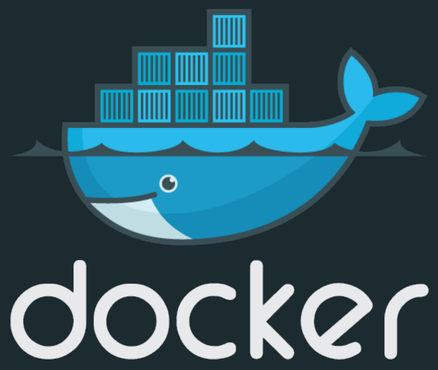
# When the going gets tough, Get TUF going!

Lily Guo - lily@docker.com
Riyaz Faizullabhoy - riyaz@docker.com / @riyazdf



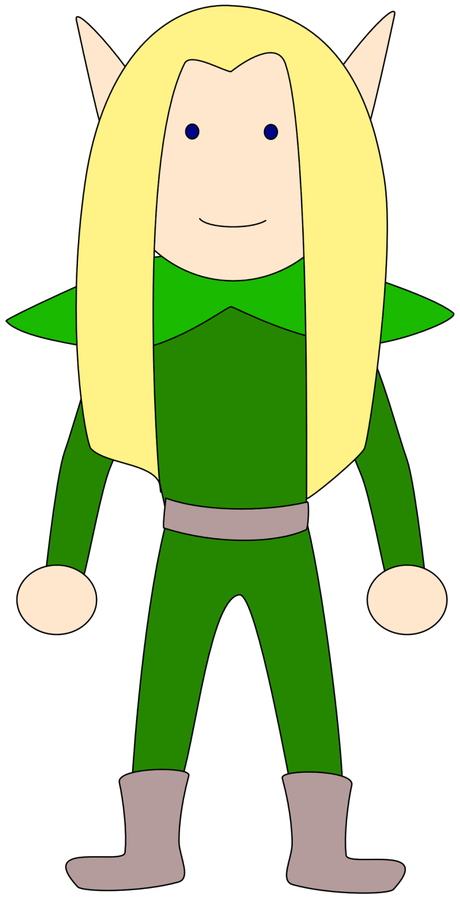
Motivation

What is TUF?

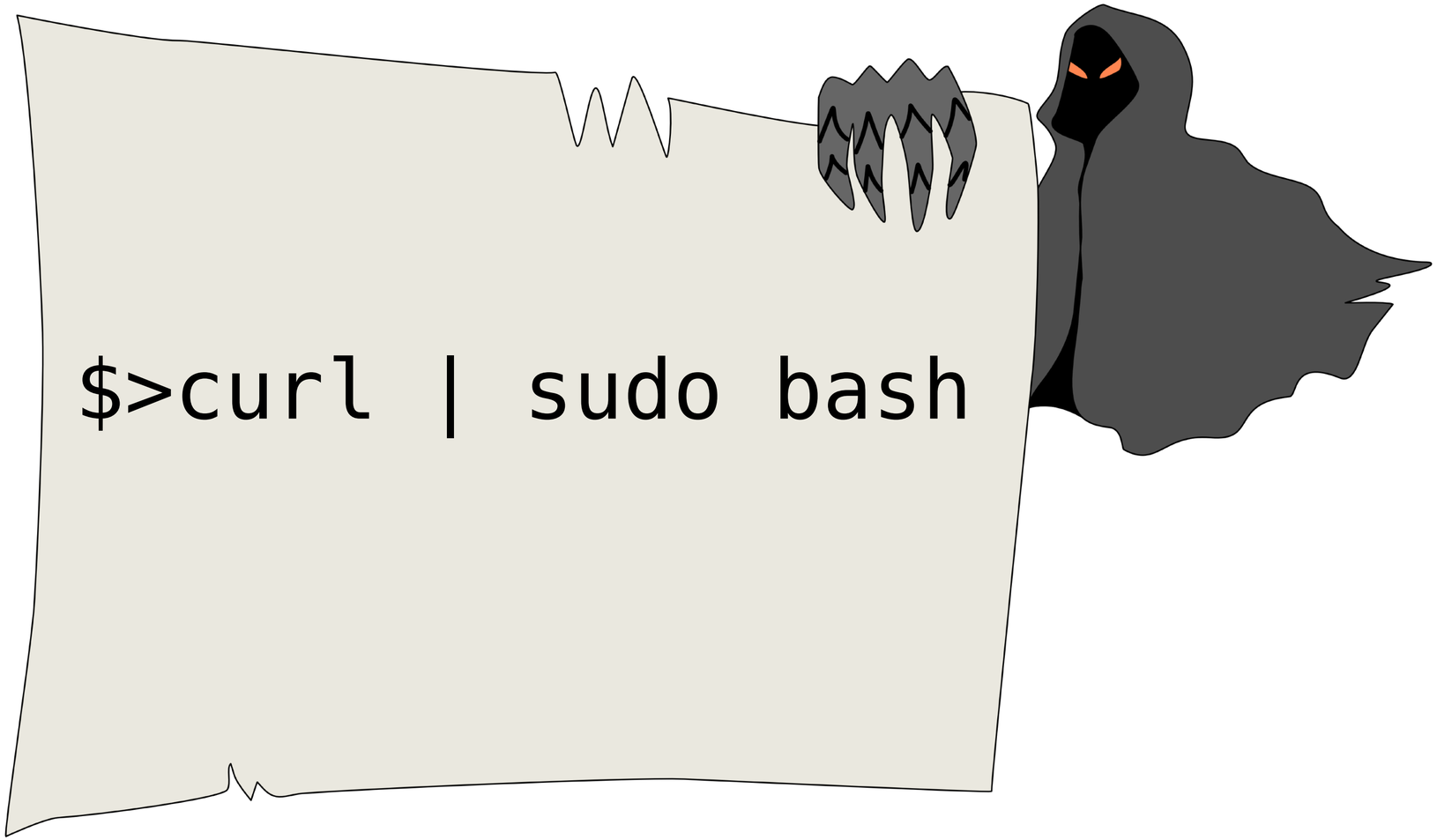
Using TUF

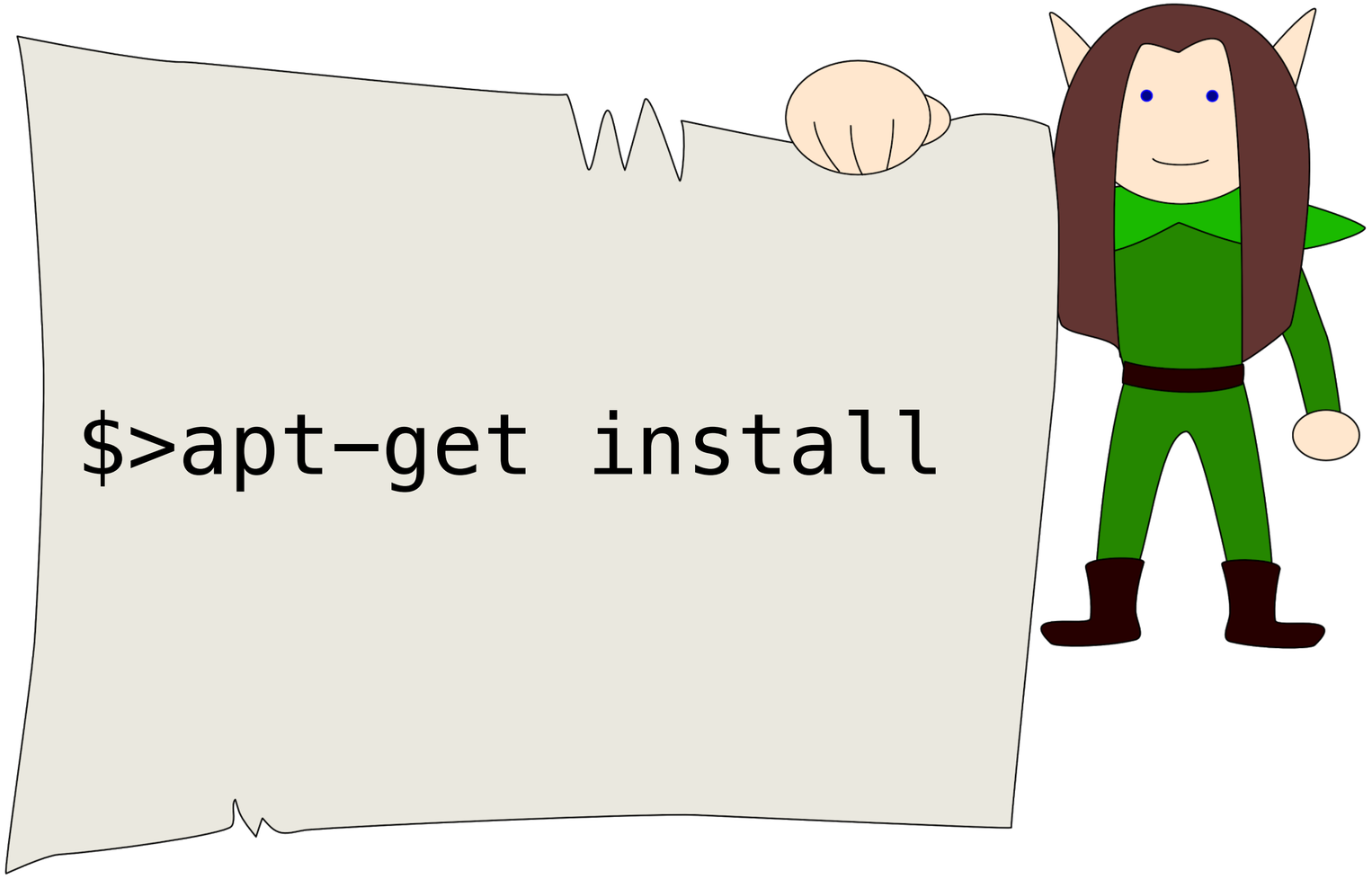
Hermetic Builds

# Where does software come from?

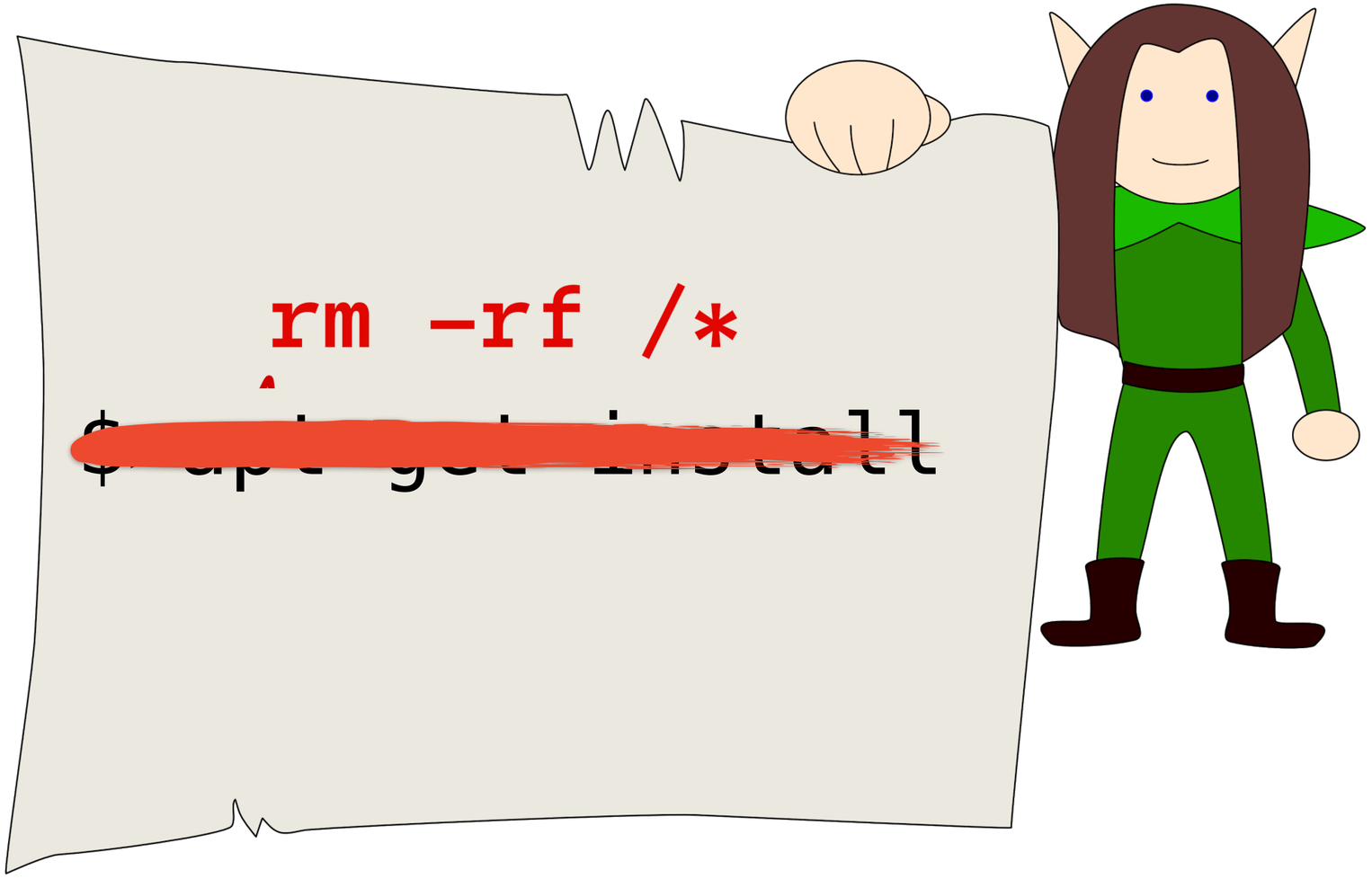


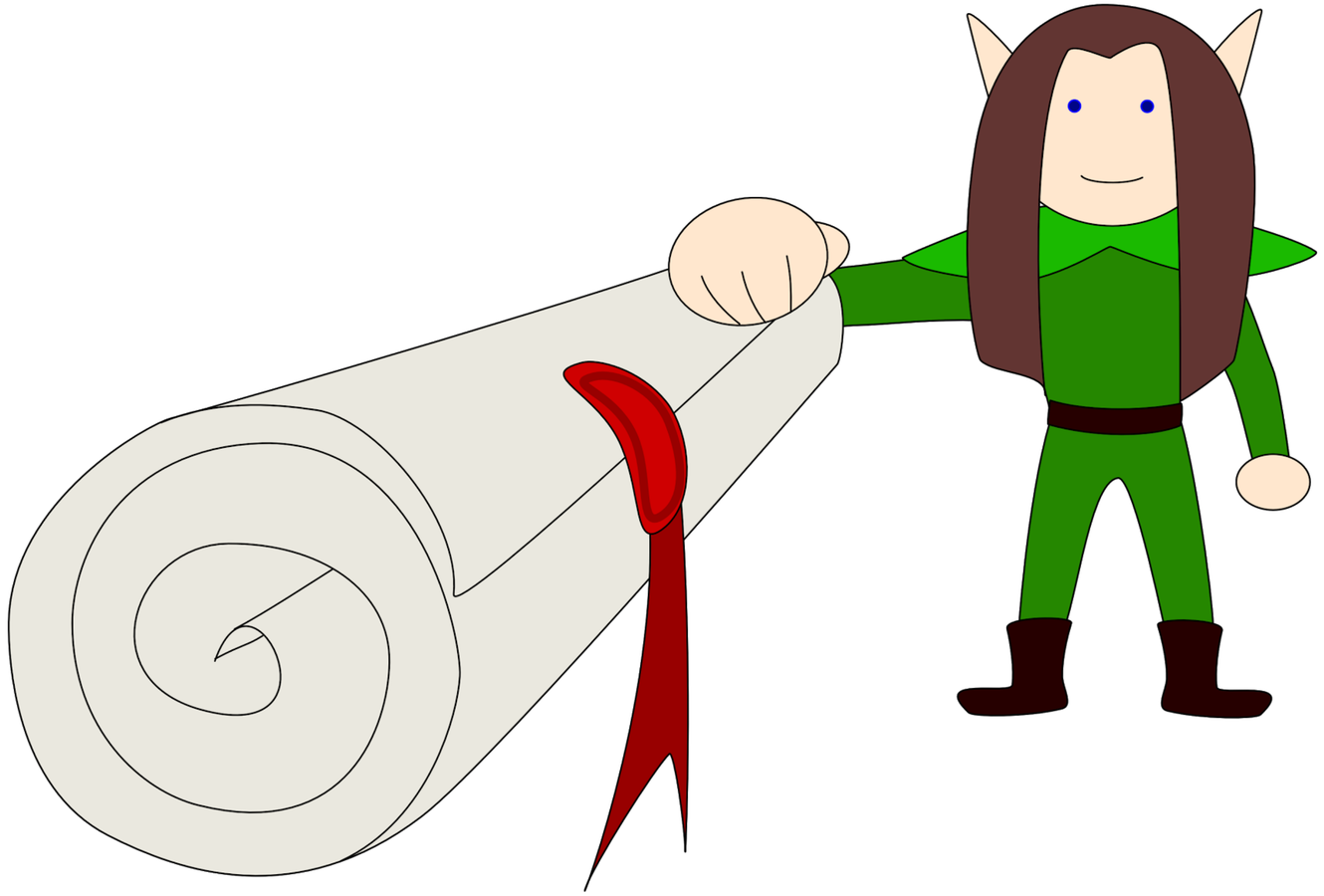
$> _


$>curl | sudo bash


$>apt-get install

- authenticity

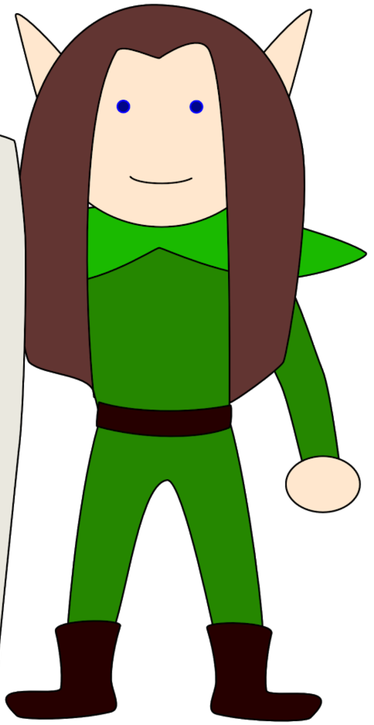

rm -rf /*



- authenticity
- integrity

$>apt-get install
really-old-foo

```
$>#not after 2007
$>apt-get install
really-old-foo
```

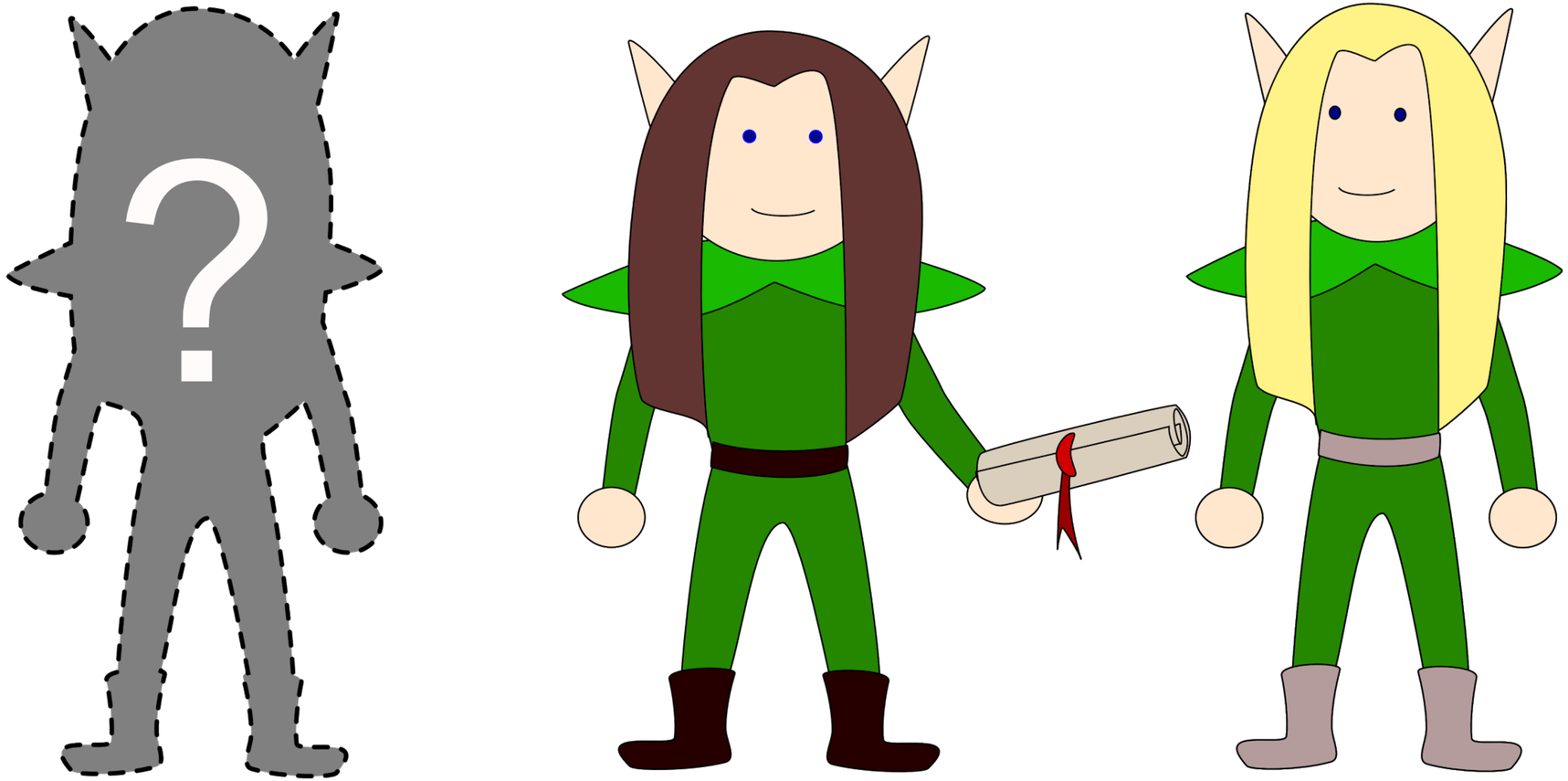


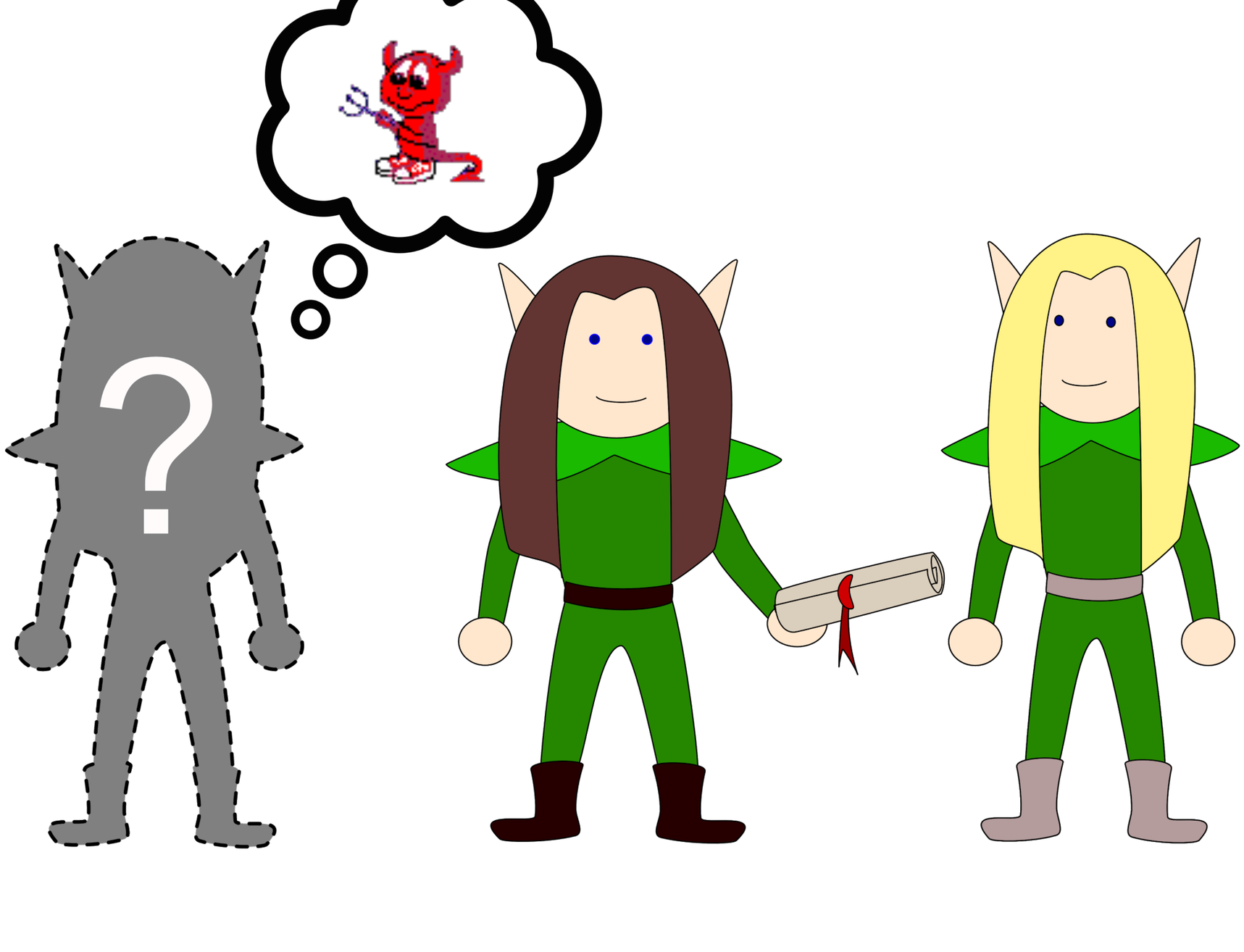
- authenticity
- integrity
- freshness

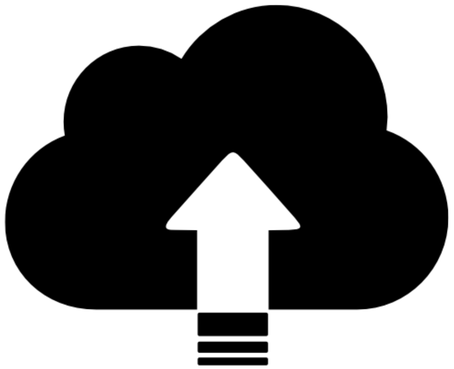
```
$> $pkg-manager install foo
```

- authenticity (TLS)
- integrity (TLS)
- freshness

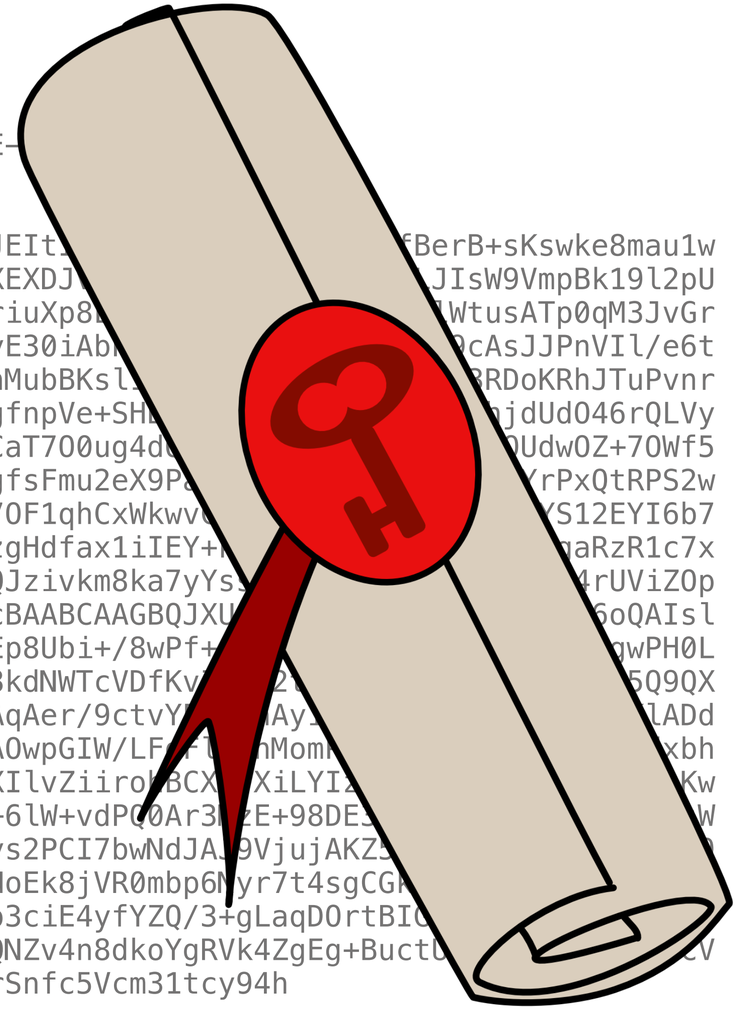
- authenticity (TLS - transport only)
- integrity (TLS - transport only)
- freshness



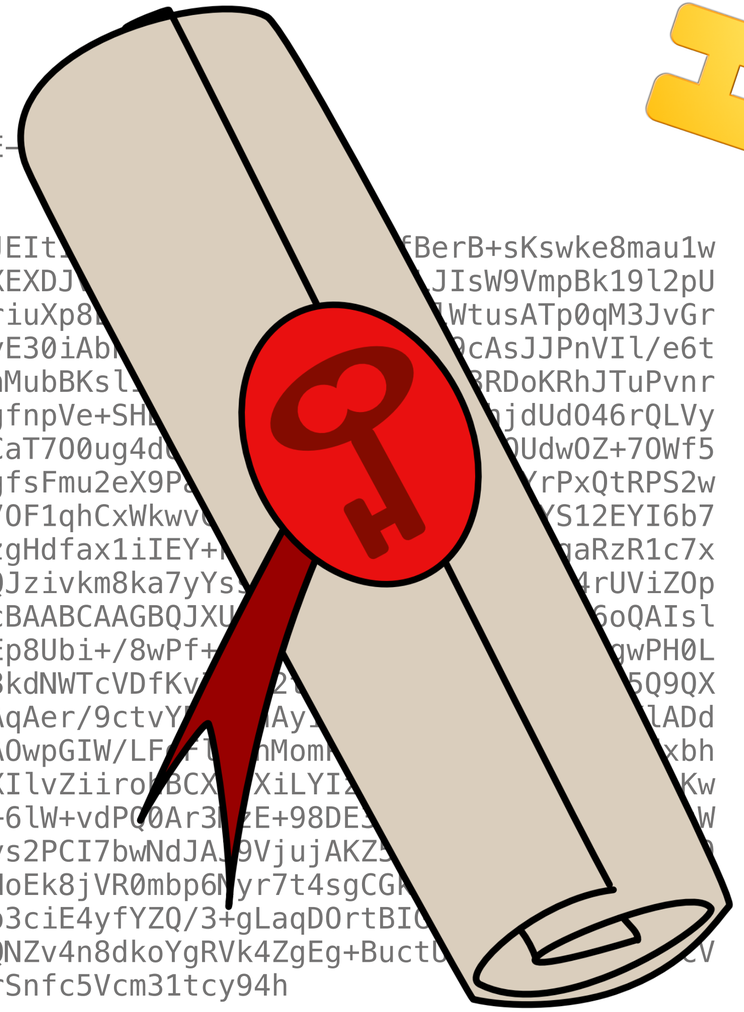




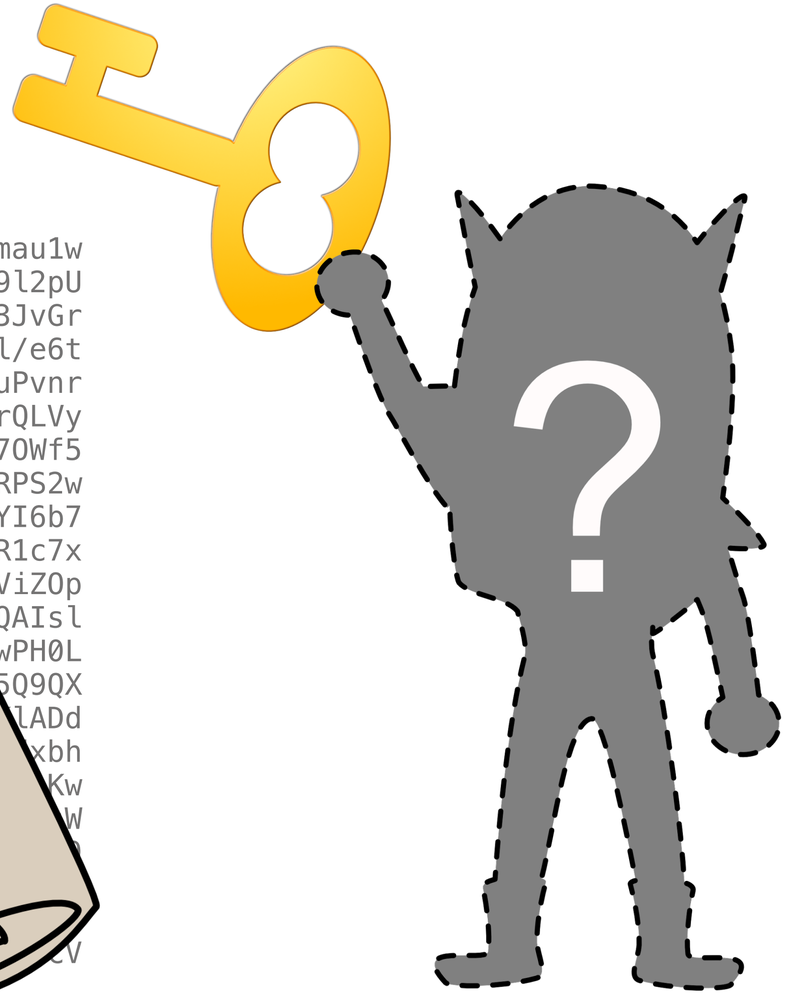
foo



```
-----BEGIN PGP SIGNATURE-
Version: GnuPG v1

iQIcBAABCAAGBQJXUtcbAAoJEIt                    fBerB+sKswke8mau1w
aalsr6MmQgARItPhsQvFUaRXEXDJ                   JIsW9VmpBk19l2pU
oWuiy6Ou9BWWA2qmS/3BKdmriuXp8                  lWtusATp0qM3JvGr
UWFDkxzQAqZycwuY7n1e4YNyE30iAb                 0cAsJJPnVIl/e6t
H3KI8F2QkQ/HwfN9KYfZKyChMubBKsl                3RDoKRhJTuPvnr
3QWH3Hlo6B/nqpX7hOcAkw6gfnpVe+SH              ajdUdO46rQLVy
iz1WcWgJkMj13kePrDC3gM+CaT7O0ug4d             QUdwOZ+7OWf5
ppdxXB5E6/XJN2X26V3KdHTgfsFmu2eX9P             YrPxQtRPS2w
oSp8Kkgh4kVTftymKvDcbFp7OF1qhCxWkwv             YS12EYI6b7
uQq0r5QjM6DzsnCRKrRh2EUzgHdfax1iIEY+             gaRzR1c7x
LRm5uxrGCq5zV9Lp+LBNGkAQJzivkm8ka7yYs             4rUViZOp
BsAoXDQUdfv80oIg39BMiQIcBAABCAAGBQJXU             6oQAIsl
7tPJpwTzKrv6r8gEQggyRvFEp8Ubi+/8wPf+             gwPH0L
js+kDrmNZRQ65PAEUf0mWwq8kdNWTcVDfKv             5Q9QX
i3h0xCZAT6hQSl7oSzWQIJkAqAer/9ctvY             lADd
Rjg2JHJJFODkciHWyQboU7UAOwpGIW/LF             xbh
o19tvoxBN32KS6yQM1oQDhvXIlvZiirok             Kw
P9m1tM0Lx+yY8OZTshedR+u+6lW+vdP             W
az6bMS07c947zbBZIAg0iO0ys2PCI7bwNdJA             
QODaFM4cGrndS53tmtjR2vPNoEk8jVR0mbp6             
seD6XOq2cWTSSisIRo8Cccdo3ciE4yfYZQ/3+gLaqDOrtBI             
7LjYz72Z5gjzK2X+jQCJFD/QNZv4n8dkoYgRVk4ZgEg+Buct             V
NRcXJtGybbGDQ85vFuzLYyUrSnfc5Vcm31tcy94h
=nSRR
-----END PGP SIGNATURE-----
```



