SLEEP BETTER WITH

CONTENT ENCRYPTION

Jens Neuhalfen

https://github.com/neuhalje/presentation_content-encryption

WHO AM I?

- Jens Neuhalfen
- Age: Forty something
- IT since: ever
- Skills: Bridge between IT and business, IT-Security Management, writing software
- https://github.com/neuhalje

SUMMARY

Regulations apply - whatever you do!

Encryption is not for free!

No encryption might be way more expensive!

Encryption is a safety net (*last* line of defence)

–> Assess risks & cost, plan, implement!

SUMMARY

Regulations apply - whatever you do!

Encryption is not for free!

No encryption might be way more expensive!

Encryption is a safety net (*last* line of defence)

–> Assess risks & cost, plan, implement!

I AM NOT A CRYPTOGRAPHER!

I AM NOT A LAWYER!

THIS TALK MADE ME A CRYPTOGRAPHY AND/OR LEGAL EXPERT

said no-one ever

I AM NOT A LAWYER!

I AM NOT A CRYPTOGRAPHER!

YOUR DATA

YOUR DATA

Collect

YOUR DATA

Collect

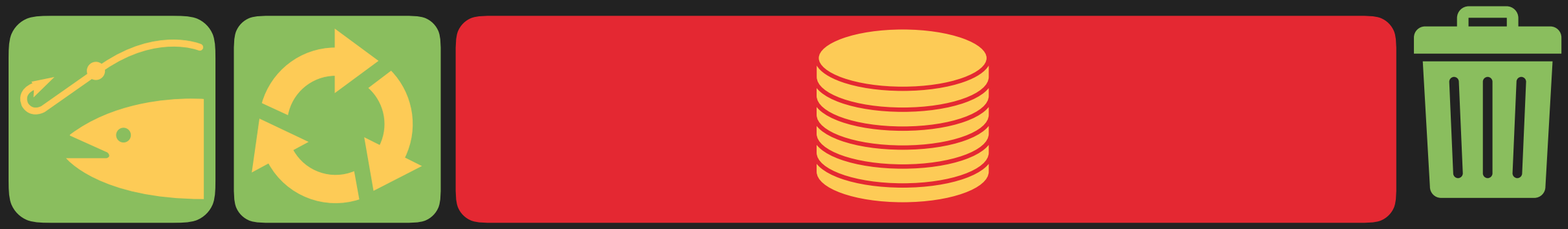
Process

YOUR DATA

Collect Store *but not use*

Process

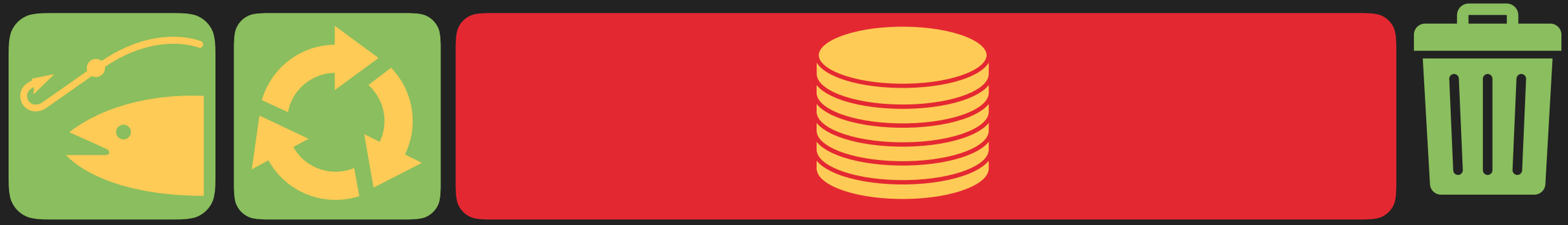
YOUR DATA



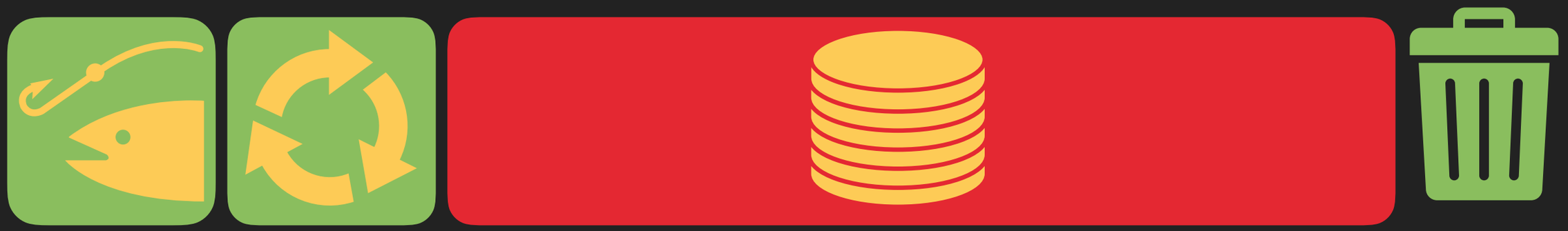
Collect Store *but not use* Delete

Process

YOUR DATA



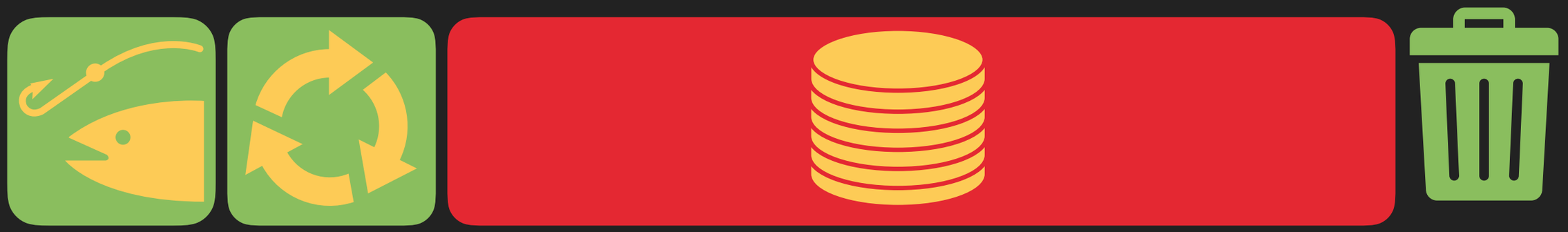
YOUR DATA



YOUR DATA

Business value

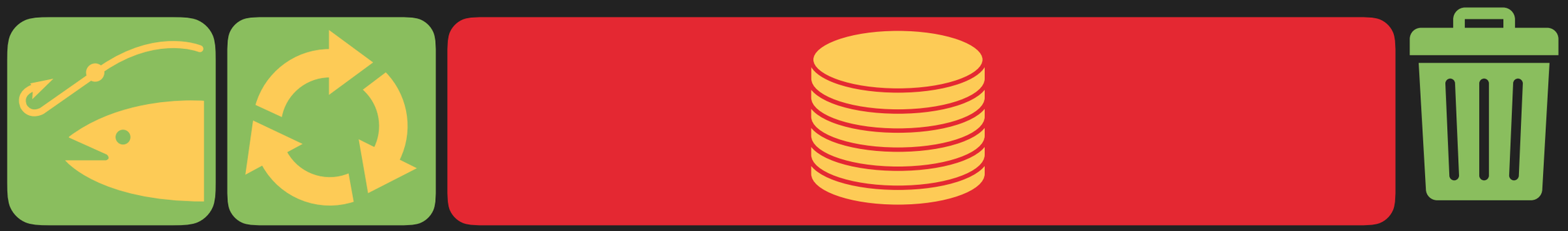
Liability



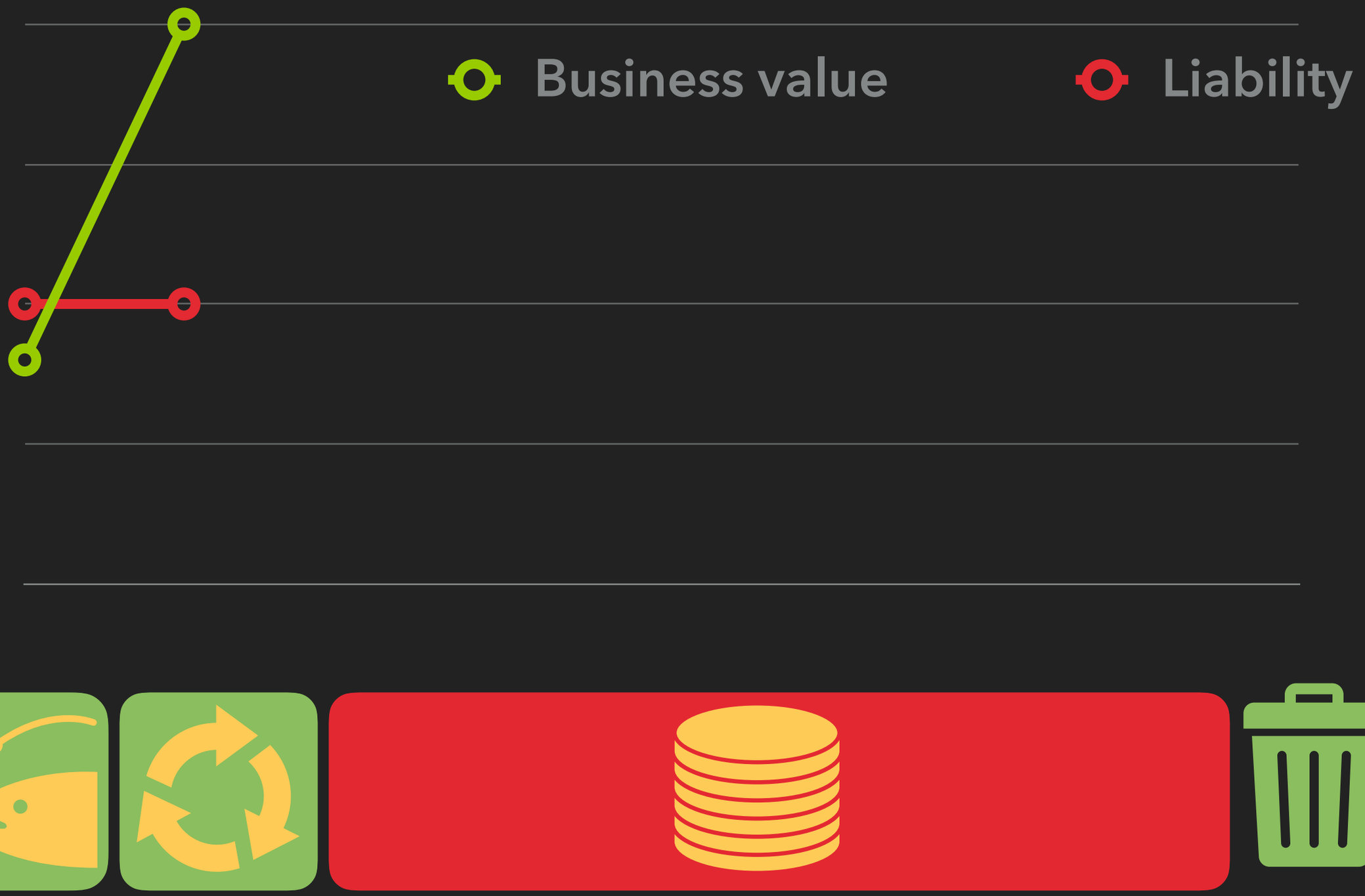
YOUR DATA

Business value

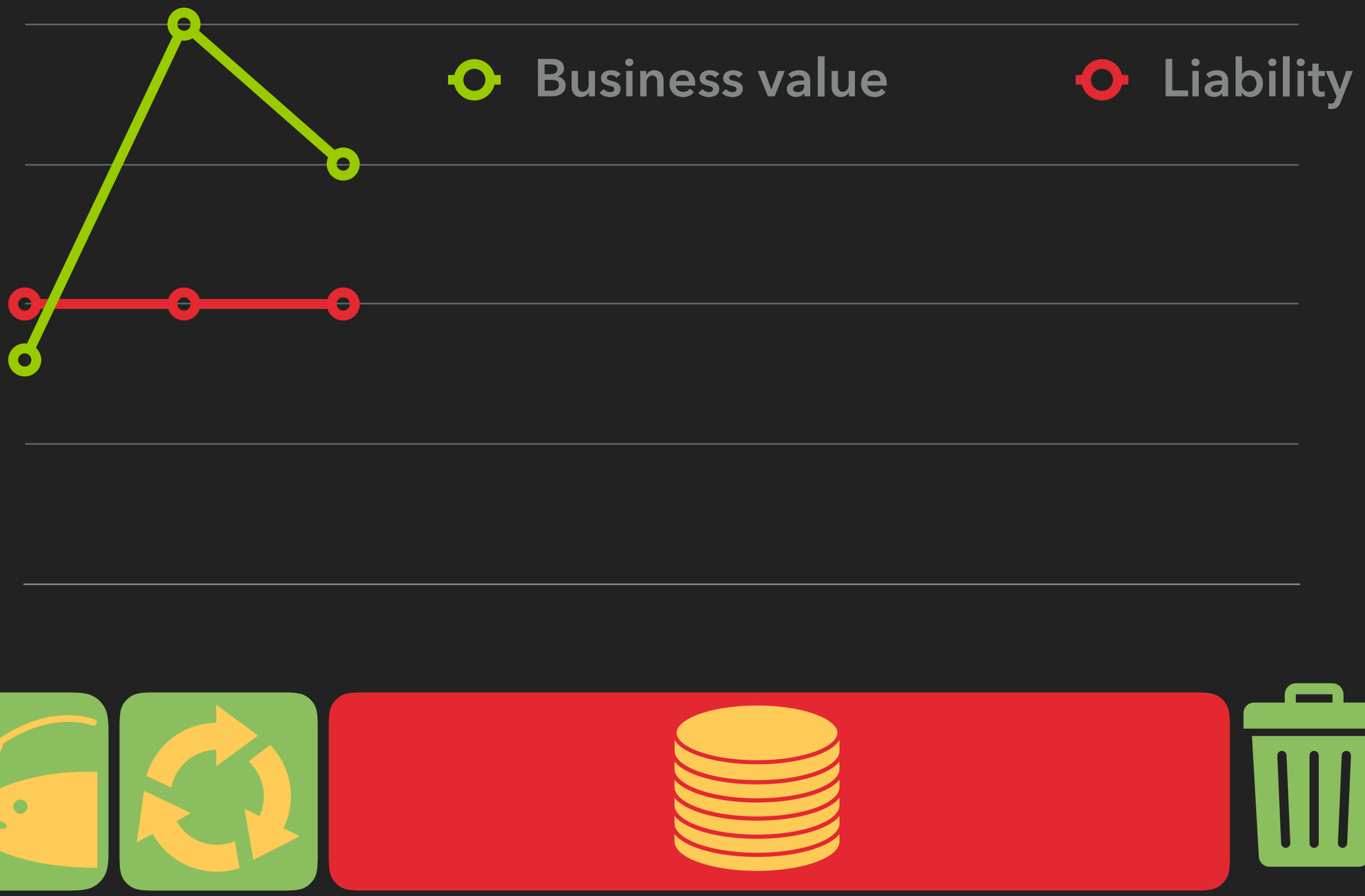
Liability



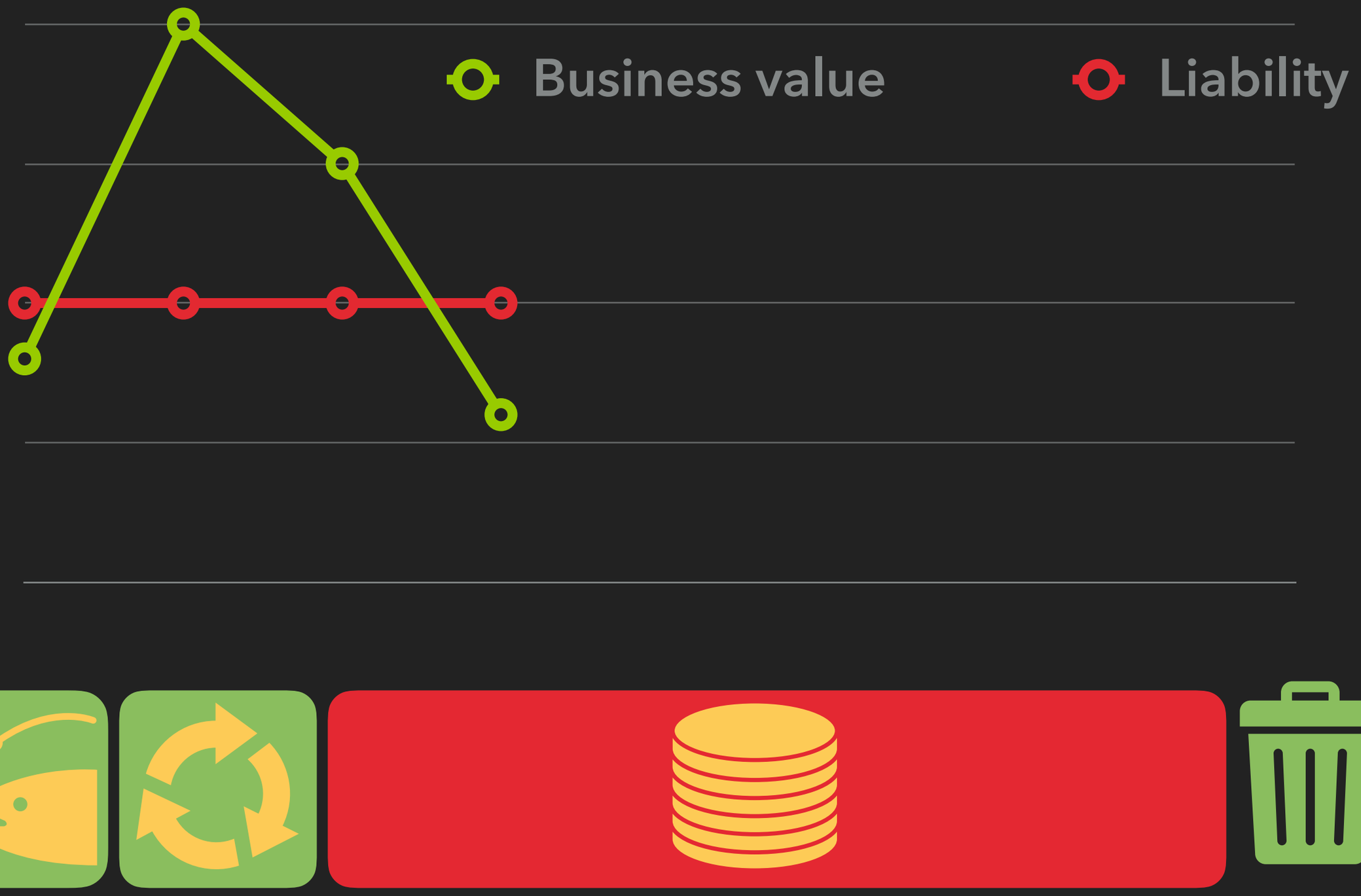
YOUR DATA



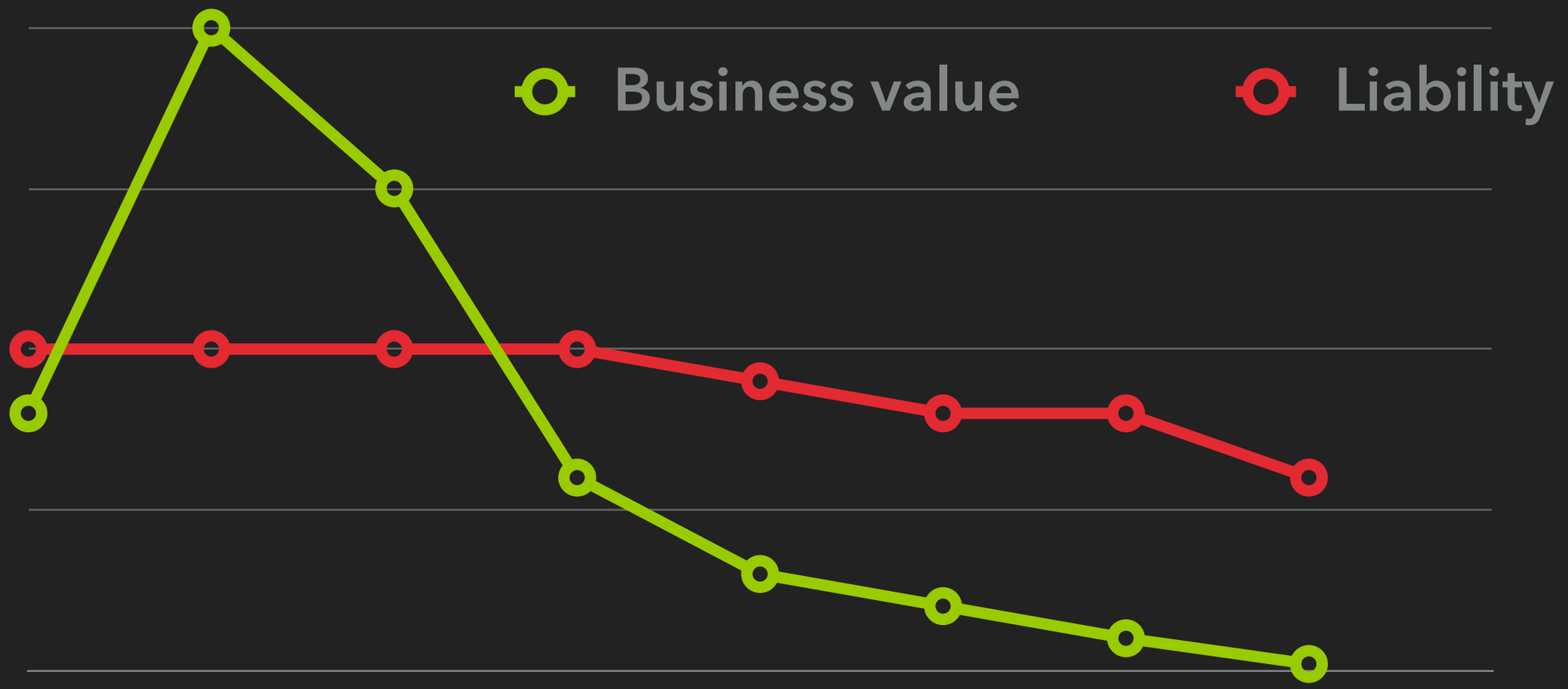
YOUR DATA

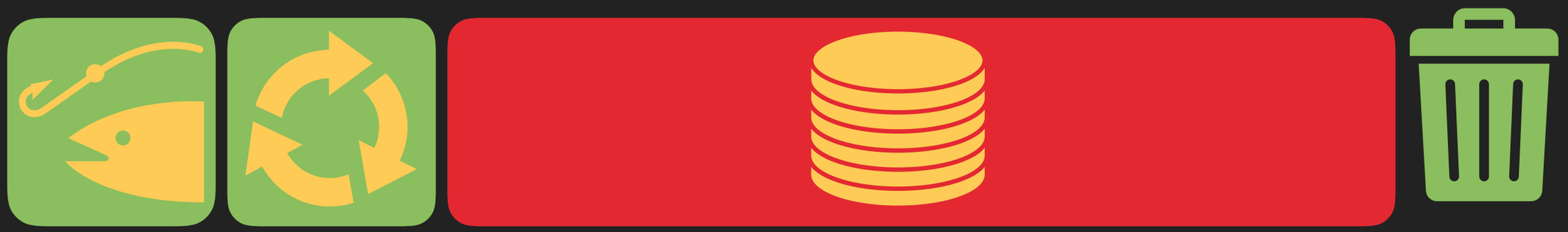


YOUR DATA

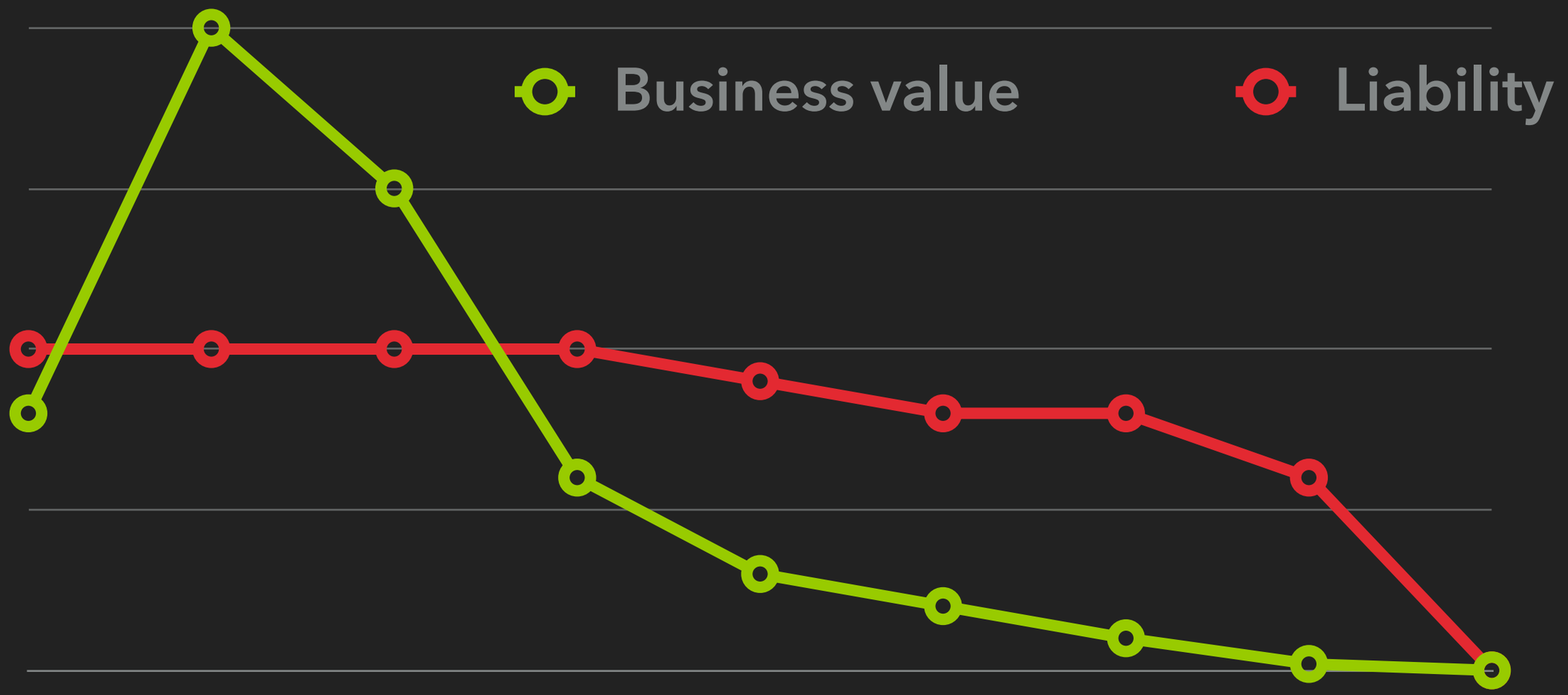


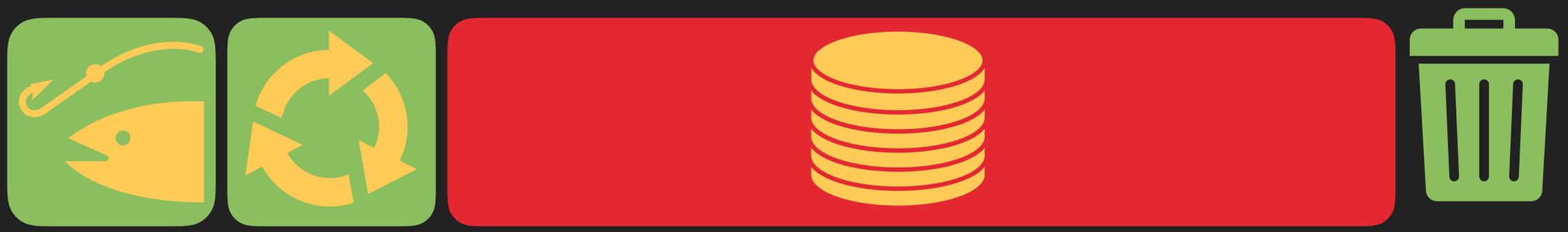
YOUR DATA



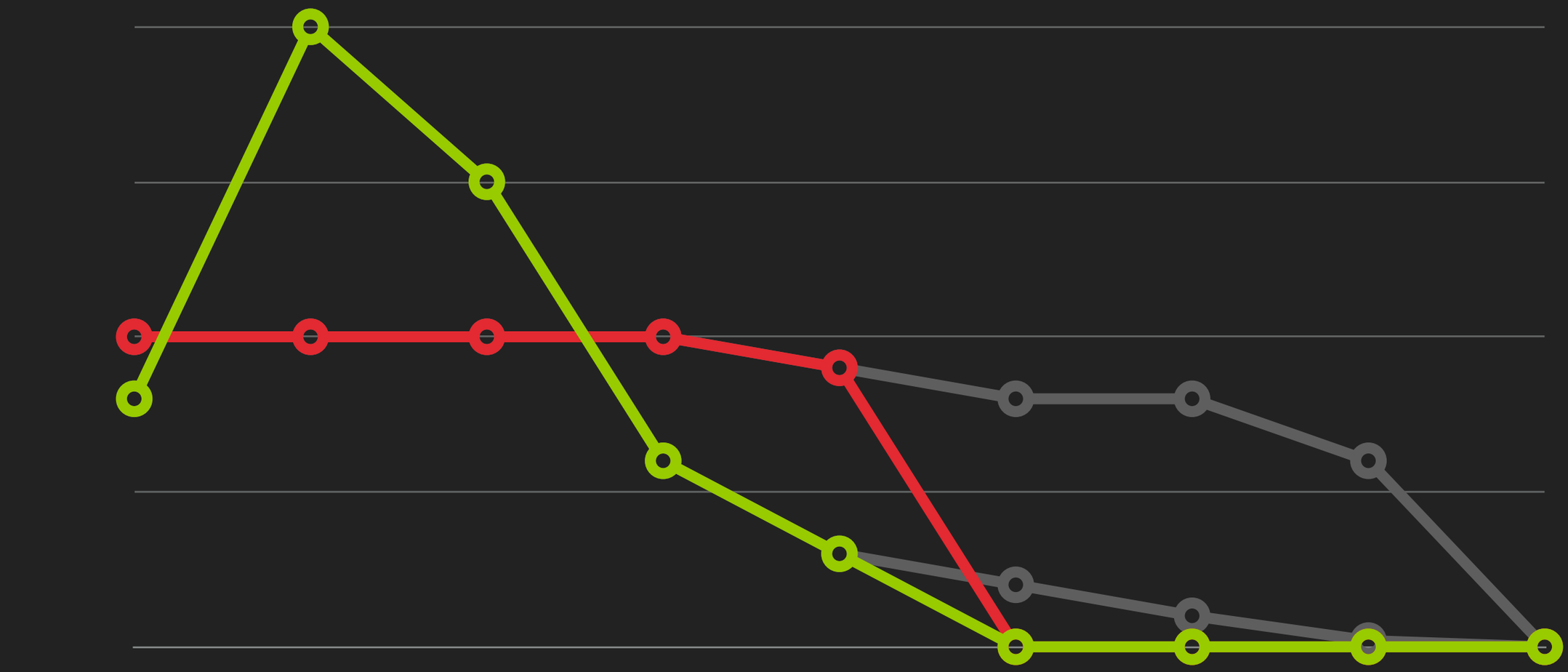


YOUR DATA





YOUR DATA



WHAT IS

CONTENT ENCRYPTION?

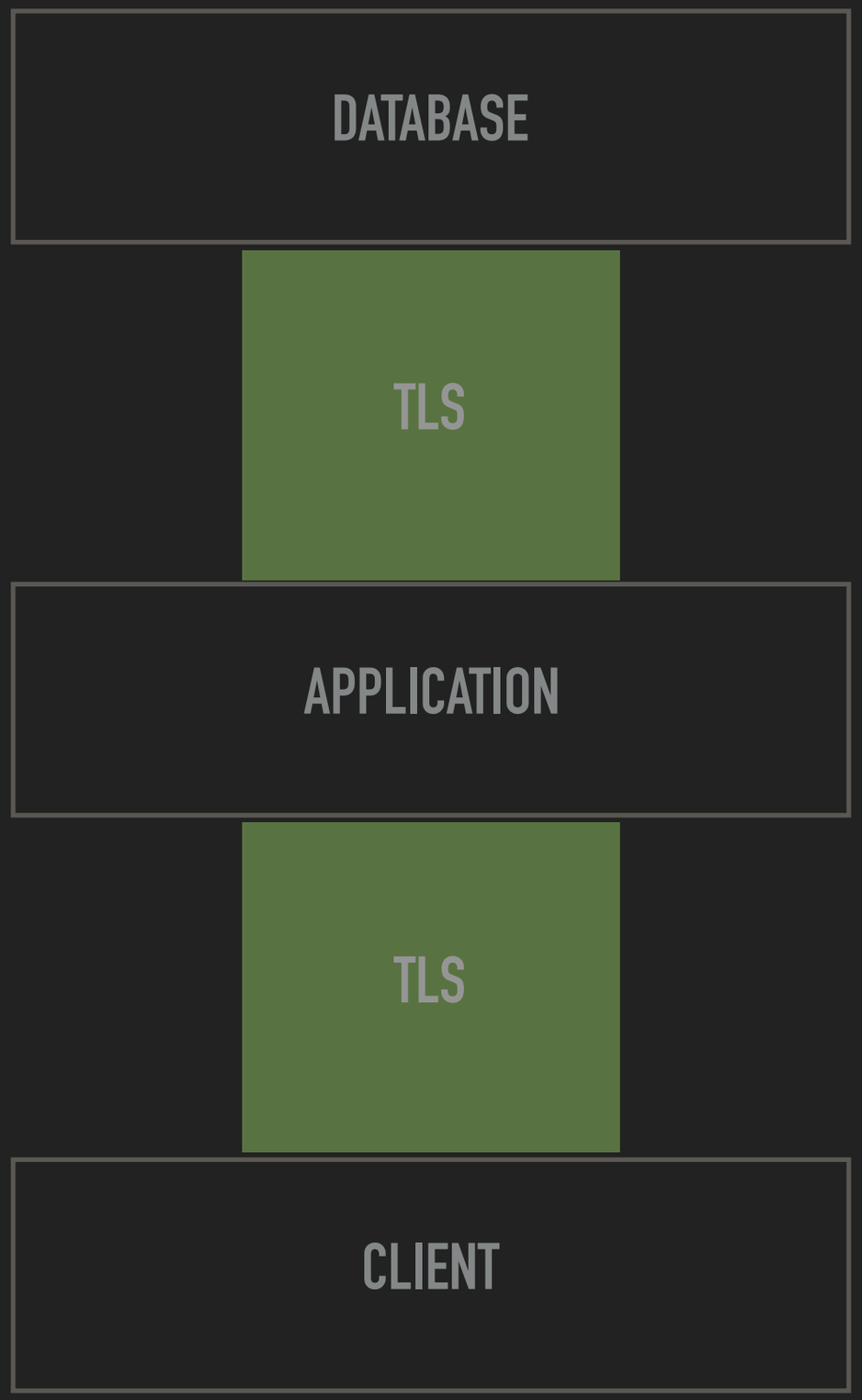
WHAT IS …?

CONTENT ENCRYPTION IS LIKE

AN AIRBAG

YOUR FATHERS CRYPTO (*)

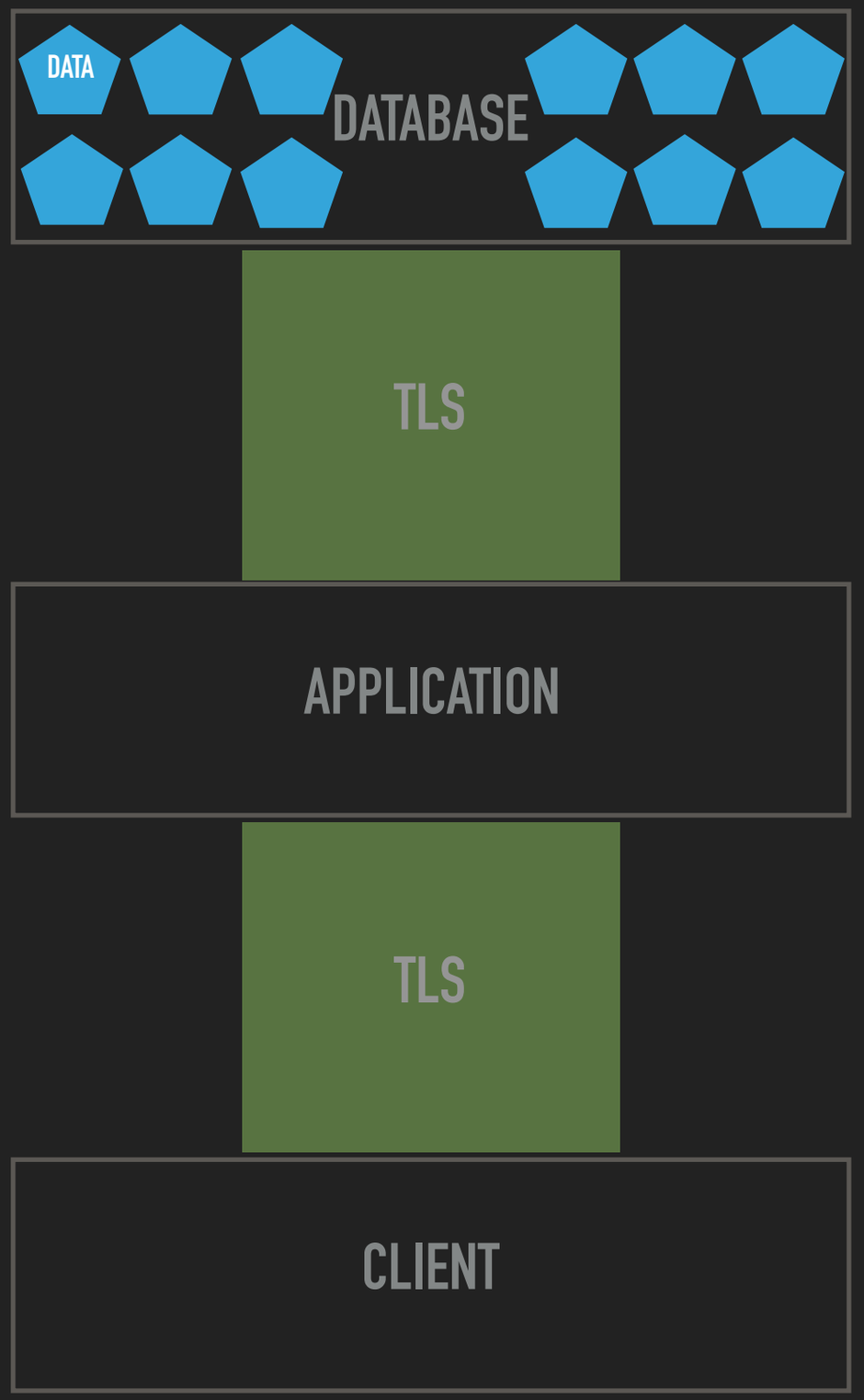
- CLIENT sends request
- APPLICATION applies logic
- DATABASE stores result
- Data encrypted via TLS



(*) I'm going to gloss over the whole cryptography nomenclature in these first slides. Bear with me.

YOUR FATHERS CRYPTO (*)

- CLIENT sends request
- APPLICATION applies logic
- DATABASE stores result
- Data encrypted via TLS

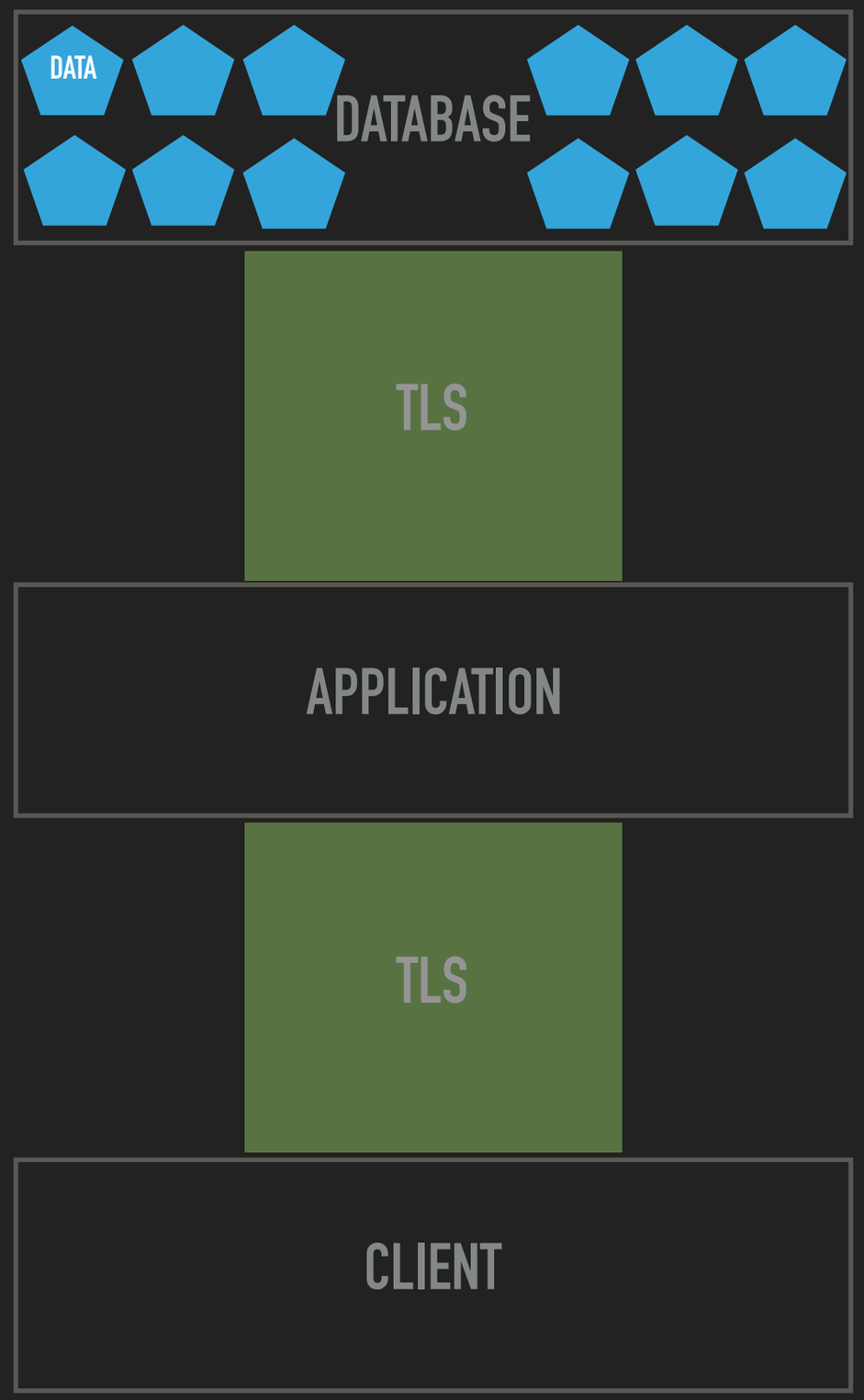


(*) I'm going to gloss over the whole cryptography nomenclature in these first slides. Bear with me.

YOUR FATHERS CRYPTO (*)

- CLIENT sends request
- APPLICATION applies logic
- DATABASE stores result
- Data encrypted via TLS

SO, EVERYTHING IS SAFE?

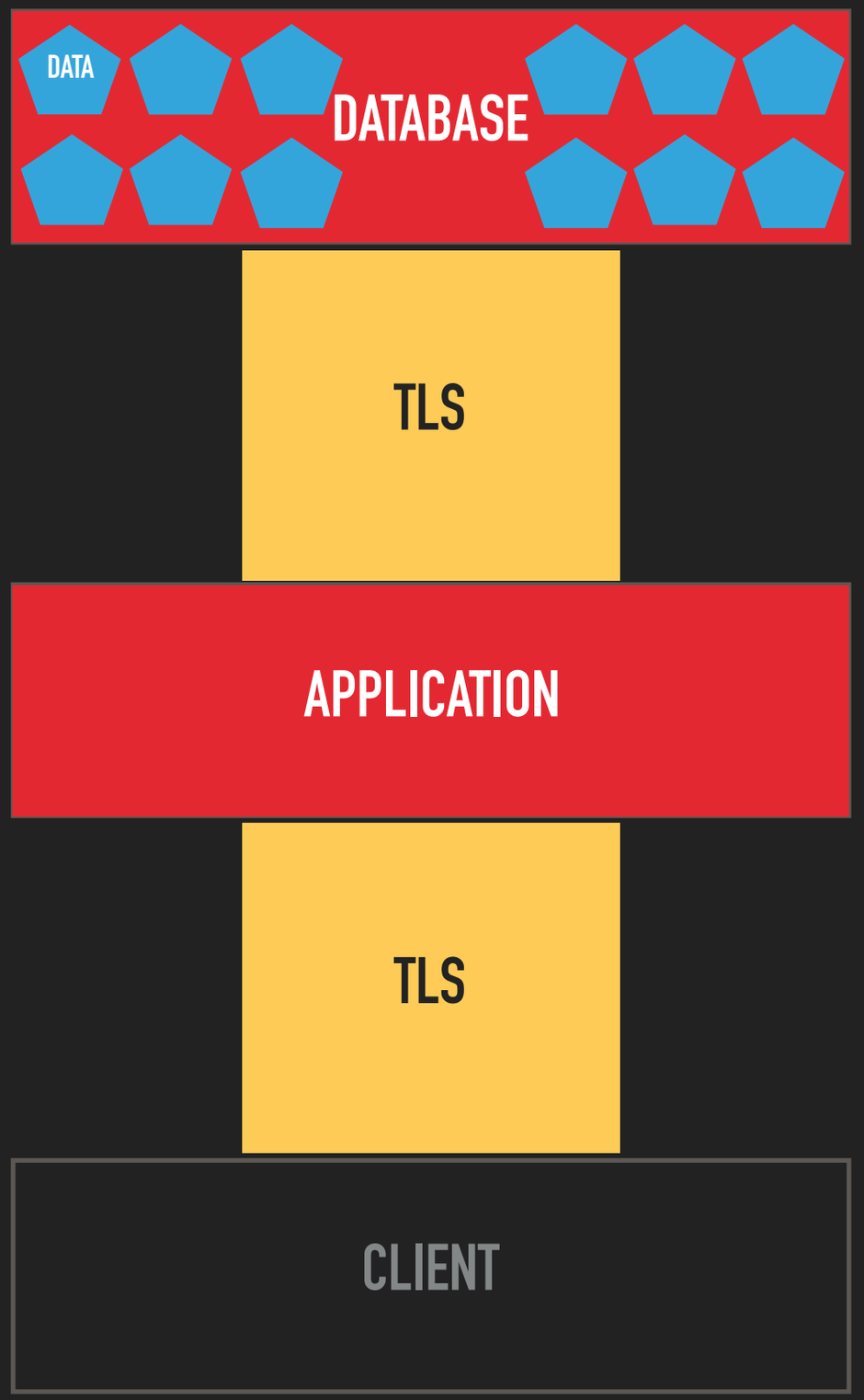


(*) I'm going to gloss over the whole cryptography nomenclature in these first slides. Bear with me.

YOUR FATHERS CRYPTO (*)

- CLIENT sends request
- APPLICATION applies logic
- DATABASE stores result
- Data encrypted via TLS

SO, EVERYTHING IS SAFE?



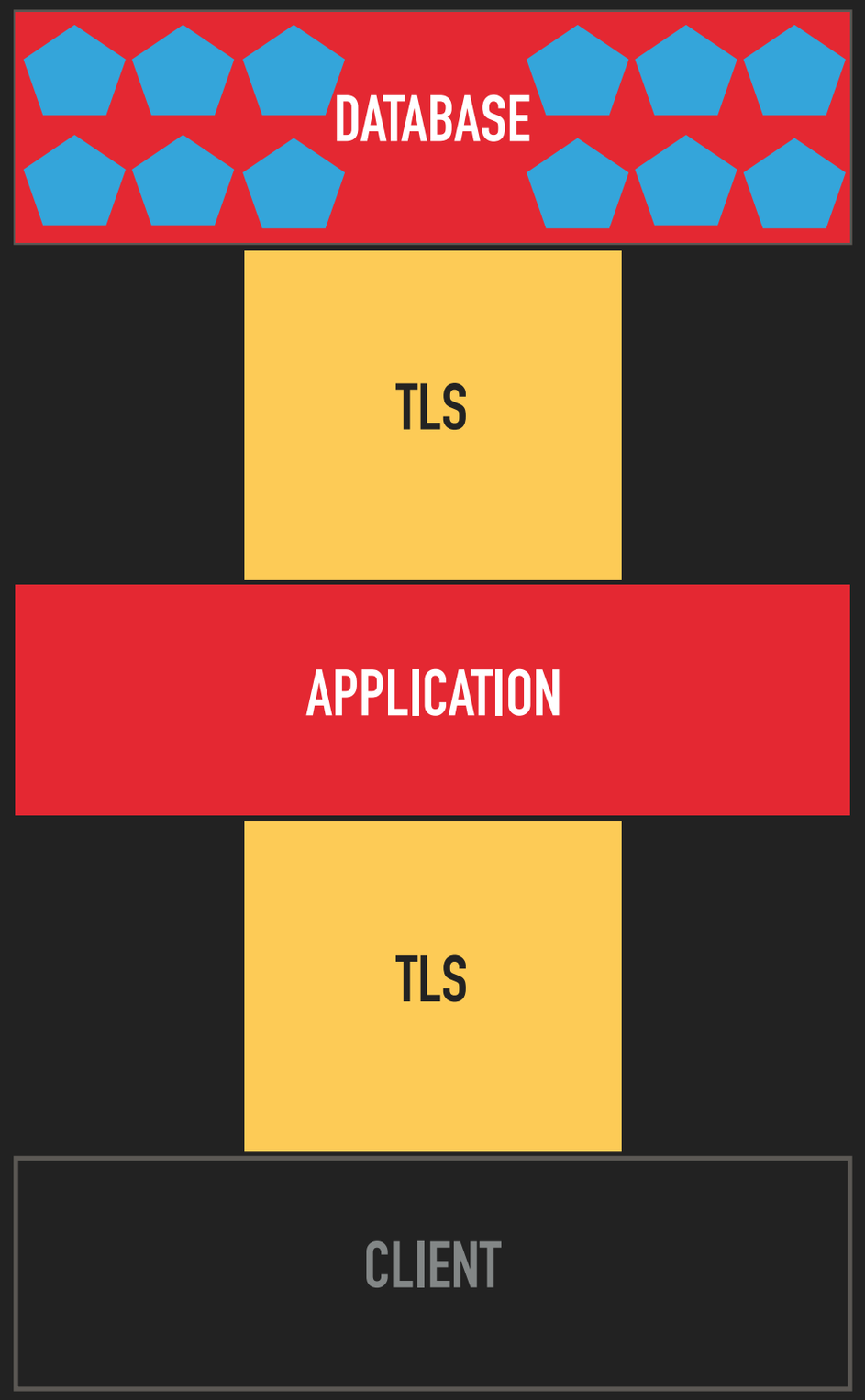
(*) I'm going to gloss over the whole cryptography nomenclature in these first slides. Bear with me.

WHAT ABOUT TLS?

