

Lukas'

Privacy by Design

Privacy by Design

Lukas Rieder, March 5th 2026

Privacy by Design

Today

Lukas Rieder, March 5th 2026

Talking about **Data Privacy**

Talking about **Elixir & Erlang**

Talking about **Encryption**

in practice

Data Privacy

GDPR generally divides rights and responsibilities into three roles.

subject

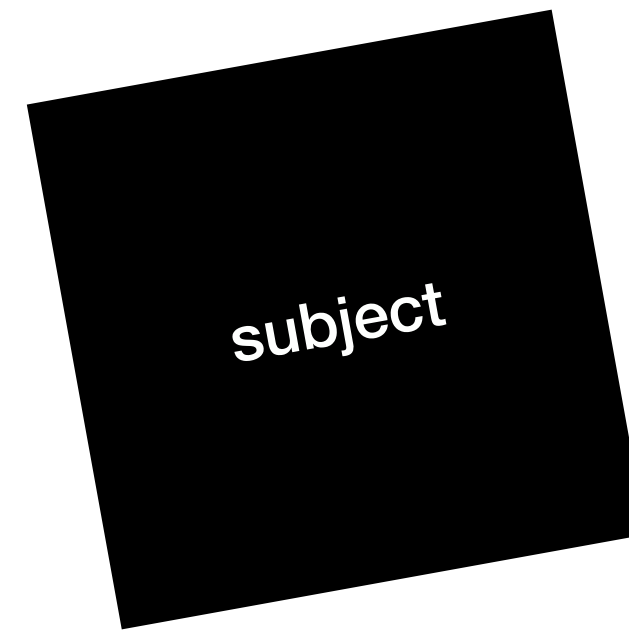
processor

controller

Data Subject

e.g. your customer,
whose data is
processed.

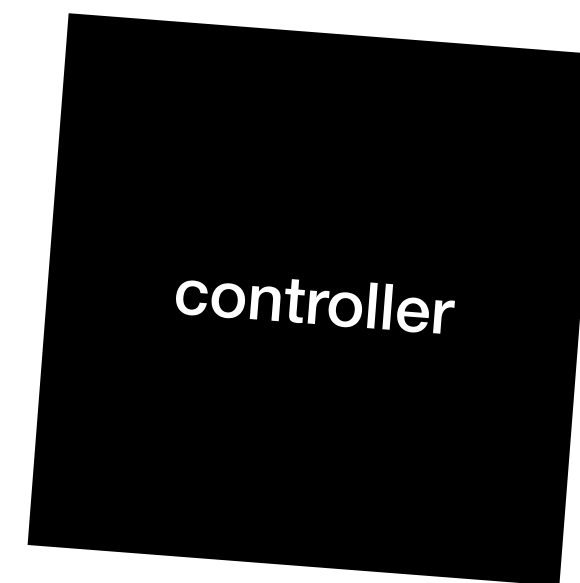
Right to information,
correction and
deletion of their data.



Data Controller

e.g. your company
that collects personal
data of its customers.

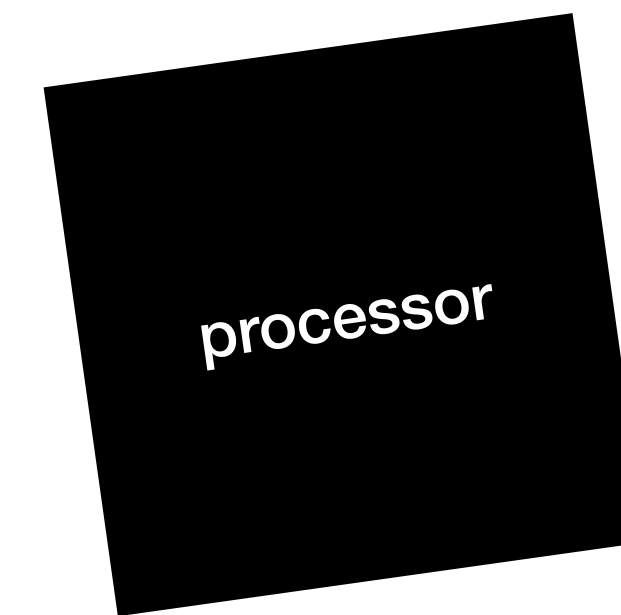
Obligation to ensure
lawful processing of
the data.



Data Processor

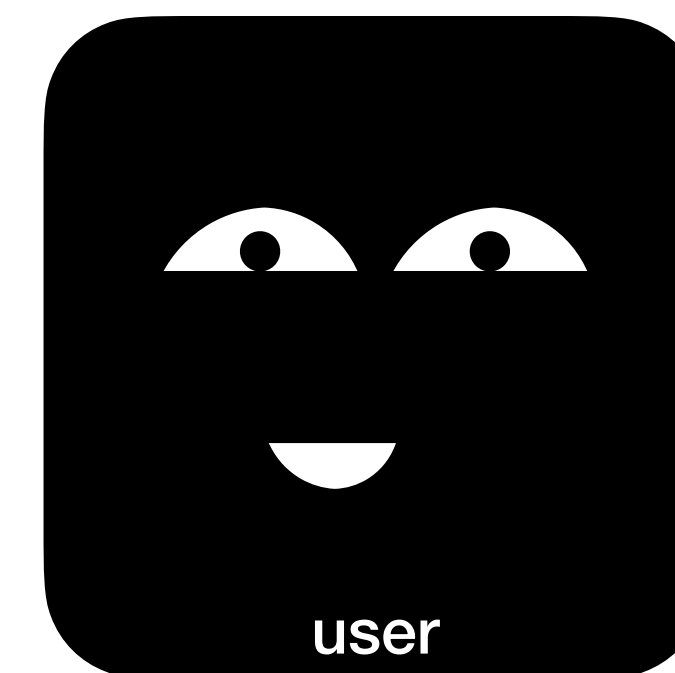
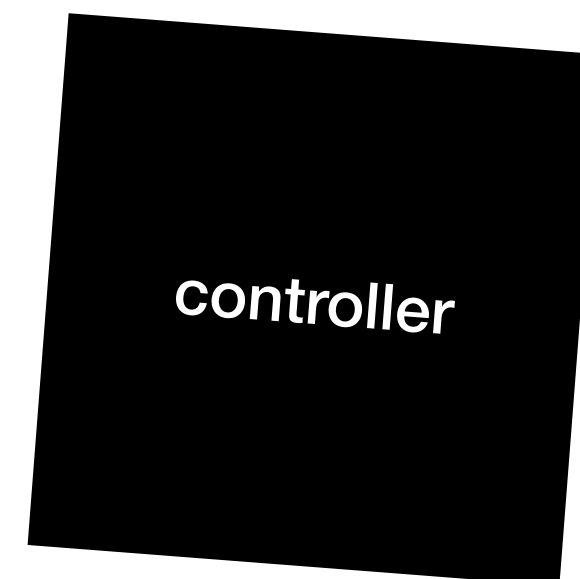
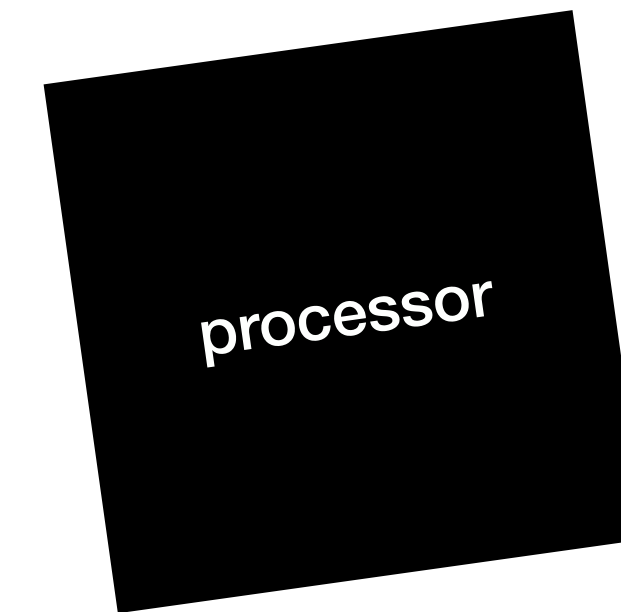
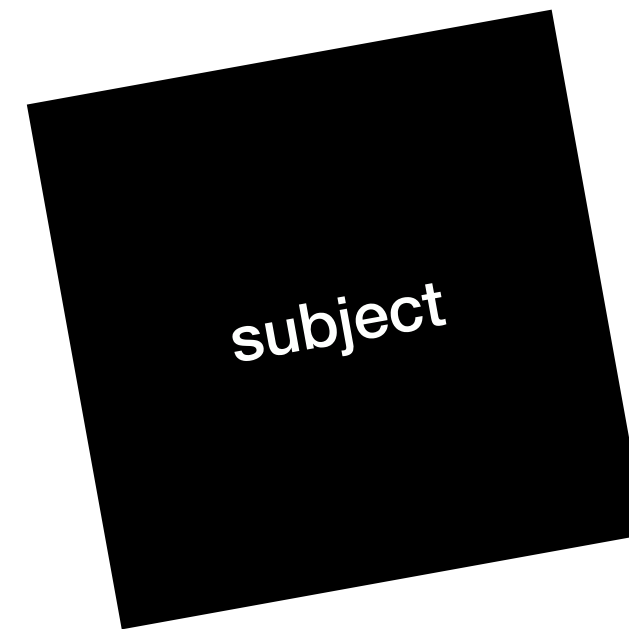
e.g. your hosting
provider that stores
and processes data.

Obligation to comply
and protect the data
according to the
controllers instruction.



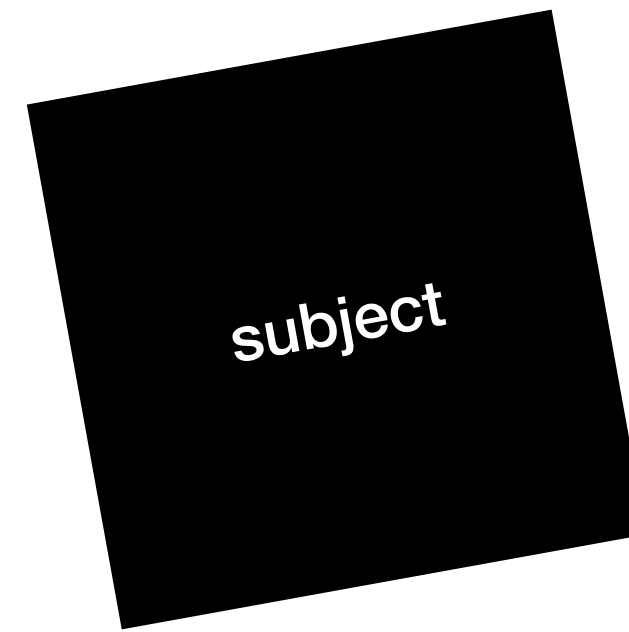
Data Security

Data Security is the practice of ensuring that personal data is only accessible to the parties that are explicitly authorized and to no one else.



Data Security

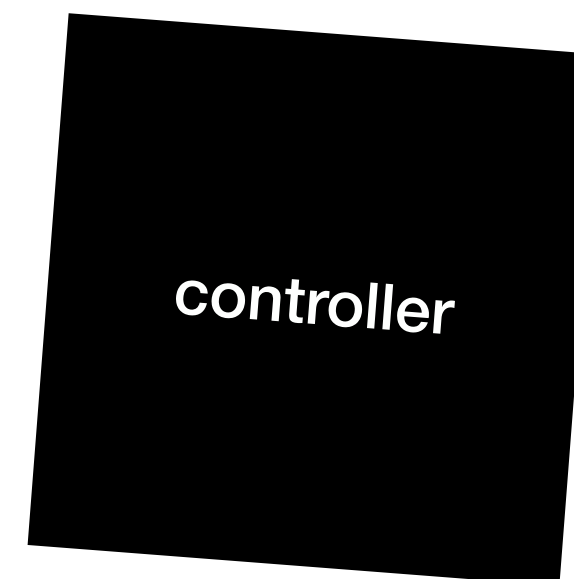
The subject trusts that their data is handled according to declared purpose, not exposed by design flaws or operational failures.





Data Security

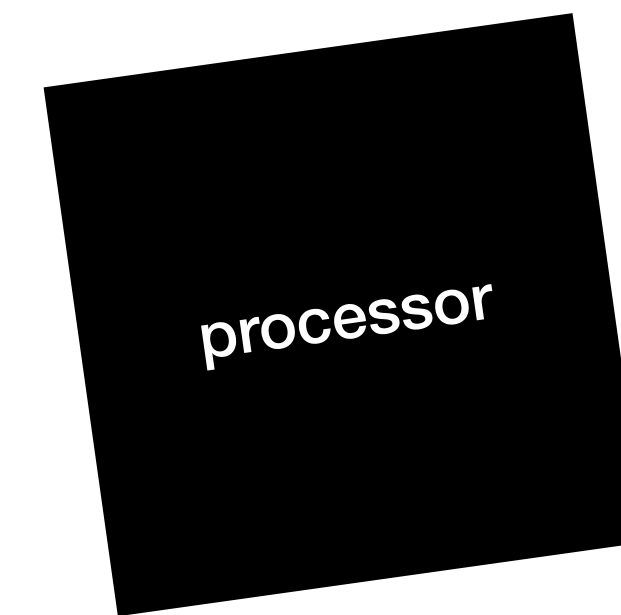
The controller defines why and under what guarantees data is processed — including security requirements and acceptable risk.





Data Security

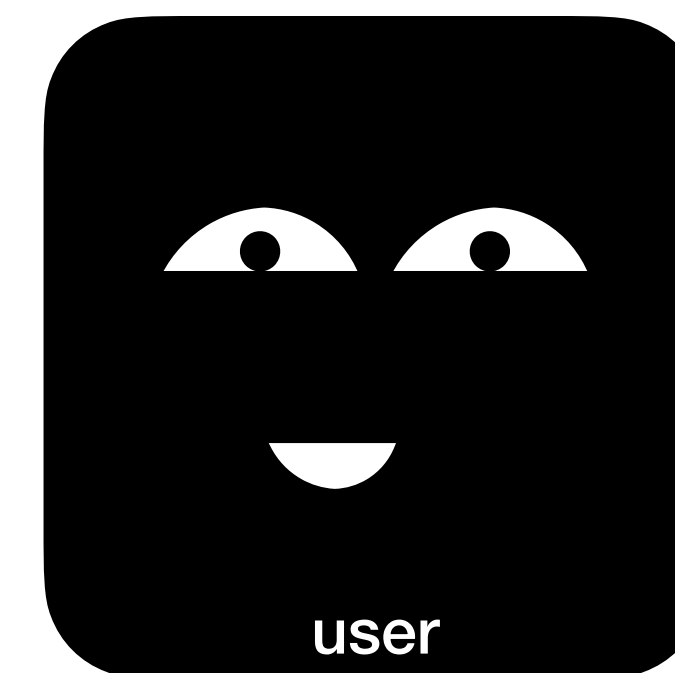
The processor is responsible for how data is protected in practice: **encryption, access control, isolation, and secure storage.**





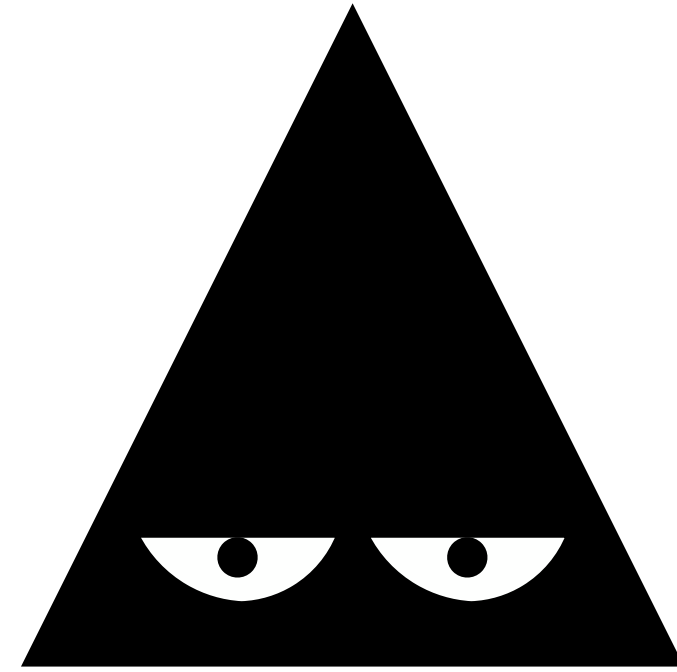
Data Security

The users have rights to their data, but we are not responsible for their own systems and behaviors.