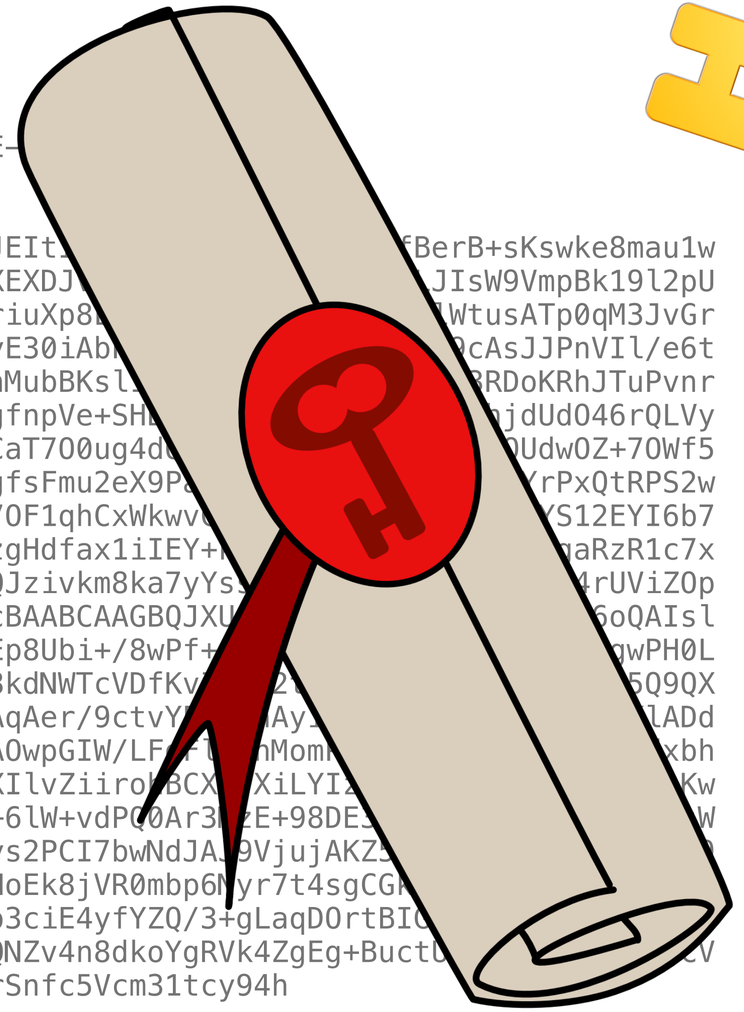
```
-----BEGIN PGP SIGNATURE-
Version: GnuPG v1

iQIcBAABCAAGBQJXUtcbAAoJEIt                    fBerB+sKswke8mau1w
aalsr6MmQgARItPhsQvFUaRXEXDJ                   JIsW9VmpBk19l2pU
oWuiy6Ou9BWWA2qmS/3BKdmriuXp8                  lWtusATp0qM3JvGr
UWFDkxzQAqZycwuY7n1e4YNyE30iAb                 0cAsJJPnVIl/e6t
H3KI8F2QkQ/HwfN9KYfZKyChMubBKsl                3RDoKRhJTuPvnr
3QWH3Hlo6B/nqpX7hOcAkw6gfnpVe+SH              ajdUdO46rQLVy
iz1WcWgJkMj13kePrDC3gM+CaT7O0ug4d             QUdwOZ+7OWf5
ppdxXB5E6/XJN2X26V3KdHTgfsFmu2eX9P             YrPxQtRPS2w
oSp8Kkgh4kVTftymKvDcbFp7OF1qhCxWkwv             YS12EYI6b7
uQq0r5QjM6DzsnCRKrRh2EUzgHdfax1iIEY+             gaRzR1c7x
LRm5uxrGCq5zV9Lp+LBNGkAQJzivkm8ka7yYs            4rUViZOp
BsAoXDQUdfv80oIg39BMiQIcBAABCAAGBQJXU            6oQAIsl
7tPJpwTzKrv6r8gEQggyRvFEp8Ubi+/8wPf+            gwPH0L
js+kDrmNZRQ65PAEUf0mWwq8kdNWTcVDfKv             5Q9QX
i3h0xCZAT6hQSl7oSzWQIJkAqAer/9ctvY             lADd
Rjg2JHJJFODkciHWyQboU7UAOwpGIW/LF             xbh
o19tvoxBN32KS6yQM1oQDhvXIlvZiirol             Kw
P9m1tM0Lx+yY8OZTshedR+u+6lW+vdP             W
az6bMS07c947zbBZIAg0iO0ys2PCI7bwNdJA             
QODaFM4cGrndS53tmtjR2vPNoEk8jVR0mbp6             
seD6XOq2cWTSSisIRo8Cccdo3ciE4yfYZQ/3+gLaqDOrtBI             
7LjYz72Z5gjzK2X+jQCJFD/QNZv4n8dkoYgRVk4ZgEg+Buct             V
NRcXJtGybbGDQ85vFuzLYyUrSnfc5Vcm31tcy94h
=nSRR
-----END PGP SIGNATURE-----
```



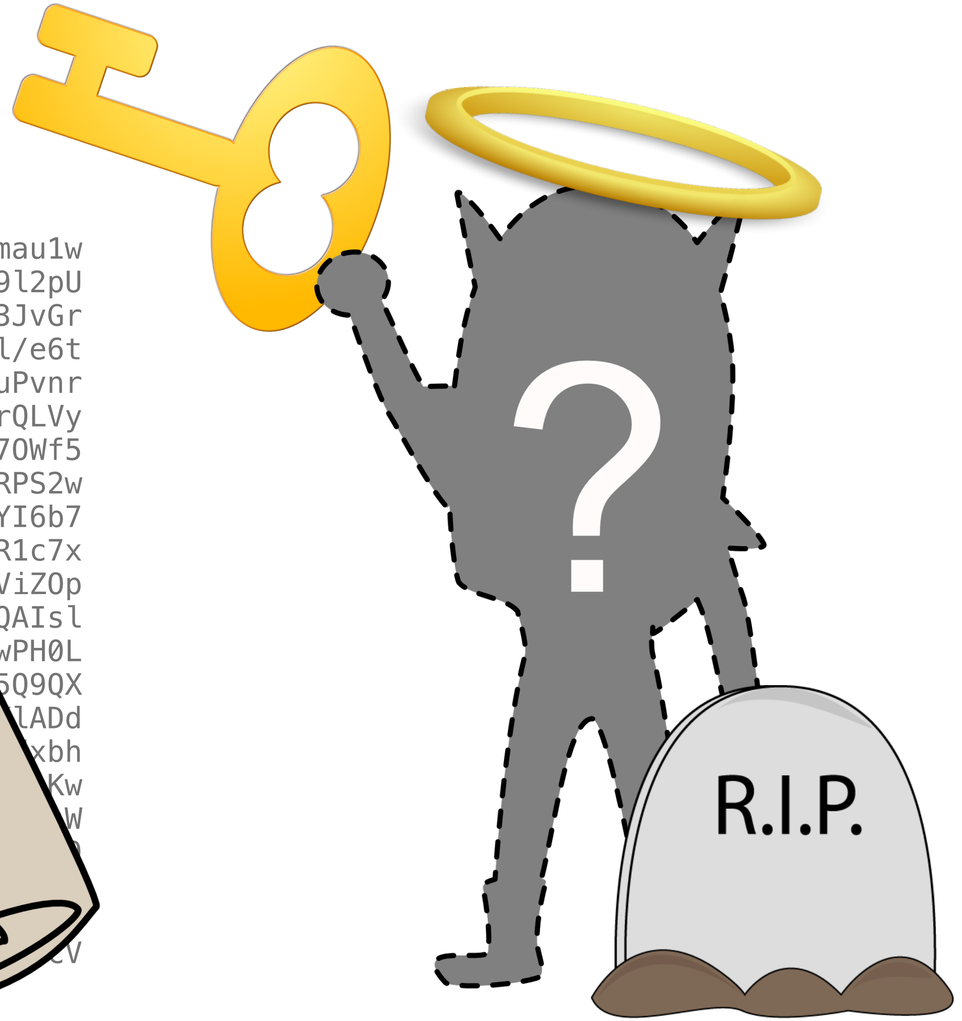
# Replay Attacks?

```
-----BEGIN PGP SIGNATURE--
Version: GnuPG v1

iQIcBAABCAAGBQJXUtcbAAoJEI
aalsr6MmQgARItPhsQvFUaRXEX
oWuiy6Ou9BWWA2qmS/3BKdmriu
UWFDkxzQAqZycwuY7n1e4YNyE3
H3KI8F2QkQ/HwfN9KYfZKyChMu
3QWH3Hlo6B/nqpX7hOcAkw6gfn
iz1WcWgJkMj13kePrDC3gM+CaT
ppdxXB5E6/XJN2X26V3KdHTgfs
oSp8Kkgh4kVTftymKvDcbFp7OF
uQq0r5QjM6DzsnCRKrRh2EUzgH
LRm5uxrGCq5zV9Lp+LBNGkAQJz
BsAoXDQUdfv80oIg39BMiQIcBA
7tPJpwTzKrv6r8gEQggyRvFEp8
js+kDrmNZRQ65PAEUf0mWwq8kd
i3h0xCZAT6hQSl7oSzWQIJkAqA
Rjg2JHJJFODkciHWyQboU7UAOv
o19tvoxBN32KS6yQM1oQDhvXI
P9m1tM0Lx+yY8OZTshedR+u+6
az6bMS07c947zbBZIAg0iO0ys
QODaFM4cGrndS53tmtjR2vPNc
seD6XOq2cWTSSisIRo8Cccdo
7LjYz72Z5gjzK2X+jQCJFD/Q
NRcXJtGybbGDQ85vFuzLYyUrSnfc5Vcm31tcy94h
=nSRR
-----END PGP SIGNATURE-----
```

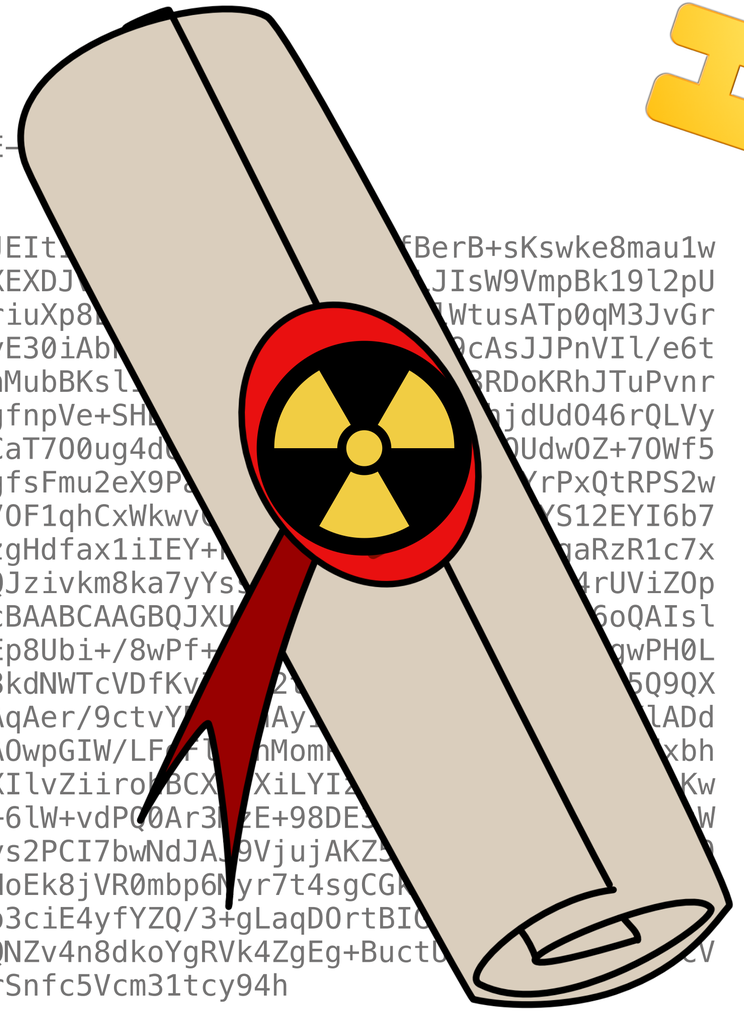
# Survivable Key Compromise?



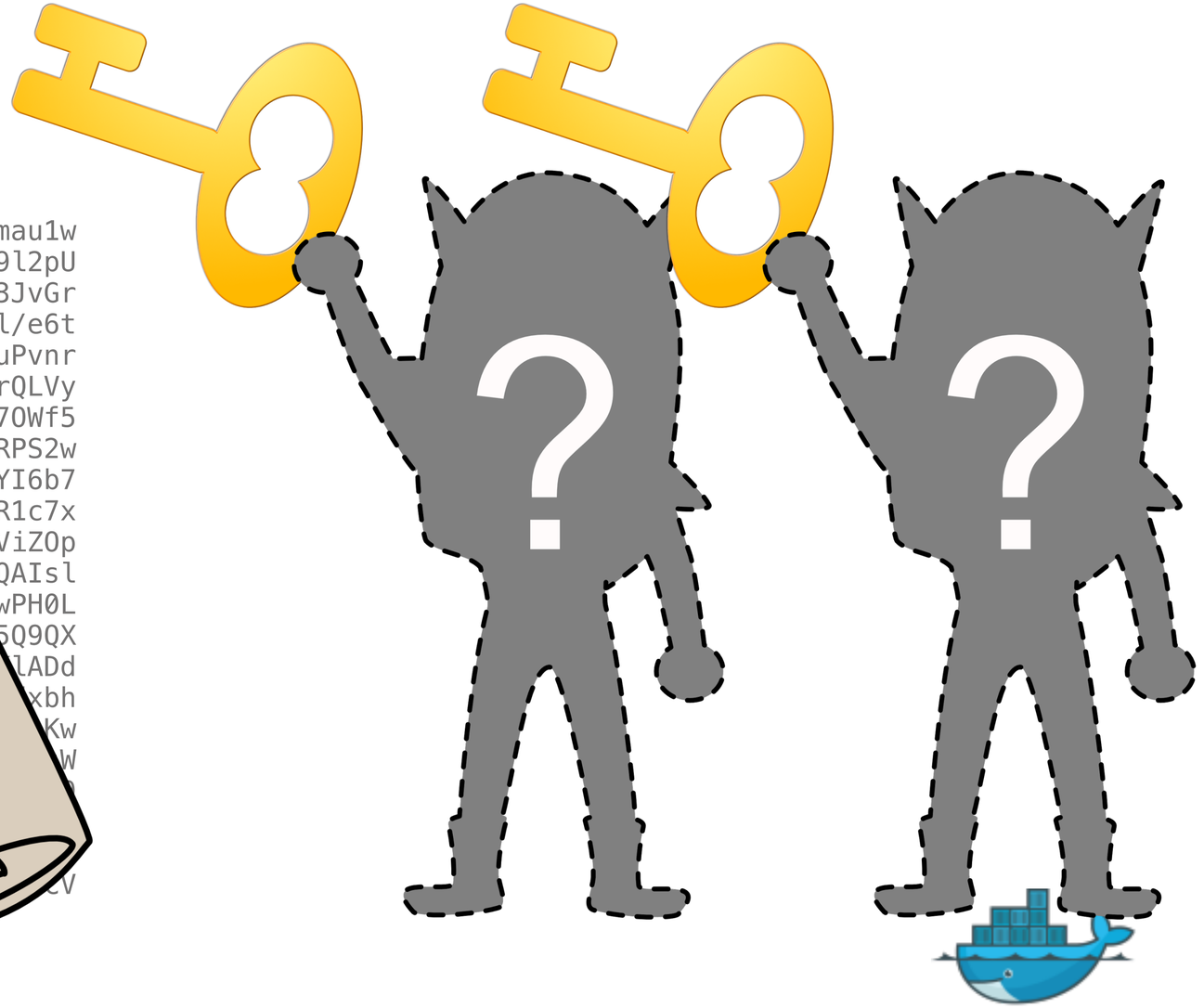
```
-----BEGIN PGP SIGNATURE-
Version: GnuPG v1

iQIcBAABCAAGBQJXUtcbAAoJEIt                    fBerB+sKswke8mau1w
aalsr6MmQgARItPhsQvFUaRXEXDJ                    JIsW9VmpBk19l2pU
oWuiy6Ou9BWWA2qmS/3BKdmriuXp8                   lWtusATp0qM3JvGr
UWFDkxzQAqZycwuY7n1e4YNyE30iAb                  0cAsJJPnVIl/e6t
H3KI8F2QkQ/HwfN9KYfZKyChMubBKsl                 3RDoKRhJTuPvnr
3QWH3Hlo6B/nqpX7hOcAkw6gfnpVe+SH               ajdUdO46rQLVy
iz1WcWgJkMj13kePrDC3gM+CaT7O0ug4d              OUdwOZ+7OWf5
ppdxXB5E6/XJN2X26V3KdHTgfsFmu2eX9P              YrPxQtRPS2w
oSp8Kkgh4kVTftymKvDcbFp7OF1qhCxWkwv             YS12EYI6b7
uQq0r5QjM6DzsnCRKrRh2EUzgHdfax1iIEY+             gaRzR1c7x
LRm5uxrGCq5zV9Lp+LBNGkAQJzivkm8ka7yYs             4rUViZOp
BsAoXDQUdfv80oIg39BMiQIcBAABCAAGBQJXU             6oQAIsl
7tPJpwTzKrv6r8gEQggyRvFEp8Ubi+/8wPf+             gwPH0L
js+kDrmNZRQ65PAEUf0mWwq8kdNWTcVDfKv             5Q9QX
i3h0xCZAT6hQSl7oSzWQIJkAqAer/9ctvY             lADd
Rjg2JHJJFODkciHWyQboU7UAOwpGIW/LF             xbh
o19tvoxBN32KS6yQM1oQDhvXIlvZiirol             Kw
P9m1tM0Lx+yY8OZTshedR+u+6lW+vdP             W
az6bMS07c947zbBZIAg0iO0ys2PCI7bwNdJA             
QODaFM4cGrndS53tmtjR2vPNoEk8jVR0mbp6             
seD6XOq2cWTSSisIRo8Cccdo3ciE4yfYZQ/3+gLaqDOrtBI             
7LjYz72Z5gjzK2X+jQCJFD/QNZv4n8dkoYgRVk4ZgEg+Buct             V
NRcXJtGybbGDQ85vFuzLYyUrSnfc5Vcm31tcy94h
=nSRR
-----END PGP SIGNATURE-----
```



# Trust Thresholding?

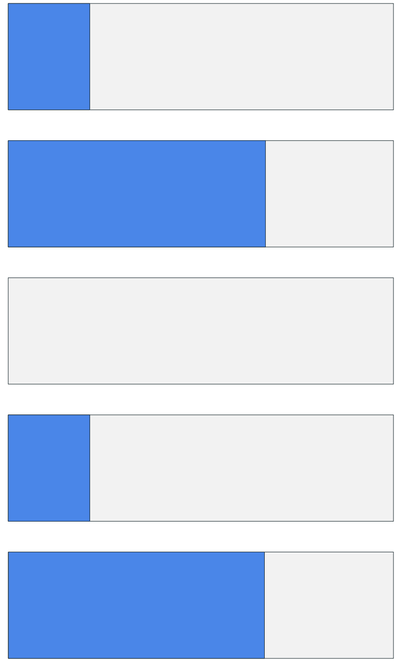


```
-----BEGIN PGP SIGNATURE-
Version: GnuPG v1

iQIcBAABCAAGBQJXUtcbAAoJEIt                    fBerB+sKswke8mau1w
aalsr6MmQgARItPhsQvFUaRXEXDJ                    JIsW9VmpBk19l2pU
oWuiy6Ou9BWWA2qmS/3BKdmriuXp8                   lWtusATp0qM3JvGr
UWFDkxzQAqZycwuY7n1e4YNyE30iAb                  0cAsJJPnVIl/e6t
H3KI8F2QkQ/HwfN9KYfZKyChMubBKsl                 3RDoKRhJTuPvnr
3QWH3Hlo6B/nqpX7hOcAkw6gfnpVe+SH               njdUdO46rQLVy
iz1WcWgJkMj13kePrDC3gM+CaT7O0ug4d              OUdwOZ+7OWf5
ppdxXB5E6/XJN2X26V3KdHTgfsFmu2eX9P              YrPxQtRPS2w
oSp8Kkgh4kVTftymKvDcbFp7OF1qhCxWkwv             YS12EYI6b7
uQq0r5QjM6DzsnCRKrRh2EUzgHdfax1iIEY+             gaRzR1c7x
LRm5uxrGCq5zV9Lp+LBNGkAQJzivkm8ka7yYs             4rUViZOp
BsAoXDQUdfv80oIg39BMiQIcBAABCAAGBQJXU             6oQAIsl
7tPJpwTzKrv6r8gEQggyRvFEp8Ubi+/8wPf+             gwPH0L
js+kDrmNZRQ65PAEUf0mWwq8kdNWTcVDfKv             5Q9QX
i3h0xCZAT6hQSl7oSzWQIJkAqAer/9ctvY             lADd
Rjg2JHJJFODkciHWyQboU7UAOwpGIW/LF             xbh
o19tvoxBN32KS6yQM1oQDhvXIlvZiirol             Kw
P9m1tM0Lx+yY8OZTshedR+u+6lW+vdP             W
az6bMS07c947zbBZIAg0iO0ys2PCI7bwNdJ             
QODaFM4cGrndS53tmtjR2vPNoEk8jVR0mbp6             
seD6XOq2cWTSSisIRo8Cccdo3ciE4yfYZQ/3+gLaqDOrtBI             
7LjYz72Z5gjzK2X+jQCJFD/QNZv4n8dkoYgRVk4ZgEg+Buct             V
NRcXJtGybbGDQ85vFuzLYyUrSnfc5Vcm31tcy94h
=nSRR
-----END PGP SIGNATURE-----
```

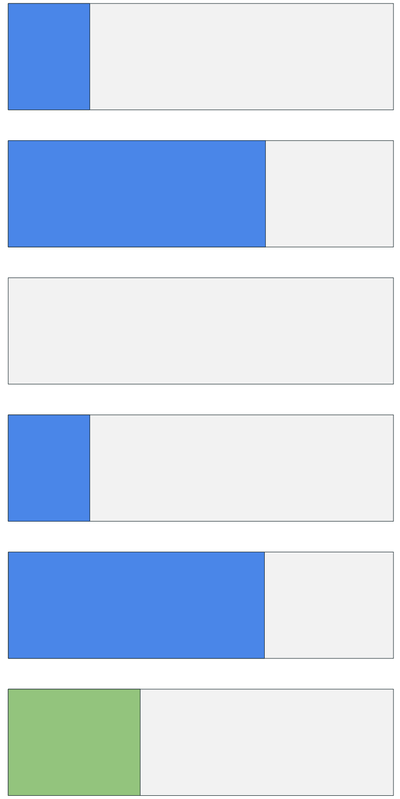


- authenticity
- integrity
- freshness
- thresholding
- survivable key compromise

- authenticity
- integrity
- freshness
- thresholding
- survivable key compromise



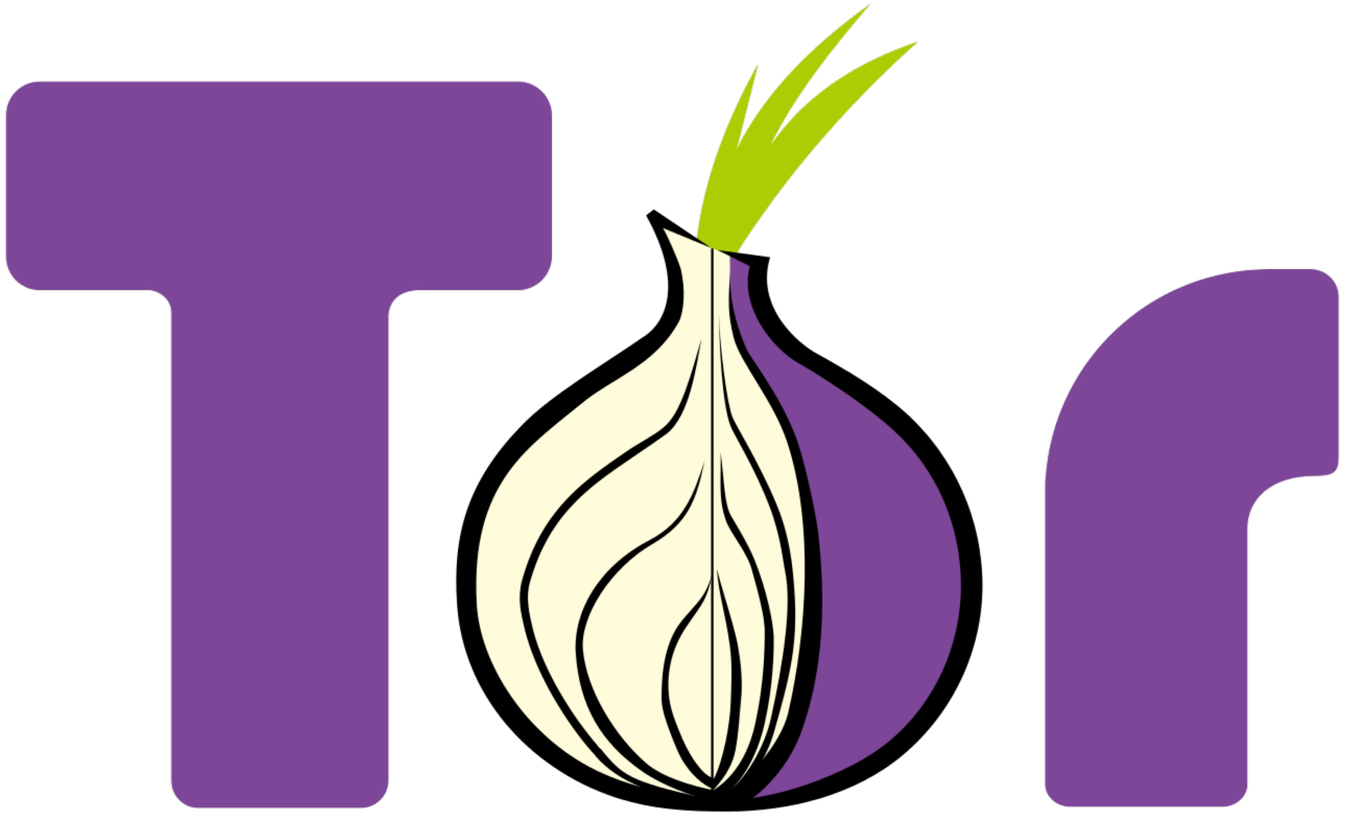
- authenticity
- integrity
- freshness
- thresholding
- survivable key compromise
- ease of use

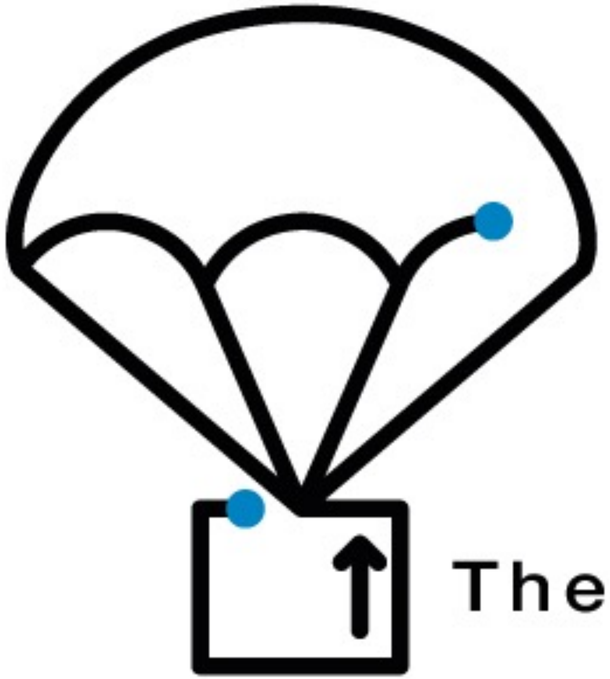


Secure Software Distribution is Hard!
arbitrary software
malicious mirrors
endless data
slow retrieval
indefinite freeze
key compromise
rollback
mix-and-match
extra dependencies
wrong software

# Get TUF

## (The Update Framework)

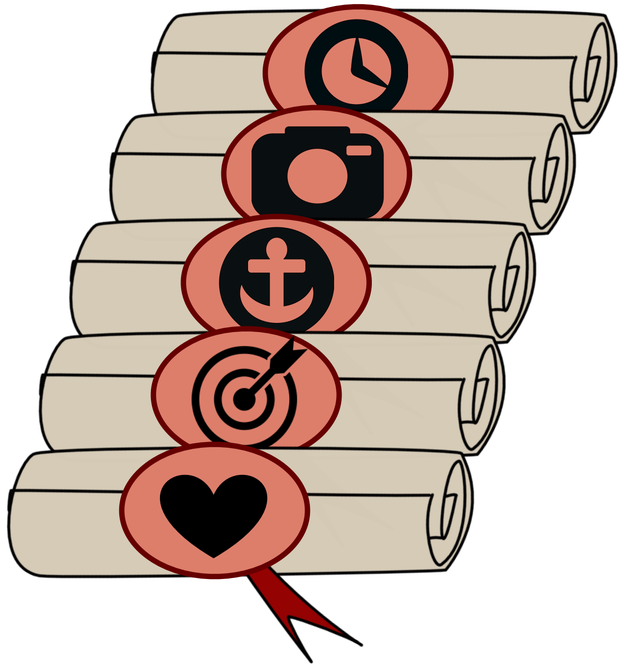

Tor



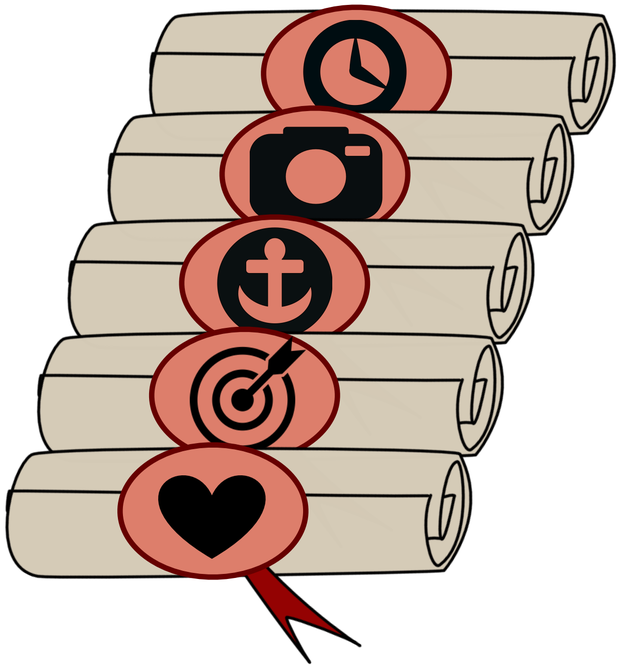
The Update Framework

- Diplomat: Using Delegations to Protect Community Repositories
- Survivable Key Compromise in Software Update Systems
- A Look in the Mirror: Attacks on Package Managers
- Package Management Security

