Macaroons

Cookies with Contextual Caveats for Decentralized Authorization in the Cloud

Photo: Daniel E. Lee, http://www.flickr.com/photos/dlee13/7609077916

Arnar Birgisson, Joe Politz,
Úlfar Erlingsson, Ankur Taly,
Michael Vrable, and Mark Lentczner

UNIVERSITY OF GOTHENBURG

CHALMERS

BROWN

Google

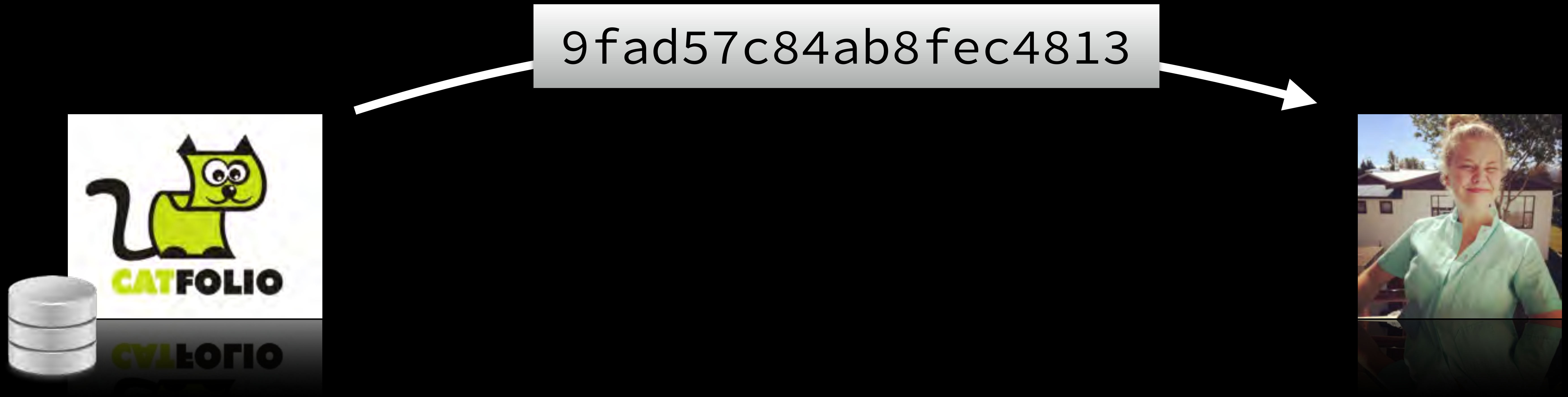
The Problem

- Cookies (and similar tokens) are ubiquitous in cloud authorization
- Good reasons: Simple, easily adopted
- But cookies are easy to steal, carry broad authority, and lack flexibility
- Macaroons = Better cookies, with arbitrary *caveats*, i.e. restrictions on access

Cookies with integrity

Standard technique:

- Protect cookies with a MAC

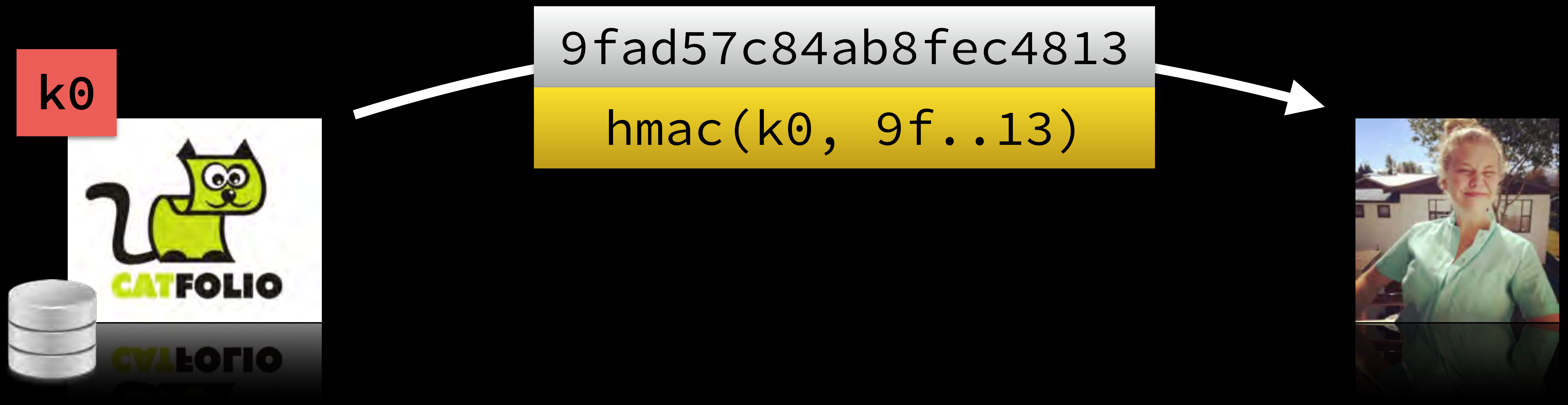


9fad57c84ab8fec4813

Cookies with integrity

Standard technique:

Protect cookies with a MAC



k0

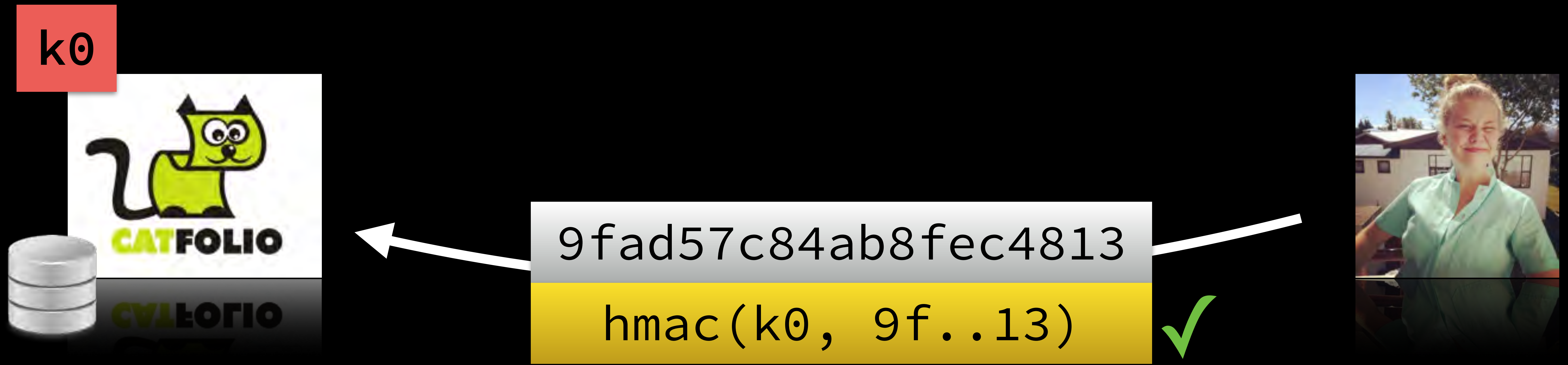
9fad57c84ab8fec4813

hmac(k0, 9f..13)

Cookies with integrity

Standard technique:

- Protect cookies with a MAC



k0

9fad57c84ab8fec4813

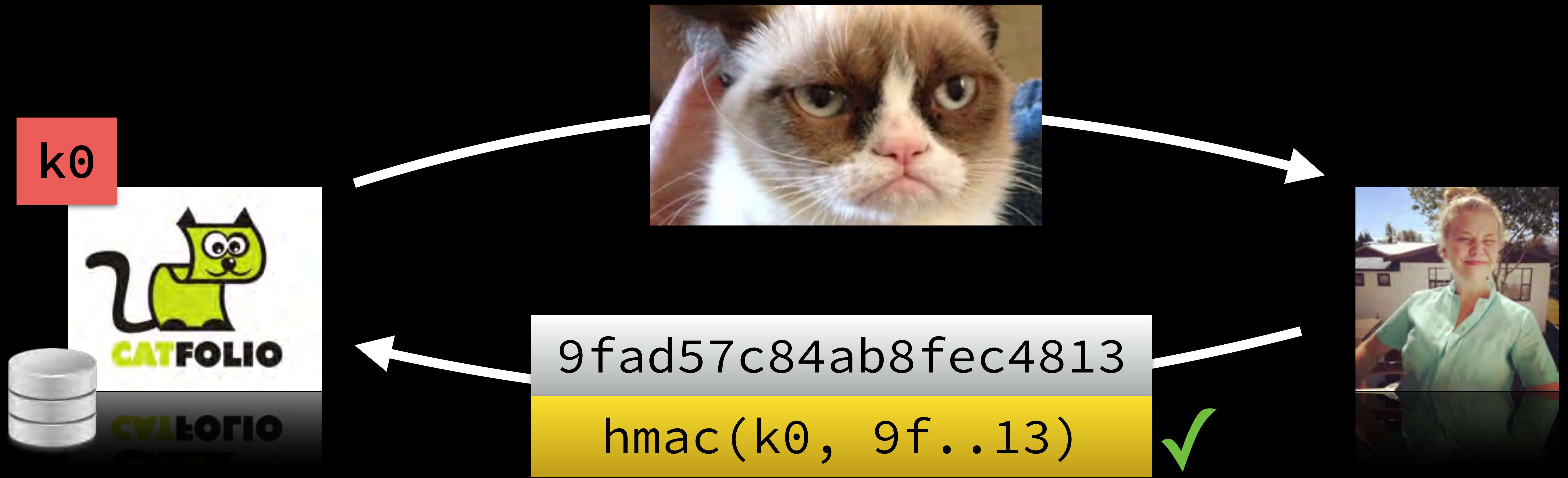
hmac(k0, 9f..13)

✓

Cookies with integrity

Standard technique:

- Protect cookies with a MAC



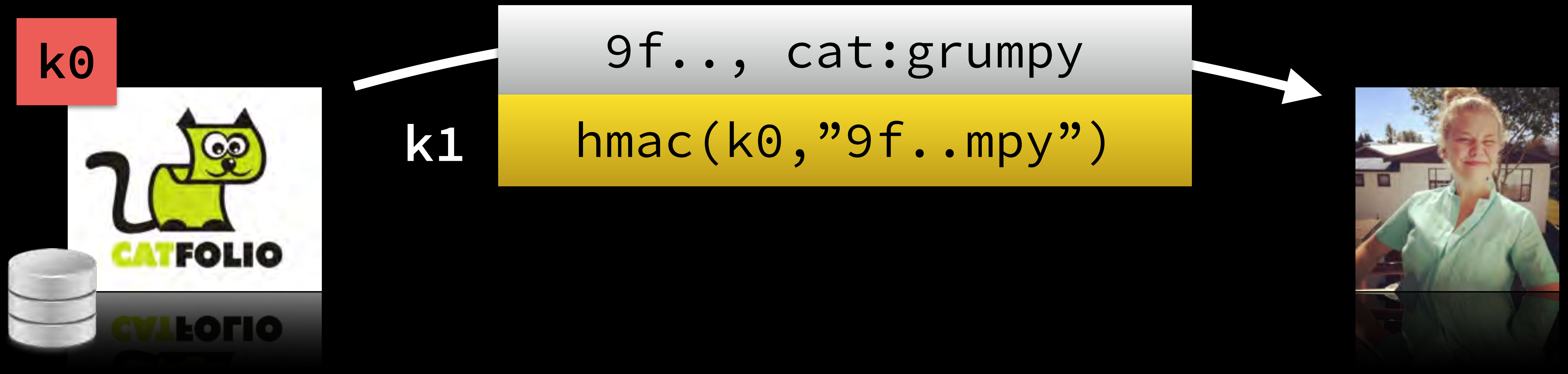
k0

9fad57c84ab8fec4813

hmac(k0, 9f..13) ✓

Hash-chaining

- Key enabling idea, dating back to Amoeba
- Macaroon: Key identifier + caveats + signature



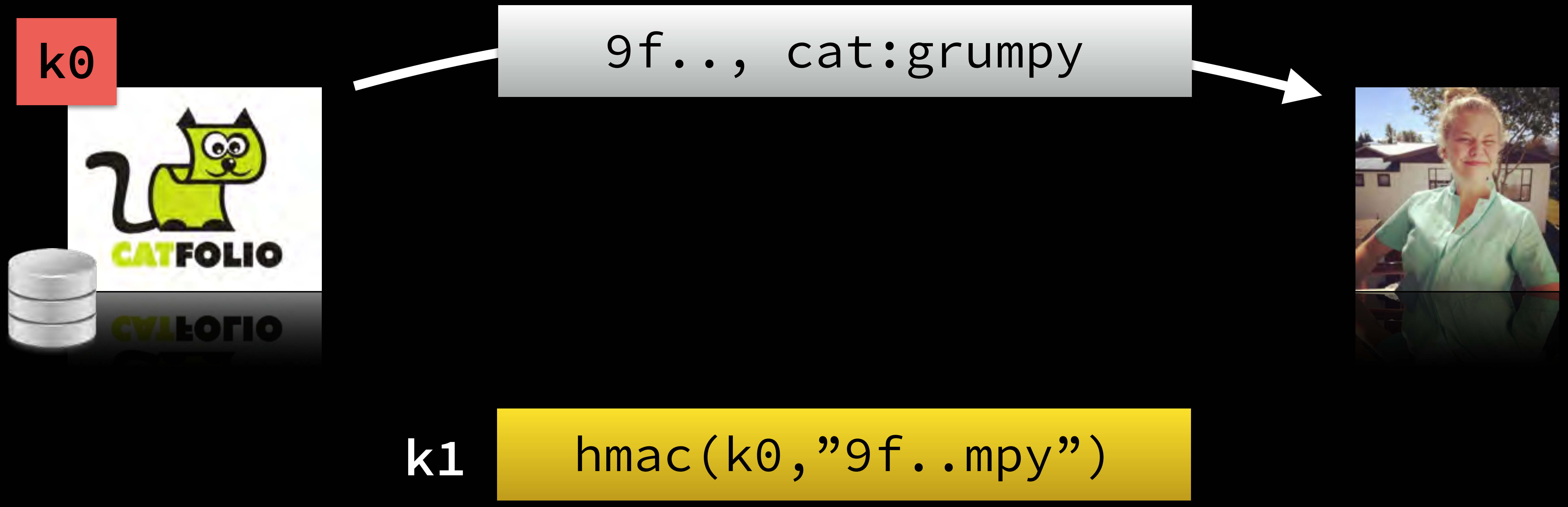
k0

k1

```
9f.., cat:grumpy
hmac(k0,”9f..mpy”)
```

Hash-chaining

- Key enabling idea, dating back to Amoeba
- Macaroon: Key identifier + caveats + signature



