



in
40 minutes

SLEEP BETTER WITH

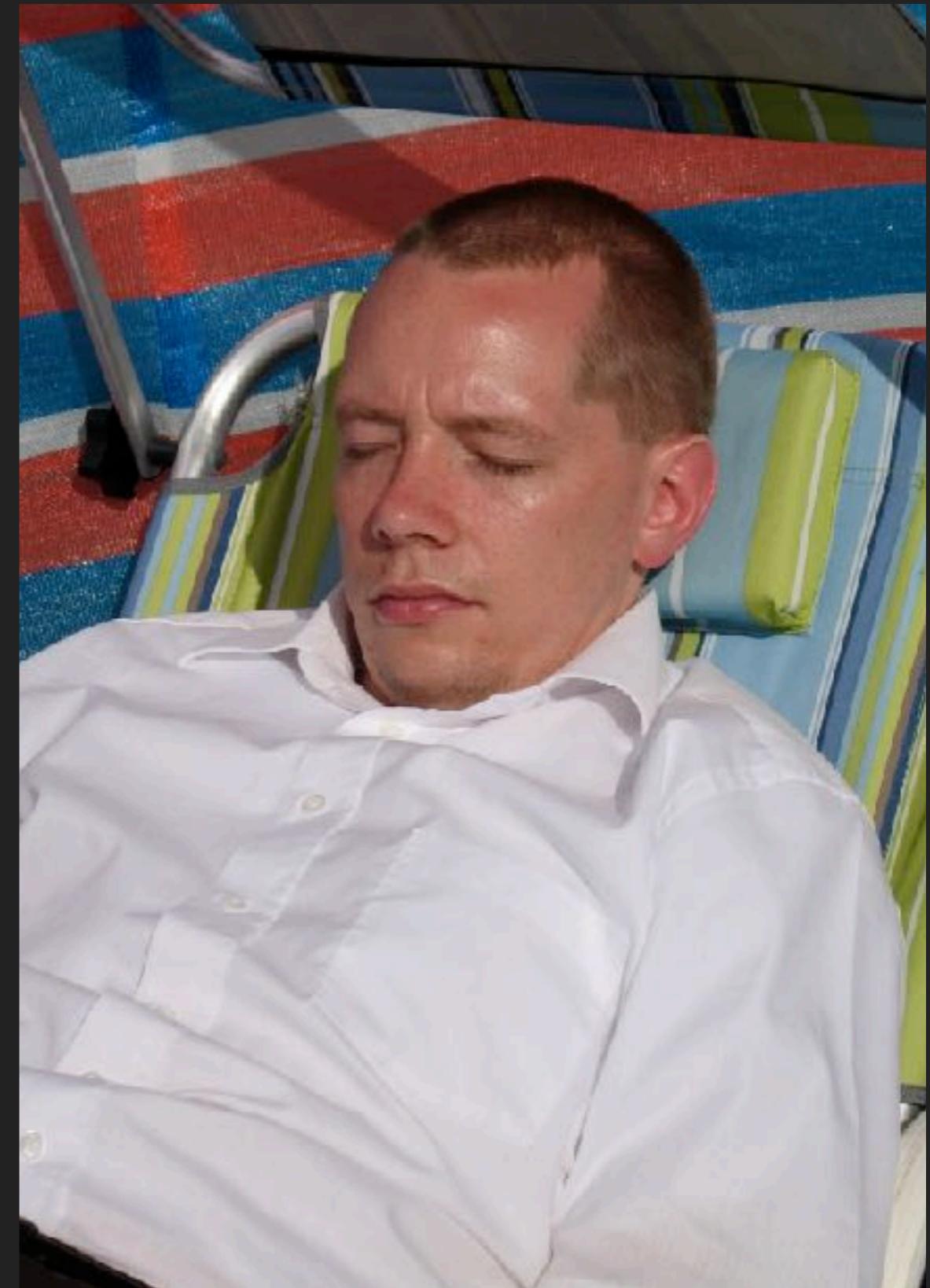
CONTENT ENCRYPTION



CC BY-SA 4.0

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

Process

Store but not use

YOUR DATA



Collect

Store *but not use*

Delete

Process

YOUR DATA



YOUR DATA



YOUR DATA

● Business value

● Liability



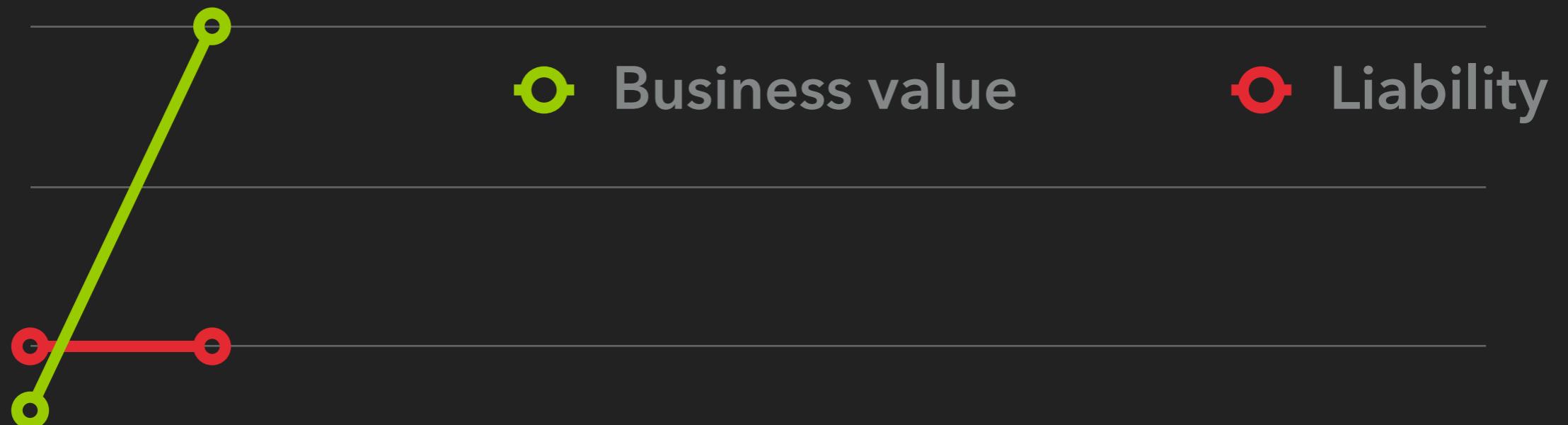
YOUR DATA

● Business value

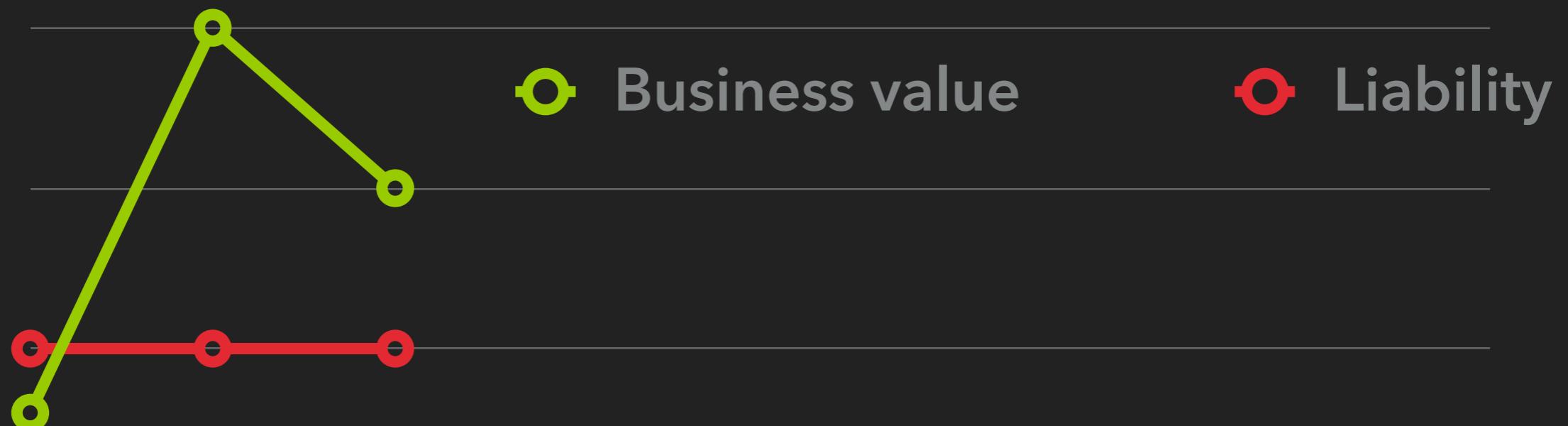
● Liability



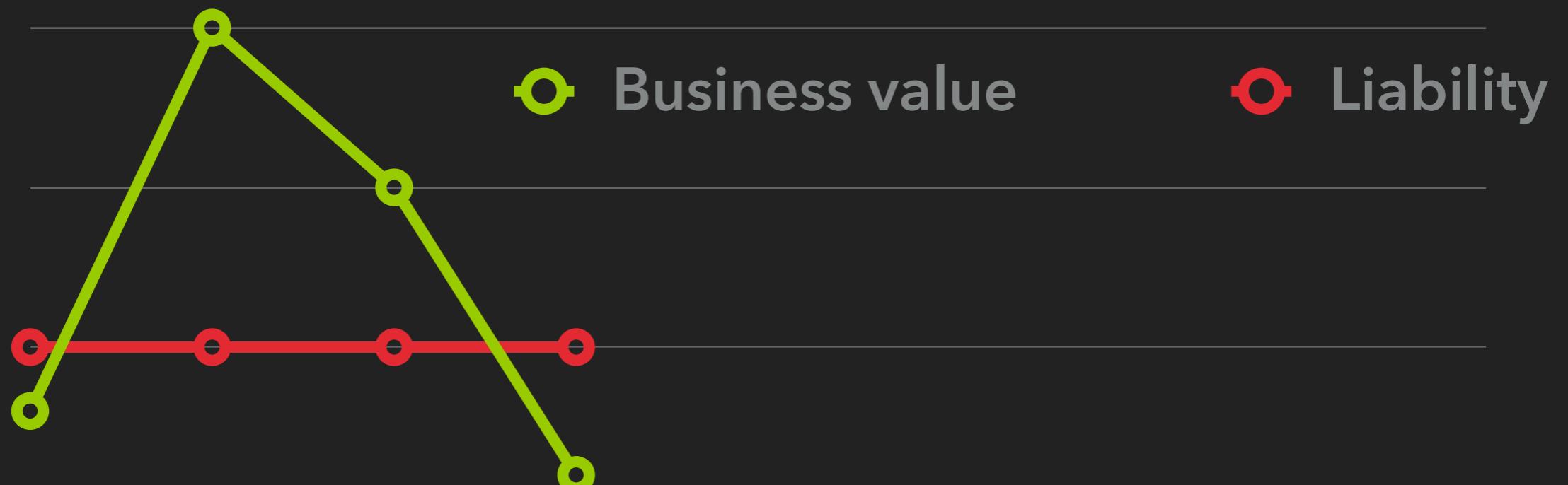
YOUR DATA



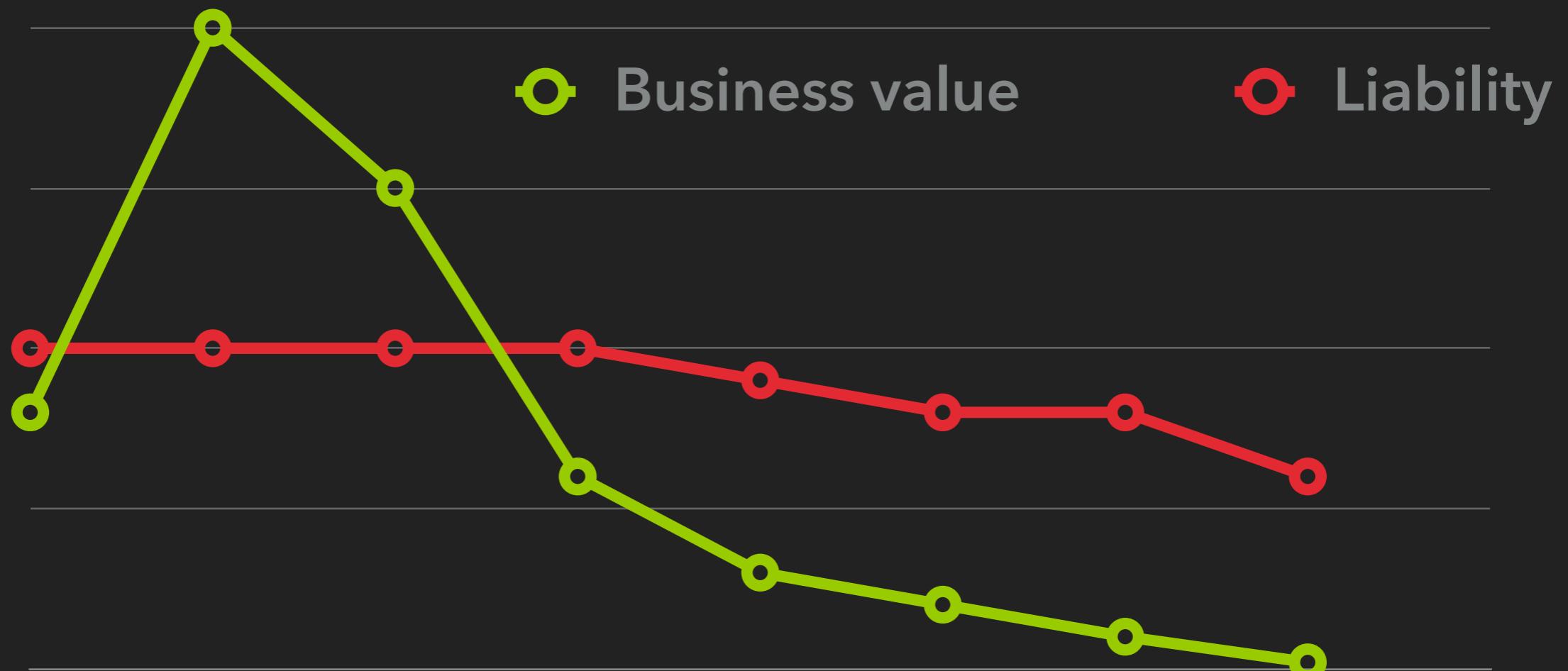
YOUR DATA



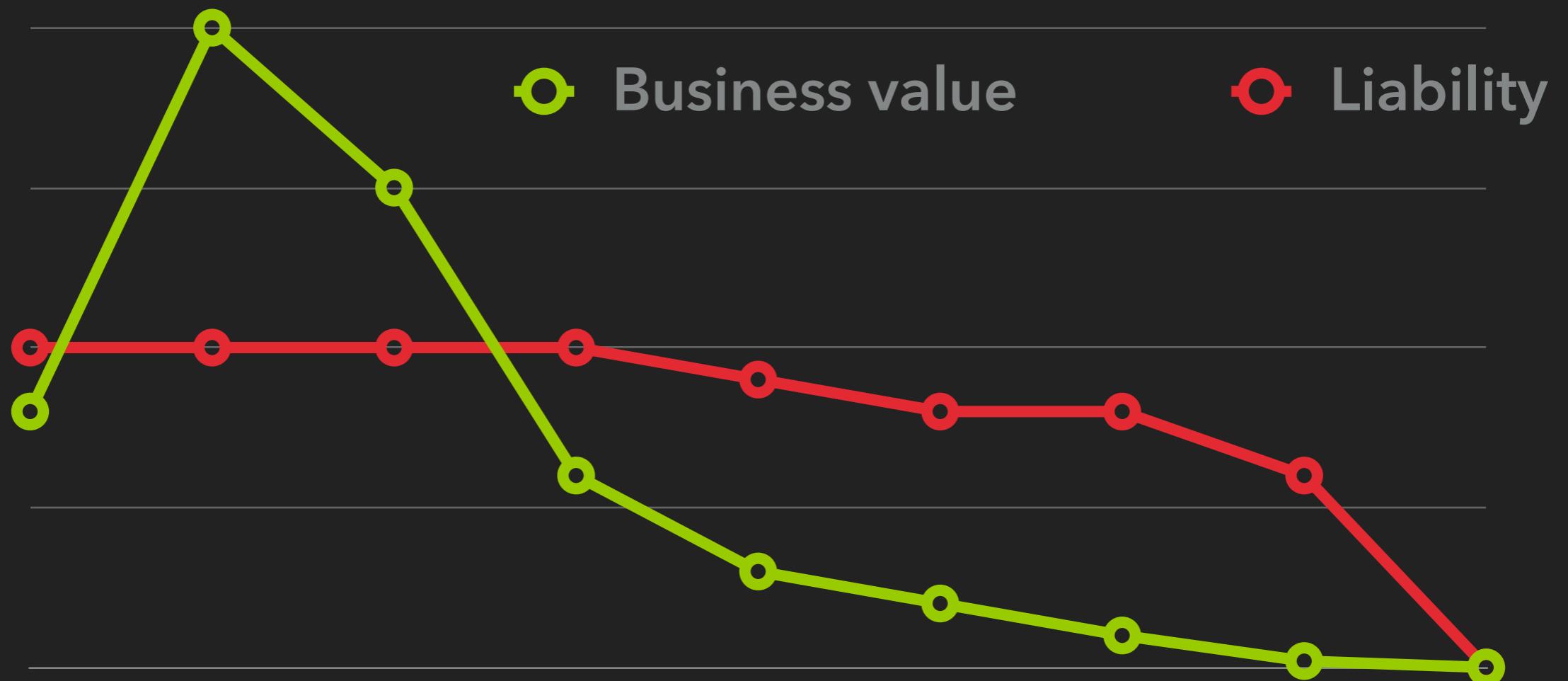
YOUR DATA



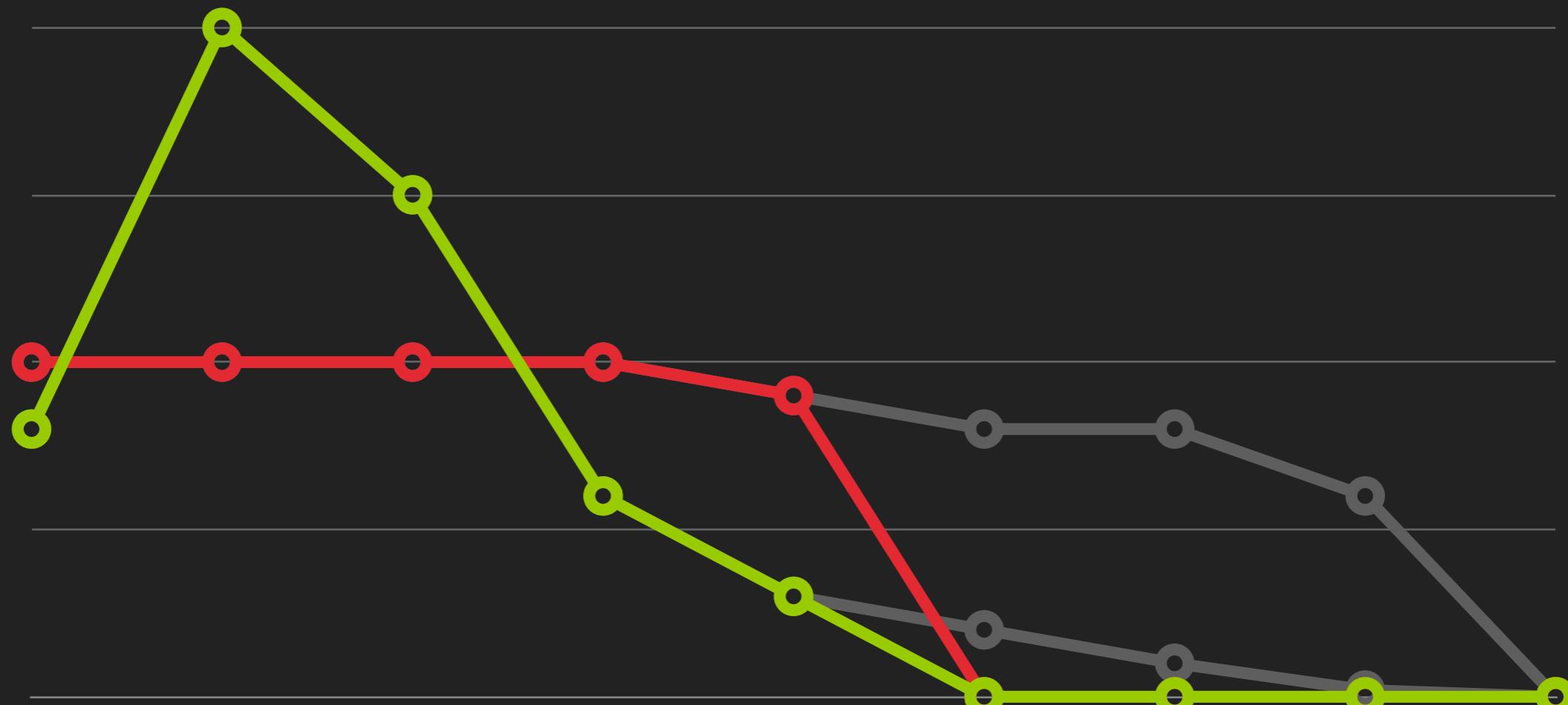
YOUR DATA



YOUR DATA



YOUR DATA



A blue house-shaped icon with a white outline. The word "DATA" is written in white capital letters on the front of the house. A white padlock is attached to the bottom right corner of the house.

DATA

WHAT IS

CONTENT
ENCRYPTION?

WHAT IS ...?

CONTENT ENCRYPTION IS LIKE AN AIRBAG

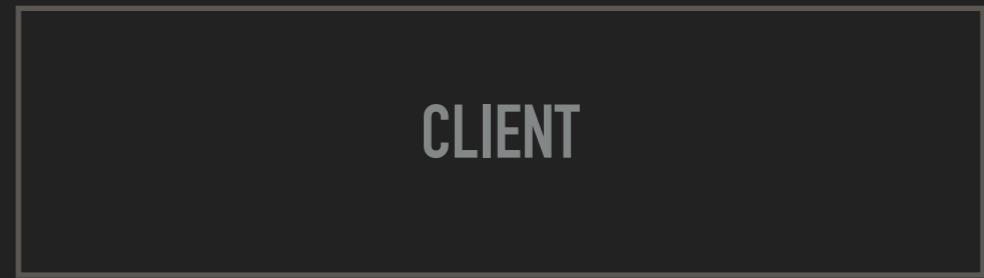
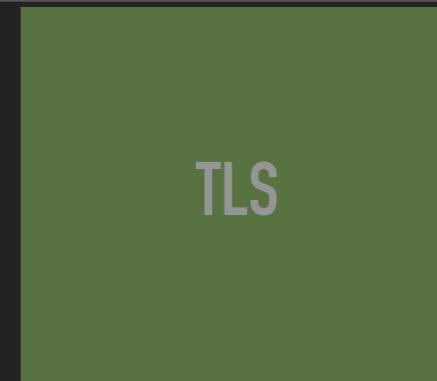
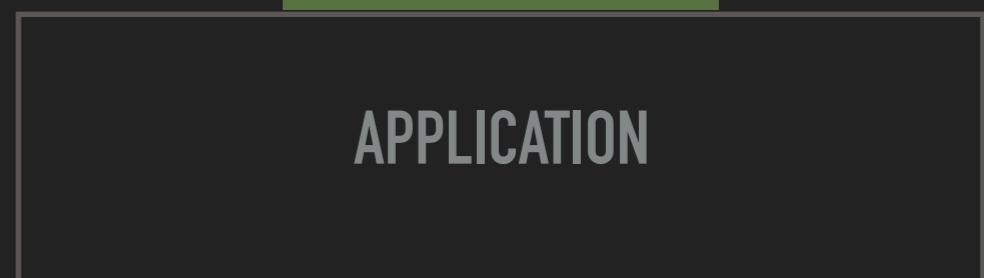
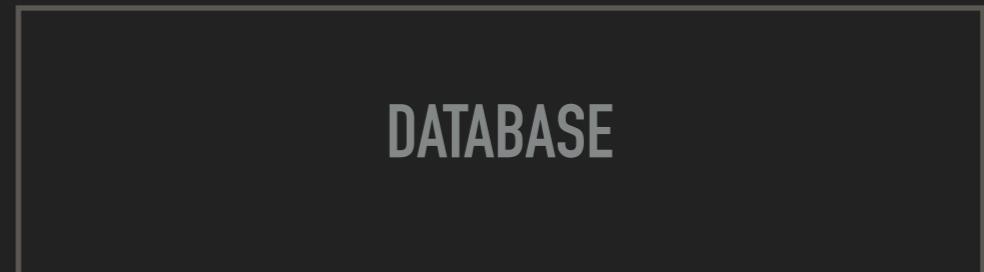
IMAGE: PINEAPPLE FEZ

THIS IMAGE IS LICENSED UNDER THE CREATIVE COMMONS ATTRIBUTION-SHARE ALIKE 3.0 UNPORTED LICENSE

[HTTPS://COMMONS.WIKIMEDIA.ORG/WIKI/FILE:SUZUKI_ALTO_BODY2_-_AIMS.JPG](https://commons.wikimedia.org/wiki/File:Suzuki_Alto_Body2_-_AIMS.jpg)

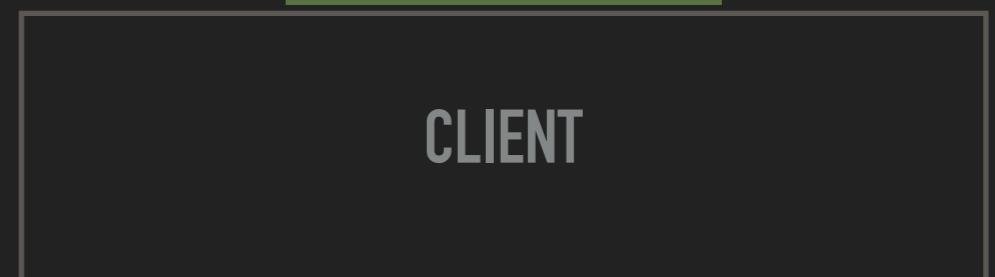
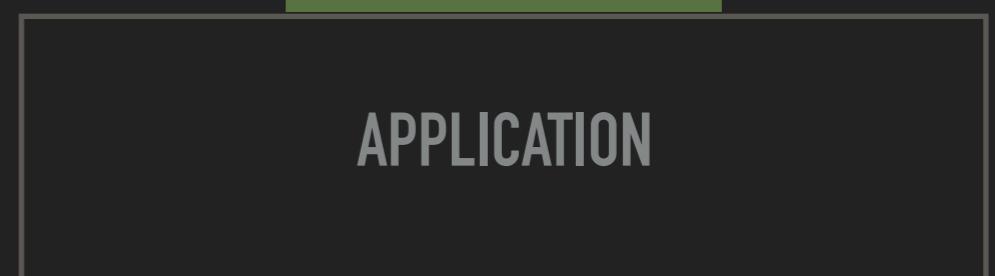
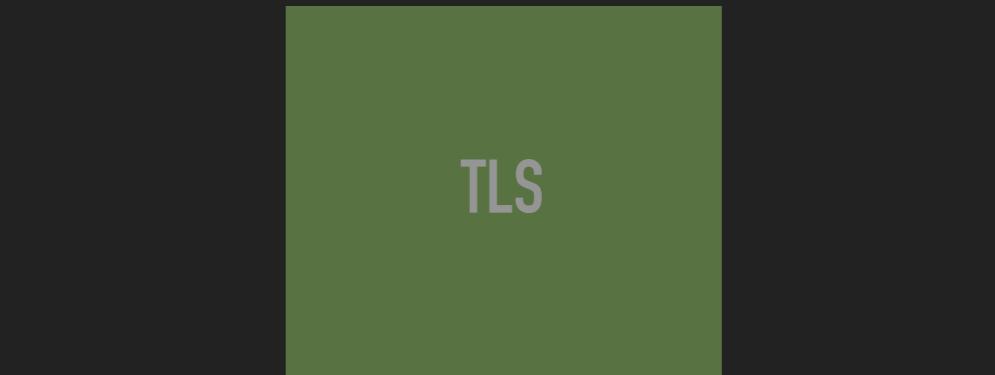
YOUR FATHERS CRYPTO (*)

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



YOUR FATHERS CRYPTO (*)

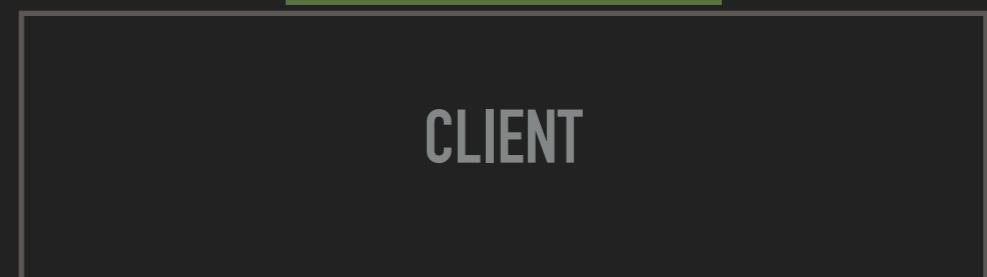
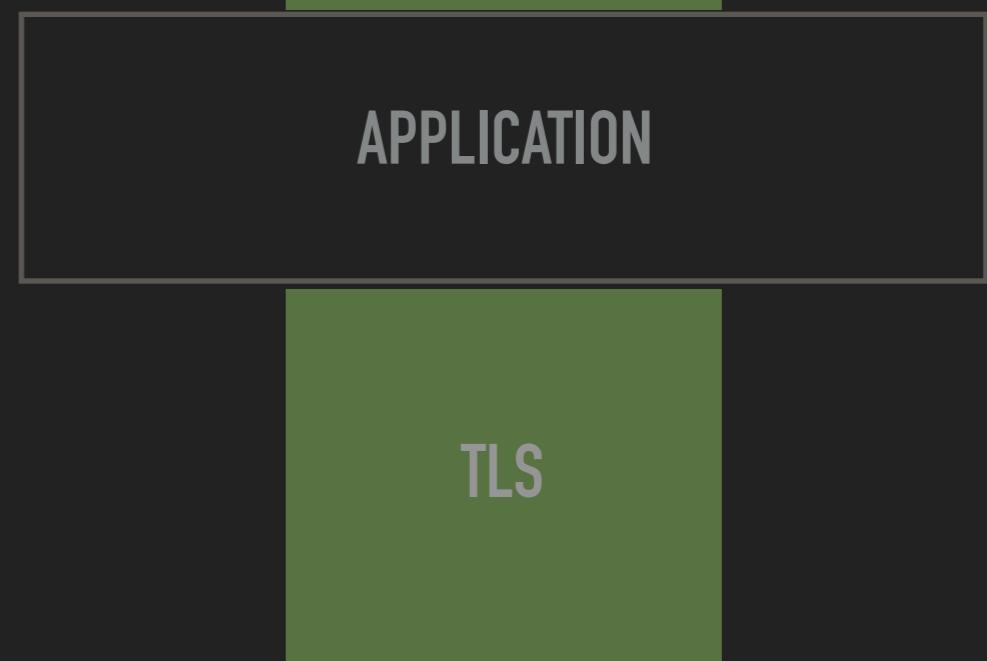
- ▶ CLIENT sends request
- ▶ APPLICATION applies logic
- ▶ DATABASE stores result
- ▶ Data encrypted via TLS 



YOUR FATHERS CRYPTO (*)

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

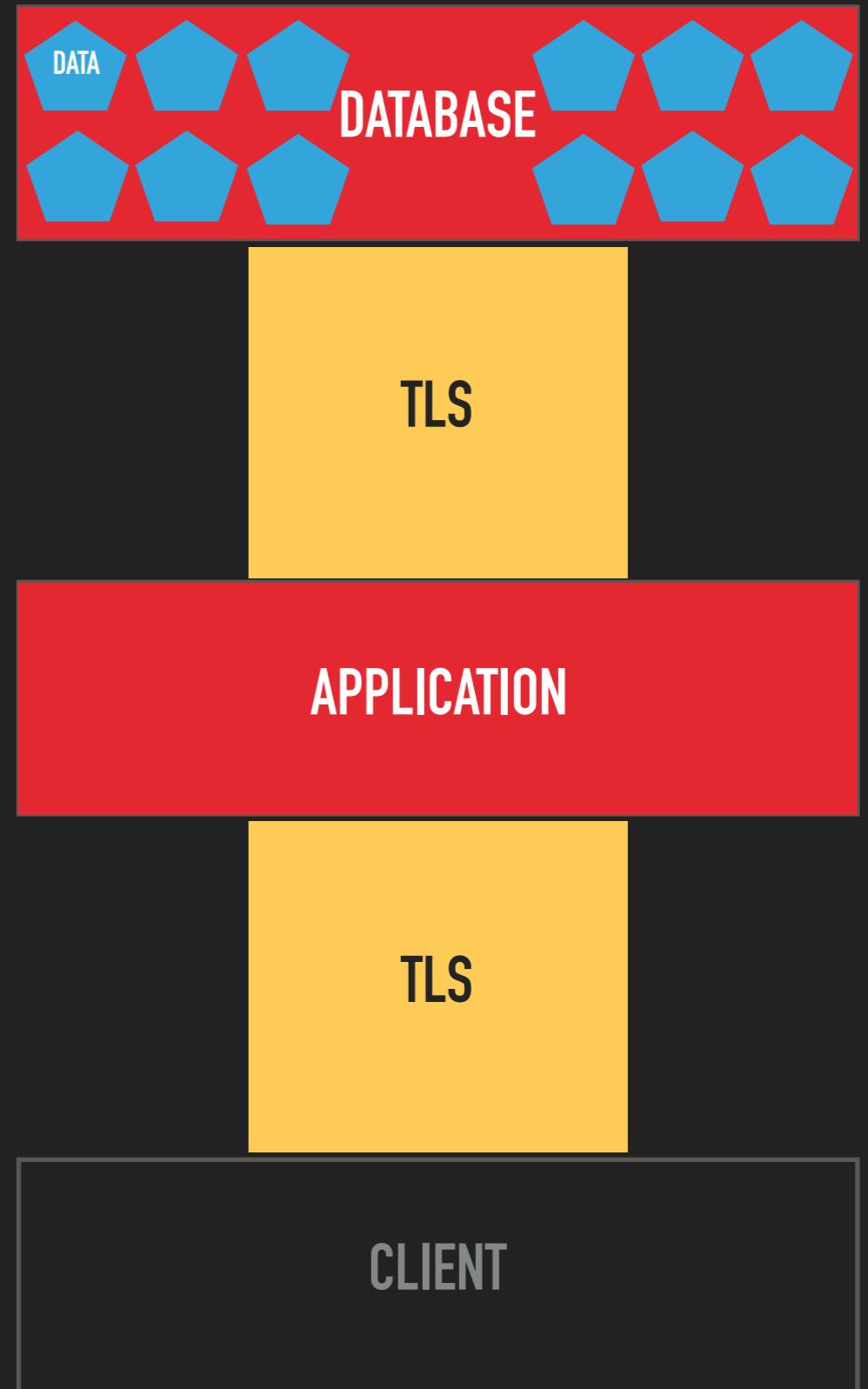
SO, EVERYTHING IS SAFE?



YOUR FATHERS CRYPTO (*)

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

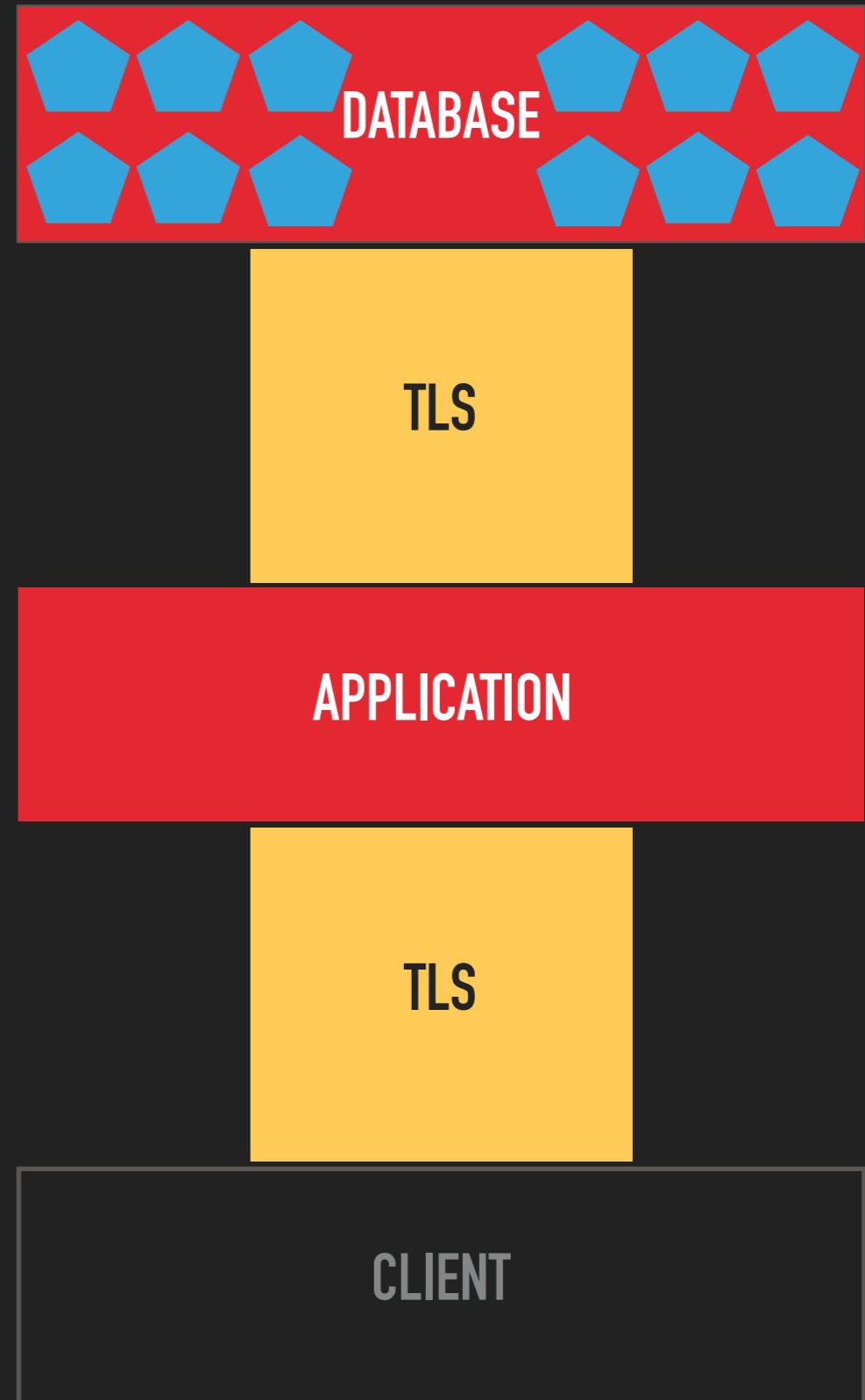
SO, EVERYTHING IS SAFE?