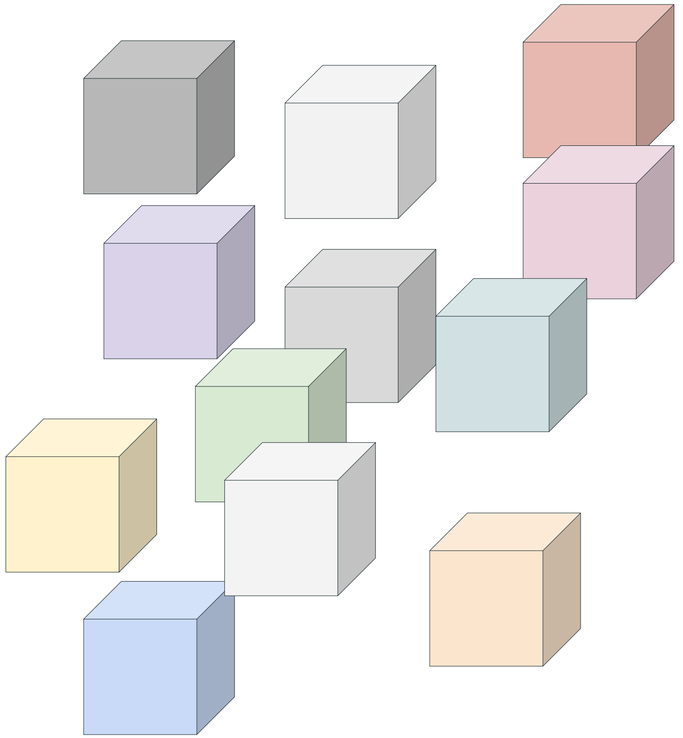
# TUF repository



TUF repository



packages



root

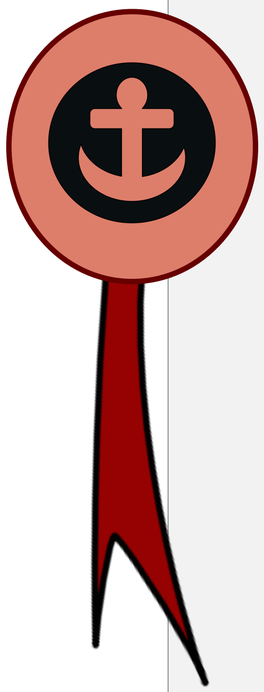
timestamp

snapshot

targets

delegation

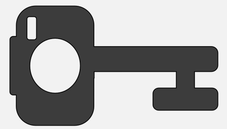
# Root Metadata



Root:

Timestamp:

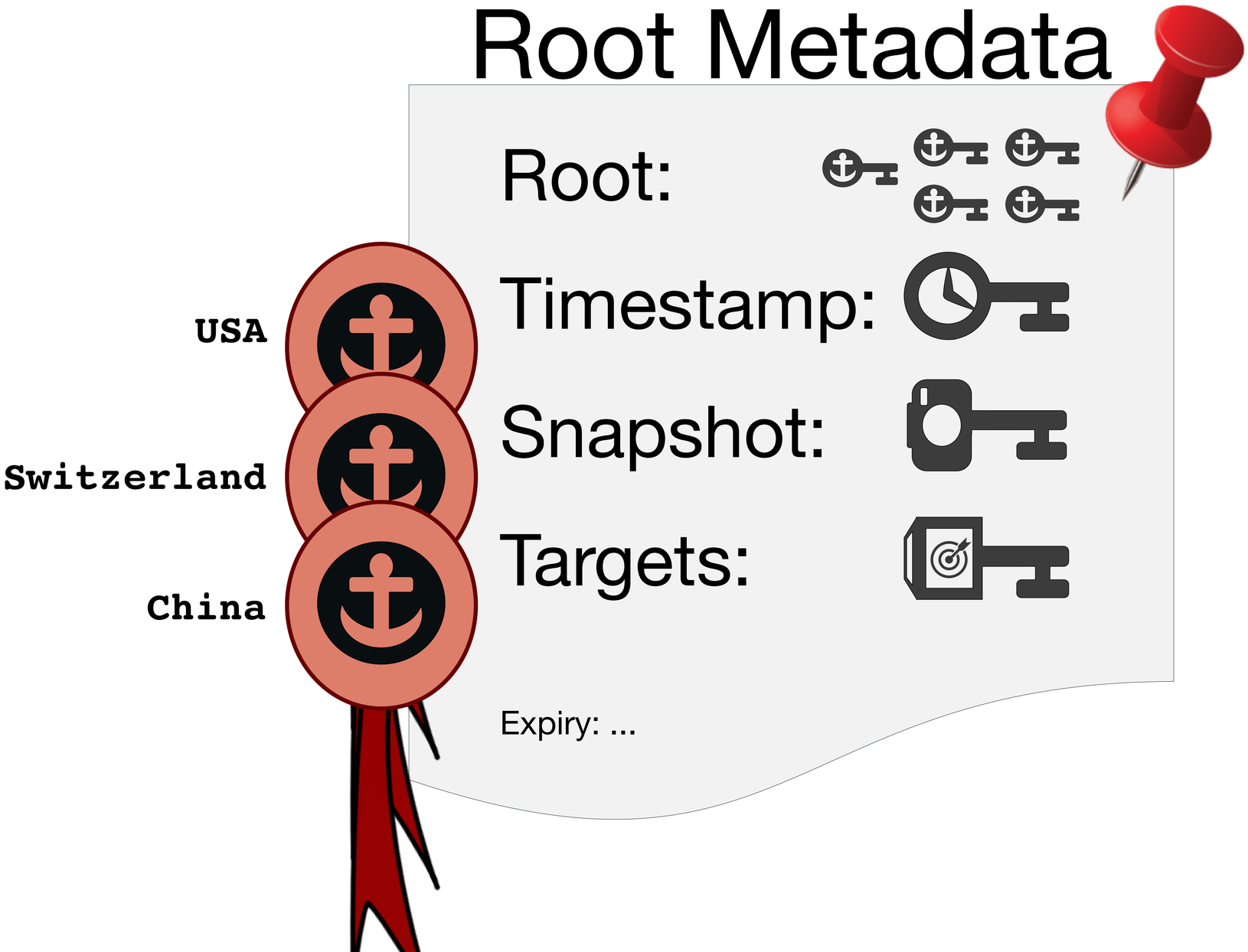
Snapshot:



Targets:

Expiry: ...

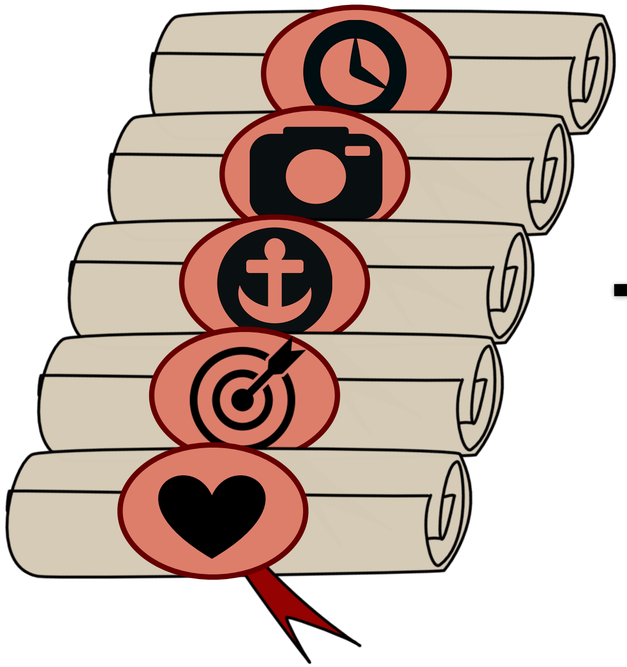
# Root Metadata



# Offline for security

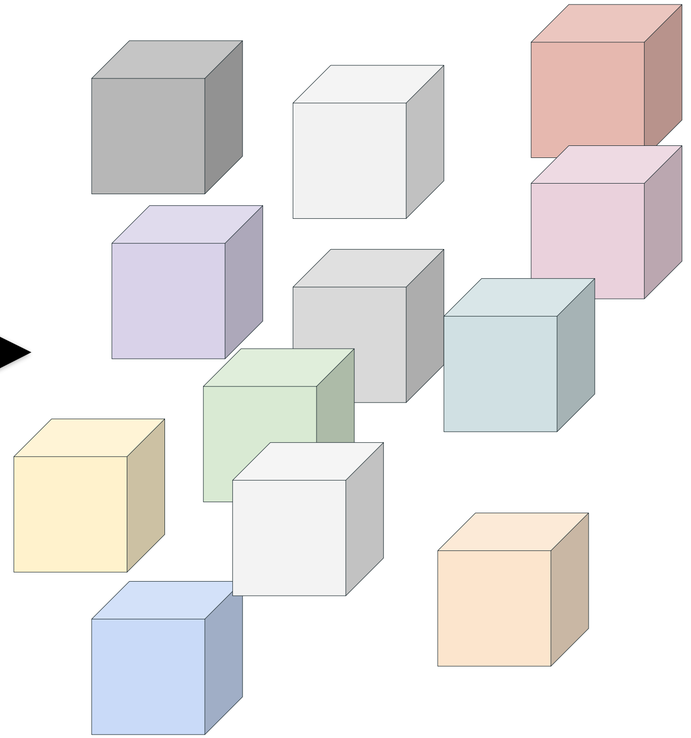
- Backup in bank vault
- Use signing hardware

TUF repository

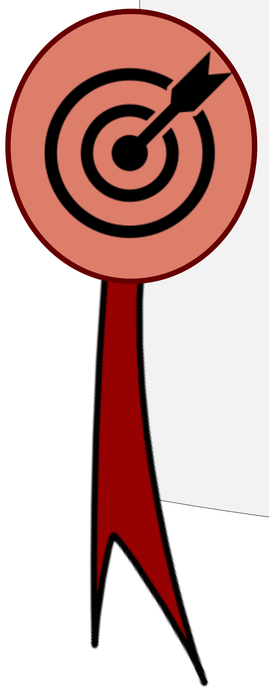


?

packages



# Targets Metadata



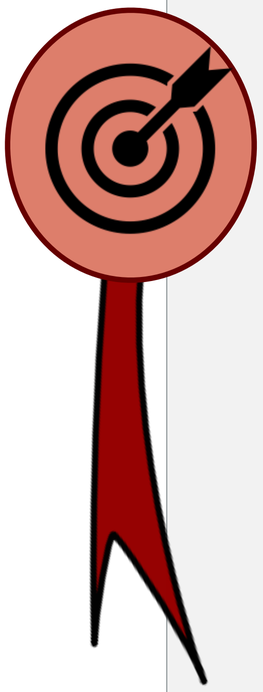
java : { hashes }

openssl : { hashes }

…

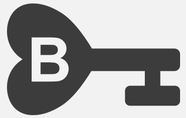
Expiry: ...

# Targets Metadata



Keys: {

Alice:


Bob: 

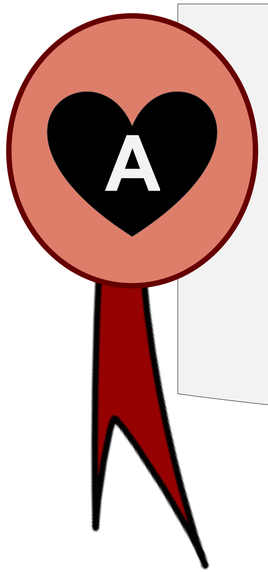

}

java: [Alice]

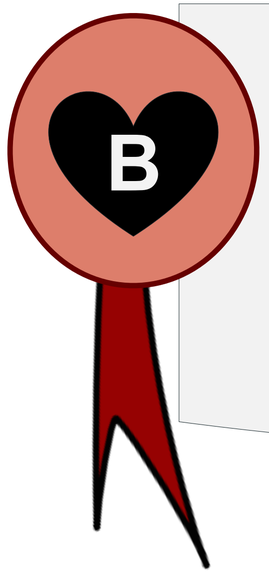
openssl: [Bob]

Expiry: ...

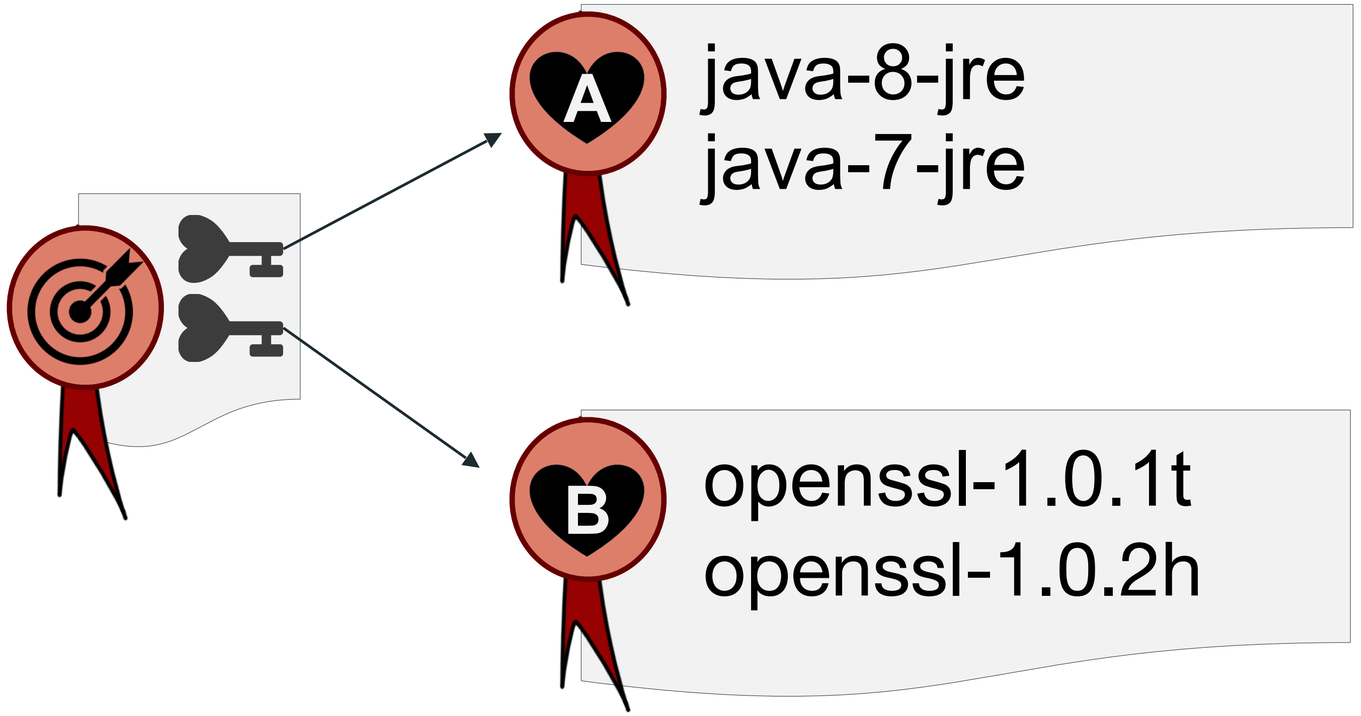
# Delegation Metadata

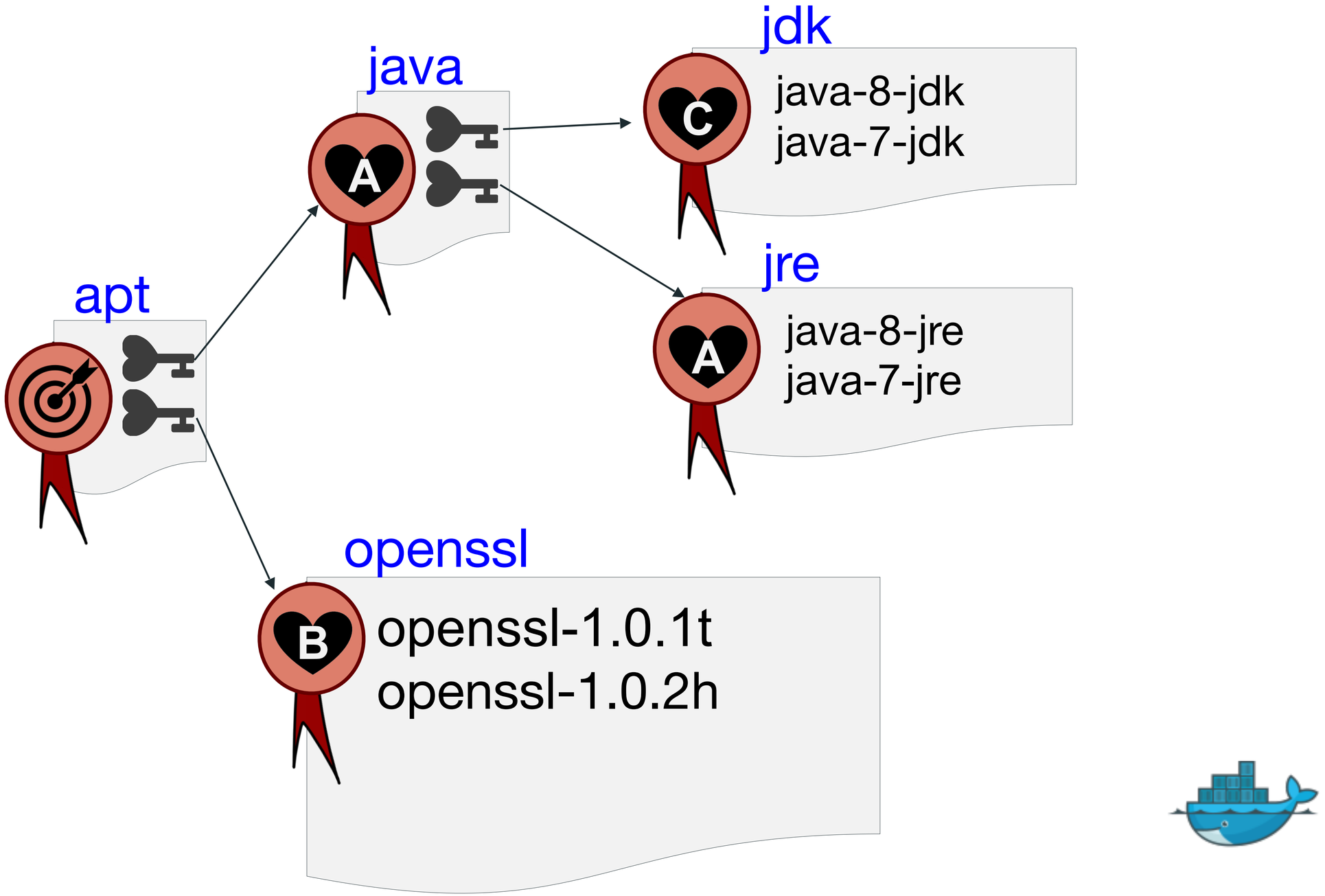


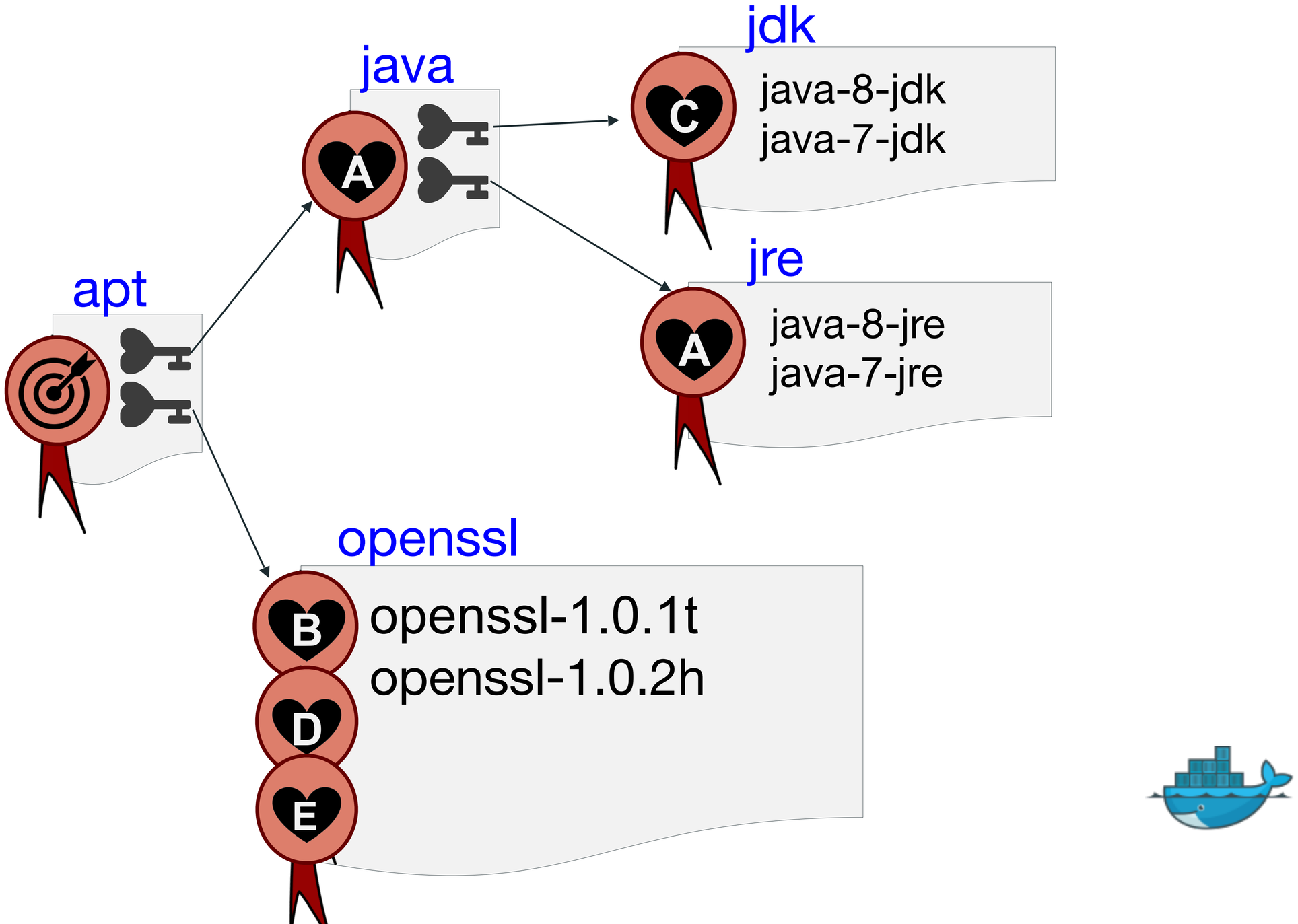

java-8-jre : { hashes }
java-7-jre : { hashes }
...
Expiry: ...




openssl-1.0.1t : { hashes }
openssl-1.0.2h : { hashes }
...
Expiry: ...

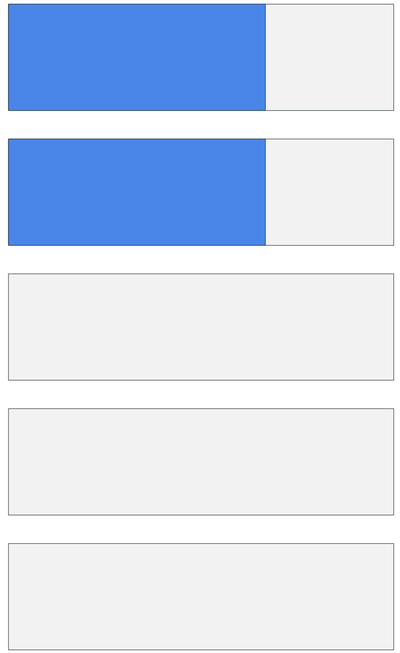

A
java-8-jre
java-7-jre
B
openssl-1.0.1t
openssl-1.0.2h


jdk
java
java-8-jdk
java-7-jdk
C
A
jre
apt
java-8-jre
java-7-jre
A
openssl
B
openssl-1.0.1t
openssl-1.0.2h

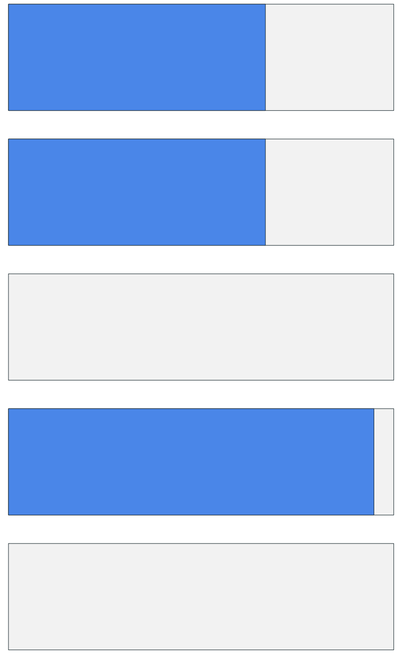

jdk
java
java-8-jdk
java-7-jdk
C
A
jre
apt
java-8-jre
java-7-jre
A
openssl
B
openssl-1.0.1t
openssl-1.0.2h
D
E

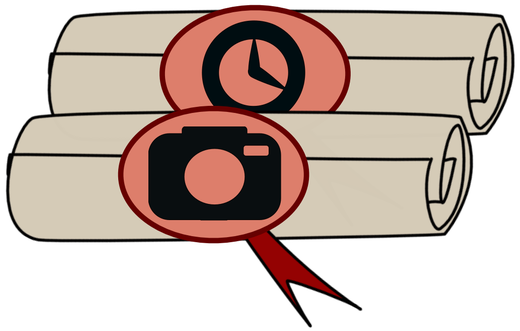
- authenticity
- integrity
- freshness
- thresholding
- survivable key compromise

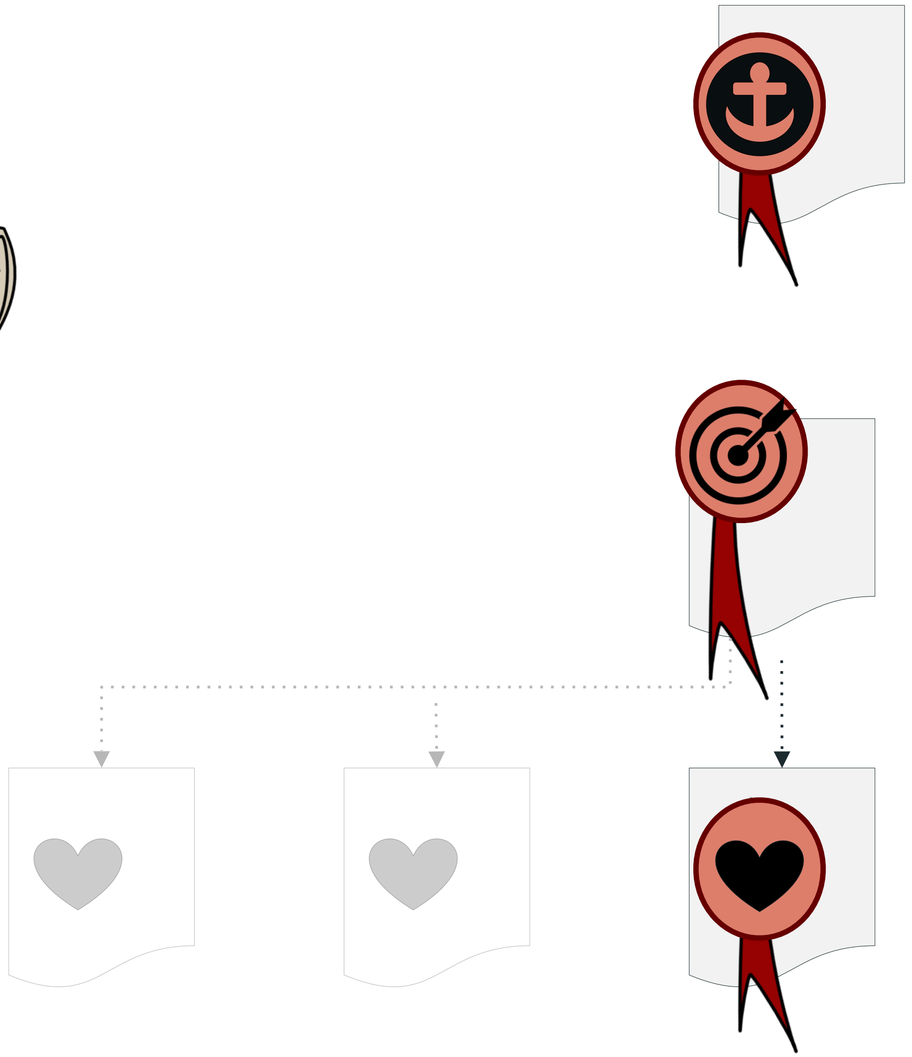
- authenticity
- integrity
- freshness
- thresholding
- survivable key compromise



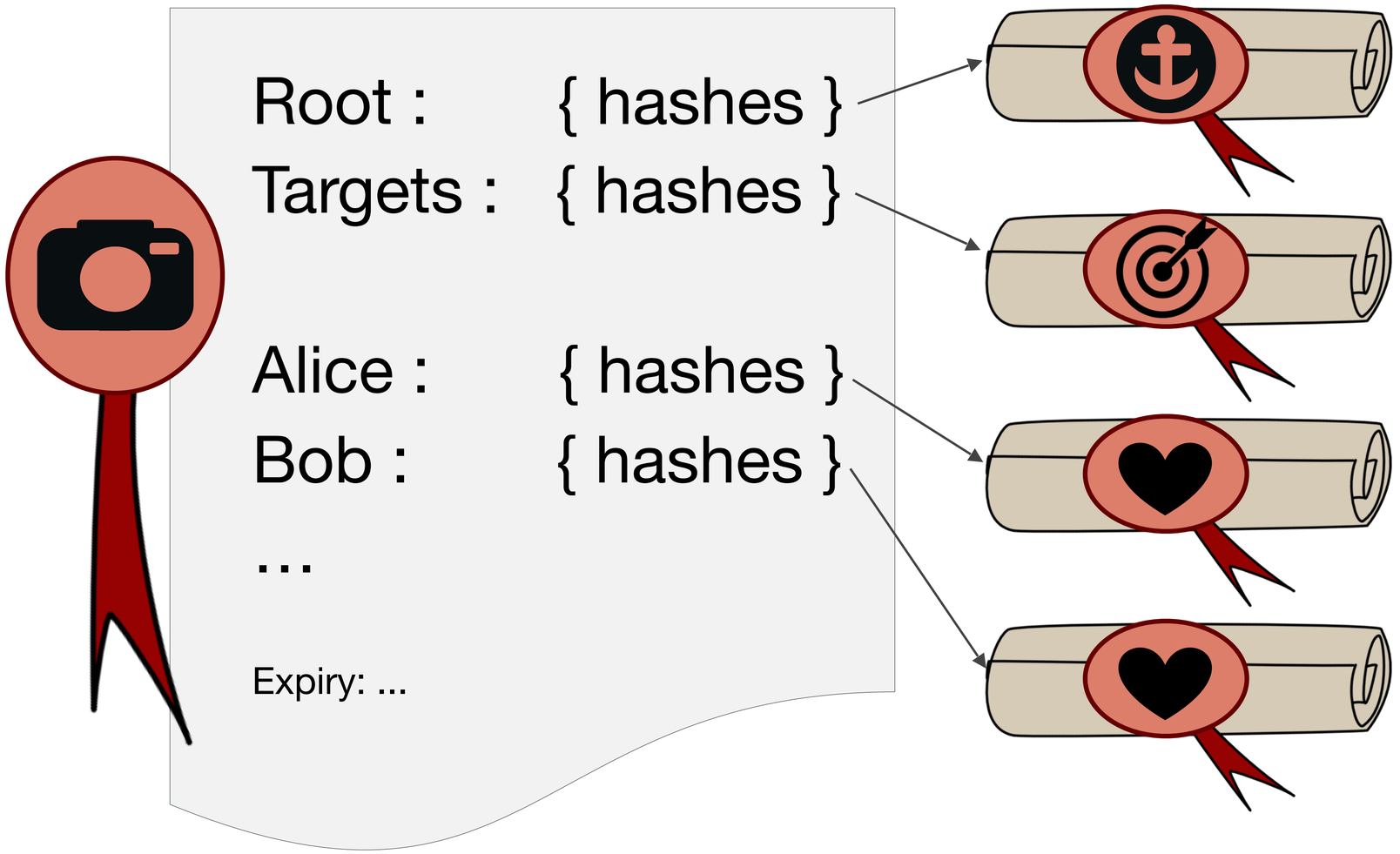
- authenticity
- integrity
- freshness
- thresholding
- survivable key compromise