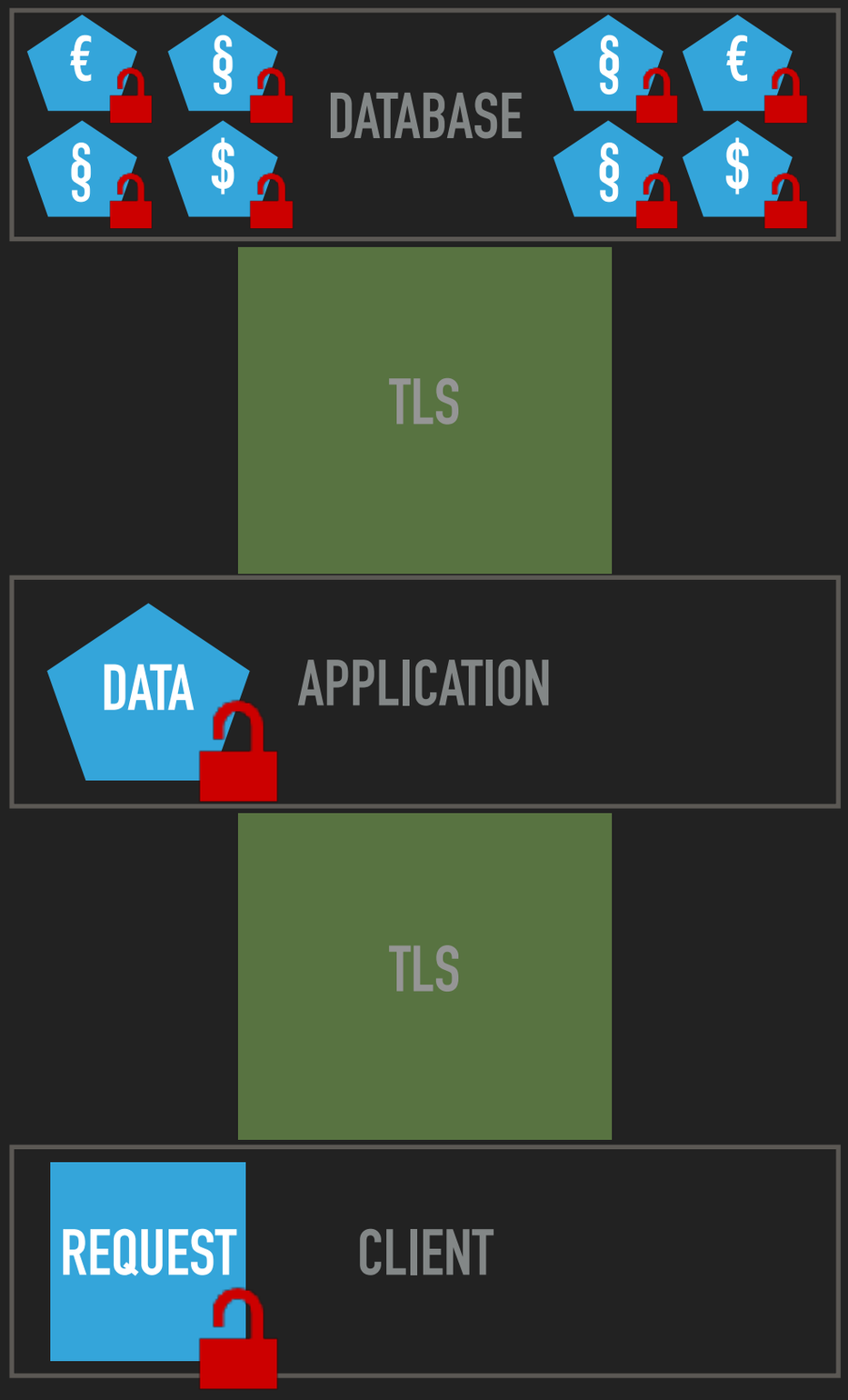
- Data is at rest for ~99.99998% of the time (*)
- Also: Heartbleed, POODLE, DROWN, Lucky13, Logjam, FREAK, …
- Also: Backups!

WHAT IS CONTENT ENCRYPTION?

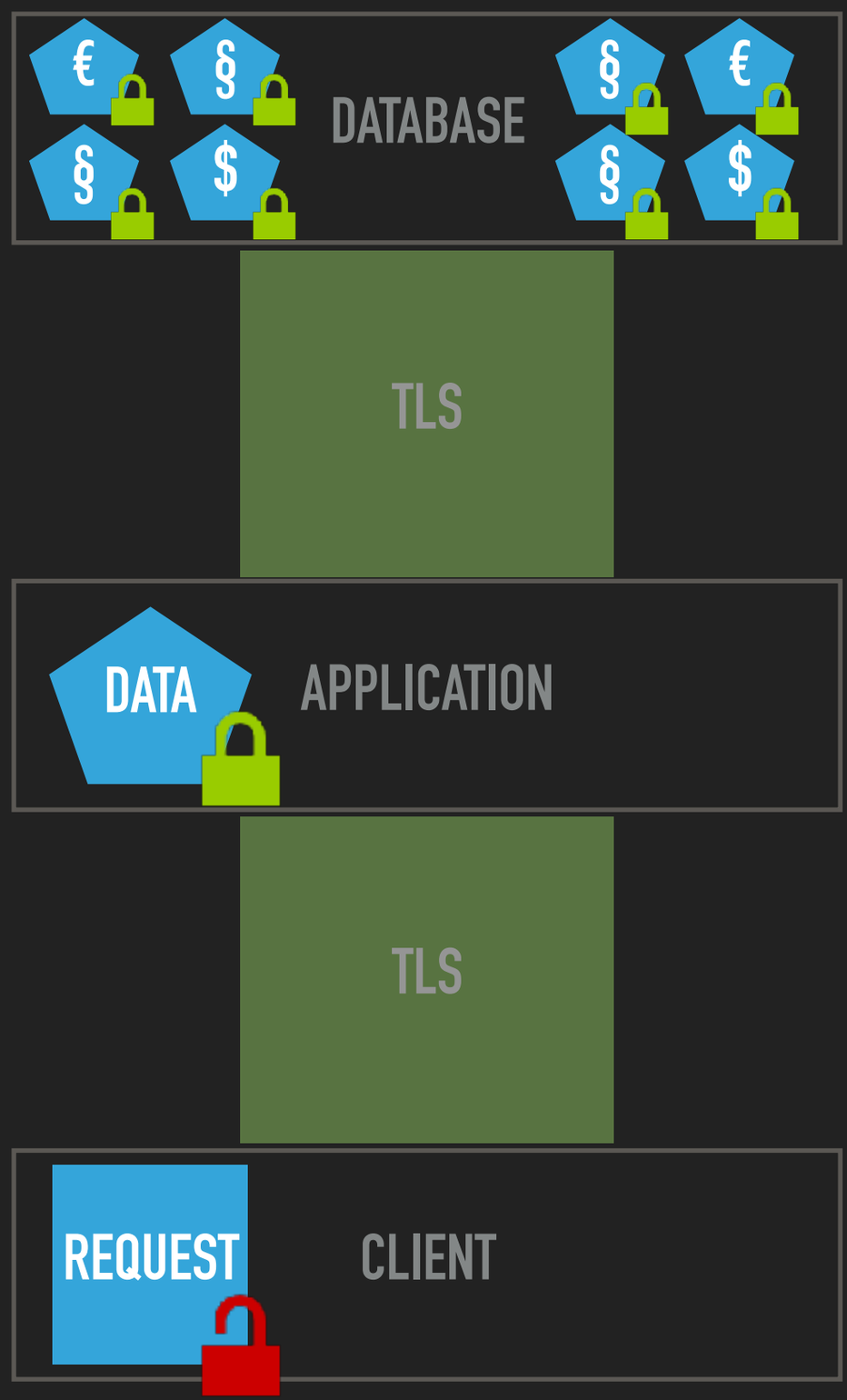
- Encrypt** data ‘itself’
- E.g. encrypted** data at rest
- Even: Protected** data while working with it



(*) 10s transit / 2 years storage (**) Cryptographically protected - details later!


DATABASE
TLS
DATA
APPLICATION
TLS
REQUEST
CLIENT

VS.


DATABASE
TLS
DATA
APPLICATION
TLS
REQUEST
CLIENT

WHY USE

CONTENT ENCRYPTION?

IT'S THE LAW

- A lot of (German) laws take data protection seriously
- Bundesdatenschutzgesetz (BDSG)
- Telemediengesetz (TMG)
- Telekommunikationsgesetz (TKG)
- Strafgesetzbuch (StGB)
- Sozialgesetzbuch (SGB)
- EUDSGV

I AM NOT A LAWYER!

BUNDESDATENSCHUTZGESETZ

Applies to: Everyone working with personal data

- **§ 3 Weitere Begriffsbestimmungen**

(1) Personenbezogene Daten sind Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person (Betroffener).
…
(9) Besondere Arten personenbezogener Daten sind Angaben über die rassische und ethnische Herkunft, politische Meinungen, religiöse oder philosophische Überzeugungen, Gewerkschaftszugehörigkeit, Gesundheit oder Sexualleben.

- **§ 42a Informationspflicht bei unrechtmäßiger Kenntniserlangung von Daten**

Stellt [eine verarbeitende Stelle] fest, dass bei ihr gespeicherte
1. besondere Arten personenbezogener Daten (§ 3 Absatz 9),
2. personenbezogene Daten, die einem Berufsgeheimnis unterliegen,
3. […] strafbare Handlungen oder Ordnungswidrigkeiten […]
4. personenbezogene Daten zu Bank- oder Kreditkartenkonten
unrechtmäßig übermittelt […], und drohen schwerwiegende Beeinträchtigungen für die Rechte oder schutzwürdigen Interessen der Betroffenen […]
[Meldung an Aufsichtsbehörde und Betroffenen, ggfs. halbseitige Anzeige]

BDSG

I AM NOT A LAWYER!

TELEMEDIENGESETZ

Applies to: Everyone providing websites for profit

- **§ 13 Pflichten des Diensteanbieters**

(7) Diensteanbieter haben […] durch technische und organisatorische Vorkehrungen sicherzustellen, dass
1. [Zugriff auf] technischen Einrichtungen möglich ist und
2.diese
a) **gegen Verletzungen des Schutzes personenbezogener Daten …**

Vorkehrungen nach Satz 1 müssen den Stand der Technik berücksichtigen. **Eine Maßnahme nach Satz 1 ist insbesondere die Anwendung eines als sicher anerkannten Verschlüsselungsverfahrens.**

TMG

- **§ 16 Bußgeldvorschriften**

…
(3) Die Ordnungswidrigkeit kann mit einer **Geldbuße bis zu fünfzigtausend Euro** geahndet werden.

I AM NOT A LAWYER!

TELEKOMMUNIKATIONSGESETZ

Applies to: Everyone providing communication services (*)

- **§ 109a Daten- und Informationssicherheit**
  (1) [...] im Fall einer Verletzung [...] unverzüglich [...] BNetzA & BfDI ...] zu benachrichtigen. Ist anzunehmen [...] schutzwürdigen Interessen beeinträchtigt werden, hat der Anbieter [...] zusätzlich die Betroffenen unverzüglich von dieser Verletzung zu benachrichtigen. In Fällen, [...] durch geeignete technische Vorkehrungen gesichert, insbesondere [...] Verschlüsselungsverfahrens gespeichert wurden, ist eine Benachrichtigung nicht erforderlich.

TKG

I AM NOT A LAWYER!

(*) to “non-closed usergroups” E.g. public E-Mail services!

STRAFGESETZBUCH

Applies to: Everyone - here to Doctors, Lawyers, Health insurance,..

- **§ 203 Verletzung von Privatgeheimnissen**

(1) Wer unbefugt ein fremdes Geheimnis, namentlich ein zum persönlichen Lebensbereich gehörendes Geheimnis oder ein Betriebs- oder Geschäftsgeheimnis, offenbart, das ihm als [Berufsgeheimnisträger] anvertraut worden oder sonst bekanntgeworden ist, wird mit Freiheitsstrafe bis zu einem Jahr oder mit Geldstrafe bestraft.

§203 StGB

I AM NOT A LAWYER!

SOZIALGESETZBUCH

Applies to: Everyone working with social data

- You know when it applies!

SGB

I AM NOT A LAWYER!

EU DATENSCHUTZGRUNDVERORDNUNG

Applies to: You (starting May 2018)

- **Art. 32 DSGVO Sicherheit der Verarbeitung**

  Unter Berücksichtigung des Stands der Technik, der Implementierungskosten […] diese Maßnahmen schließen unter anderem Folgendes ein:

  A. die **Pseudonymisierung und Verschlüsselung** personenbezogener Daten;

  B. die Fähigkeit, die **Vertraulichkeit, Integrität**, Verfügbarkeit und Belastbarkeit der Systeme und Dienste im Zusammenhang mit der Verarbeitung auf Dauer sicherzustellen;

  C. …

- Erwägungsgrund 83

DSGVO

I AM NOT A LAWYER!

IT'S COMPLIANCE

- Company rules require encryption
- PCI DSS
- ISO 27001
- ...

NOT CONVINCED YET?

ASK THE COMPETITION!

I grew tired of updating this!

ASK THE COMPETITION PT. 1 (A)

Dido Harding (here still) CEO of Talk Talk

https://www.grahamcluley.com/hacked-talktalk-says-received-ransom-demand/
http://www.bbc.com/news/uk-34611857

ASK THE COMPETITION PT. 1 (A)

This could be YOUR CEO!

Dido Harding (here still) CEO of Talk Talk

https://www.grahamcluley.com/hacked-talktalk-says-received-ransom-demand/
http://www.bbc.com/news/uk-34611857

ASK THE COMPETITION PT. 1 (A)

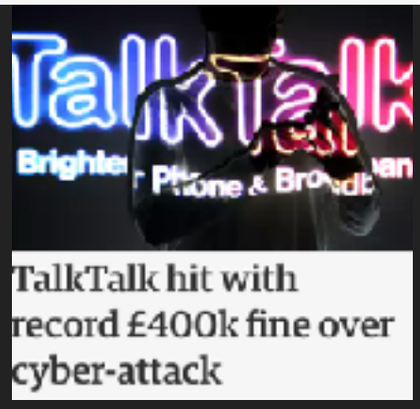
This could be YOUR CEO!

And you will want a happy CEO :-)

Dido Harding (here still) CEO of Talk Talk

https://www.grahamcluley.com/hacked-talktalk-says-received-ransom-demand/
http://www.bbc.com/news/uk-34611857

ASK THE COMPETITION PT. 1 (B)



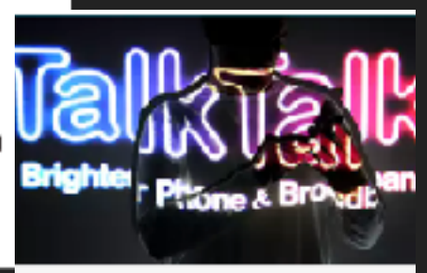
https://www.bloomberg.com/quote/TALK:LN
https://www.theguardian.com/business/2016/may/12/talktalk-profits-halve-hack-cyber-attack

ASK THE COMPETITION PT. 1 (C)

TalkTalk has been fined a record £400,000 fine for security failings which led to the theft of personal data of almost 157,000 customers.

The cyber attack in October last year exposed the latest security failure for the company, which was forced to admit it had not encrypted some personal details of customers.

The Information Commissioner's Office (ICO) said the attack could have been prevented if TalkTalk had taken basic steps to protect customers' information.



TalkTalk hit with record £400k fine over cyber-attack

https://www.independent.co.uk/news/business/news/talktalk-fine-data-breach-theft-customers-information-stolen-record-penalty-a7346316.html
https://www.theguardian.com/business/2016/feb/02/talktalk-cyberattack-costs-customers-leave

ASK THE COMPETITION PT. 1 (D)

TalkTalk chief executive Dido Harding to step down

Has TalkTalk's security been breached yet again?

Customers claim scammers have fresh details of their accounts – and even a new router password

https://www.theguardian.com/business/2017/feb/01/talktalk-chief-executive-dido-harding-cyber-attack
https://www.theguardian.com/money/2017/mar/11/talktalk-security-breached-again-scammers-india

ASK THE COMPETITION PT. 2

Adultery Website AshleyMadison Seeks IPO as Demand Booms

Kristen Schweizer
15 April 2015, 13:27 CEST

before

after

A data dump, 9.7 gigabytes in size, was posted on Tuesday to the dark web using an Onion address accessible only through the Tor browser. The files appear to include account details and log-ins for some 32 million users of the social networking site, touted as the premier site for married individuals seeking partners for affairs. Seven years worth of credit card and other payment transaction details are also part of the dump.

In August 2015, after its customer records were leaked by hackers, a $576 million class-action lawsuit was filed against the company.[48]

https://www.bloomberg.com/news/articles/2015-04-15/adultery-website-ashleymadison-seeks-ipo-as-demand-booms

https://www.wired.com/2015/08/happened-hackers-posted-stolen-ashley-madison-data/

https://en.wikipedia.org/wiki/Ashley_Madison

ASK THE COMPETITION PT. 3

Breached Organizations Lose Millions in Market Value, Finds New Report

> *"For a typical FTSE 100 firm the impact of 1.8 per cent equates to a permanent loss of market capitalization of £120 million," explains the report – which amounts close to $150 million USD.*

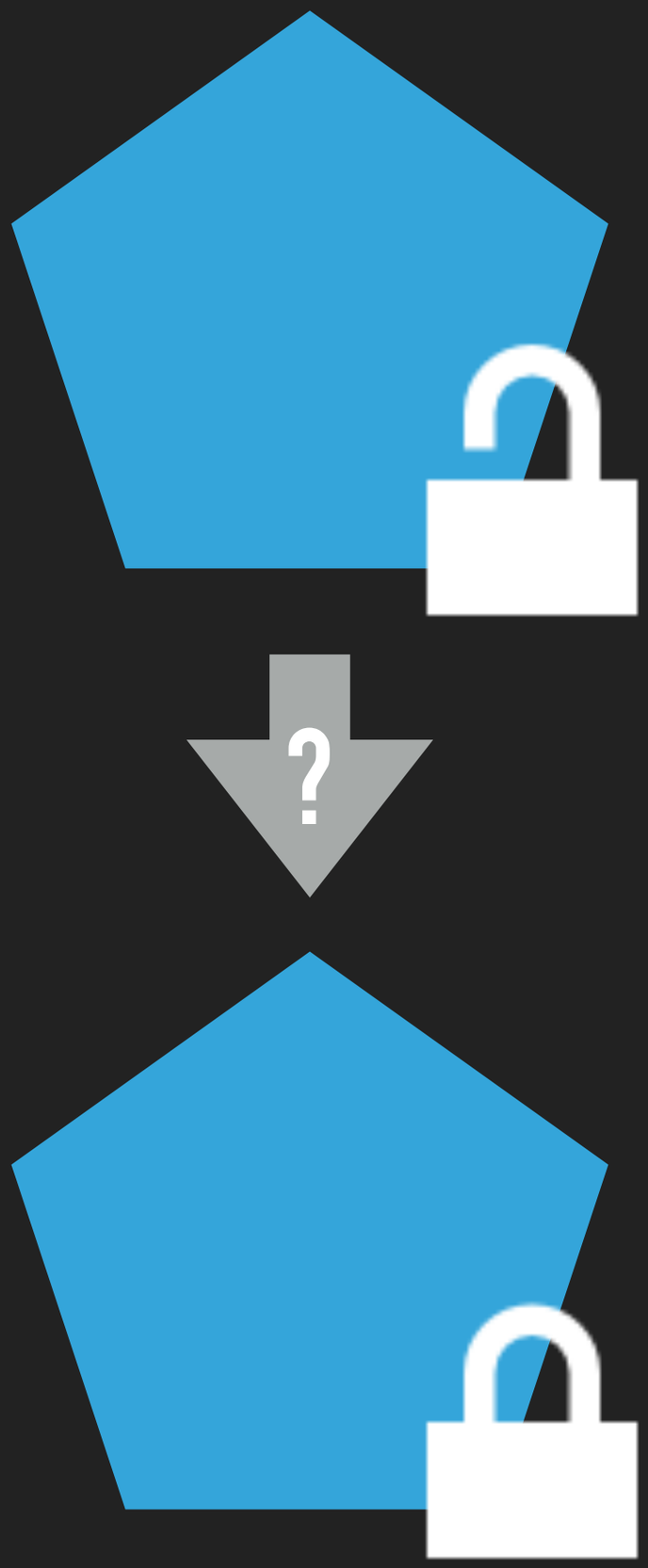
> *"Lost shareholder value across European markets could rise by as much as a factor of 10 when the new regulations take effect in May 2018," Rogoyski told Infosecurity Magazine.*

https://www.tripwire.com/state-of-security/latest-security-news/breached-organizations-lose-millions-marke
value-finds-new-report/

http://breachlevelindex.com/

ASK THE COMPETITION PT. 4

https://mobile.twitter.com/CNBCnow/status/992133162347237376



HOW TO DESIGN

CONTENT ENCRYPTION

0 – REGULATIONS & INITIAL RISK ASSESSMENT

1 – DATA CLASSIFICATION

2 – DATA TREATMENT PLAN

3 – IMPLEMENTATION PLAN

0 – REGULATIONS & INITIAL RISK ASSESSMENT

1. List all relevant **laws, regulations**, etc. with a legal expert
2. List all **data items** (“Adress”, “Bank Account”, …)
3. Estimate **initial risks** (damage & probability) for CIA violations with management and legal expert

1 – DATA CLASSIFICATION

Classify all data items with respect to regulations

Item	Regulation	
Customer		
- Name	BDSG	Pers. bez. Datum
- Bank Account	BDSG	bes. PbD
…	…	…
Marketing E-Mail		
- Recipient	BDSG	Pers. bez. Datum
- Text	???	???
- Protocol	???	???

THIS DEPENDS ON YOUR SPECIFIC APPLICATION/SCENARIO!

I AM NOT A LAWYER!

1 – DATA CLASSIFICATION

Classify all data items with respect to regulations

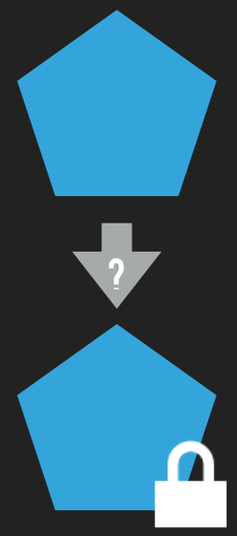
Item	Regulation	
Customer		
- Name	BDSG	Pers. bez. Datum
- Bank Account	BDSG	bes. PbD
...	...	...
Marketing E-Mail		
- Recipient	BDSG	Pers. bez. Datum
- Text	???	???
- Protocol	???	???

Classification depends on context!

THIS DEPENDS ON YOUR SPECIFIC APPLICATION/SCENARIO!

I AM NOT A LAWYER!

2 – DATA TREATMENT PLAN



Create a data treatment plan, and for each data item

- describe how (*) this data must be
- **... stored**
- **... transmitted**
- **... logged**
- **... backed up**
- and when it must be **deleted**!

(*) how: plain, masked (“XXXXX 123”), pseudonymised, anonymised, encrypted, hashed.

Also: consider integrity protection.

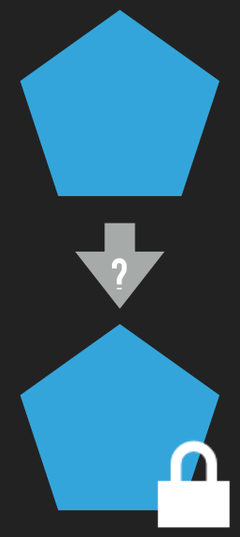
Item	Store	Transmit	Log	Backup	Delete
Customer					
- Name	Plain	Encrypted	Pseud.	Encrypted	BDSG
- Bank	Encrypted	Encrypted	Masked	Encrypted	BDSG
…	…	…			
Marketing E-Mail					
- Recipient	Plain	Encrypted	Pseud.	Encrypted	BDSG
- Text	???	???			
- Protocol	???	???			

Also: consider integrity protection.

THIS DEPENDS ON YOUR SPECIFIC APPLICATION/SCENARIO!

I AM NOT A LAWYER!

3 – IMPLEMENTATION PLAN



Update the requirements

- Merge Use Cases & data treatment plan
- Add cryptographic use cases

LONGER STORY: WRITE A CRYPTOGRAPHY & SECURITY CONCEPT

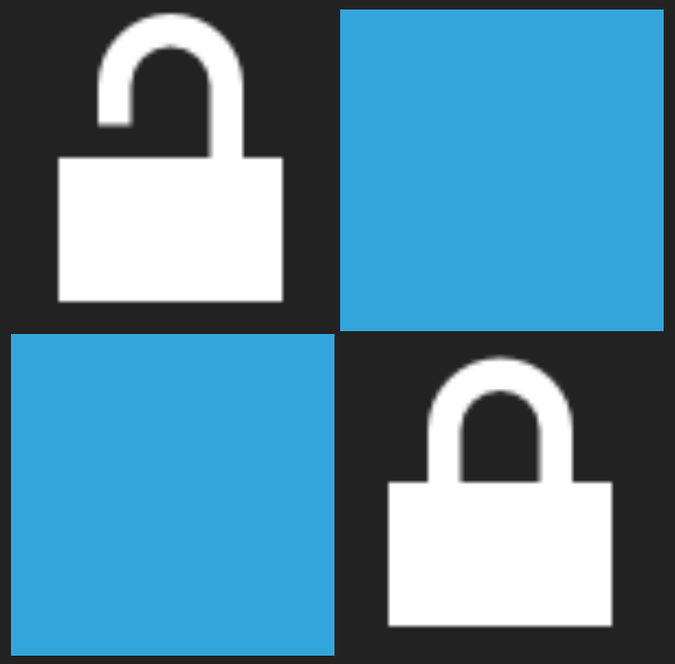
RECAP

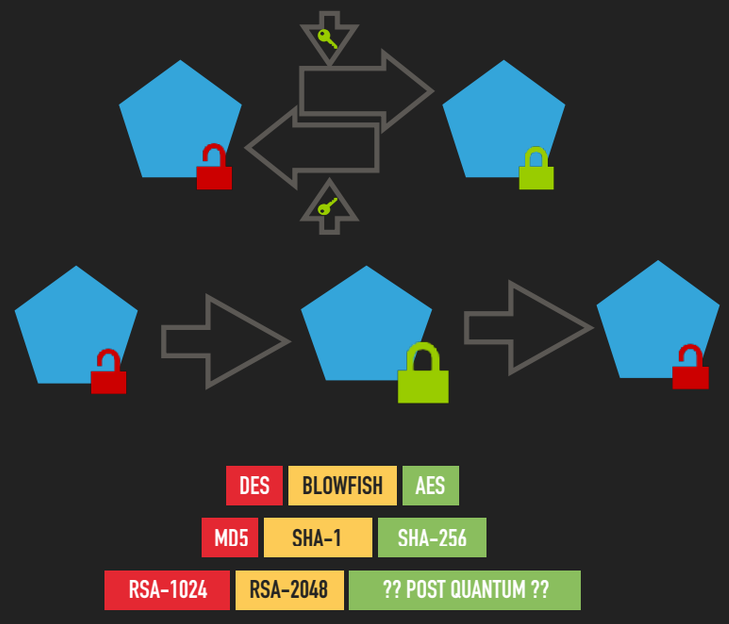
0 - Regulations & initial risk assessment ✓

1 - Data classification ✓

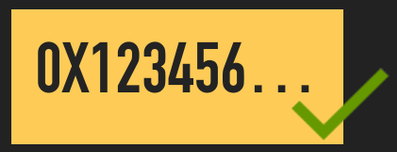
2 - Data treatment plan ✓

3 - Implementation Plan ✓



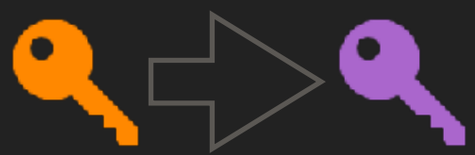


- [x] Data treatment …
- [x] Use existing …
- [x] …

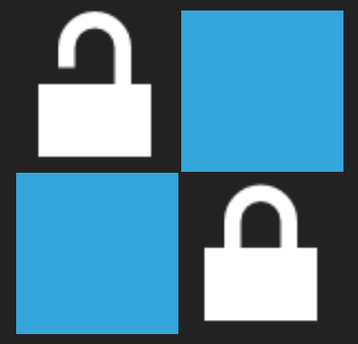


CONTENT ENCRYPTION

PATTERNS



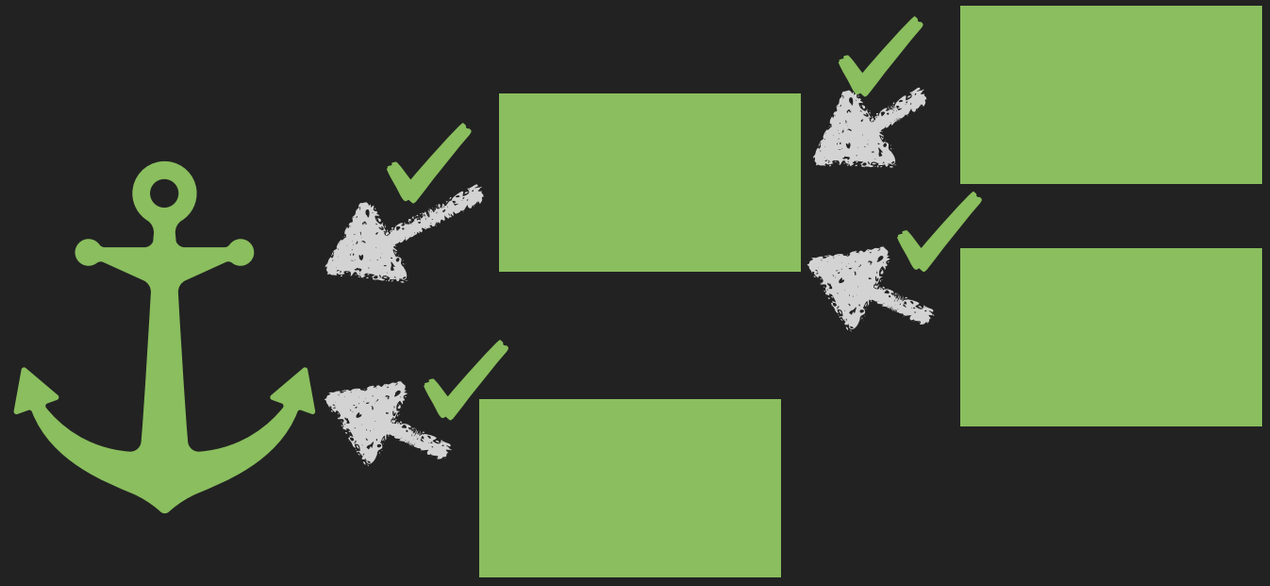

```
int getRandomNumber()
{
    return 4;  // chosen by four dice roll.
               // guaranteed to be random.
}
```



MOST IMPORTANT ADVICE: GET AN EXPERT OR AT LEAST READ AND UNDERSTAND THE DOCUMENTATION!

Like all power tools: Better RTFM than to lose an eye!

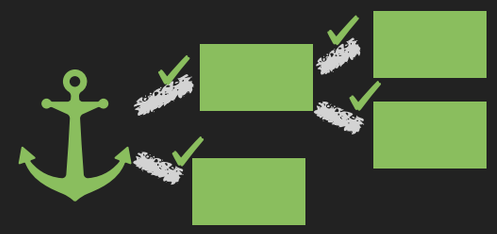
- At least be able to explain: “Hash vs. encryption”, “Integrity vs. encryption”, “Stream vs. block”, “Mode of operation”, “IV”, “Nonce”, “Padding”, “Key derivation”
- Identify and name your trust anchors



PATTERNS

TRUST ANCHORS

TRUST ANCHORS



Problem: You do not know when to start trusting / stop discussing

Solution: Define your trust anchors

Reasoning about the security of a system is impossible without

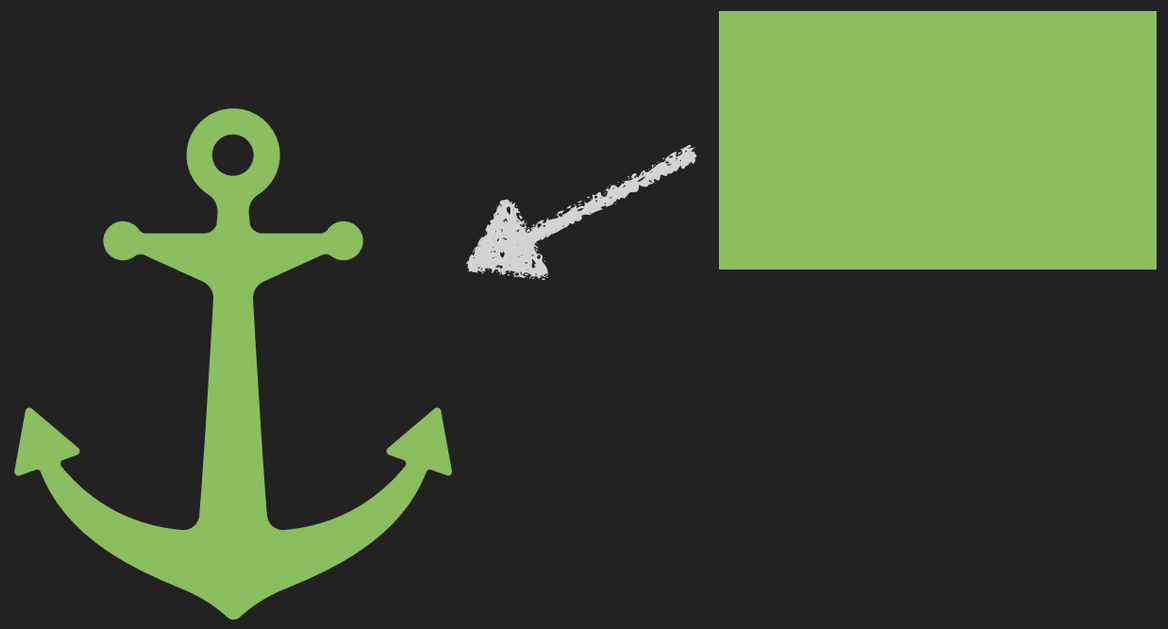
- knowing what you want to protect against (threat model)
- knowing what you can ultimately rely upon (trust anchor)

** Threat modelling not discussed here but it is really important!*

TRUST ANCHORS

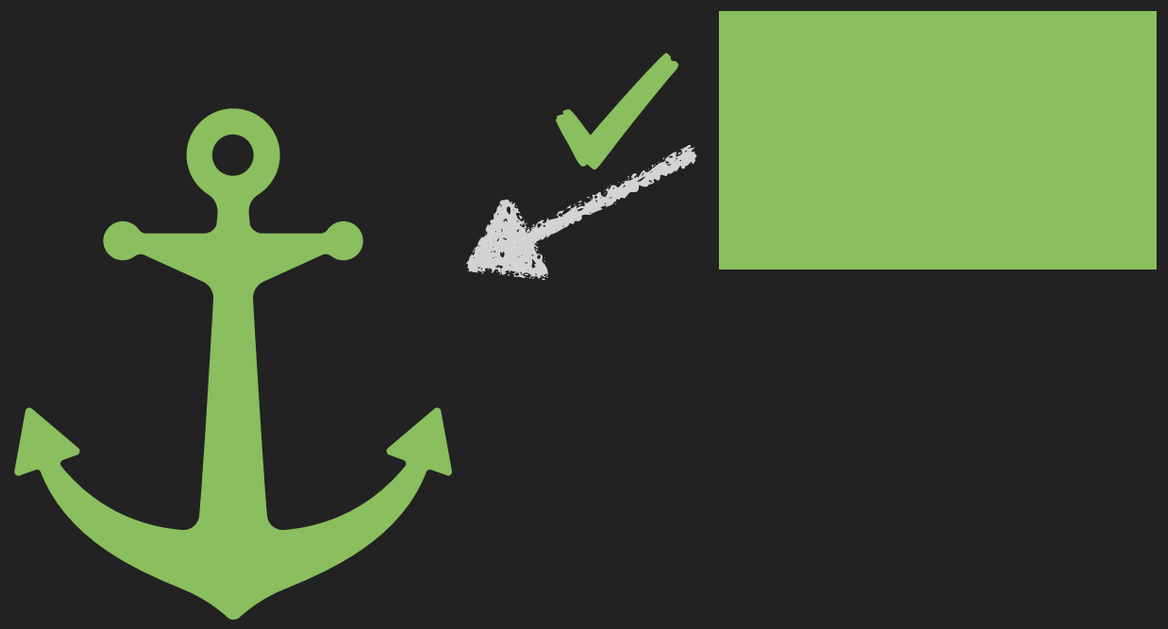
1. You trust your trust anchor(s) – they are secure by definition

TRUST ANCHORS



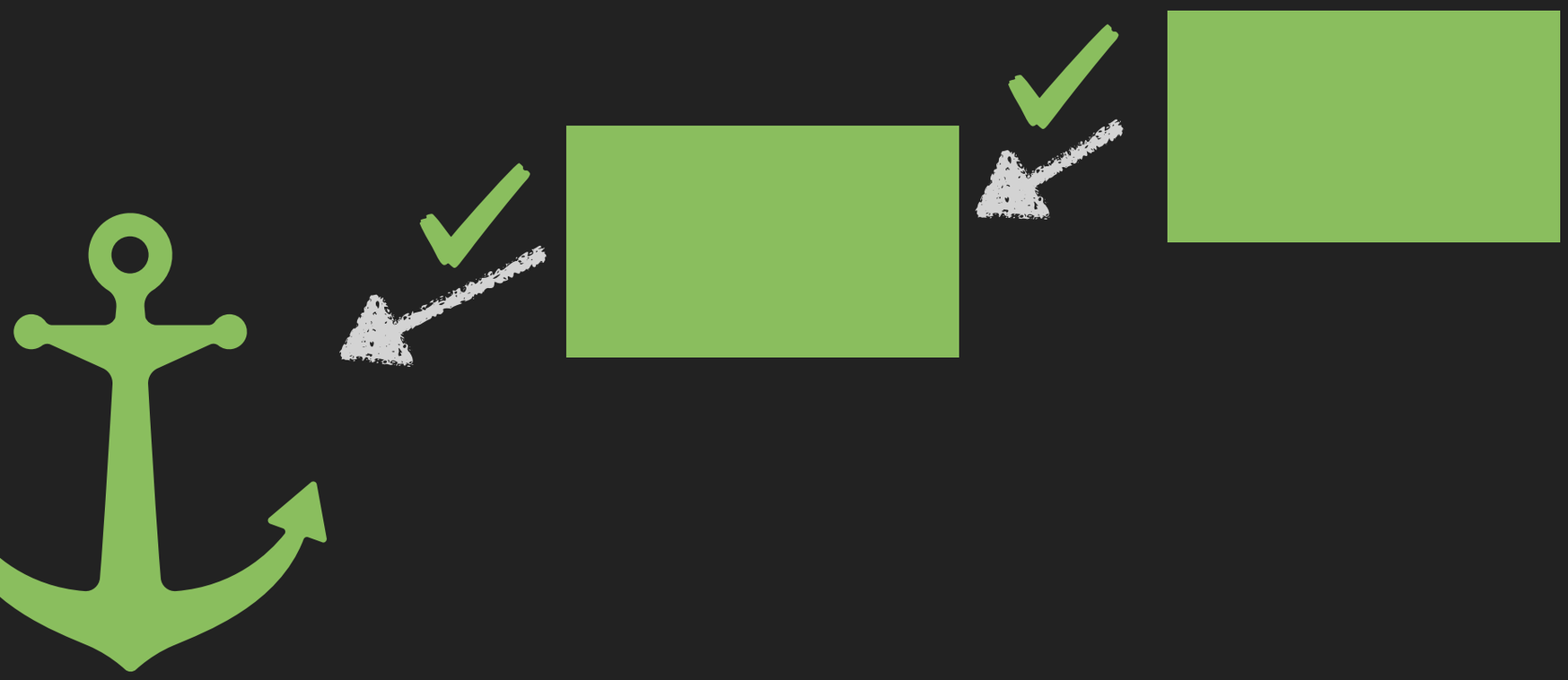
1. You trust your trust anchor(s) – they are secure by definition
2. You reason that something else is also trusted by relying on the trust anchor(s)

TRUST ANCHORS



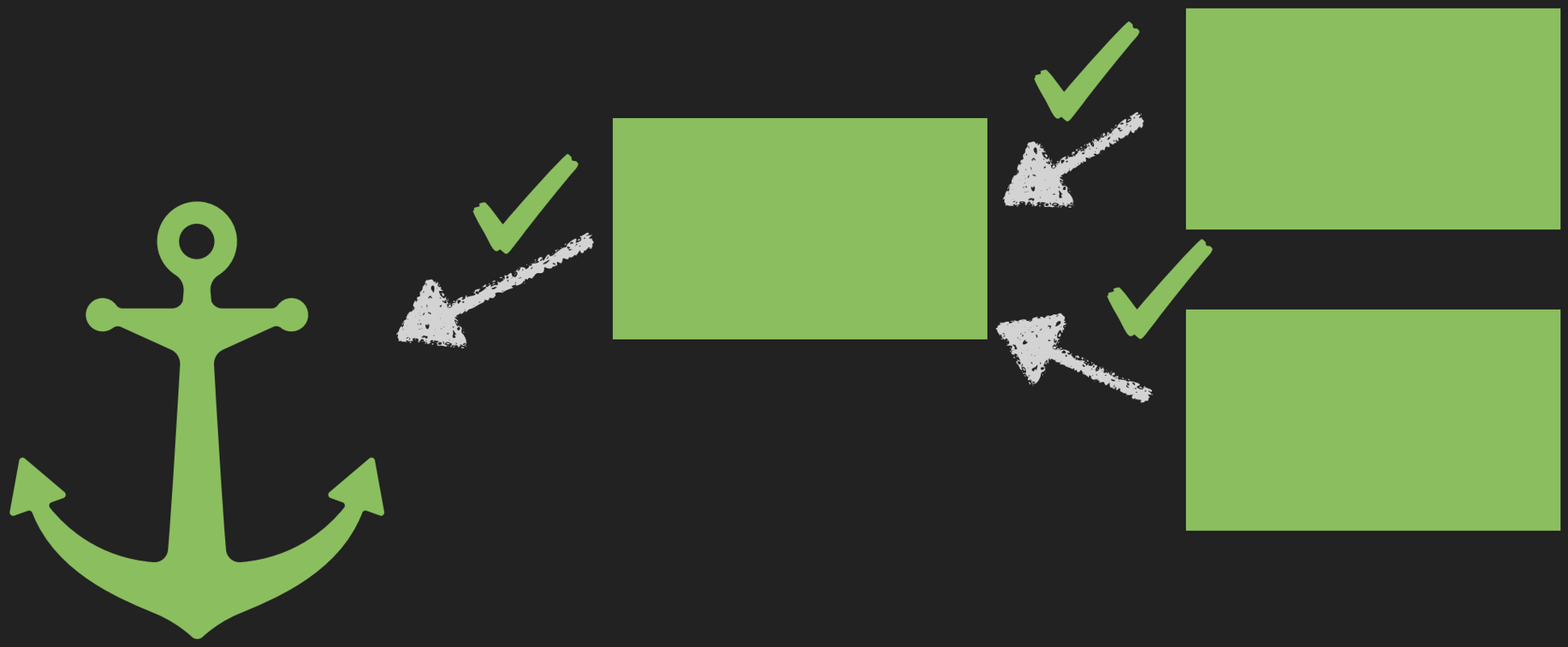
1. You trust your trust anchor(s) – they are secure by definition
2. You reason that something else is also trusted by relying on the trust anchor(s)
3. Therefore trust can be extended

TRUST ANCHORS



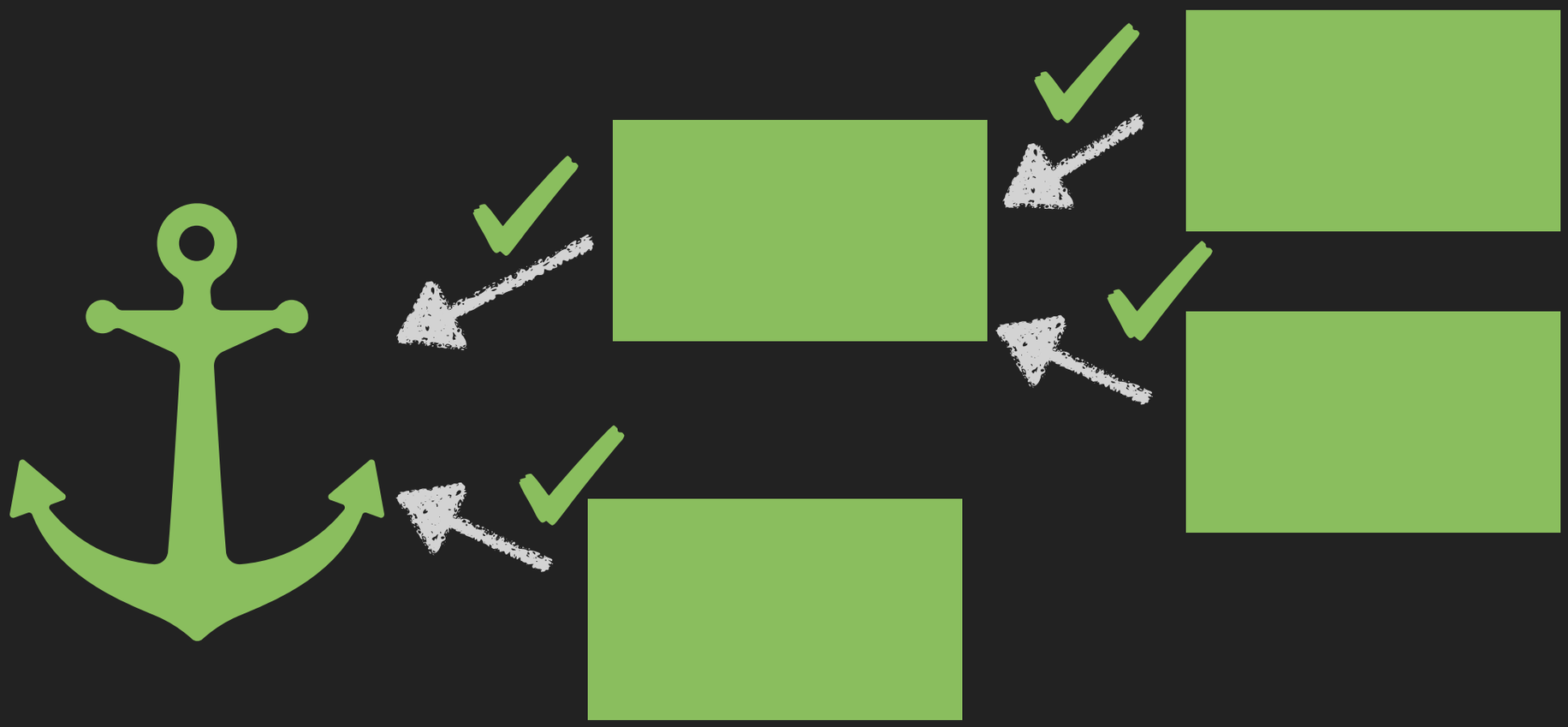
1. You trust your trust anchor(s) – they are secure by definition
2. You reason that something else is also trusted by relying on the trust anchor(s)
3. Therefore trust can be extended

TRUST ANCHORS



1. You trust your trust anchor(s) – they are secure by definition
2. You reason that something else is also trusted by relying on the trust anchor(s)
3. Therefore trust can be extended

TRUST ANCHORS

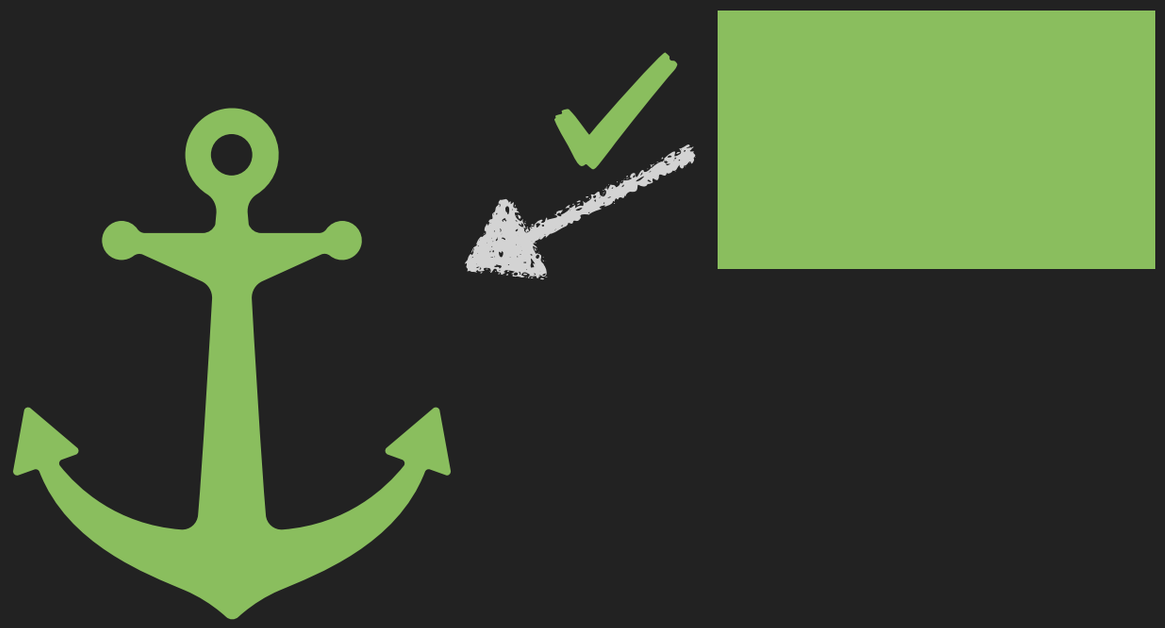


1. You trust your trust anchor(s) – they are secure by definition
2. You reason that something else is also trusted by relying on the trust anchor(s)
3. Therefore trust can be extended

TRUST ANCHORS: EXAMPLE

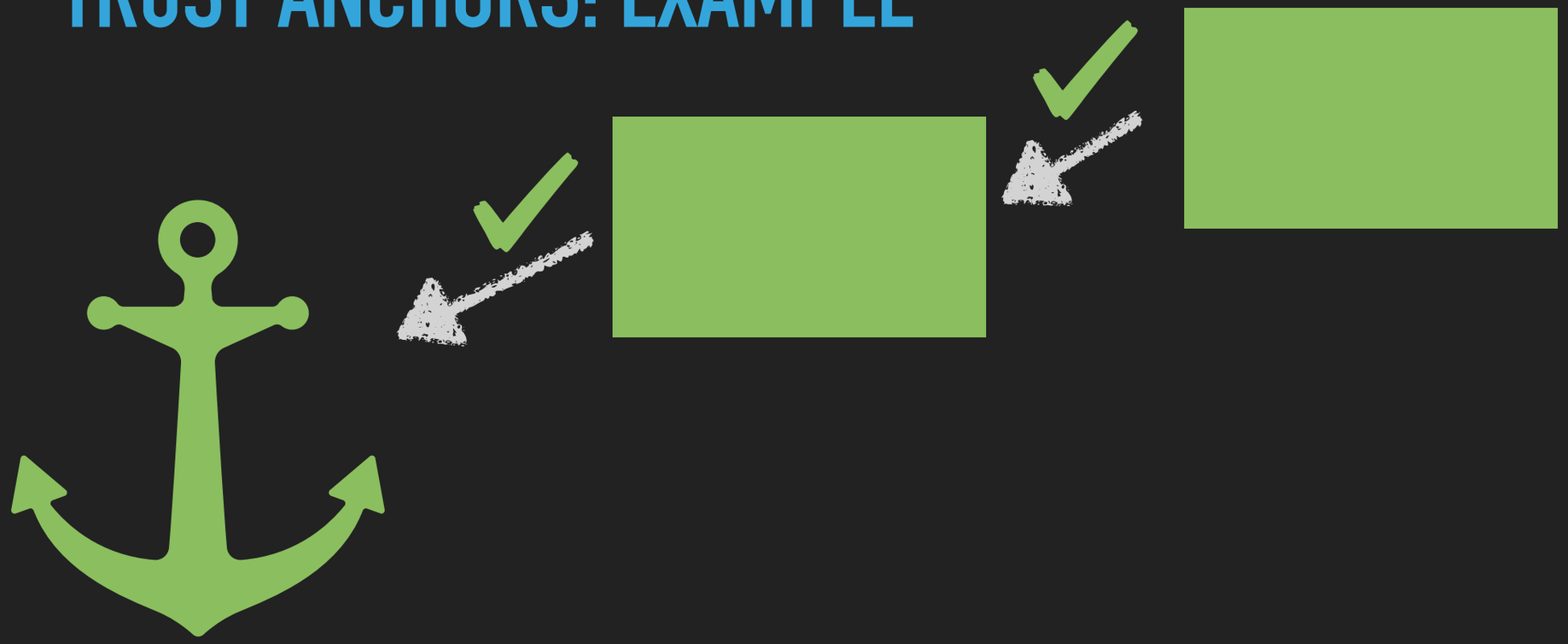
1. I trust my cloud provider

TRUST ANCHORS: EXAMPLE



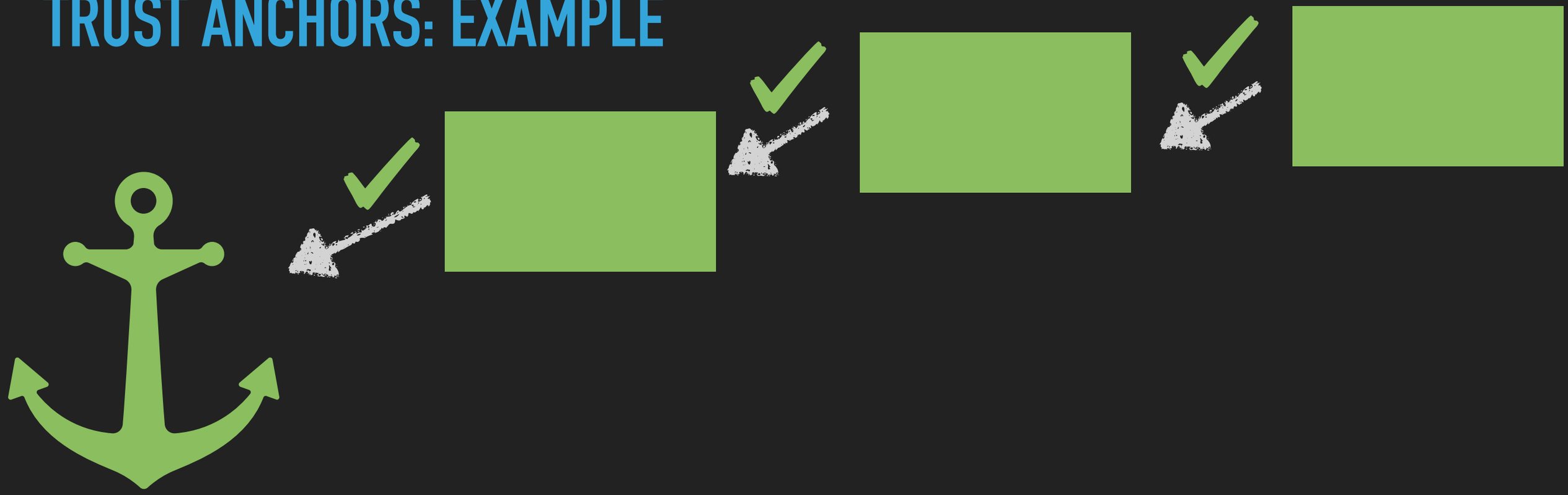
1. I trust my cloud provider
2. I trust that my cloud provider has a secure key storage