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

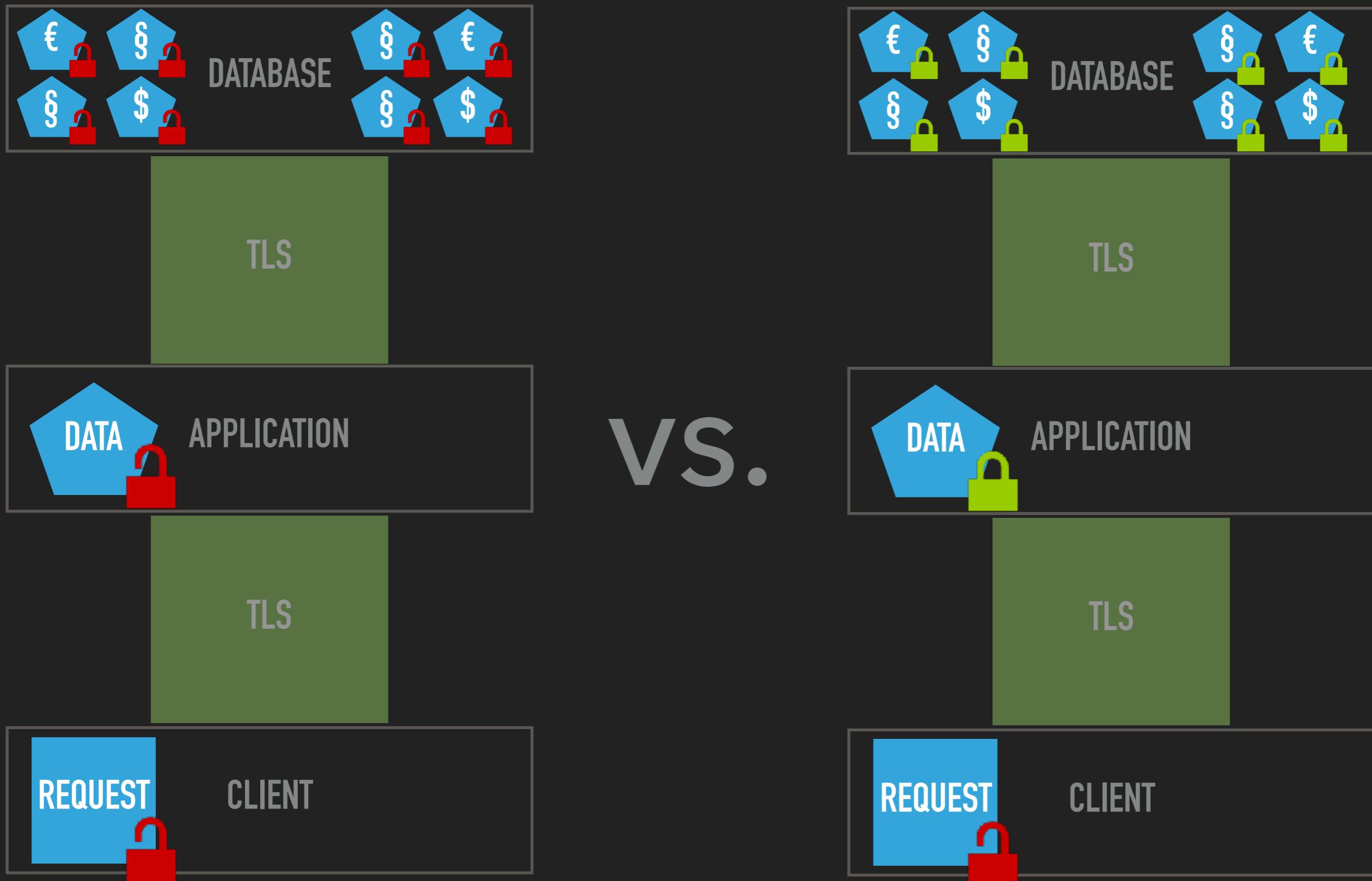
WHAT ABOUT TLS?

- ▶ 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





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



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, Gewerkschaftzugehörigkeit, 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]



BDG

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.**

► § 16 Bußgeldvorschriften

...

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



TMG

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!

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.

STGB

SOZIALGESETZBUCH

Applies to: Everyone working with social data

- ▶ You know when it applies!



SGB

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

IT'S COMPLIANCE

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



NOT
CONVINCED
YET?

**ASK THE
COMPETITION!**

ASK THE COMPETITION PT. 1



<https://www.bloomberg.com/quote/TALK:LN>

<https://www.theguardian.com/business/2016/may/12/talktalk-profits-halve-hack-cyber-attack>

<https://www.theguardian.com/business/2016/feb/02/talktalk-cyberattack-costs-customers-leave>

ASK THE COMPETITION PT. 2

Adultery Website AshleyMadison Seeks IPO as Demand Booms

Kristen Schweizer

18 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

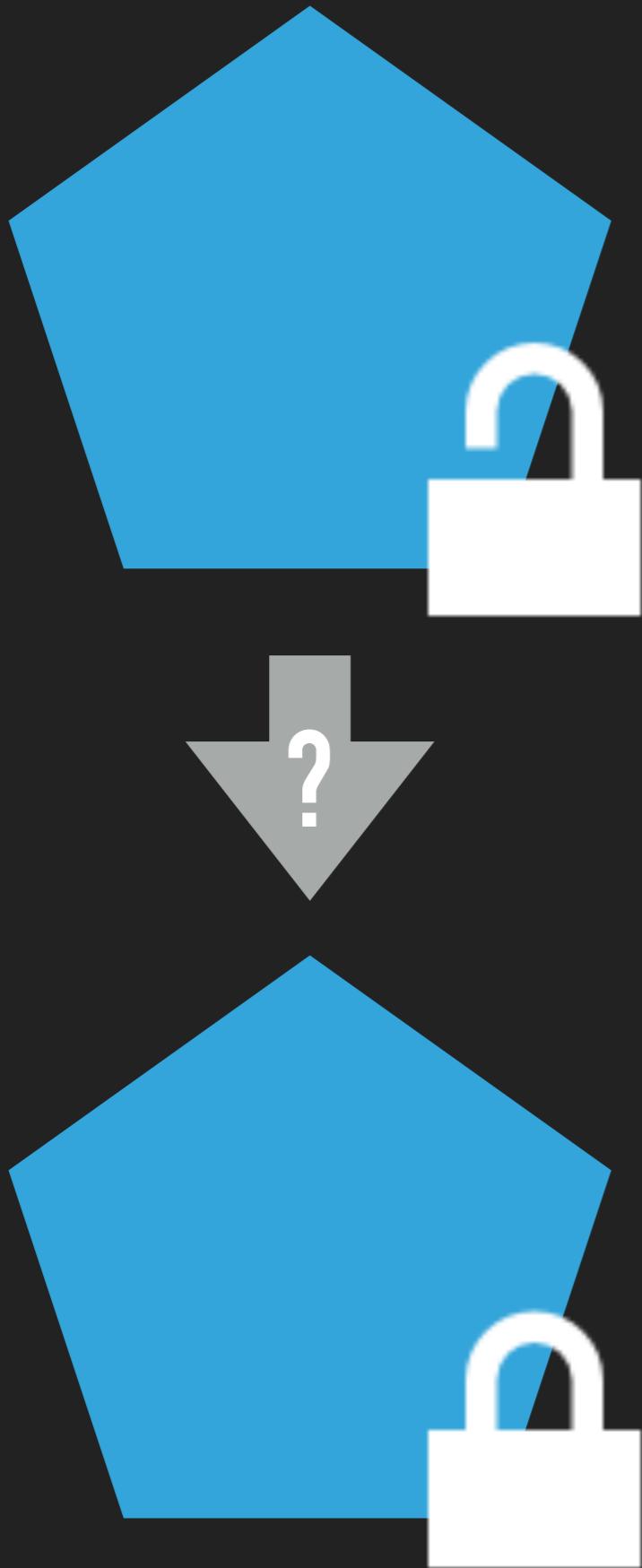
Breached Organizations Lose Millions in Market Value, Finds New Report

“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-market-value-finds-new-report/>

<http://breachlevelindex.com/>



HOW TO DESIGN CONTENT ENCRYPTION

0 - REGULATIONS & INITIAL RISK ASSESSMENT

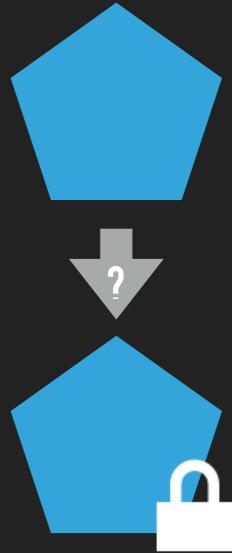
1 - DATA CLASSIFICATION

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.

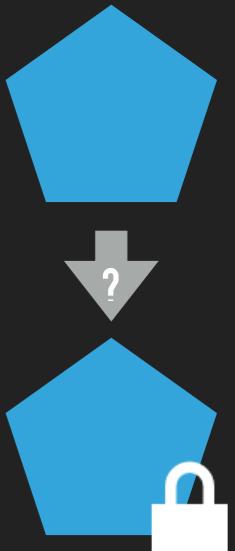
THIS DEPENDS ON YOUR SPECIFIC APPLICATION/SCENARIO!

I AM NOT A LAWYER!

3 - IMPLEMENTATION PLAN

Update the requirements

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



RECAP

0 - Regulations & initial risk assessment



1 - Data classification

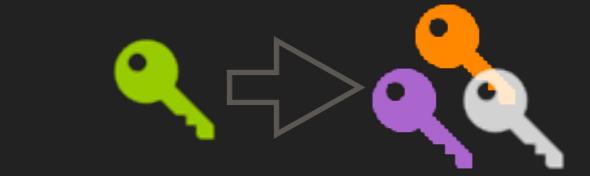
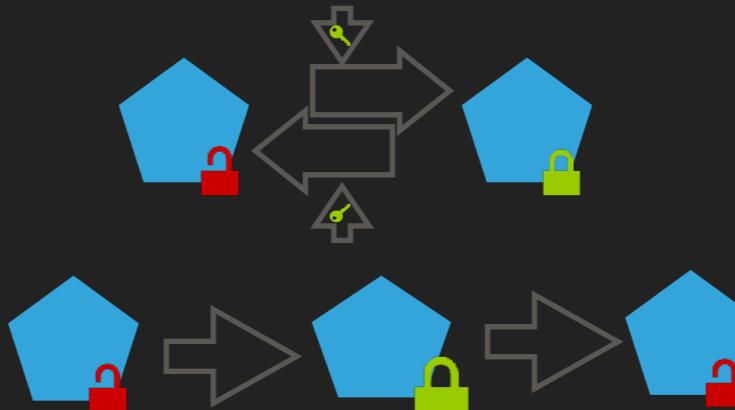


2 - Data treatment plan



3 - Implementation Plan





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

- Data treatment ...
- Use existing ...
- ...