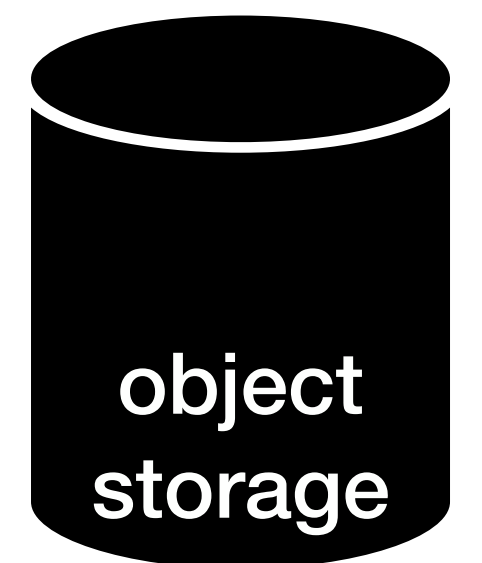
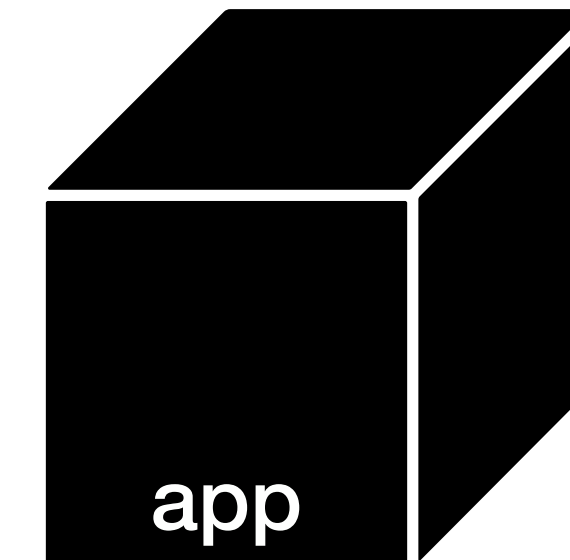
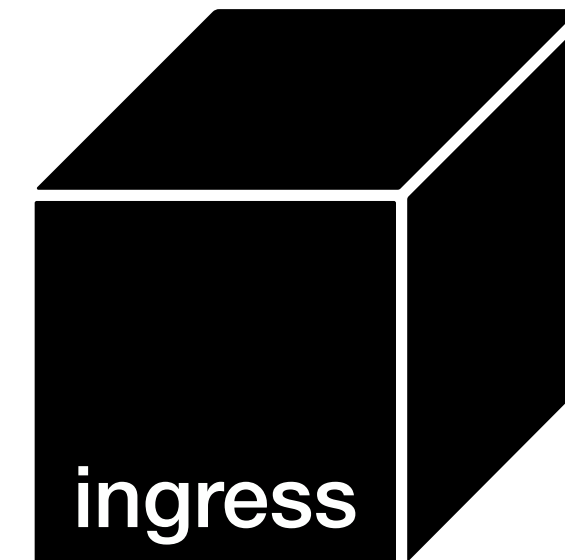
Encryption, Access Control, Isolation, and Secure Storage





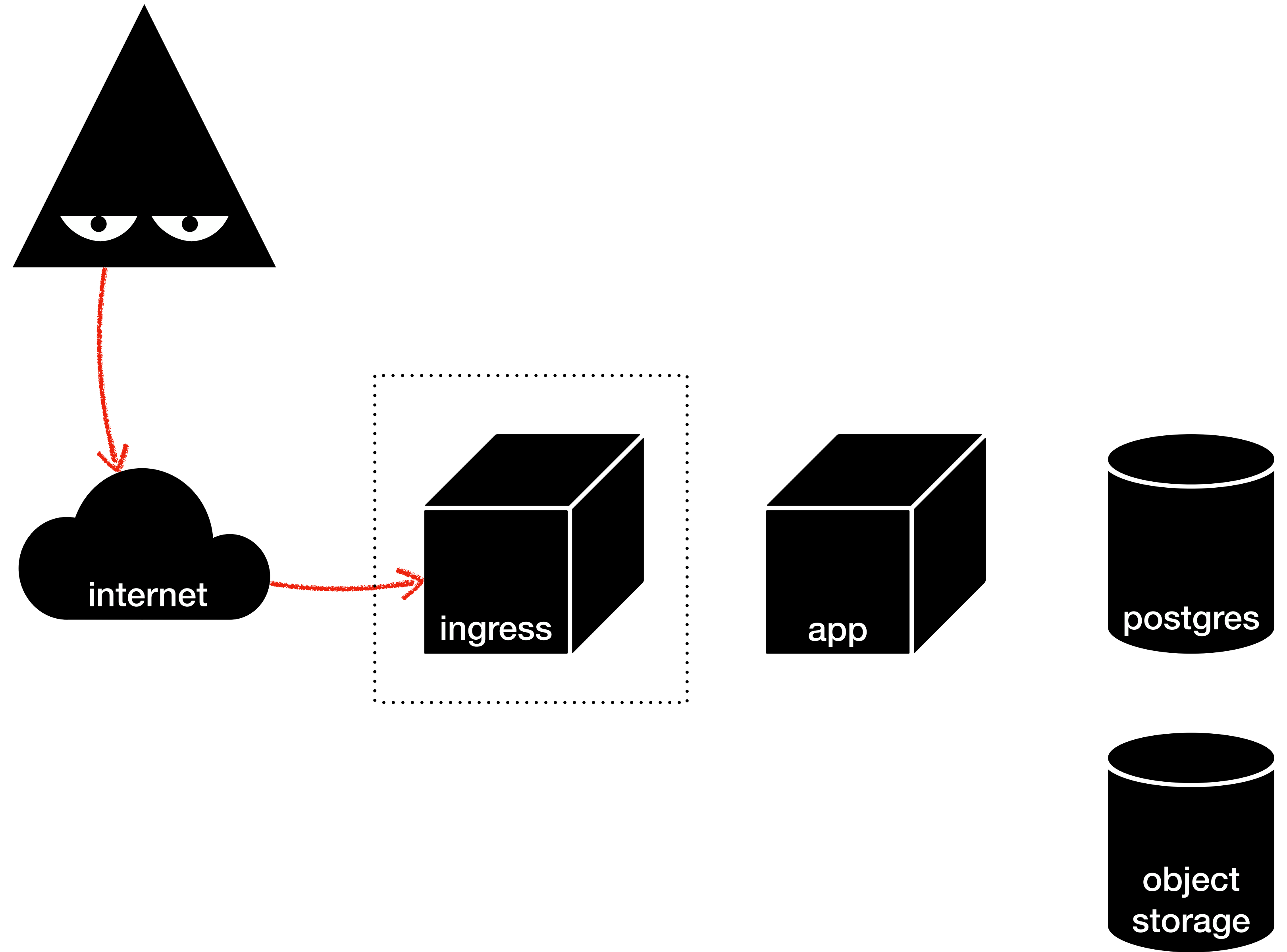
Data Security

Data security is not a single mechanism. It is a layered response to different adversary capabilities.



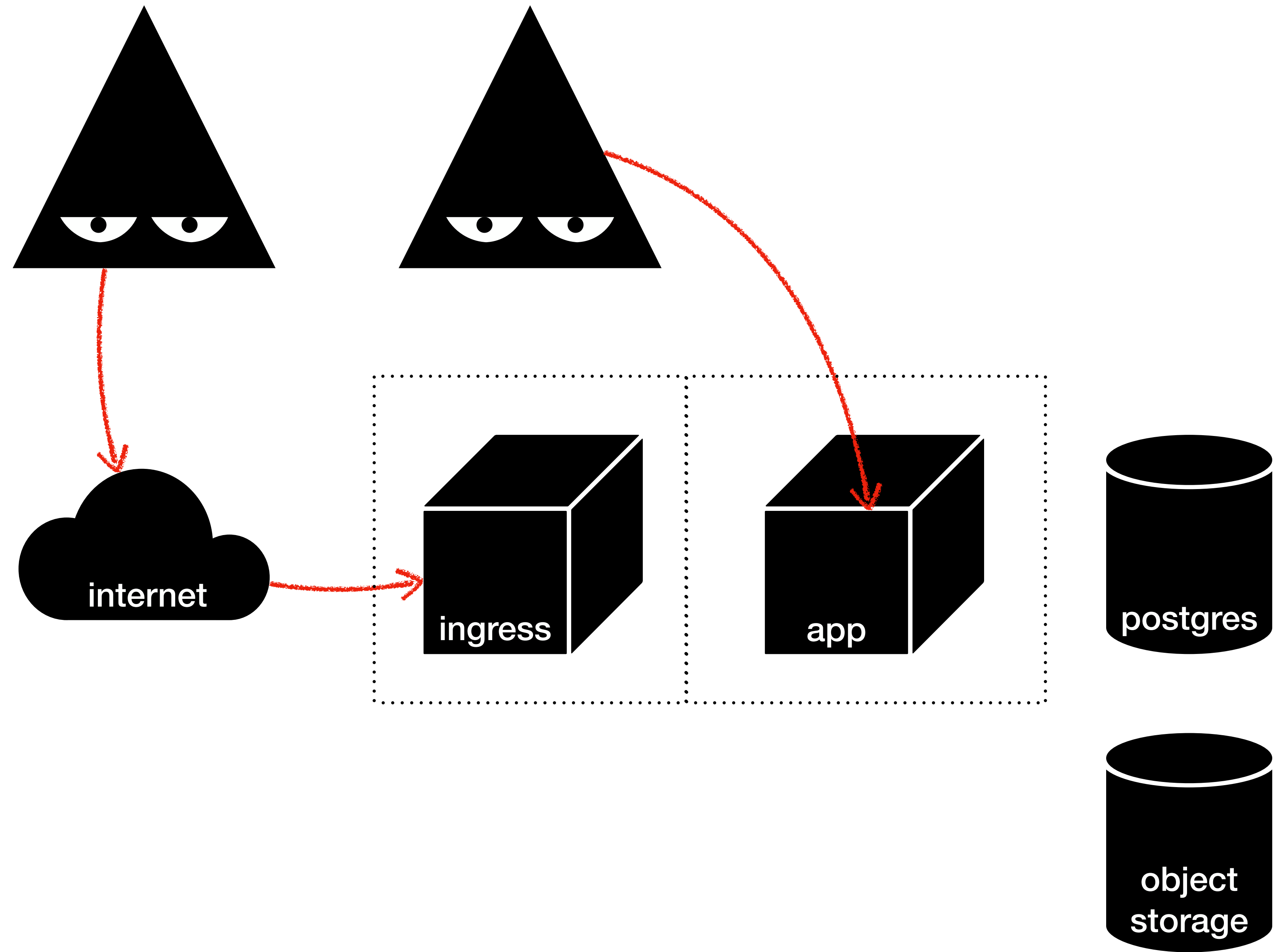
Data Security

Data security is not a single mechanism. It is a layered response to different adversary capabilities.



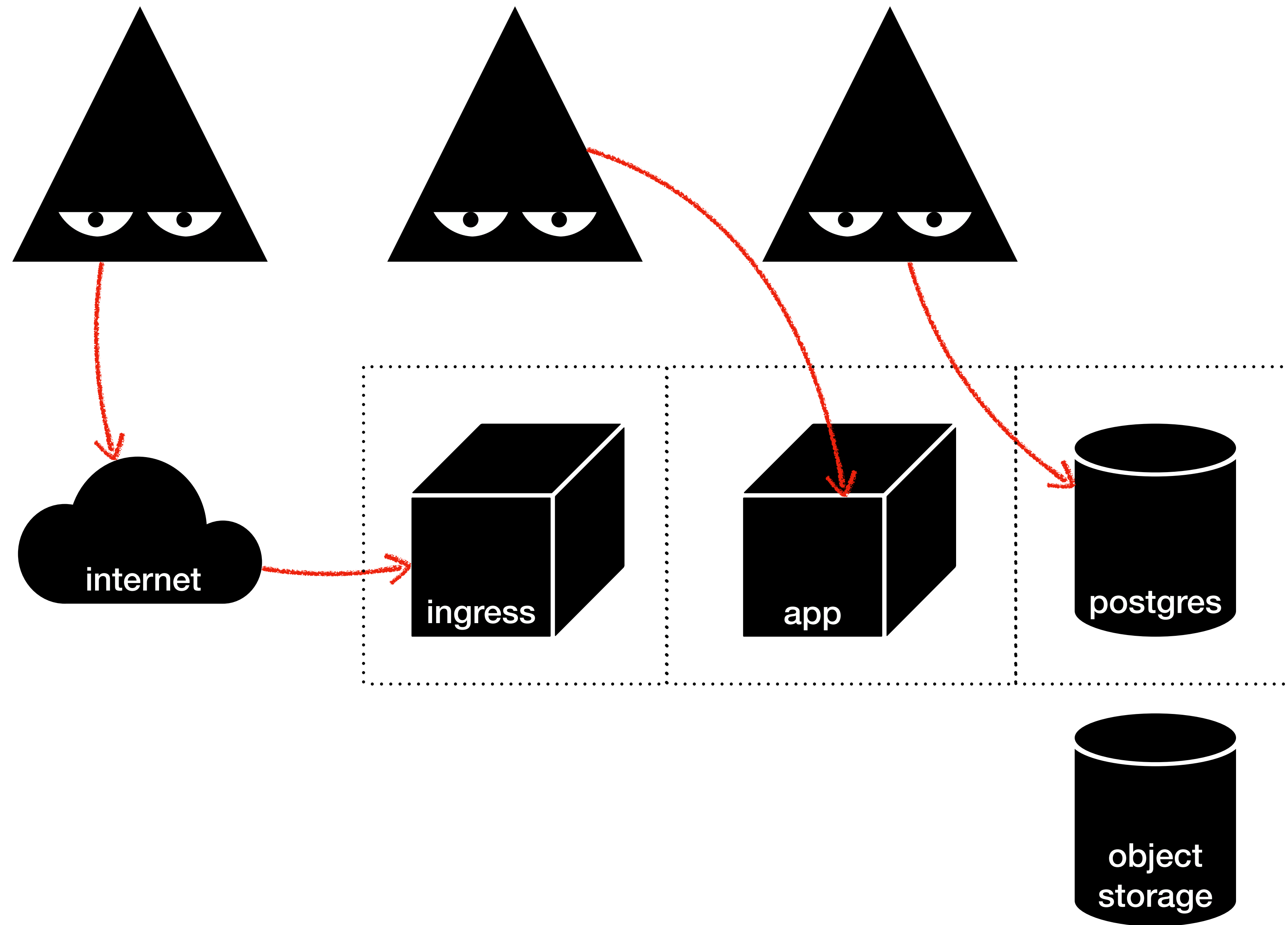
Data Security

Data security is not a single mechanism. It is a layered response to different adversary capabilities.