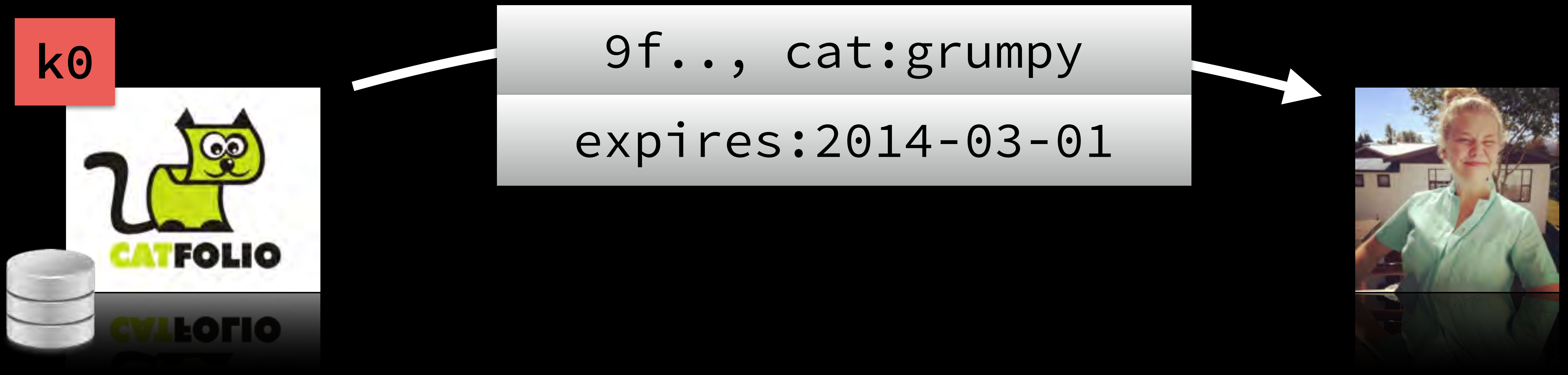
k0

`9f.., cat:grumpy`

k1 `hmac(k0,”9f..mpy”)`

Hash-chaining

- Key enabling idea, dating back to Amoeba
- Macaroon: Key identifier + caveats + signature



k0

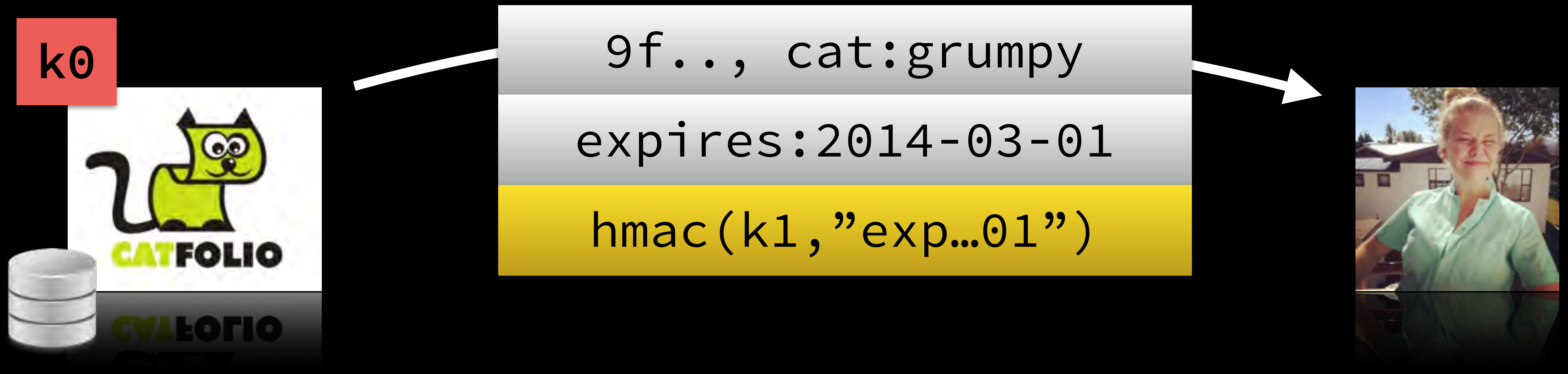
9f.., cat:grumpy

expires:2014-03-01

k1 hmac(k0,”9f..mpy”)

Hash-chaining

- Key enabling idea, dating back to Amoeba
- Macaroon: Key identifier + caveats + signature



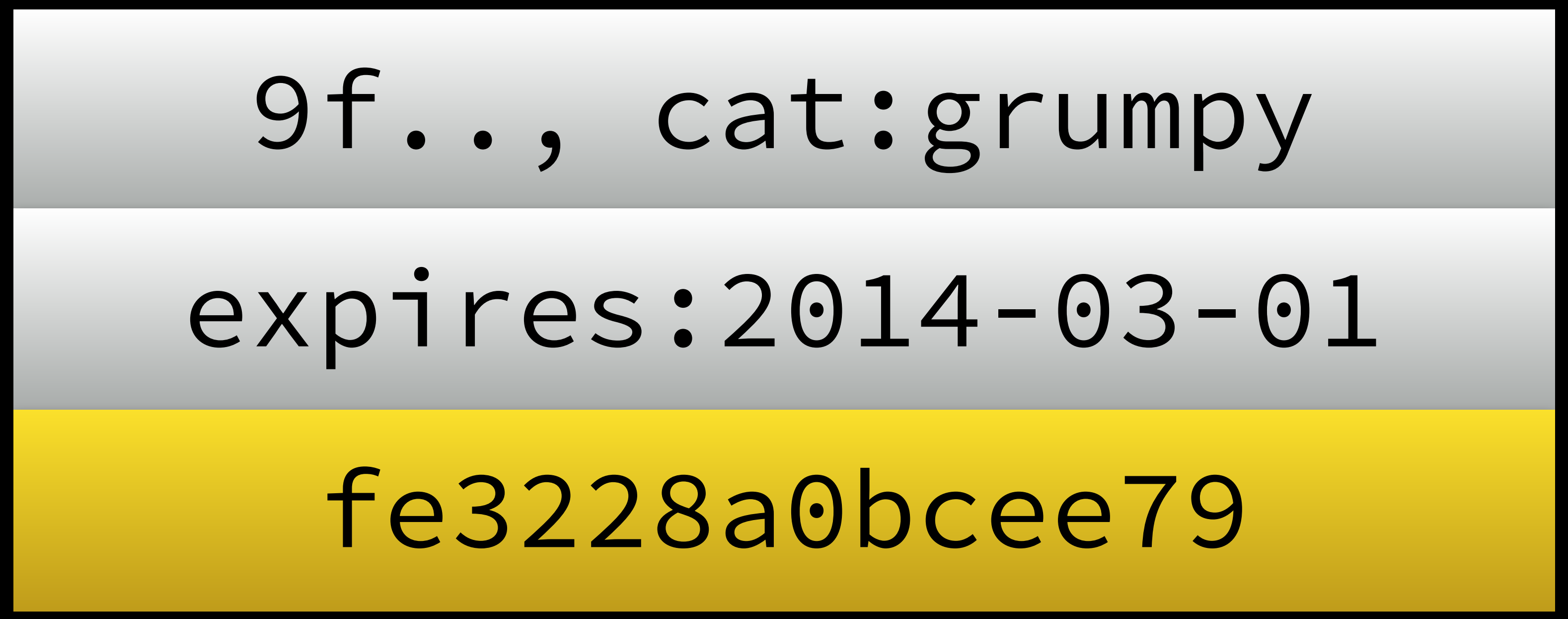
9f.., cat:grumpy

expires:2014-03-01

hmac(k1,”exp…01”)

Verification

- Verify integrity by recomputing signature

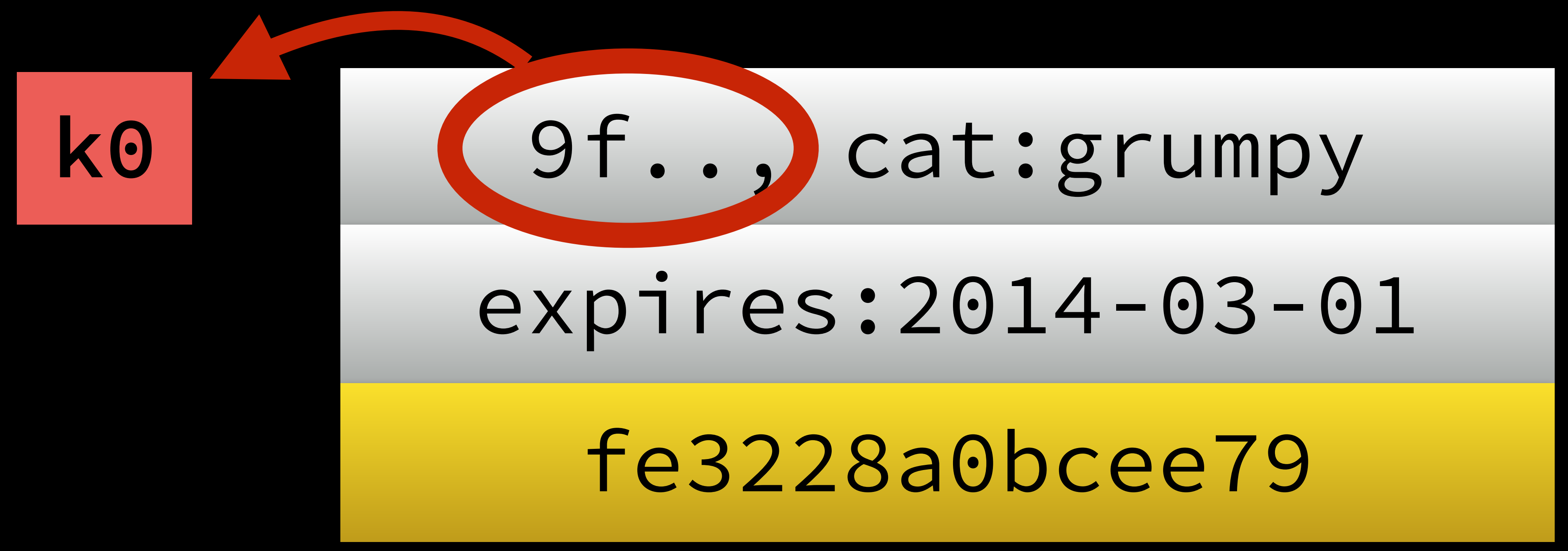


```
9f.., cat:grumpy
expires:2014-03-01
fe3228a0bcee79
```

CATFOLIO

Verification

- Verify integrity by recomputing signature



k0

9f.., cat:grumpy

expires:2014-03-01

fe3228a0bcee79

Verification

- Verify integrity by recomputing signature

```
hmac( k0 ,    9f.., cat:grumpy     ) = k1
           expires:2014-03-01
              fe3228a0bcee79
```

Verification

- Verify integrity by recomputing signature

```
hmac( k0 ,   9f.., cat:grumpy          ) = k1
hmac( k1 ,   expires:2014-03-01        ) = k2
      k2 ==      fe3228a0bcee79       ✓
```

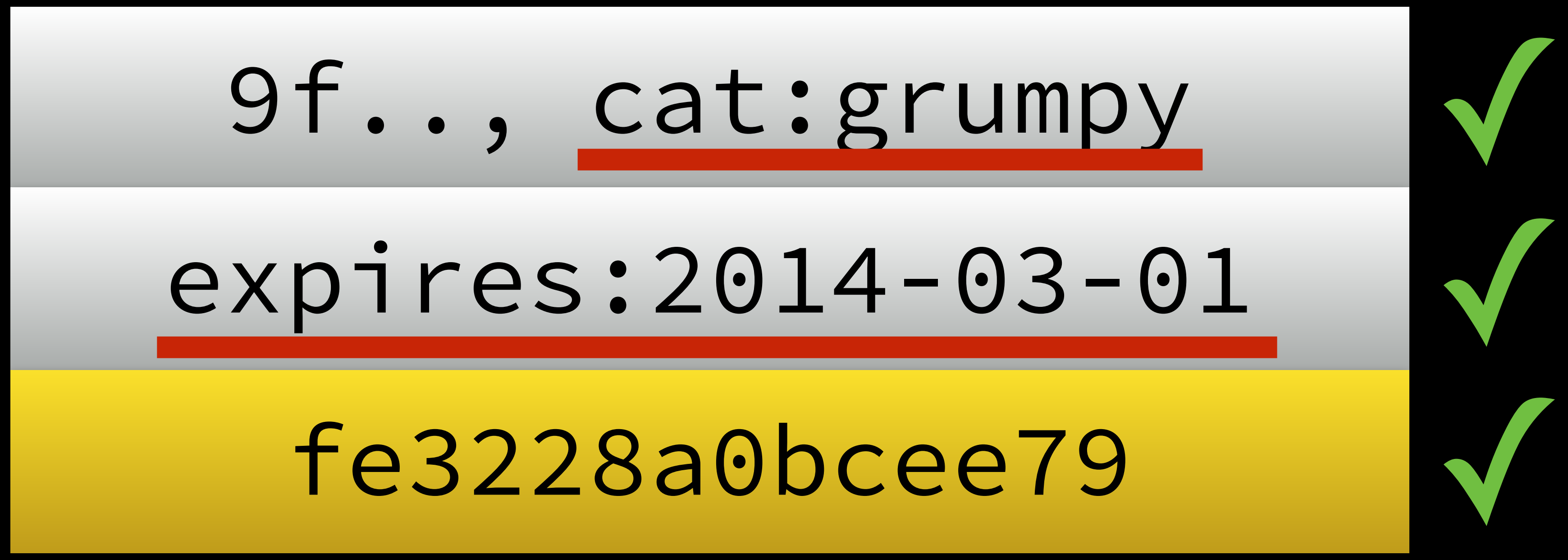
Verification

- Verify integrity by recomputing signature

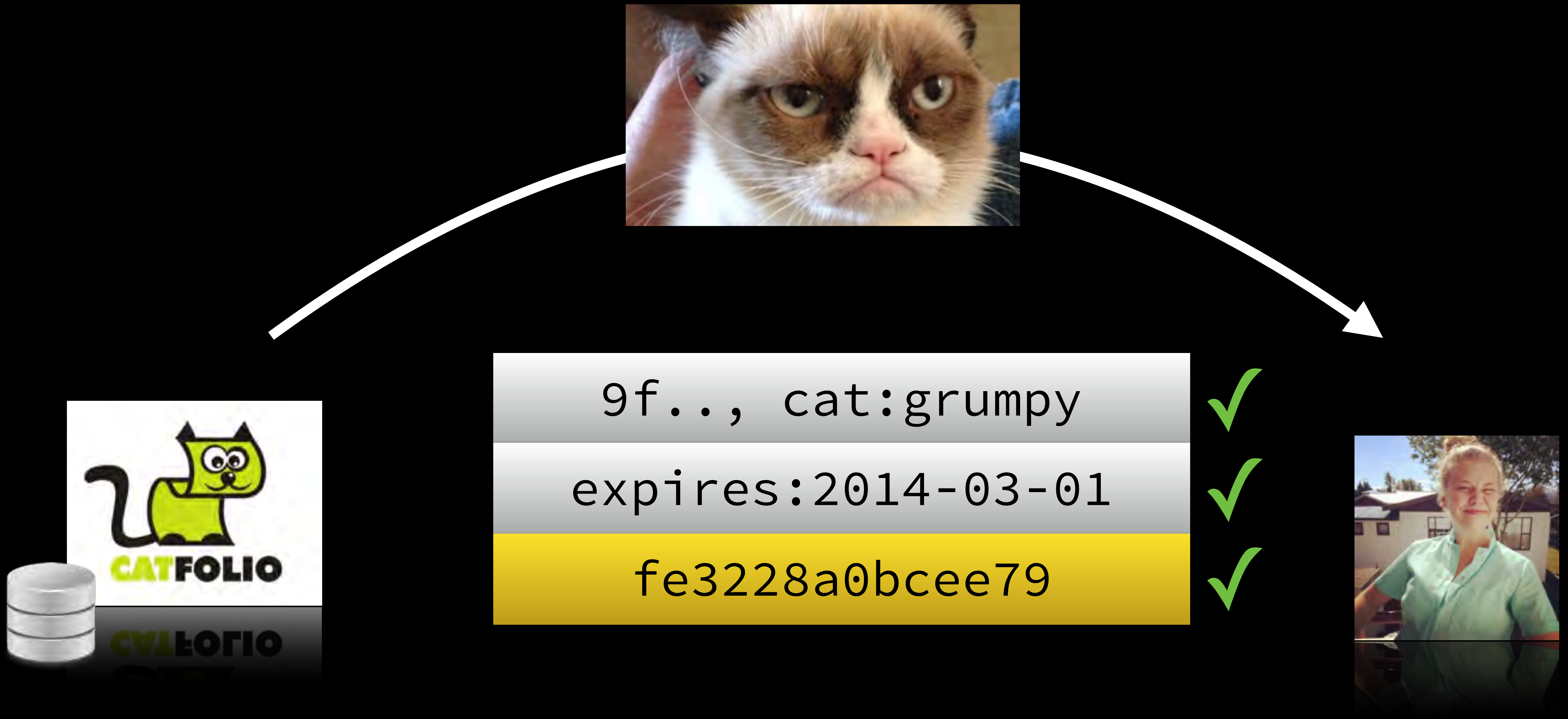
```
9f.., cat:grumpy
expires:2014-03-01
fe3228a0bcee79 ✓
```

Verification

- Verify integrity by recomputing signature
- Check that all caveats are satisfied



```
9f.., cat:grumpy ✓
expires:2014-03-01 ✓
fe3228a0bcee79 ✓
```