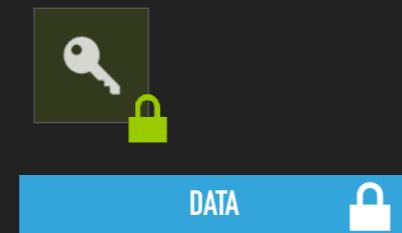


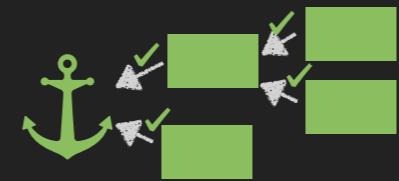
CONTENT ENCRYPTION

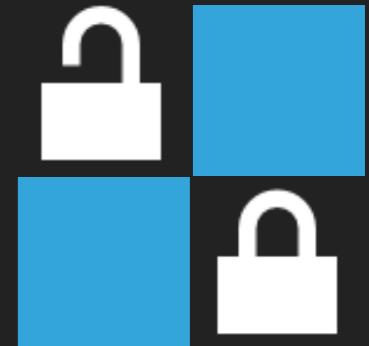
PATTERNS

```

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



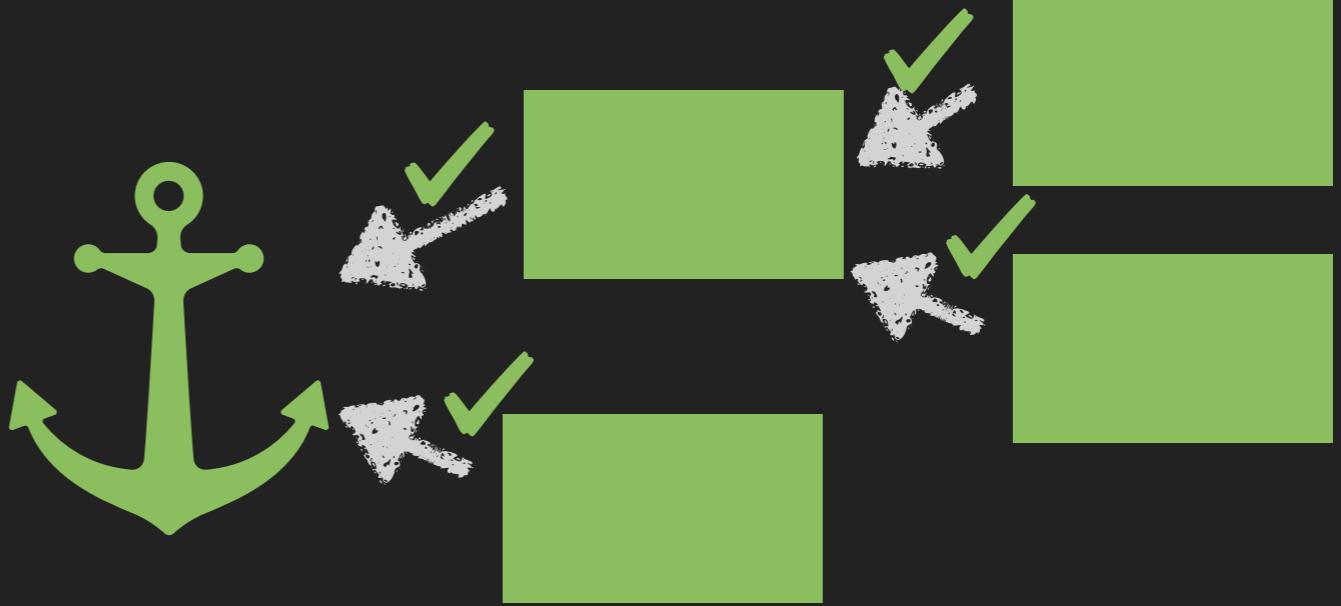
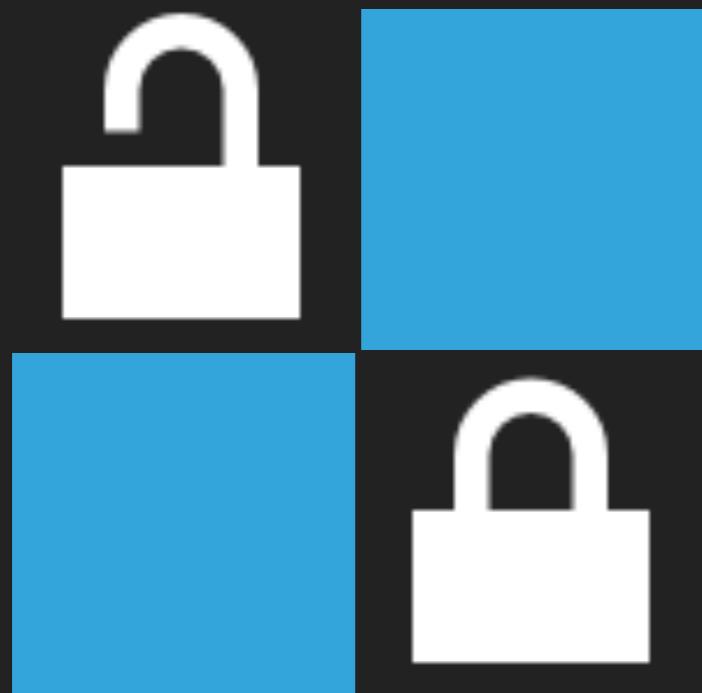




MOST IMPORTANT ADVICE: GET AN EXPERT OR AT LEAST READ AND UNDERSTAND THE DOCUMENTATION!

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 modeling not discussed here but it is really important!*

TRUST ANCHORS



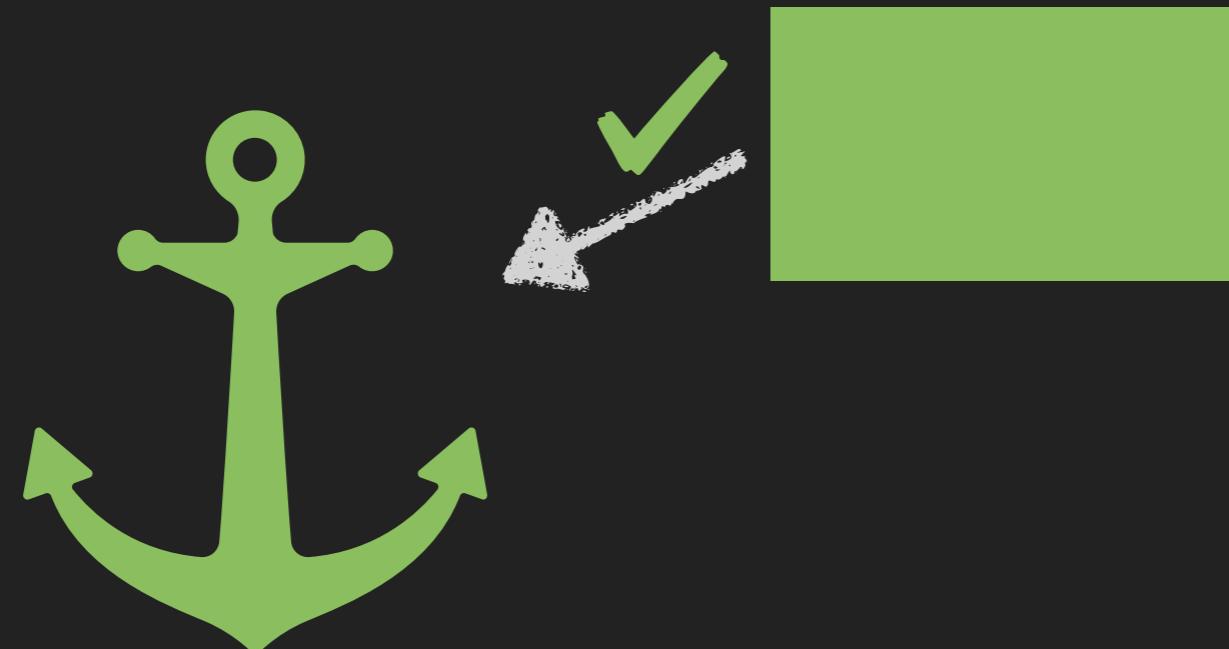
1. You trust your trust anchor(s) – they are secure by definition

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



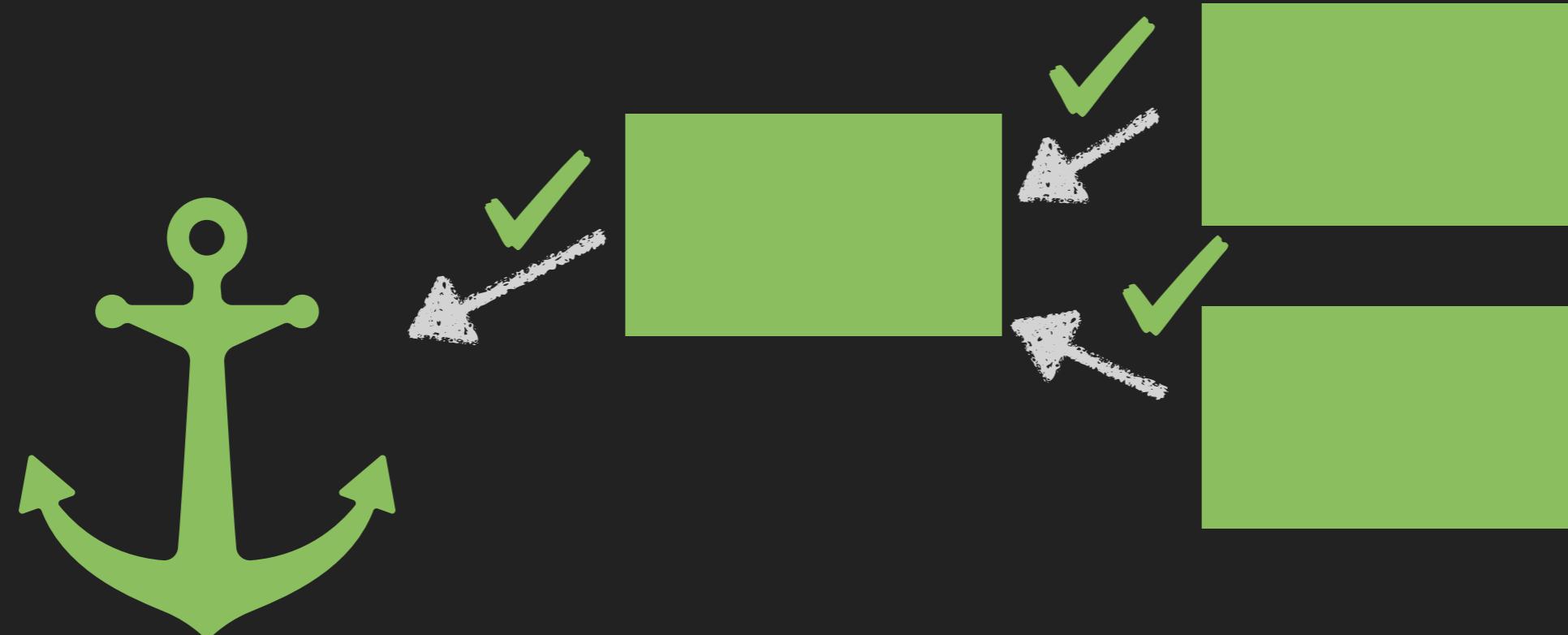
1. You trust your trust anchor(s) – they are secure by definition
2. You reason that something else is also trusted by relying on the trust anchor(s)
3. Therefore trust can be extended

TRUST ANCHORS



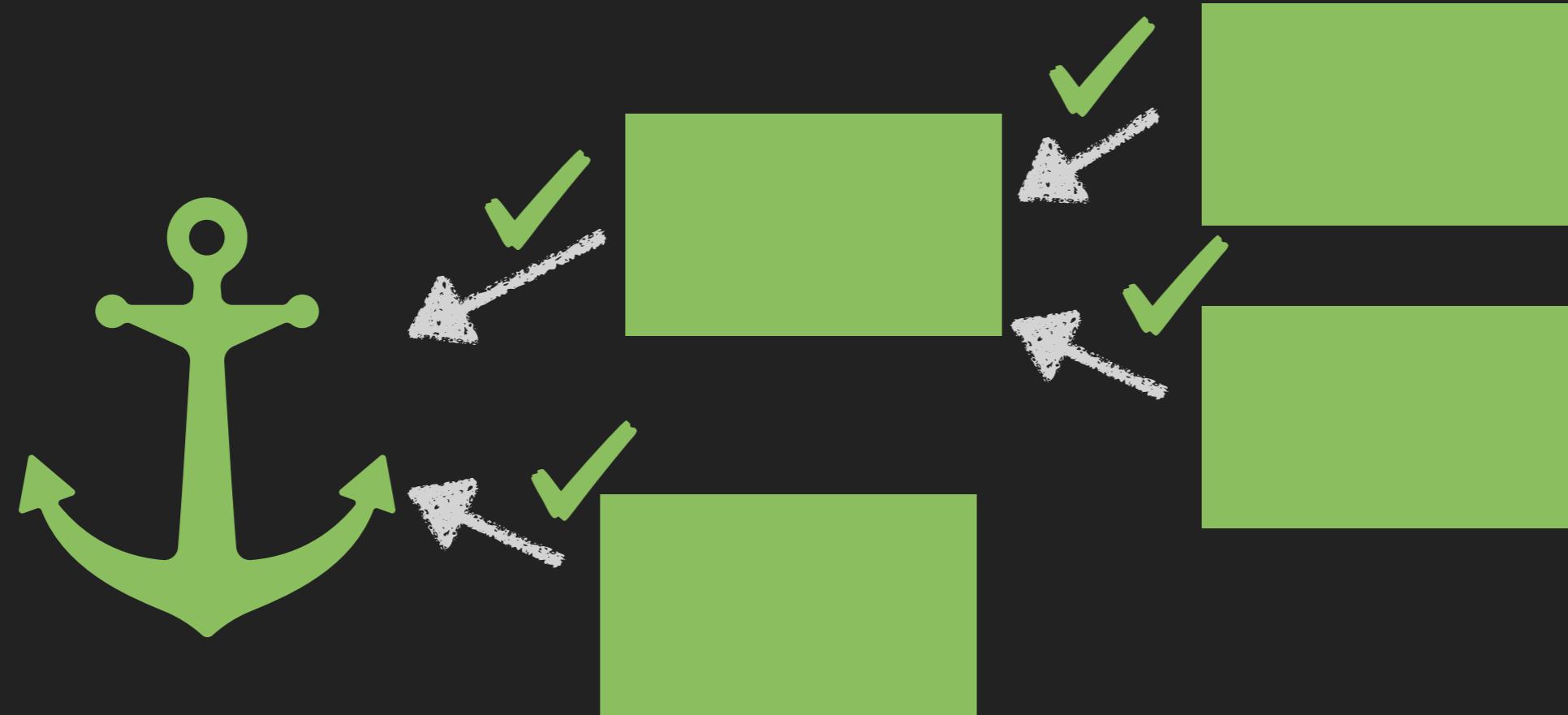
1. You trust your trust anchor(s) – they are secure by definition
2. You reason that something else is also trusted by relying on the trust anchor(s)
3. Therefore trust can be extended

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



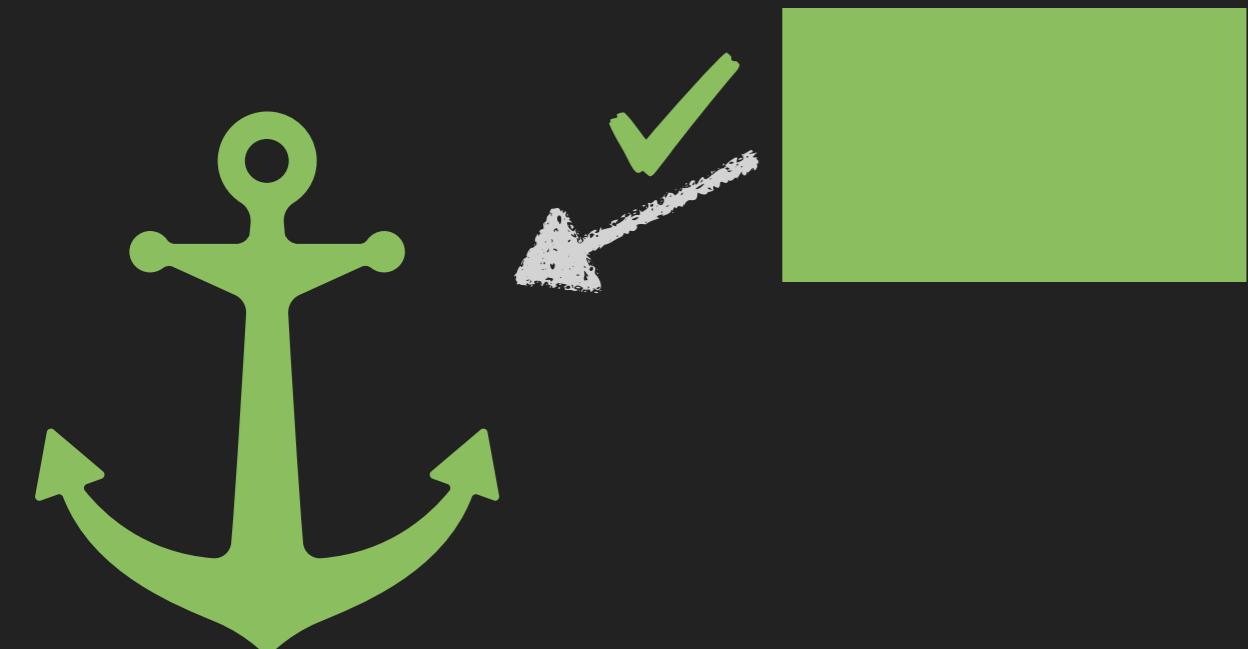
1. You trust your trust anchor(s) – they are secure by definition
2. You reason that something else is also trusted by relying on the trust anchor(s)
3. Therefore trust can be extended

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



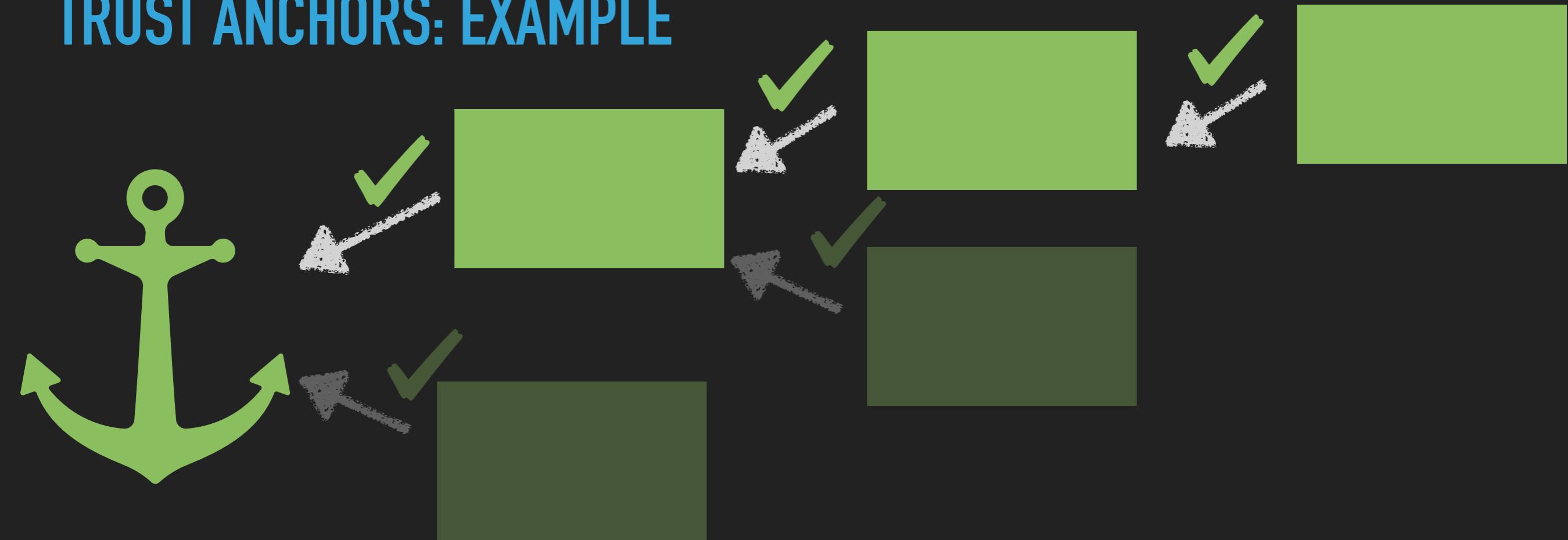
1. I trust my cloud provider
2. I trust that my cloud provider has a secure key storage
3. Data encrypted with keys stored in the keyring is secure

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

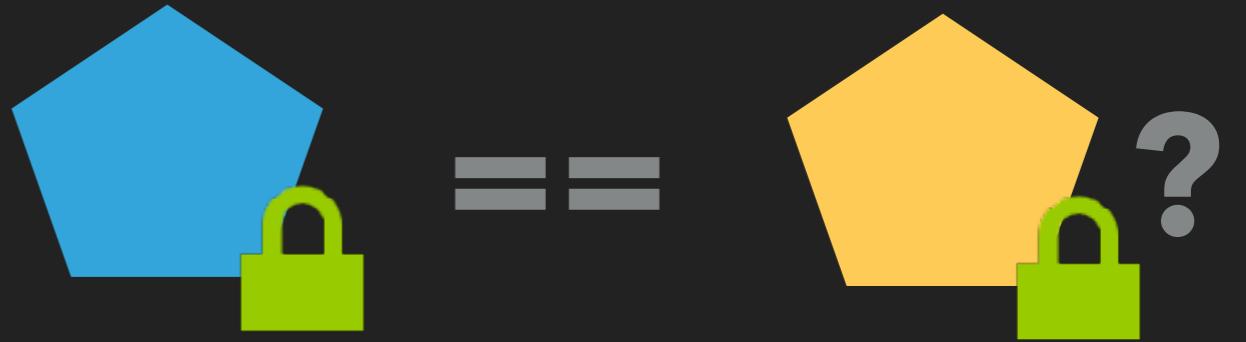
TRUST ANCHORS: EXAMPLE



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

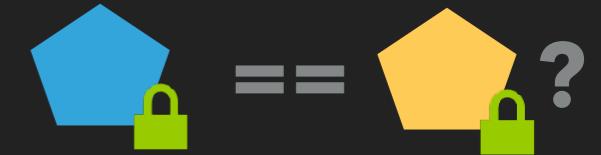




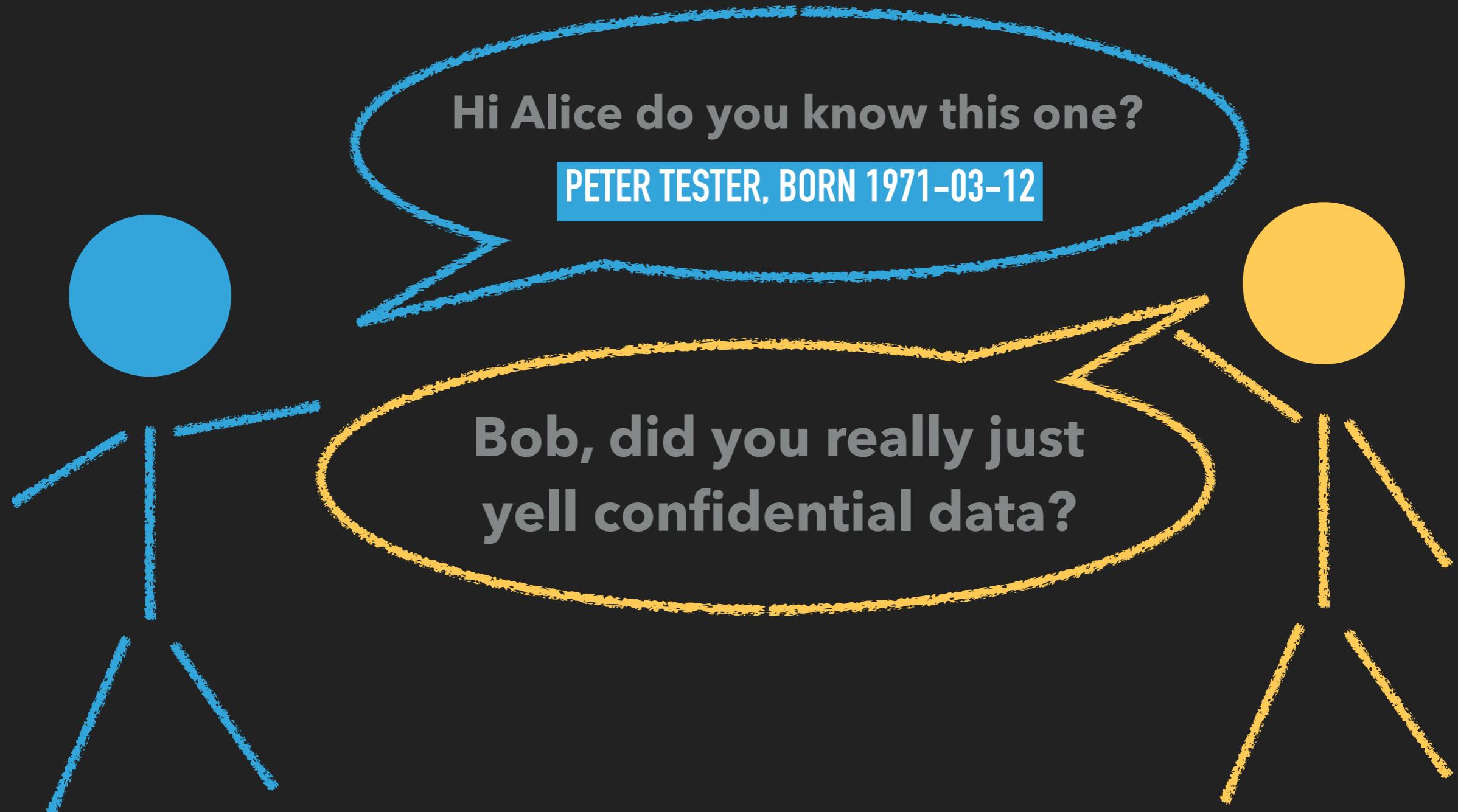
PATTERNS

COMPARING

COMPARE DATA



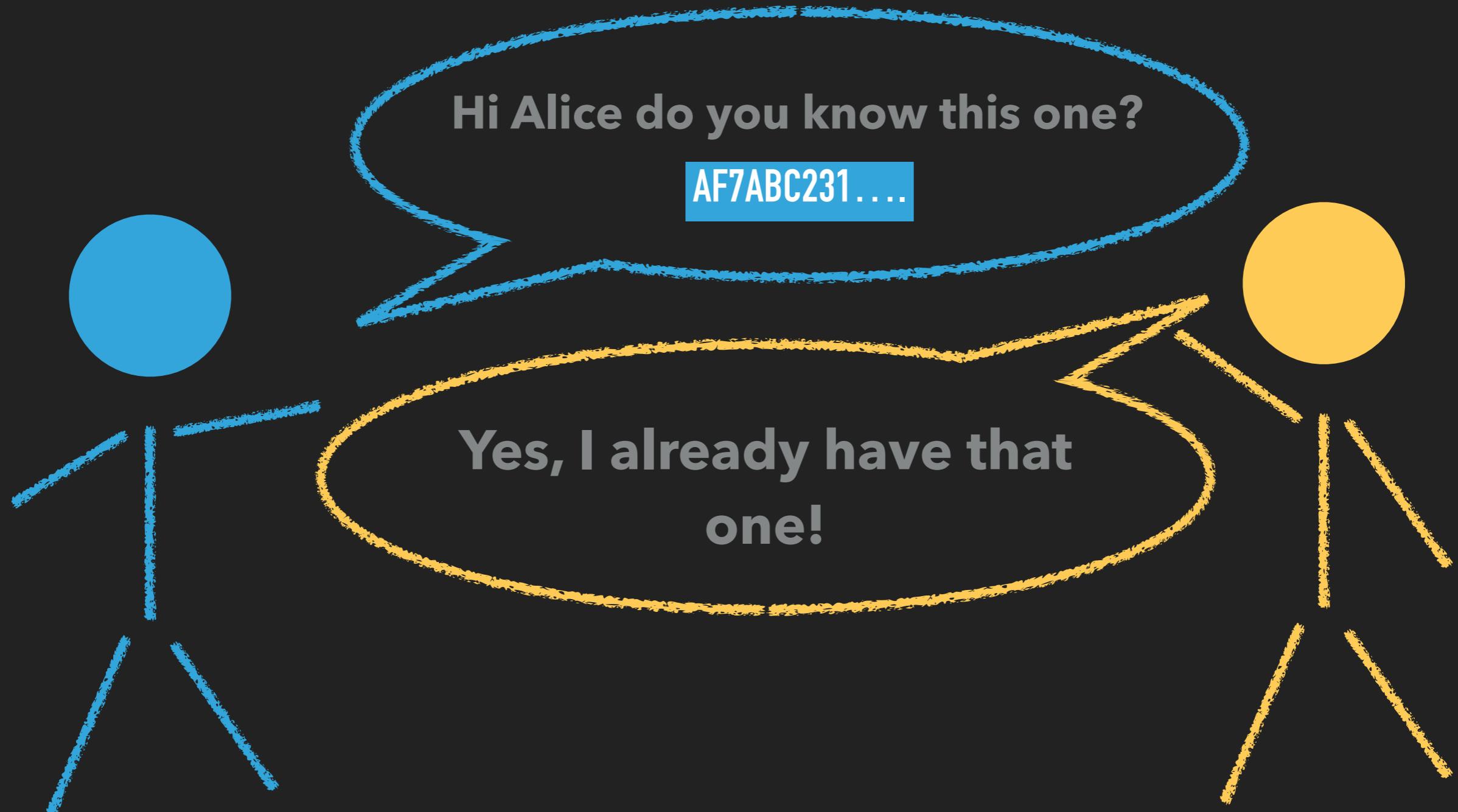
Problem: Securely compare two data items



COMPARE DATA



Problem: Securely compare two data items



COMPARE DATA



Problem: Securely compare two data items

Solution: Normalise & hash data, compare hashes

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

COMPARE DATA



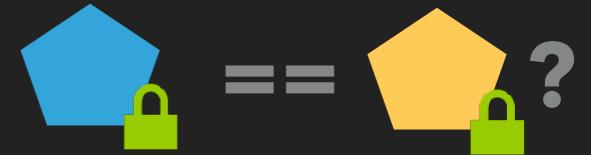
Problem: Securely compare two data items

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

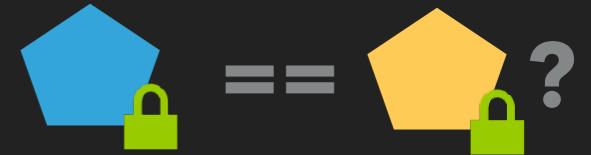
=>
★

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 ...

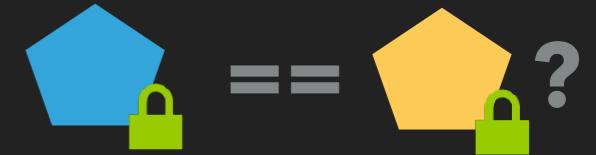
sha256(LOREM IPSUM ...) == sha256(LOREM IPSUM ...)

<=>

4C53E9C9... == 4C53E9C9...

- ★ Collisions [$A \neq B$ but $\text{sha256}(A) == \text{sha256}(B)$] 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 $\text{hash}(\text{salt} + \text{data})$ 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



PATTERNS

TRANSPARENT
ENCRYPTION

TRANSPARENT ENCRYPTION



What?

**Transparent encryption
is also transparent to the attacker**

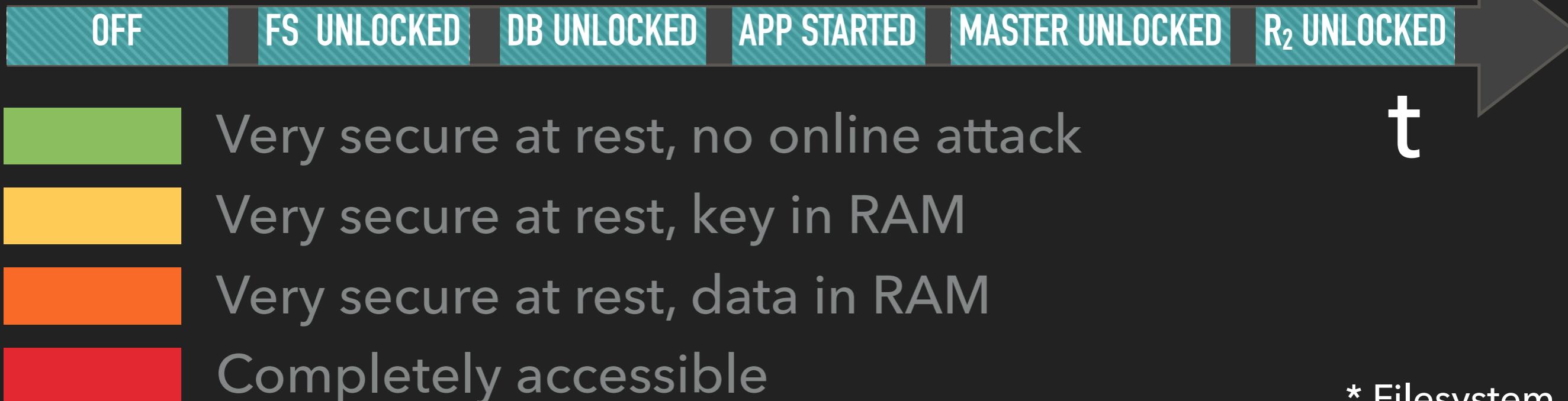
?!?!
?!?!



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

Help against stolen hard disks.

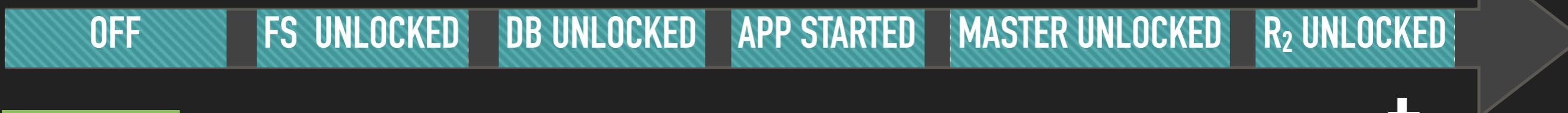
TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



FS* 



 Very secure at rest, no online attack 

 Very secure at rest, key in RAM

 Very secure at rest, data in RAM

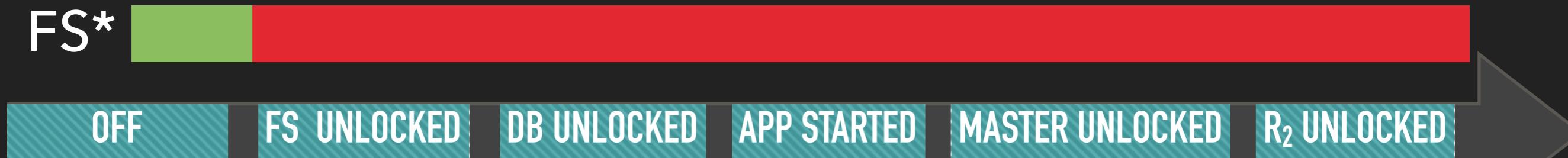
 Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



FS*



Very secure at rest, no online attack

t

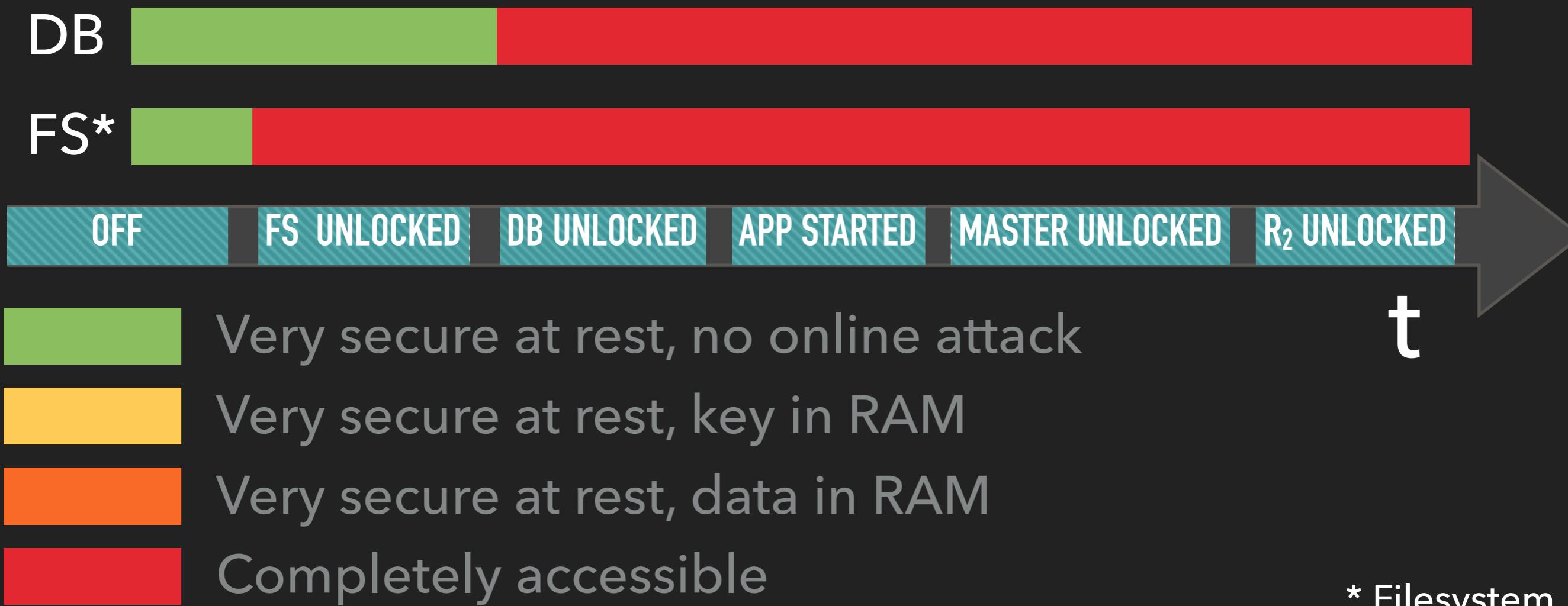
Very secure at rest, key in RAM

Very secure at rest, data in RAM

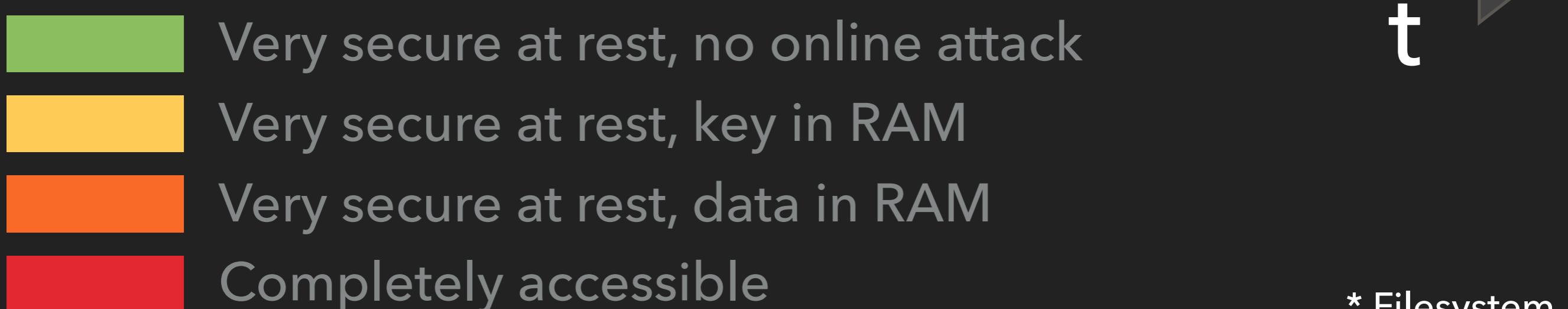
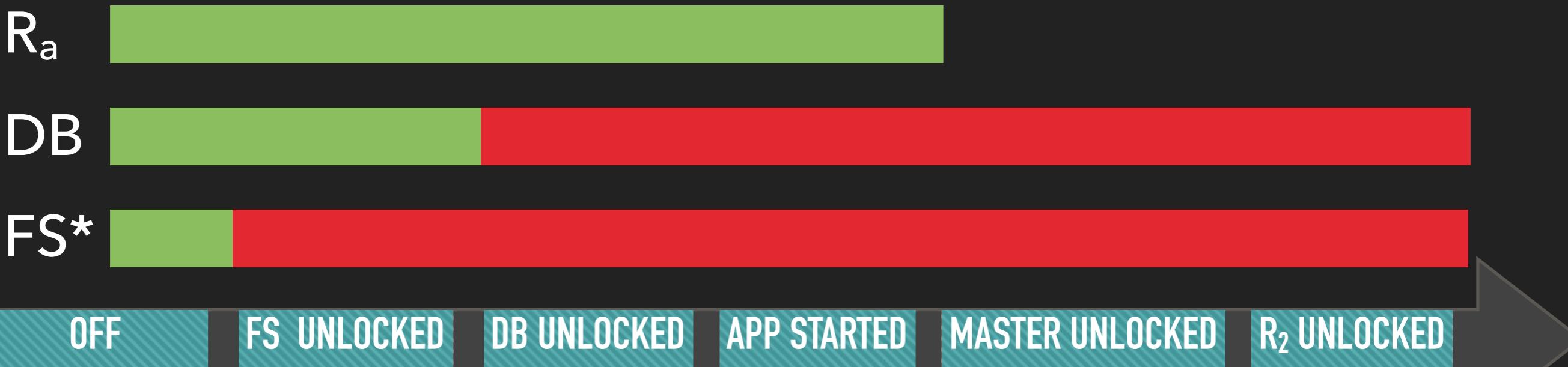
Completely accessible

* Filesystem

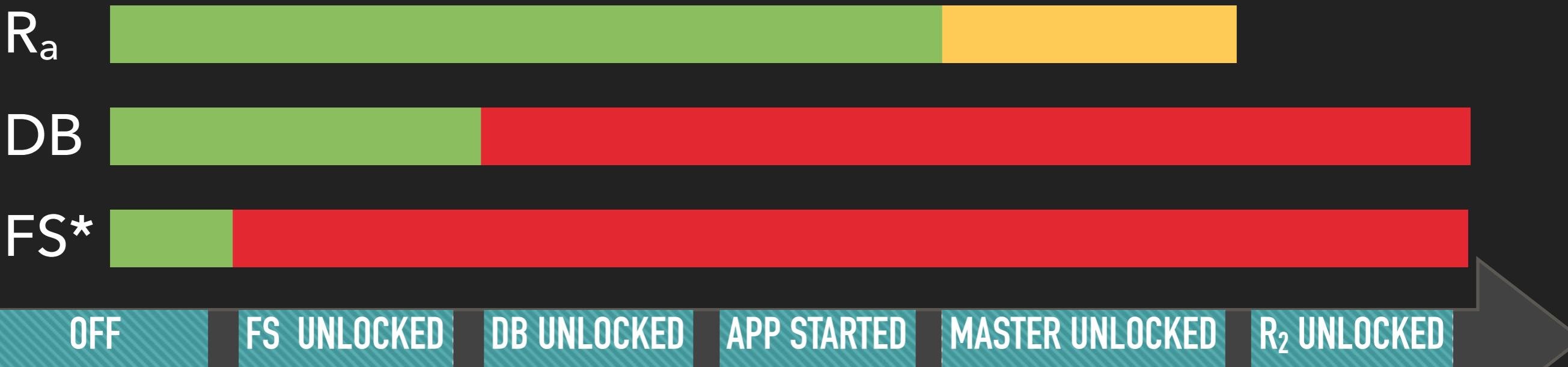
TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



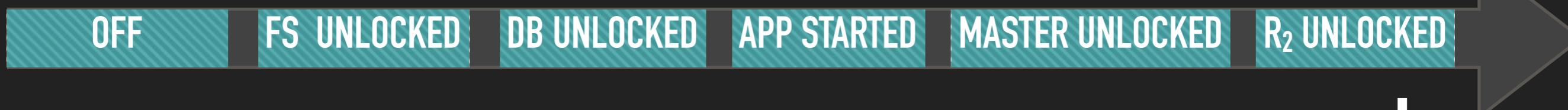
TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



-
- A legend consisting of four colored squares with corresponding labels:
- Green square: Very secure at rest, no online attack
 - Yellow square: Very secure at rest, key in RAM
 - Orange square: Very secure at rest, data in RAM
 - Red square: Completely accessible
- On the far right, there is a large italicized letter 't' and a note: *** Filesystem**.

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)

Attack



Very secure at rest, no online attack t

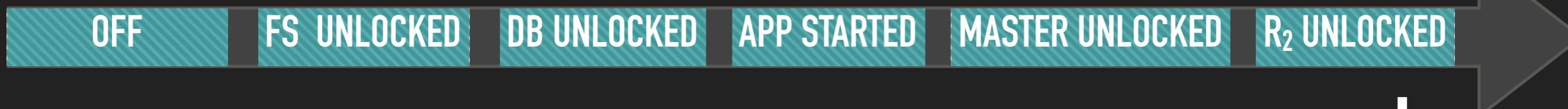
Very secure at rest, key in RAM

Very secure at rest, data in RAM

Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



Very secure at rest, no online attack t

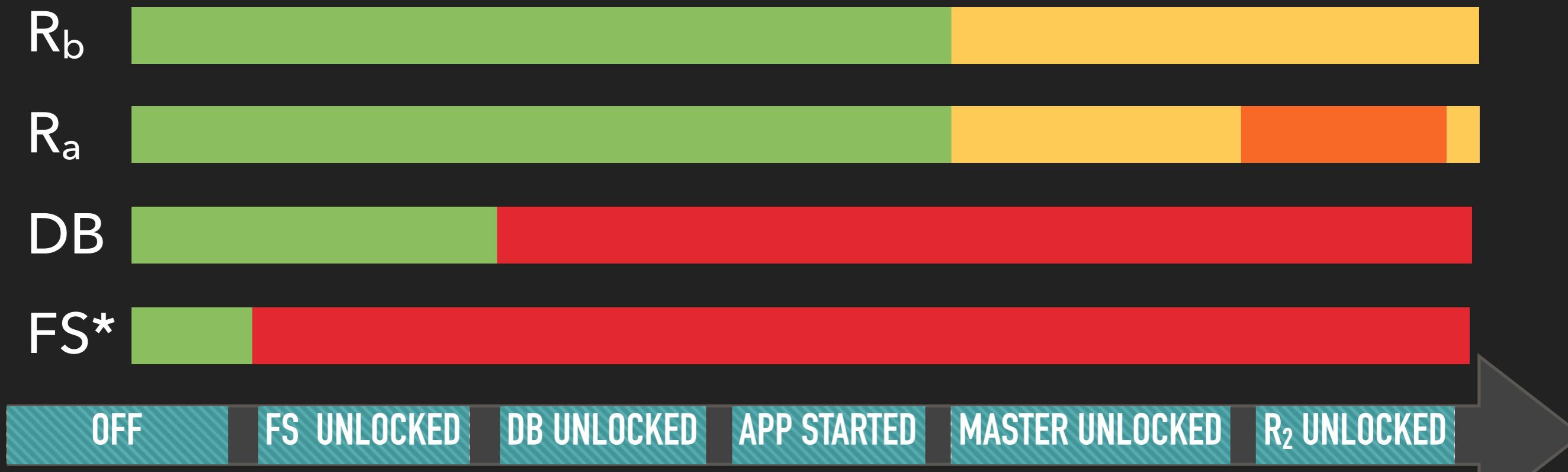
Very secure at rest, key in RAM

Very secure at rest, data in RAM

Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



Very secure at rest, no online attack t

Very secure at rest, key in RAM

Very secure at rest, data in RAM

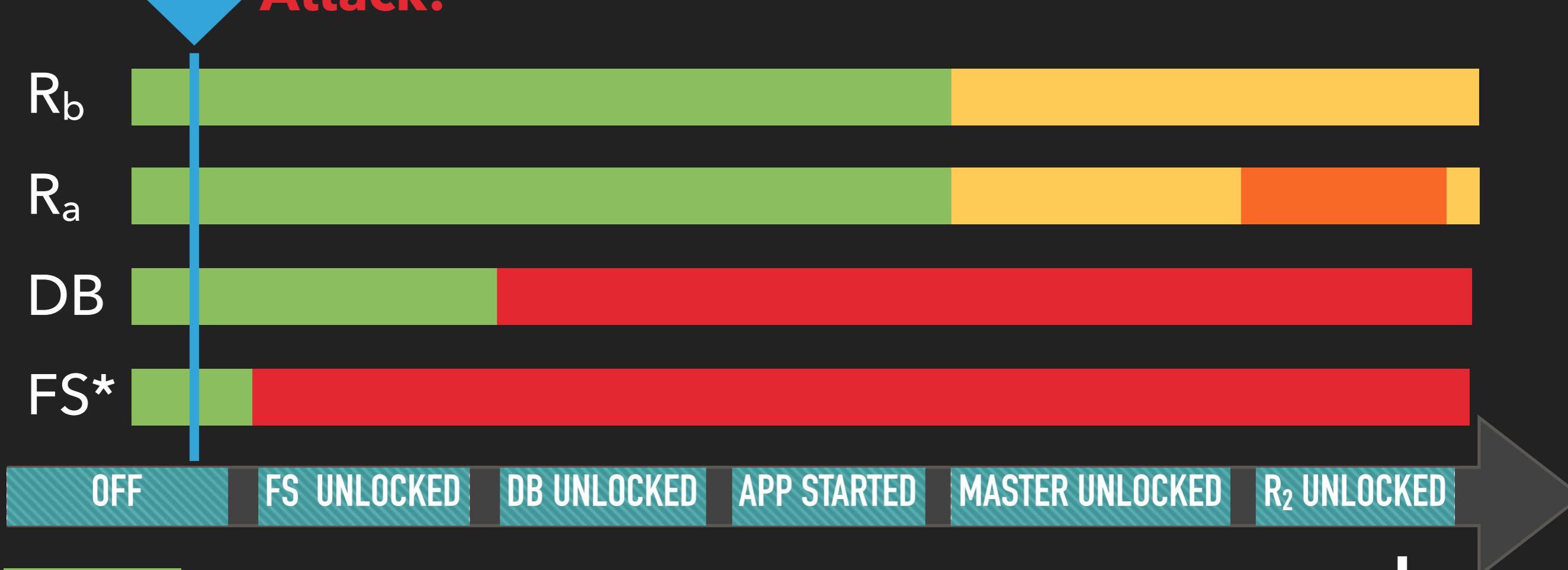
Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



Attack!



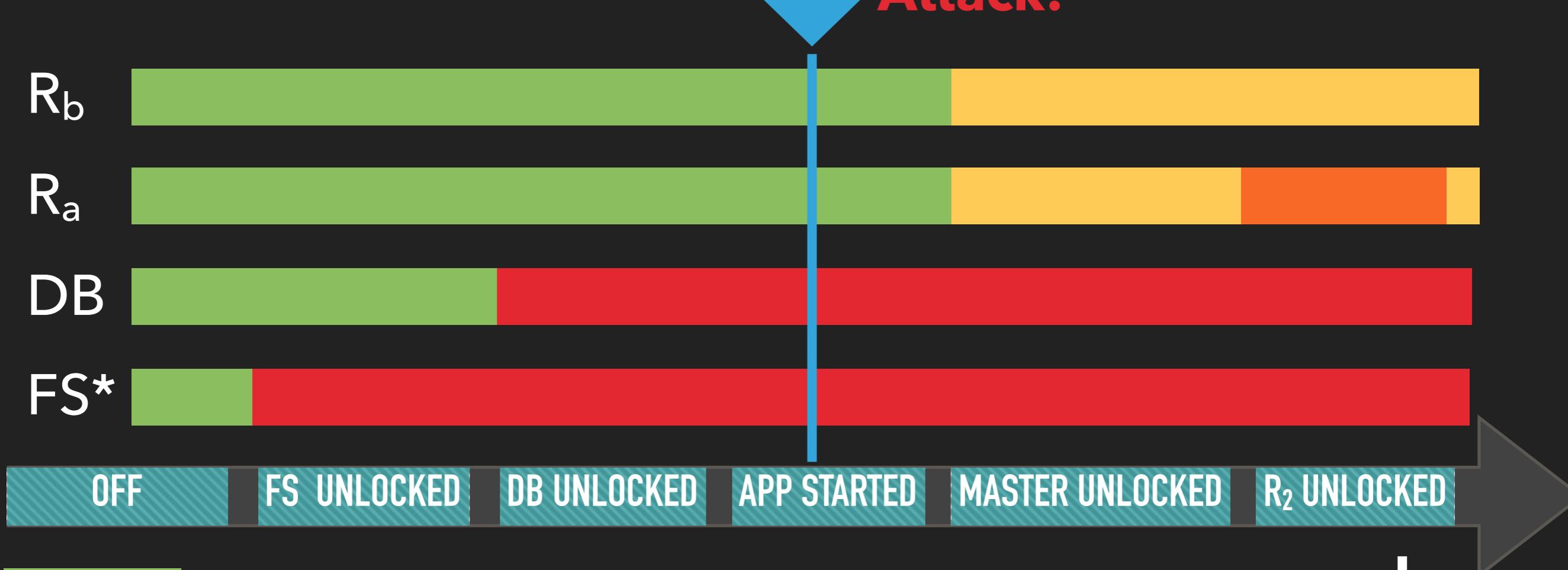
- Very secure at rest, no online attack t
- Very secure at rest, key in RAM
- Very secure at rest, data in RAM
- Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



Attack!



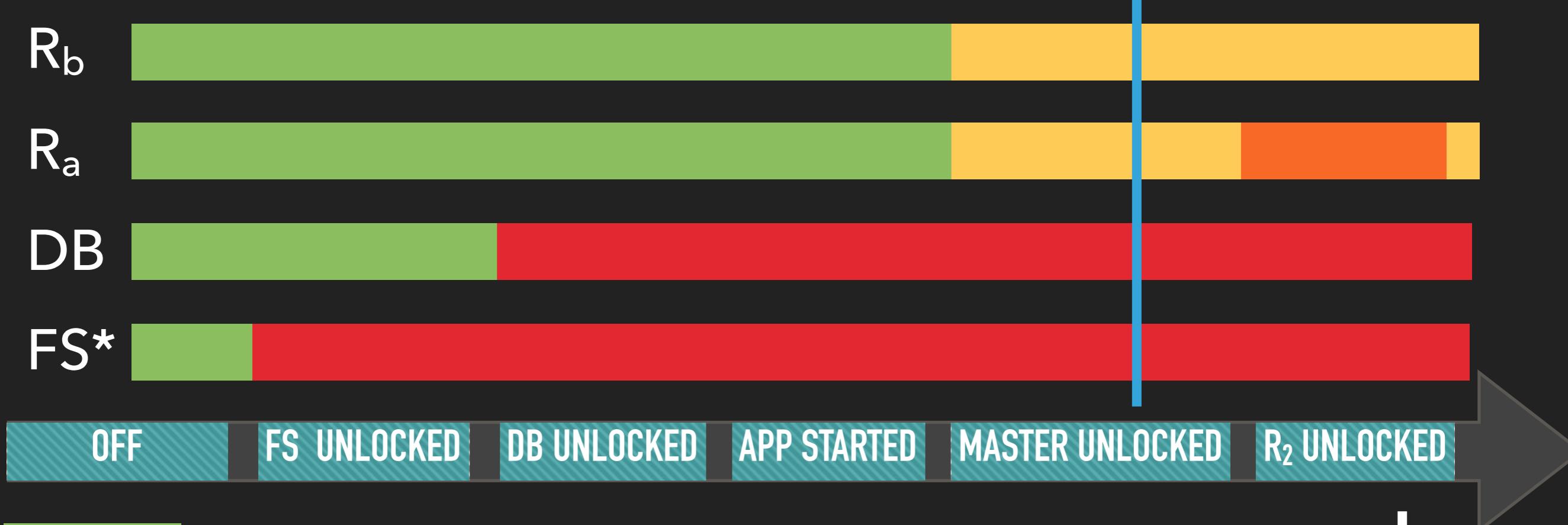
- Very secure at rest, no online attack t
- Very secure at rest, key in RAM
- Very secure at rest, data in RAM
- Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



Attack!



Very secure at rest, no online attack

t

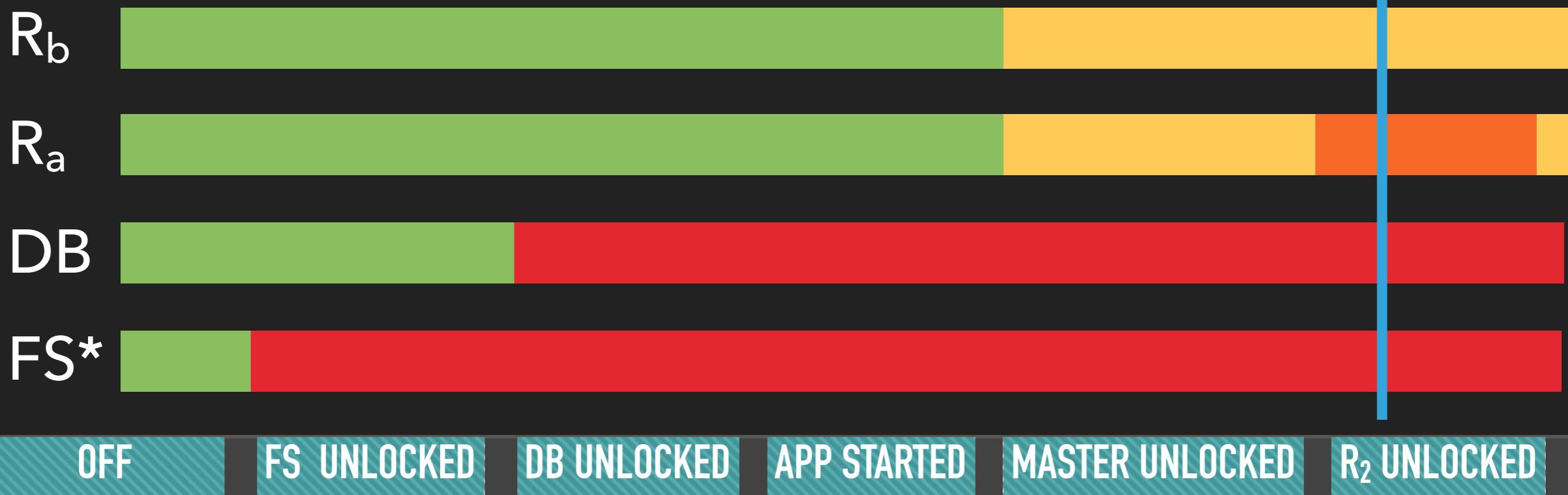
Very secure at rest, key in RAM

Very secure at rest, data in RAM

Completely accessible

* Filesystem

TRANSPARENT ENCRYPTION (ATTACKS VIA APPLICATION)



Very secure at rest, no online attack

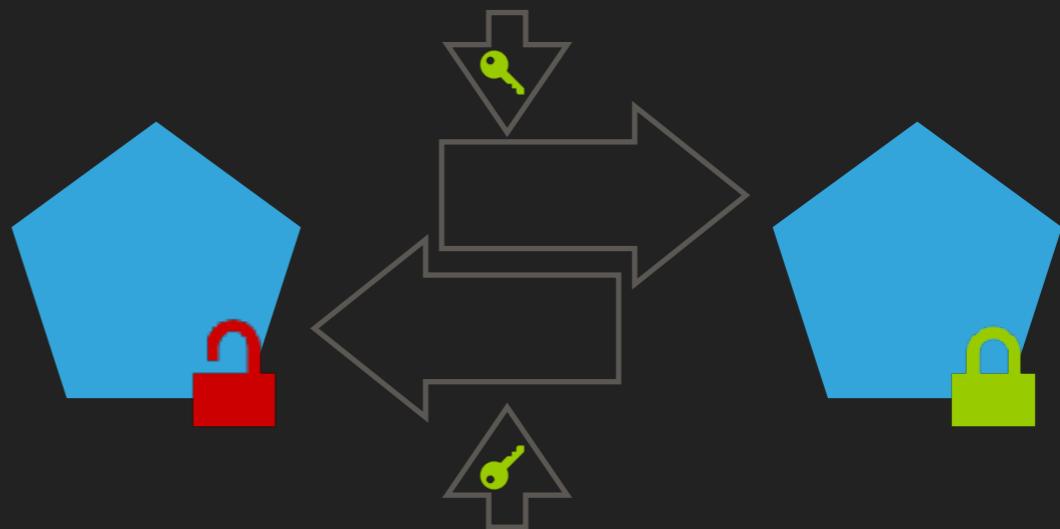
t

Very secure at rest, key in RAM

Very secure at rest, data in RAM

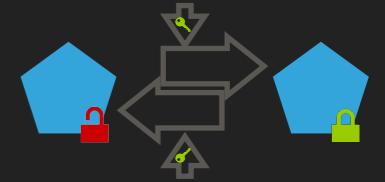
Completely accessible

* Filesystem



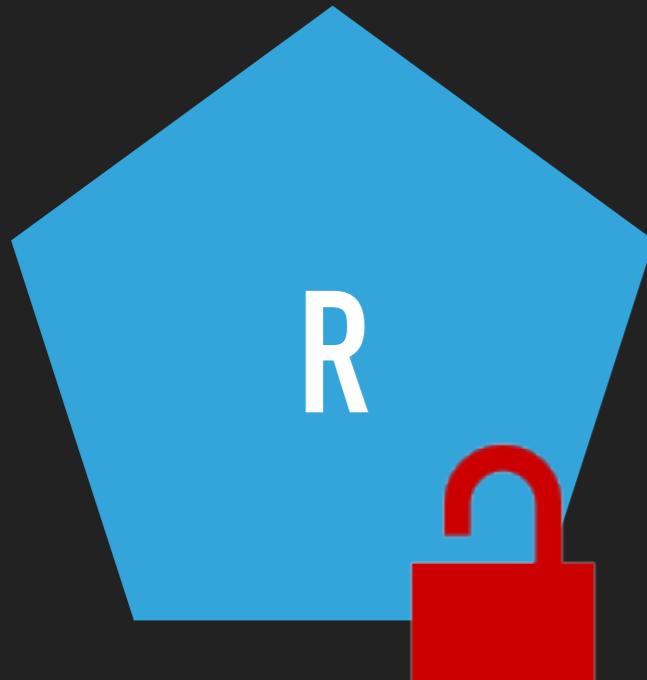
PATTERNS STORING

SECURE MULTIPLE DATA RECORDS



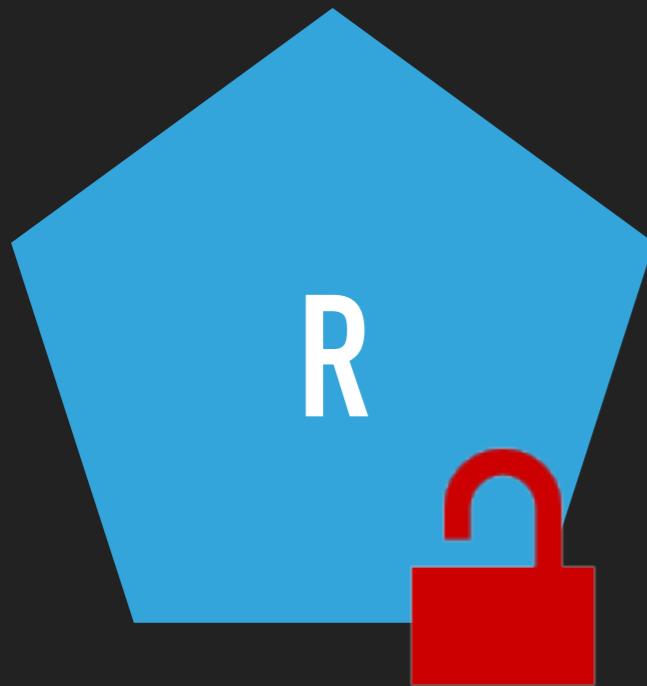
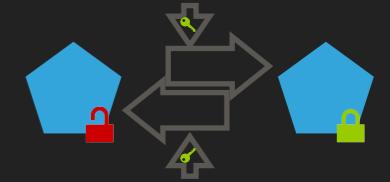
Problem: Multiple records are read and written by the same application.

Solution: Use symmetric encryption to protect the records.



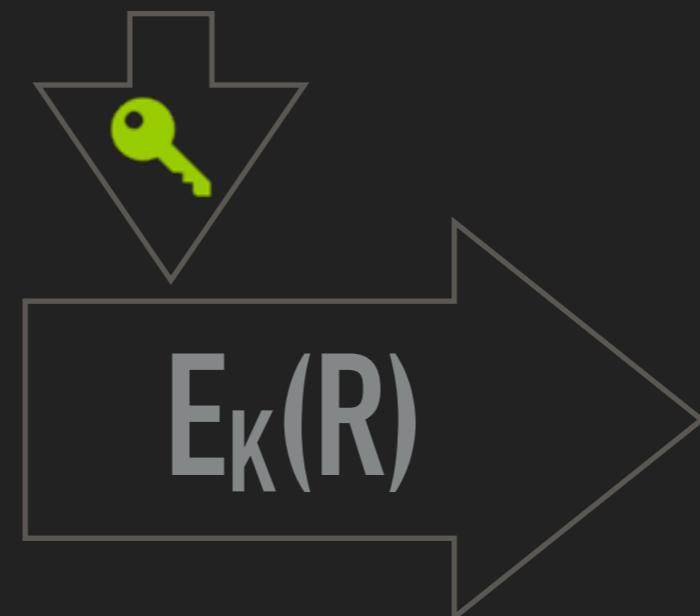
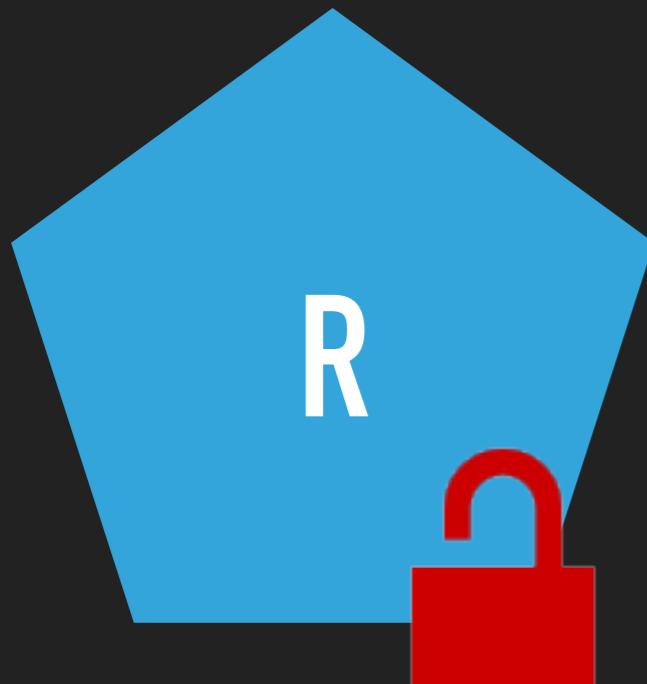
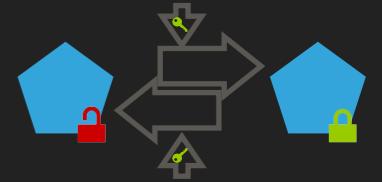
Reading and writing plaintext record 'R' 

SECURE MULTIPLE DATA RECORDS



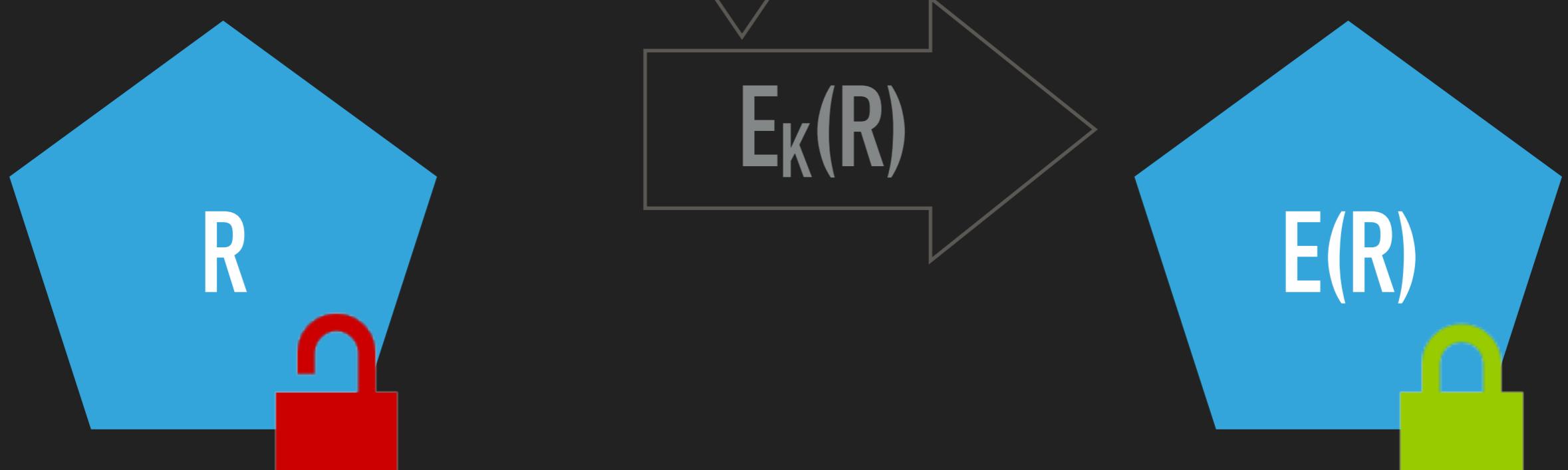
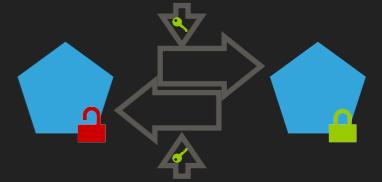
En-/decrypt Data 'R' (record) with key 'K'. 

SECURE MULTIPLE DATA RECORDS



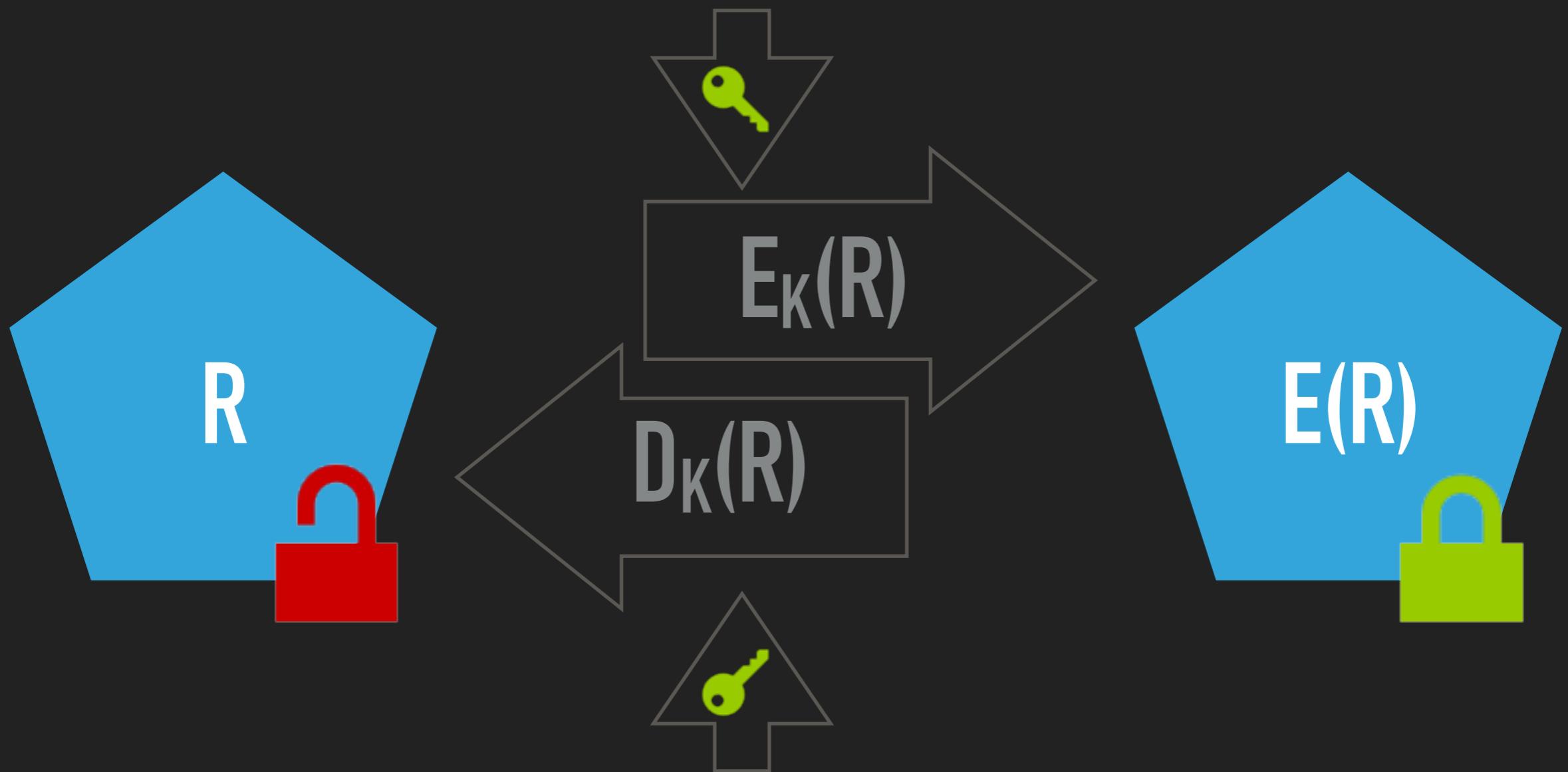
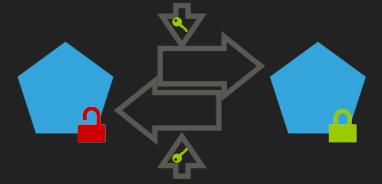
En-/decrypt Data 'R' (record) with key 'K'. 

SECURE MULTIPLE DATA RECORDS



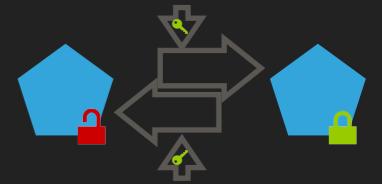
En-/decrypt Data 'R' (record) with key 'K'. 

SECURE MULTIPLE DATA RECORDS

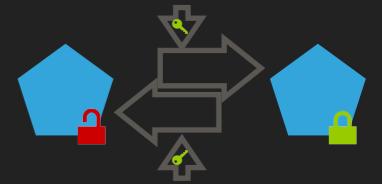


En-/decrypt Data 'R' (record) with key 'K'. 🔑

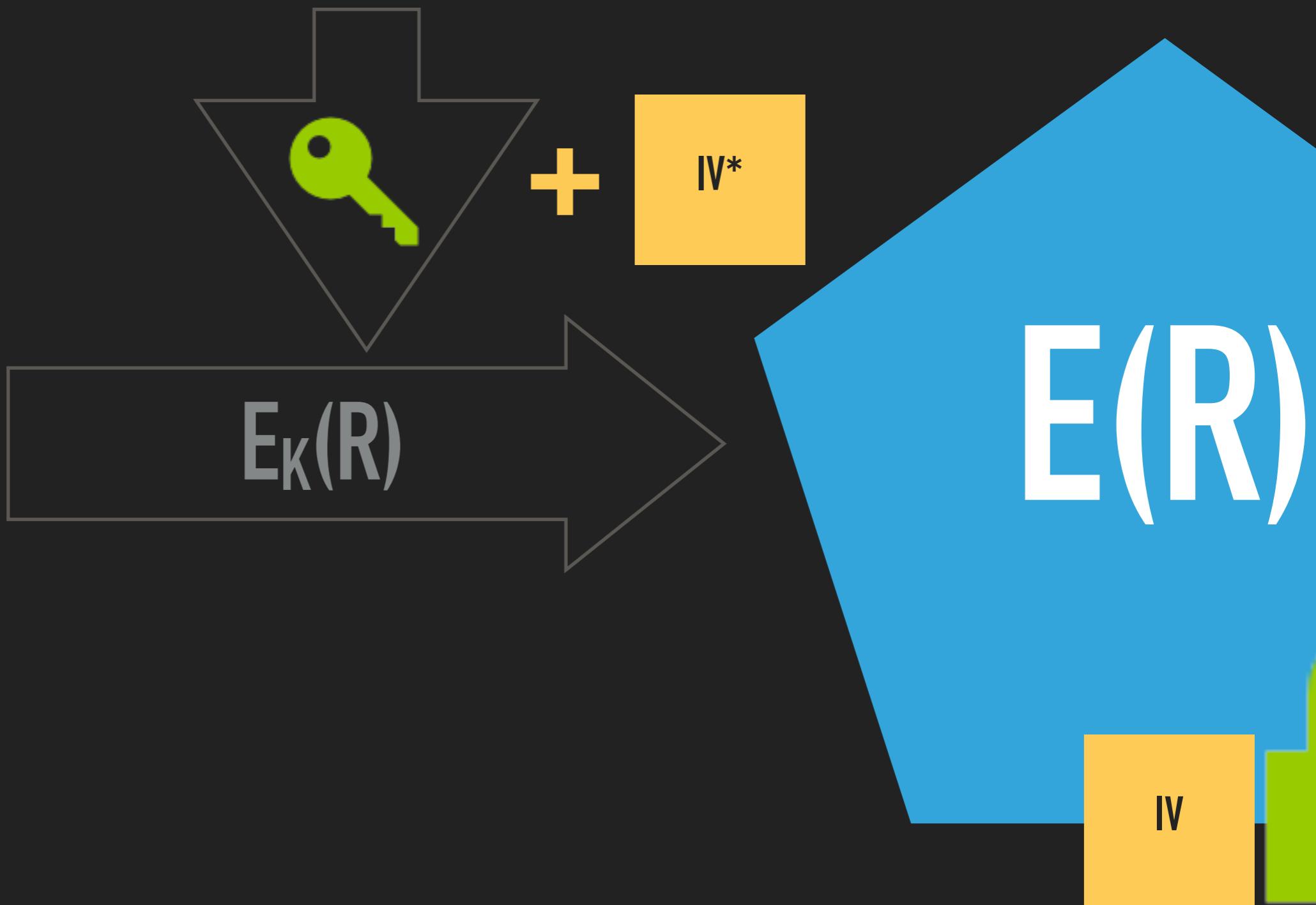
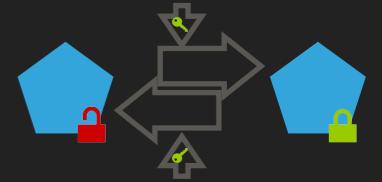
SECURE SMALL KEY(S)

 $E_K(R)$ $E(R)$

SECURE SMALL KEY(S)

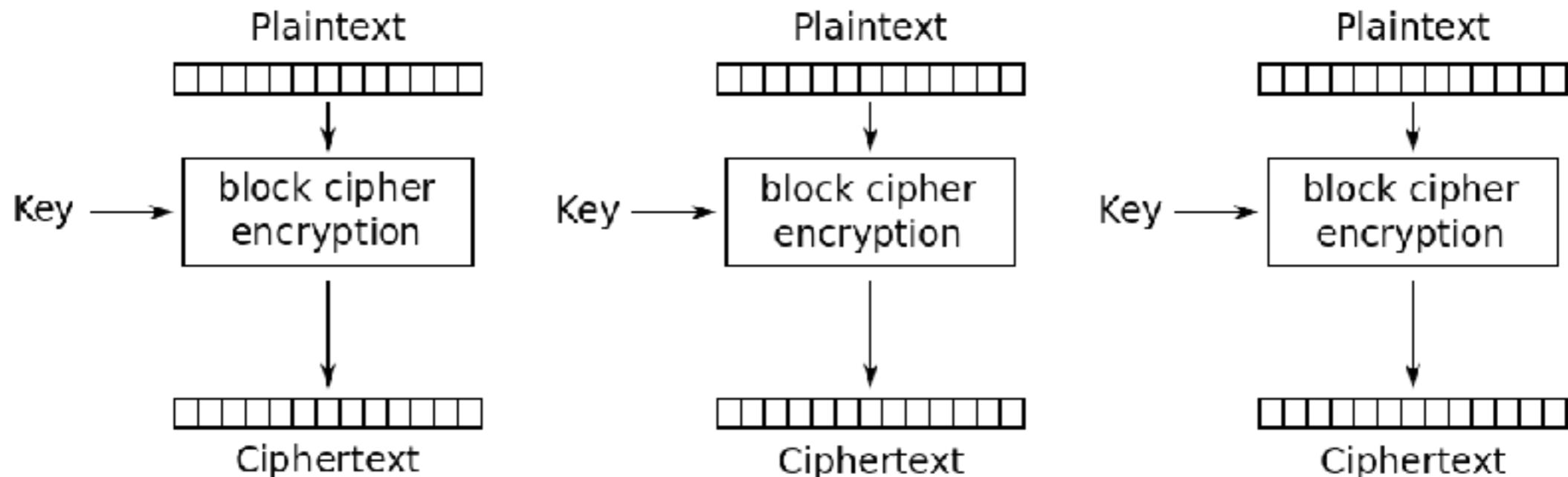


SECURE SMALL KEY(S)

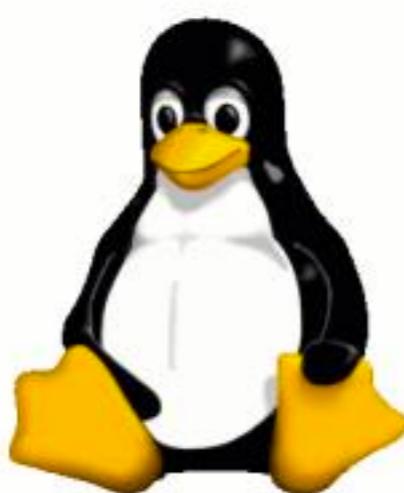


(*) IV - Initialisation Vector. Used in many cryptographic operations. Like a salt. Must be random, might be public.

BLOCK CIPHERS AND THE IV



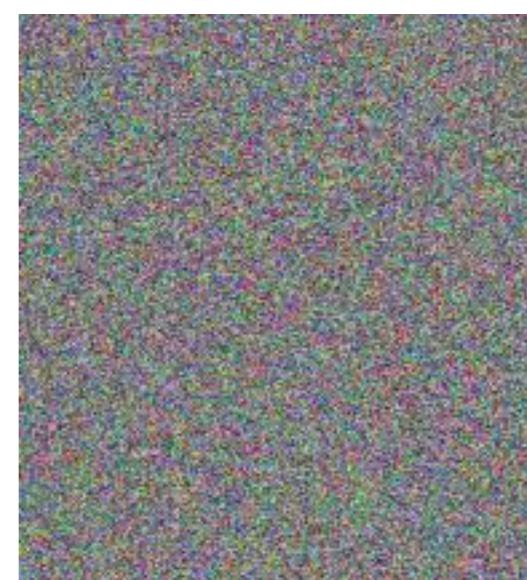
Electronic Codebook (ECB) mode encryption



Original

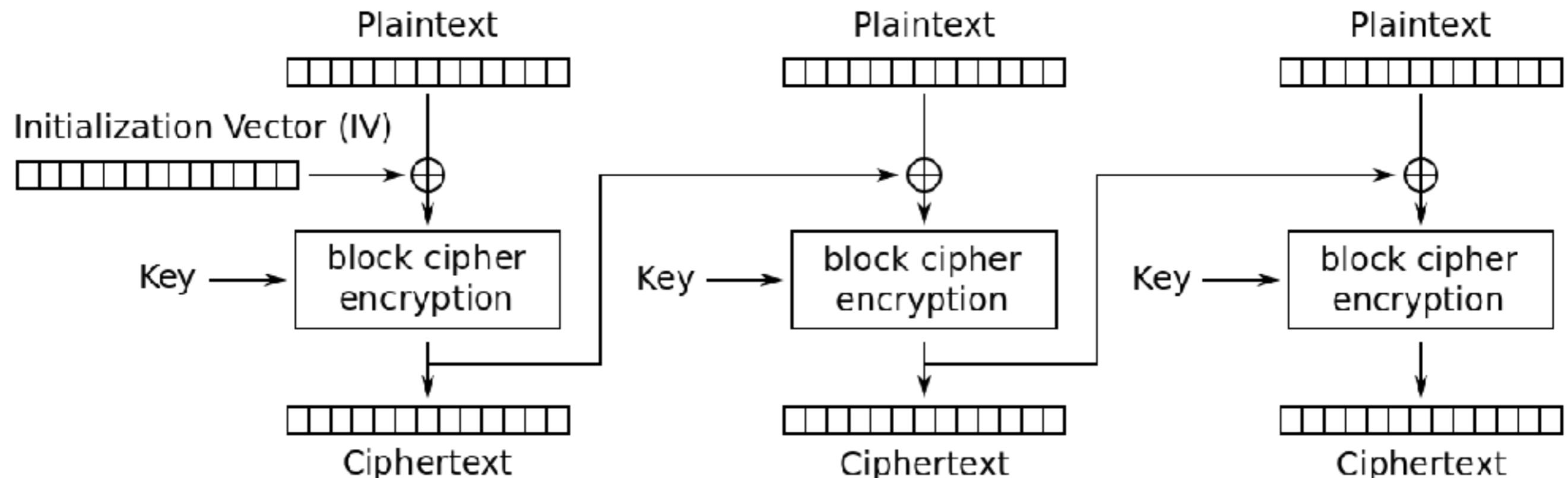


ECB



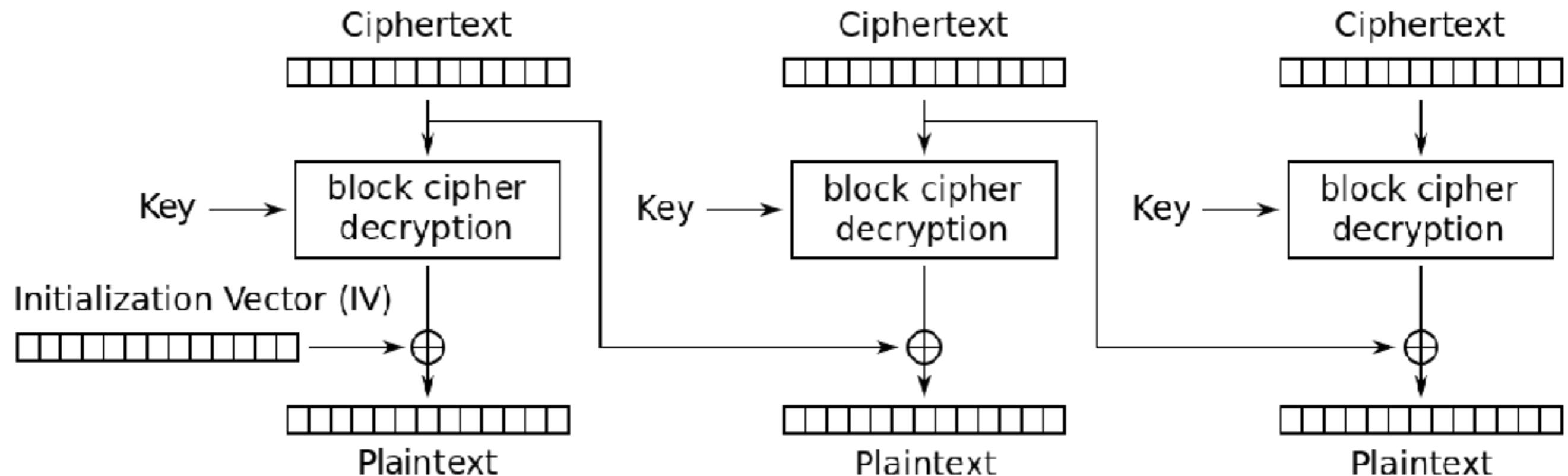
CBC (or other)

BLOCK CIPHERS AND THE IV



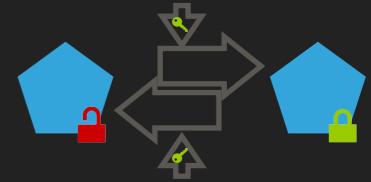
Cipher Block Chaining (CBC) mode encryption

BLOCK CIPHERS AND THE IV



Cipher Block Chaining (CBC) mode decryption

SOLUTIONS FOR SECURING KEY(S)



Master key in different storage

E.g. records in DB, master key on filesystem.

Baseline. Easy. Protects (only) against 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".





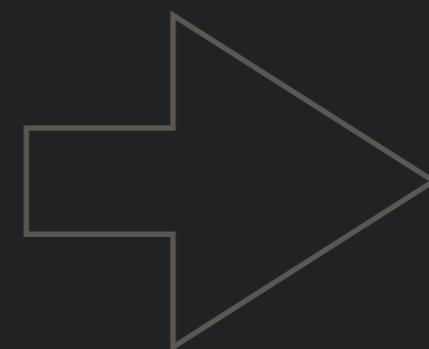
PATTERNS

KEY DERIVATION 1: PASSWORD TO KEY

FROM PASSWORD TO KEY



password

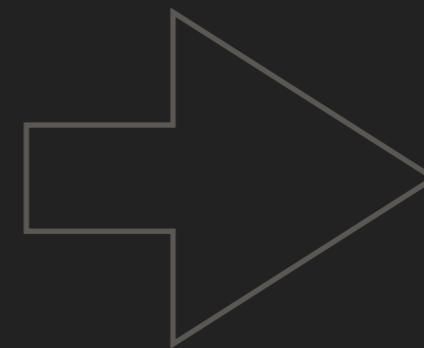


128 bit key

FROM PASSWORD TO KEY



password



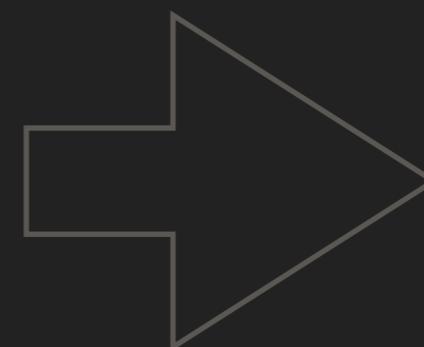
128 bit key

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

FROM PASSWORD TO KEY



password



128 bit key

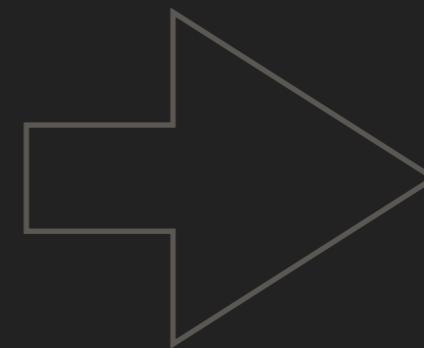
- ▶ A *random* password has ~ 6 bits per character (*)
- ▶ A 128 bit key needs ≥ 21 character passwords

Secure passwords are *very long*

FROM PASSWORD TO KEY



password



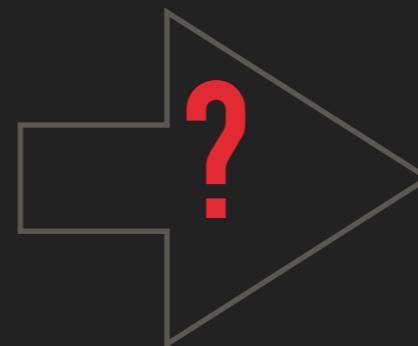
128 bit key

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

FROM PASSWORD TO KEY

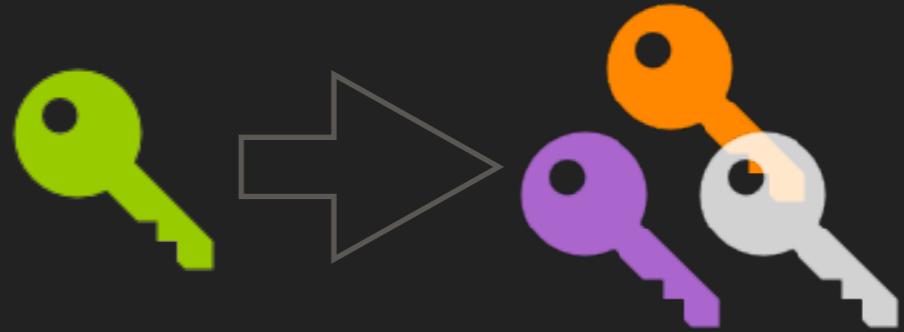


password



128 bit key

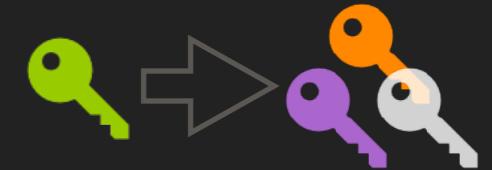
- ▶ Key derivation functions (KDF) convert passwords to keys
- ▶ For good (21+ chars) passwords use HKDF ([RFC5869](#))
- ▶ Else: use a KDF with brute force protection (*)
 - ▶ SCRYPT ([RFC7914](#))
 - ▶ PBKDF2 ([RFC2898](#))



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.

Master Key + $r_1.id + r_1.ver \rightarrow$ Derived Key

Master Key + $r_2.id + r_2.ver \rightarrow$ Derived Key

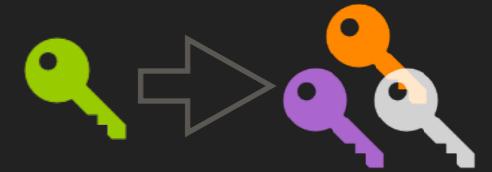
...

Master Key + $r_n.id + r_n.ver \rightarrow$ Derived Key

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.

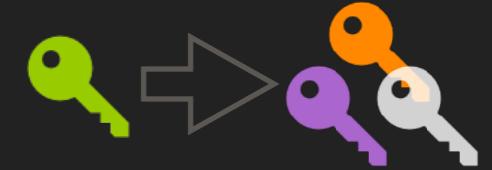
Solution: Use key derivation to derive per-record keys.



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 )
```