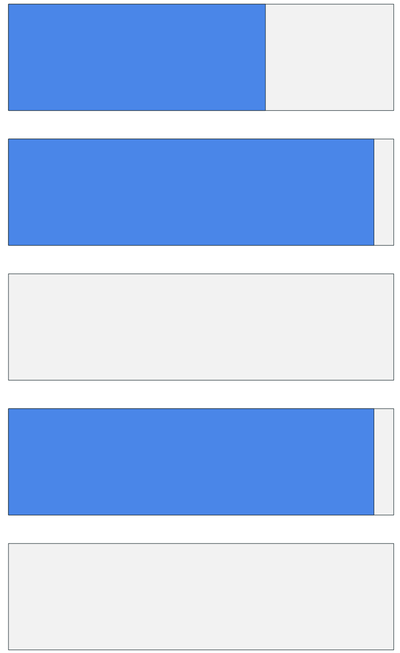




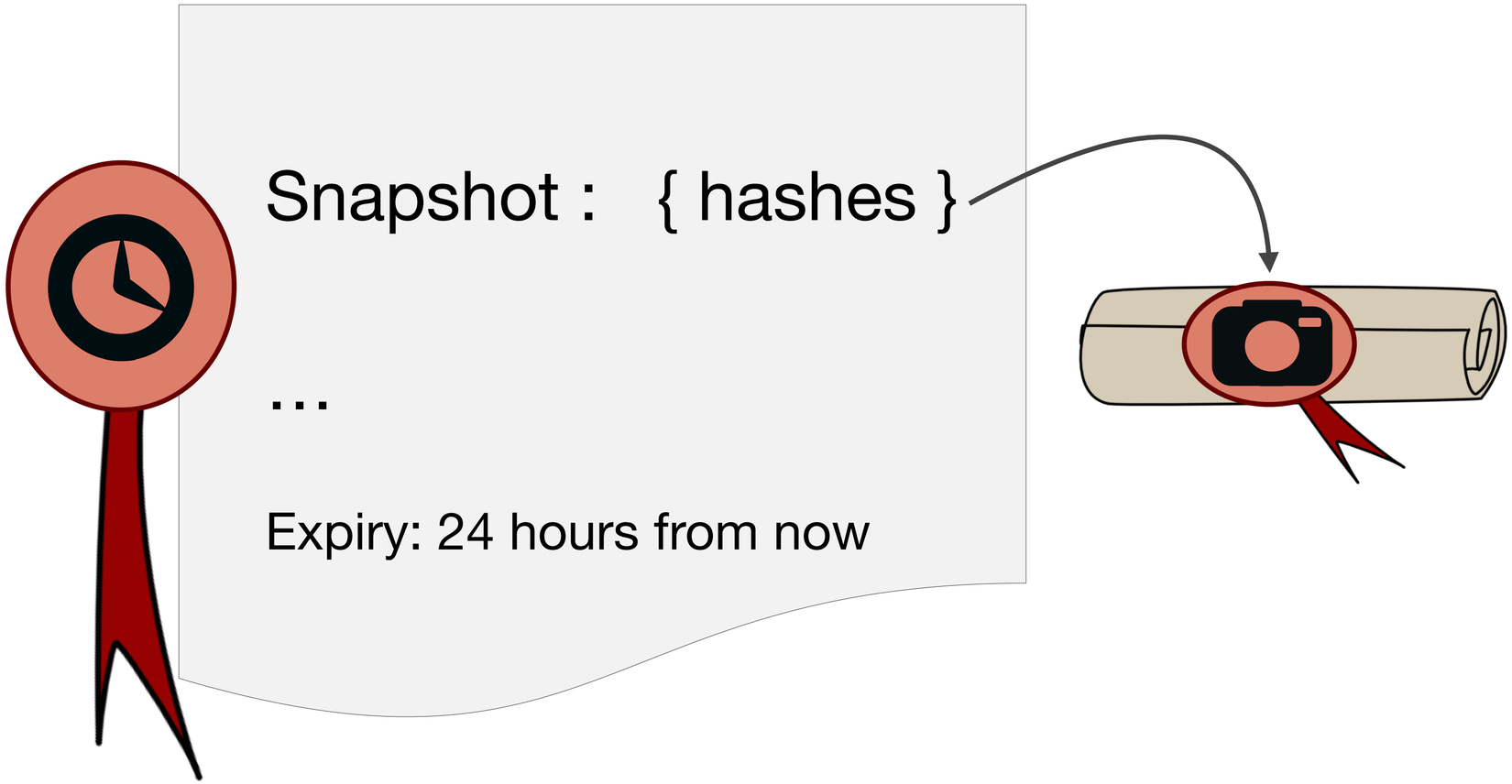
# Snapshot Metadata



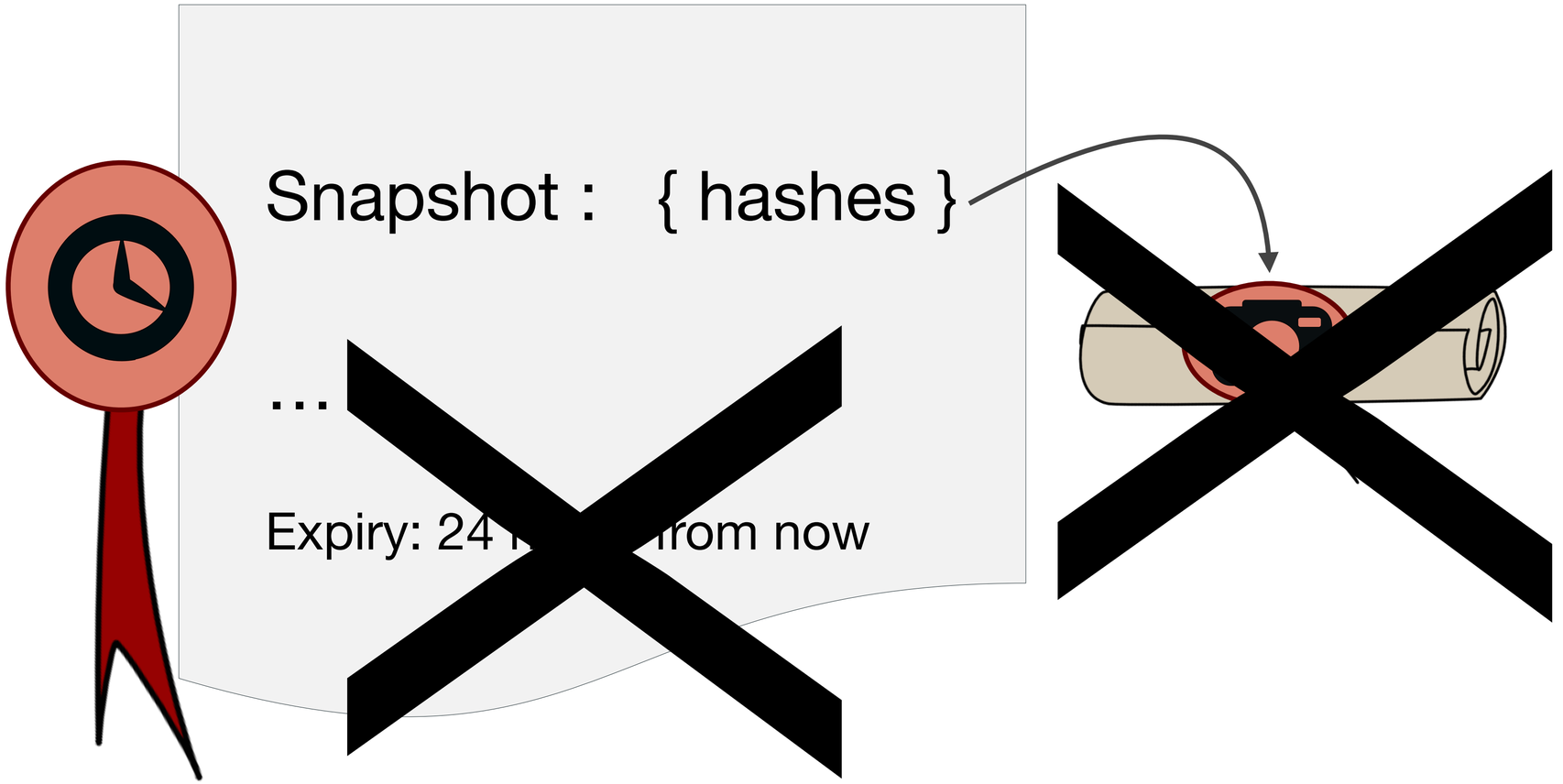
- authenticity
- integrity
- freshness
- thresholding
- survivable key compromise

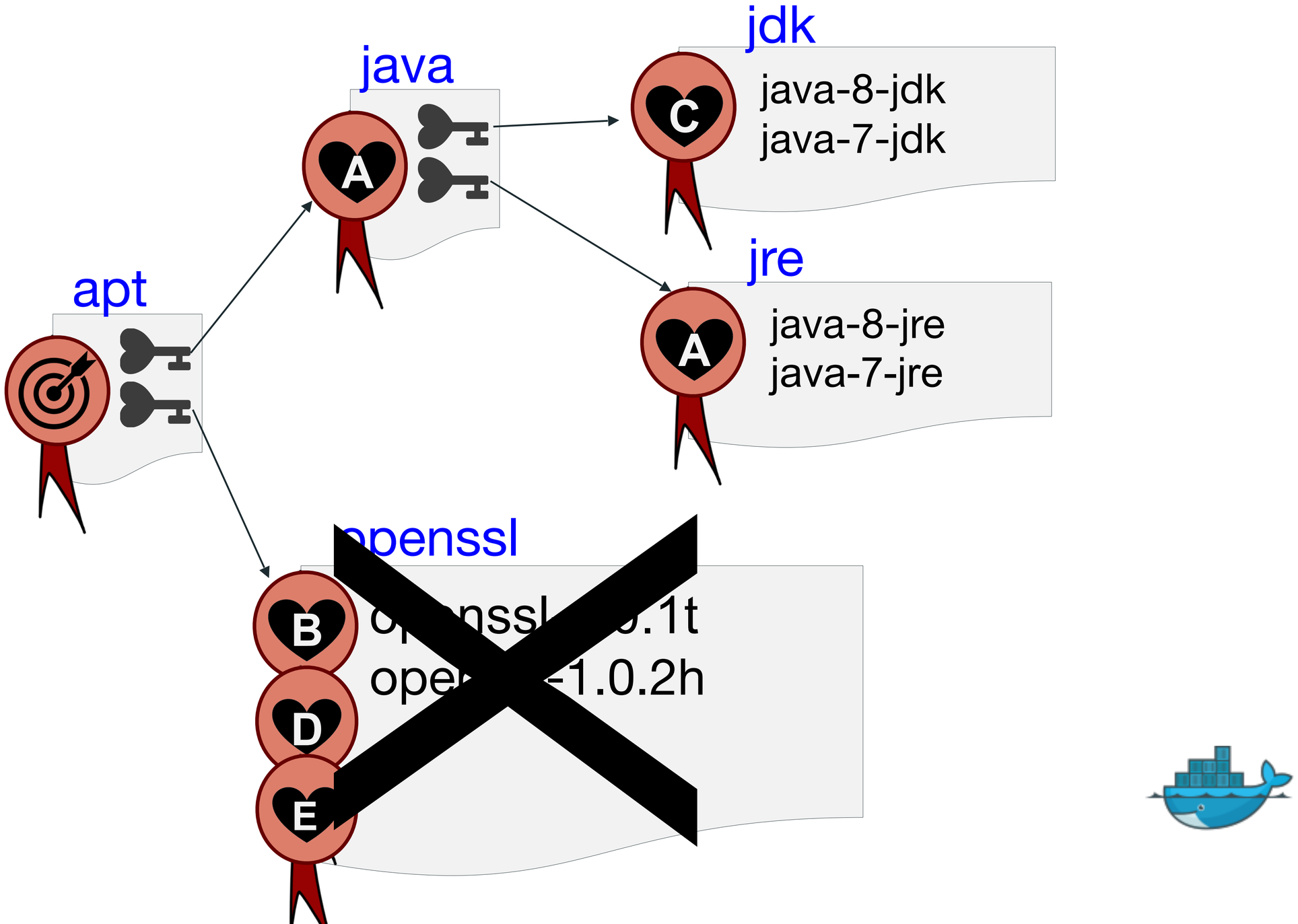


# Timestamp Metadata

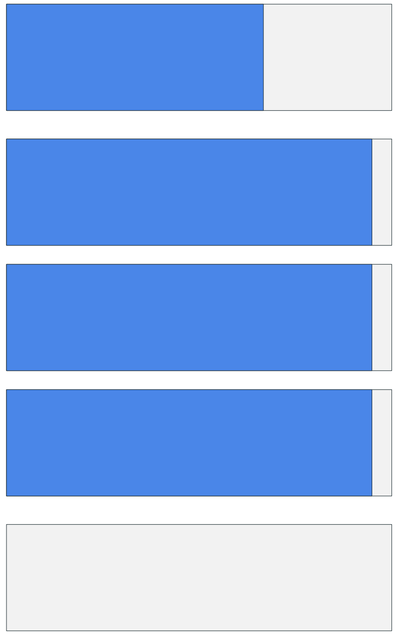


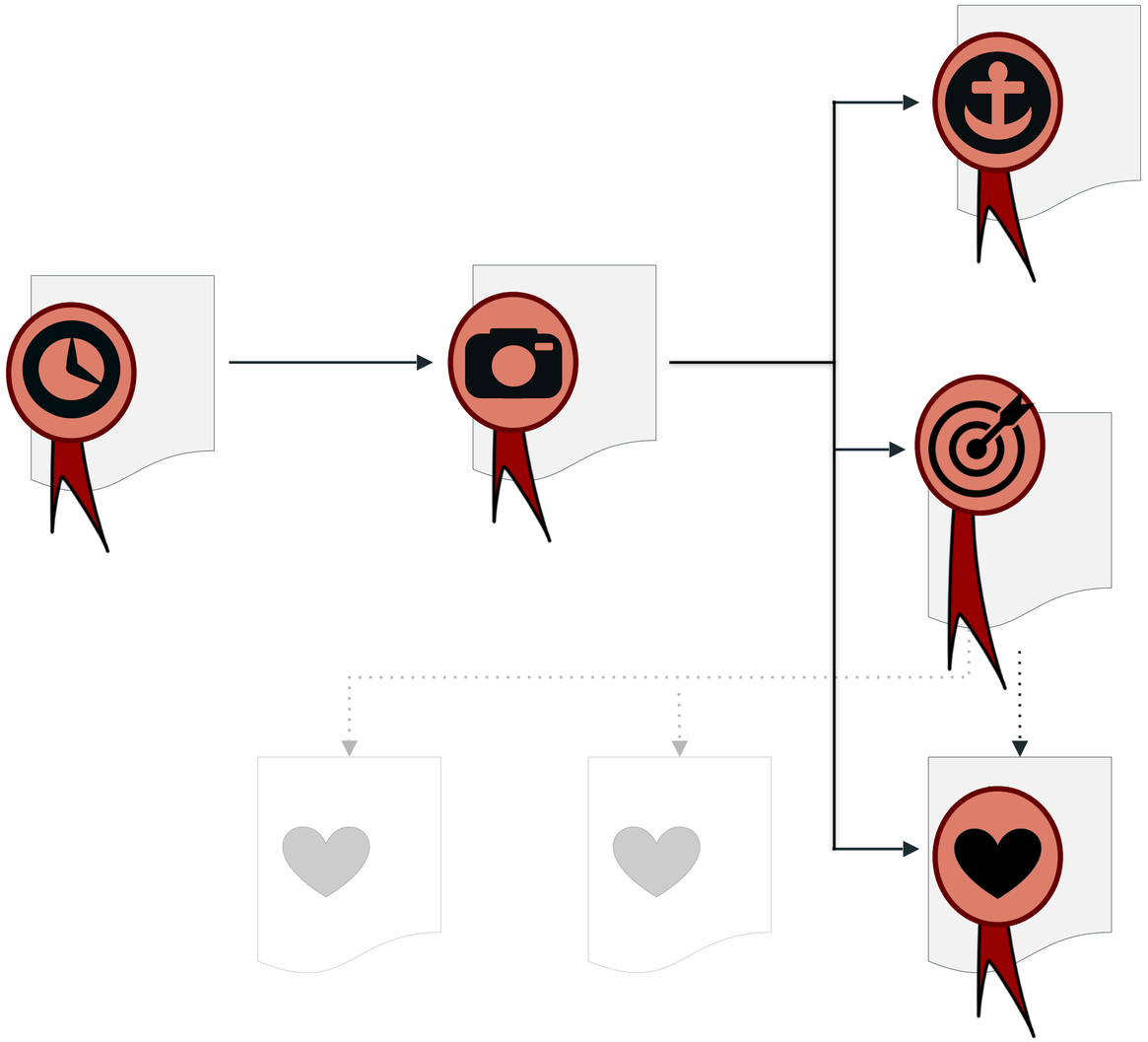
# Timestamp Metadata

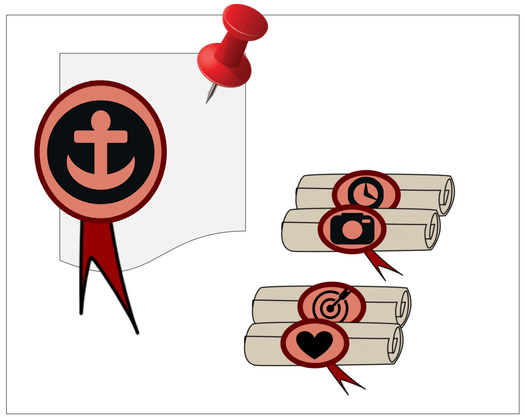


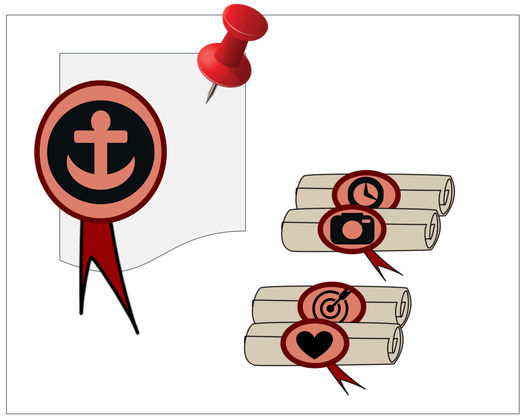

jdk
java
java-8-jdk
java-7-jdk
C
A
apt
jre
java-8-jre
java-7-jre
A
openssl
B
D
E

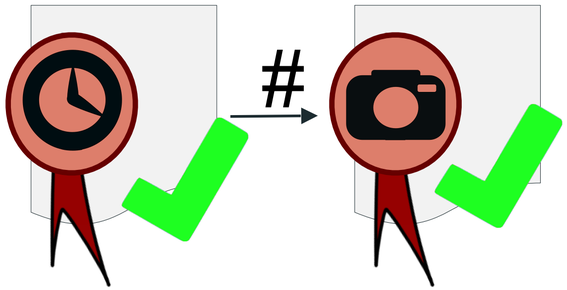
- authenticity
- integrity
- freshness
- thresholding
- survivable key compromise