9f.., cat:grumpy ✓

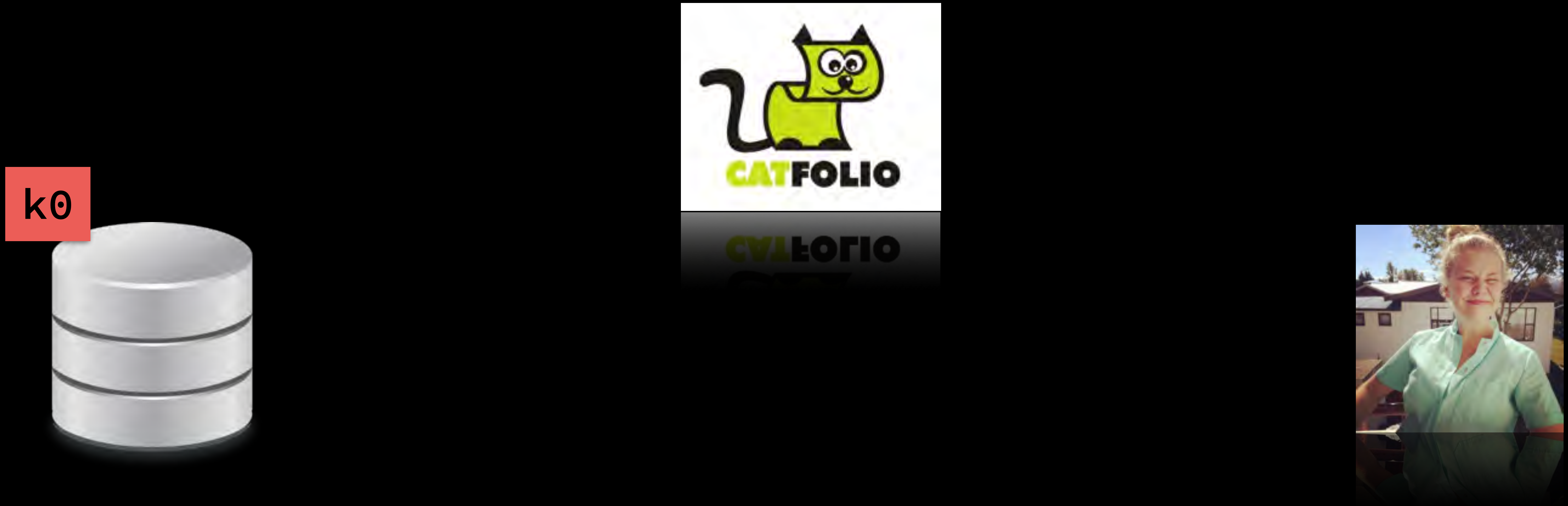
expires:2014-03-01 ✓

fe3228a0bcee79 ✓

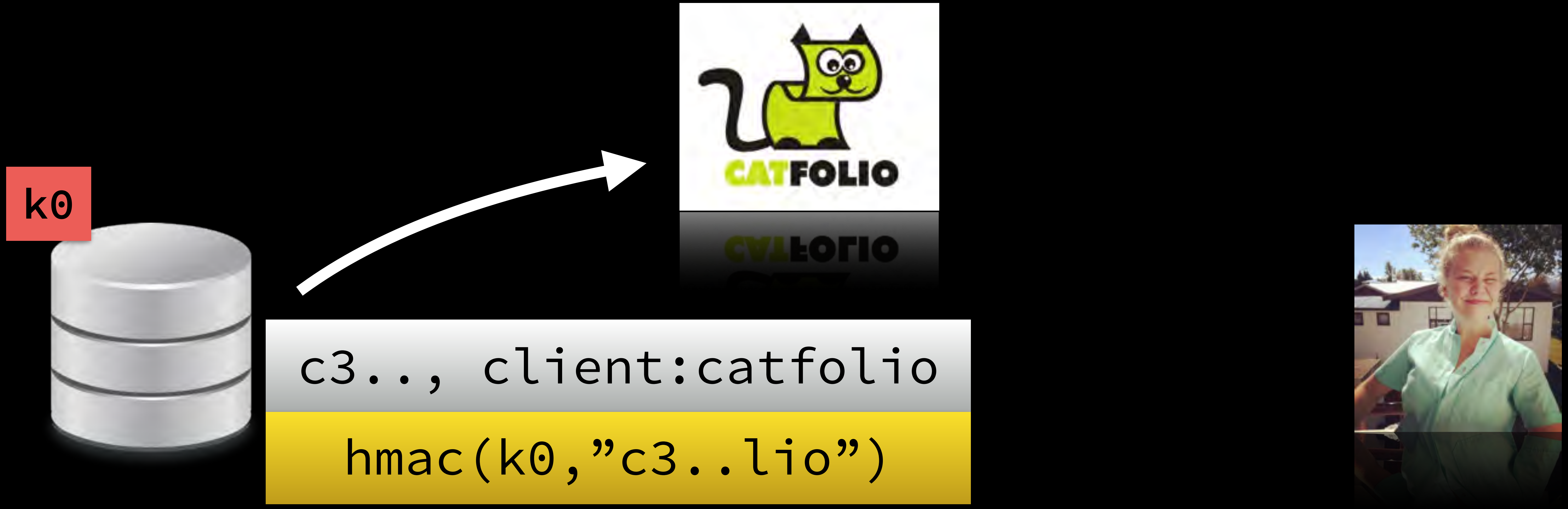
First-party caveats

- Can be added by any holder of a macaroon
- No constraints on the assertion language
- Naturally support attenuation and delegation

Delegation

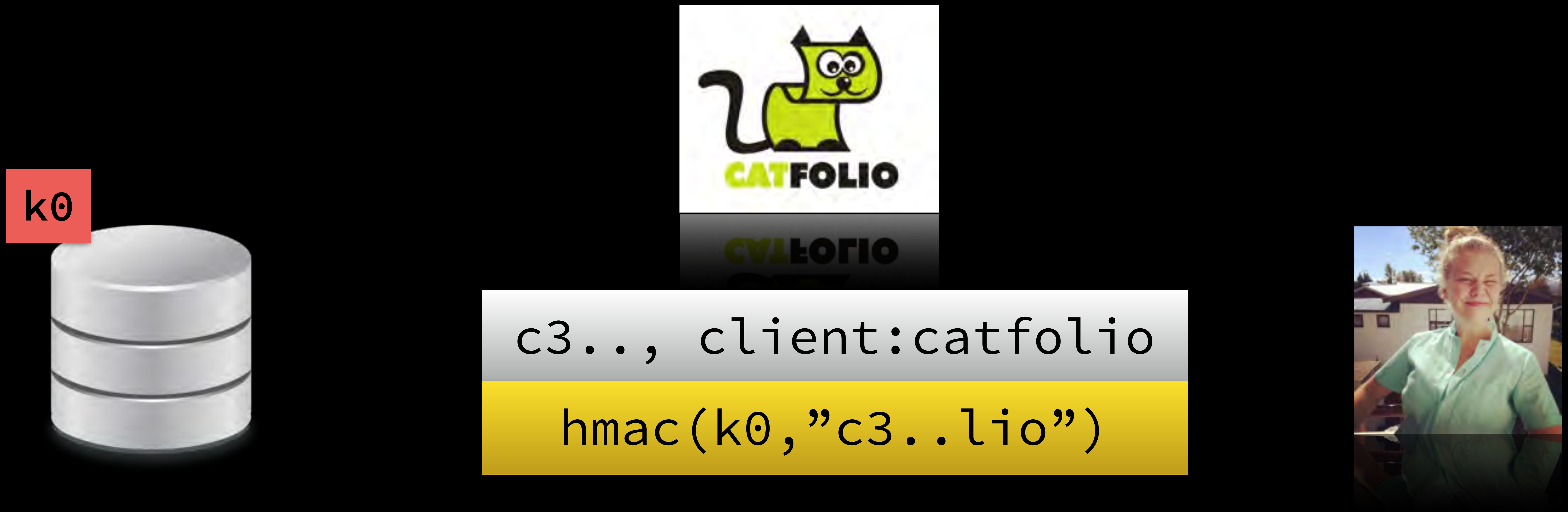


Delegation



```
c3.., client:catfolio
hmac(k0,”c3..lio”)
```

Delegation

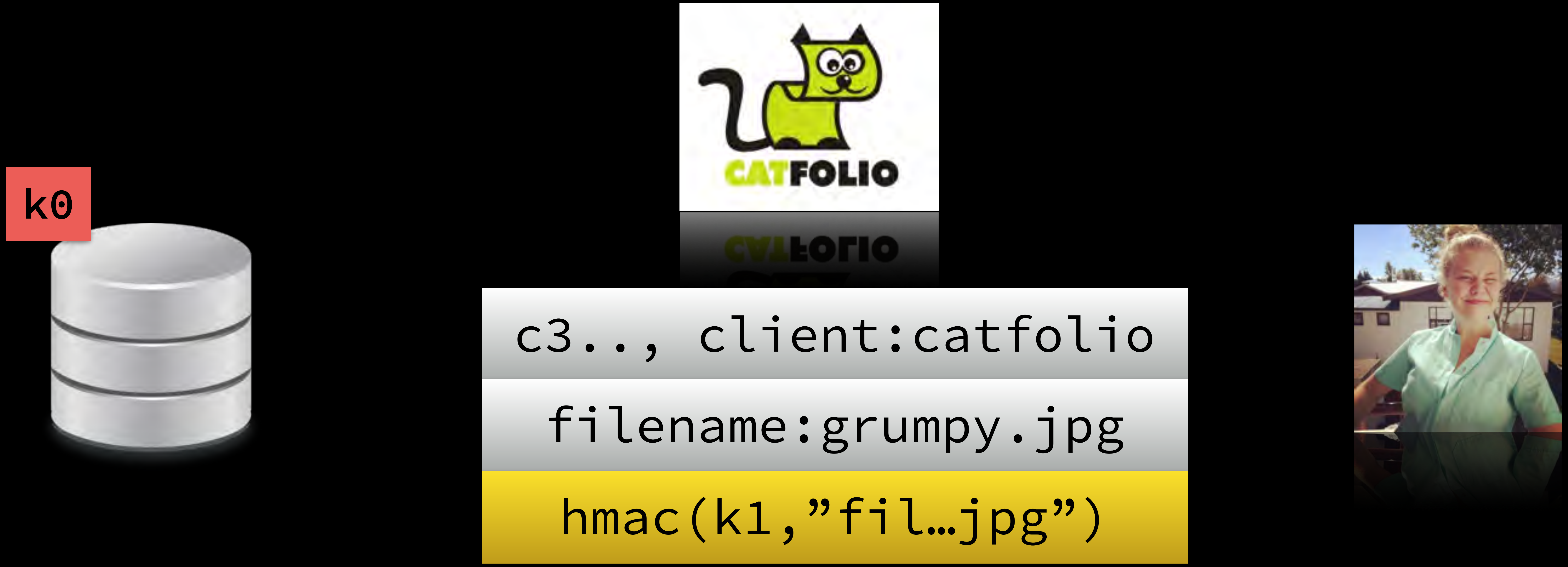


k0

c3.., client:catfolio

hmac(k0,”c3..lio”)

Delegation



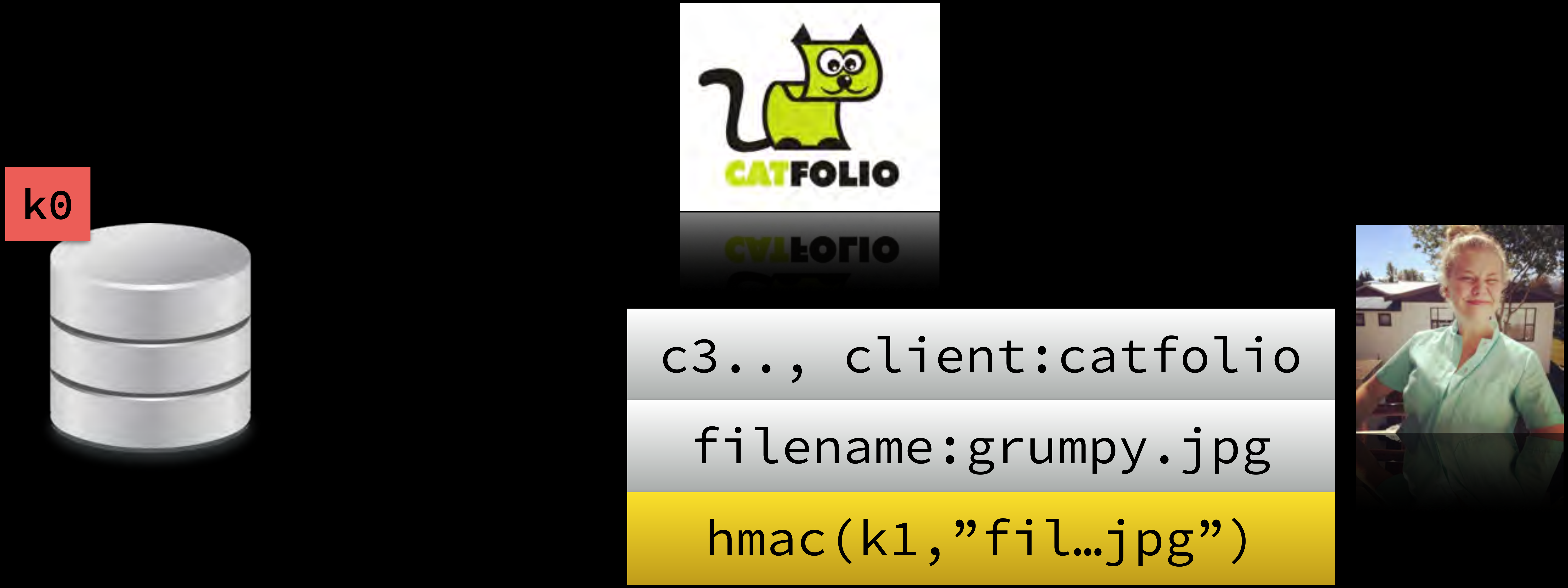
k0

c3.., client:catfolio

filename:grumpy.jpg

hmac(k1,”fil…jpg”)

