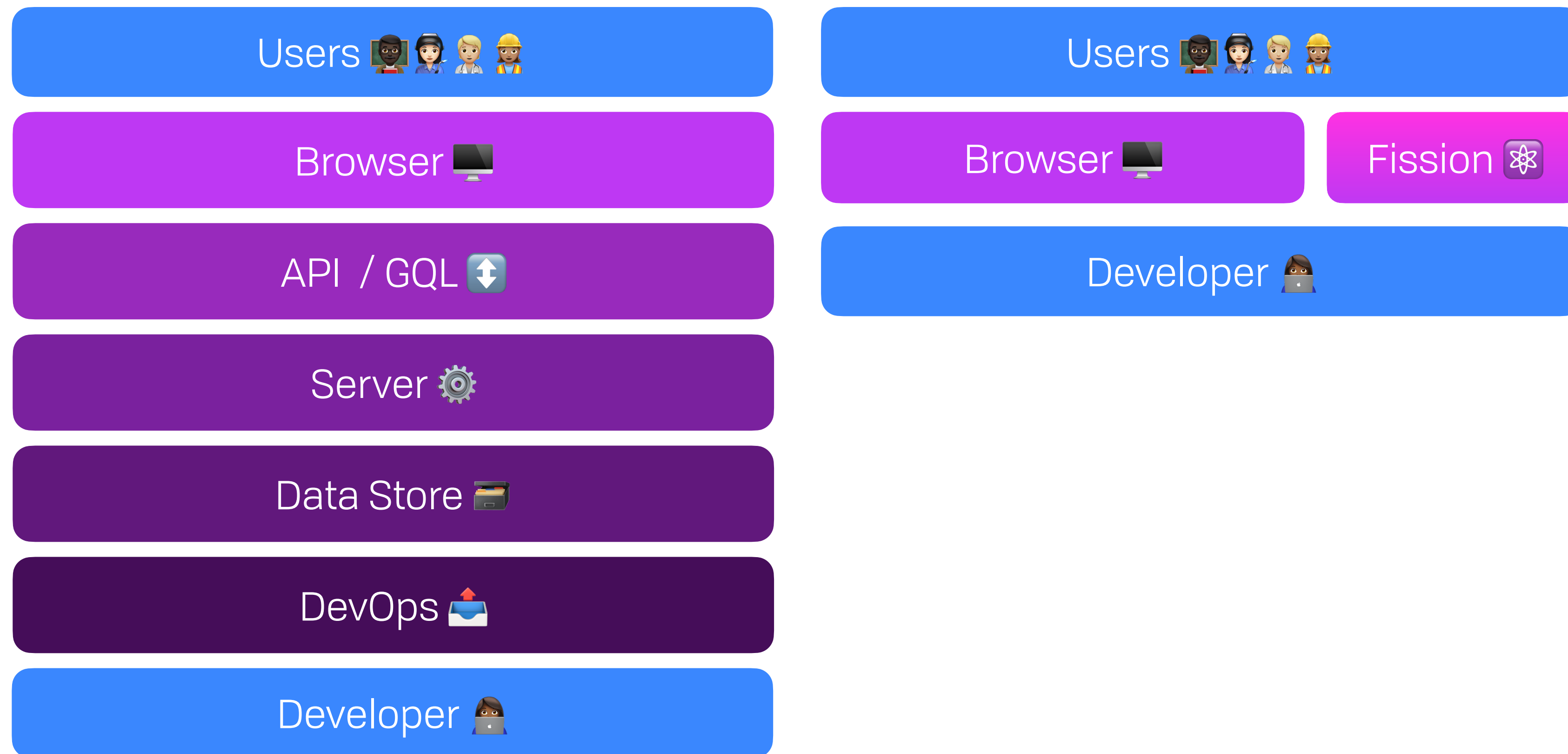
Web Native

Web Apps are Too Hard — Shrink the Development Cycle



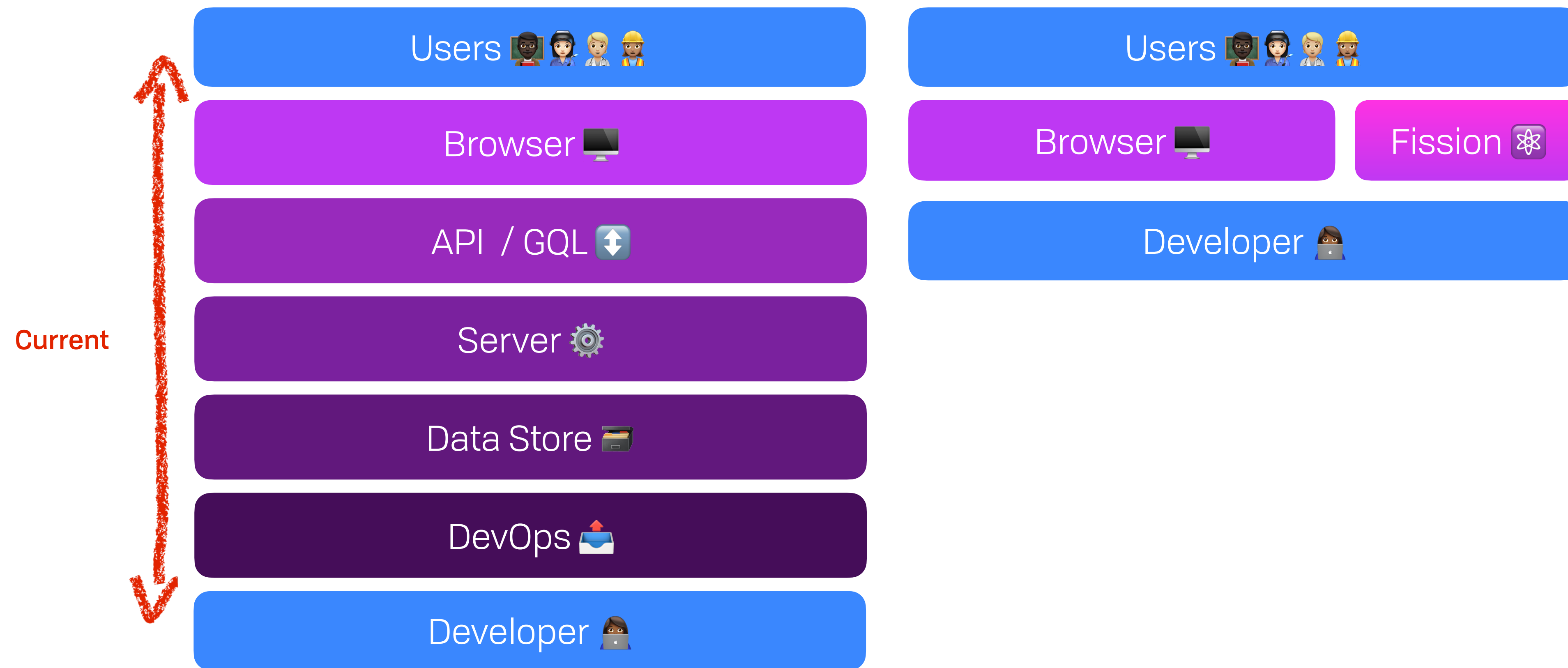
Web Native

Web Apps are Too Hard — Shrink the Development Cycle



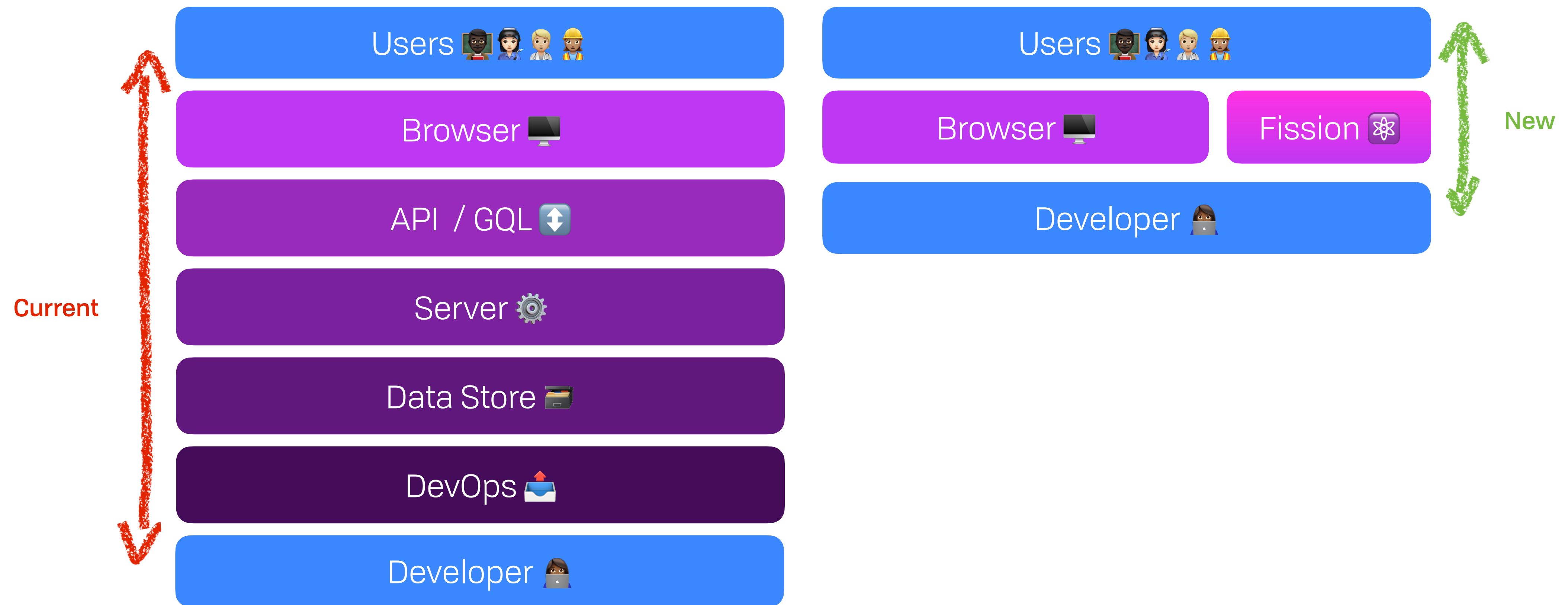
Web Native

Web Apps are Too Hard — Shrink the Development Cycle



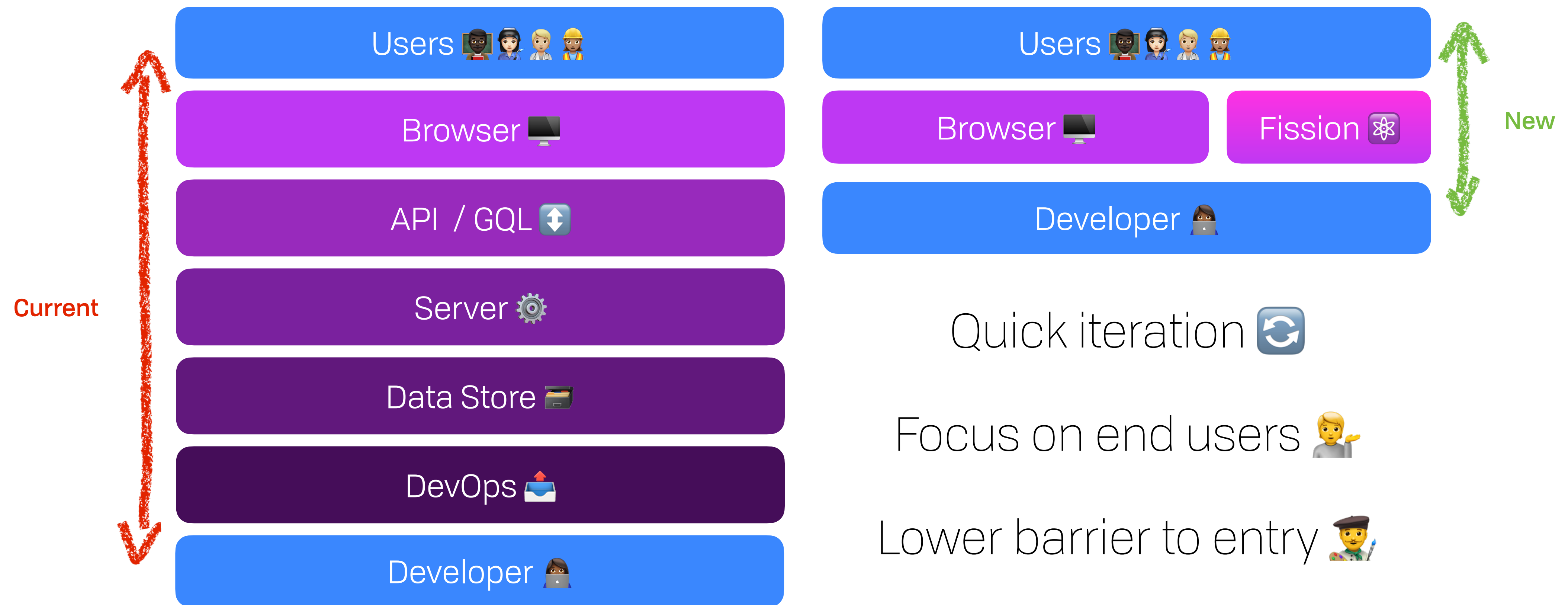
Web Native

Web Apps are Too Hard — Shrink the Development Cycle



Web Native

Web Apps are Too Hard — Shrink the Development Cycle



Web Native

New assumptions ~ new approach

- Different world from web's beginnings
- Don't need to rely on client/server
- New features naturally fall out of the architecture
- Location independent data
- Client side encryption
- Networked *machines* → networked *data*

