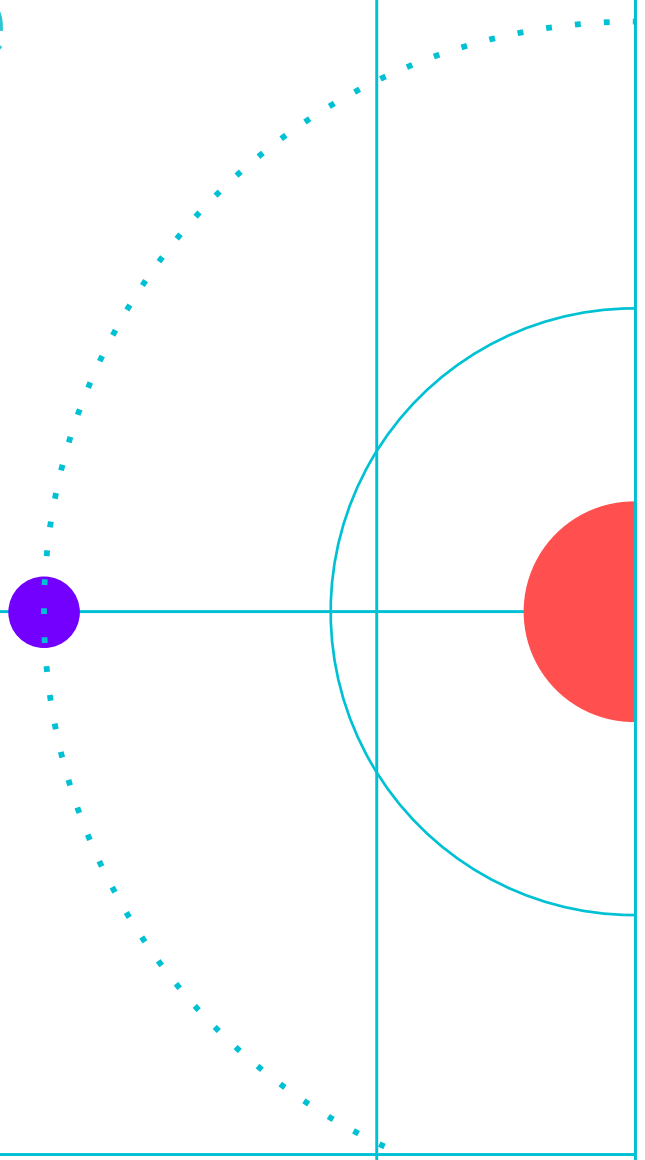
reUpNix: Reconfigurable and Updateable Embedded Systems



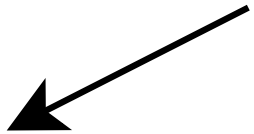
TUHH
Technische Universität Hamburg

2023-06-18

Niklas Gollenstede

Sending stuff 🛰️ into space 🚀 is expensive 💰

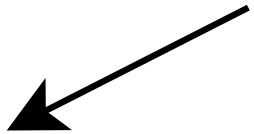
Sending stuff 🛰️ into space 🚀 is expensive 💰



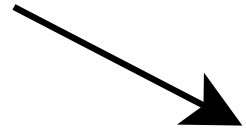
update

Embedded: Updates & Reconfiguration

Sending stuff 🛰️ into space 🚀 is expensive 💰



update



reconfigure

Sending stuff 🛰️ into space 🚀 is expensive 💰

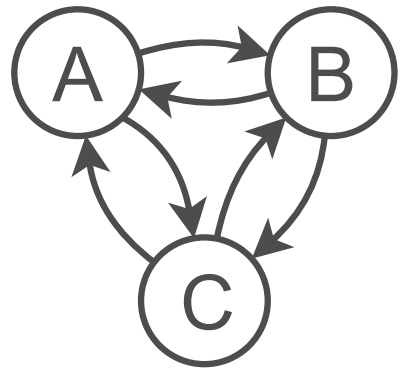
update

reconfigure

Sending stuff 🛰️ into space 🚀 is expensive 💰

update

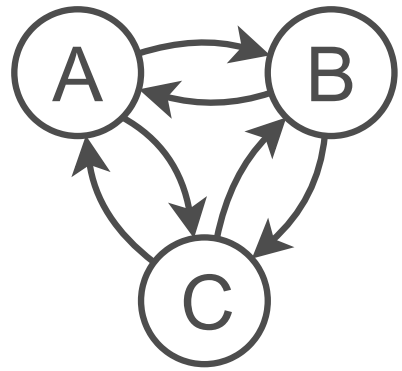
reconfigure



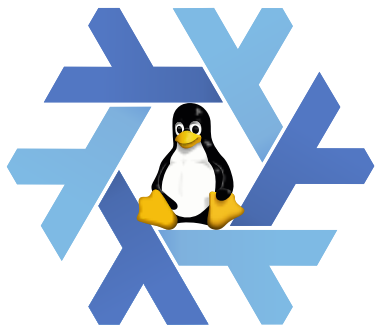
Sending stuff 🛰️ into space 🚀 is expensive 💰

update

reconfigure



NixOS

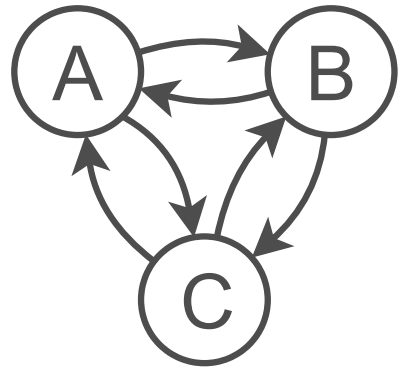


Embedded: Updates & Reconfiguration

Sending stuff 🛰️ into space 🚀 is expensive 💰

update

reconfigure



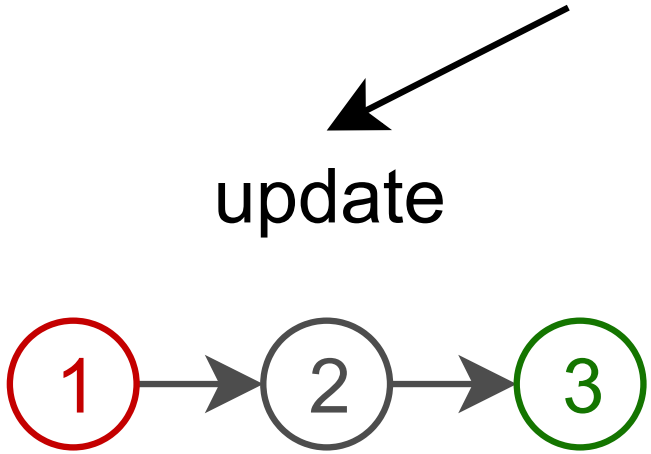
NixOS

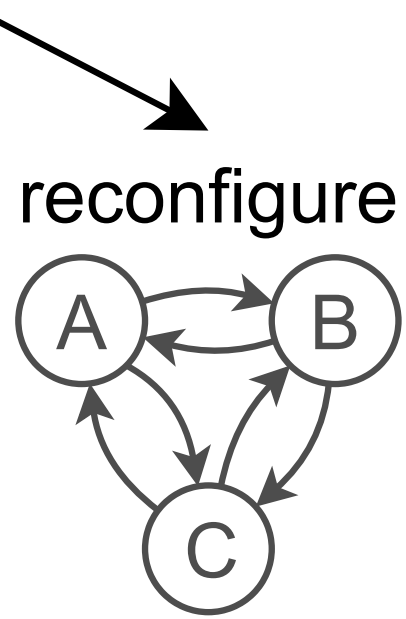
:

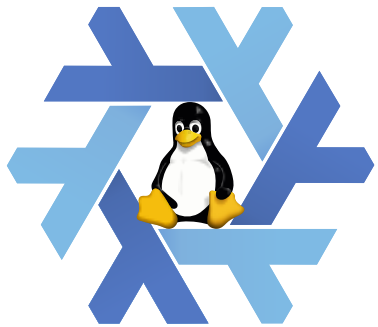
~~embedded~~

Embedded: Updates & Reconfiguration

Sending stuff 🛰️ into space 🚀 is expensive 💰

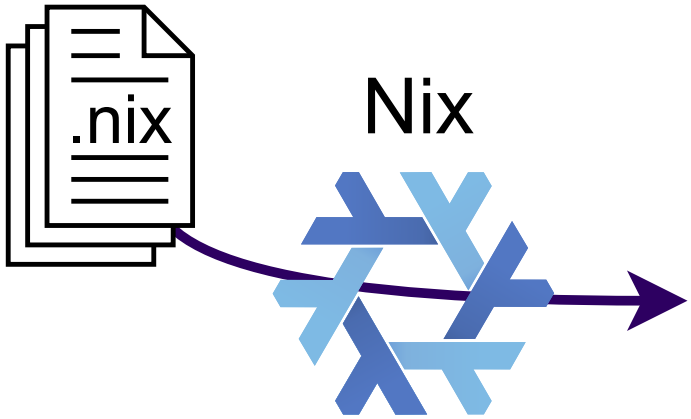


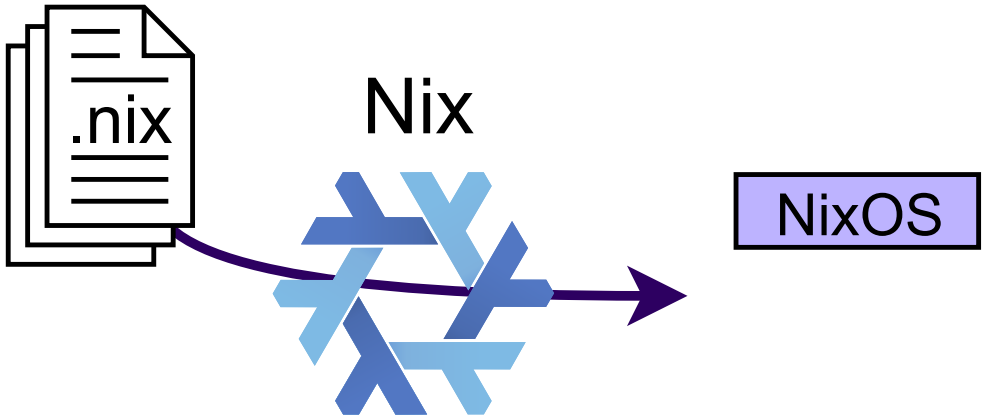


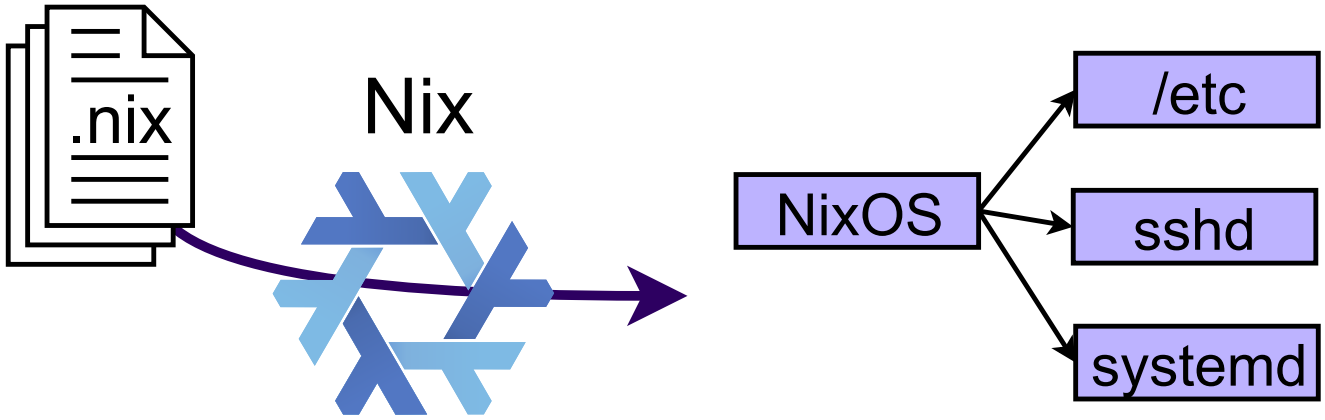
NixOS  : why?