Delegation

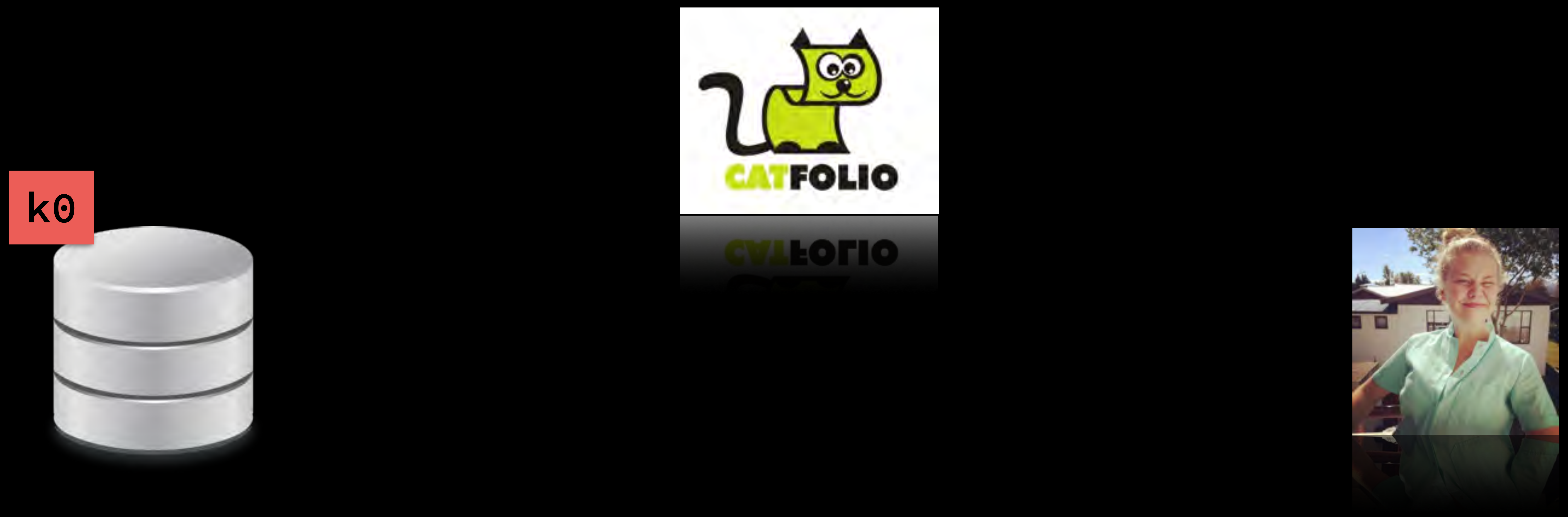


c3.., client:catfolio

filename:grumpy.jpg

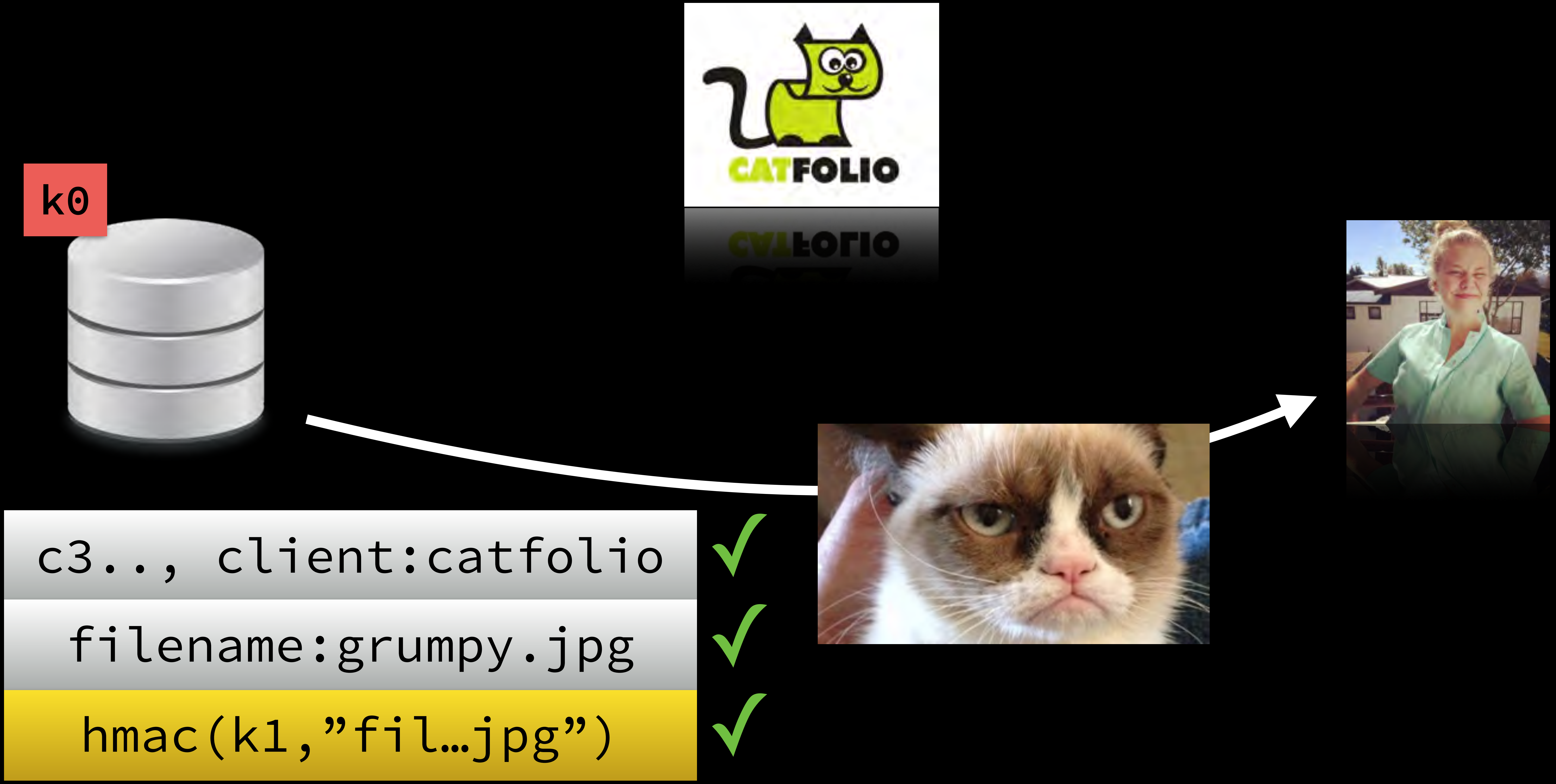
hmac(k1,”fil…jpg”)

Delegation



```
c3.., client:catfolio
filename:grumpy.jpg
hmac(k1,”fil…jpg”)
```

Delegation



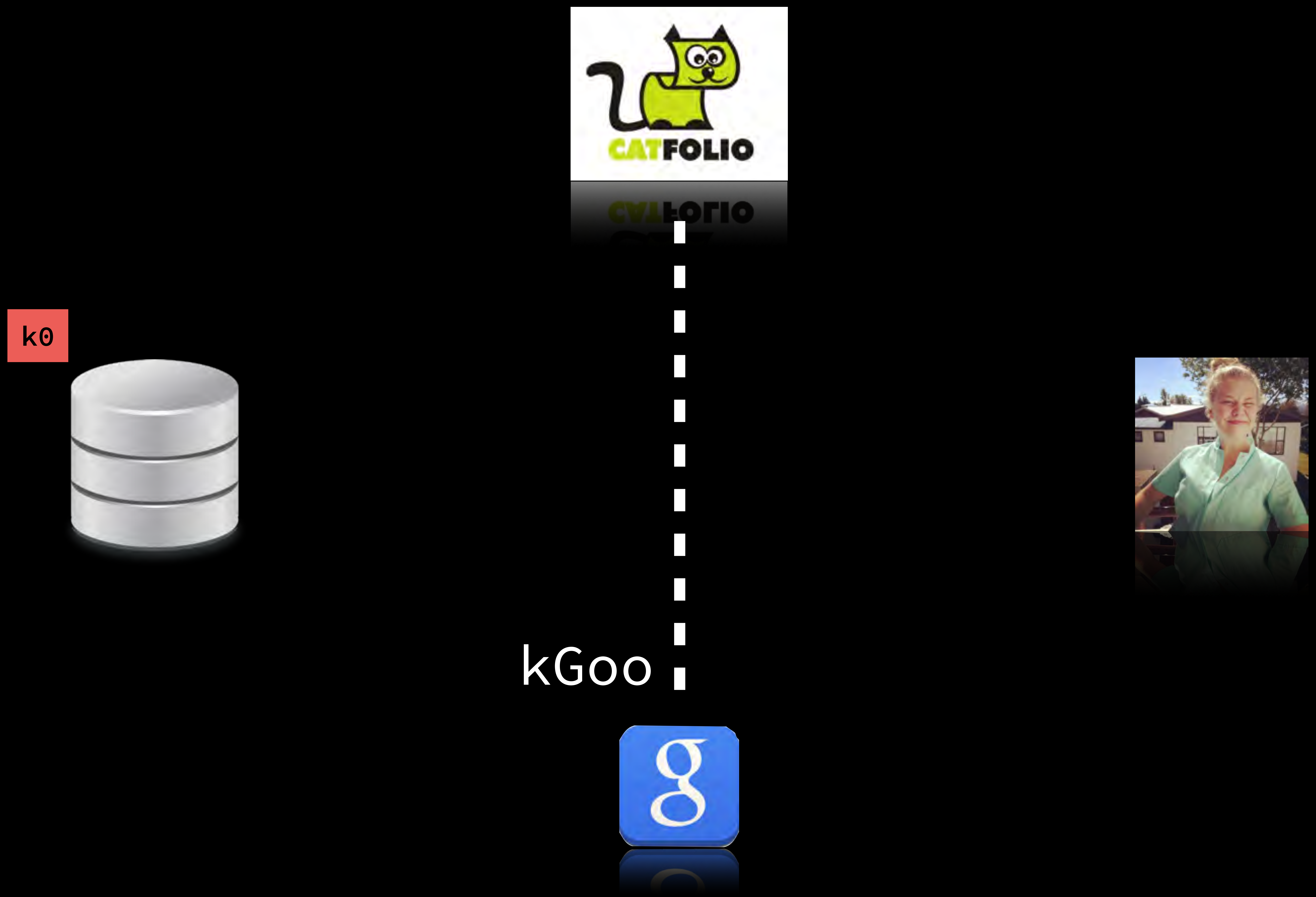
c3.., client:catfolio ✓

filename:grumpy.jpg ✓

hmac(k1,”fil…jpg”) ✓

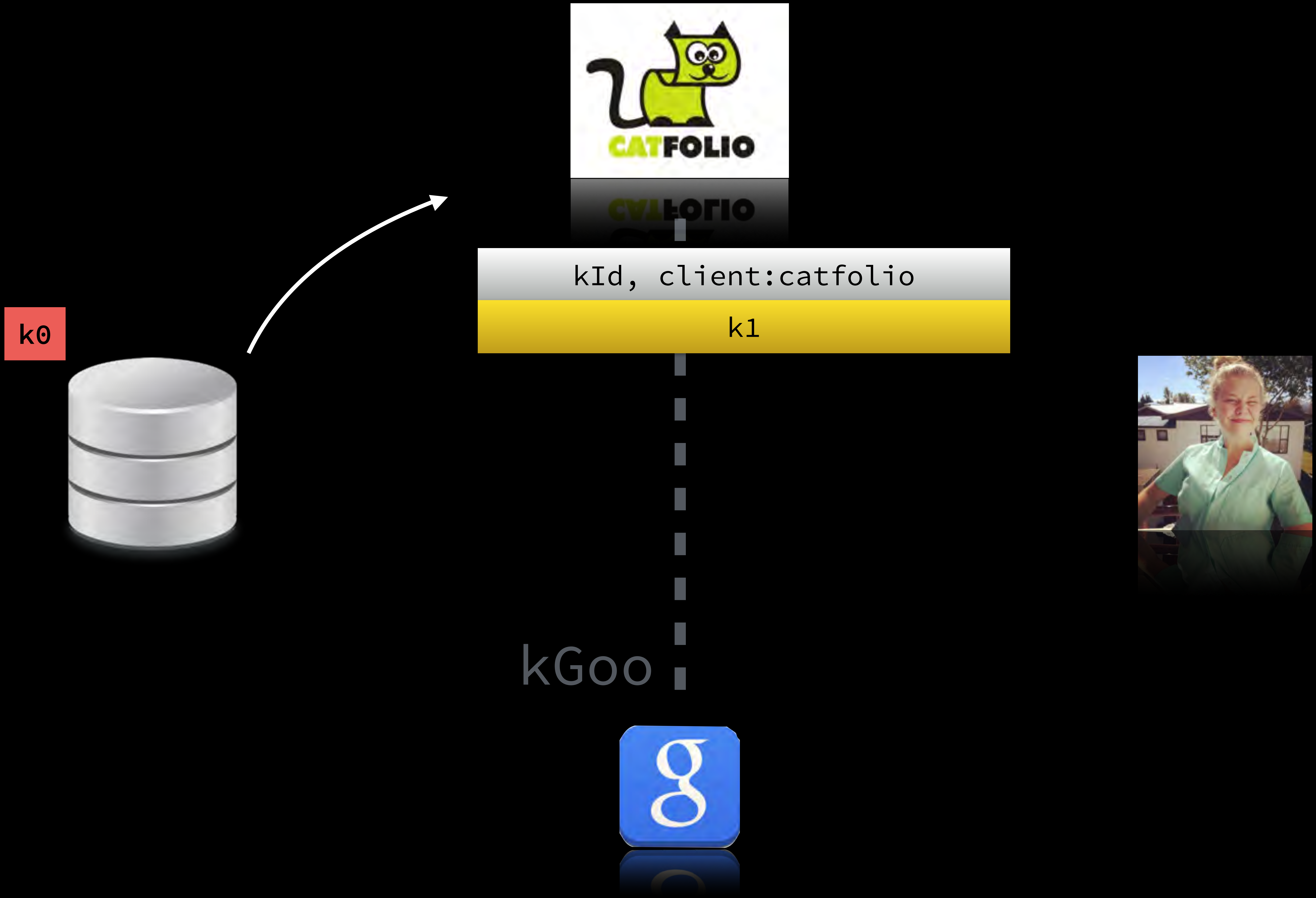
Third-party caveats

- So far, only the target server is trusted for authorization decisions.
- Third-party caveats require another service to check restrictions.



