Lukas'

Privacy by Design

Privacy by Design

Lukas Rieder, March 5th 2026

Privacy by Design

Today

Lukas Rieder, March 5th 2026

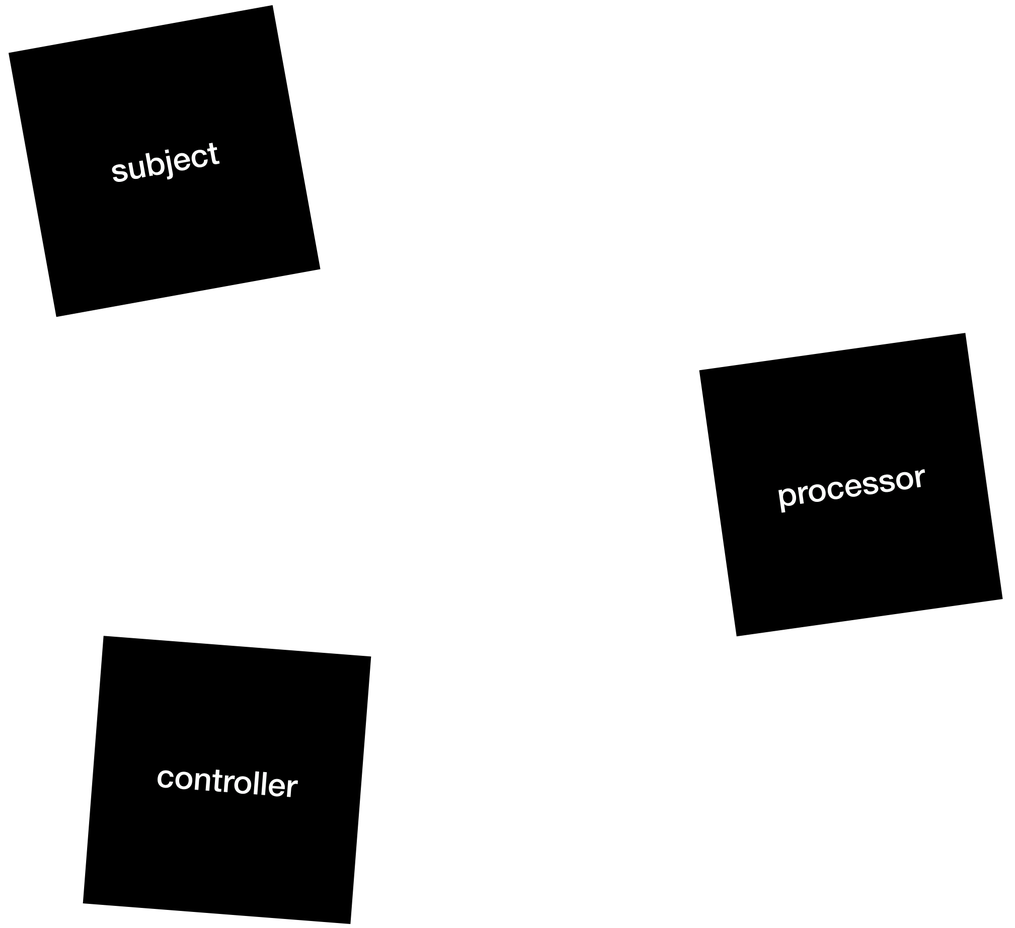
Talking about **Data Privacy**

Talking about **Elixir & Erlang**

Talking about **Encryption** *in practice*

Data Privacy

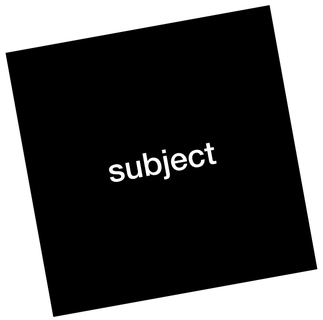
GDPR generally divides rights and responsibilities into three roles.



Data Subject

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

Right to information, correction and deletion of their data.



Data Controller

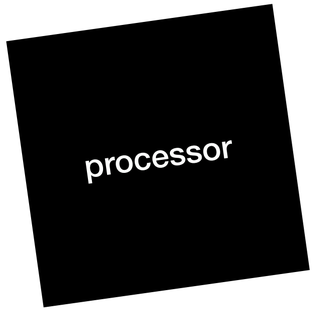
e.g. your company that collects personal data of its customers.

Obligation to ensure lawful processing of the data.

Data Processor

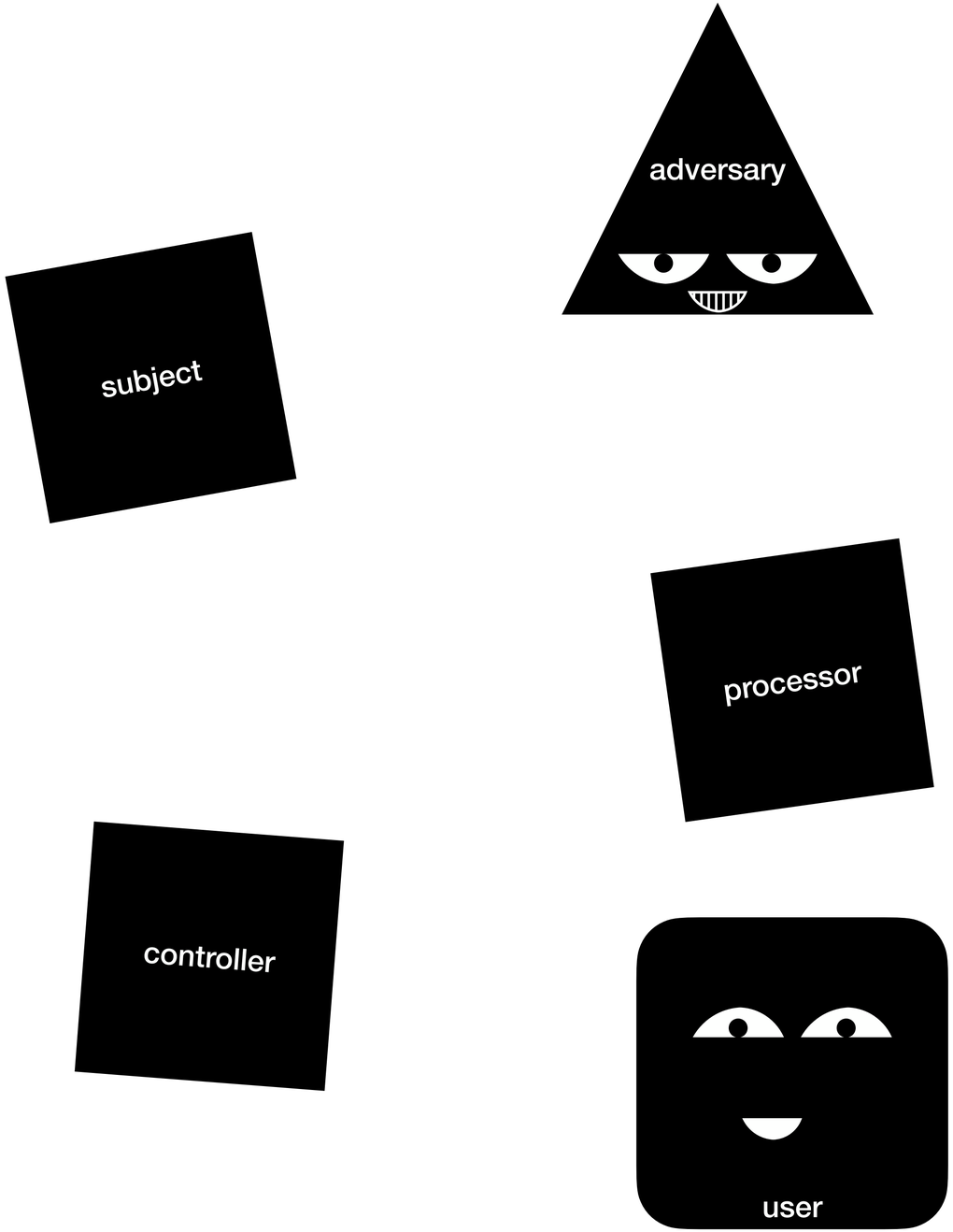
e.g. your hosting provider that stores and processes data.

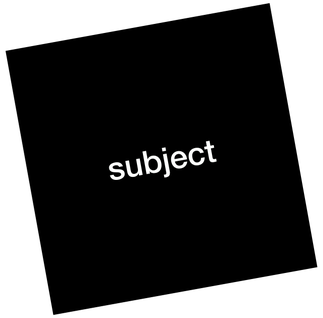
Obligation to comply and protect the data according to the controllers instruction.



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

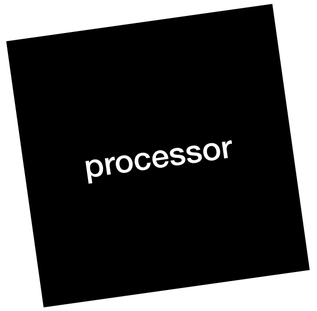
The subject trusts that their data is handled according to declared purpose, not exposed by design flaws or operational failures.

Data Security

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

Data Security

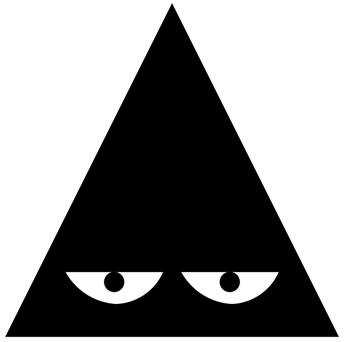
The processor is responsible for how data is protected in practice: **encryption**, **access control**, **isolation**, and **secure storage**.



Data Security

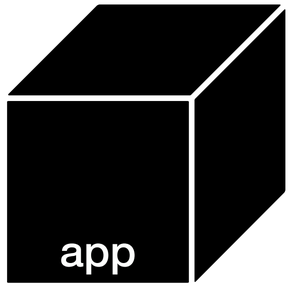
The users have rights to their data, but we are not responsible for their own systems and behaviors.