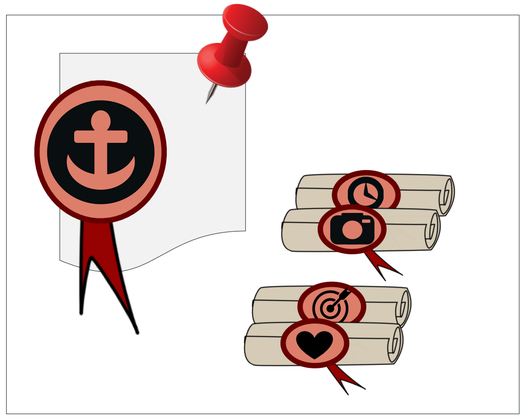


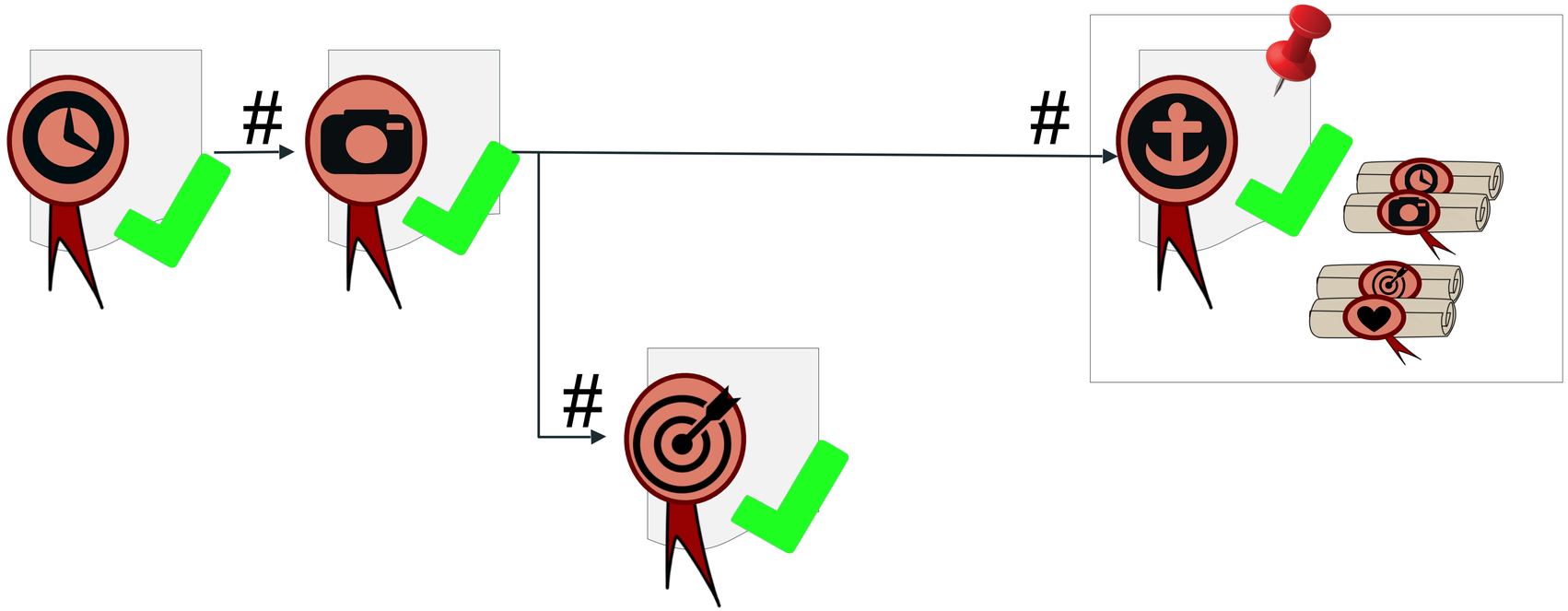


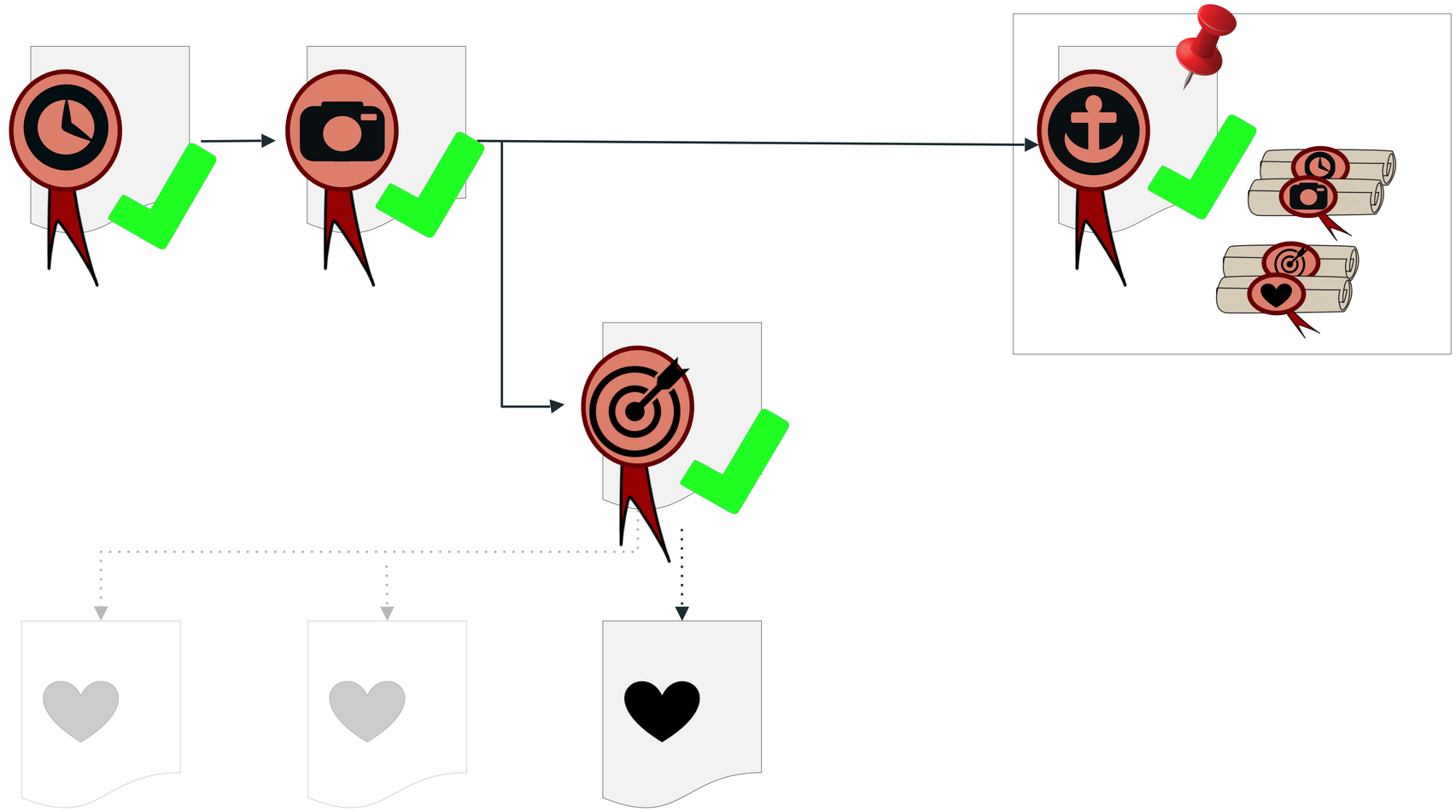


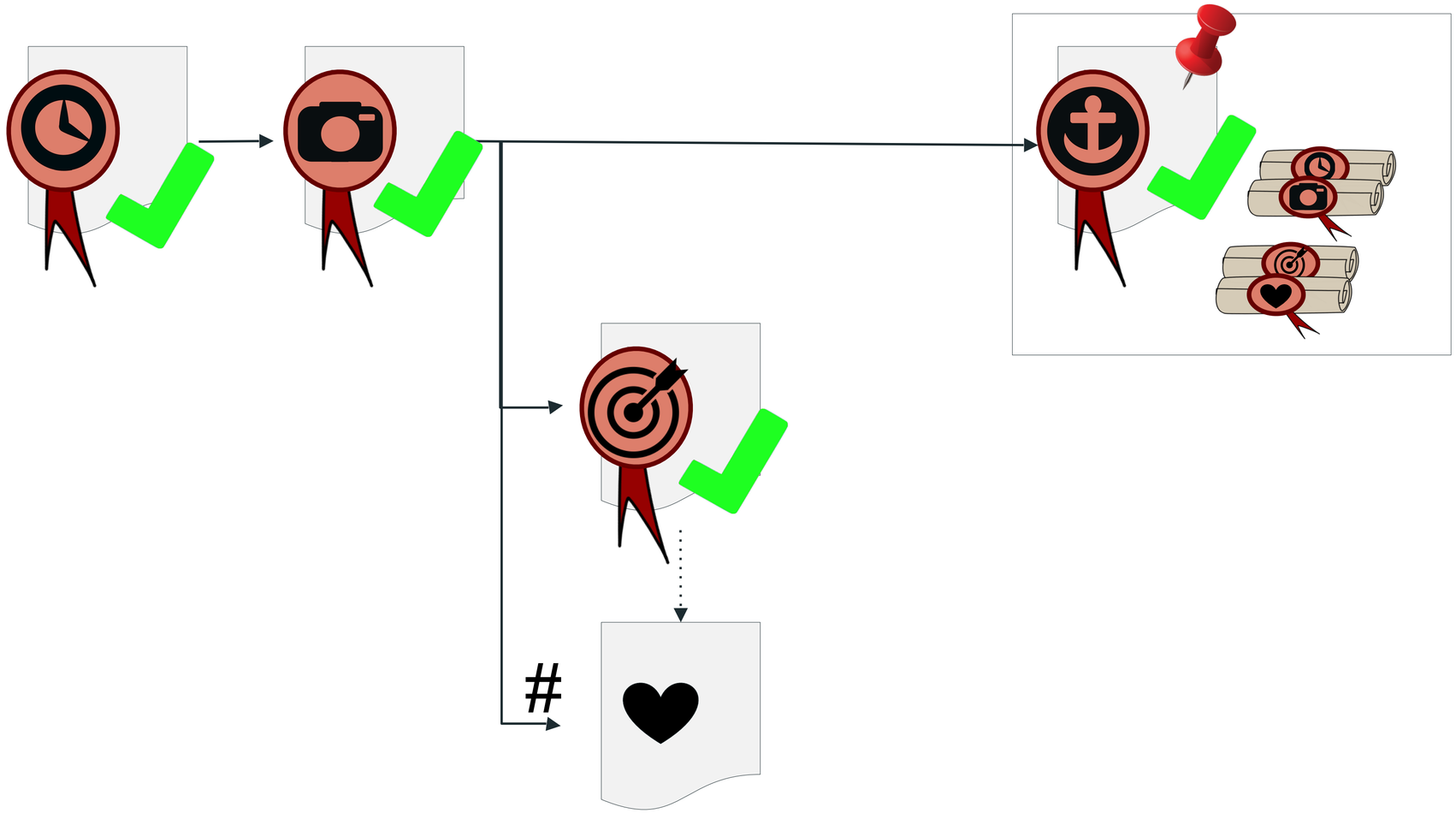


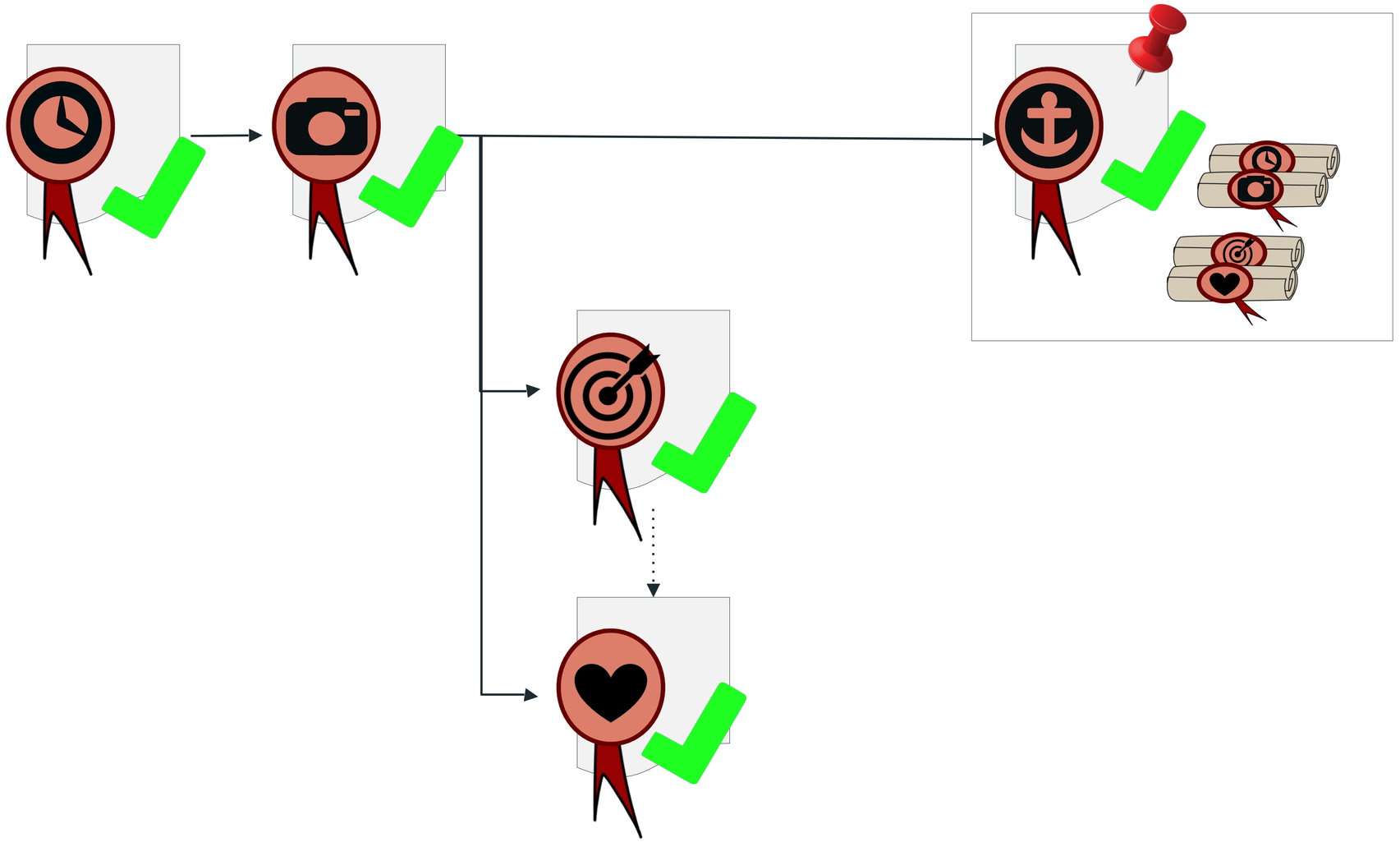


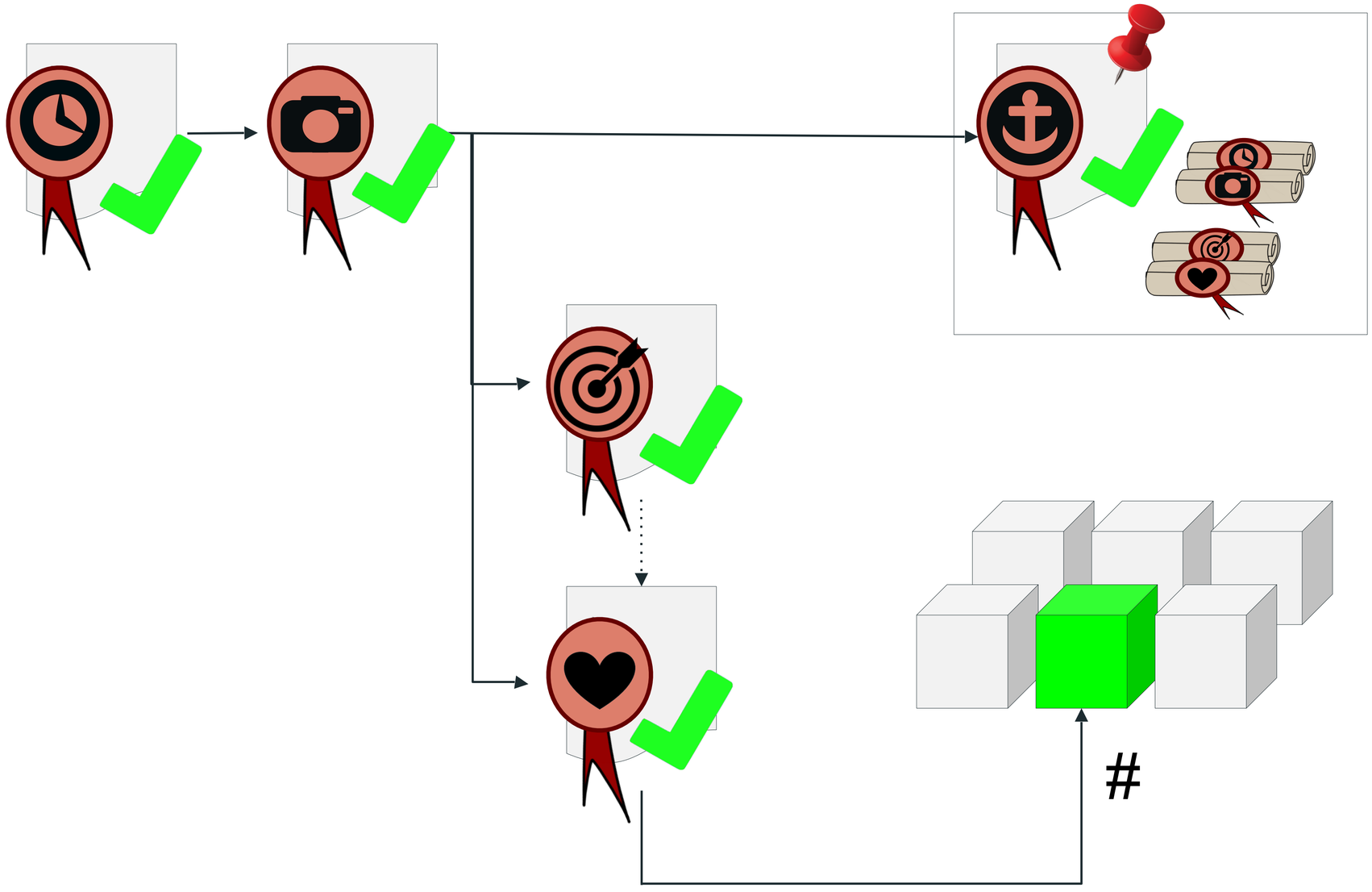




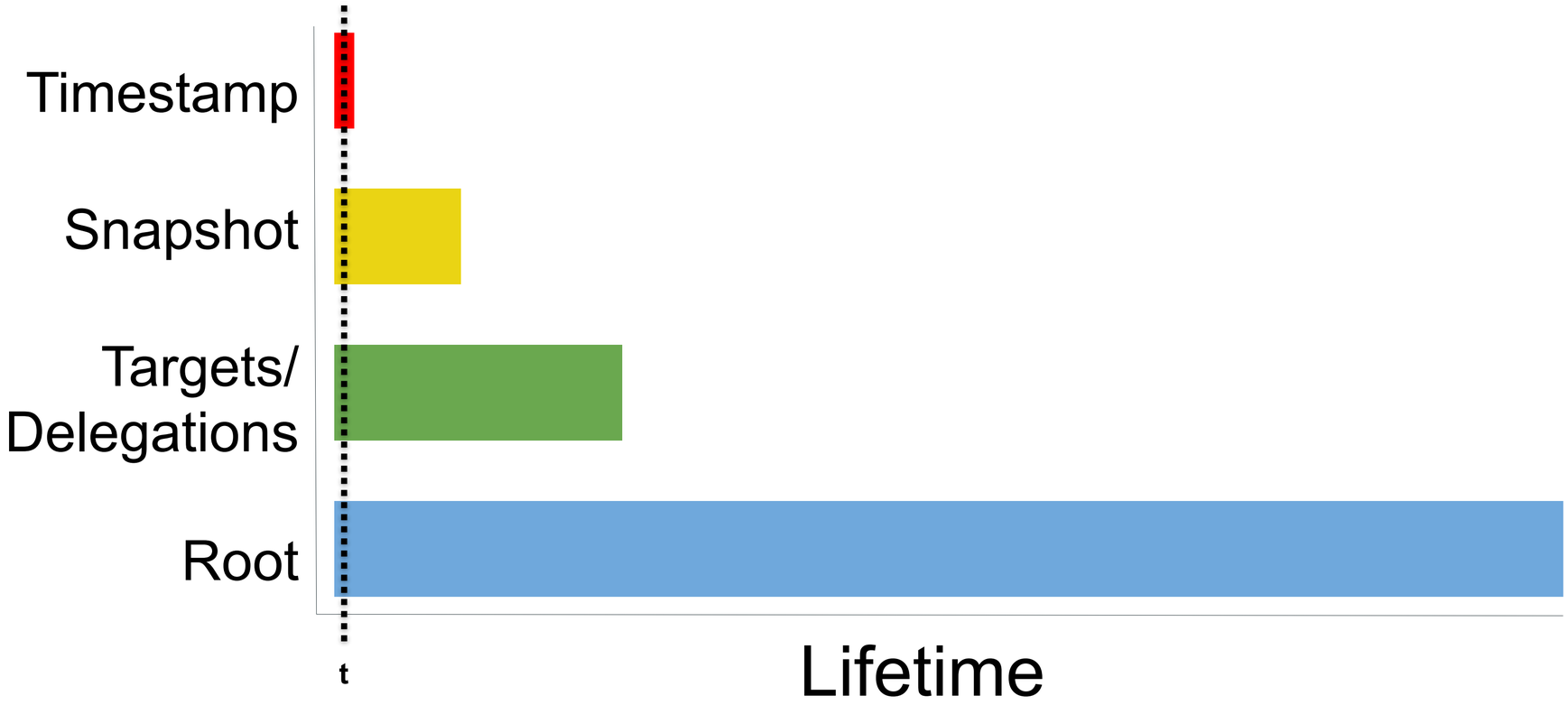




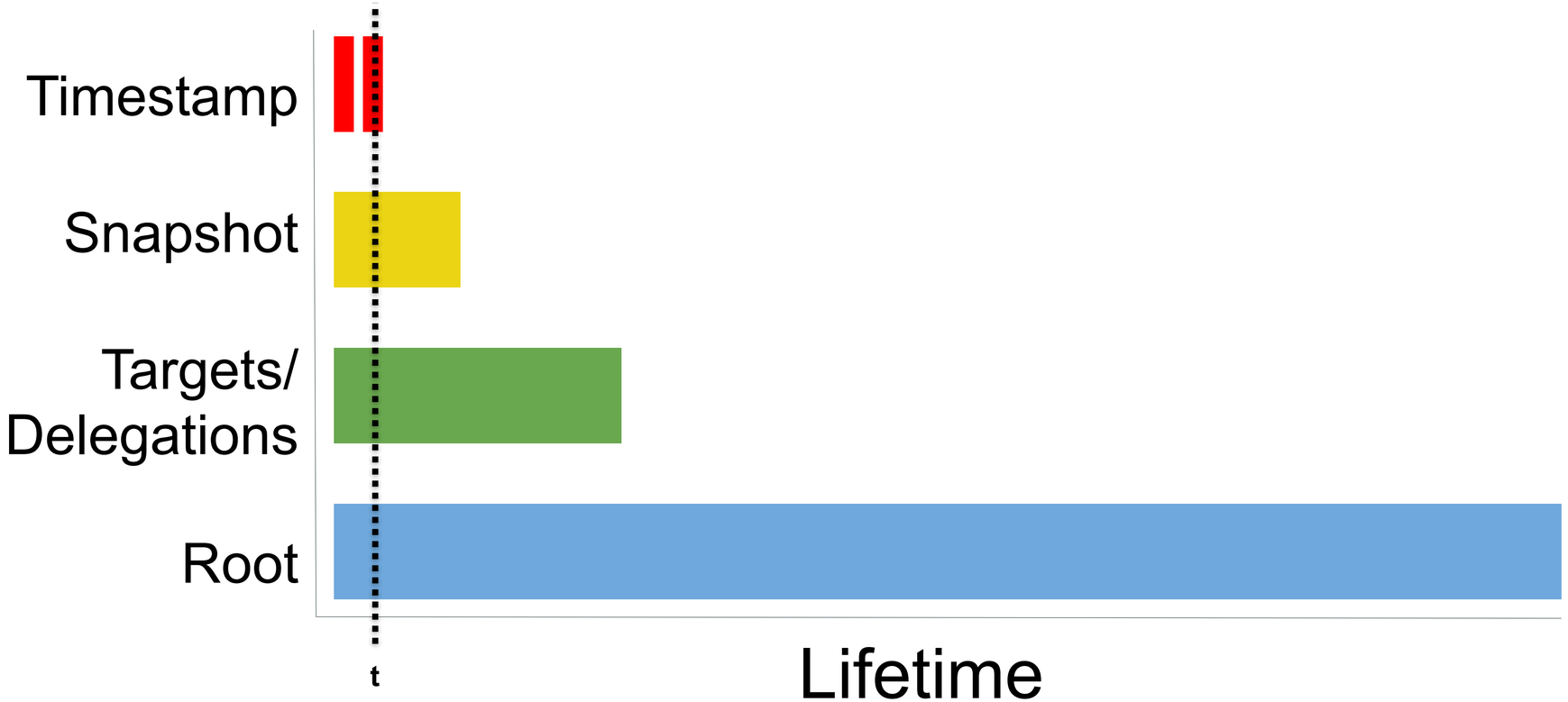




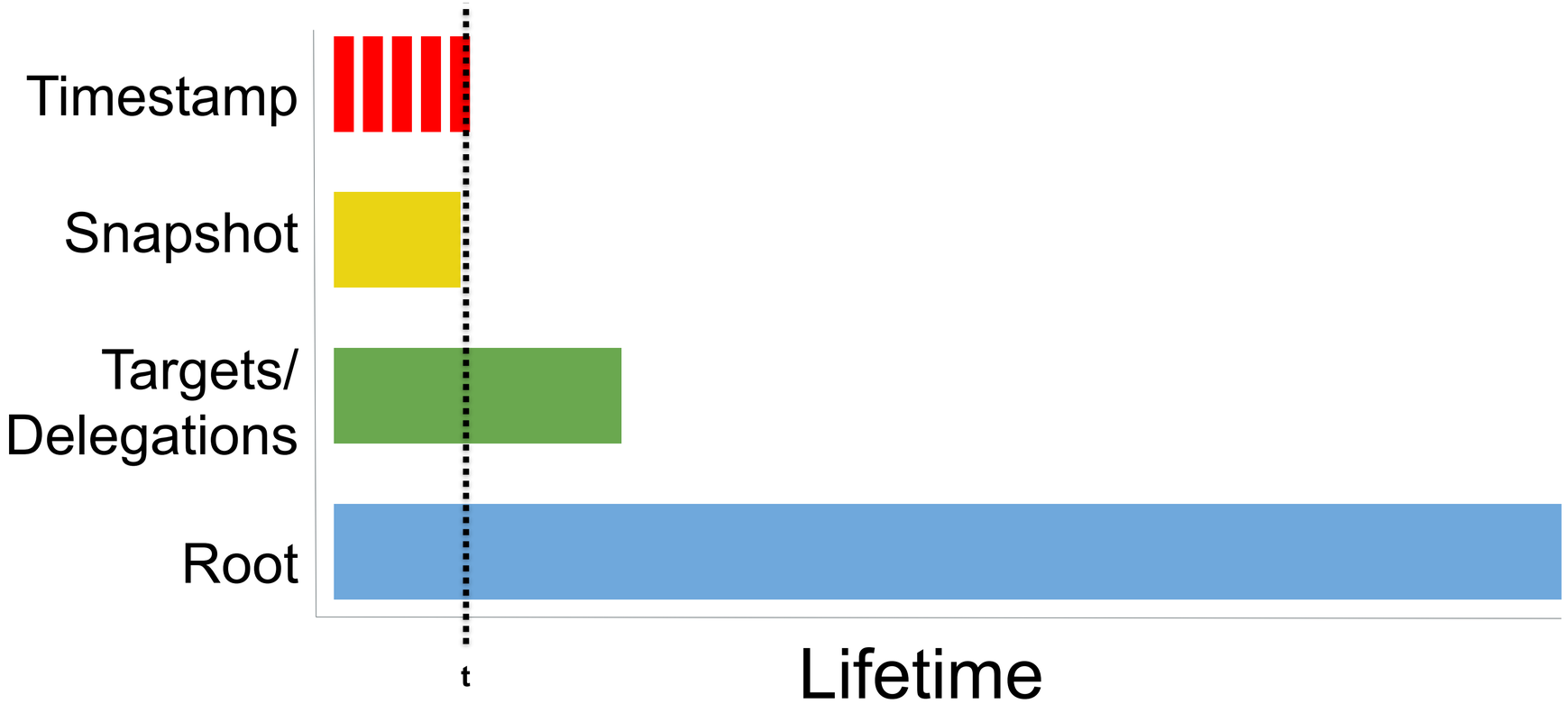
# Metadata Lifetime



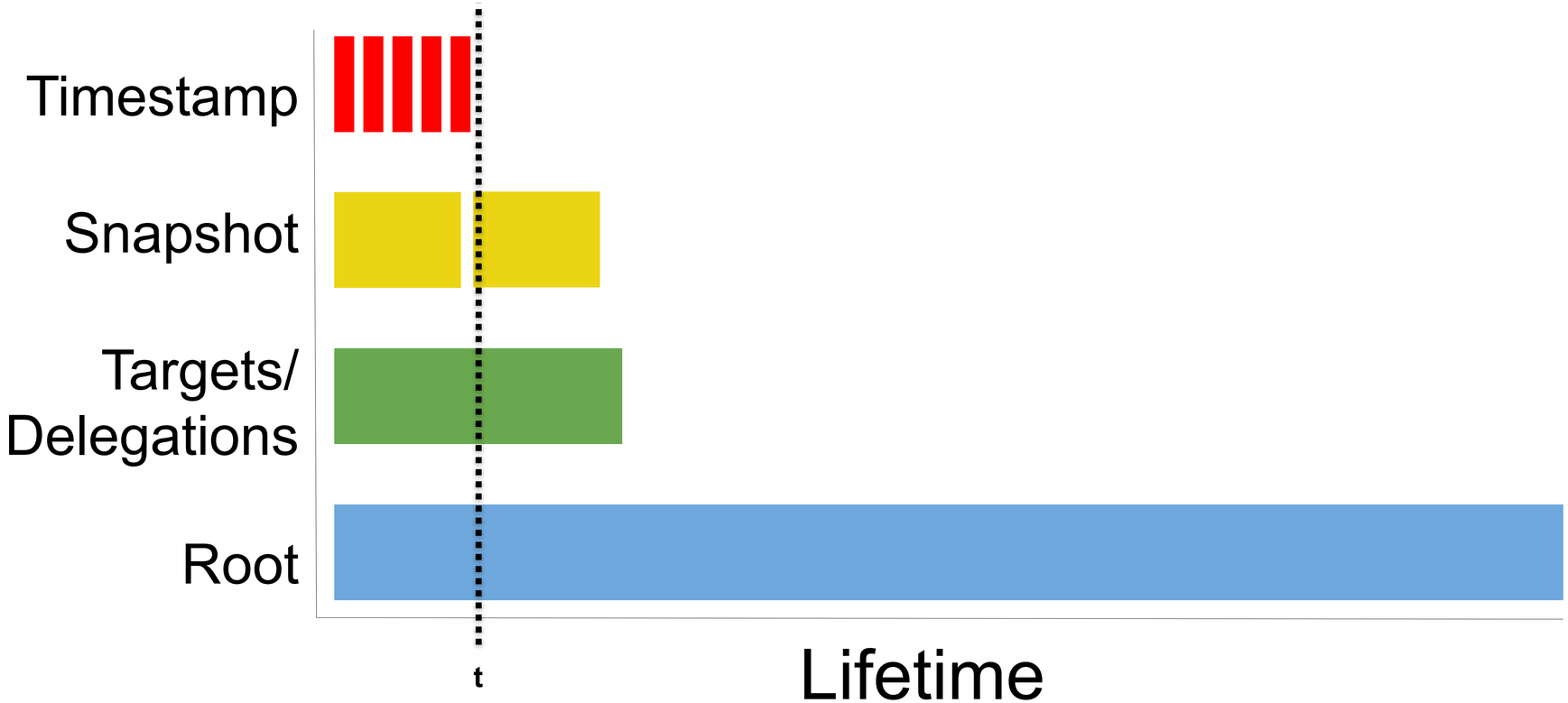
# Keeping Freshness



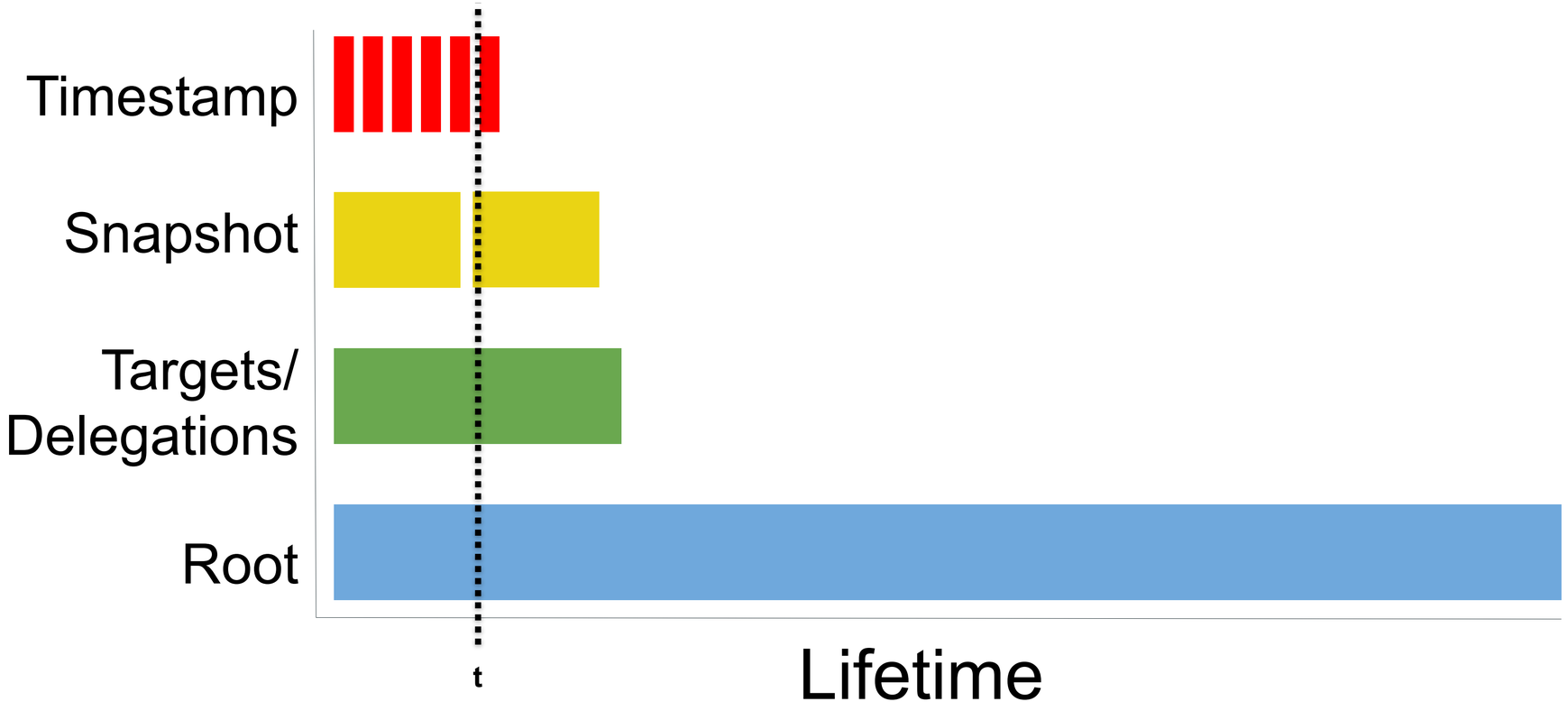
# Snapshot Expired!



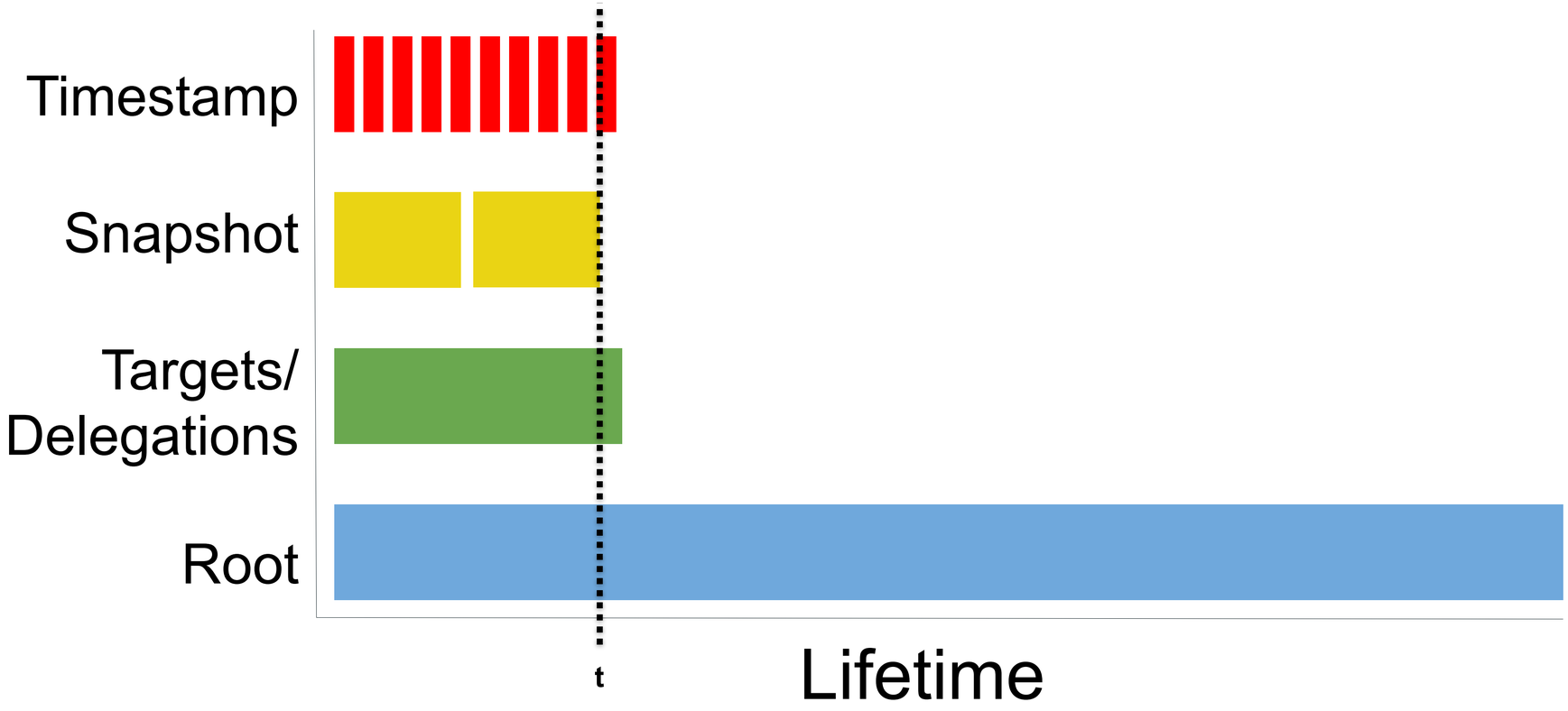
# Sign a new Snapshot



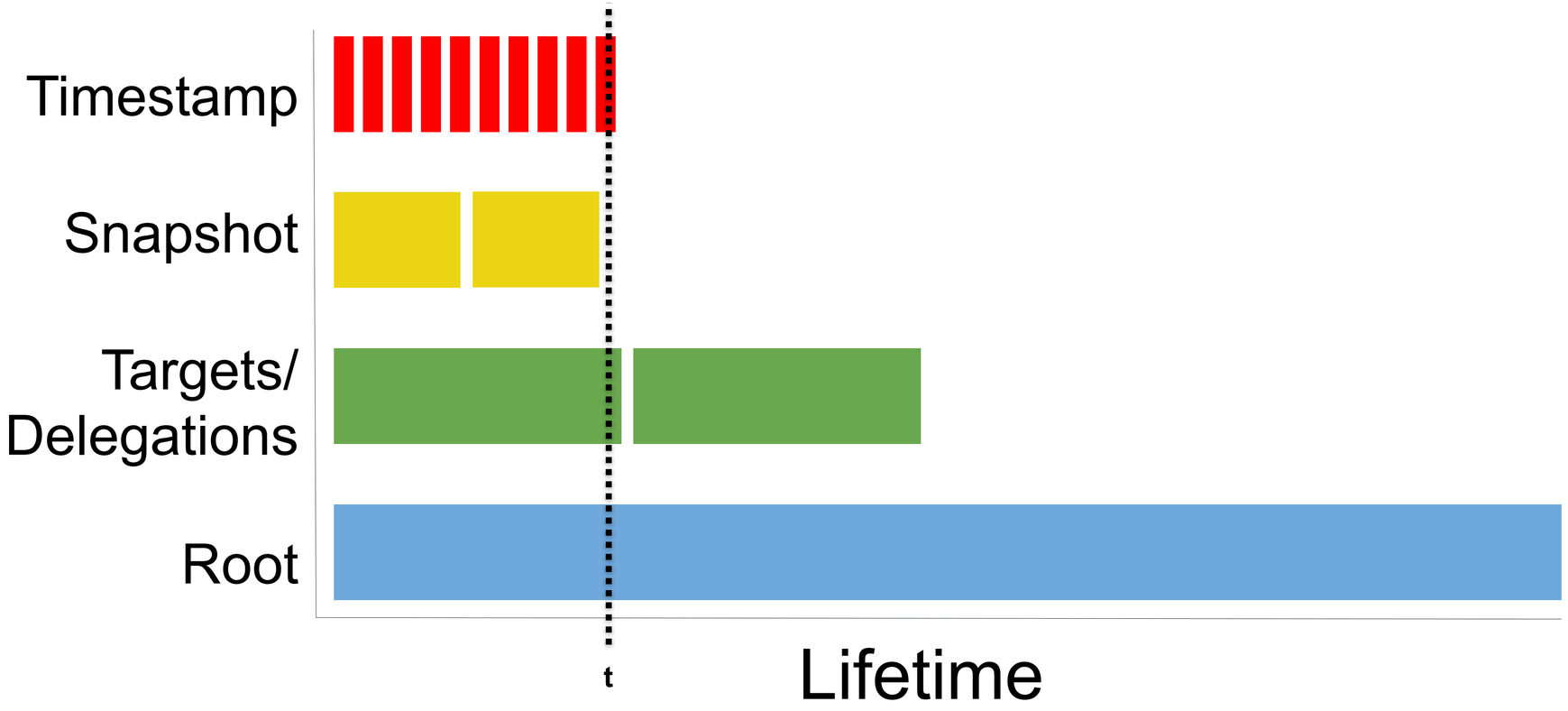
# Sign a new Timestamp to point the Snapshot



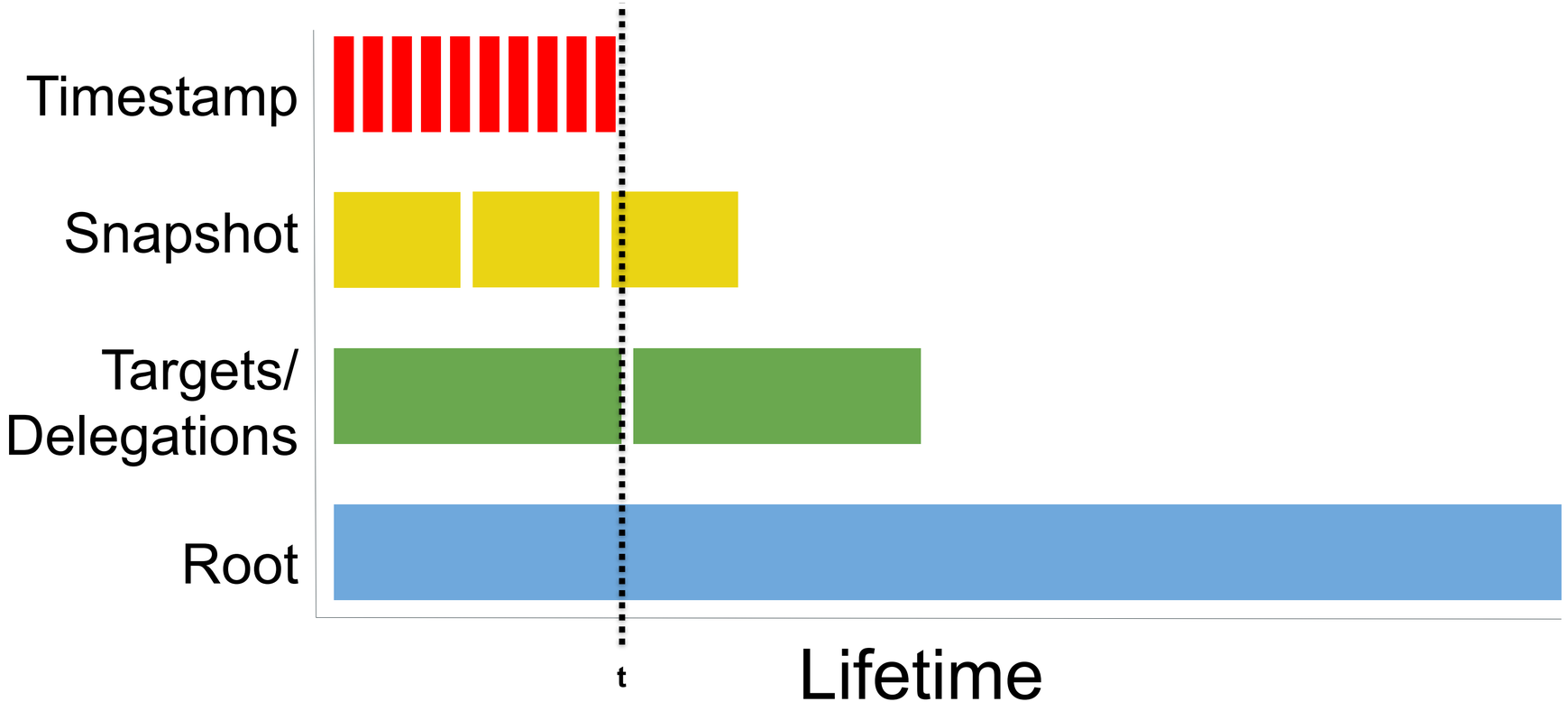
# Want to publish something?