k0

kGoo

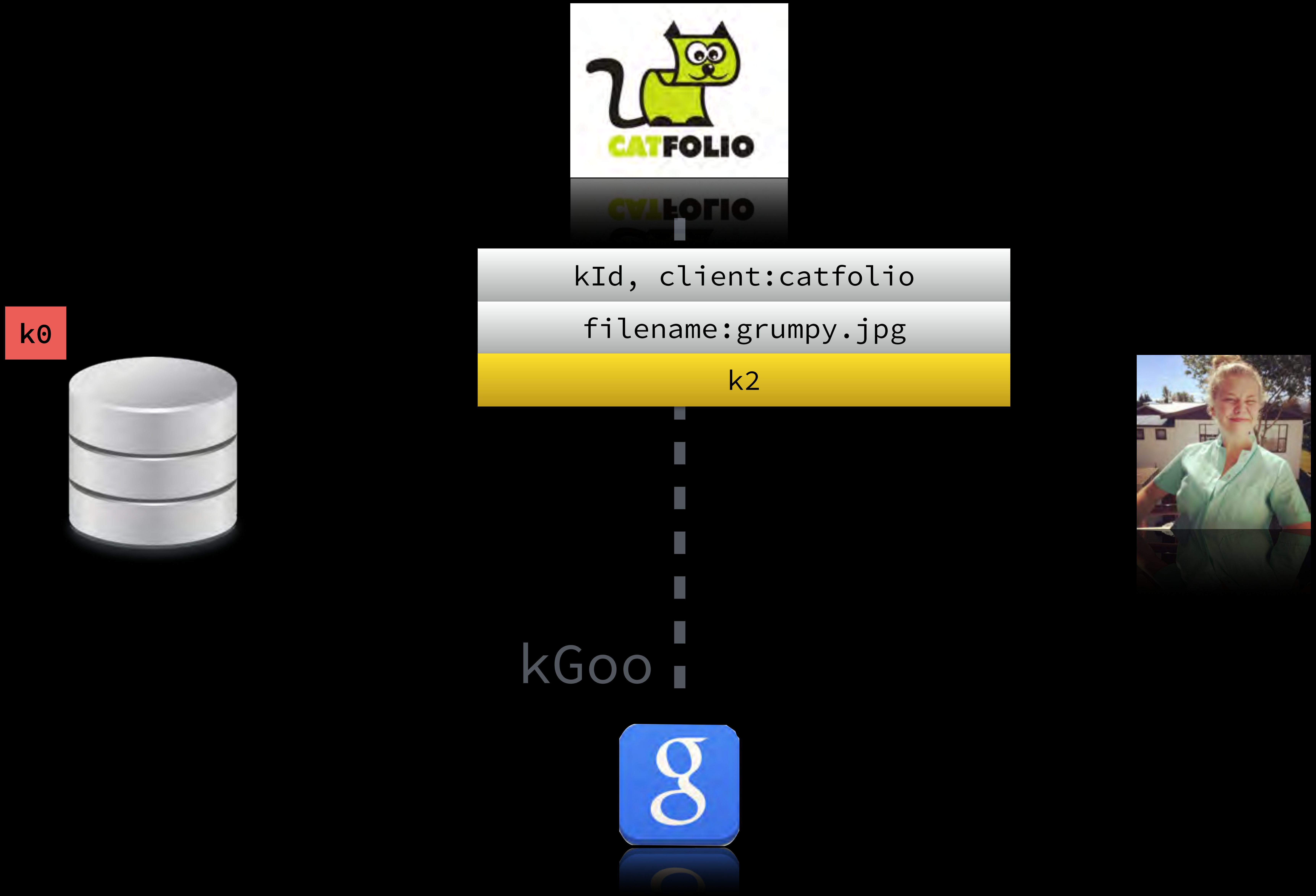


k0

kId, client:catfolio

k1

kGoo



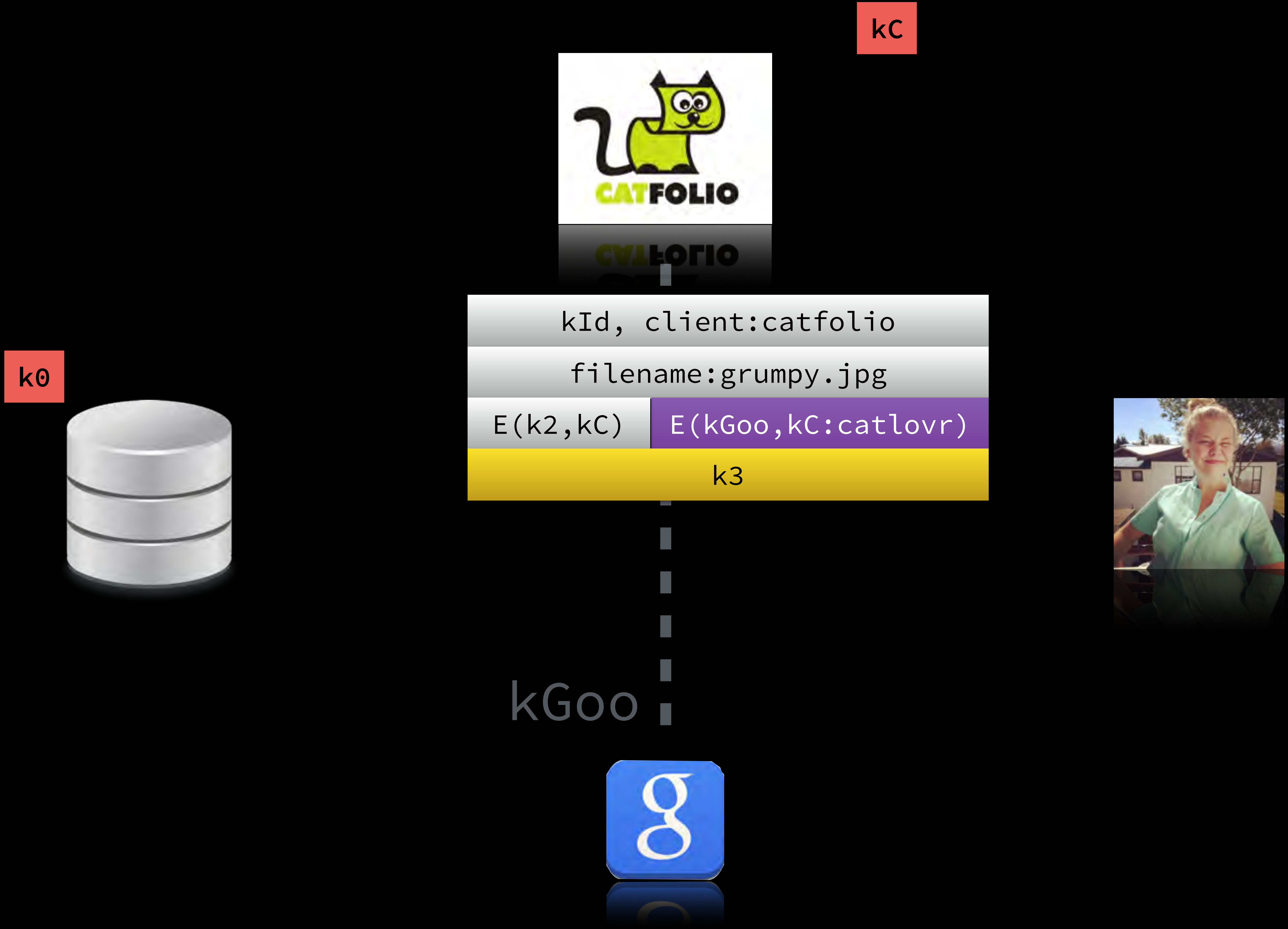
k0

kId, client:catfolio

filename:grumpy.jpg

k2

kGoo



kC

k0

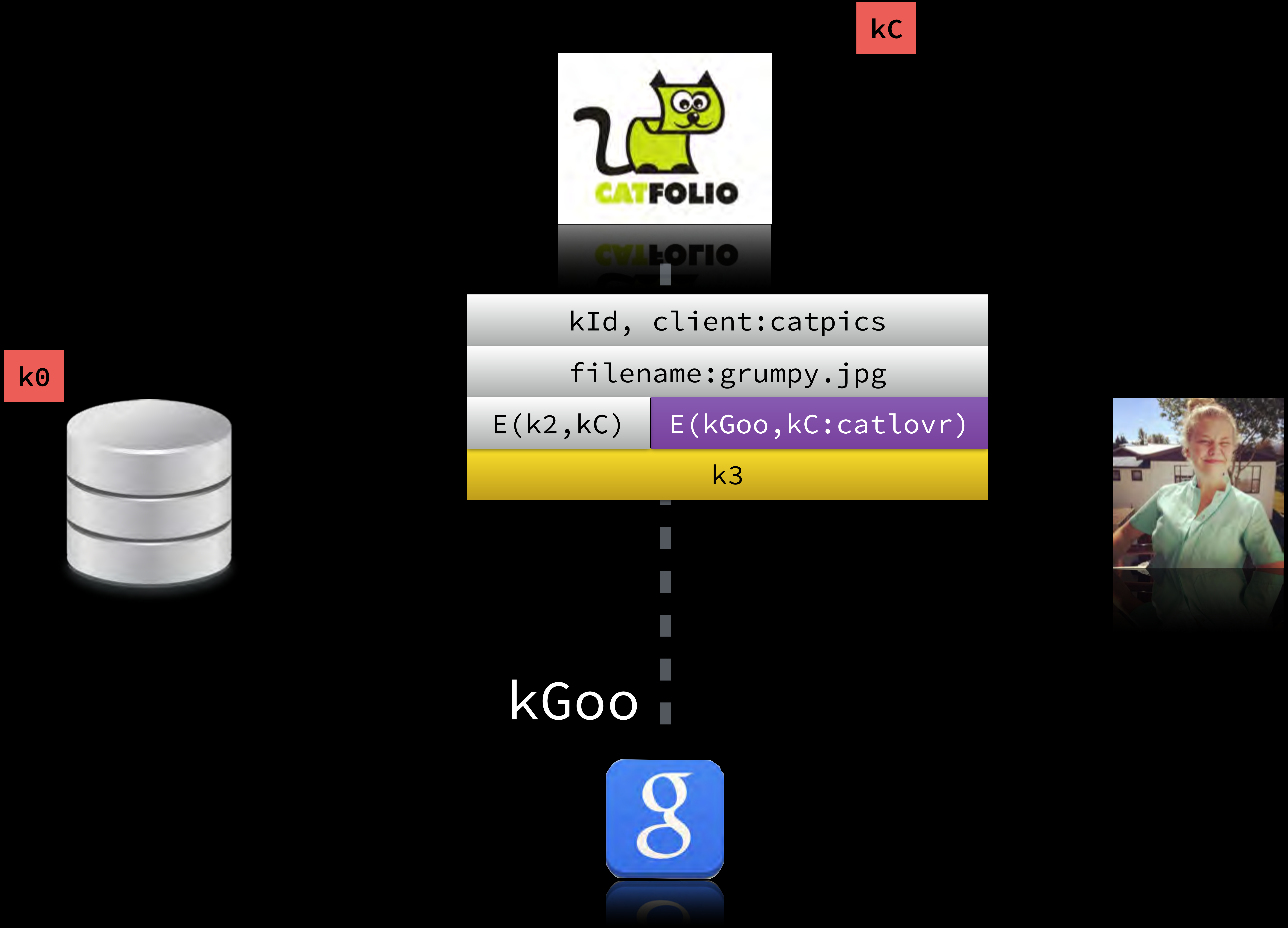
kId, client:catfolio

filename:grumpy.jpg

E(k2,kC) E(kGoo,kC:catlovr)

k3

kGoo



kC

CATFOLIO

kId, client:catpics

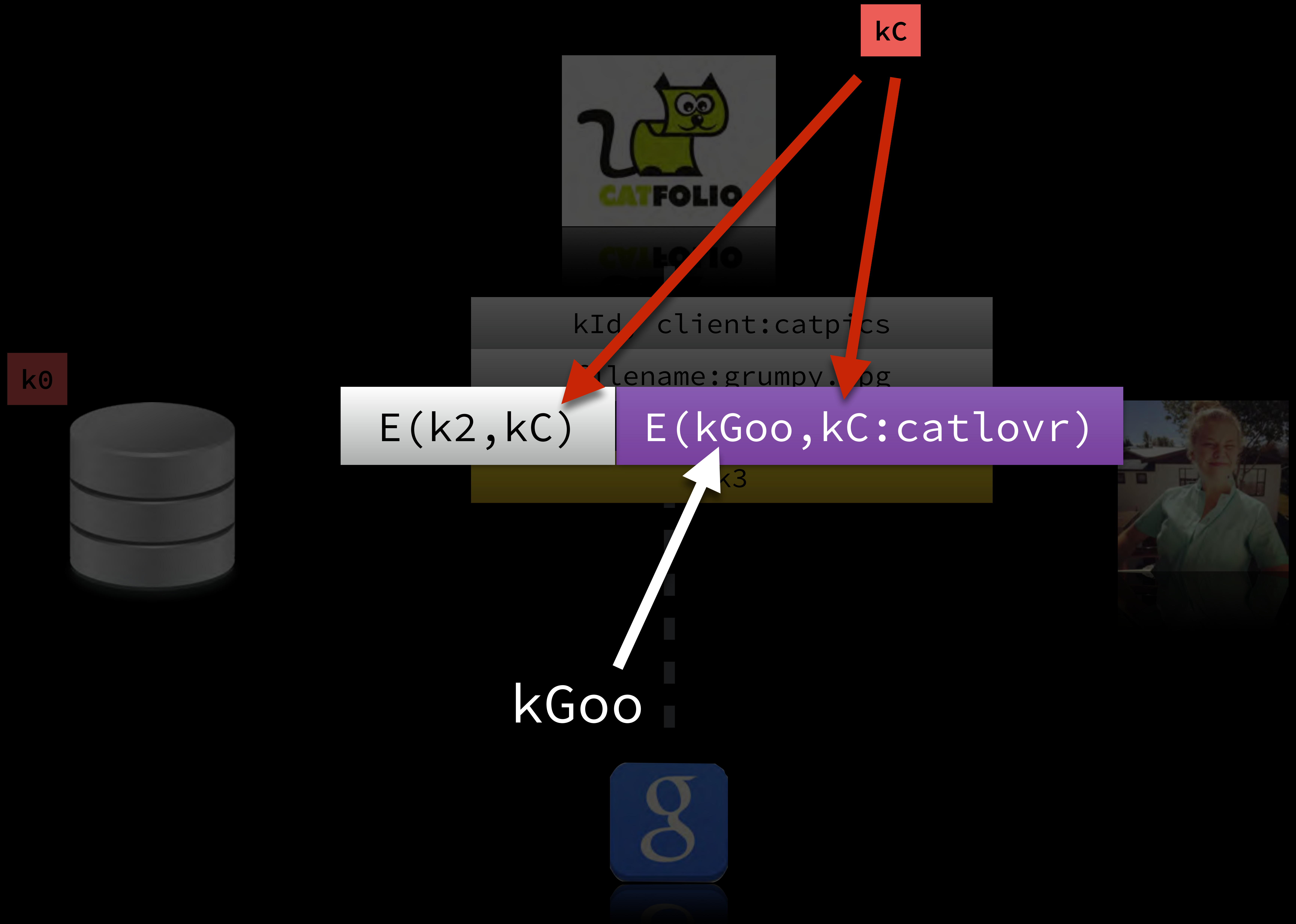
filename:grumpy.jpg

E(k2,kC) E(kGoo,kC:catlovr)

k3

k0

kGoo



kC

kId: client:catpics

filename:grumpy.jpg

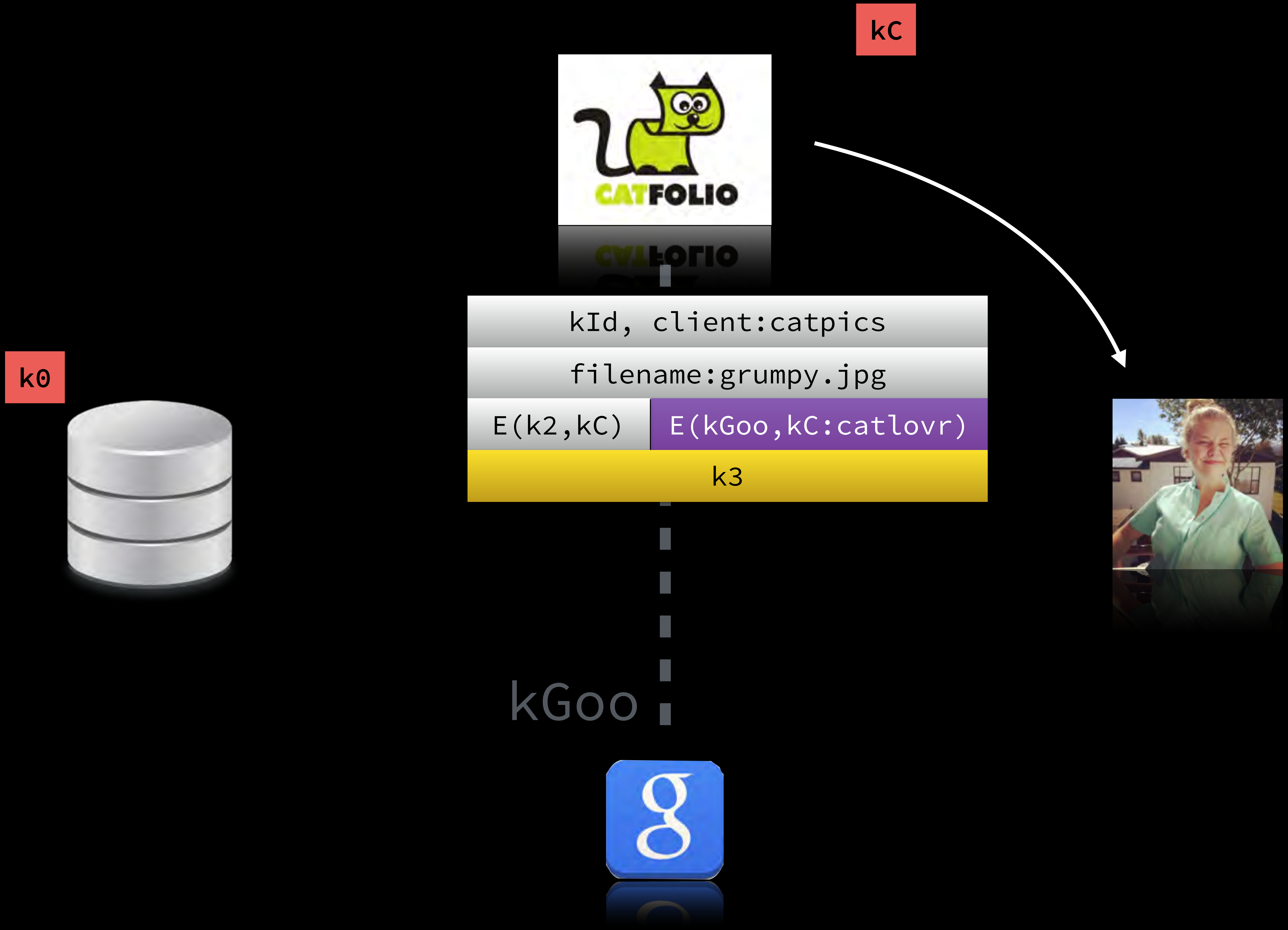
E(k2,kC)