TRUST ANCHORS: EXAMPLE



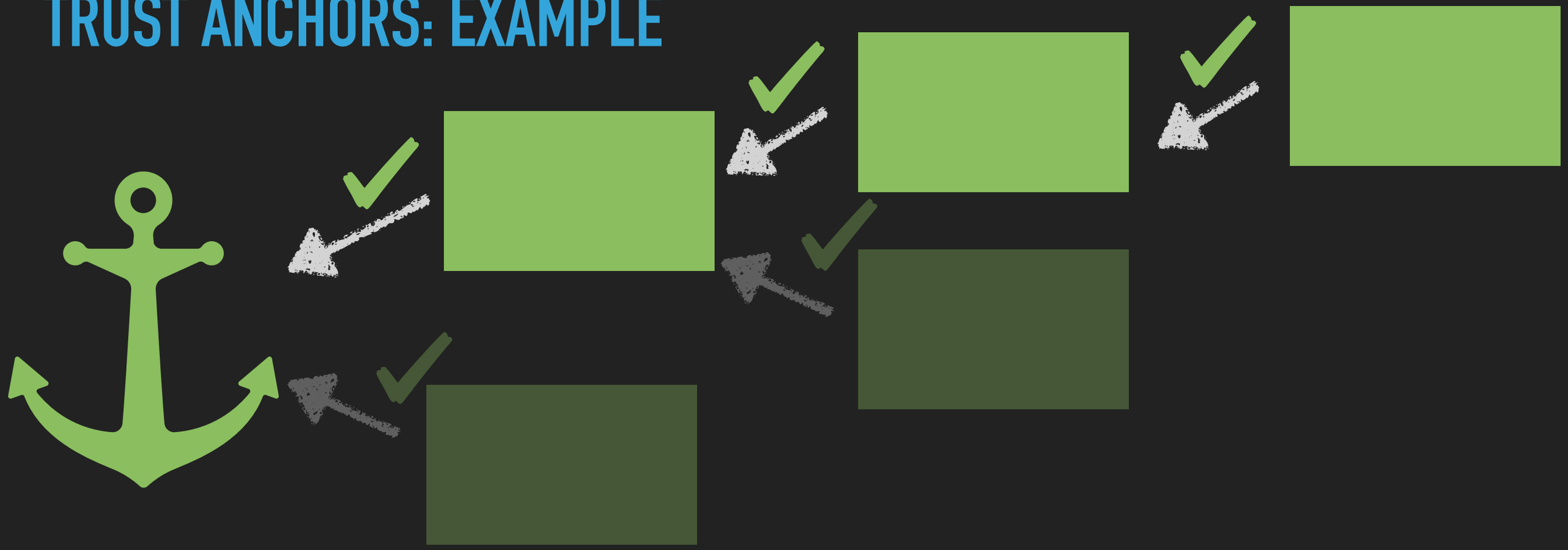
1. I trust my cloud provider
2. I trust that my cloud provider has a secure key storage
3. Data encrypted with keys stored in the keyring is secure

TRUST ANCHORS: EXAMPLE



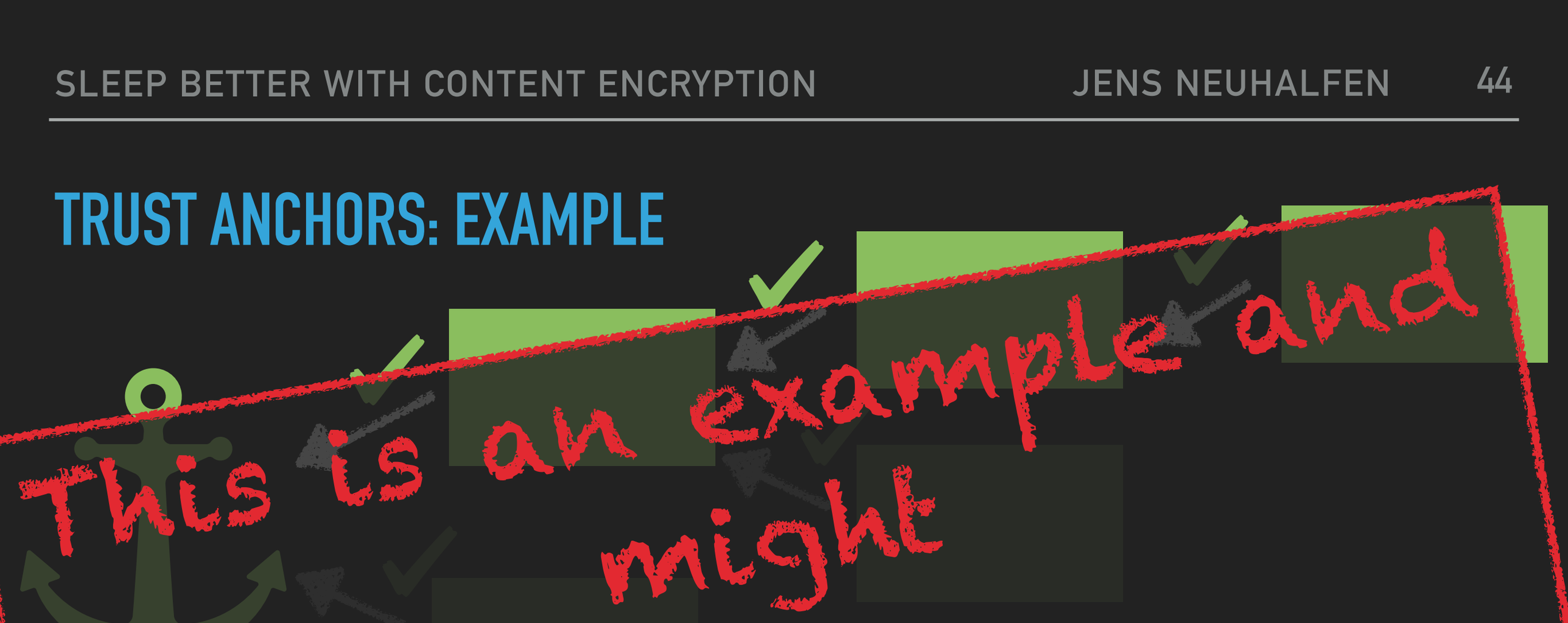
1. I trust my cloud provider
2. I trust that my cloud provider has a secure key storage
3. Data encrypted with keys stored in the keyring is secure
4. Therefore I can store sensitive data in my cloud

TRUST ANCHORS: EXAMPLE



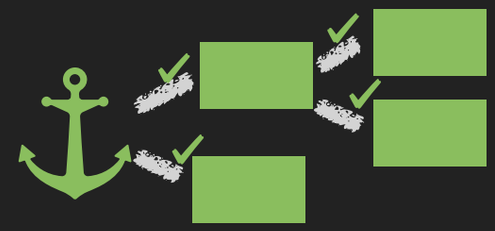
1. I trust my cloud provider
2. I trust that my cloud provider has a secure key storage
3. Data encrypted with keys stored in the keyring is secure
4. Therefore I can store sensitive data in my cloud

TRUST ANCHORS: EXAMPLE

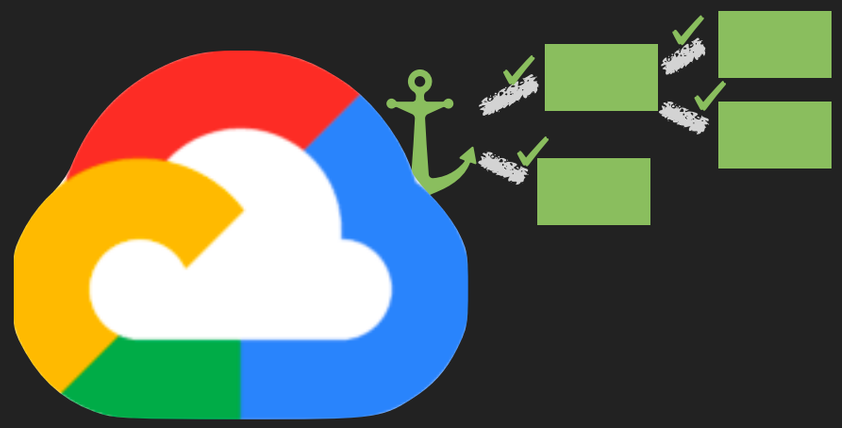


1. I trust my cloud provider
2. I trust that my cloud provider has a secure key storage
3. Data encrypted with keys stored in the keyring is secure
4. Therefore I can store sensitive data in my cloud

TRUST ANCHORS



TRUST ANCHORS



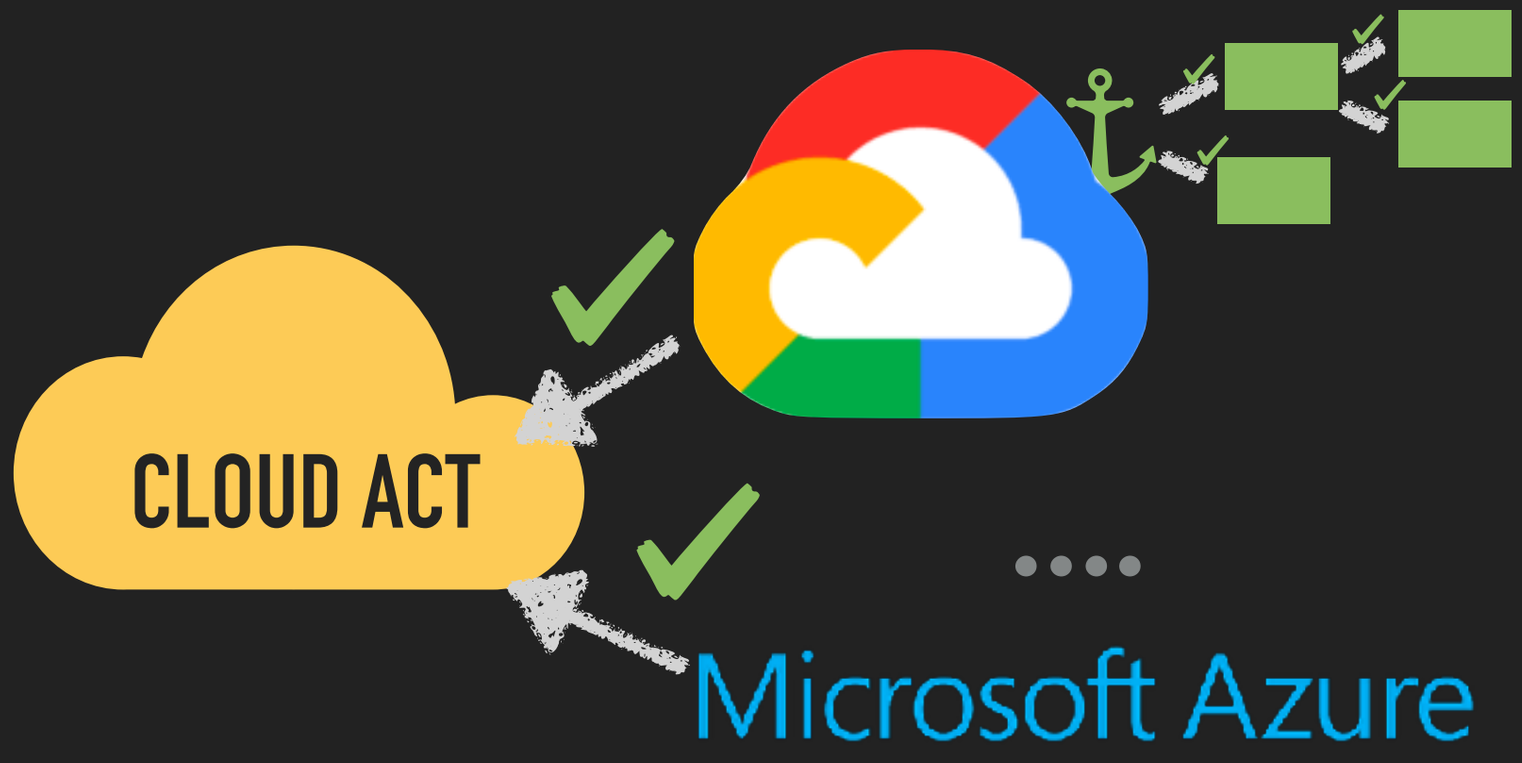
....

Microsoft Azure

....

any other ....

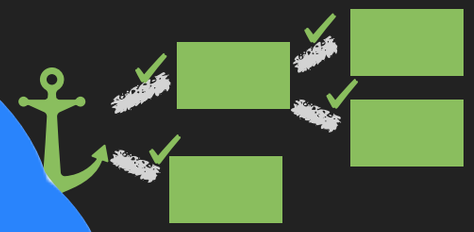
TRUST ANCHORS



....

any other ....

TRUST ANCHORS



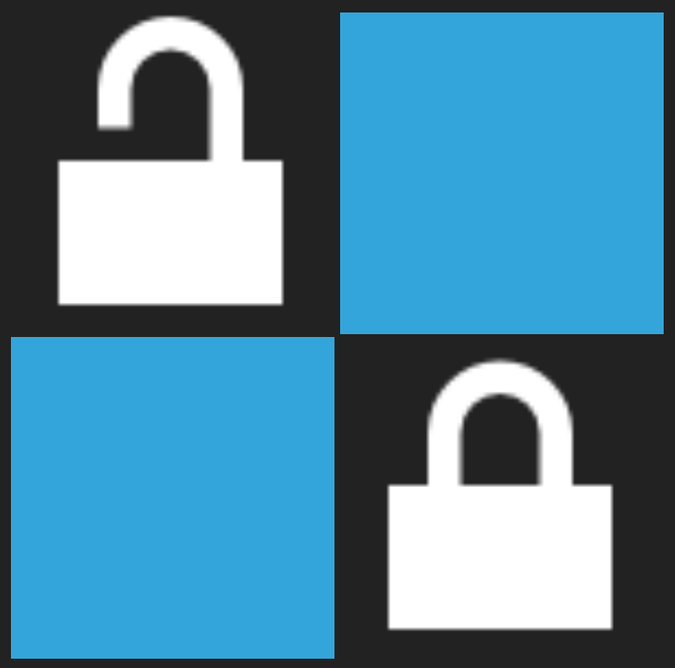
....

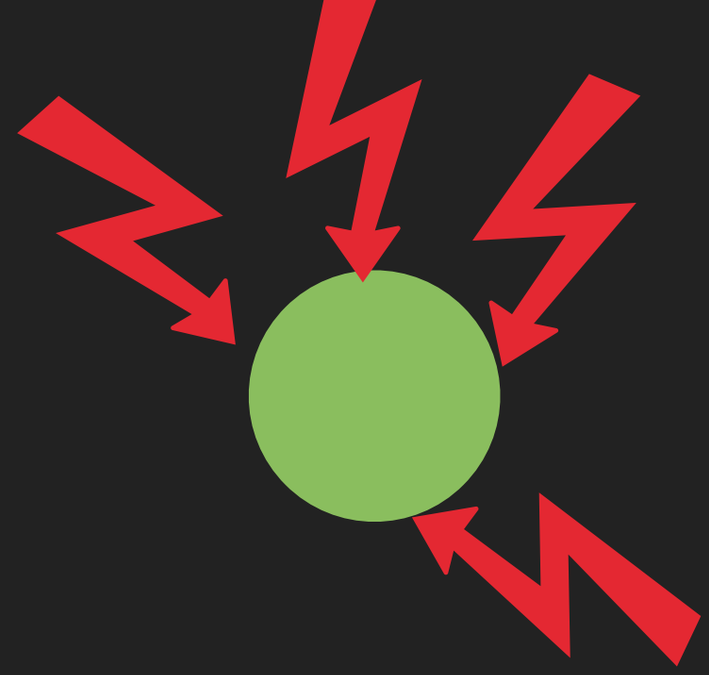
Microsoft Azure

....

any other ....

- http://time.com/collection/most-influential-people-2018/5217621/donald-trump-2/
- https://cloud.google.com/
- https://commons.wikimedia.org/wiki/File:Windows_Azure_logo.png

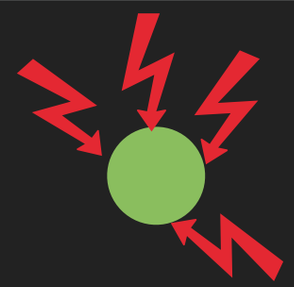




PATTERNS

THREAT MODEL

THREAT MODEL



Problem: You need to know what to protect against

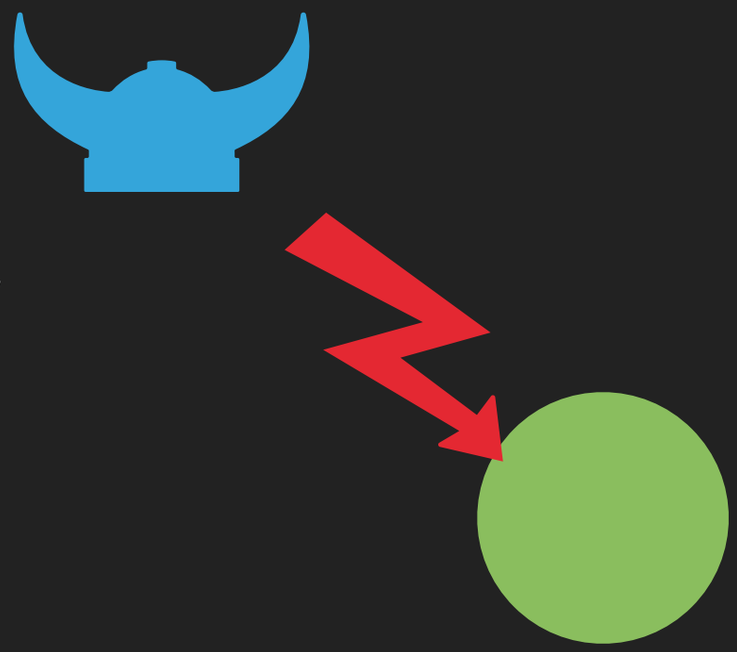
Solution: Write down your threat model

- Model the **technical side** (classical STRIDE etc)
- Model the **business side** of your situation

Also see “Trust Anchors”

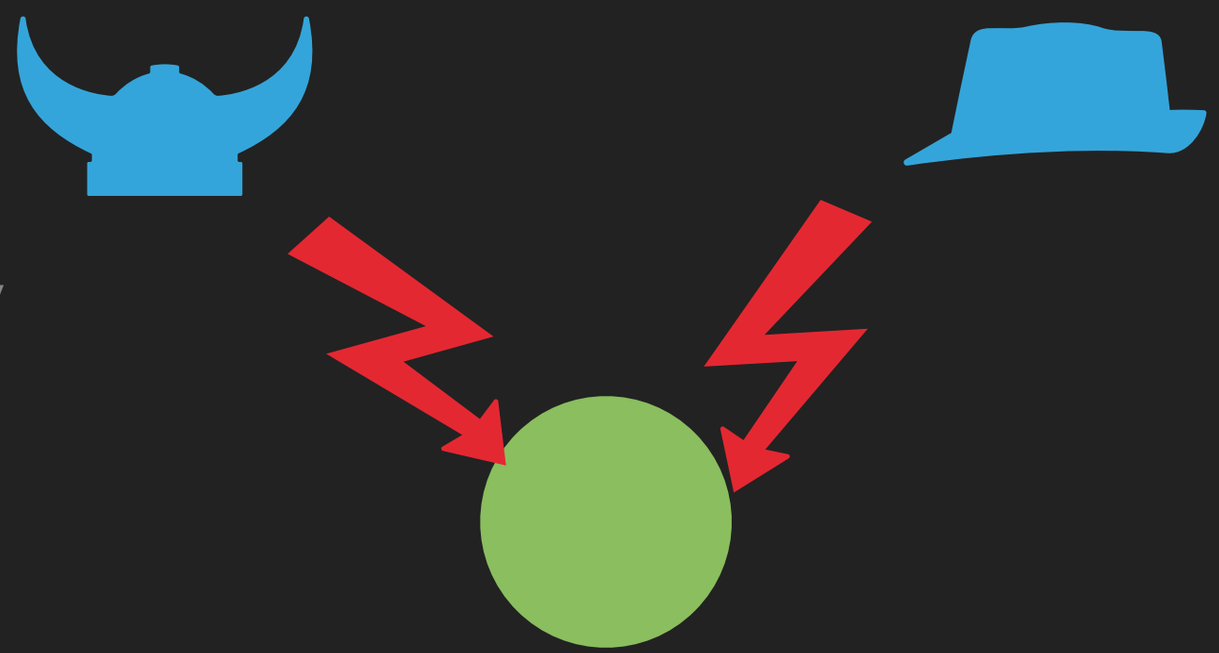
These are just a few examples

Hacker hordes attack through the front door, loud and noisy



These are just a few examples

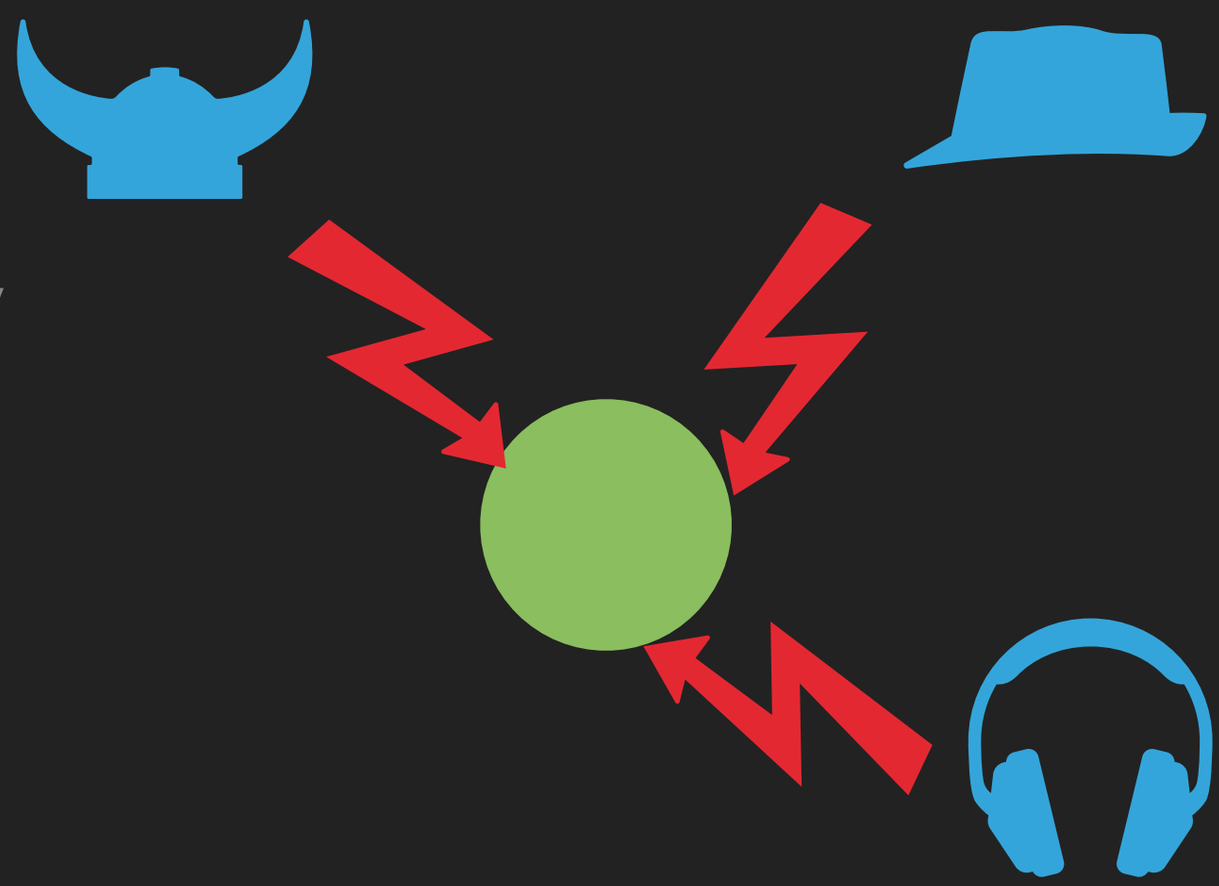
Hacker hordes attack through the front door, loud and noisy



Sophisticated attack combining technical flaws and calling your customer service

These are just a few examples

Hacker hordes attack through the front door, loud and noisy



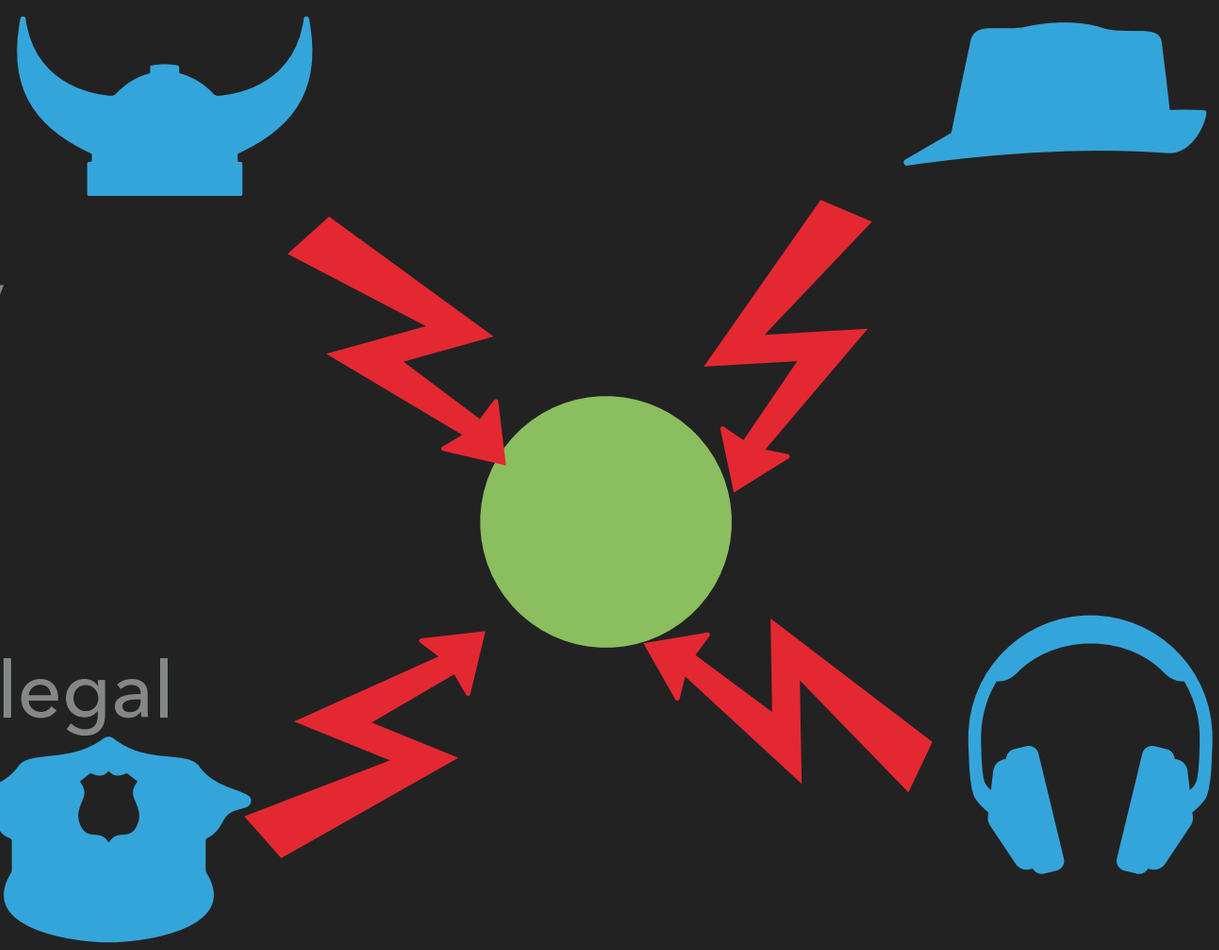
Sophisticated attack combining technical flaws and calling your customer service

Sneaky, sneaky spies (*)

These are just a few examples

(*) https://www.ndr.de/nachrichten/netzwelt/Snowden-NSA-spioniert-Wirtschaft-aus,snowden235.html

Hacker hordes attack through the front door, loud and noisy



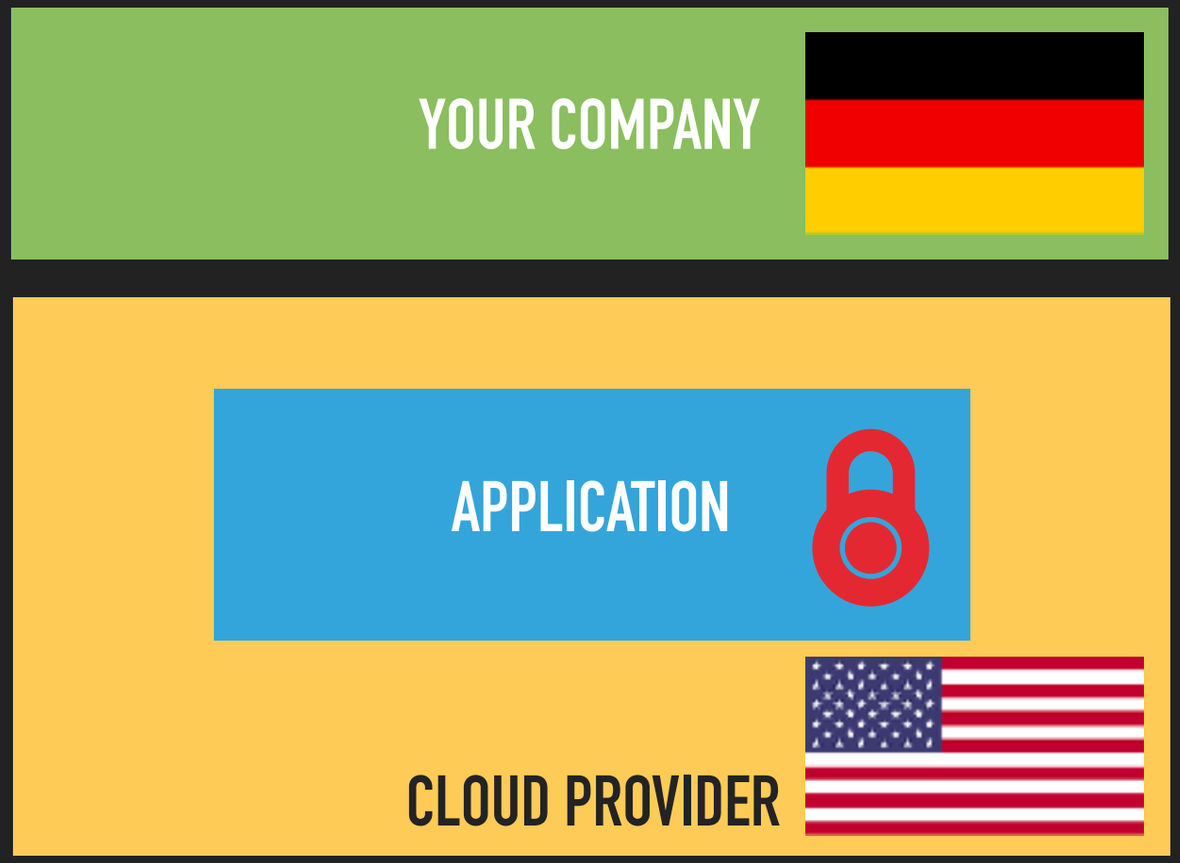
Sophisticated attack combining technical flaws and calling your customer service

Police or legal warrant

Sneaky, sneaky spies (*)

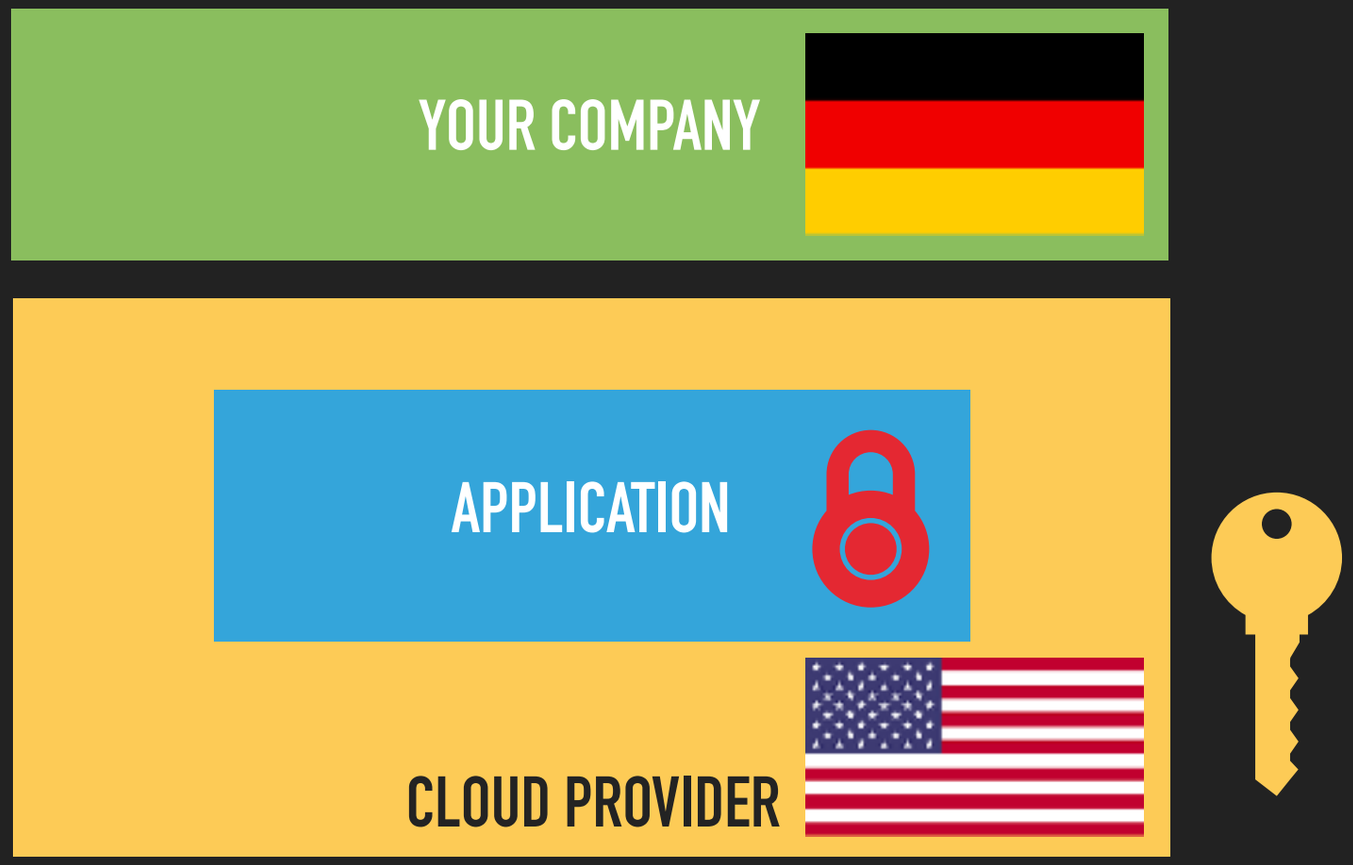
These are just a few examples

(*) https://www.ndr.de/nachrichten/netzwelt/Snowden-NSA-spioniert-Wirtschaft-aus,snowden235.html



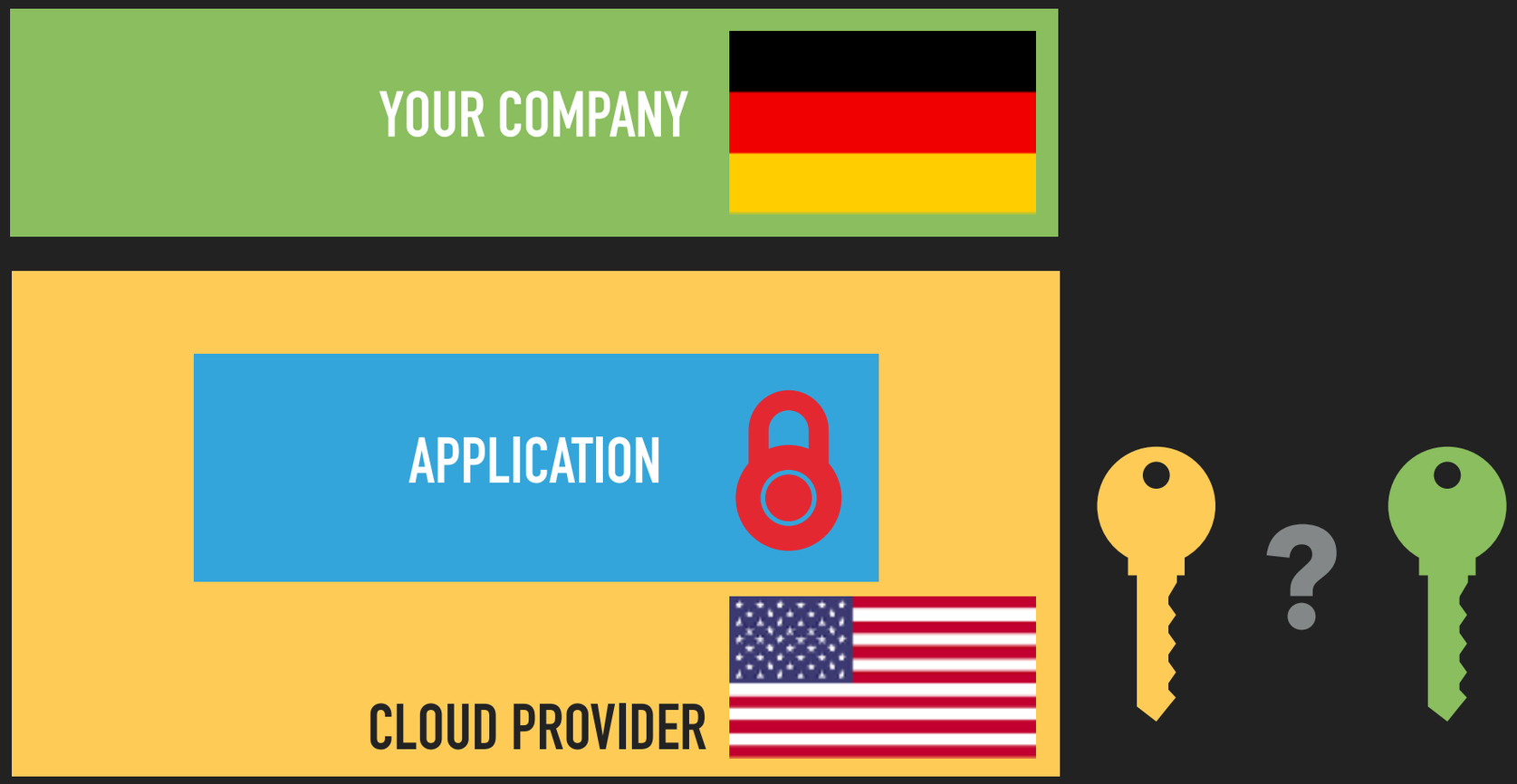
Watch for legal compulsions of your providers
A warrant can include all data – including keys

PS: You will most likely never know that your keys are lost



Watch for legal compulsions of your providers
A warrant can include all data – including keys

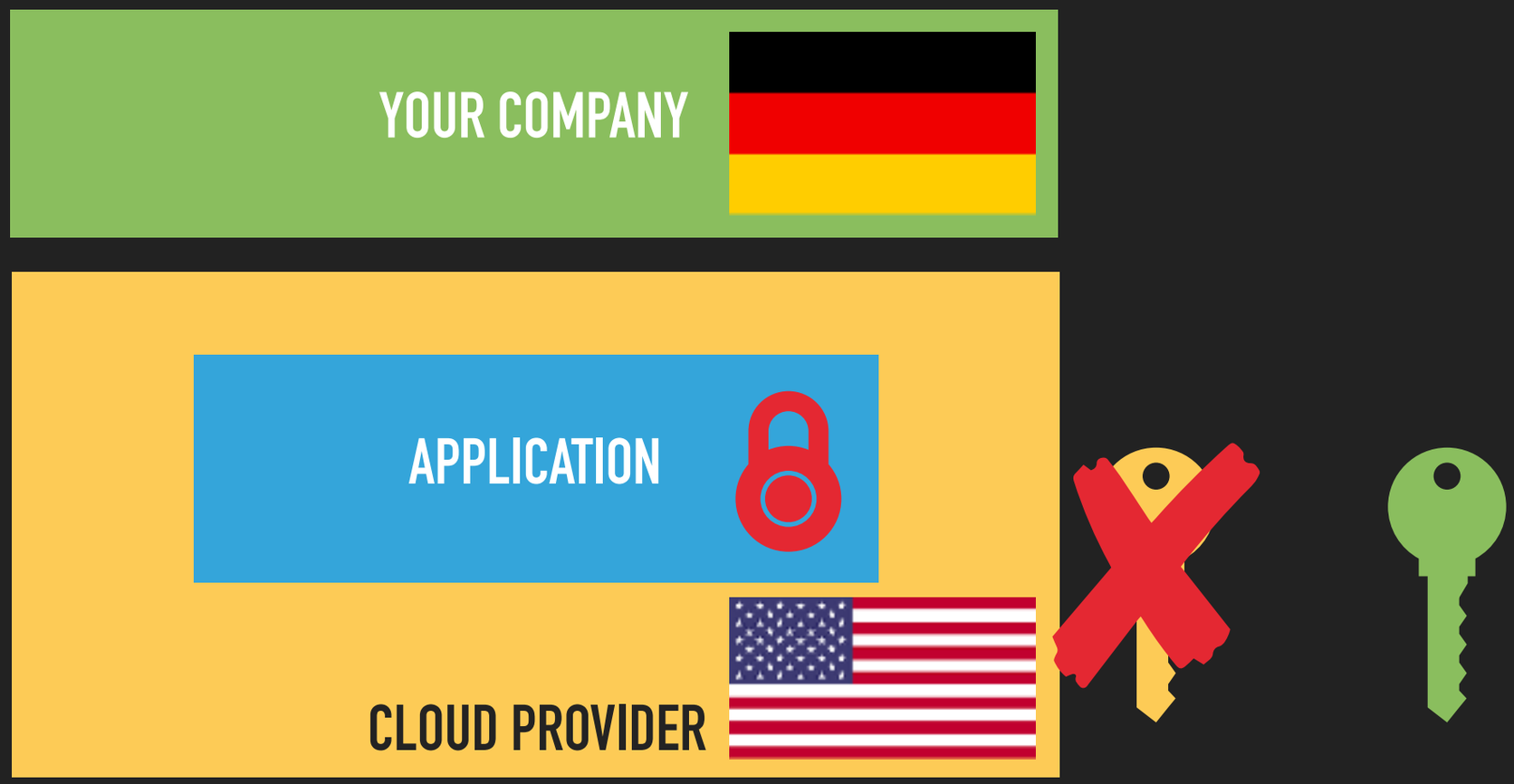
PS: You will most likely never know that your keys are lost



- Who owns the key?

Watch for legal compulsions of your providers
A warrant can include all data – including keys

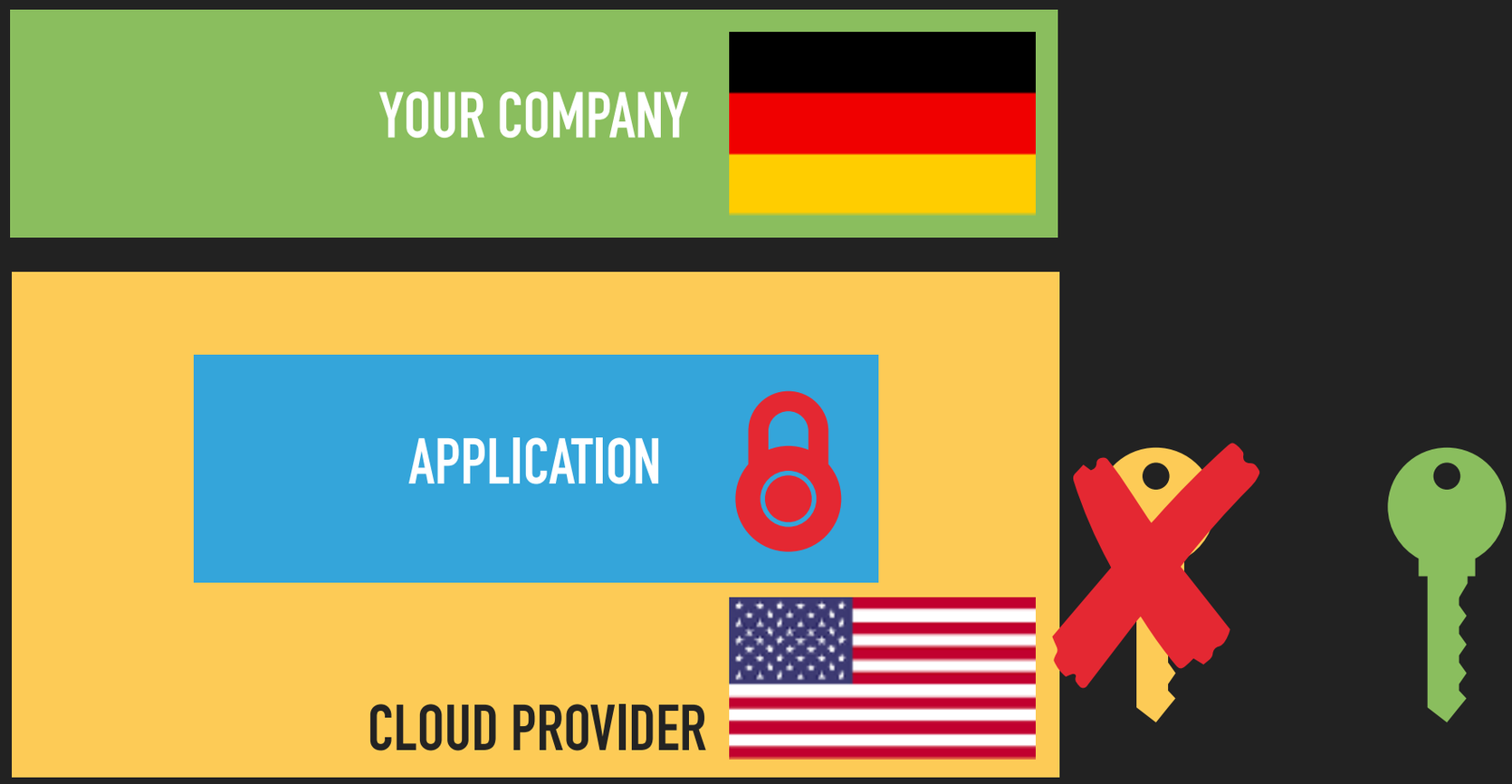
PS: You will most likely never know that your keys are lost



- **Who <u>owns</u> the key?**

Watch for legal compulsions of your providers
A warrant can include all data – including keys

PS: You will most likely never know that your keys are lost



- Who owns the key?
- Who has the key?

Watch for legal compulsions of your providers
A warrant can include all data – including keys

PS: You will most likely never know that your keys are lost

- Who owns the key?
- Who has the key?

Watch for legal compulsions of your providers
A warrant can include all data – including keys

PS: You will most likely never know that your keys are lost

PATTERNS

COMPARING

COMPARE DATA

Problem: Securely compare two data items

Hi Alice do you know this one?

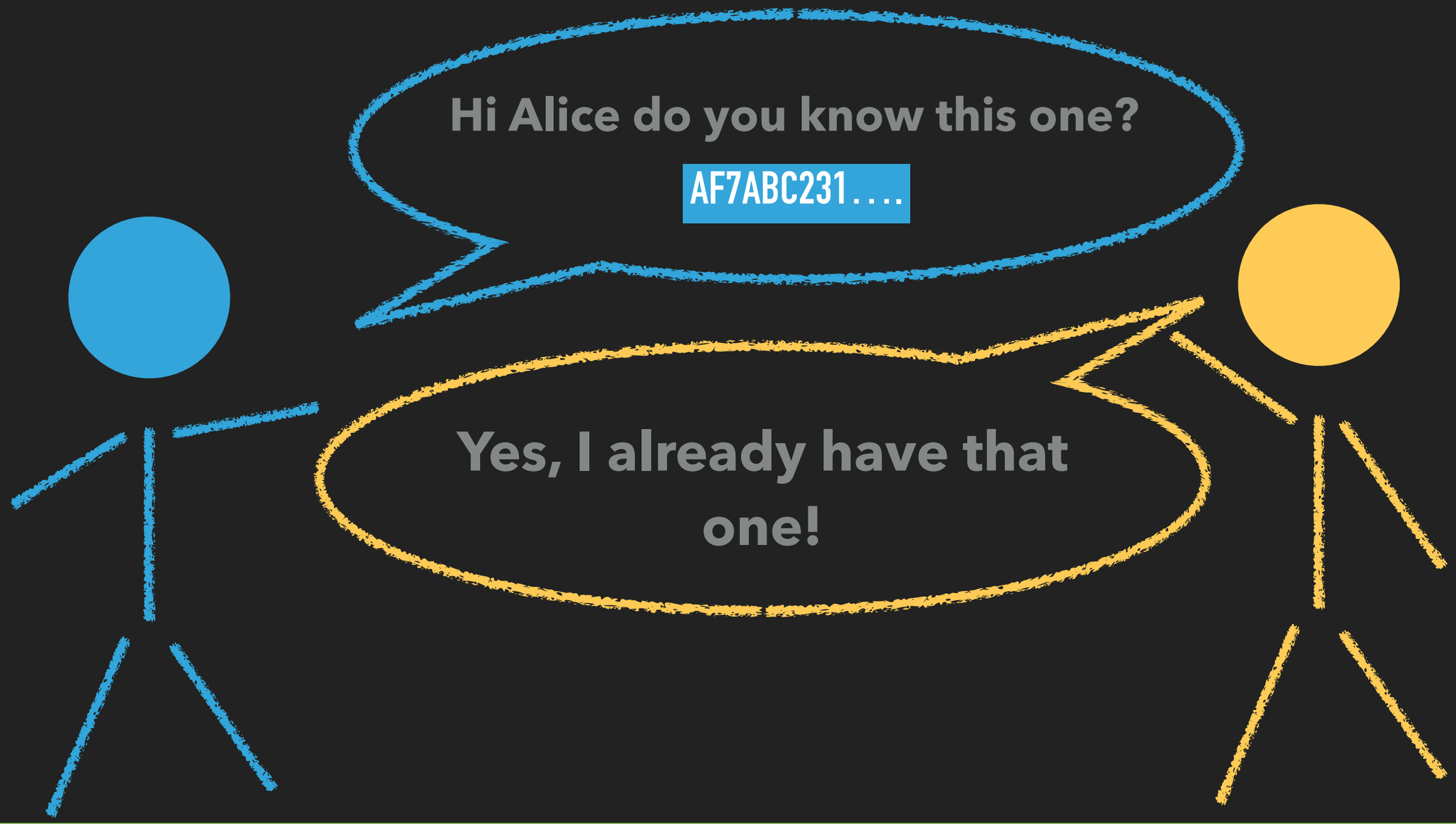
PETER TESTER, BORN 1971-03-12

Bob, did you really just yell confidential data?