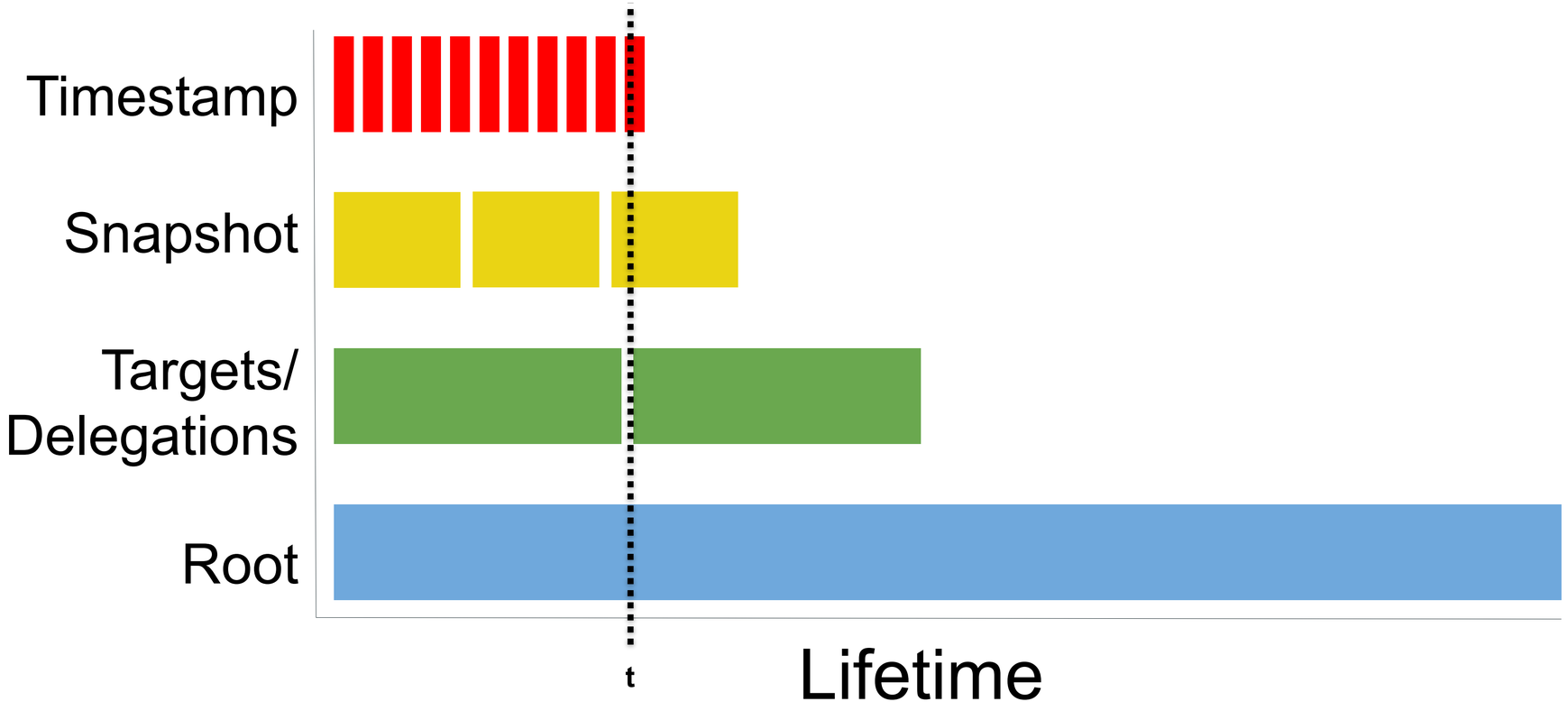
# Sign the hash into a new Targets or Delegation file



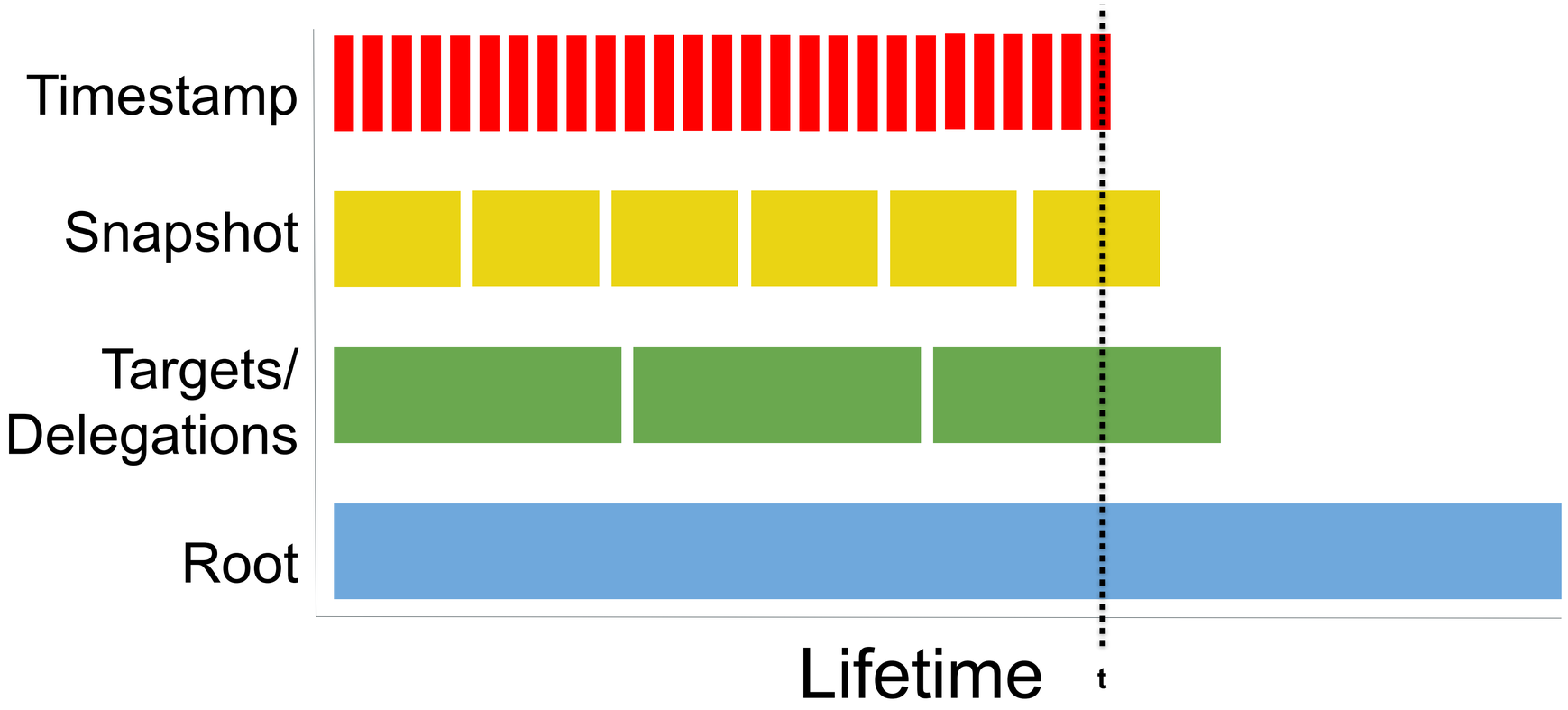
# Sign a new Snapshot that references this Targets file



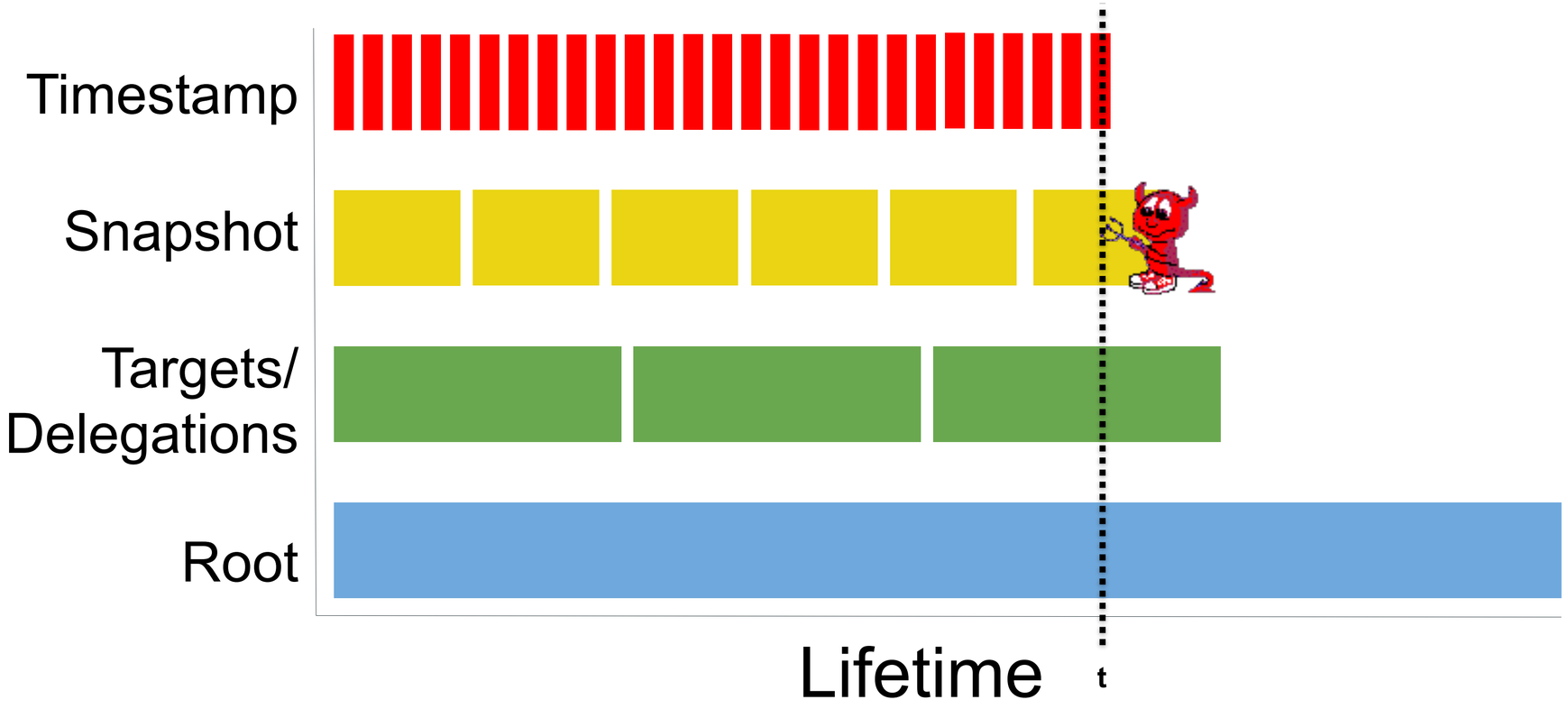
# Sign a new Timestamp that references the new Snapshot



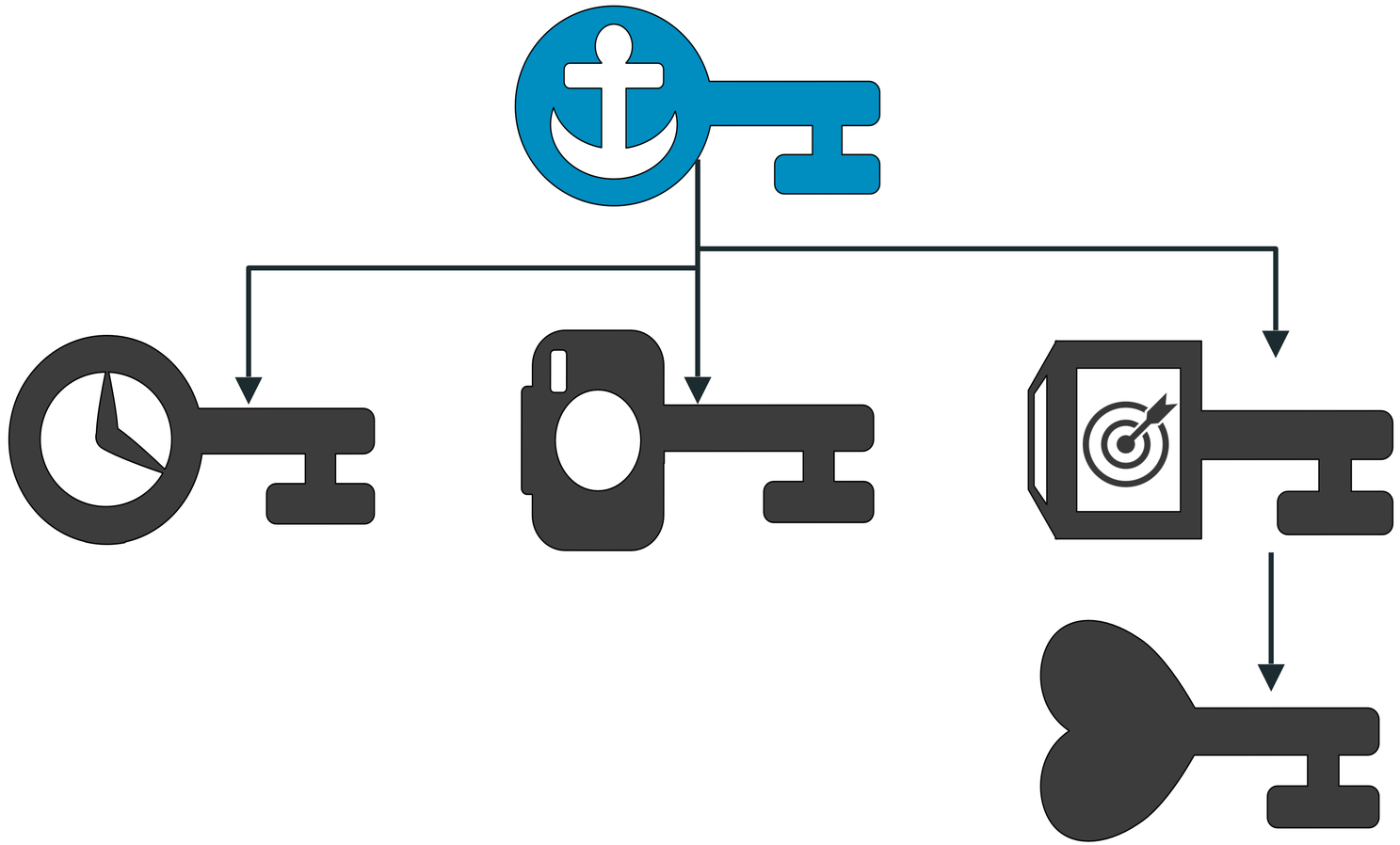
# Situation normal



# Oh no, I think my Snapshot key was compromised!



# Compromise is “when” not “if”



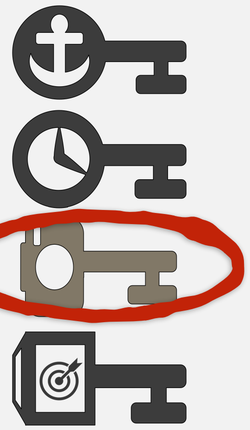
Root
Metadata

Root:

Timestamp:

Snapshot:

Targets:



Root
Metadata

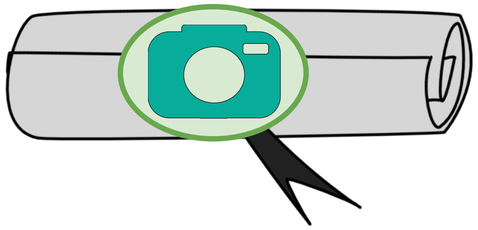
Root:

Timestamp:

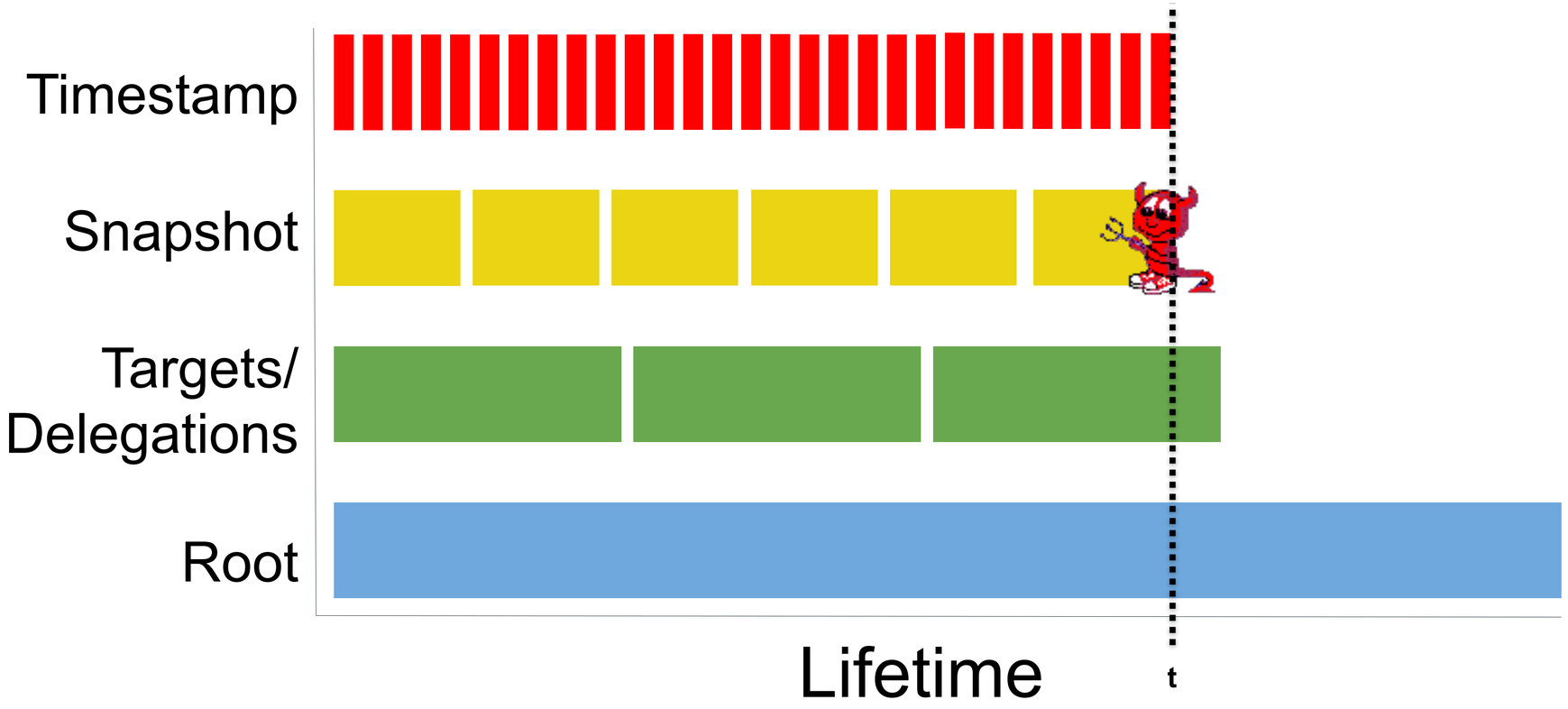
Snapshot:

Targets:

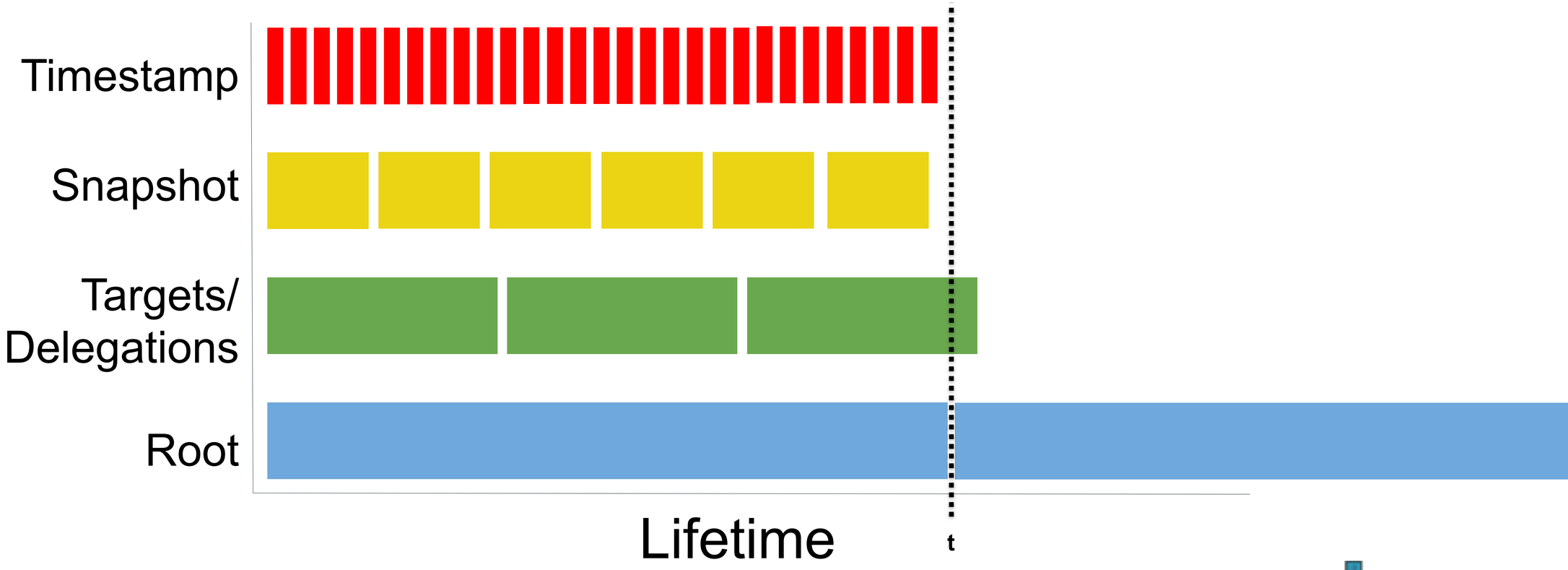
Snapshot
Metadata



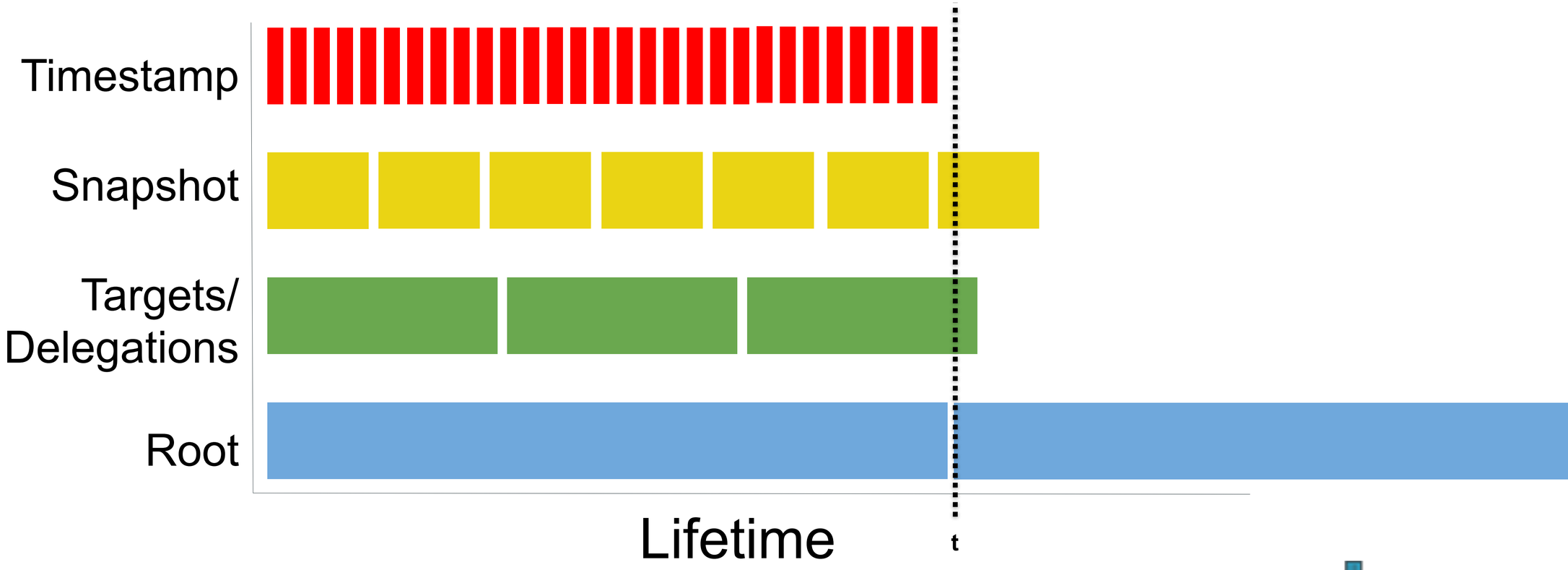
# Before recovery



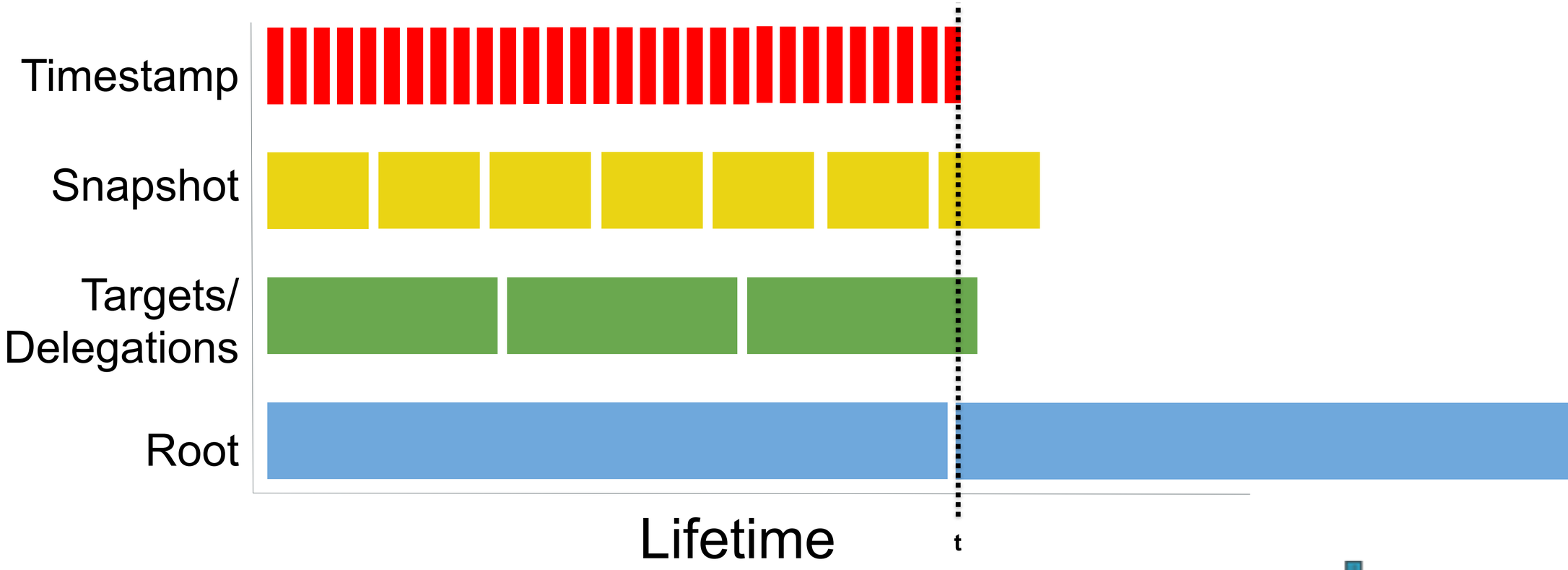
# Create and sign the new Snapshot key into Root



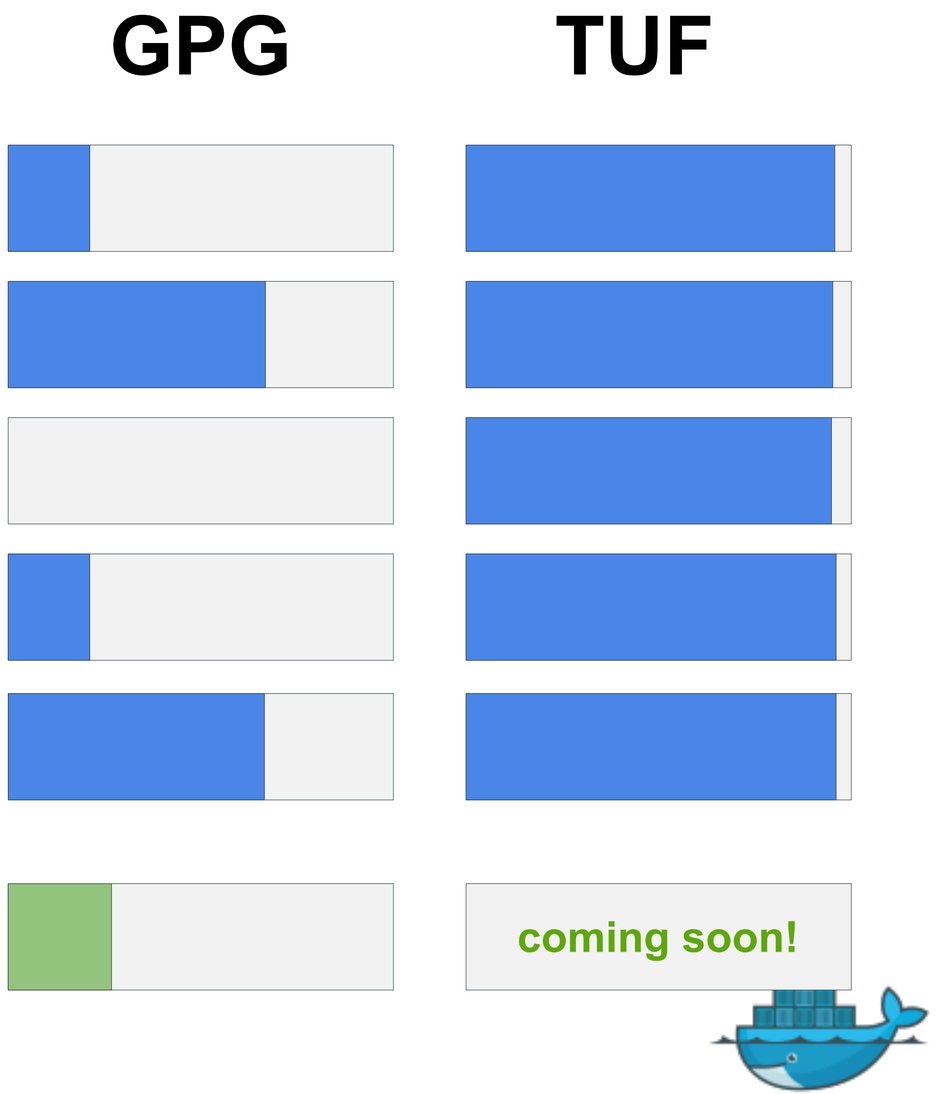
# Sign a new Snapshot with the new key



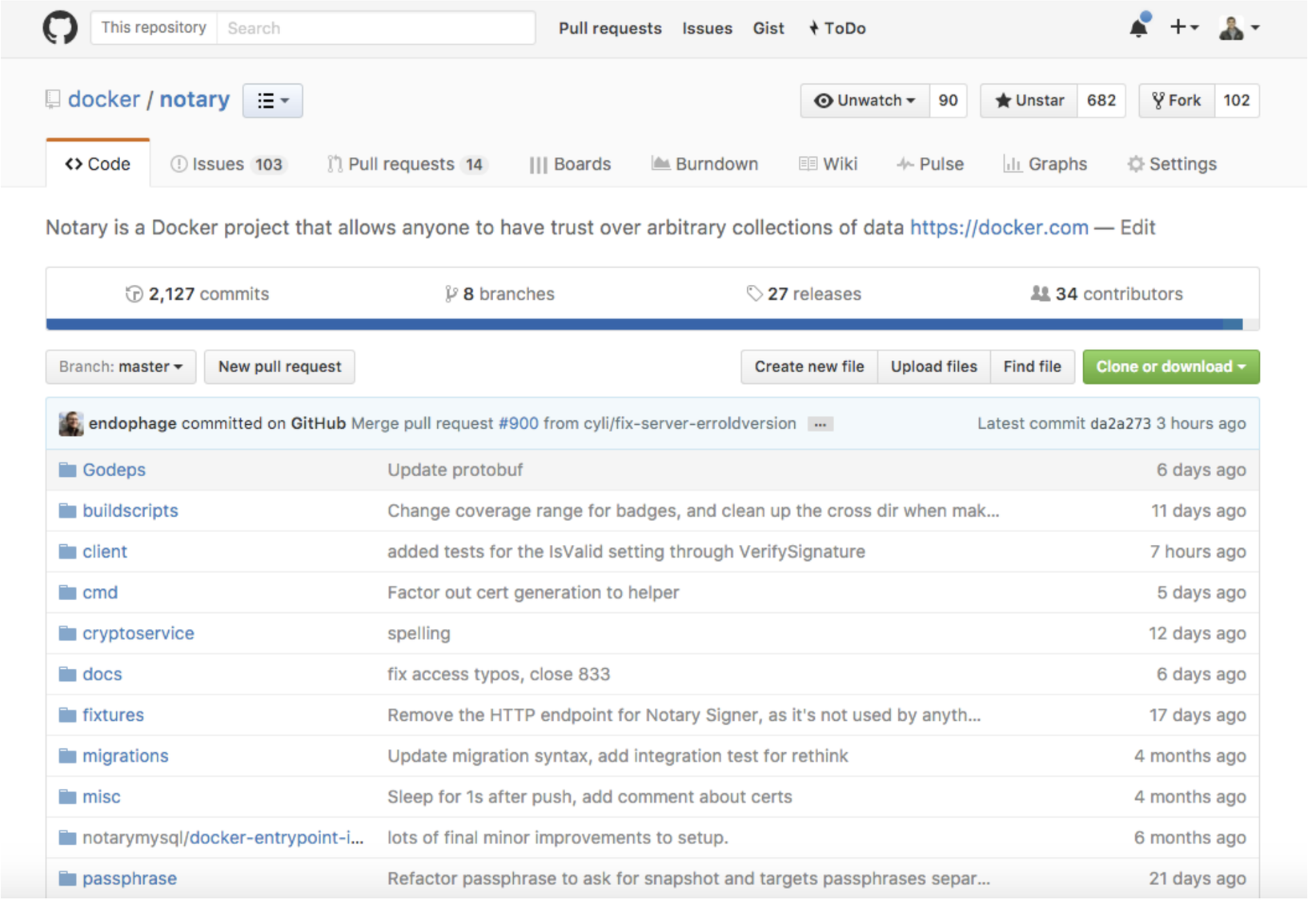
# Sign new Timestamp to reference new Snapshot