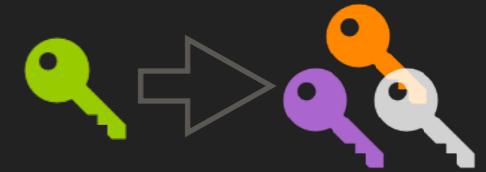
```
byte[ 16 ] derived_iv    = keyAndIV[ 0..15 ]  
byte[ 16 ] derived_key   = keyAndIV[ 16..31 ]
```

- ▶ `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!

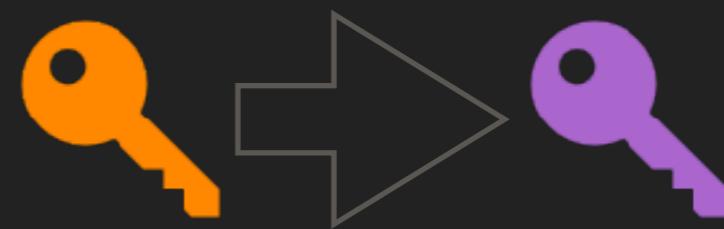


- ▶ Encrypting different data with the same key and IV can lead to complete loss of confidentiality / integrity (*)
- ▶ **When updating encrypted records a new IV must be used** (better: a new key and IV)
- ▶ This can be achieved by incrementing a record-version on each encryption and using it in the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

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



DES BLOWFISH AES

MD5 SHA-1 SHA-256

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

PATTERNS

ALGORITHM ROLLOVER

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

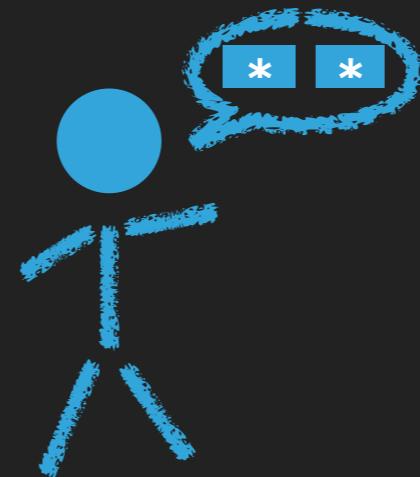
ALGORITHM ROLLOVER

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

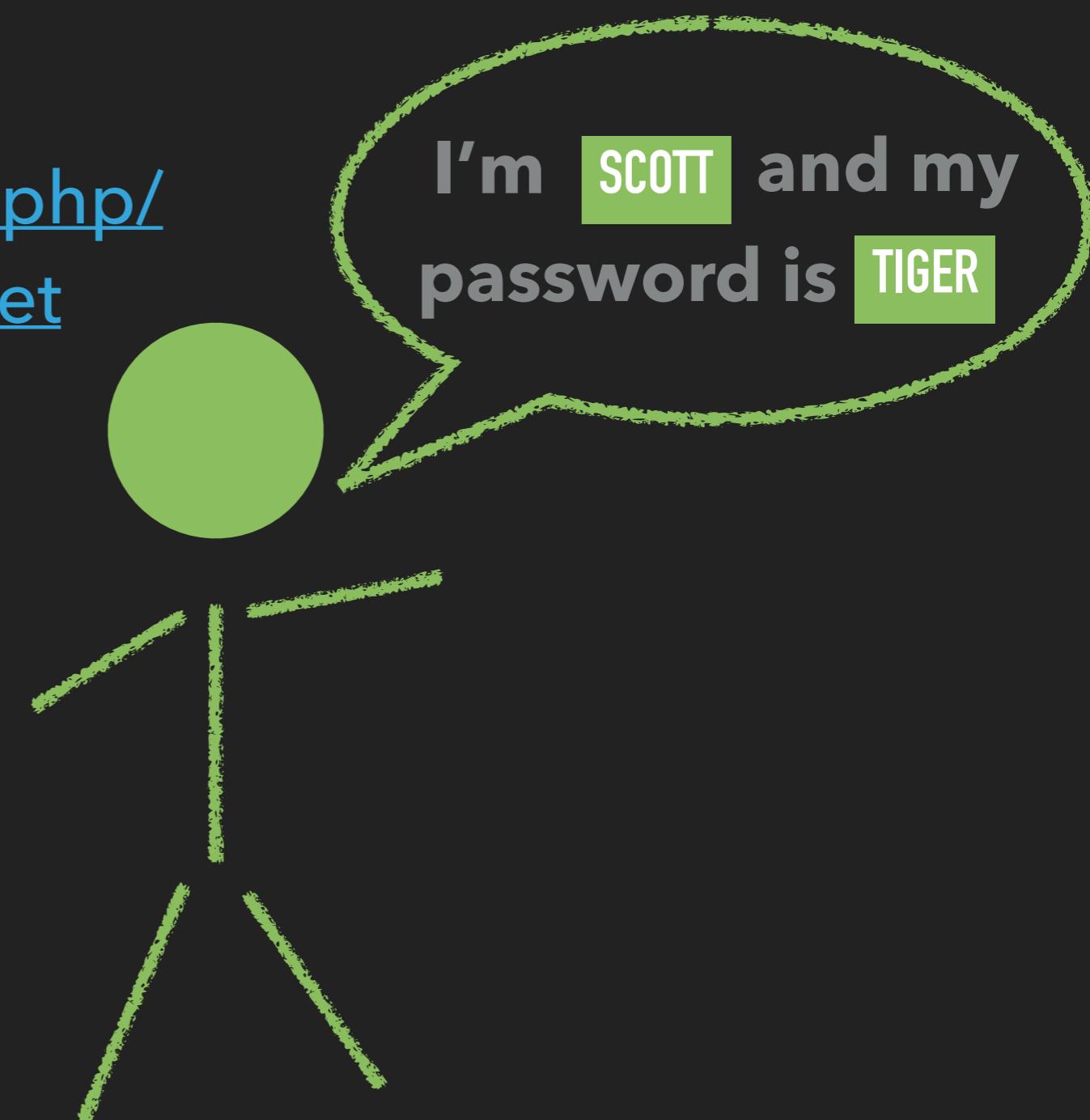
USER LOGIN

Problem: Login a user

Solution: Not in scope here

[https://www.owasp.org/index.php/
Password_Storage_Cheat_Sheet](https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet)

Also: OAUTH, Kerberos, ...

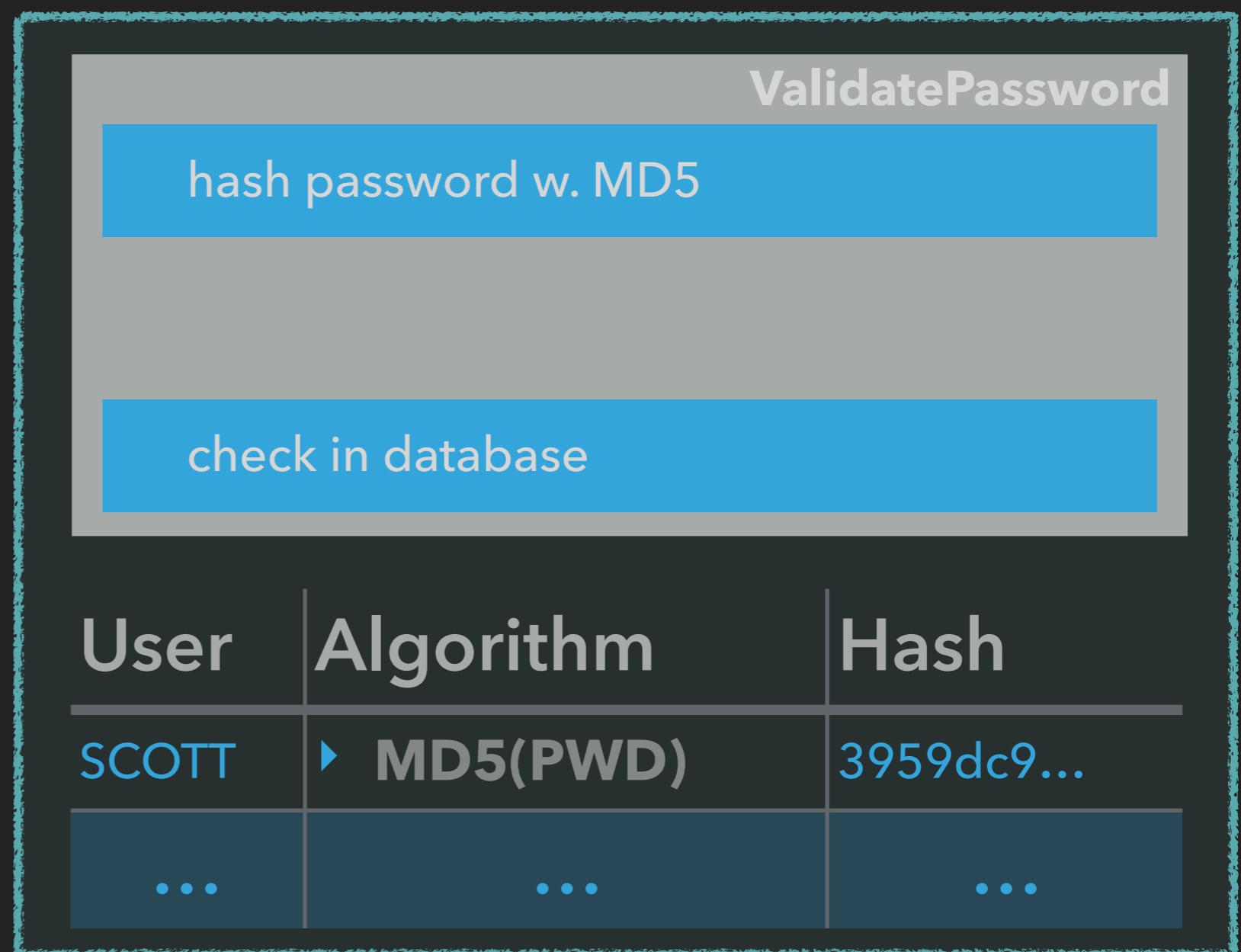
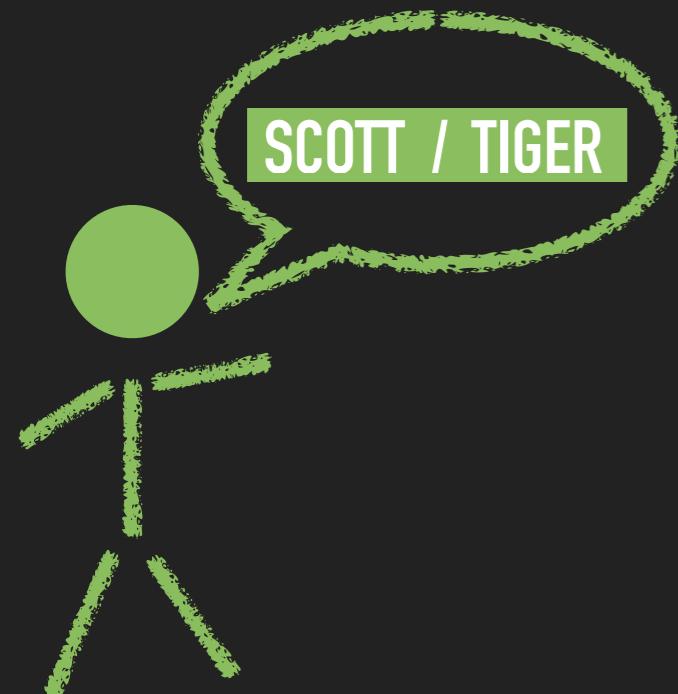


USER LOGIN: MIGRATE PASSWORDS



Problem: Migrate password hashes to new algorithms

Solution: Chain hashing functions

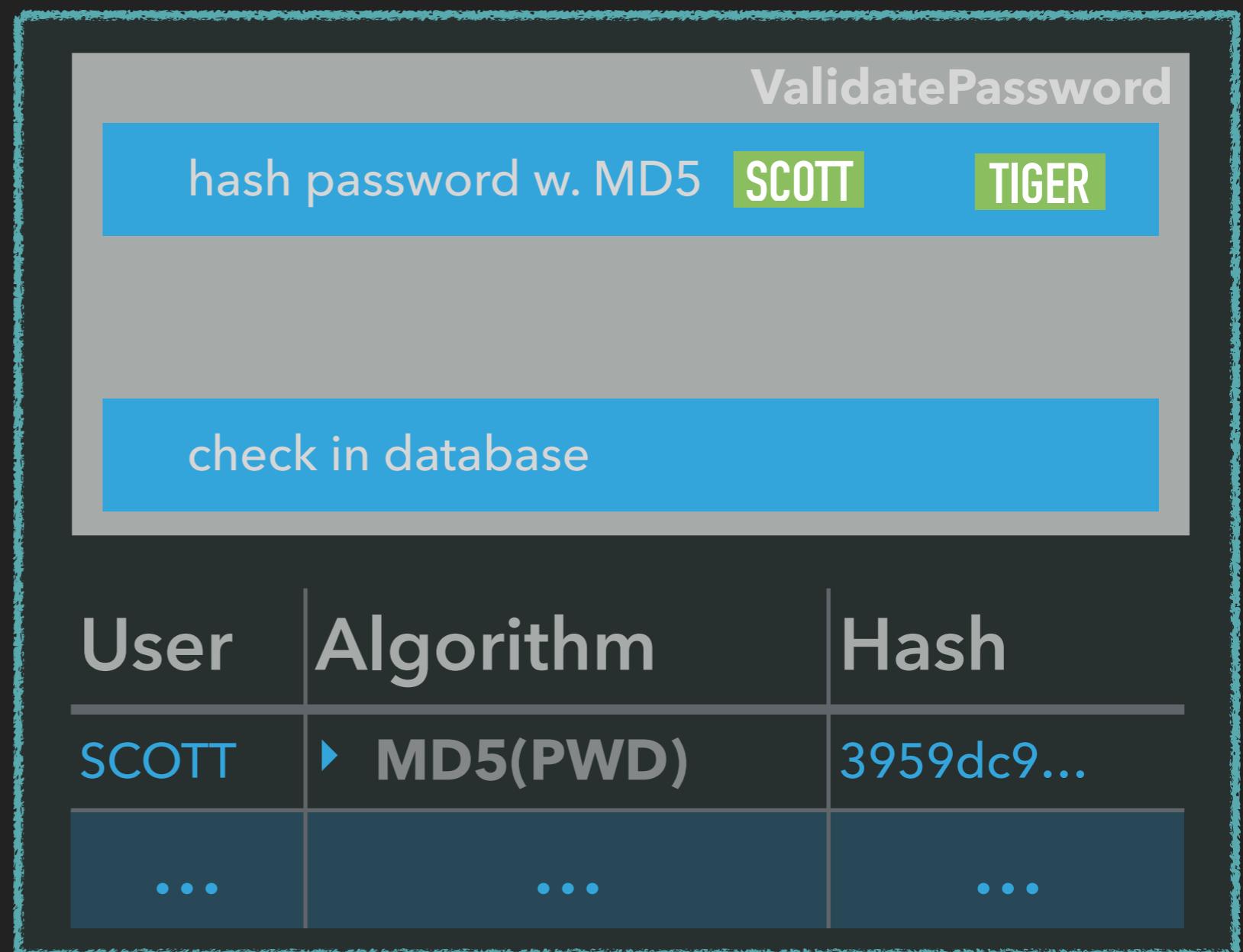
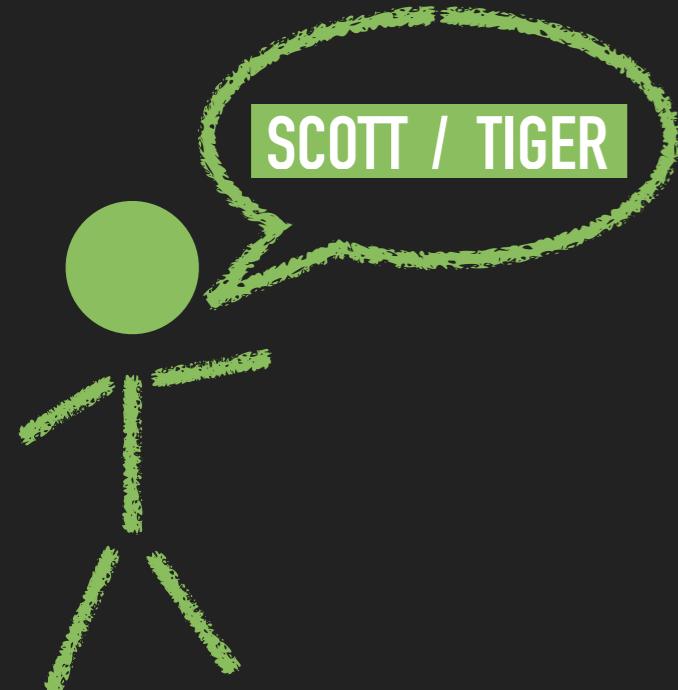