CONFIDENTIAL DATA EXCHANGED!

COMPARE DATA

Problem: Securely compare two data items



NO CONFIDENTIAL DATA EXCHANGED.

COMPARE DATA

Problem: Securely compare two data items

Solution: Normalise & hash data, compare hashes

LOREM IPSUM … == LOREM IPSUM …

COMPARE DATA

Problem: Securely compare two data items

Solution: Normalise & hash data, compare hashes

LOREM IPSUM . . . == LOREM IPSUM . . .

=> sha256(LOREM IPSUM . . .) == sha256(LOREM IPSUM . . .)

COMPARE DATA

Problem: Securely compare two data items

Solution: Normalise & hash data, compare hashes

LOREM IPSUM . . . == LOREM IPSUM . . .

=> **sha256(** LOREM IPSUM . . . **) == sha256(** LOREM IPSUM . . . **)**

- Collisions [**A != B** but **sha256(A) == sha256(B)**] are mathematically possible, but practically not relevant

COMPARE DATA

Problem: Securely compare two data items

Solution: Normalise & hash data, compare hashes

LOREM IPSUM . . . == LOREM IPSUM . . .

=> sha256(LOREM IPSUM . . .) == sha256(LOREM IPSUM . . .)

<=> 4C53E9C9. . . == 4C53E9C9. . .

★ Collisions [**A != B** but **sha256(A) == sha256(B)**] are mathematically possible, but practically not relevant

COMPARE DATA

Problem: Securely compare two data items

Solution: Normalise & hash data, compare hashes

1 - Normalize

E.g. J. EDGAR HOOVER → HOOVER, JOHN EDGAR →* H160, J500 E326

2 - Hash

Use *hash(salt + data)* to prevent precomputing attacks. Use multiple iterations of hashing.

- *public salt* => treat hash as *pseudonymised*
- *secret salt* => treat hash as *anonymised*

** Soundex - but choose whatever normalisation works for you*

HASHING DOES NOT INCREASE THE ENTROPY! (THINK OF HASHED PHONE NUMBERS)

COMPARE DATA

Hashed personal data sometimes is longer personal data!

Dr. Grace Nacimiento von der Kanzlei KLEINER: „Der Einsatz von Salt-Wert und kryptographischer Hashfunktion durch Posteo sorgt dafür, dass eine vom Kunden eingegebene Mobilfunknummer anonymisiert wird, bevor eine Übermittlung an Posteo erfolgt. **Für die bei Posteo allein gespeicherten Hashwerte fehlt es daher an einem Personenbezug** […] Die gespeicherten Hashwerte sind daher kein Bestandsdatum im Sinne des § 95 TKG […]" ()

Bundesdatenschutzbeauftragte Andrea Voßhof:

„**Auch aus meiner Sicht ist der gespeicherte gesaltete Hashwert kein personenbeziehbares Datum**. (…) Zusammenfassend stelle ich fest, dass Posteo im Sinne des § 95 TKG keine Bestandsdaten erhebt".

https://posteo.de/bfdi_pruefbericht.pdf

https://posteo.de/blog/bnetza-entscheidung-zu-posteo-kryptographisch-bearbeitete-daten-nicht-auskunftspflichtig

PATTERNS

TRANSPARENT ENCRYPTION

TRANSPARENT ENCRYPTION

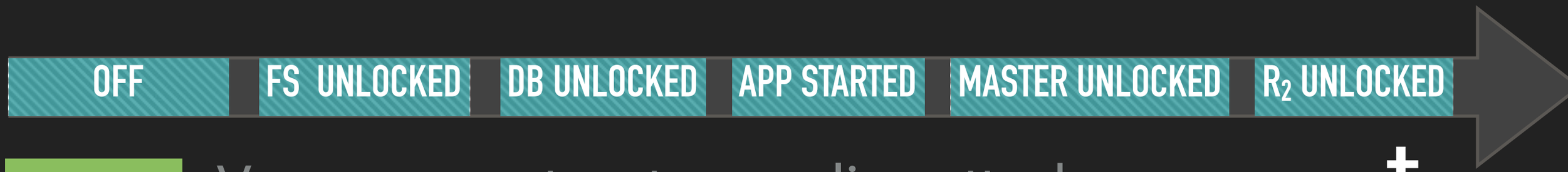
What?
Transparent encryption
is also transparent to the attacker
?!?!

- Full disk encryption
- File system encryption
- Transparent database encryption

Help against stolen hard disks.

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)

Attack



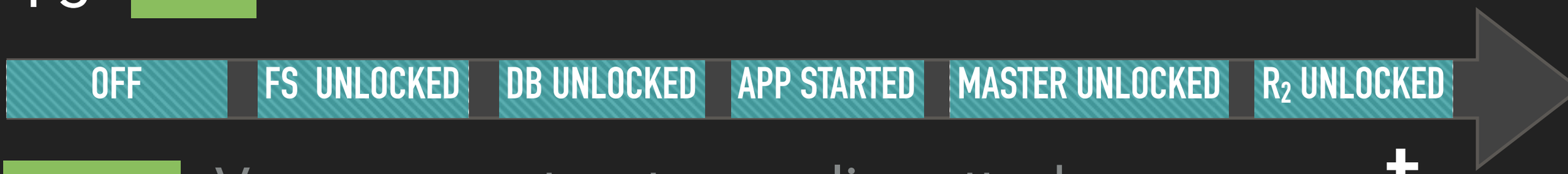
- Very secure at rest, no online attack
- Very secure at rest, key in RAM
- Very secure at rest, data in RAM
- Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)

Attack

FS*



Very secure at rest, no online attack

Very secure at rest, key in RAM

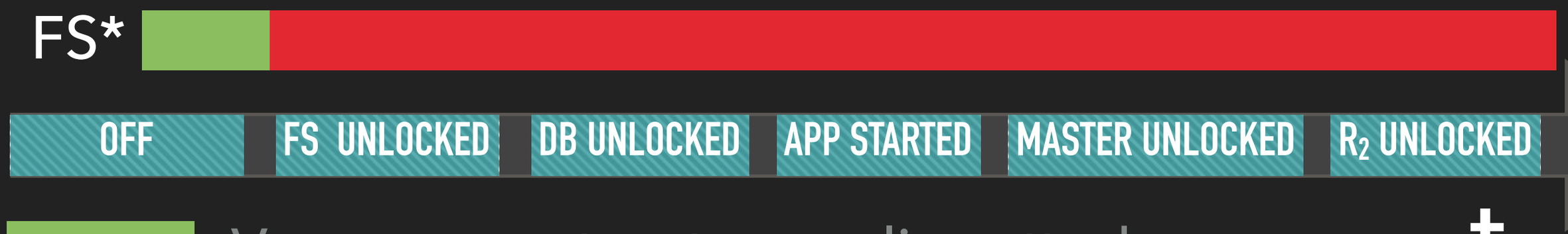
Very secure at rest, data in RAM

Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)

Attack



Very secure at rest, no online attack

Very secure at rest, key in RAM

Very secure at rest, data in RAM

Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)

Attack

Very secure at rest, no online attack

Very secure at rest, key in RAM

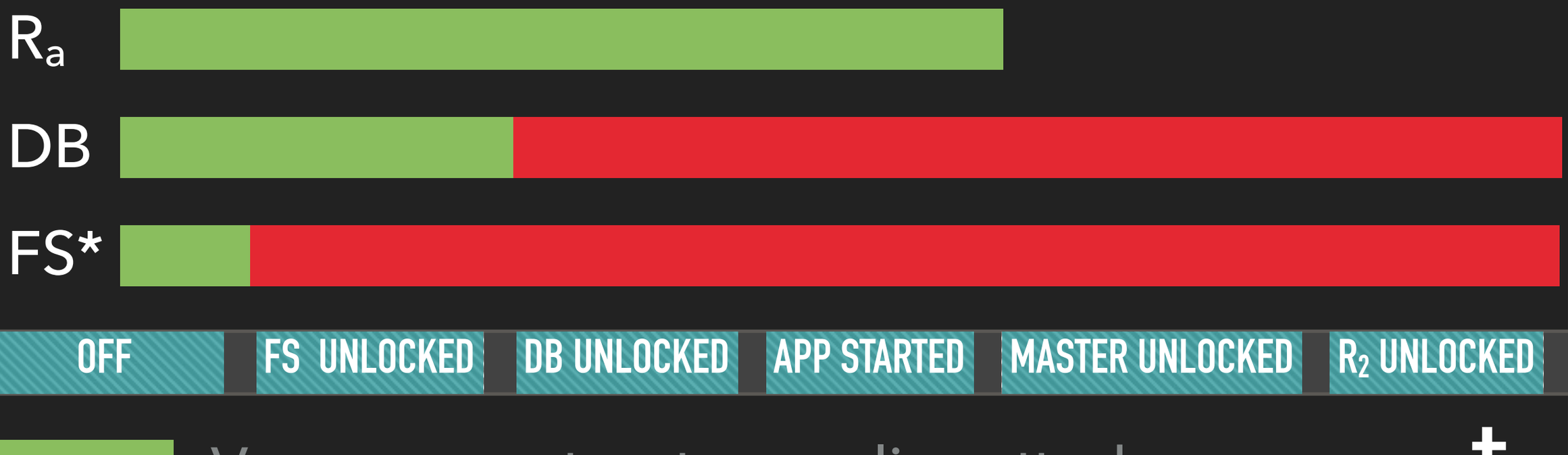
Very secure at rest, data in RAM

Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)

Attack



Very secure at rest, no online attack

Very secure at rest, key in RAM

Very secure at rest, data in RAM

Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)

Attack

Very secure at rest, no online attack

Very secure at rest, key in RAM

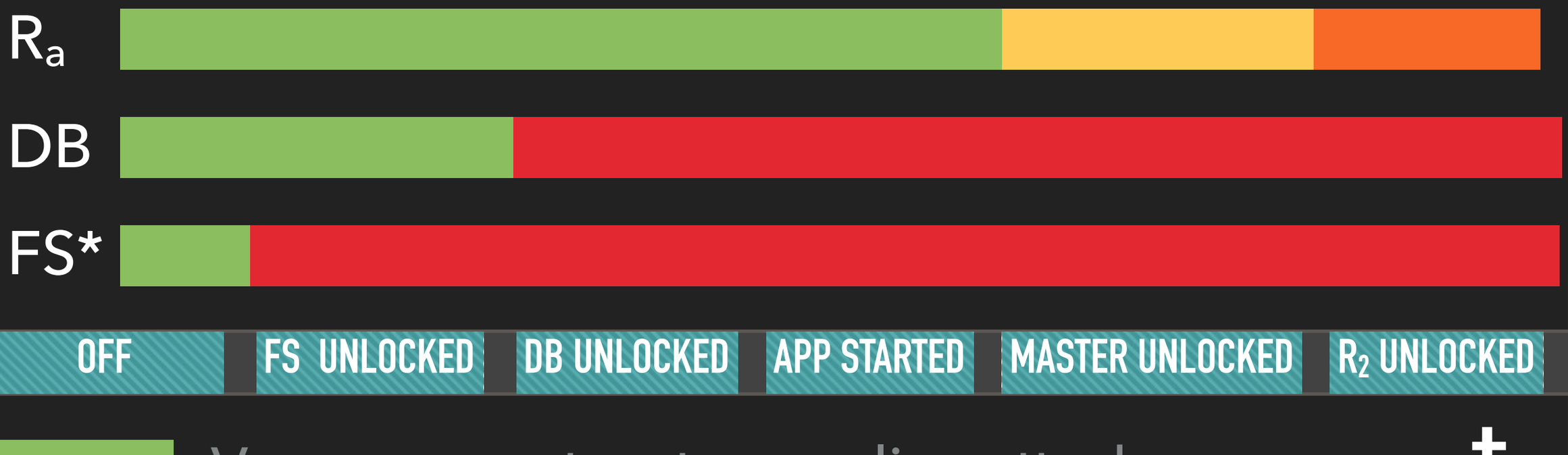
Very secure at rest, data in RAM

Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)

Attack



Very secure at rest, no online attack

Very secure at rest, key in RAM

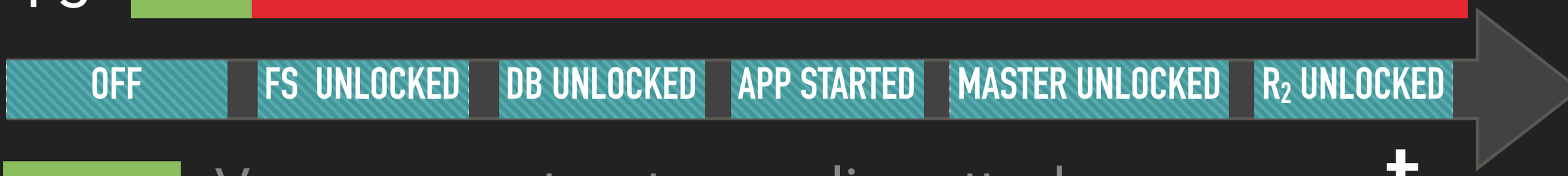
Very secure at rest, data in RAM

Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)

Attack



Very secure at rest, no online attack

Very secure at rest, key in RAM

Very secure at rest, data in RAM

Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)

R_b

R_a

DB

FS*

OFF | FS UNLOCKED | DB UNLOCKED | APP STARTED | MASTER UNLOCKED | R_2 UNLOCKED

t

Very secure at rest, no online attack

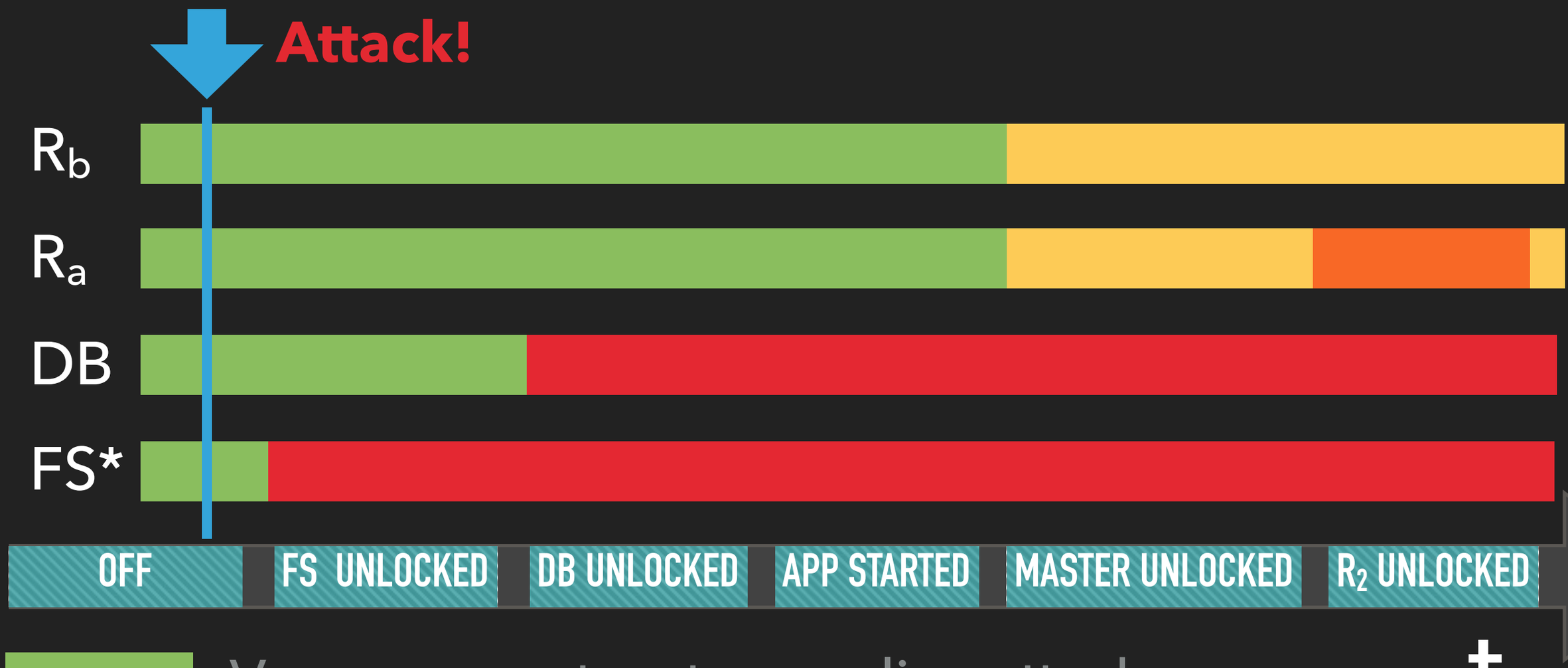
Very secure at rest, key in RAM

Very secure at rest, data in RAM

Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



Very secure at rest, no online attack

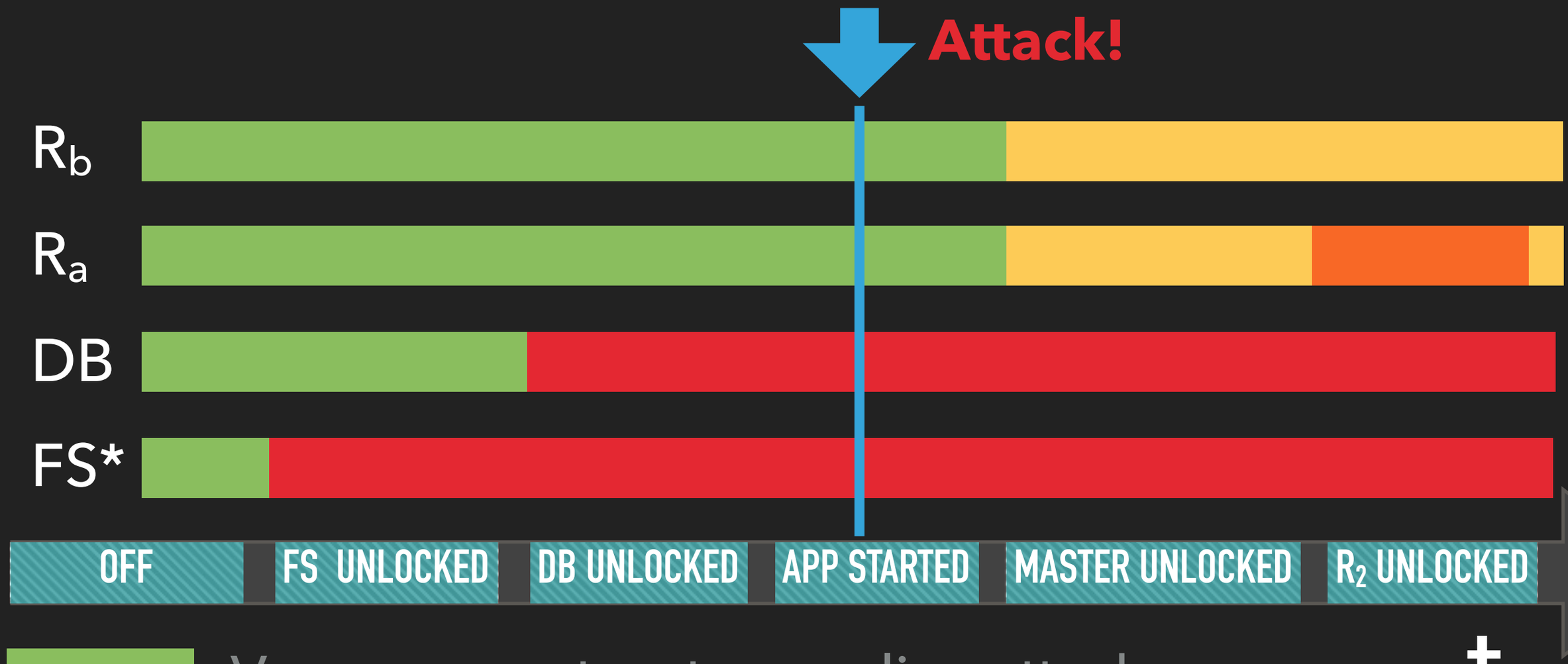
Very secure at rest, key in RAM

Very secure at rest, data in RAM

Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



Very secure at rest, no online attack

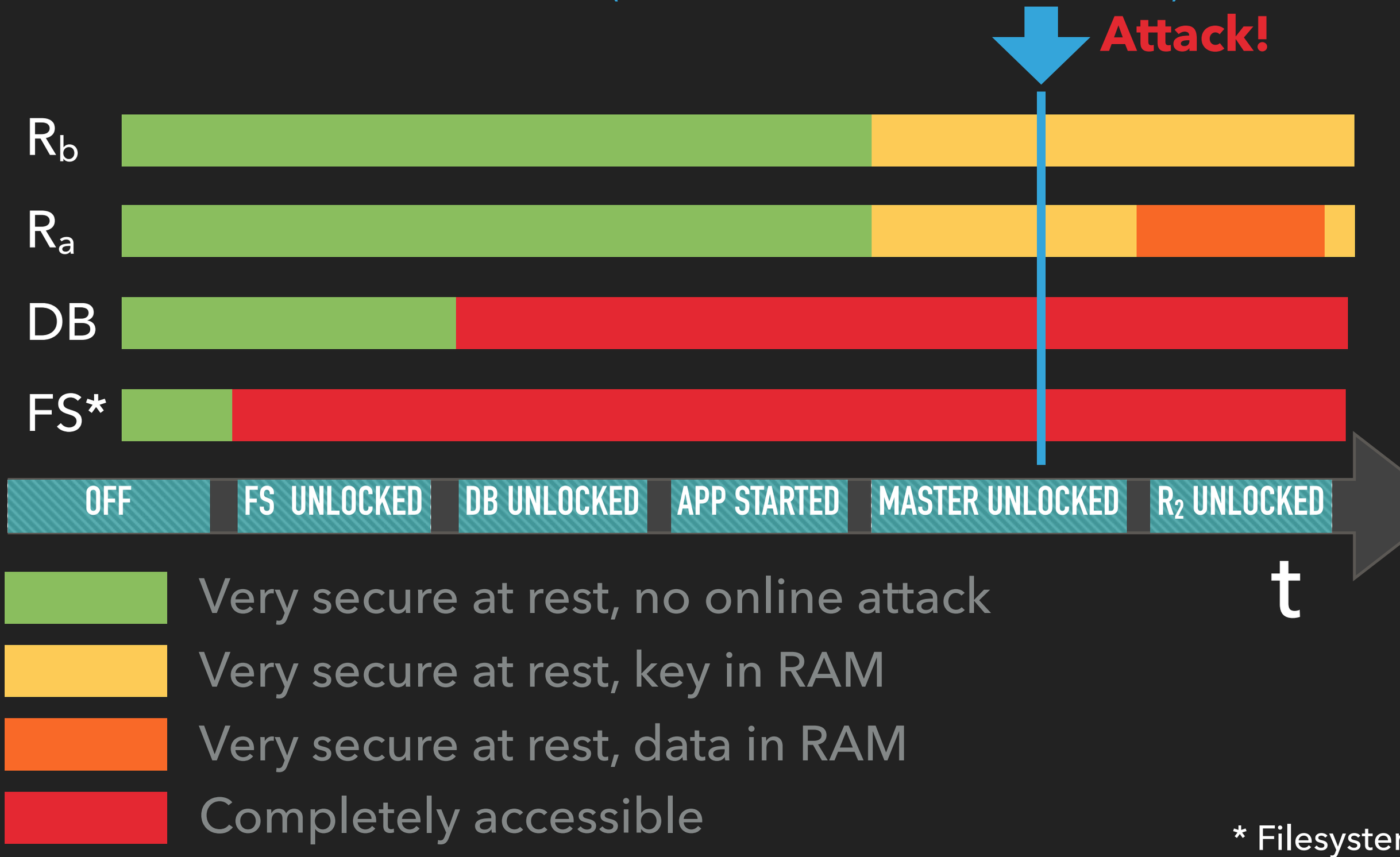
Very secure at rest, key in RAM

Very secure at rest, data in RAM

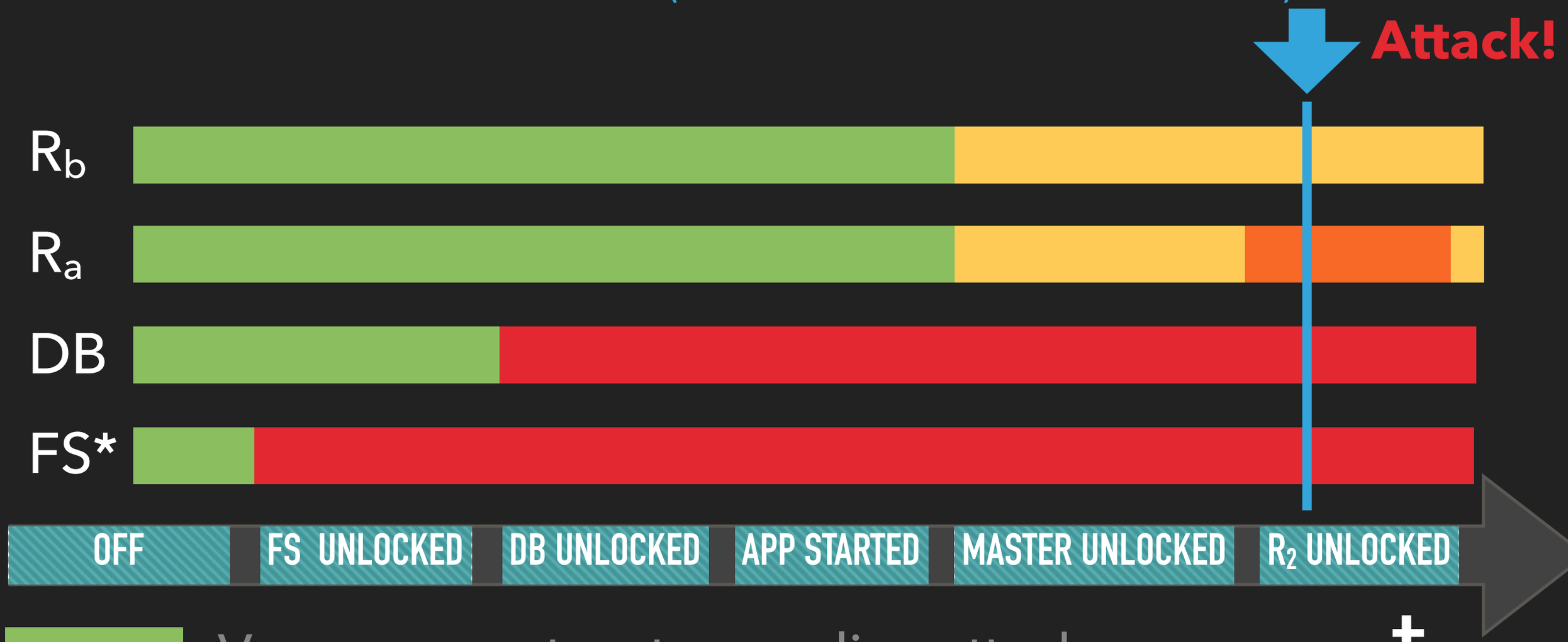
Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



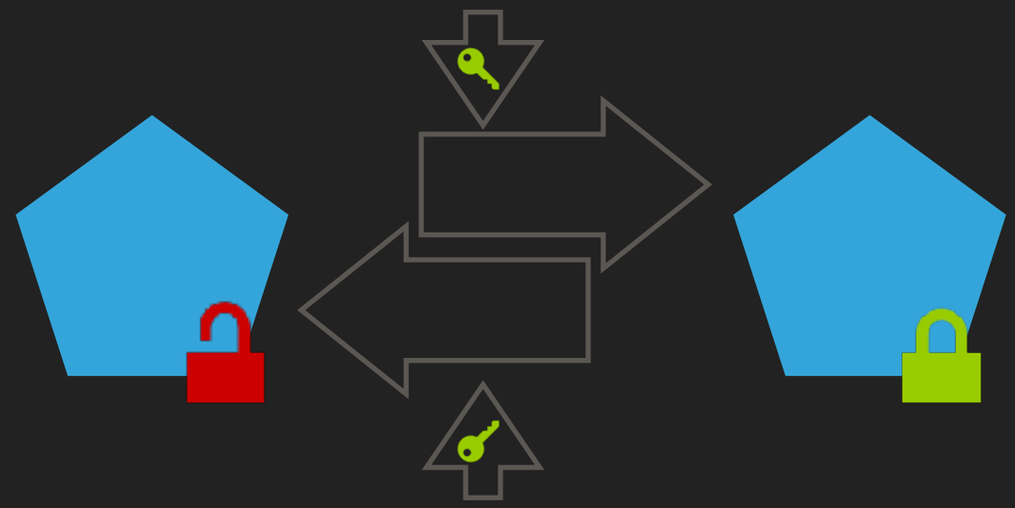
Very secure at rest, no online attack

Very secure at rest, key in RAM

Very secure at rest, data in RAM

Completely accessible

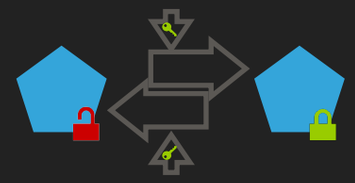
* Filesystem



PATTERNS

STORING

SECURE MULTIPLE DATA RECORDS

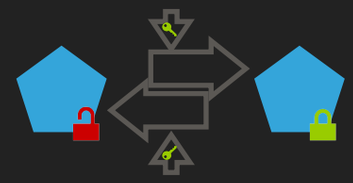


Problem: Multiple records are read and written by the same application.

Solution: Use symmetric encryption to protect the records.

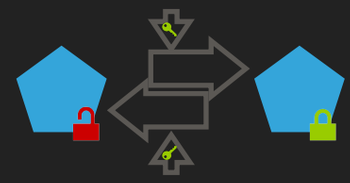
Reading and writing plaintext record ‘R’ 

SECURE MULTIPLE DATA RECORDS



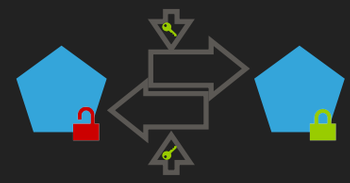
En-/decrypt Data ‘R’ (record) with key ‘K’.🔑

SECURE MULTIPLE DATA RECORDS



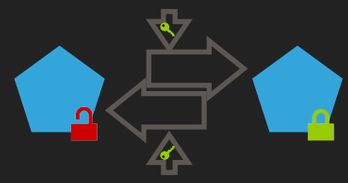
En-/decrypt Data ‘R’ (record) with key ‘K’.🔑

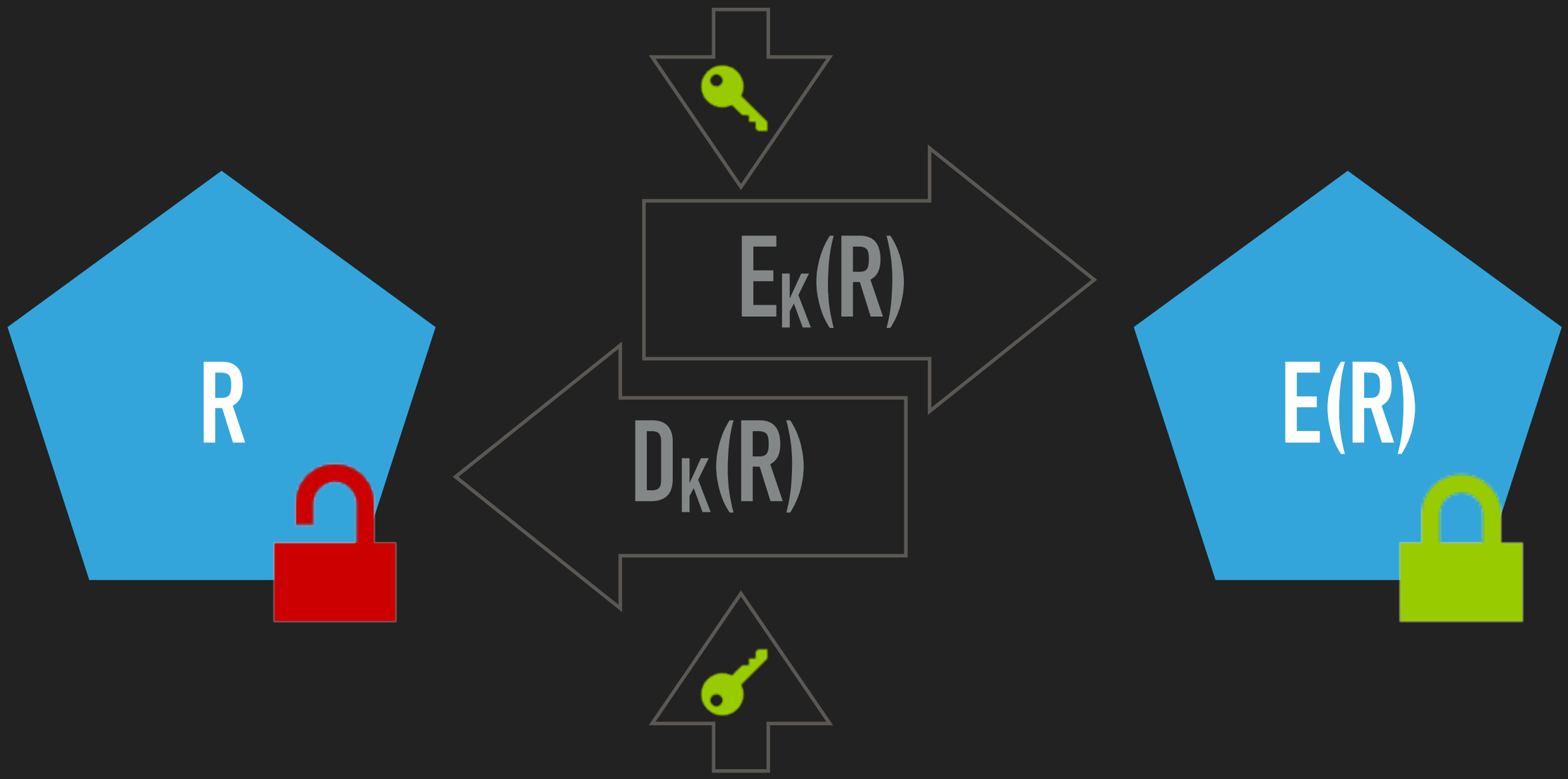
SECURE MULTIPLE DATA RECORDS



En-/decrypt Data ‘R’ (record) with key ‘K’.🔑

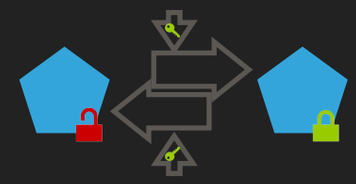
SECURE MULTIPLE DATA RECORDS





En-/decrypt Data ‘R’ (record) with key ‘K’.

SECURE SMALL KEY(S)

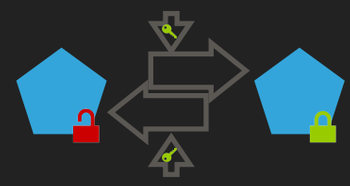


R

$E_K(R)$

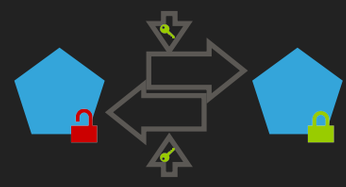
E(R)

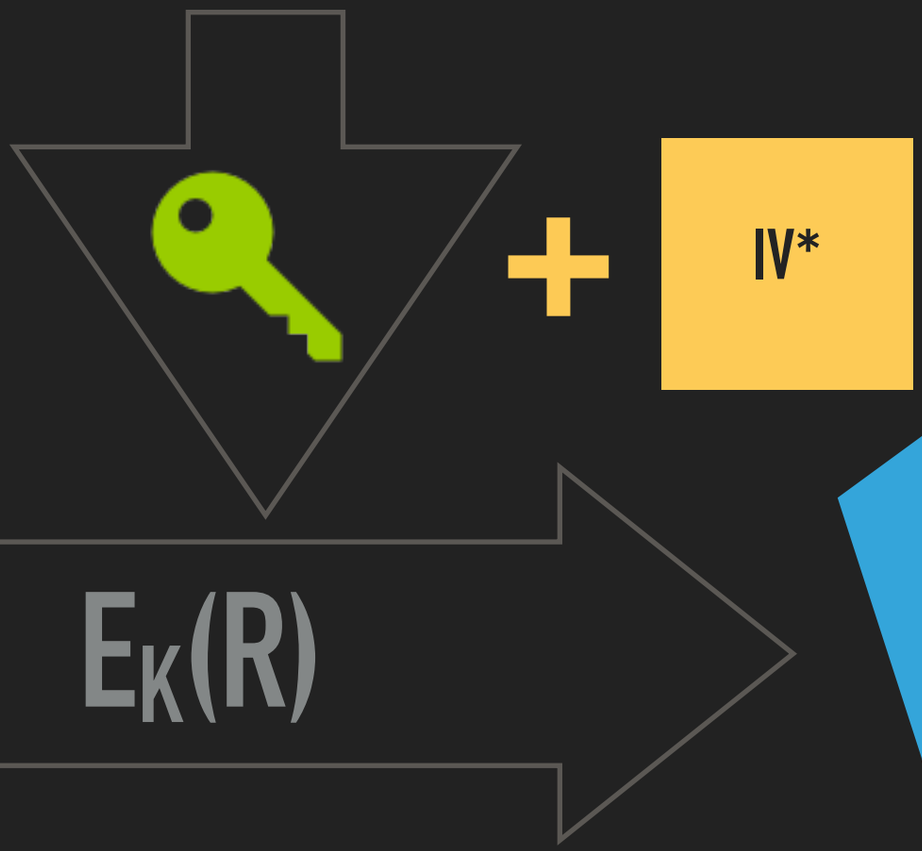
SECURE SMALL KEY(S)



E(R)

SECURE SMALL KEY(S)



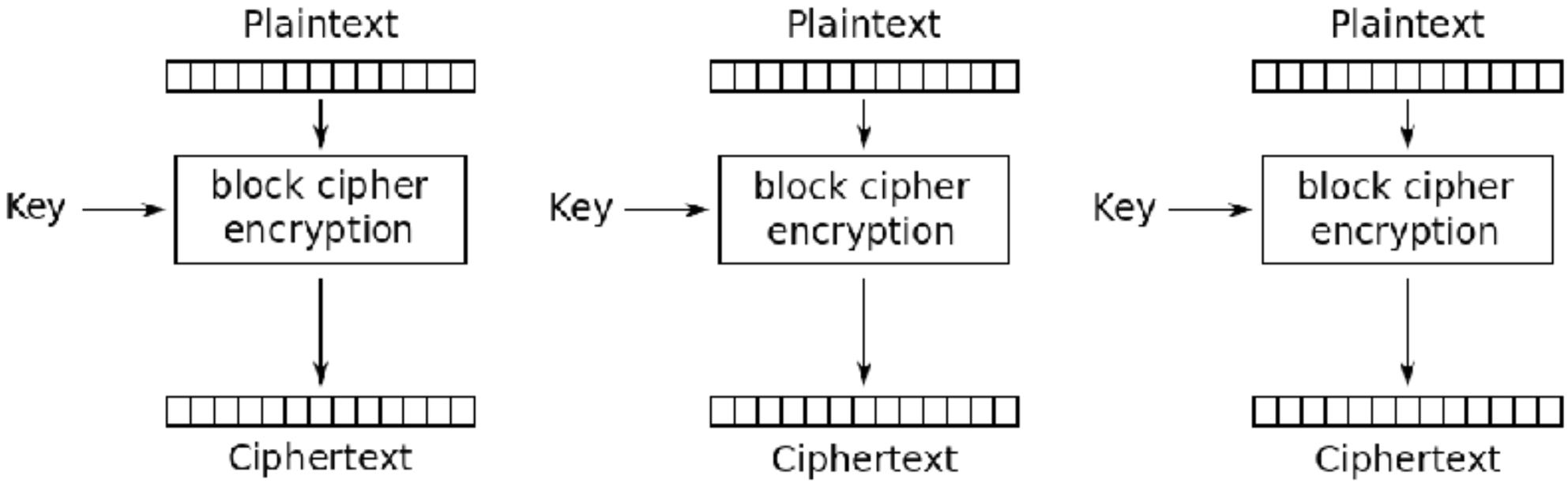


E(R)

IV

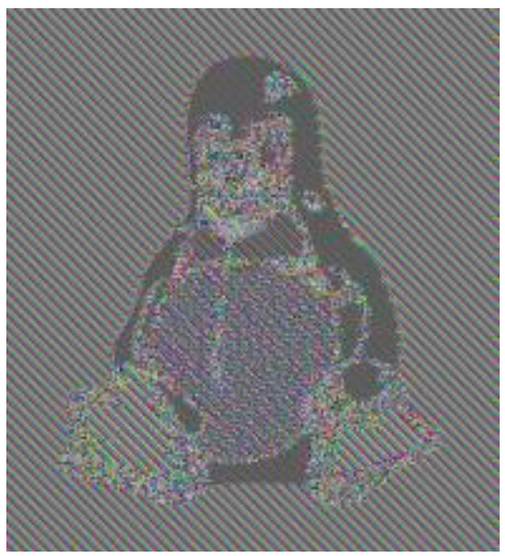
(*) IV - Initialisation Vector. Used in many cryptographic operations. Like a salt. Must be random, might be public.

BLOCK CIPHERS AND THE IV



Electronic Codebook (ECB) mode encryption

Original

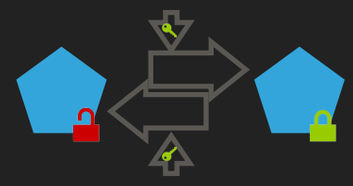


ECB

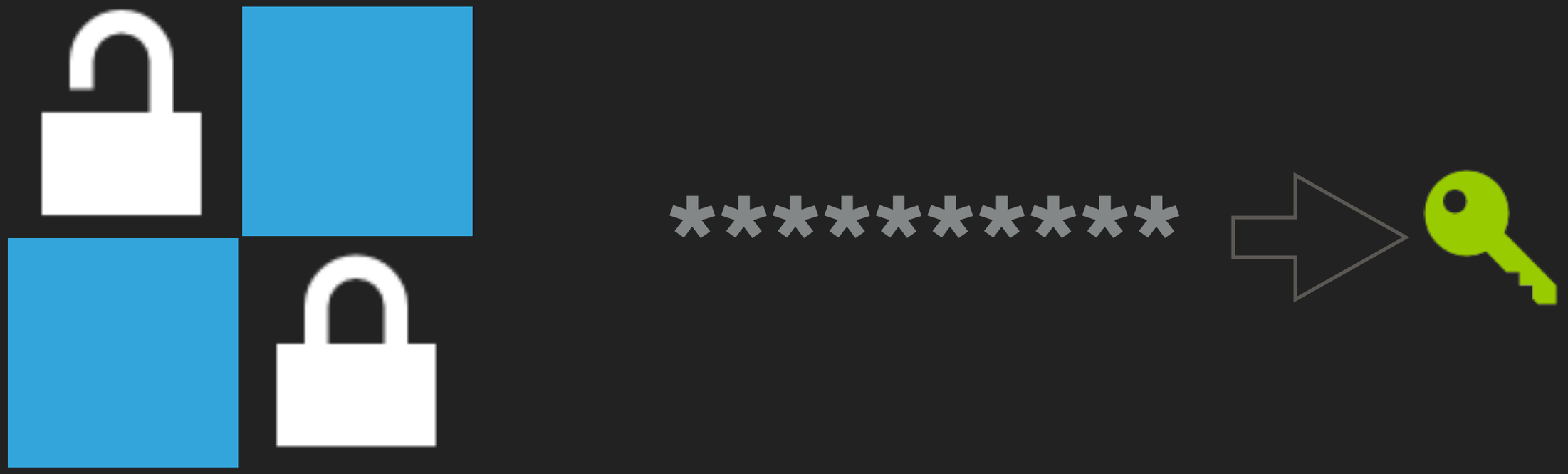
CBC (or other)

https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

SOLUTIONS FOR SECURING KEY(S)



Master key in different storage	**E.g. records in DB, master key on filesystem.**	Baseline. Easy. Protects (only) agains DB theft (e.g. SQL injection)
Encrypt master key	**Use baked in 'obfuscation key' to encrypt master key. Better: Store master key in OS keyring.**	Easy. Some protection against FS access (e.g. remote file inclusion)
Derive per-record key	**Unique per record key derived from master key.** *Bonus: Protect integrity. Bind to record.*	Mostly easy. Protects against some cipher text attacks. *Use AEAD!*
Crypto Host	**All crypto operations on a dedicated host. (Master)key never leaves Crypto Host.**	Depends on architecture. Helps w. key distribution. Makes key theft difficult.
HSM	**Use Hardware Security Module as Crypto Host.**	Expensive & difficult. "Crypto Host on steroids".



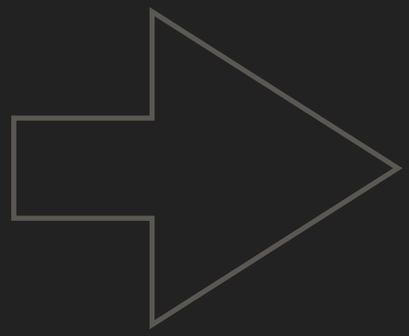
PATTERNS

KEY DERIVATION 1: PASSWORD TO KEY

FROM PASSWORD TO KEY

****

****



password

128 bit key

FROM PASSWORD TO KEY

****

****

password

128 bit key

- A *random* password has
  ~ 6 bits per character (*)
- A 128 bit key needs >= 21 character passwords

(*) a-zA-Z0-9 –> ~64 different values per character (2^6). A 128 bit passwords must be >= 21 characters long (128/6)

FROM PASSWORD TO KEY

****

****

password

128 bit key

- A *random* password has
  ~ 6 bits per character (*)
- A 128 bit key needs >= 21 character passwords

Secure passwords are *very* long

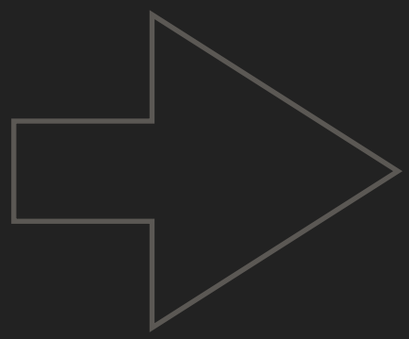
(*) a-zA-Z0-9 –> ~64 different values per character (2^6). A 128 bit passwords must be >= 21 characters long (128/6)

FROM PASSWORD TO KEY

********** 

****

password

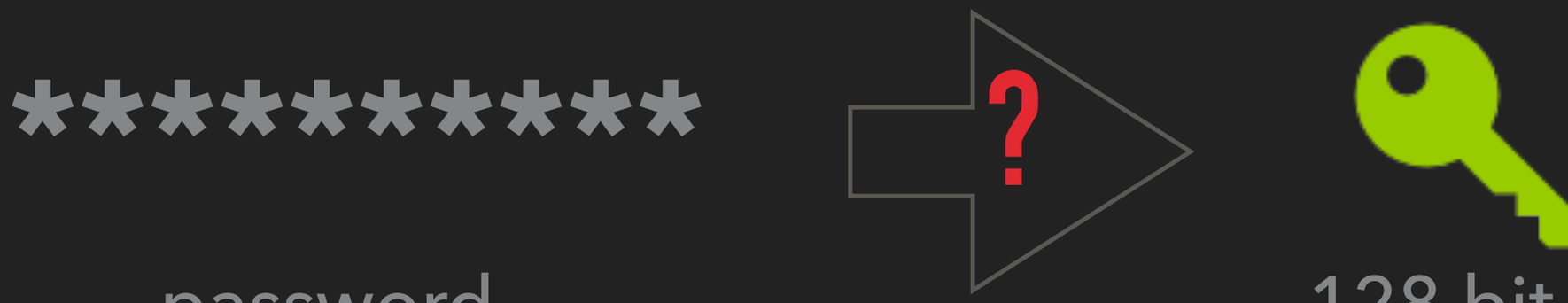


128 bit key

- `aw92SDAVglkqusabvgw38` ✓
  - 128 bit
- `3o8uGsdA` ✗
  - 8 chars, 48 bit (cracked in hours to days)

CREATE YOU PERSONAL PASSWORDS USING A PASSWORD MANAGER!

FROM PASSWORD TO KEY

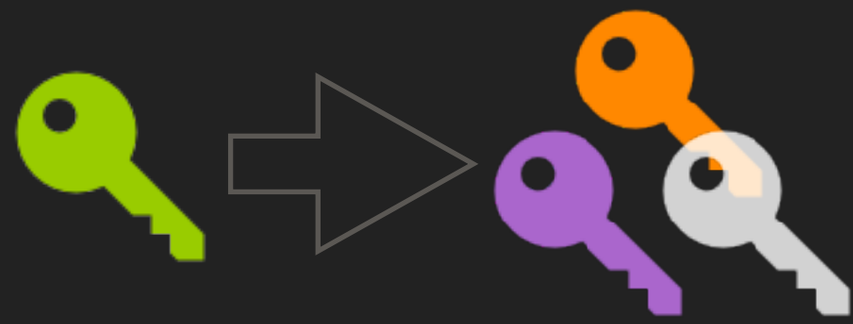


password

128 bit key

- Key derivation functions (KDF) convert passwords to keys
- For good (21+ chars) passwords use HKDF (RFC5869)
- Else: use a KDF with brute force protection (*)
  - SCRYPT (RFC7914)
  - PBKDF2 (RFC2898)

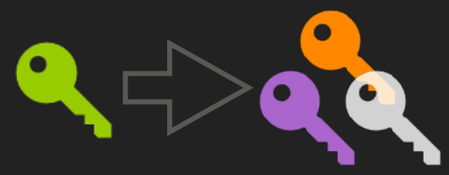
(*) *Brute force protection*: The function is designed to be very slow (up to seconds). This prevents enumeration attacks.