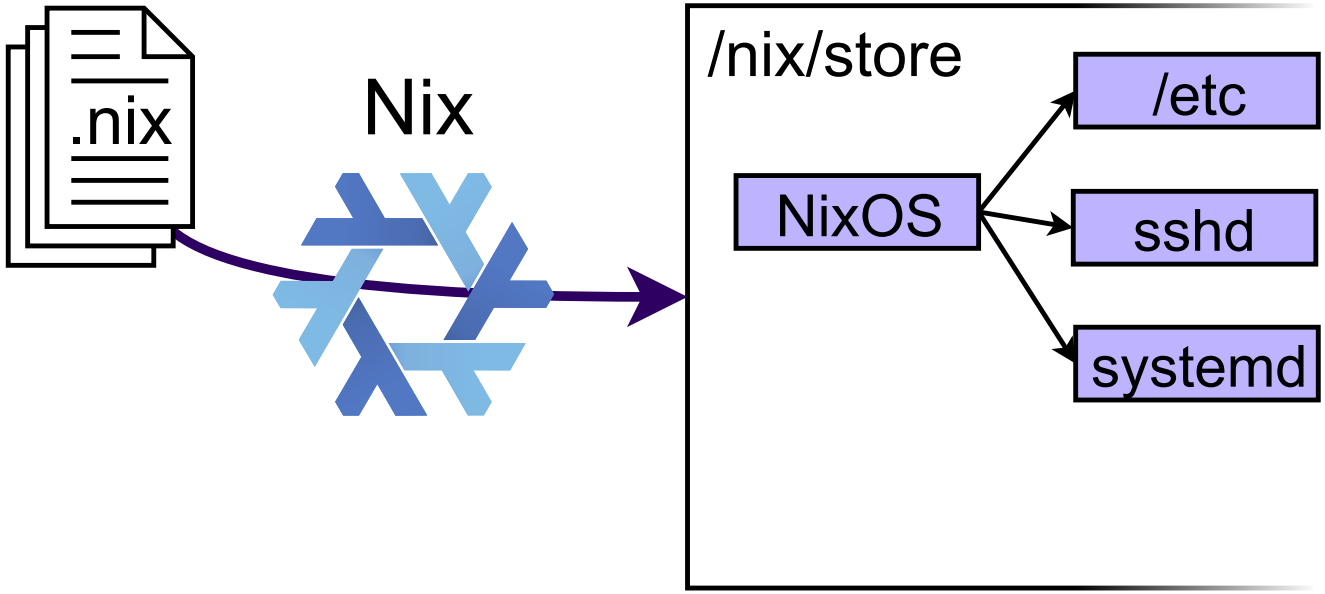
~~embedded~~

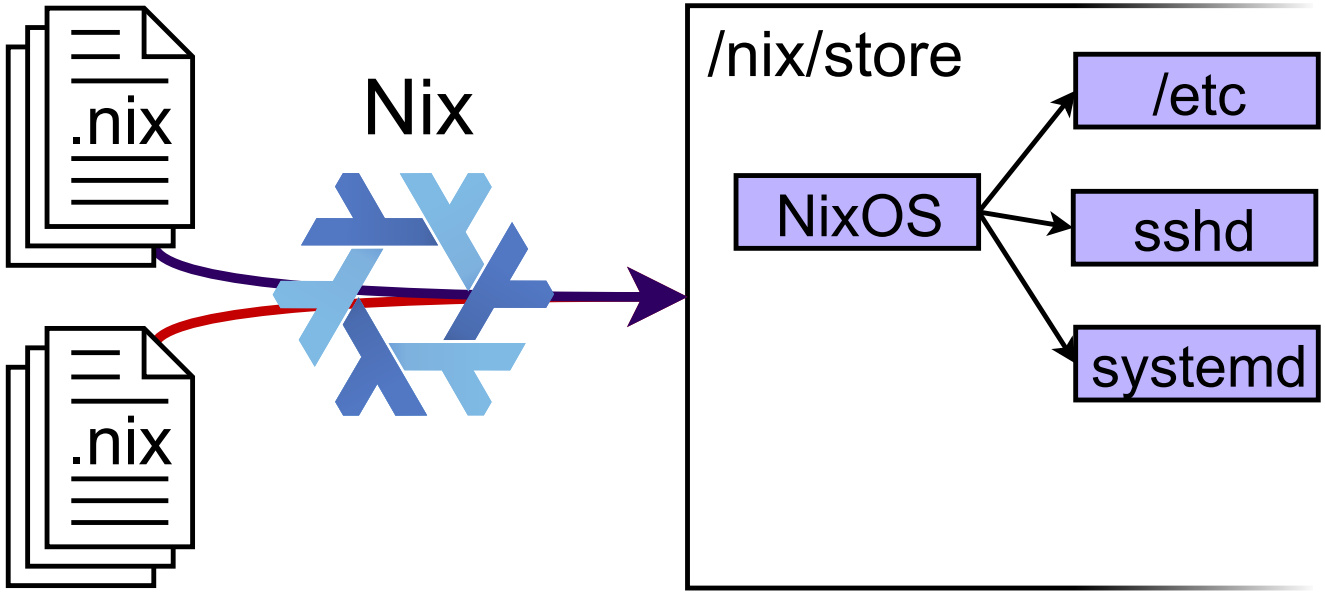
Nix

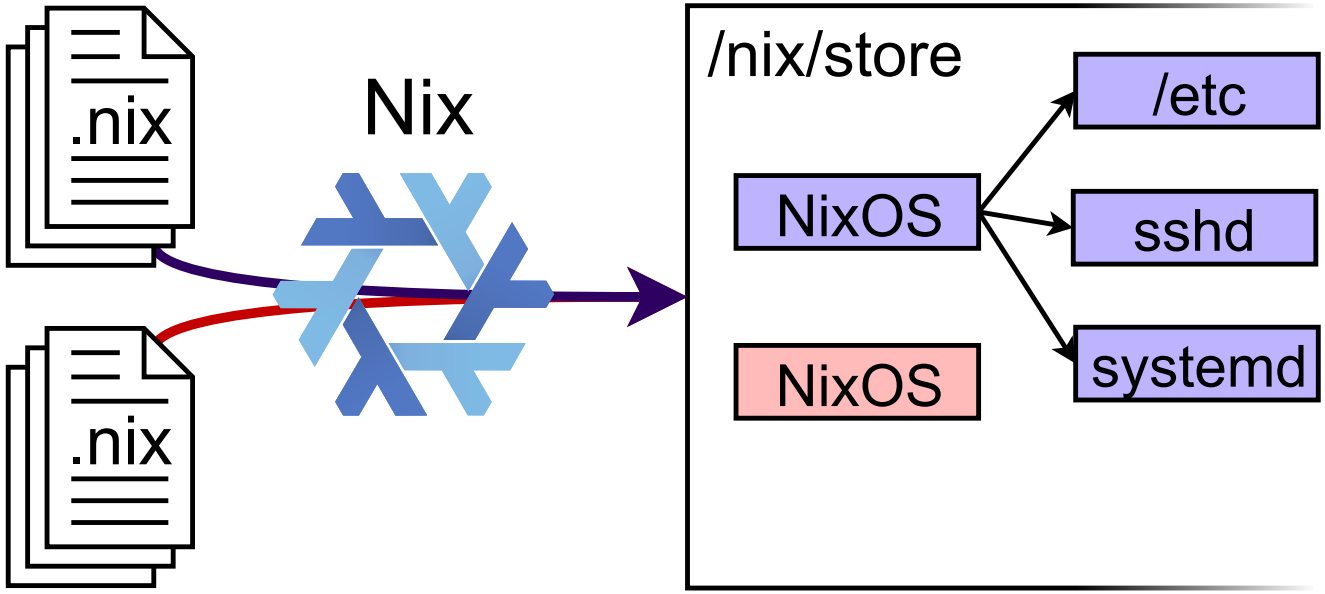

.nix
Nix

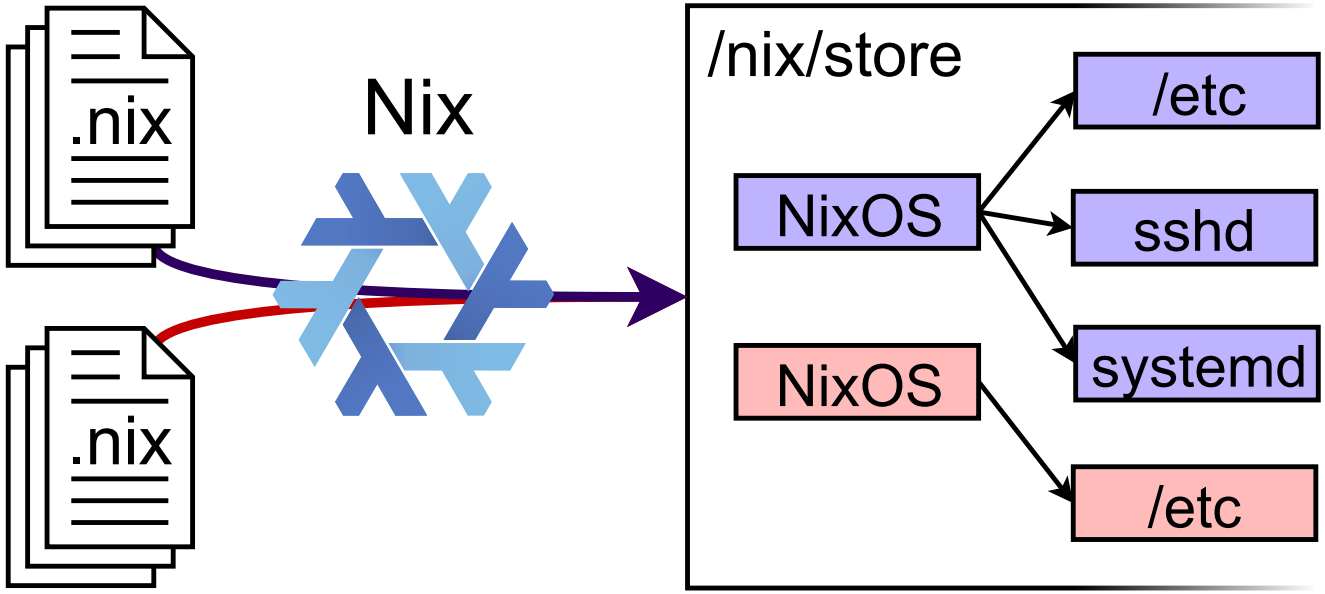

.nix
Nix
NixOS

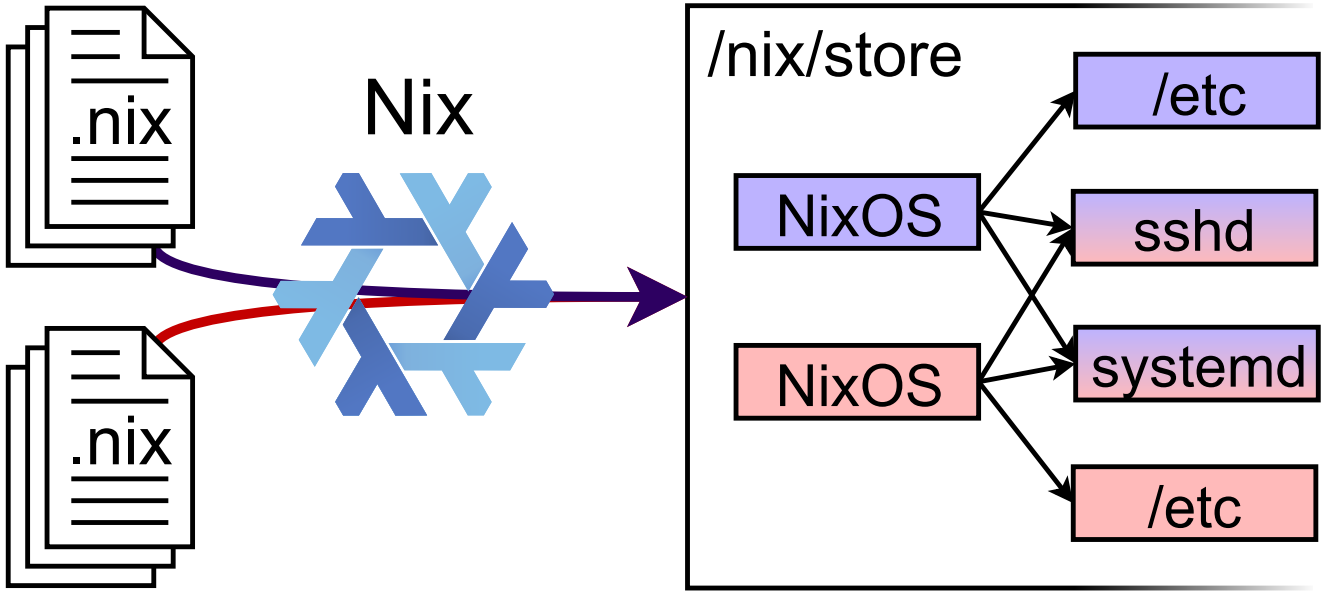

.nix
Nix
NixOS
/etc
sshd
systemd

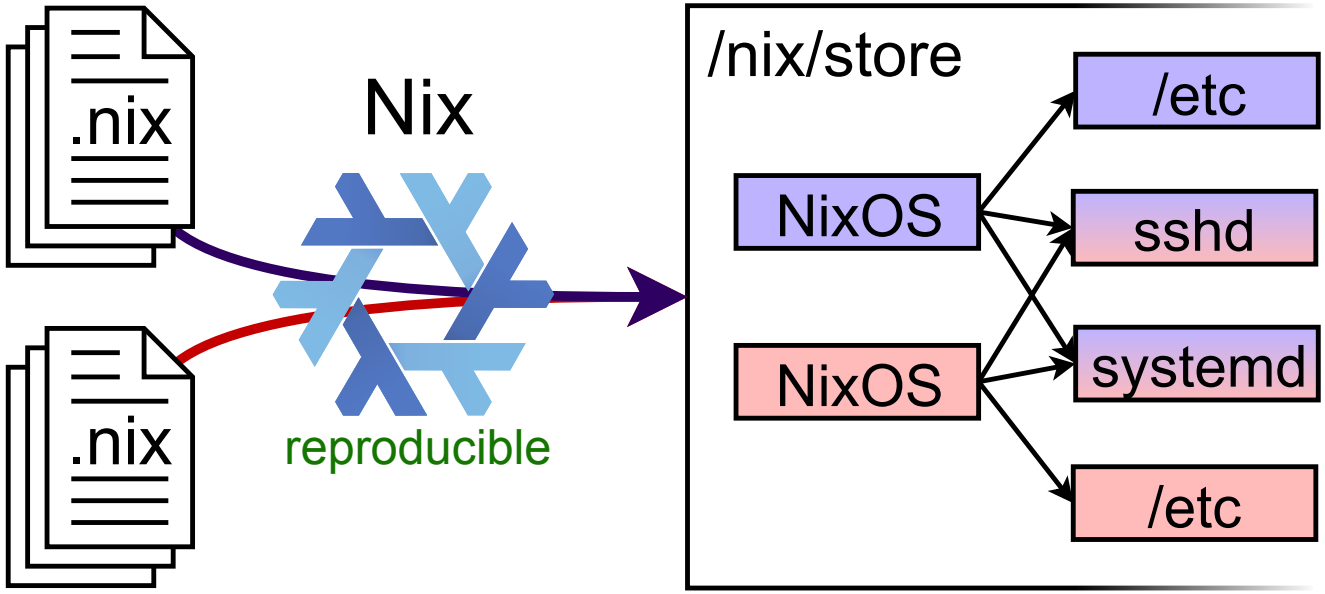

.nix
Nix
/nix/store
NixOS
/etc
sshd
systemd

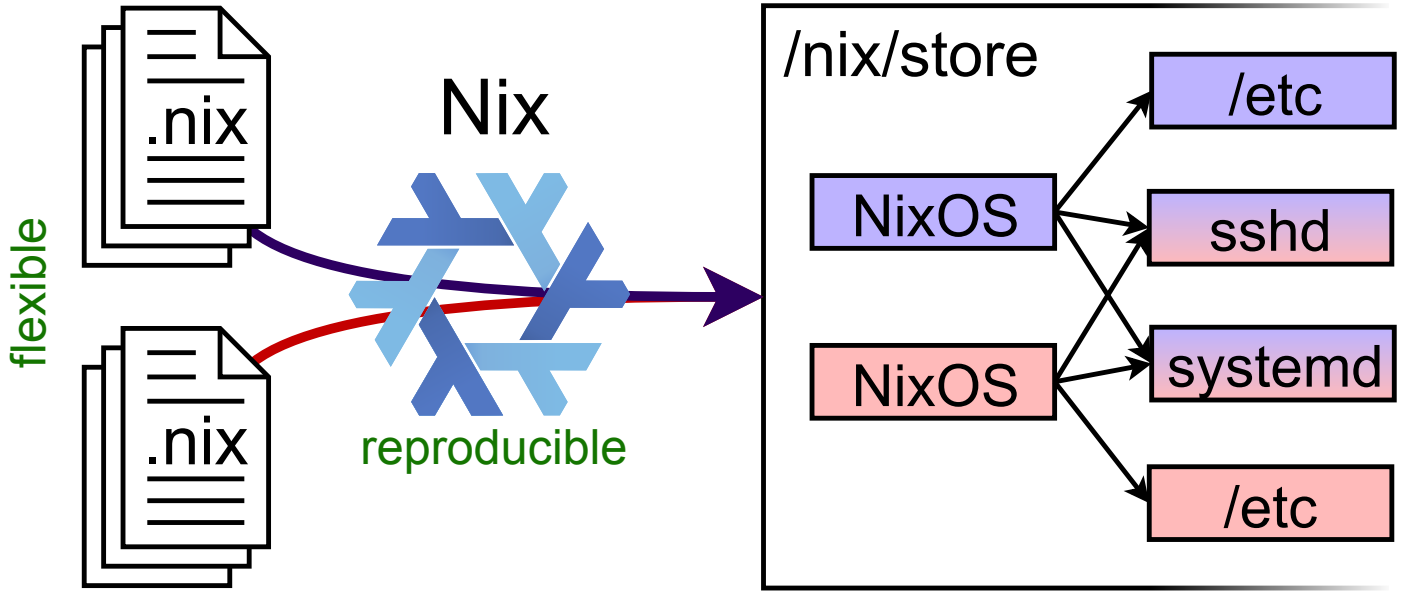

.nix
.nix
Nix
/nix/store
NixOS
/etc
sshd
systemd

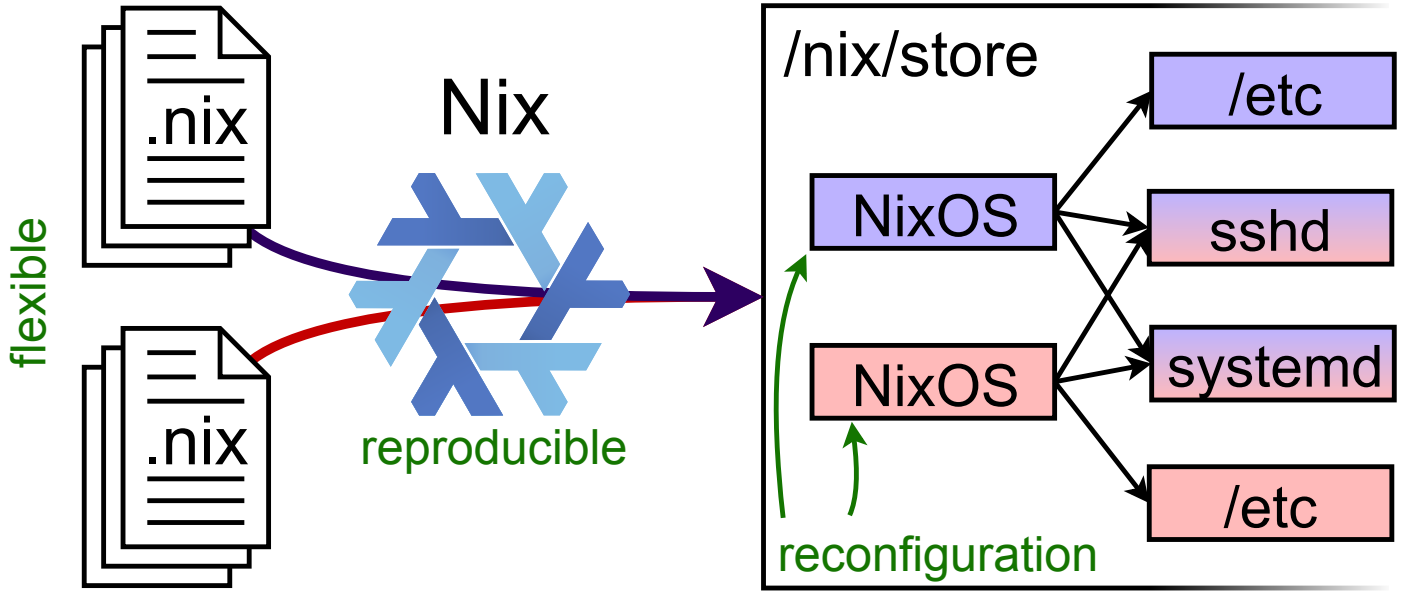

.nix
.nix
Nix
/nix/store
NixOS
NixOS
/etc
sshd
systemd

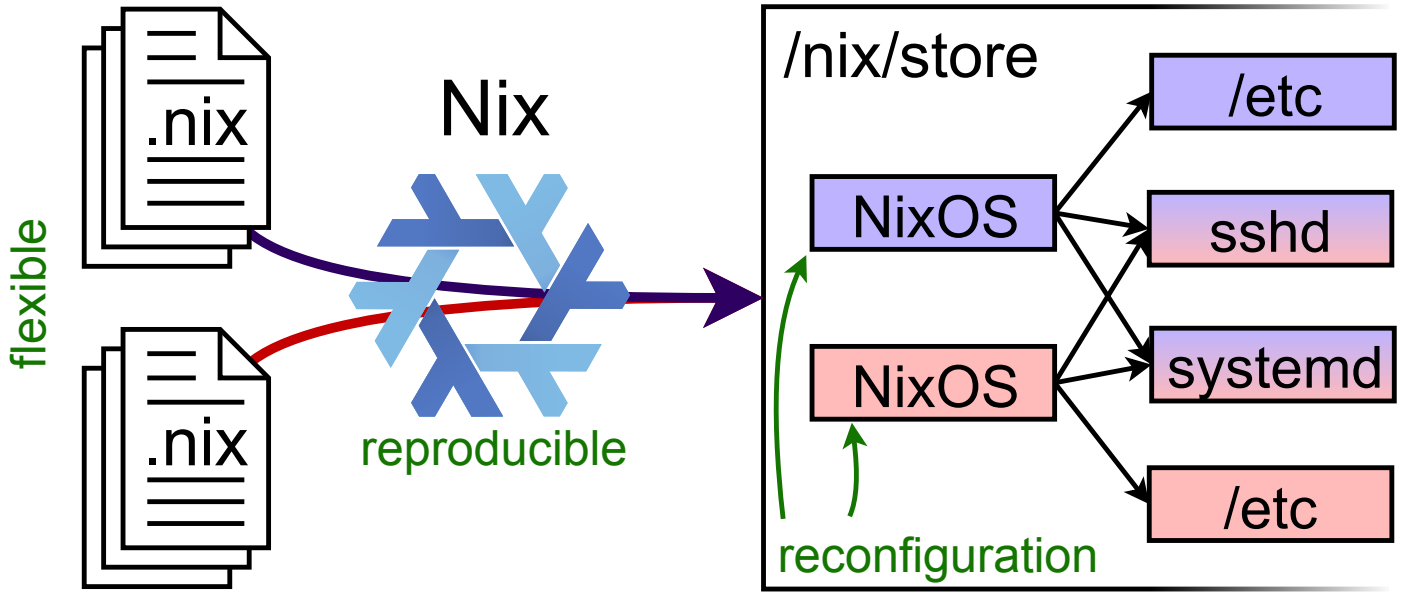

.nix
.nix
Nix
/nix/store
NixOS
NixOS
/etc
sshd
systemd
/etc


.nix
.nix
Nix
/nix/store
NixOS
NixOS
/etc
sshd
systemd
/etc

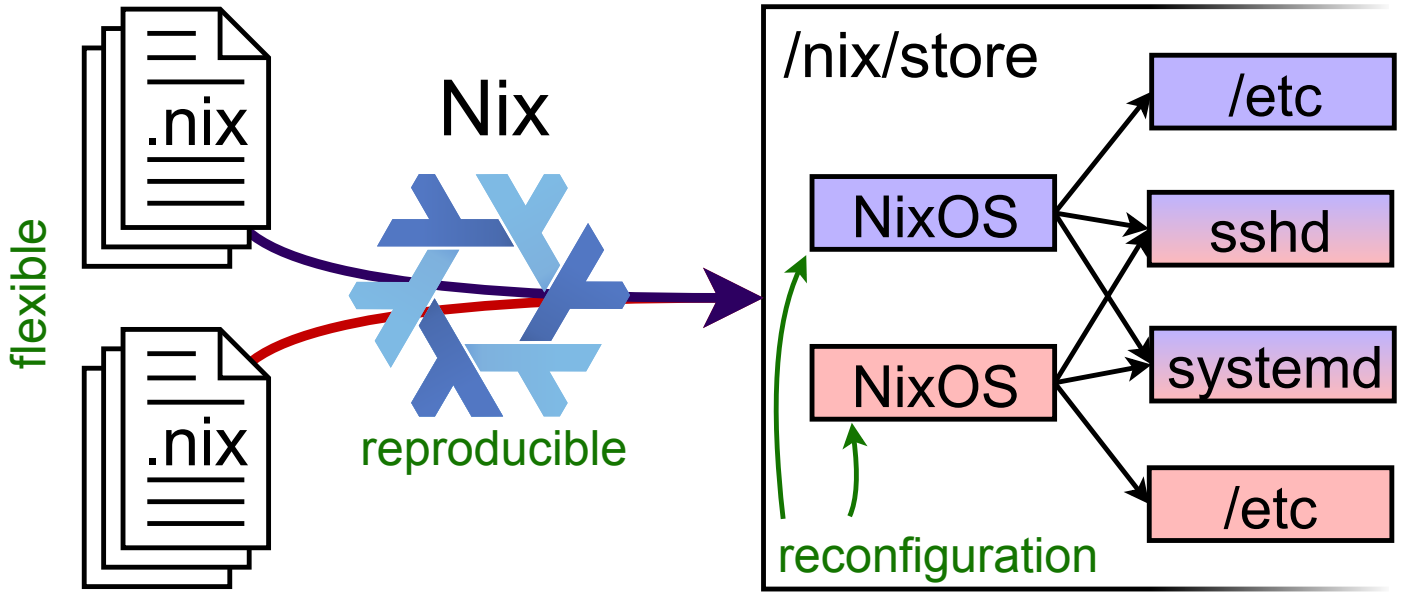

.nix
.nix
Nix
reproducible
/nix/store
NixOS
NixOS
/etc
sshd
systemd
/etc


flexible
.nix
.nix
Nix
reproducible
/nix/store
NixOS
NixOS
/etc
sshd
systemd
/etc


flexible
.nix
.nix
Nix
reproducible
/nix/store
NixOS
NixOS
/etc
sshd
systemd
/etc
reconfiguration