E(kGoo,kC:catlovr)

k3

k0

kGoo



kC

k0

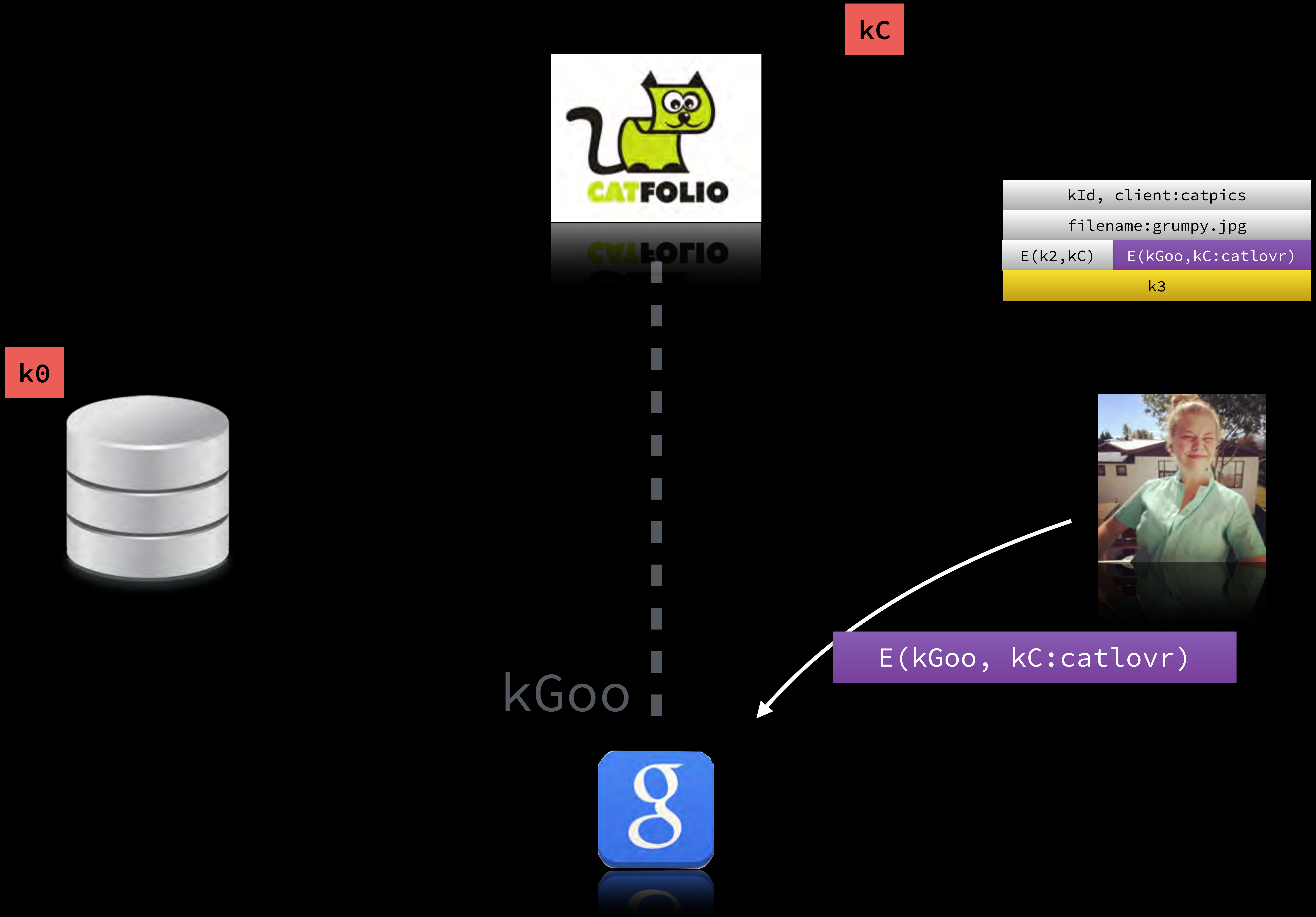
kId, client:catpics

filename:grumpy.jpg

E(k2,kC) E(kGoo,kC:catlovr)

k3

kGoo



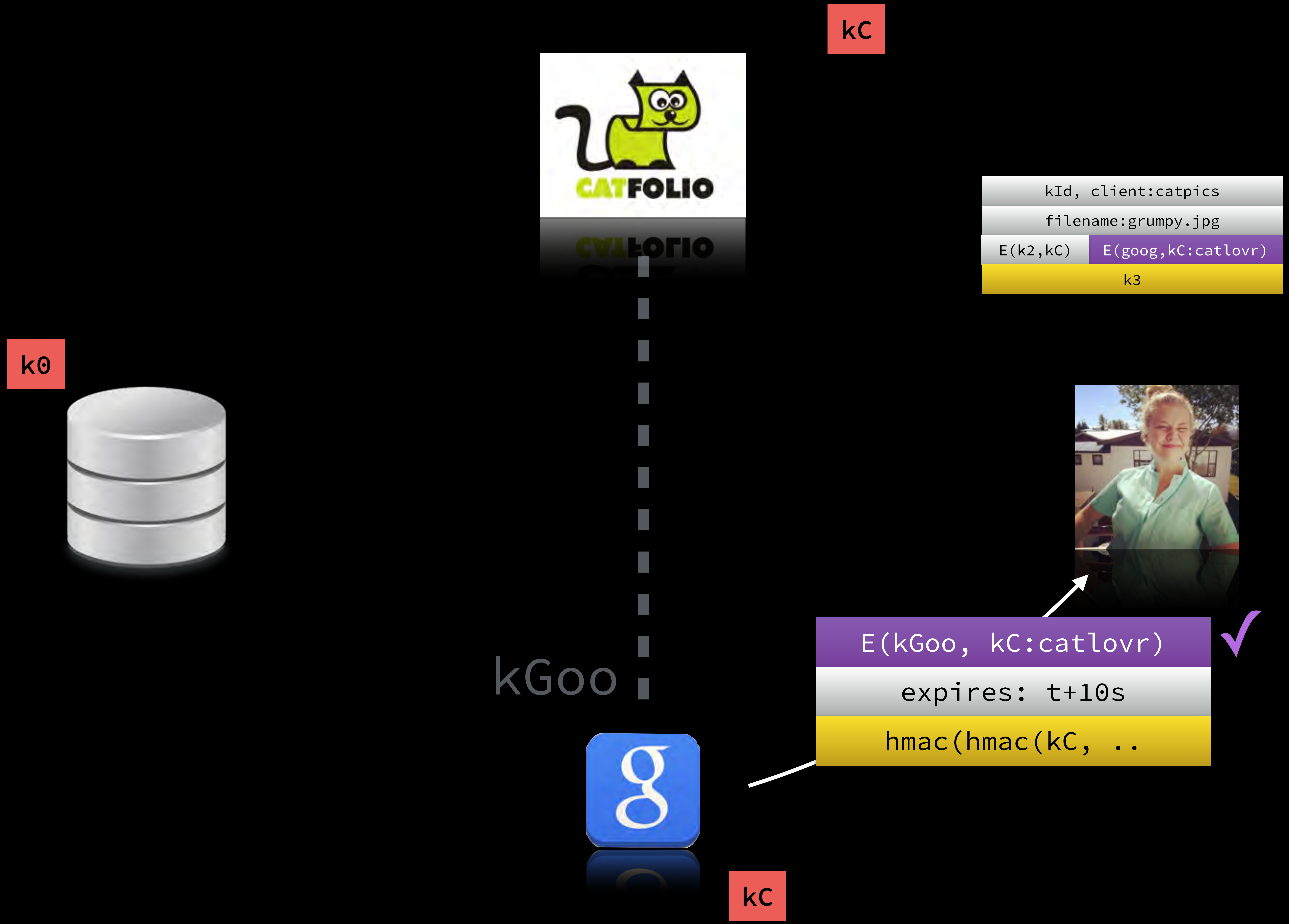
kId, client:catpics

filename:grumpy.jpg

E(k2,kC) E(kGoo,kC:catlovr)

k3

E(kGoo, kC:catlovr)



kC

kId, client:catpics

filename:grumpy.jpg

E(k2,kC) E(goog,kC:catlovr)

k3

k0

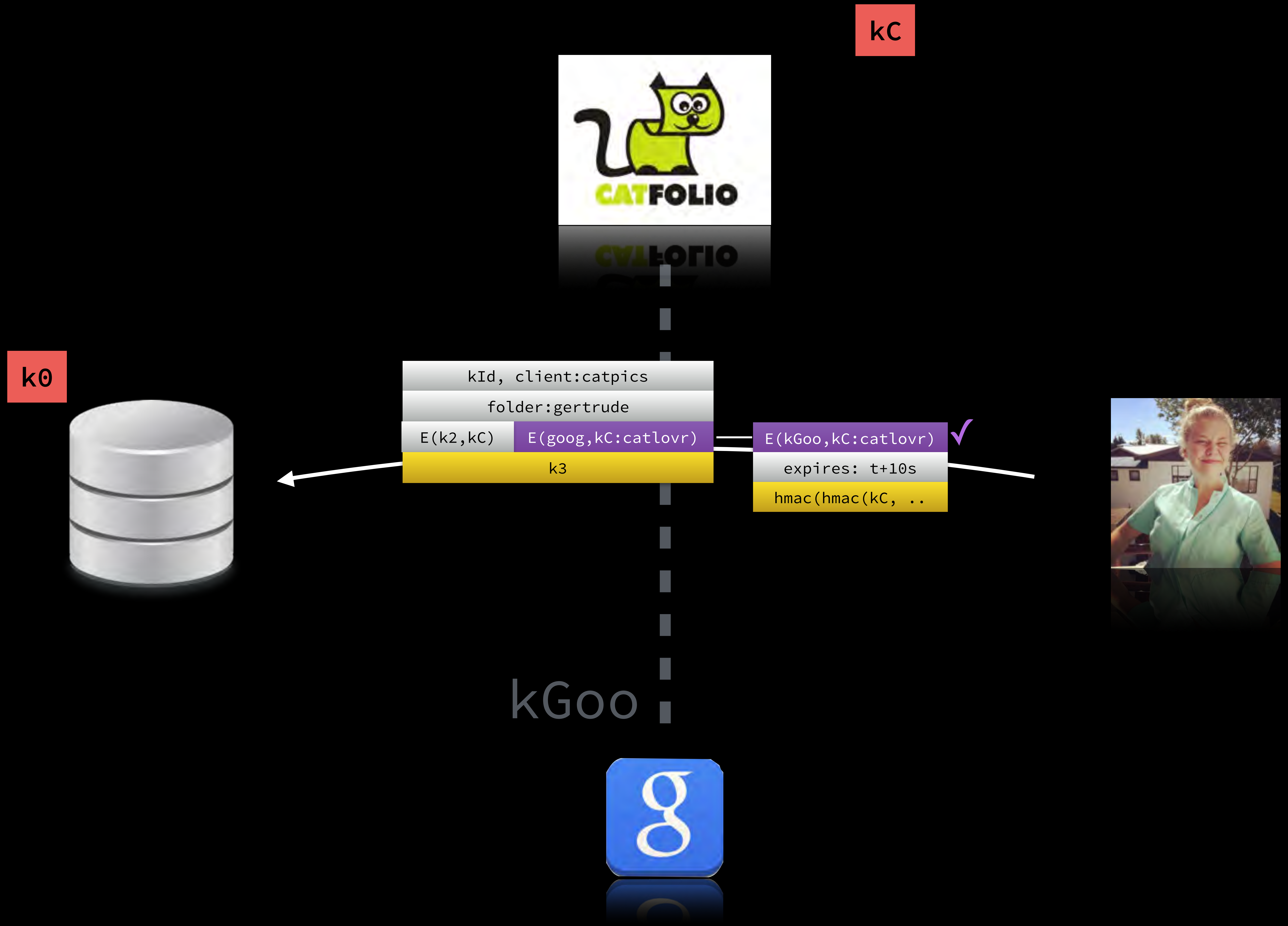
kGoo

E(kGoo, kC:catlovr) ✓

expires: t+10s

hmac(hmac(kC, ..

kC



kC

k0

kId, client:catpics

folder:gertrude

E(k2,kC) E(goog,kC:catlovr)

k3

E(kGoo,kC:catlovr) ✓

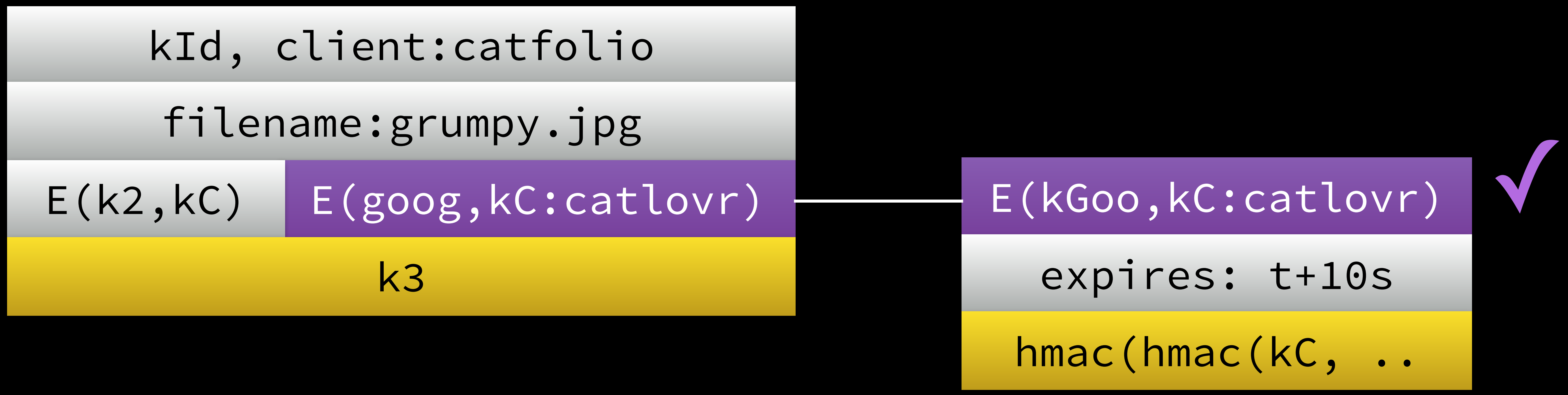
expires: t+10s

hmac(hmac(kC, ..

kGoo

Verification

- As before, check signature & caveats
- For each third-party caveat, recursively:
  - Verify its discharge macaroon
  - Check discharge macaroon caveats



kId, client:catfolio

filename:grumpy.jpg

E(k2,kC) E(goog,kC:catlovr)

k3

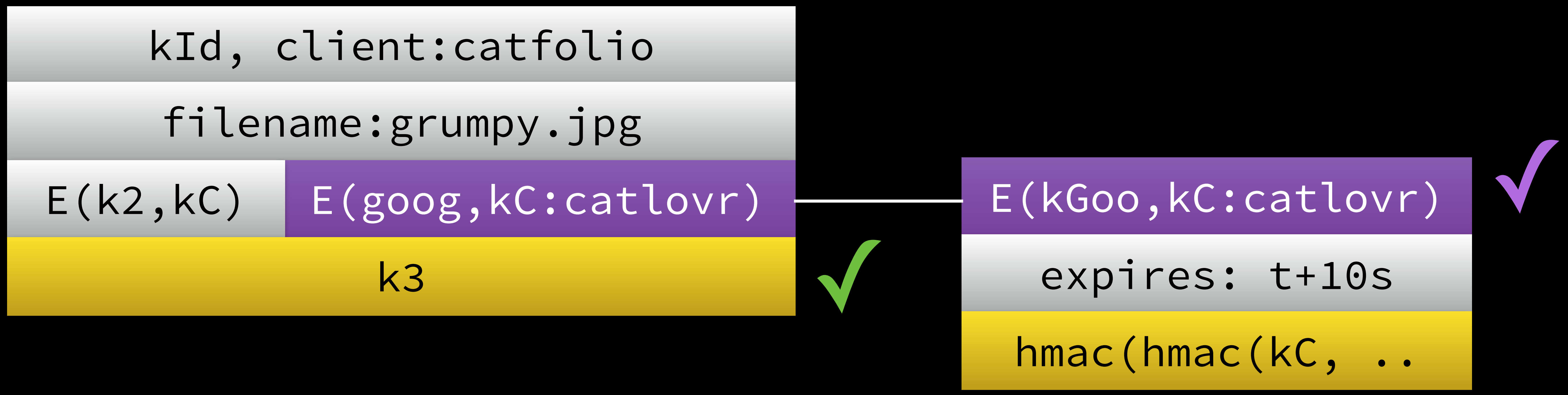
E(kGoo,kC:catlovr) ✓

expires: t+10s

hmac(hmac(kC, ..

Verification

- As before, check signature & caveats
- For each third-party caveat, recursively:
  - Verify its discharge macaroon
  - Check discharge macaroon caveats



kId, client:catfolio

filename:grumpy.jpg

E(k2,kC) E(goog,kC:catlovr)

k3 ✓

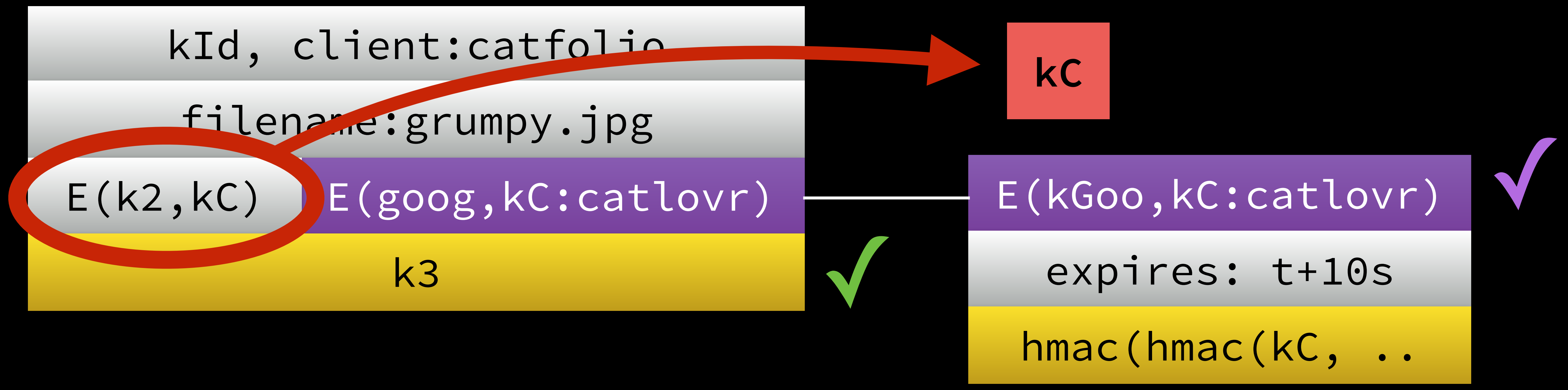
E(kGoo,kC:catlovr) ✓

expires: t+10s

hmac(hmac(kC, ..