Web Native

Abstraction Layers

Web Native

Abstraction Layers



Application Layer

Business Logic



Platform Layer

WebNative File System



Protocol Layer

IPFS

Web Native Components

Web Native Components

AuthN & AuthZ

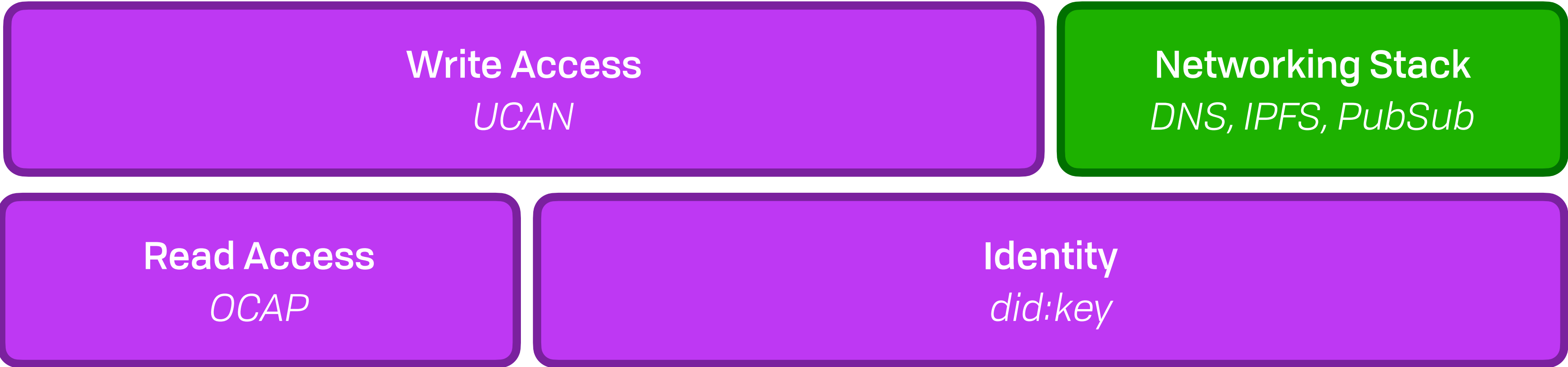
Write Access
UCAN

Read Access
OCAP

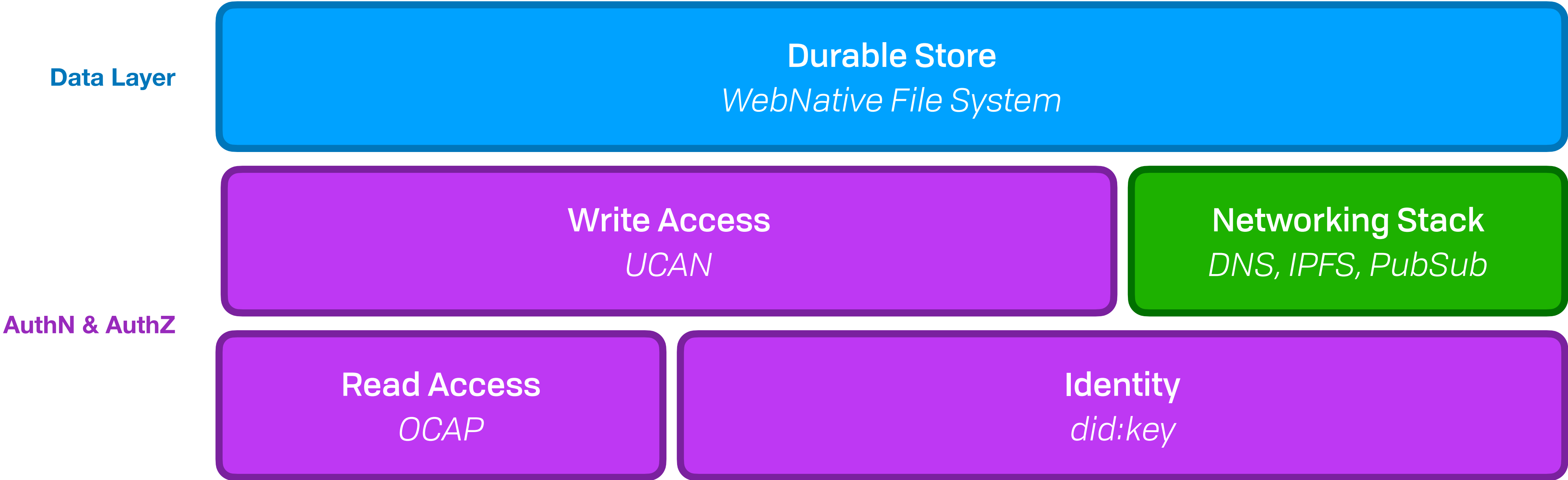
Identity
did:key

Web Native Components

AuthN & AuthZ



Web Native Components



Web Native Components

Multi Device / User

Offline & Async Sharing
Exchange Store

Soft Realtime Multiuser
Automerge CRDT

Data Layer

Durable Store
WebNative File System

AuthN & AuthZ

Write Access
UCAN

Networking Stack
DNS, IPFS, PubSub

Read Access
OCAP

Identity
did:key

Web Native Components

Multi Device / User

App Abstractions

Fission SDK

Offline & Async Sharing

Exchange Store

Soft Realtime Multiuser

Automerge CRDT

Data Layer

Durable Store

WebNative File System

AuthN & AuthZ

Write Access

UCAN

Networking Stack

DNS, IPFS, PubSub

Read Access

OCAP

Identity

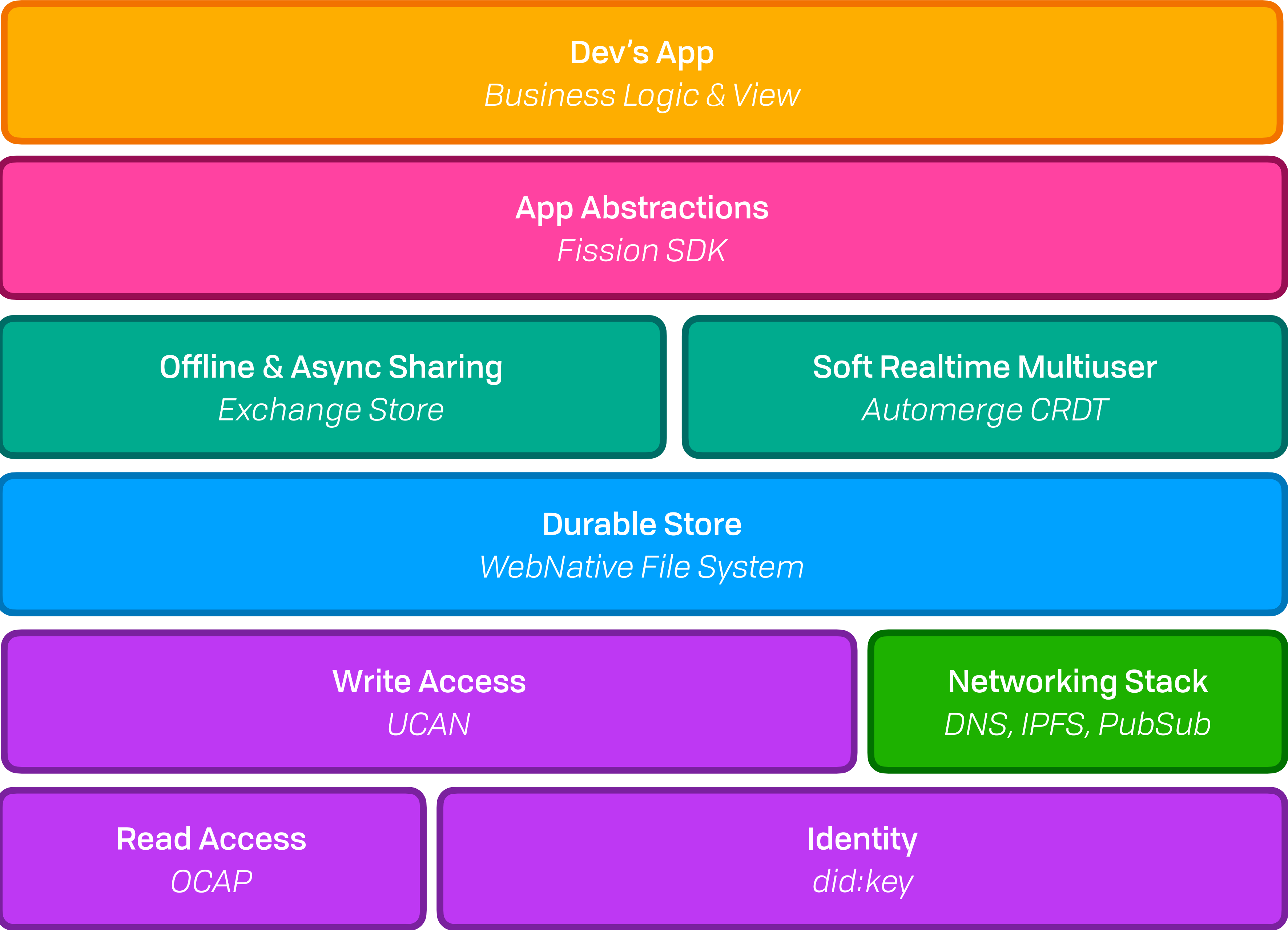
did:key

Web Native Components

Multi Device / User

Data Layer

AuthN & AuthZ



Web Native File System

File System

 High Level Layout

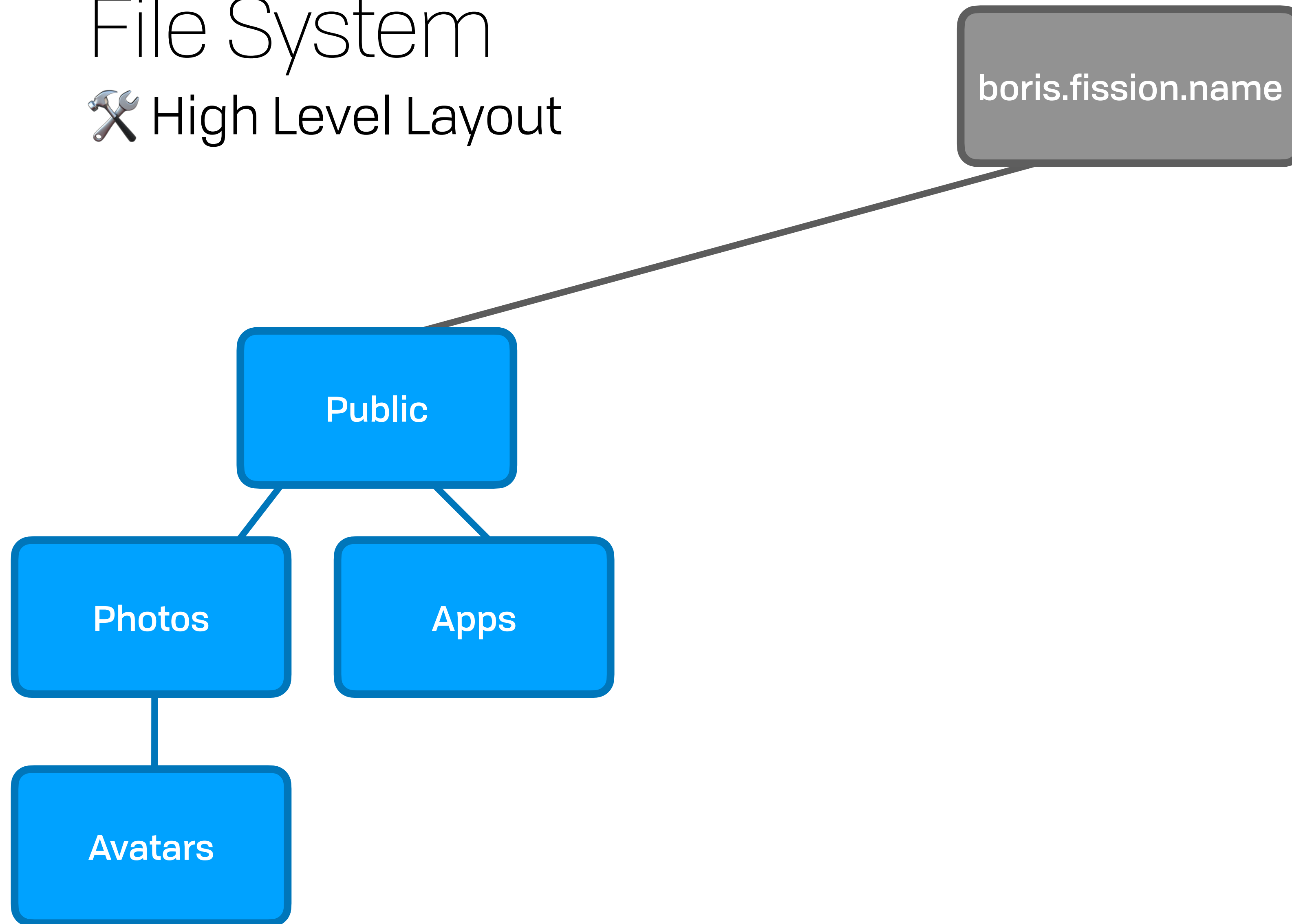
File System

 High Level Layout

boris.fission.name

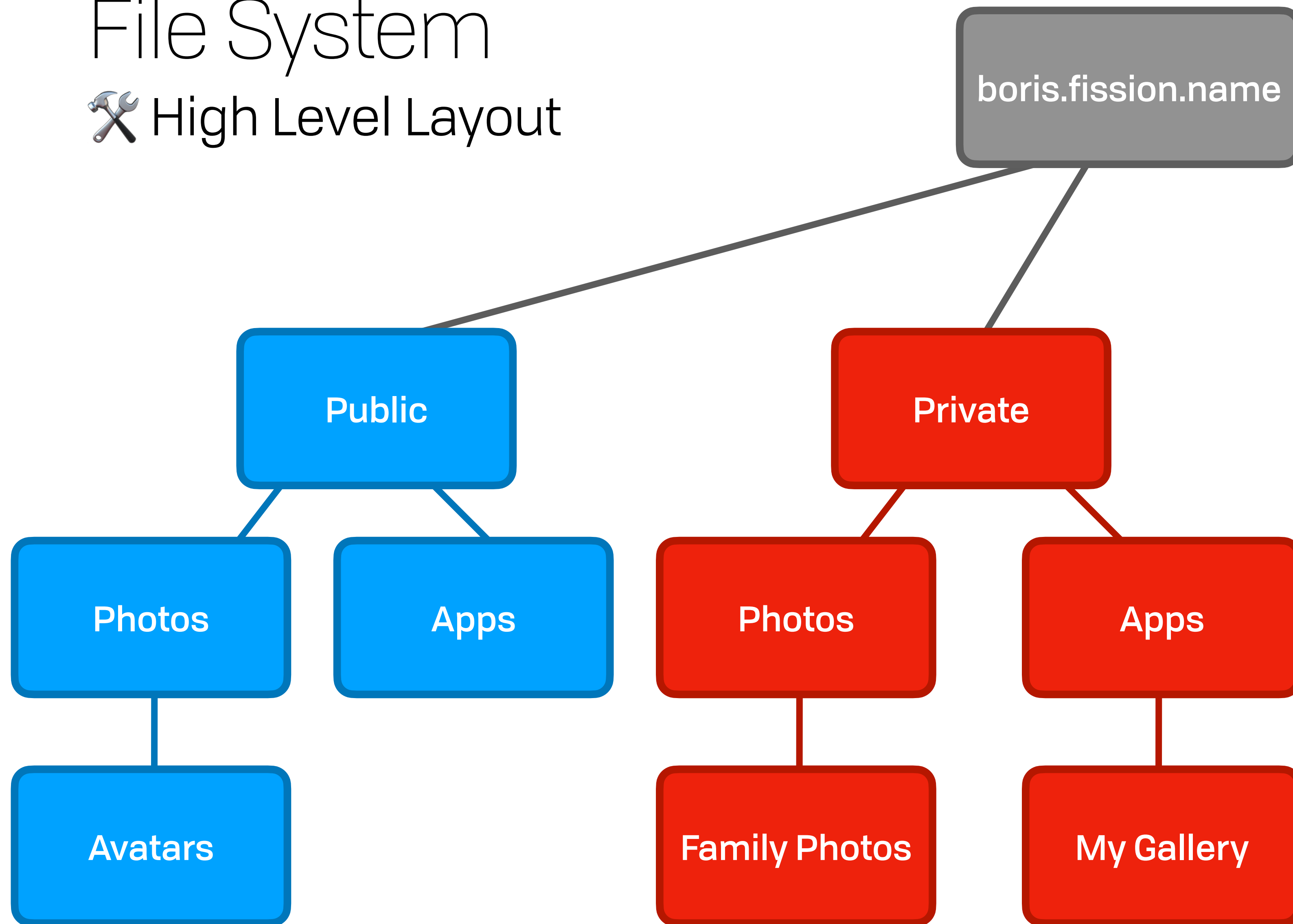
File System

🔧 High Level Layout



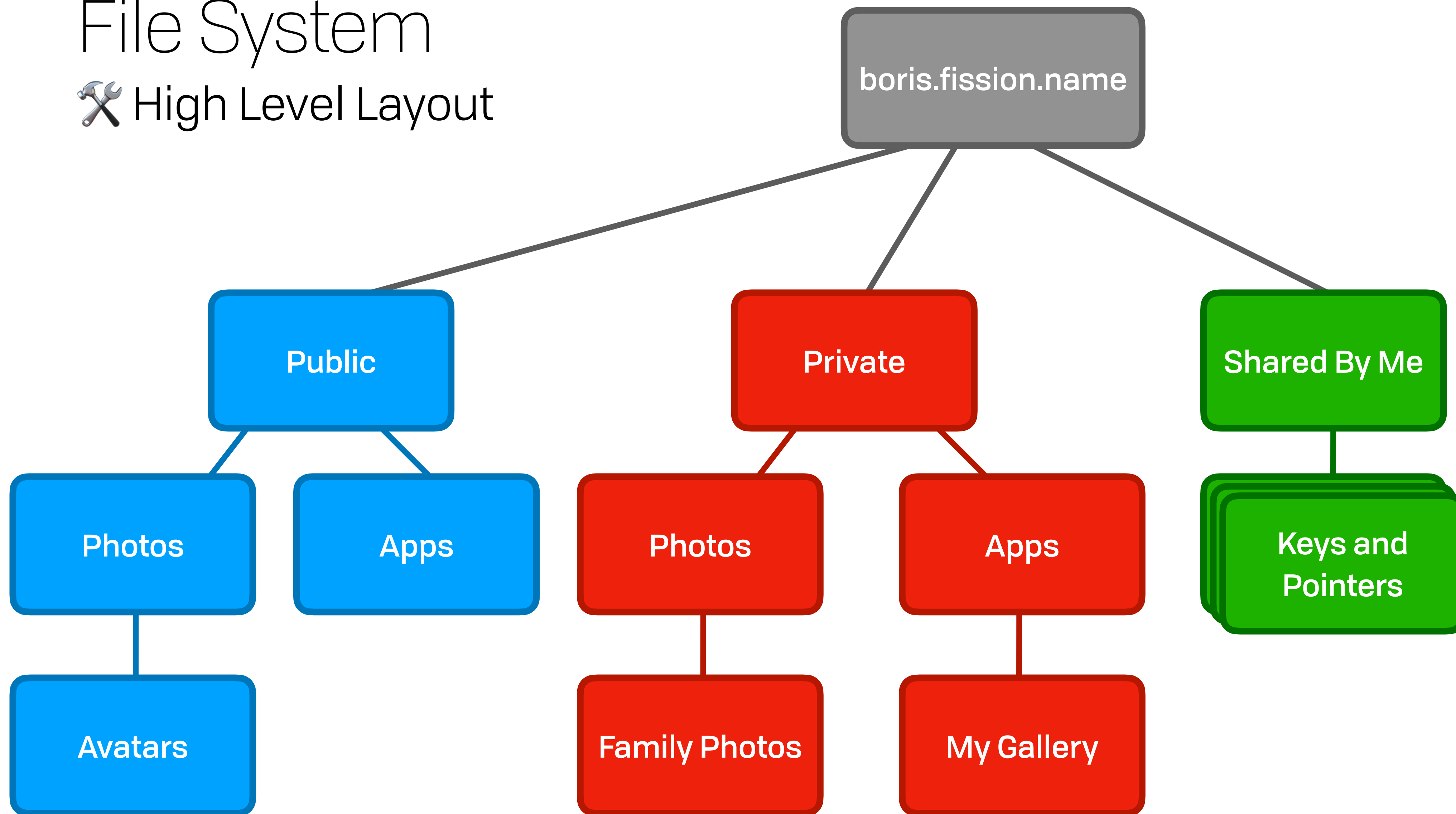
File System

🔧 High Level Layout



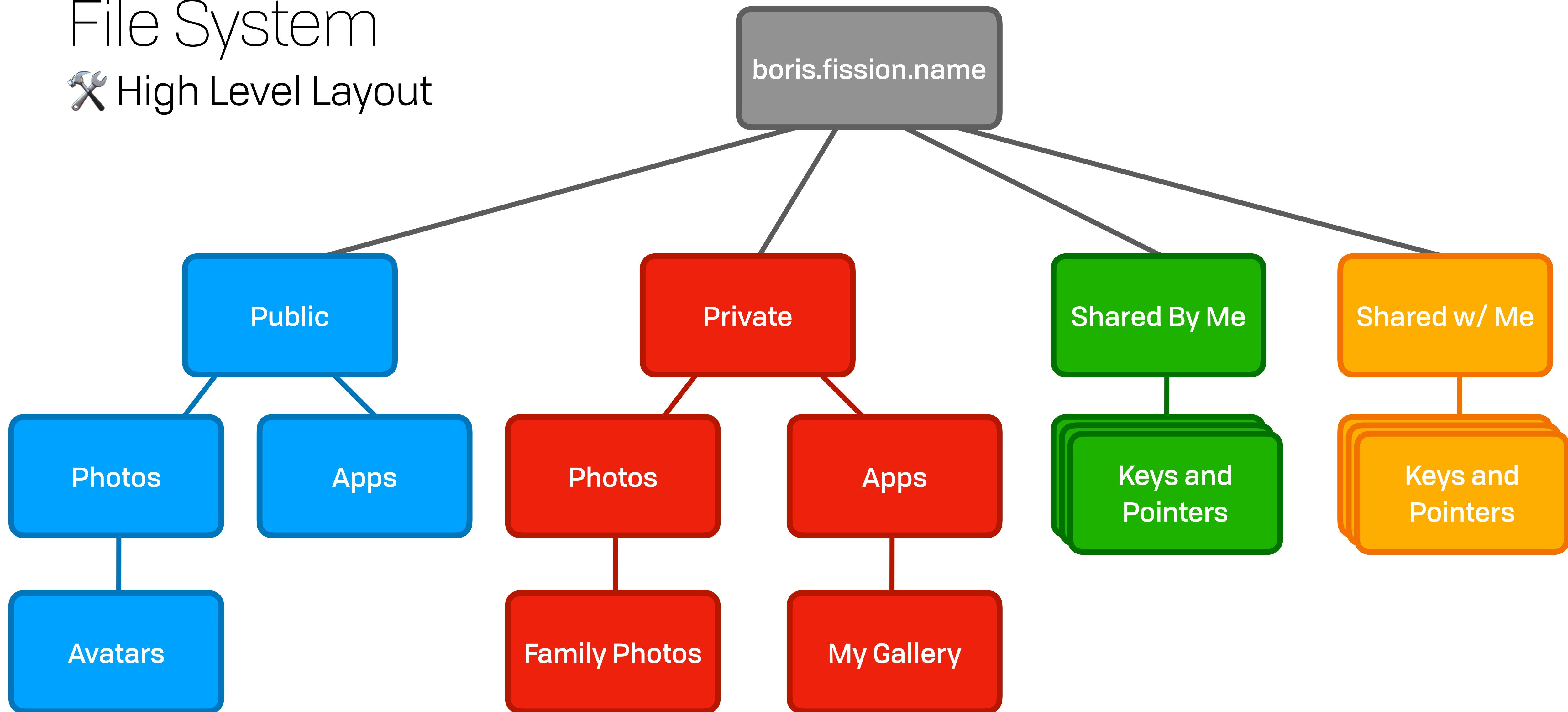
File System

🔧 High Level Layout



File System

🔧 High Level Layout



File System

Quick Demo



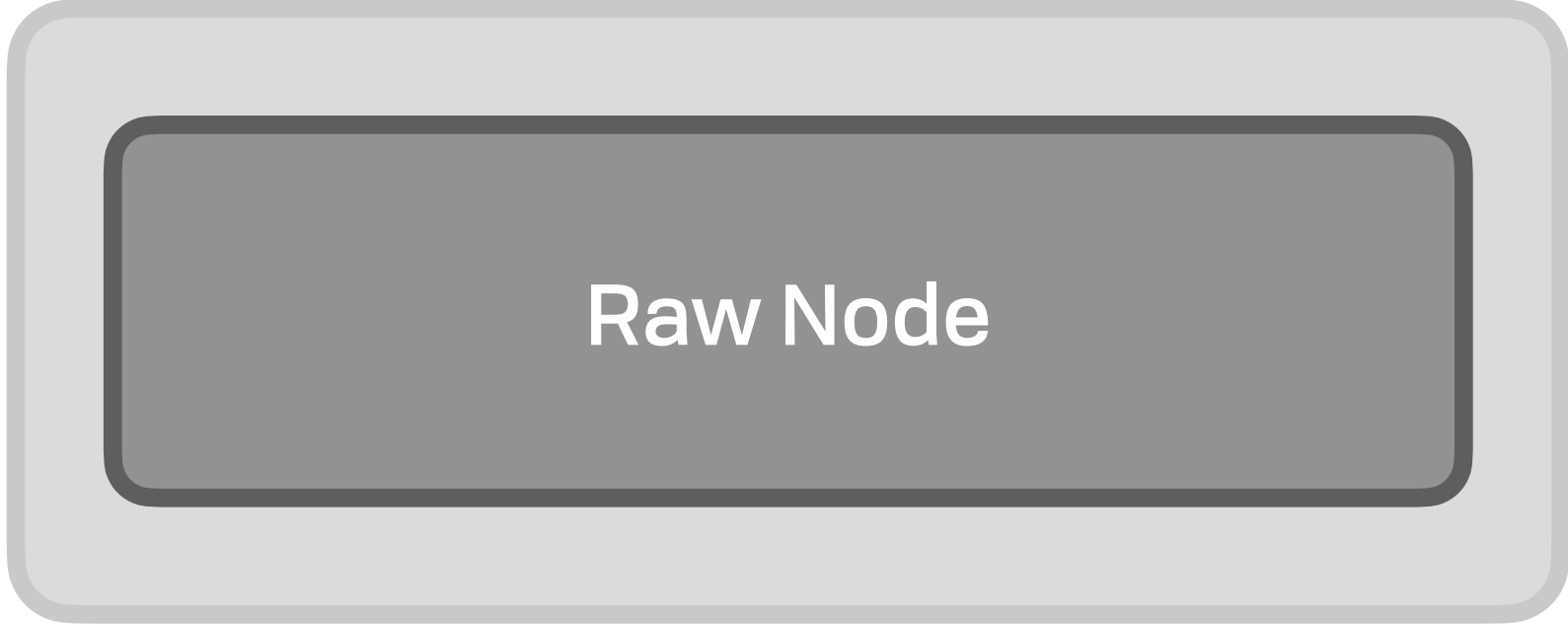
Common Concepts 🤝

Common Concepts

  Virtual Nodes

Common Concepts

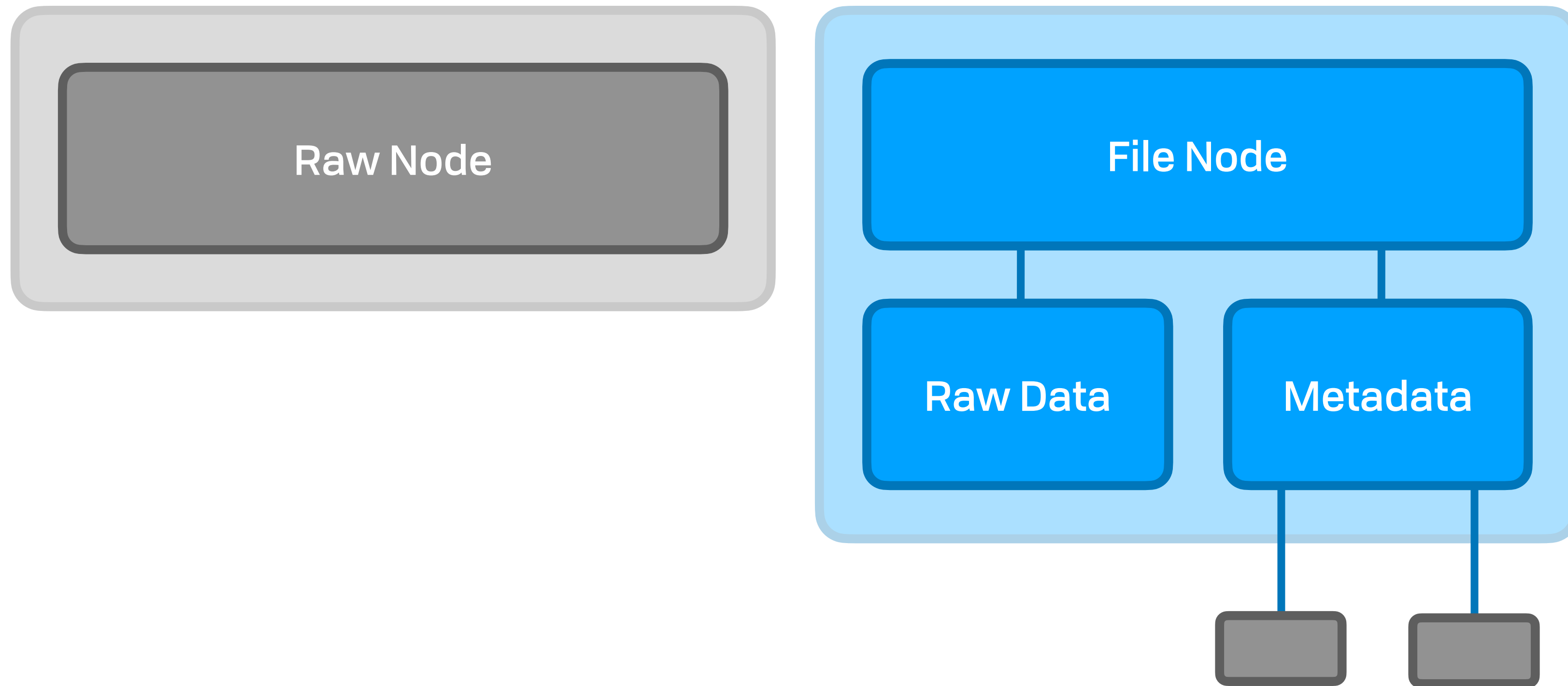
  Virtual Nodes



Raw Node

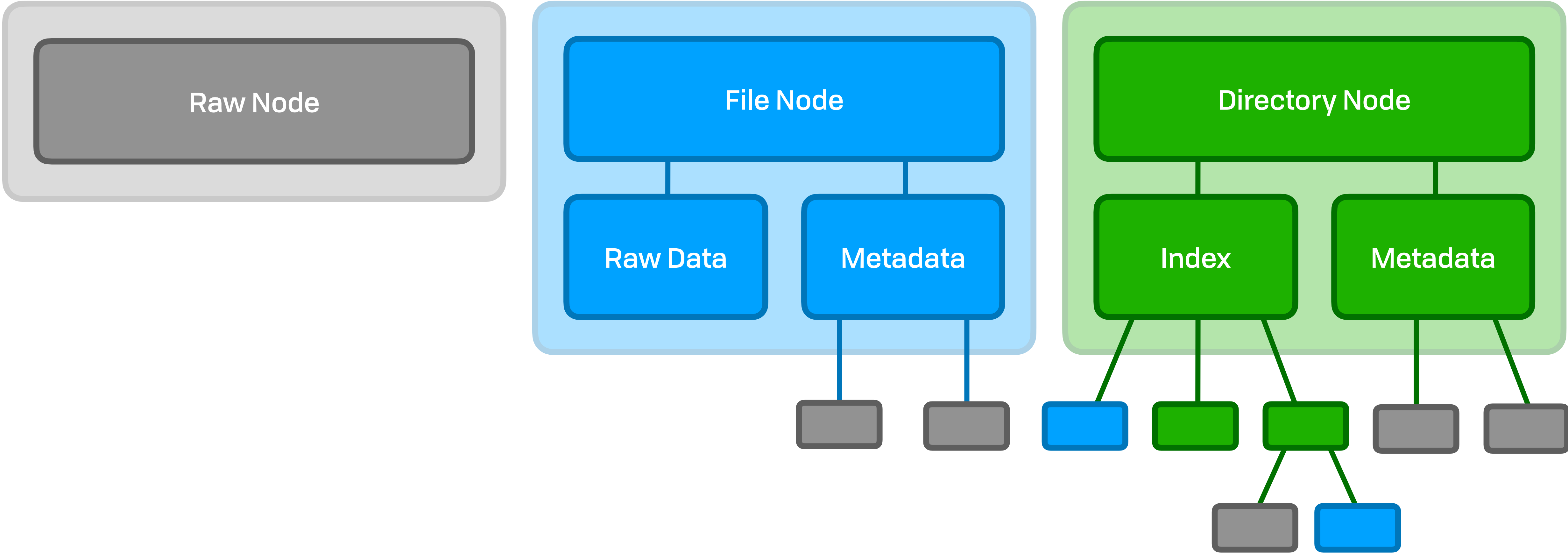
Common Concepts

🔧🌐 Virtual Nodes



Common Concepts

🔧🌐 Virtual Nodes



Common Concepts

Virtual Nodes

Raw Node

- Virtual Node
 - Consistent interface
- Arbitrary metadata
 - More than POSIX
 - Tags, creators, MIME, sources, &c

File Node

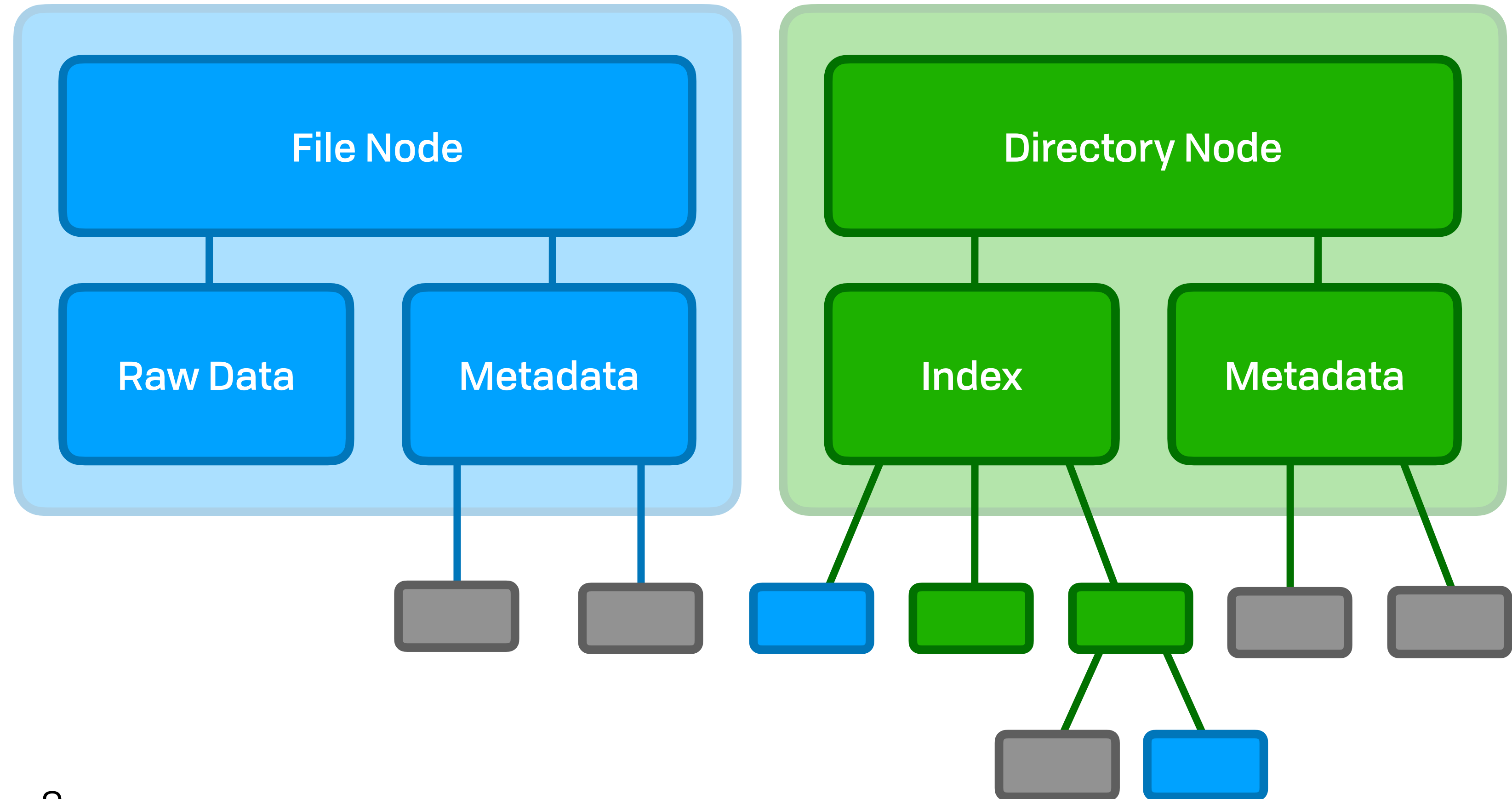
Raw Data

Metadata

Directory Node

Index

Metadata



Common Concepts

 Hard vs Soft Links

Common Concepts

Hard vs Soft Links

- Hard links
 - Direct reference
 - 2 pointers ~ duplicate

Common Concepts

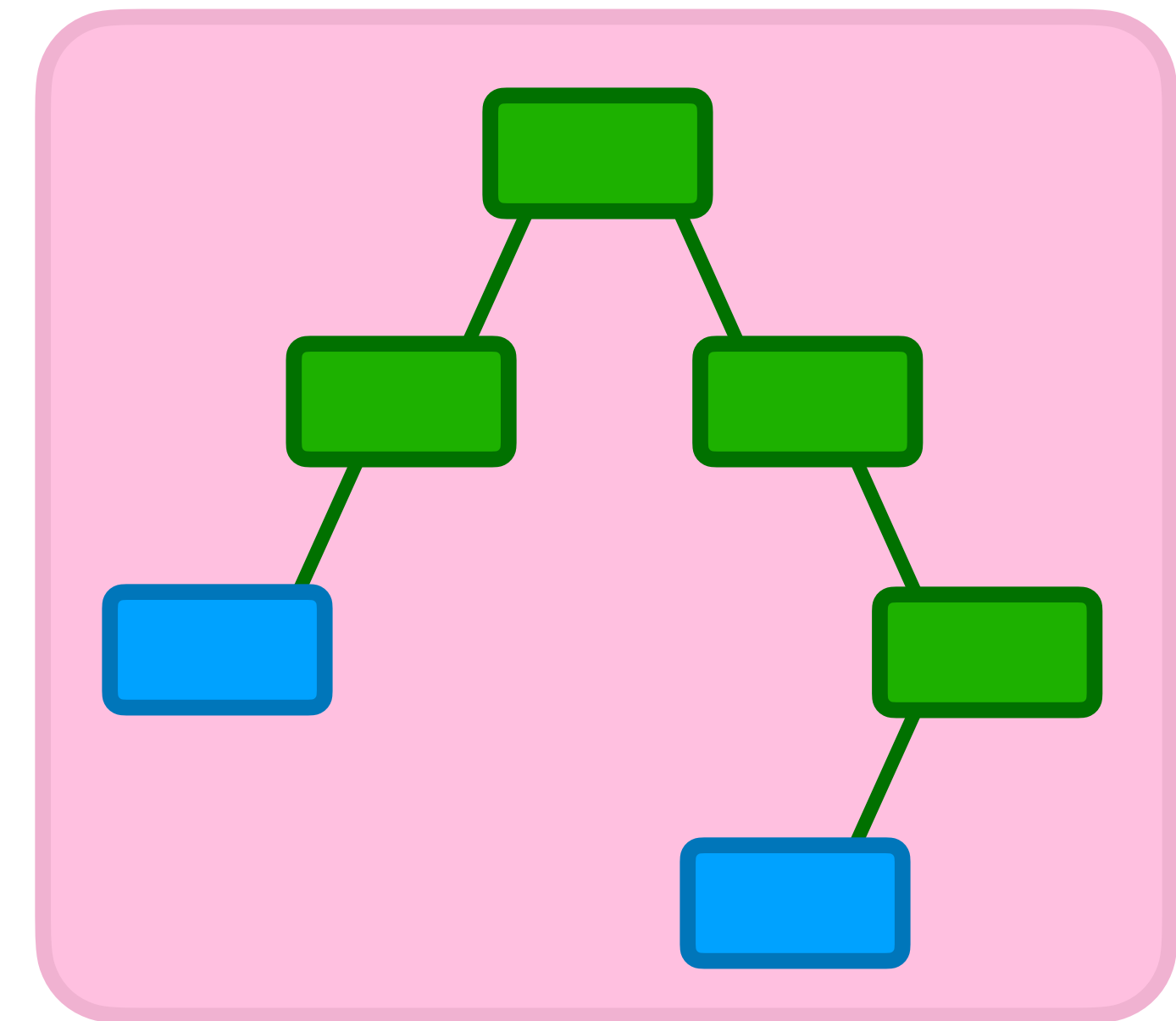
Hard vs Soft Links

- Hard links
 - Direct reference
 - 2 pointers ~ duplicate
- Soft links
 - Like a symlink or web link
 - 2 pointers ~ latest
 - May break
 - Always *some* version available

Common Concepts

🔧 Hard vs Soft Links

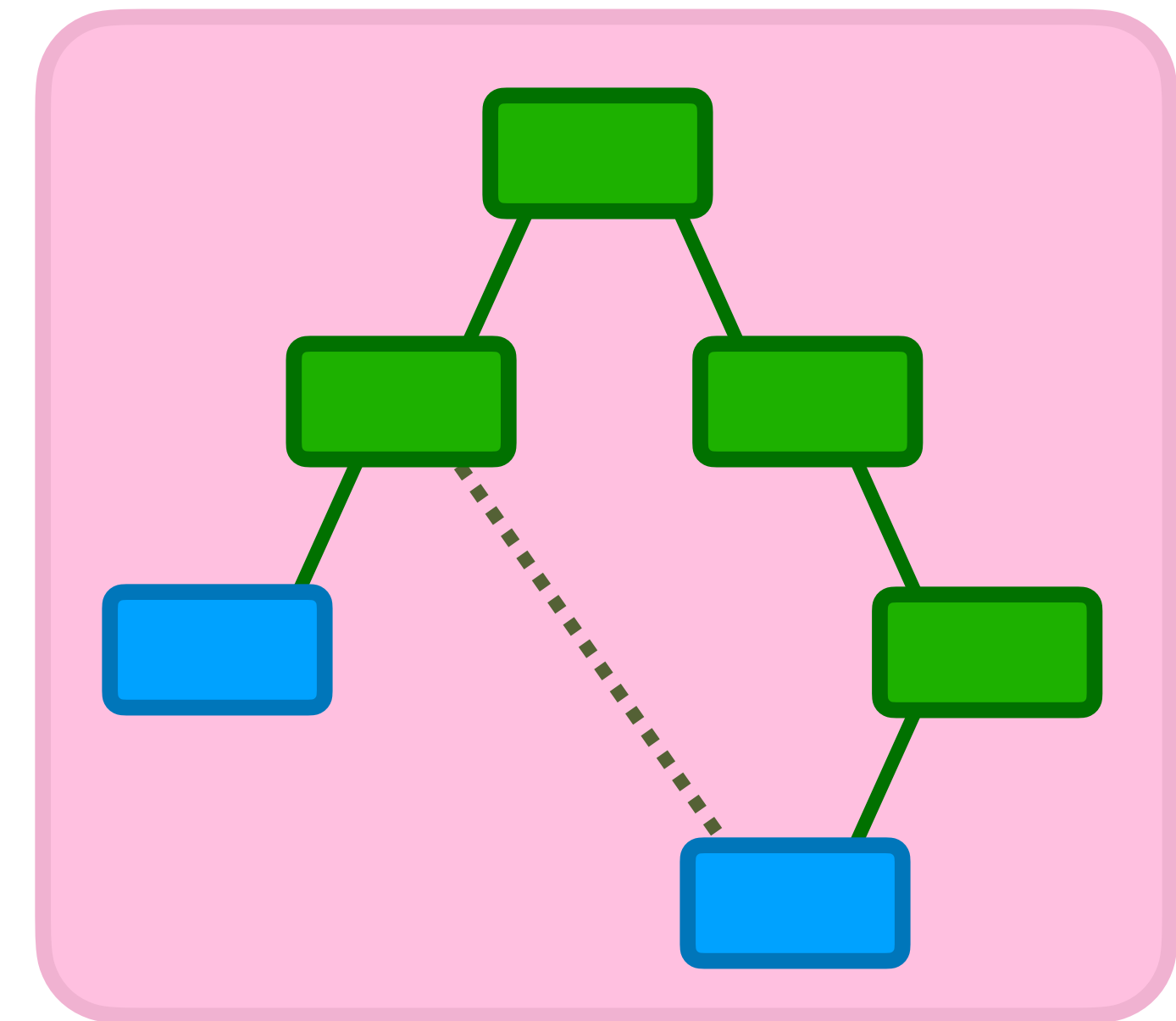
- Hard links
 - Direct reference
 - 2 pointers ~ duplicate
- Soft links
 - Like a symlink or web link
 - 2 pointers ~ latest
 - May break
 - Always *some* version available