.nix
.nix
flexible
Nix
reproducible
/nix/store
NixOS
NixOS
/etc
sshd
systemd
/etc
reconfiguration

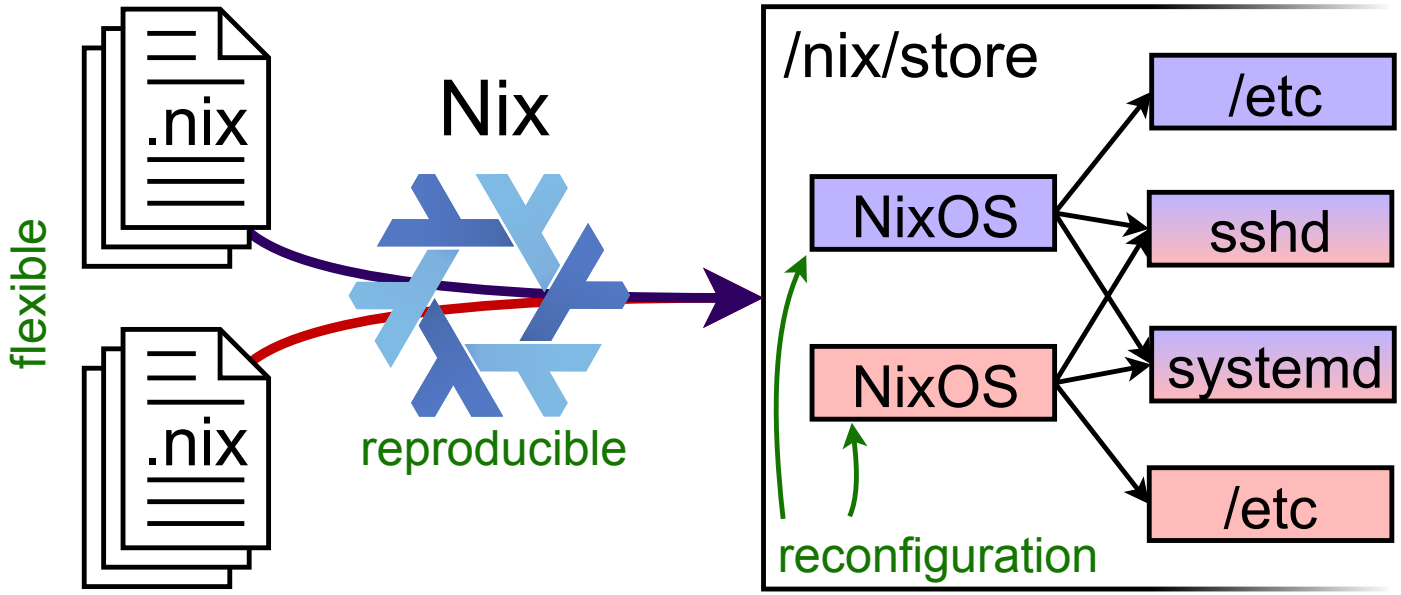
NixOS ✗



NixOS ✗
Large base system

⇒

reUpNix contribution
Shrink (~1.1 GiB → 155 MiB)



NixOS ✗

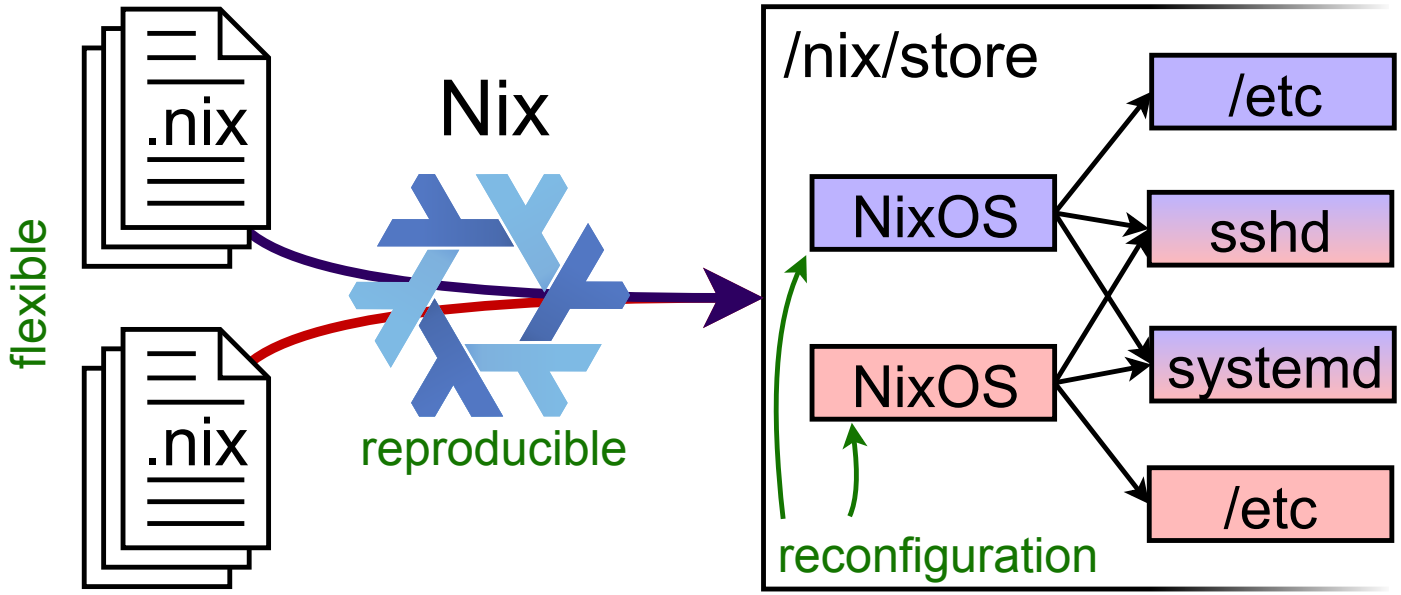
Large base system ⇒

OCI containers inefficient ⇒

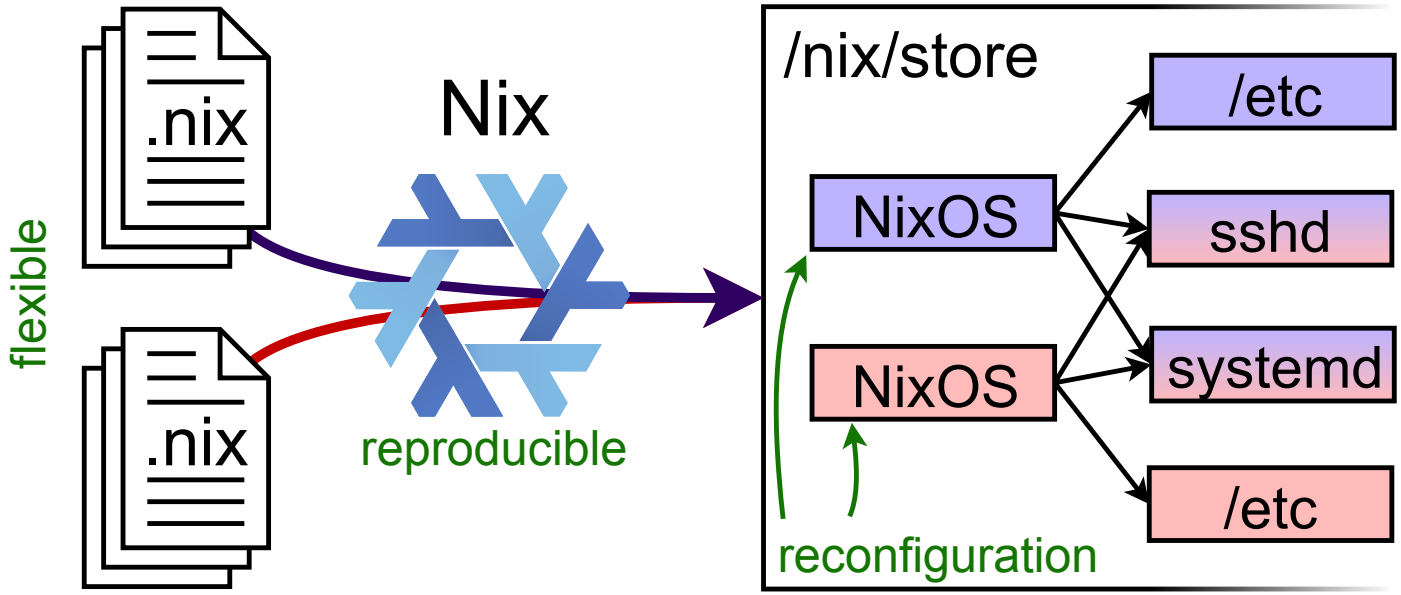
reUpNix contribution

Shrink (~1.1 GiB → 155 MiB)

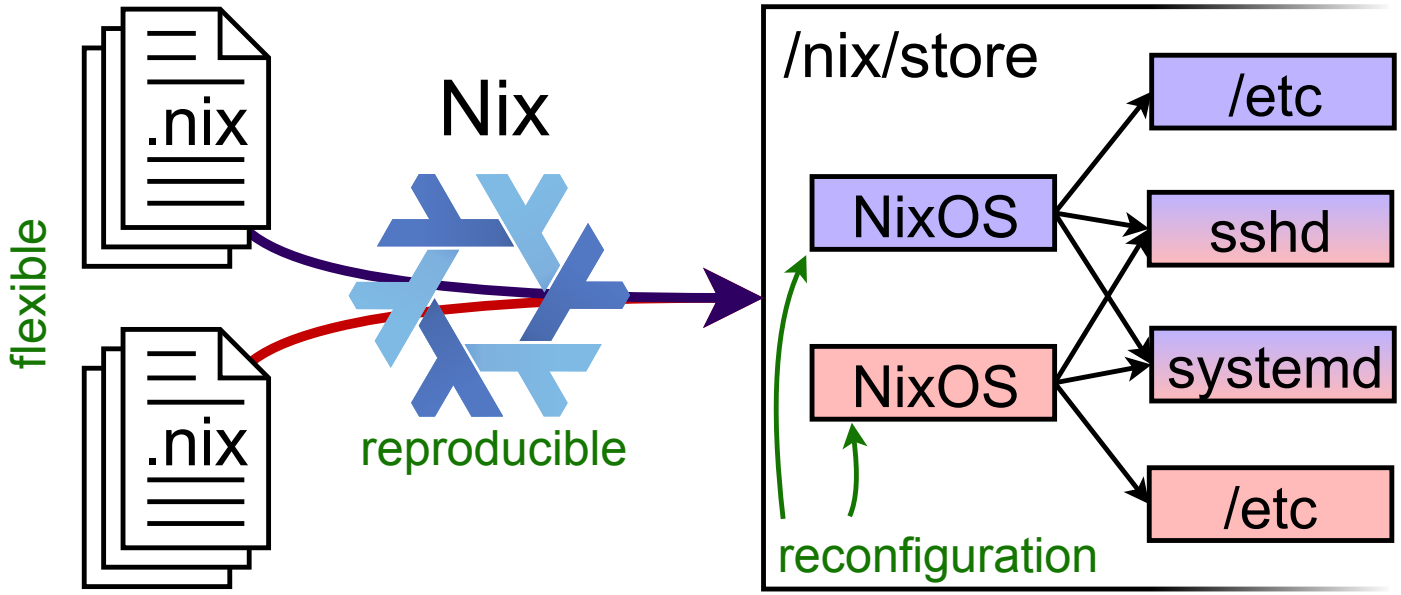
Remove Docker daemon & deduplicate files across images



NixOS ✗		**reUpNix contribution**
Large base system	⇒	Shrink (~1.1 GiB → 155 MiB)
OCI containers inefficient	⇒	Remove Docker daemon & deduplicate files across images
Large system updates	⇒	Differential update transfer

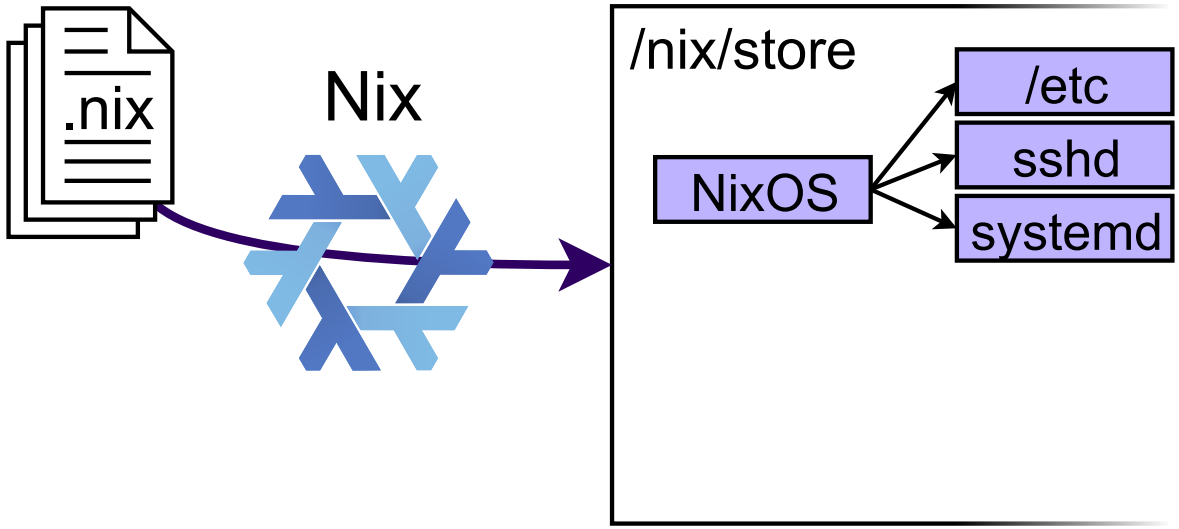


NixOS ✗		**reUpNix contribution**
Large base system	⇒	Shrink (~1.1 GiB → 155 MiB)
OCI containers inefficient	⇒	Remove Docker daemon & deduplicate files across images
Large system updates	⇒	Differential update transfer
Bootloader update fragile	⇒	Make update atomic & Integrate reconfiguration

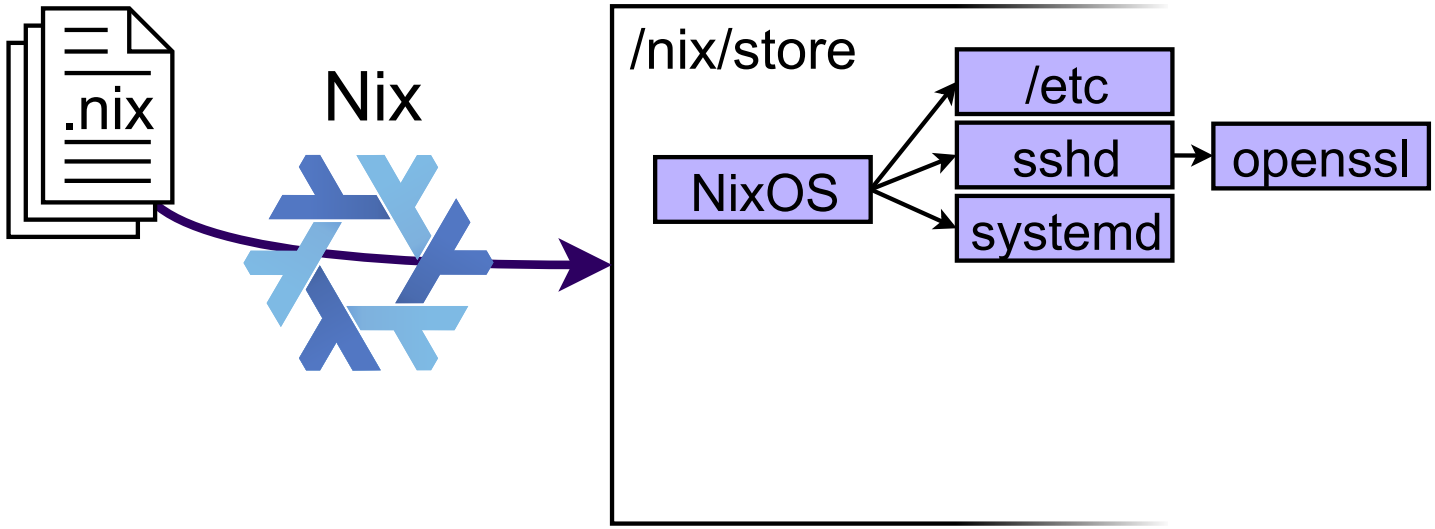


NixOS ✗		**reUpNix contribution**
Large base system	⇒	Shrink (~1.1 GiB → 155 MiB)
OCI containers inefficient	⇒	Remove Docker daemon & deduplicate files across images
Large system updates	⇒	Differential update transfer
Bootloader update fragile	⇒	Make update atomic & Integrate reconfiguration