Encryption, Access Control, Isolation, and Secure Storage



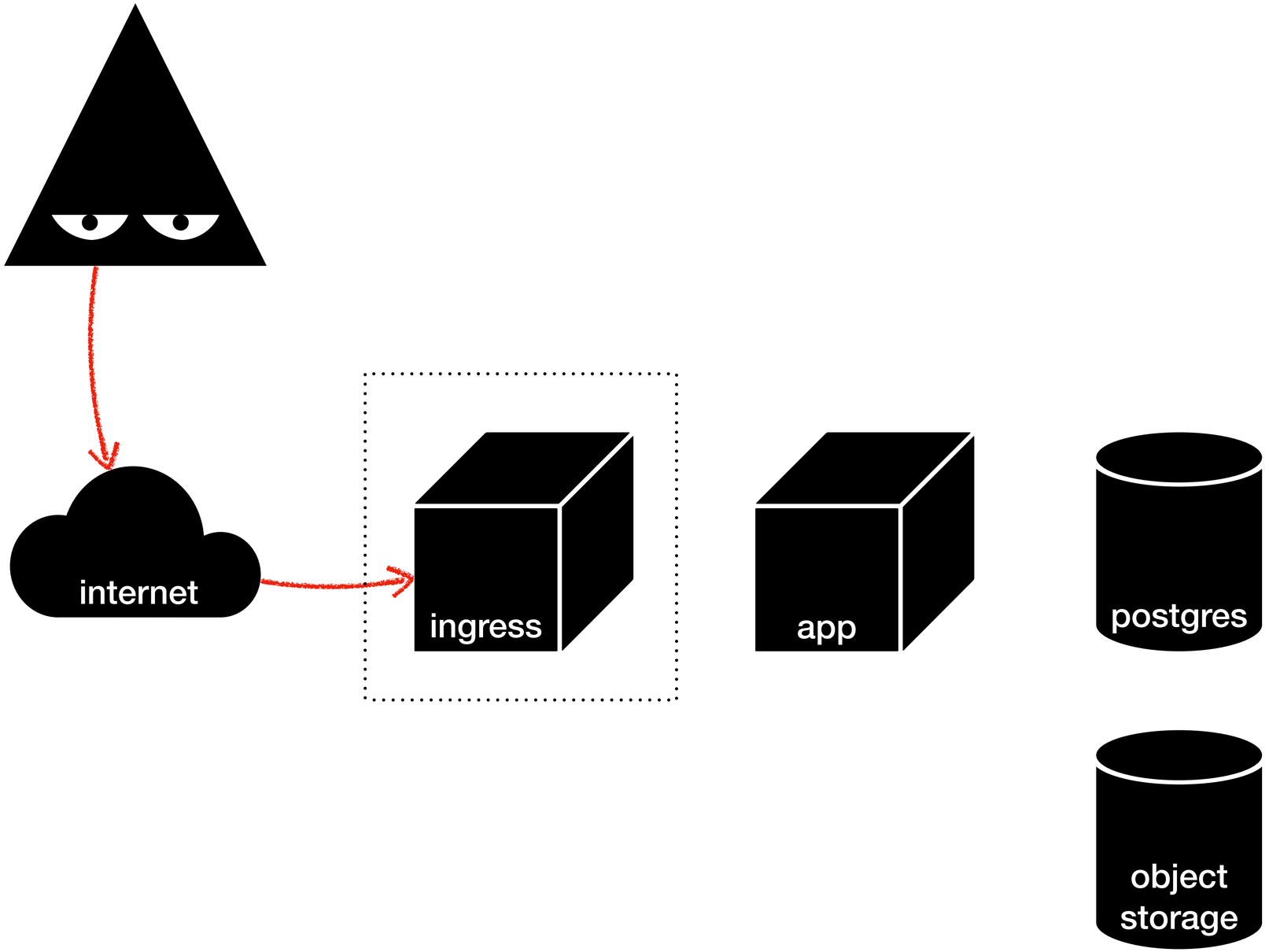
Data Security

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



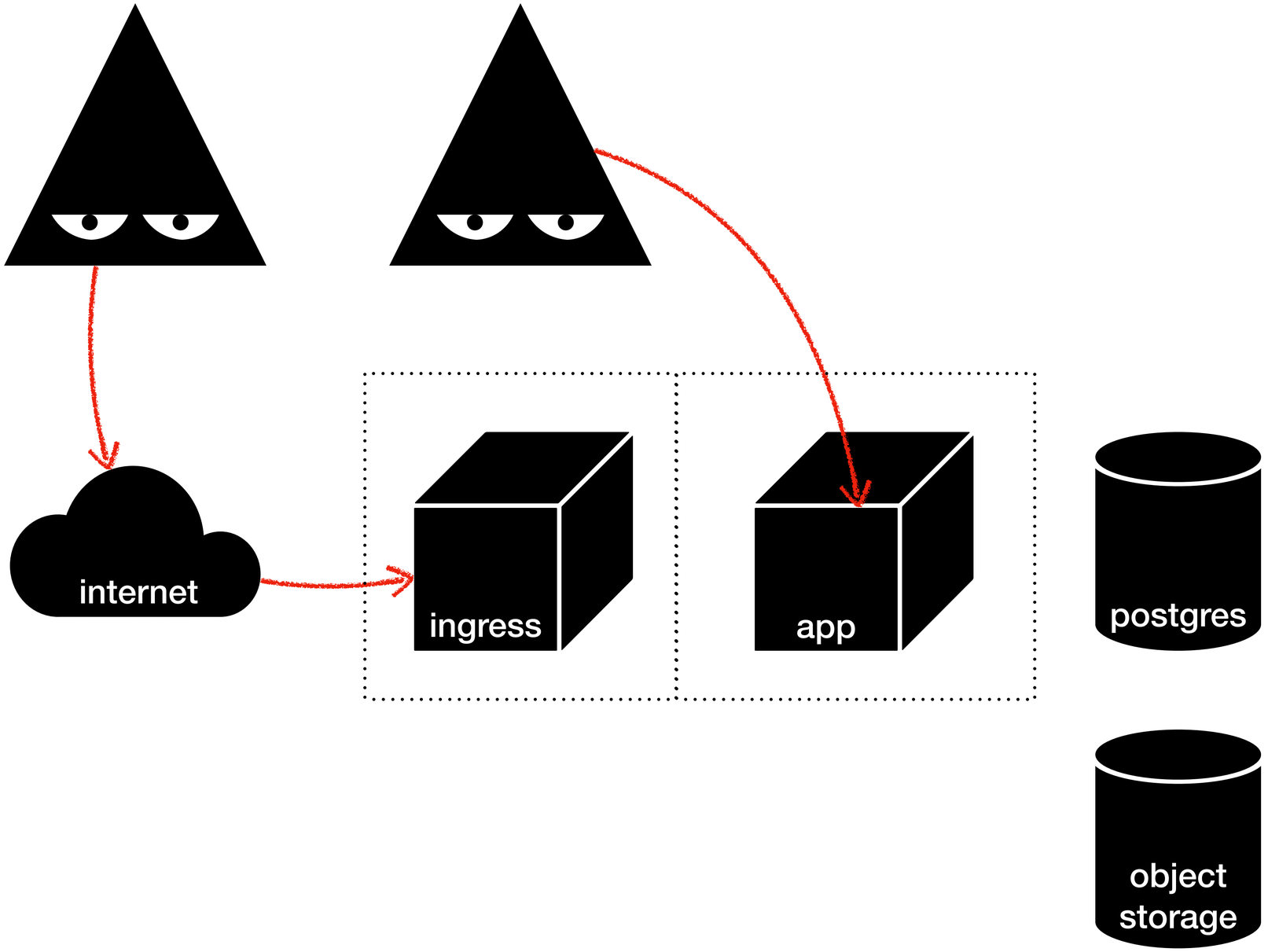
Data Security

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



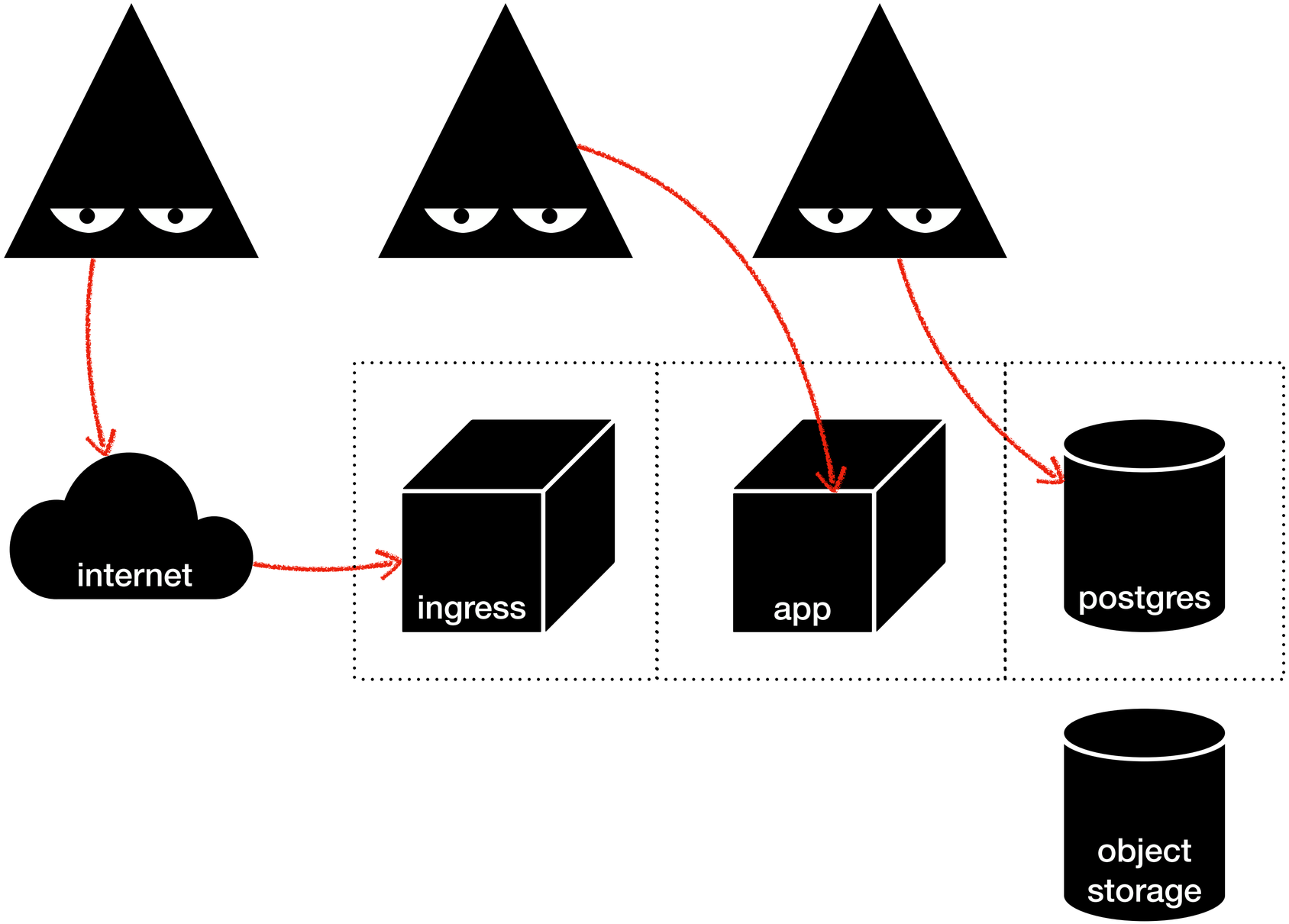
Data Security

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



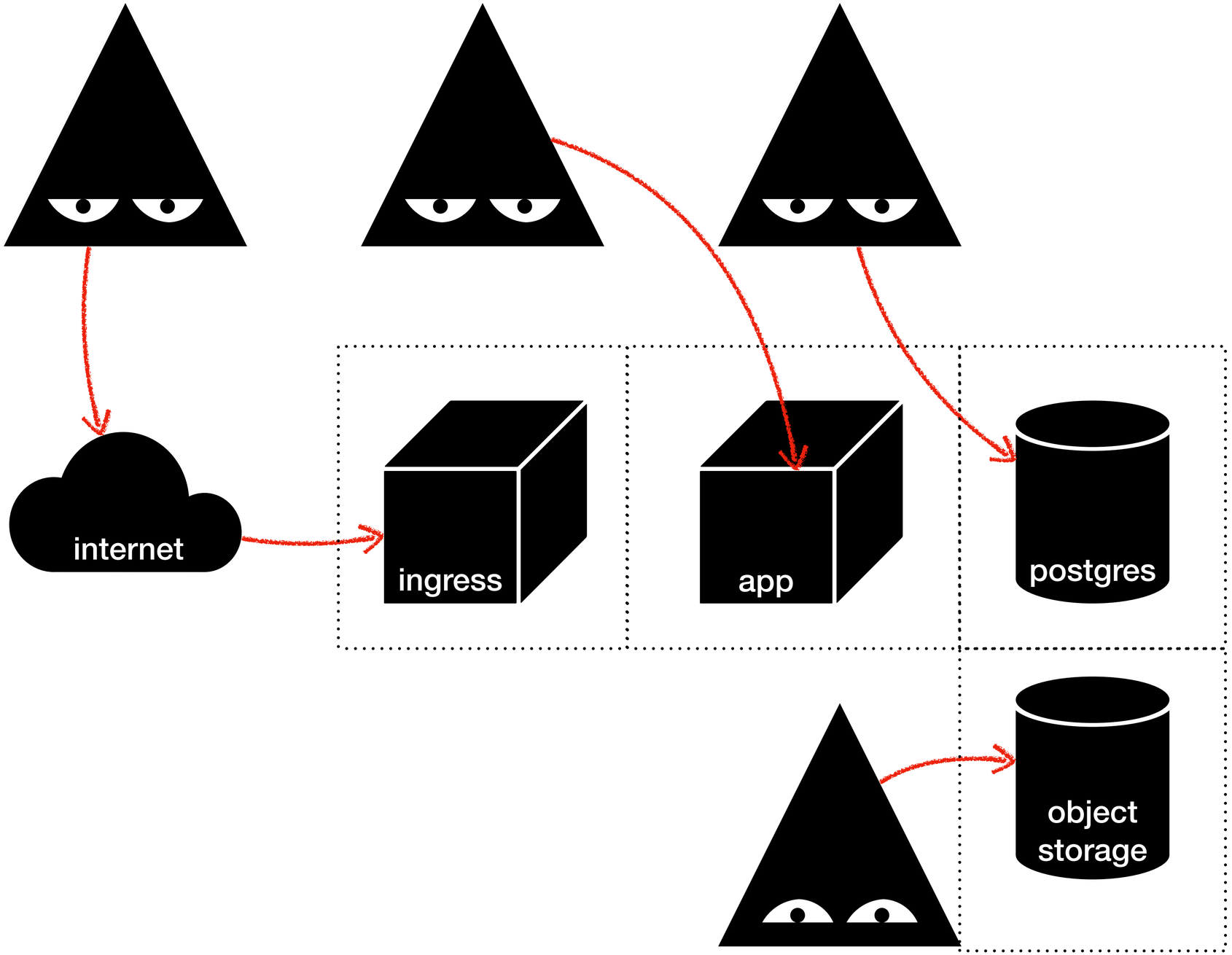
Data Security