PATTERNS

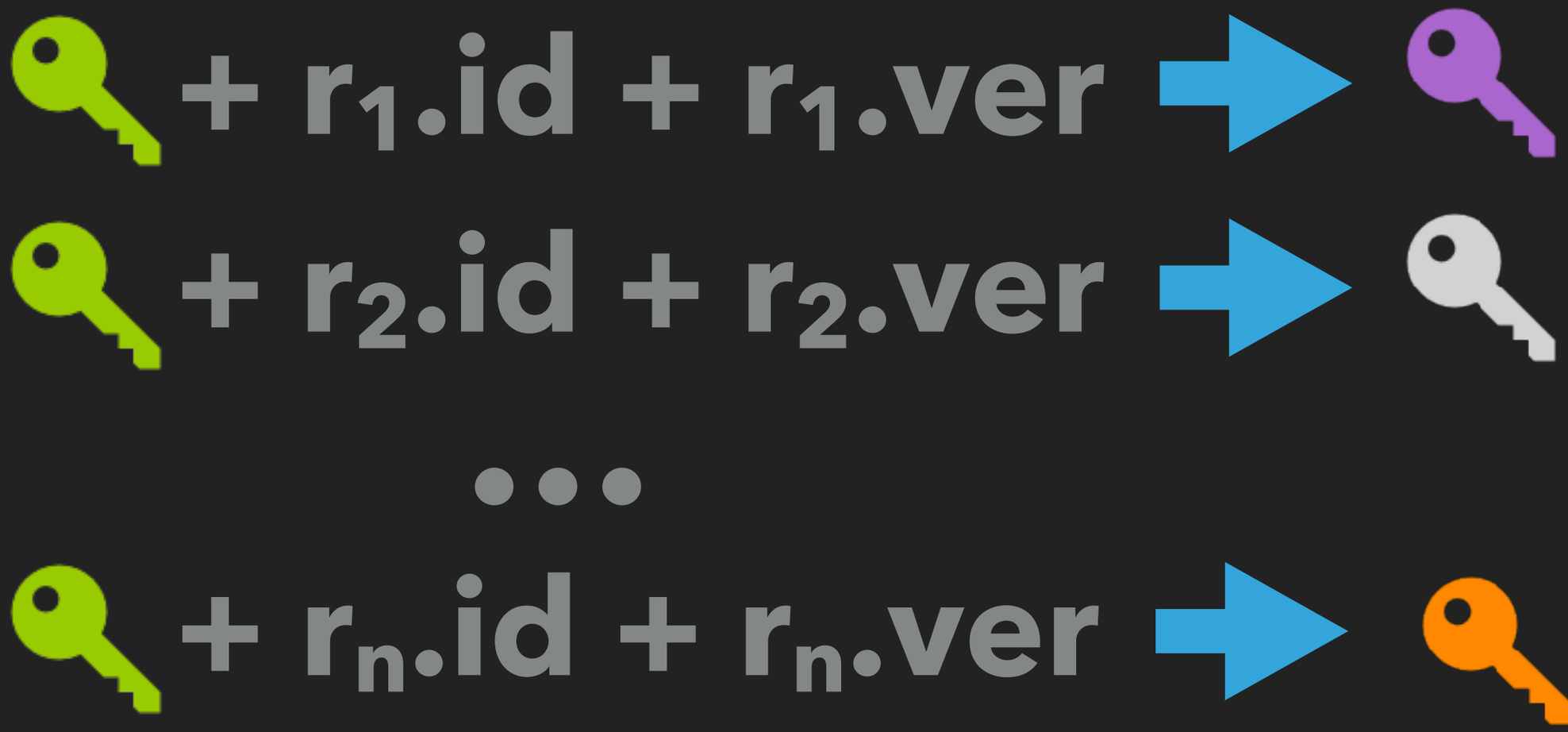
KEY DERIVATION 2: FROM 1 TO N

DERIVE PER RECORD KEYS



Problem: Use different keys for different records, only store master key.

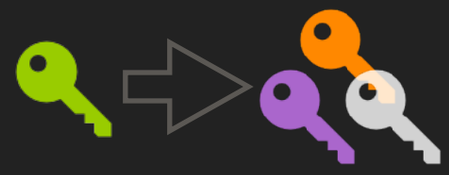
Solution: Use key derivation to derive per-record keys.



IMPORTANT: NEVER USE THE SAME KEY/IV TO ENCRYPT DIFFERENT DATA

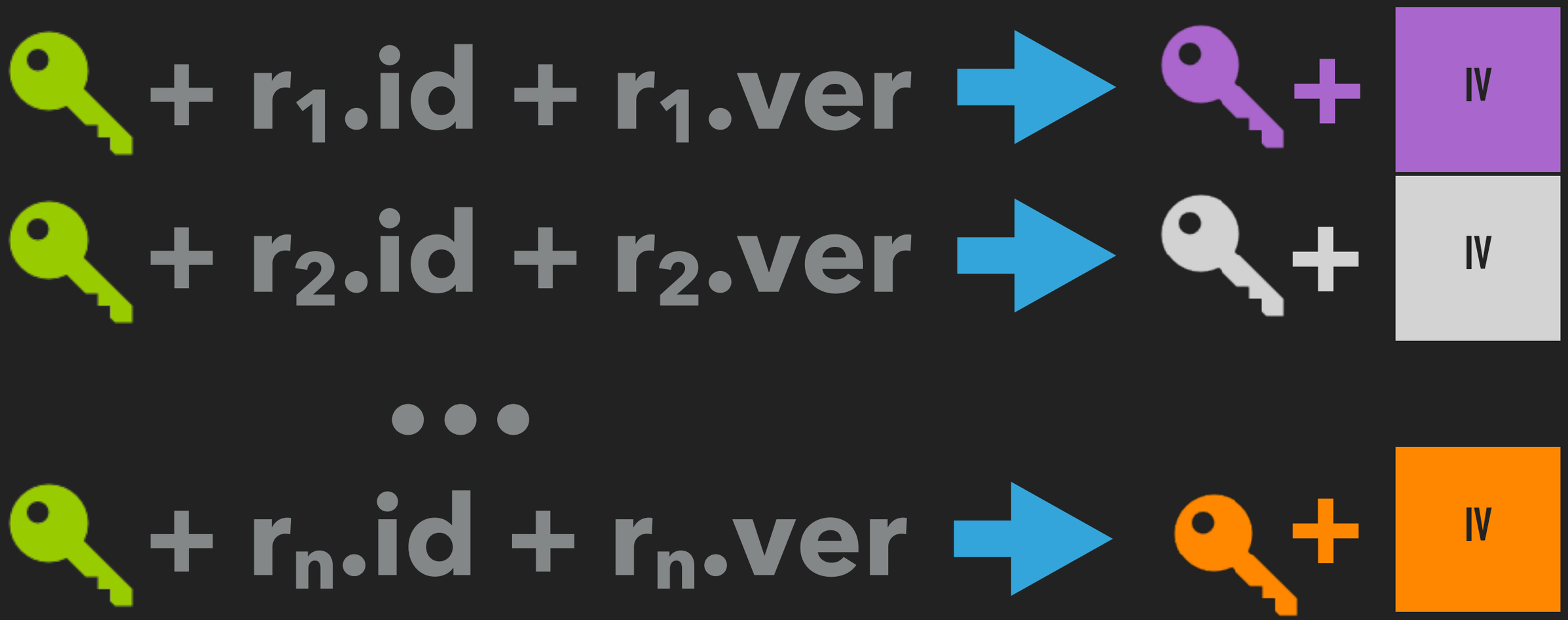
MAKE SURE THAT THE MASTER KEY HAS ENOUGH ENTROPY FOR DERIVED KEY AND DERIVED IV

DERIVE PER RECORD KEYS



Problem: Use different keys for different records, only store master key.

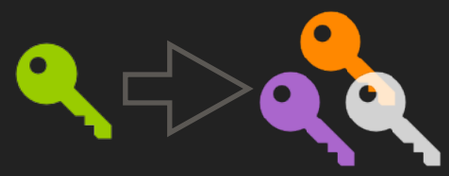
Solution: Use key derivation to derive per-record keys.



IMPORTANT: NEVER USE THE SAME KEY/IV TO ENCRYPT DIFFERENT DATA

MAKE SURE THAT THE MASTER KEY HAS ENOUGH ENTROPY FOR DERIVED KEY AND DERIVED IV

SOLUTIONS FOR DERIVING KEY(S)



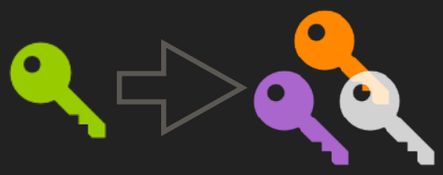
```
// Input:
//  Master_key and
//  (DB) record_id target record DB id
// Output:
//  AES-Key and
//  salt for encrypting target record
// AES-Key and salt for target record. "||" concatenates
// AES-CBC uses 128 bit IV. AES-GCM uses a 96 bit IV
byte[32] keyAndIV = derive_key( master_key ||
                                record_id || record_version, 256 bit)

byte[16] derived_iv   = keyAndIV[0..15]
byte[16] derived_key  = keyAndIV[16..31]
```

- derive_key needs an additional *installation specific* salt of >= 128 bit. PBKDF2 with HMAC sha256 is an example of derive_key, as is scrypt or argon2.
- Use same process for decryption.
- No need to store the *generated* IV value.

IMPORTANT: NEVER USE THE SAME KEY/IV TO ENCRYPT DIFFERENT DATA

MAKE SURE THAT THE MASTER KEY HAS ENOUGH ENTROPY FOR DERIVED KEY AND DERIVED SALT

NEVER REUSE KEYS!

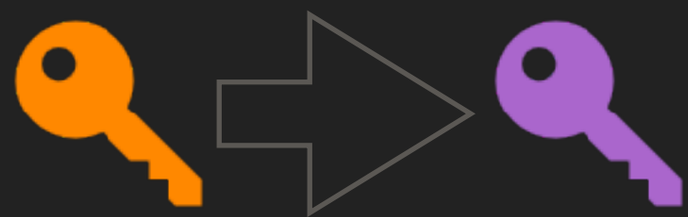


- Encrypting different data with the same key and IV can lead to complete loss of confidentiality / integrity (*)
- **When updating encrypted records a new IV must be used** (better: a new key and IV)
- This can be achieved by incrementing a record-version on each encryption and using it in the key derivation.

(*) This is because of the way CTR/GCM/CBC/… work. See e.g Appendix B of NIST Sp. Pub. 800-38A

IMPORTANT: NEVER USE THE SAME KEY/IV TO ENCRYPT DIFFERENT DATA

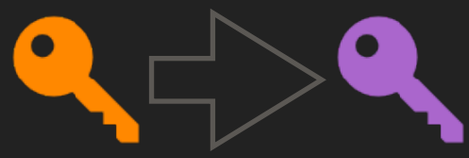
MAKE SURE THAT THE MASTER KEY HAS ENOUGH ENTROPY FOR DERIVED KEY AND DERIVED SALT



PATTERNS

KEY REFRESH

KEY REFRESH



Problem: Keys must only be used for a limited amount of data

Solution: Design for (constant but not frequent) key rollover

Record-ID	Version	Masterkey ID	(Data…)
B9E10DEE-C97E-…	1	**B874920B-E801…**	…
FDE0C6E3-8BF0-…	1	**9A6580FC-1248…**	…
…	3	**9A6580FC-1248…**	…

ATTENTION: REENCRYPTING OPENS A WINDOW OF ATTACK

DES BLOWFISH AES

MD5 SHA-1 SHA-256

RSA-1024 RSA-2048 ?? POST QUANTUM ??

PATTERNS

ALGORITHM ROLLOVER

ALGORITHM ROLLOVER

Problem: Algorithms must be changed and data migrated

Solution: Design for online data migration

Record-ID	…	Masterkey ID	(Data…)
B9E10DEE-C97E-…	…	B874920B-E801-…	…
FDE0C6E3-8BF0-…	…	9A6580FC-1248-…	…
…	…	9A6580FC-1248-…	…

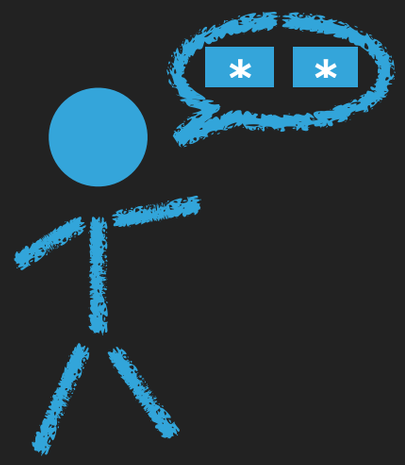
ALGORITHM ROLLOVER

DES BLOWFISH AES
MD5 SHA-1 SHA-256
RSA-1024 RSA-2048 ?? POST QUANTUM ??

Problem: Algorithms must be changed and data migrated

Solution: Design for online data migration

Record-ID	Algorithms	Masterkey ID	(Data…)
B9E10DEE-C97E-…	▸ PBKDF2(…) ▸ AES128–GCM	B874920B-E801-…	…
FDE0C6E3-8BF0-…	▸ SCRYPT(…) ▸ AES256–CBC ▸ PKCS#5	9A6580FC-1248-…	…
…	…	9A6580FC-1248-…	…



PATTERNS

PASSWORD VERIFICATION

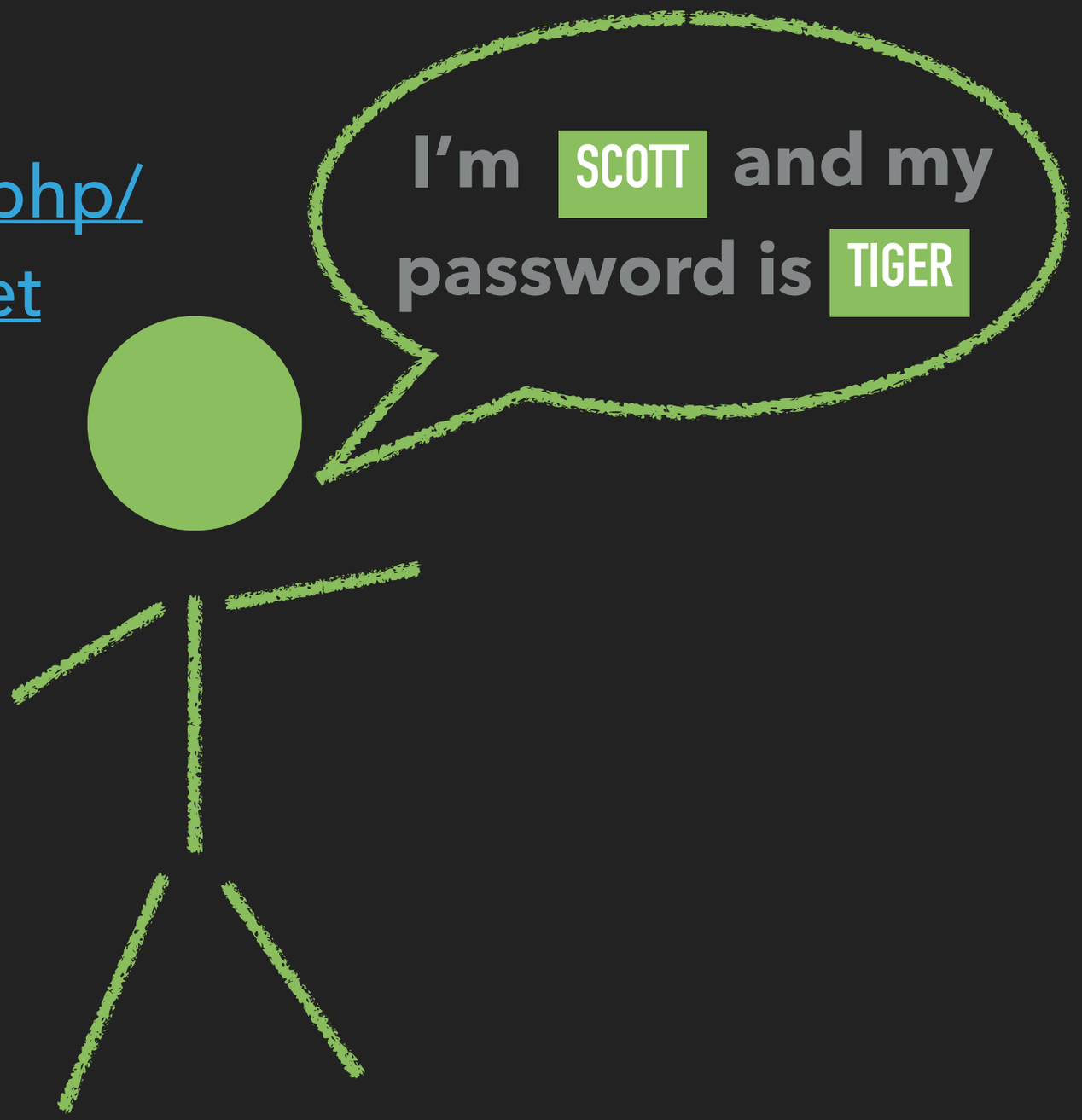
USER LOGIN

Problem: Login a user

Solution: Not in scope here

https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet

Also: OAUTH, Kerberos, …



USER LOGIN: MIGRATE PASSWORDS

Problem: Migrate password hashes to new algorithms

Solution: Chain hashing functions

ValidatePassword

hash password w. MD5

check vs. database

User	Algorithm	Hash
SCOTT	▸ MD5(PWD)	3959dc9…
…	…	…

USER LOGIN: MIGRATE PASSWORDS

Problem: Migrate password hashes to new algorithms

Solution: Chain hashing functions

ValidatePassword

hash password w. MD5 SCOTT TIGER

check vs. database

User	Algorithm	Hash
SCOTT	▸ MD5(PWD)	3959dc9…
…	…	…

USER LOGIN: MIGRATE PASSWORDS

Problem: Migrate password hashes to new algorithms

Solution: Chain hashing functions

ValidatePassword

hash password w. MD5

check vs. database SCOTT 3959DC9...

User	Algorithm	Hash
SCOTT	MD5(PWD)	3959dc9...
...	...	...

USER LOGIN: MIGRATE PASSWORDS

Problem: Migrate password hashes to new algorithms

Solution: Chain hashing functions

ValidatePassword

hash password w. MD5

check vs. database

User	Algorithm	Hash
SCOTT SCOTT	MD5(PWD)	3959dc9... 3959DC9...
...	...	...

USER LOGIN: MIGRATE PASSWORDS

Problem: Migrate password hashes to new algorithms

Solution: Chain hashing functions

ValidatePassword

hash password w. MD5

check vs. database

User	Algorithm	Hash
SCOTT SCOTT	MD5(MD5)	3959dc9... 3959DC9...
...	...	...

USER LOGIN: MIGRATE PASSWORDS

Problem: Migrate password hashes to new algorithms

Solution: Chain hashing functions

ValidatePassword

hash password w. MD5

check vs. database

User	Algorithm	Hash
SCOTT	MD5(PWD)	3959dc9...
...	PURE HASH CONSIDERED HARMFUL	...

Even consumer grade graphic cards calculates giga-hashes (2^30) per second.

- https://www.troyhunt.com/our-password-hashing-has-no-clothes/
- https://gist.github.com/epixoip/a83d38f412b4737e99bbef804a270c40
- http://cynosureprime.blogspot.de/2017/08/320-million-hashes-exposed.html
- https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet
- https://en.wikipedia.org/wiki/Pepper_(cryptography)

Even consumer grade graphic cards calculates giga-hashes (2^30) per second.

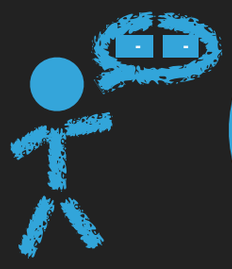
HASHES – EVEN SALTED – OFFER NO PROTECTION AGAINST OFFLINE ATTACKS

SWITCH TO BRUTE-FORCE PROOF (== SLOWER) ALGORITHMS

INCREASE THE ENTROPY BY USING A PEPPER

- https://www.troyhunt.com/our-password-hashing-has-no-clothes/
- https://gist.github.com/epixoip/a83d38f412b4737e99bbef804a270c40
- http://cynosureprime.blogspot.de/2017/08/320-million-hashes-exposed.html
- https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet
- https://en.wikipedia.org/wiki/Pepper_(cryptography)

USER LOGIN: MIGRATE PASSWORDS

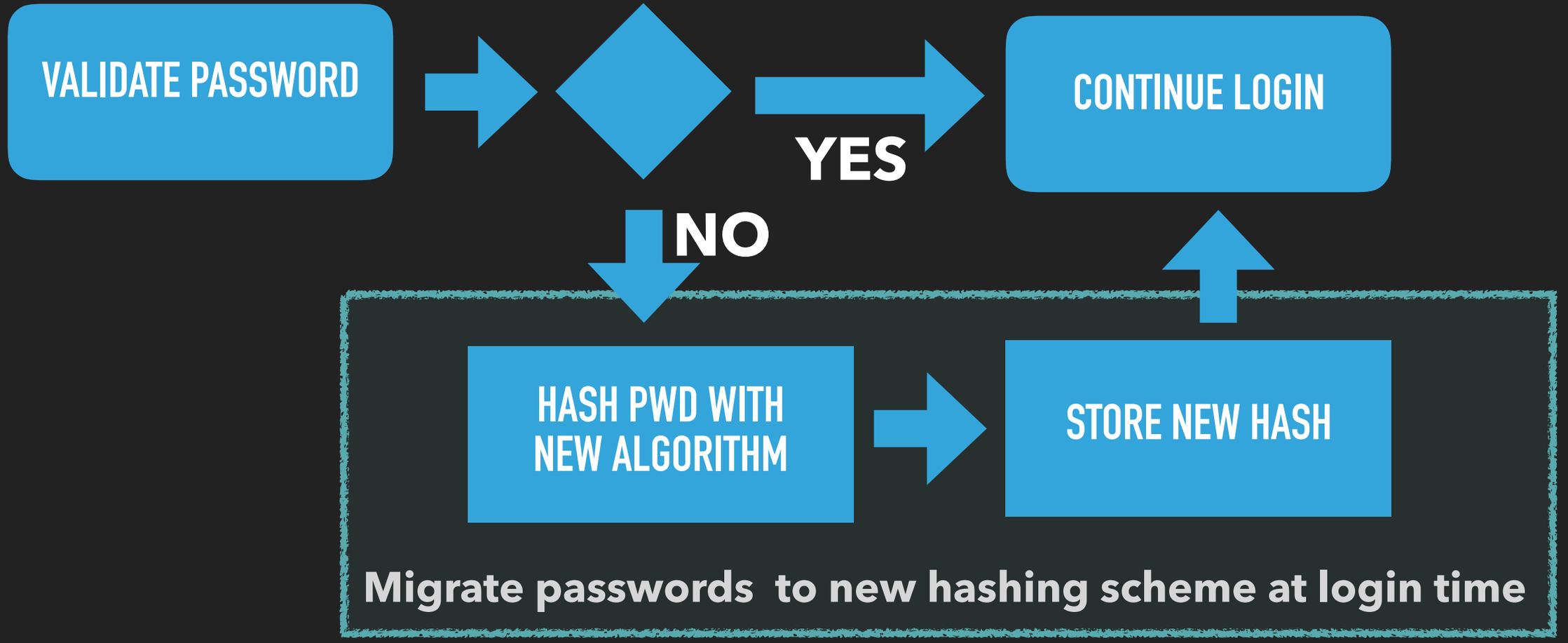


LOGIN

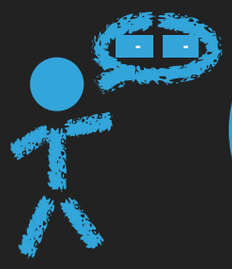
USER
PASSWORD

- Online migration of passwords to a new hashing scheme

Already migrated?



USER LOGIN: MIGRATE PASSWORDS

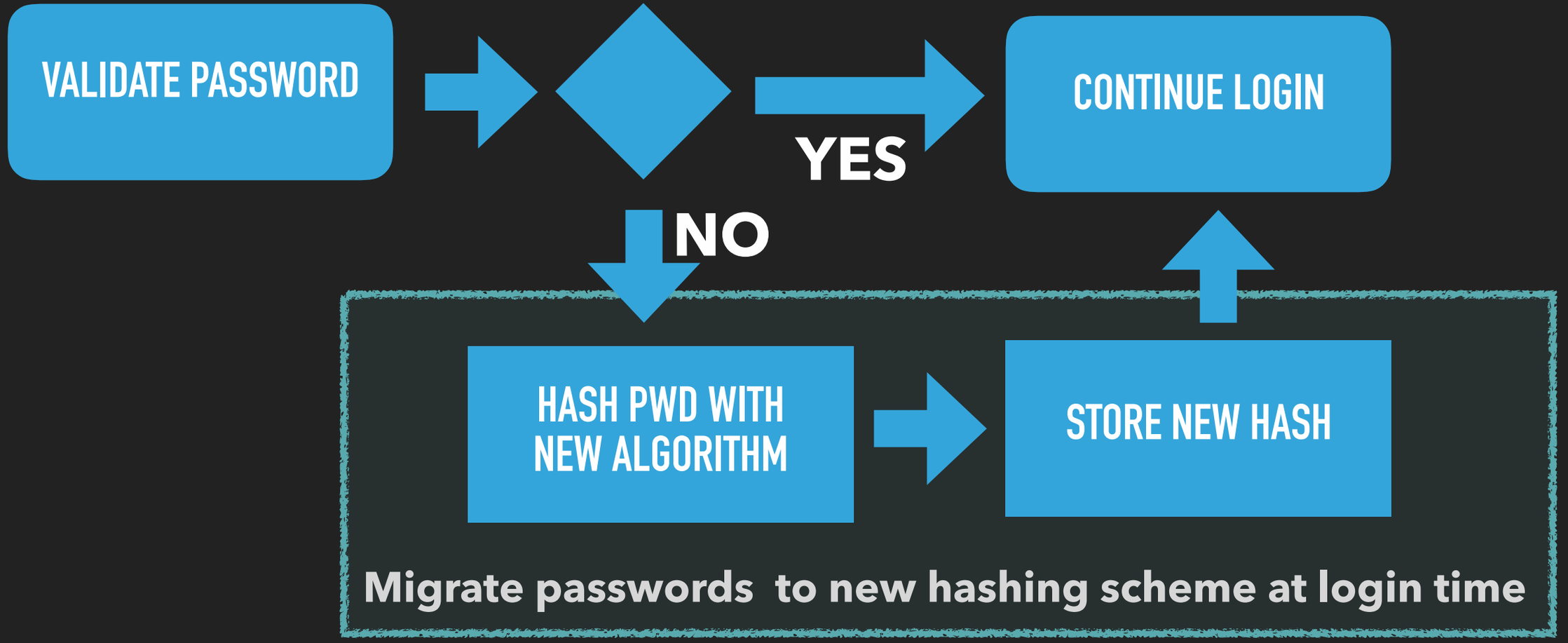


LOGIN

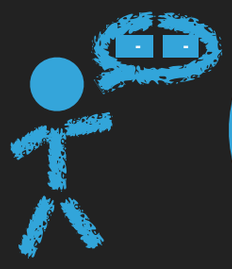
USER
PASSWORD

- Online migration of passwords to a new hashing scheme

Already migrated?



USER LOGIN: MIGRATE PASSWORDS

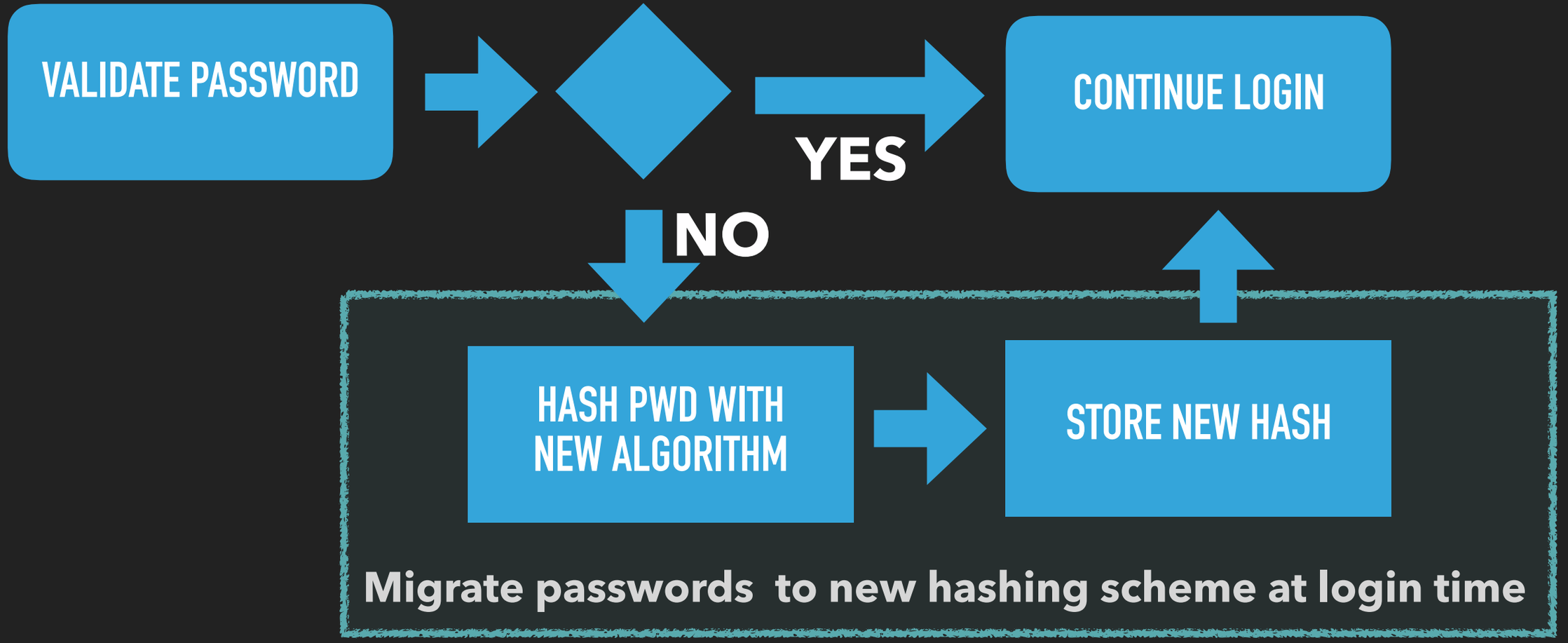


LOGIN

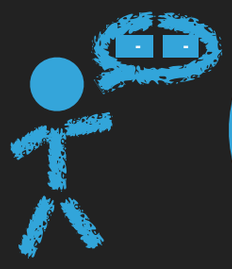
USER
PASSWORD

- Online migration of passwords to a new hashing scheme

Already migrated?



USER LOGIN: MIGRATE PASSWORDS

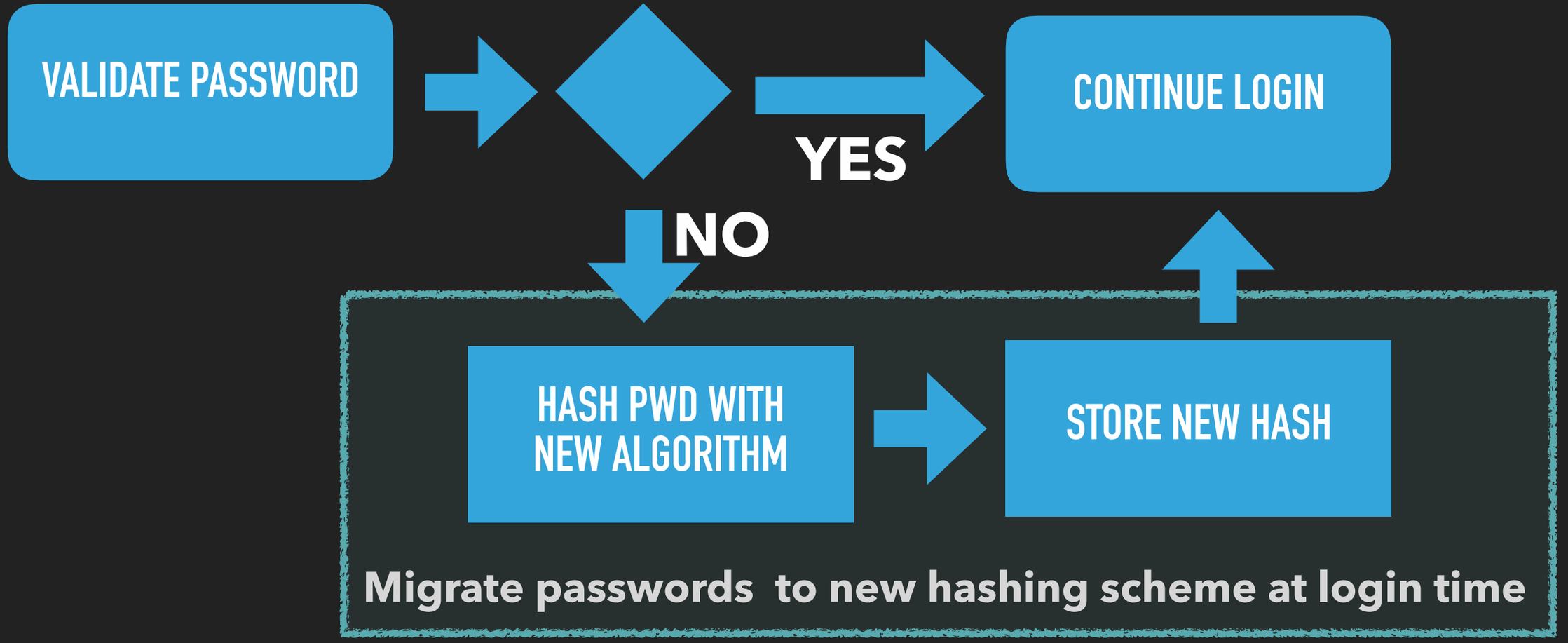


LOGIN

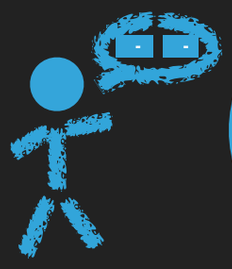
USER
PASSWORD

- Online migration of passwords to a new hashing scheme

Already migrated?



USER LOGIN: MIGRATE PASSWORDS

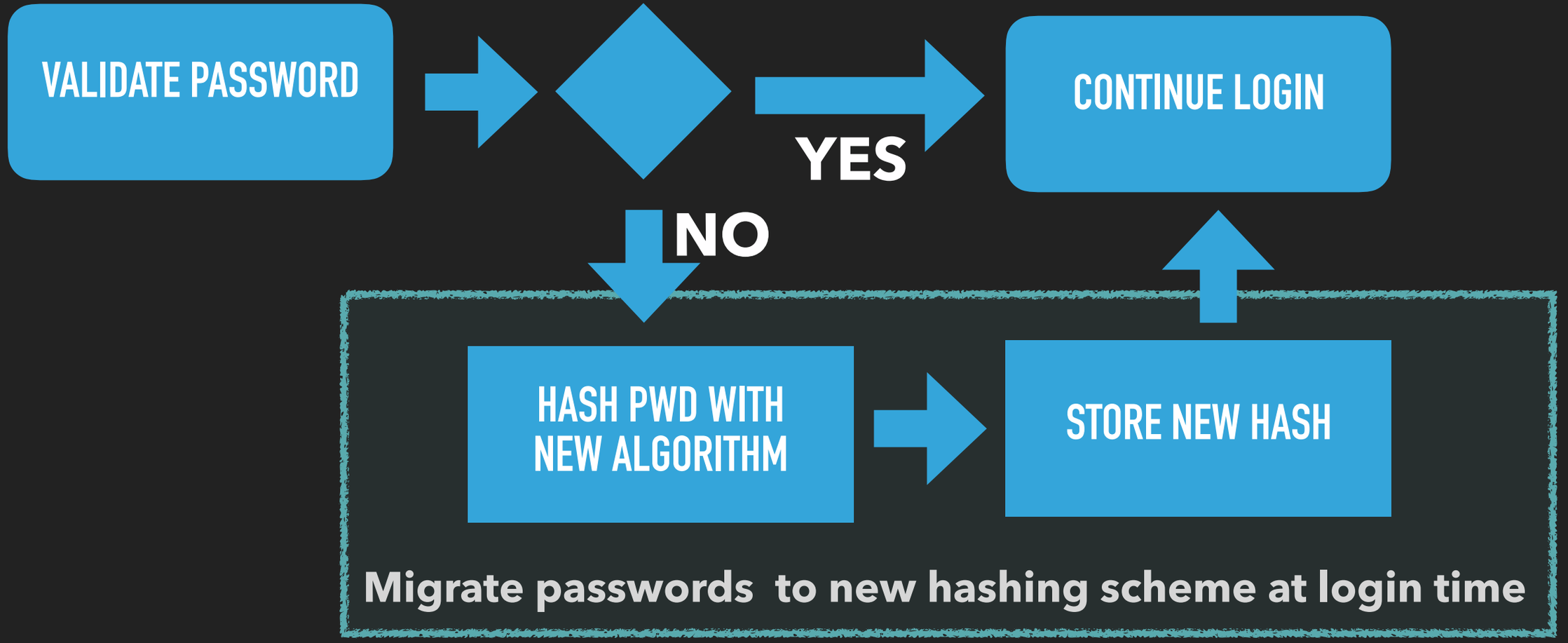


LOGIN

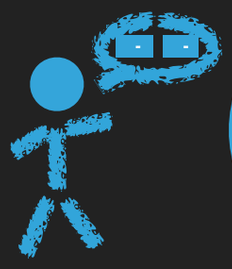
USER
PASSWORD

- Online migration of passwords to a new hashing scheme

Already migrated?



USER LOGIN: MIGRATE PASSWORDS

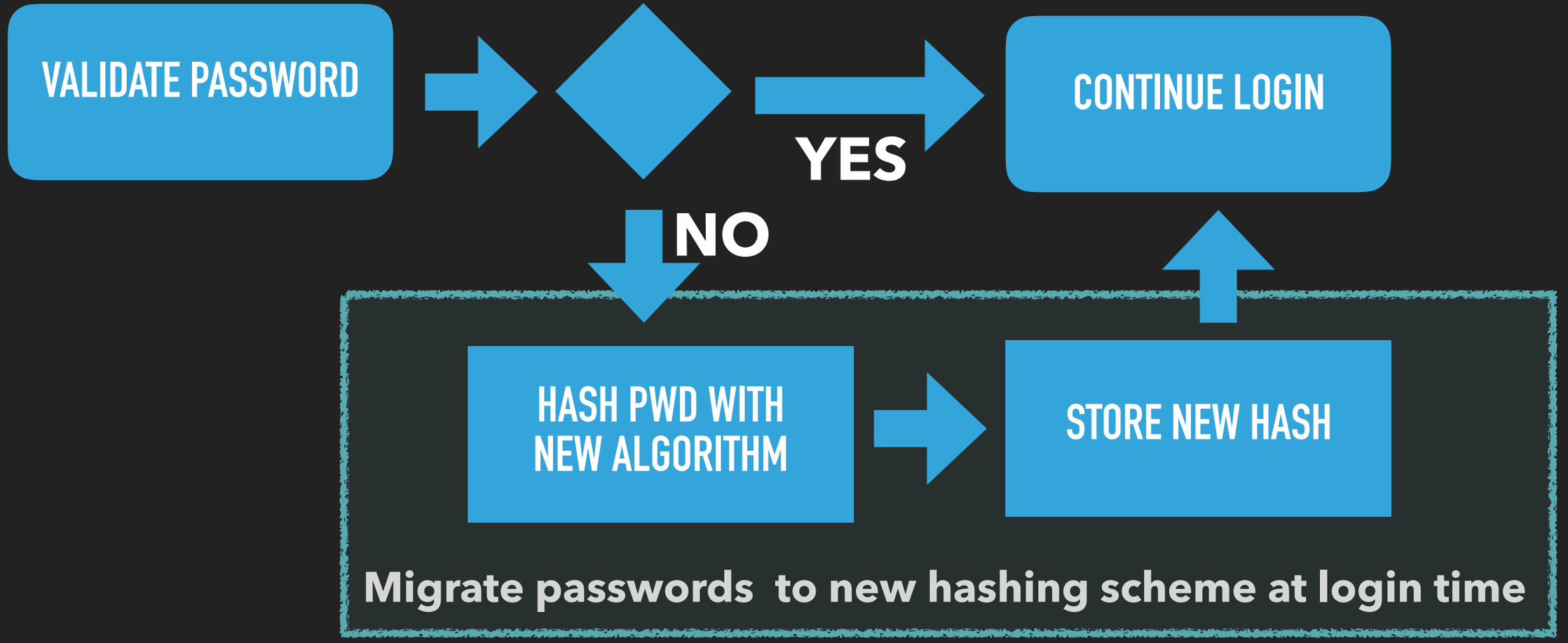


LOGIN

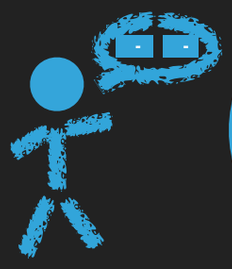
USER
PASSWORD

- Online migration of passwords to a new hashing scheme

Already migrated?



USER LOGIN: MIGRATE PASSWORDS

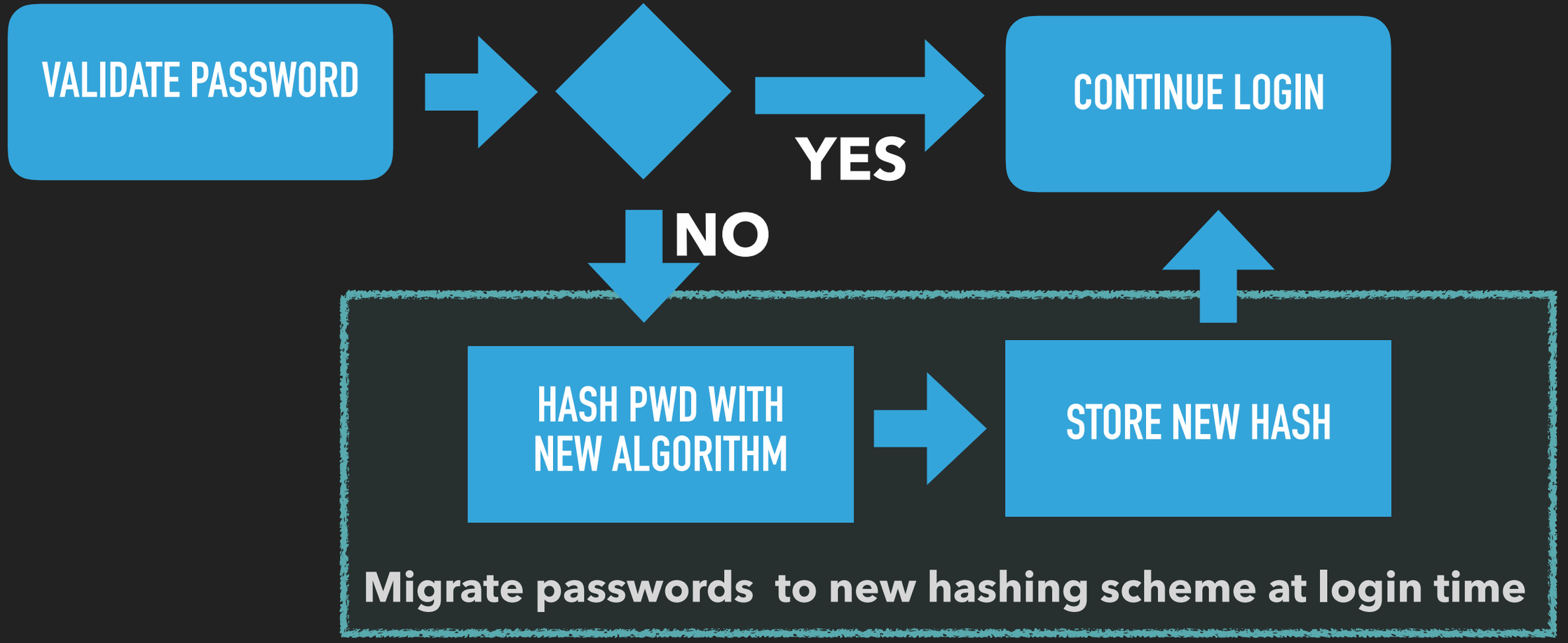


LOGIN

USER
PASSWORD

- Online migration of passwords to a new hashing scheme

Already migrated?



USER LOGIN: MIGRATE PASSWORDS

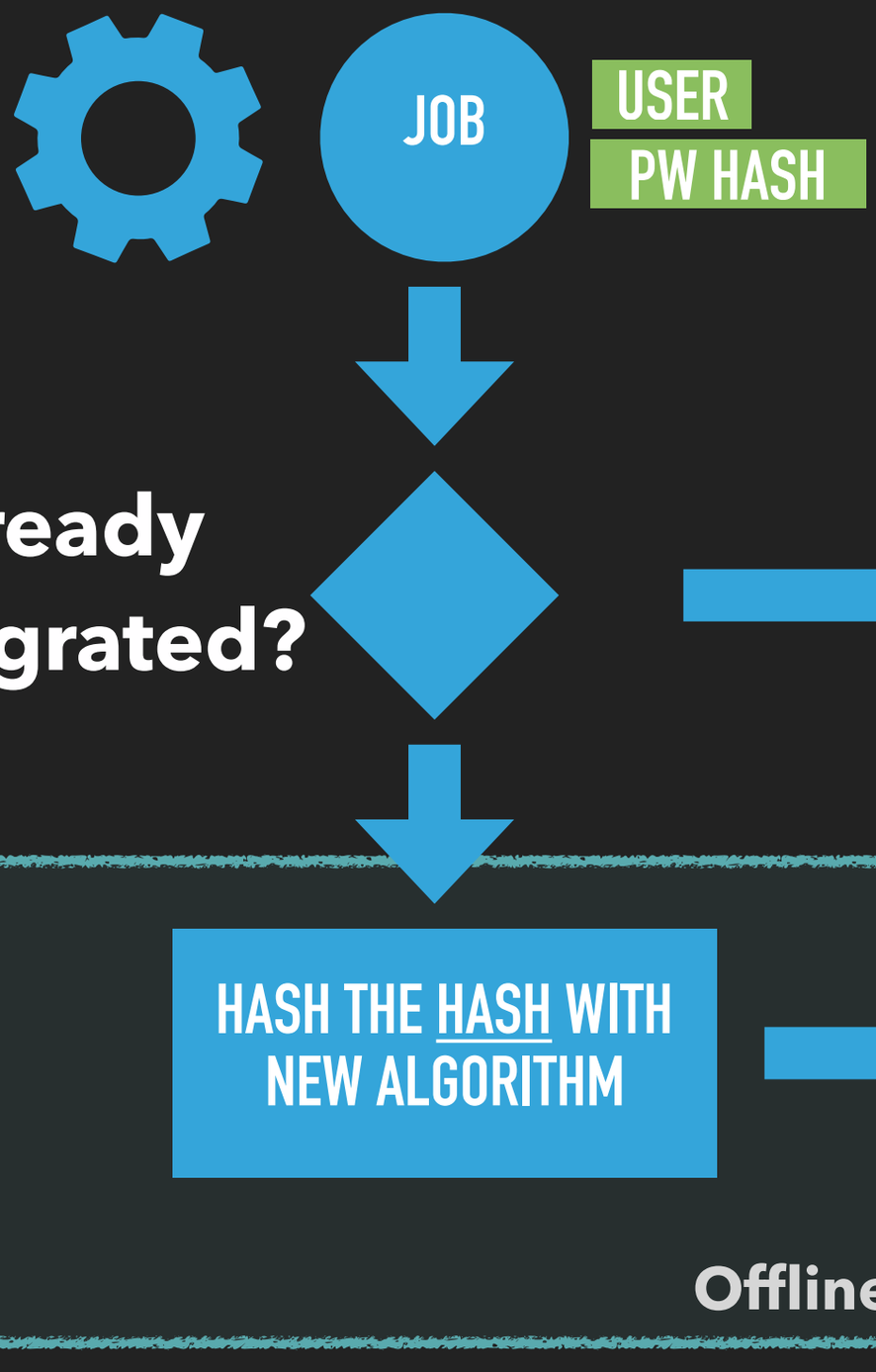
User	Algorithm	Hash	Last Login
SCOTT	PBKDF2(PWD)	3959dc9…	Now
PETER	MD5(PWD)	…	2 years ago
…	MD5(PWD)	…	4 months ago
…	MD5(PWD)	…	…
…	MD5(PWD)	…	…
…	MD5(PWD)	…	…

USER LOGIN: MIGRATE PASSWORDS

User	Algorithm	Hash	Last Login
SCOTT	**PBKDF2(PWD)**	3959dc9…	Now
PETER	**MD5(PWD)**	…	2 years ago
…	**MD5(PWD)**	…	4 months ago
…	**MD5(PWD)**	…	…
…	**MD5(PWD)**	…	…
…	**MD5(PWD)**	…	…

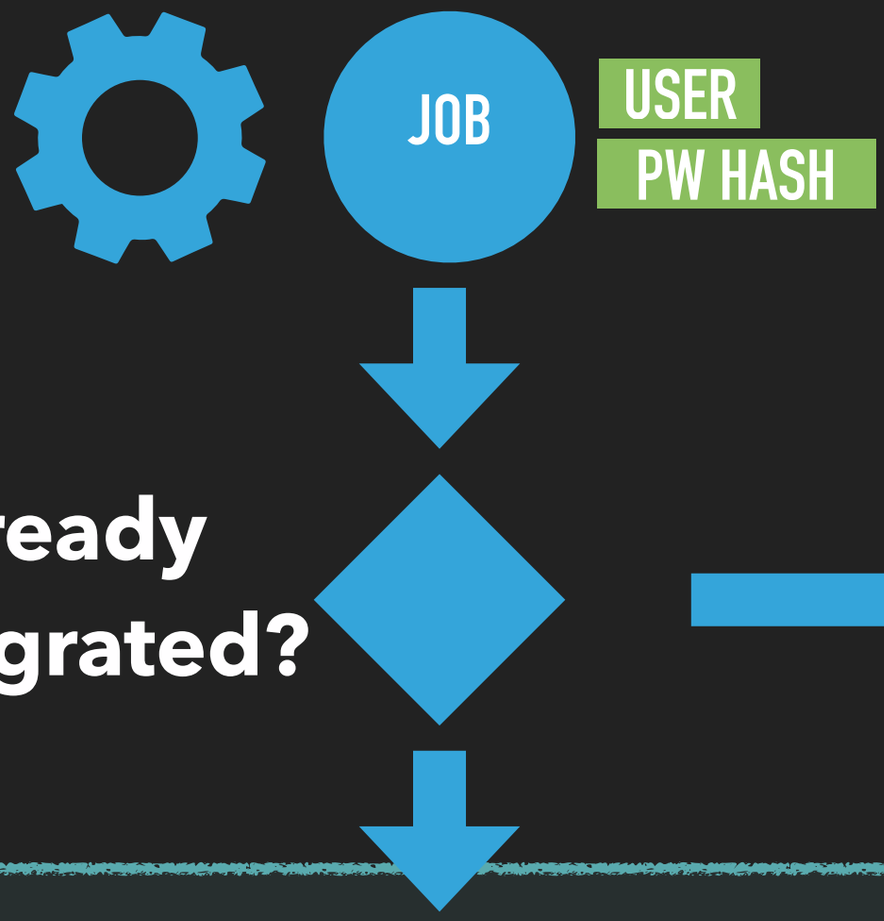
- **How to migrate passwords of users that do not login frequently?**