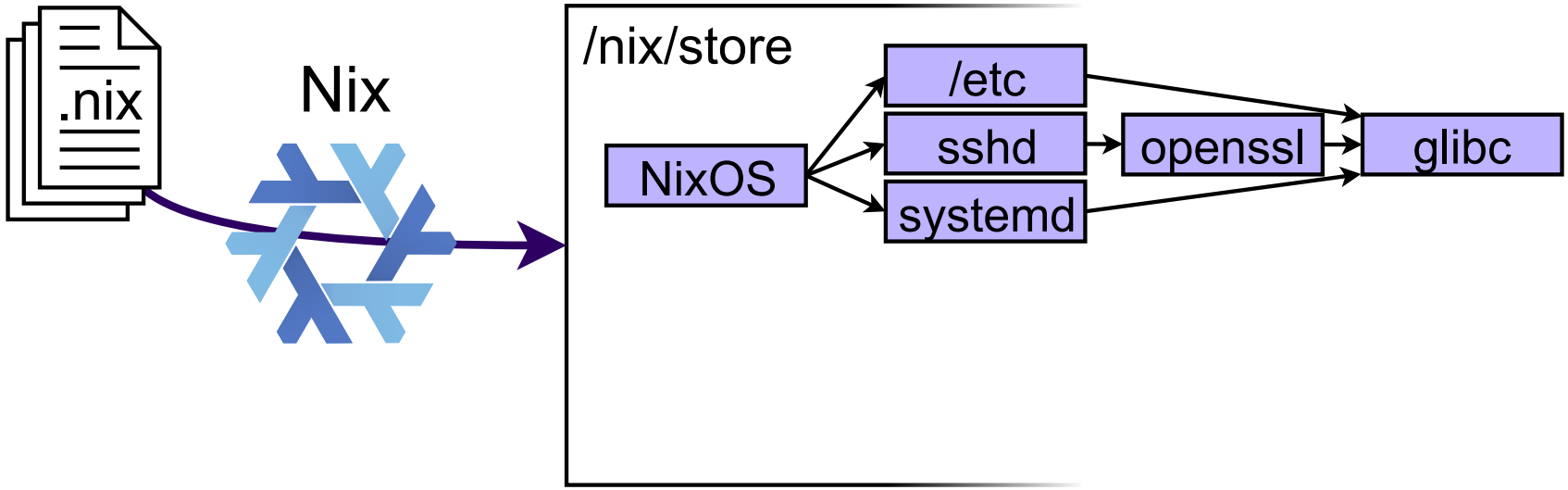
Differential Update Transfer



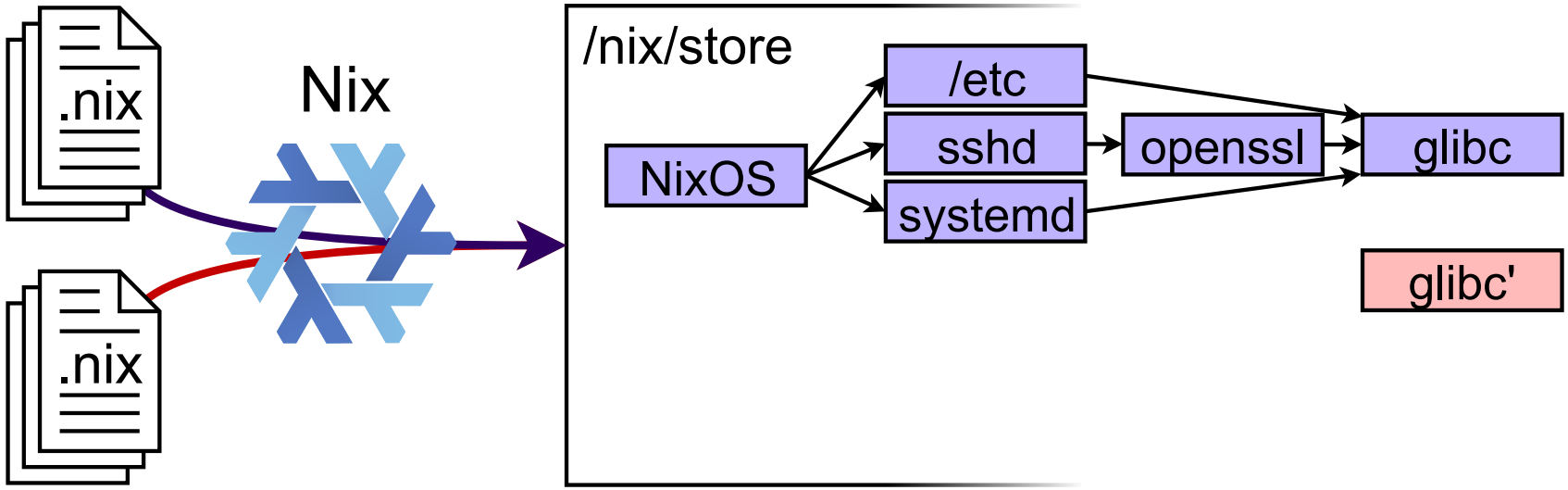
Differential Update Transfer



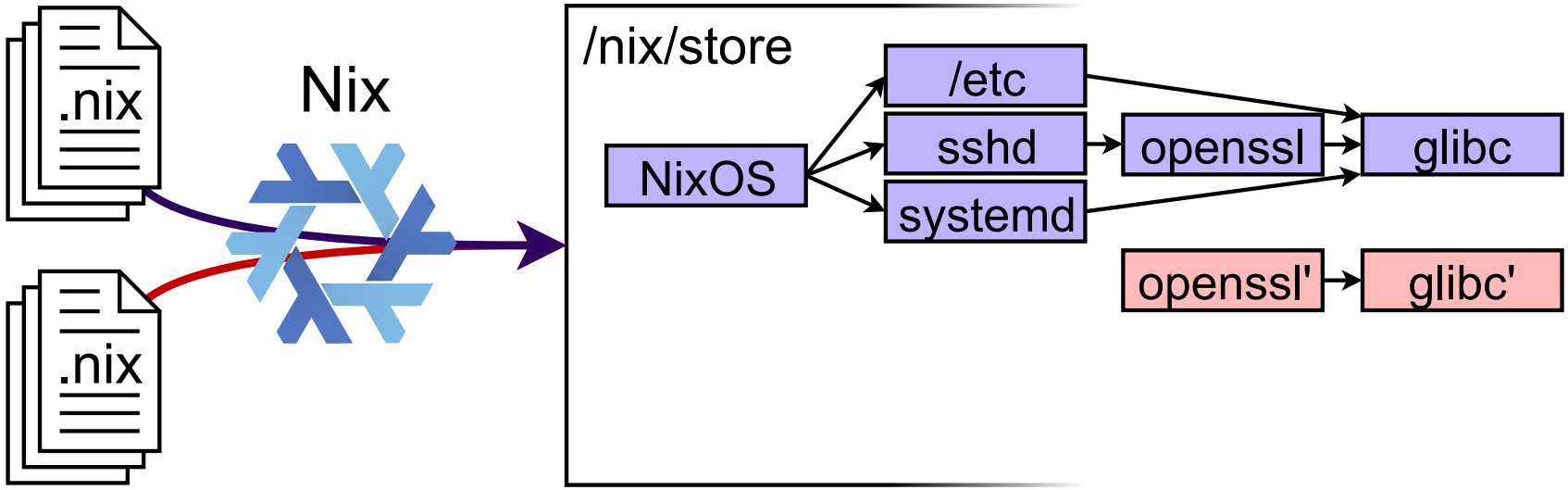
Differential Update Transfer



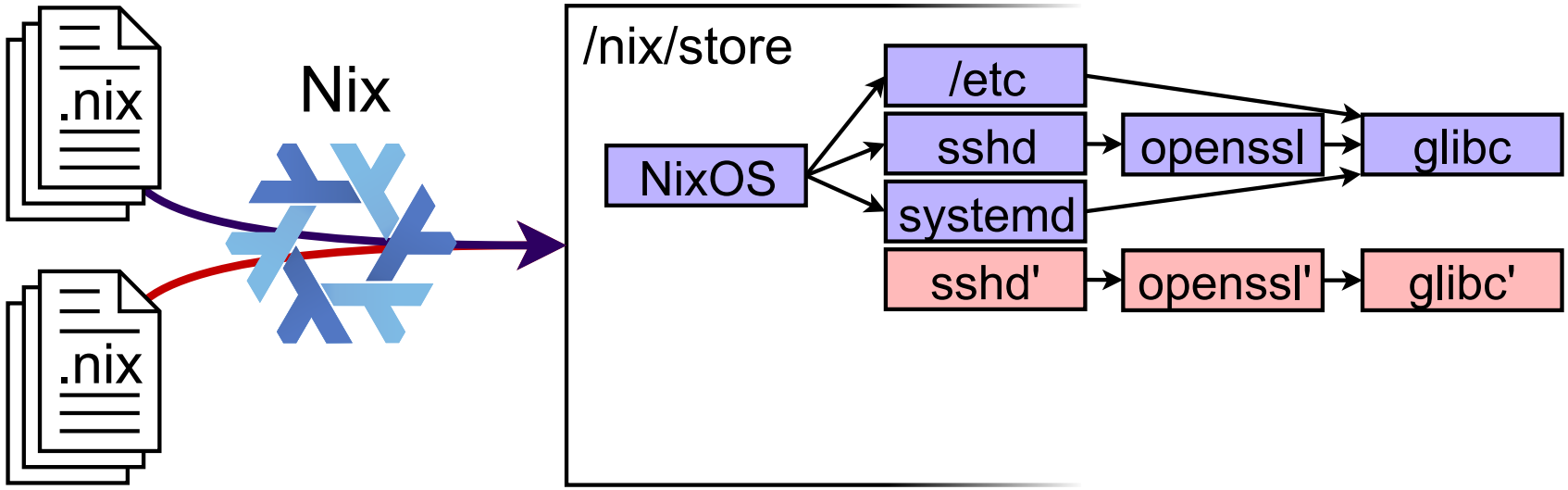
Differential Update Transfer



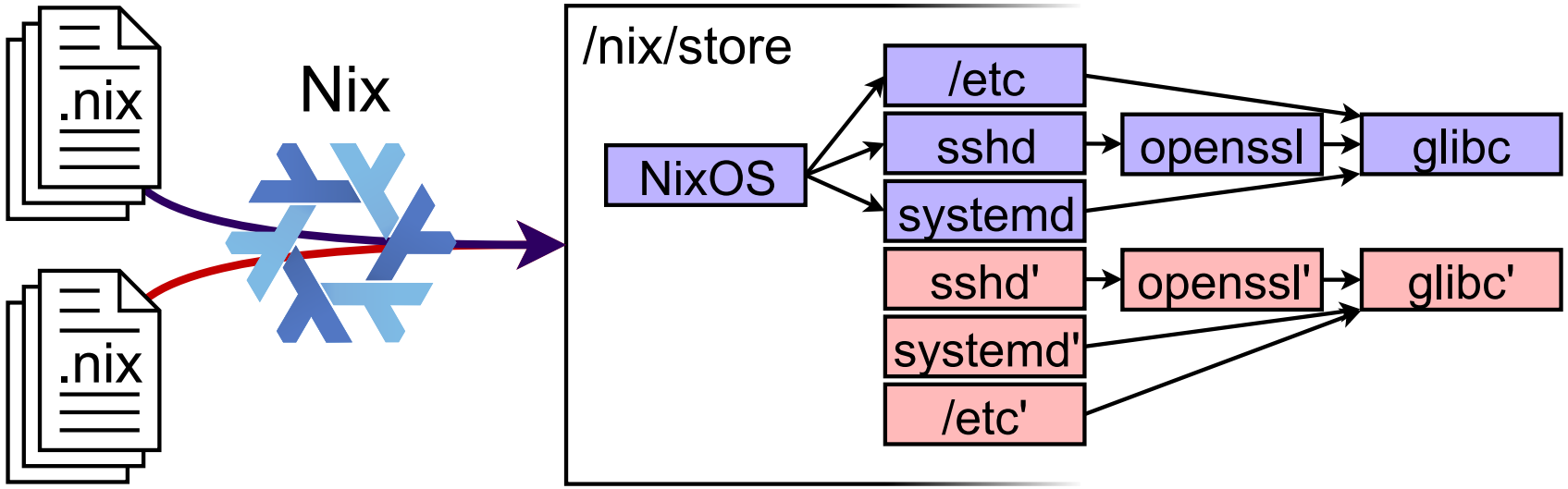
Differential Update Transfer



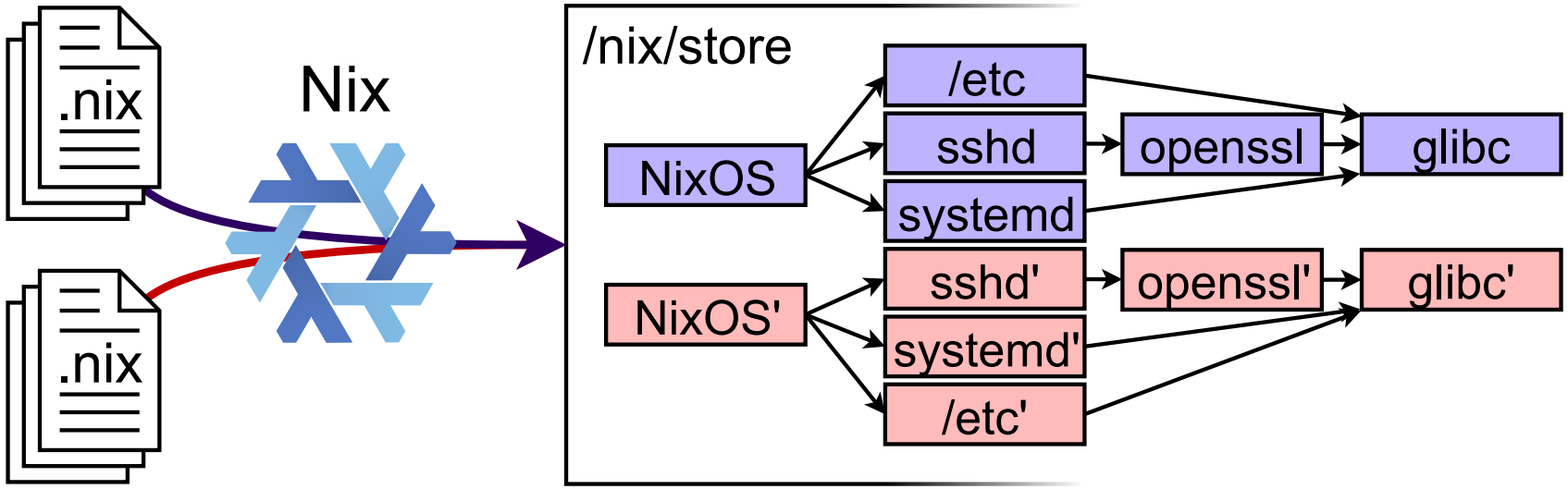
Differential Update Transfer



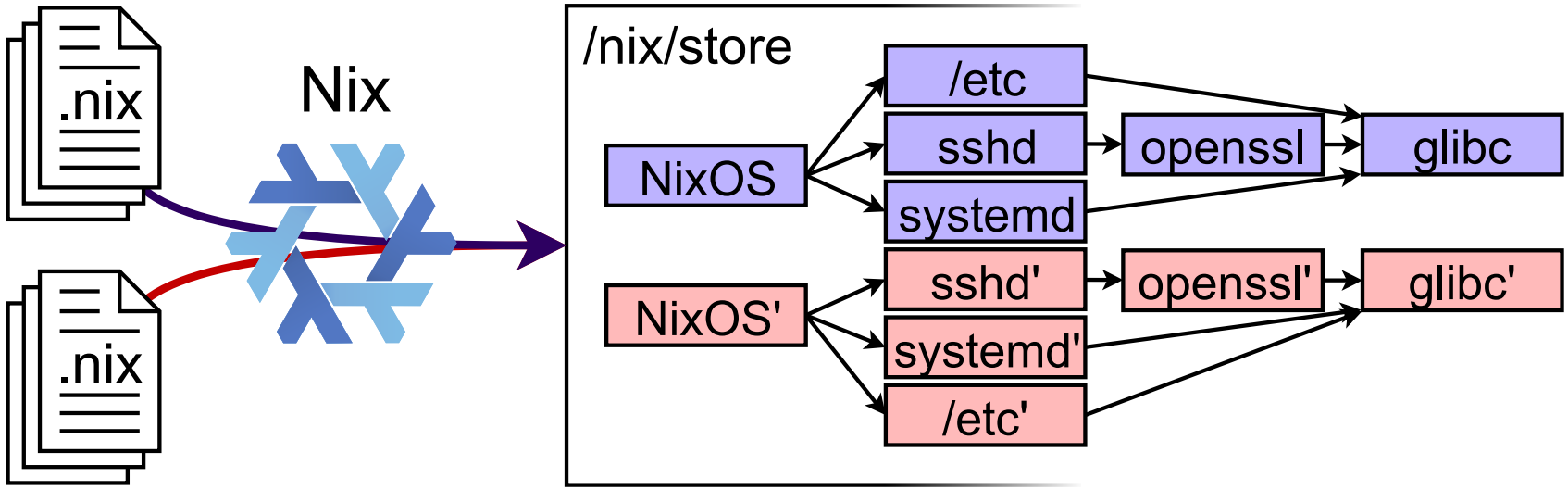
Differential Update Transfer



Differential Update Transfer

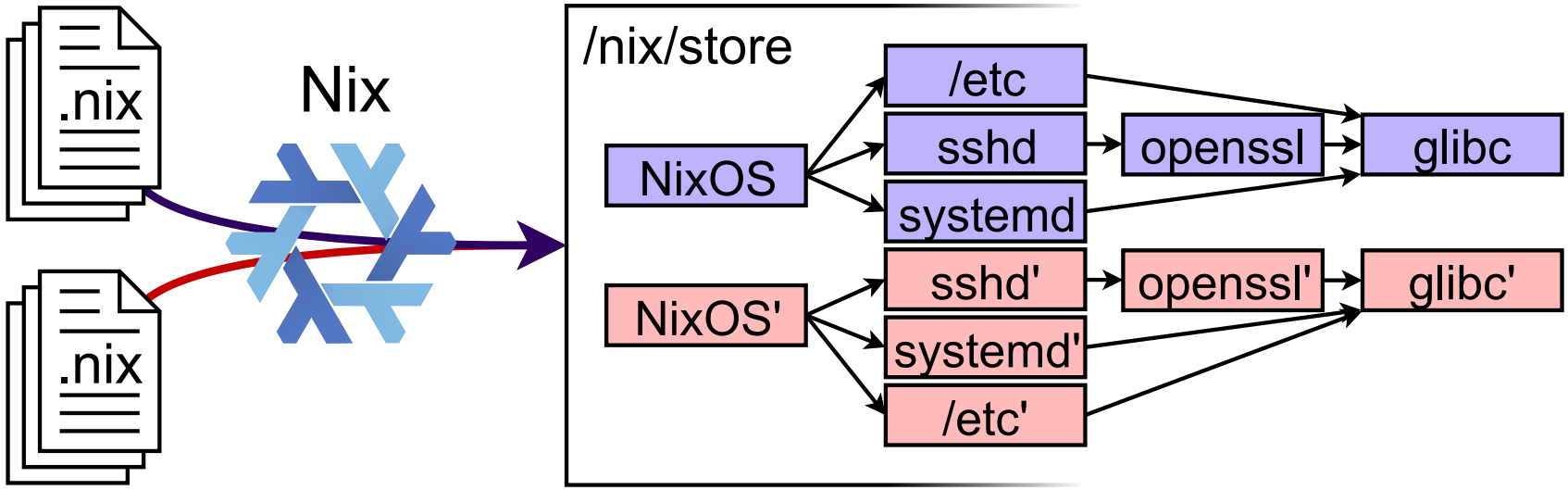


Differential Update Transfer



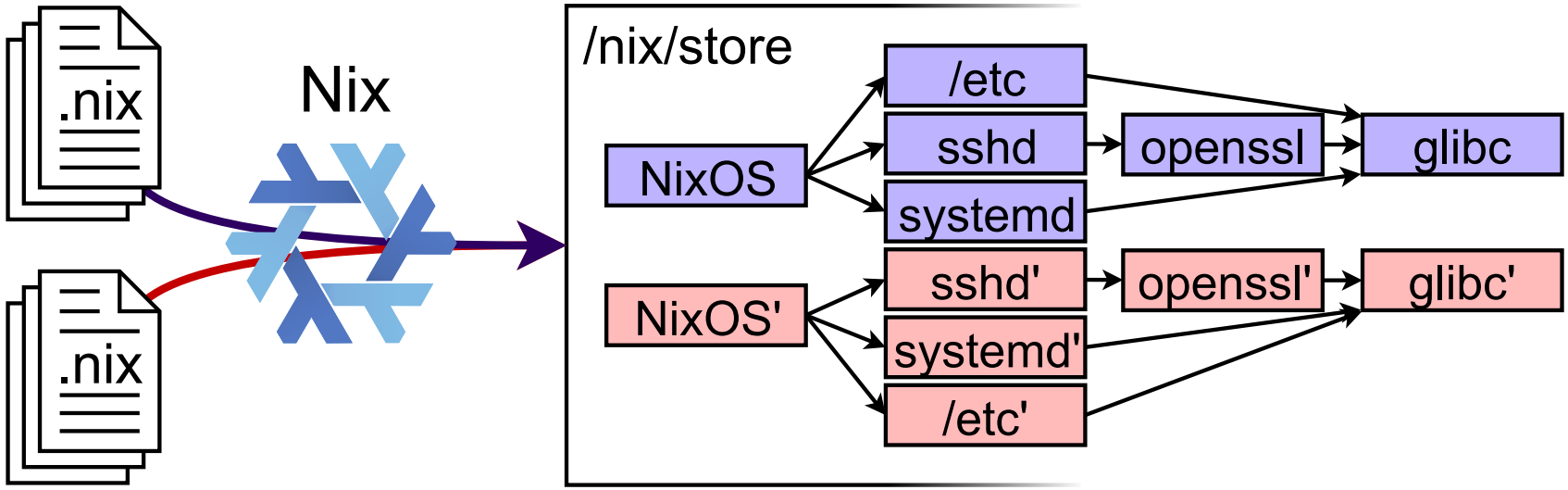
- Many/all new components, but very similar component exists