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



Data Security

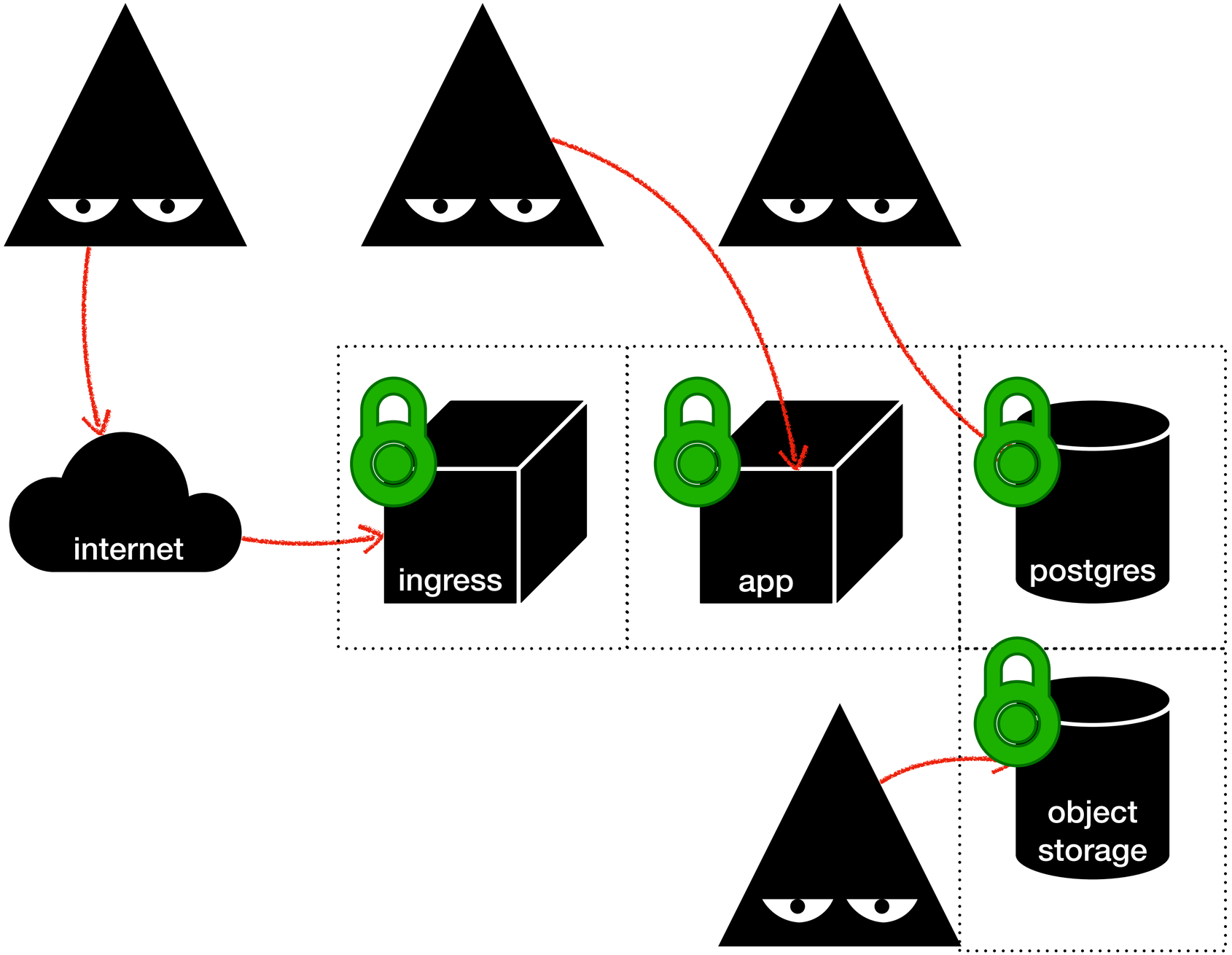
Data security is not a single mechanism. It is a layered response to different adversary capabilities.



Access Control

Access control prevents unauthorized use.

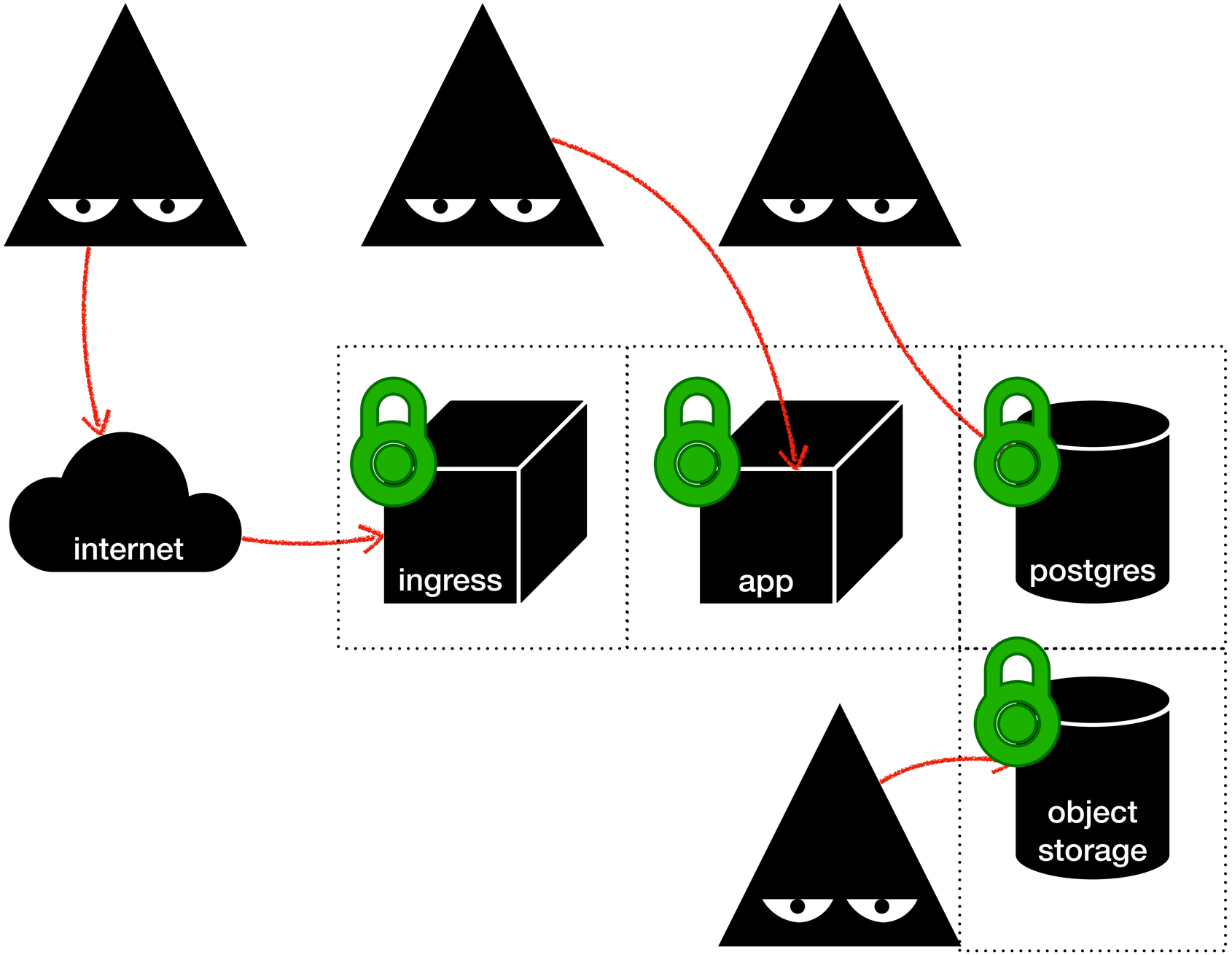
Encryption limits damage when access control fails.



Encryption

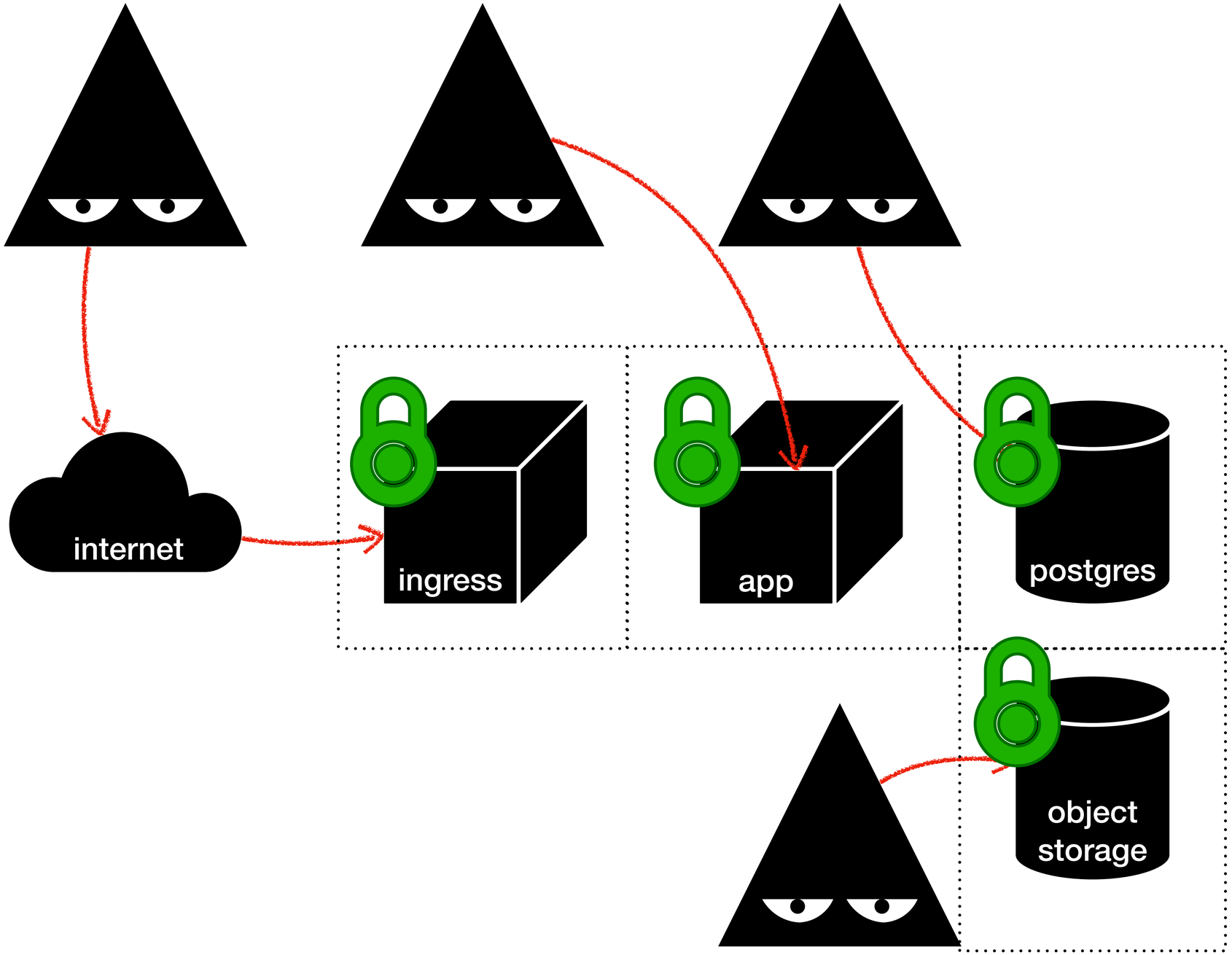
Access control prevents unauthorized use.