Common Concepts

🔧 Hard vs Soft Links

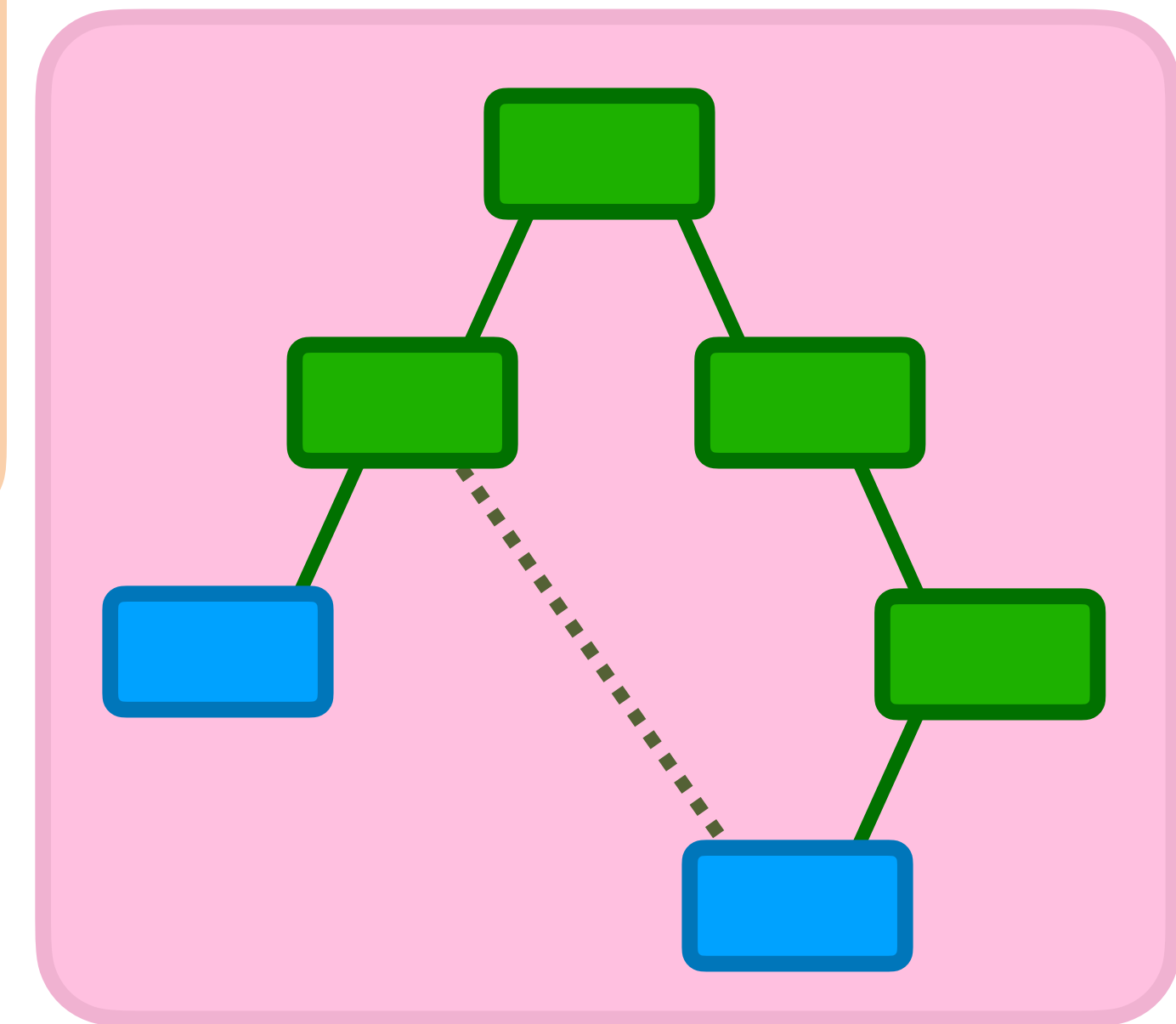
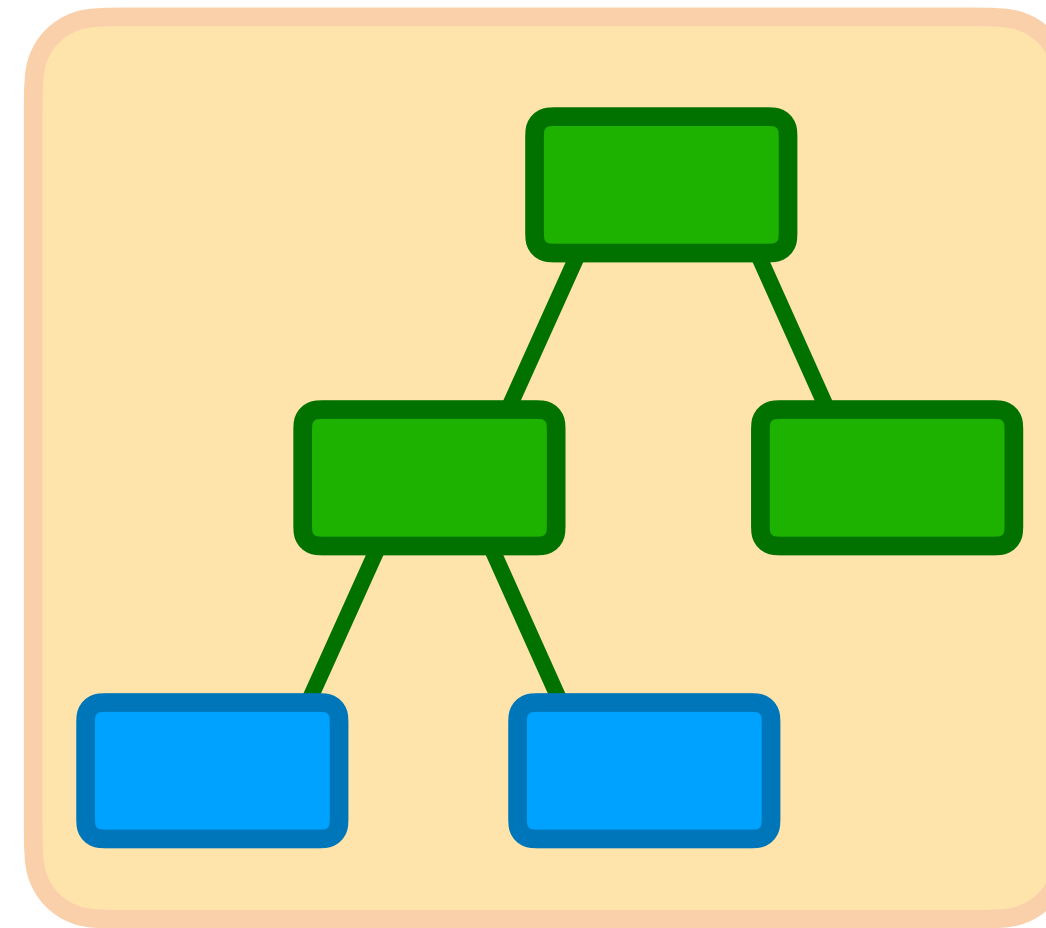
- Hard links
 - Direct reference
 - 2 pointers ~ duplicate
- Soft links
 - Like a symlink or web link
 - 2 pointers ~ latest
 - May break
 - Always *some* version available



Common Concepts

🔧 Hard vs Soft Links

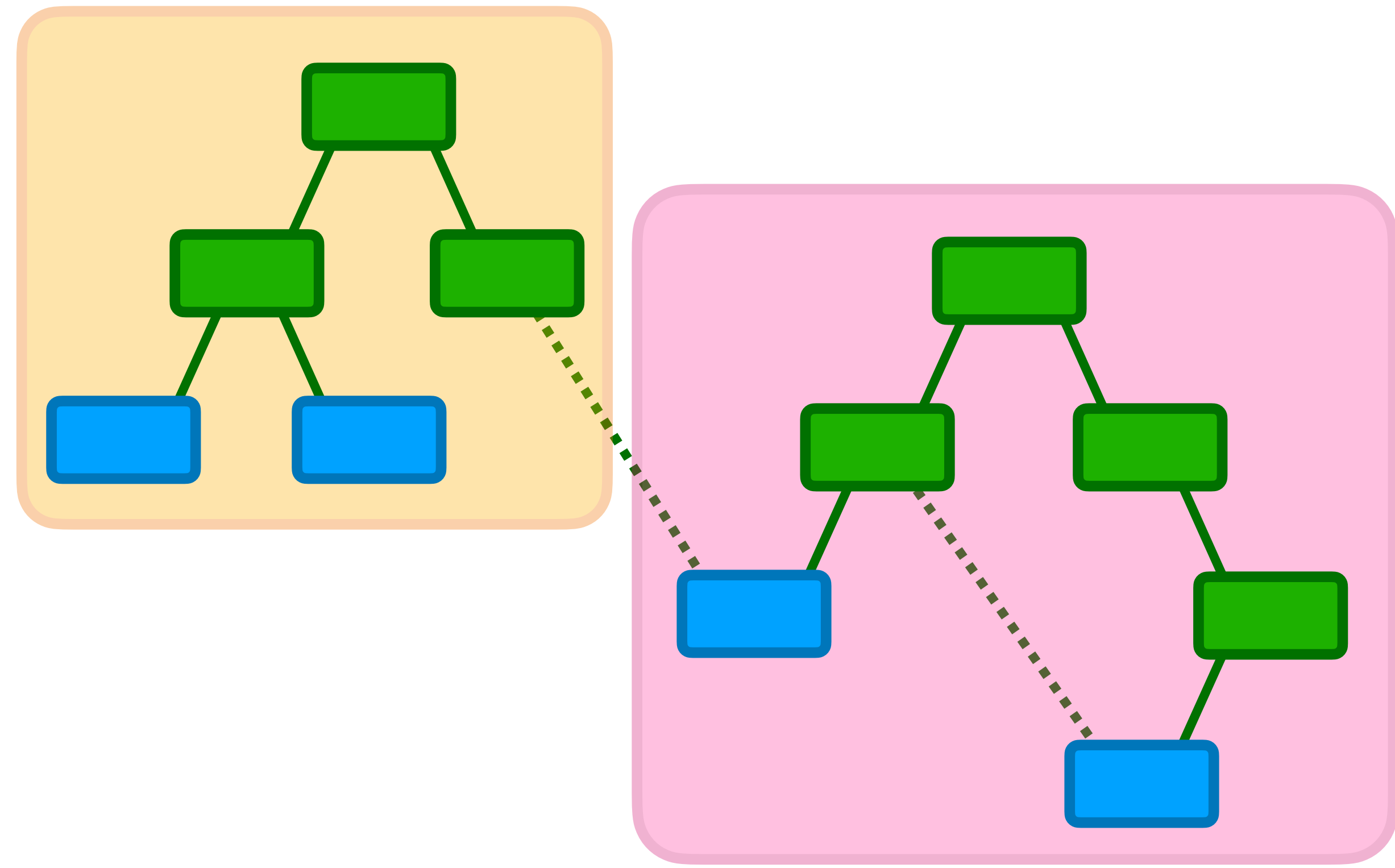
- Hard links
 - Direct reference
 - 2 pointers ~ duplicate
- Soft links
 - Like a symlink or web link
 - 2 pointers ~ latest
 - May break
 - Always *some* version available



Common Concepts

🔧 Hard vs Soft Links

- Hard links
 - Direct reference
 - 2 pointers ~ duplicate
- Soft links
 - Like a symlink or web link
 - 2 pointers ~ latest
 - May break
 - Always *some* version available