Differential Update Transfer



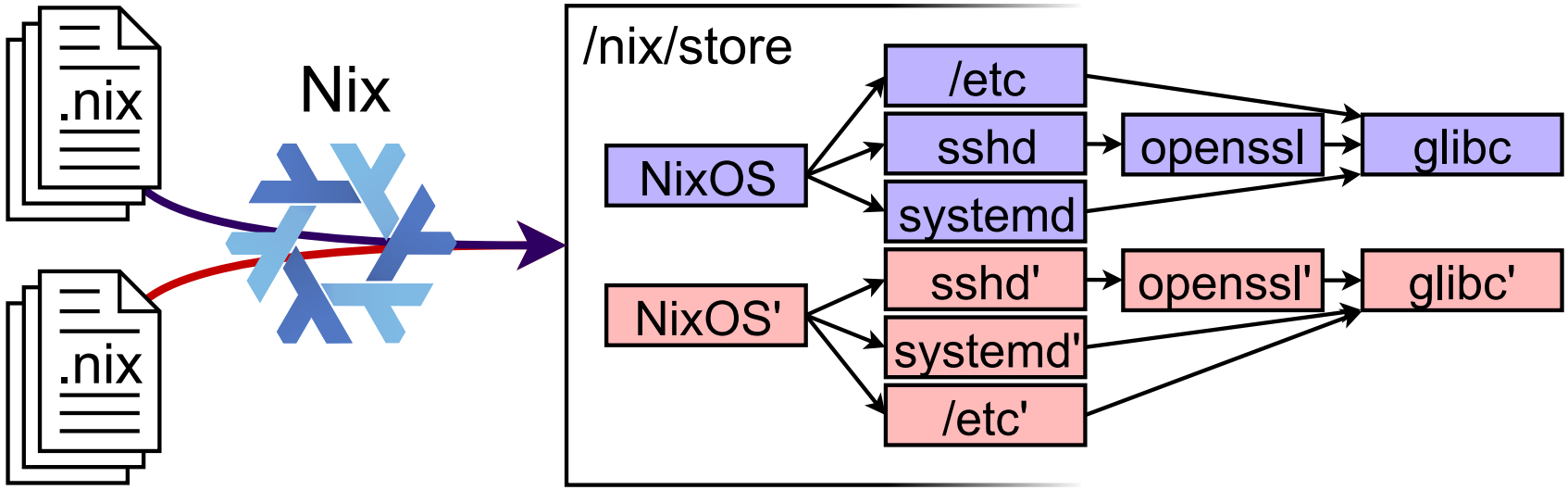
- Many/all new components, but very similar component exists
- Find “before” component, update differentially

Differential Update Transfer



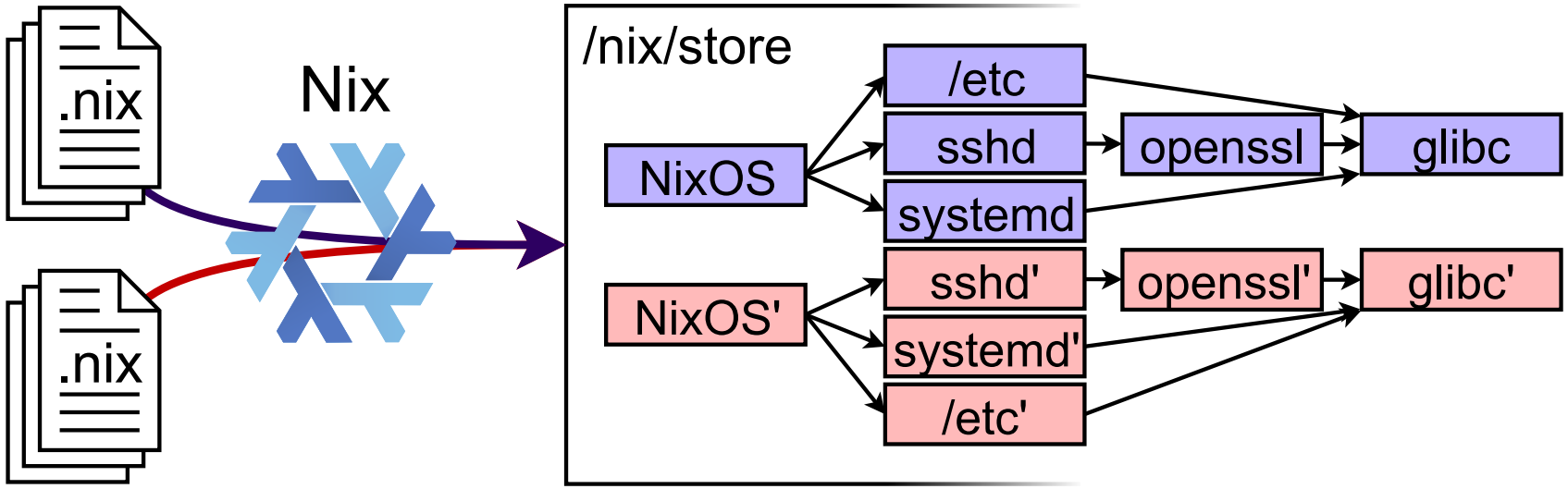
- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage

Differential Update Transfer



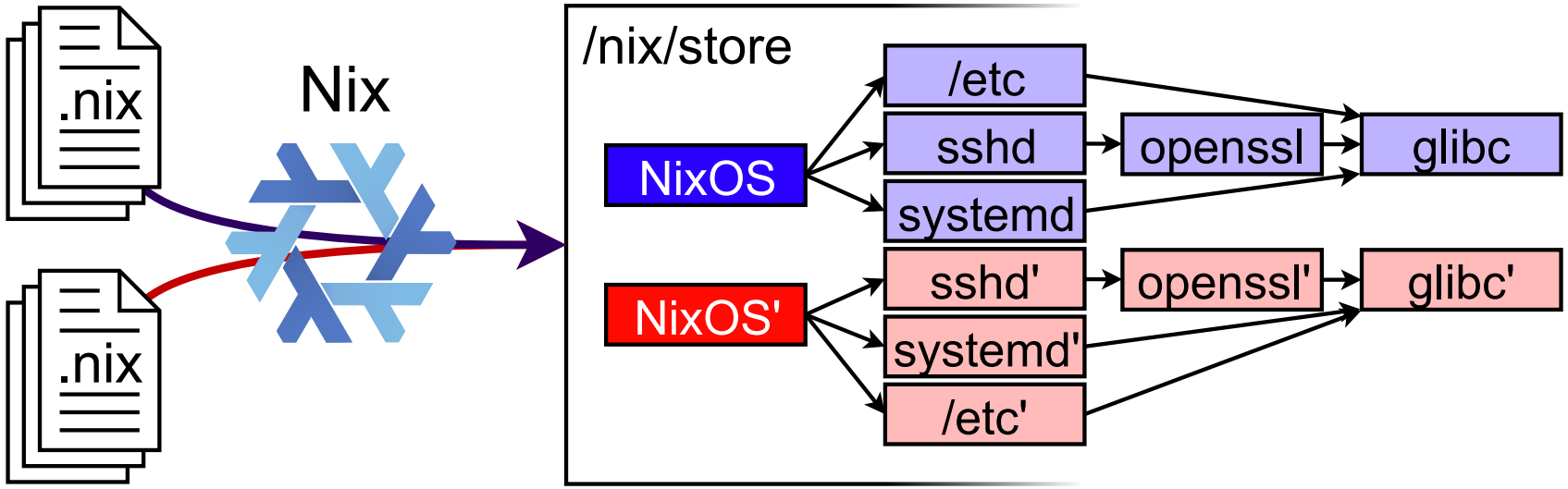
- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs

Differential Update Transfer



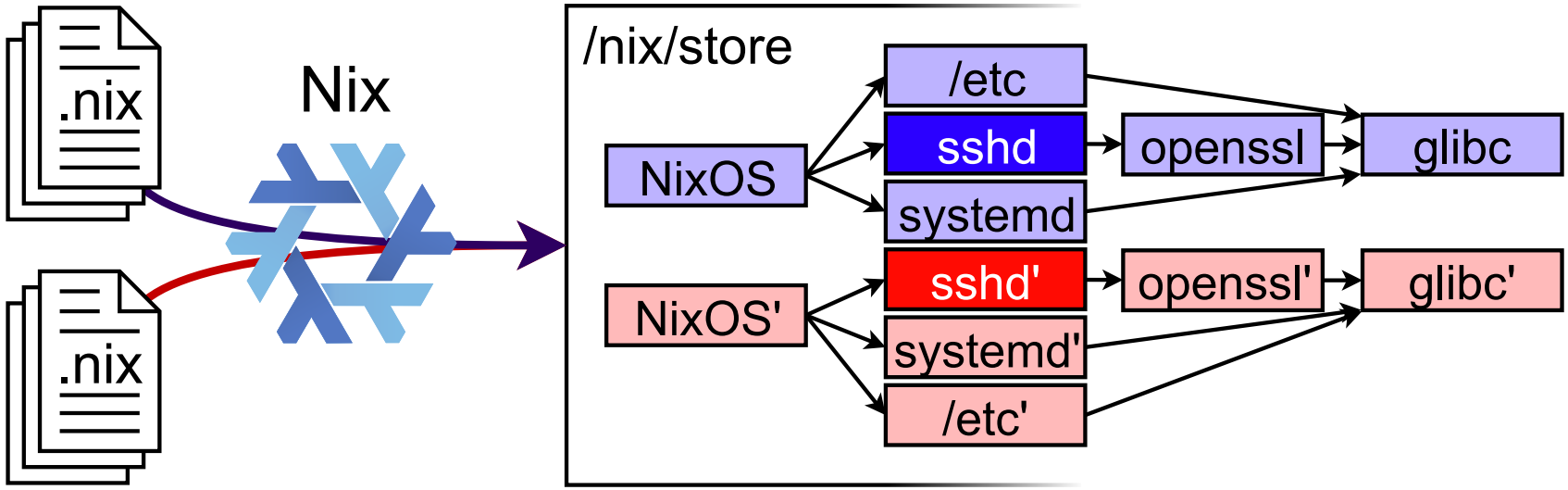
- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer:

Differential Update Transfer



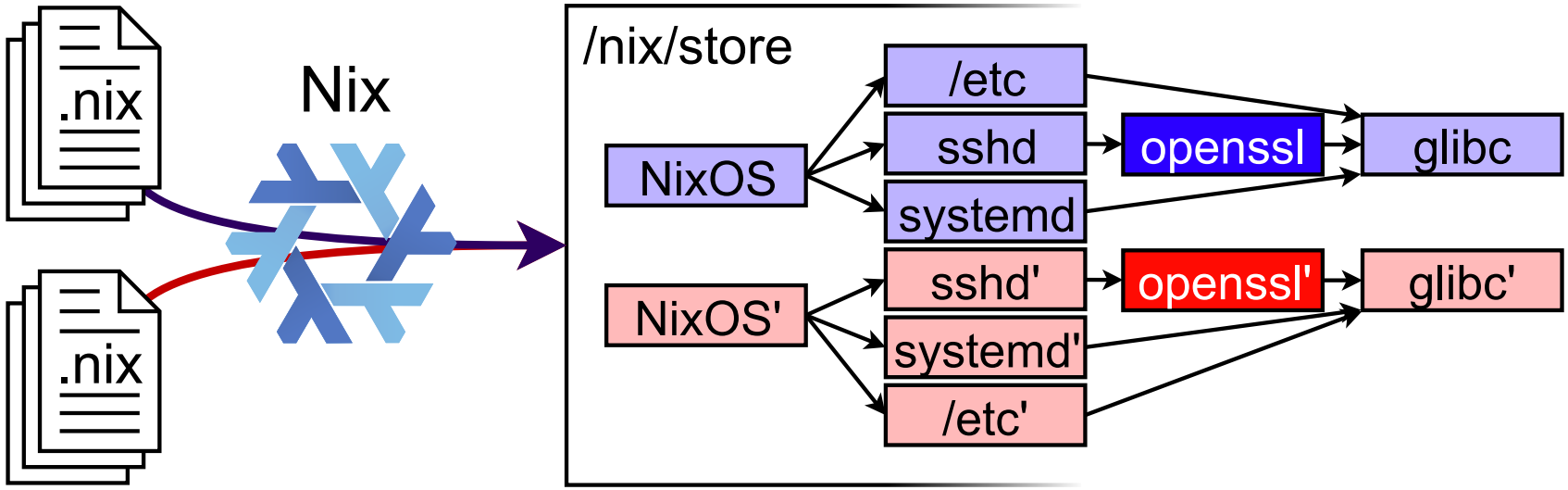
- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’),

Differential Update Transfer



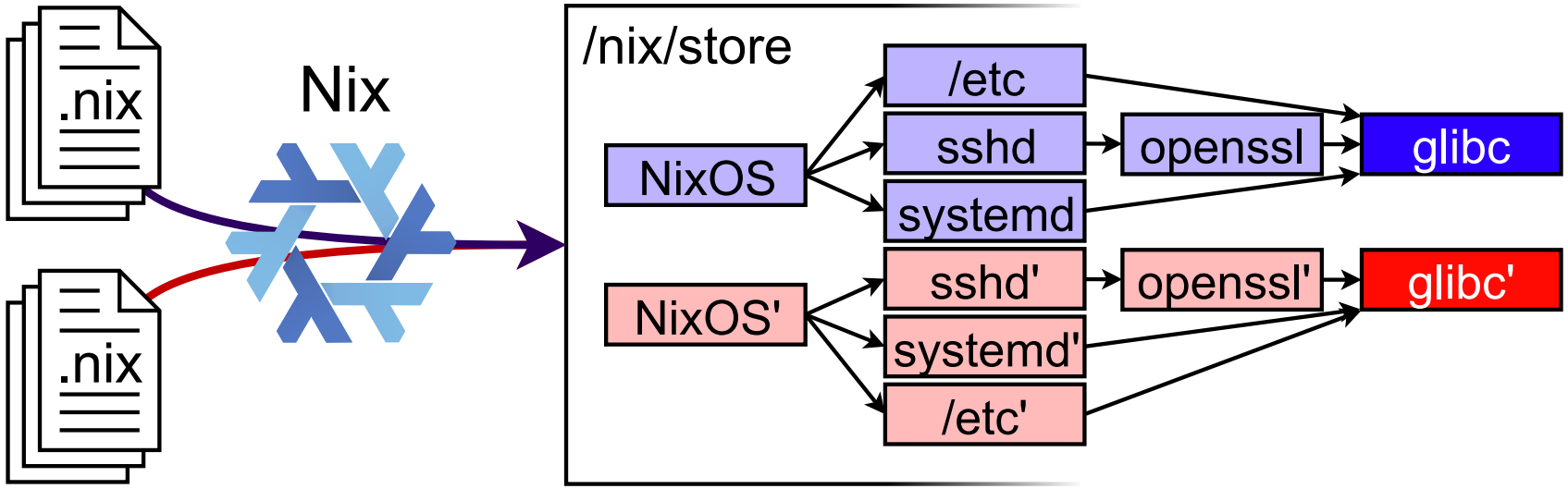
- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’), diff(sshd, sshd’),

Differential Update Transfer



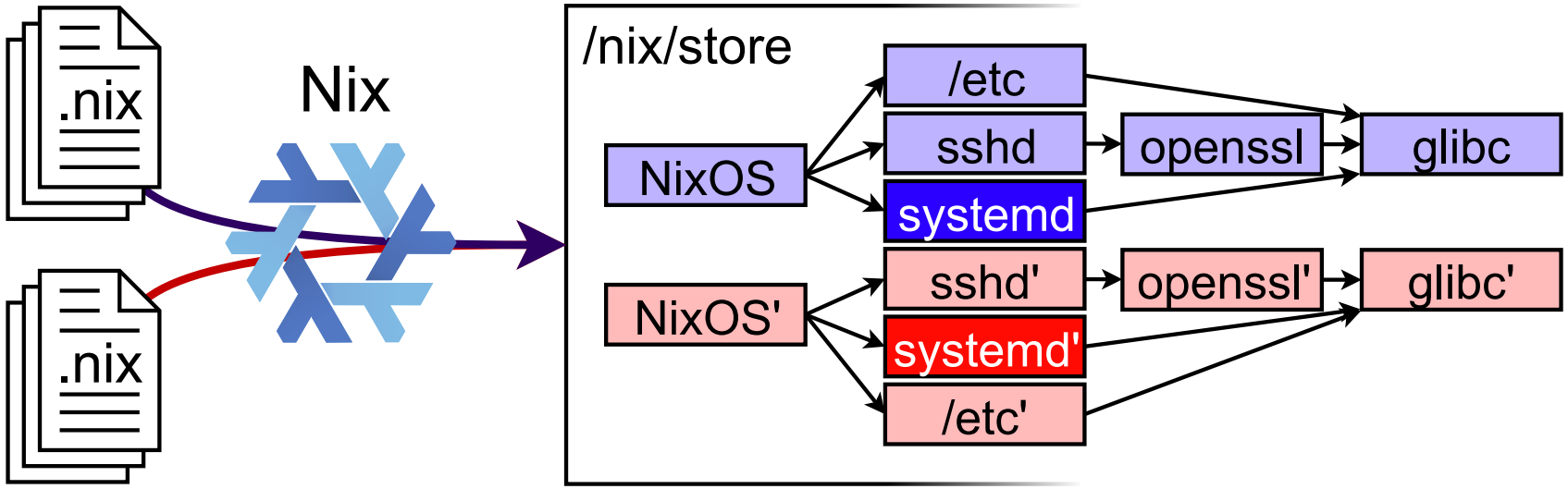
- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’), diff(sshd, sshd’), diff(openssl, openssl’),