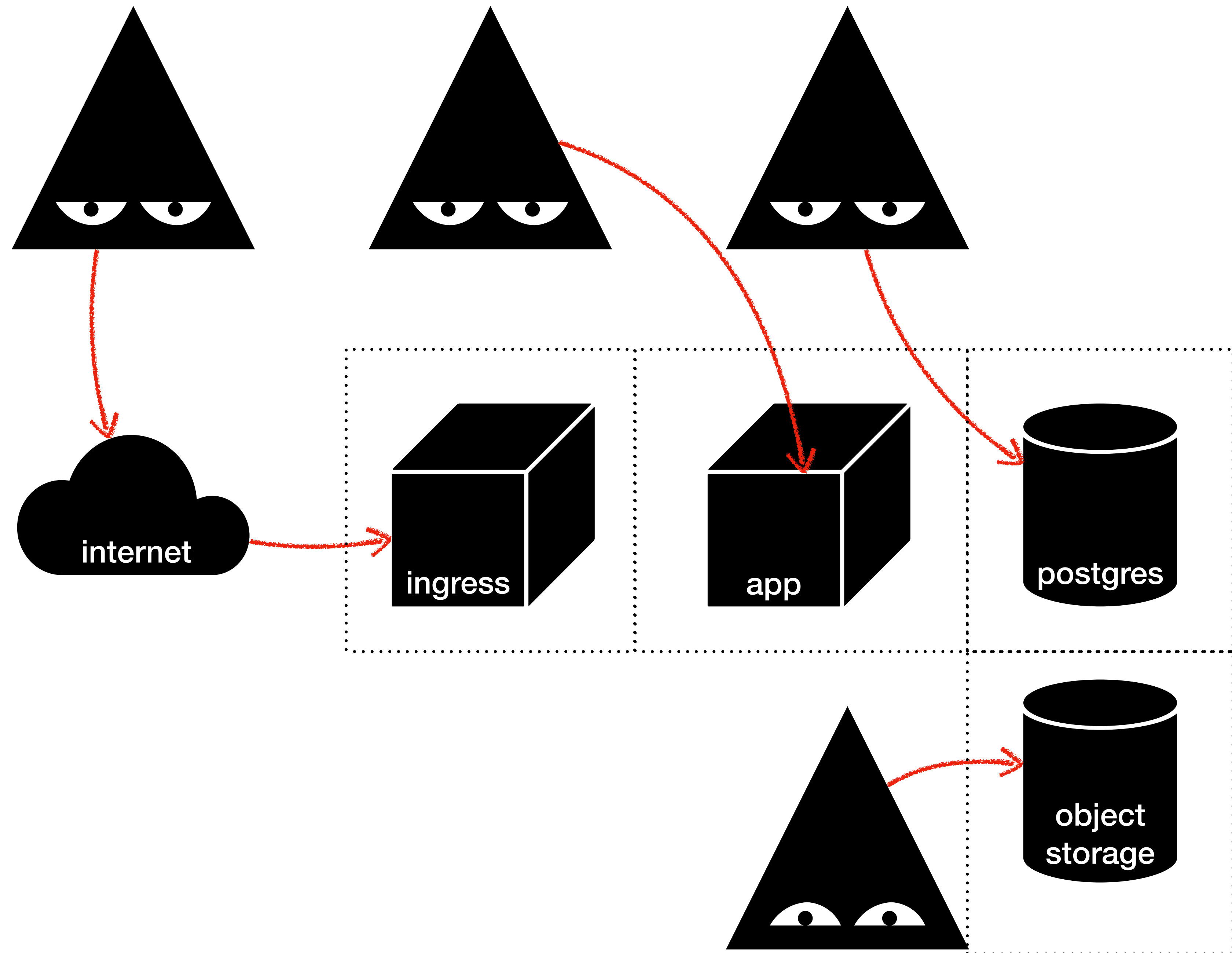
Data Security

Data security is not a single mechanism. It is a layered response to different adversary capabilities.



Data Security

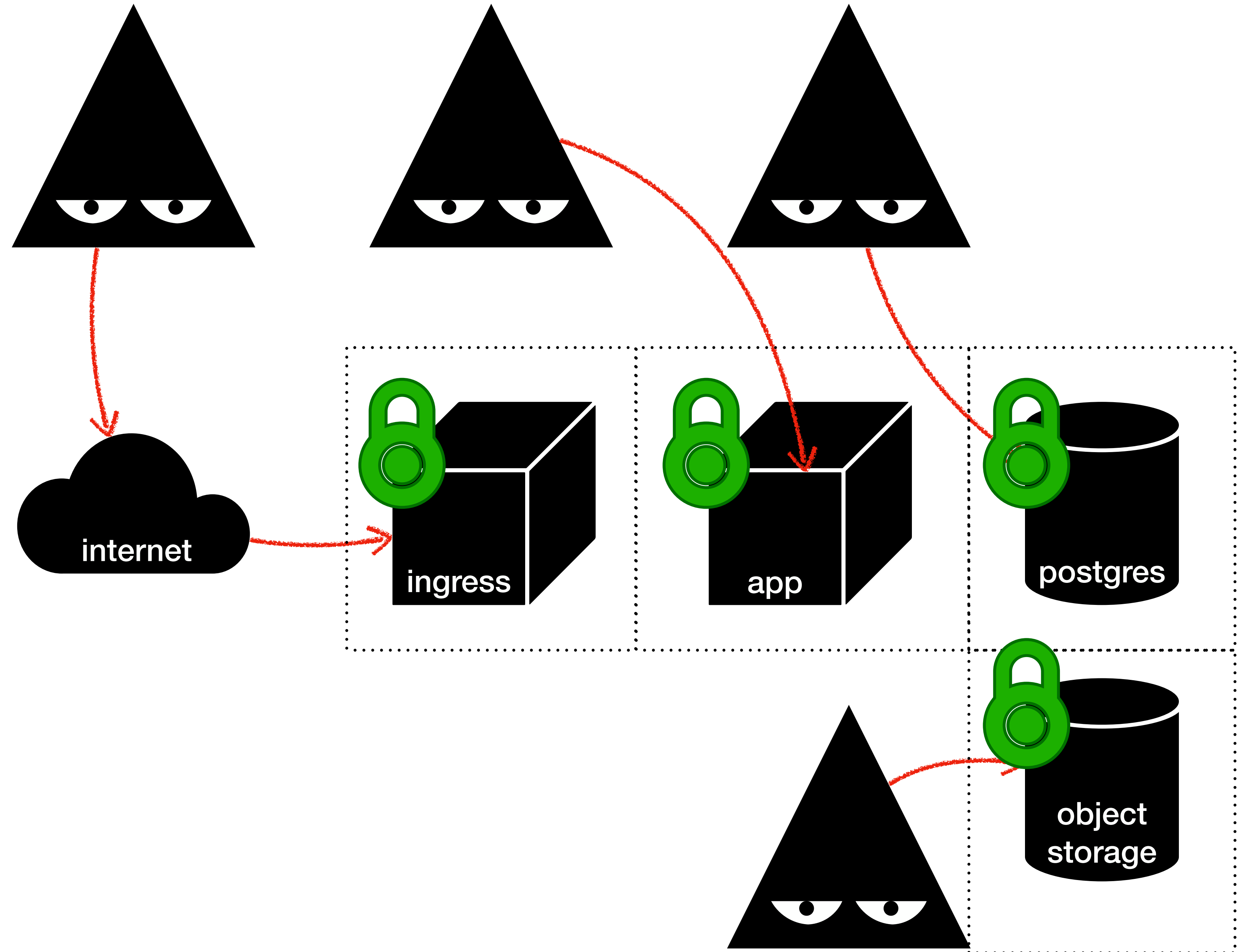
Data security is not a single mechanism. It is a layered response to different adversary capabilities.

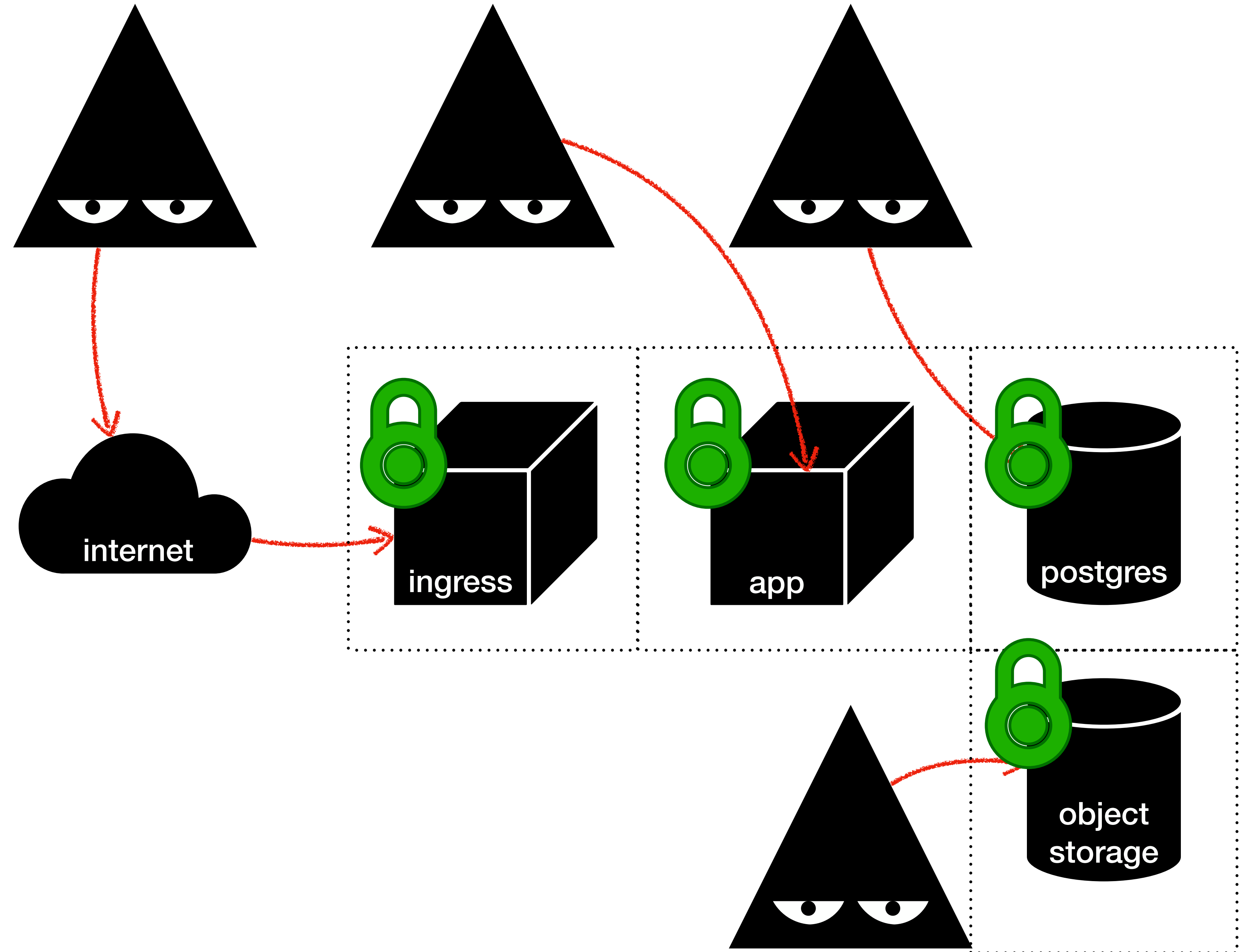


Access Control

Access control prevents unauthorized use.

Encryption limits damage when access control fails.





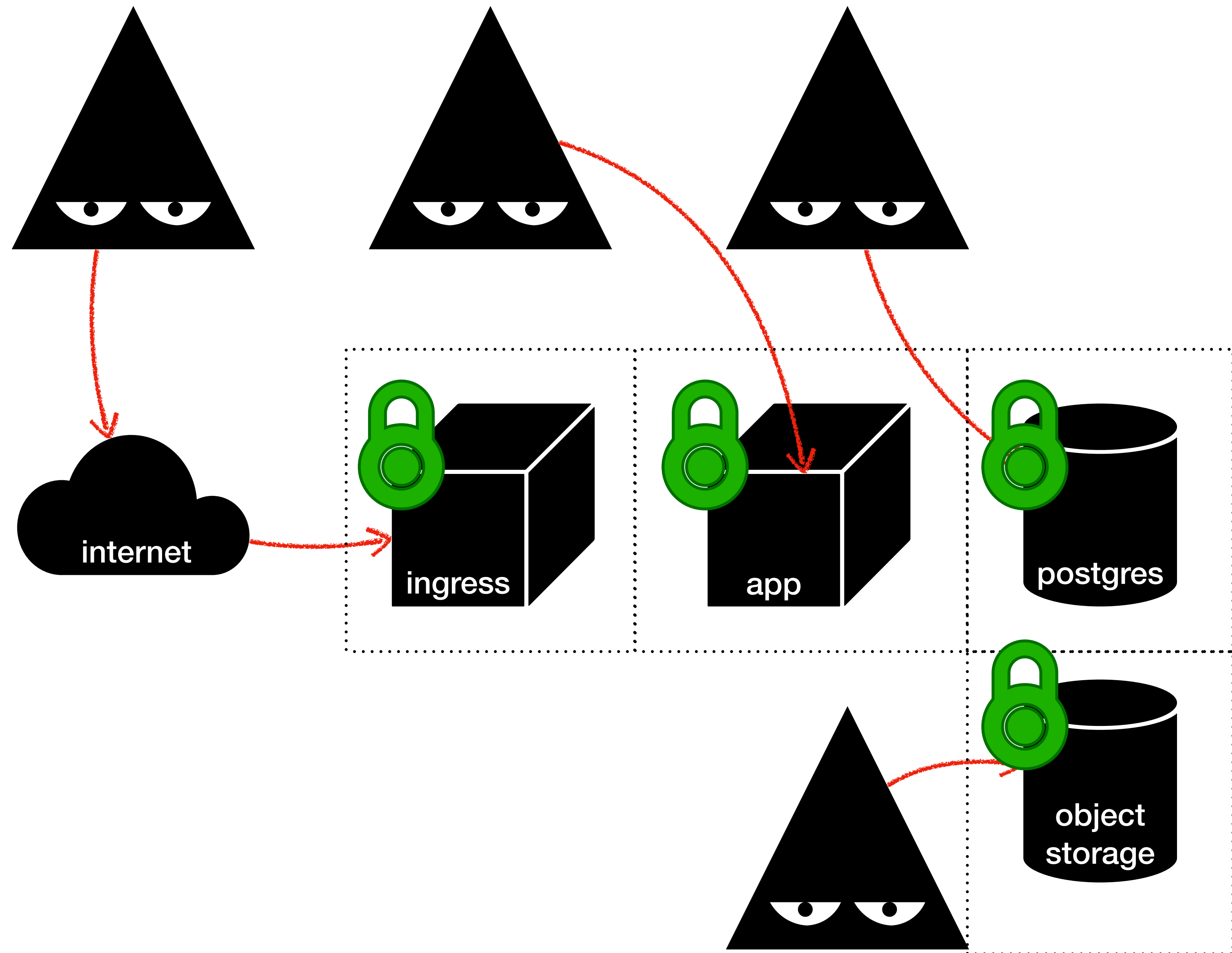
Encryption

Access control prevents unauthorized use.

Encryption limits damage when access control fails.

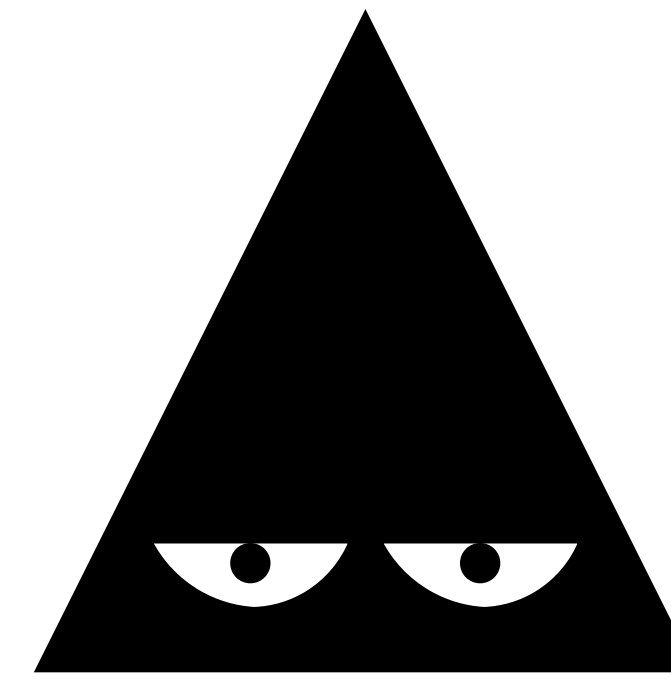
Secure Storage

Secure storage does not mean "hard to access", it means "useless without the correct cryptographic context".