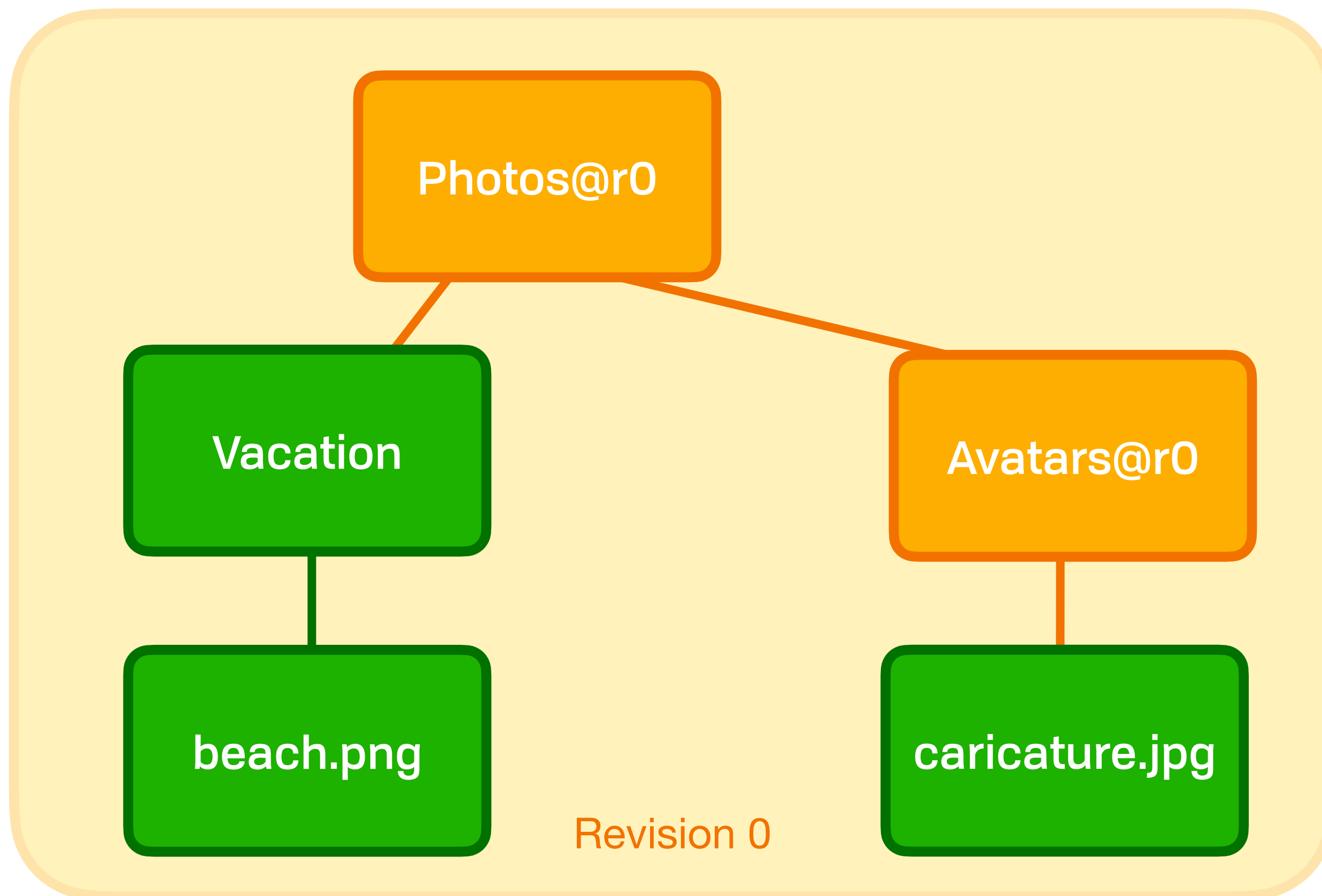


File System

🔧 The z-dimension: versioning & events

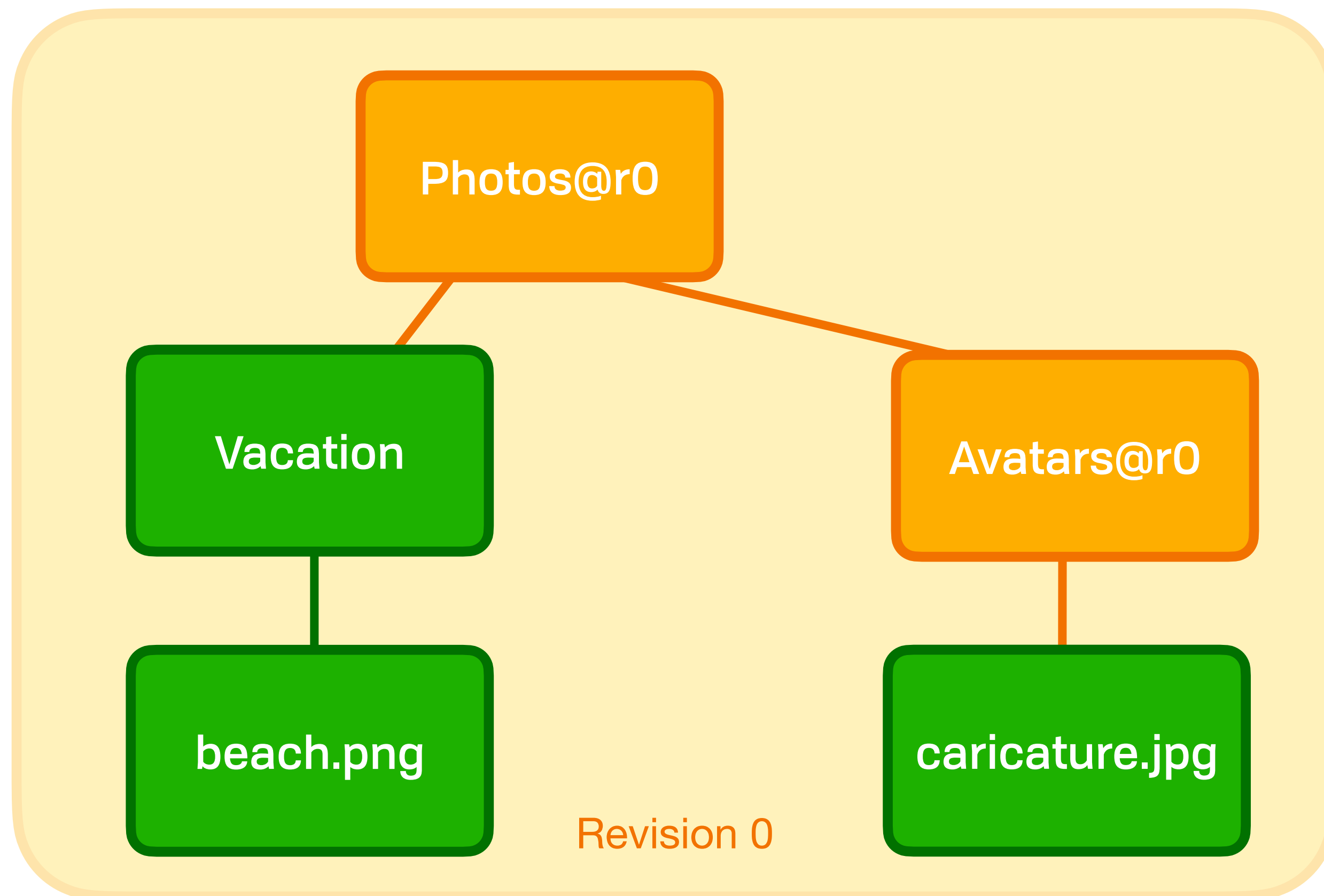
File System

🔧 The z-dimension: versioning & events



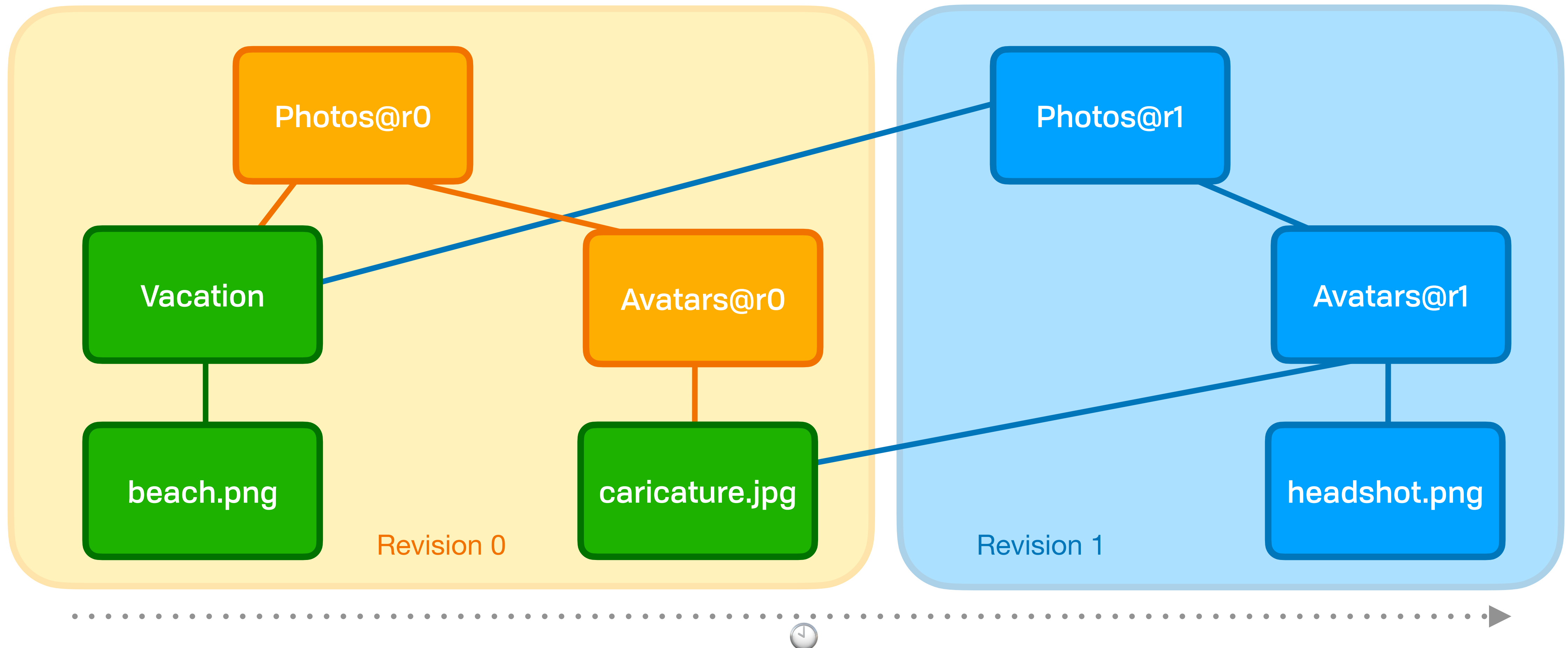
File System

🔧 The z-dimension: versioning & events



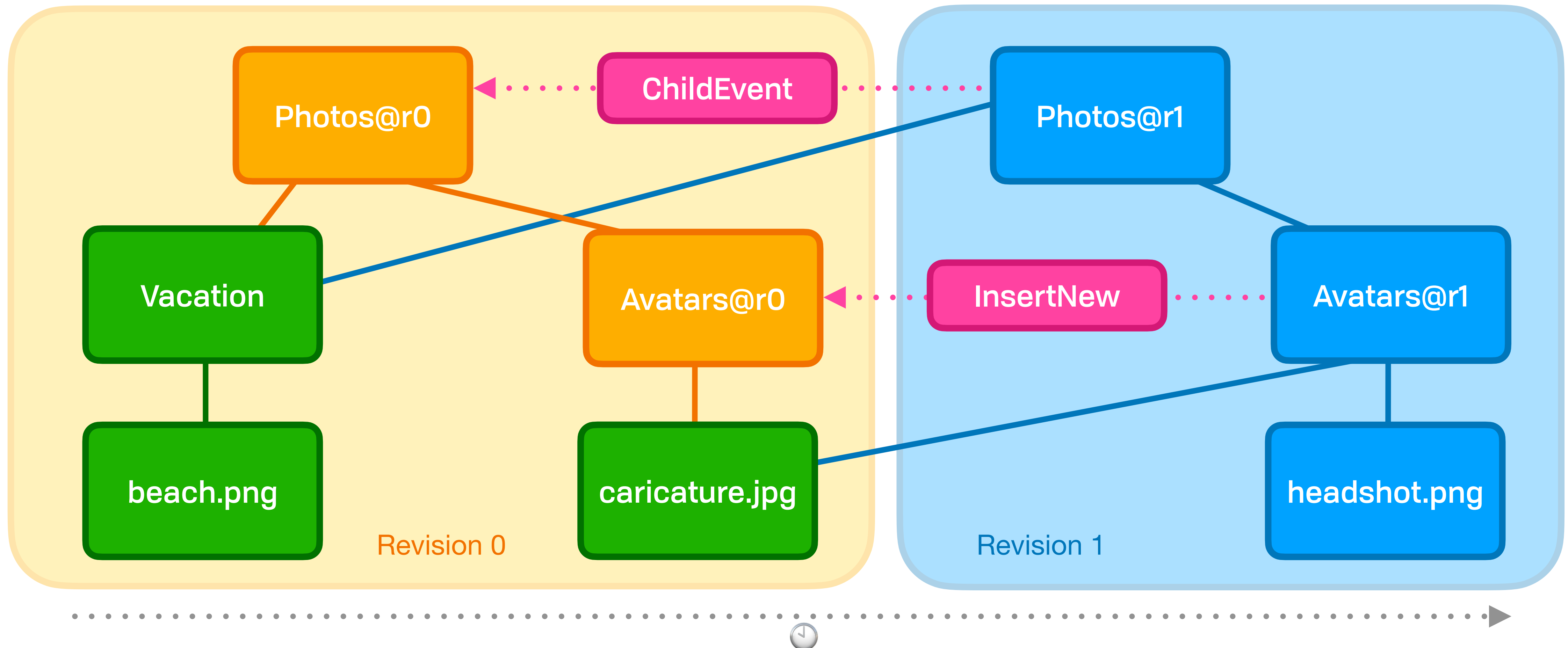
File System

🔧 The z-dimension: versioning & events



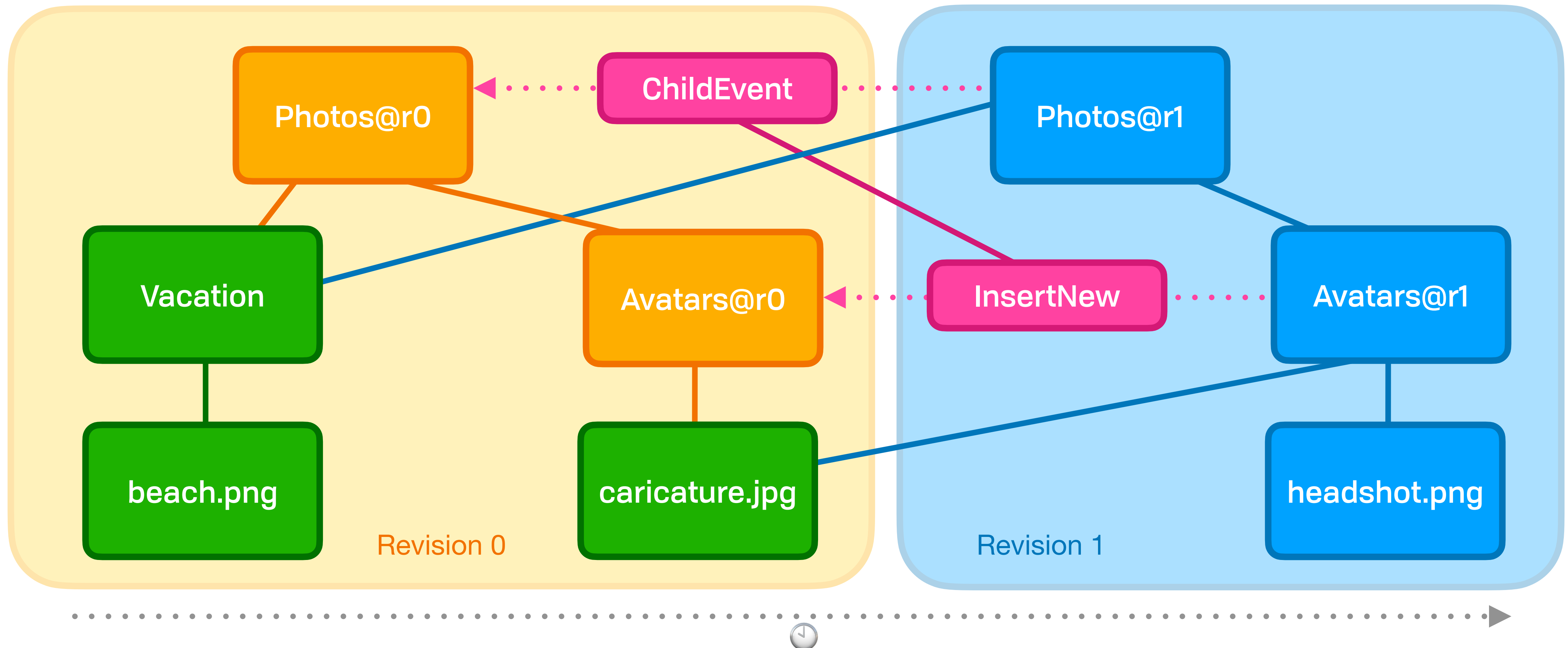
File System

🔧 The z-dimension: versioning & events



File System

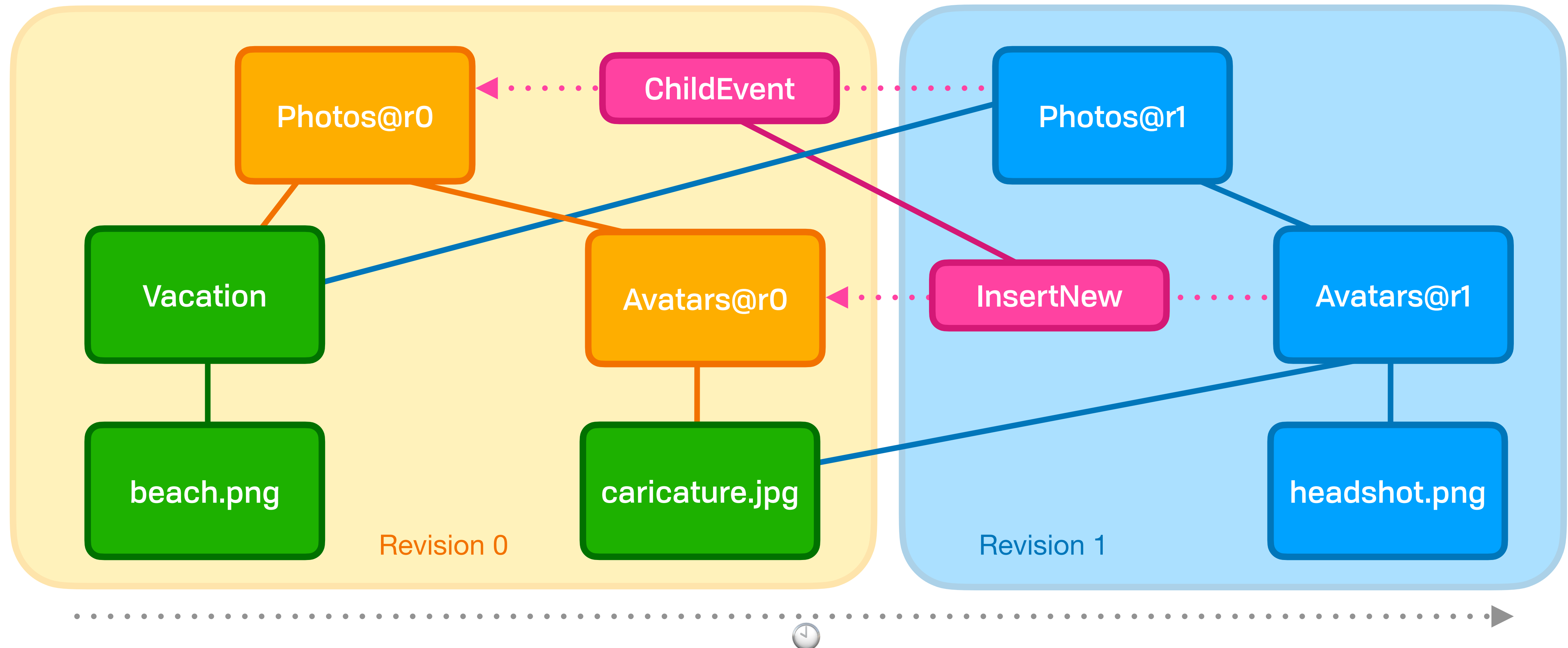
🔧 The z-dimension: versioning & events



File System

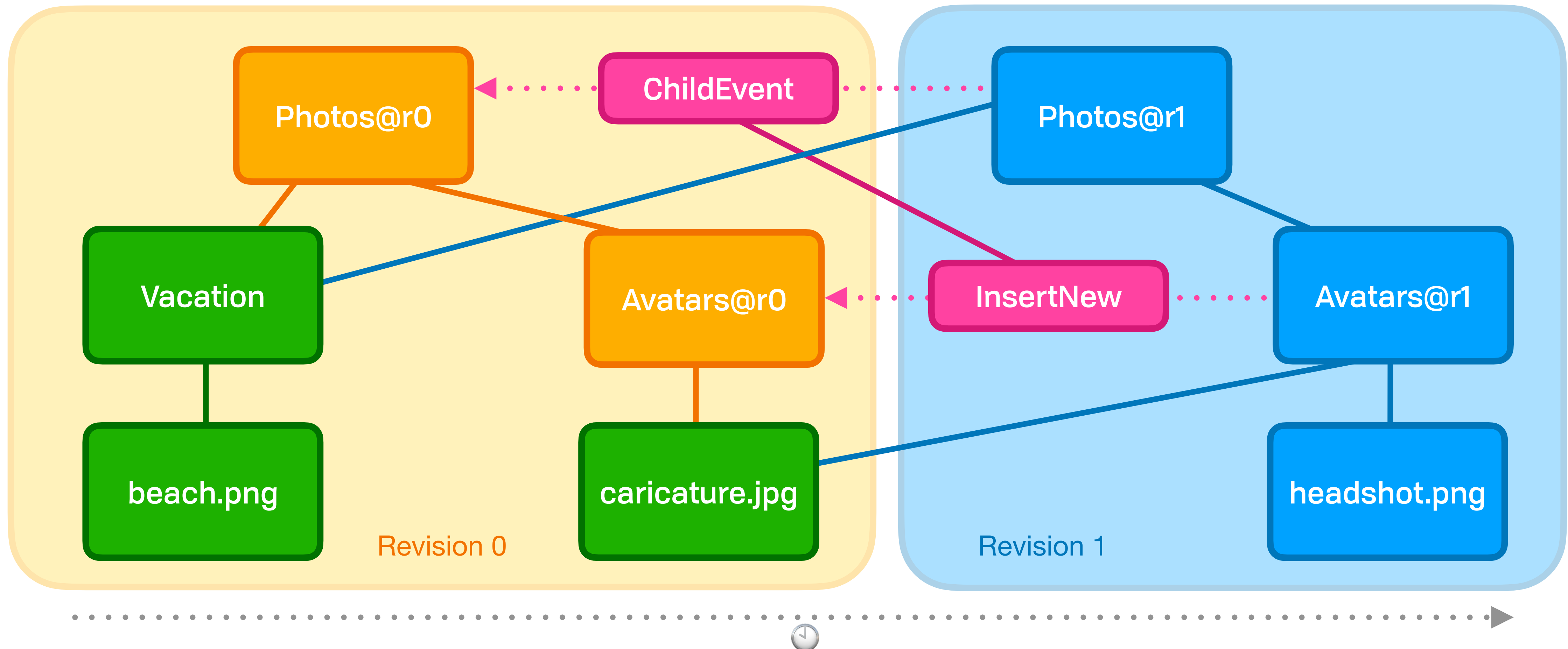
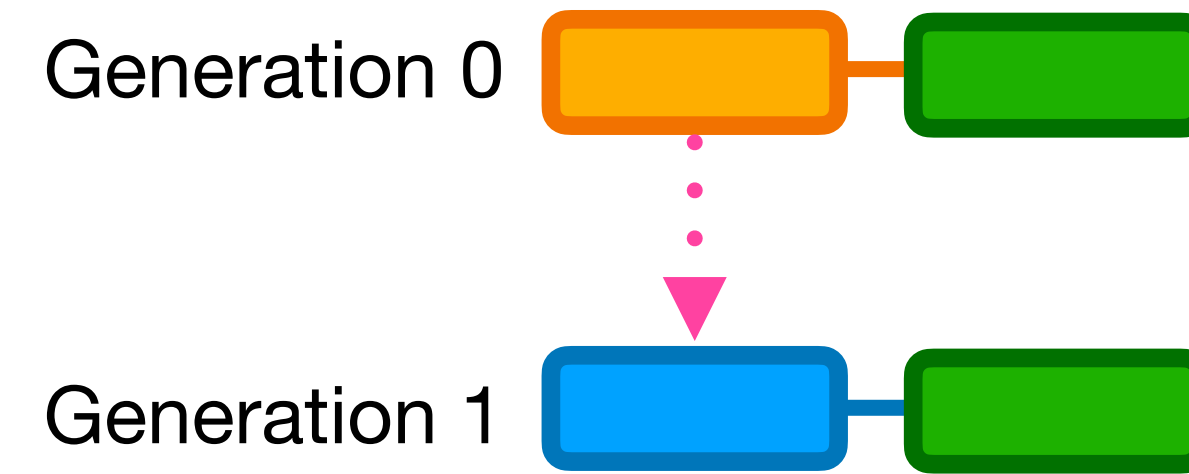
🔧 The z-dimension: versioning & events