Differential Update Transfer



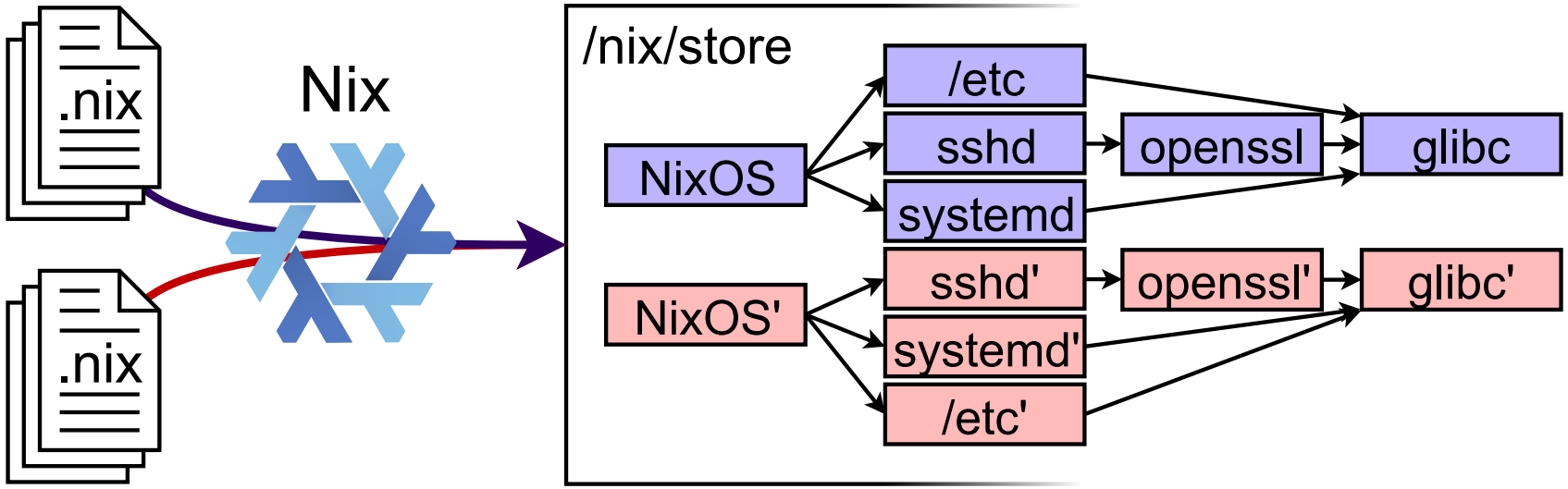
- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’), diff(sshd, sshd’), diff(openssl, openssl’), diff(glibc, glibc’),

Differential Update Transfer



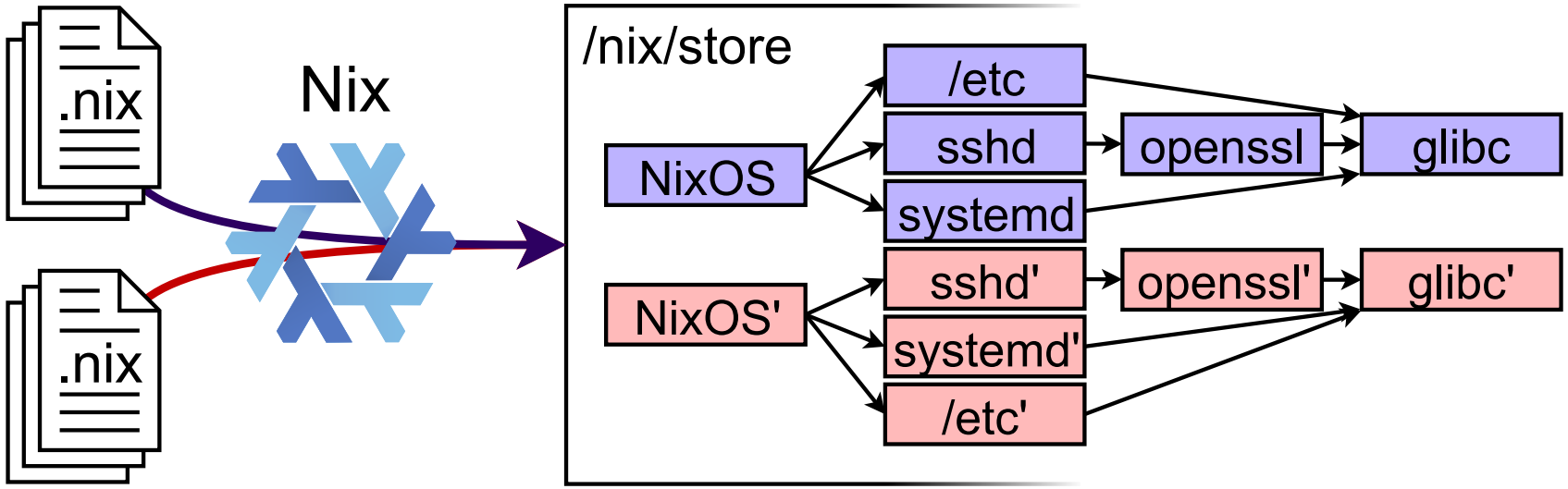
- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’), diff(sshd, sshd’), diff(openssl, openssl’), diff(glibc, glibc’), diff(systemd, systemd’),

Differential Update Transfer



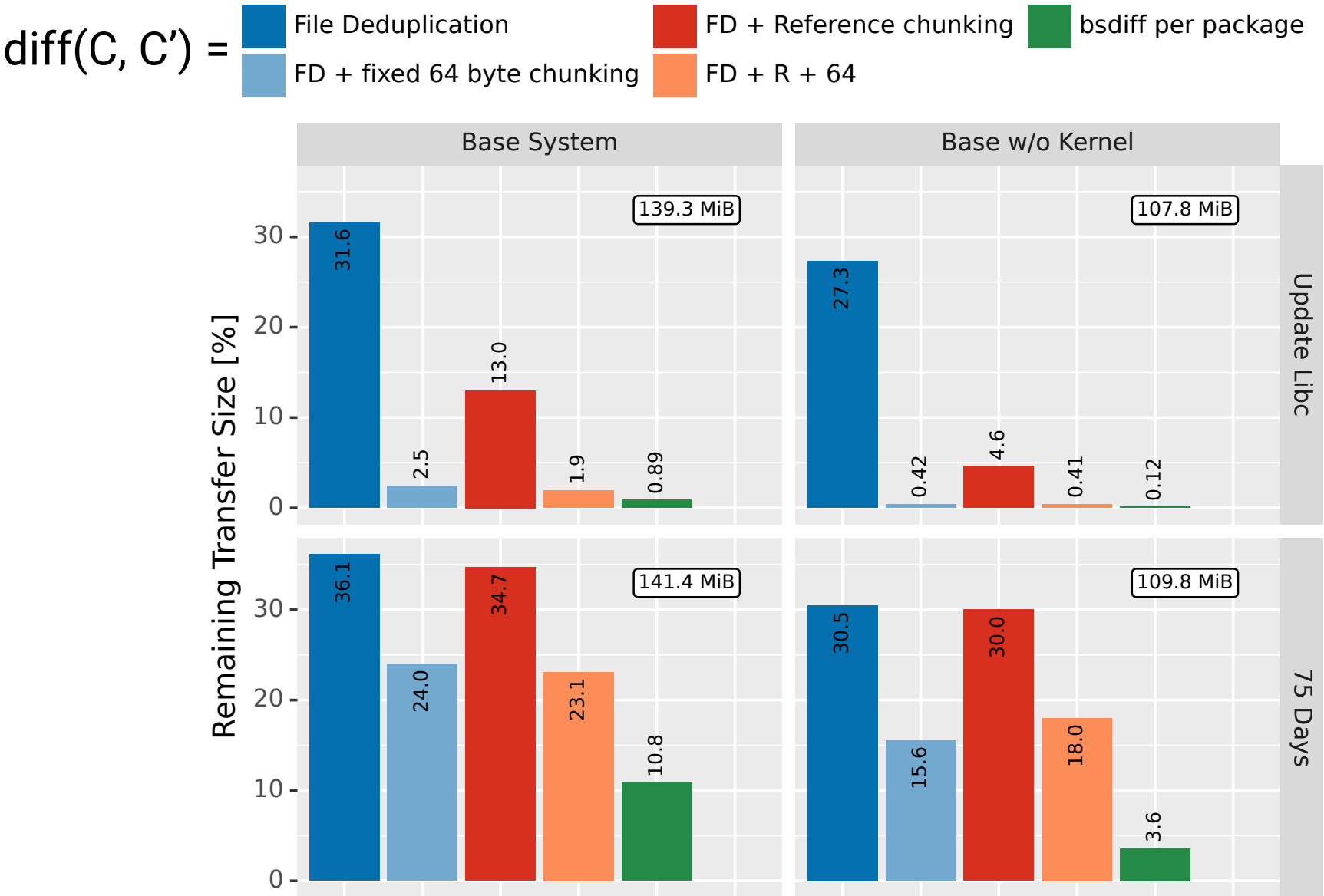
- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’), diff(sshd, sshd’), diff(openssl, openssl’), diff(glibc, glibc’), diff(systemd, systemd’), ...

Differential Update Transfer



- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’), diff(sshd, sshd’), diff(openssl, openssl’), diff(glibc, glibc’), diff(systemd, systemd’), ...
- “diff(C, C’)” by sender (knows C & C’) ⇒ unidirectional transfer

Differential Update Transfer: Results



Atomic Bootloader Updates

TUHH

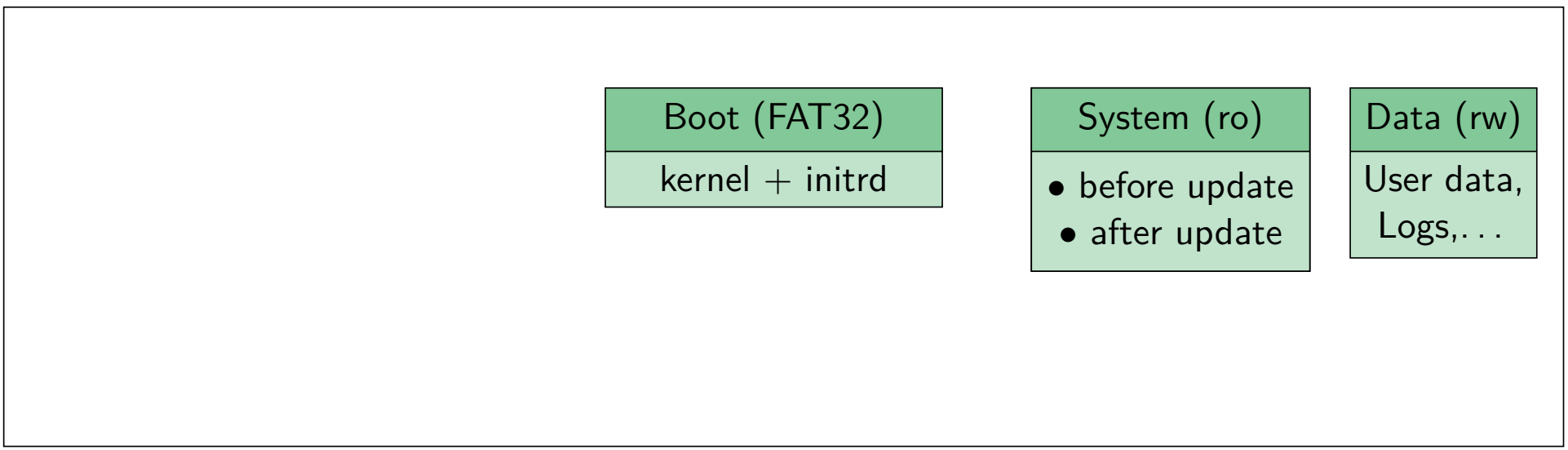
- **Goal**: safe/atomic updates for NixOS on any hardware/firmware

Atomic Bootloader Updates

TUHH

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility

Flash Disk

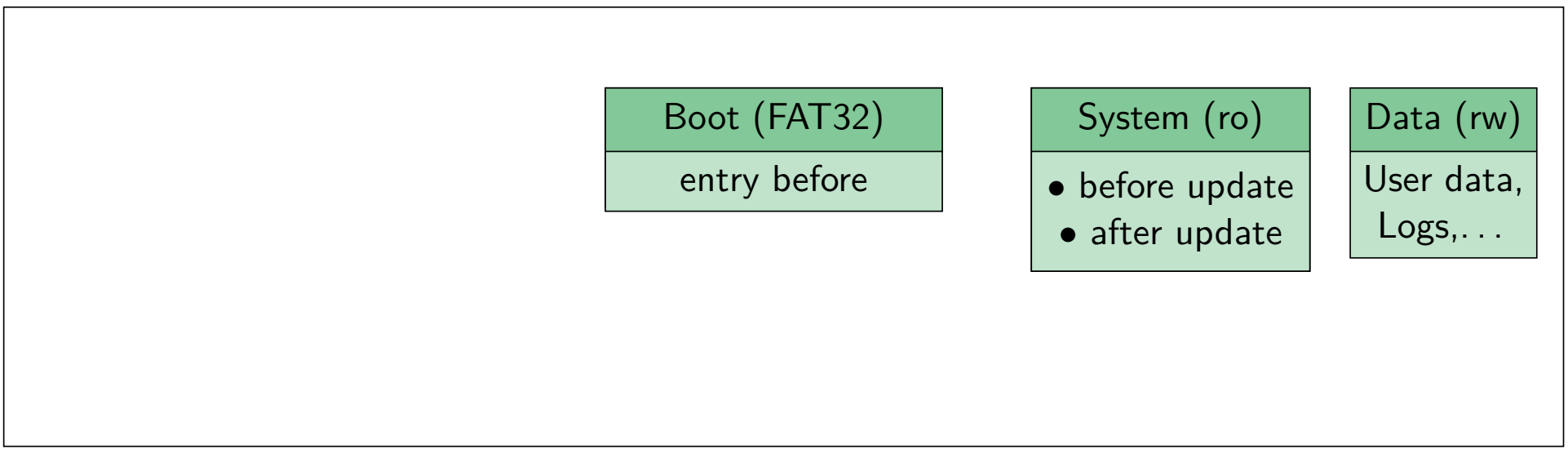


Atomic Bootloader Updates

TUHH

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility

Flash Disk

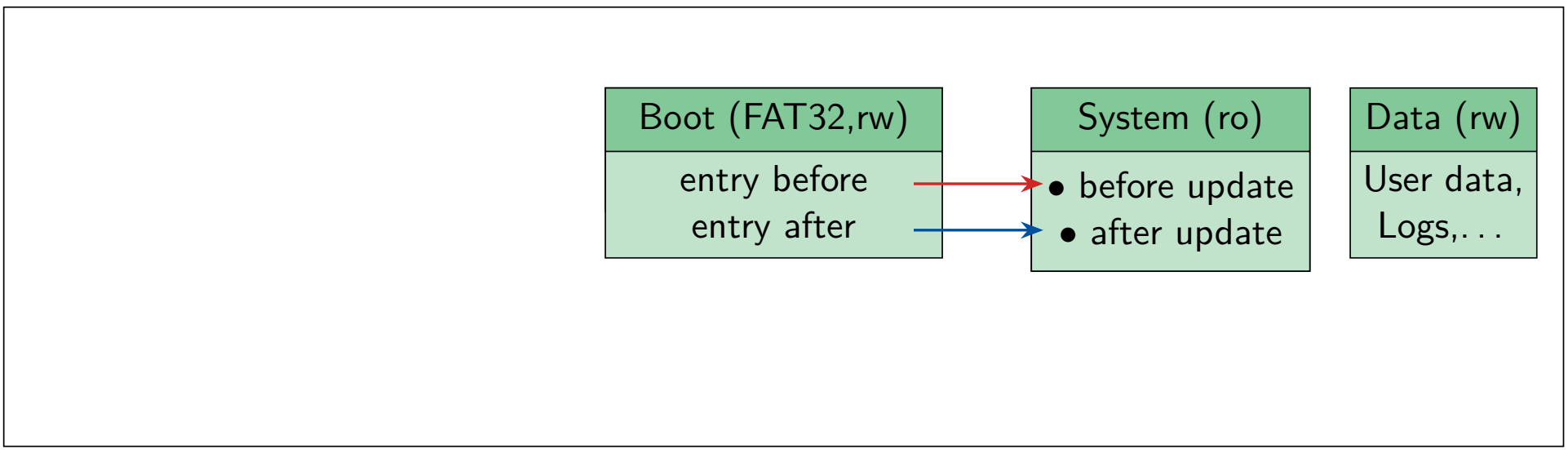


Atomic Bootloader Updates

TUHH

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility

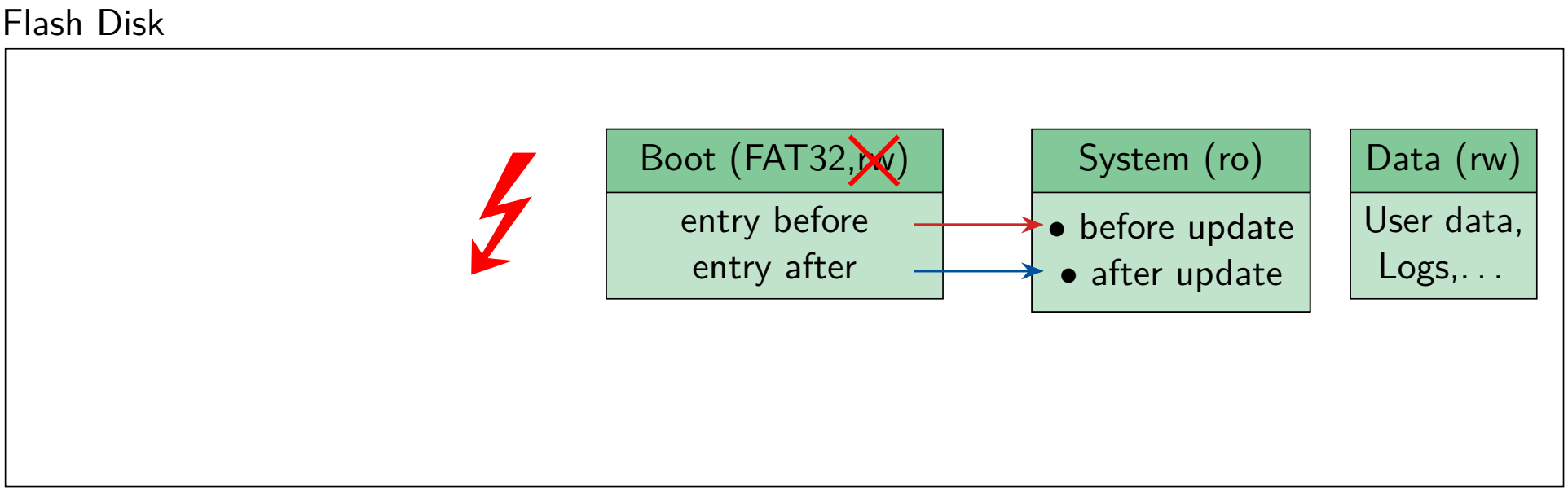
Flash Disk



Atomic Bootloader Updates

TUHH

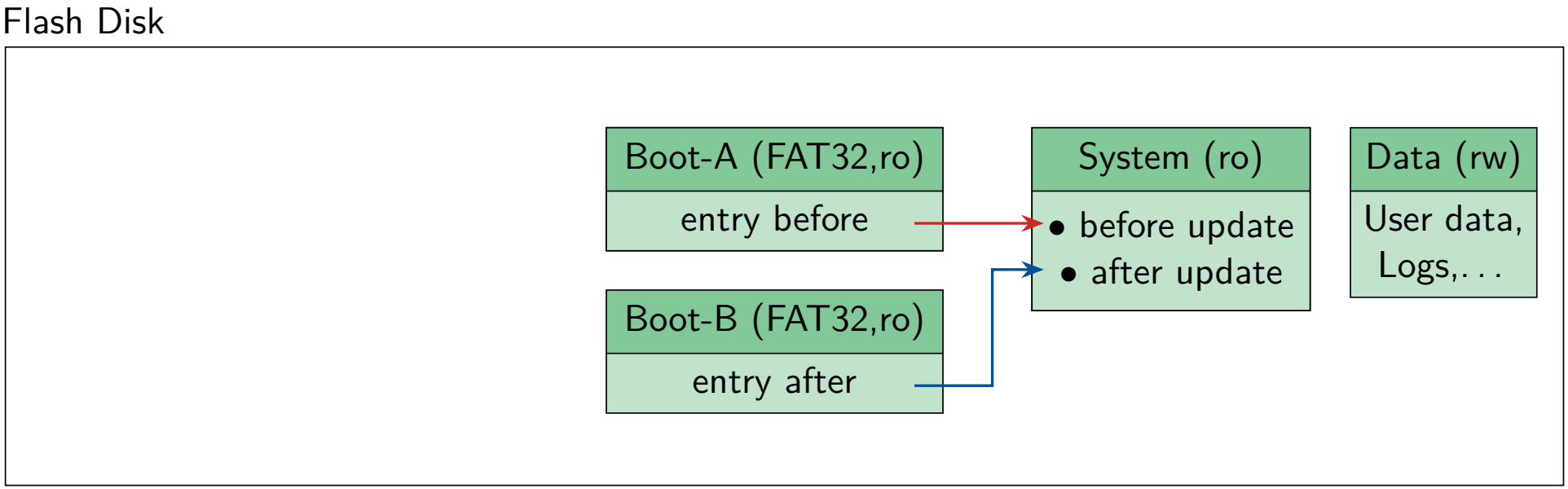
- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility but unreliable