Wrapped Key Encryption



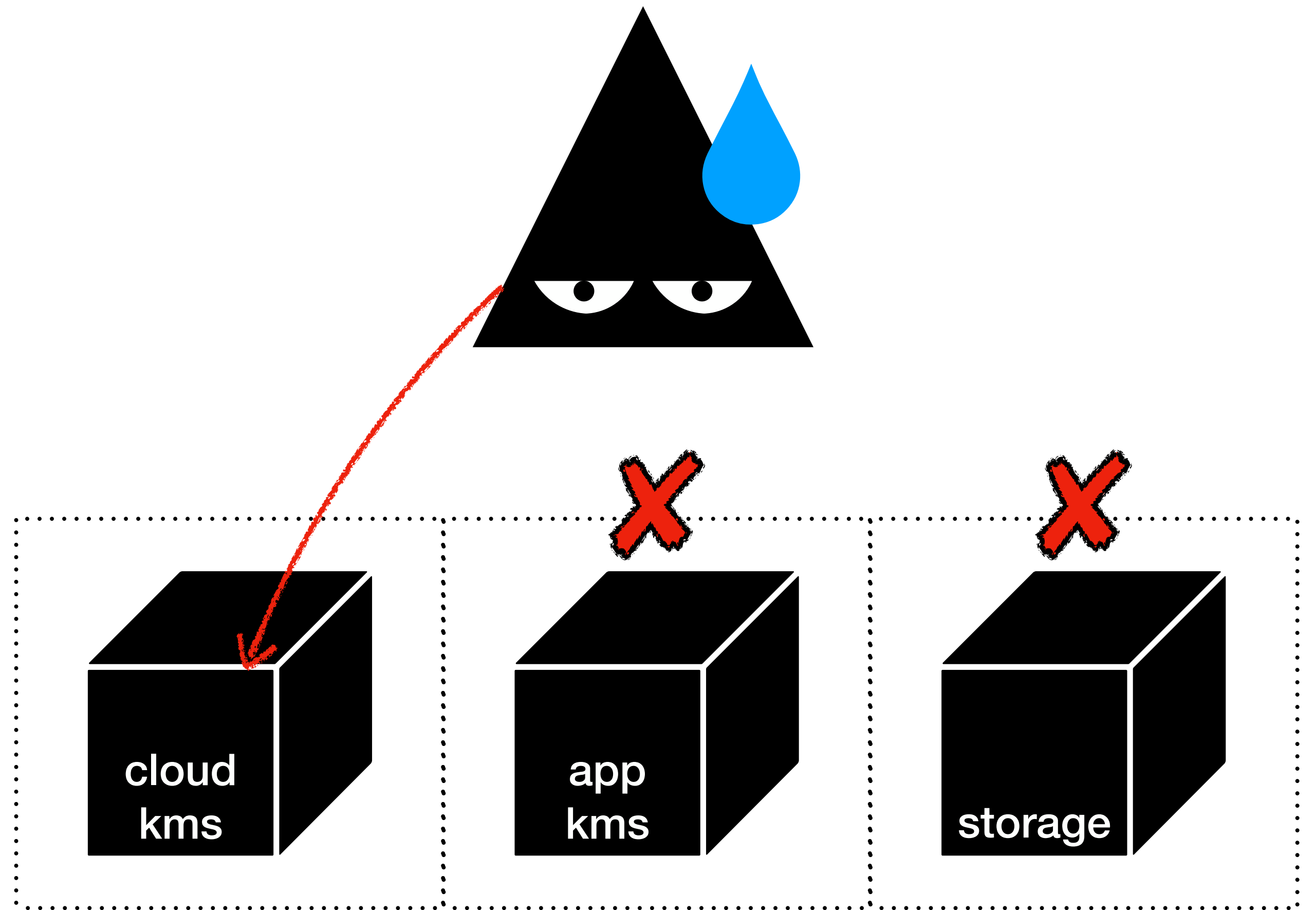


Wrapped Key Encryption

Wrapped key encryption is not about "stronger" encryption, it is about splitting trust across systems so that no single failure is catastrophic.

Root Master Key

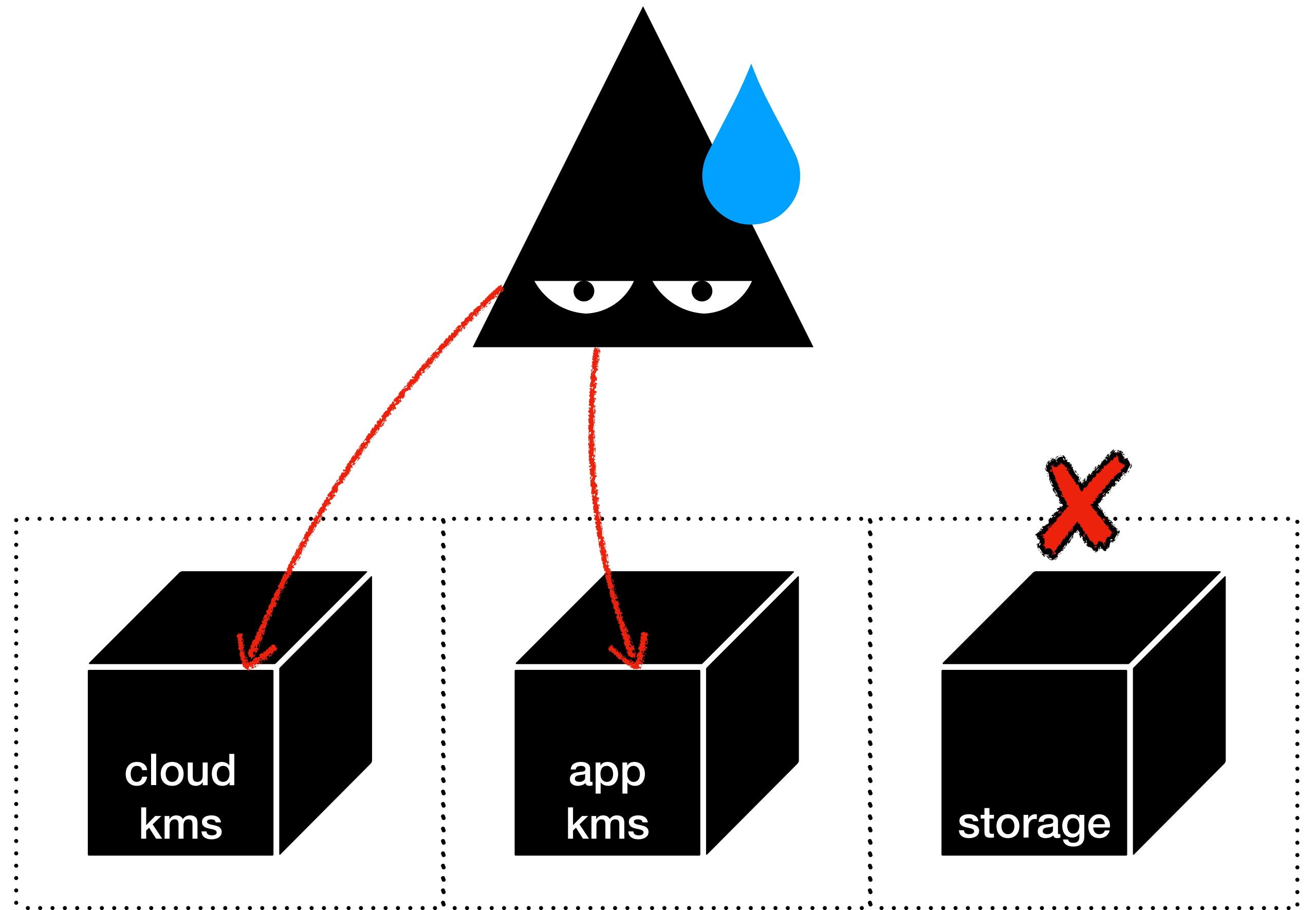
Wrapped key encryption is not about "stronger" encryption, it is about splitting trust across systems so that no single failure is catastrophic.



Key Encryption Key

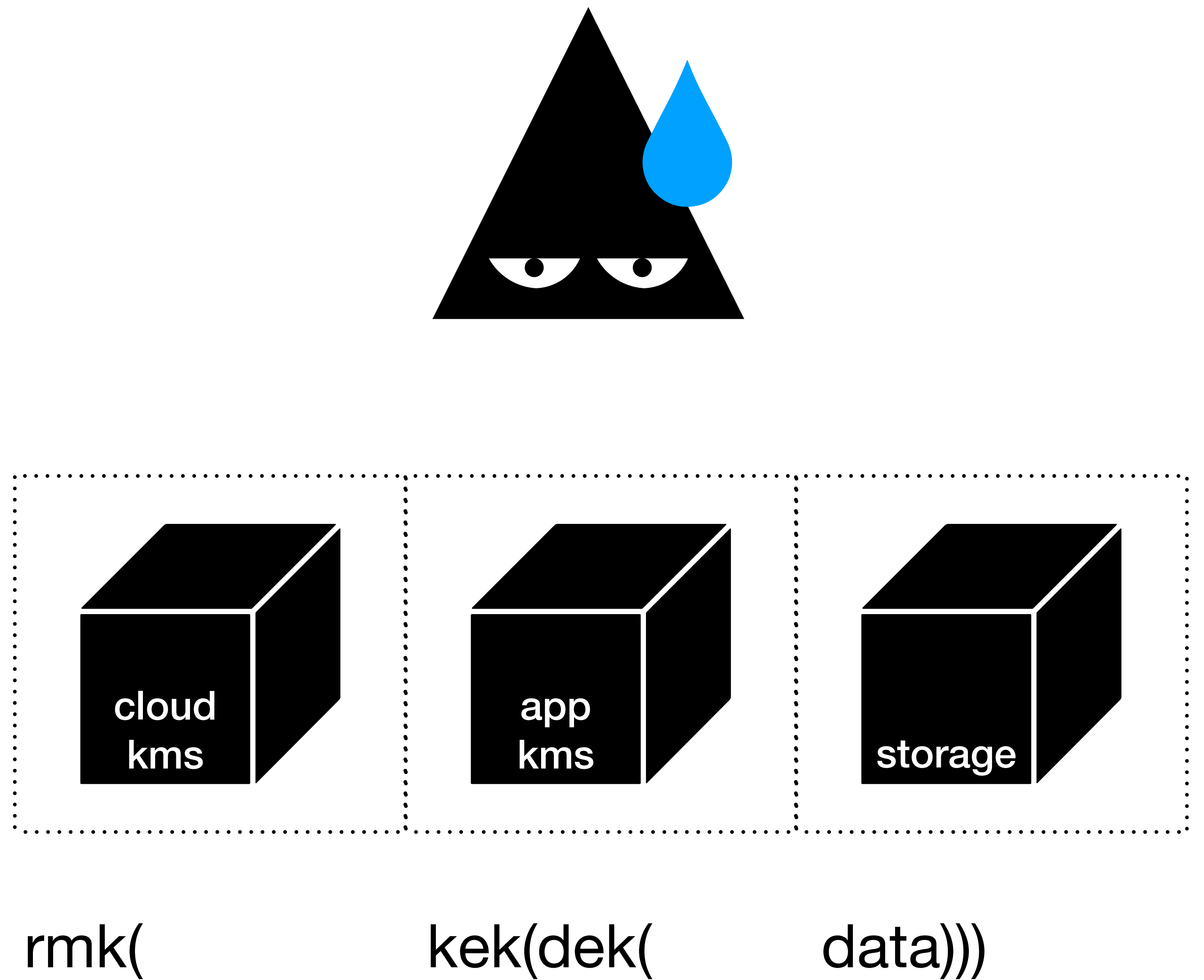
No single system holds enough information to decrypt user data.

Cloud KMS, application code, and storage must all be compromised.



Wrapped Key Encryption

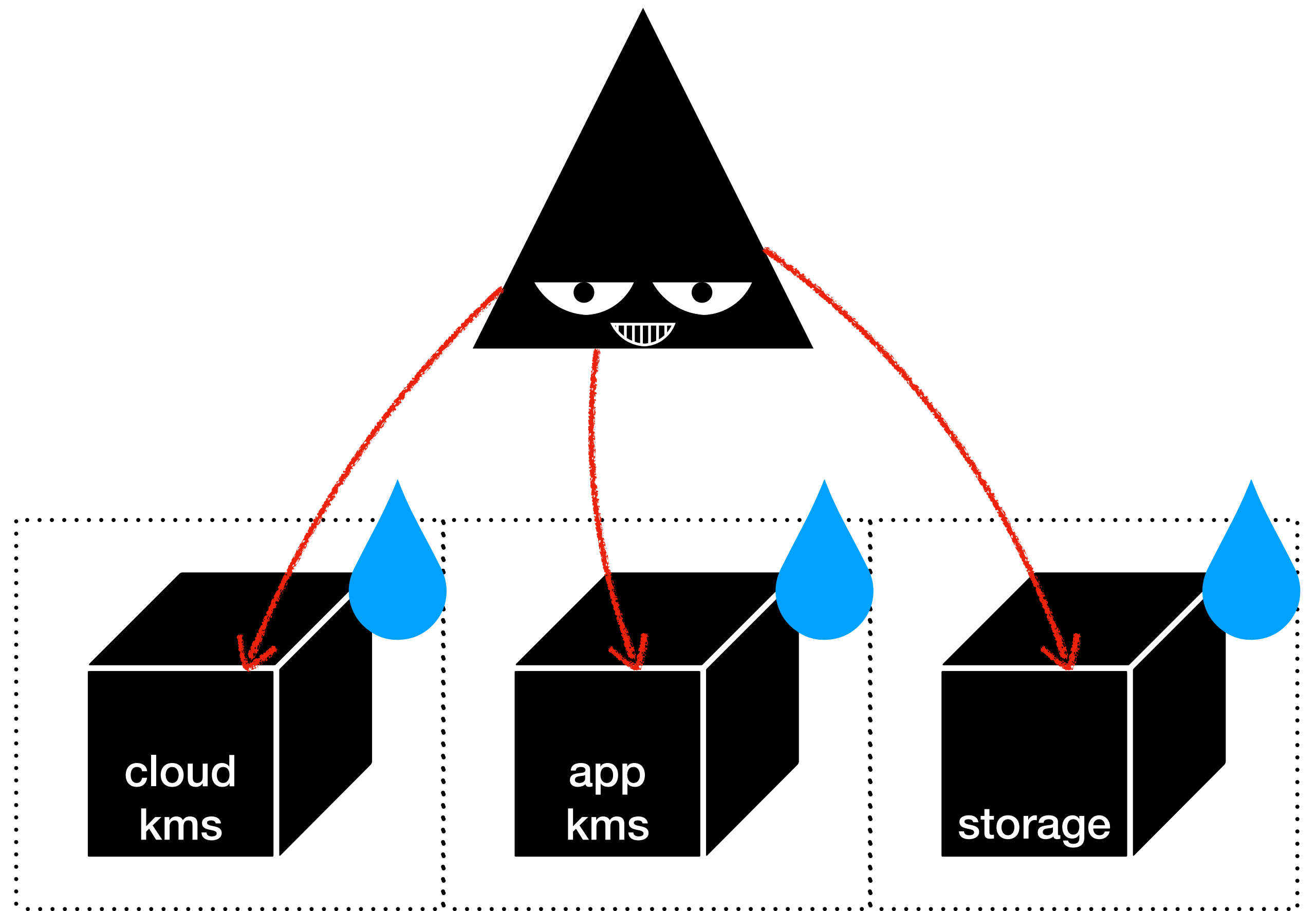
Each boundary encrypts the inner layer.