Generation 0  



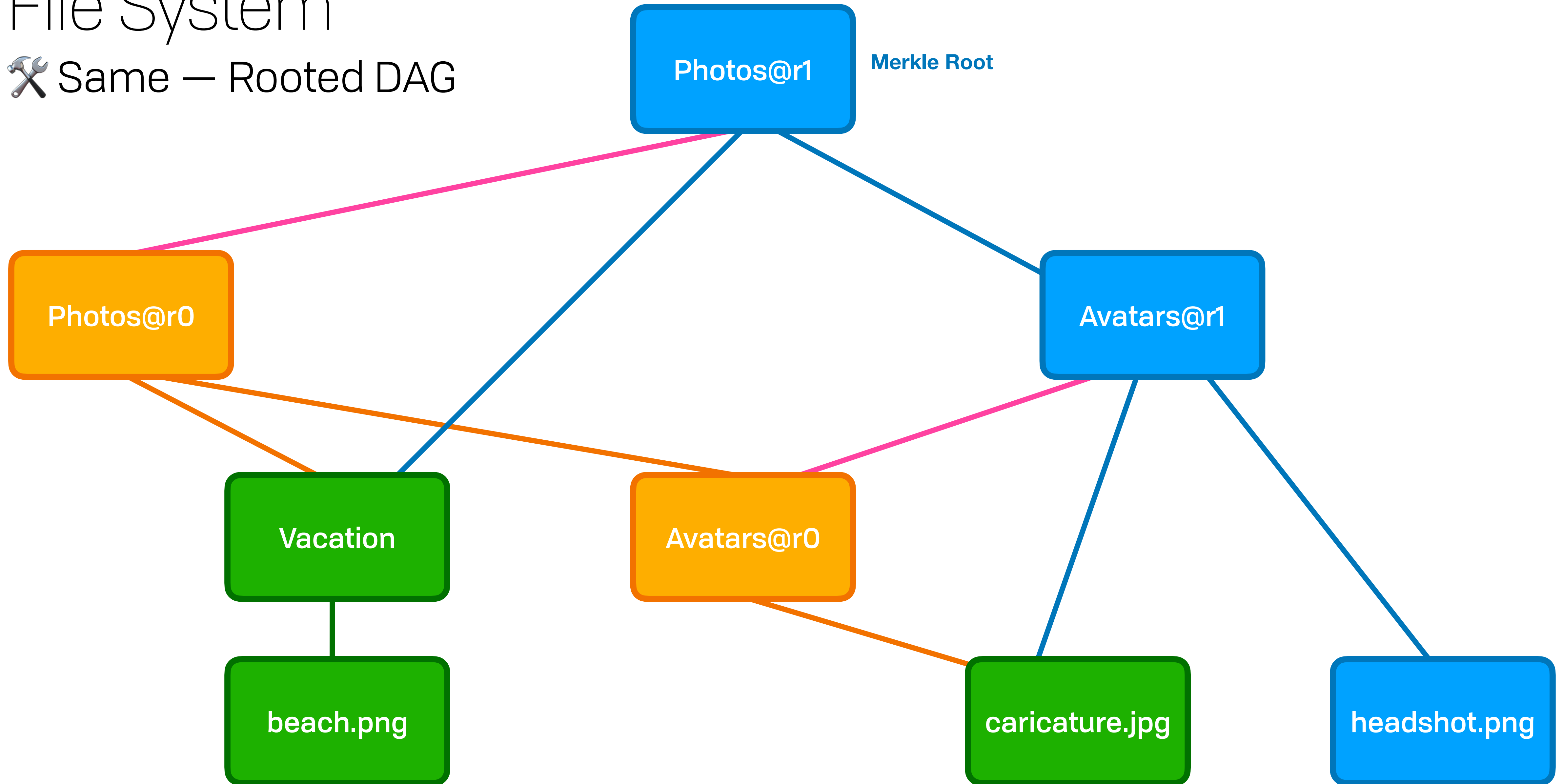
File System

🔧 The z-dimension: versioning & events



File System

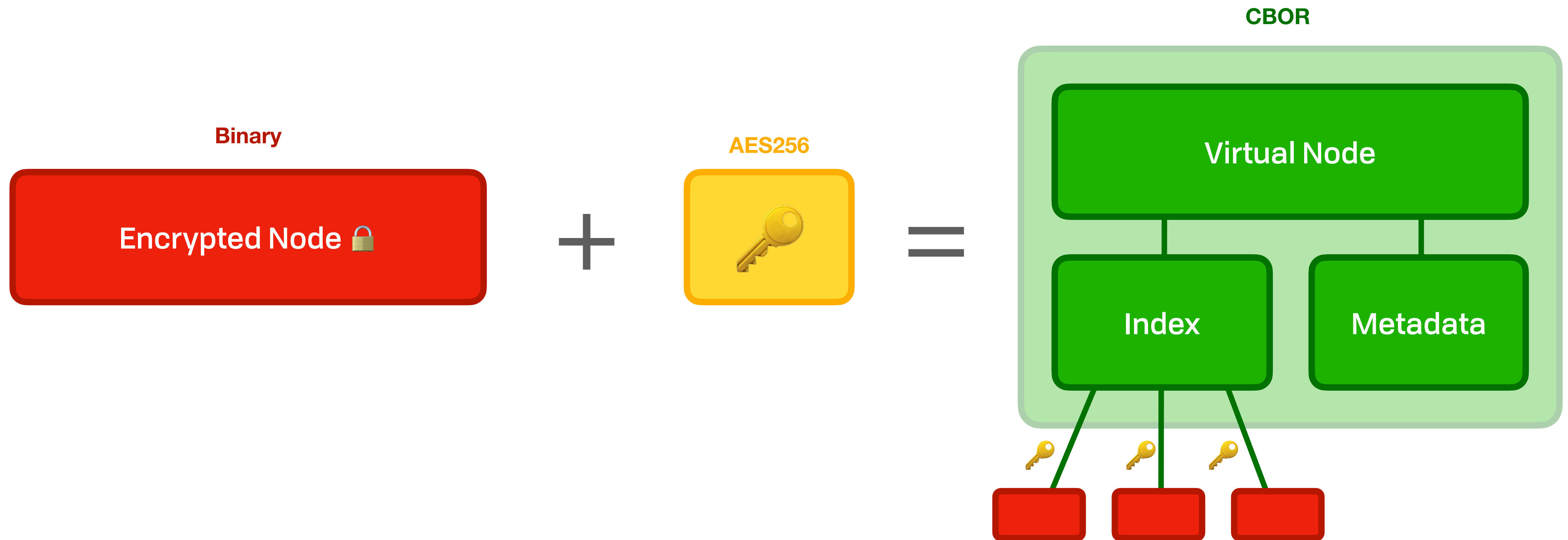
🔧 Same — Rooted DAG



Private Nodes 🤫

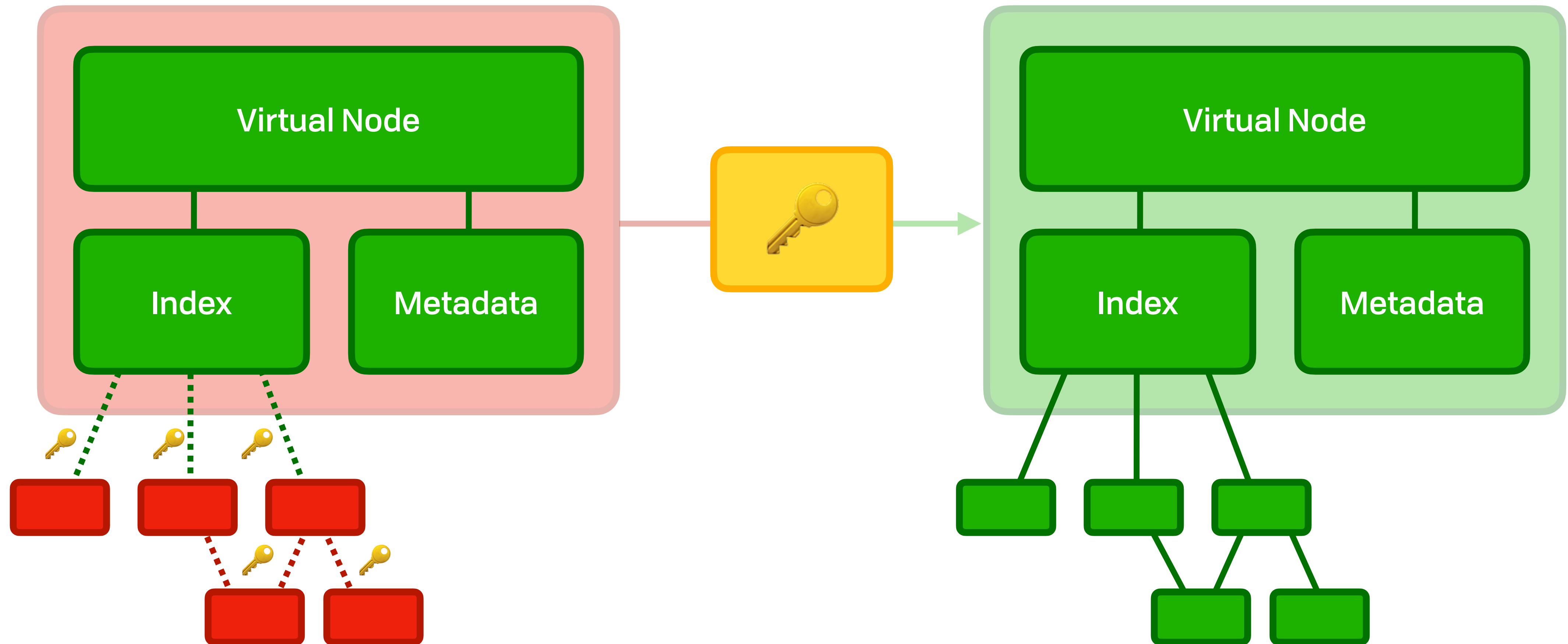
Private Nodes

🔧 Components



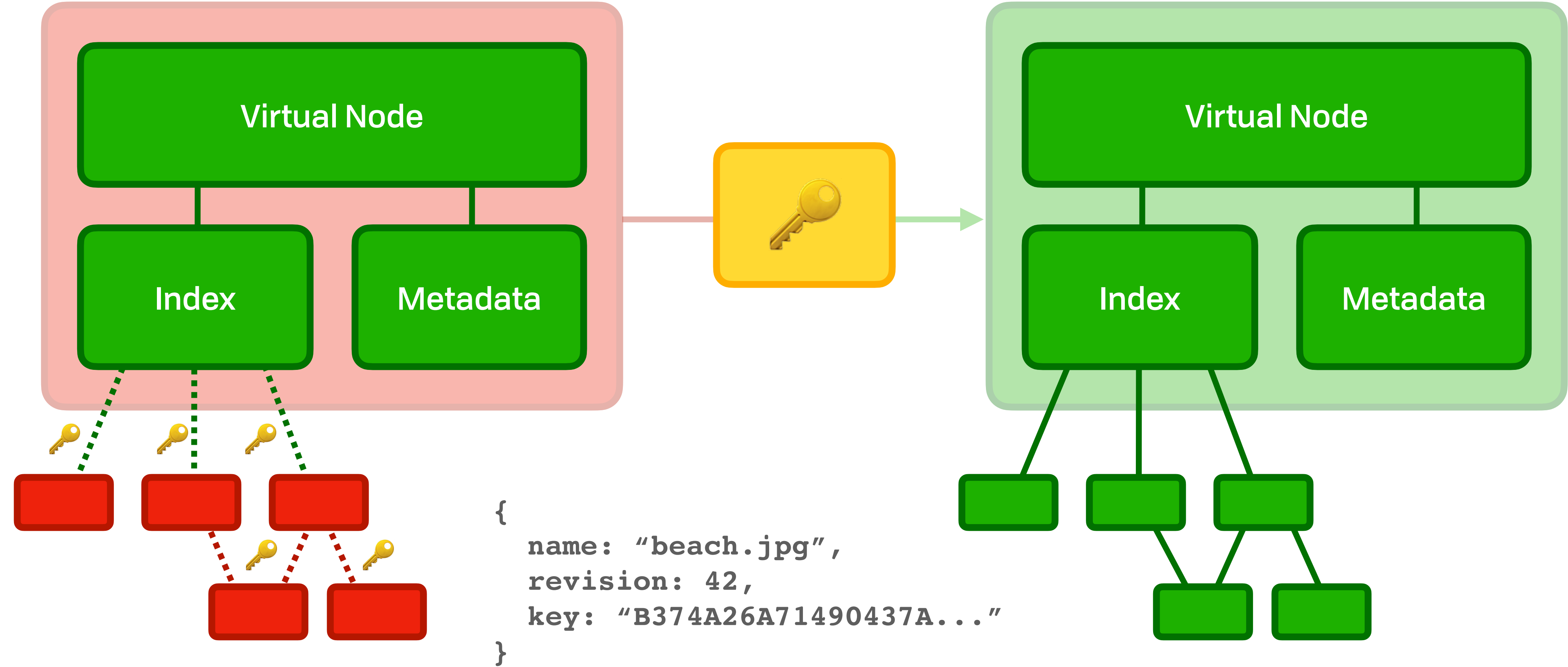
Private Nodes

🔧 Cryptree



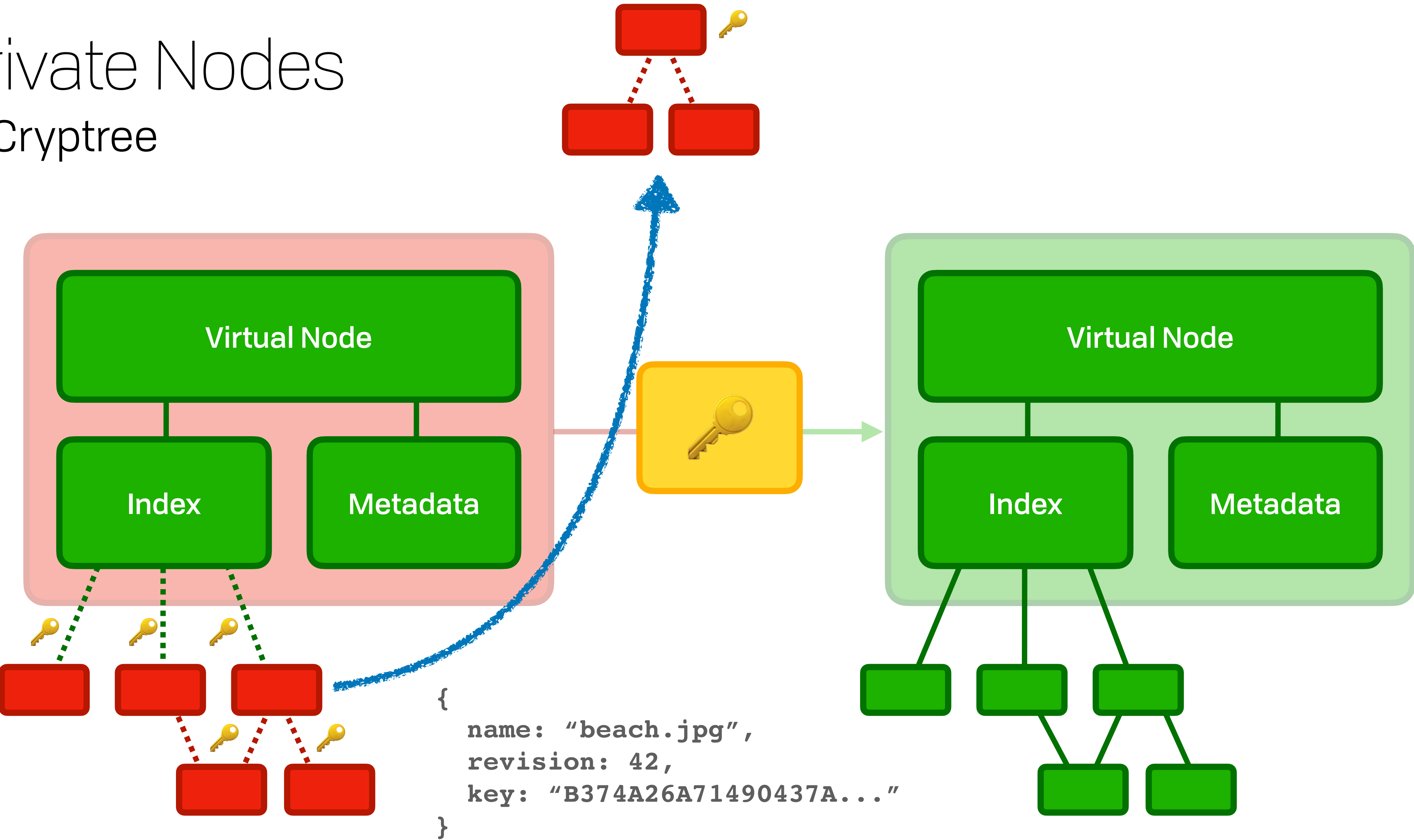
Private Nodes

🔧 Cryptree



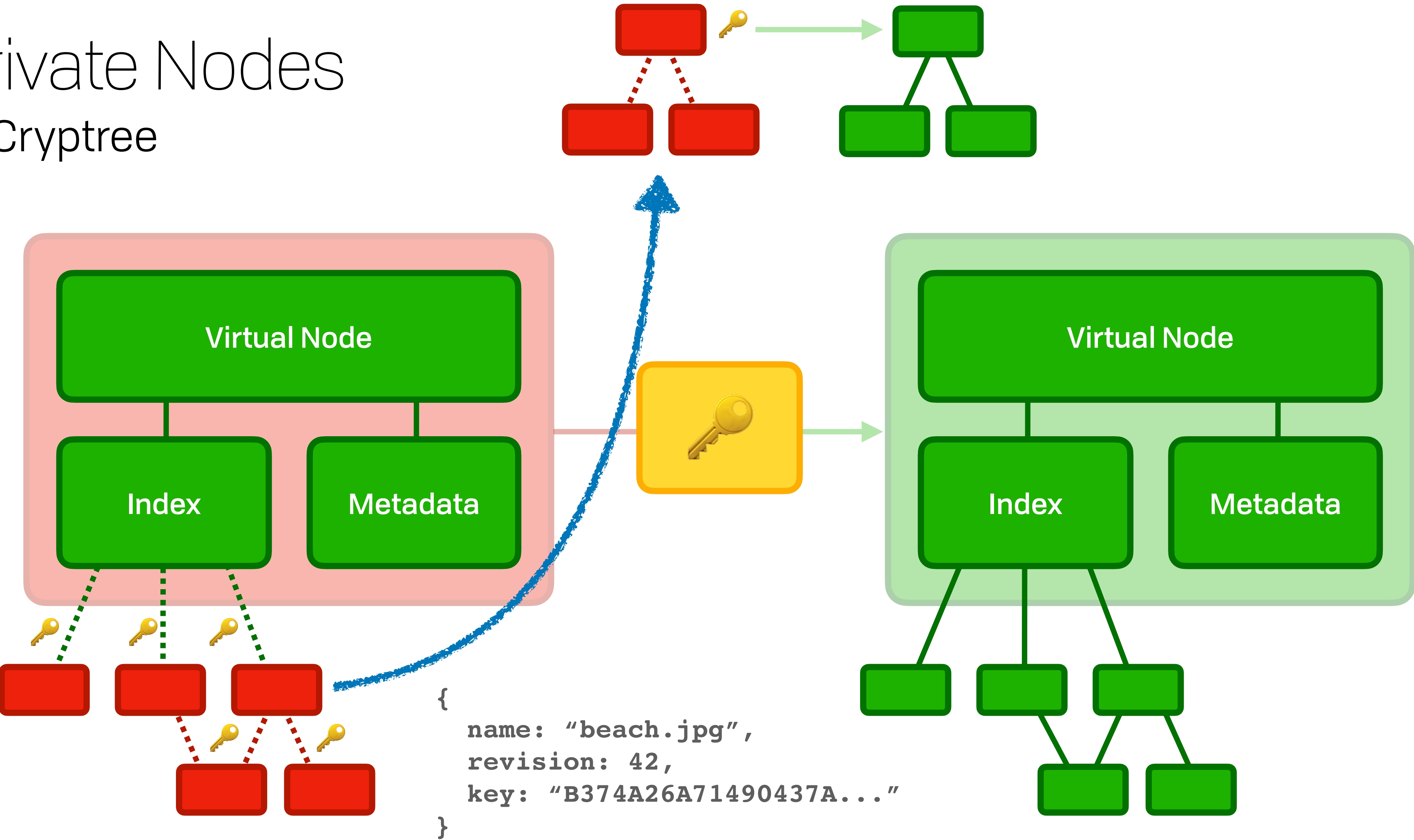
Private Nodes

🔧 Cryptree



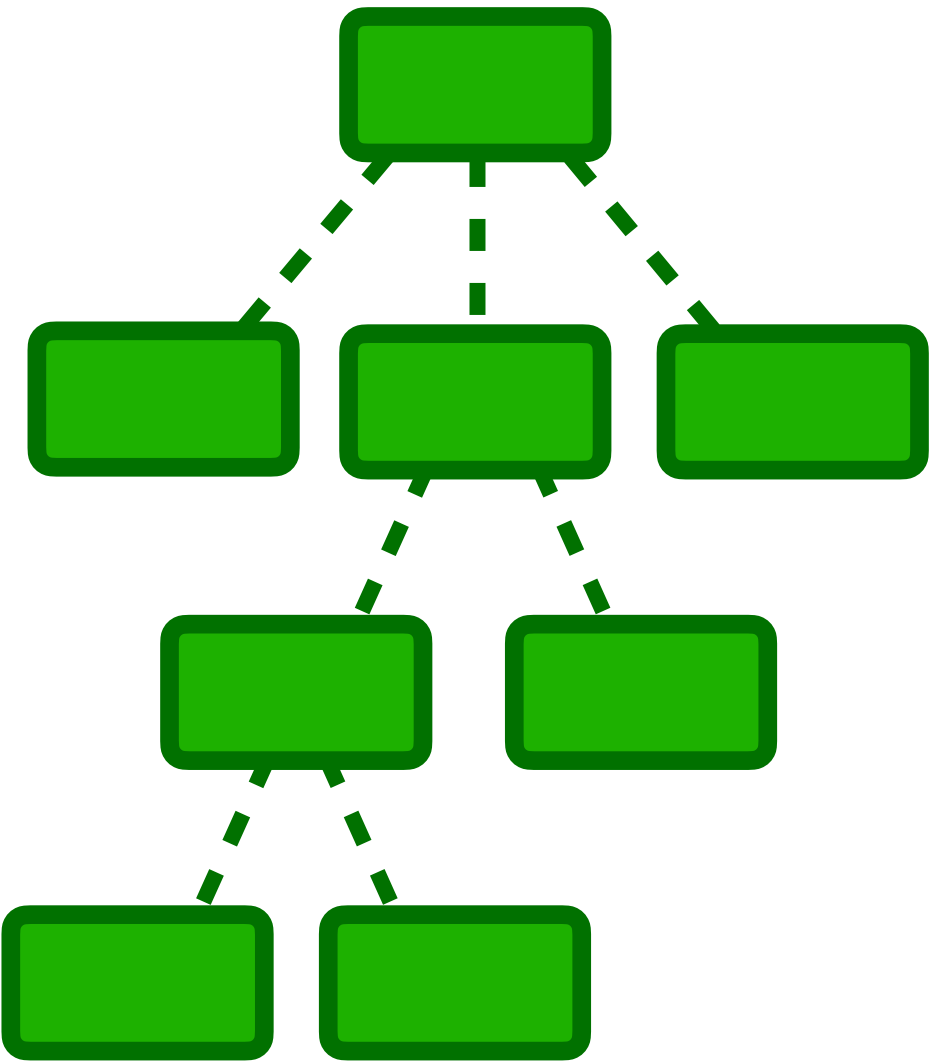
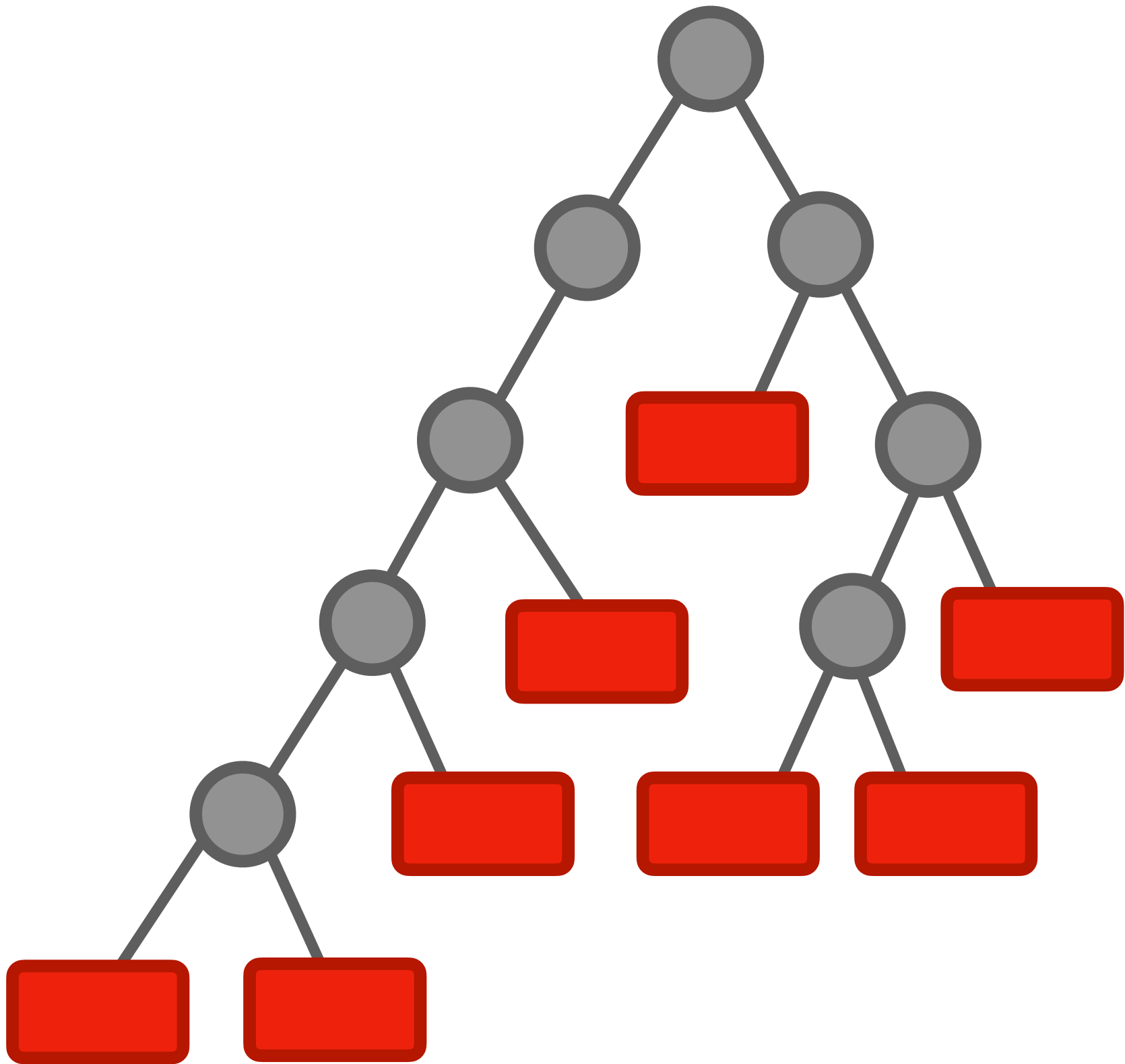
Private Nodes

🔧 Cryptree



Private Nodes

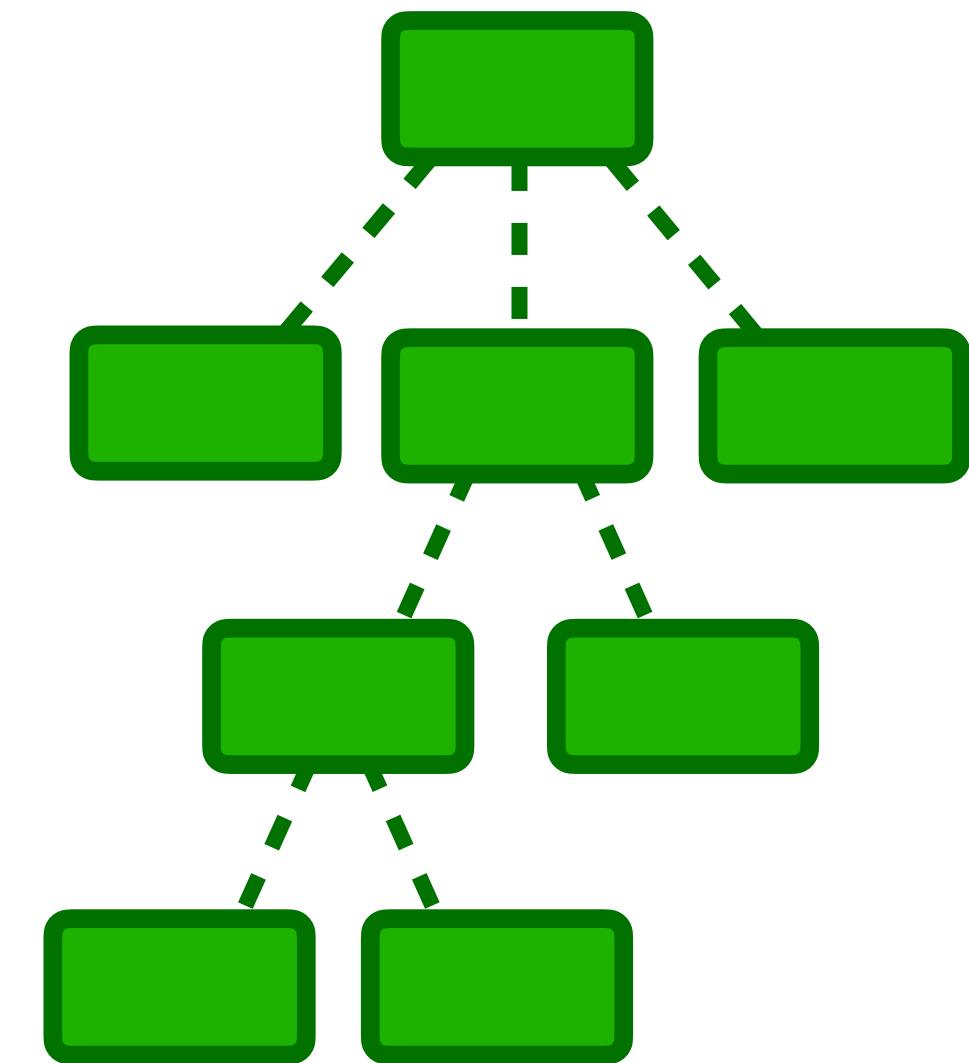
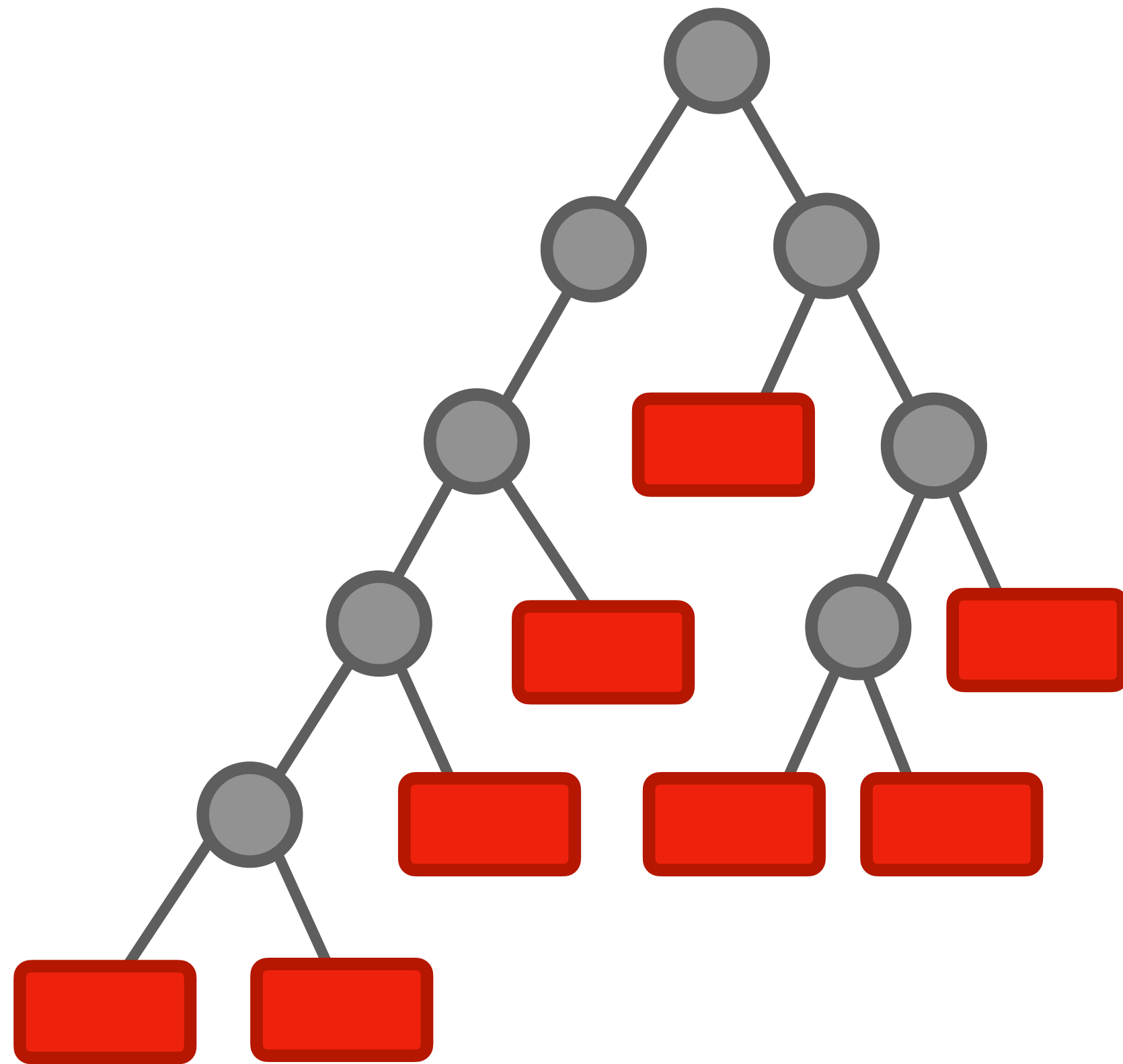
 Private Data Store



Private Nodes

 Private Data Store

MMPT $16^3 = 4,096$ items
(weight 16) $16^4 = 65,536$ items

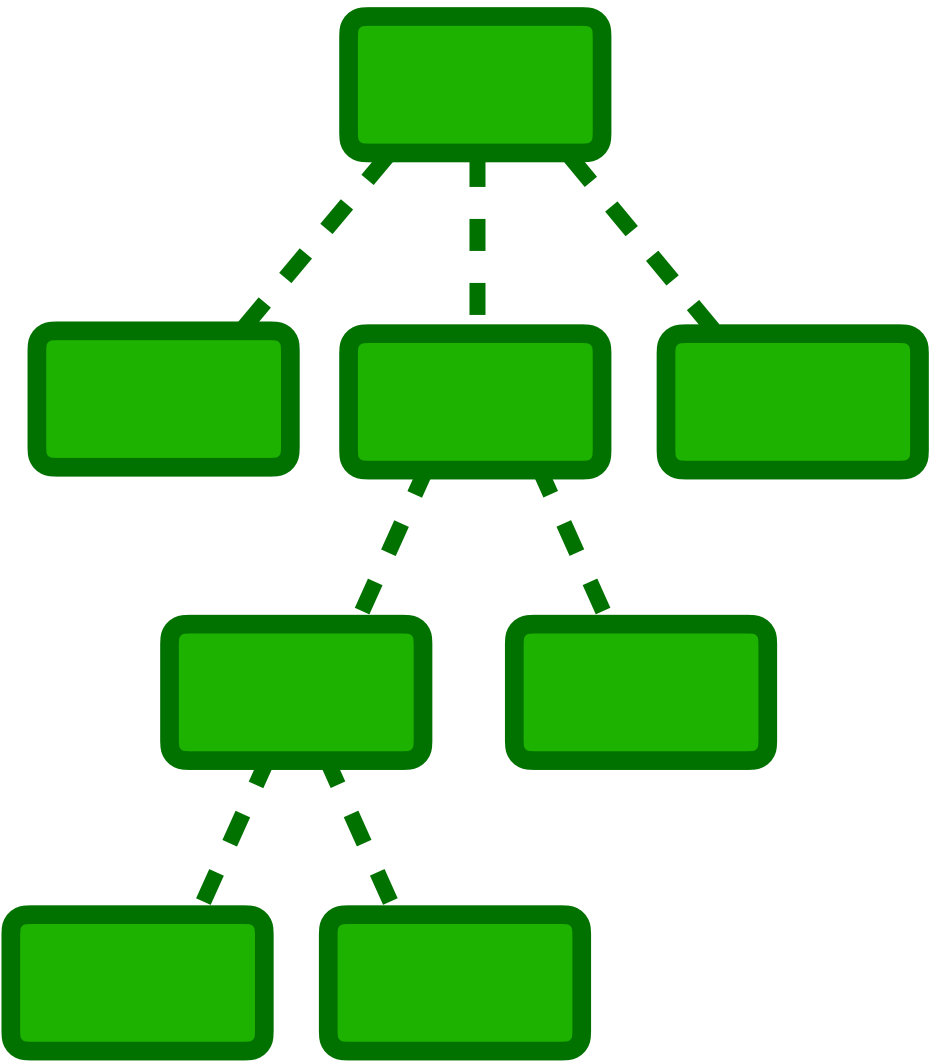
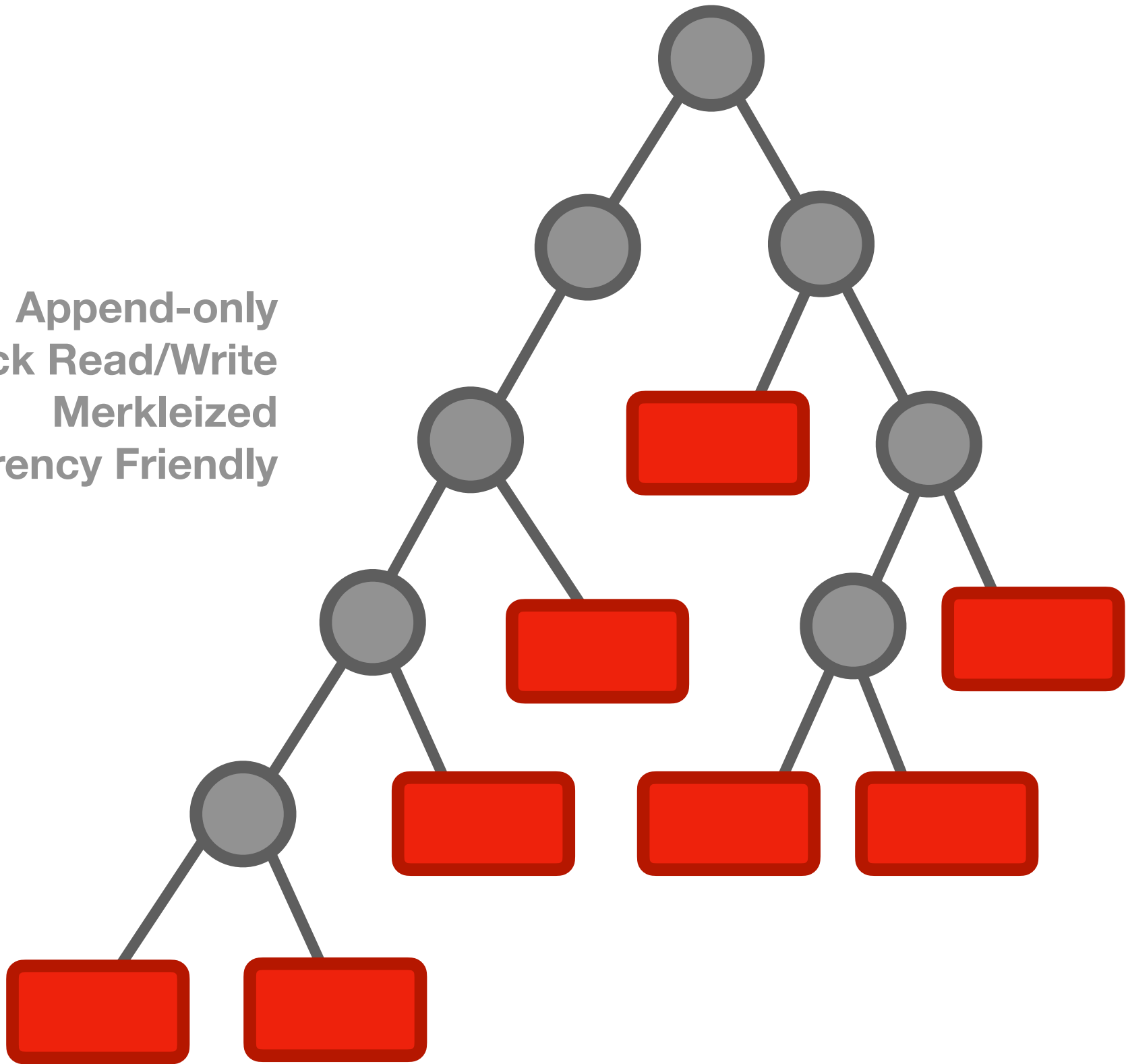


Private Nodes

 Private Data Store

MMPT $16^3 = 4,096$ items
(weight 16) $16^4 = 65,536$ items

Append-only
Quick Read/Write
Merkleized
Concurrency Friendly

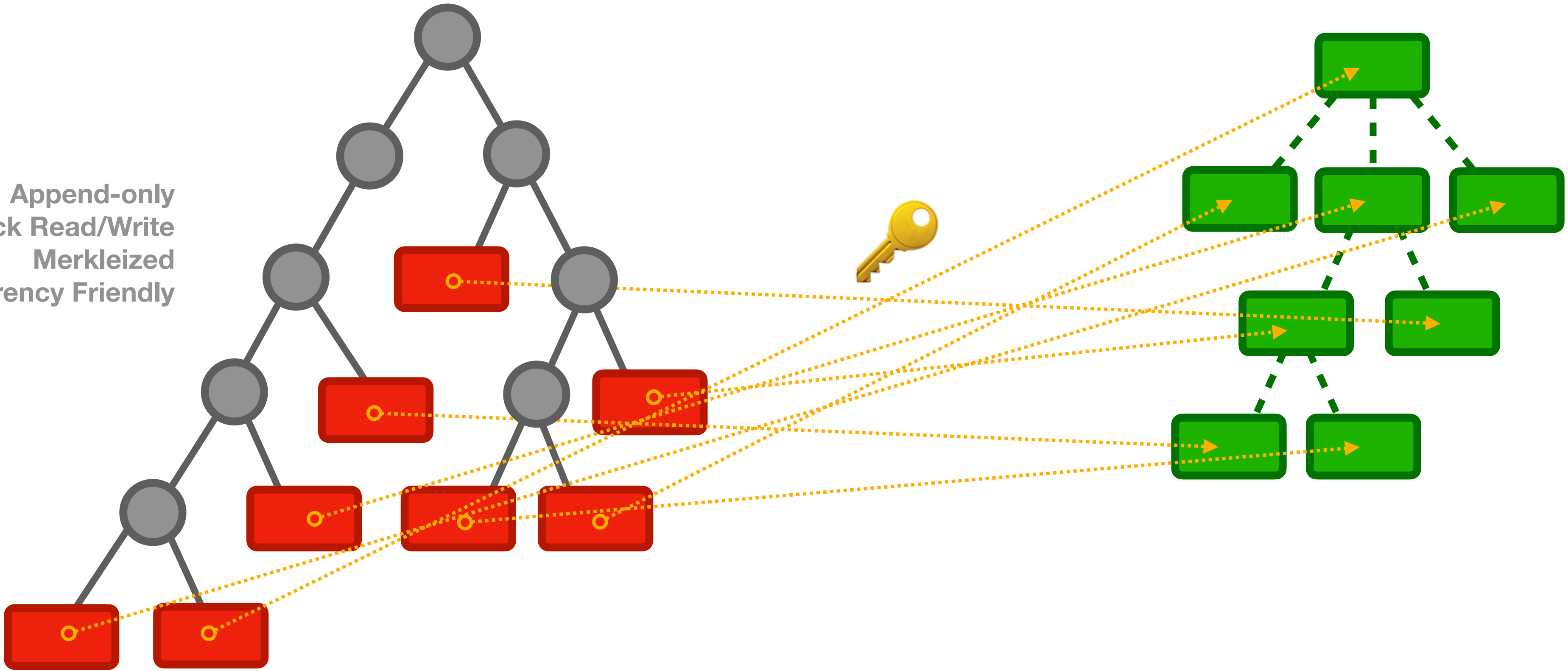


Private Nodes

 Private Data Store

MMPT $16^3 = 4,096$ items
(weight 16) $16^4 = 65,536$ items

Append-only
Quick Read/Write
Merkleized
Concurrency Friendly



Private Nodes

Namefilters

Private Nodes

Namefilters

- Constraints:
 - Deterministic
 - Versioned
 - Addressable
 - Prove subpath for UCAN
 - Minimal knowledge
- AES keys ~ path segments *but secret*

Private Nodes

Namefilters

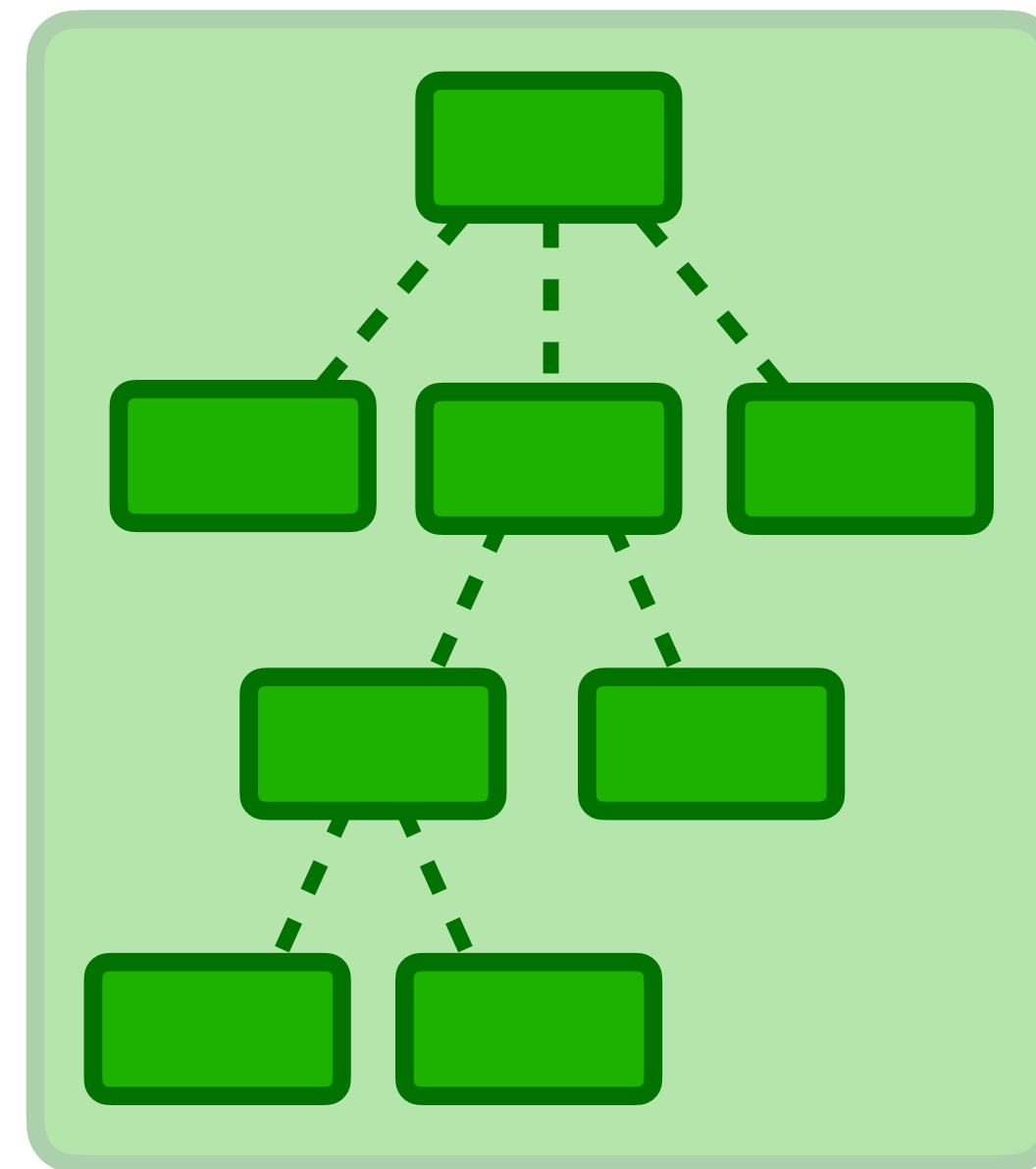
- Constraints:
 - Deterministic
 - Versioned
 - Addressable
 - Prove subpath for UCAN
 - Minimal knowledge
 - AES keys ~ path segments *but secret*
- bareFilter
 - **parentFilter**
 - **AND bloom(SHA(aesKey))**
 - **AND bloom(SHA(aesKey ++ revision))**
 - Saturation
 - **nameFilter AND bloom(SHA(nameFilter))**
 - Repeat until threshold bits flipped