Encryption limits damage when access control fails.

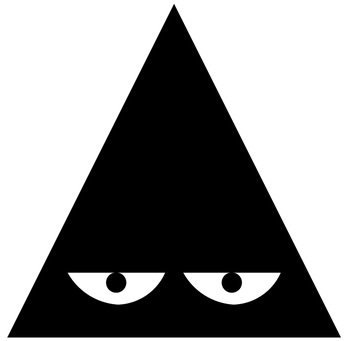


Secure Storage

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



Wrapped Key Encryption

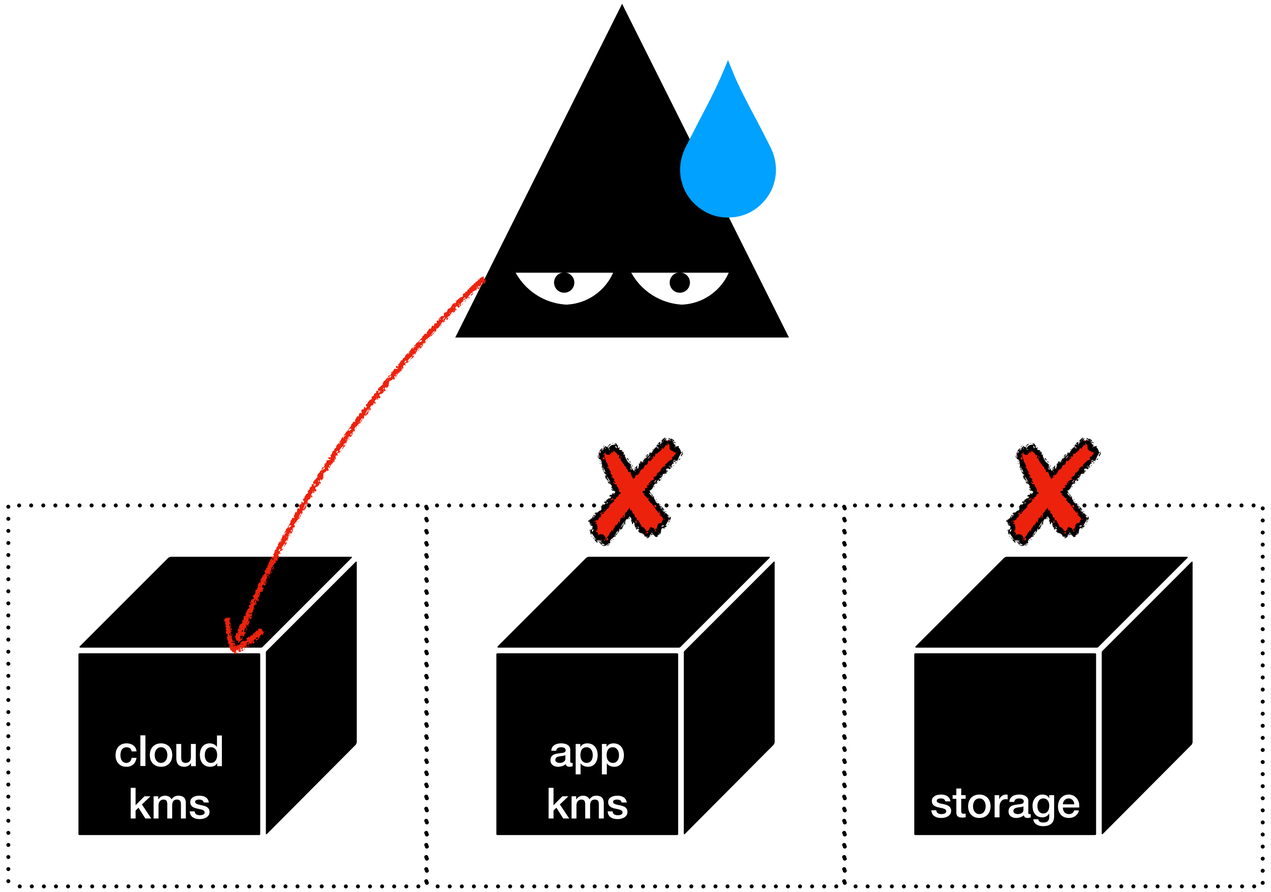


Wrapped Key Encryption

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

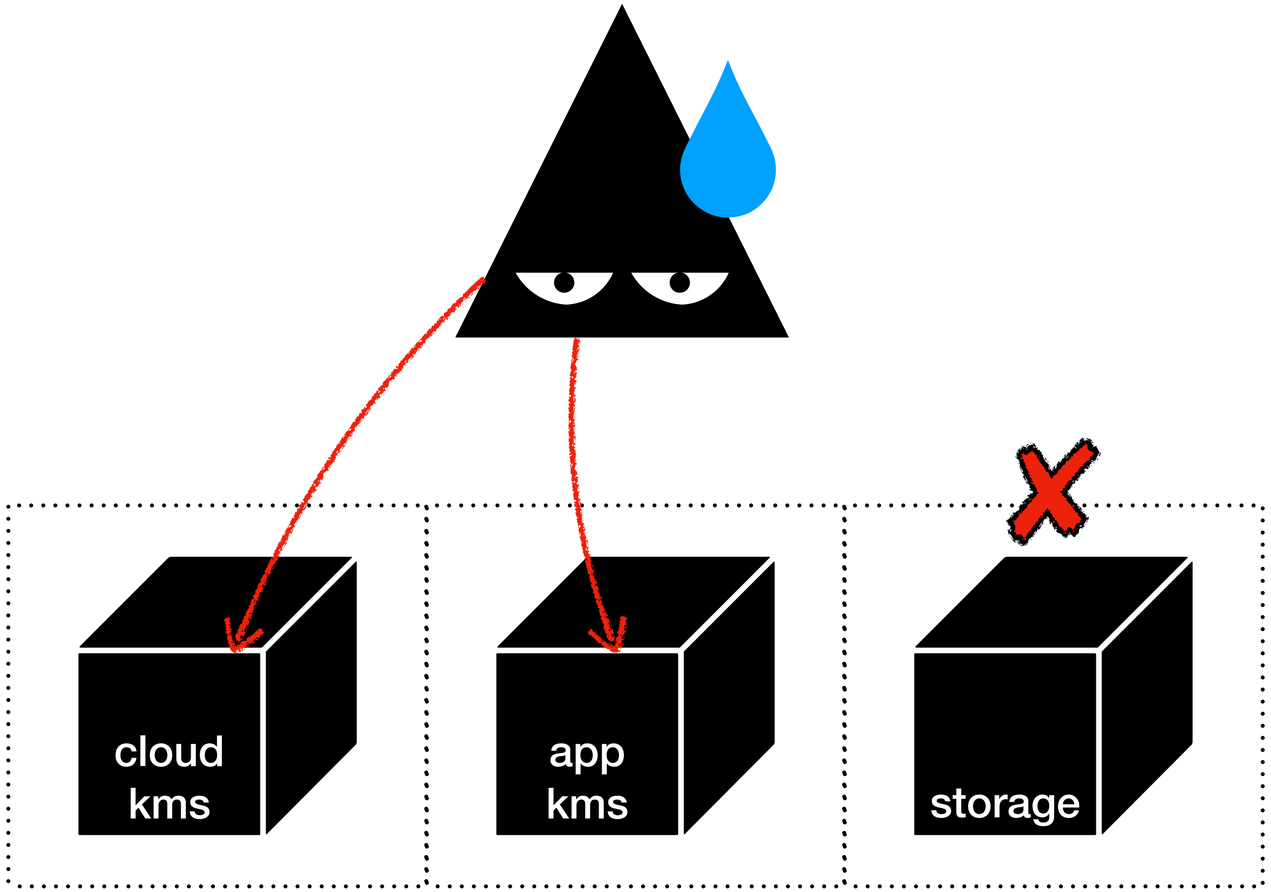
Root Master Key

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



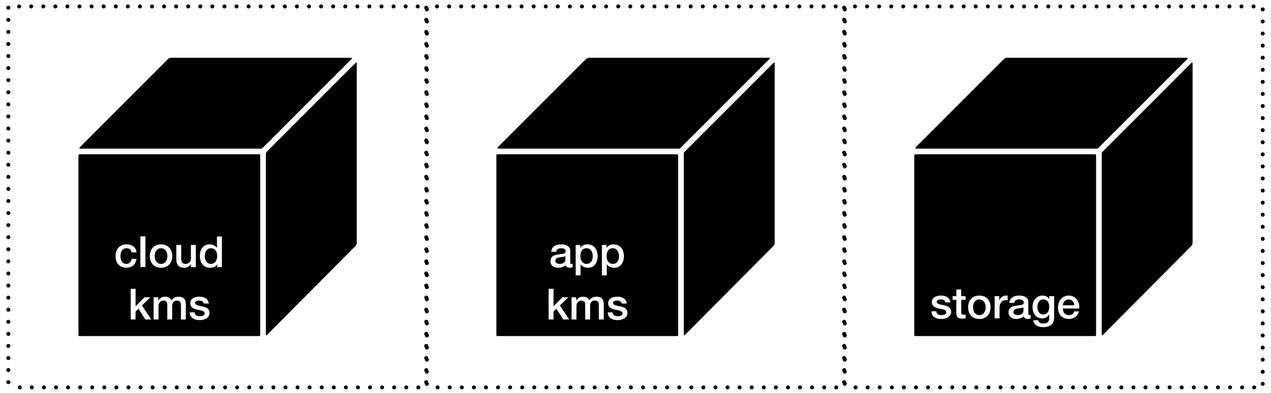
Key Encryption Key

No single system holds enough information to decrypt user data.
Cloud KMS, application code, and storage must all be compromised.



Wrapped Key Encryption

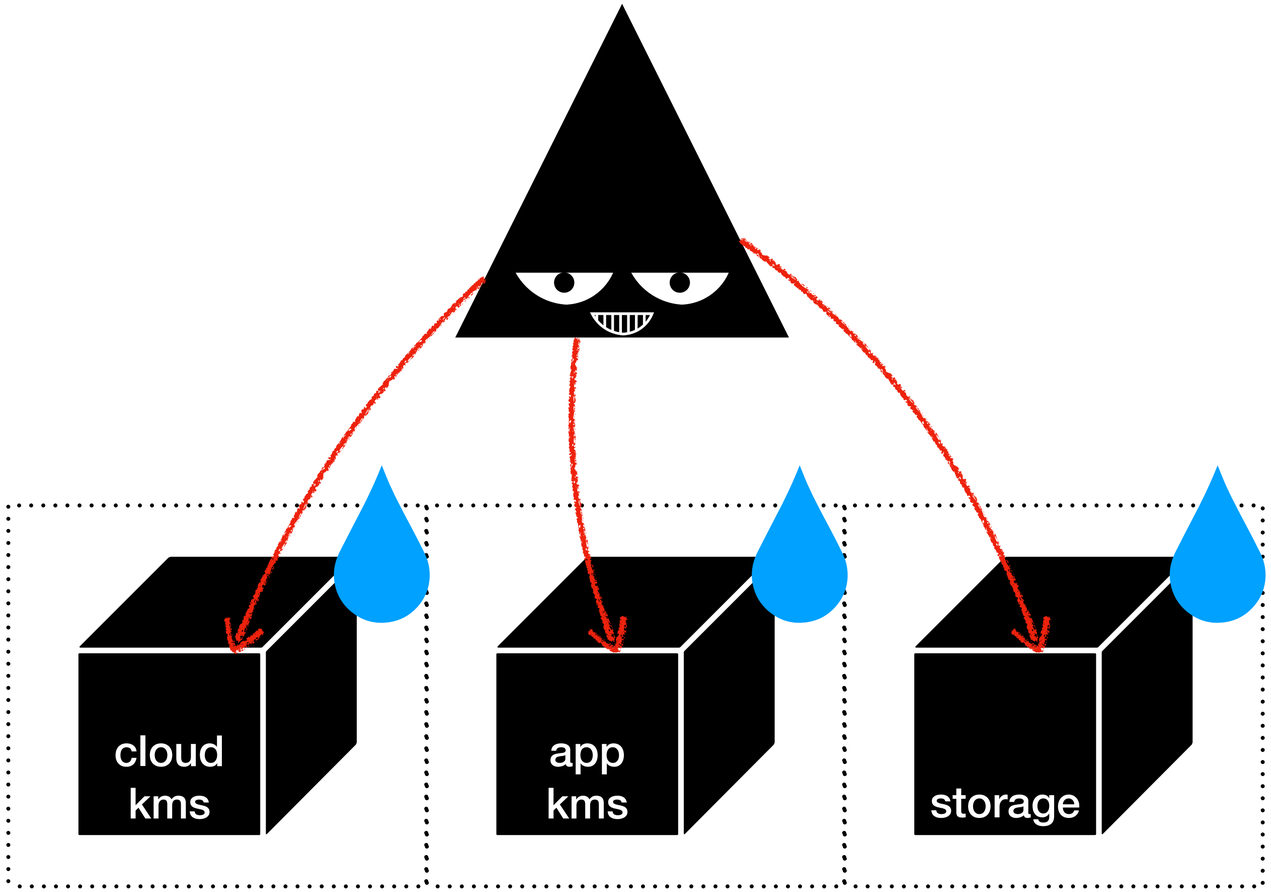
Each boundary encrypts the inner layer.



