

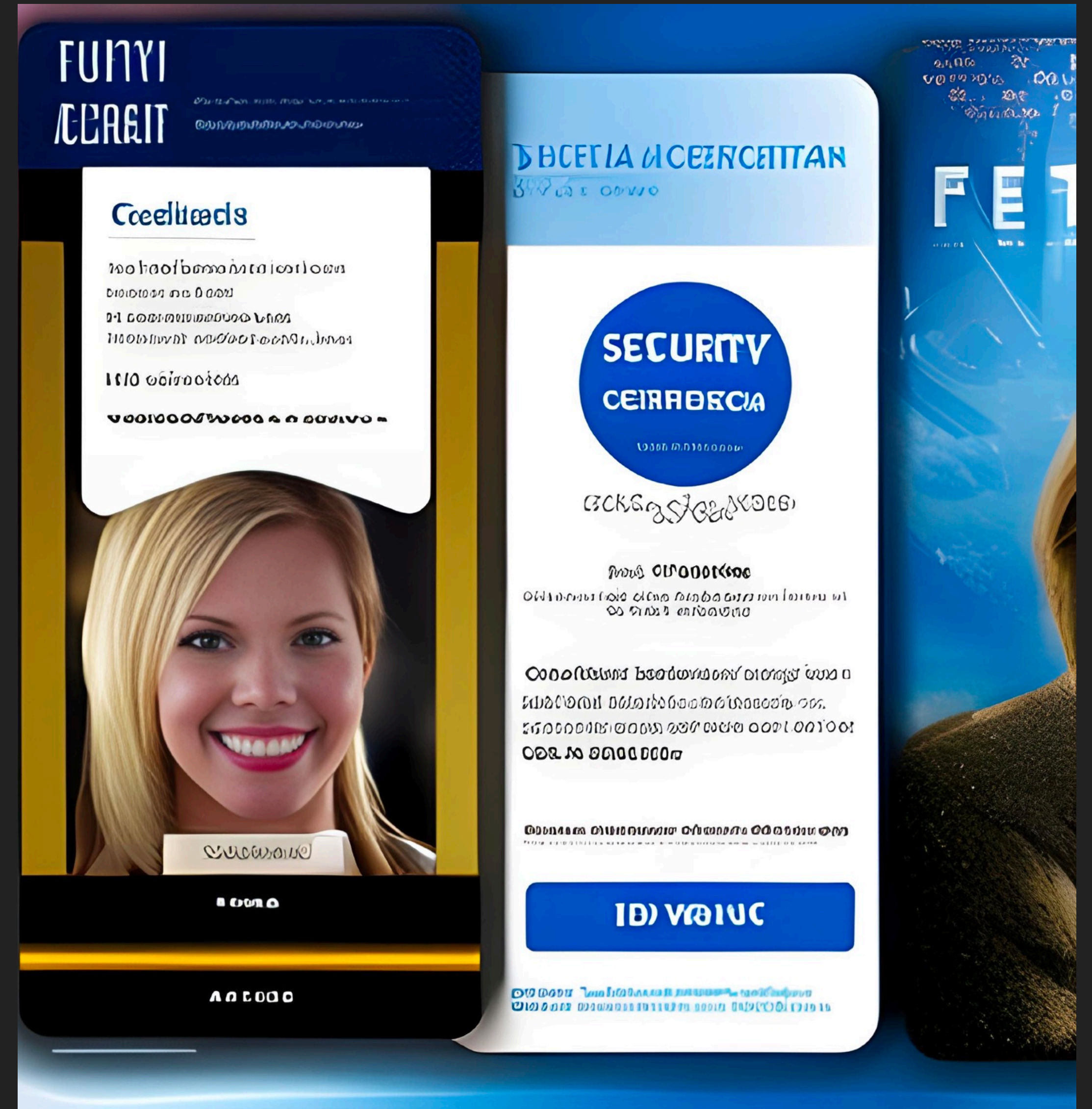


BLOCKCHAIN COMMONS

THE NEXT STEP IN DIGITAL CREDENTIALS

DIGITAL CREDENTIALS ARE A BETTER WAY OF SHARING QUALIFICATIONS

- ▶ They Simplify Administration
 - ▶ Create a credential.
 - ▶ Sign it.
 - ▶ Put Public Keys in a PKI.
 - ▶ You're done!