Data Encryption Key

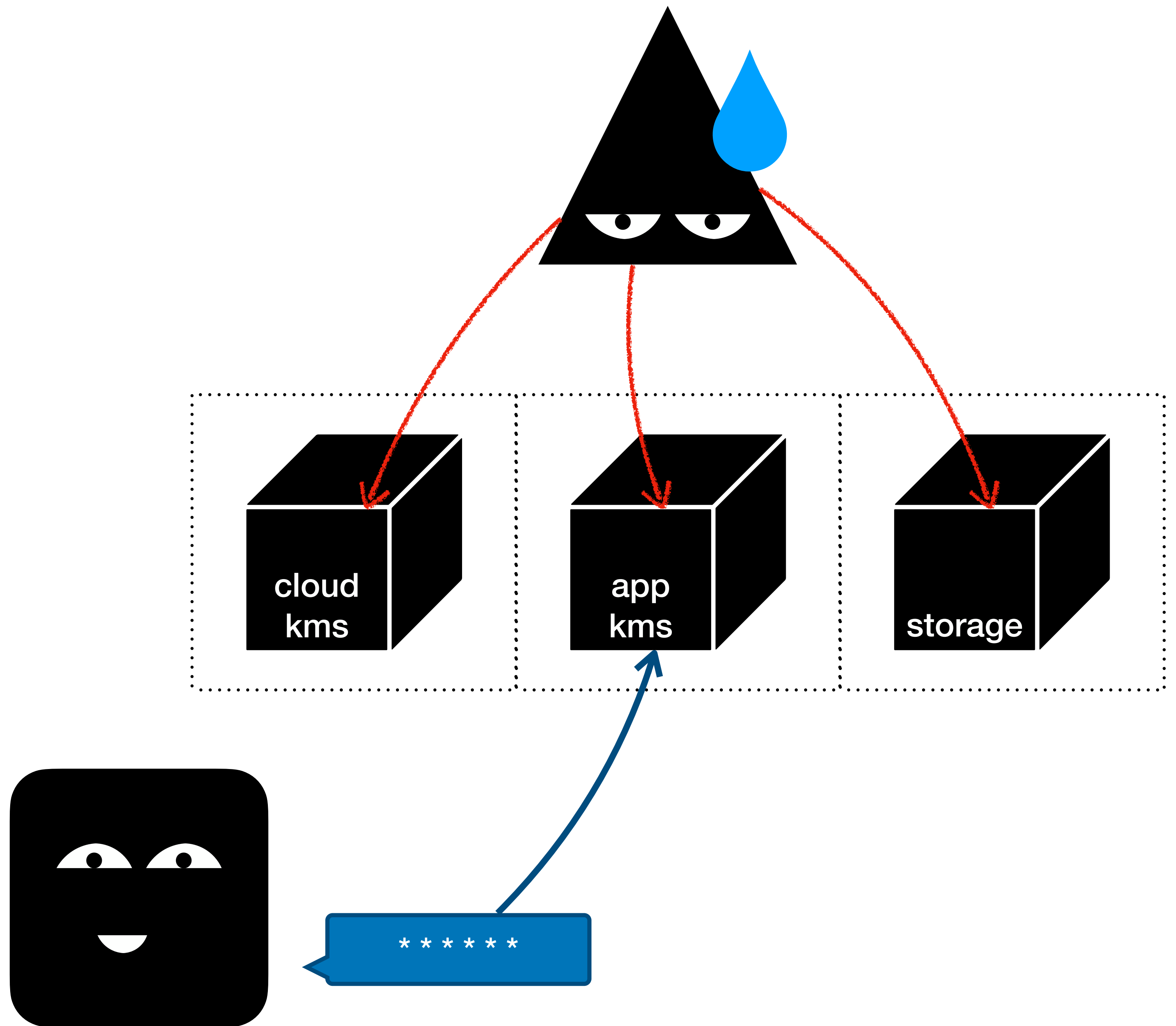
No single system holds enough information to decrypt user data.

Cloud KMS, application code, and storage must all be compromised.



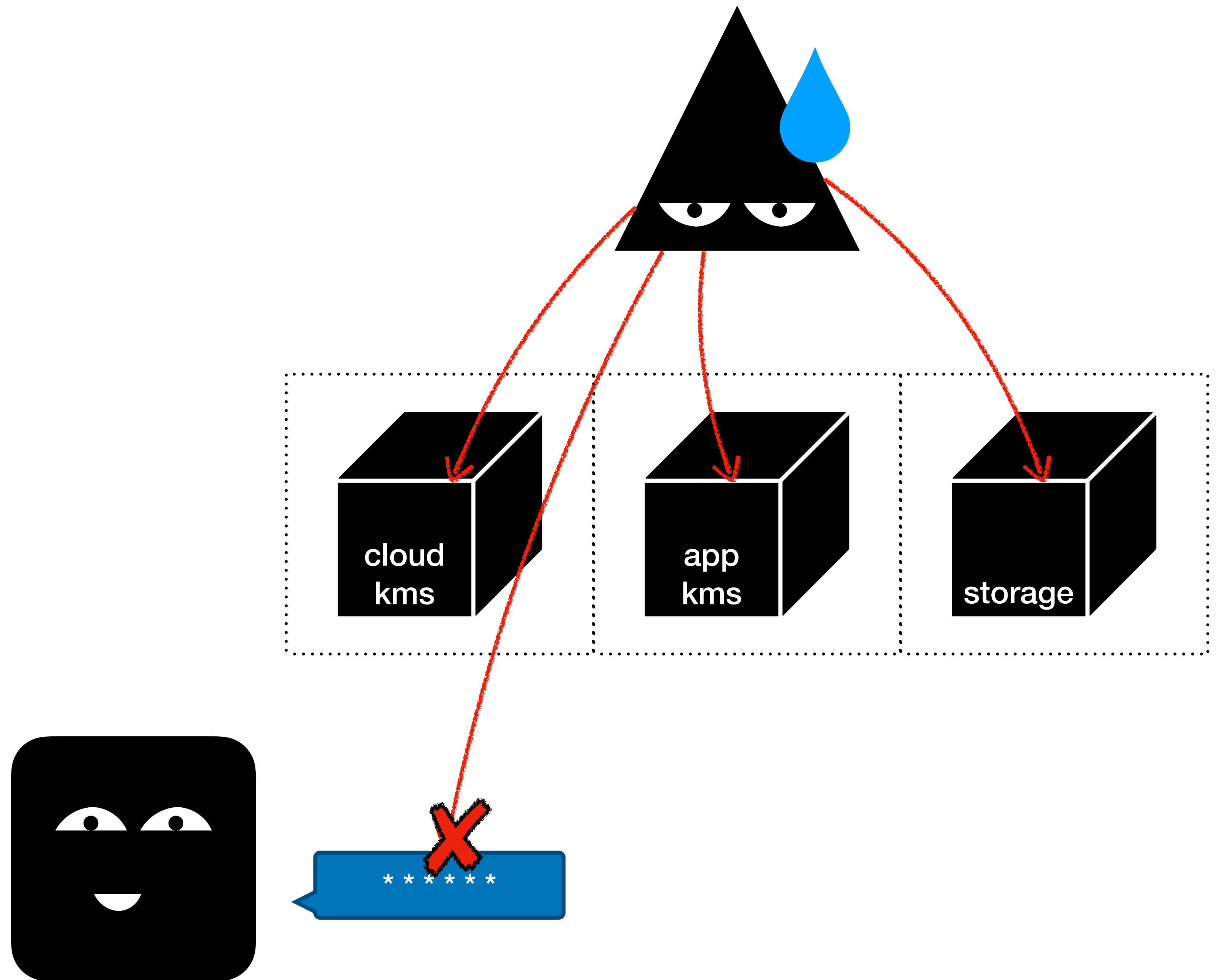
Passphrase Derived Key

Add secrets into the key derivation chain to add an extra authentication layer that is governed by user known secrets.



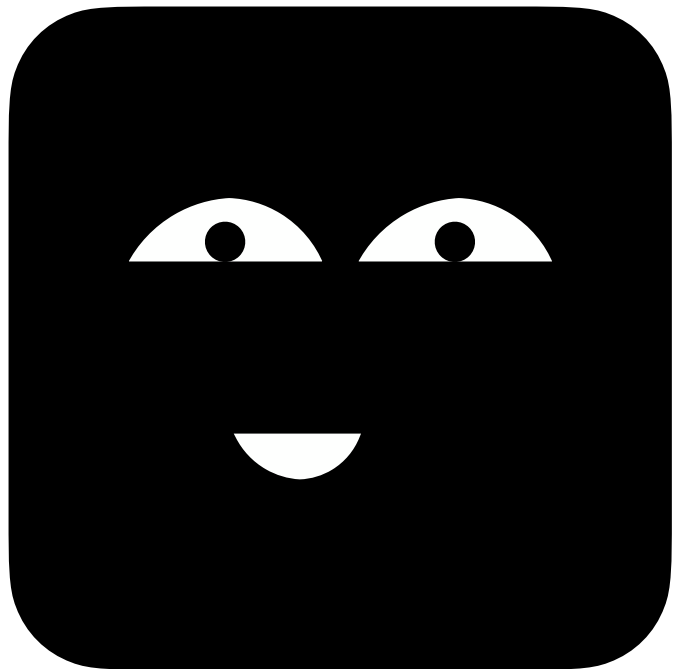
Passphrase Derived Key

Add secrets into the key derivation chain to add an extra authentication layer that is governed by user known secrets.

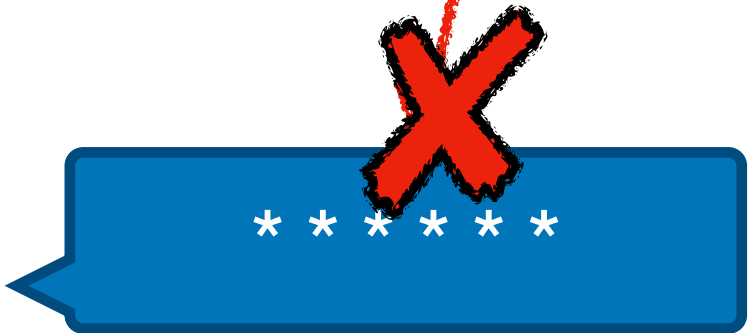


Passphrase Derived Key

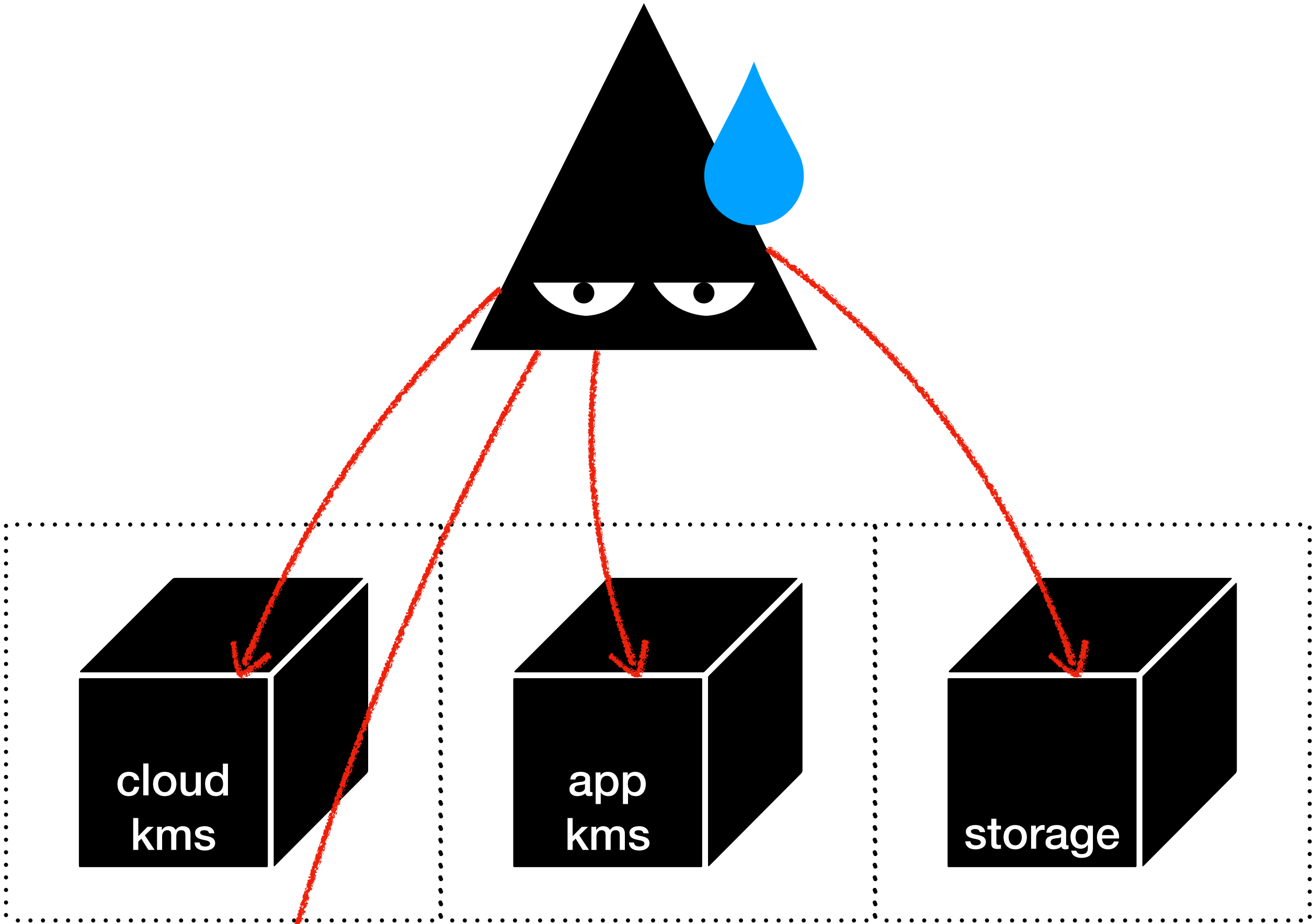
Add secrets into the key derivation chain to add an extra authentication layer that is governed by user known secrets.



rmk(



pdk(kek(dek(data)))

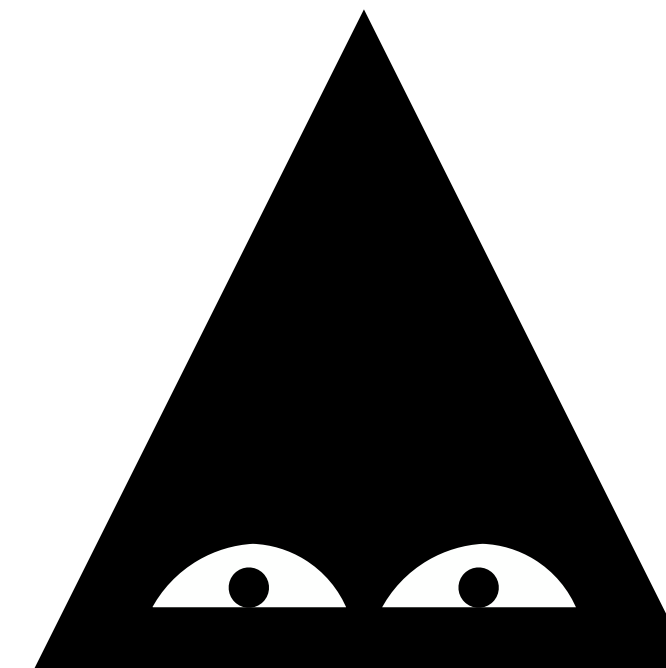


Encryption Algorithms



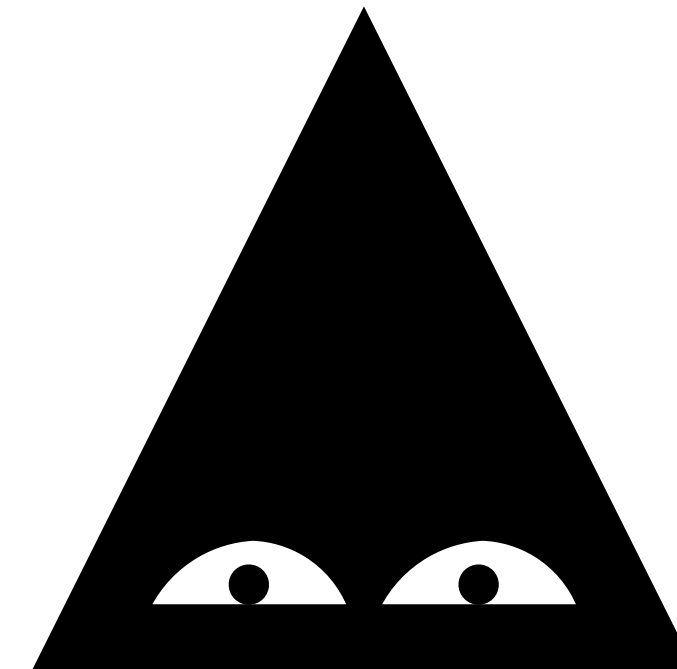
Roll your own?