Atomic Bootloader Updates

TUHH

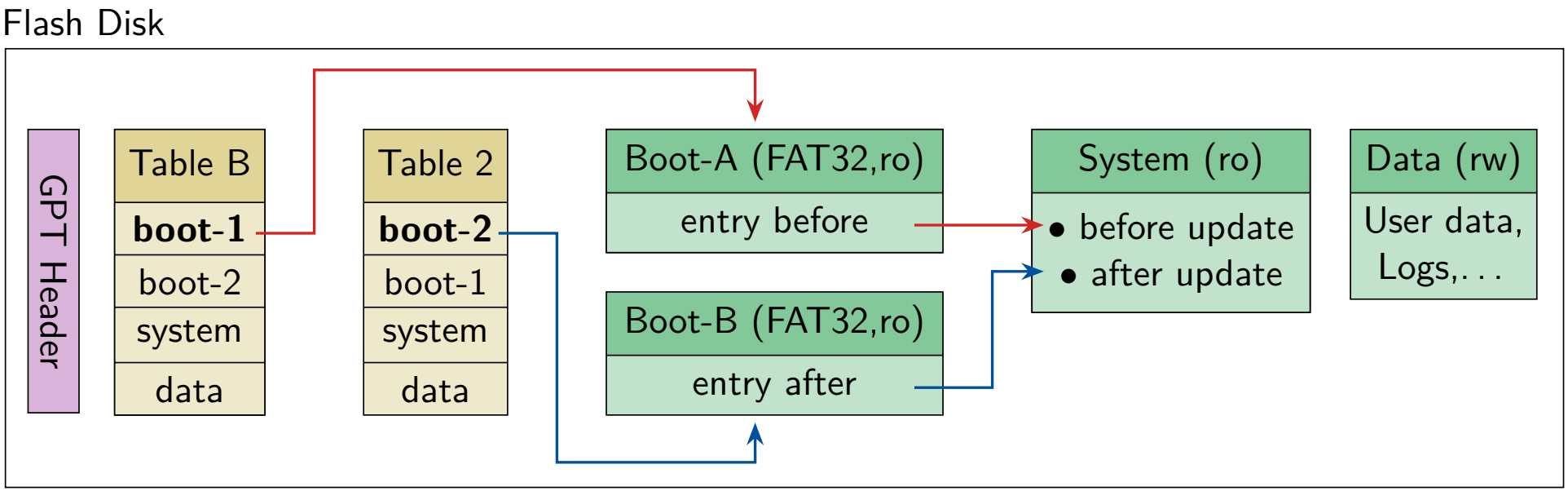
- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility but unreliable
- Solution: A/B partitioning for Boot



Atomic Bootloader Updates

TUHH

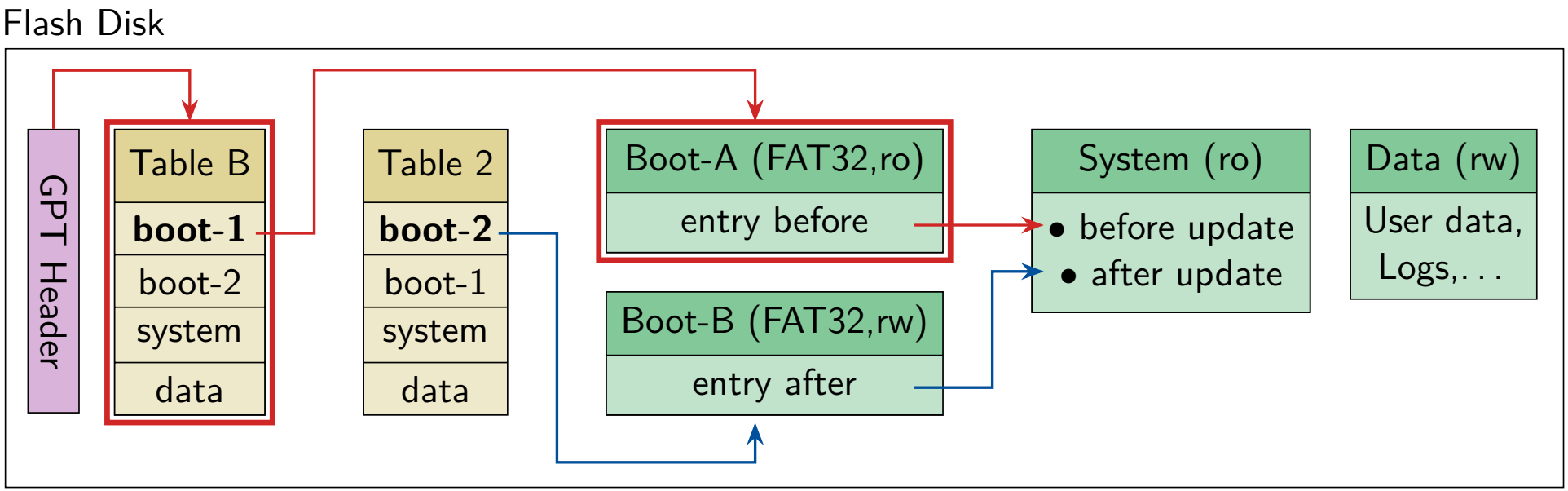
- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility but unreliable
- Solution: A/B partitioning for Boot



Atomic Bootloader Updates

TUHH

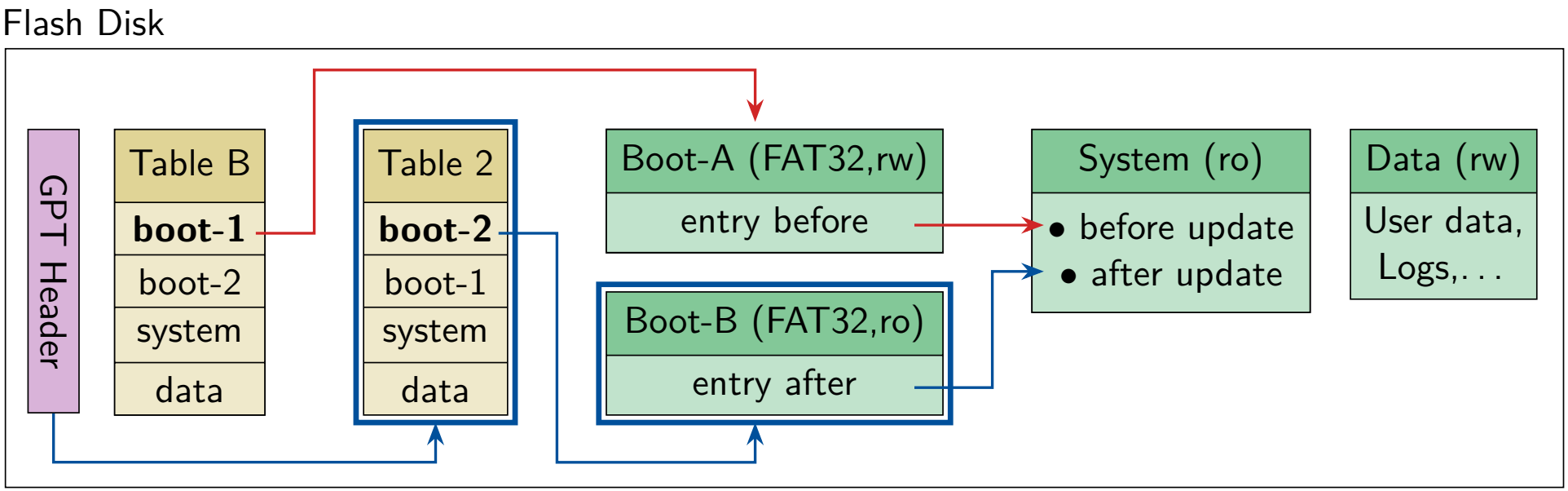
- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility but unreliable
- Solution: A/B partitioning for Boot switched via GPT



Atomic Bootloader Updates

TUHH

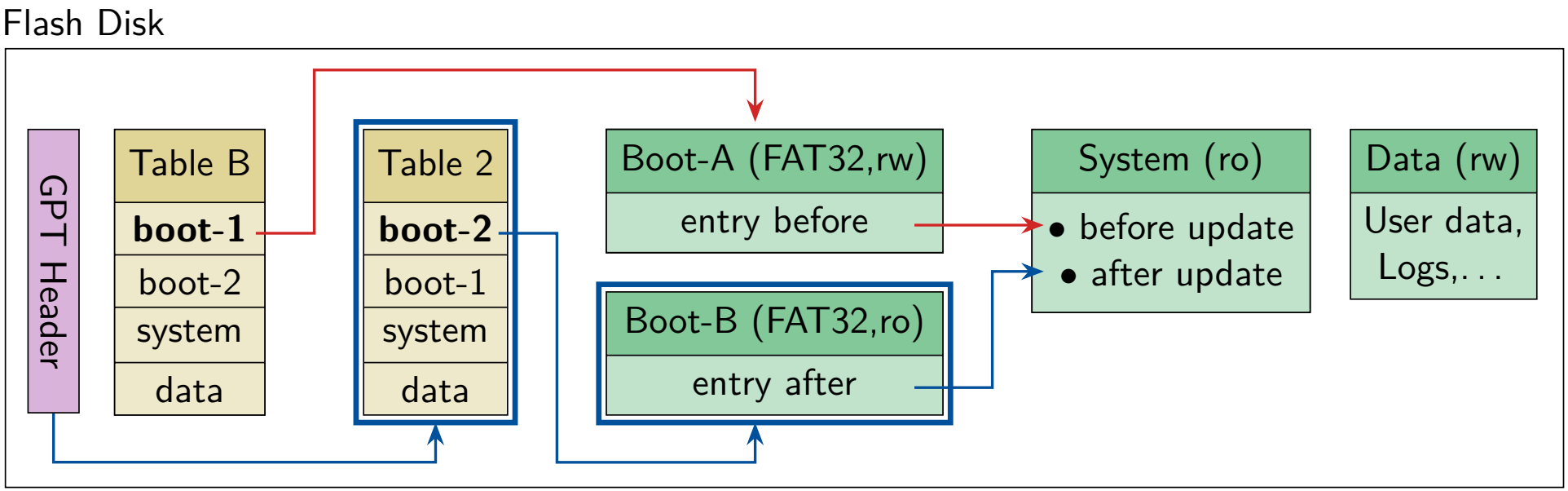
- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility but unreliable
- Solution: A/B partitioning for Boot switched via GPT



Atomic Bootloader Updates

TUHH

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility but unreliable
- Solution: A/B partitioning for Boot switched via GPT



⇒ applicable to other operating systems!

Reconfiguration

TUHH

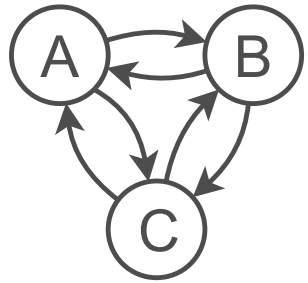
- Change system’s executed software configuration

- Change system's executed software configuration

Update:

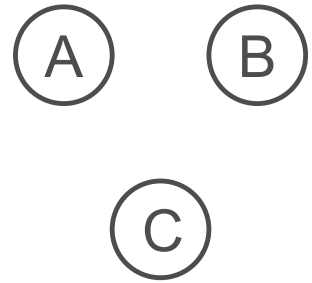
- Change system's executed software configuration

Update:

Reconfigure: 

- Change system's executed software configuration

Update:

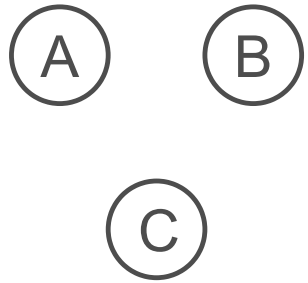
Reconfigure: 

Reconfiguration

TUHH

- Change system's executed software configuration

Update:

Reconfigure: 

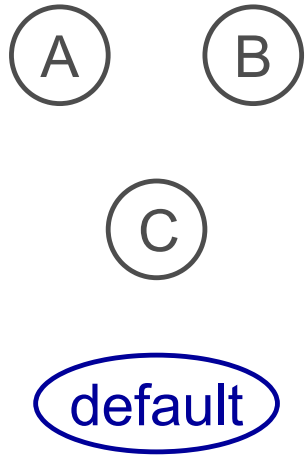
- NixOS: multiple variants ("profiles")

Reconfiguration

TUHH

- Change system's executed software configuration

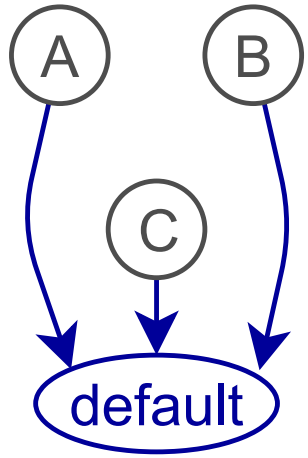
Update:

Reconfigure: 

- NixOS: multiple variants ("profiles")

- Change system's executed software configuration

Update:

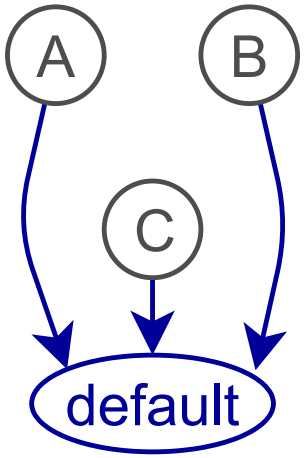
Reconfigure: 

- NixOS: multiple variants ("profiles")

- Change system's executed software configuration

Update:

Reconfigure:

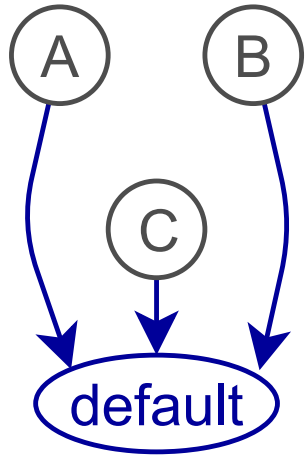


- NixOS: multiple variants ("profiles")
- reUpNix: reconfig ⇒ reboot

Reconfiguration

- Change system's executed software configuration

Update:

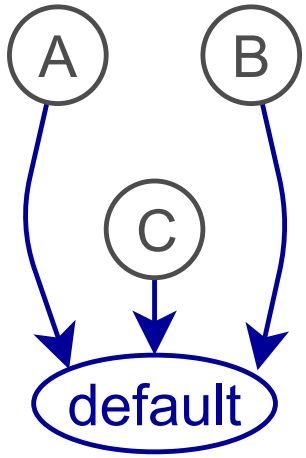
Reconfigure: 

- NixOS: multiple variants ("profiles")
- reUpNix: reconfig ⇒ reboot
  - Correctness: independent of previous profile

- Change system's executed software configuration

Update:

Reconfigure:

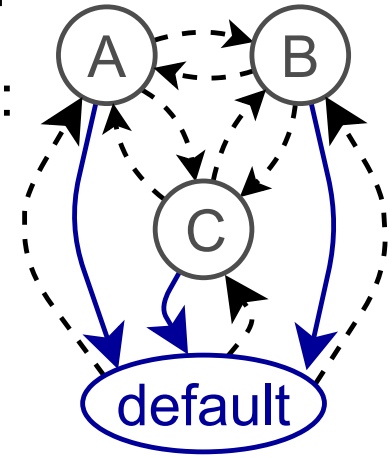


- NixOS: multiple variants ("profiles")
- reUpNix: reconfig ⇒ reboot
  - Correctness: independent of previous profile
  - Completeness: e.g. change kernel

Reconfiguration

- Change system's executed software configuration

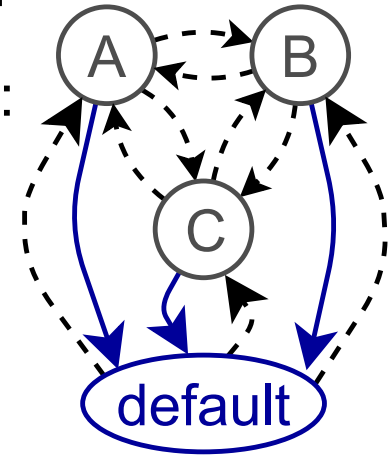
Update:

Reconfigure: 