DIGITAL CREDENTIALS ARE A BETTER WAY OF SHARING QUALIFICATIONS

- ▶ They Simplify Usage
 - ▶ Student can retrieve at will.
 - ▶ It is not necessary for institutions to verify. (The signature does that.)
 - ▶ “No phone home”



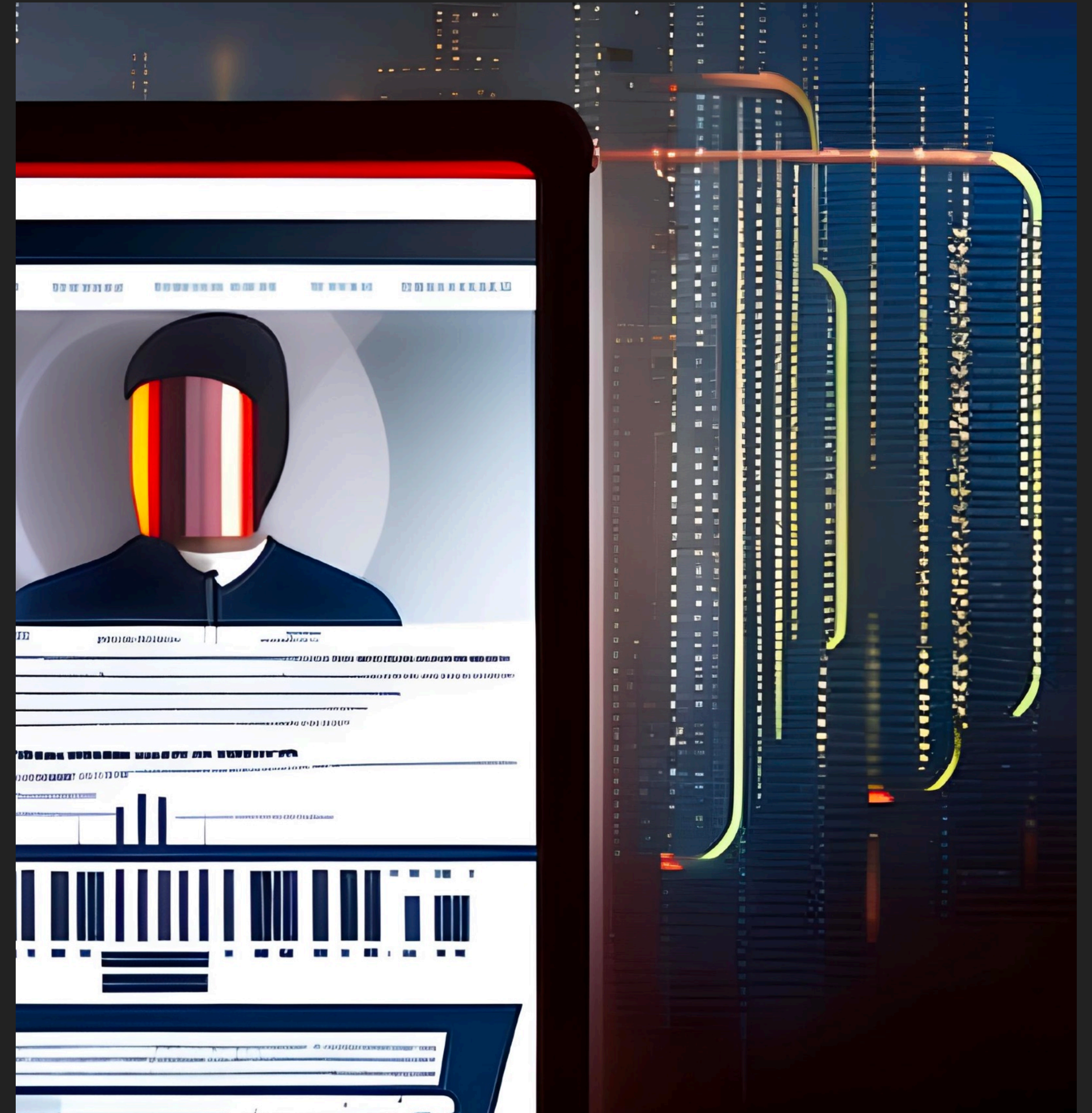
DIGITAL CREDENTIALS CAN BE DANGEROUS TOO!

- ▶ How do you protect student's privacy?
- ▶ How do you protect against discrimination?
- ▶ How do institutions reduce liability, especially with new laws such as GDPR, CCPA, and more to come?