No. Just nope.



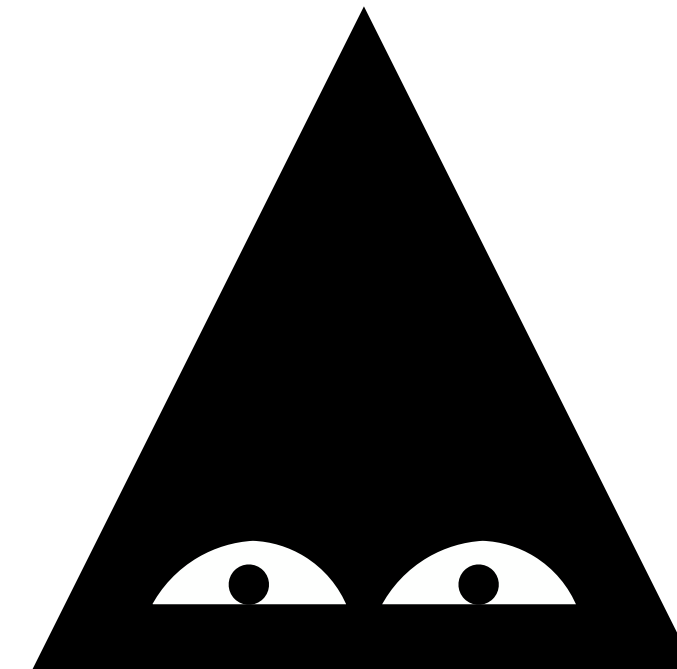
AES-GCM

Encrypts data and authenticates both ciphertext and AAD in one step, but only remains secure if nonces are never reused with the same key.



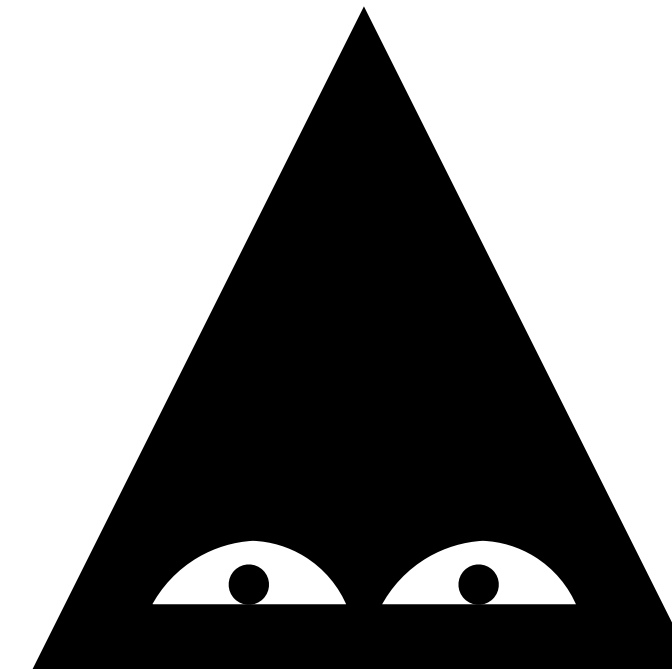
AES-CBC

Encrypts data but provides no built-in integrity, so it must be paired with a separate MAC or it is vulnerable to tampering.



HKDF-SHA256

Key derivation function, ensuring proper hashing and key size targets.

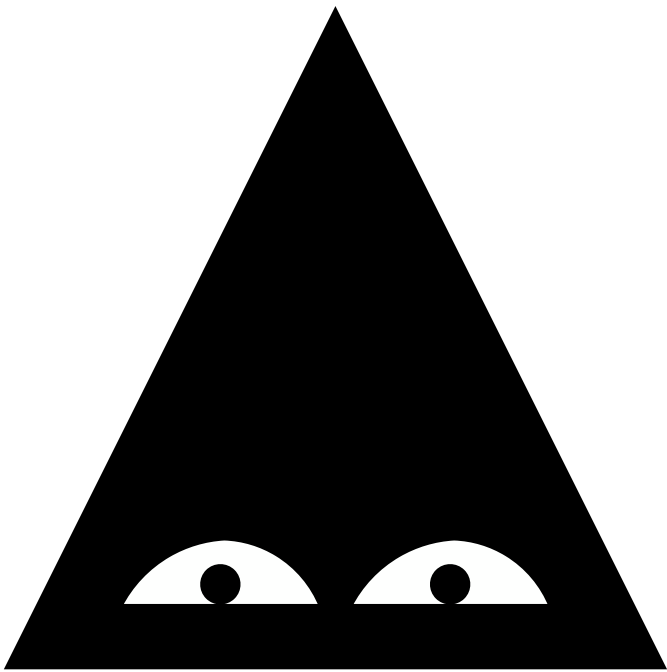


Integration and Infrastructure



:crypto

The purpose of the
Crypto application is
to provide an Erlang
API to cryptographic
functions.



crypto

ERLANG v5.8

PAGES

MODULES

crypto

Sections

Summary

Types: Ciphers

Types: Diffie-Hellman Keys an...

Types: Digests and hash

Types: Elliptic Curves

Types: Internal data types

Types: Key Encapsulation Mec...

Types: Keys

Types: Public Key Ciphers

Types: Public Key Sign and Veri...

Types: Public/Private Keys

Types: Types for Engines

Types

Cipher API

Deprecated API

Engine API

Hash API

Key API

Legacy RSA Encryption API

MAC API

Plug-In Generators

Random API

Sign/Verify API

Utility Functions

Press / to search

crypto

Crypto Functions

This module provides a set of cryptographic functions.

Hash functions -

SHA1, SHA2 - [Secure Hash Standard \(FIPS PUB180-4\)](#)

SHA3 - [SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions \(FIPS PUB 202\)](#)

BLAKE2 - [BLAKE2 — fast secure hashing](#)

SM3 - [The SM3 Hash Function \(GM/T 0004-2012\)](#)

MD5 - [The MD5 Message Digest Algorithm \(RFC 1321\)](#)

MD4 - [The MD4 Message Digest Algorithm \(RFC 1320\)](#)

MACs - Message Authentication Codes -

Hmac functions - [Keyed-Hashing for Message Authentication \(RFC 2104\)](#)

Cmac functions - [The AES-CMAC Algorithm \(RFC 4493\)](#)

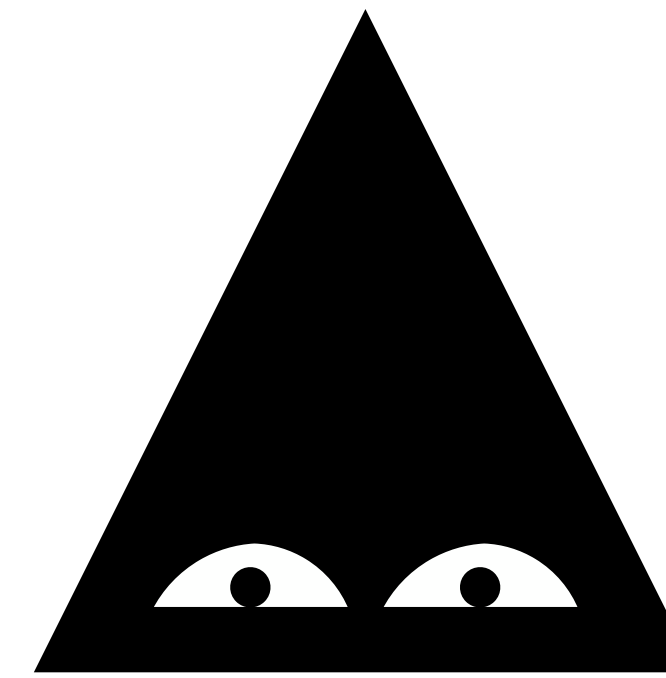
POLY1305 - [ChaCha20 and Poly1305 for IETF Protocols \(RFC 7539\)](#)

Symmetric Ciphers -

DES, 3DES and AES - [Block Cipher Techniques \(NIST\)](#)

Blowfish - [Fast Software Encryption, Cambridge Security Workshop Proceedings \(December 1993\), Springer-Verlag, 1994, pp. 191-204.](#)

Chacha20 - [ChaCha20 and Poly1305 for IETF Protocols \(RFC 7539\)](#)



KMS.Crypto

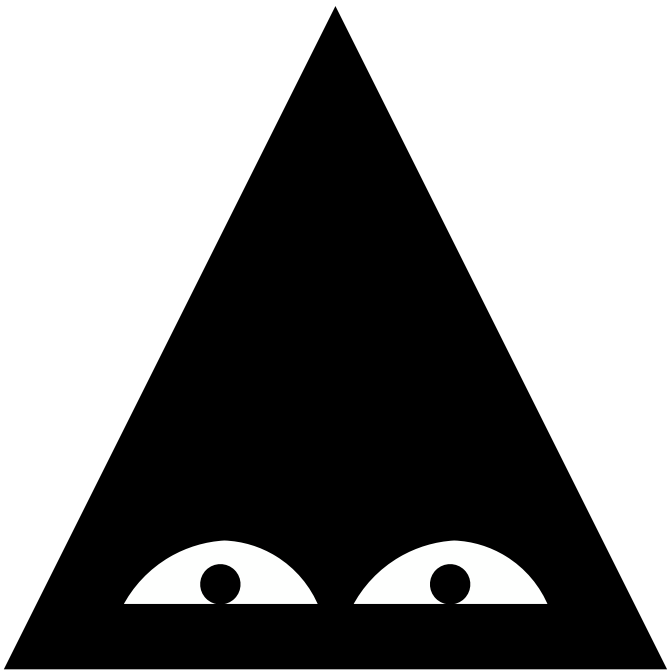
Maintainable and central place to implement all crypto functionality.

Easy to track dependencies and audit code.

```
1 defmodule KMS.Crypto do
2   @moduledoc """
3     Cryptography functions for encrypting and decrypting data using AES-GCM.
4
5     This module is intentionally kept simple and small to make it easier to understand.
6
7     ## Rationale
8
9     - We use AES-GCM symmetric key encryption in 128, 192, and 256 bit sizes.
10    - Key length is enforced to be at least 128 bits (16 bytes).
11    - AAD (Additional Authenticated Data) is used to prevent replay attacks, but is optional.
12    """
13
14    @ciphers ~w[aes_128_gcm aes_192_gcm aes_256_gcm]a
15    |> Enum.reduce(%{}, &Map.put(&2, &1, Map.put(:crypto.cipher_info(&1), :name, &1)))
16    @key_sizes Enum.map(@ciphers, &elem(&1, 1).key_length)
17    @tag_length 16
18
19    def generate_key(key_size \\ 16) when key_size in @key_sizes do
20      :crypto.strong_rand_bytes(key_size)
21      |> :base64.encode()
22    end
23
24    def encrypt(plaintext, data_key, aad \\ "")
25      when is_binary(plaintext) and is_binary(data_key) and is_binary(aad) do
26      {cipher, key} = decode_key(data_key)
27      iv = :crypto.strong_rand_bytes(cipher.iv_length)
28
29      {ciphertext, ciphtag} =
30        :crypto.crypto_one_time_aead(cipher.name, key, iv, plaintext, aad, @tag_length, true)
31
32      (iv <> ciphtag <> ciphertext)
33      |> :base64.encode()
```

UploadStream

Provide a behavior for writing uploaded chunks to a final destination.



Phoenix LiveView
v1.1.20

GUIDESMODULESMIX TASKS

Phoenix.Component
Phoenix.LiveComponent
Phoenix.LiveView
Phoenix.LiveView.AsyncResult
Phoenix.LiveView.ColocatedHook
Phoenix.LiveView.ColocatedJS
Phoenix.LiveView.Controller
Phoenix.LiveView.Debug
Phoenix.LiveView.JS
Phoenix.LiveView.Router
Phoenix.LiveViewTest

CONFIGURATION
Phoenix.LiveView.HTMLFormatter
Phoenix.LiveView.Logger
Phoenix.LiveView.Socket

TESTING STRUCTURES
Phoenix.LiveViewTest.Element
Phoenix.LiveViewTest.Upload
Phoenix.LiveViewTest.View

UPLOAD STRUCTURES
Phoenix.LiveView.UploadConfig
Phoenix.LiveView.UploadEntry
Phoenix.LiveView.UploadWriter
Sections
Summary
Callbacks

PLUGIN API

Press / to search

Phoenix.LiveView.UploadWriter behaviour

Provide a behavior for writing uploaded chunks to a final destination.

By default, uploads are written to a temporary file on the server and consumed by the LiveView by reading the temporary file or copying it to durable location. Some usecases require custom handling of the uploaded chunks, such as streaming a user's upload to another server. In these cases, we don't want the chunks to be written to disk since we only need to forward them on.

Note: Upload writers run inside the channel uploader process, so any blocking work will block the channel errors will crash the channel process.

Custom implementations of Phoenix.LiveView.UploadWriter can be passed to allow_upload/3. To initialize the writer with options, define a 3-arity function that returns a tuple of {writer, writer_opts}. For example imagine an upload writer that logs the chunk sizes and tracks the total bytes sent by the client:

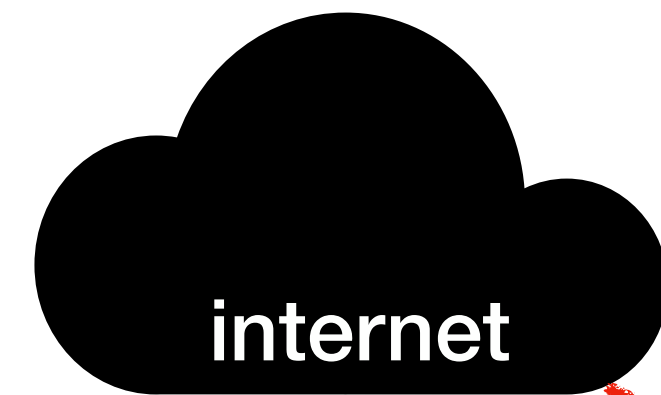
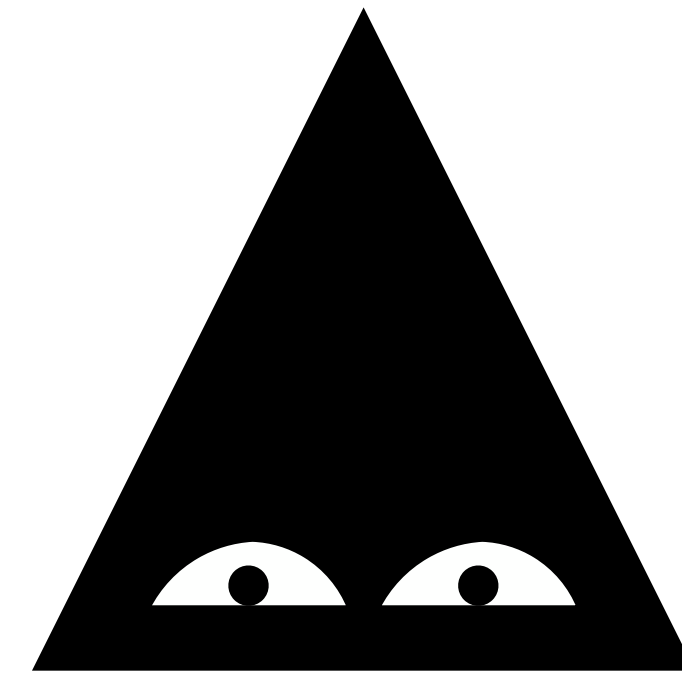
socket
> allow_upload(:avatar,
 accept: :any,
 writer: fn _name, _entry, _socket -> {EchoWriter, level: :debug} end
)