USER LOGIN: MIGRATE PASSWORDS



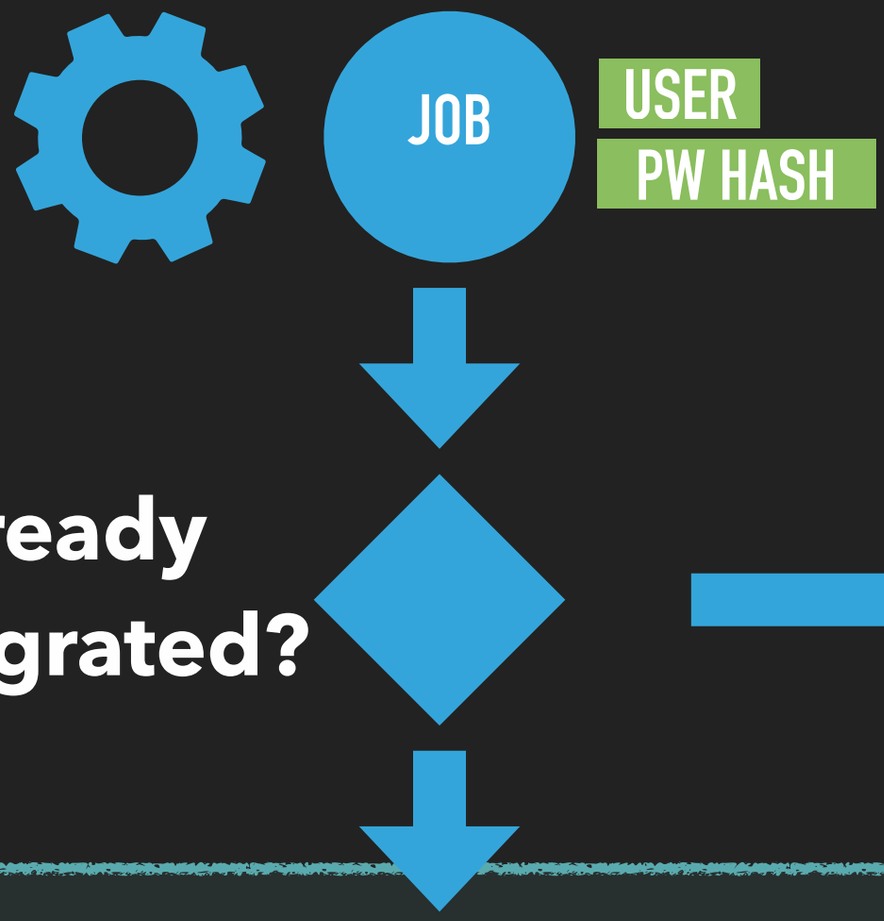
USER LOGIN: MIGRATE PASSWORDS

- Offline migrate passwords to new hashing schemes without user interaction



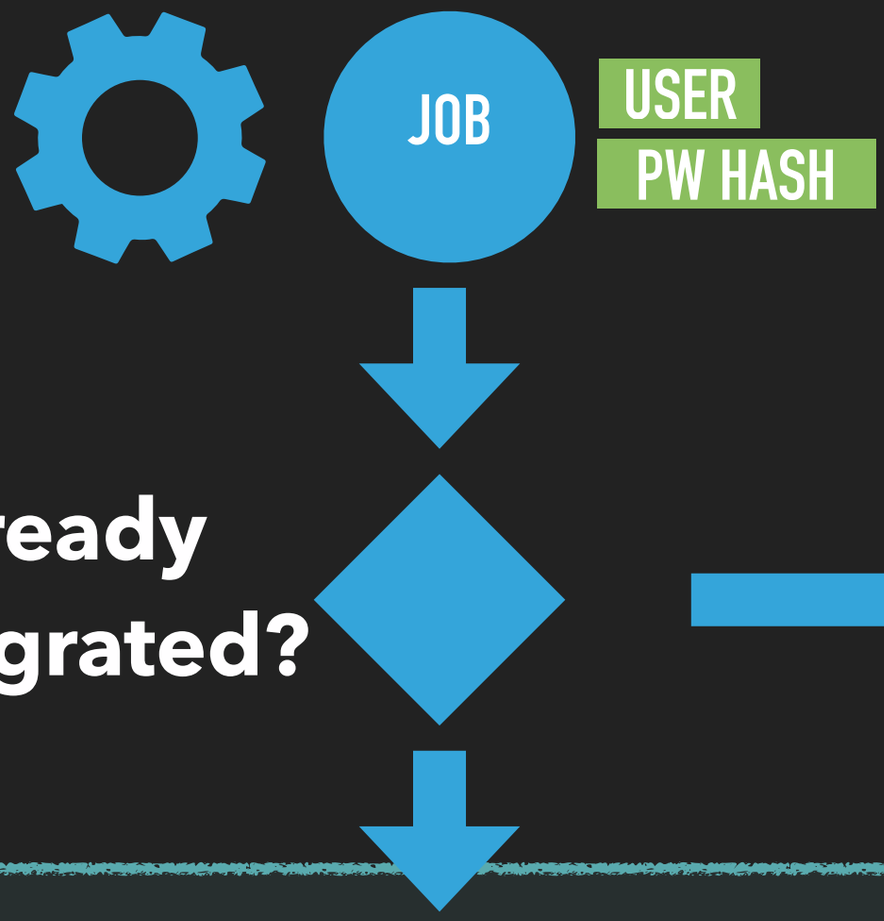
USER LOGIN: MIGRATE PASSWORDS

- Offline migrate passwords to new hashing schemes without user interaction



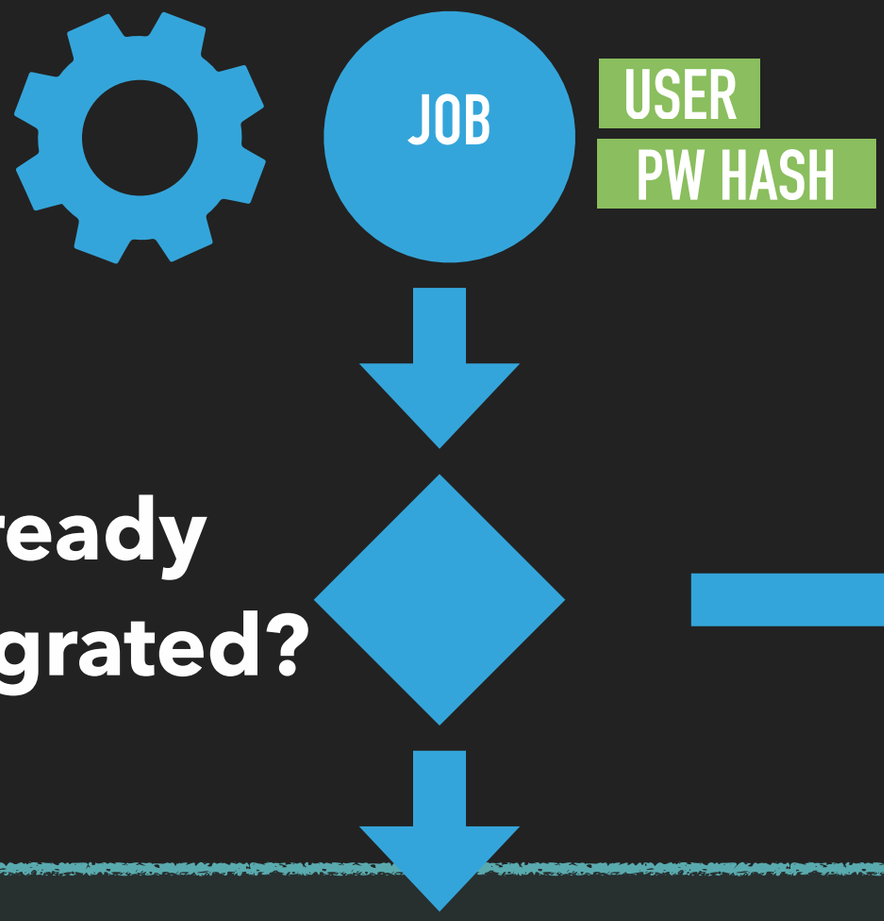
USER LOGIN: MIGRATE PASSWORDS

- Offline migrate passwords to new hashing schemes without user interaction



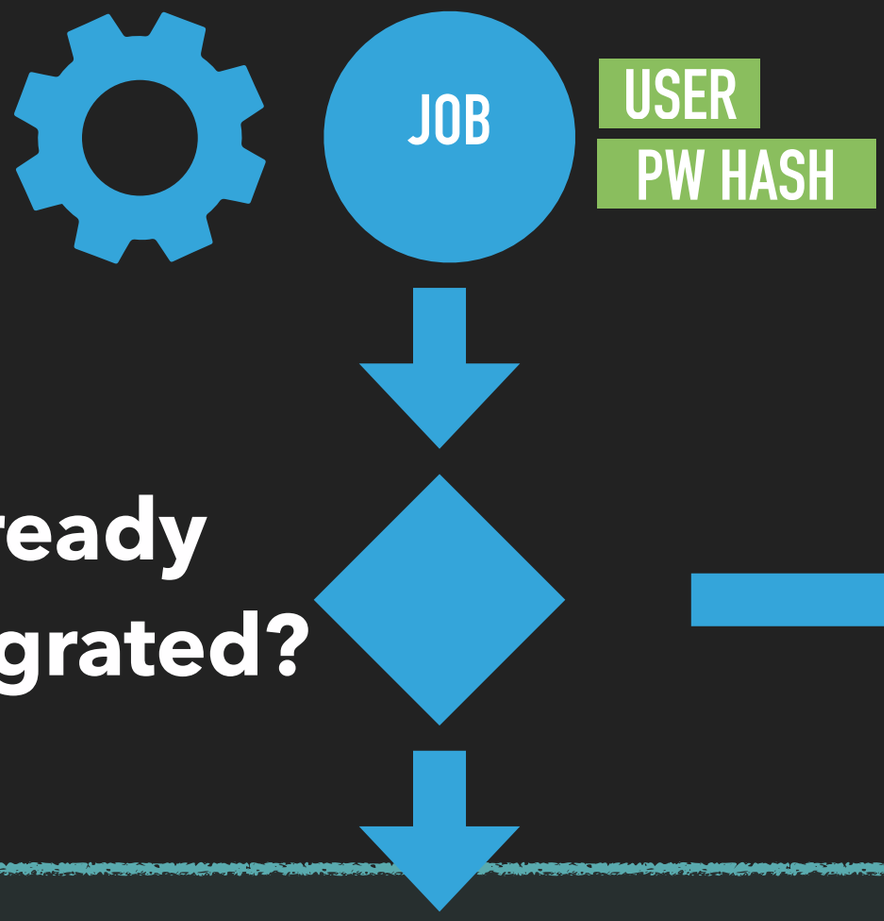
USER LOGIN: MIGRATE PASSWORDS

- Offline migrate passwords to new hashing schemes without user interaction



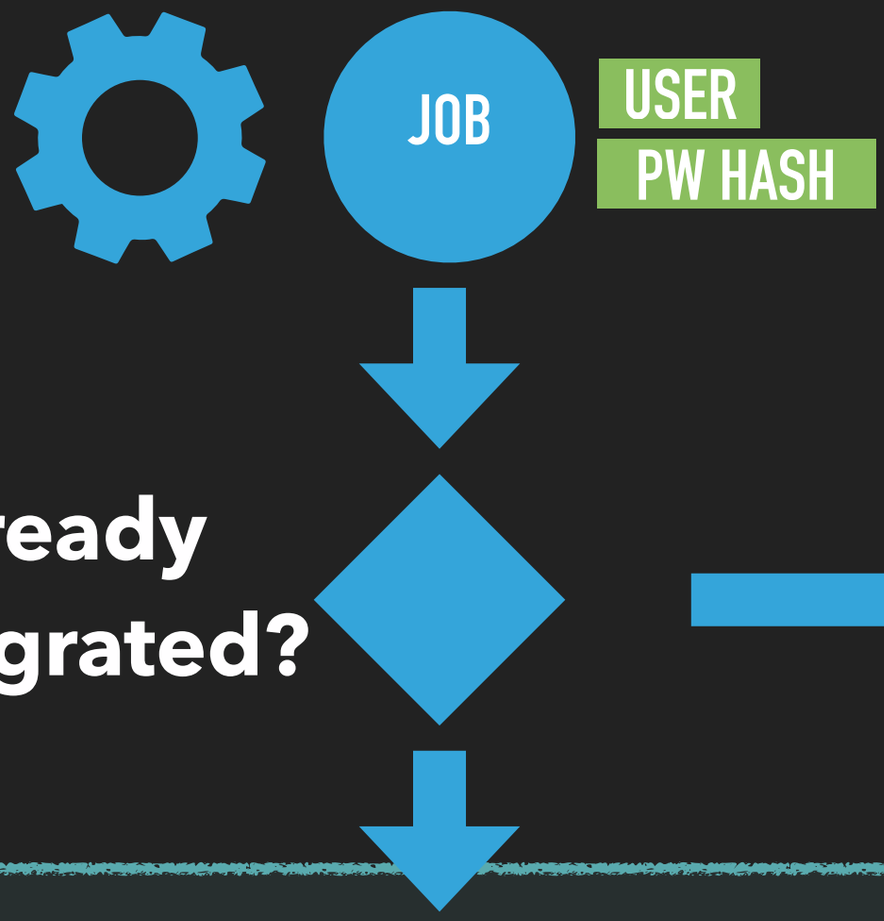
USER LOGIN: MIGRATE PASSWORDS

- Offline migrate passwords to new hashing schemes without user interaction



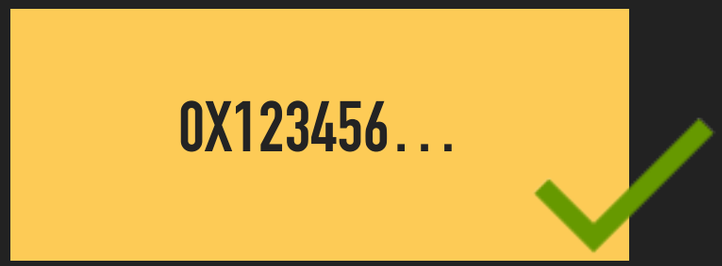
USER LOGIN: MIGRATE PASSWORDS

- Offline migrate passwords to new hashing schemes without user interaction



USER LOGIN: MIGRATE PASSWORDS

User	Algorithm	Hash	Last Login
SCOTT	PBKDF2(PWD)	3959dc9…	Now
PETER	PBKDF2(MD5(PWD))	…	2 years ago
…	PBKDF2(MD5(PWD))	…	4 months ago
…	PBKDF2(MD5(PWD))	…	…
…	PBKDF2(MD5(PWD))	…	…
…	PBKDF2(MD5(PWD))	…	…



PATTERNS

INTEGRITY

DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Sensitive data can be manipulated.

	User	Salary	…
	Alice	3,141€	…
	Eve	2,718 €	…
	…	…	…

DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Sensitive data can be manipulated.

	User	Salary	…
	Alice	3,141€	…
	Eve	10,000 €	…
	…	…	…

DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Sensitive data can be manipulated.

	User	Salary	…
	Alice	3,141€	…
	Eve	10,000 €	…
…			…

EVE GETS AN INSTANT PROMOTION

DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Sensitive data can be manipulated.

Solution: Use cryptographic checksums with a secret.

User	Salary	MAC*
Alice	3,141€	0x4711…
Eve	2,718 €	0xabcd…
…	…	…

* Checksum with a secret: hmac, AEAD, public key signatures

DATA INTEGRITY & ASSOCIATION

0X123456...

Problem: Sensitive data can be manipulated.

Solution: Use cryptographic checksums with a secret.

User	Salary	MAC*
Alice	3,141€	0x4711...
Eve	10,000 €	0xabcd...
...	...	...

* Checksum with a secret: hmac, AEAD, public key signatures

DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Sensitive data can be manipulated.

Solution: Use cryptographic checksums with a secret.

User	Salary	MAC*
Alice	3,141€	0x4711…
Eve	10,000 €	0xabcd…

THE CHECKSUMS DON'T MATCH, PROMOTION IS DECLINED

* Checksum with a secret: hmac, AEAD, public key signatures

DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Protected data can be “replayed”.

	User	Salary	MAC*
	Alice	3,141€	0x4711…
	Eve	2,718 €	0xabcd…
	…	…	…

DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Protected data can be “replayed”.

	User	Salary	MAC*
	Alice	3,141€	0x4711…
	Eve	3,141€	0x4711…
	…	…	…

DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Protected data can be “replayed”.

	User	Salary	MAC*
	Alice	3,141€	0x4711…
	Eve	3,141€	0x4711…

…

EVE GETS AN INSTANT PROMOTION

…

DATA INTEGRITY & ASSOCIATION

0X123456...

Problem: Protected data can be “replayed”.

Solution: Cryptographically bind data to context.

	User	Salary	MAC*
	Alice	3,141€	0xabcd...
	Eve	2,718€	0x9876...
	...	...	...

* including the username in the hash/hmac

DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Protected data can be “replayed”.

Solution: Cryptographically bind data to context.

	User	Salary	MAC*
	Alice	3,141€	0xabcd…
	Eve	3,141€	0xabcd…
	…	…	…

* including the username in the hash/hmac

DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Protected data can be “replayed”.

Solution: Cryptographically bind data to context.

	User	Salary	MAC*
	Alice	3,141€	0xabcd…
	Eve	3,141€	0xabcd…

THE CHECKSUMS DON’T MATCH, PROMOTION IS DECLINED

* including the username in the hash/hmac

DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Protected data can be “replayed”.

Solution: Cryptographically bind data to context.

User	Password Hash *	…
Alice	0X123456…	…
Eve	0XABCDEF…	…
…	…	…

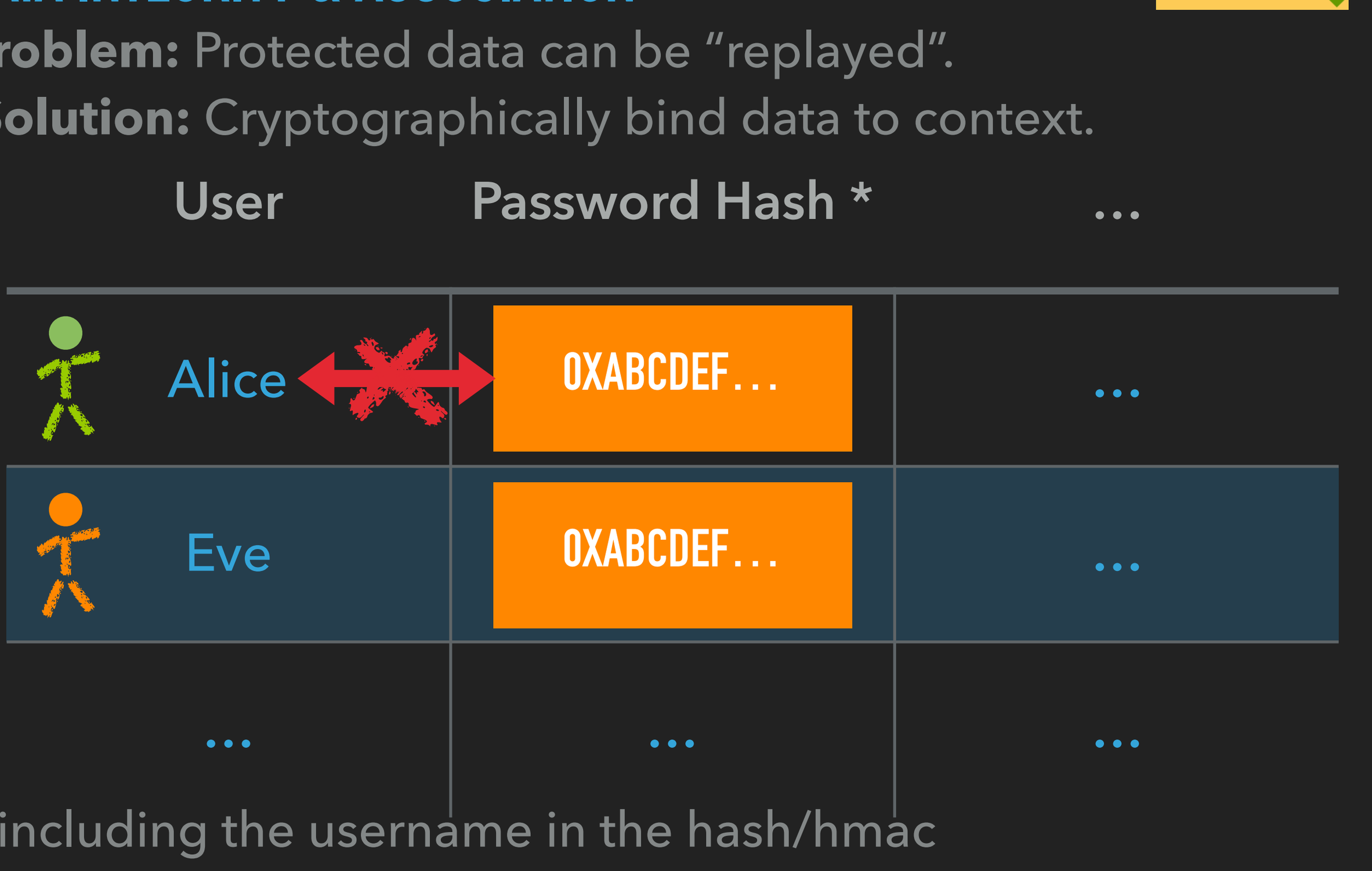
* including the username in the hash/hmac

DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Protected data can be “replayed”.

Solution: Cryptographically bind data to context.



* including the username in the hash/hmac

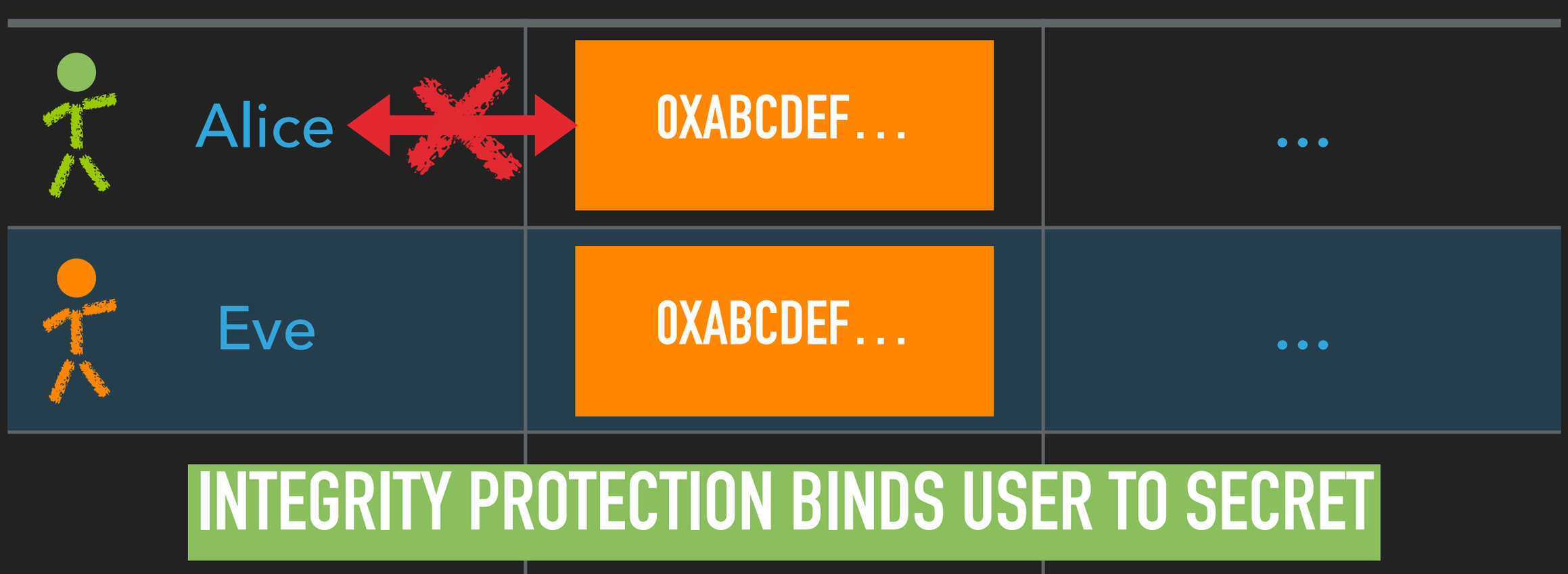
DATA INTEGRITY & ASSOCIATION

0X123456…

Problem: Protected data can be “replayed”.

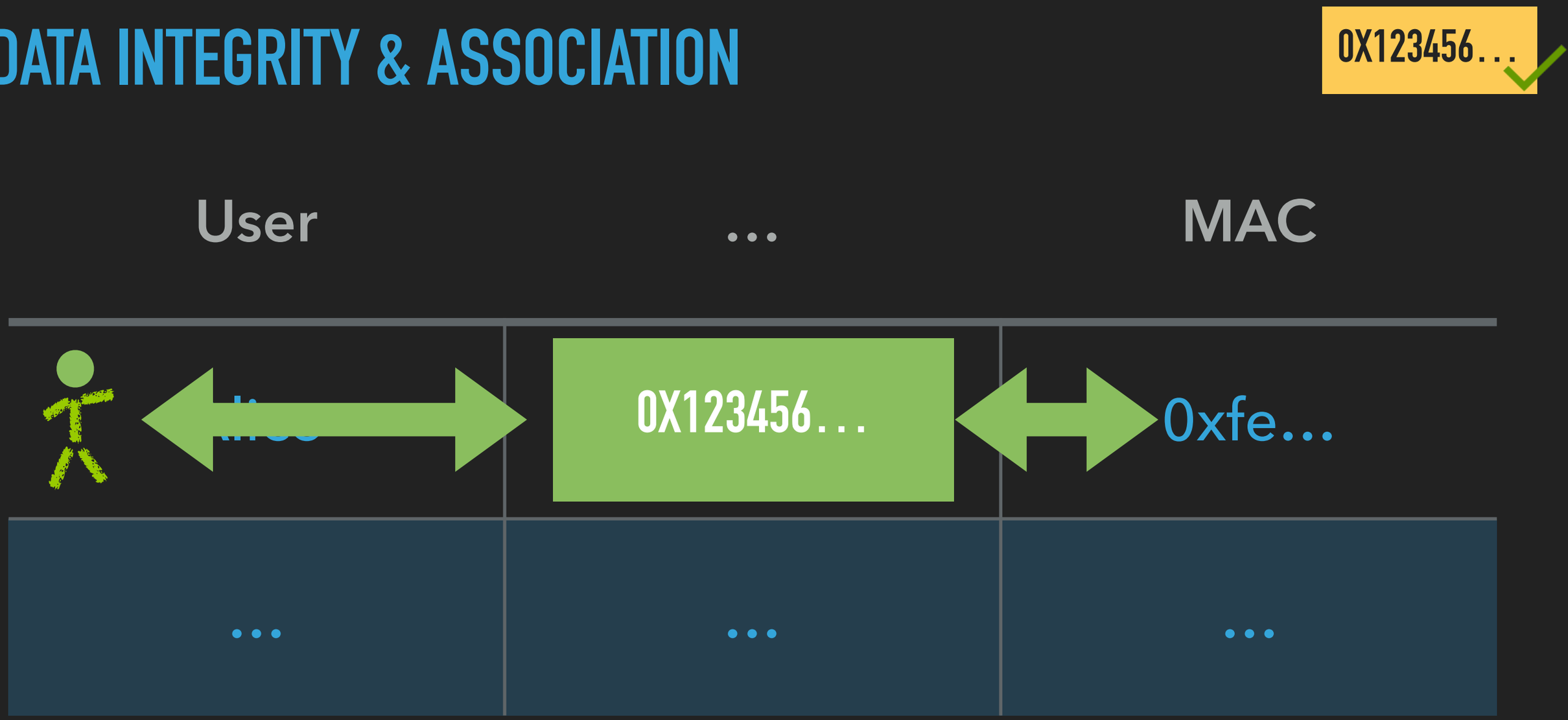
Solution: Cryptographically bind data to context.

User Password Hash * …



* including the username in the hash/hmac

DATA INTEGRITY & ASSOCIATION



- Add integrity checks to the data (HMAC, AEAD encryption, signatures)
- Include an association (here: “User”) in the integrity check

PATTERNS

MOVING

MOVE DATA BETWEEN PARTIES

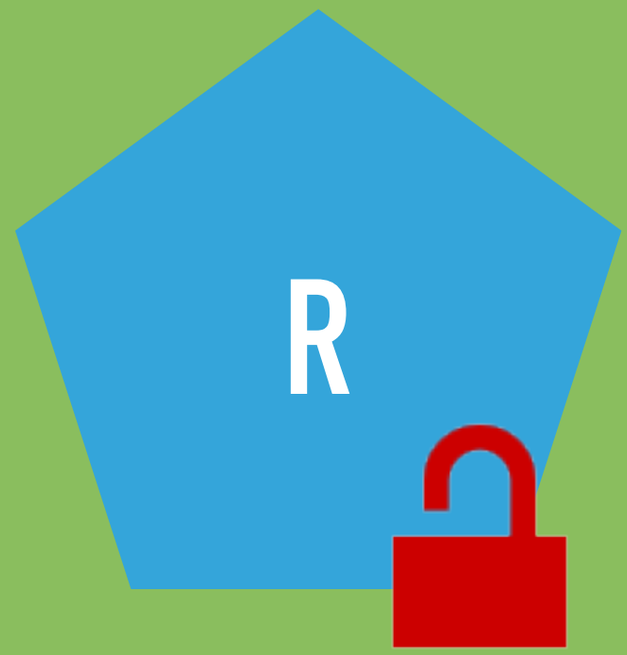
Problem: Data is exchanged between parties.

MOVE DATA BETWEEN PARTIES

Problem: Data is exchanged between parties.

SENDER

STORE & FORWARD

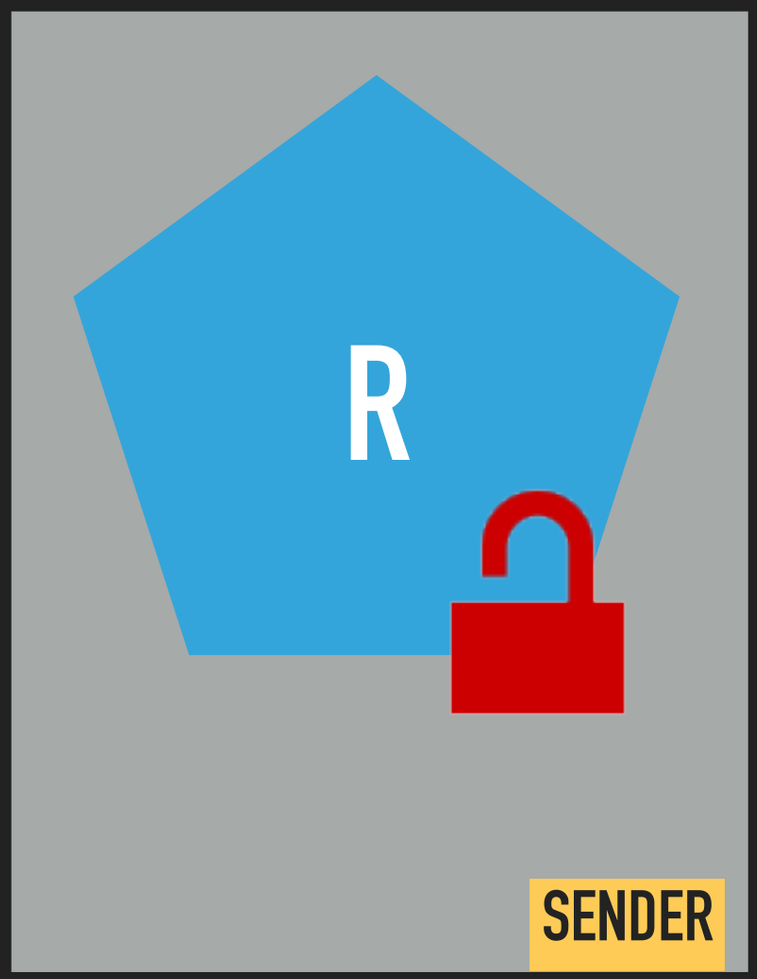


RECIPIENT

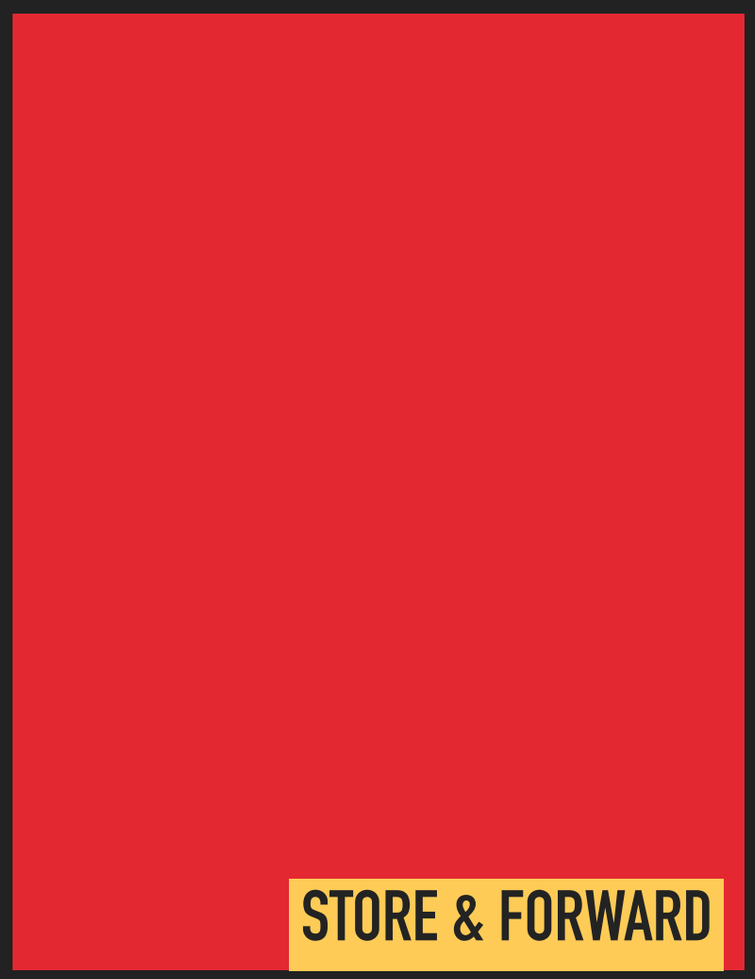
MOVE DATA BETWEEN PARTIES

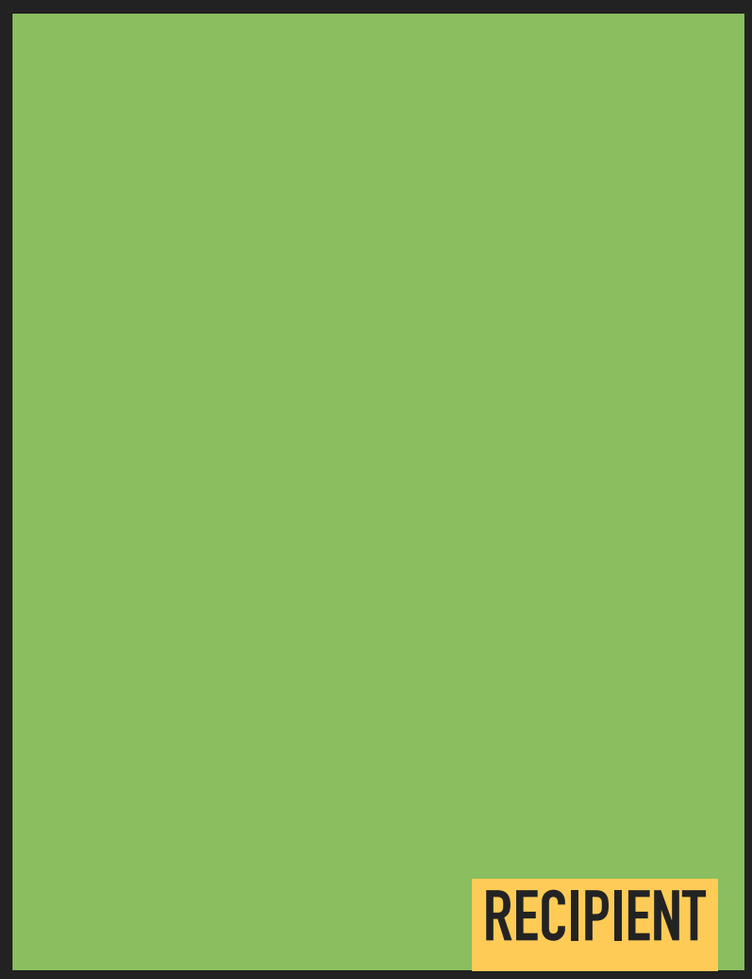
Problem: Data is exchanged between parties.

Solution: Use public key cryptography to protect data.



Encrypt with public key





Decrypt with private key

NO SECRETS ON SENDER / STORE & FORWARD SERVER

MOVE DATA BETWEEN PARTIES

Problem: Data is exchanged between parties.

Solution: Use public key cryptography to protect data.

Encrypt with public key

Decrypt with private key

NO SECRETS ON SENDER / STORE & FORWARD SERVER

MOVE DATA BETWEEN PARTIES

```
KeyringConfig keyringConfig = KeyringConfigs
        .withKeyRingsFromFiles(
         “/…/pubring.gpg”,
         “/…/secring.gpg”,
         withPassword(secKeyRingPassword));
try (
    final InputStream cipherTextStream = Files.newInputStream(sourceFile);
    final OutputStream fileOutput = Files.newOutputStream(destFile);
    final BufferedOutputStream bufferedOut = …

    final InputStream plaintextStream = BouncyGPG
            .decryptAndVerifyStream()
            .withConfig(keyringConfig)
            .andRequireSignatureFromAllKeys(“sender@example.com”)
            .fromEncryptedInputStream(cipherTextStream)
    ) {
            Streams.pipeAll(plaintextStream, bufferedOut);
     }
```