USER LOGIN: MIGRATE PASSWORDS



Problem: Migrate password hashes to new algorithms

Solution: Chain hashing functions

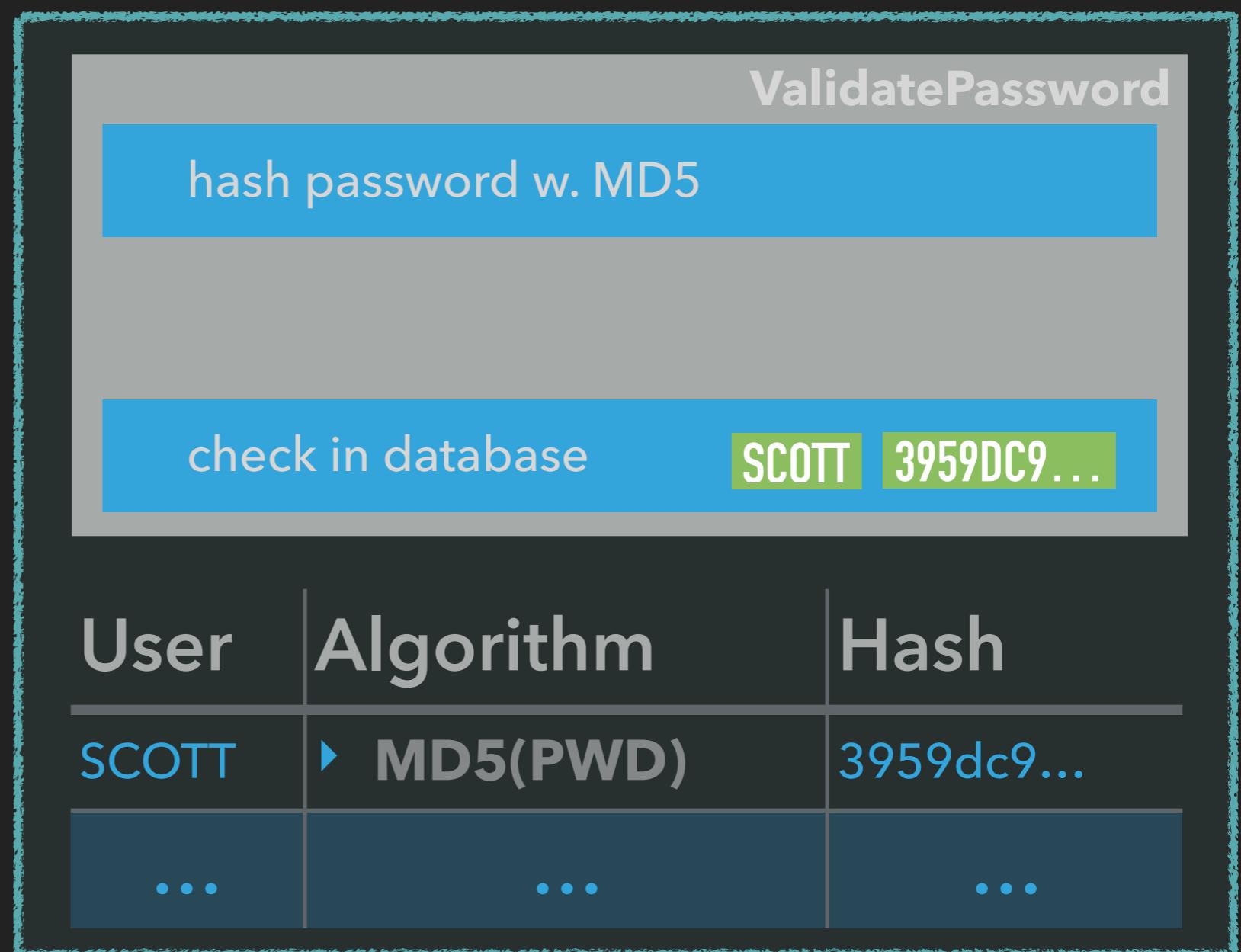
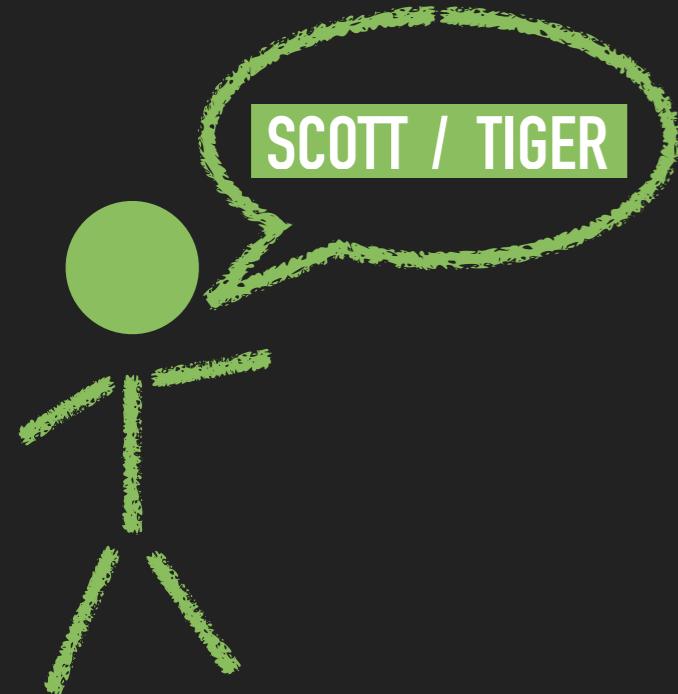


USER LOGIN: MIGRATE PASSWORDS



Problem: Migrate password hashes to new algorithms

Solution: Chain hashing functions

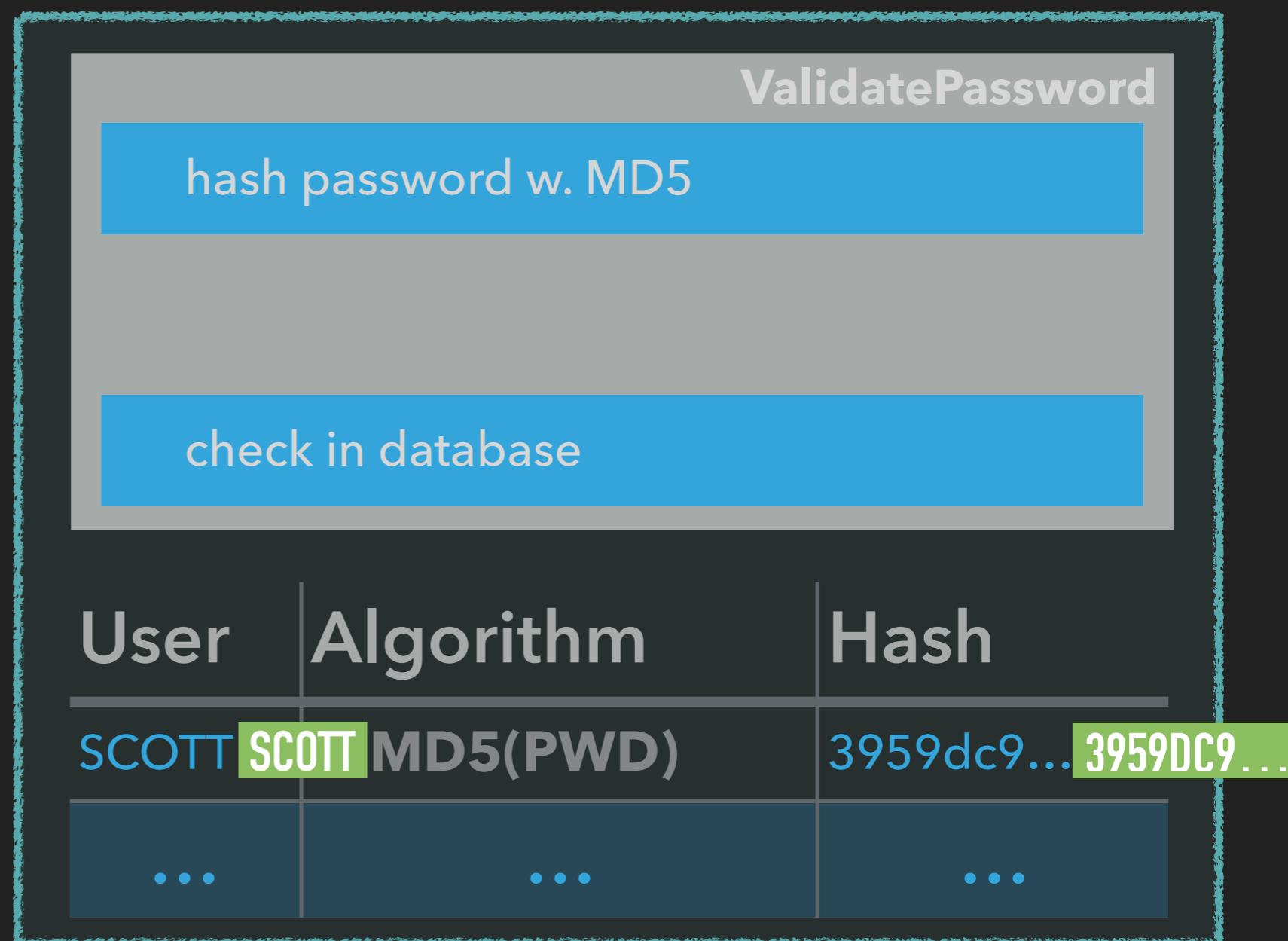
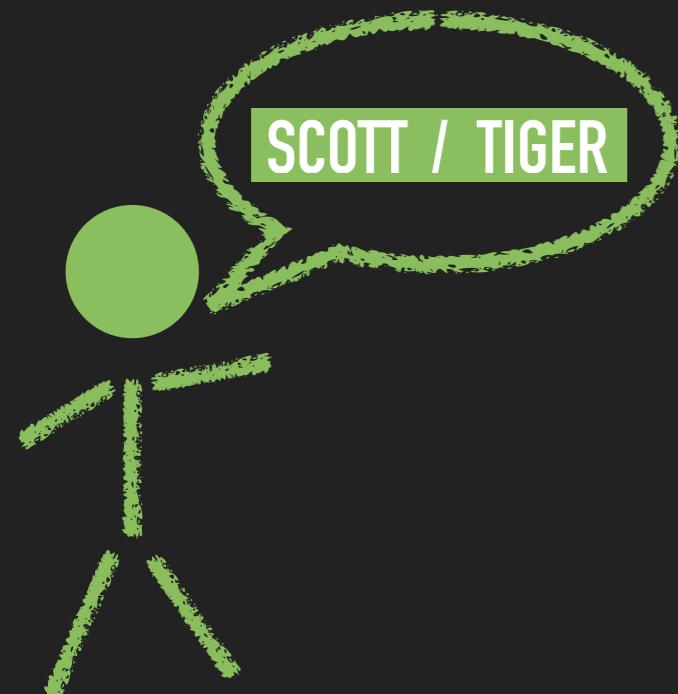


USER LOGIN: MIGRATE PASSWORDS



Problem: Migrate password hashes to new algorithms

Solution: Chain hashing functions

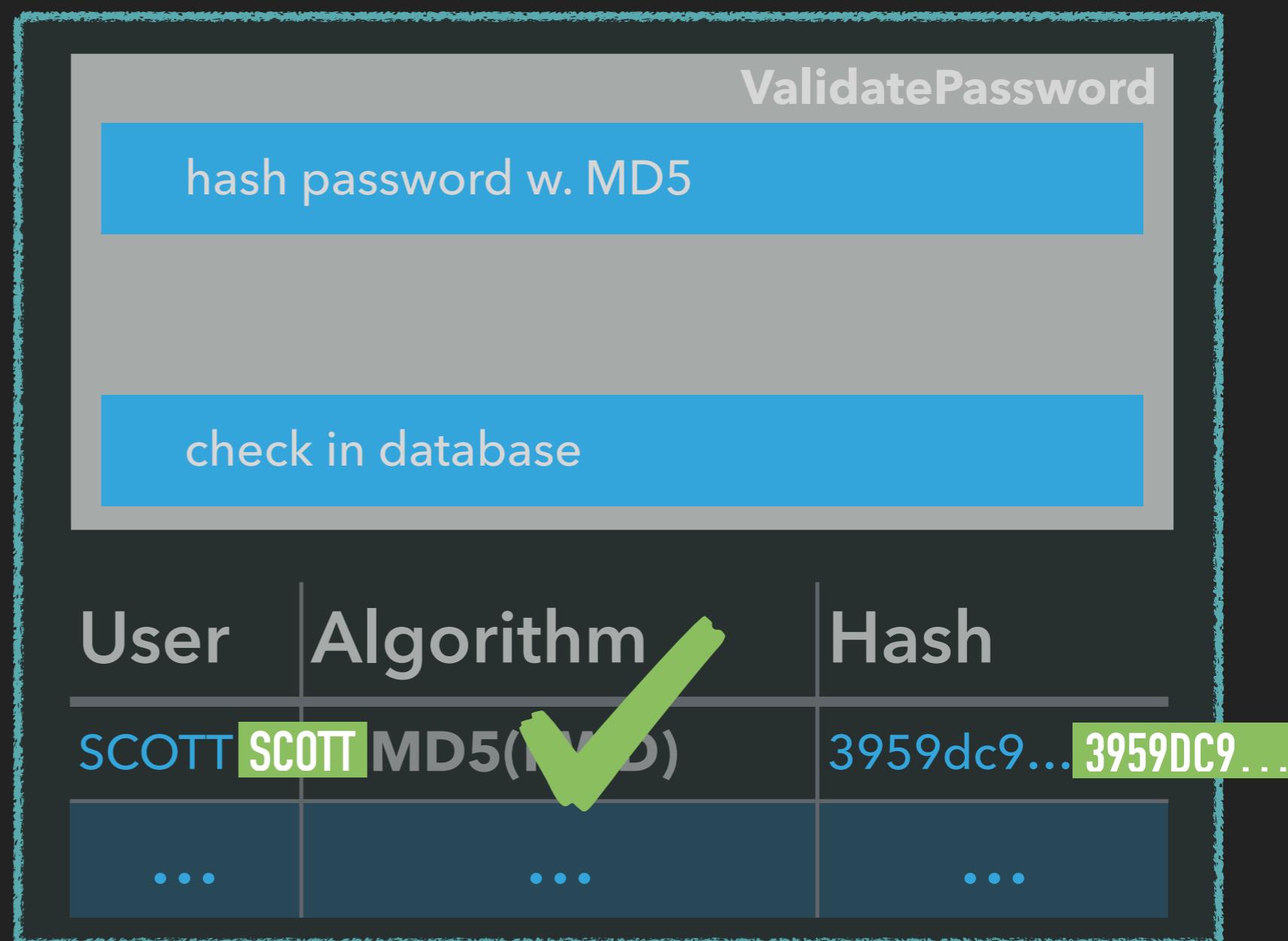
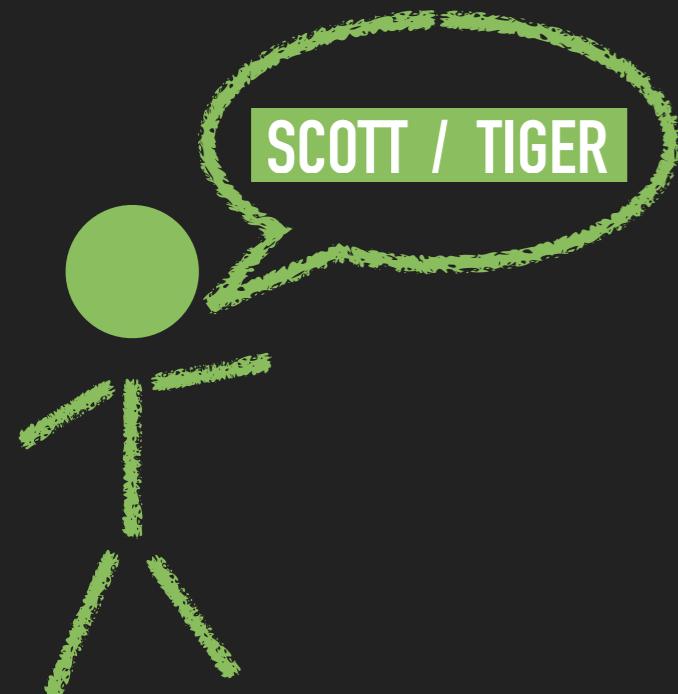


USER LOGIN: MIGRATE PASSWORDS



Problem: Migrate password hashes to new algorithms

Solution: Chain hashing functions

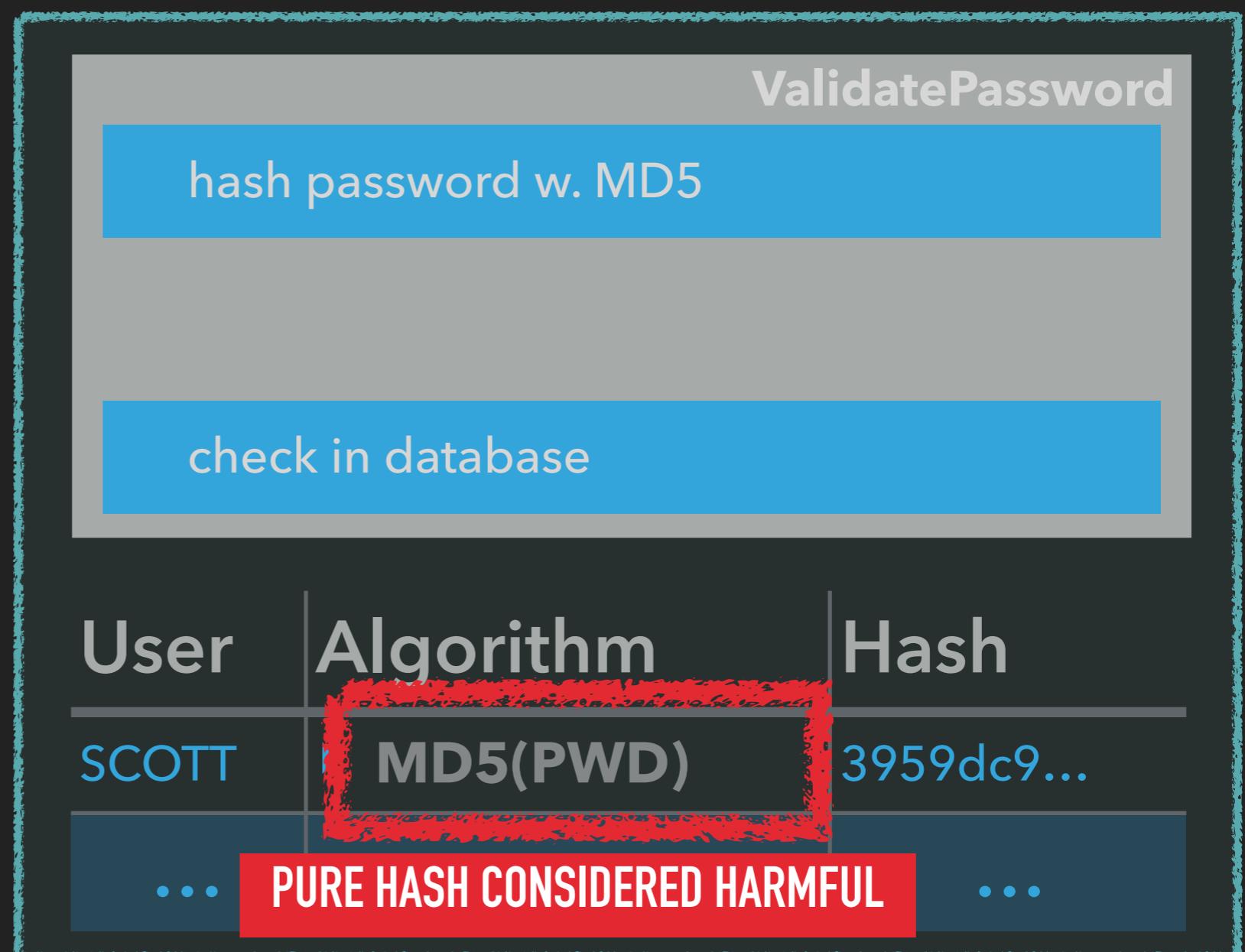
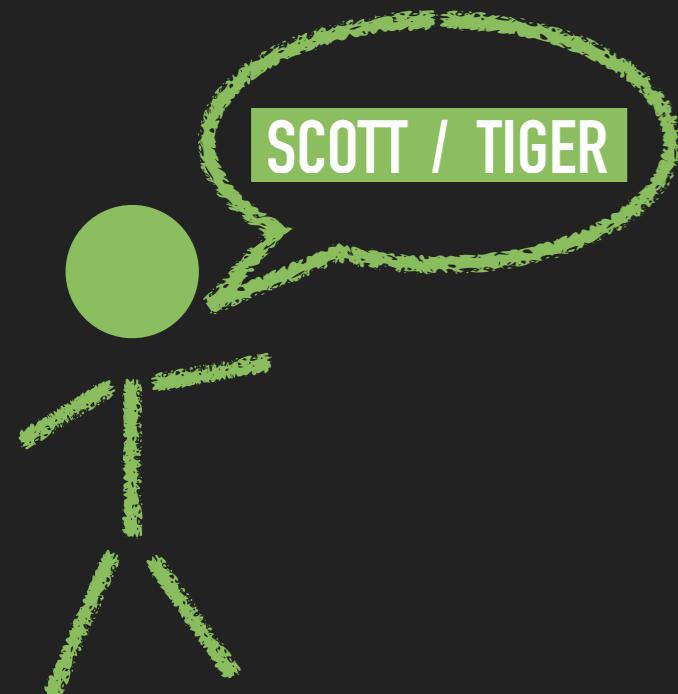


USER LOGIN: MIGRATE PASSWORDS



Problem: Migrate password hashes to new algorithms

Solution: Chain hashing functions



User	Algorithm	Hash
SCOTT	MD5(MD5)	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

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

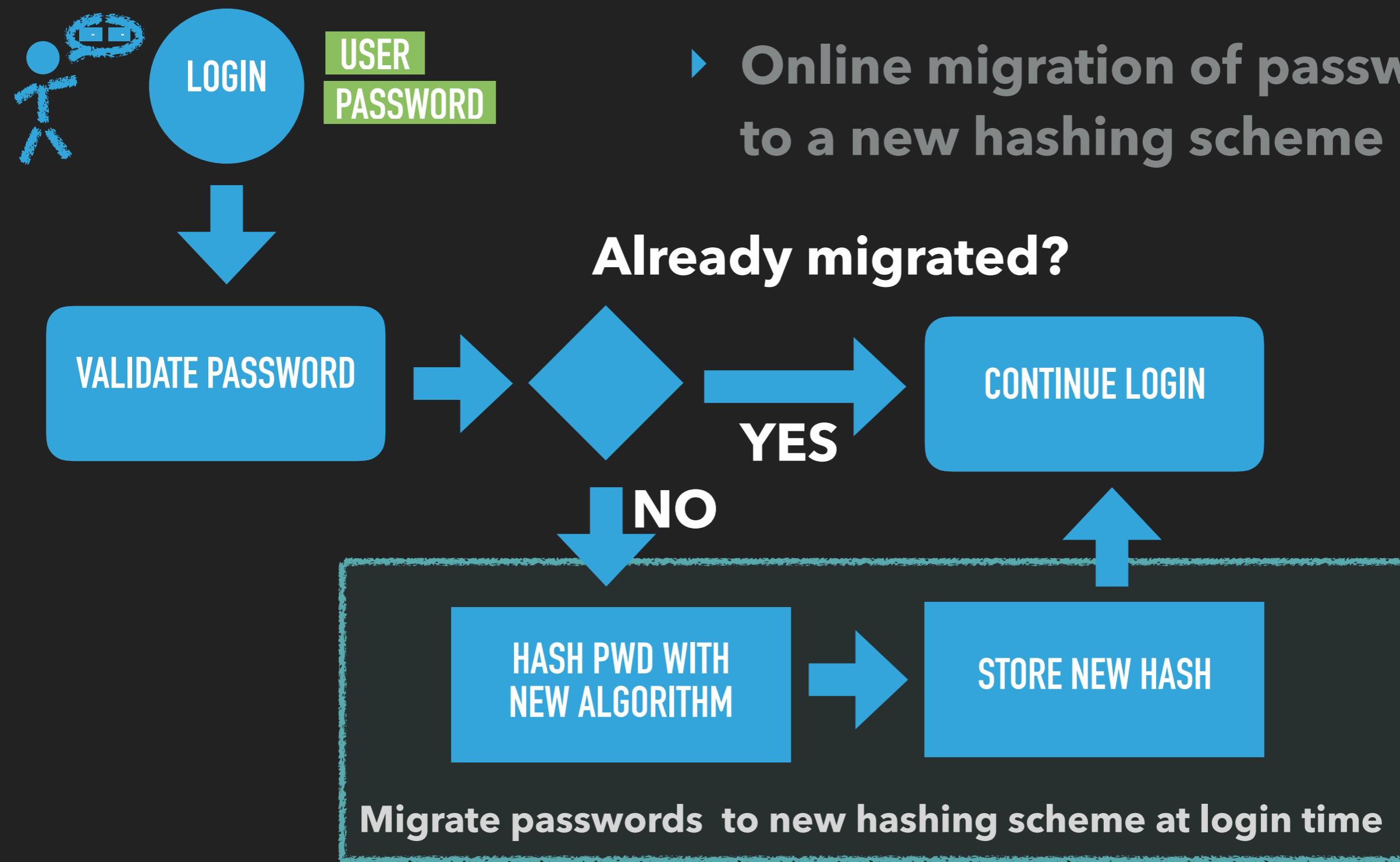
Even consumer grade graphic cards calculates giga-hashes (2^{30}) per second.

HASHES - EVEN SALTED - OFFER NO PROTECTION

SWITCH TO BRUTE-FORCE PROOF SCHEMES

- <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

USER LOGIN: MIGRATE PASSWORDS



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)

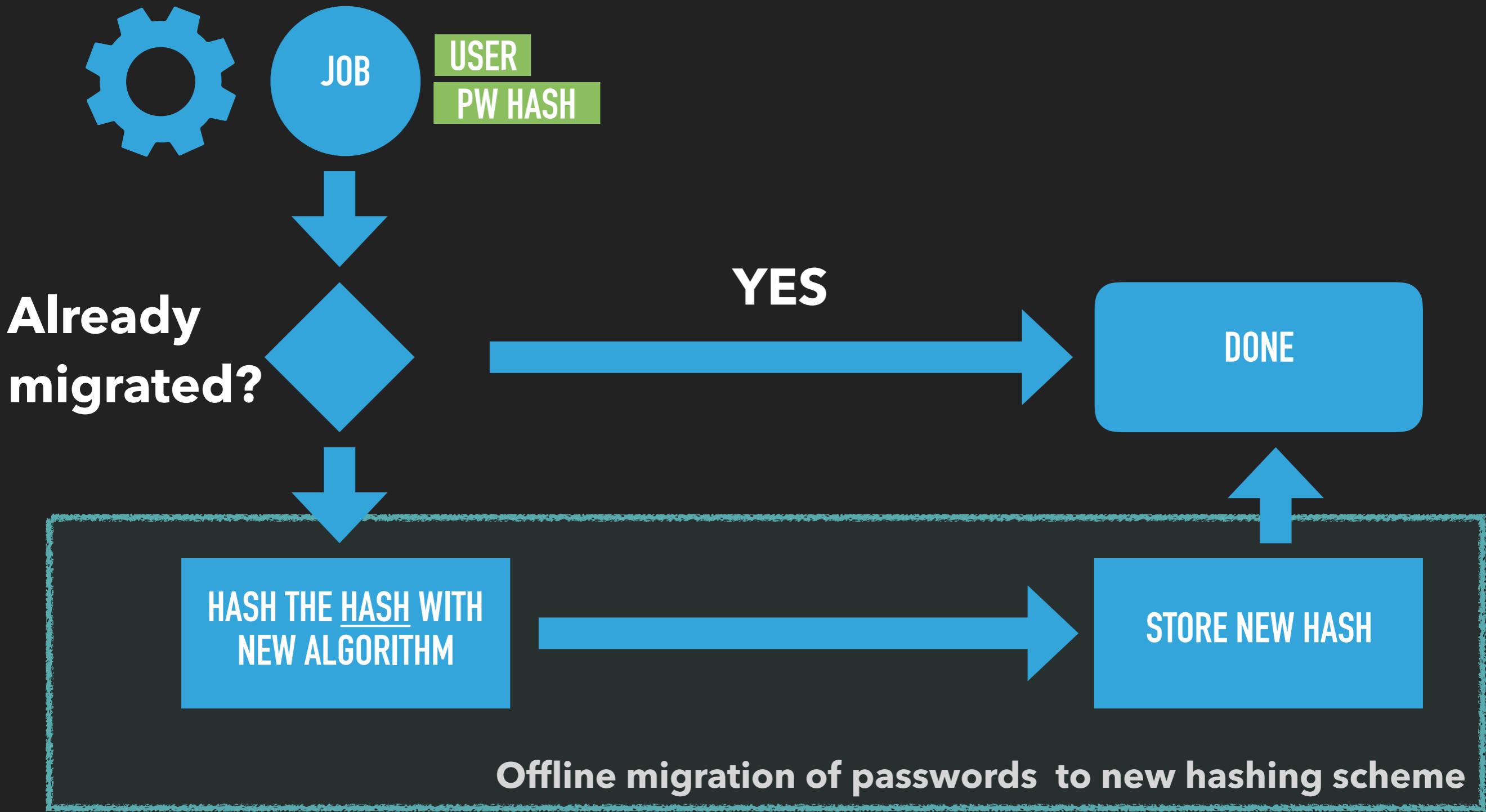
USER LOGIN: MIGRATE PASSWORDS



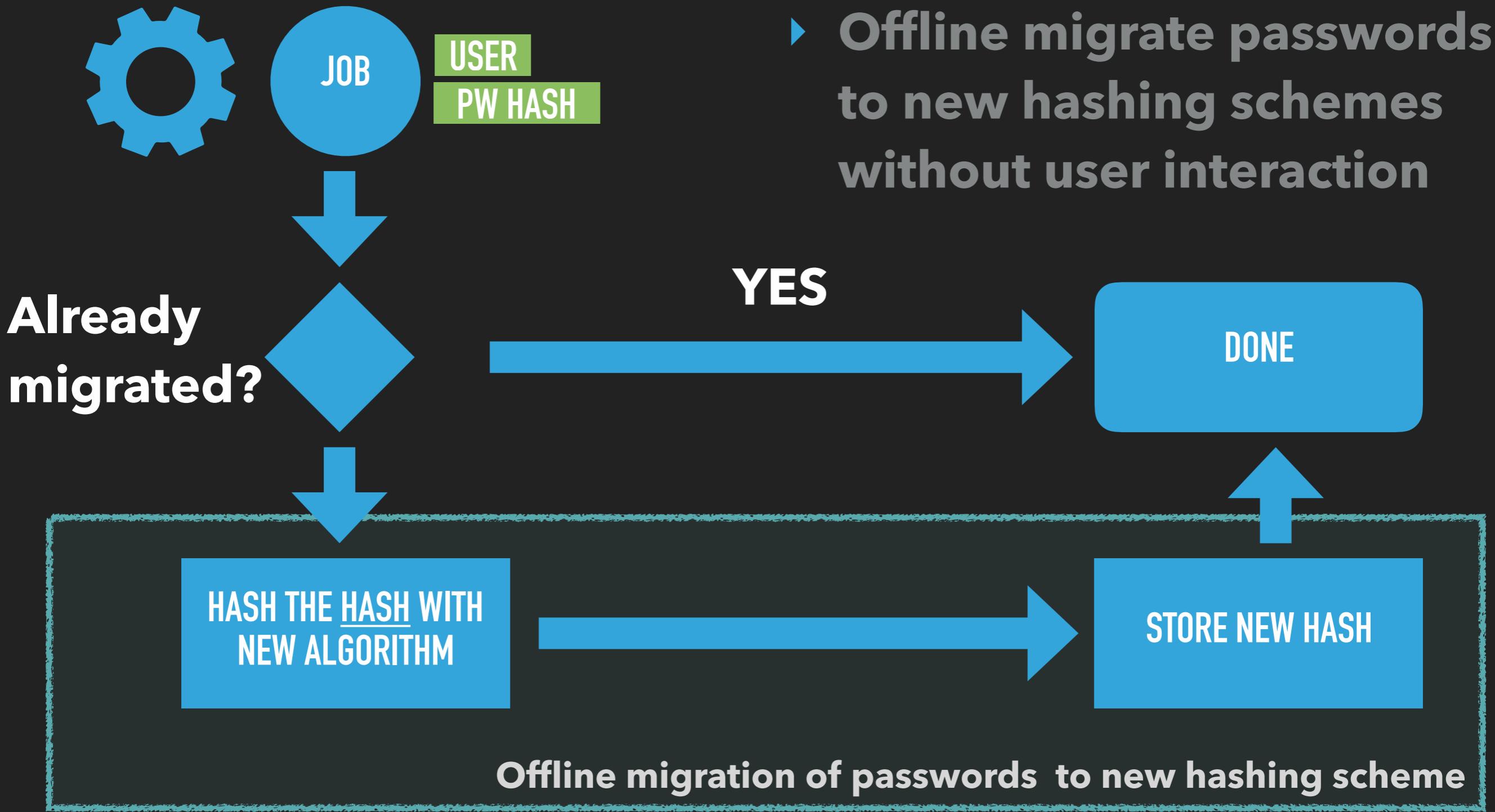
User	Algorithm	Hash	Last Login
SCOTT	PBKDF2(PWD)	3959dc9...	Now
PETER	MD5(PWD)	...	2 years ago
...	MD5(PWD)	...	4 months ago
...	MD5(PWD)
...	MD5(PWD)
...	MD5(PWD)

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

USER LOGIN: MIGRATE PASSWORDS



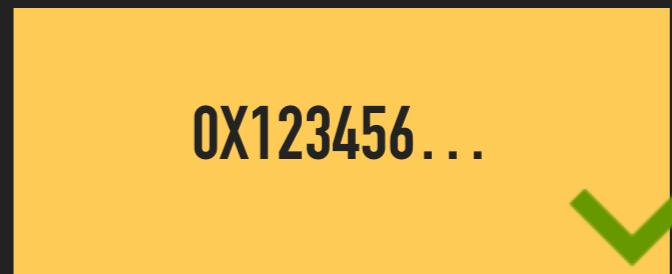
USER LOGIN: MIGRATE PASSWORDS



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 €	...
...	EVE GETS AN INSTANT PROMOTION	...

DATA INTEGRITY & ASSOCIATION

0x123456... ✓

Problem: Sensitive data can be manipulated.

Solution: Use cryptographic checksums with a secret.

User	Salary	Checksum*
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	Checksum*
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	Checksum*
Alice	3,141€	0x4711...
Eve	2,718 €	0xabcd...
...

DATA INTEGRITY & ASSOCIATION

0X123456... ✓

Problem: Protected data can be “replayed”.