Private Nodes

 Writing as Much as UCAN

Private Nodes

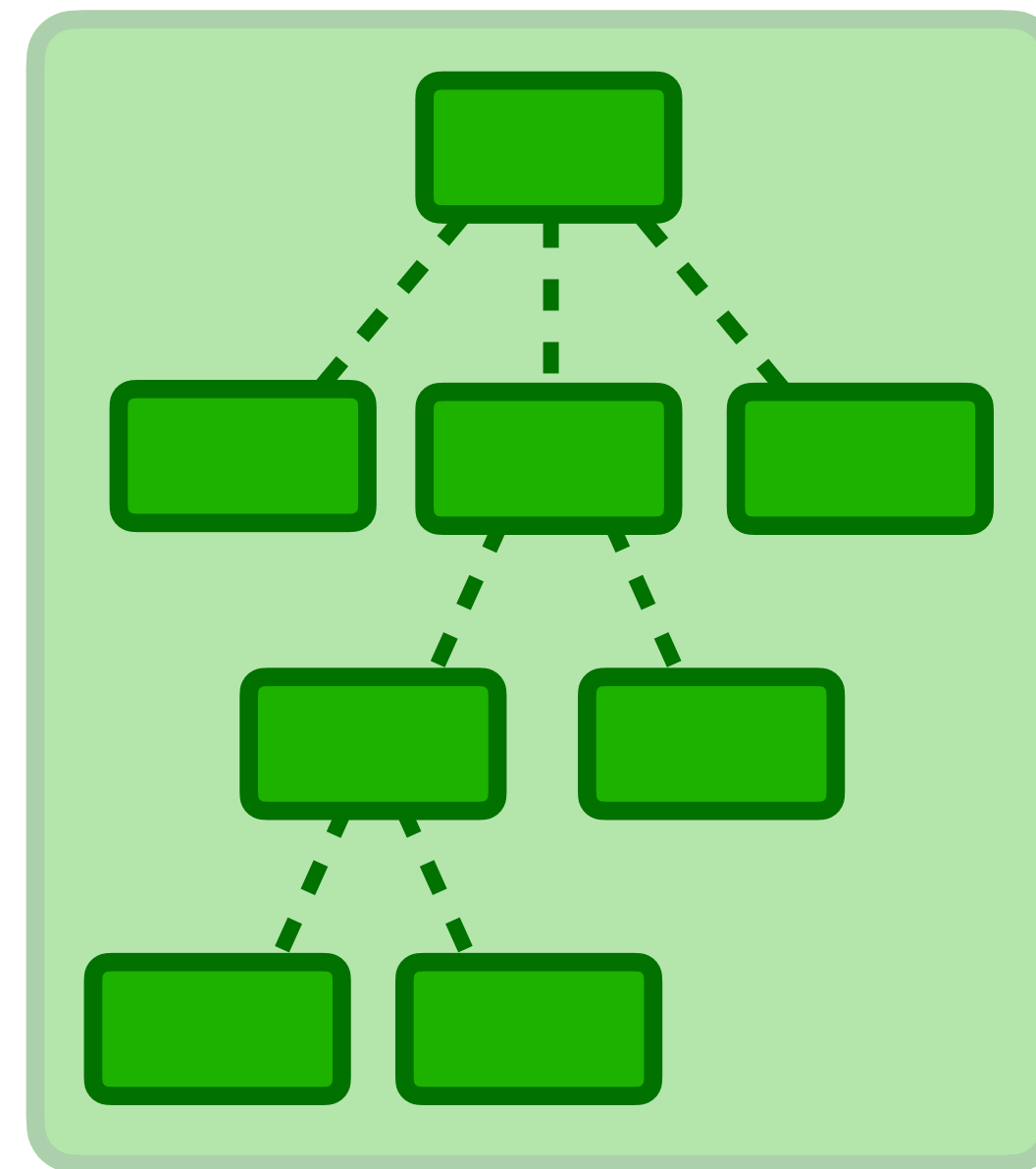
🔧 Writing as Much as UCAN



Rev 0

Private Nodes

🔧 Writing as Much as UCAN

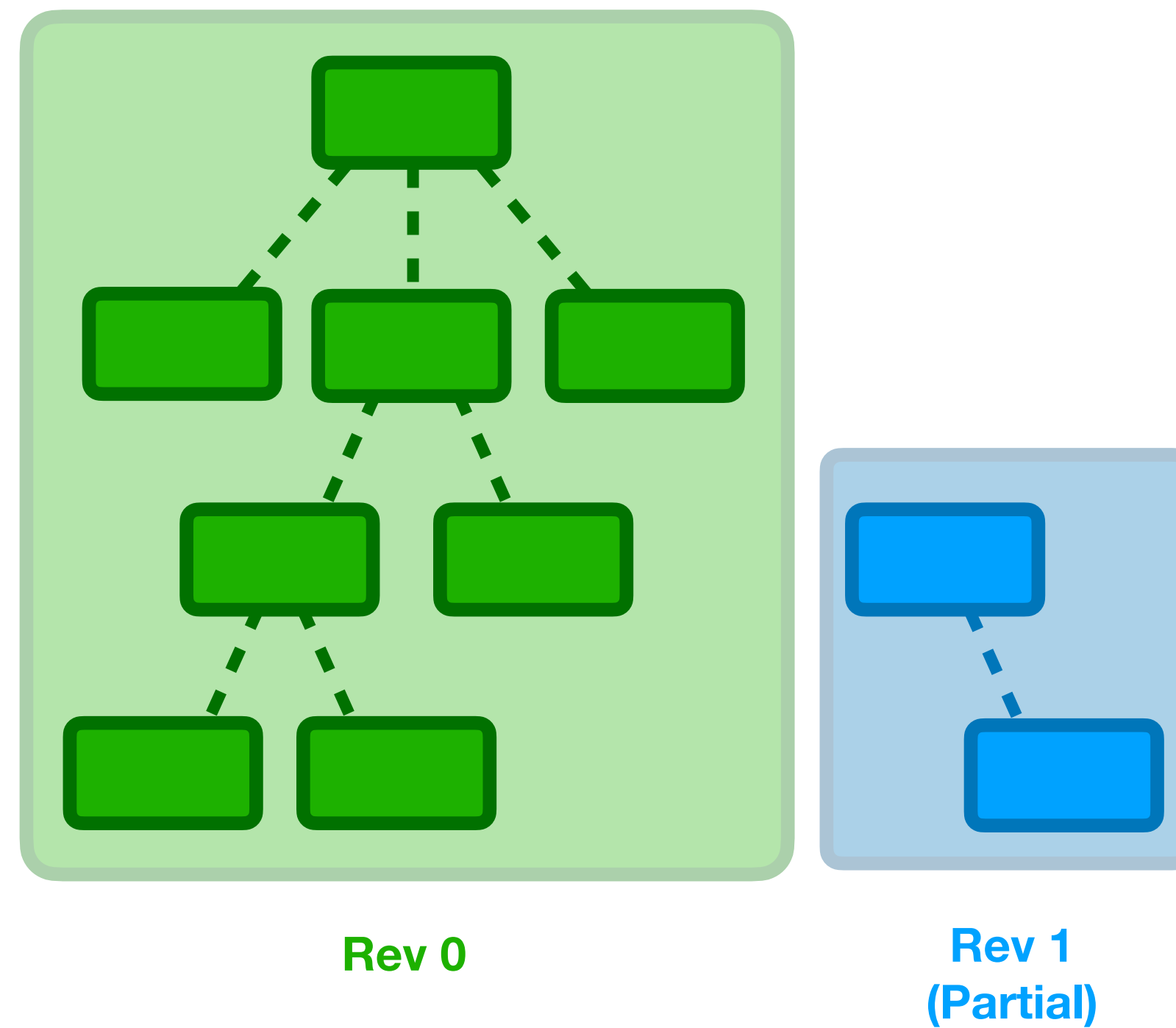


Rev 0



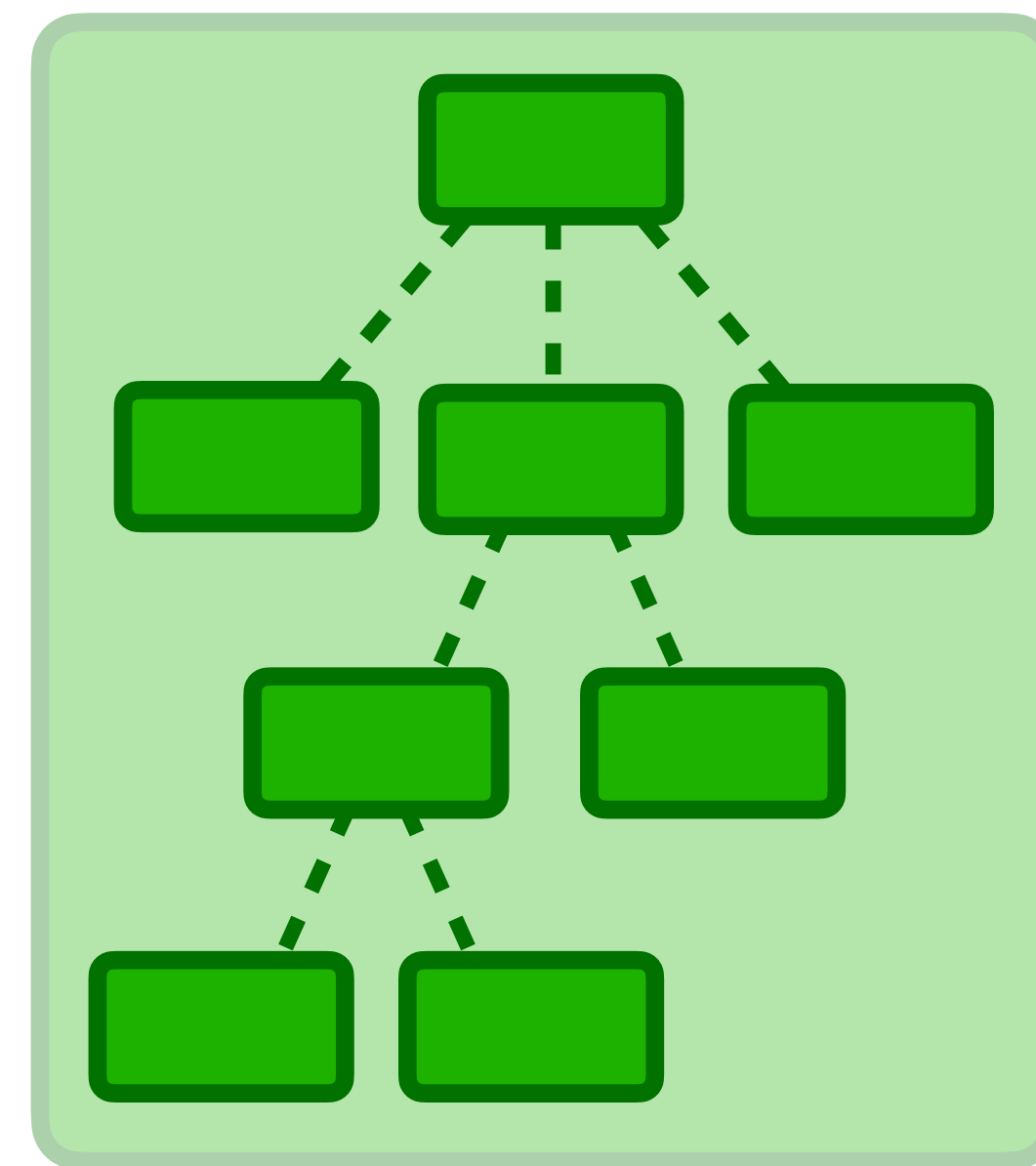
Private Nodes

🔧 Writing as Much as UCAN

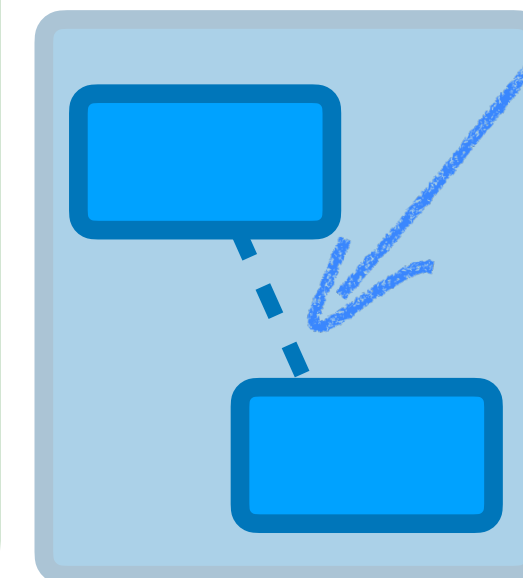


Private Nodes

🔧 Writing as Much as UCAN



Rev 0

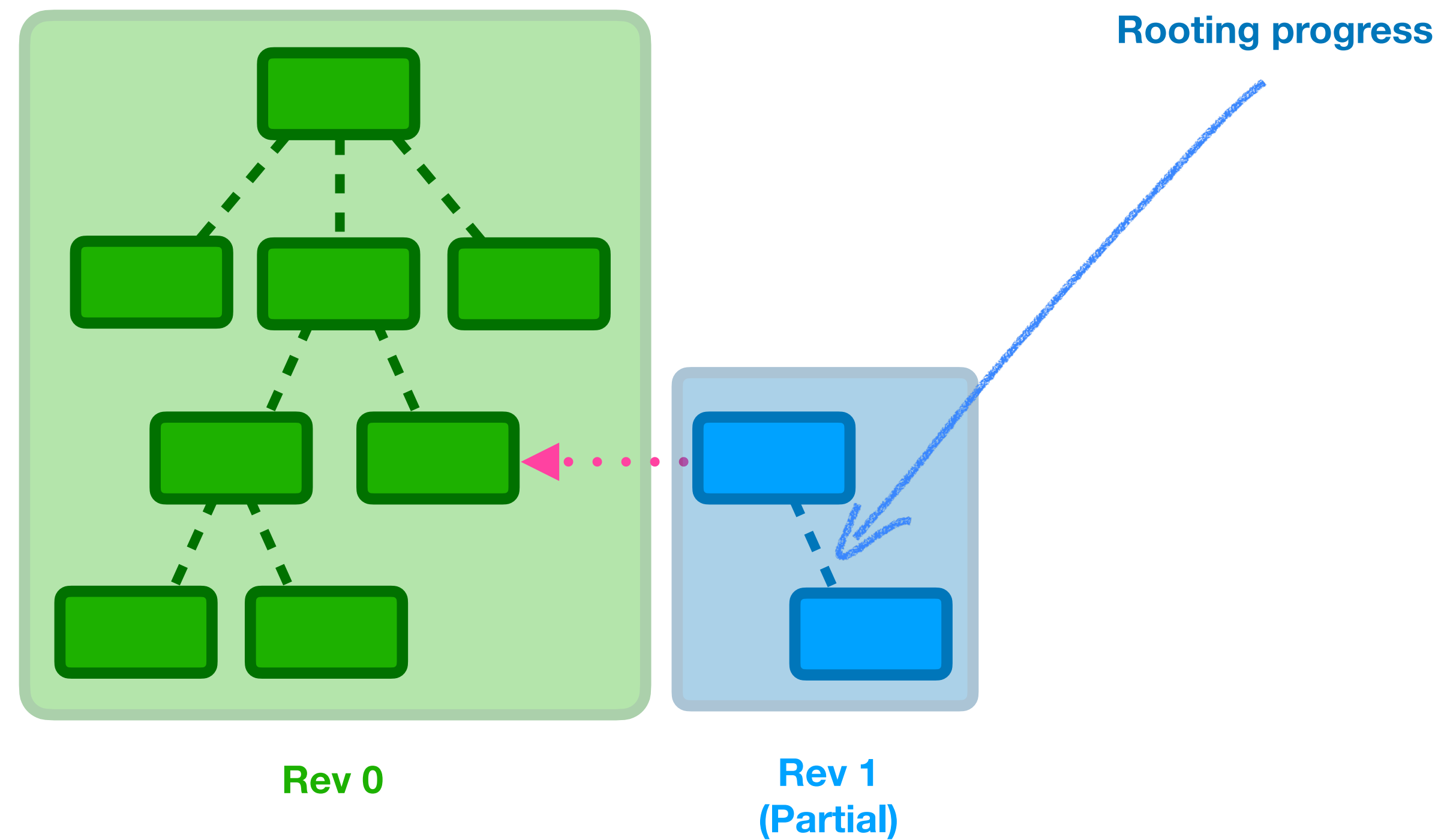


Rev 1
(Partial)

Rooting progress

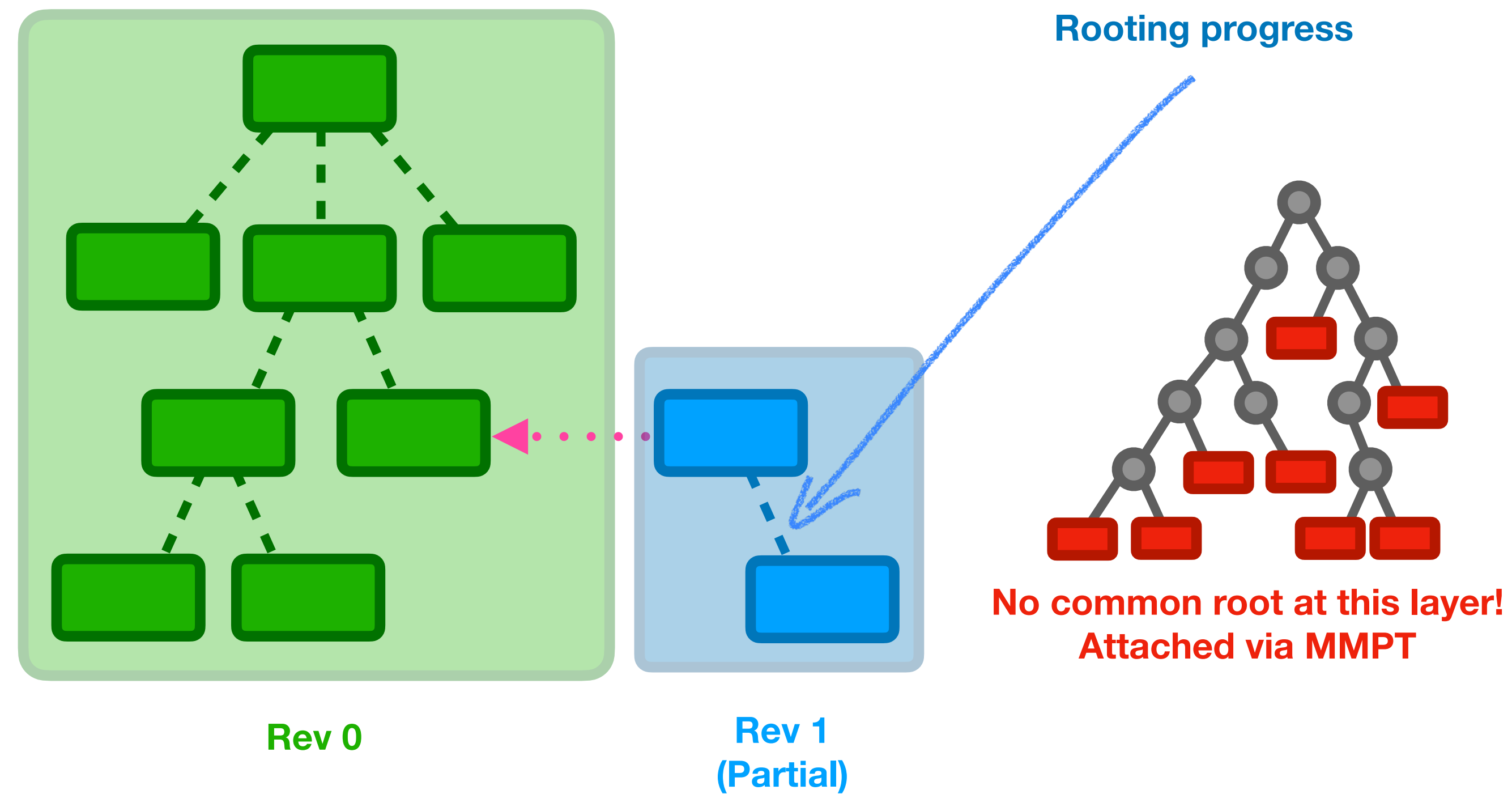
Private Nodes

🔧 Writing as Much as UCAN



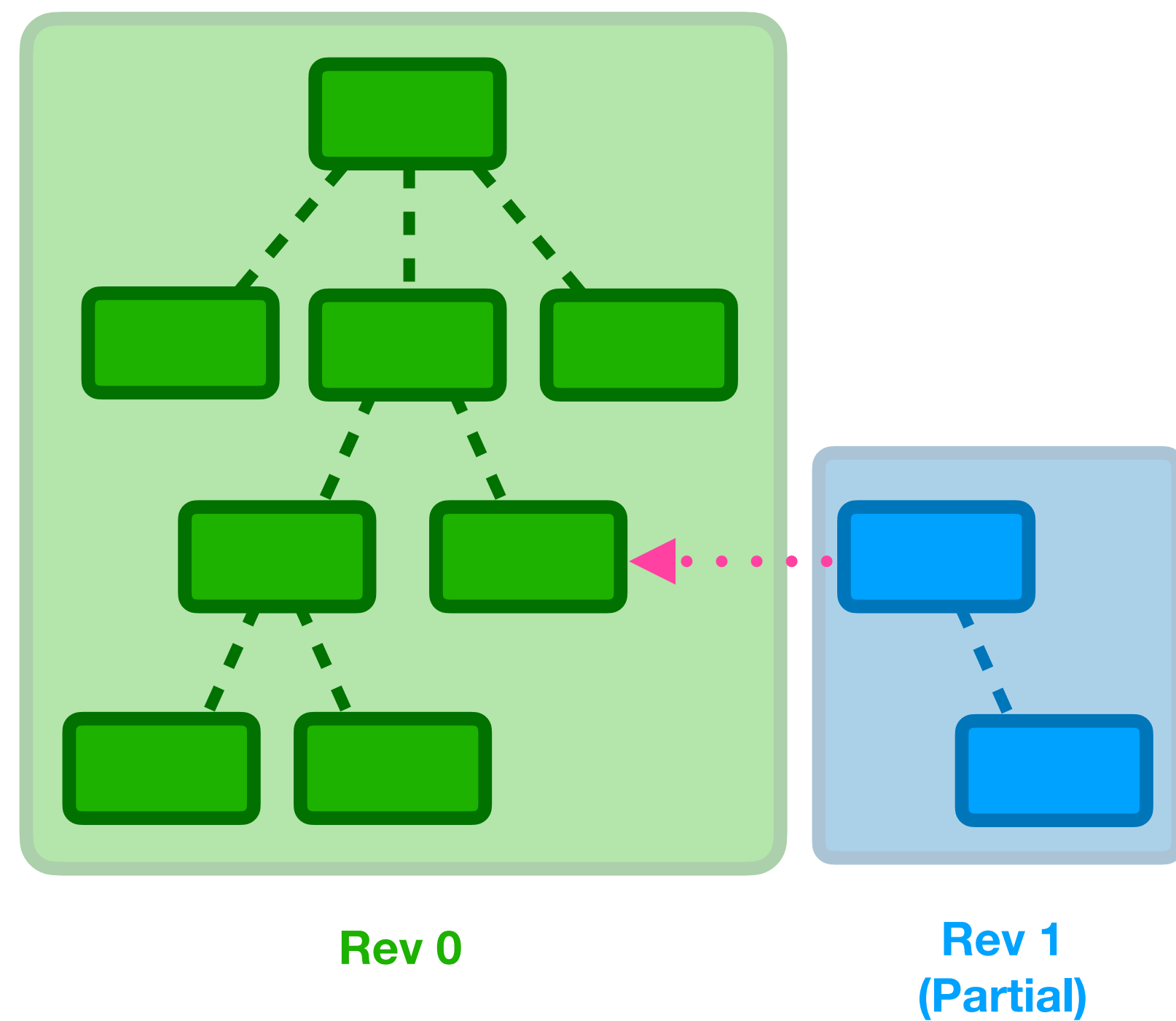
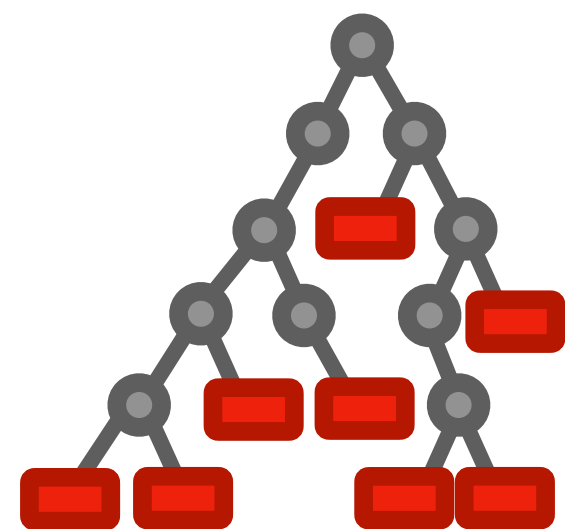
Private Nodes