https://github.com/neuhalje/bouncy-gpg/

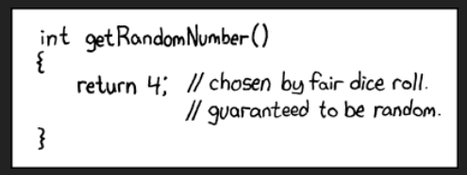
```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

https://xkcd.com/221/

PATTERNS

ENTROPY

ENTROPY



- Bad entropy compromises keys
- Computers are very bad at making things up! (not always)
- Entropy therefore often is limited (esp. after booting!)
- Use what the API provides (SecureRandom)
- **RTFM**

PATTERNS

ACCESS CONTROL

ACCESS CONTROL

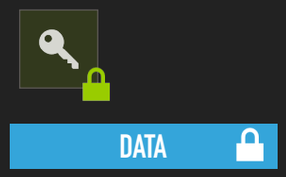
Problem: Make sure that data can only be accessed by some users

Solution: Use cryptographic access controls

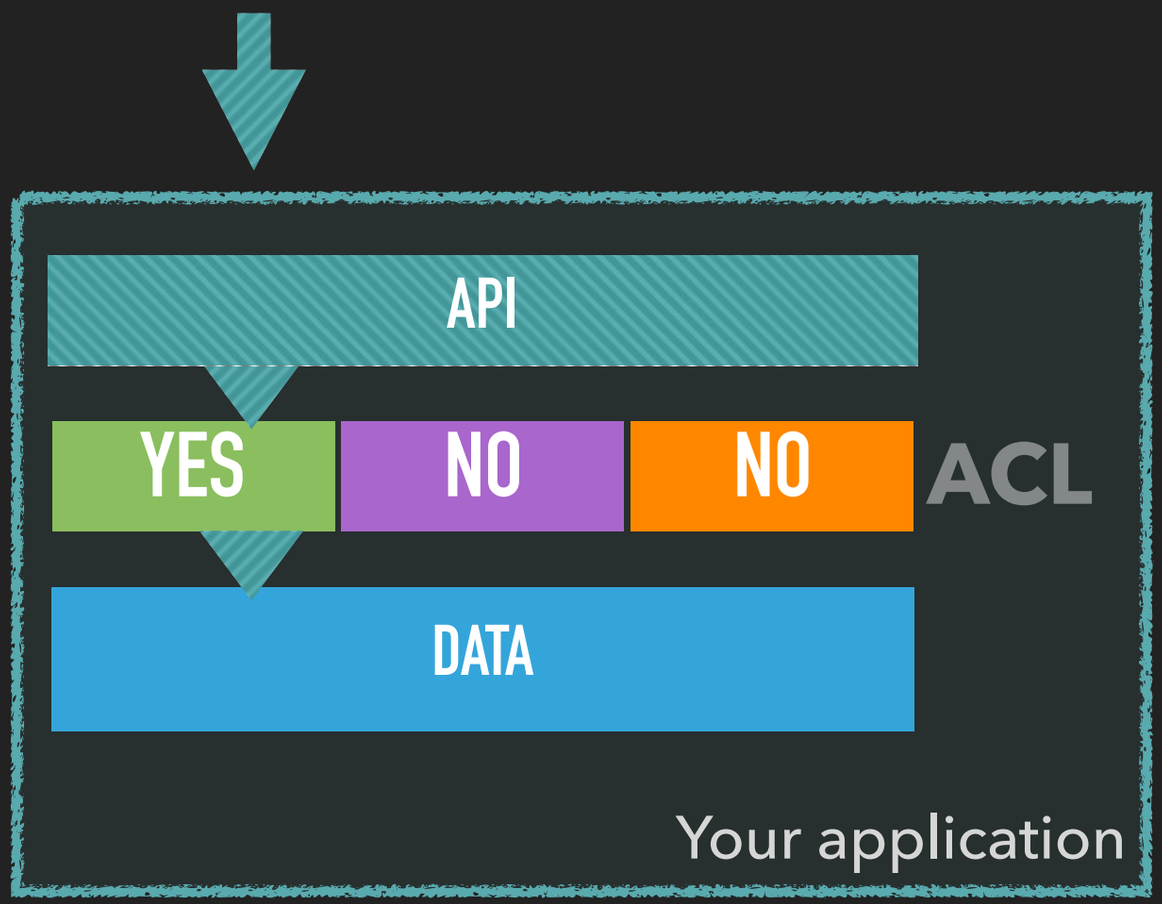
Applies primarily to

- Personal data (e.g. health data, personell records, ...)
- Top Secret data (e.g. company secrets)
- When “provable” access control is required

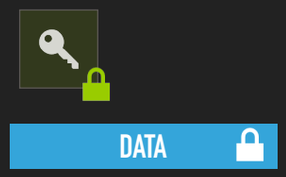
ACCESS CONTROL



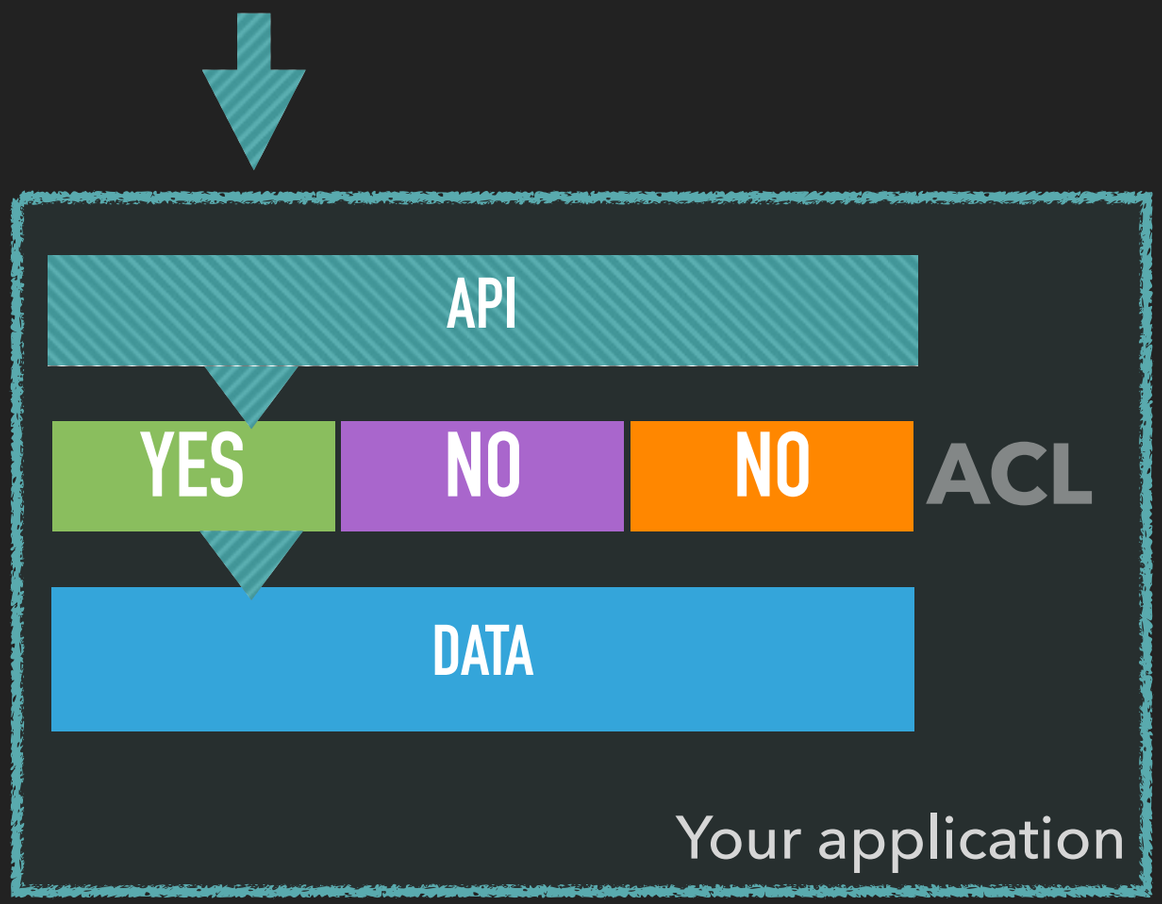
1. Alice tries to read data



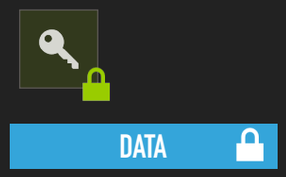
ACCESS CONTROL



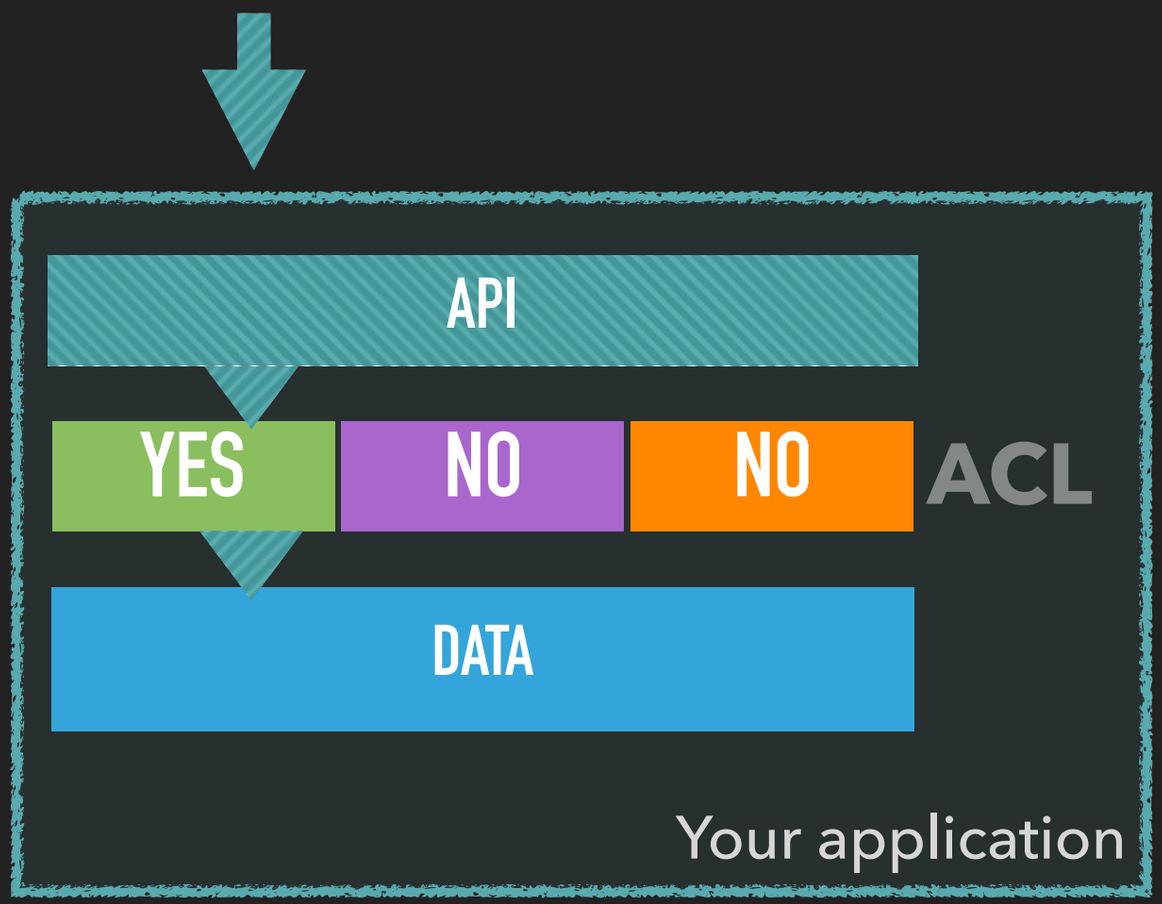
1. Alice tries to read data



ACCESS CONTROL

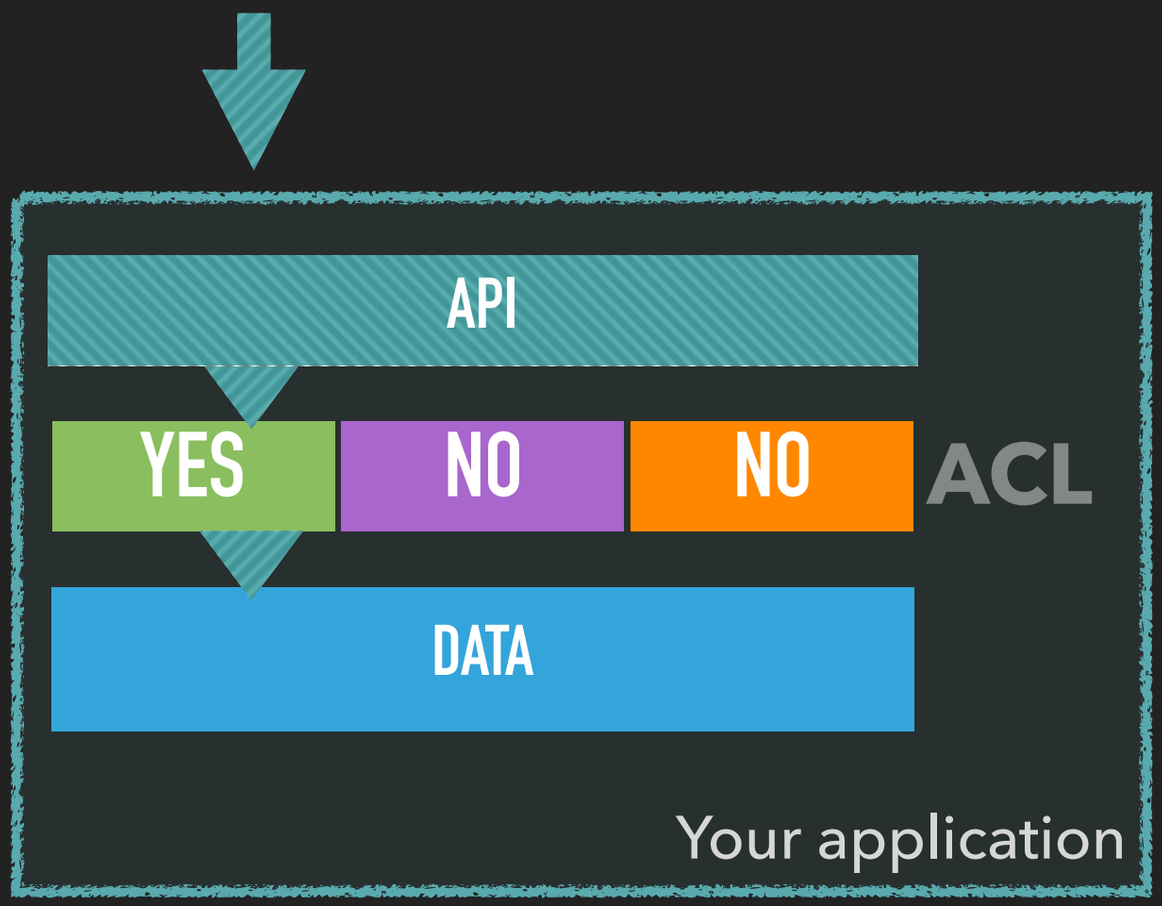


1. Alice tries to read data
2. Application validates ACL



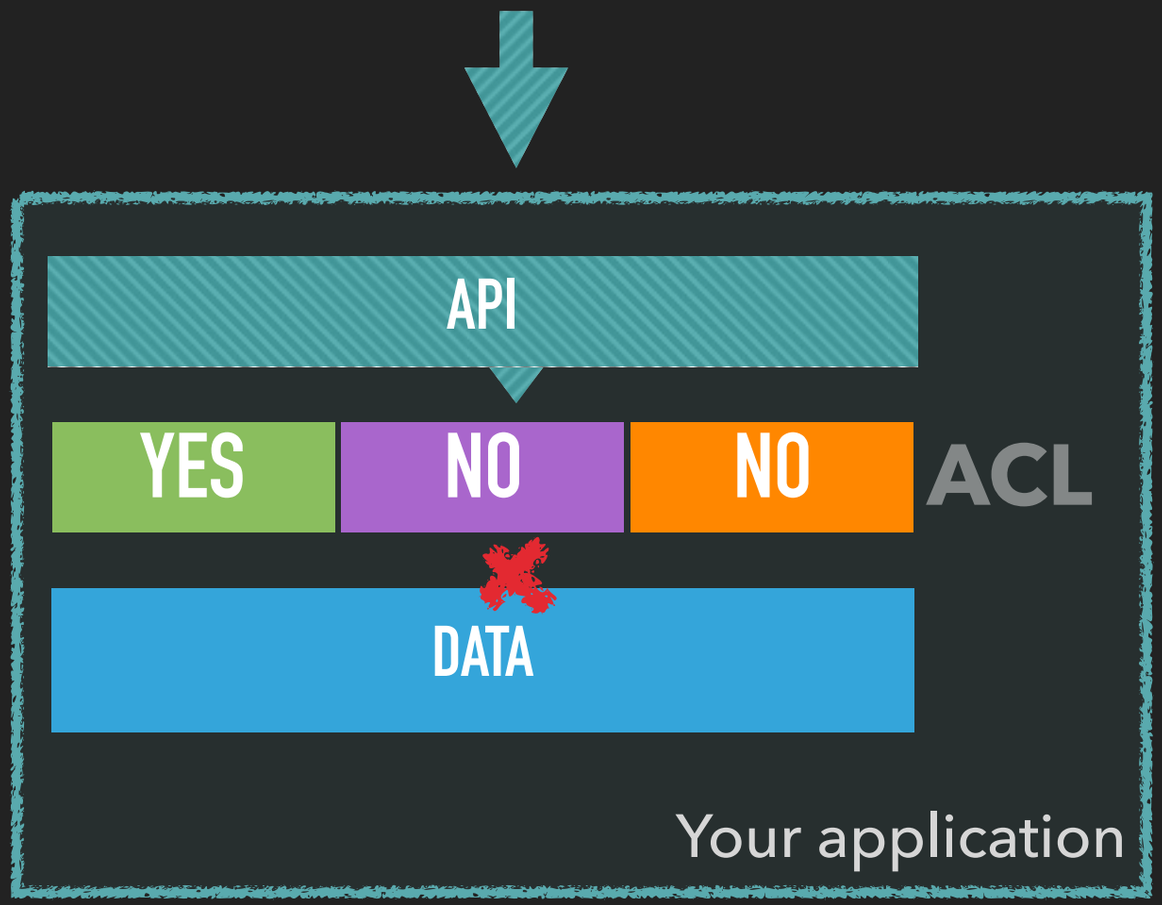
ACCESS CONTROL

1. Alice tries to read data
2. Application validates ACL
3. Alice is granted access



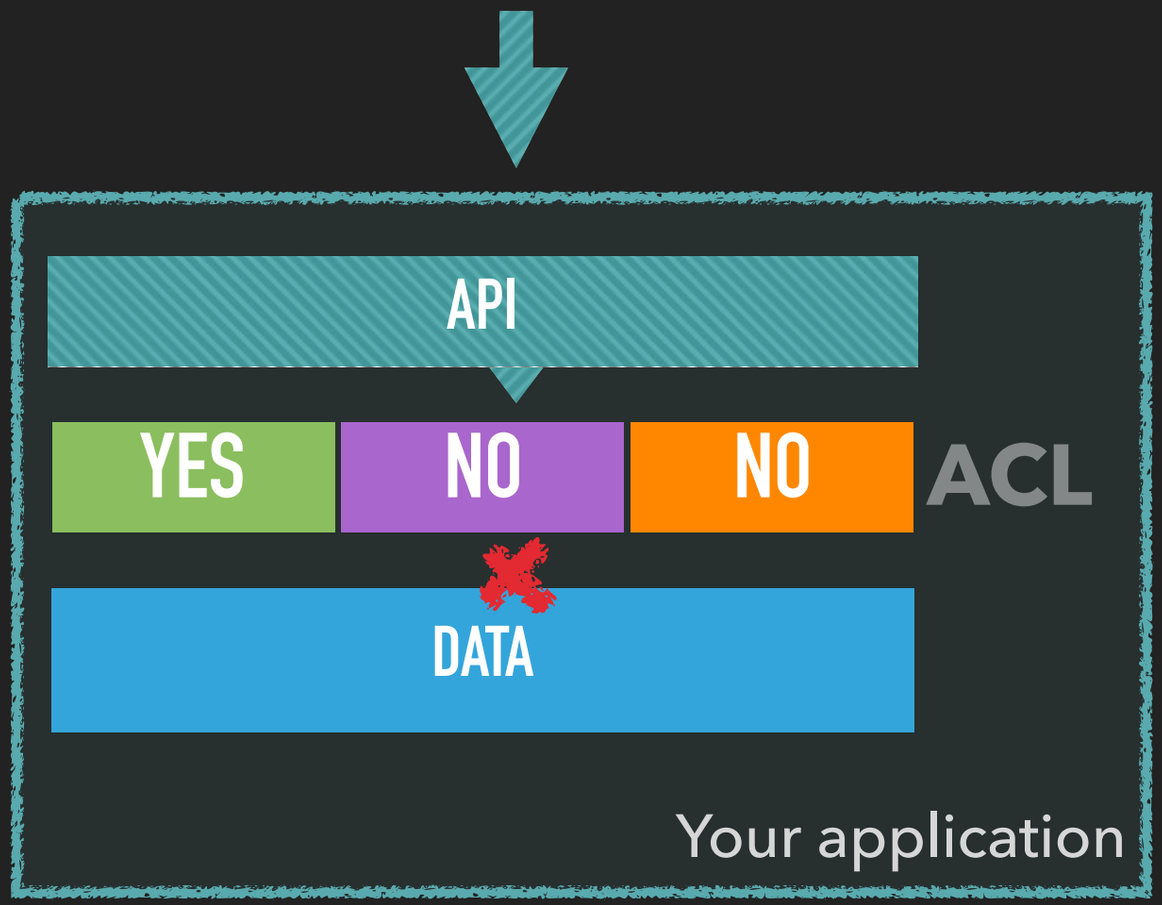
ACCESS CONTROL

1. Bob tries to read data



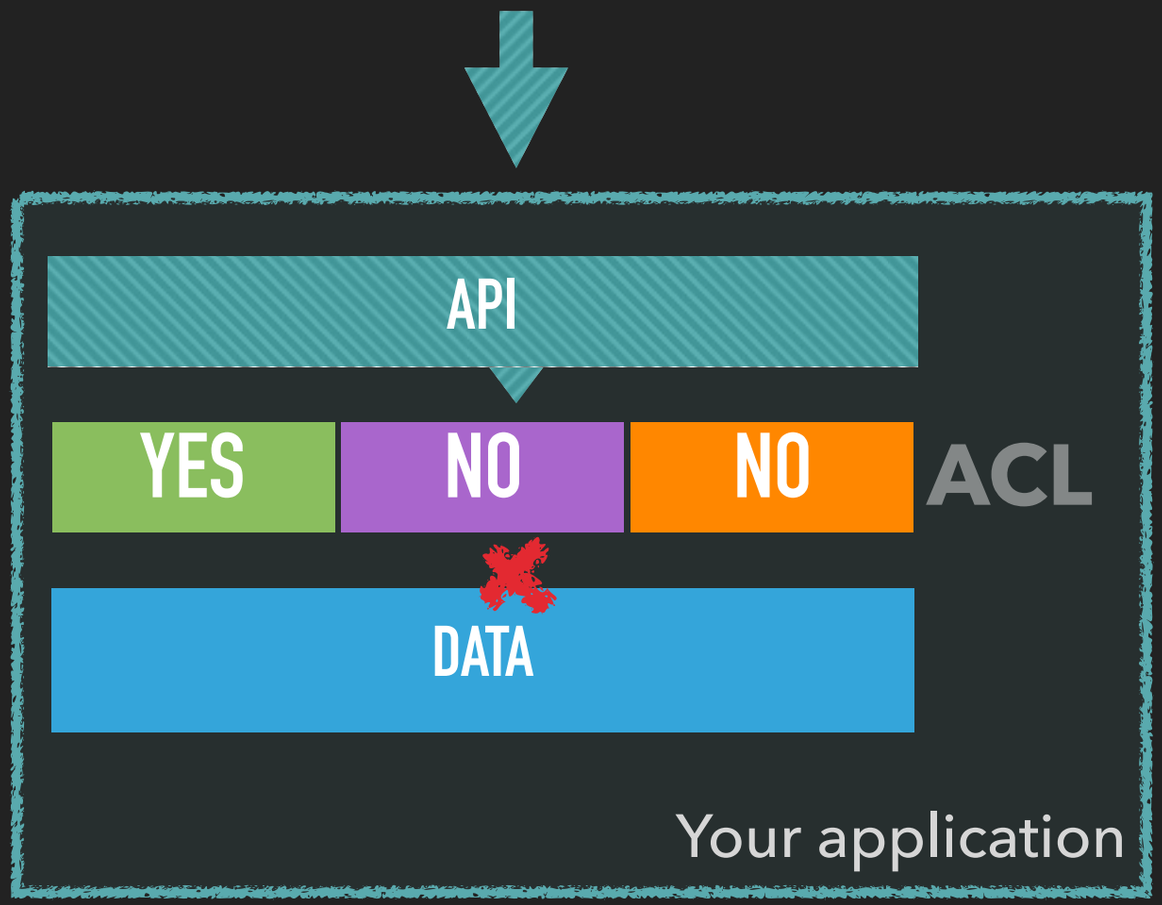
ACCESS CONTROL

1. Bob tries to read data



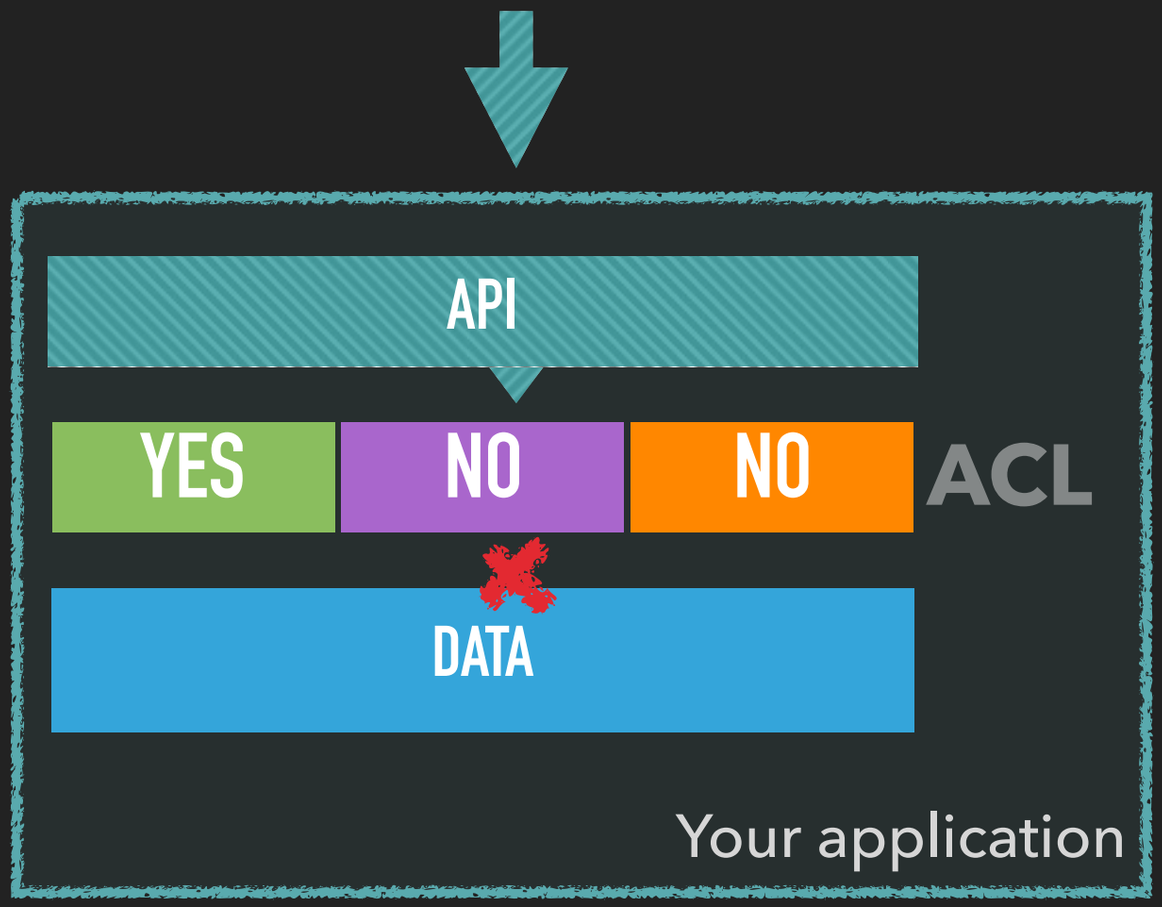
ACCESS CONTROL

1. Bob tries to read data
2. Application validates ACL

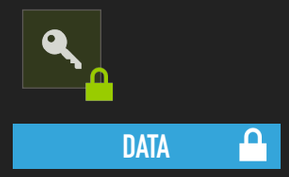


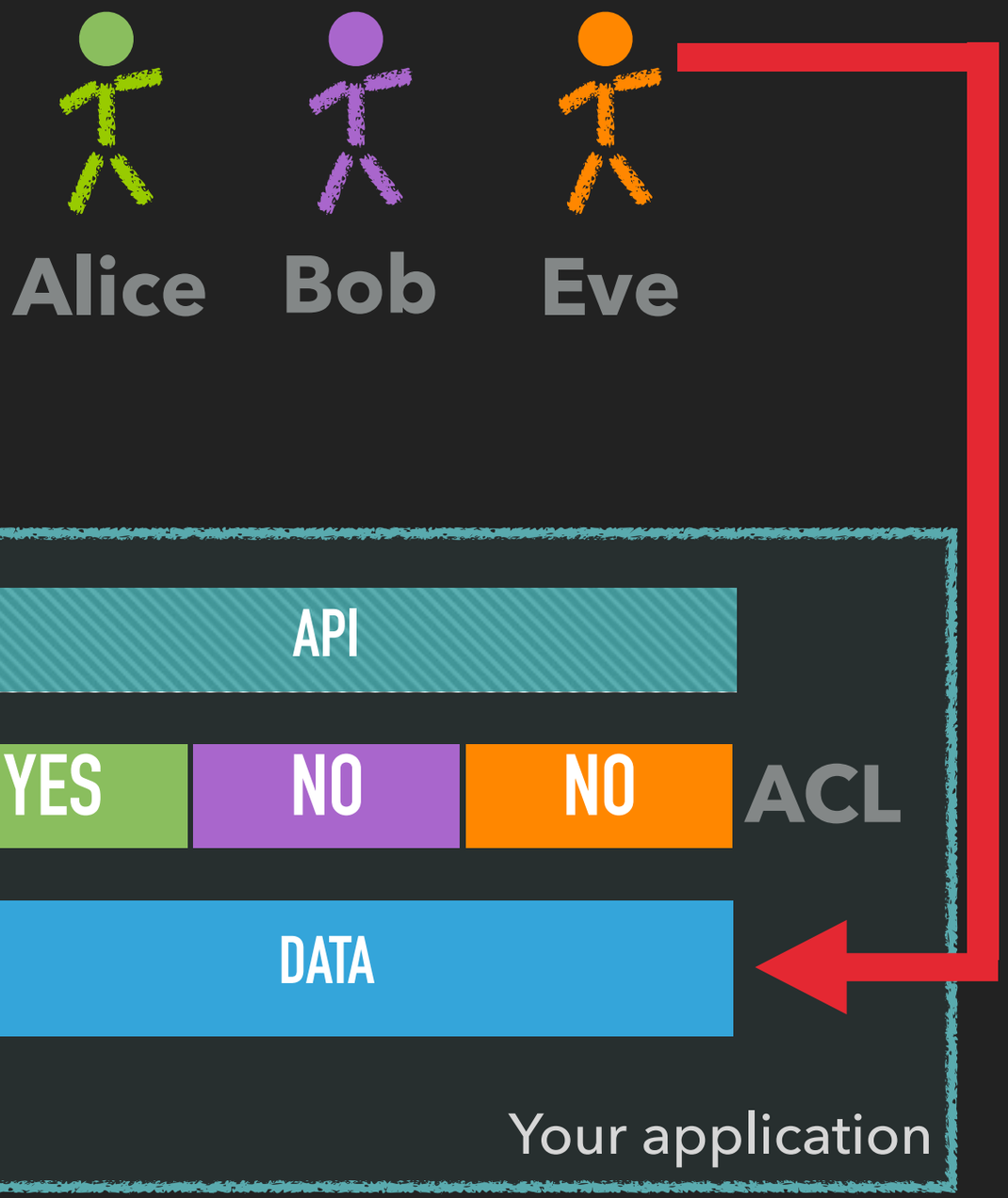
ACCESS CONTROL

1. Bob tries to read data
2. Application validates ACL
3. Access is denied



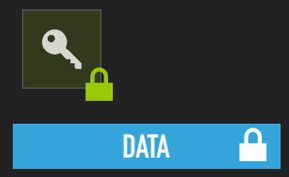
ACCESS CONTROL

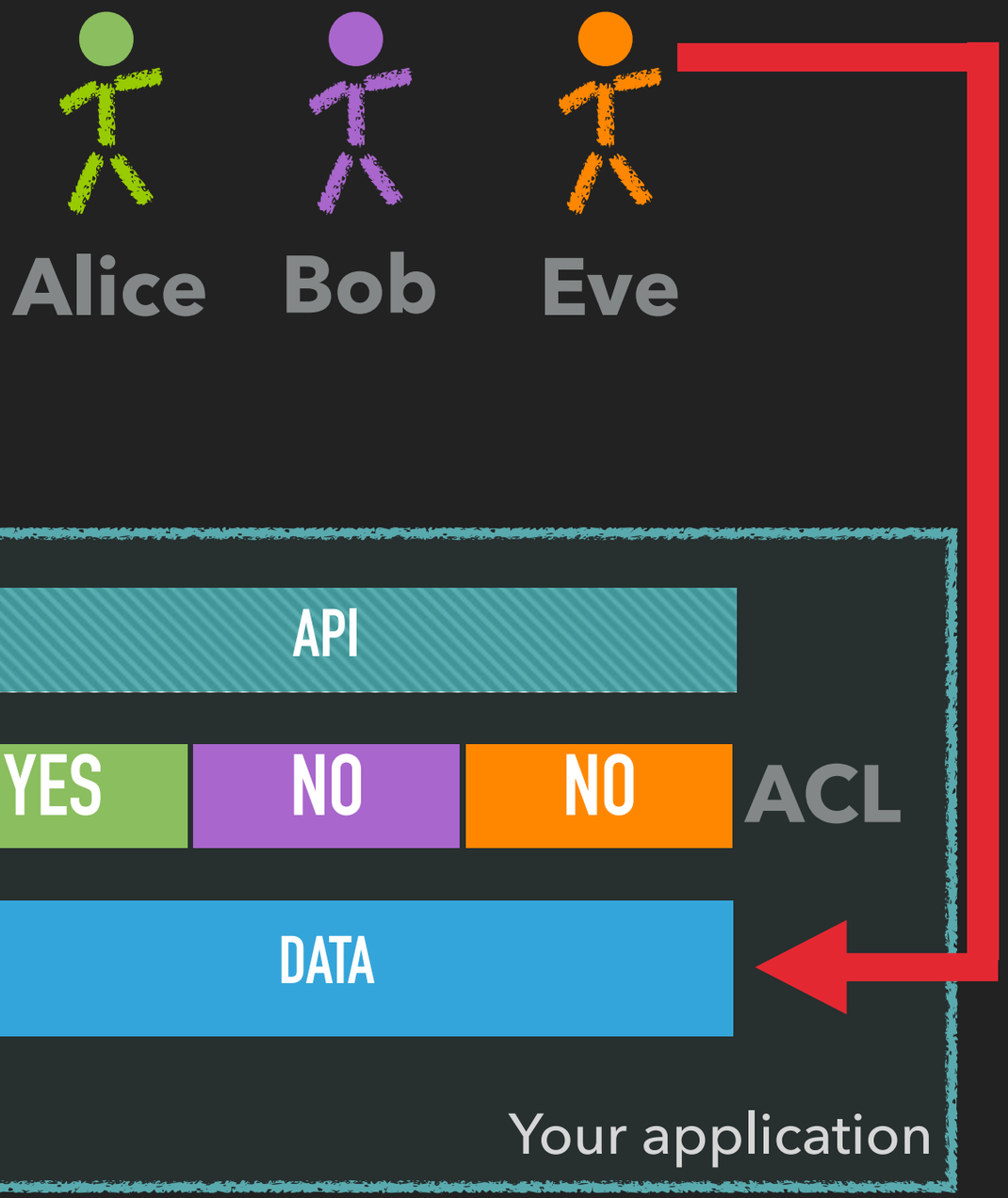




1. Eve exploits application

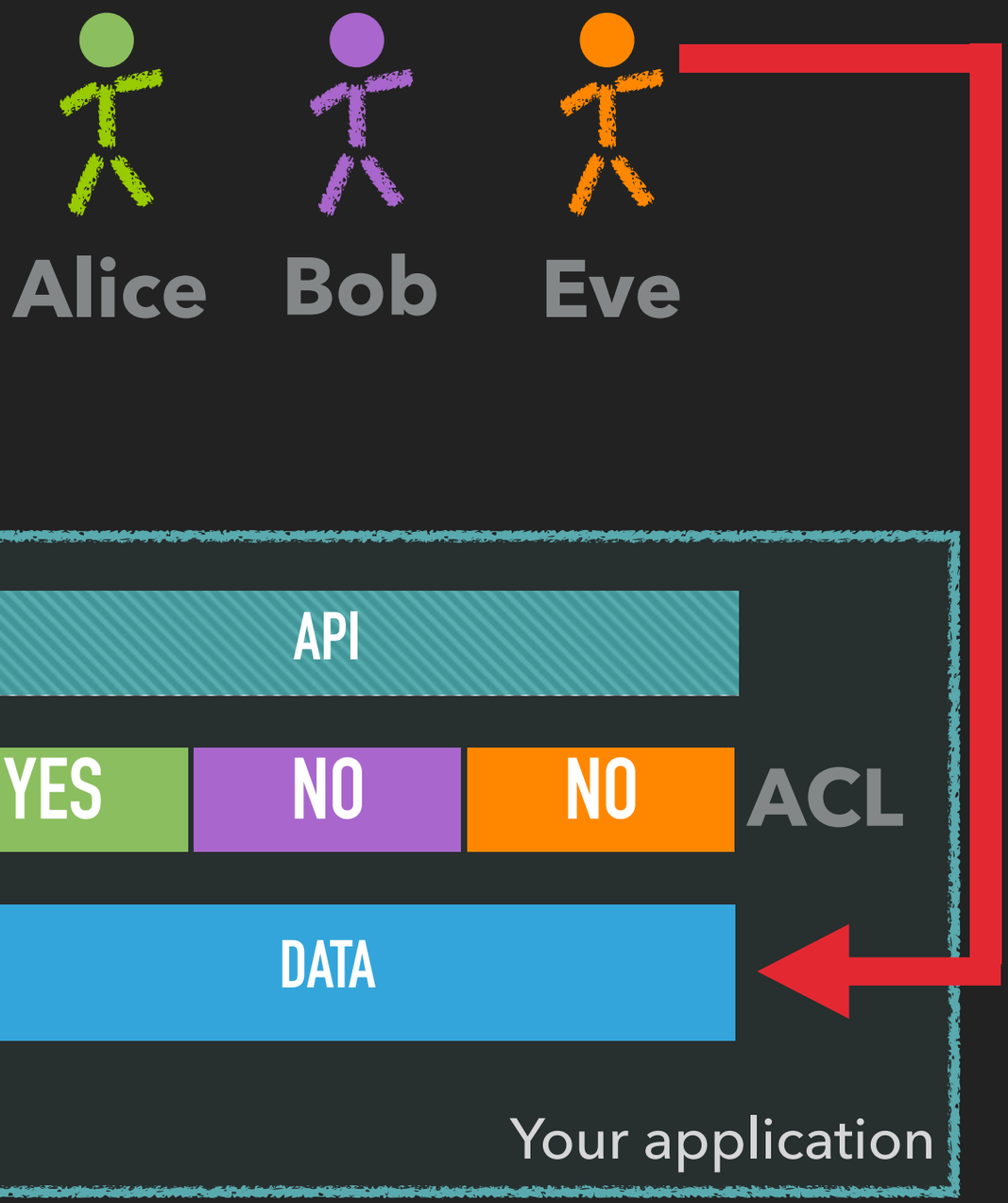
ACCESS CONTROL





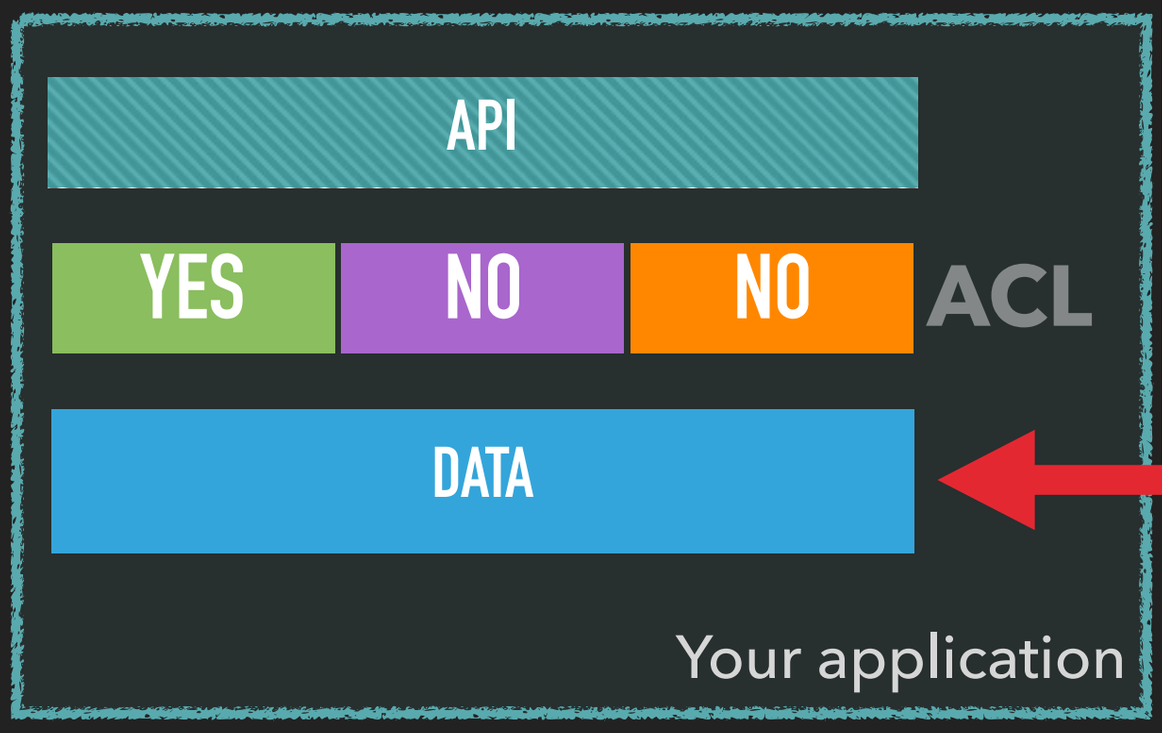
1. Eve exploits application

ACCESS CONTROL

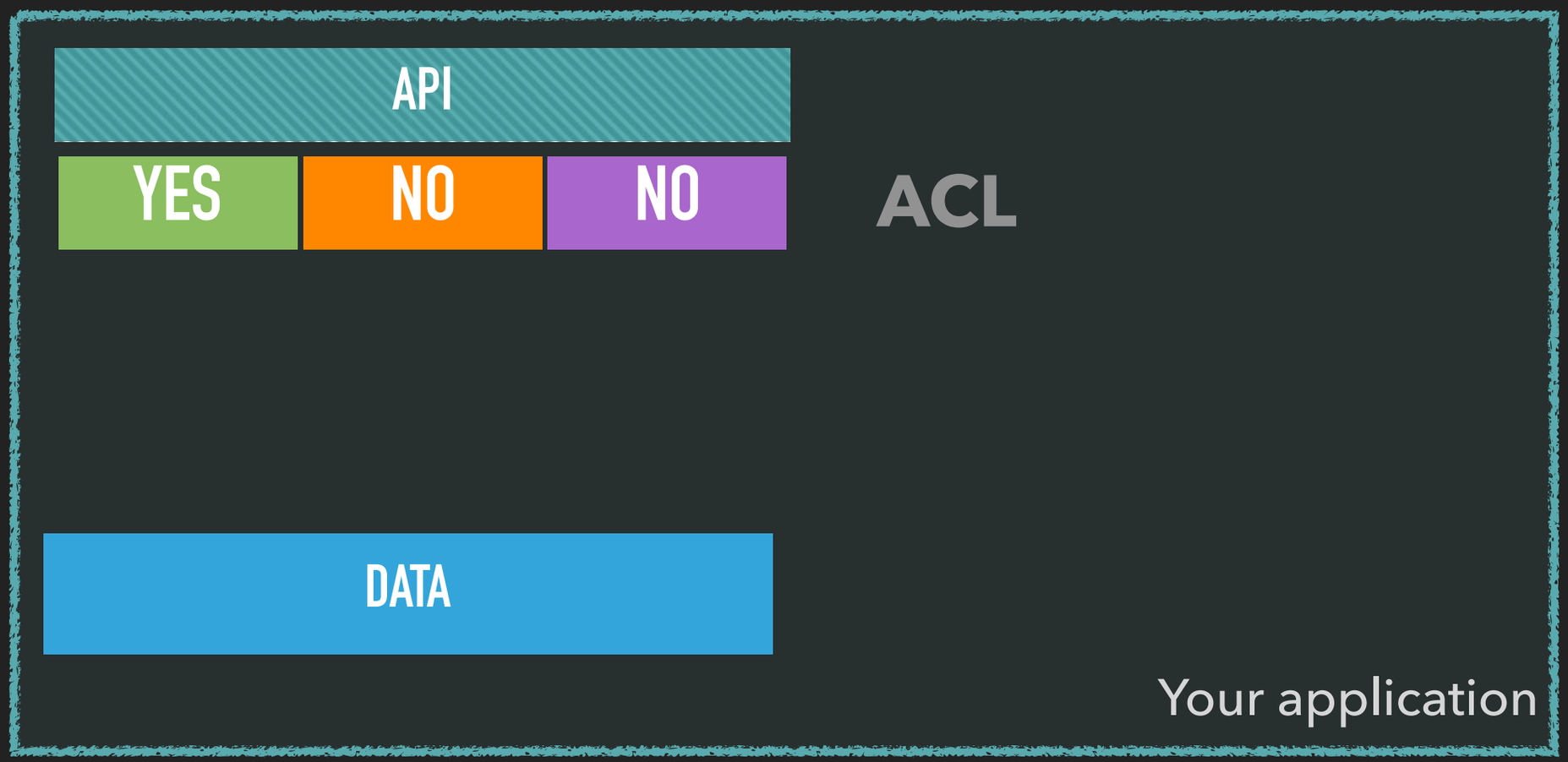


ACCESS CONTROL

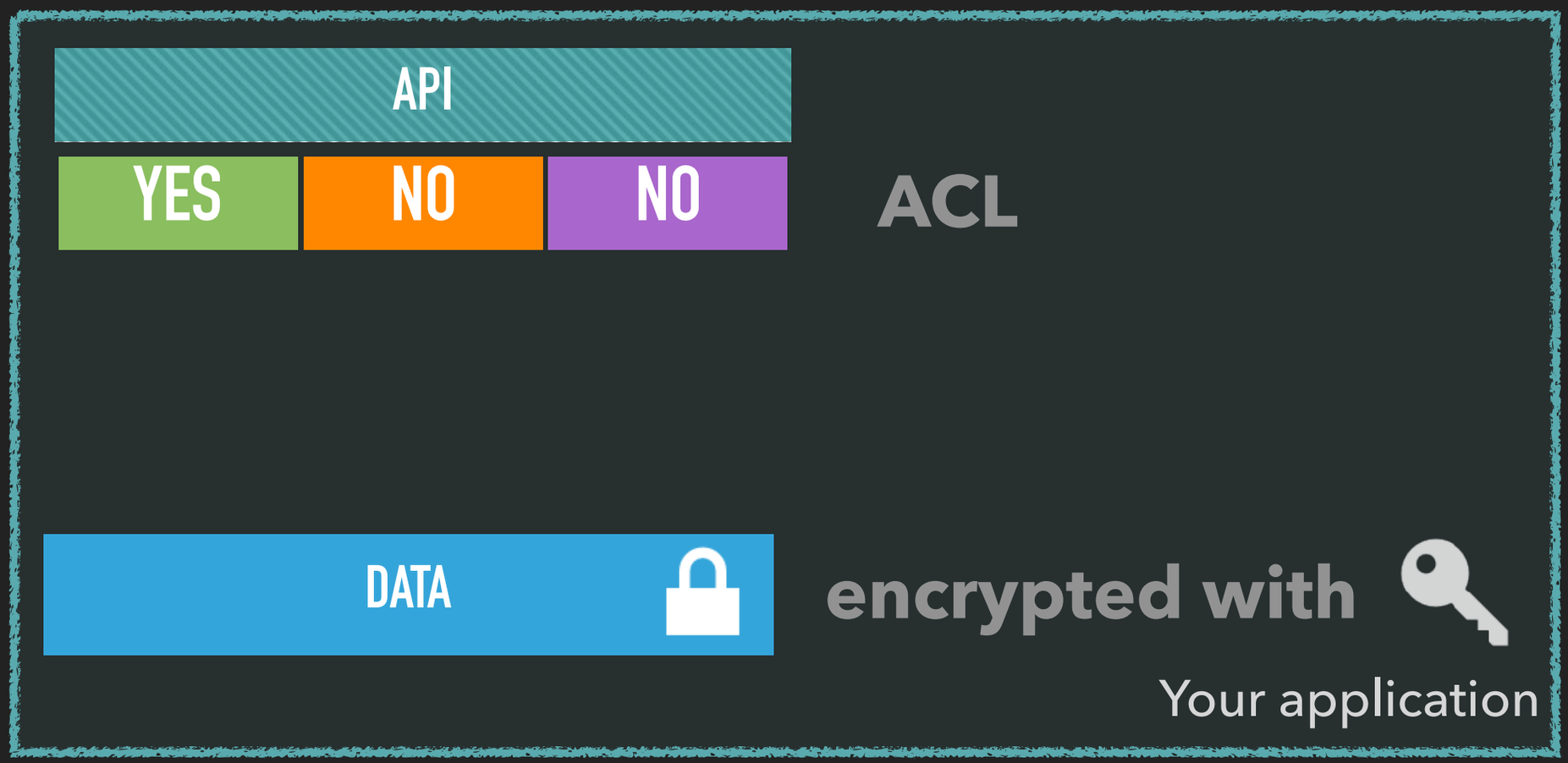
1. Eve exploits application
2. Application ????
3. Eve has access to data



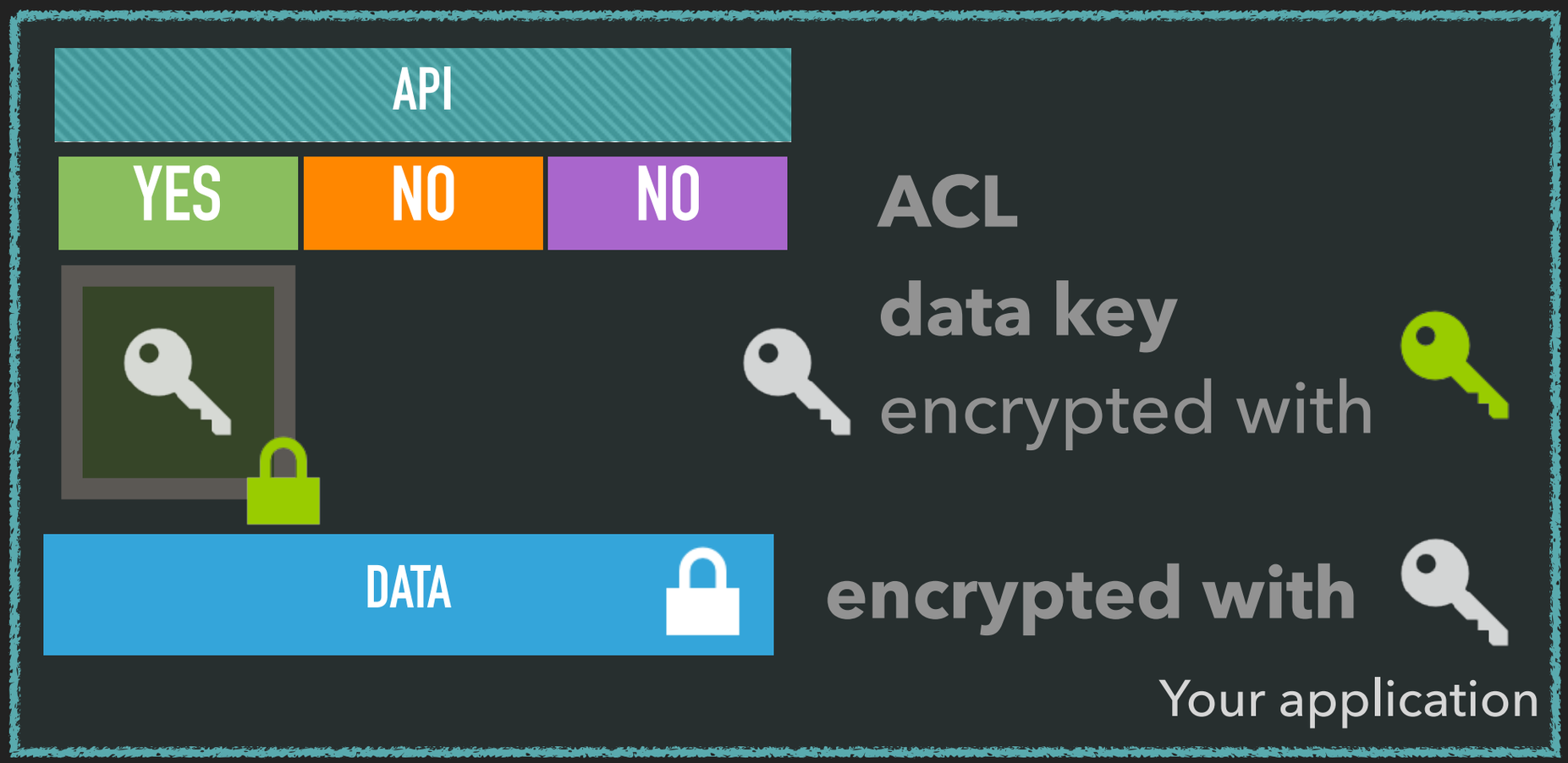
ACCESS CONTROL



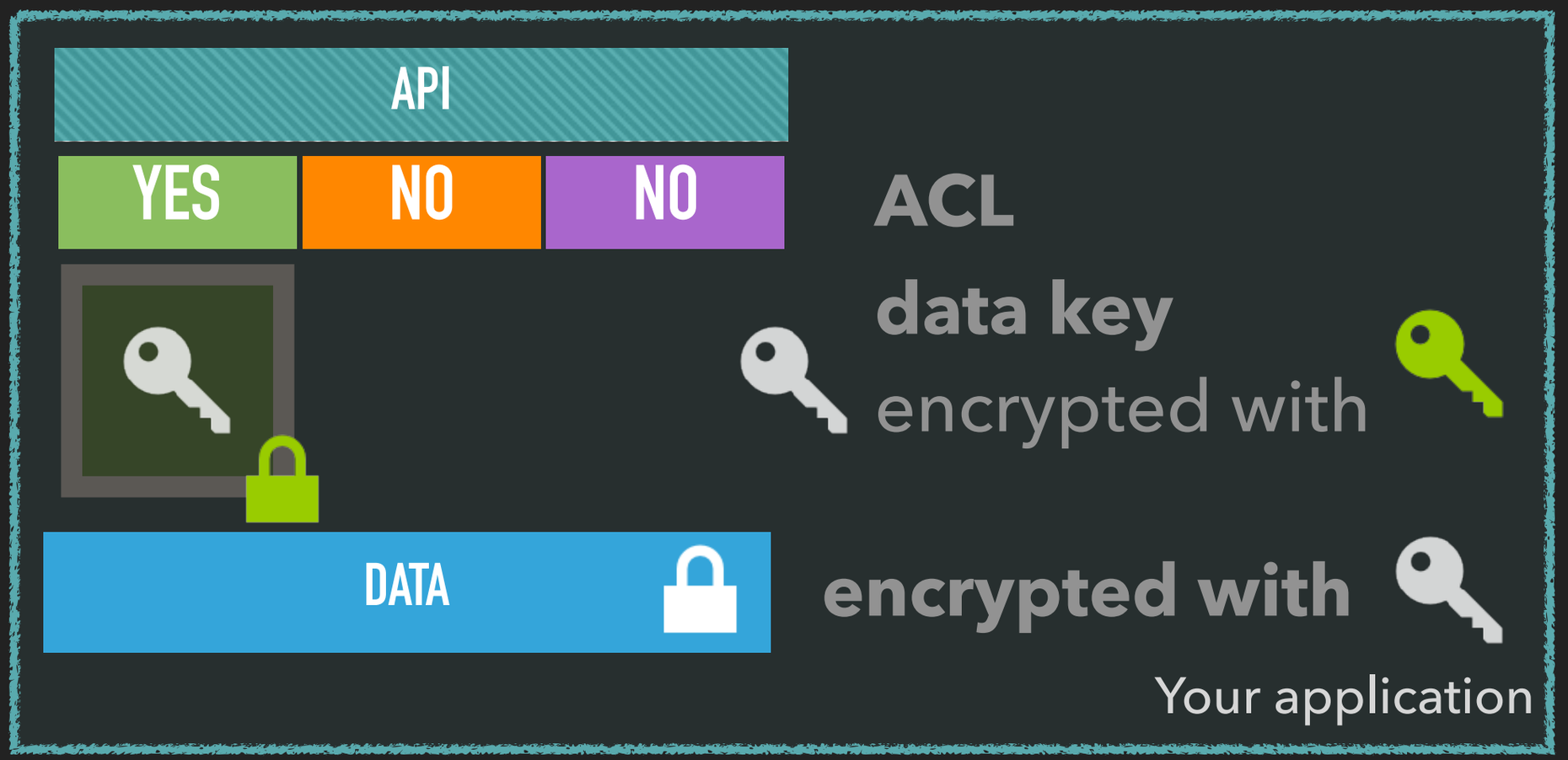
ACCESS CONTROL



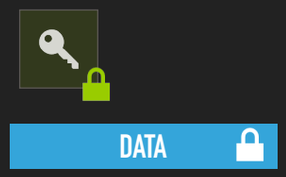
ACCESS CONTROL

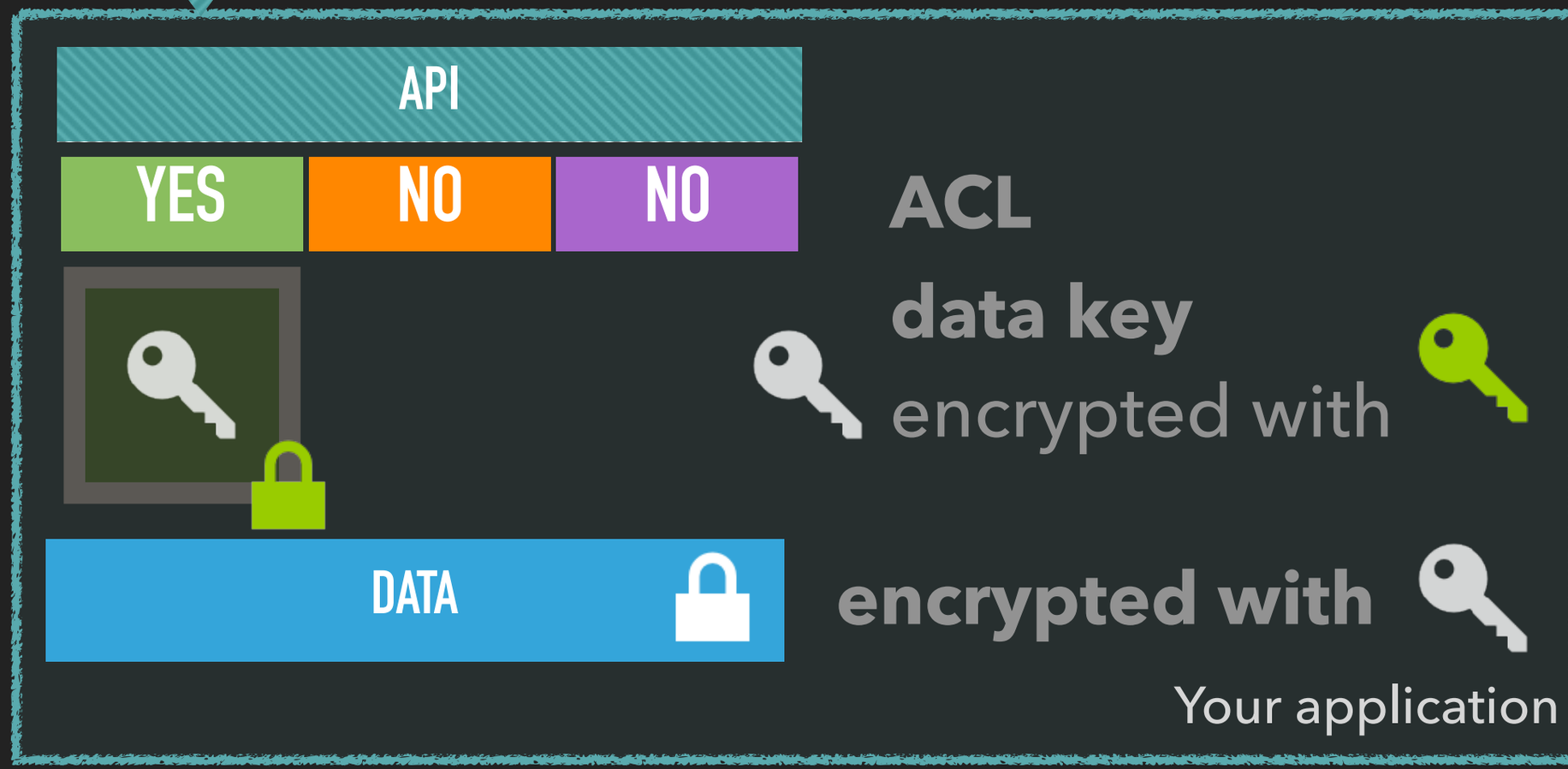


ACCESS CONTROL



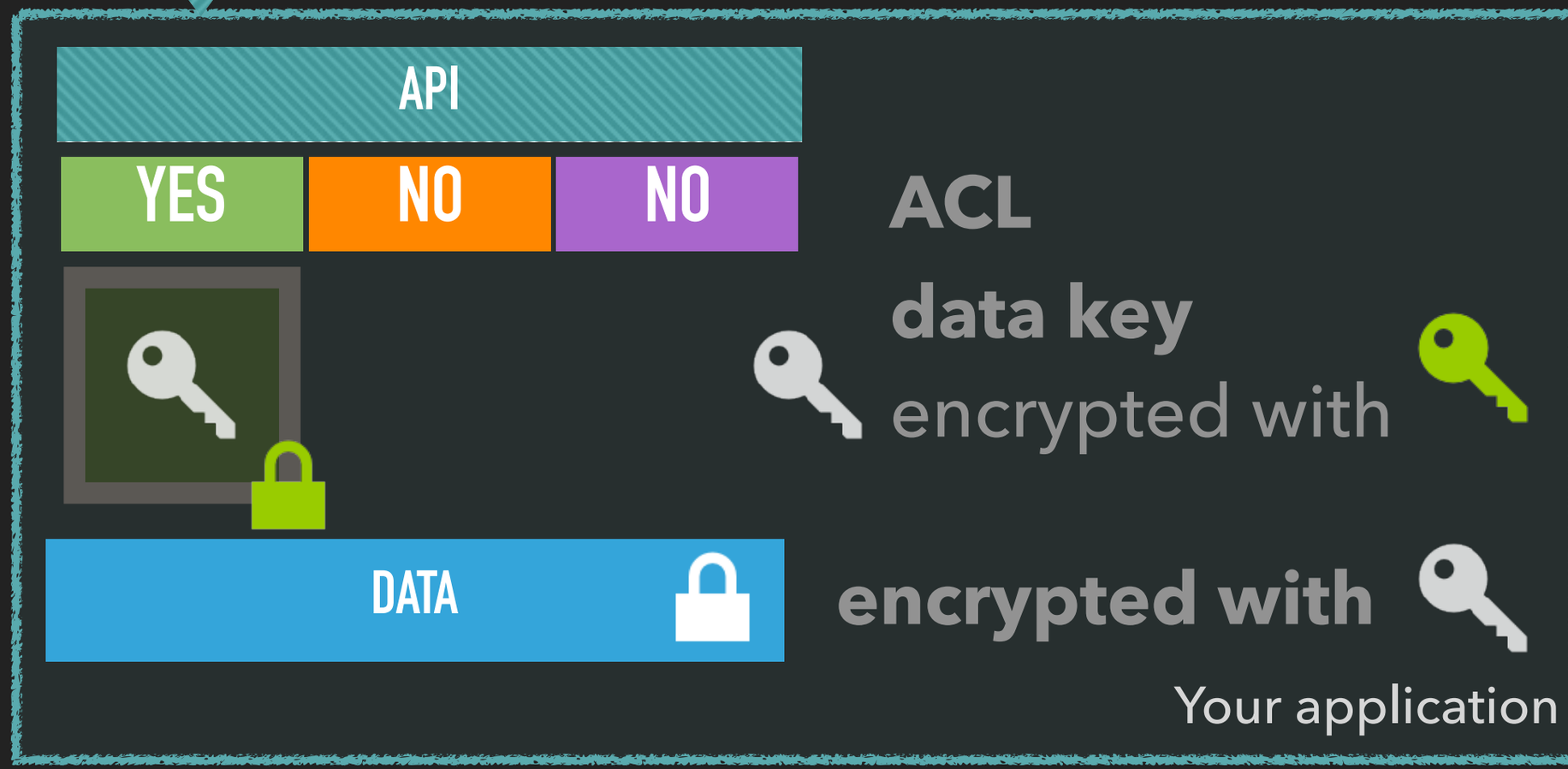
ACCESS CONTROL





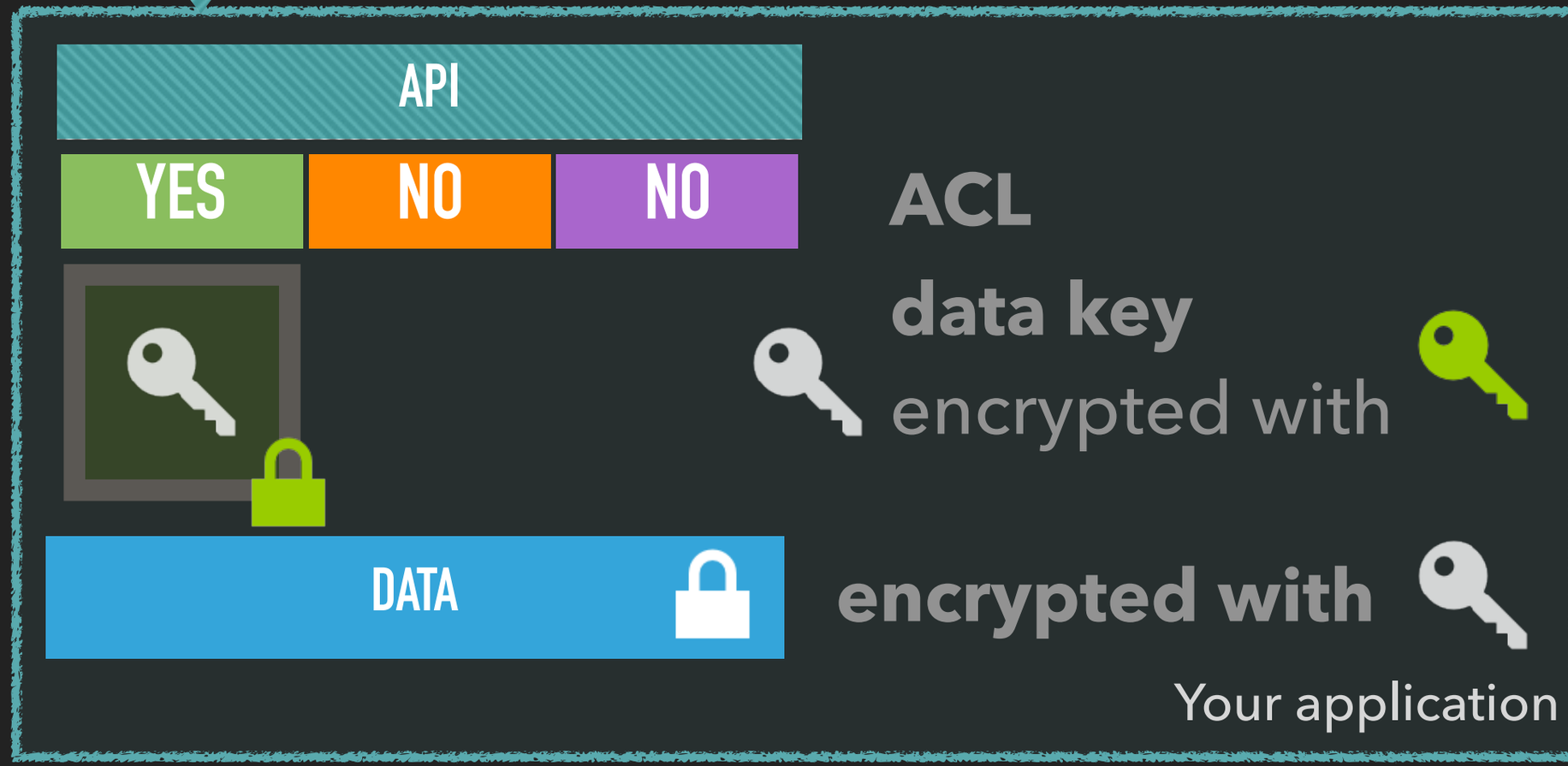
ACCESS CONTROL

1. Alice tries to read data



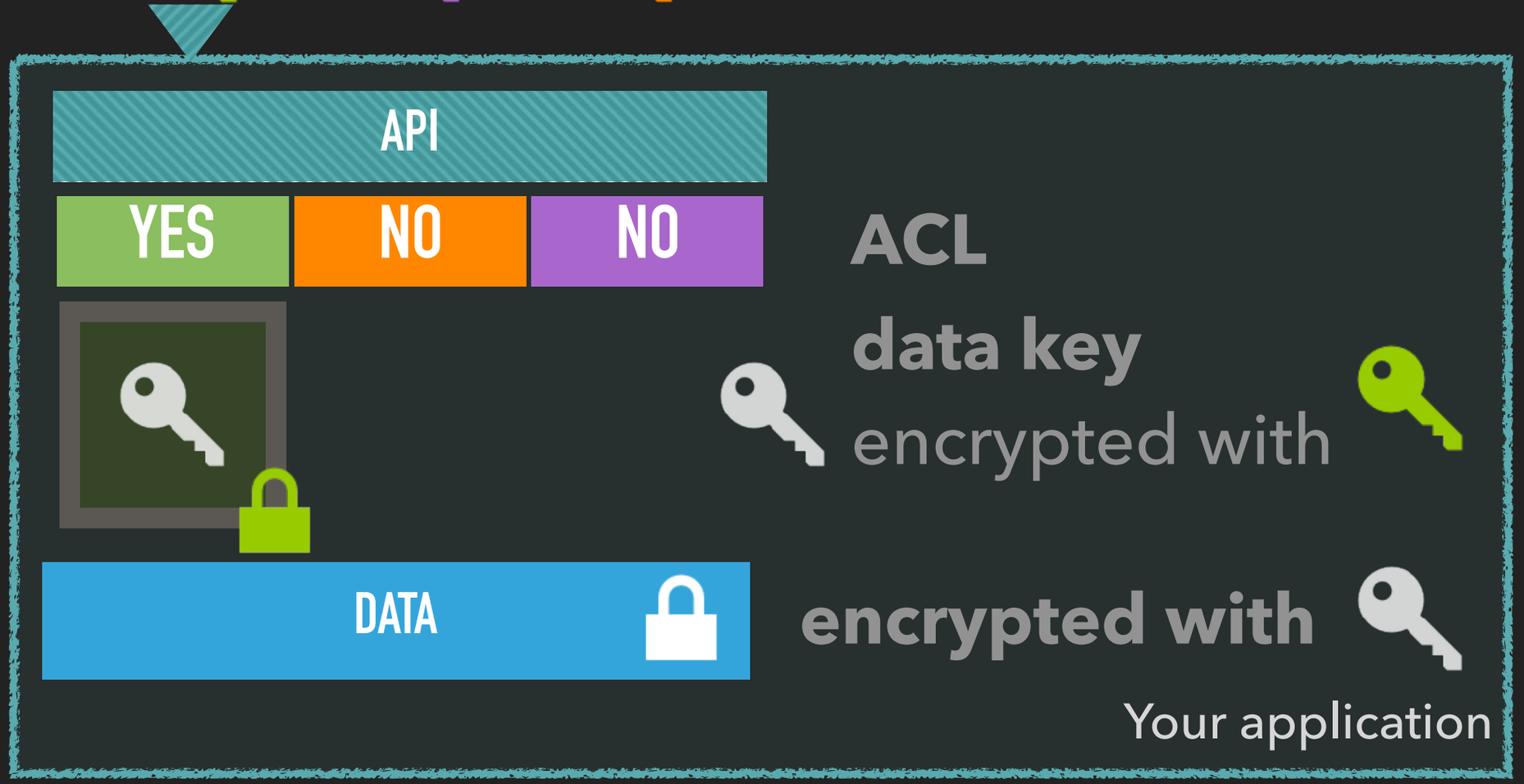
ACCESS CONTROL

1. Alice tries to read data
2. Application validates ACL



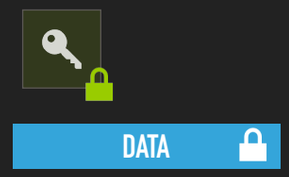
ACCESS CONTROL

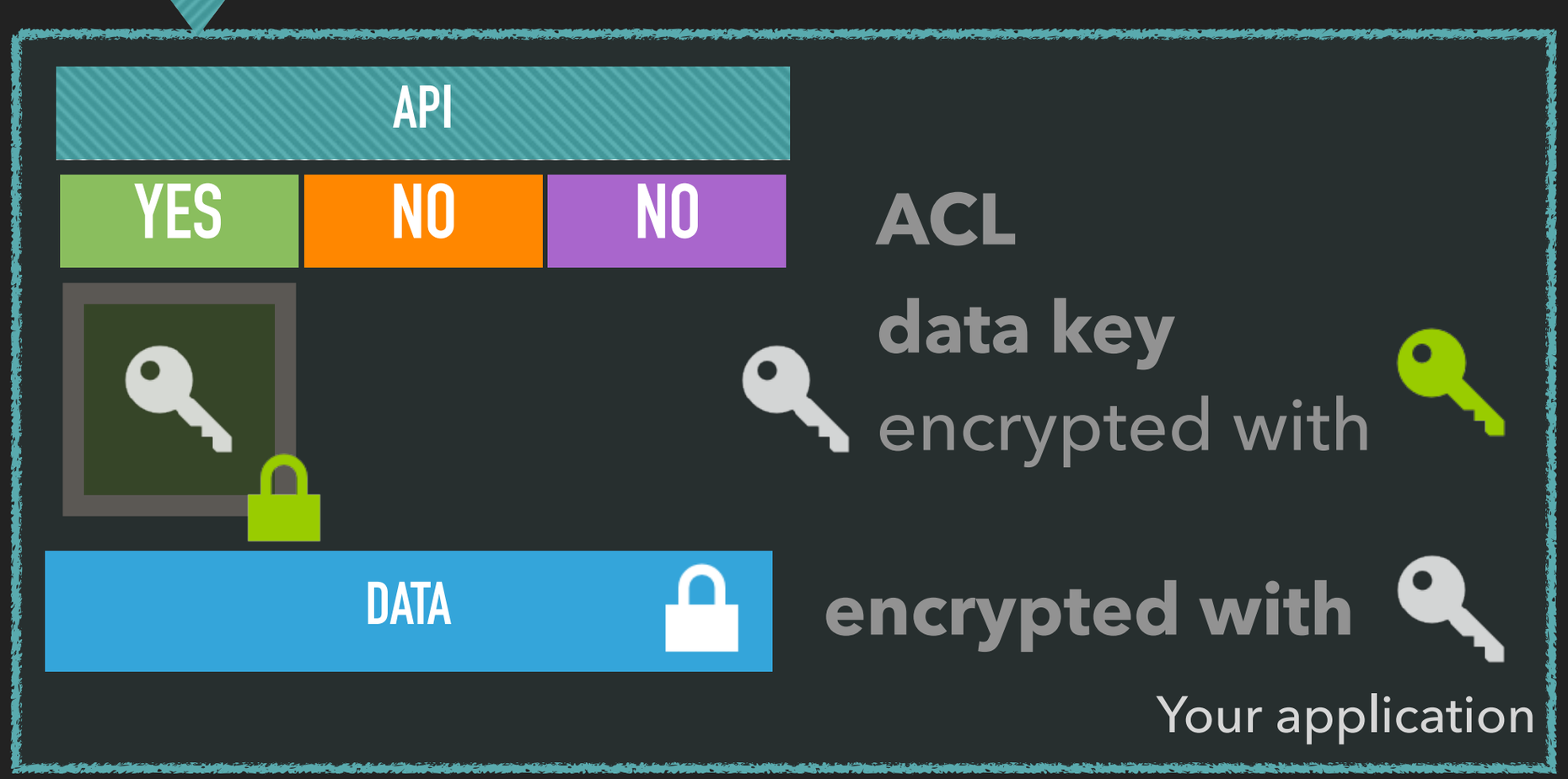
1. Alice tries to read data
2. Application validates ACL
3. Alice is granted access



ACCESS CONTROL

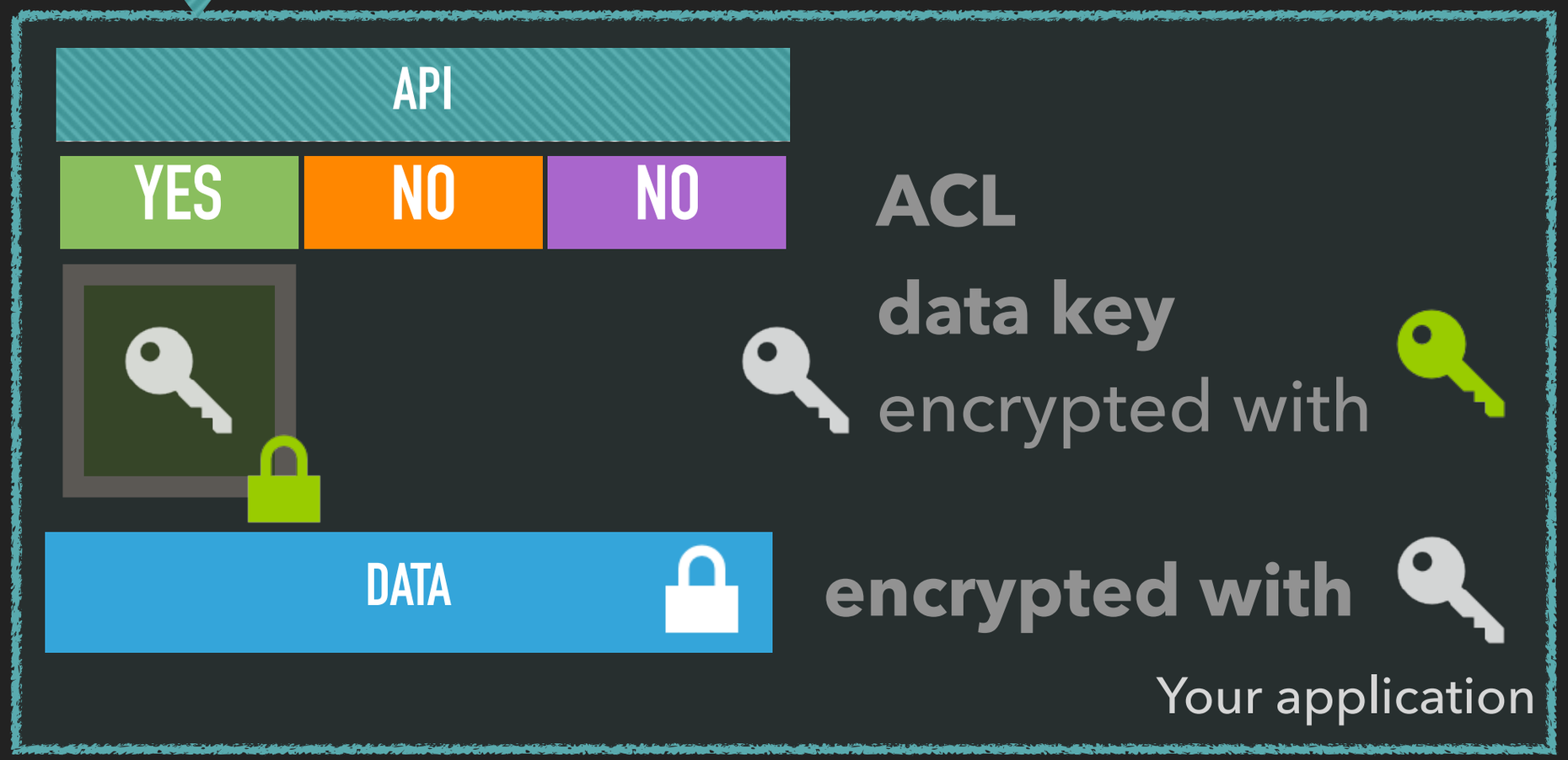
1. Alice tries to read data
2. Application validates ACL
3. Alice is granted access
4. Alice decrypts the “data key”



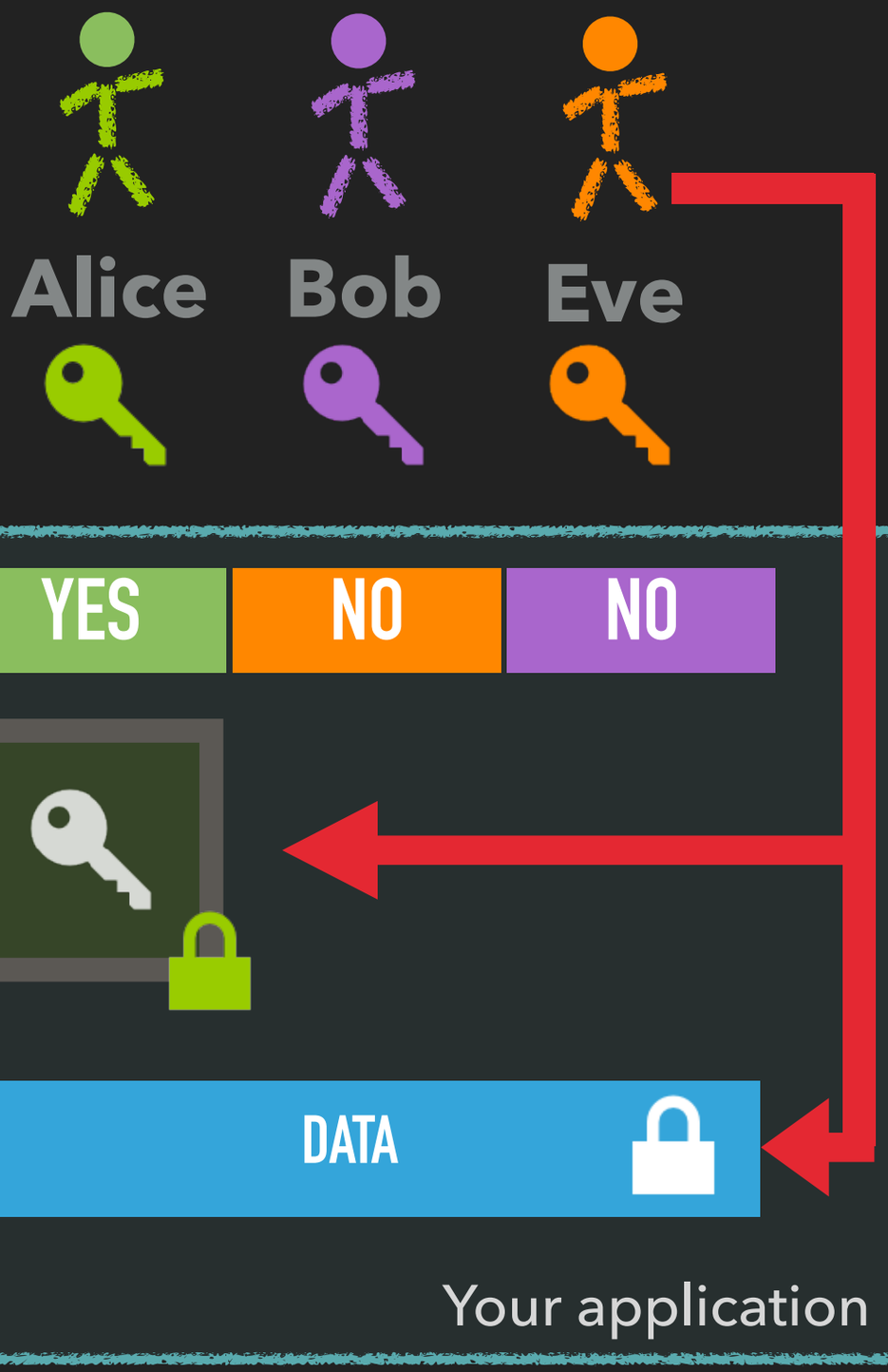


ACCESS CONTROL

1. Alice tries to read data
2. Application validates ACL
3. Alice is granted access
4. Alice decrypts the “data key”
5. Alice decrypts data

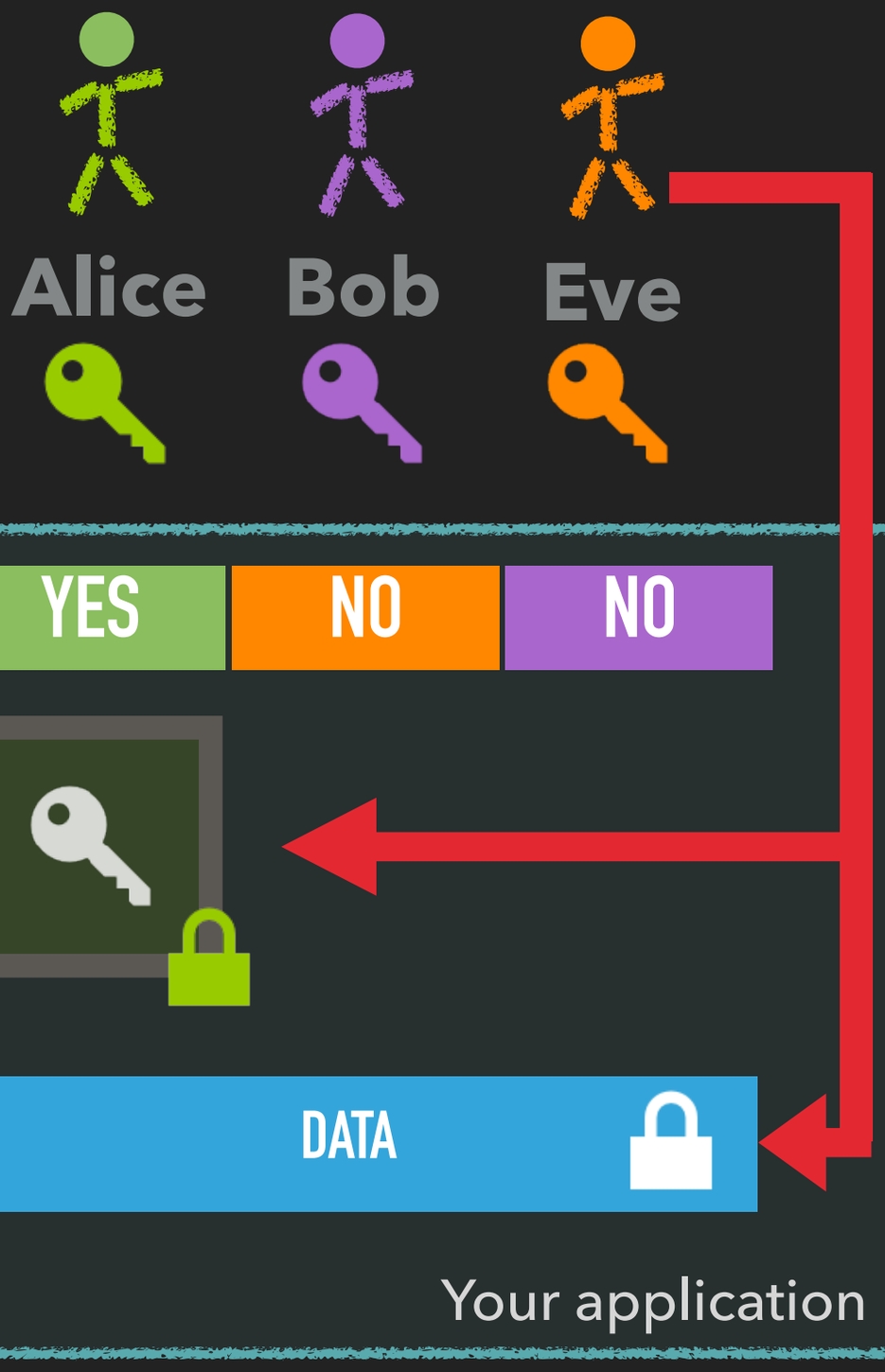


ACCESS CONTROL



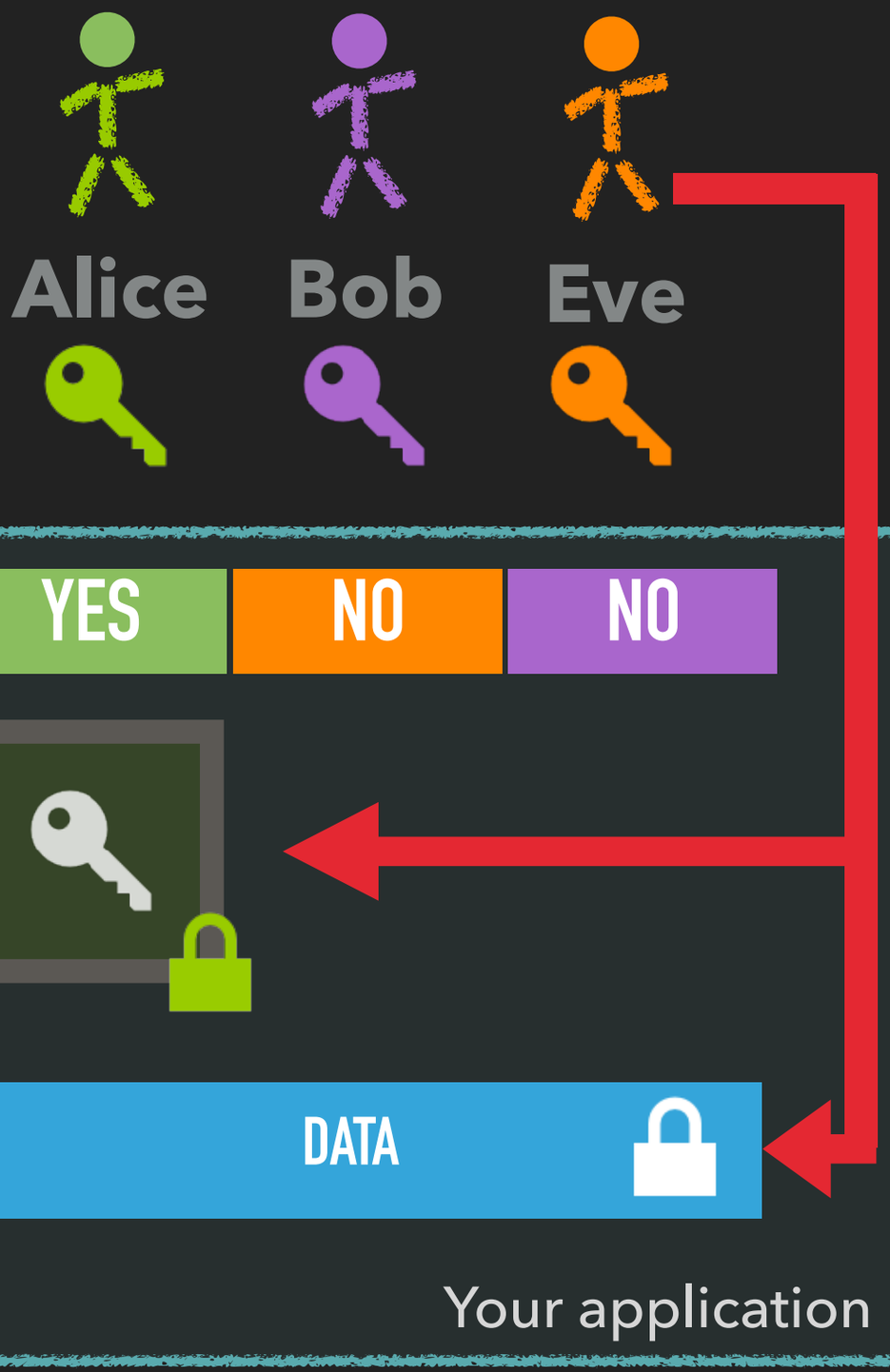
1. Eve exploits application

ACCESS CONTROL



1. Eve exploits application
2. Eve gets encrypted data key
   AND
   Eve gets encrypted data

ACCESS CONTROL



1. Eve exploits application
2. Eve gets encrypted data key
   AND
   Eve gets encrypted data

Eve cannot decrypt the data!

ACCESS CONTROL

1. Alice enters her password

DECOUPLE PASSWORD AND LONGTERM KEY TO ALLOW EASY PASSWORD CHANGES

ACCESS CONTROL

(1)

Alice

*****

1. Alice enters her password

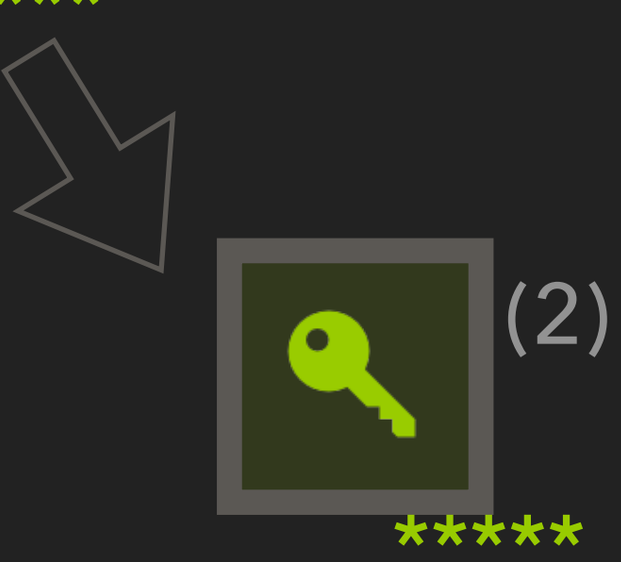
DECOUPLE PASSWORD AND LONGTERM KEY TO ALLOW EASY PASSWORD CHANGES

ACCESS CONTROL

(1)

Alice

*****



1. Alice enters her password
2. Application decrypts her strong long-term key

Alice's long term secret key
encrypted with her password.

DECOUPLE PASSWORD AND LONGTERM KEY TO ALLOW EASY PASSWORD CHANGES

ACCESS CONTROL

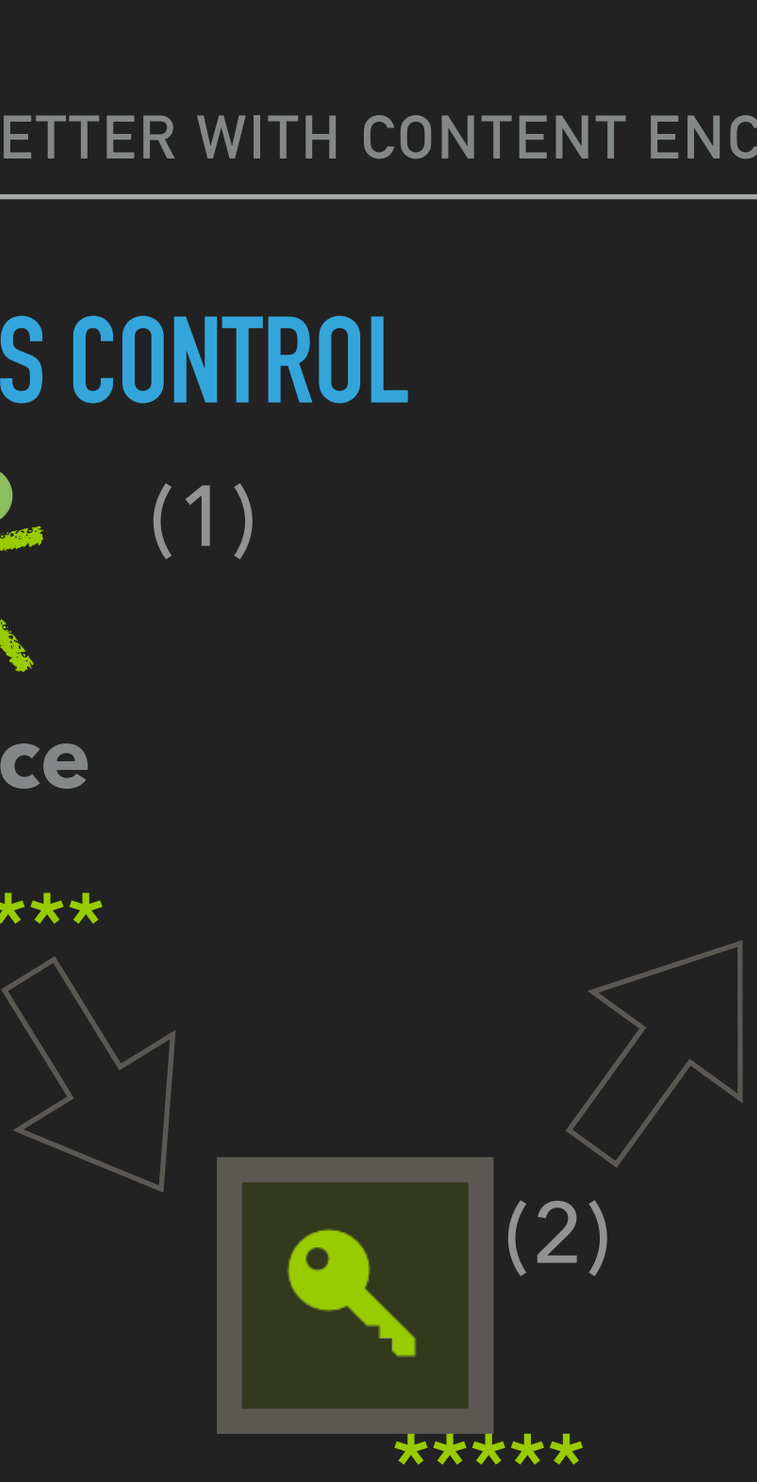
(1)

Alice

*****

(3)

Alice



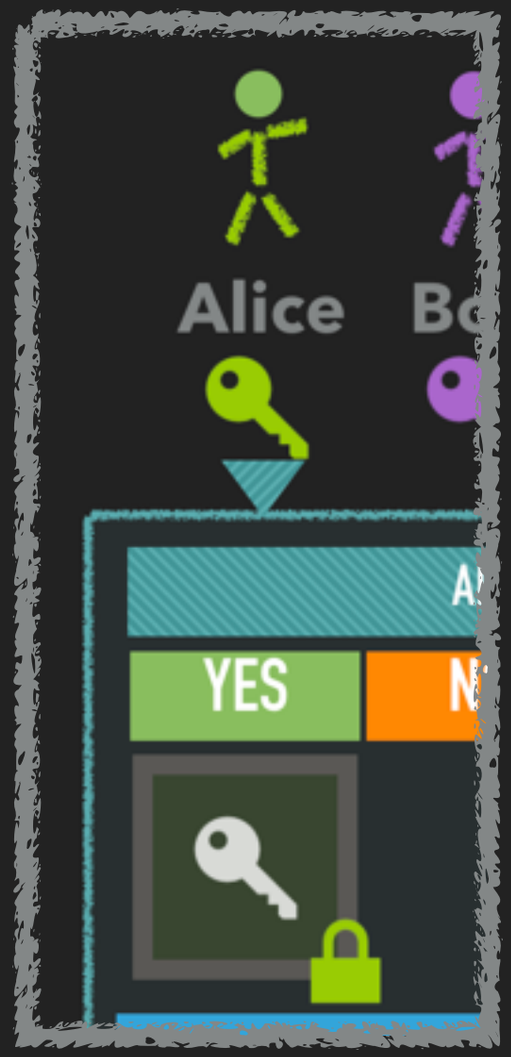
1. Alice enters her password
2. Application decrypts her strong long-term key
3. Alice now has her long-term key