- authenticity
- integrity
- freshness
- thresholding
- survivable key compromise
- ease of use

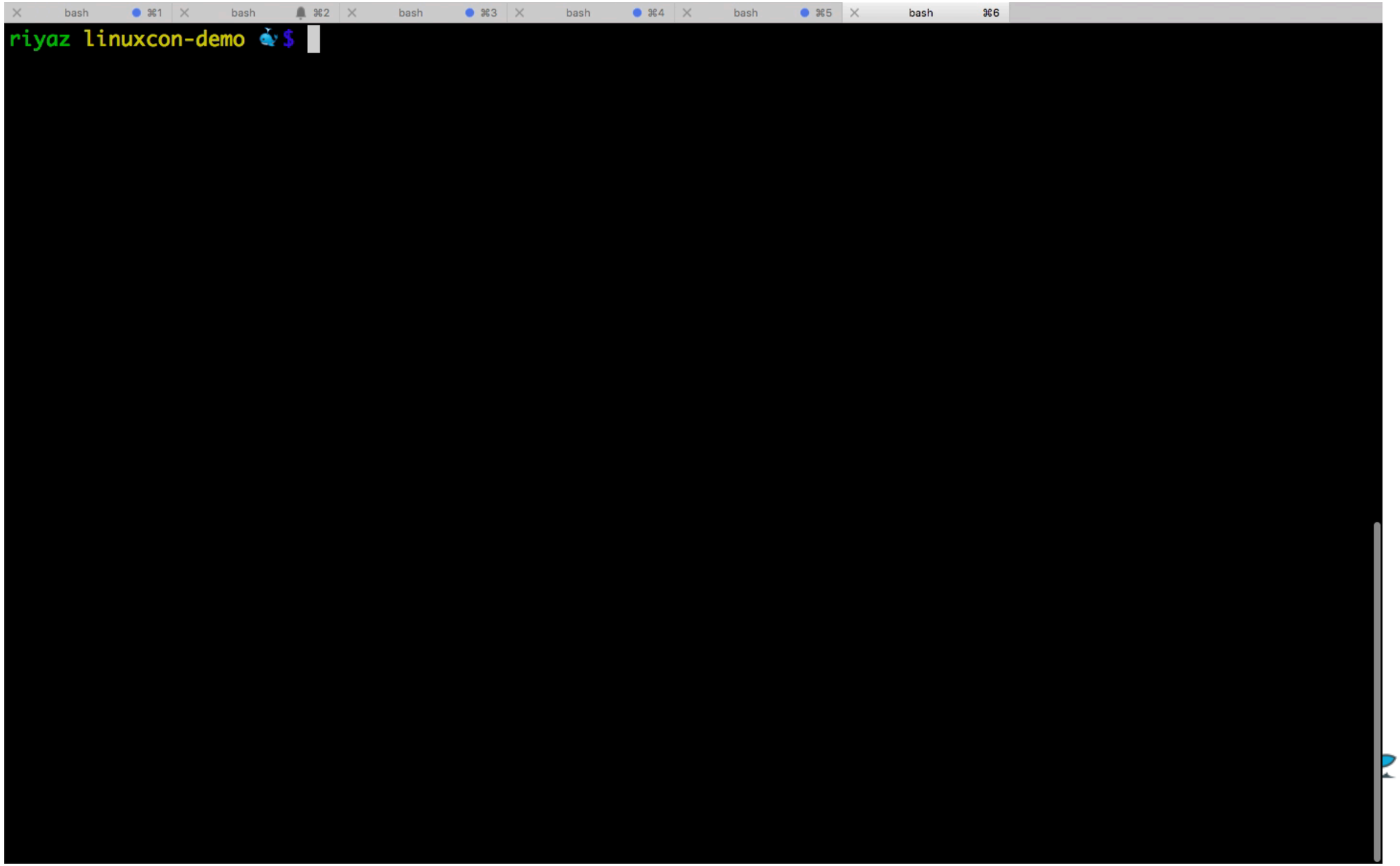


# How can we start using TUF?

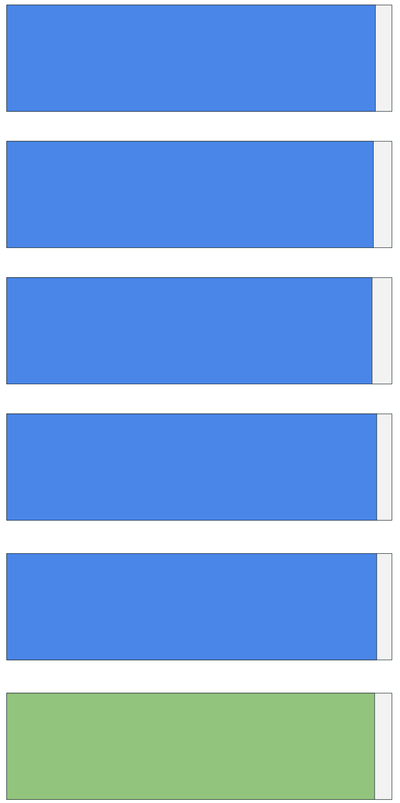

This repository
Search
Pull requests
Issues
Gist
ToDo
docker / notary
Unwatch 90
Unstar 682
Fork 102
Code
Issues 103
Pull requests 14
Boards
Burndown
Wiki
Pulse
Graphs
Settings
Notary is a Docker project that allows anyone to have trust over arbitrary collections of data https://docker.com — Edit
2,127 commits
8 branches
27 releases
34 contributors
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
endophage committed on GitHub Merge pull request #900 from cyli/fix-server-erroldversion
Latest commit da2a273 3 hours ago
Godeps Update protobuf 6 days ago
buildscripts Change coverage range for badges, and clean up the cross dir when mak... 11 days ago
client added tests for the IsValid setting through VerifySignature 7 hours ago
cmd Factor out cert generation to helper 5 days ago
cryptoservice spelling 12 days ago
docs fix access typos, close 833 6 days ago
fixtures Remove the HTTP endpoint for Notary Signer, as it's not used by anyth... 17 days ago
migrations Update migration syntax, add integration test for rethink 4 months ago
misc Sleep for 1s after push, add comment about certs 4 months ago
notarymysql/docker-entrypoint-i... lots of final minor improvements to setup. 6 months ago
passphrase Refactor passphrase to ask for snapshot and targets passphrases separ... 21 days ago

# Demo

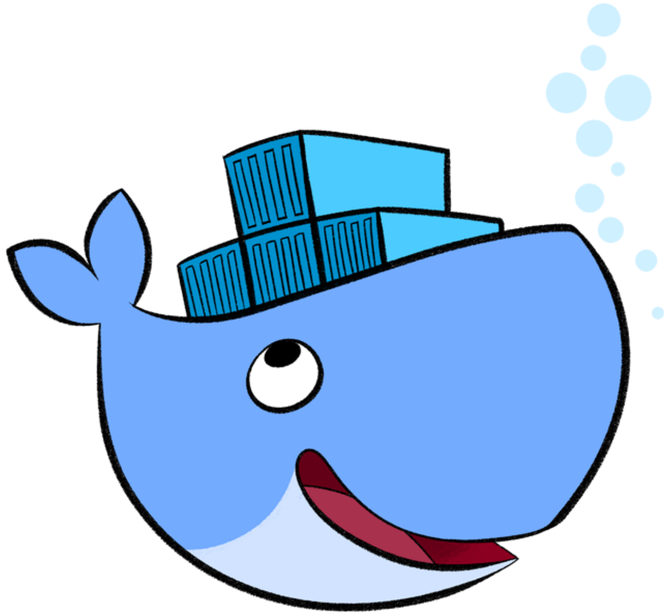
- ease of use?


bash ⌘1
bash ⌘2
bash ⌘3
bash ⌘4
bash ⌘5
bash ⌘6
riyaz linuxcon-demo $

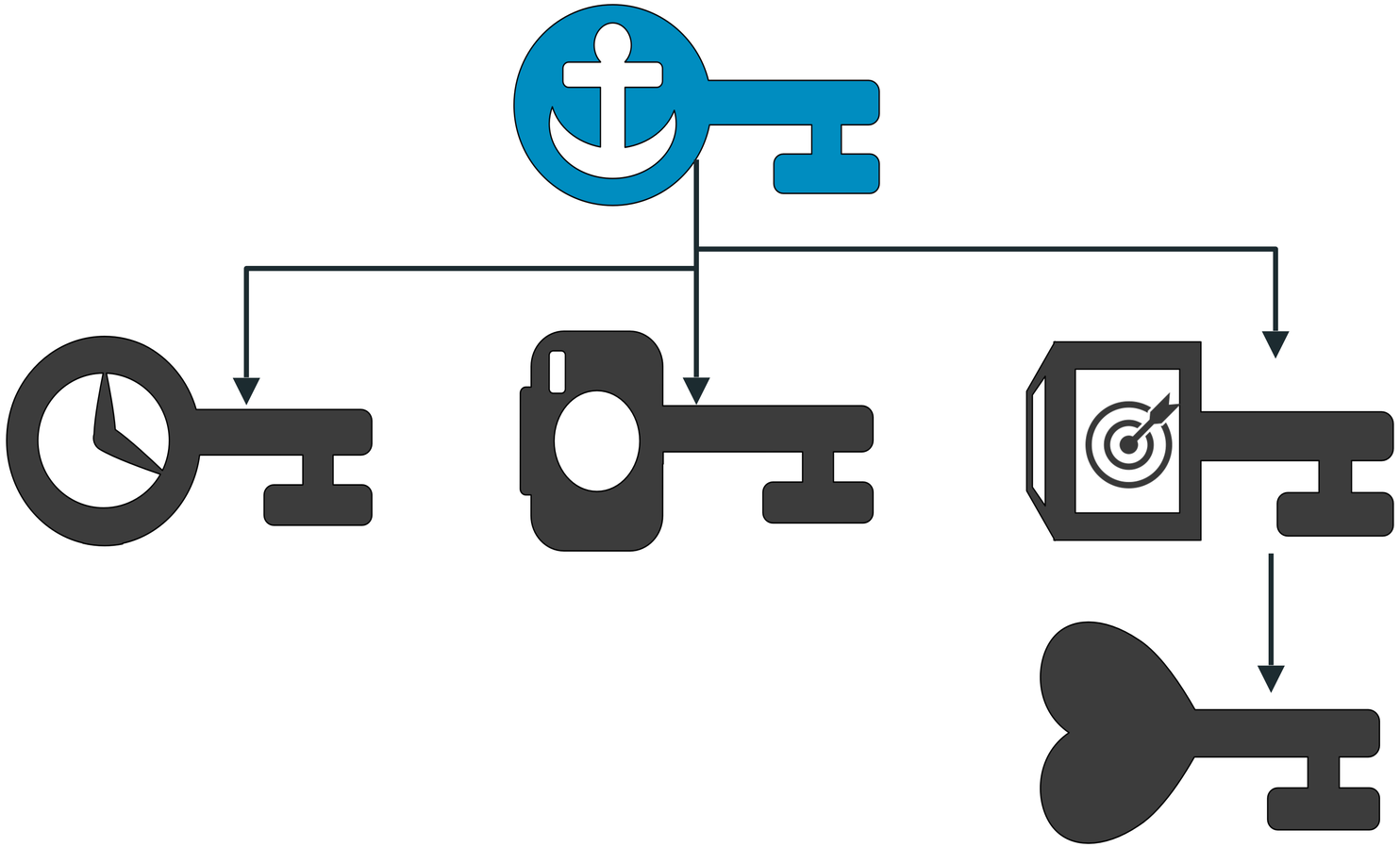
- authenticity
- integrity
- freshness
- thresholding
- survivable key compromise
- ease of use

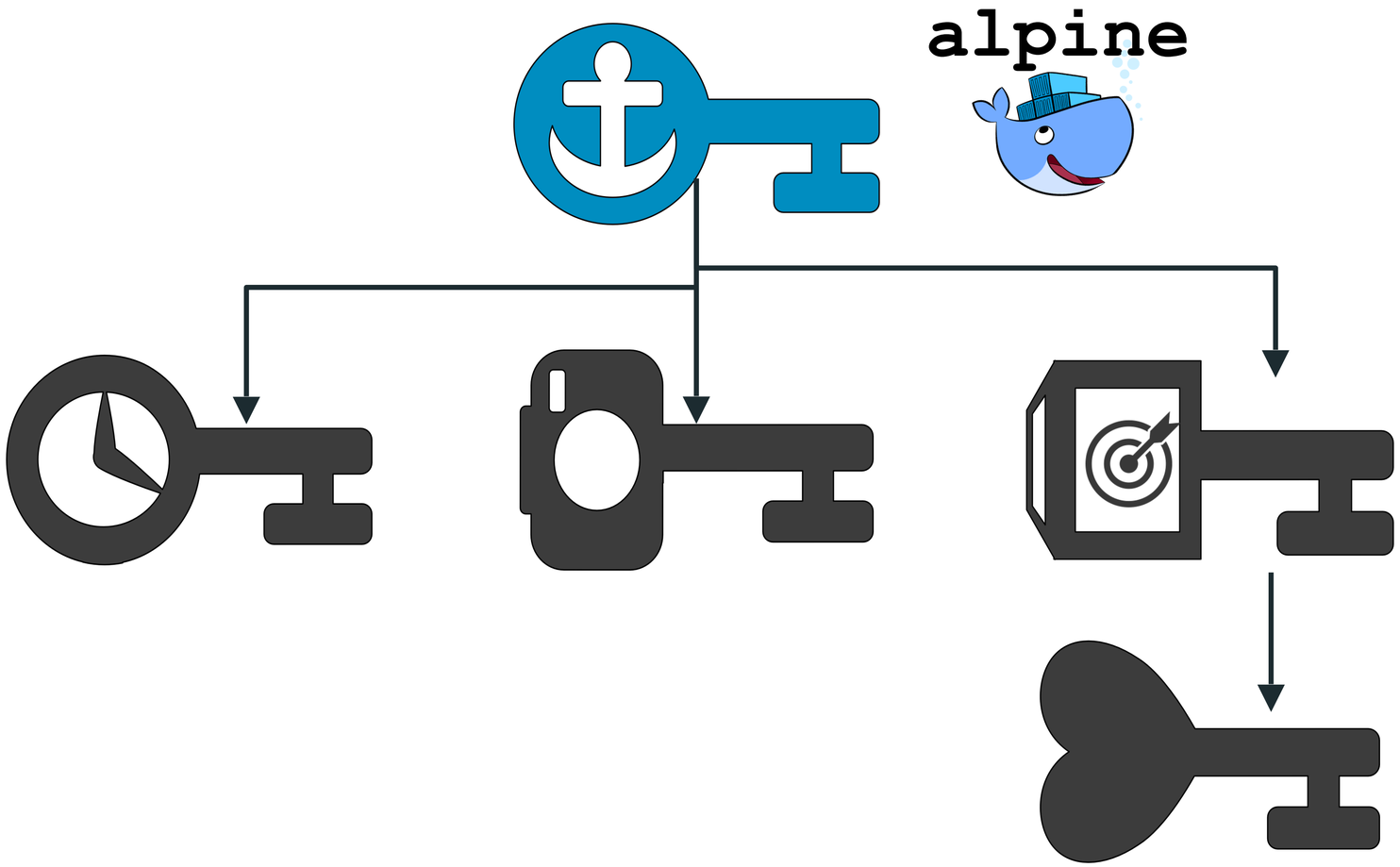


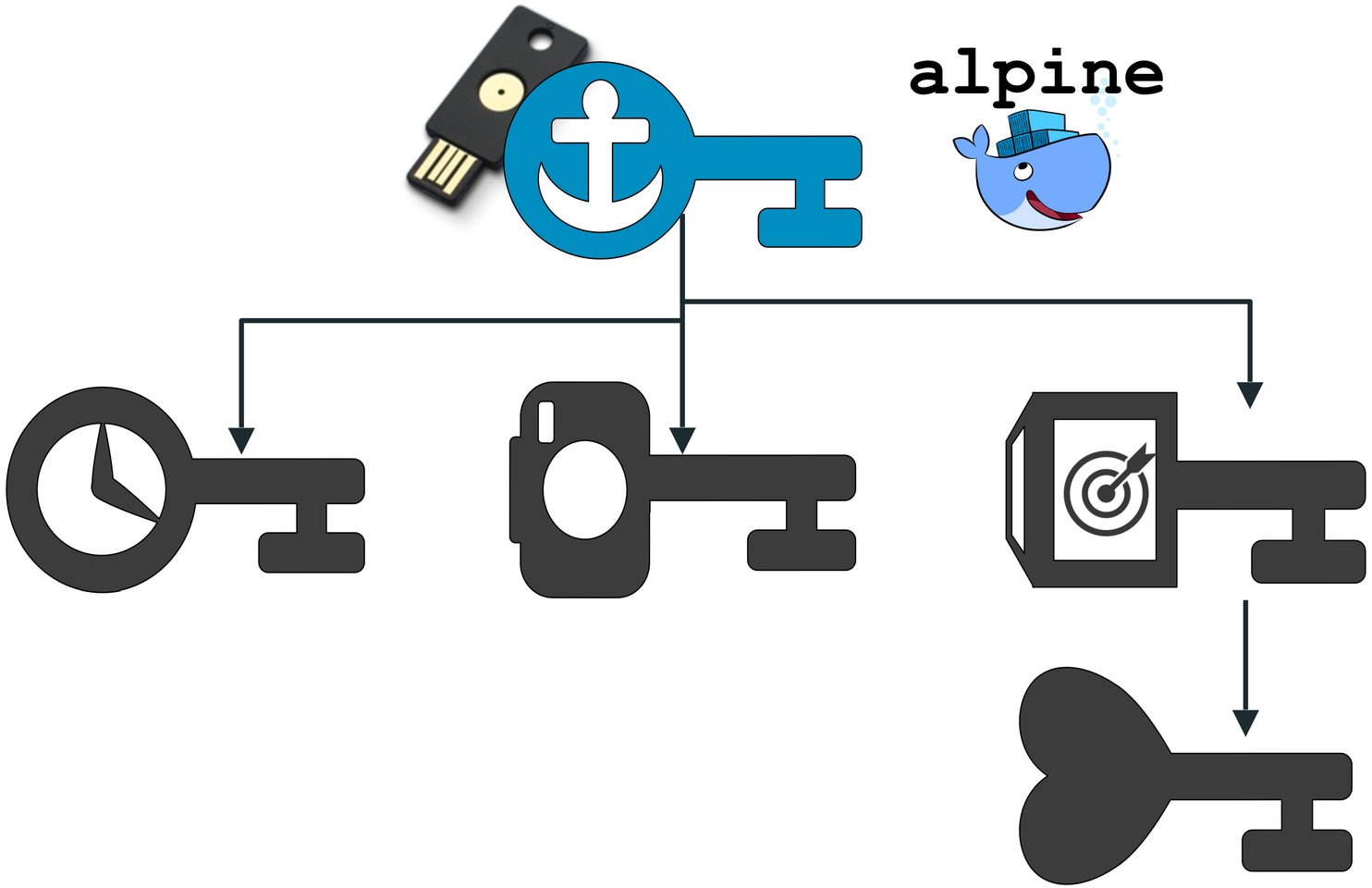
github.com/docker/notary

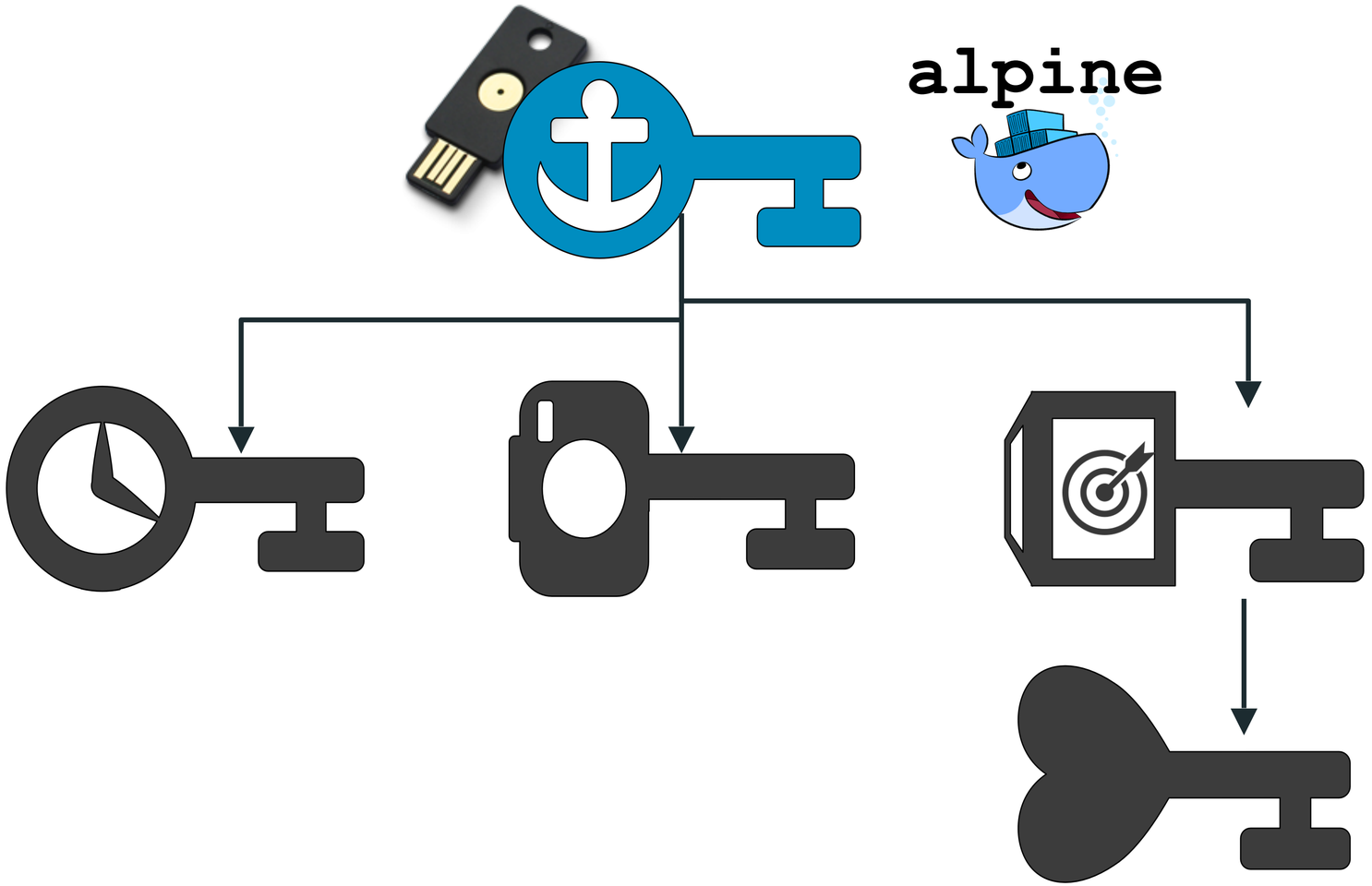


IN
WE TRUST




alpine


alpine




```
latest:  {hash}
edge:    {hash}
2.6:     {hash}
3.3:     {hash}
3.4:     {hash}
```

```
$> export DOCKER_CONTENT_TRUST=1
```

# TODO(riyazdf): DCT Demo Video

```
$> $pkg-manager install openssl
```

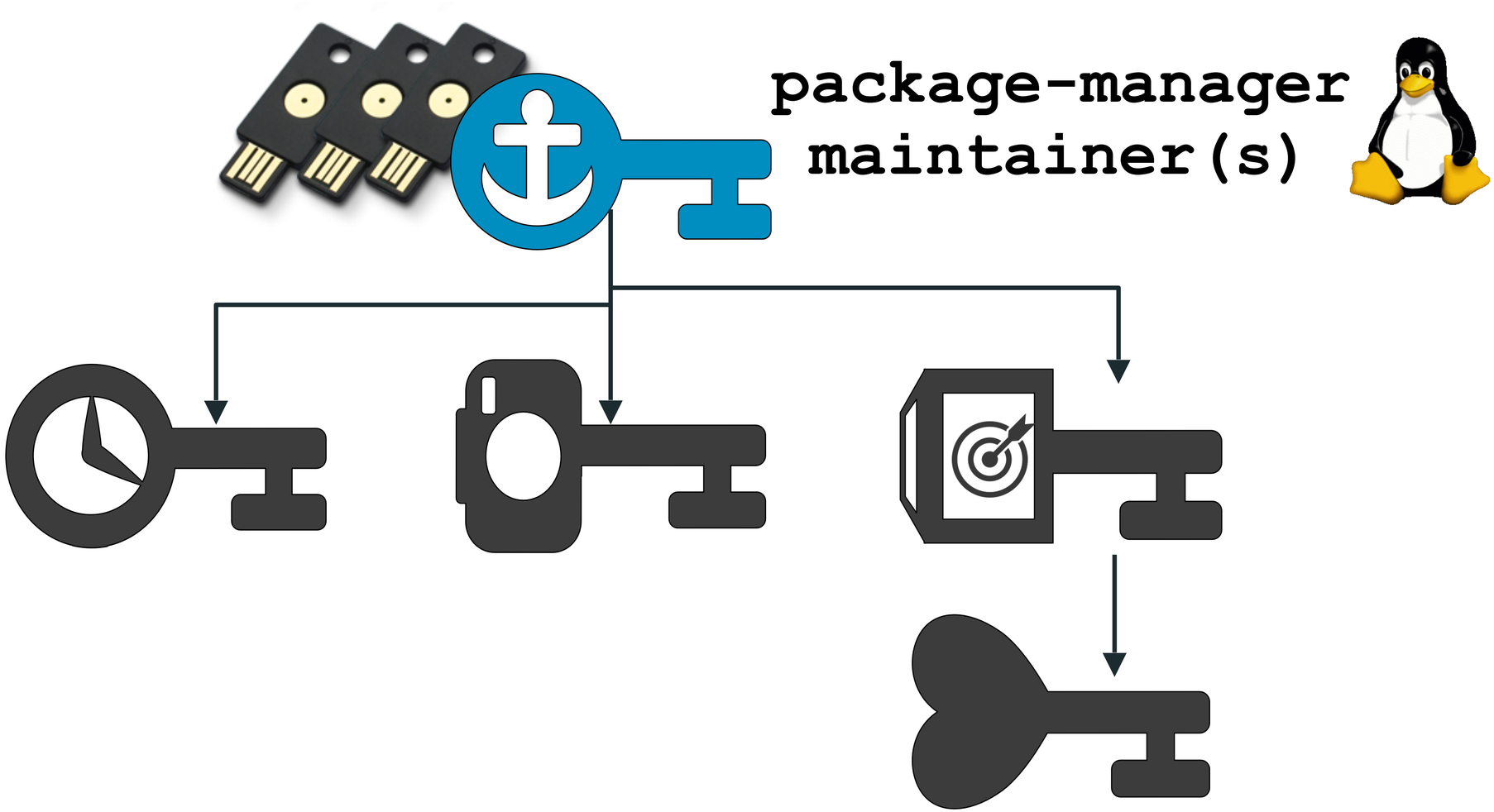
# Design Goals:

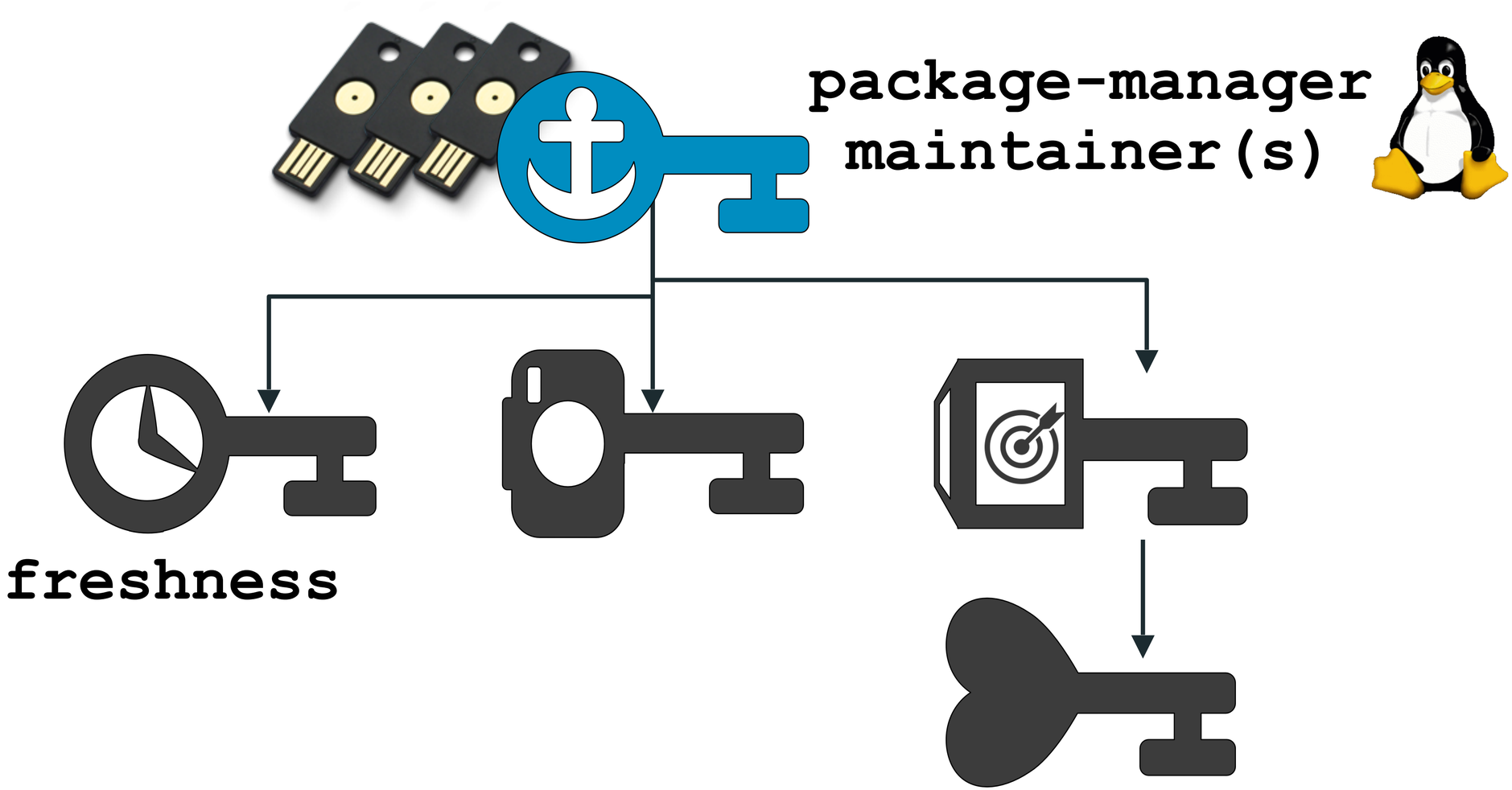
- root of trust in package manager maintainers
  - *with thresholding*