User	Salary	Checksum*
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	Checksum*
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	Checksum*
Alice	3,141€	0xabcd...
Eve	3,141€	0xabcd...

```
graph LR; AliceUser[User Alice] -- "green arrow" --> AliceSalary[Salary 3,141€]; AliceUser -- "blue arrow" --> AliceChecksum["Checksum* 0xabcd..."]; EveUser[User Eve] -- "red double arrow" --> EveSalary[Salary 3,141€]; EveUser -- "red double arrow" --> EveChecksum["Checksum* 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.

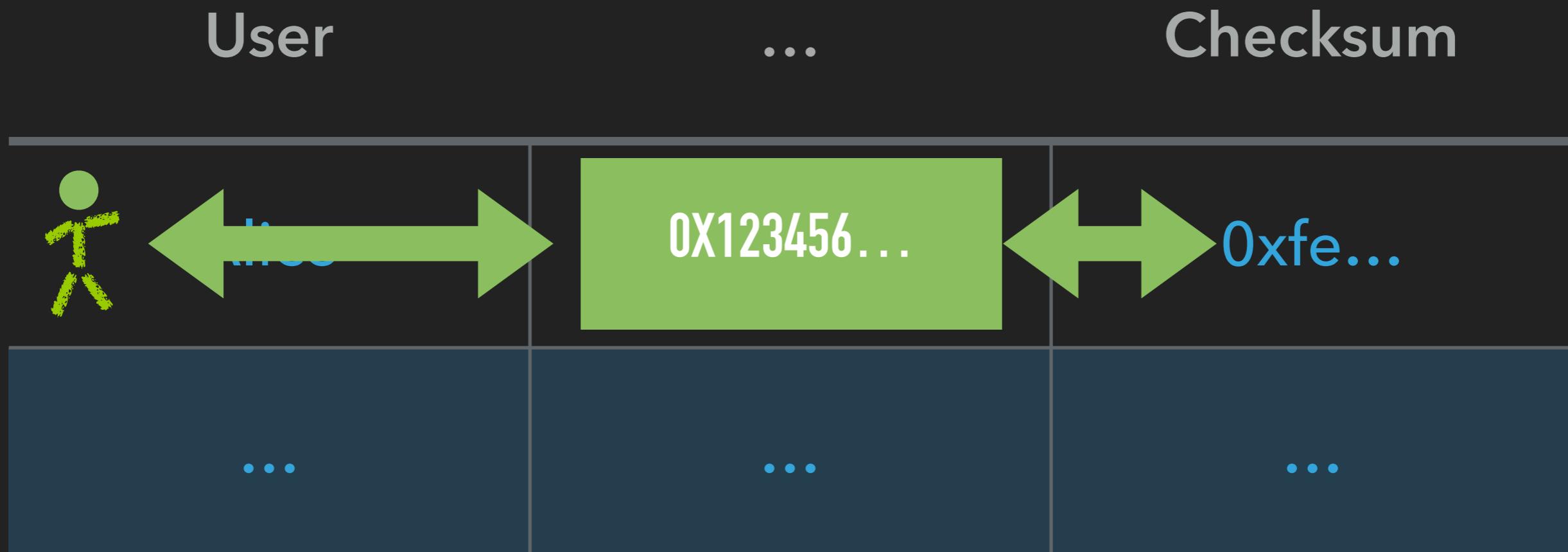
User	Password Hash *	...
Alice	OXABCDEF...	...
Eve	OXABCDEF...	...

INTEGRITY PROTECTION BINDS USER TO SECRET

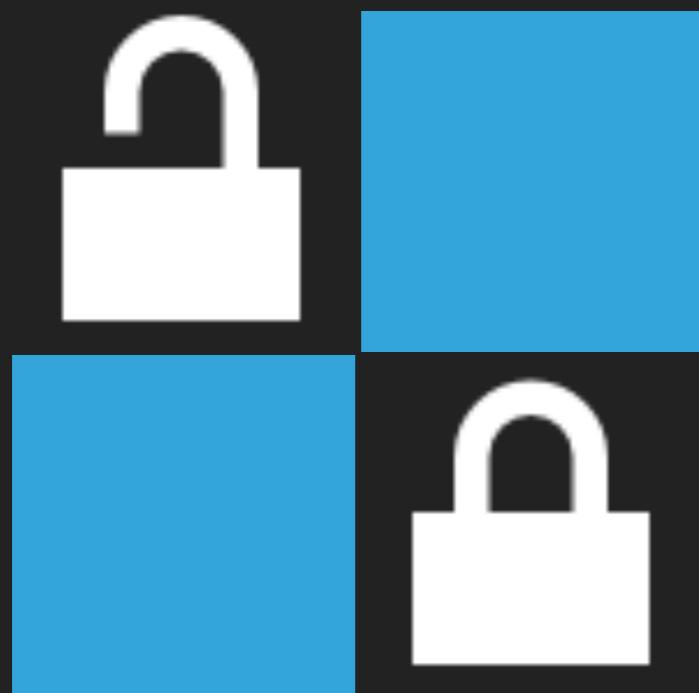
* including the username in the hash/hmac

DATA INTEGRITY & ASSOCIATION

0X123456... ✓



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



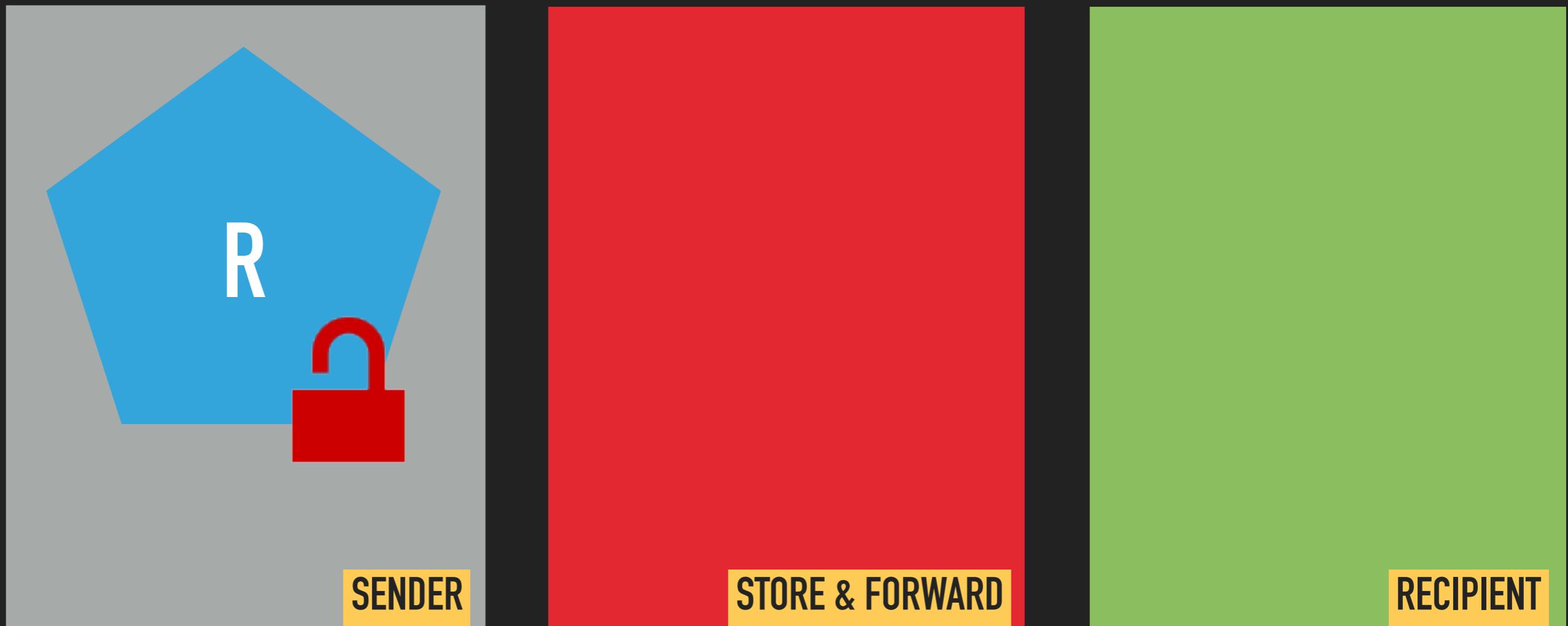
PATTERNS

MOVING

MOVE DATA BETWEEN PARTIES

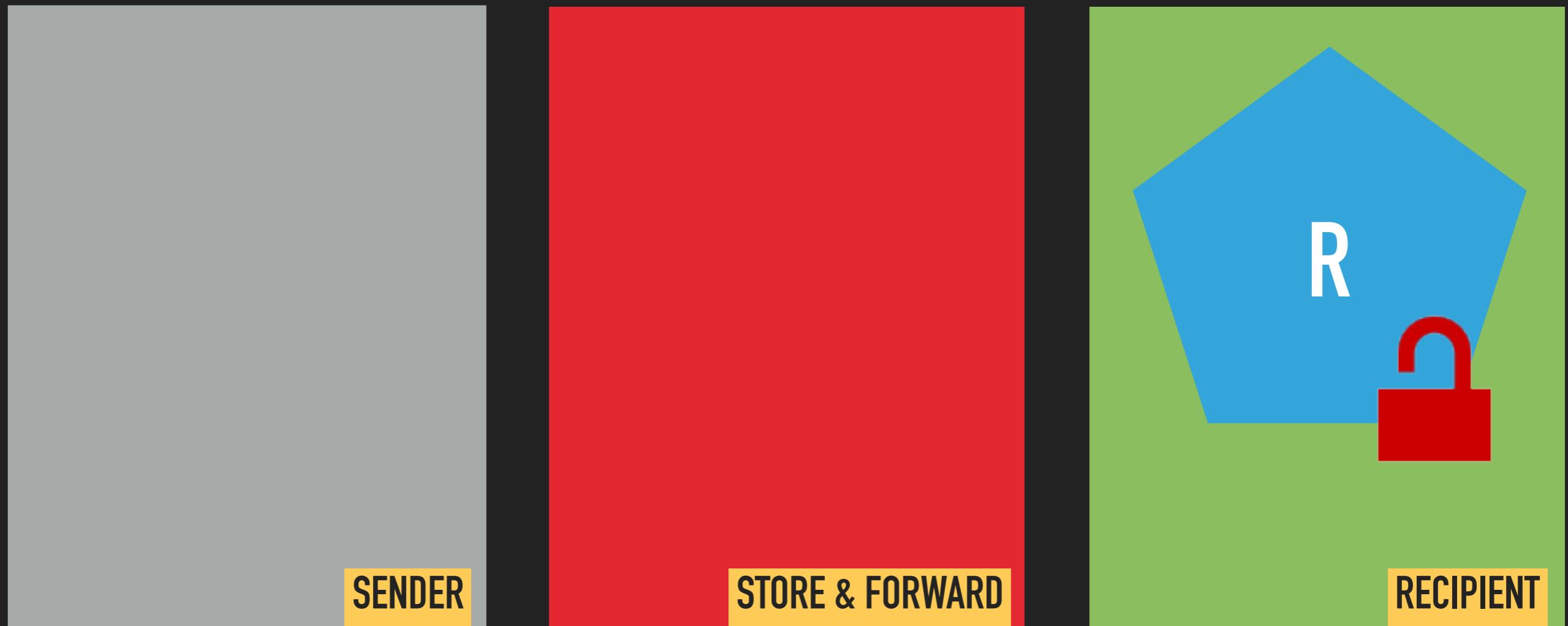


Problem: Data is exchanged between parties.



MOVE DATA BETWEEN PARTIES

Problem: Data is exchanged between parties.

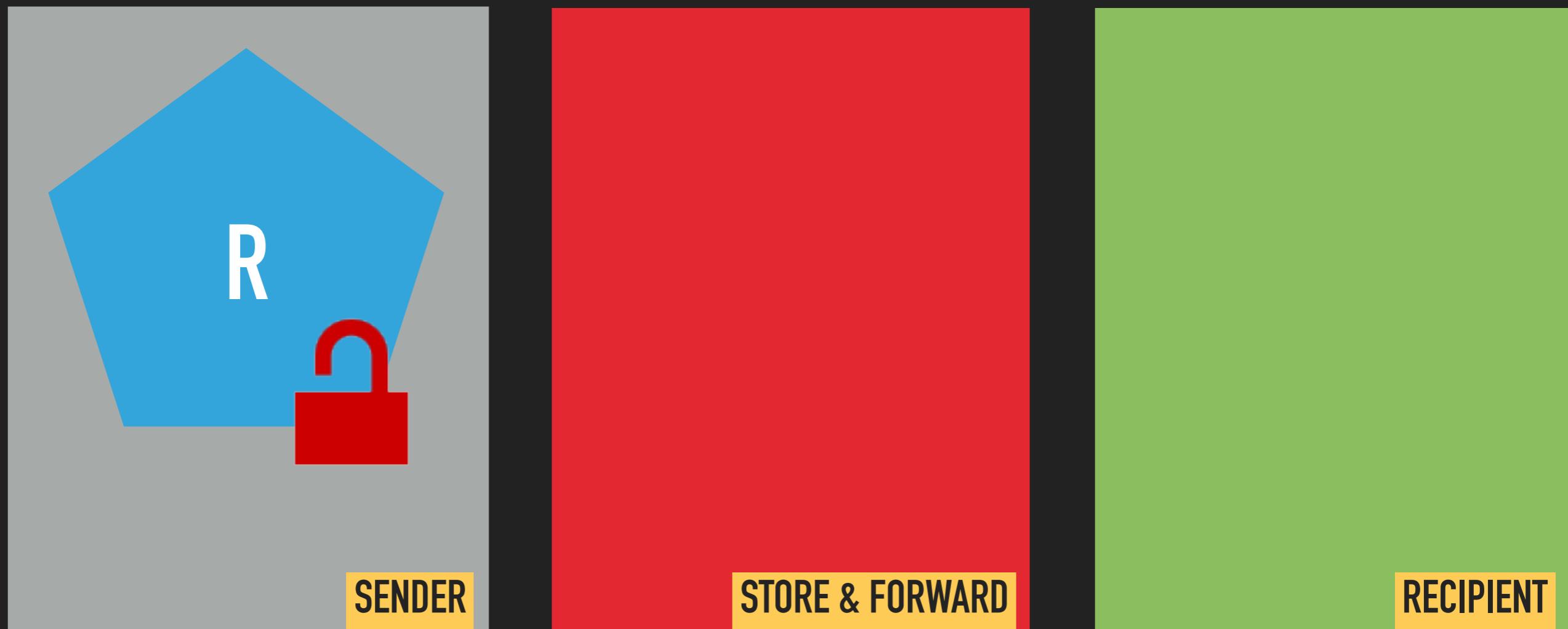


MOVE DATA BETWEEN PARTIES



Problem: Data is exchanged between parties.

Solution: Use public key cryptography to protect data.



Encrypt with public key

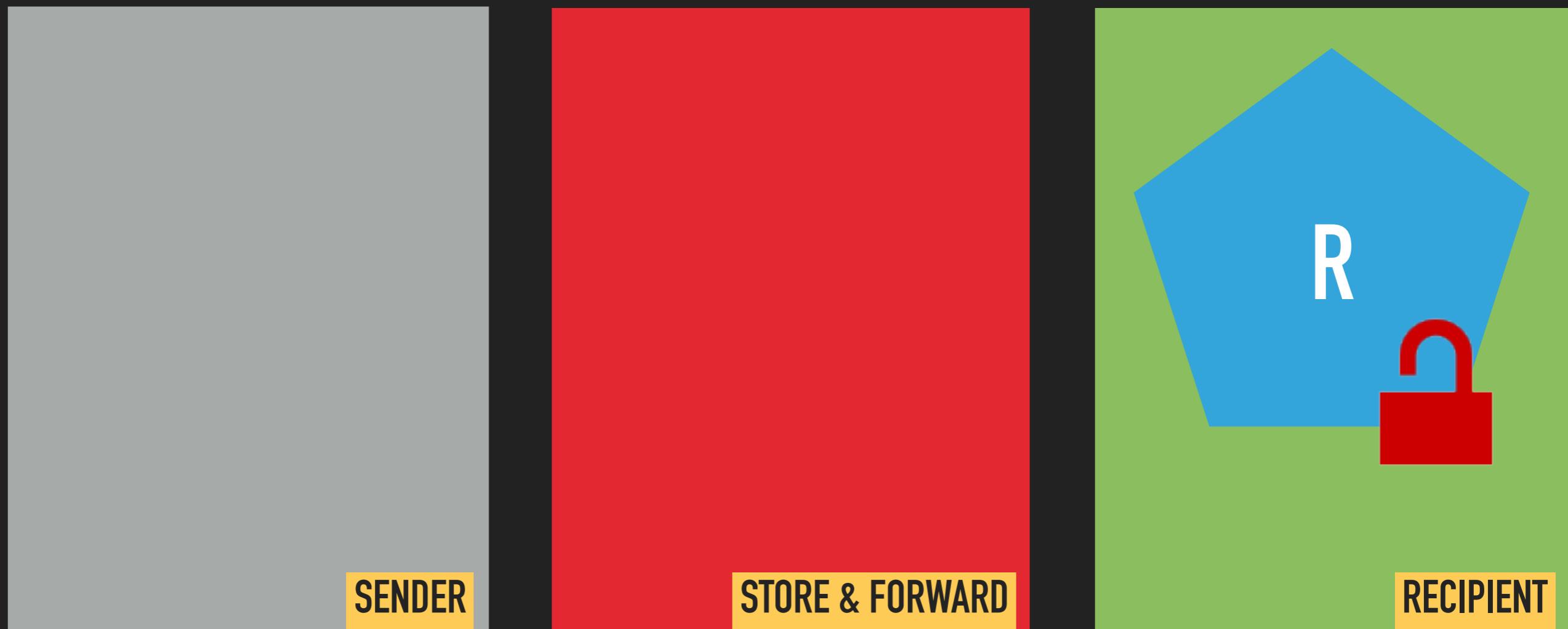
Decrypt with private key

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

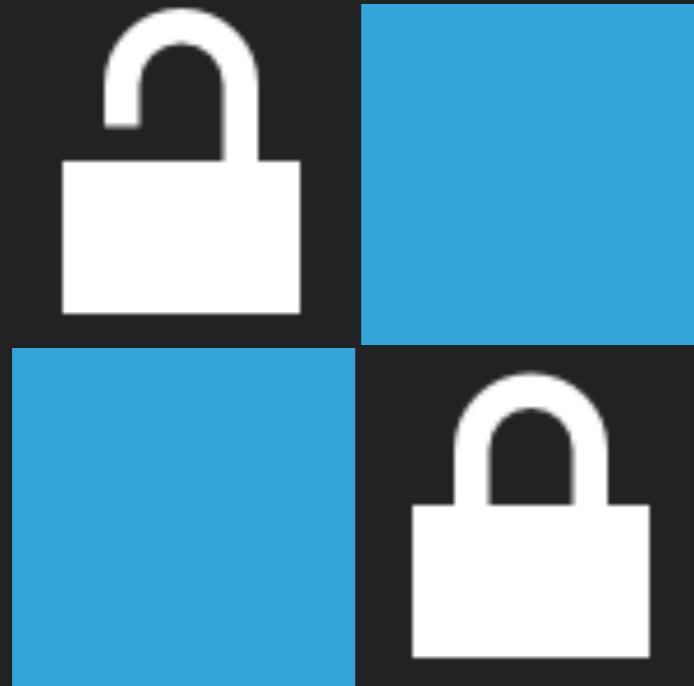
MOVE DATA BETWEEN PARTIES