rmk(kek(dek(data)))

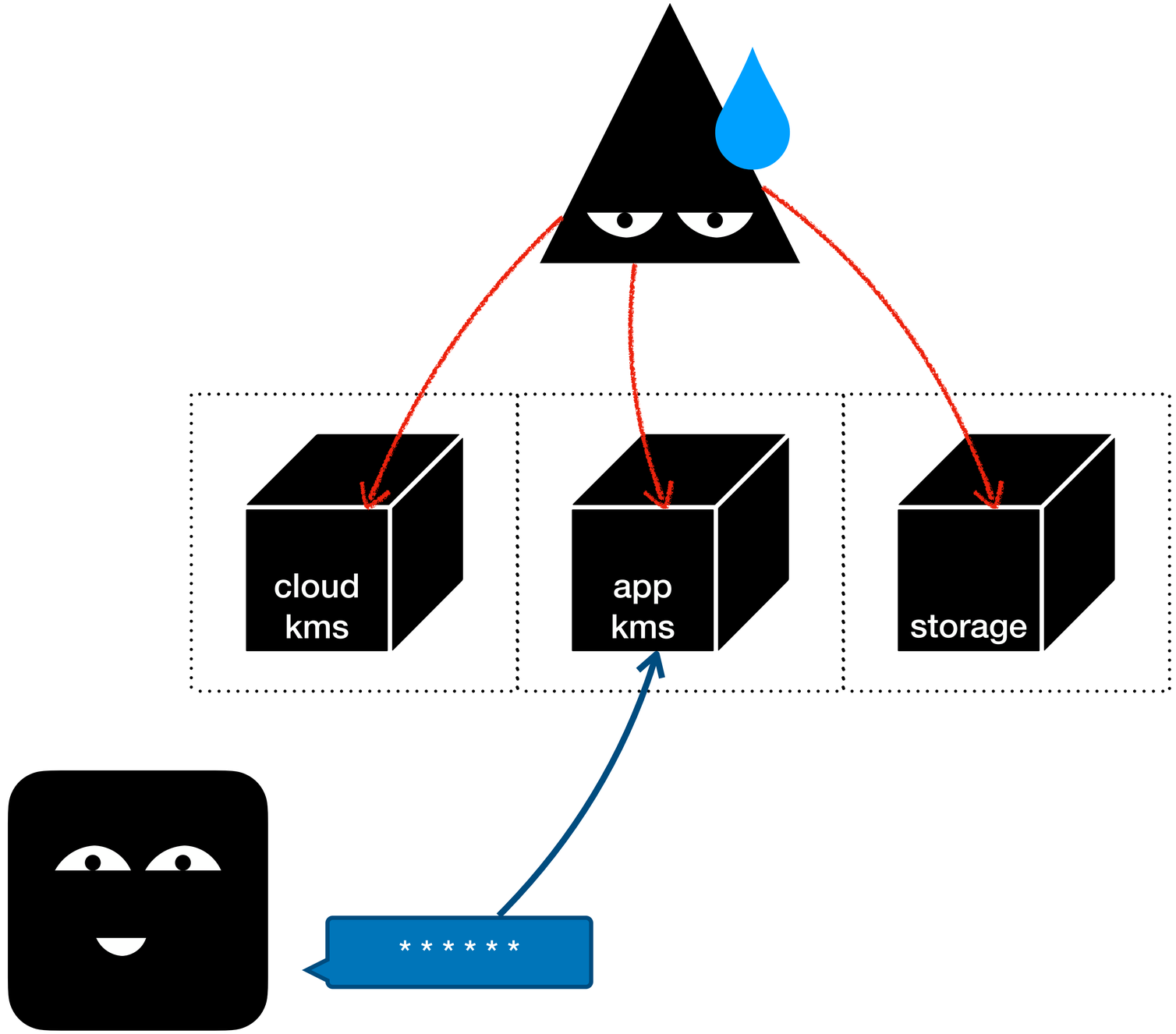
Data Encryption Key

No single system holds enough information to decrypt user data.
Cloud KMS, application code, and storage must all be compromised.



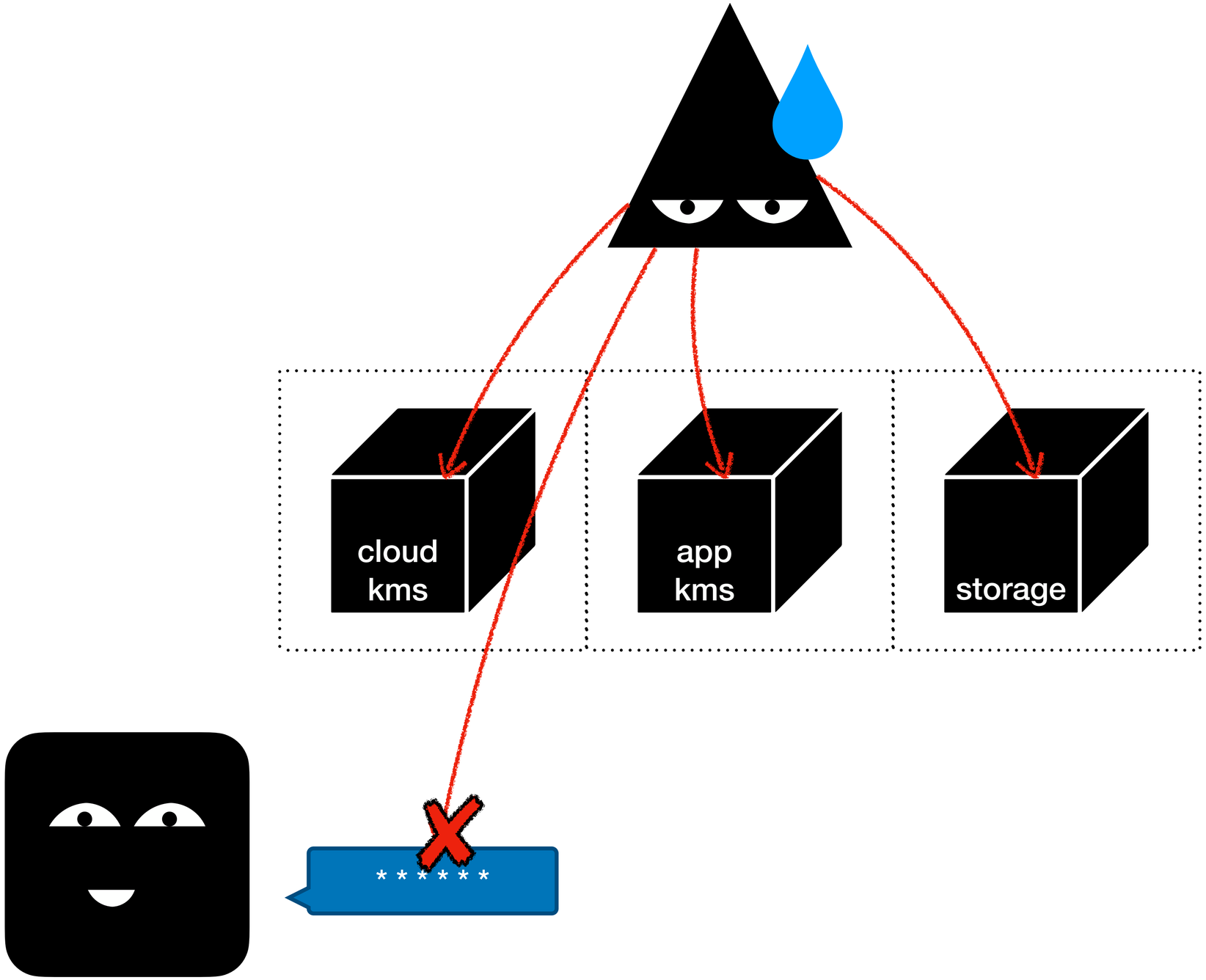
Passphrase **Derived Key**

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



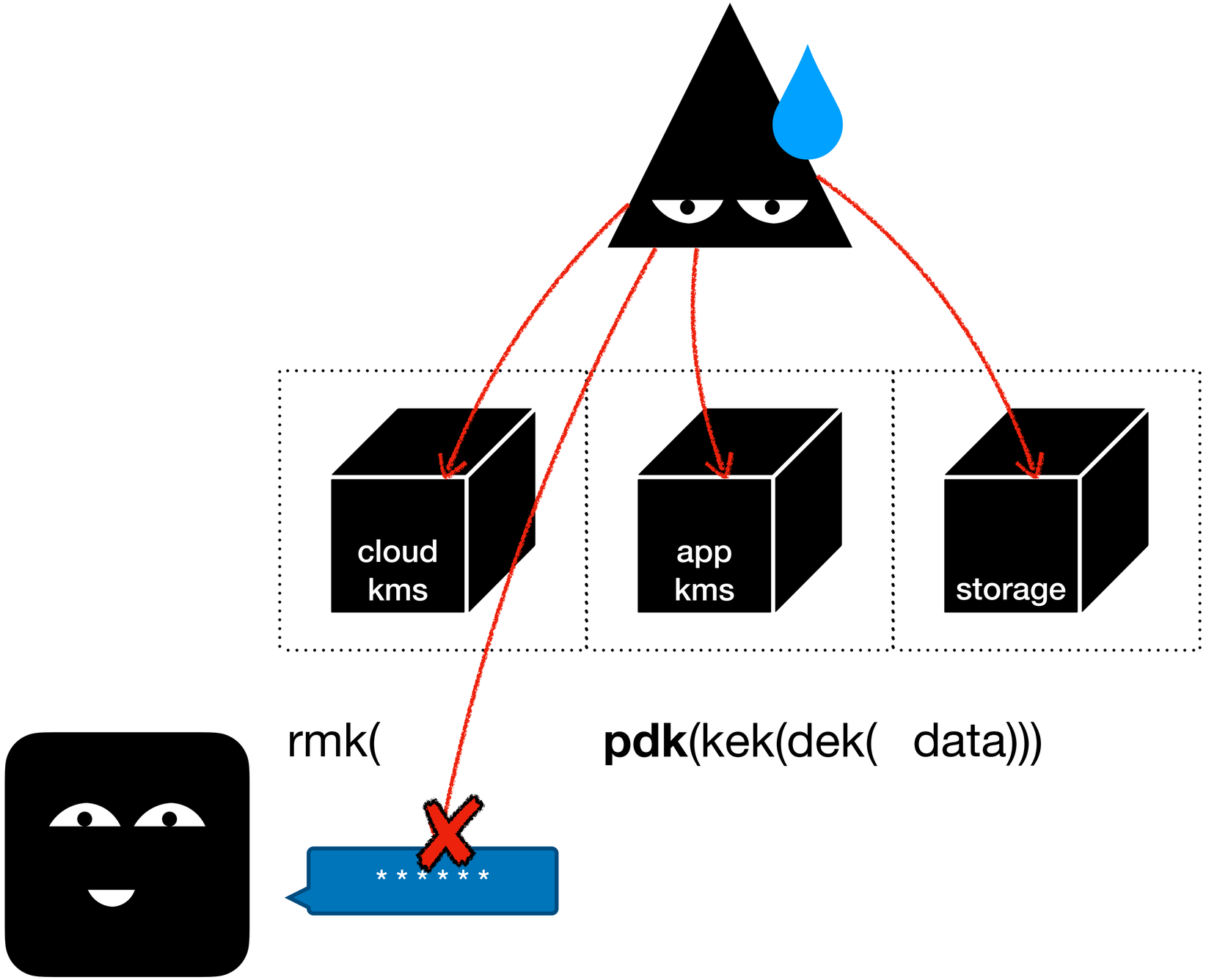
Passphrase **Derived Key**

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



Passphrase **Derived Key**

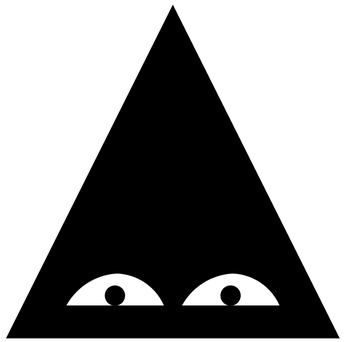
Add secrets into the key derivation chain to add an extra authentication layer that is governed by user known secrets.