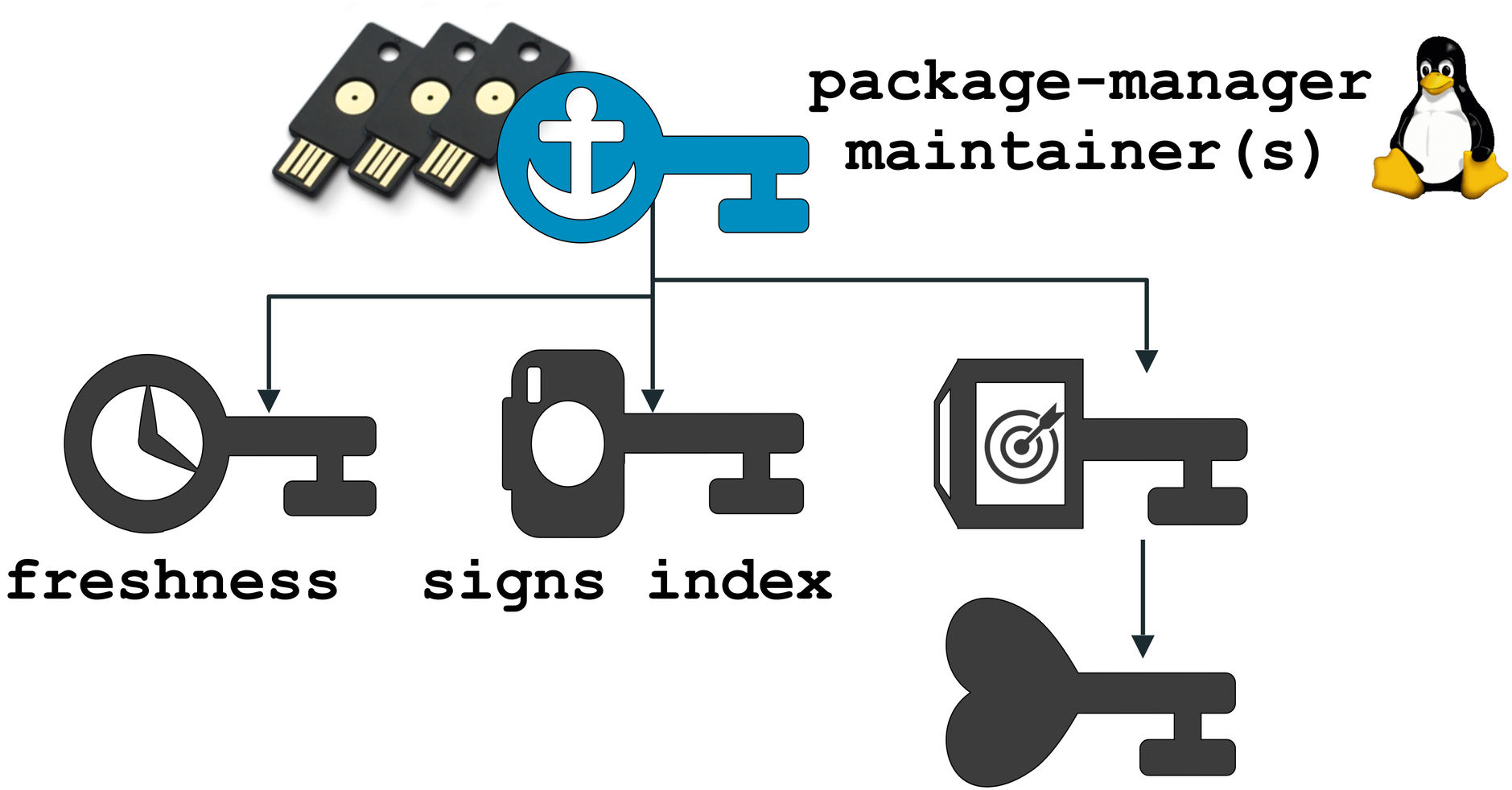
- freshness guarantees

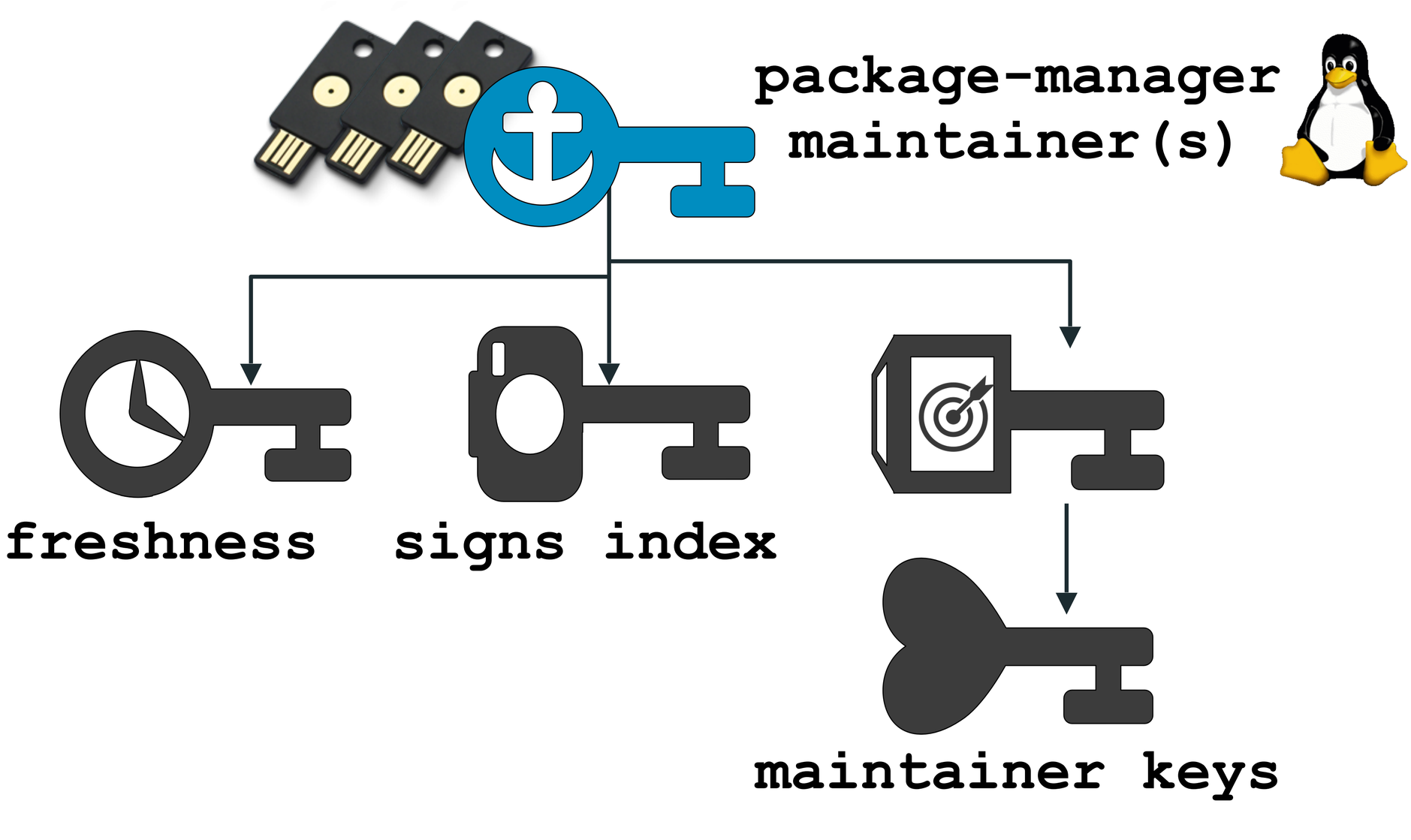
- signed index of all packages

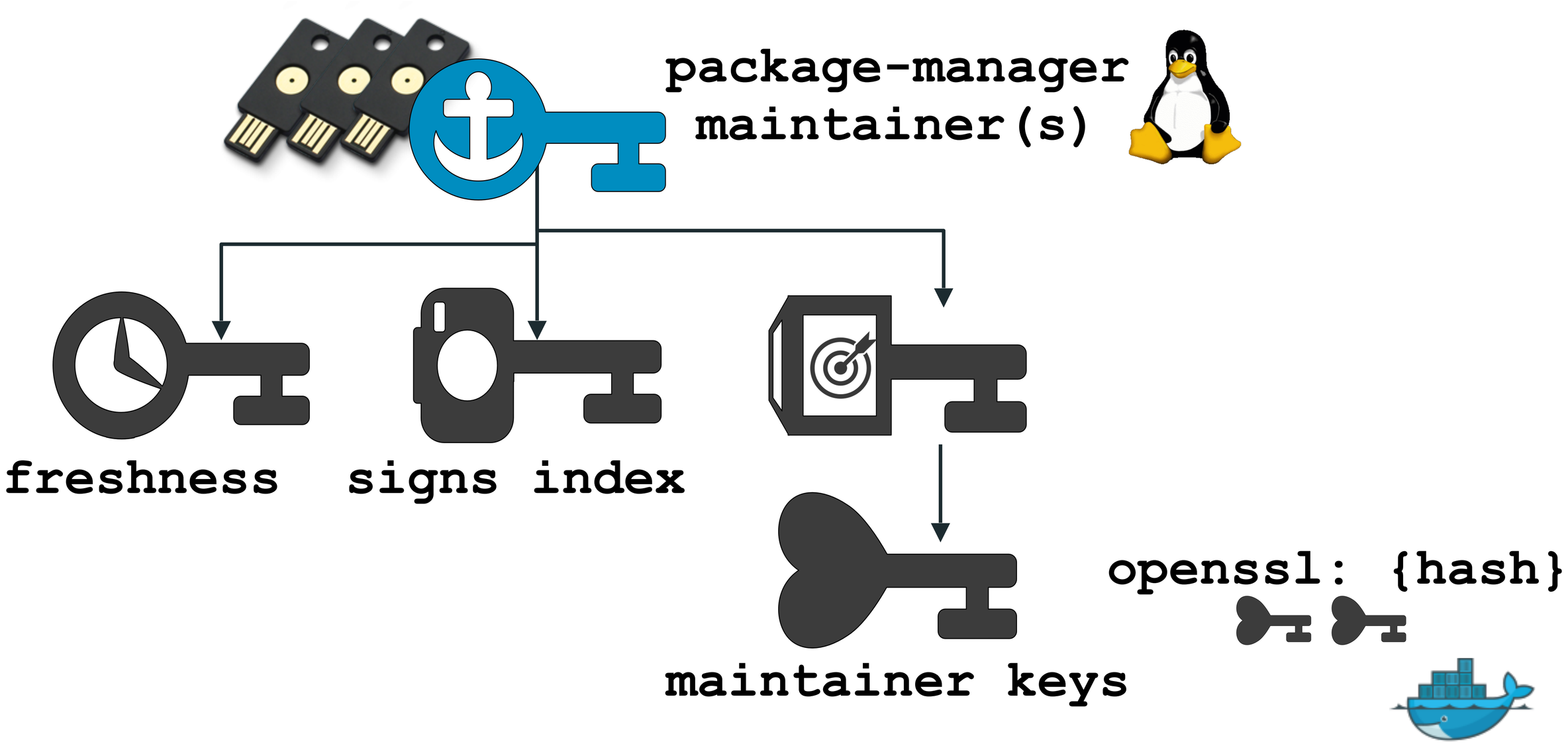
- package targets signed by package maintainers
  - *with thresholding*
  - name to hash resolution


package-manager
maintainer(s)


package-manager
maintainer(s)
freshness


package-manager
maintainer(s)
freshness
signs
index


package-manager
maintainer(s)
freshness
signs
index
maintainer keys

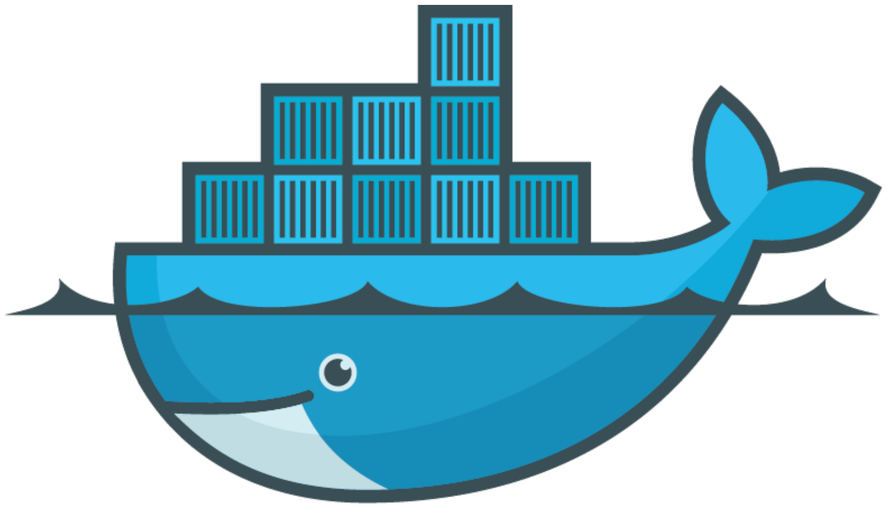

package-manager
maintainer(s)
freshness
signs index
maintainer keys
openssl: {hash}

# Future work: **hermetic builds**

```
FROM golang:1.7.1

RUN apt-get update && apt-get install -y \
        curl \
        clang \
        libltdl-dev \
        libsqlite3-dev \
        patch \
        tar \
        xz-utils \
        python \
        python-pip \
        --no-install-recommends \
        && rm -rf /var/lib/apt/lists/*

RUN useradd -ms /bin/bash notary \
        && pip install codecov \
        && go get golang.org/x/tools/cmd/cover
```

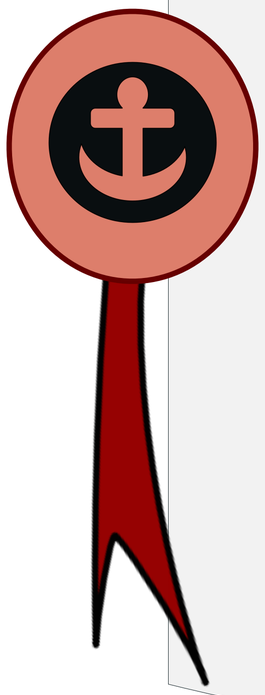
# Learn More

- Read the spec:
  - github.com/theupdateframework/tuf/ (docs/tuf-spec.txt)
- Look at Notary:
  - github.com/docker/notary
- Read the Docker Content Trust docs:
  - docs.docker.com/engine/security/trust/content_trust/



# THANK YOU

# Root Metadata



Root:

Timestamp:

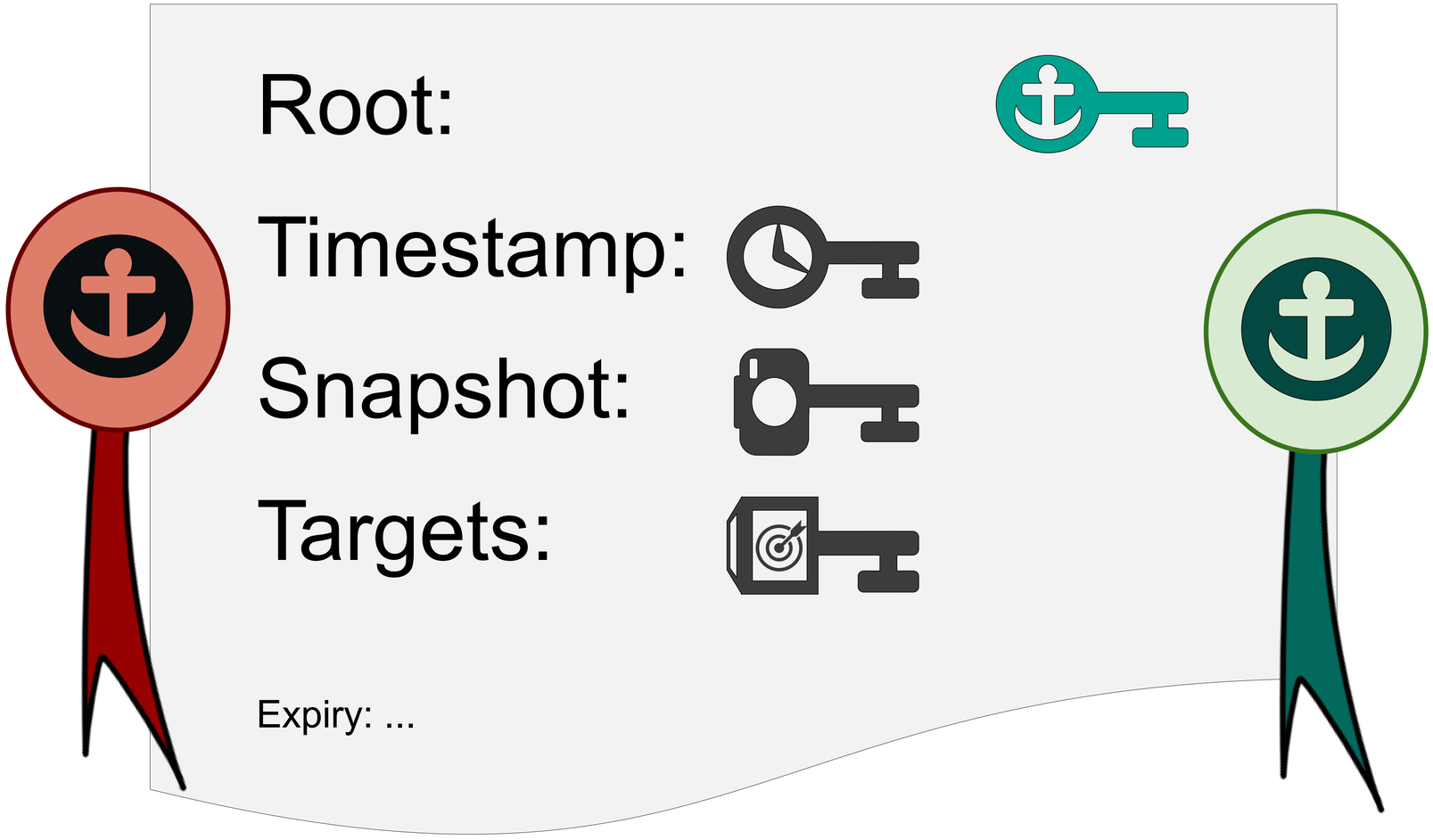
Snapshot:

Targets:

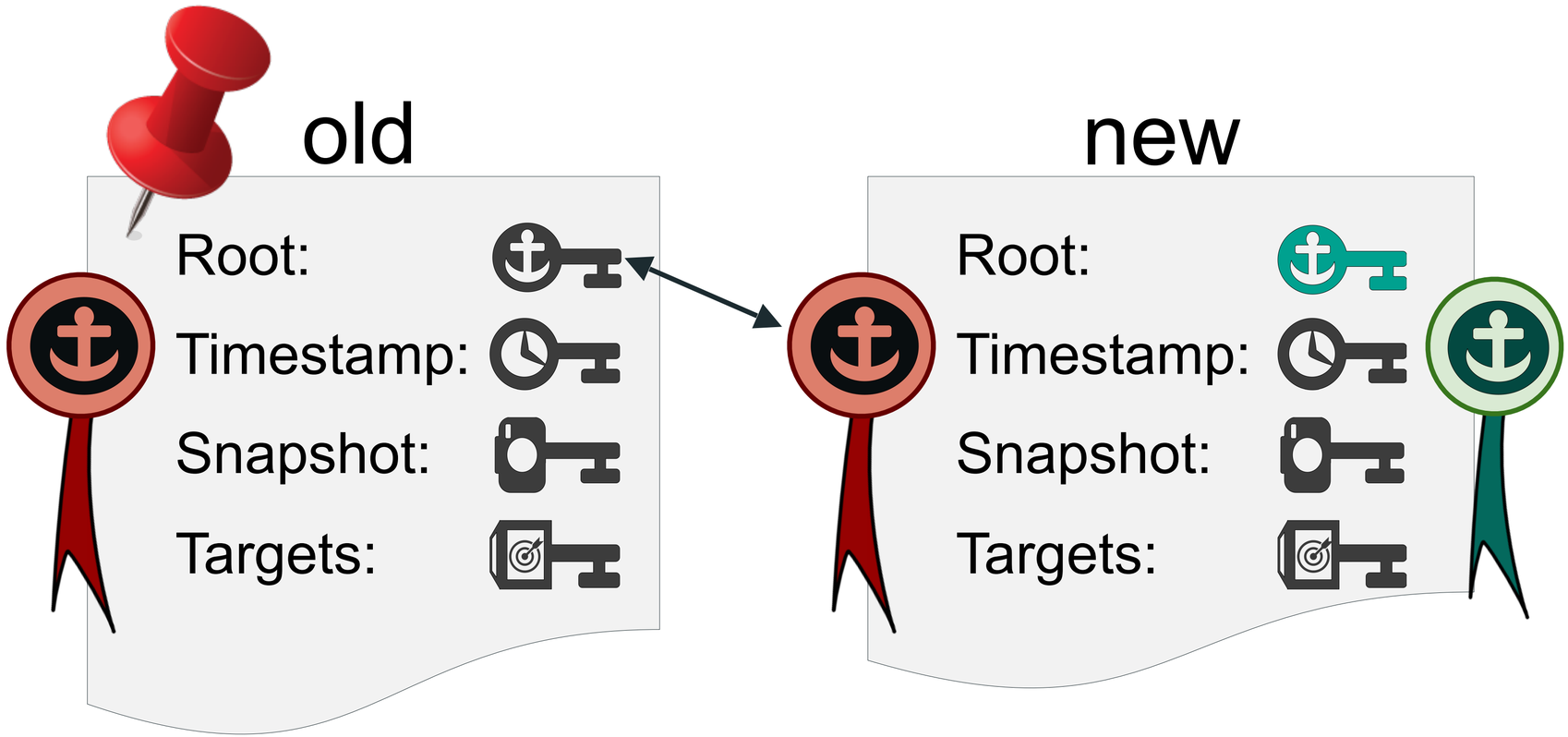
Expiry: ...

**Appendix: root key rotations**

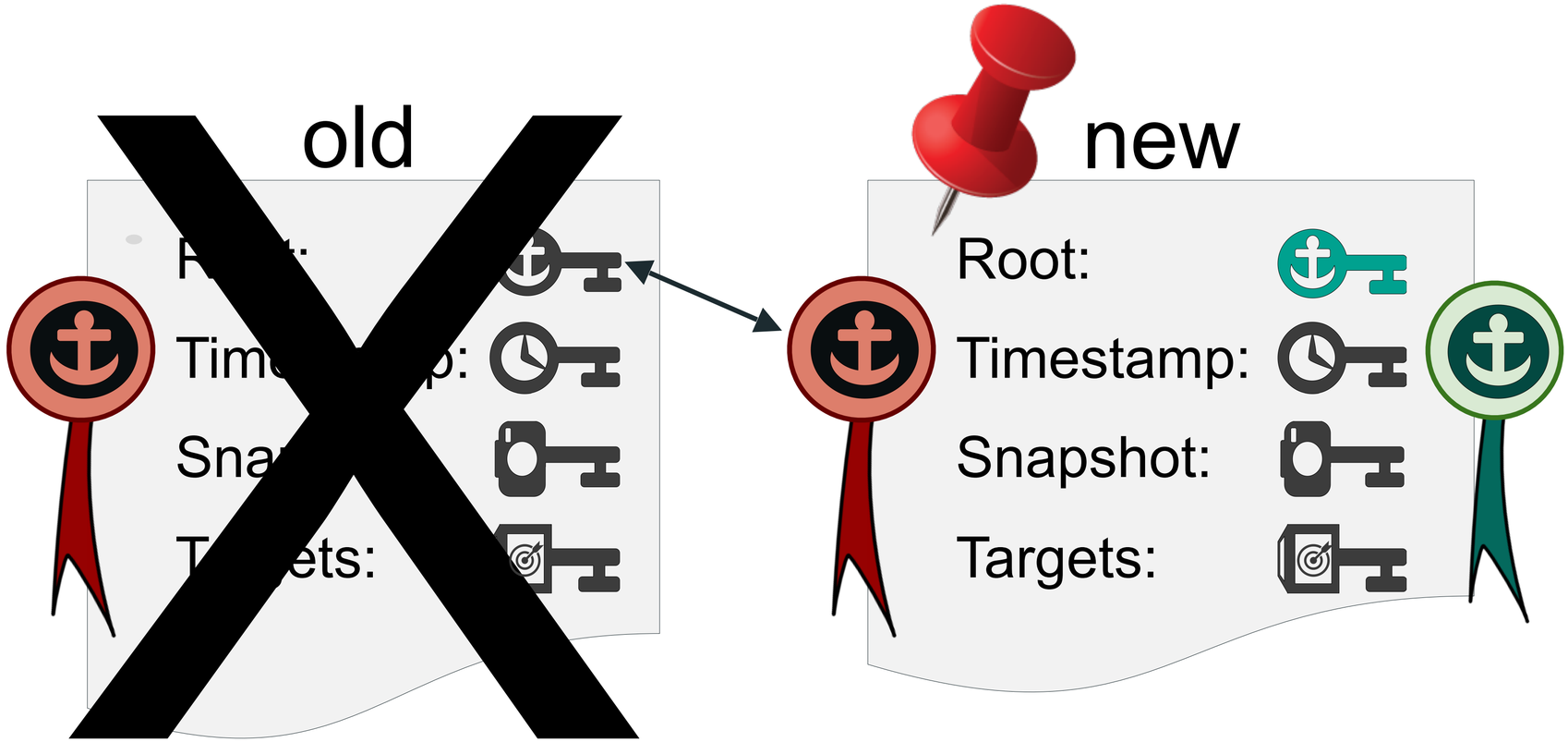
# Root Metadata



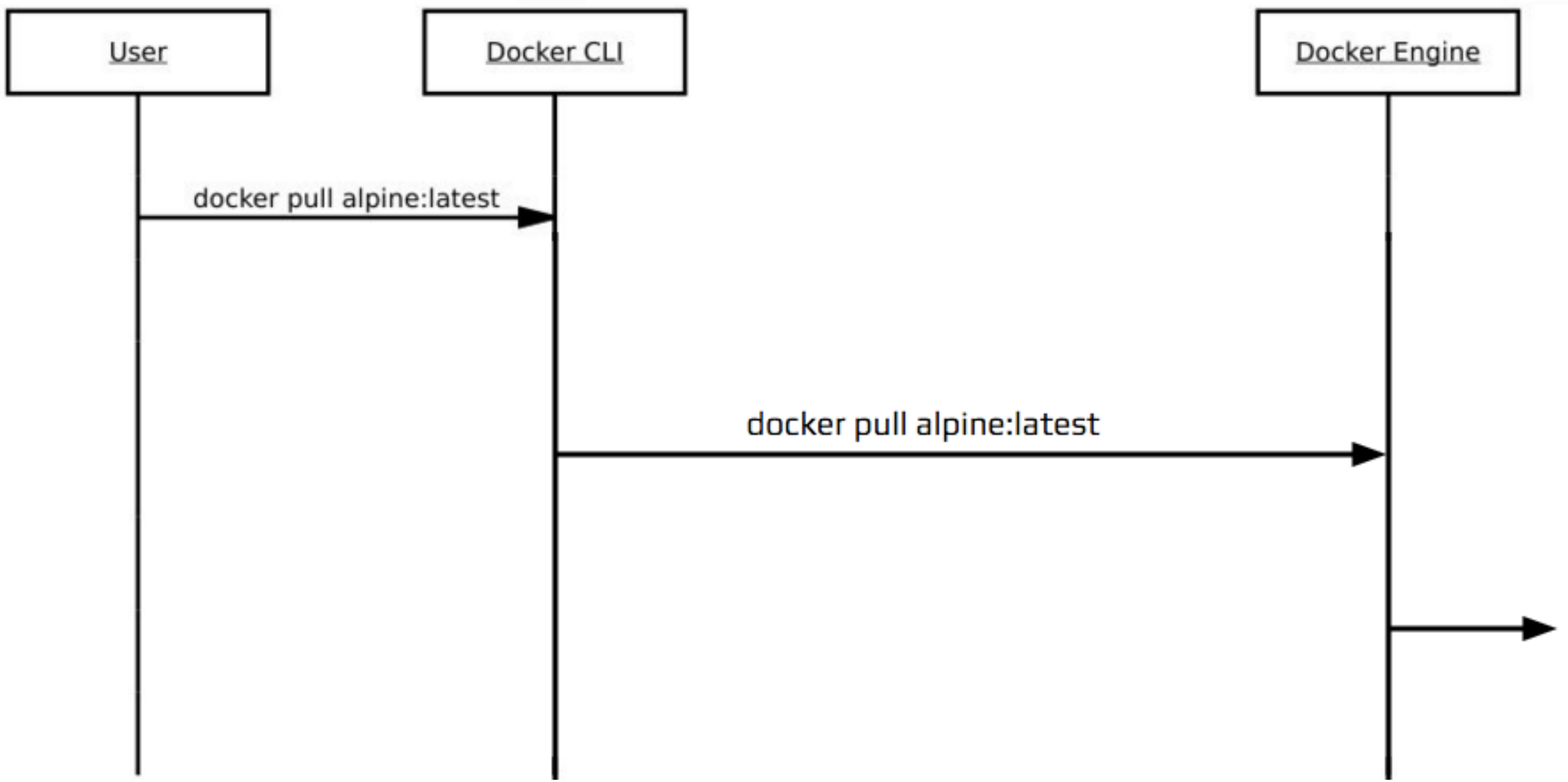

**Appendix: root key rotations**



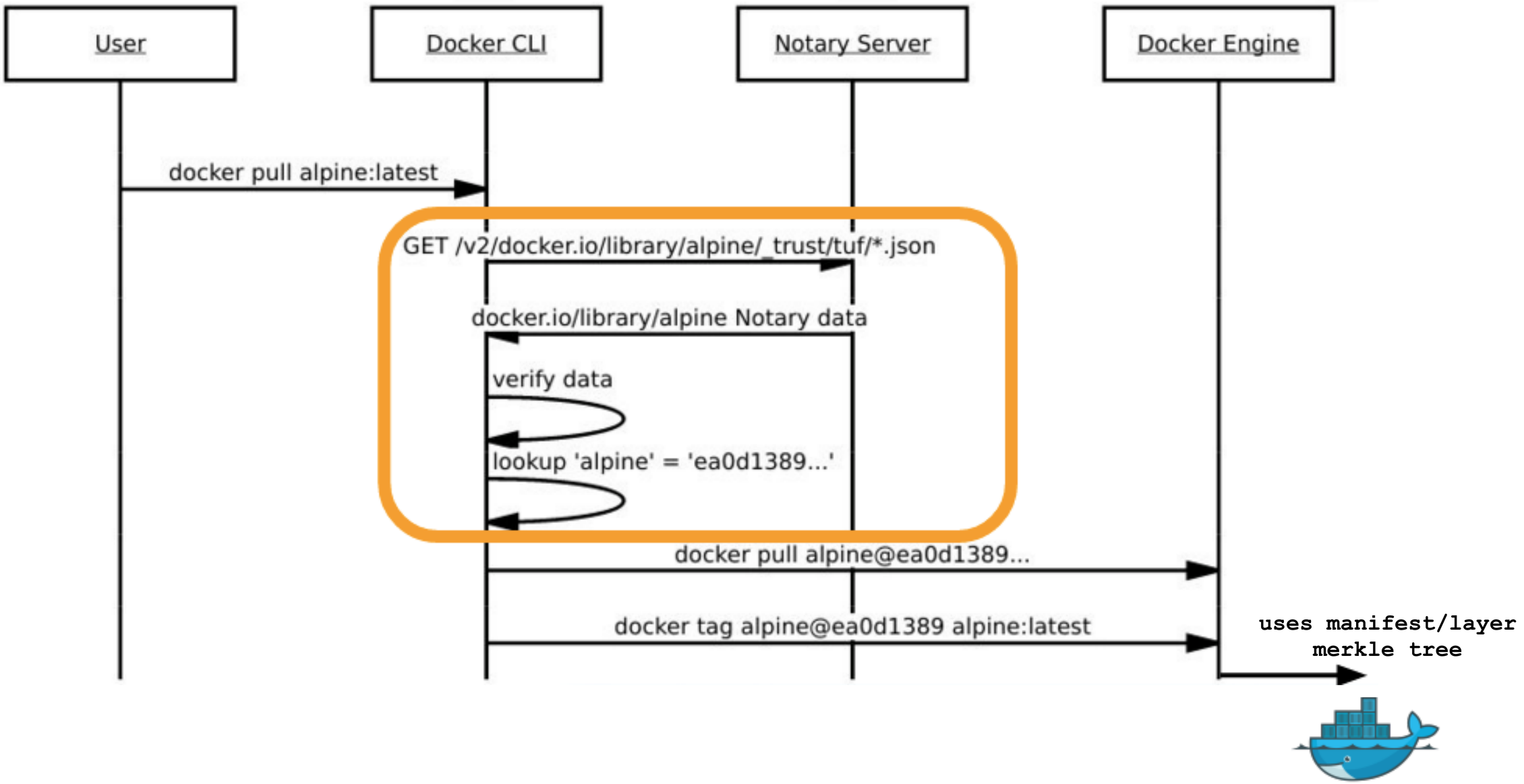

# Appendix: root key rotations




# Appendix: root key rotations




# Appendix: DCT pull flow




# Appendix: DCT pull flow