```
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);
}
```



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

<https://xkcd.com/221/>

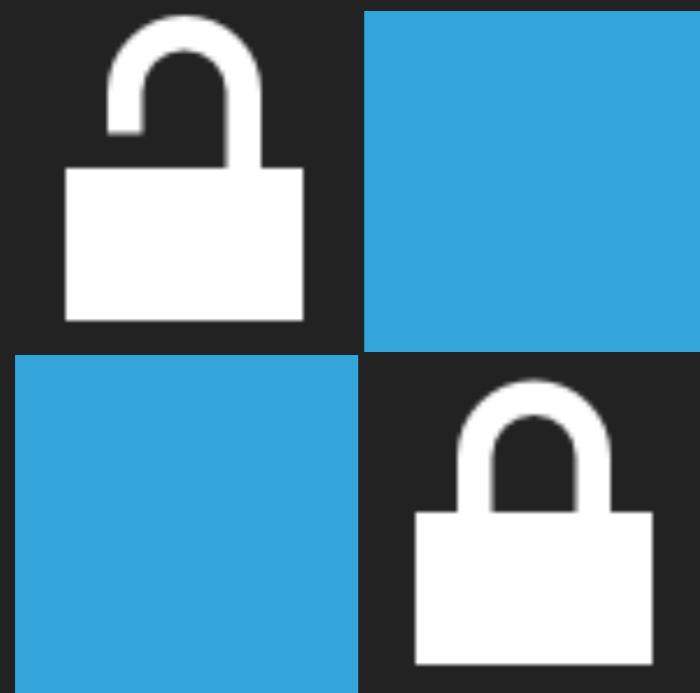
PATTERNS

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

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



- Data treatment ...
- Use existing ...
- ...

PATTERNS

CRYPTO CHECKLIST

CRYPTO CHECKLIST

- Data treatment ...
- Use existing ...
- ...

- 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)

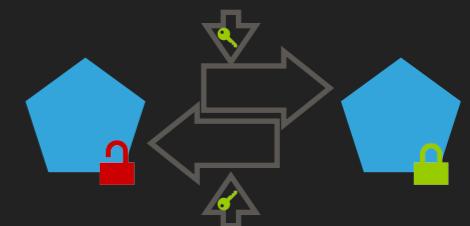
→ Assess risks & cost, plan, implement!

SUMMARY

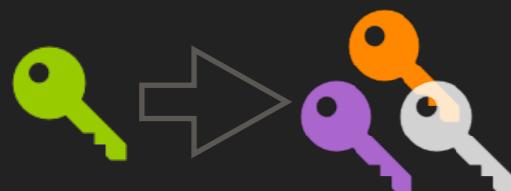
Comparing data



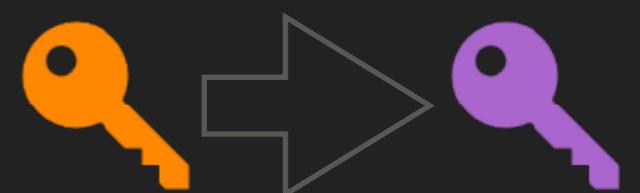
Transparent encryption



Storing data



Key derivation



Key refresh

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

Algorithm rollover

SUMMARY

0X123456...



Integrity

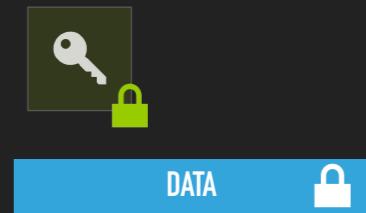
```
int getRandomNumber()
{
    return 4; // chosen by fair dice roll.
              // guaranteed to be random.
}
```

Entropy

Access Control

- Data treatment ...
- Use existing ...
- ...

Crypto Checklist

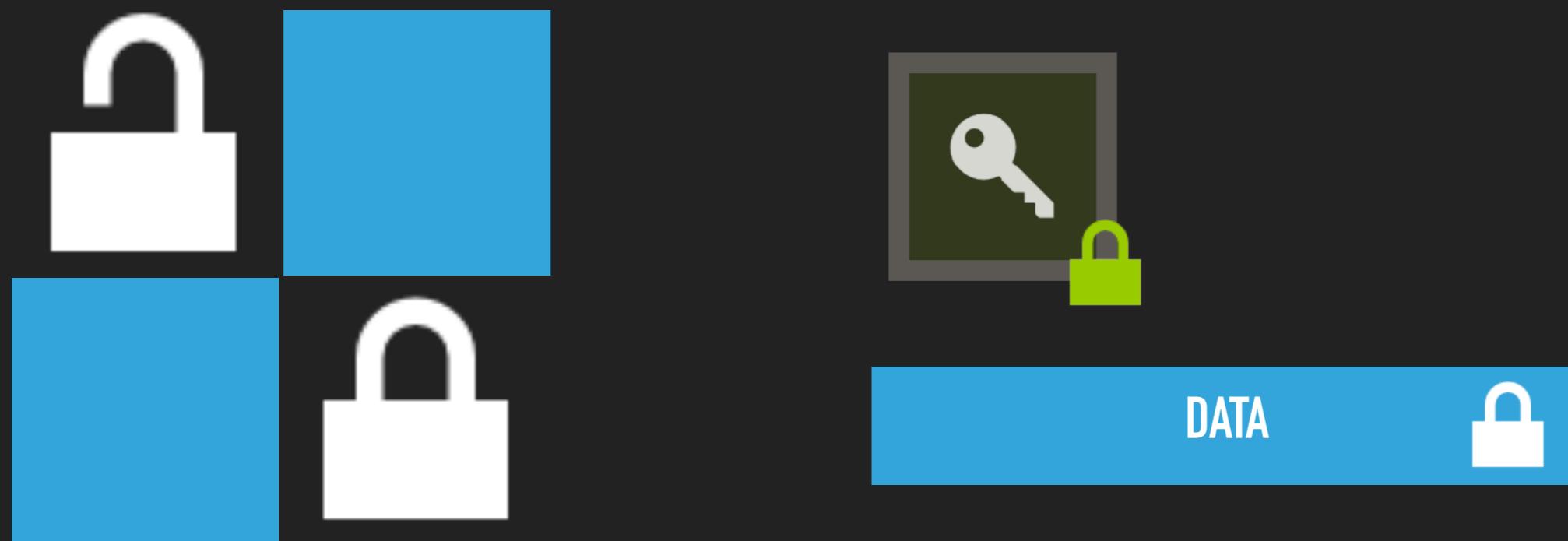


Q & A



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

BACKUP



PATTERNS

ACCESS
CONTROL

ACCESS CONTROL

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

Solution: Use cryptographic access controls

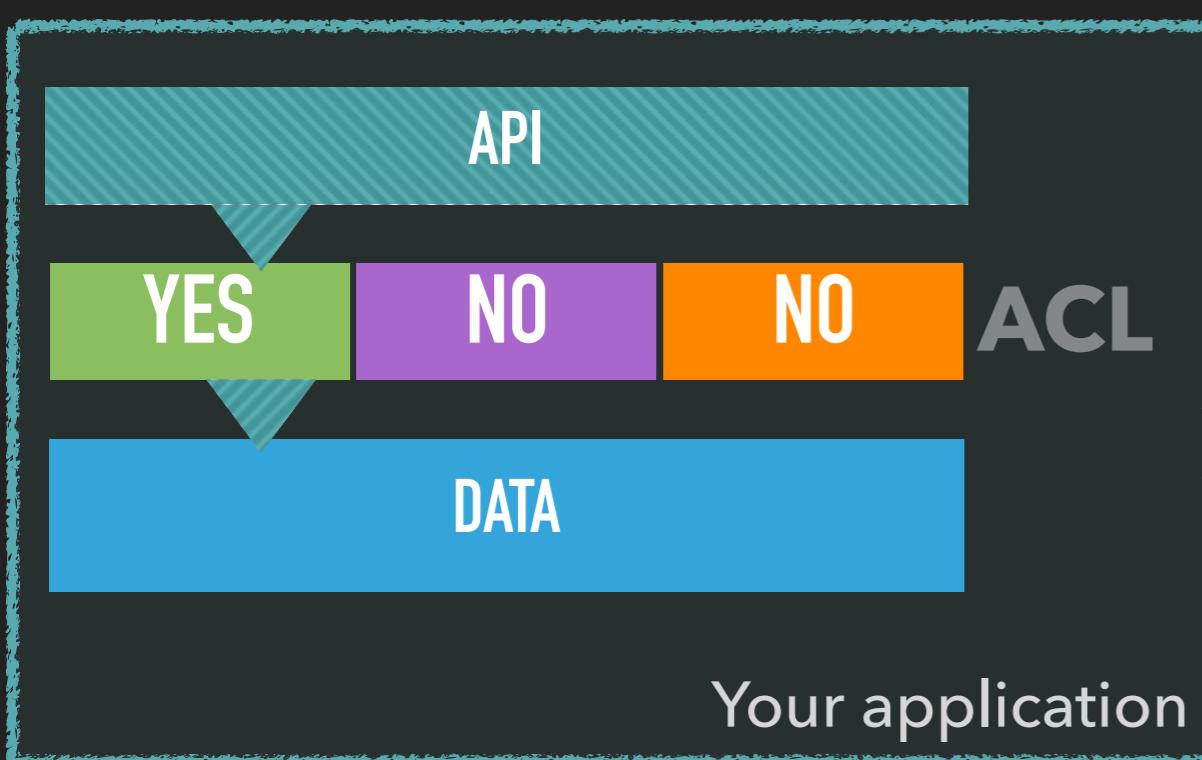
Applies primarily to

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

ACCESS CONTROL

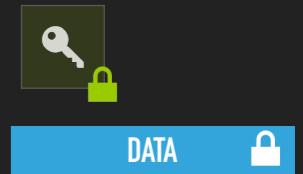


Alice Bob Eve

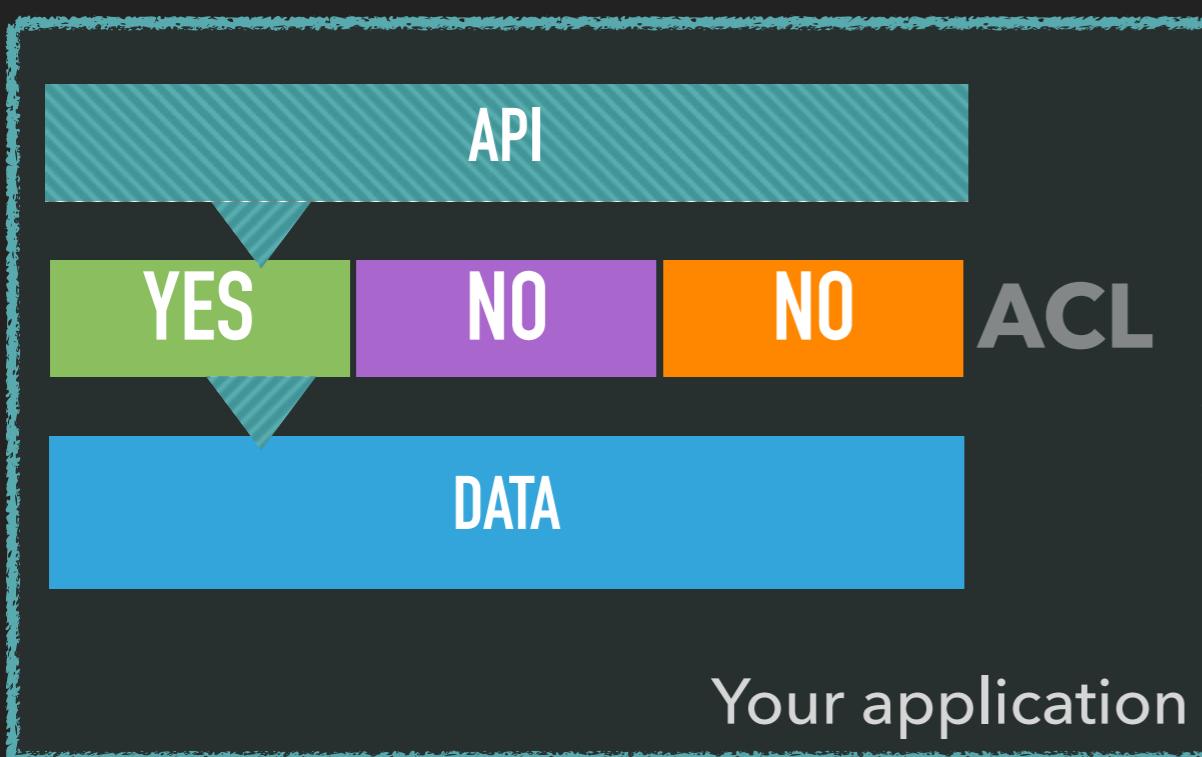


1. Alice tries to read data

ACCESS CONTROL



Alice Bob Eve

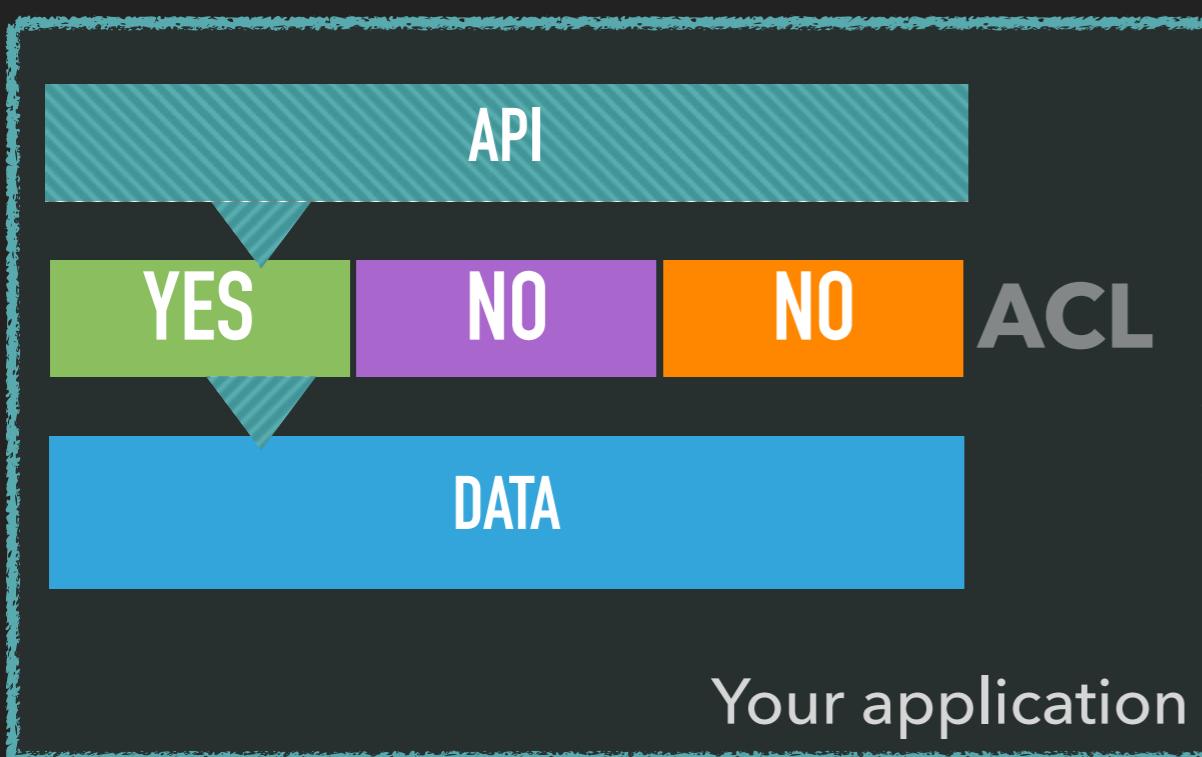


1. Alice tries to read data

ACCESS CONTROL



Alice Bob Eve

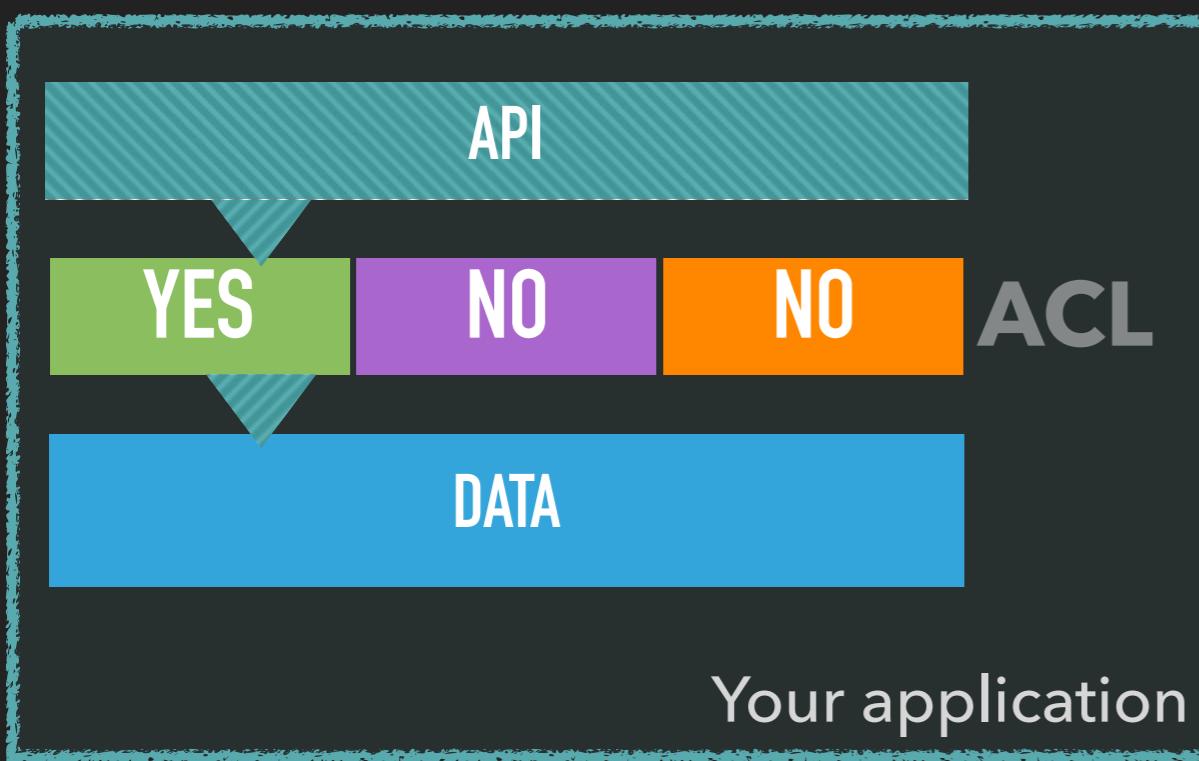


1. Alice tries to read data
2. Application validates ACL

ACCESS CONTROL

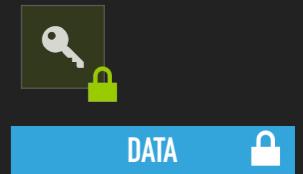


Alice Bob Eve



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

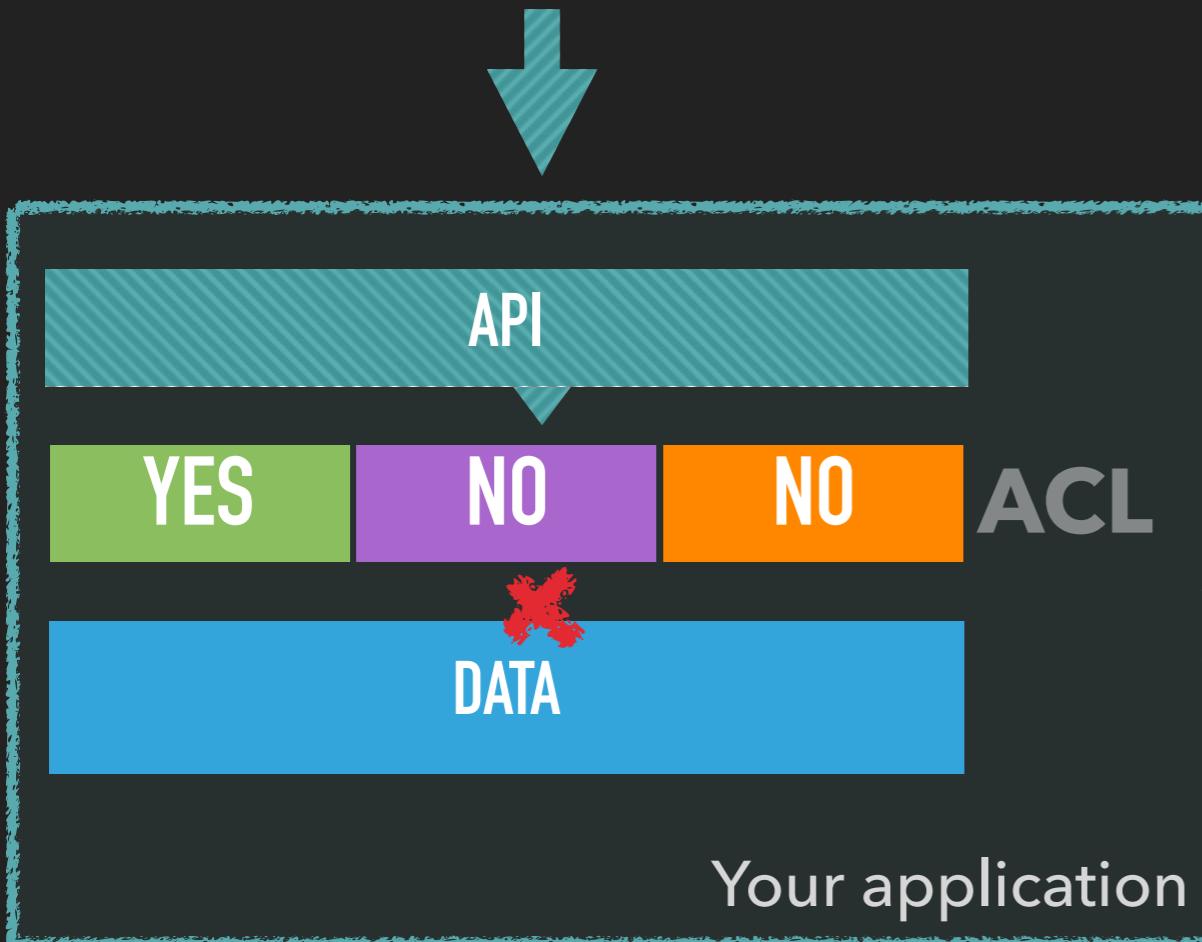
ACCESS CONTROL



DATA



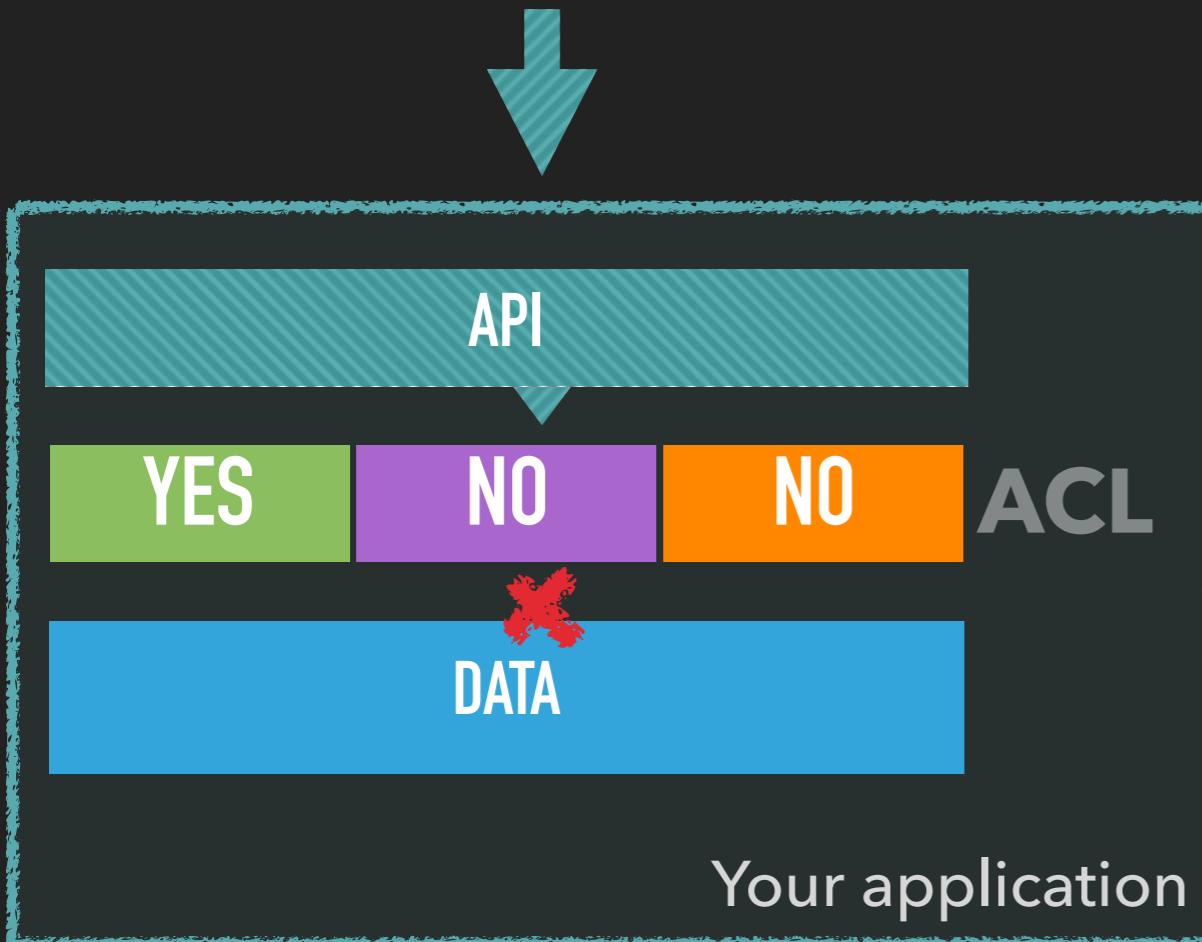
1. Bob tries to read data



ACCESS CONTROL

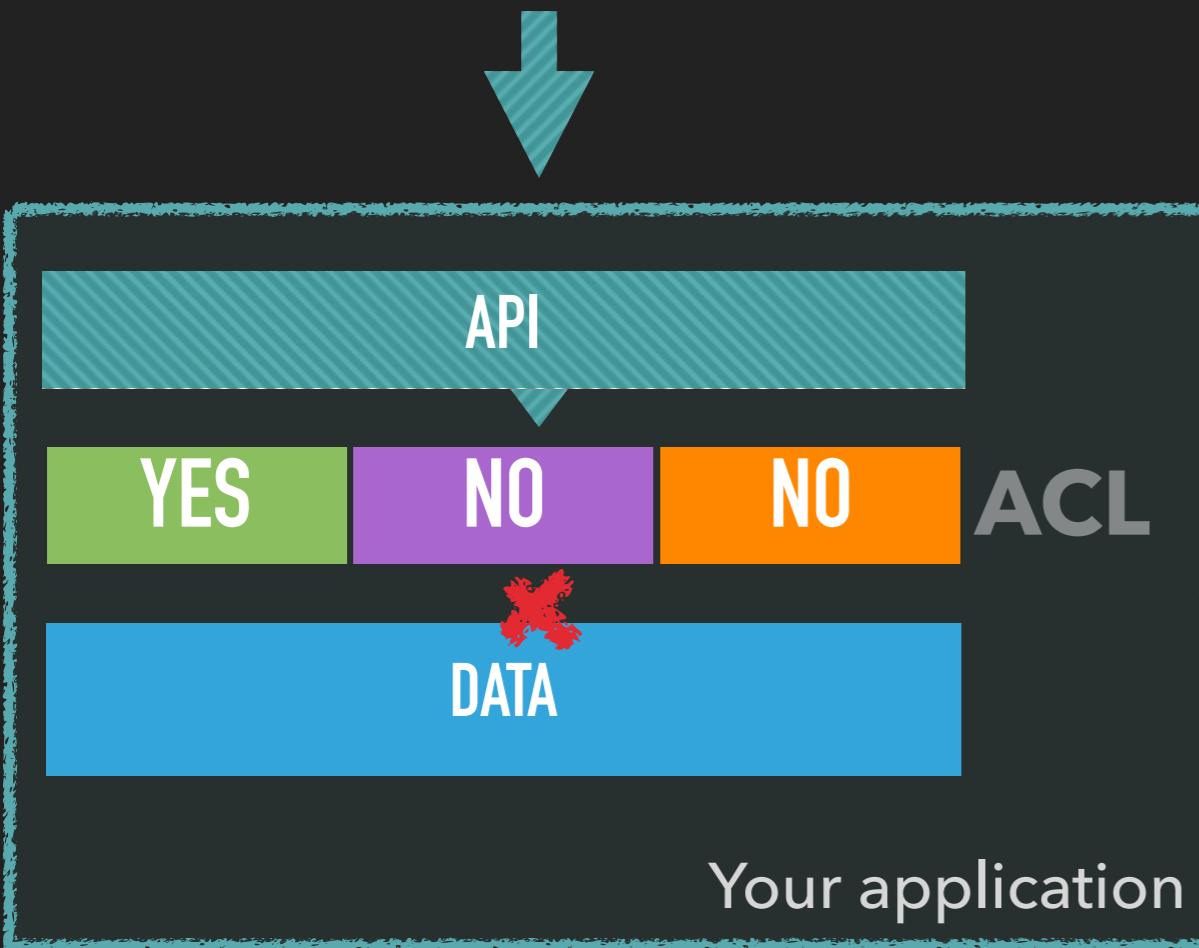


Alice Bob Eve



1. Bob tries to read data

ACCESS CONTROL

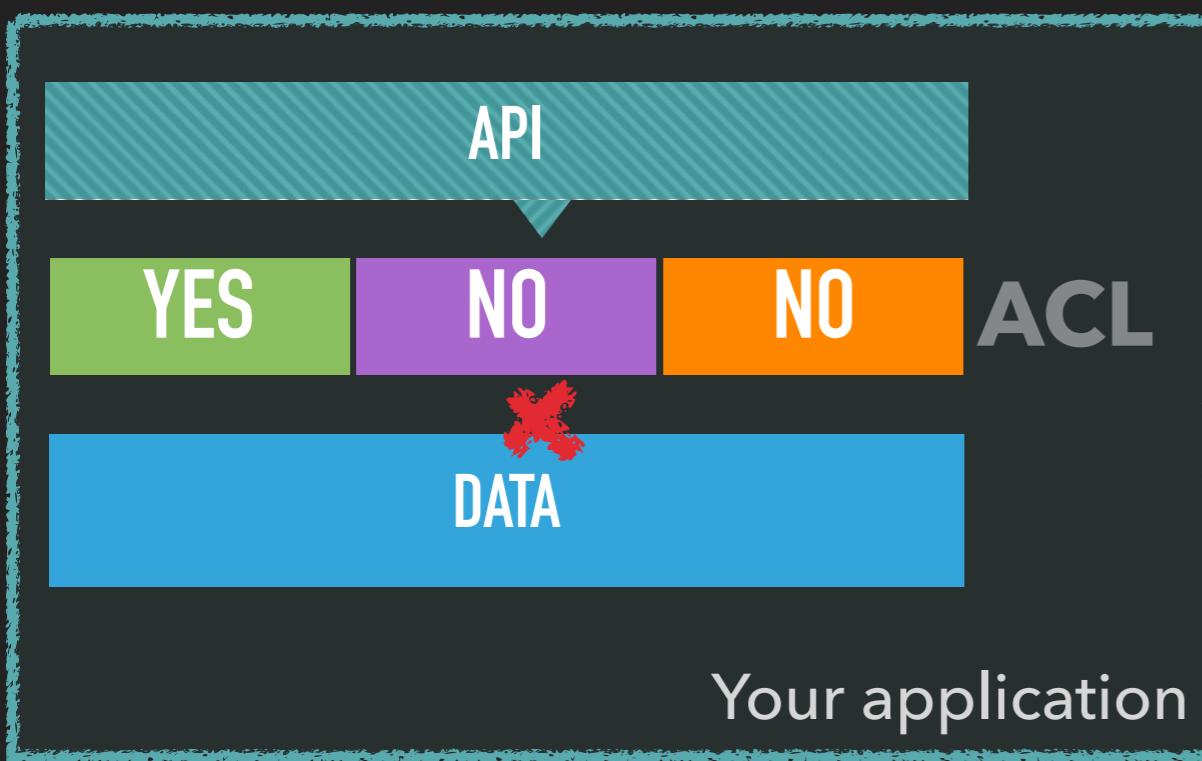


1. Bob tries to read data
2. Application validates ACL

ACCESS CONTROL

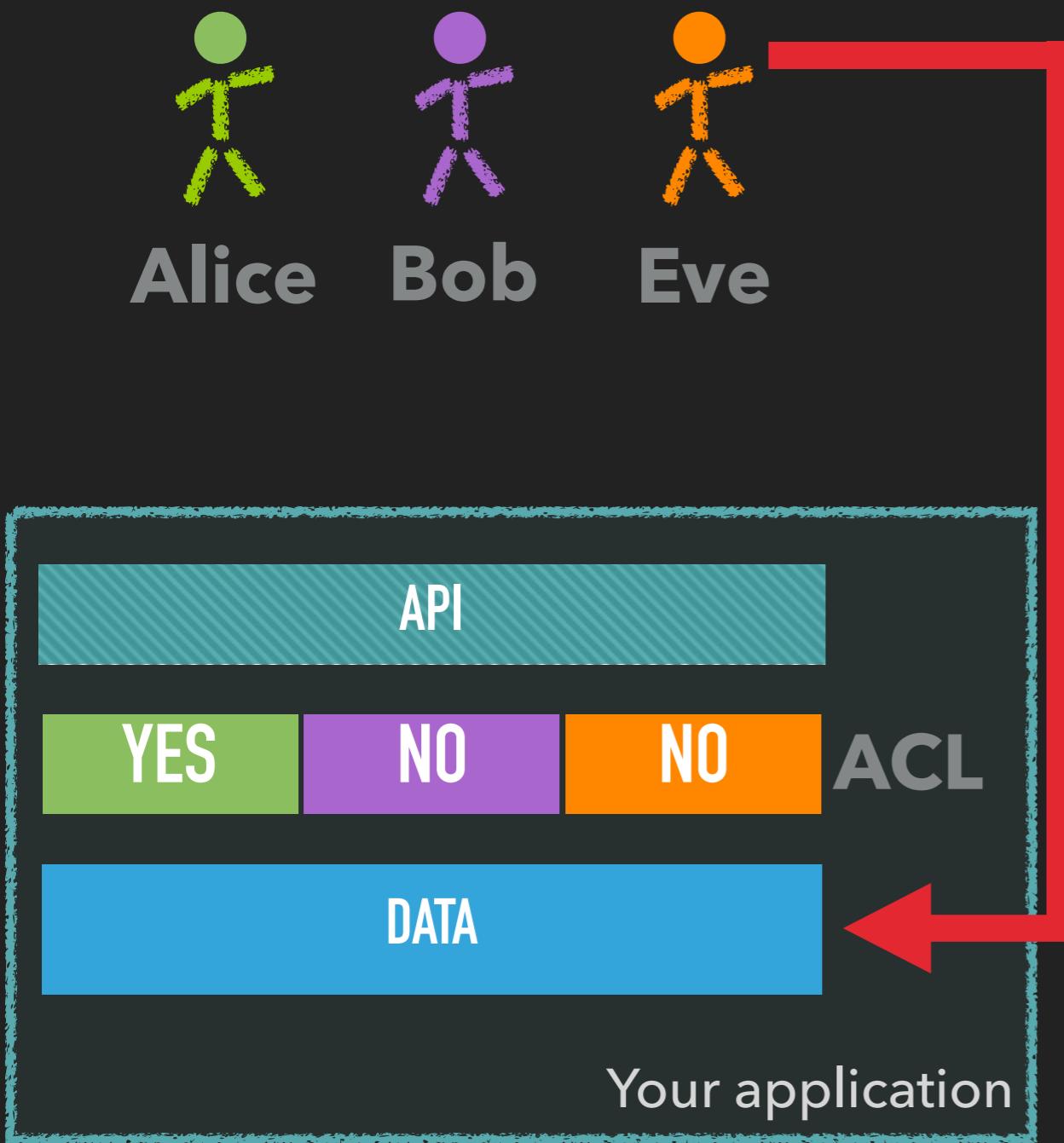


Alice Bob Eve



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

ACCESS CONTROL

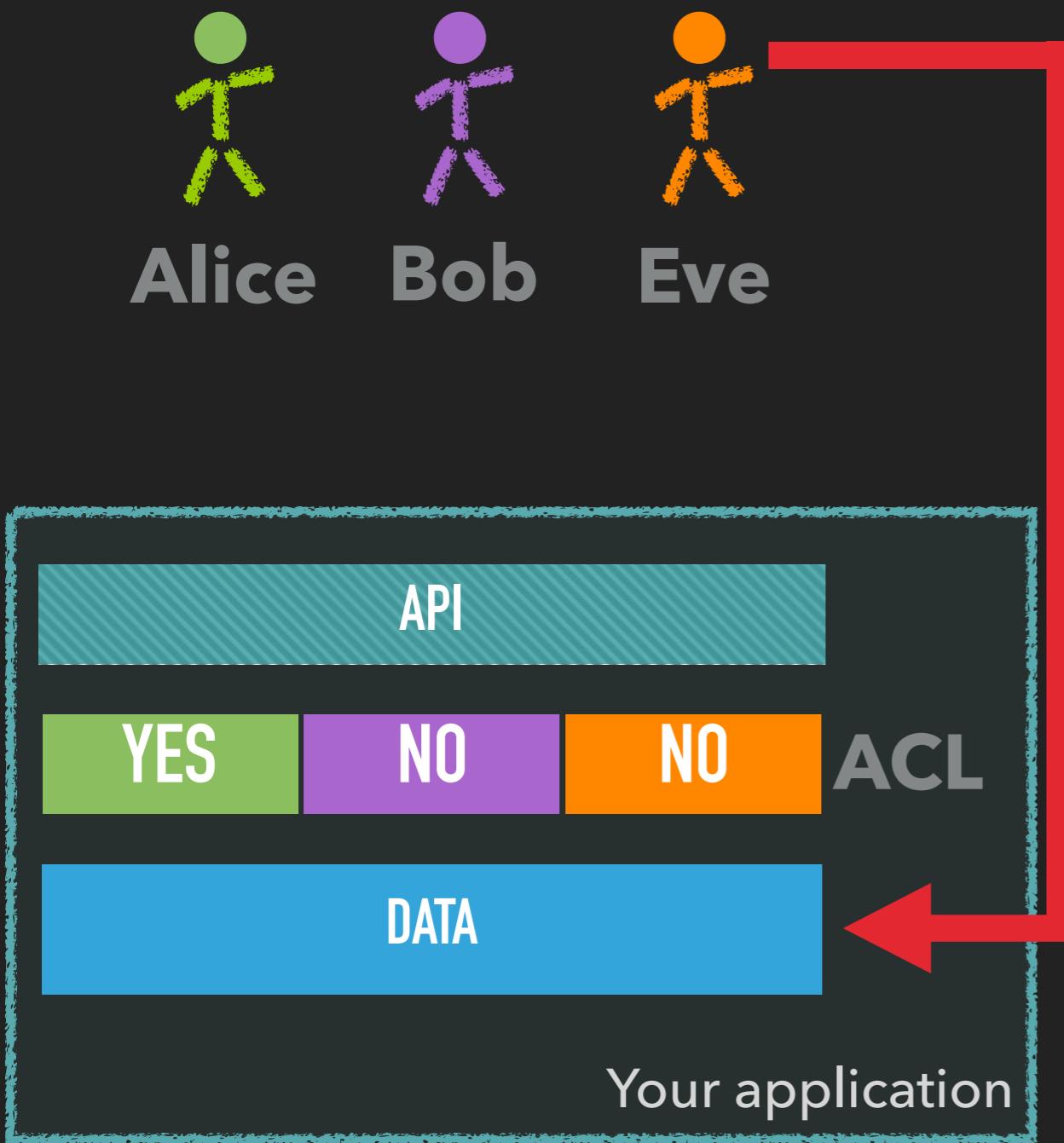


1. Eve exploits application

ACCESS CONTROL

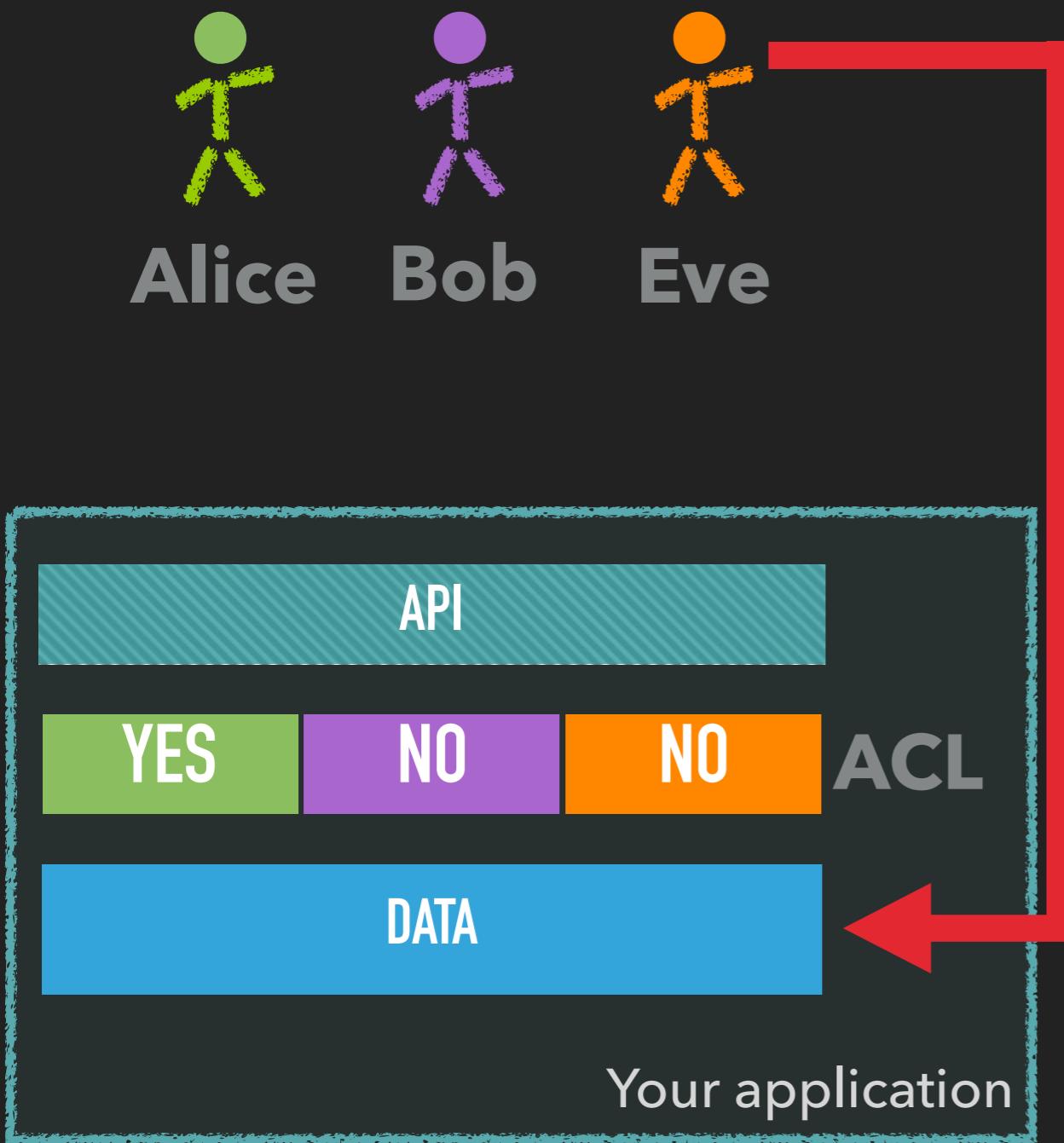
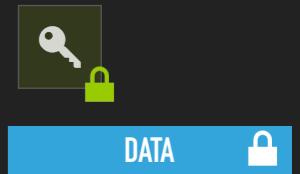


DATA



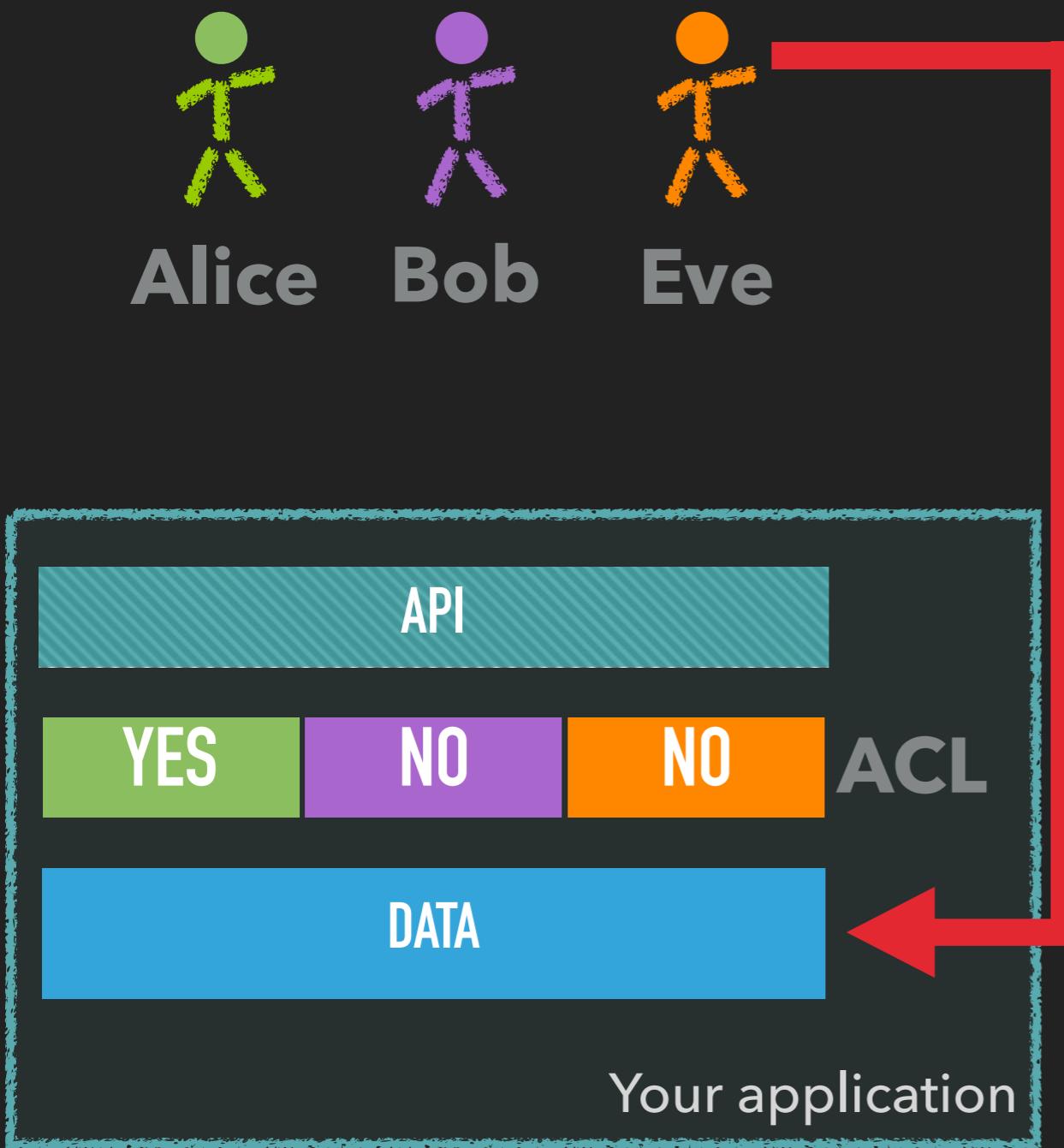
1. Eve exploits application

ACCESS CONTROL



1. Eve exploits application
2. Application ????

ACCESS CONTROL

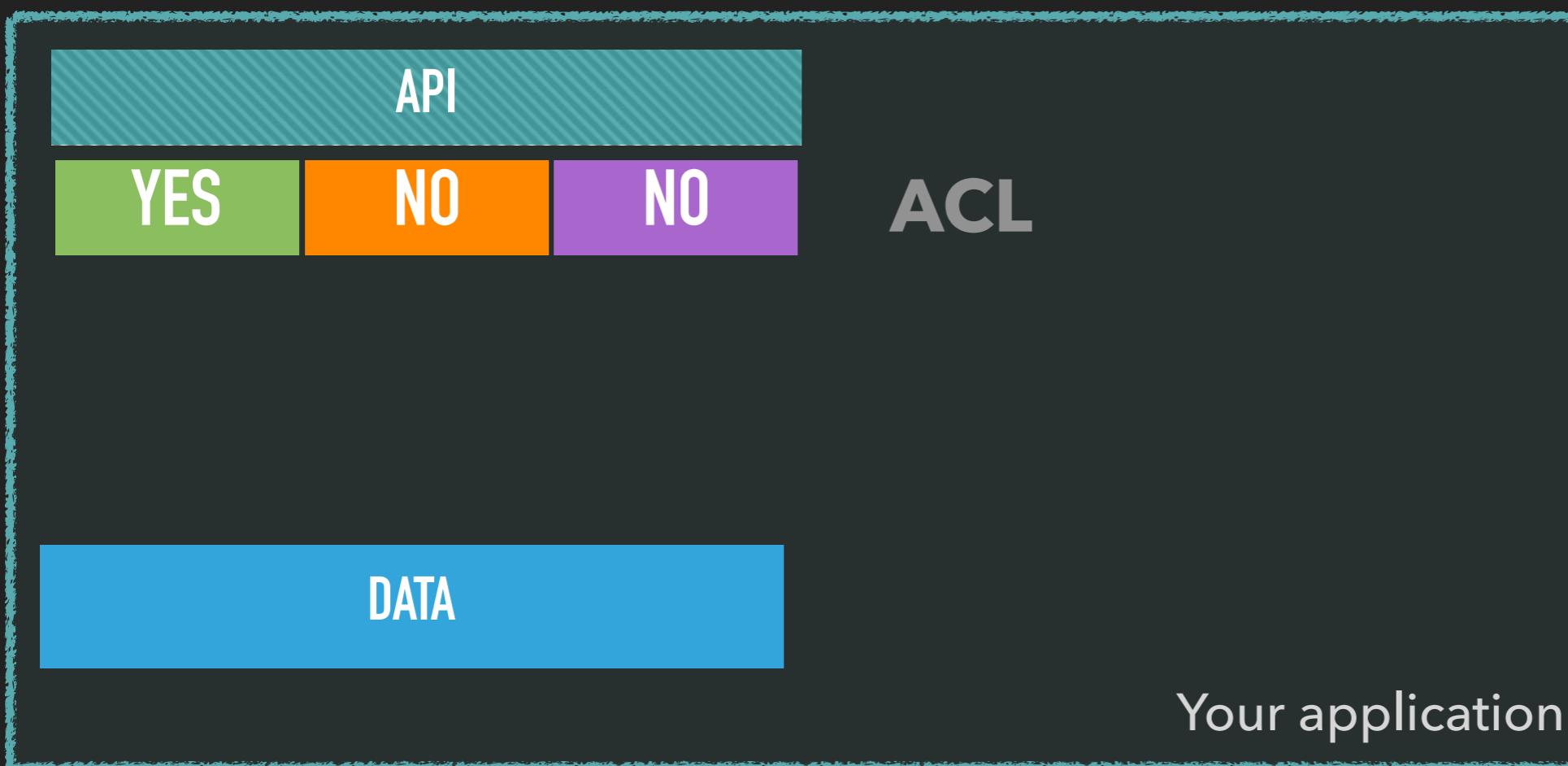


1. Eve exploits application
2. Application ????
3. Eve has access to data

ACCESS CONTROL



Alice Bob Eve



ACCESS CONTROL



Alice Bob Eve

API

YES

NO

NO

ACL

DATA



encrypted with

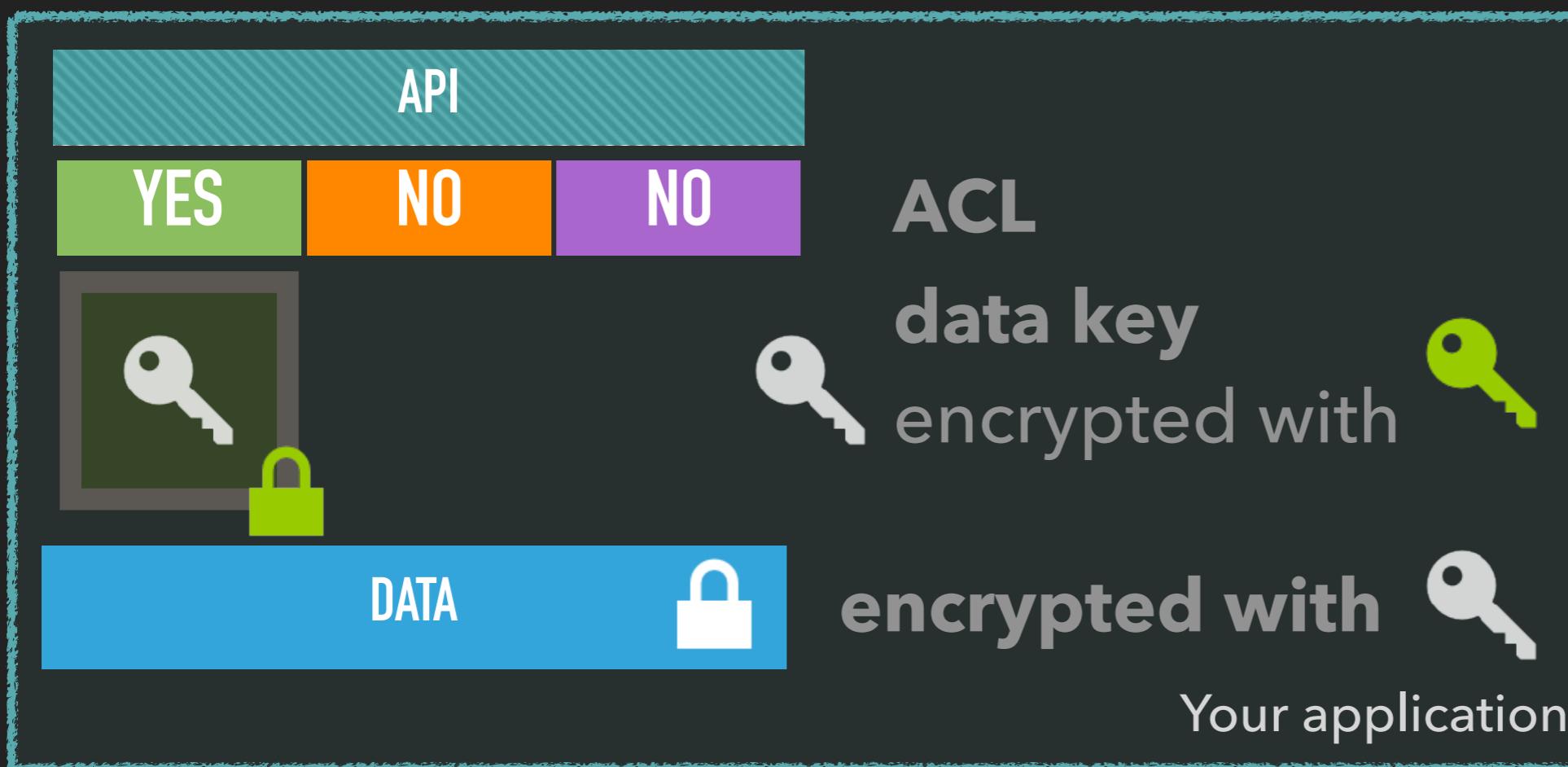
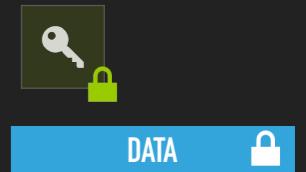


Your application

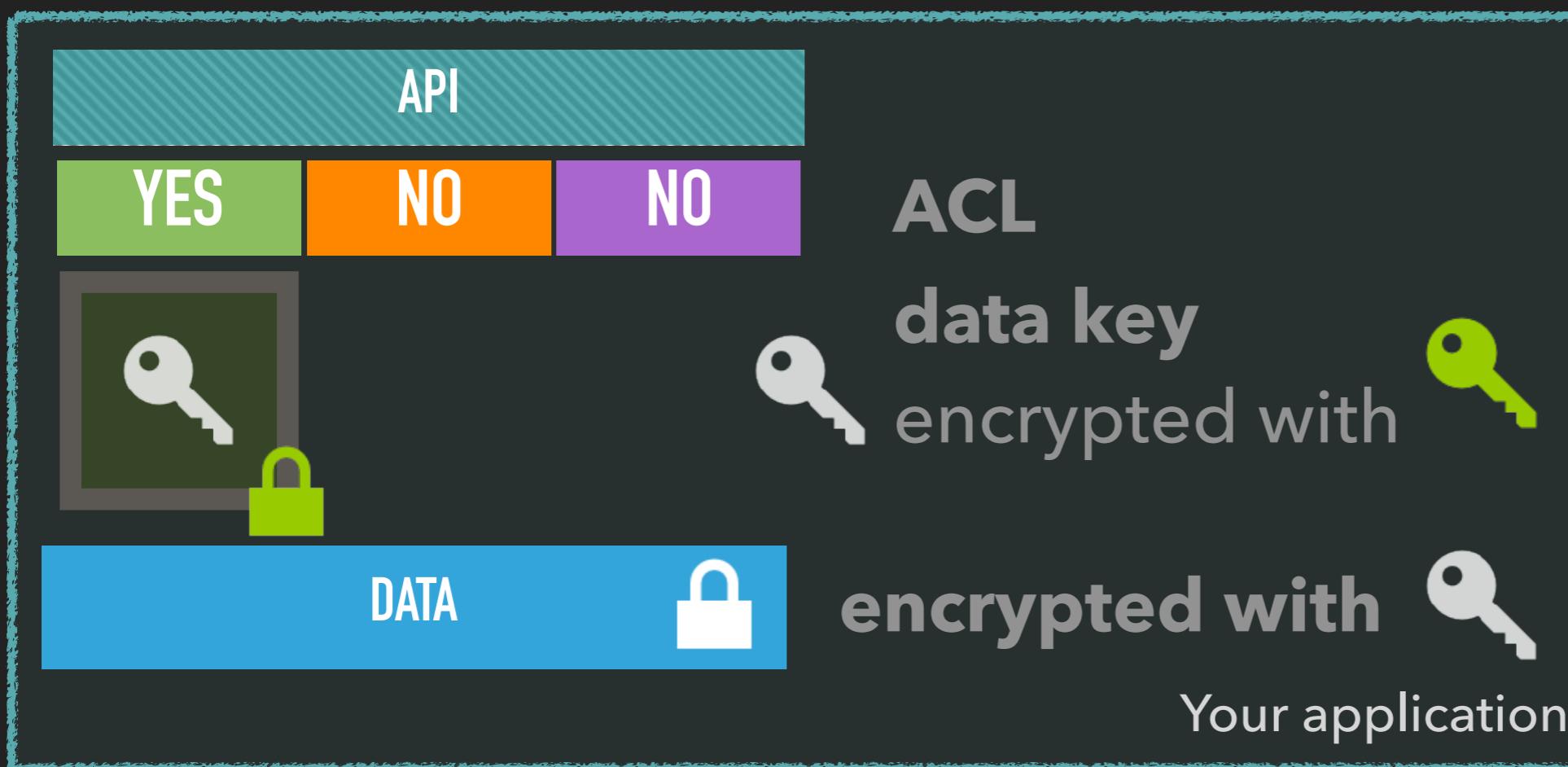
ACCESS CONTROL



Alice Bob Eve



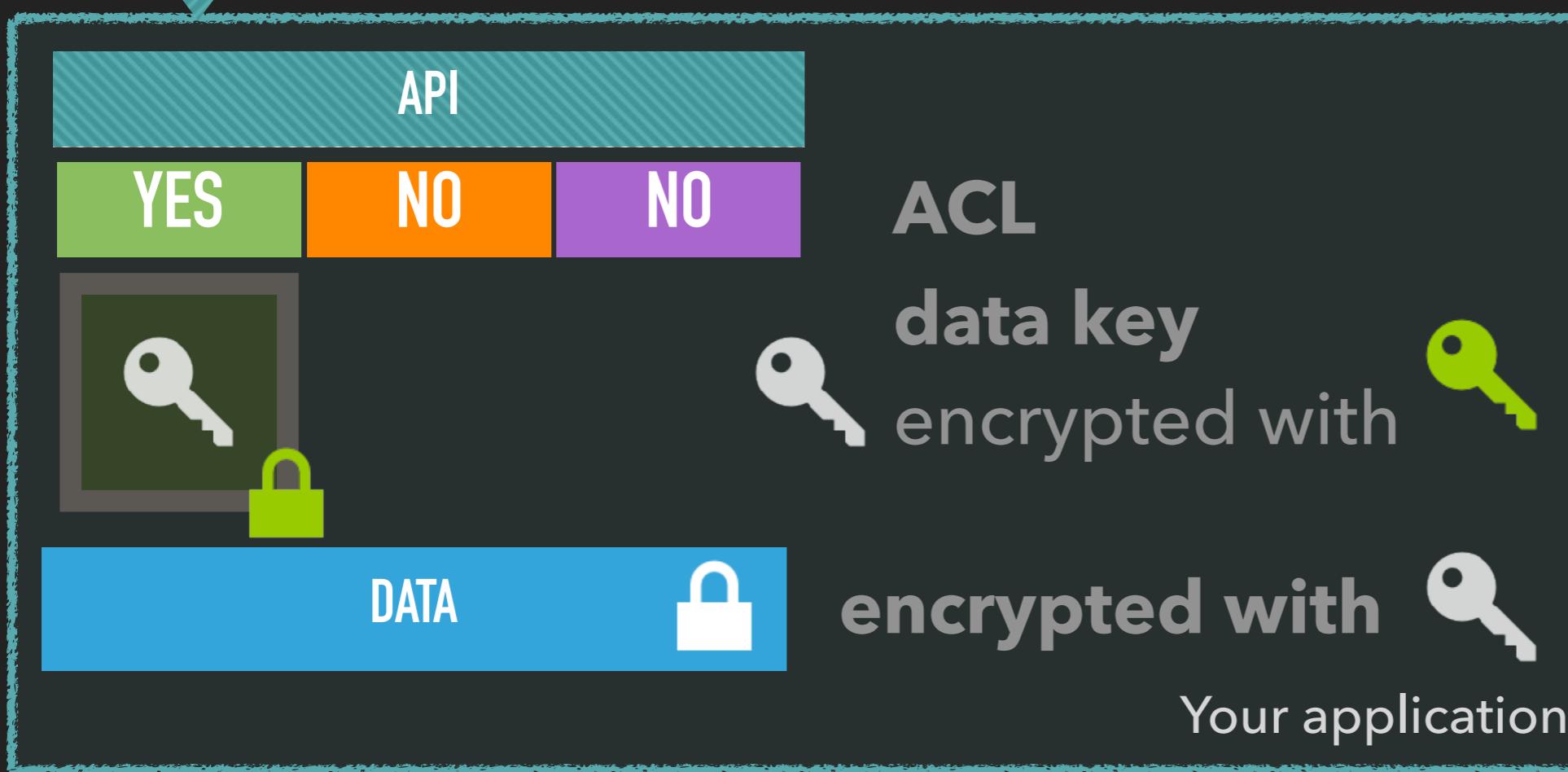
ACCESS CONTROL



ACCESS CONTROL

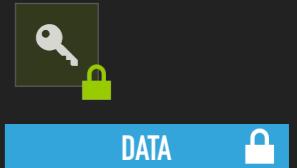


Alice Bob Eve

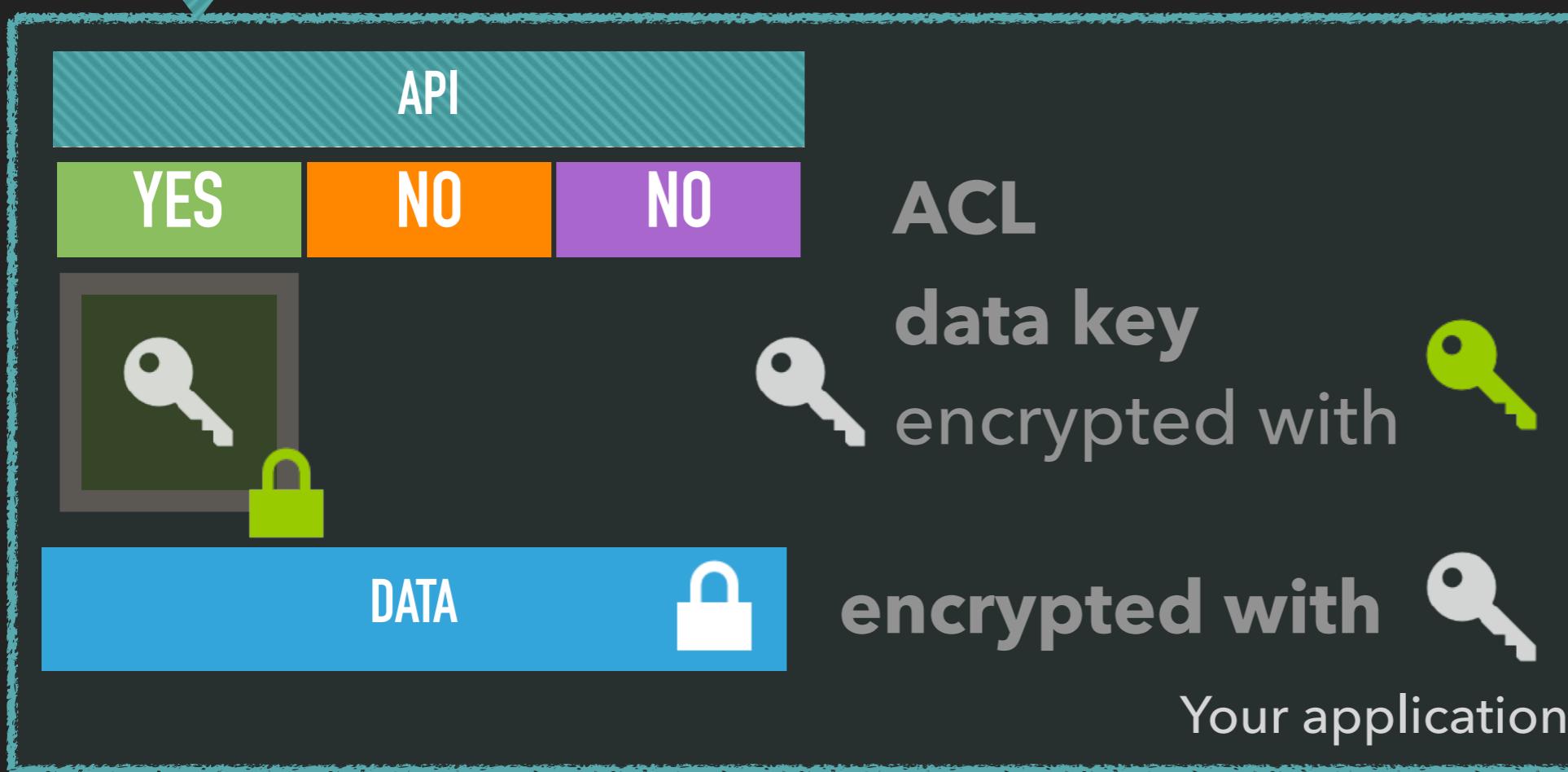
Three stick figure icons representing users: Alice (green), Bob (purple), and Eve (orange). Each user has a corresponding key icon below them.

ACCESS CONTROL

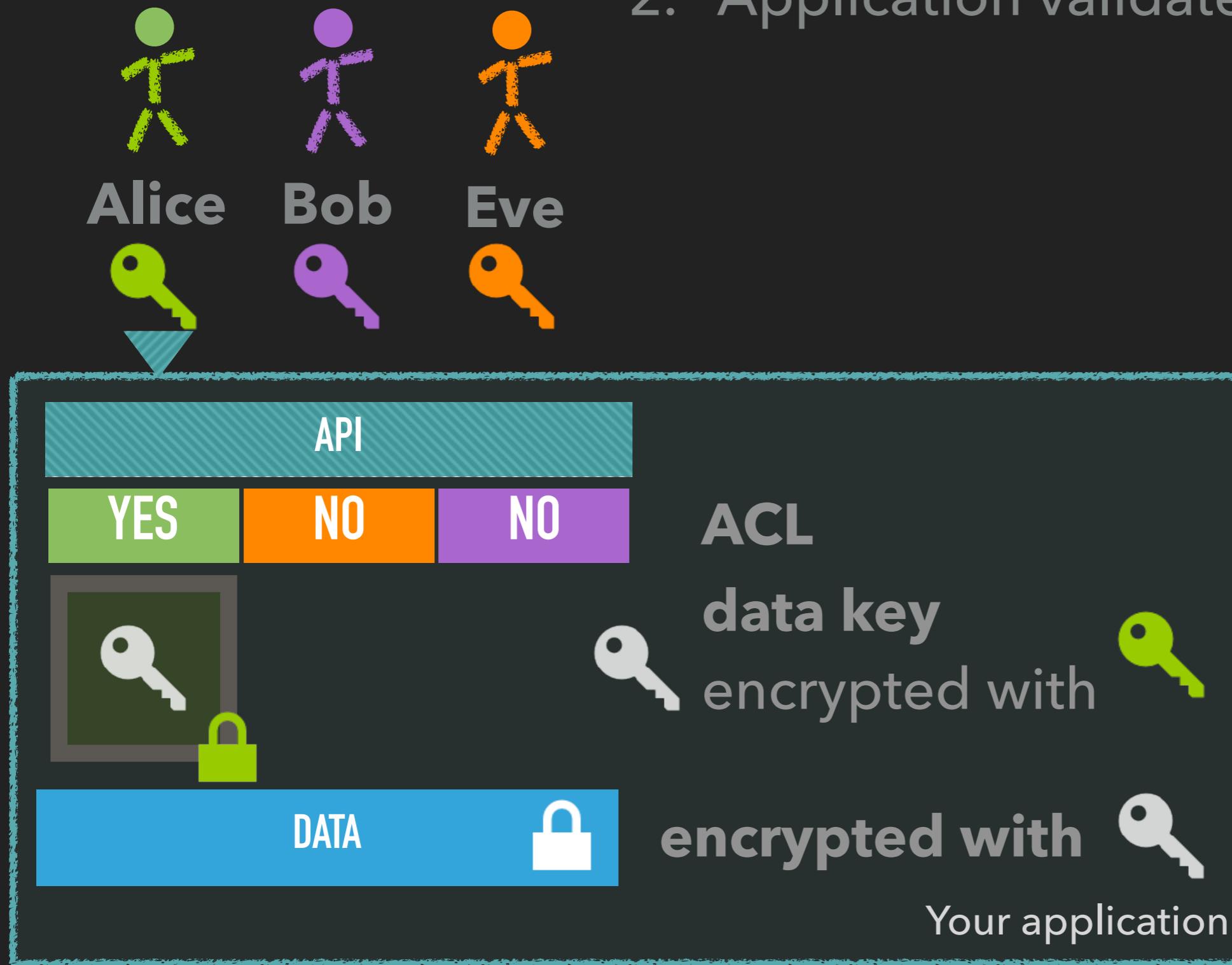
1. Alice tries to read data



Alice Bob Eve
Three key icons corresponding to the stick figures: a green key, a purple key, and an orange key.



ACCESS CONTROL