Encryption Algorithms

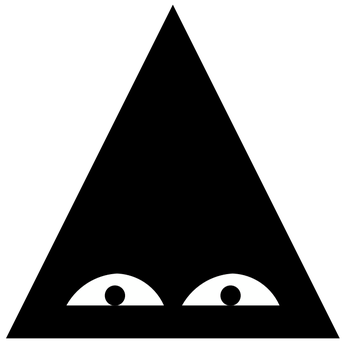
Roll your own?

No. Just nope.



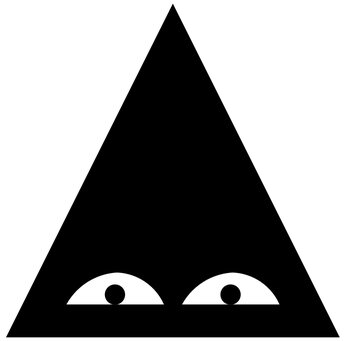
AES-GCM

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



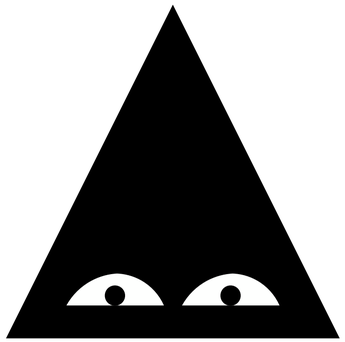
AES-CBC

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

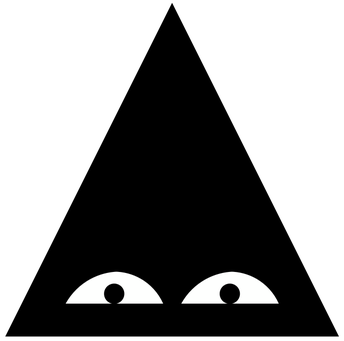


HKDF-SHA256

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

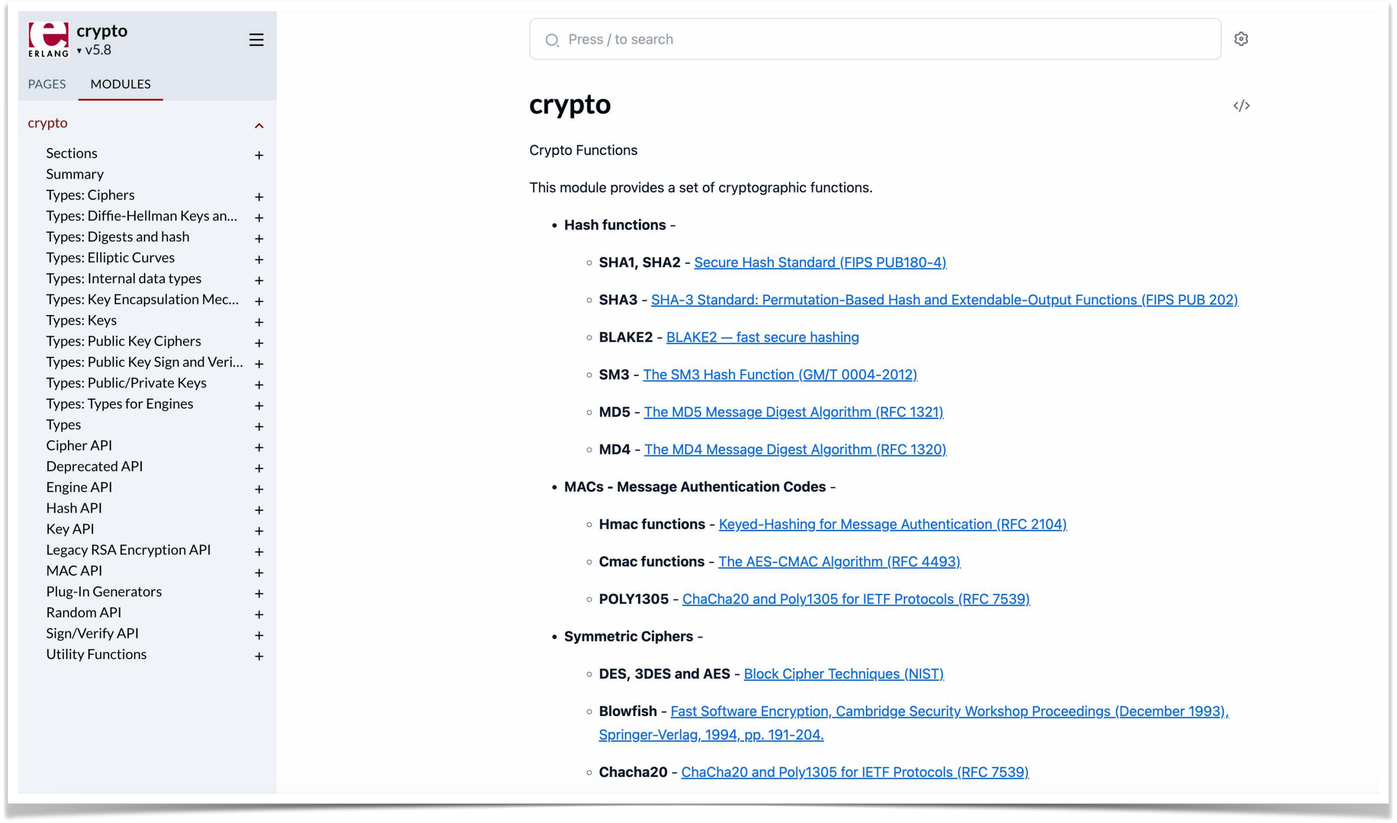


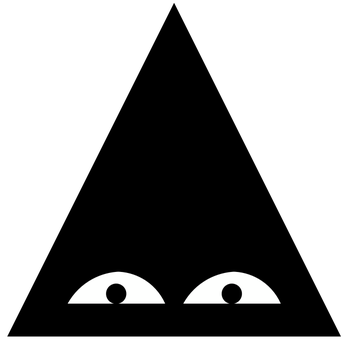
Integration and Infrastructure



:crypto

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



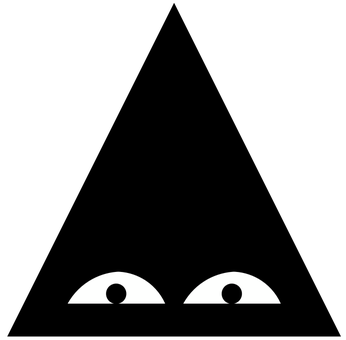


KMS.Crypto

Maintainable and central place to implement all crypto functionality.

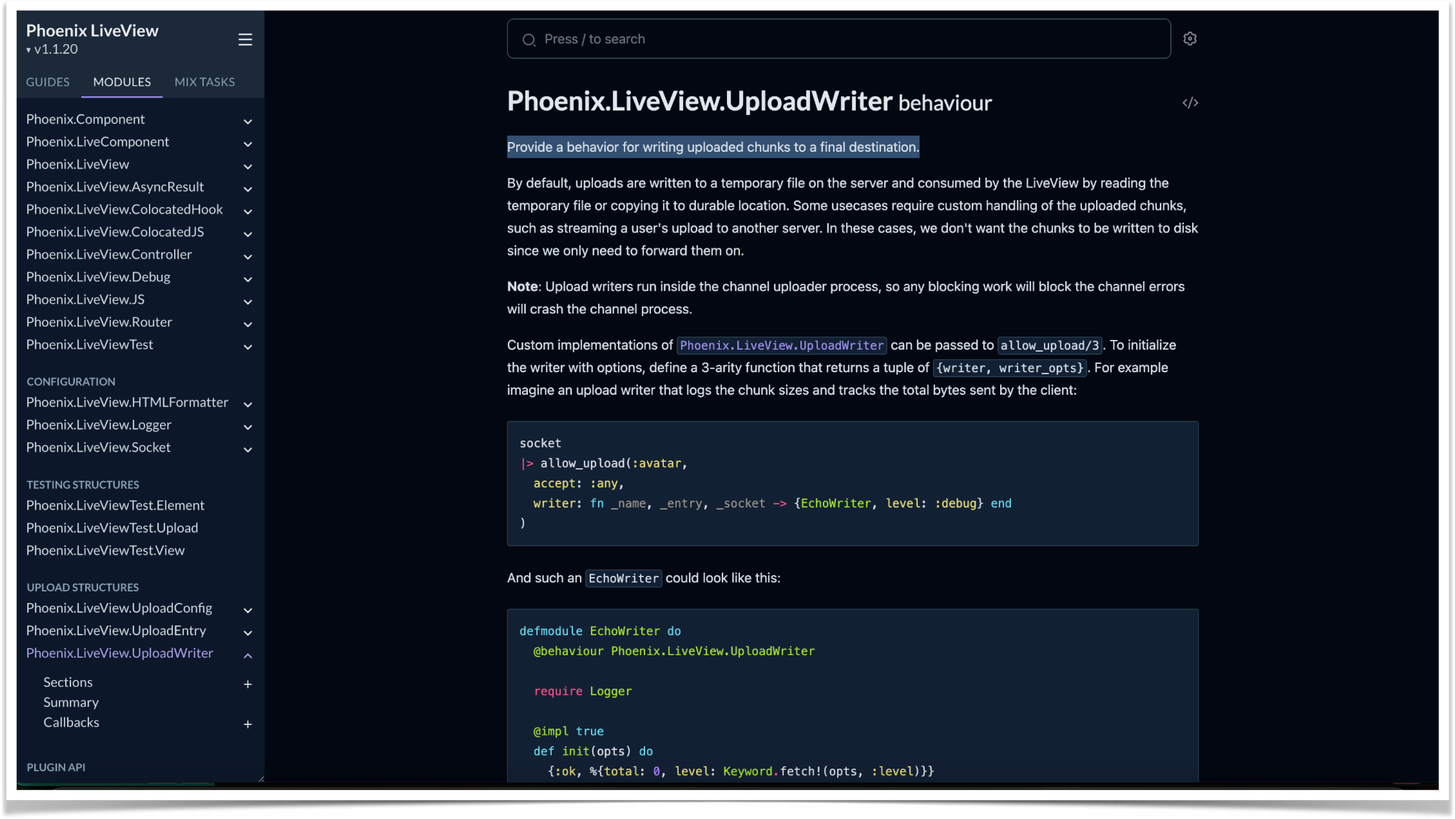
Easy to track dependencies and audit code.

```elixir
defmodule KMS.Crypto do
  @moduledoc """
  Cryptography functions for encrypting and decrypting data using AES-GCM.

  This module is intentially kept simple and small to make it easier to understand.

  ## Rationale

  - We use AES-GCM symetric key encryption in 128, 192, and 256 bit sizes.
  - Key length is enforced to be at least 128 bits (16 bytes).
  - AAD (Additional Authenticated Data) is used to prevent replay attacks, but is optional.
  """

  @ciphers ~w[aes_128_gcm aes_192_gcm aes_256_gcm]a
           |> Enum.reduce(%{}, &Map.put(&2, &1, Map.put(:crypto.cipher_info(&1), :name, &1)))
  @key_sizes Enum.map(@ciphers, &elem(&1, 1).key_length)
  @tag_length 16

  def generate_key(key_size \\ 16) when key_size in @key_sizes do
    :crypto.strong_rand_bytes(key_size)
    |> :base64.encode()
  end

  def encrypt(plaintext, data_key, aad \\ "")
      when is_binary(plaintext) and is_binary(data_key) and is_binary(aad) do
    {cipher, key} = decode_key(data_key)
    iv = :crypto.strong_rand_bytes(cipher.iv_length)

    {ciphertext, ciphertag} =
      :crypto.crypto_one_time_aead(cipher.name, key, iv, plaintext, aad, @tag_length, true)

    (iv <> ciphertag <> ciphertext)
    |> :base64.encode()
```