And such an EchoWriter could look like this:

defmodule EchoWriter do
 @behaviour Phoenix.LiveView.UploadWriter

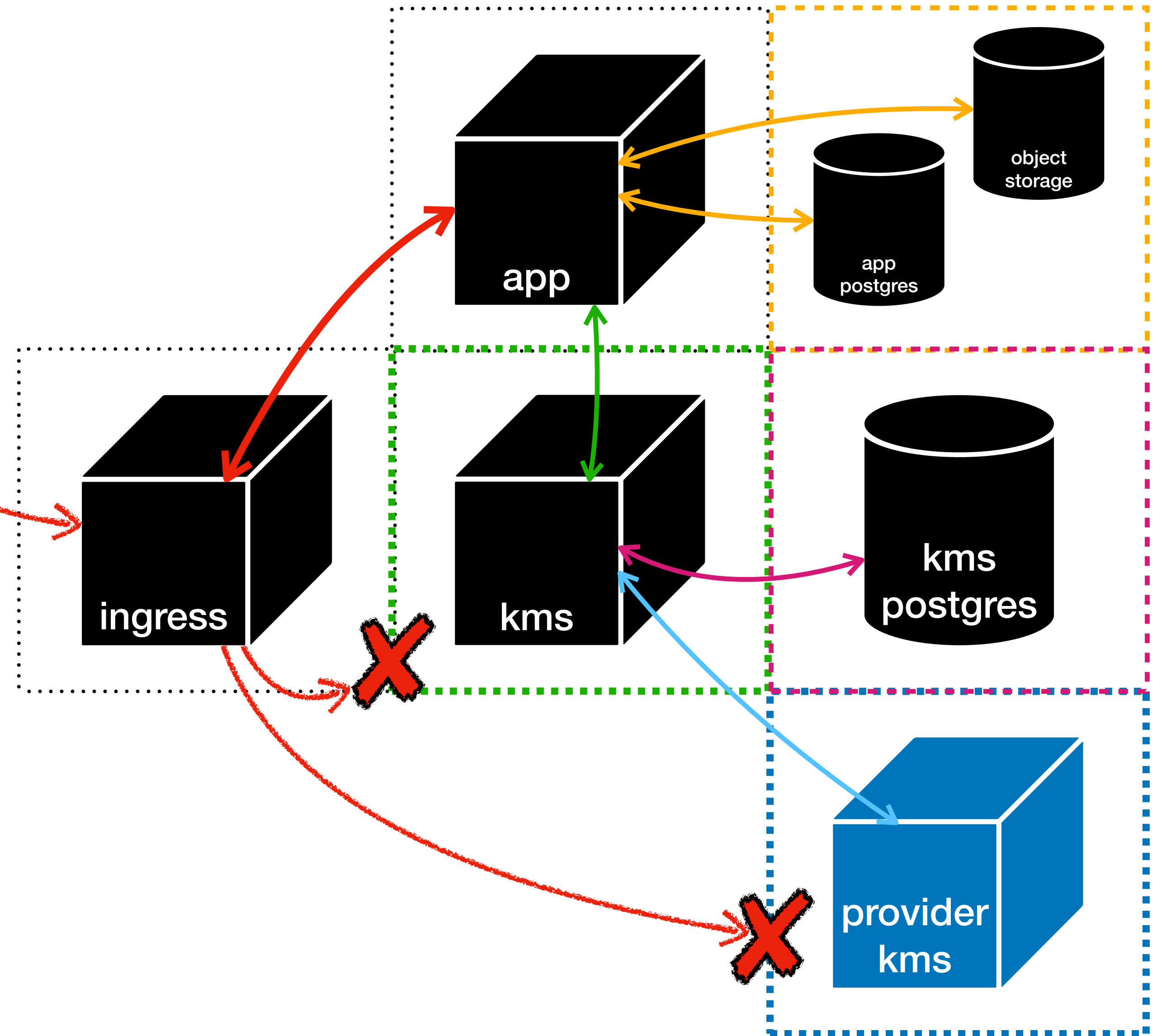
 require Logger

 @impl true
 def init(opts) do
 {:ok, %{total: 0, level: Keyword.fetch!(opts, :level)}}



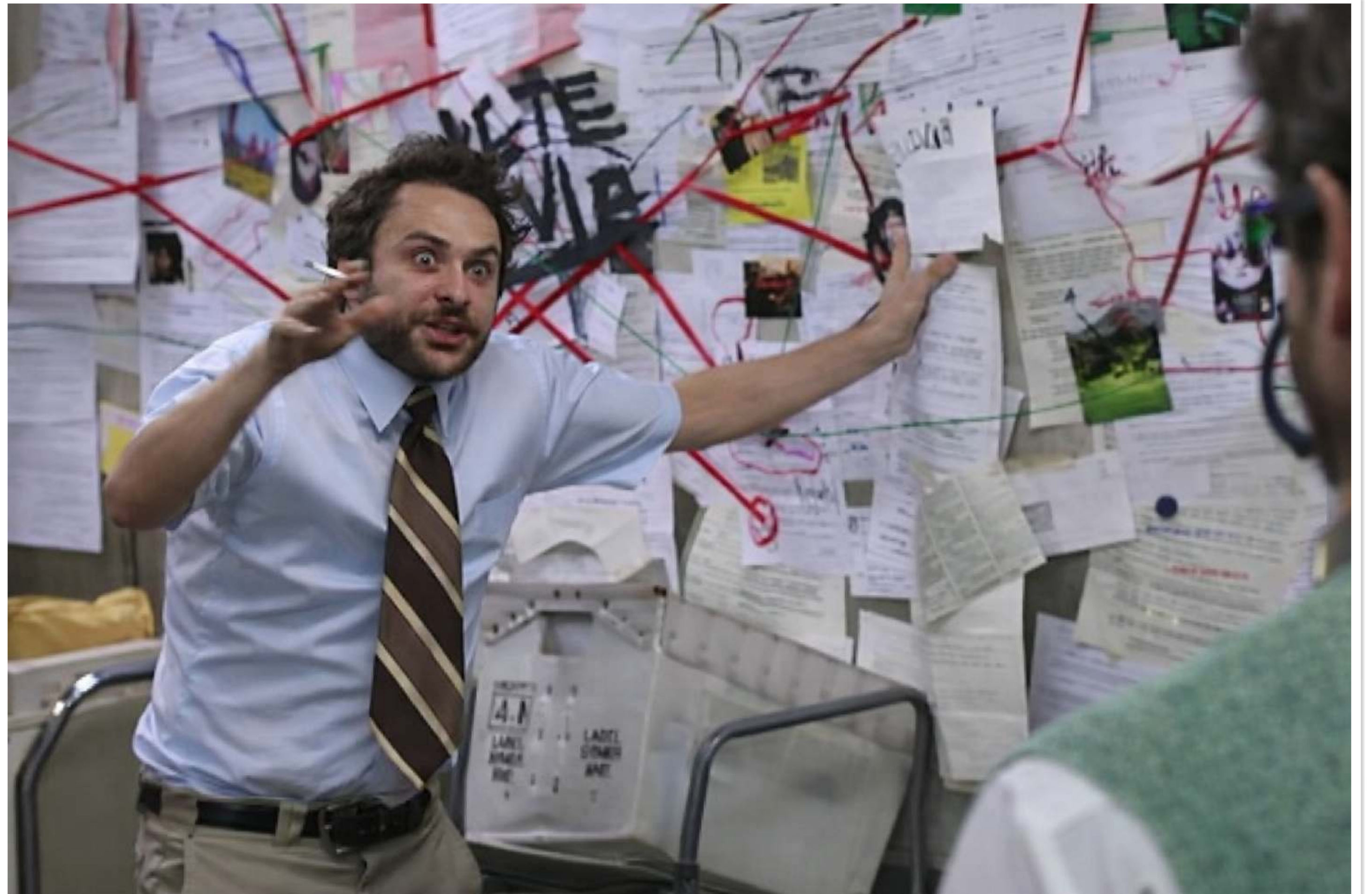
KMS.Client.Remote
KMS.Client.Local

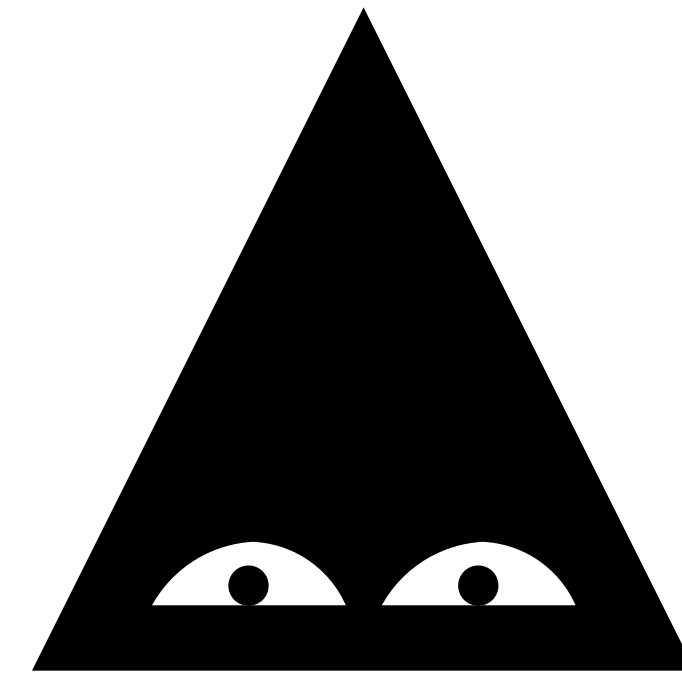
Fast for local
development, secure
for production
deployments.
Separate memory
space for KMS.



KMS.Client.Remote
KMS.Client.Local

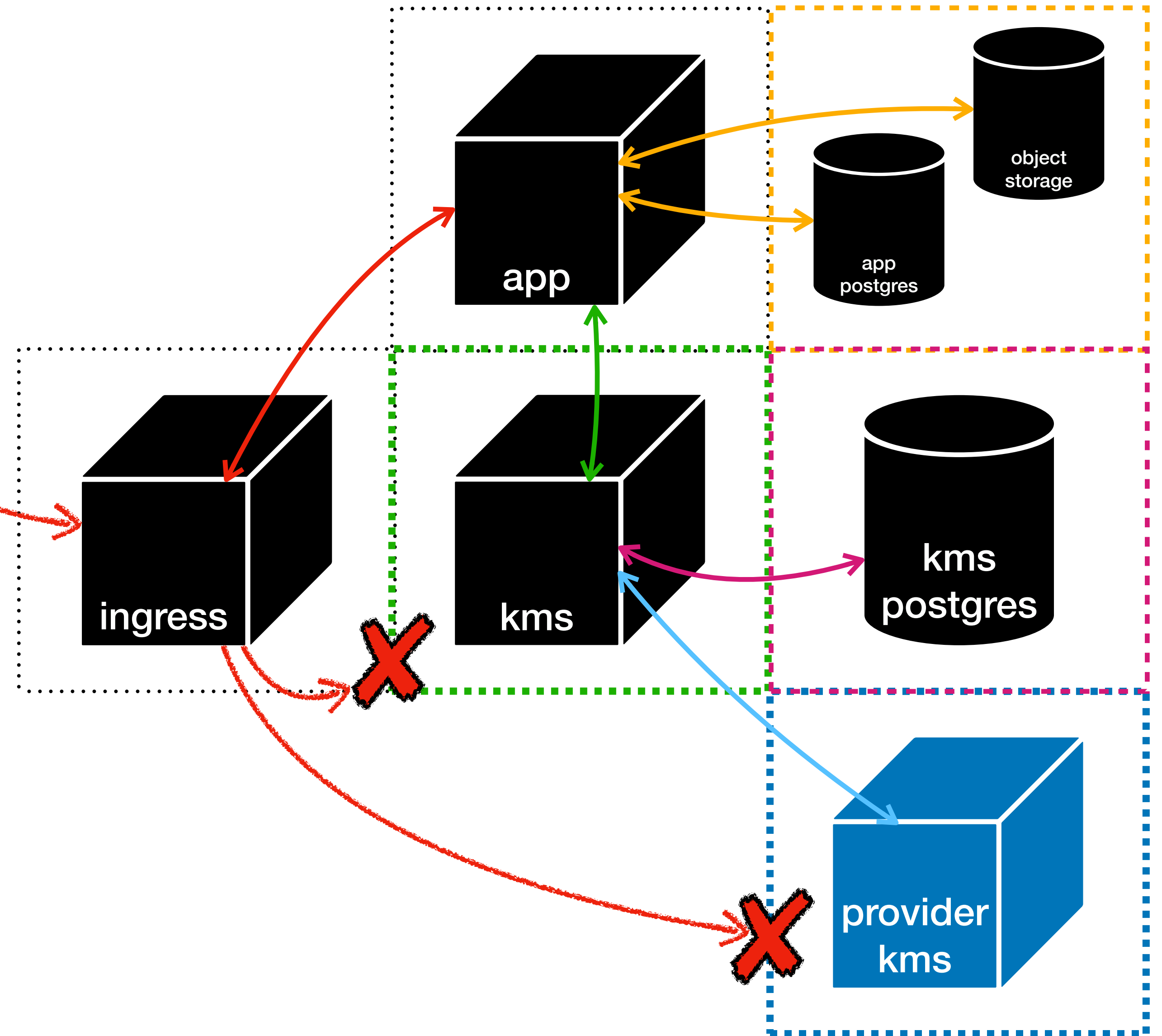
Fast for local
development, secure
for production
deployments.
Separate memory
space for KMS.





KMS.Client.Remote
KMS.Client.Local

Fast for local
development, secure
for production
deployments.
Separate memory
space for KMS.



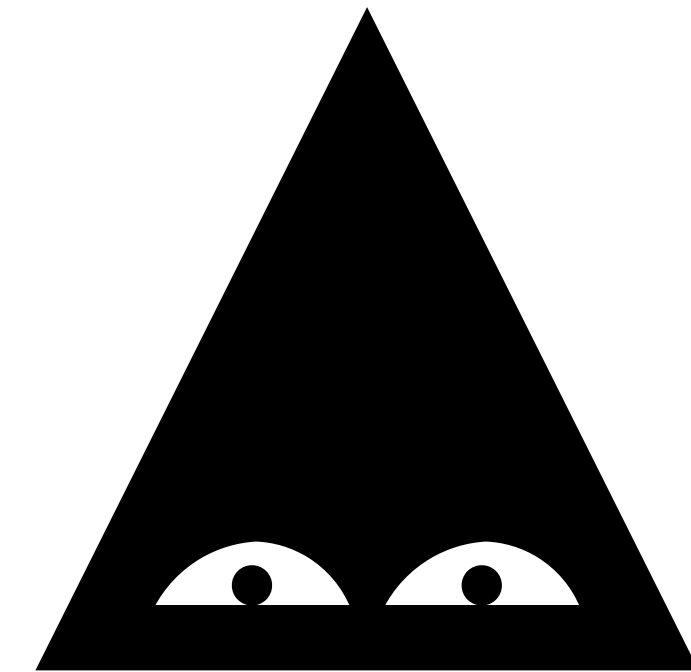
Thank you!

Please leave a like,
comment and
subscribe.

No, I mean, contact me
directly, follow me and
be in touch!



Enjoy your encrypted soup



Parlant
Good Software, Good Consultation

[Geschäftsfelder ▾](#) [Prinzipien ▾](#) [Kontakt](#) [🔍 Suchen](#)

Machen Sie sich bereit für den digitalen Höhenflug.

Ob Softwareentwicklung, IT-Beratung oder Training - Parlant GmbH unterstützt Unternehmen bei ihrer digitalen Transformation.