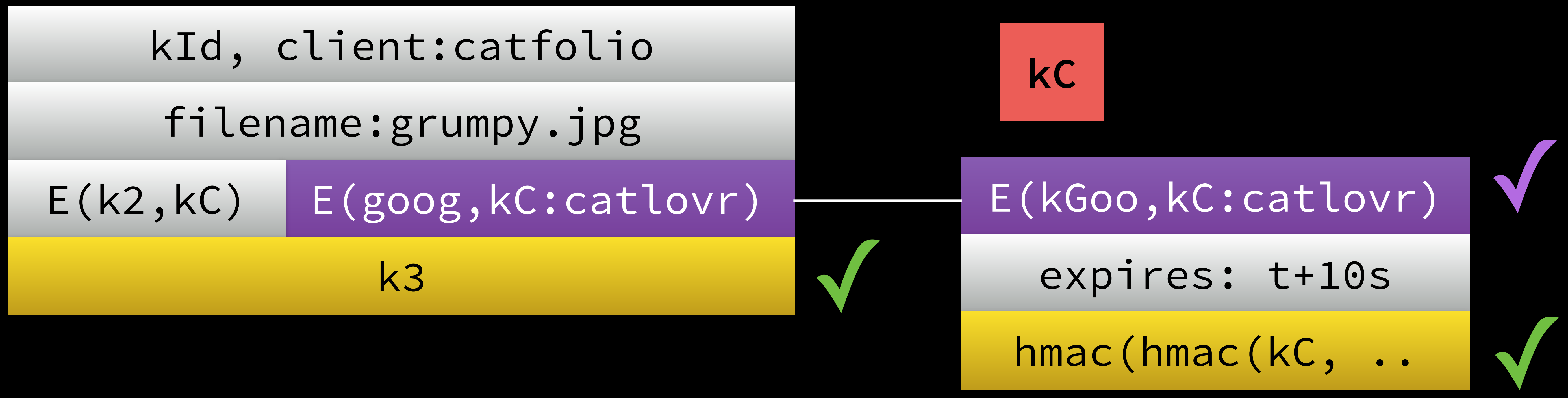
Verification

- As before, check signature & caveats
- For each third-party caveat, recursively:
  - Verify its discharge macaroon
  - Check discharge macaroon caveats



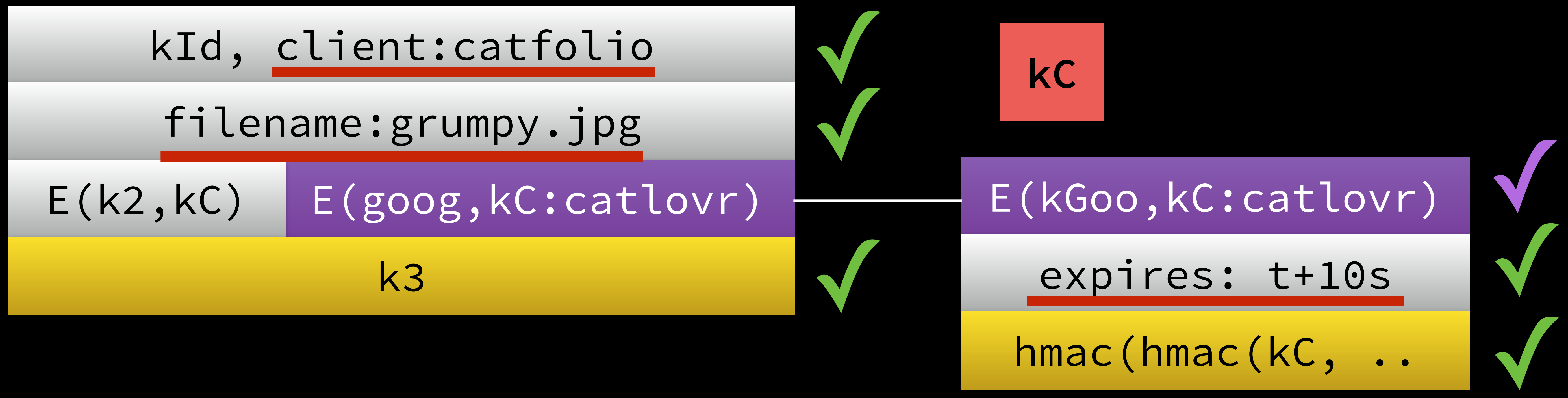
Verification

- As before, check signature & caveats
- For each third-party caveat, recursively:
  - Verify its discharge macaroon
  - Check discharge macaroon caveats



Verification

- As before, check signature & caveats
- For each third-party caveat, recursively:
  - Verify its discharge macaroon
  - Check discharge macaroon caveats



Benefits

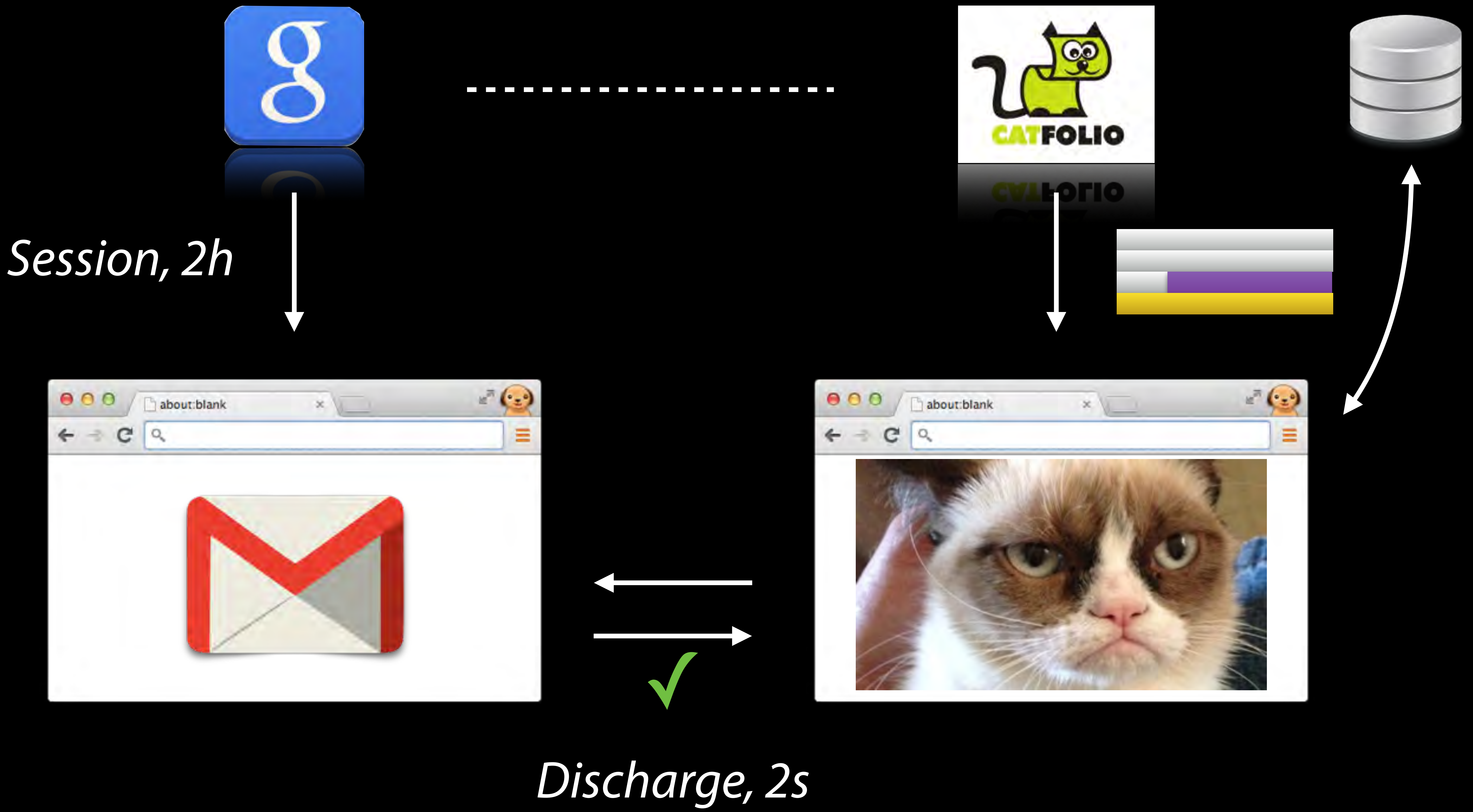
Speed: HMACs = Very light-weight, and fast.

Timeliness: Can require fresh credentials and revocation checks on every request.

Flexibility: Contextual confinements, attenuation, delegation, third-party caveats.

Adoptability: HMACs can run anywhere.

Example: Identity

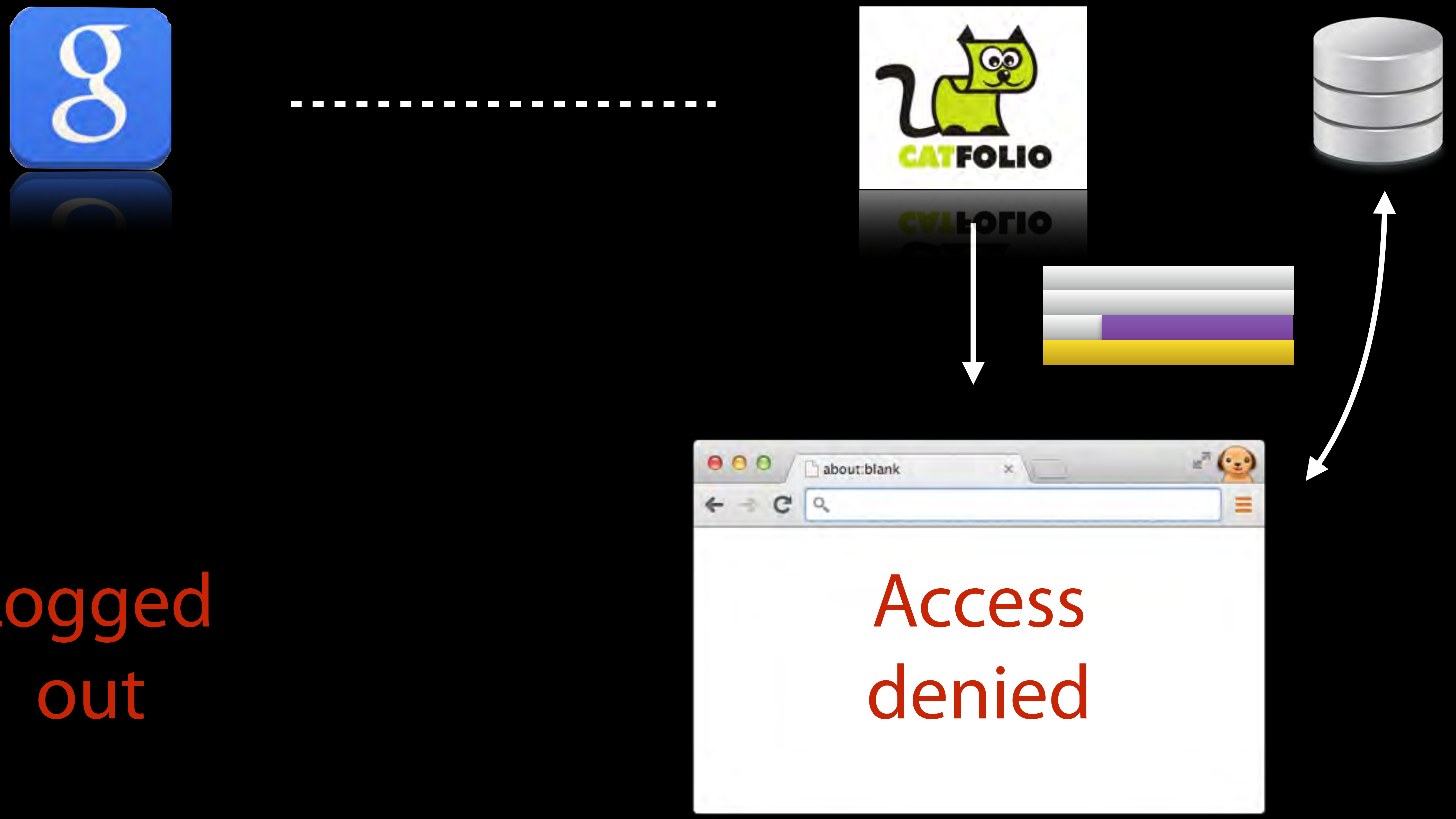


Session, 2h

✓

Discharge, 2s

Example: Identity



Logged out

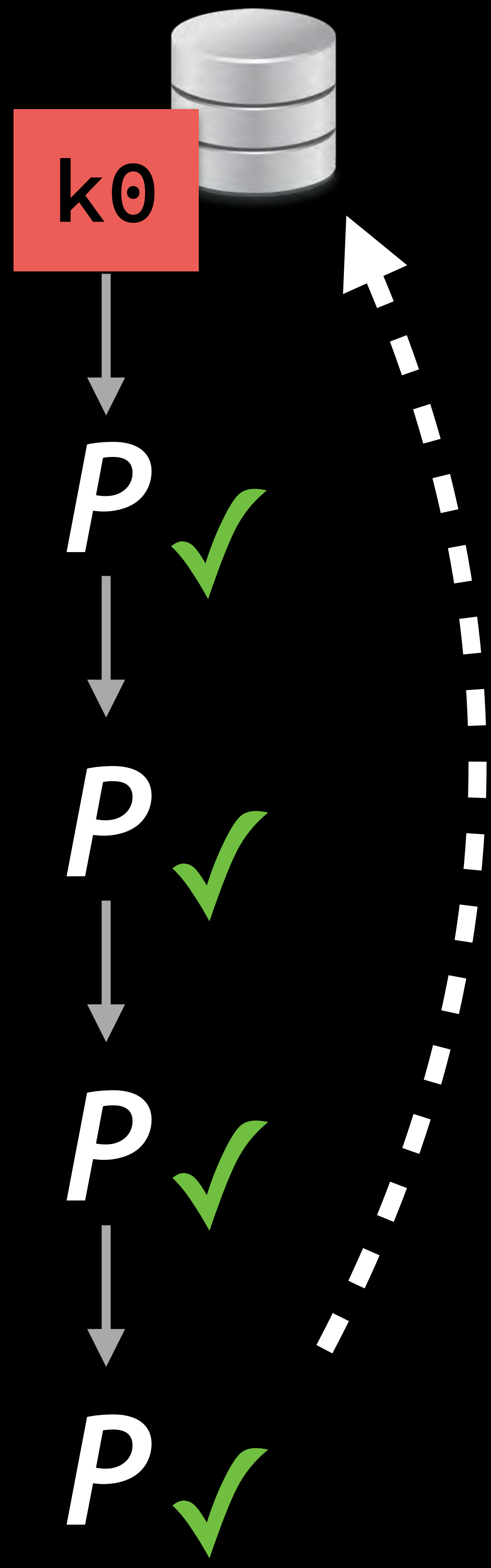
Access denied

Performance

	JS/Chrome	Node.js
Minting	27 µs	19 µs
Add caveat	~300 µs	56 µs
Verify	~360 µs	70 µs
To/from JSON	~3.5 µs	~4.0 µs

Performance

	JS/Chrome	Node.js
Minting	37k /s	53k /s
Add caveat	3.3k /s	18k /s
Verify	3.7k /s	14k /s
To/from JSON	380k /s	250k /s