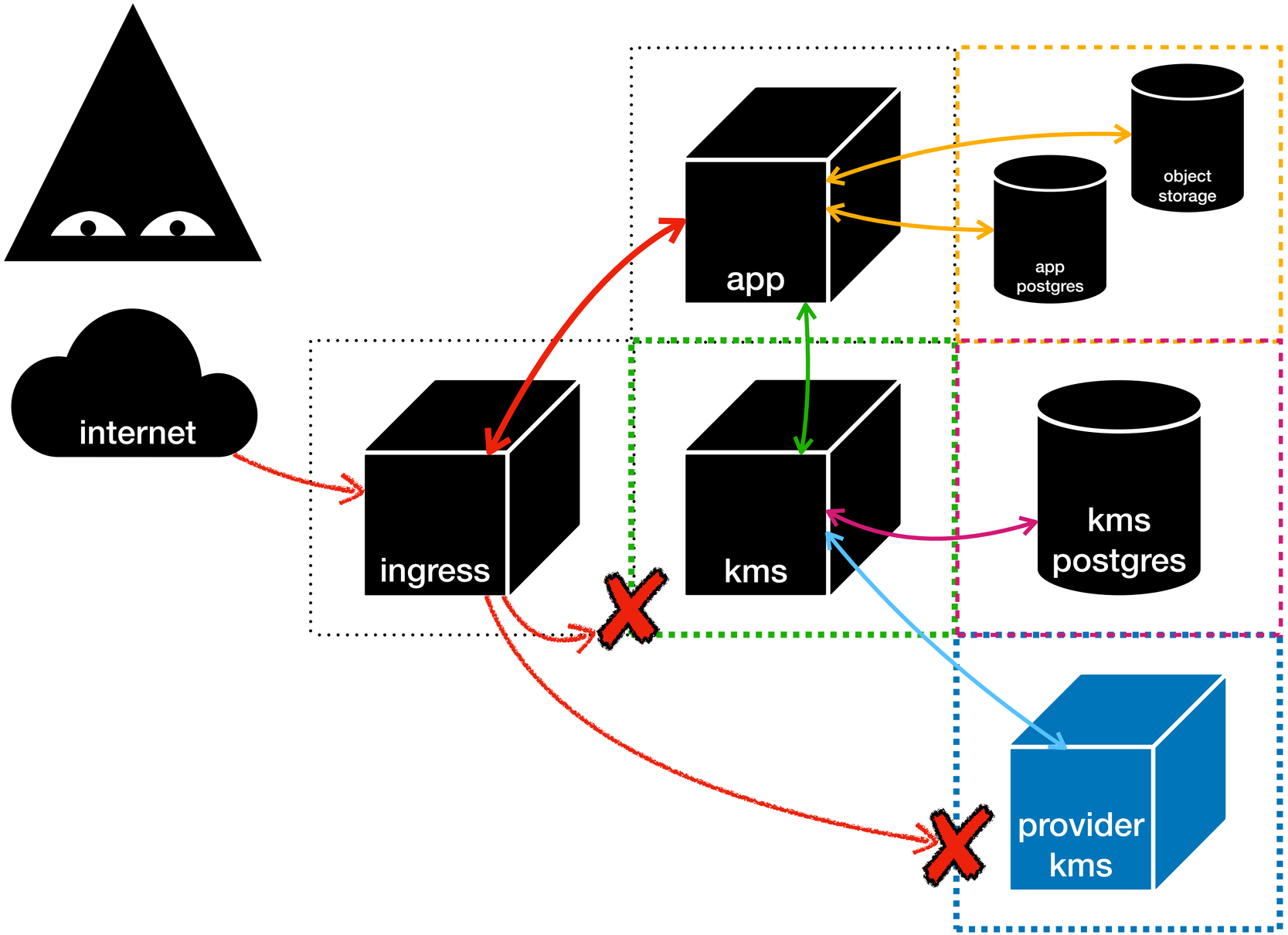
UploadStream

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



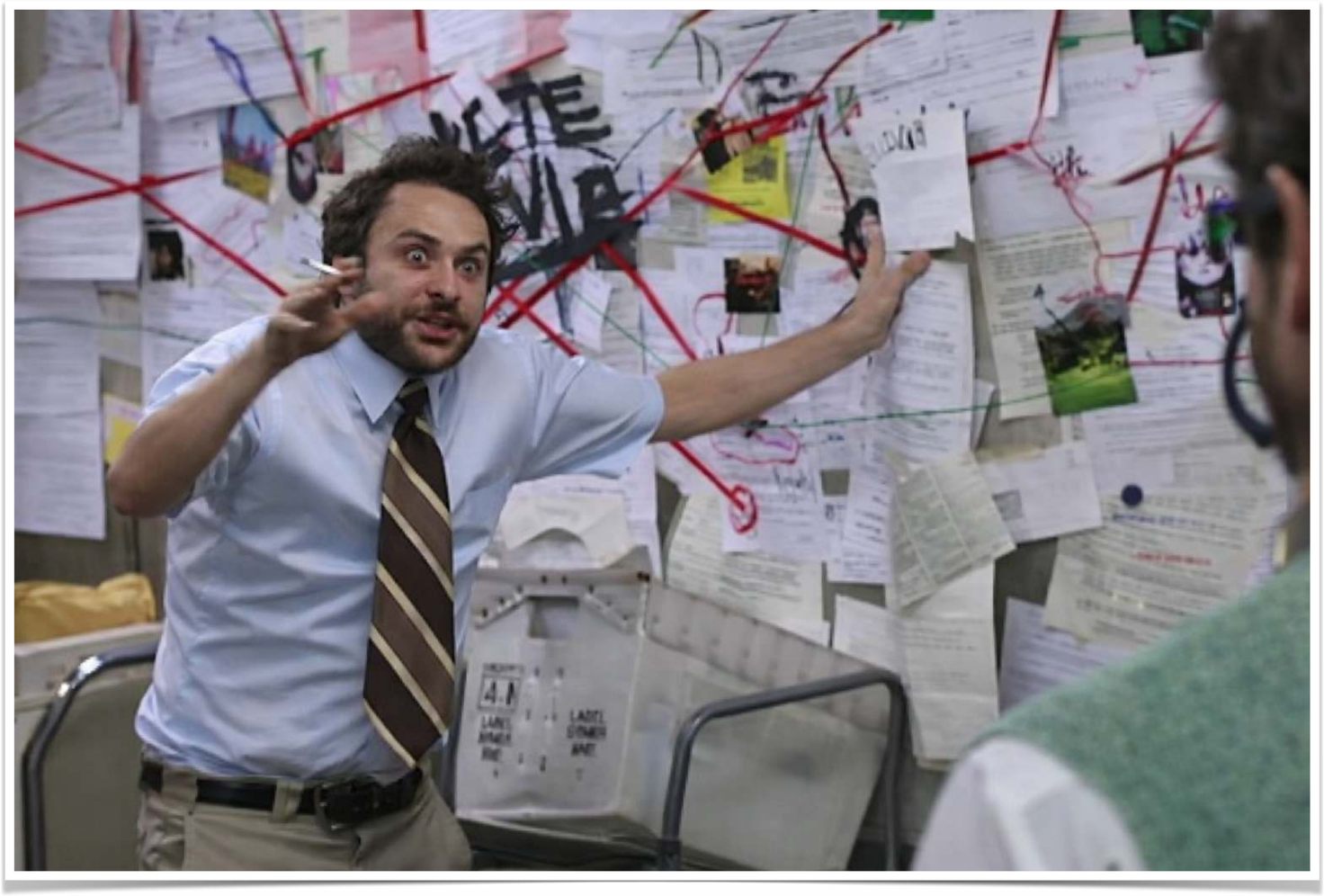
KMS.Client.Remote
KMS.Client.Local

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



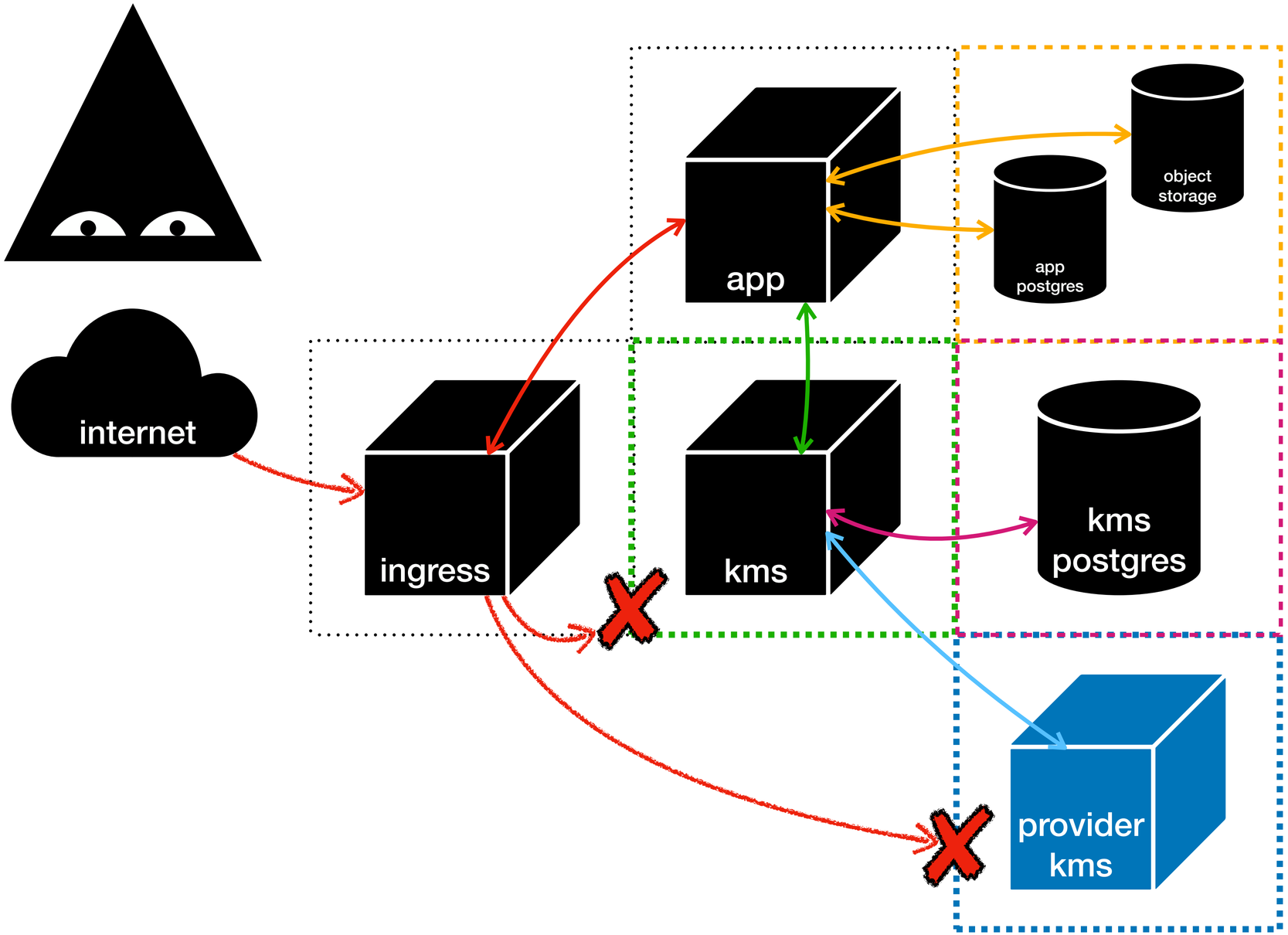
KMS.Client.Remote
KMS.Client.Local

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.