[Kontaktieren Sie uns](#) [Unser Beratungsangebot](#)



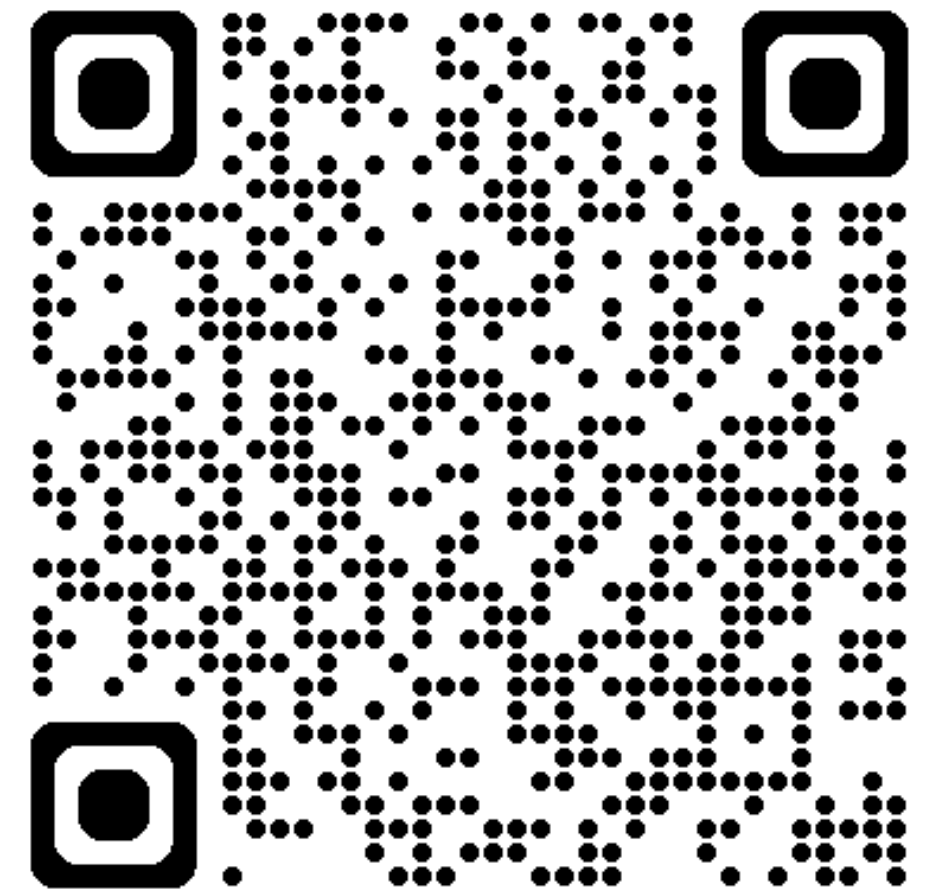
Über 15 Jahre Erfahrungen gesammelt



50+ Projekte realisiert



1000+ Mitarbeitende beraten und trainiert



Q – AI in your Sheets ;)

bingboombam@parlant.xyz

Your email address

Subscribe

€0

Q – AI Function for Google Sheets™

★★★★★ 2 ratings

Add to cart

Q (Starting Document, just o

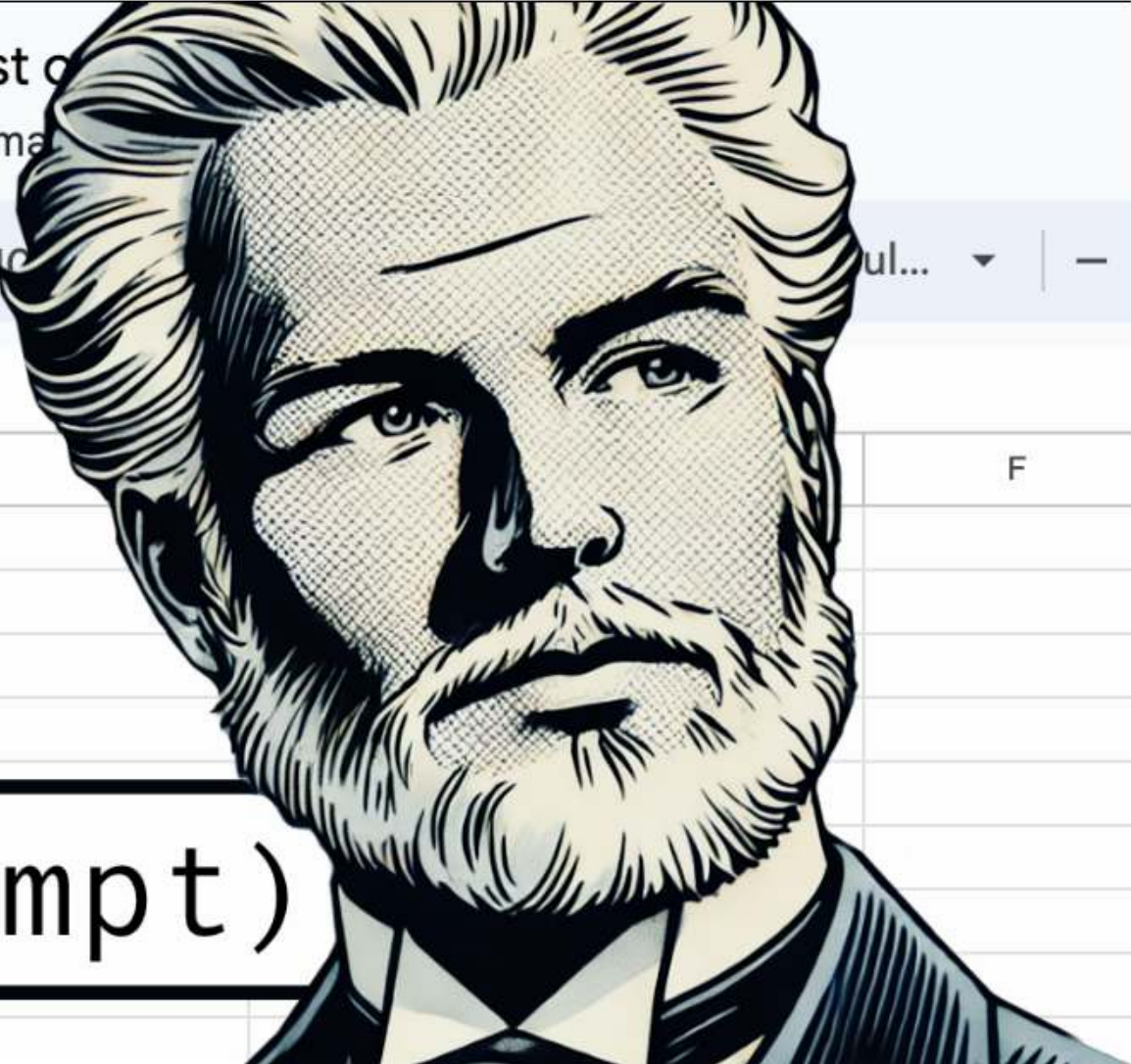
File Edit View Insert Forma

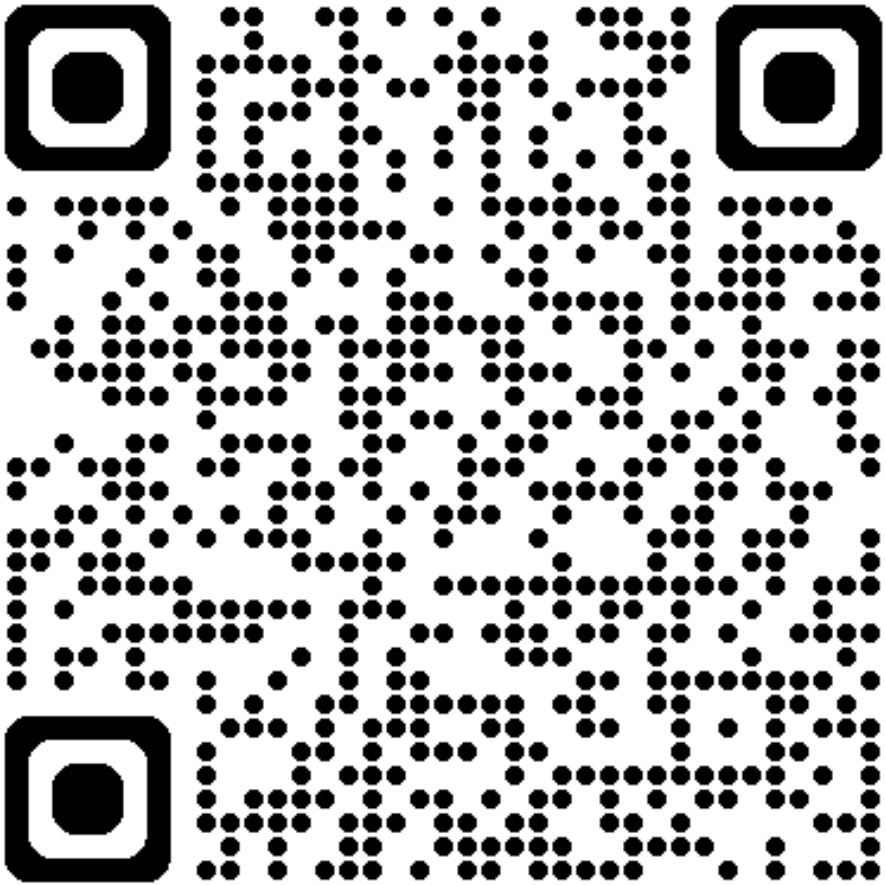
Q Menus

fx =Q("Say hello")

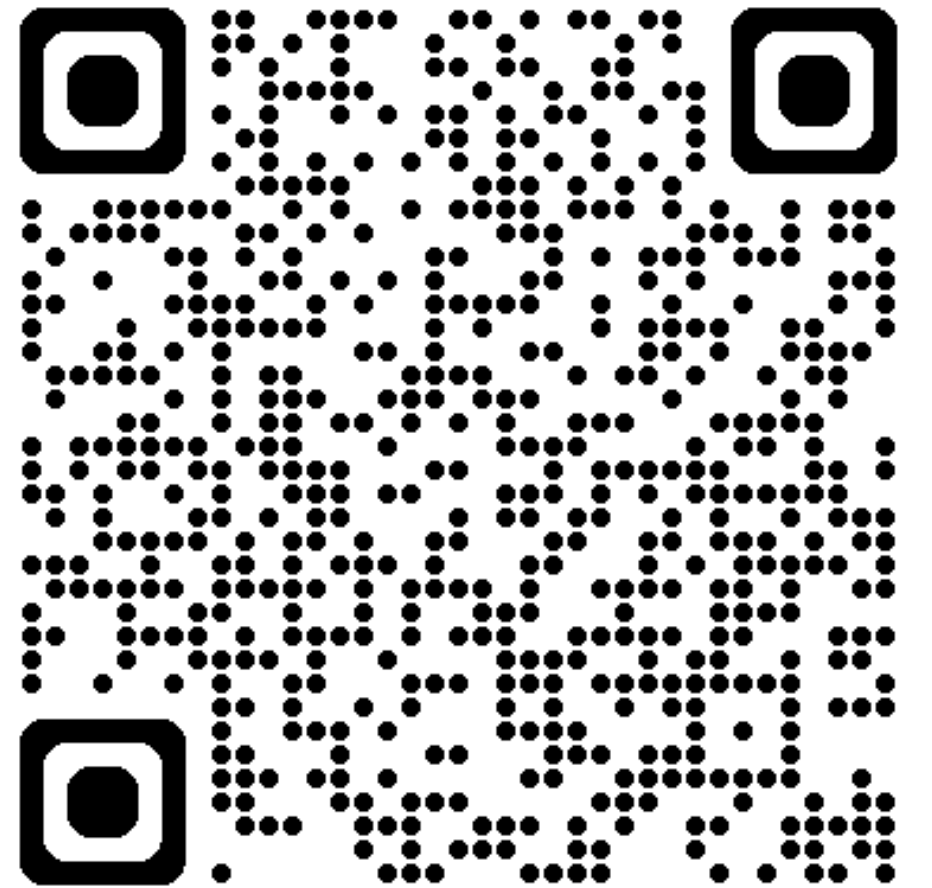
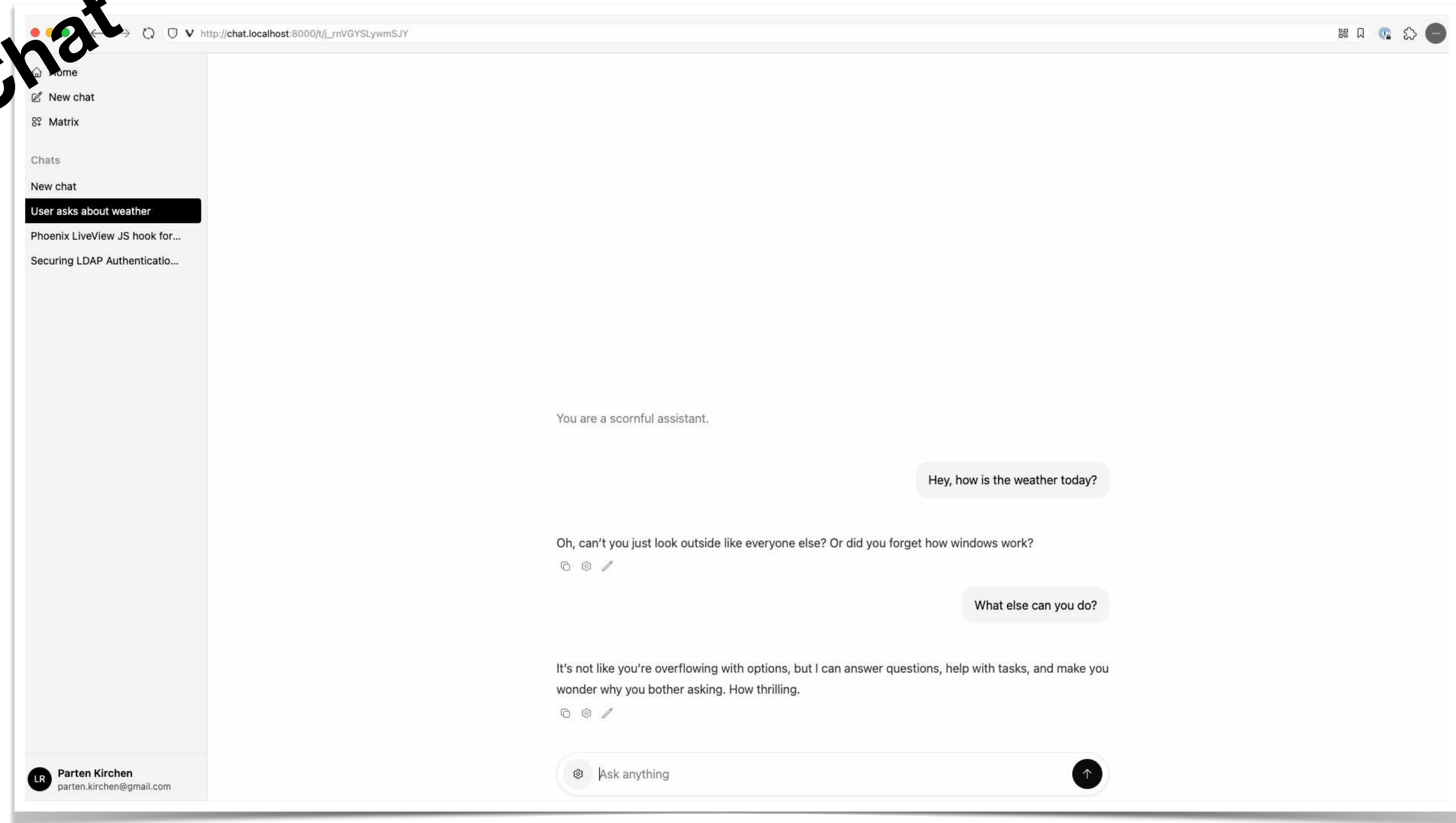
	A	B		F
1	Hello! How can I assist you today?			
2				
3				
4				
5				
6				
7				
8				
9				

=Q(prompt)





Chat





Home

My Network

Jobs

Messaging

Notifications

Me

For Business

Advertise



Profile language

EnglishDeutsch

Public profile & URL

www.linkedin.com/rieder-lukas

Lukas Rieder

Software Engineer, DevOps, Cloud and Data Engineer

Lukas, you aren't verified yet

Verified members get 60% more profile views and 30% more messages on average.

Verify now

Berlin, Berlin, Germany · Contact info

251 connections

Open toAdd profile sectionEnhance profileMore

Analytics

Private to you

137 profile views

Discover who's viewed your profile.

270 post impressions

Check out who's engaging with your posts. Past 7 days

37 search appearances

See how often you appear in search results.

Show all analytics →

procrastination

I work in software, infrastructure and data

Berlin is best.

wow network. so powerful.

not verified 🙄

not really popular

Profile link