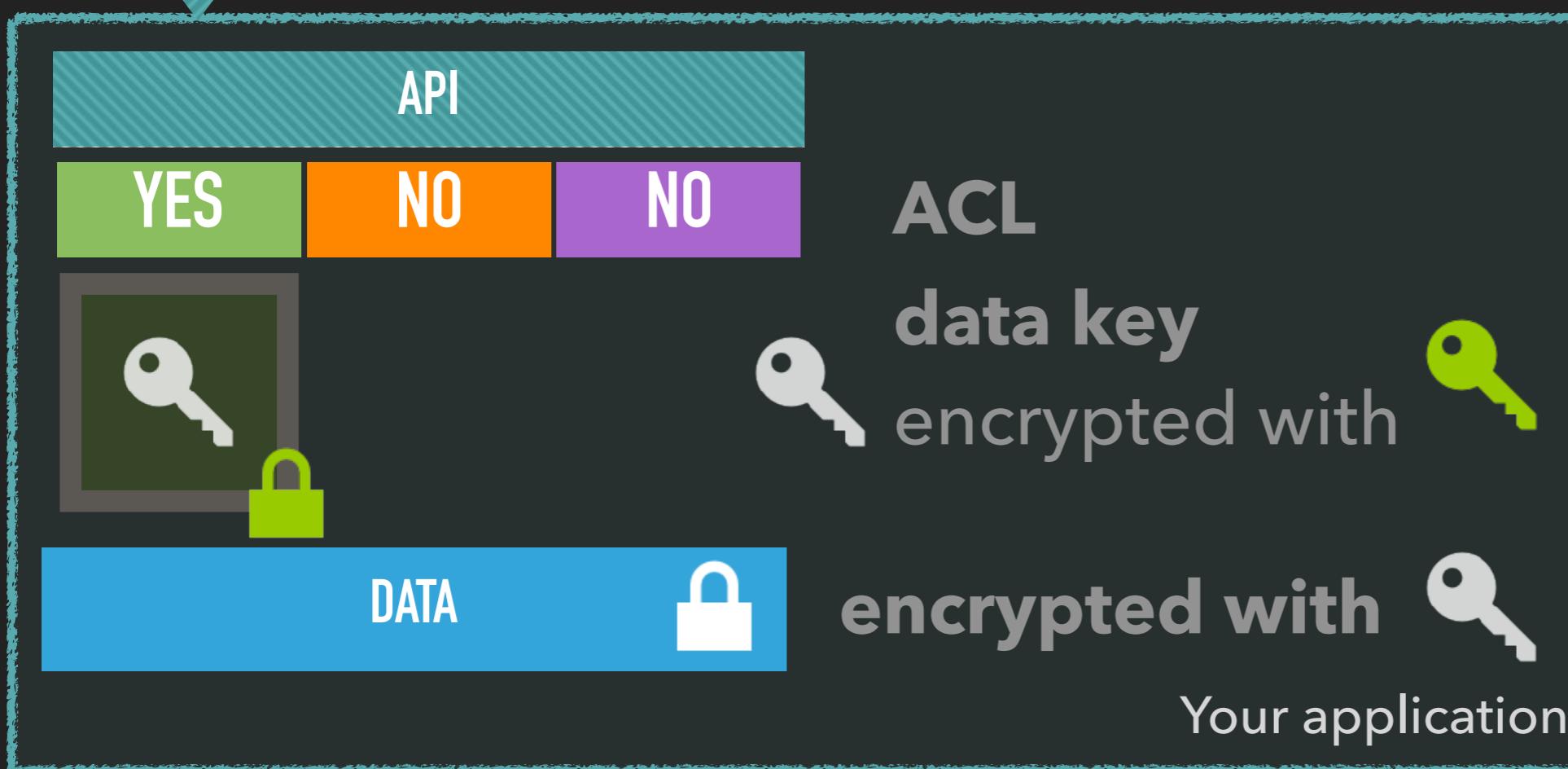
ACCESS CONTROL



Alice Bob Eve

Below each stick figure is a magnifying glass icon. The green magnifying glass is held by the green stick figure, the purple by the purple one, and the orange by the orange one. A blue downward-pointing arrow is positioned between the Alice and Bob icons.

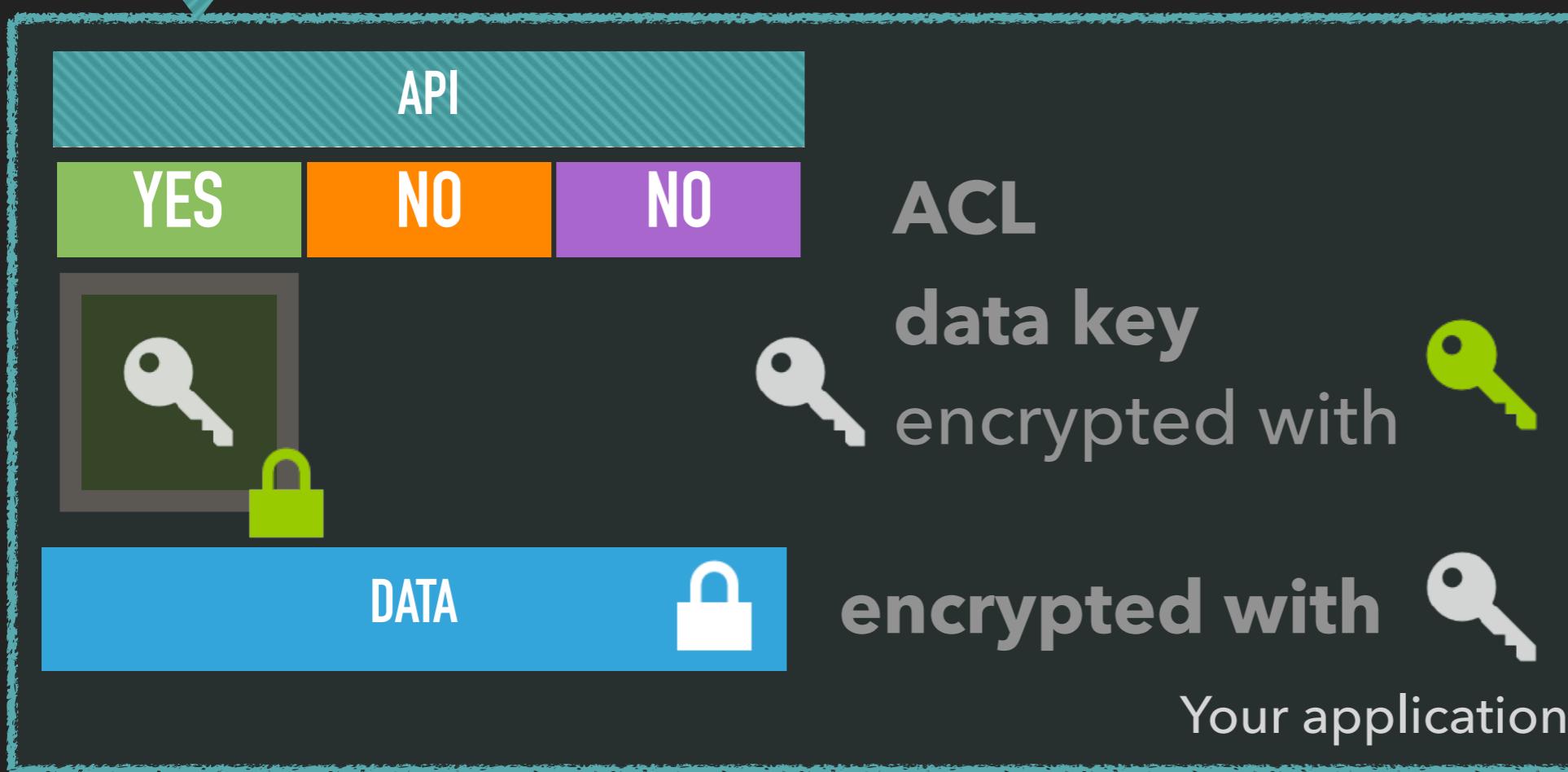
1. Alice tries to read data
2. Application validates ACL
3. Alice is granted access



ACCESS CONTROL



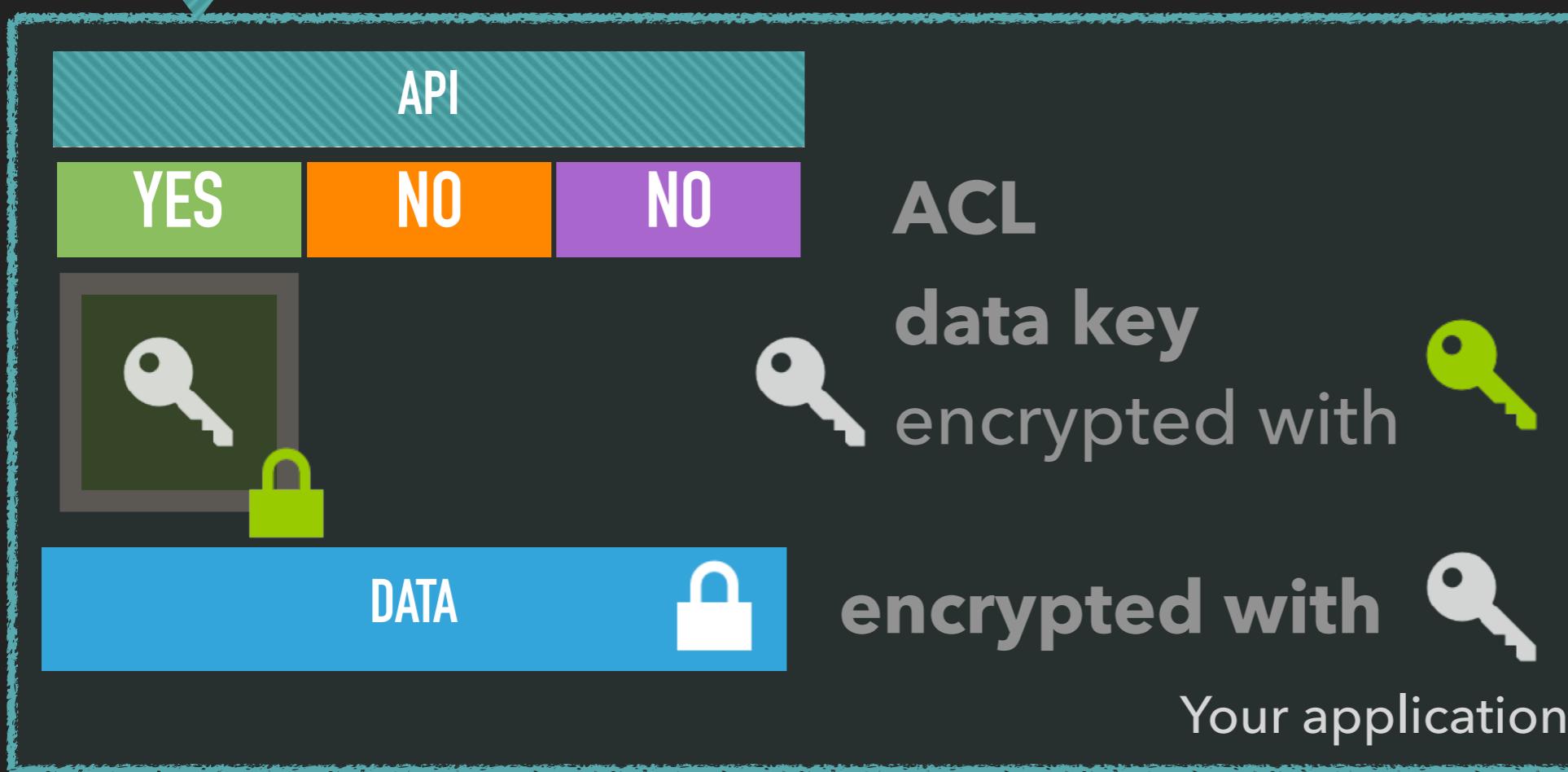
1. Alice tries to read data
2. Application validates ACL
3. Alice is granted access
4. Alice decrypts the “data key”



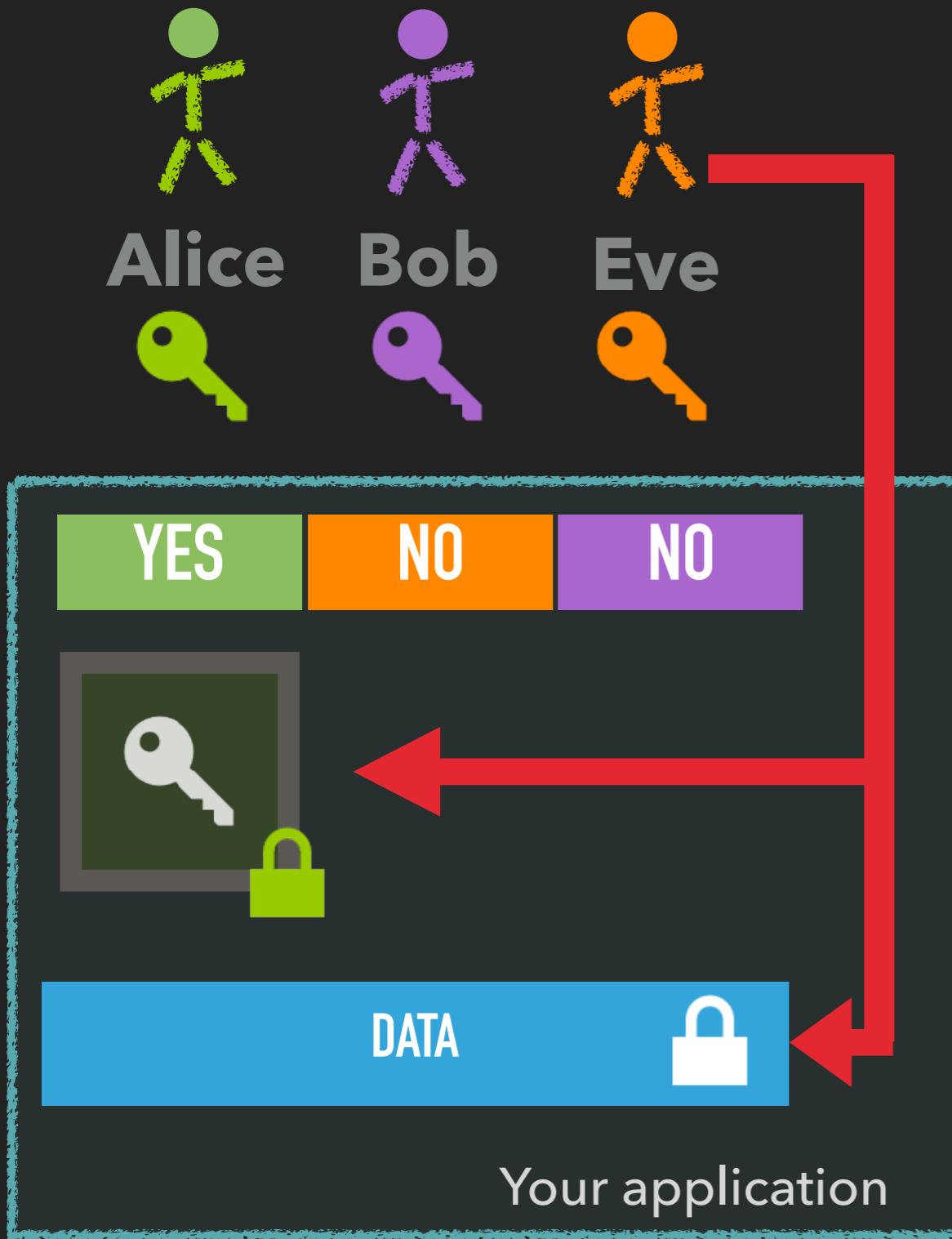
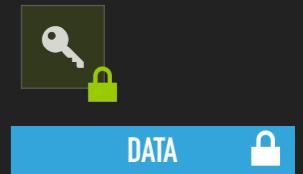
ACCESS CONTROL



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

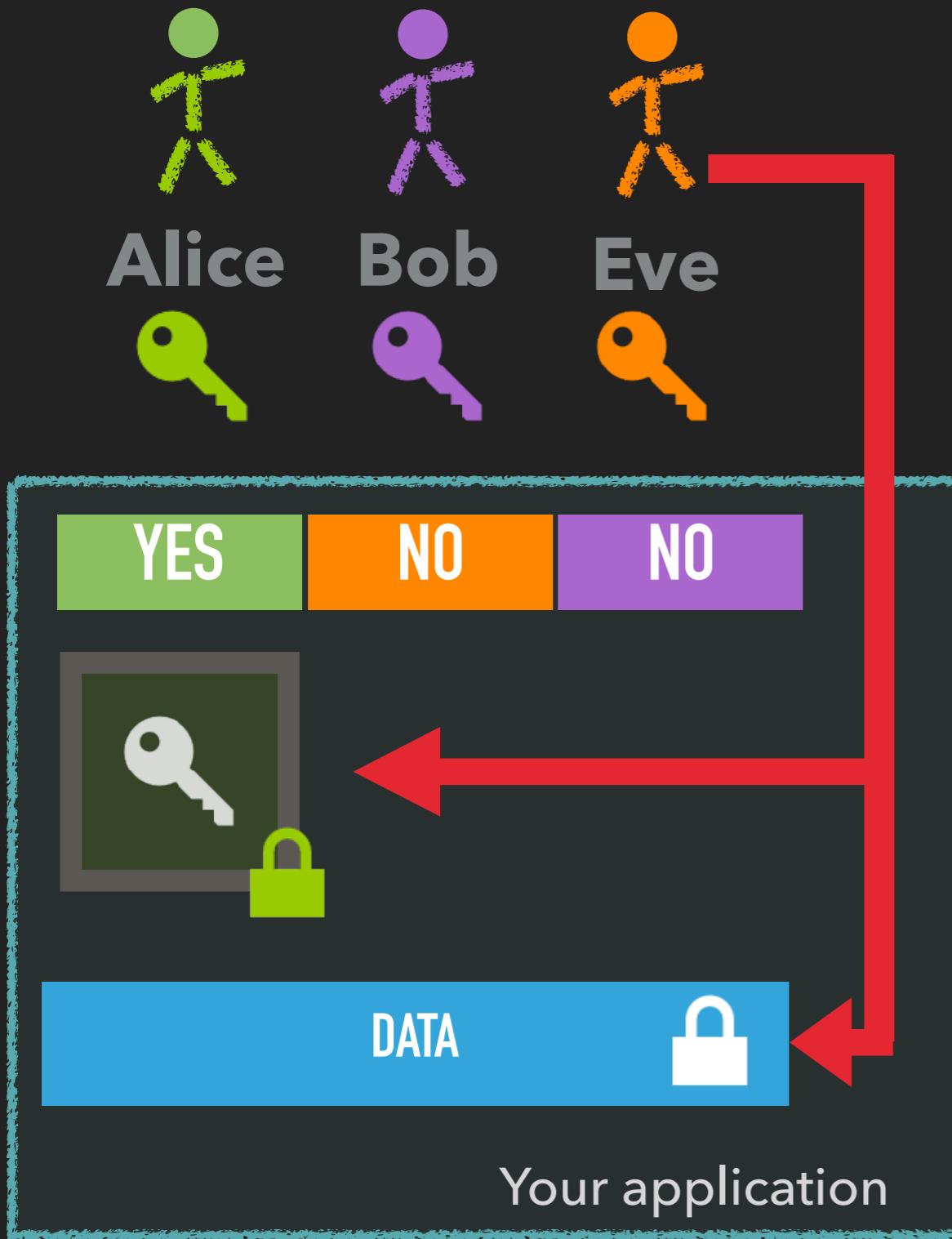


ACCESS CONTROL



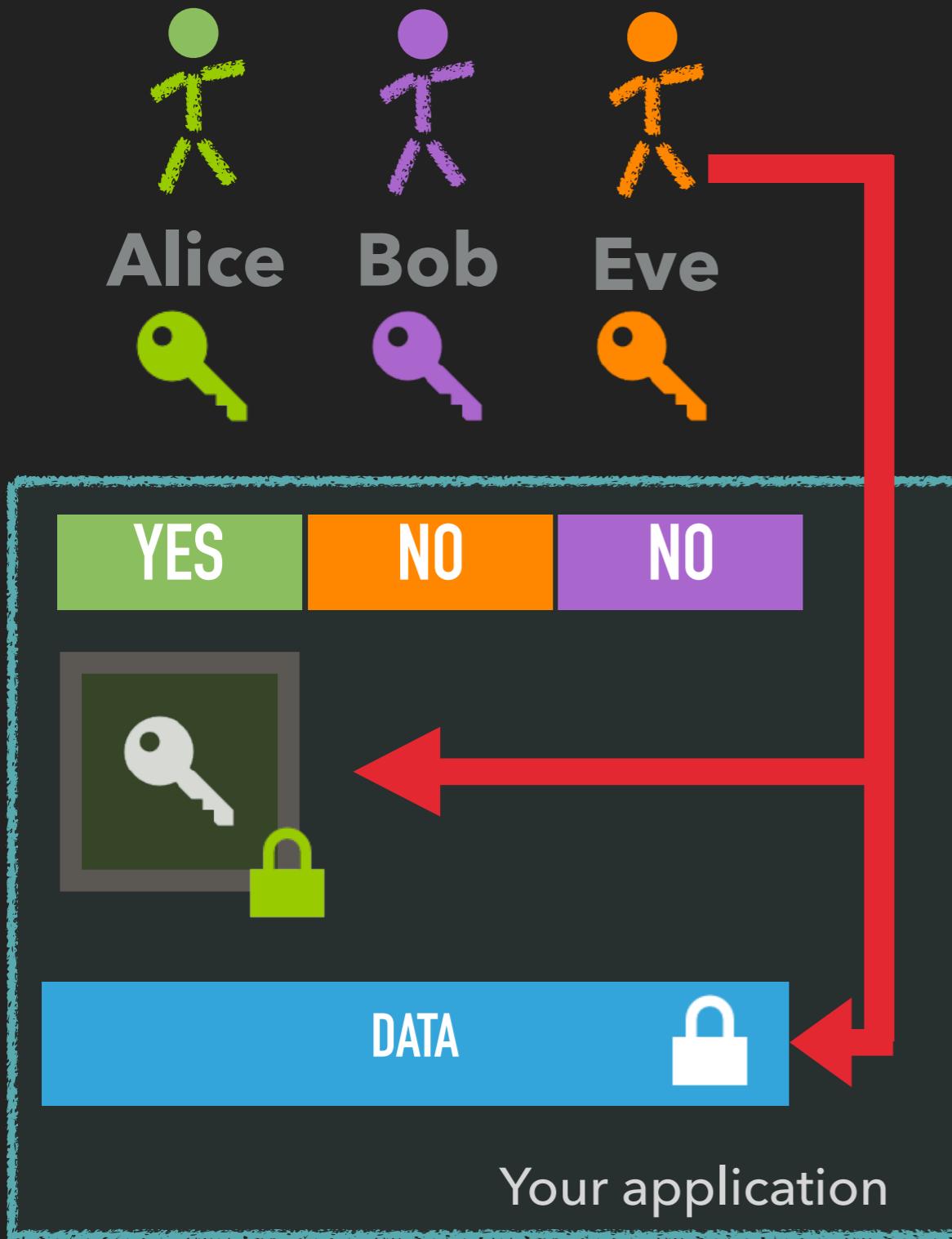
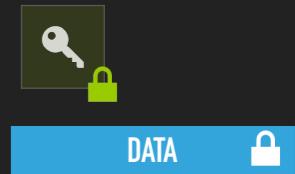
1. Eve exploits application

ACCESS CONTROL



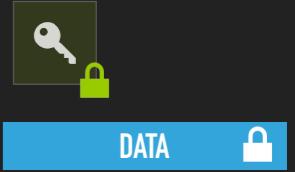
1. Eve exploits application
2. Eve gets encrypted data key
OR
Eve gets encrypted data
Eve cannot decrypt the data!

ACCESS CONTROL



1. Eve exploits application
2. Eve gets encrypted data key
OR
Eve gets encrypted data
Eve cannot decrypt the data!

ACCESS CONTROL



1. Alice enters her password

ACCESS CONTROL



(1)

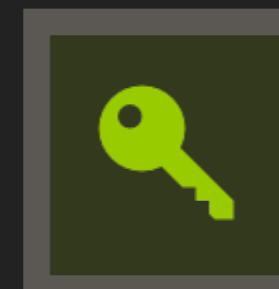
Alice

1. Alice enters her password

ACCESS CONTROL



Alice



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

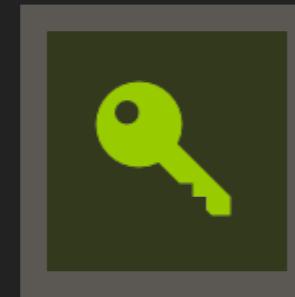
Alice' long term secret key
encrypted with her
password.

ACCESS CONTROL



(1)

Alice

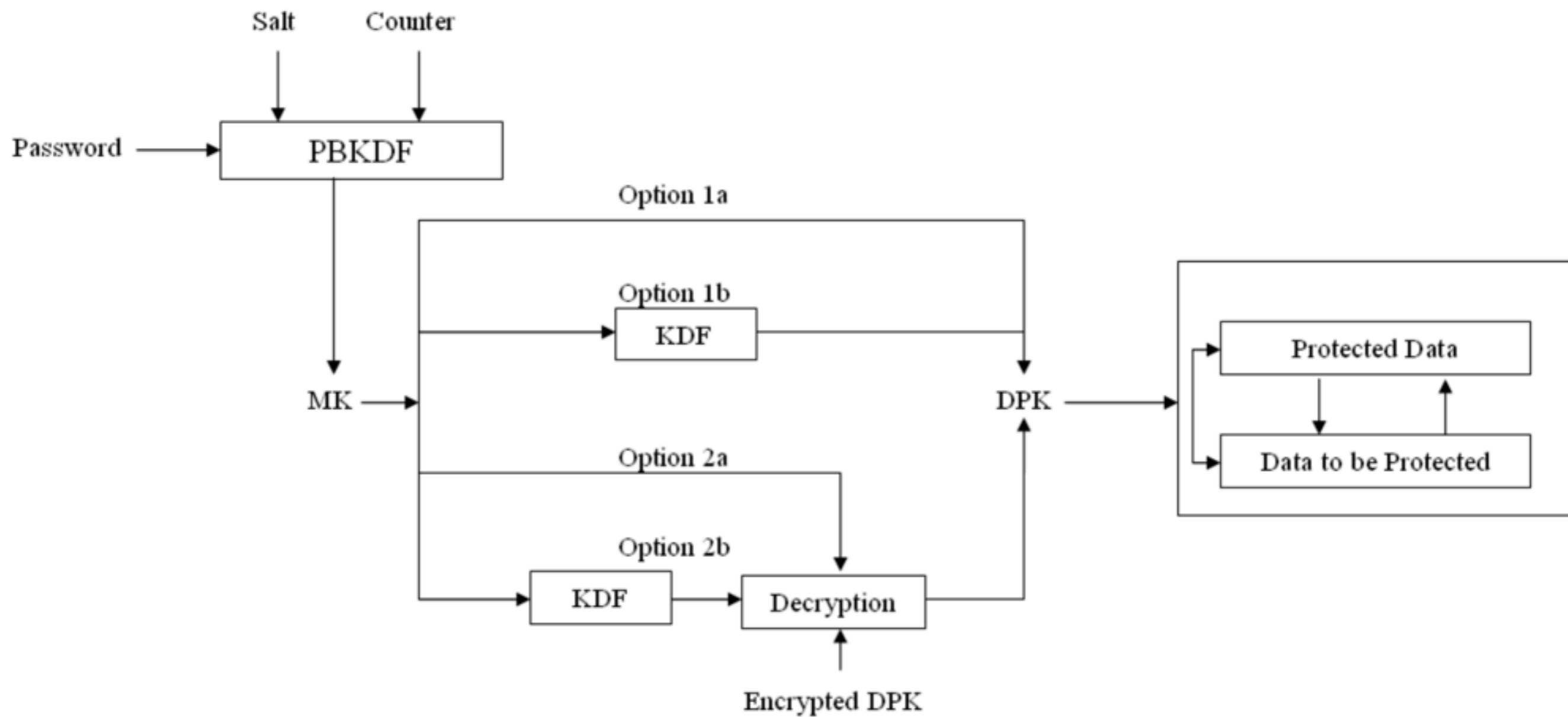


(2)

**Alice**

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

Alice' long term secret key
encrypted with her
password.



Option 1a: The MK or a segment of the MK is used as DPK.

Option 1b: The DPK is derived from the MK using an **approved** key derivation function.

Option 2a: The MK is used to recover the DPK through an **approved** authenticated encryption or an **approved** key protection/recovery method.

Option 2b: The key to recover the DPK through an **approved** authenticated encryption or an **approved** key protection/recovery method is derived from the MK using an **approved** key derivation function.

§

LEGAL

(PSEUDO|
ANONYM)ISED?

(PSEUDO|ANONYM)ISED?

ANONYMISIERUNG

„das Verändern personenbezogener Daten derart, dass die Einzelangaben über persönliche oder sachliche Verhältnisse nicht mehr oder nur mit einem unverhältnismäßig großen Aufwand an Zeit, Kosten und Arbeitskraft einer bestimmten oder bestimmbaren natürlichen Person zugeordnet werden können.“ § 3 Abs. 6 BDSG



§

PSEUDONYMISIERUNG

„das Ersetzen des Namens und anderer Identifikationsmerkmale durch ein Kennzeichen zu dem Zweck, die Bestimmung des Betroffenen auszuschließen oder wesentlich zu erschweren.“ § 3 Abs. 6a BDSG

VERSCHLÜSSELUNG

„Technisch versteht man unter Verschlüsselung den Vorgang, bei dem ein klar lesbarer Text (Klartext) oder auch Informationen anderer Art wie Ton- oder Bildaufzeichnungen mit Hilfe eines Verschlüsselungsverfahrens (Kryptosystem) in eine „unleserliche“, das heißt nicht einfach interpretierbare Zeichenfolge (Geheimtext) umgewandelt wird. Als entscheidend wichtige Parameter der Verschlüsselung werden hierbei ein oder auch mehrere Schlüssel verwendet. Man unterscheidet deshalb zwischen symmetrischer und asymmetrischer Verschlüsselung. Bei der symmetrischen Verschlüsselung wird zum Ver- und Entschlüsseln der gleiche Schlüssel verwendet; bei der asymmetrischen Verschlüsselung sind dies immer verschiedene Schlüssel, die aber zueinander passen müssen. Das Gesetz lässt allerdings offen, welche Art der Verschlüsselung Verwendung finden soll.“ BDSG-Kommentar von Simitis (§ 9, Rn. 166)

~BACKUP