🔧 Writing as Much as UCAN



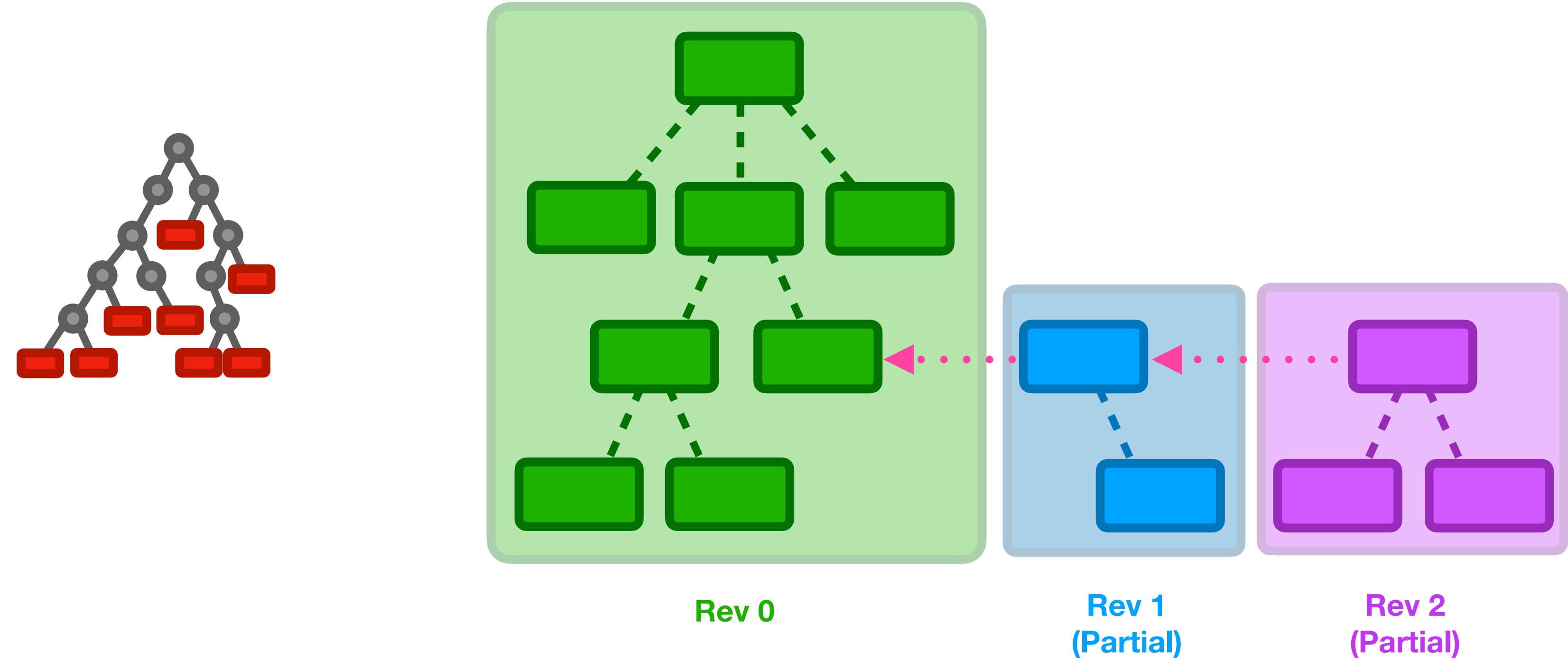
Private Nodes

🔧 Progressive Fast Forward



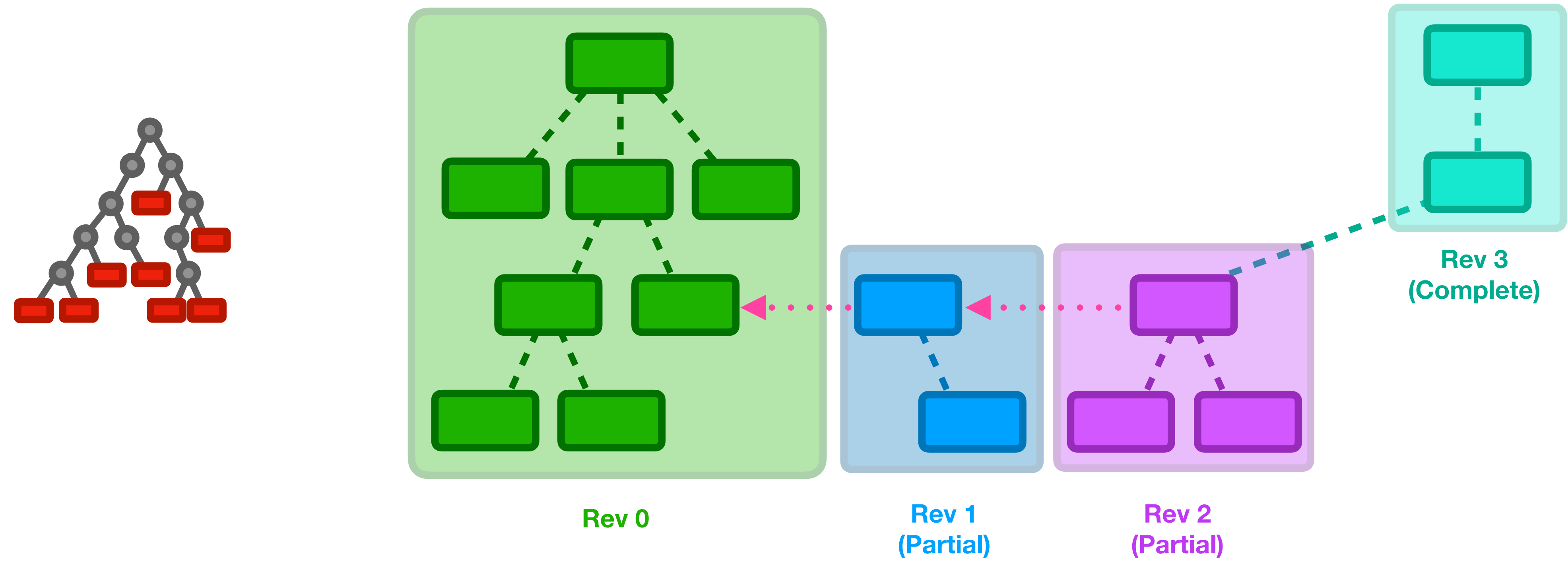
Private Nodes

🔧 Progressive Fast Forward



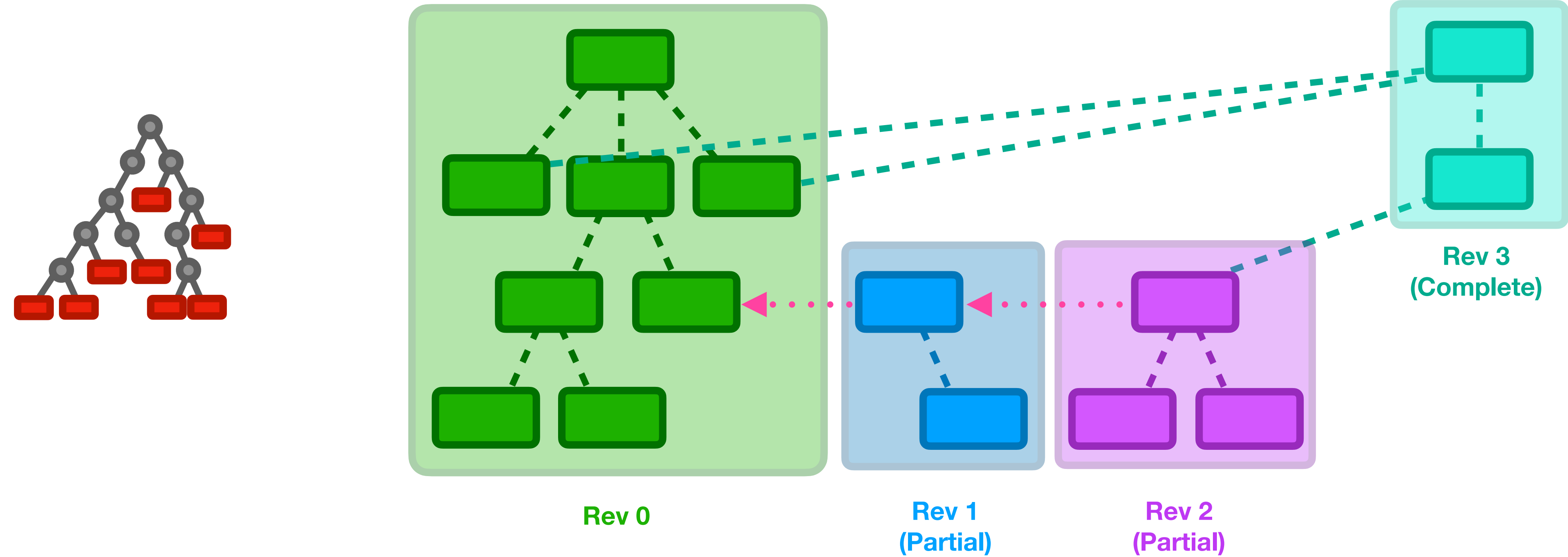
Private Nodes

🔧 Progressive Fast Forward



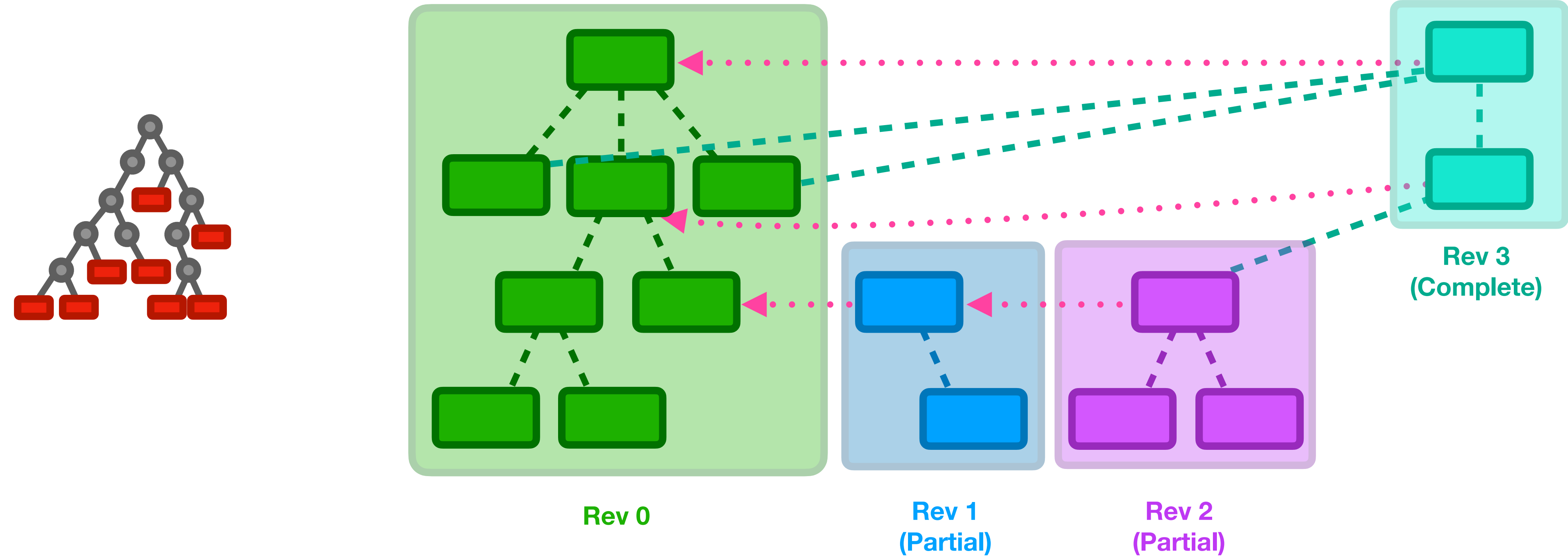
Private Nodes

🔧 Progressive Fast Forward



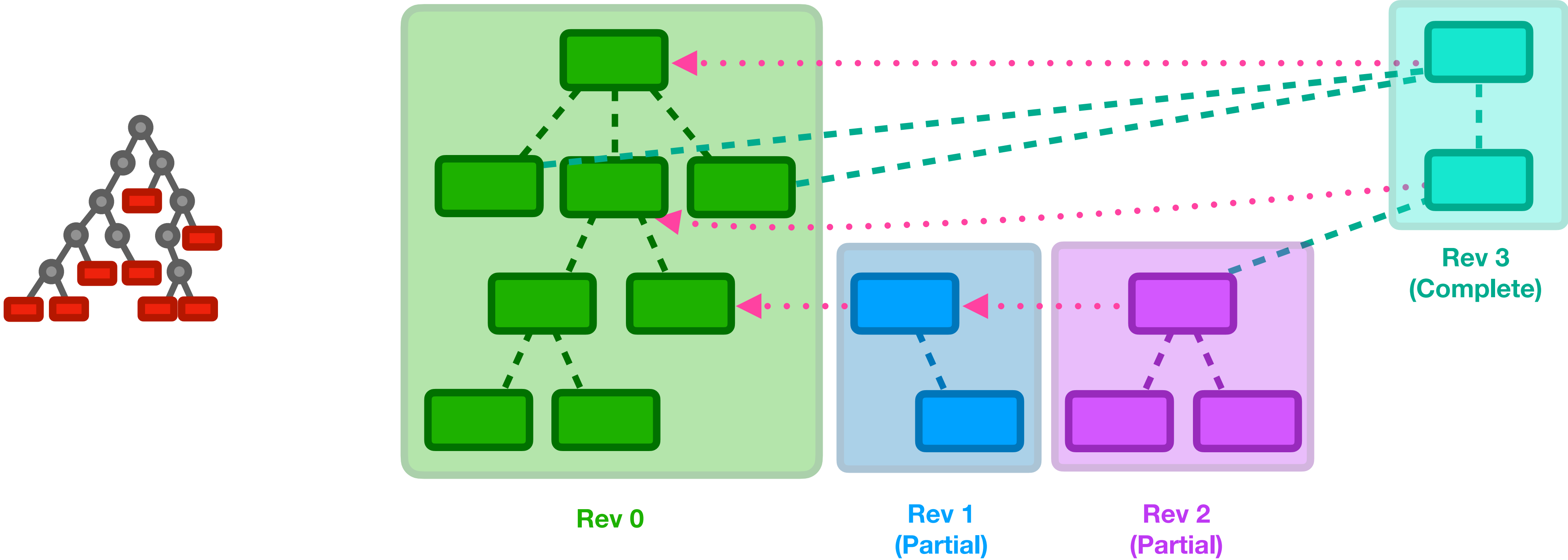
Private Nodes

🔧 Progressive Fast Forward



Private Nodes

🔧 Progressive Fast Forward

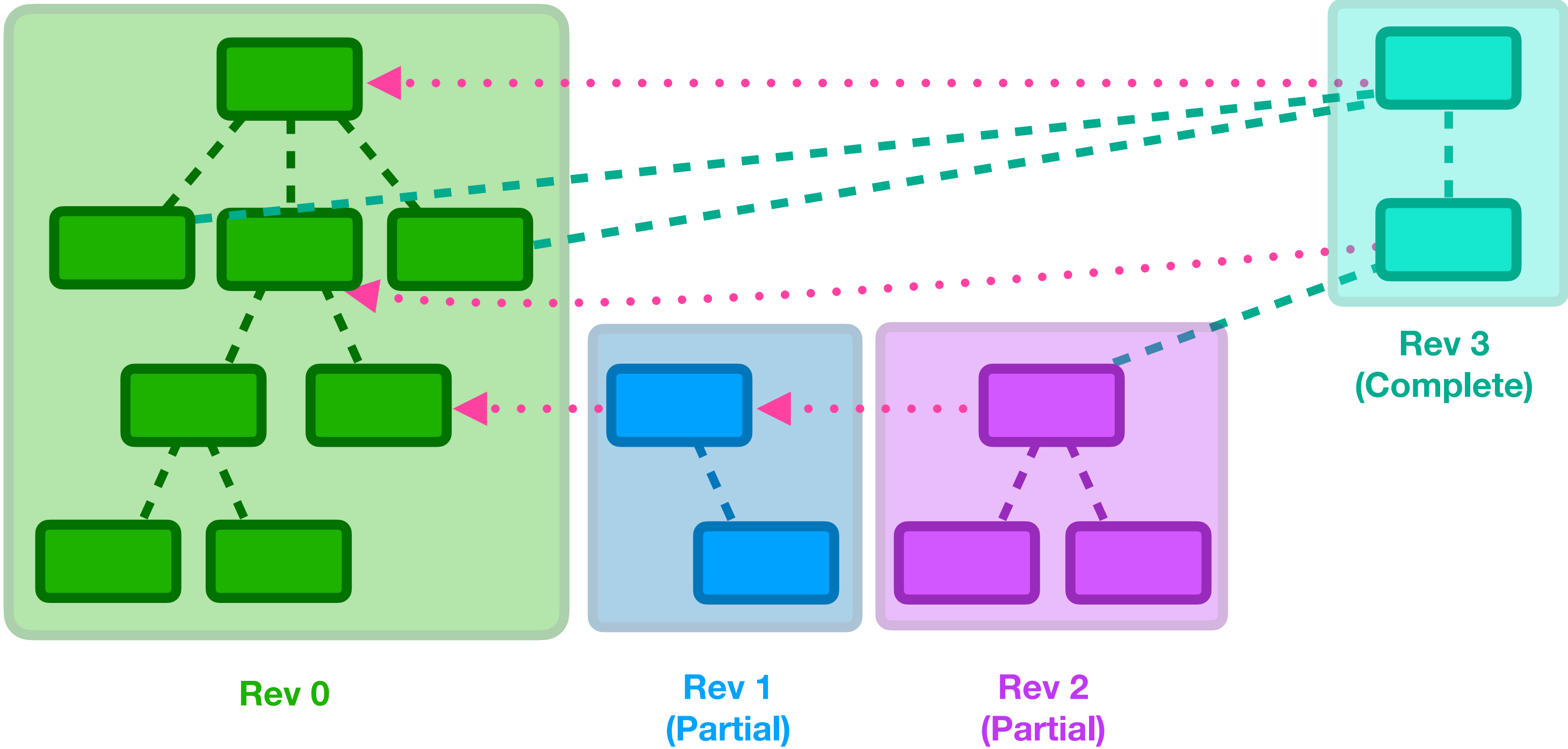
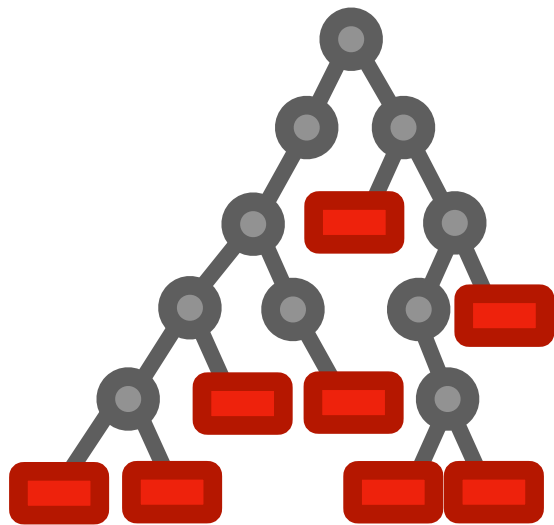


Private Nodes

🔧 Progressive Fast Forward

current rev + 2ⁿ

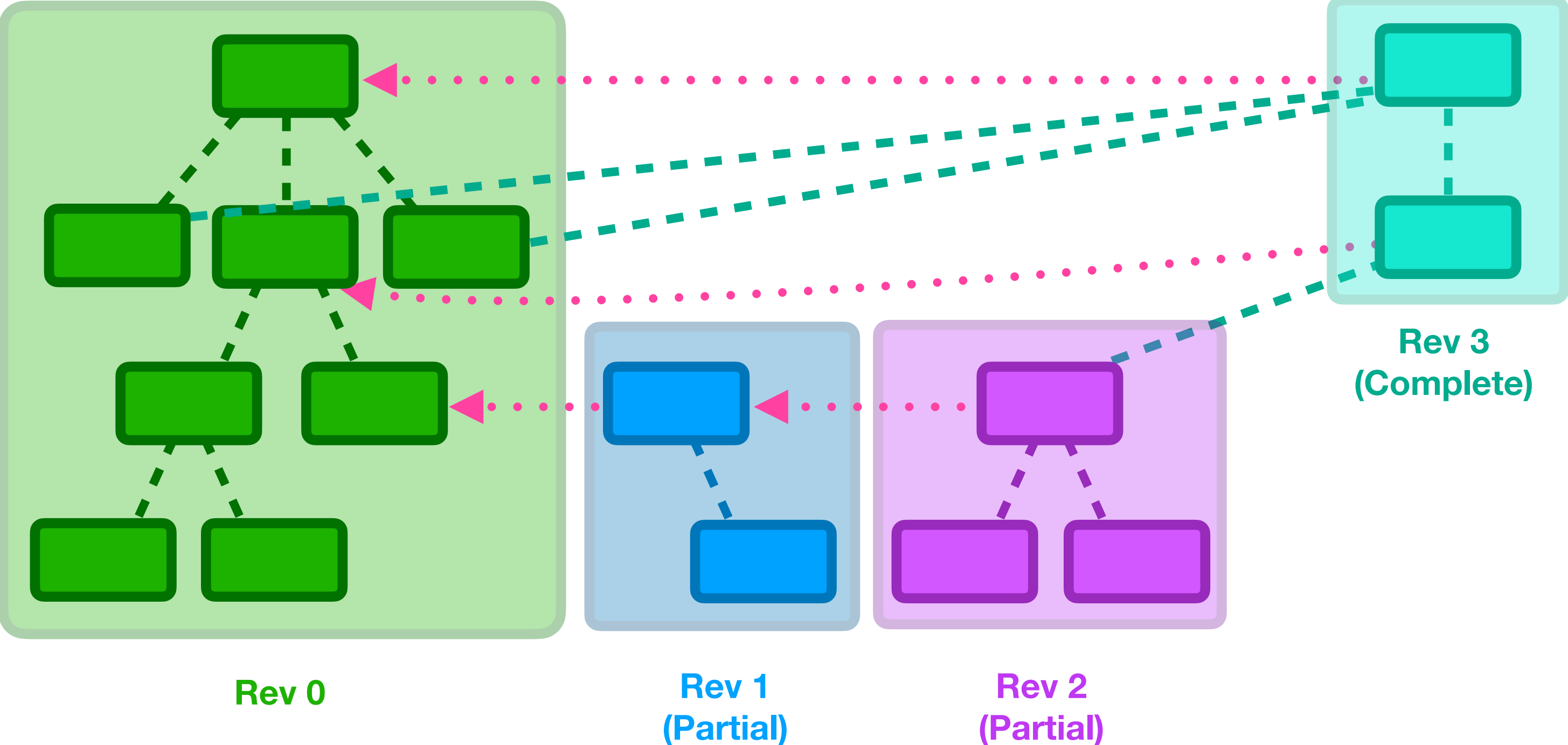
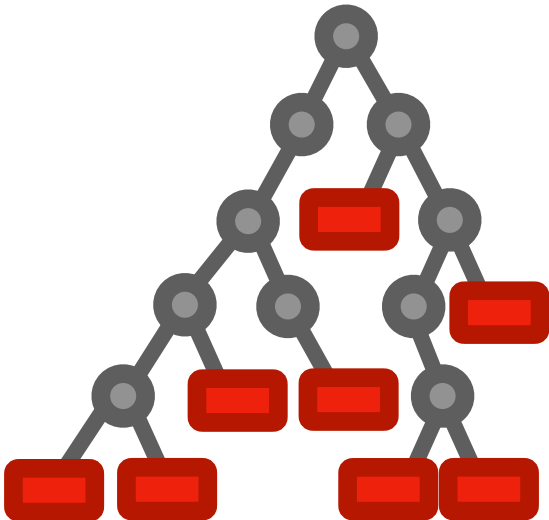
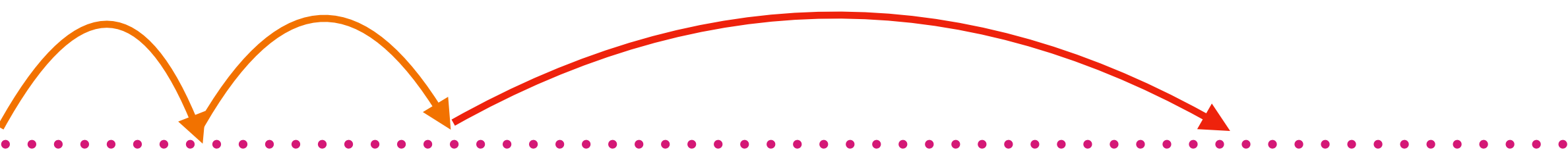
.....



Private Nodes

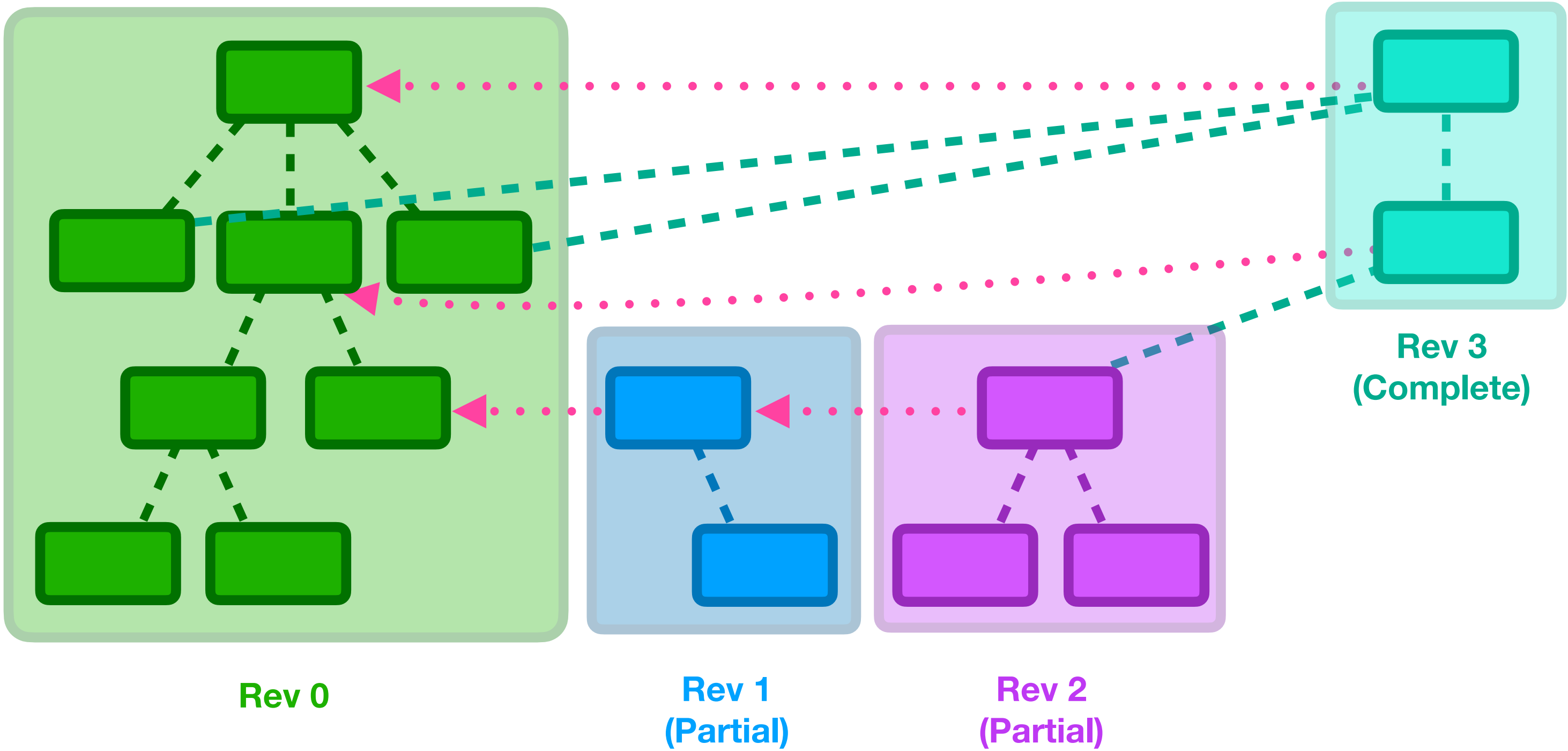
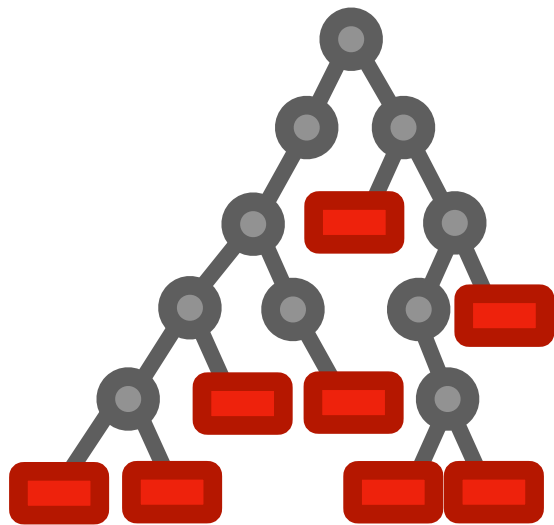
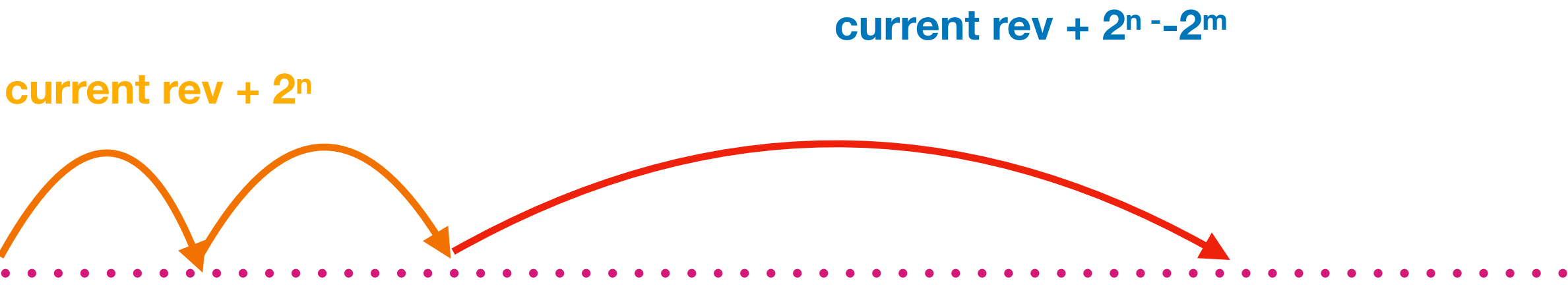
🔧 Progressive Fast Forward

current rev + 2^n



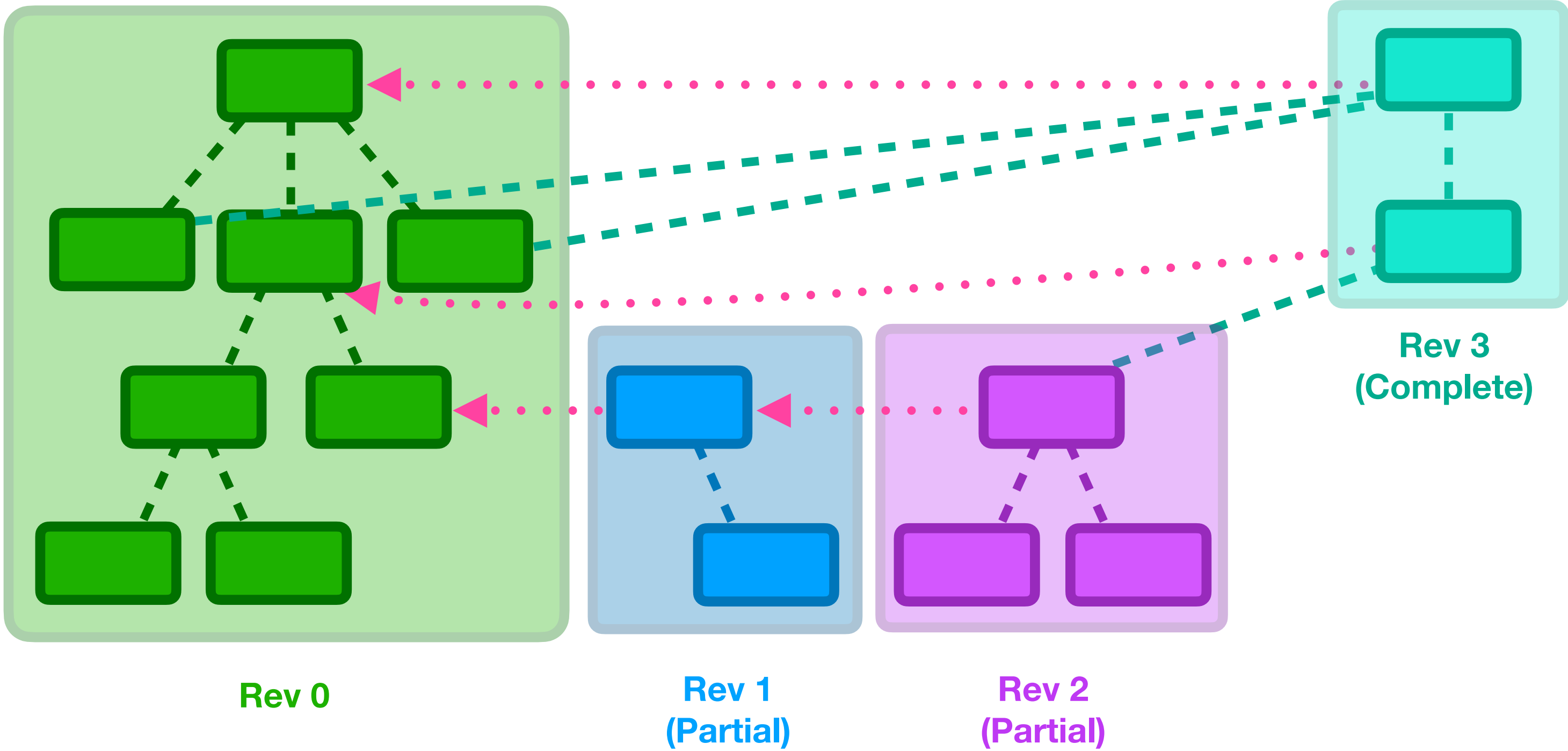
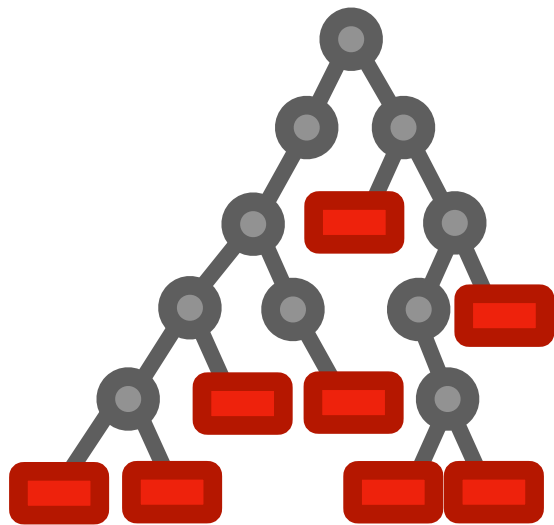
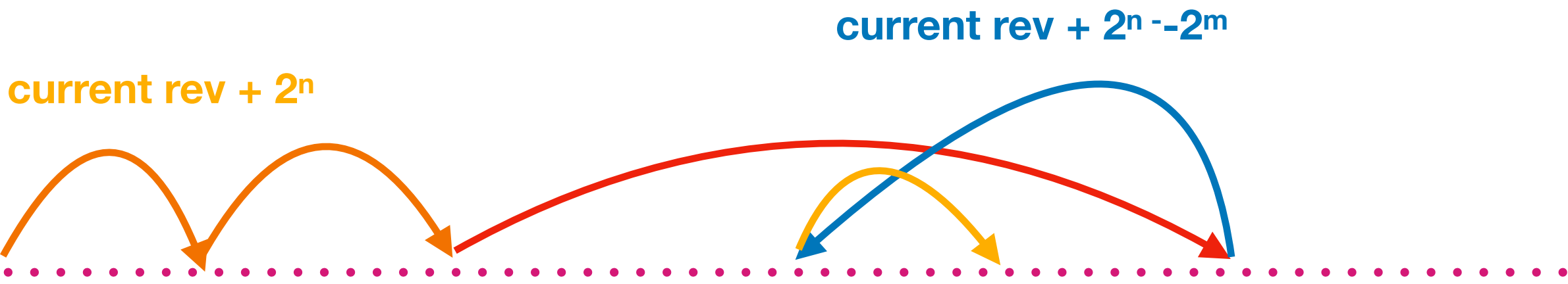
Private Nodes