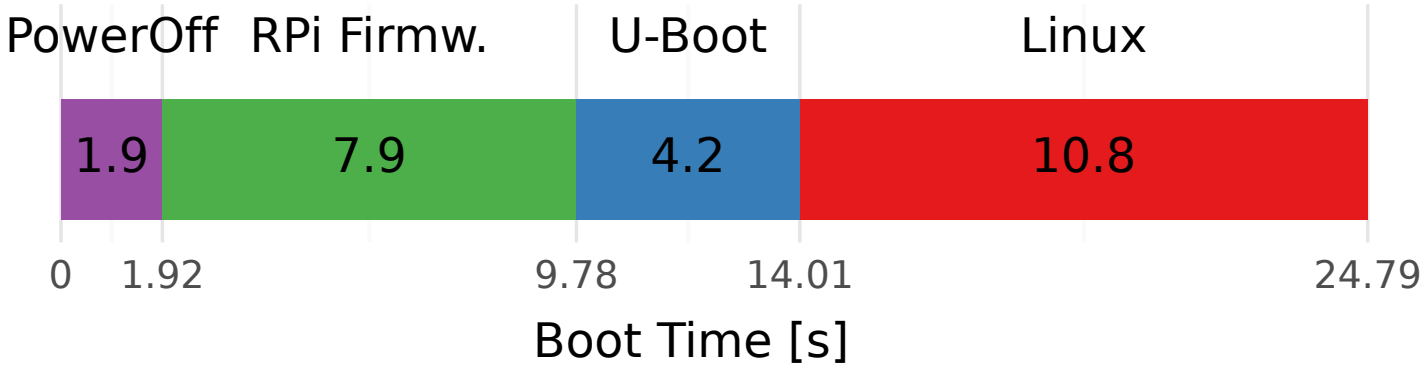
- NixOS: multiple variants ("profiles")
- reUpNix: reconfig ⇒ reboot
  - Correctness: independent of previous profile
  - Completeness: e.g. change kernel

- Change system's executed software configuration

Update:

Reconfigure: 

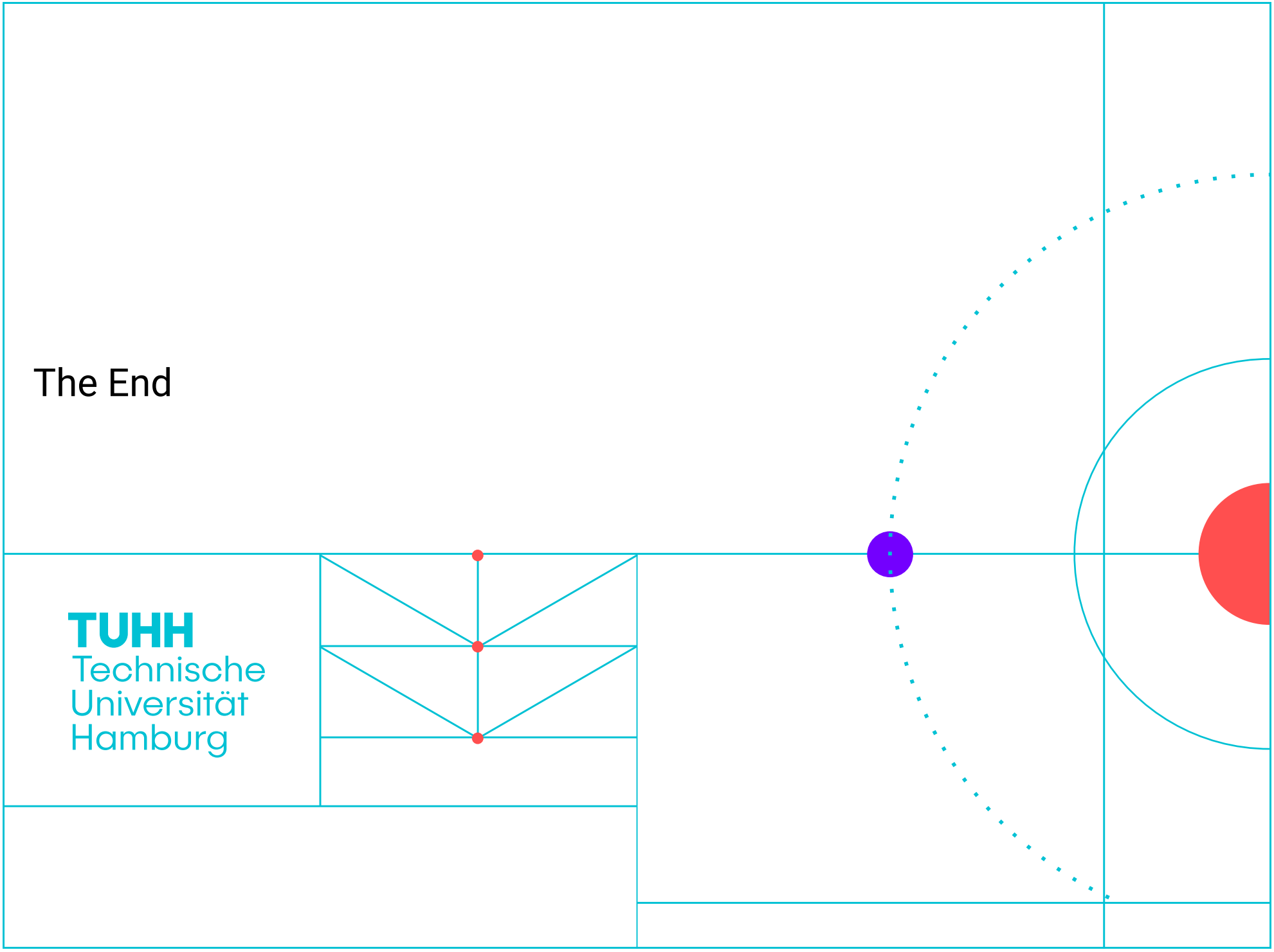
- NixOS: multiple variants ("profiles")
- reUpNix: reconfig ⇒ reboot
  - Correctness: independent of previous profile
  - Completeness: e.g. change kernel

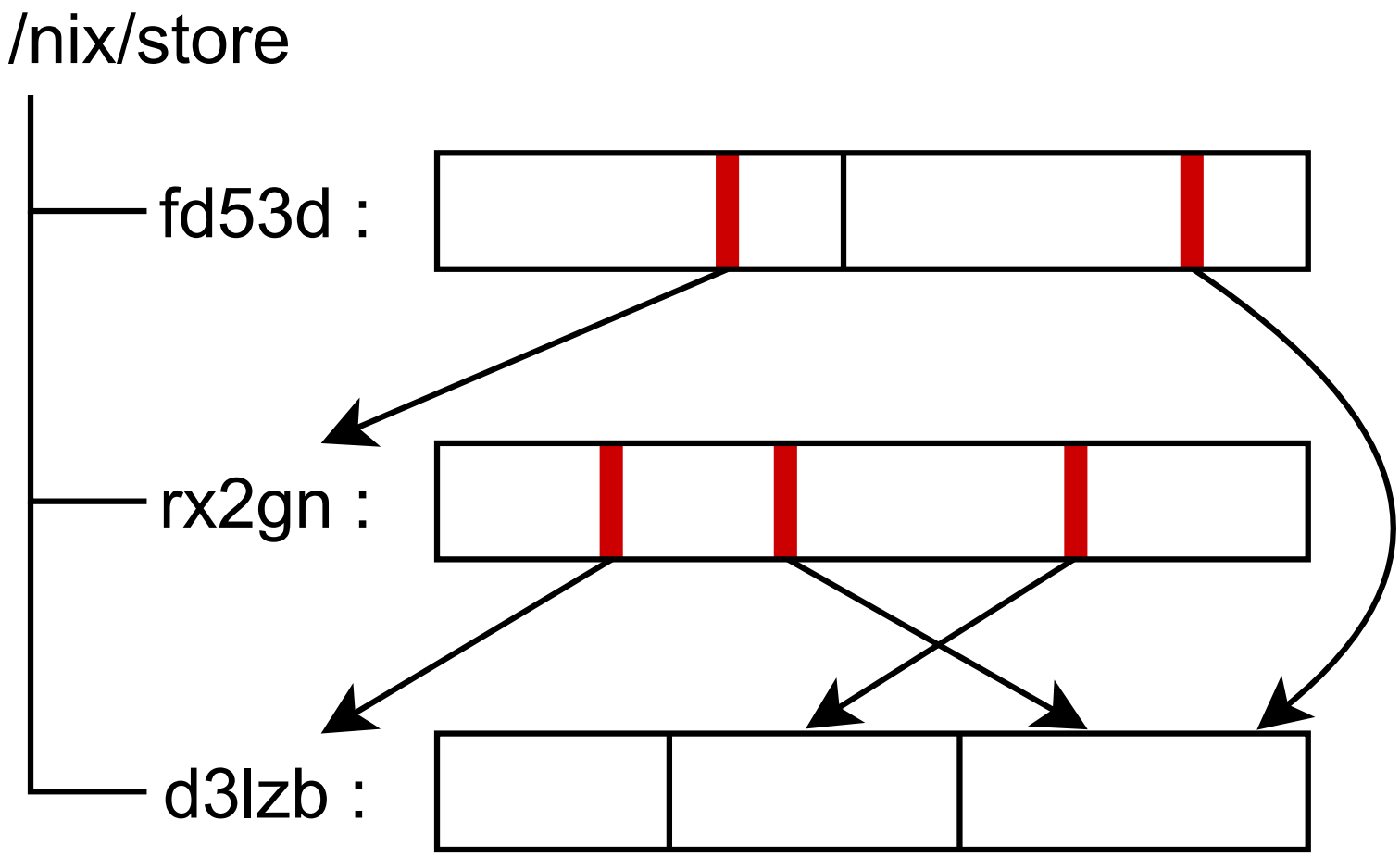


Summary

- NixOS: flexible, reproducible, predictable, reconfigurable
  - Not suited for embedded
- Solved NixOS problems for embedded
  - Reduced system size (~1.1 GiB → 155 MiB)
  - Updates up to 99.88 % smaller (mitigate change amplification)
  - Atomic bootloader config update (switch A/B part via GPT)
- reUpNix key features
  - Reconfiguration / multi-system setup
  - Efficient OCI services (w/o Docker, file-deduplicated)
  - Unidirectional communication update transfers

The End

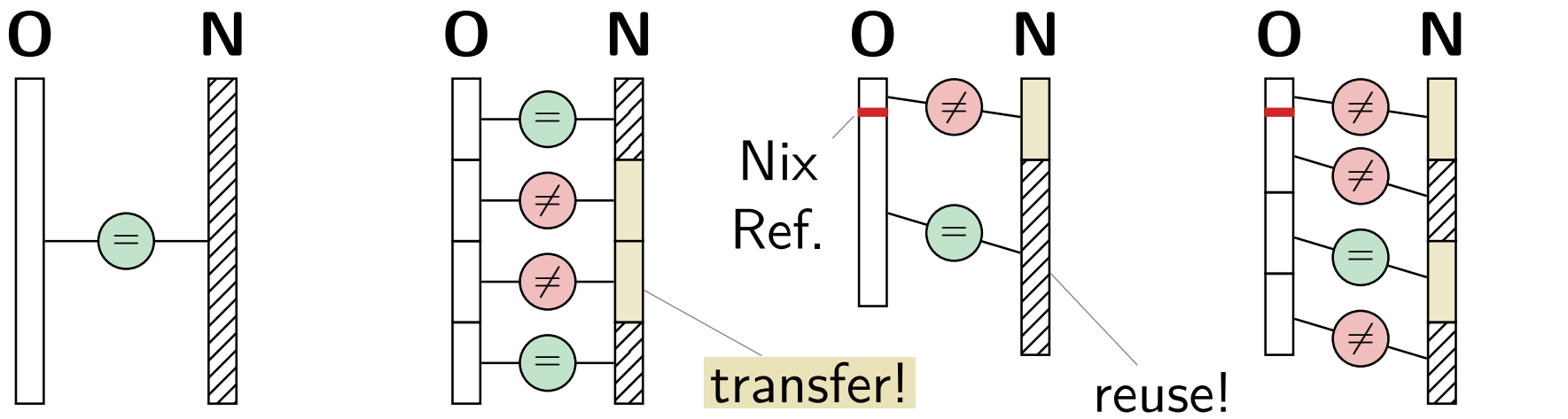



/nix/store
fd53d :
rx2gn :
d3lzb :

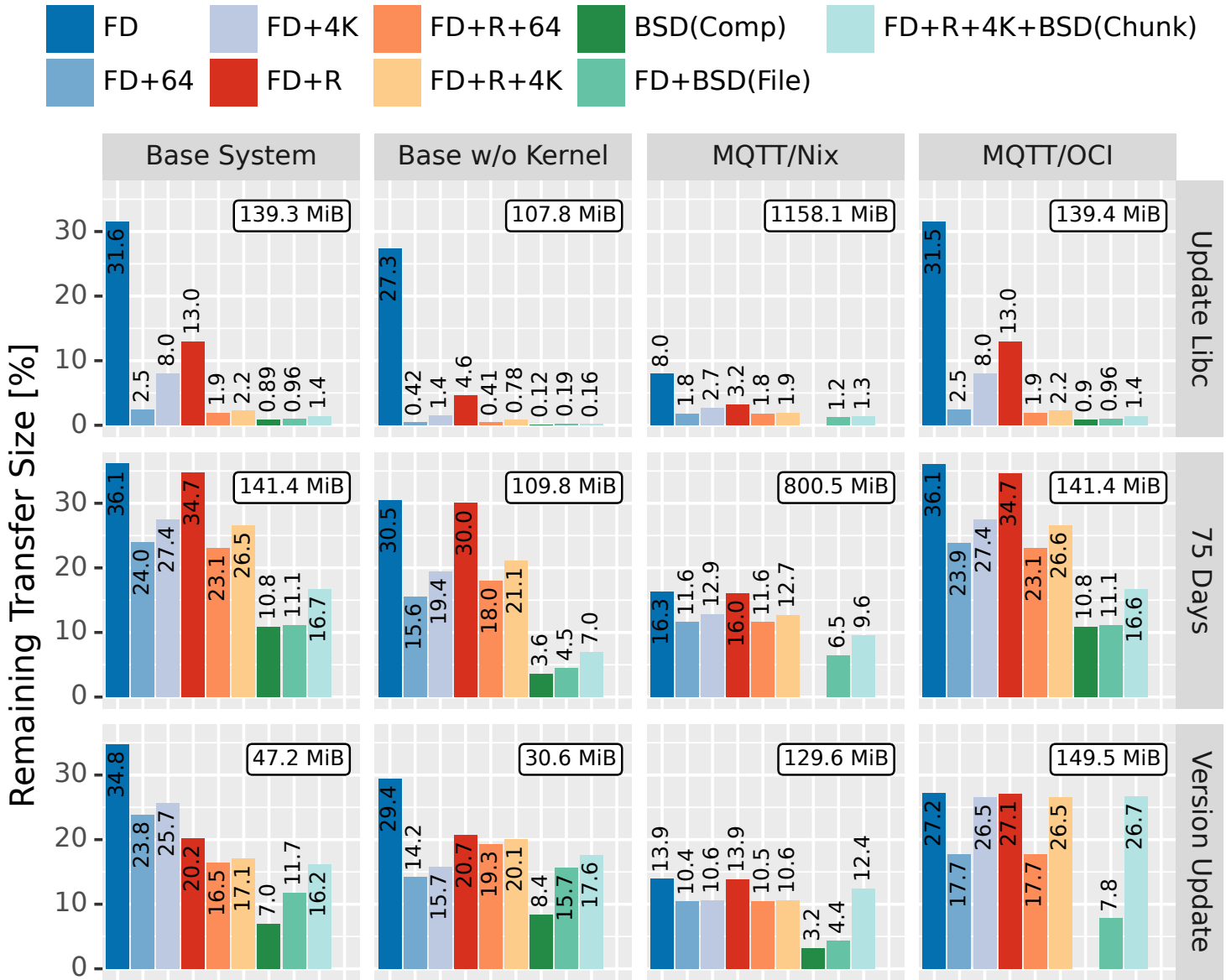
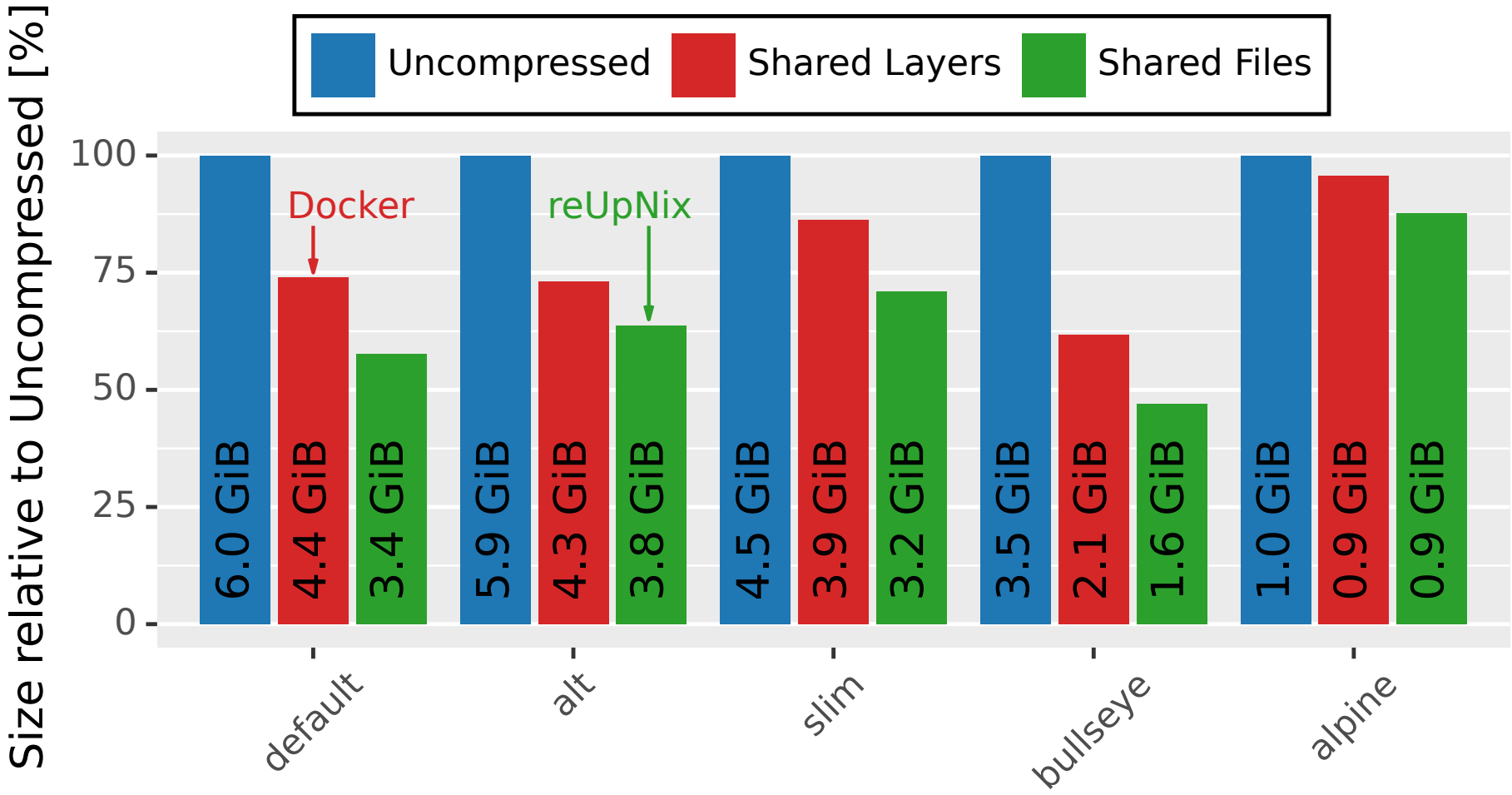
Update Deduplication

TUHH

(a) File Dedup. (b) Blocks (c) Refs (d) Ref+Block



Update Transfer Size



OCI Container Storage Cost