Alice's long term secret key
encrypted with her password.

DECOUPLE PASSWORD AND LONGTERM KEY TO ALLOW EASY PASSWORD CHANGES

ACCESS CONTROL

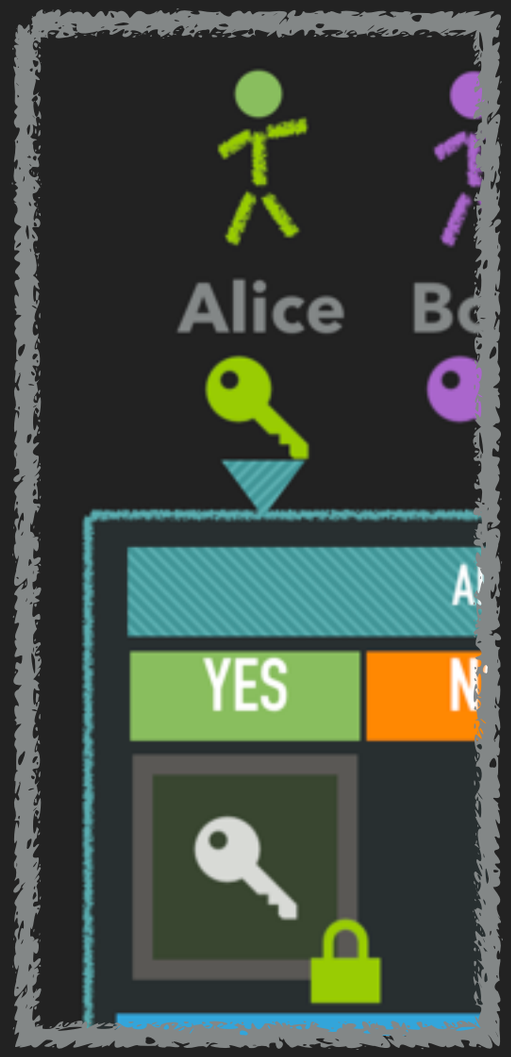
Situation:



ACCESS CONTROL

Situation:

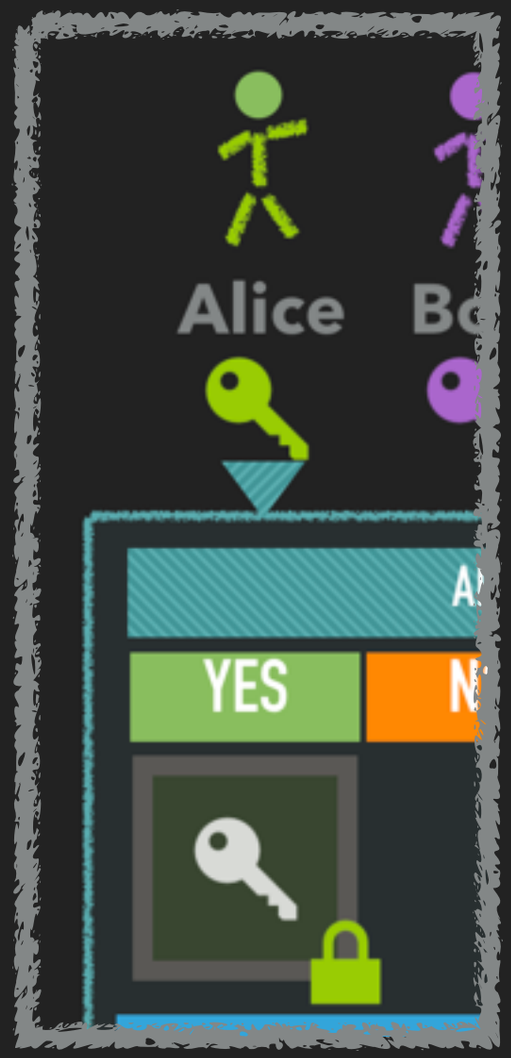
- ✓ All data is secured under Alices key



ACCESS CONTROL

Situation:

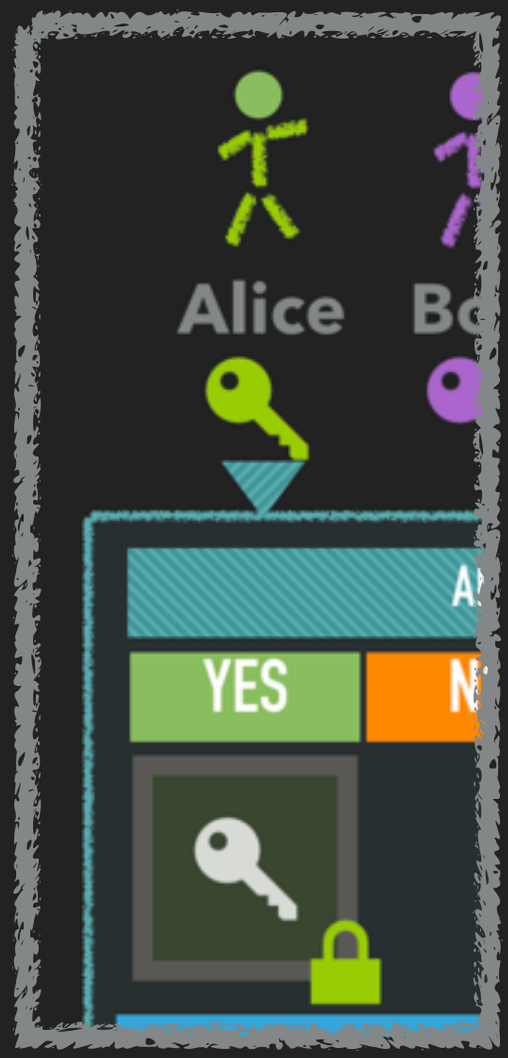
- ✓ All data is secured under Alices key
- ✓ Alice can change her password by reencrypting her longterm key



ACCESS CONTROL

Situation:

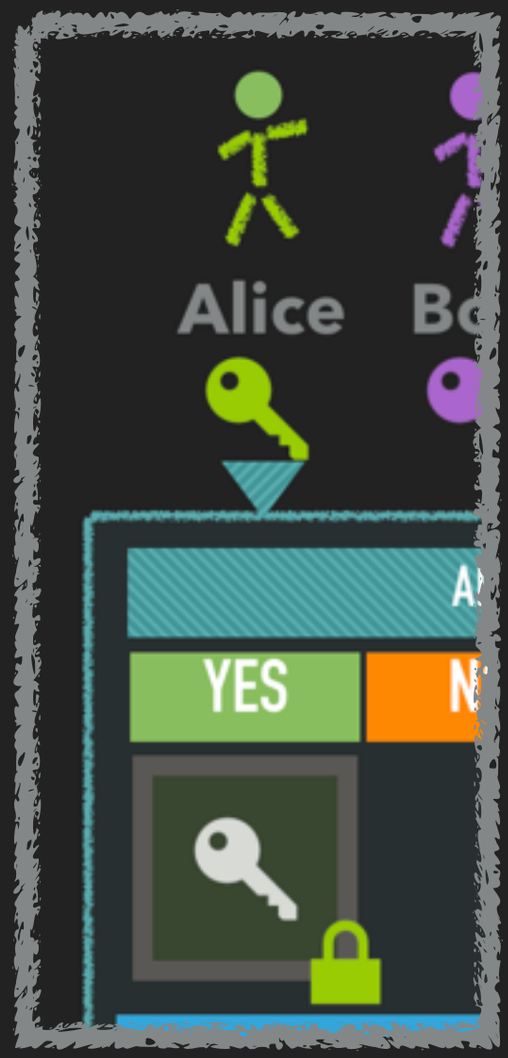
- ✓ All data is secured under Alices key
- ✓ Alice can change her password by reencrypting her longterm key
- ⚠ Alice may forget her password



ACCESS CONTROL

Situation:

- ✓ All data is secured under Alices key
- ✓ Alice can change her password by reencrypting her longterm key
- ⚠ Alice may forget her password



Question:

What happens when Alice forgets her password?

ACCESS CONTROL

Problem: What happens when Alice forgets her password?

Solution: Use cryptographic secret sharing for recovery

ACCESS CONTROL

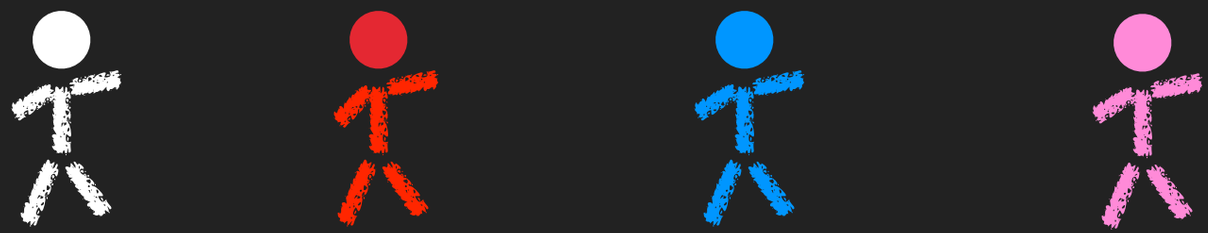
Problem: What happens when Alice forgets her password?

Solution: Use cryptographic secret sharing for recovery

ACCESS CONTROL

Problem: What happens when Alice forgets her password?

Solution: Use cryptographic secret sharing for recovery



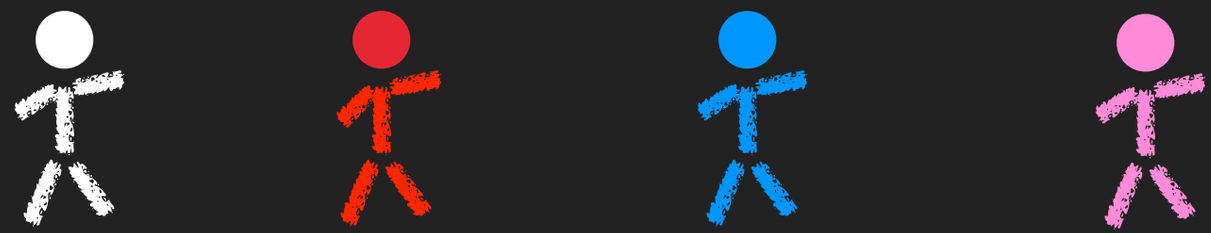
Justus Peter Bob Mathilda

Alice

ACCESS CONTROL

Problem: What happens when Alice forgets her password?

Solution: Use cryptographic secret sharing for recovery



Justus **Peter** **Bob** **Mathilda**

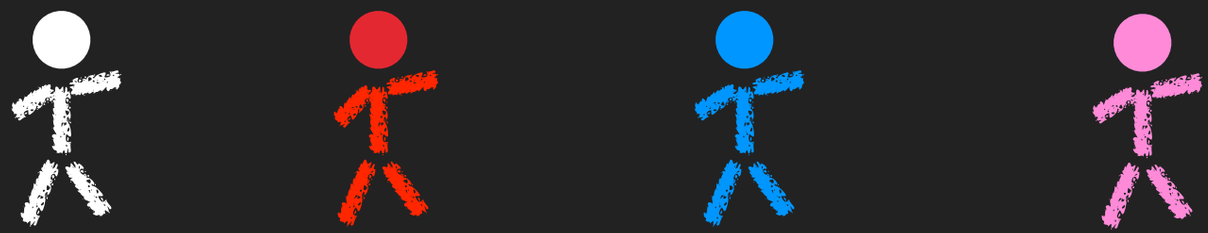
Alice

Alice trusts her friends only so far. But she thinks it is very unlikely that three of them conspire together against her.

ACCESS CONTROL

Problem: What happens when Alice forgets her password?

Solution: Use cryptographic secret sharing for recovery



Justus **Peter** **Bob** **Mathilda**

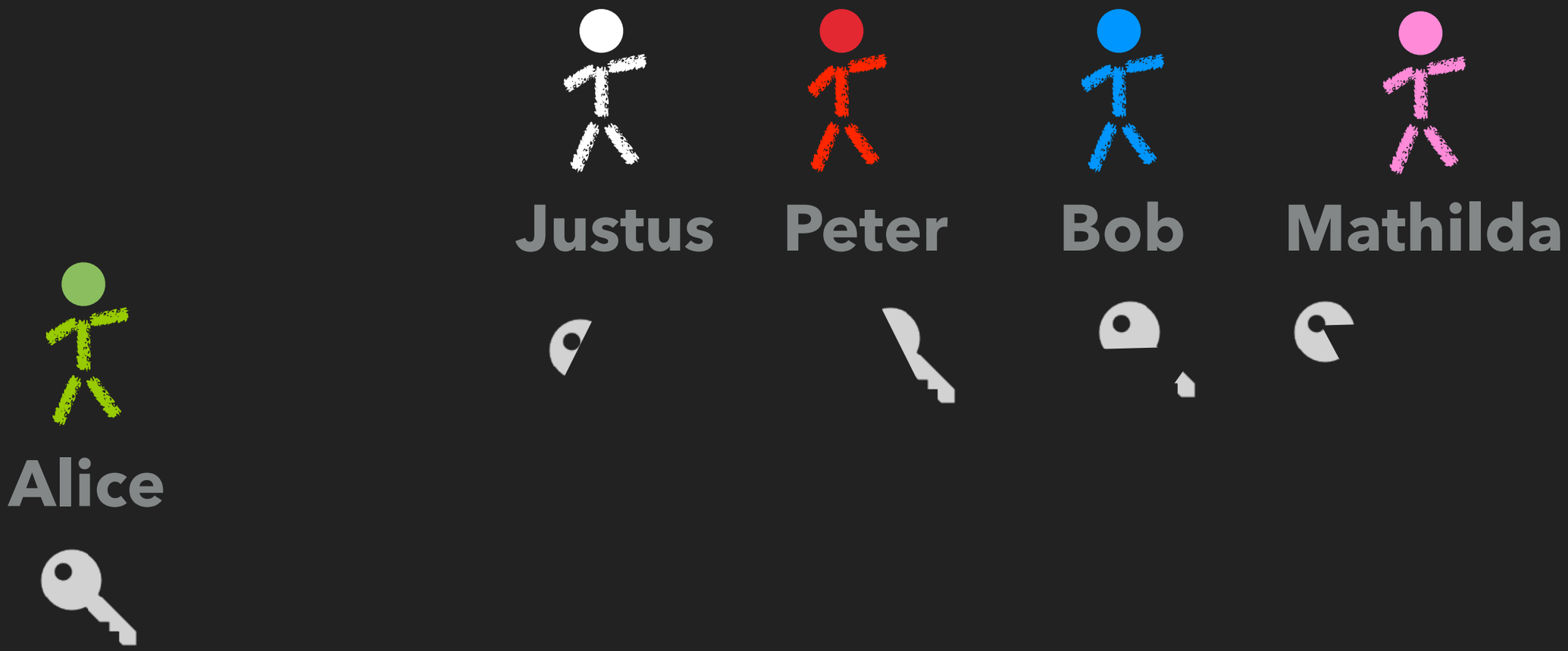
Alice

Alice will split her secret key (e.g. with Shamir) in such a way, that any three of her four trusted friends can restore the key.

ACCESS CONTROL

Problem: What happens when Alice forgets her password?

Solution: Use cryptographic secret sharing for recovery



Alice will split her secret key (e.g. with Shamir) in such a way, that any three of her four trusted friends can restore the key.

DO NOT TRY THIS WITHOUT A CRYPTOGRAPHER

ACCESS CONTROL

Secret sharing (“t out of n”) shares a secret such, that the secret can be restored with any t (here 3) of the n (here 4) parts.

This can be used for secret recovery without a single point of trust (failure).

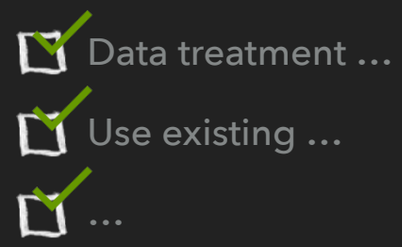
DO NOT TRY THIS WITHOUT A CRYPTOGRAPHER

- [x] Data treatment …
- [x] Use existing …
- [x] …

PATTERNS

CRYPTO CHECKLIST

CRYPTO CHECKLIST



- [ ] Data treatment plan created and consequences accepted by management
- [ ] Trust anchors identified and named
- [ ] Sensitive operations (crypt,sign,..) require client authentication (applies to services too!)
- [ ] Existing (e.g. RFC 4880) protocols & formats used wherever possible
- [ ] Nonces used only once. Random salt used where possible
- [ ] Cryptographic concept written down & challenged in review
- [ ] (Master-)Key offsite backup established
- [ ] Key refresh after a few GiB of encrypted data implemented and tested
- [ ] Algorithm rollover implemented and tested
- [ ] Entropy source with enough entropy used
- [ ] Test cases include restore of old data (key/algorithm rollover)

SUMMARY

Regulations apply - whatever you do!

Encryption is not for free!

No encryption might be way more expensive!

Encryption is a safety net (*last* line of defence)

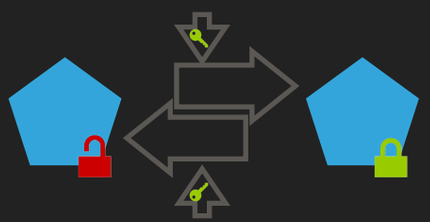
–> Assess risks & cost, plan, implement!

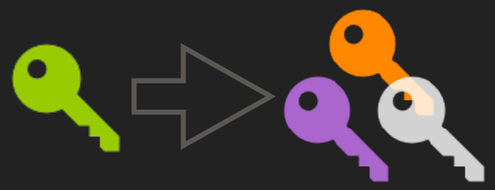
SUMMARY

Comparing data

Transparent encryption

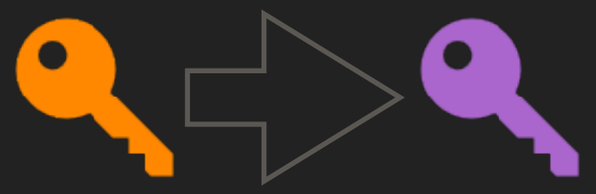
Storing data

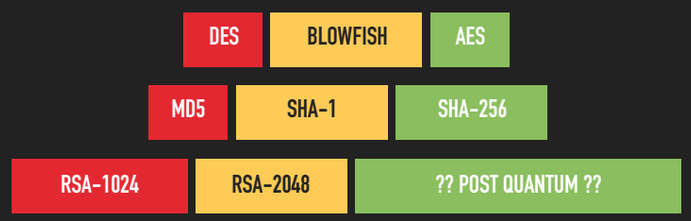




Key derivation

Key refresh





Algorithm rollover

SUMMARY

0X123456…

Integrity

Moving Data

```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

Entropy

Access Control

- Data treatment …
- Use existing …
- …

Crypto Checklist

Q & A

https://github.com/neuhalje/presentation_content-encryption

Something missing?

Boring?

Awesome?

Feedback helps!

FEEDBACK

https://github.com/neuhalje/presentation_content-encryption

BACKUP

~BACKUP