🔧 Progressive Fast Forward



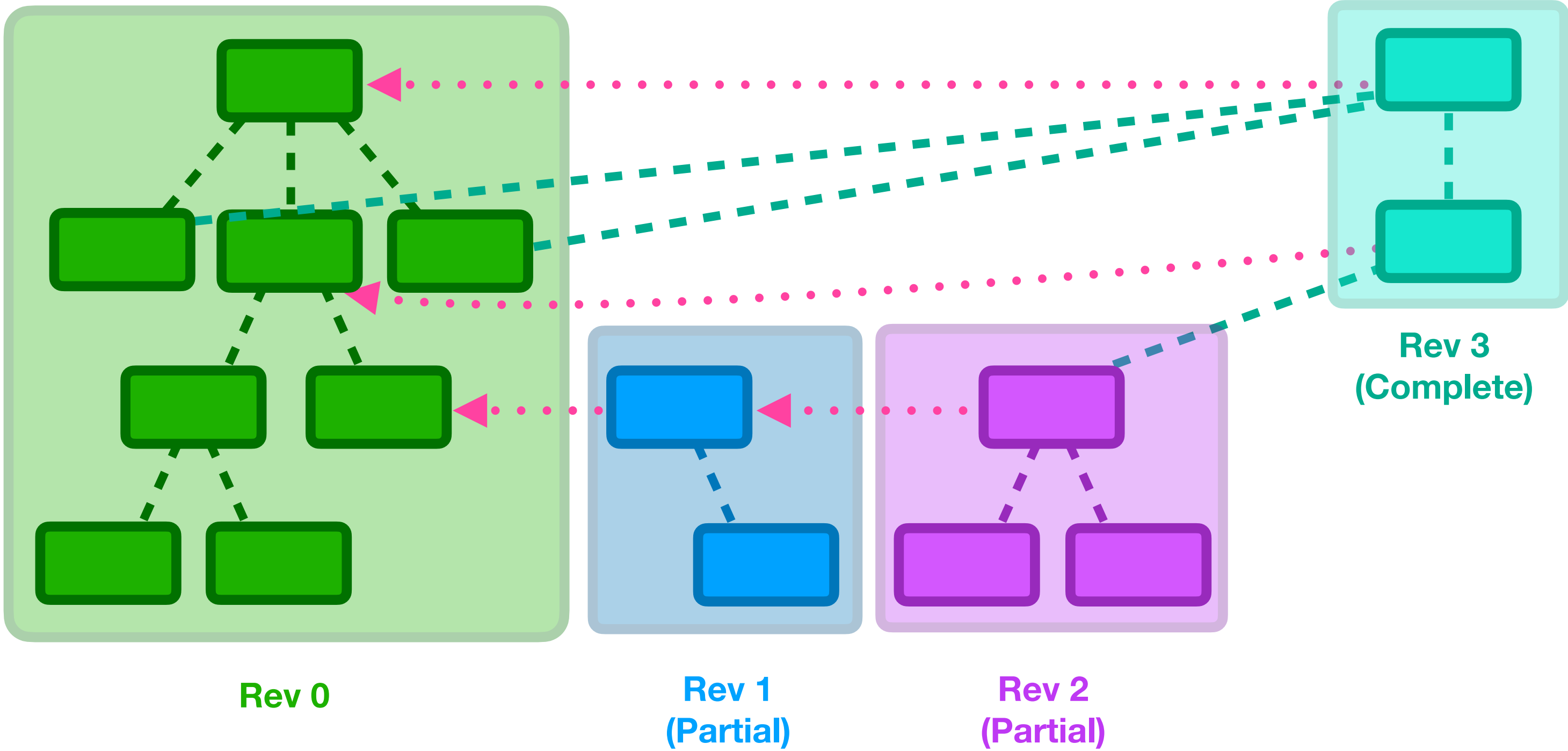
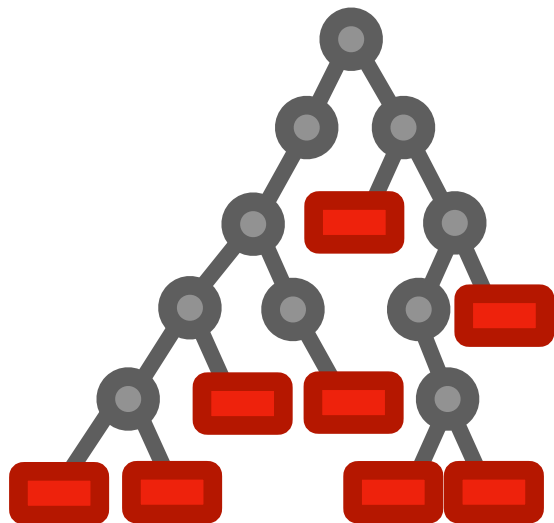
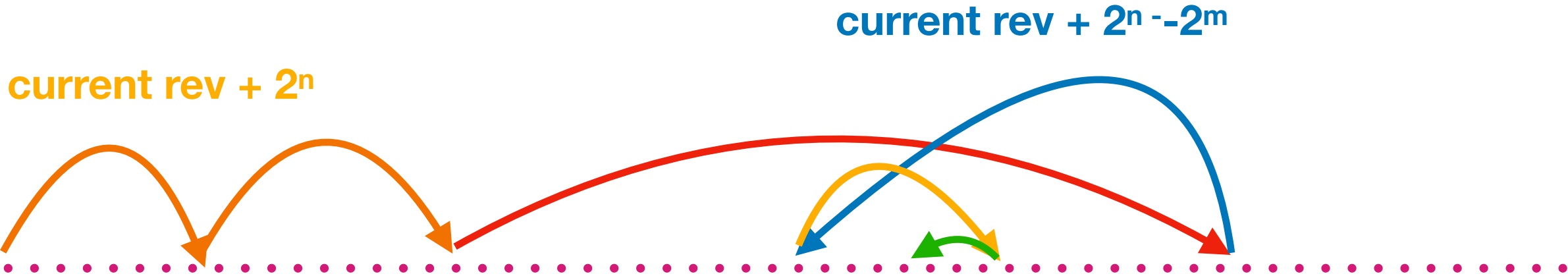
Private Nodes

🔧 Progressive Fast Forward



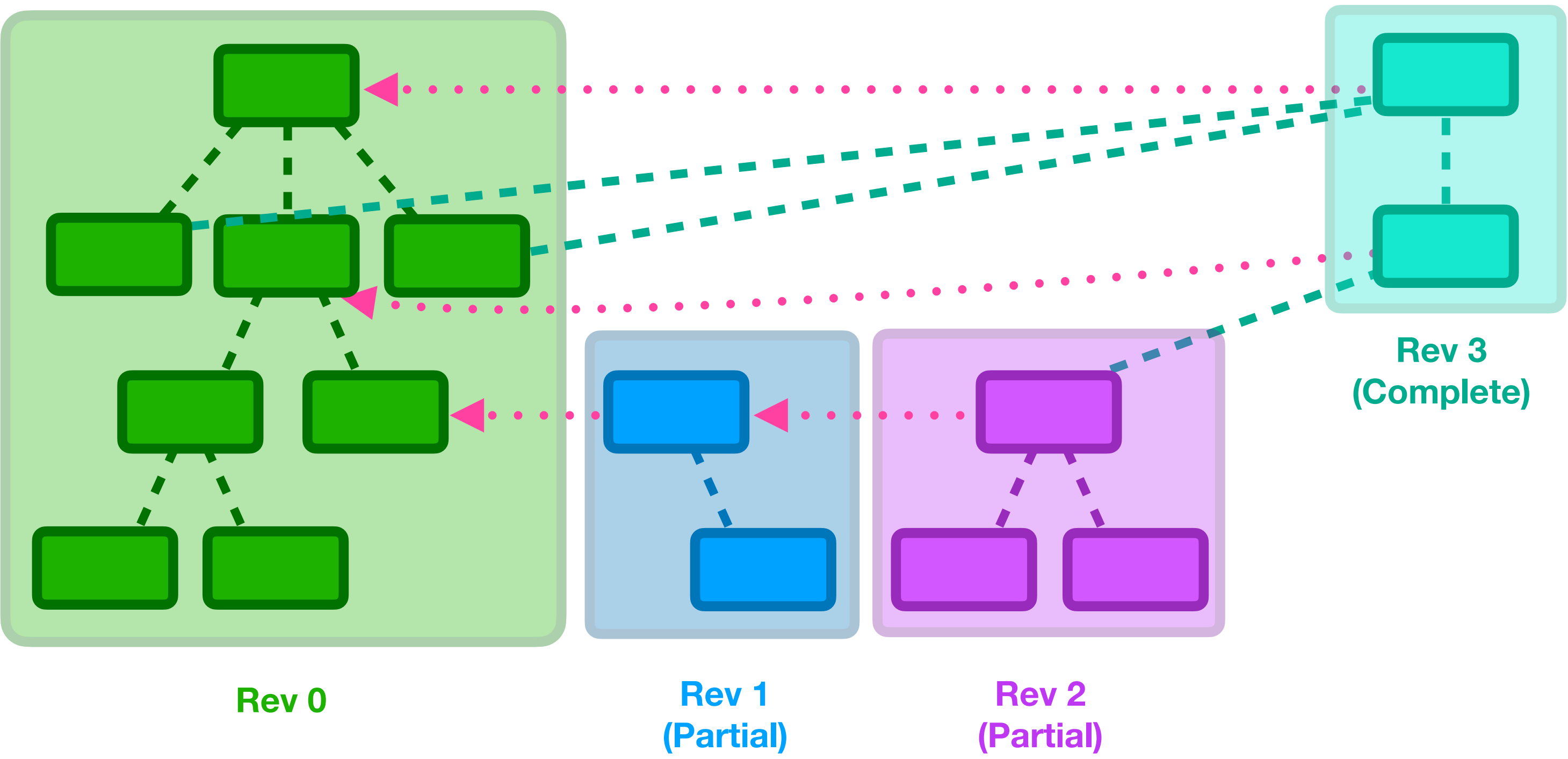
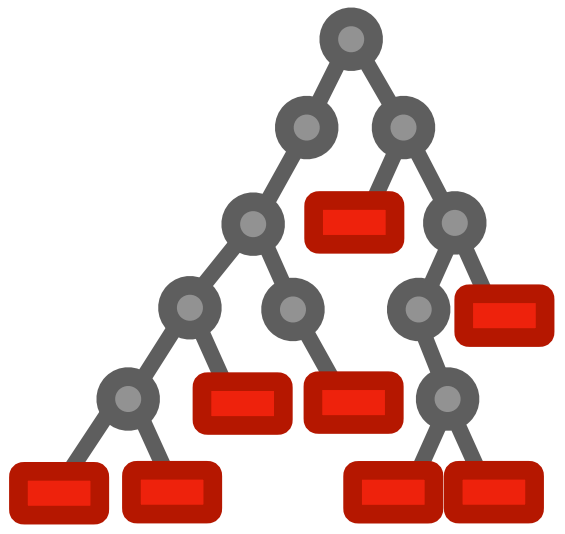
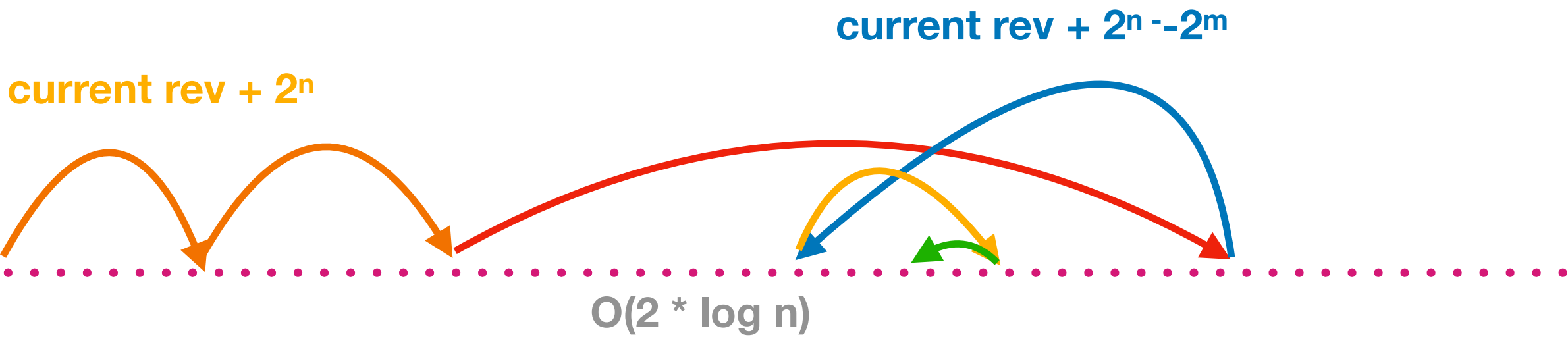
Private Nodes

⚒ Progressive Fast Forward



Private Nodes

🔧 Progressive Fast Forward



File Sharing



File Sharing

Shared by/with Me

File Sharing

Shared by/with Me

Shared *by* Me

File Sharing

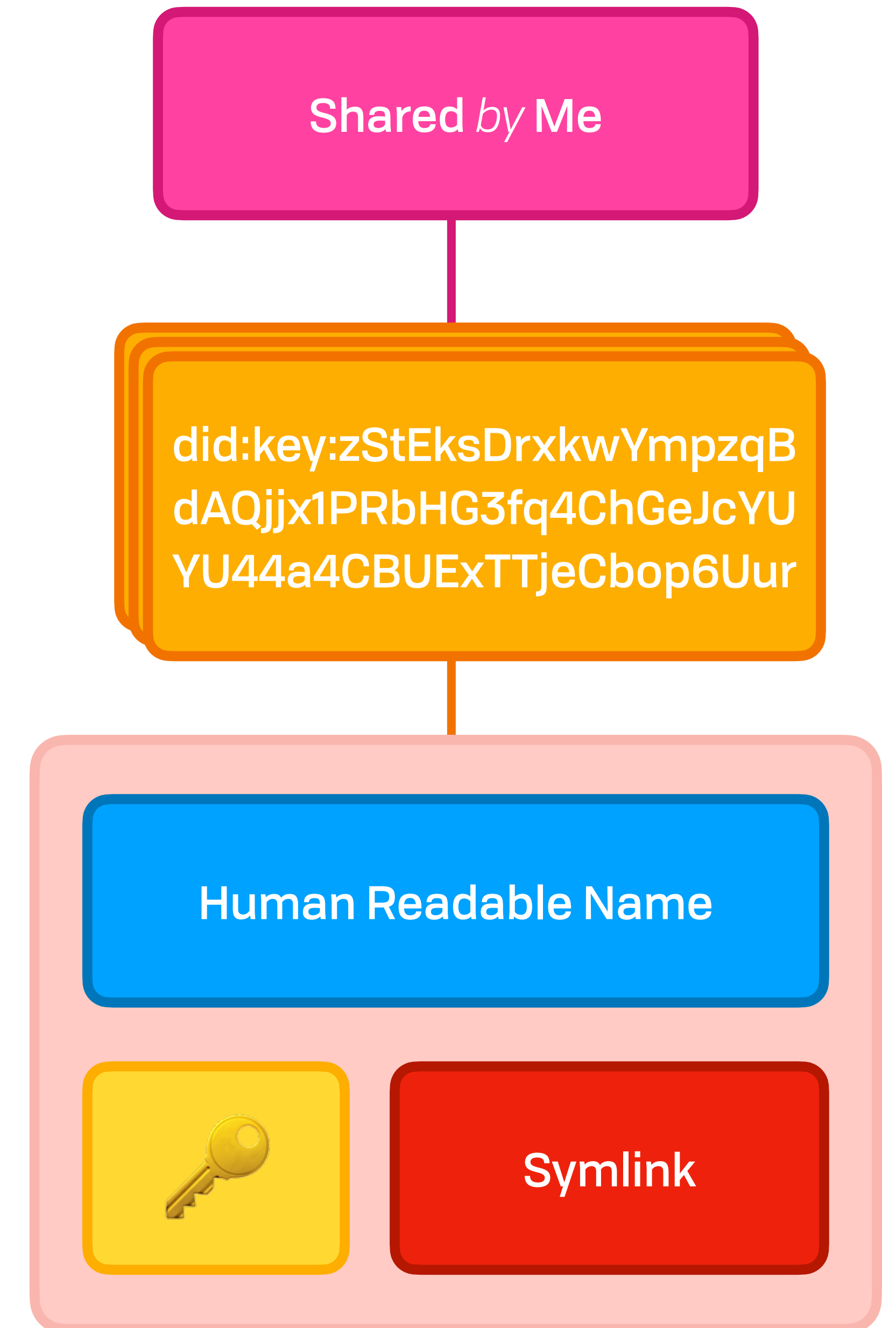
Shared by/with Me

Shared *by* Me

did:key:zStEksDrxkwYmpzqB
dAQjjx1PRbHG3fq4ChGeJcYU
YU44a4CBUExTTjeCbop6Uur

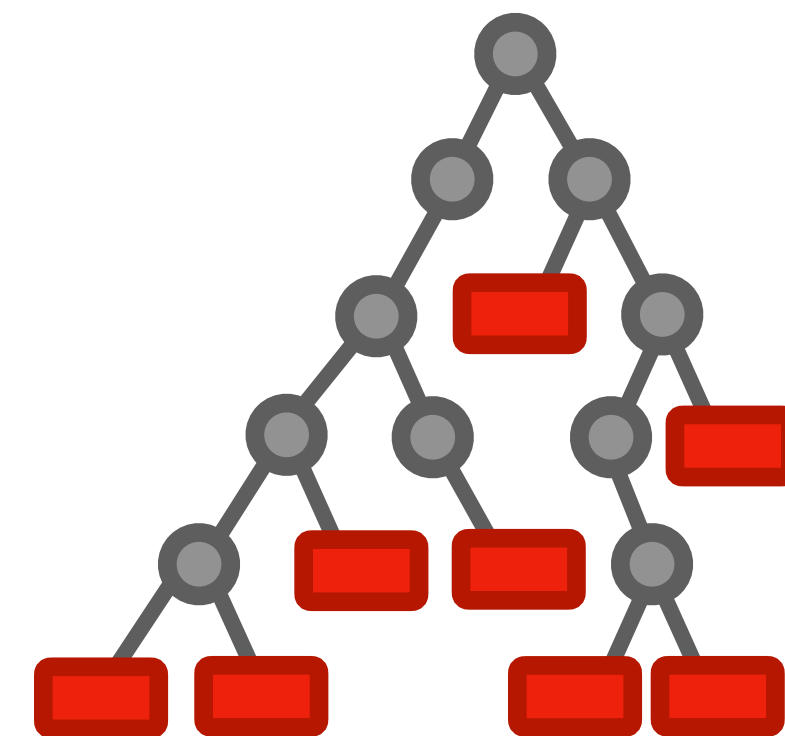
File Sharing

Shared by/with Me



File Sharing

Shared by/with Me



Shared *by* Me

did:key:zStEksDrxkwYmpzqB
dAQjjx1PRbHG3fq4ChGeJcYU
YU44a4CBUExTTjeCbop6Uur

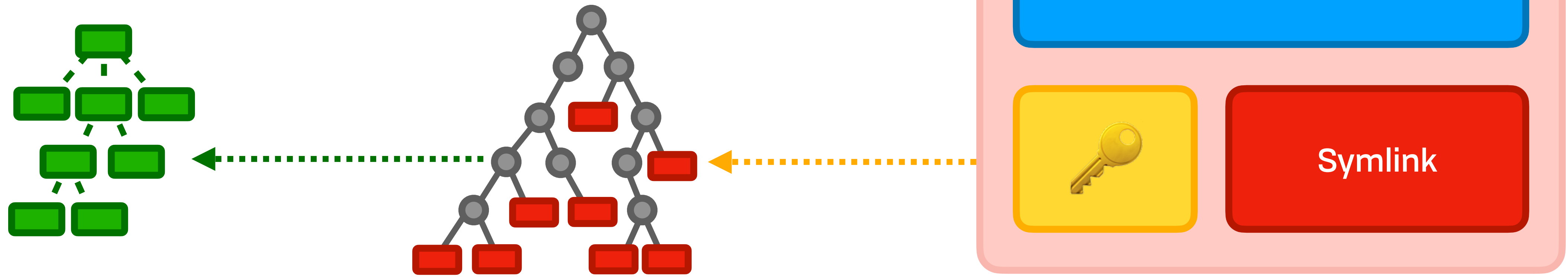
Human Readable Name



Symlink

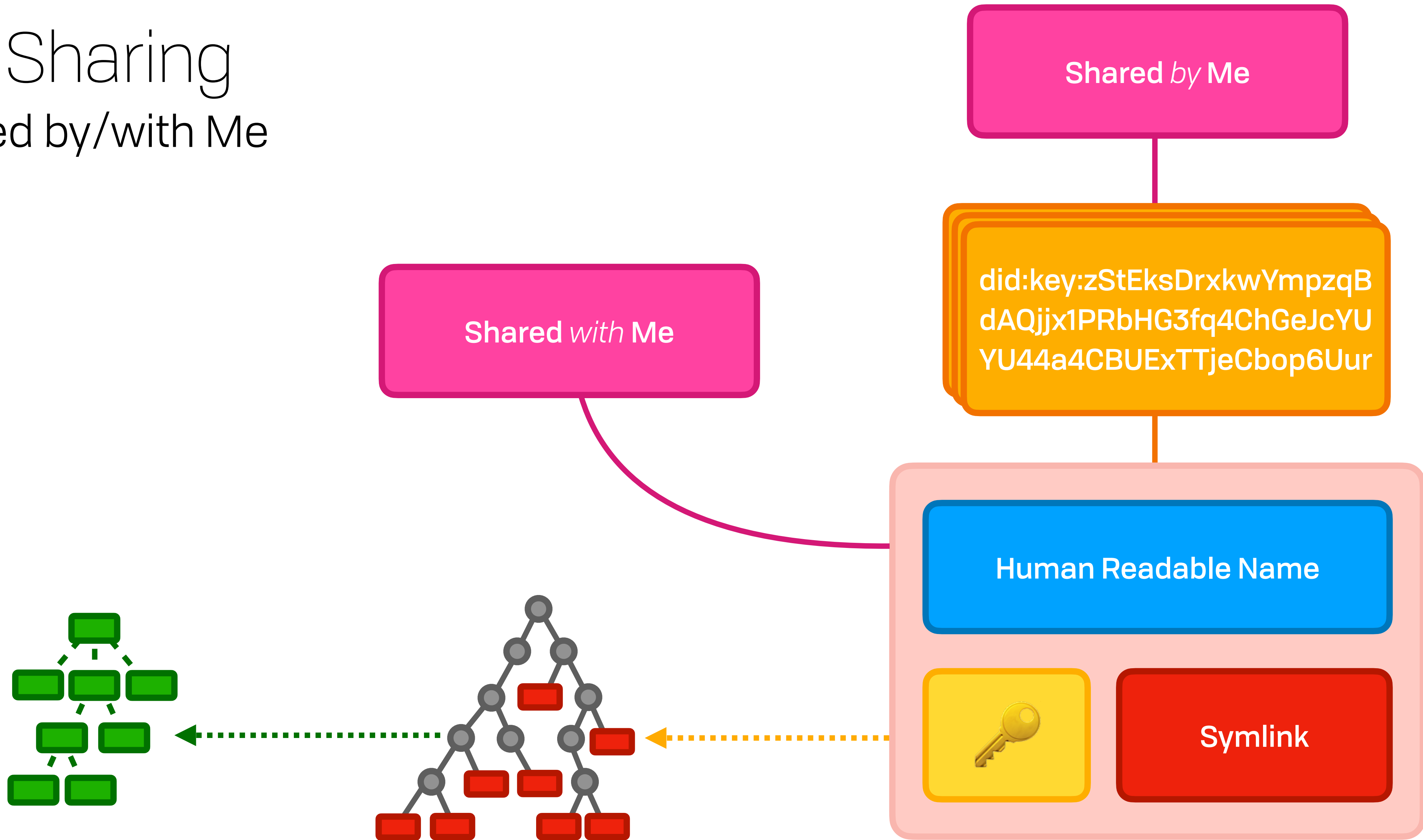
File Sharing

Shared by/with Me

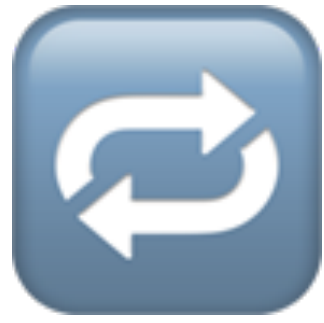


File Sharing

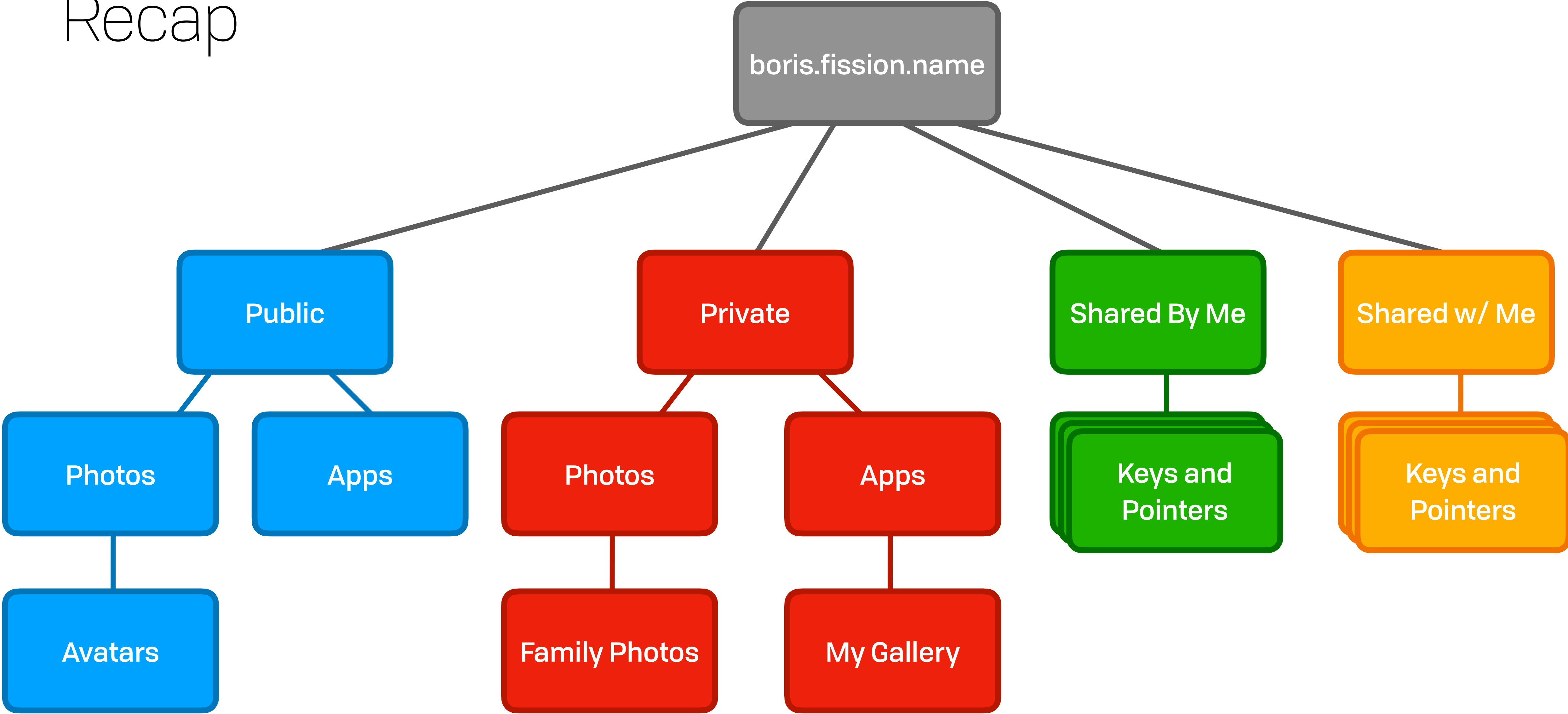
Shared by/with Me



Recap



Recap



<https://fission.codes>
<https://talk.fission.codes>
<https://discord.gg/zAQBDEq>



THANK YOU



brooklyn@fission.codes
github.com/expede
[@expede](#)

`https://fission.codes`
`https://talk.fission.codes`
`https://discord.gg/zAQBDEq`



THANK YOU



`brooklyn@fission.codes`
`github.com/expede`
`@expede`

GREAT DEVELOPER EXPERIENCES, LIVE FROM YOUR LAPTOP

Subscribe to our product updates newsletter

newsletter@fission.codes

Subscribe

[Visit the subscribe page](#) to pick from event, product, and blog link newsletters