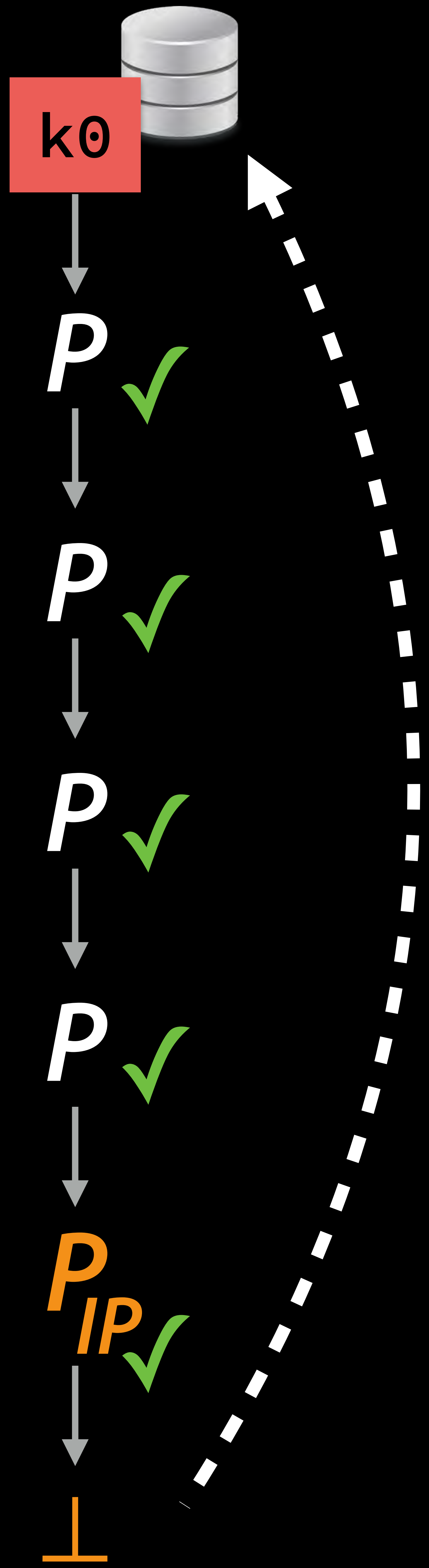
k0

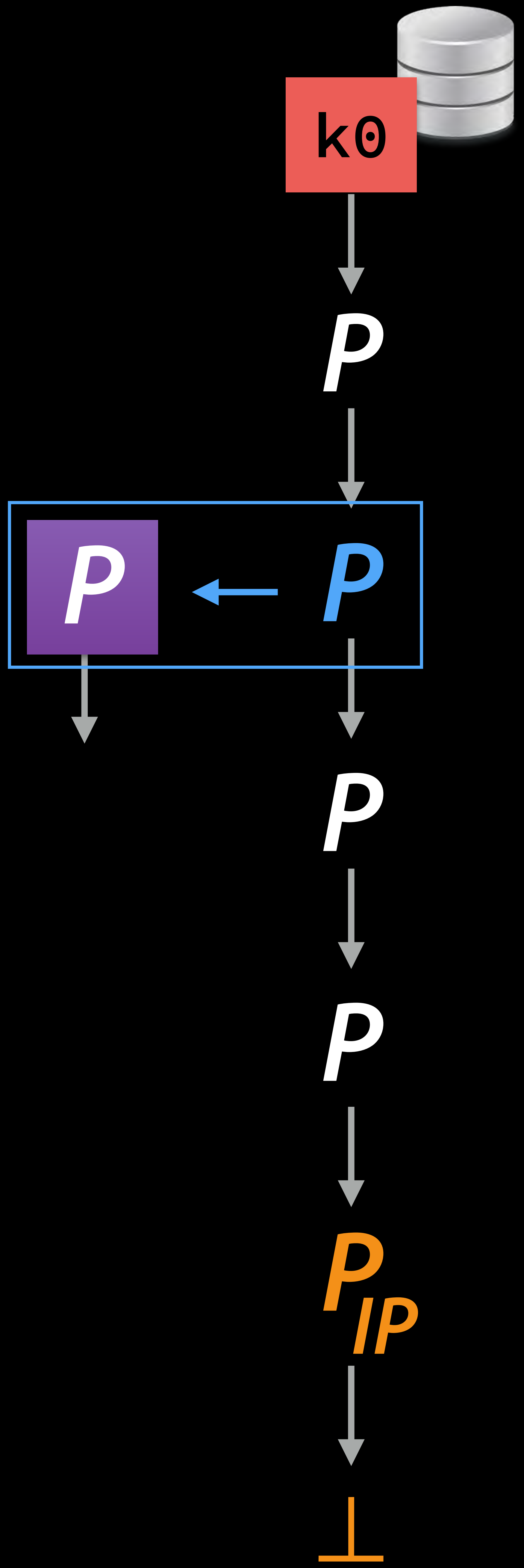
P ✓

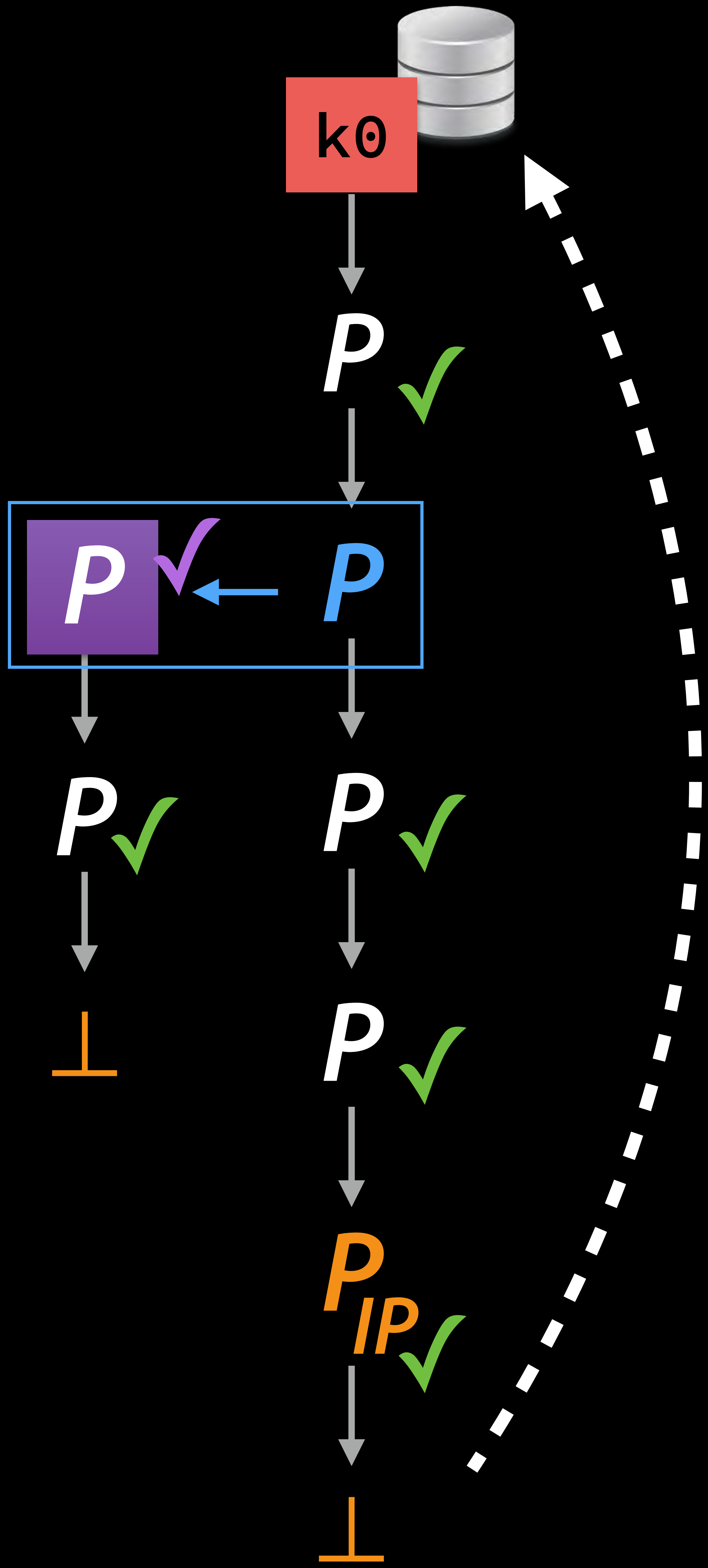
P ✓

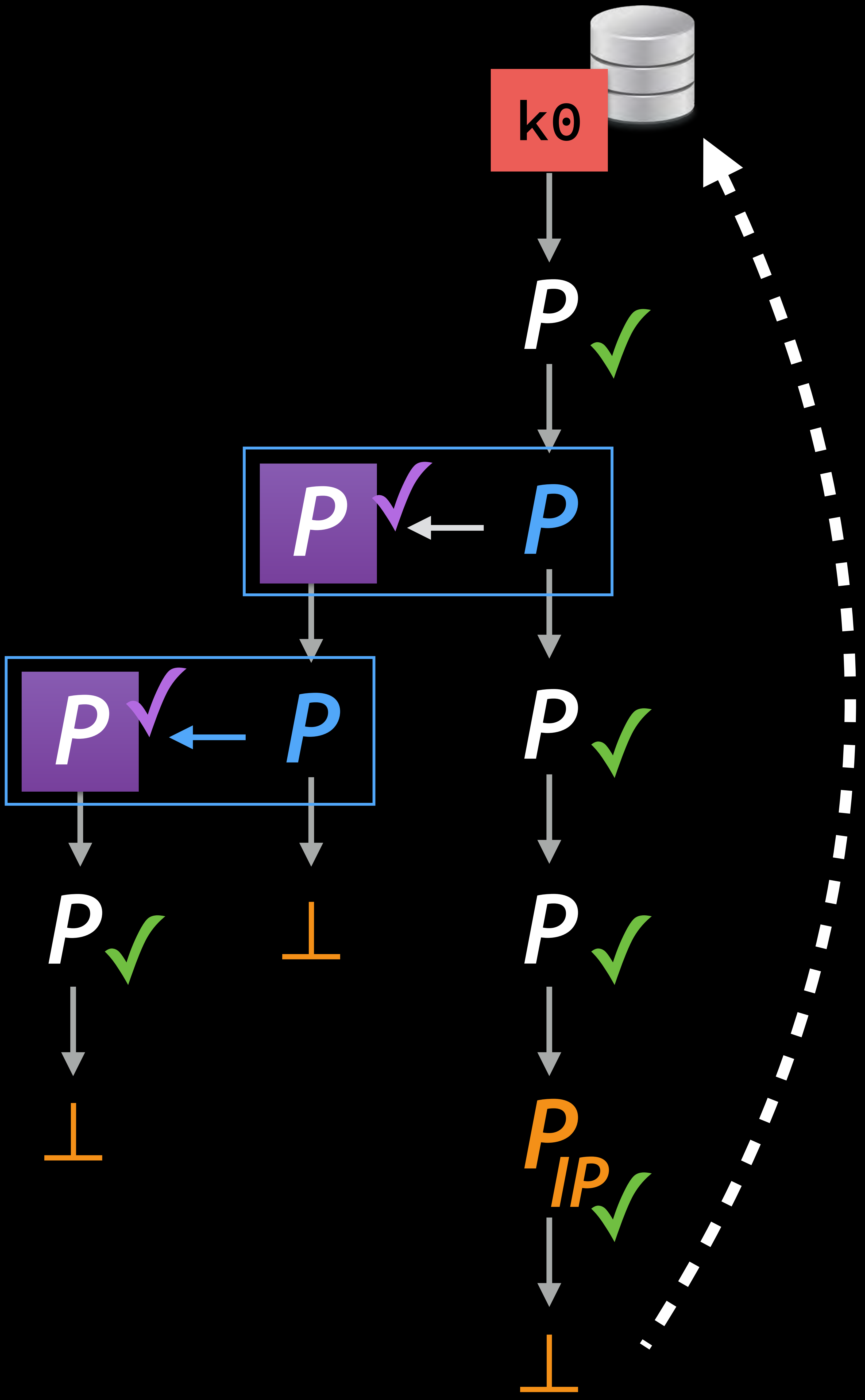
P ✓

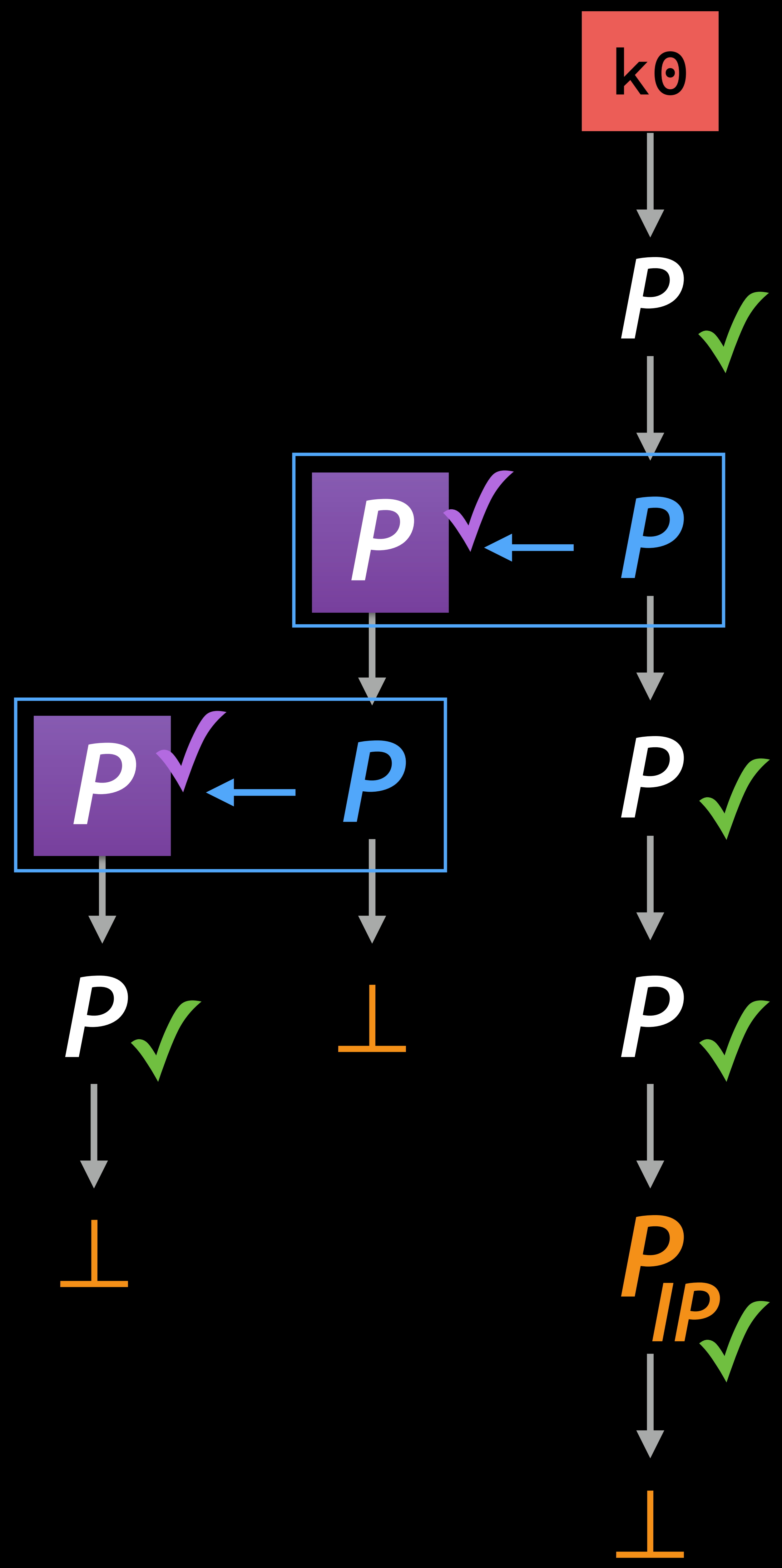
P ✓


k0
P ✓
P ✓
P ✓
P ✓
P_IP ✓
⊥


k0
P
P
P
P
P
P_{IP}
⊥


k0
P ✓
P ✓
P
P ✓
P ✓
P ✓
⊥
P_IP ✓
⊥


k0
P ✓
P ✓ ← P
P ✓ ← P
P ✓
P ✓
⊥
⊥
P ✓
P_IP ✓
⊥


k0
P ✓
P ✓ ← P
P ✓ ← P
P ✓
⊥
P ✓
P ✓
⊥
P_IP ✓
⊥

Fun stuff in the paper

- Formal definitions and verification
- A variant based on public-key crypto instead of hash-chains
- Comparison to SPKI/SDSI
- Other uses for first- and third-party caveats

TL;DR

- Macaroons are better cookies.
- Fast: Can be used anywhere, often.
- Flexible: Fit many applications, delegable.
- Secure:
  Caveats limit scope; Contextual caveats make stolen macaroons useless.