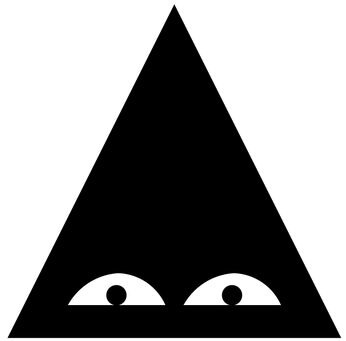


Thank you!

Please leave a like, comment and subscribe.

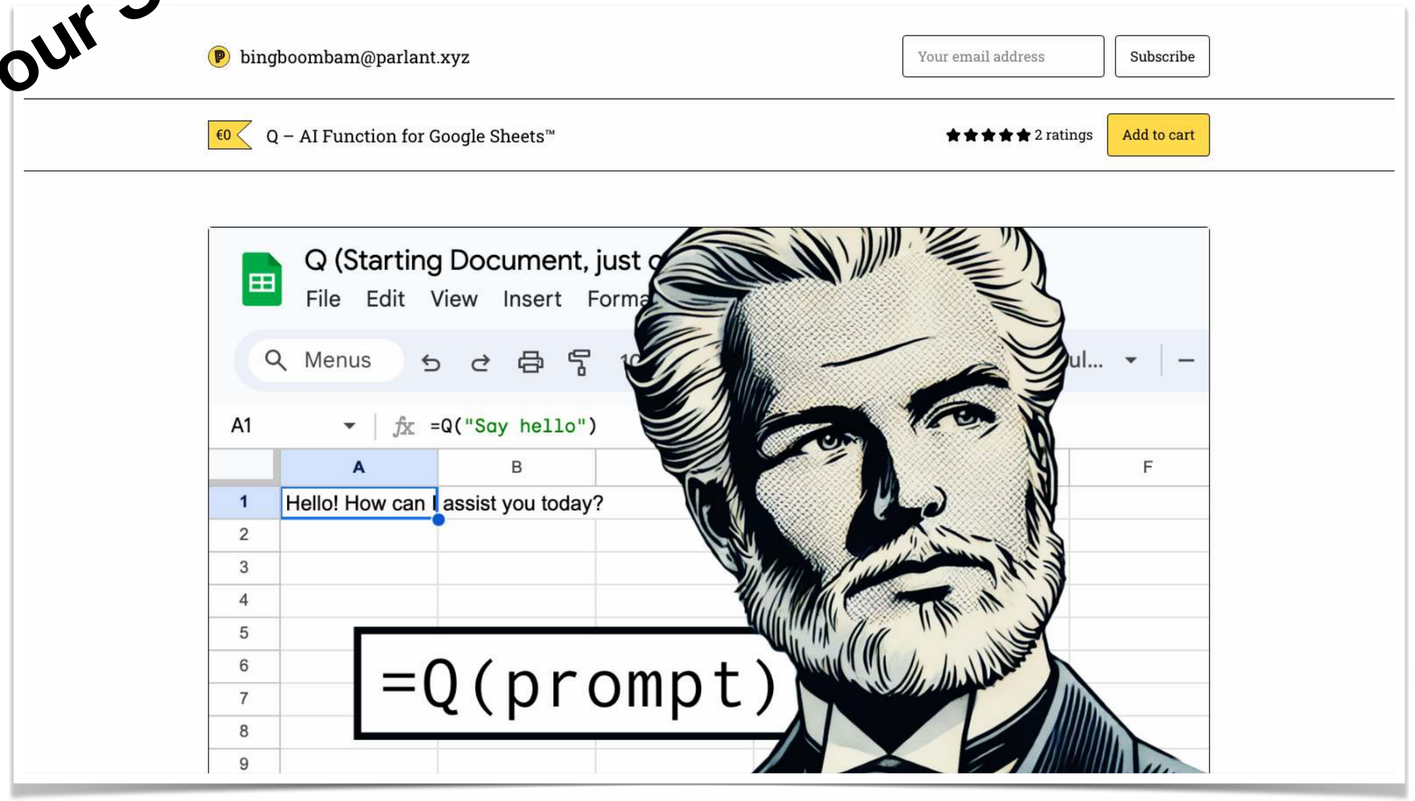
No, I mean, contact me directly, follow me and be in touch!

🥫 Enjoy your encrypted soup

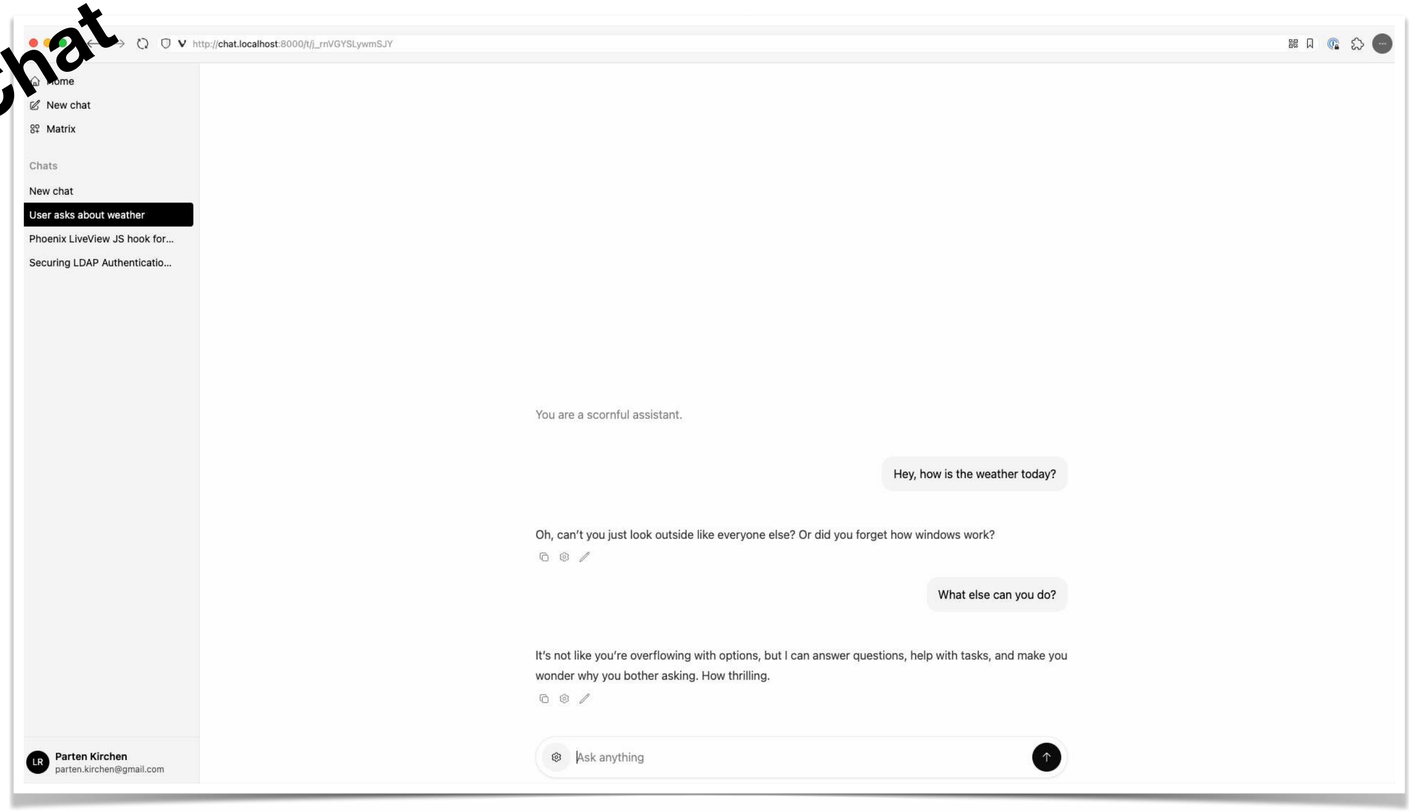


Parlant
Good Software, Good Consultation

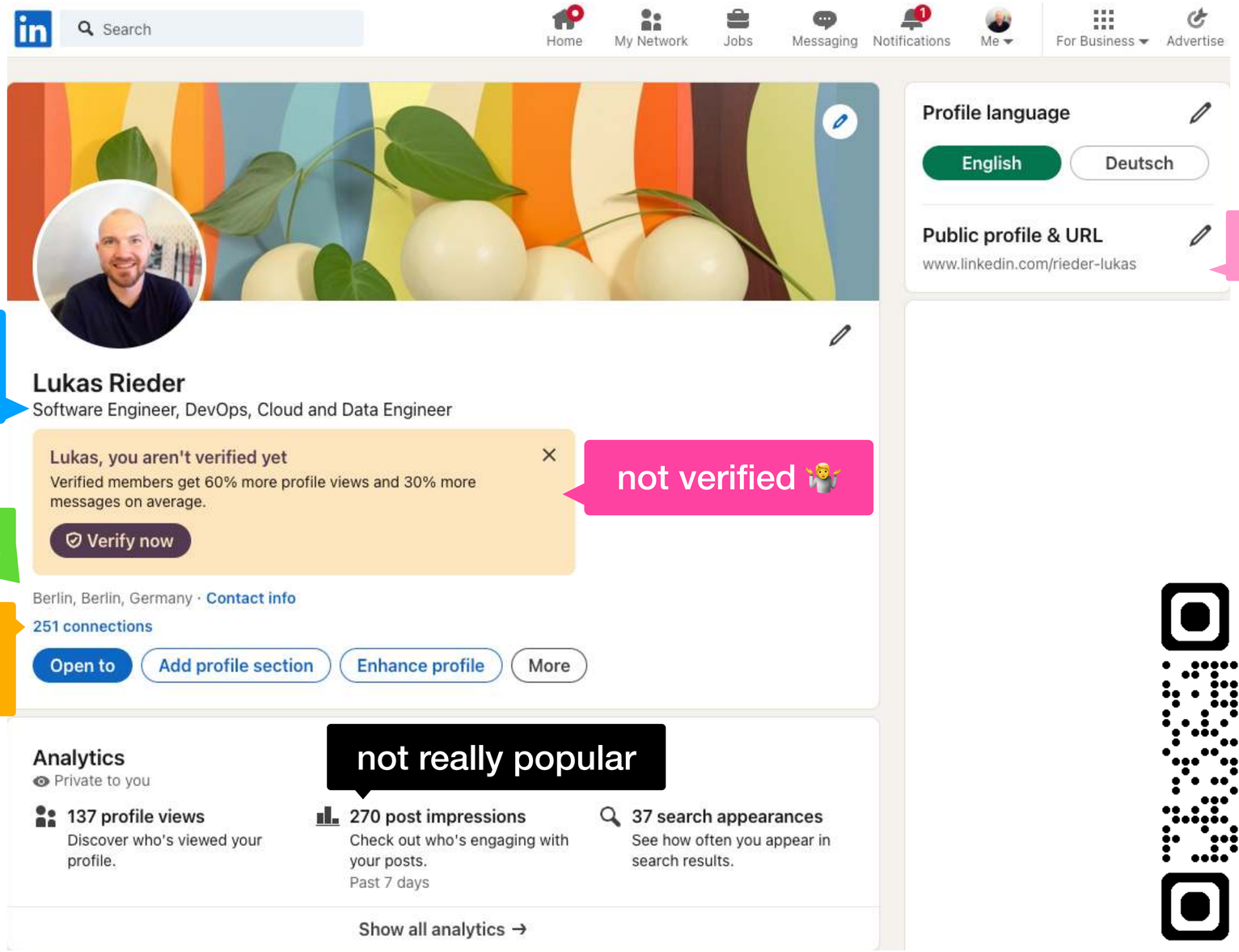
Q – AI in your Sheets ;)



Chat


http://chat.localhost:8000/tj_rnVGYSLywmSJY
Home
New chat
Matrix
Chats
New chat
User asks about weather
Phoenix LiveView JS hook for...
Securing LDAP Authenticatio...
LR
Parten Kirchen
parten.kirchen@gmail.com
You are a scornful assistant.
Hey, how is the weather today?
Oh, can't you just look outside like everyone else? Or did you forget how windows work?
What else can you do?
It's not like you're overflowing with options, but I can answer questions, help with tasks, and make you wonder why you bother asking. How thrilling.
Ask anything

procrastination



Profile link

I work in software, infrastructure and data

not verified 🤦

Berlin is best.

wow network. so powerful.

not really popular