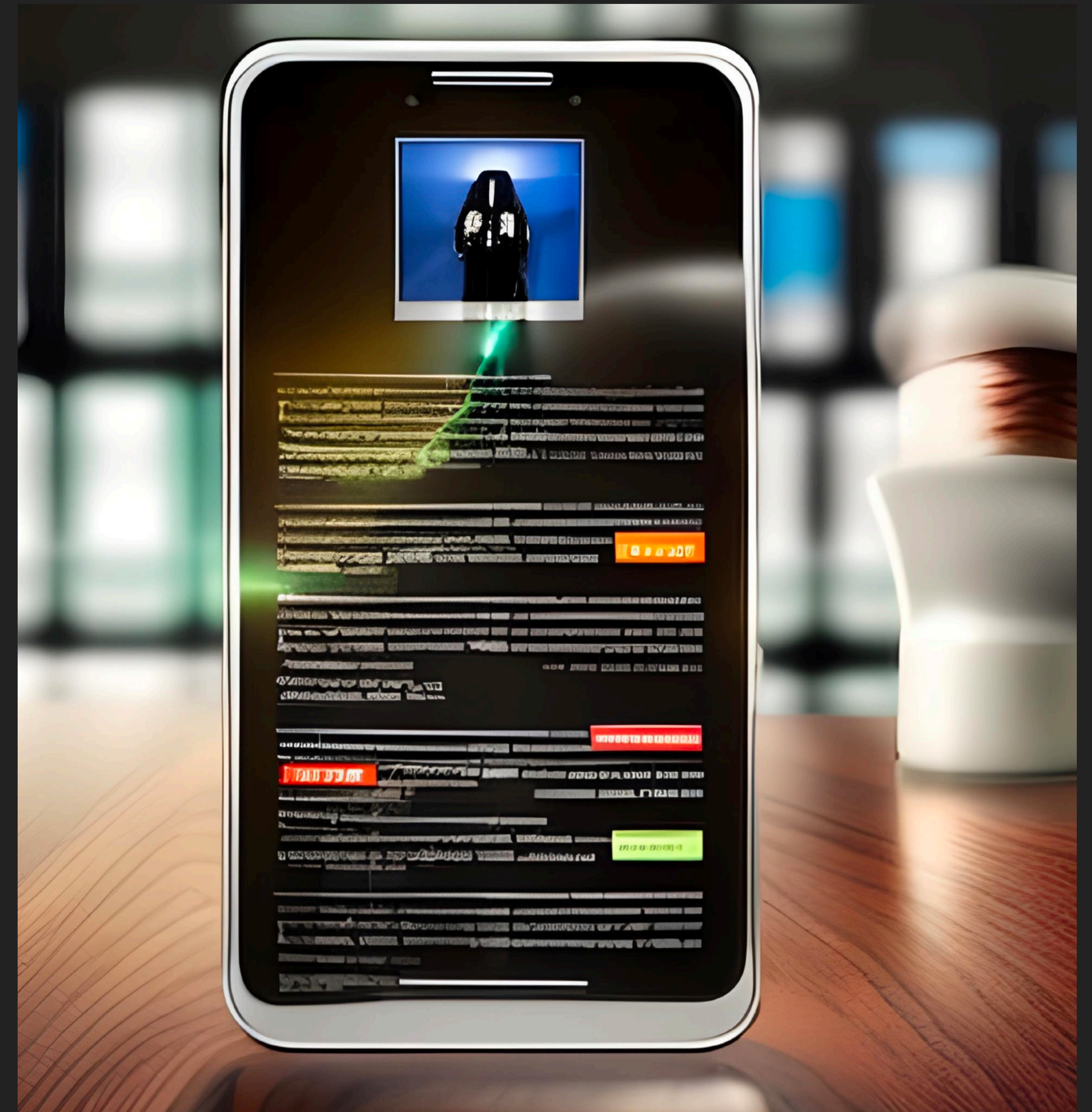
DIGITAL CREDENTIAL PROBLEMS

- ▶ The biggest problem is identity theft.
- ▶ Credentials can contain huge amounts of info!
- ▶ Names, addresses, birthdays, ID #s.
- ▶ These are used as identity questions!
- ▶ But specific data can cause problems too!



DIGITAL DATA PROBLEMS

- ▶ Gender: *gender discrimination*
 - ▶ Especially problematic for students from vulnerable countries & transgender students in US.
- ▶ Name, Birthplace, Address, Issuer location: *racial discrimination*
- ▶ Age, date of credentialing: *age discrimination*
- ▶ Faith-based school info: *religious discrimination*
- ▶ The more data, the more problems!

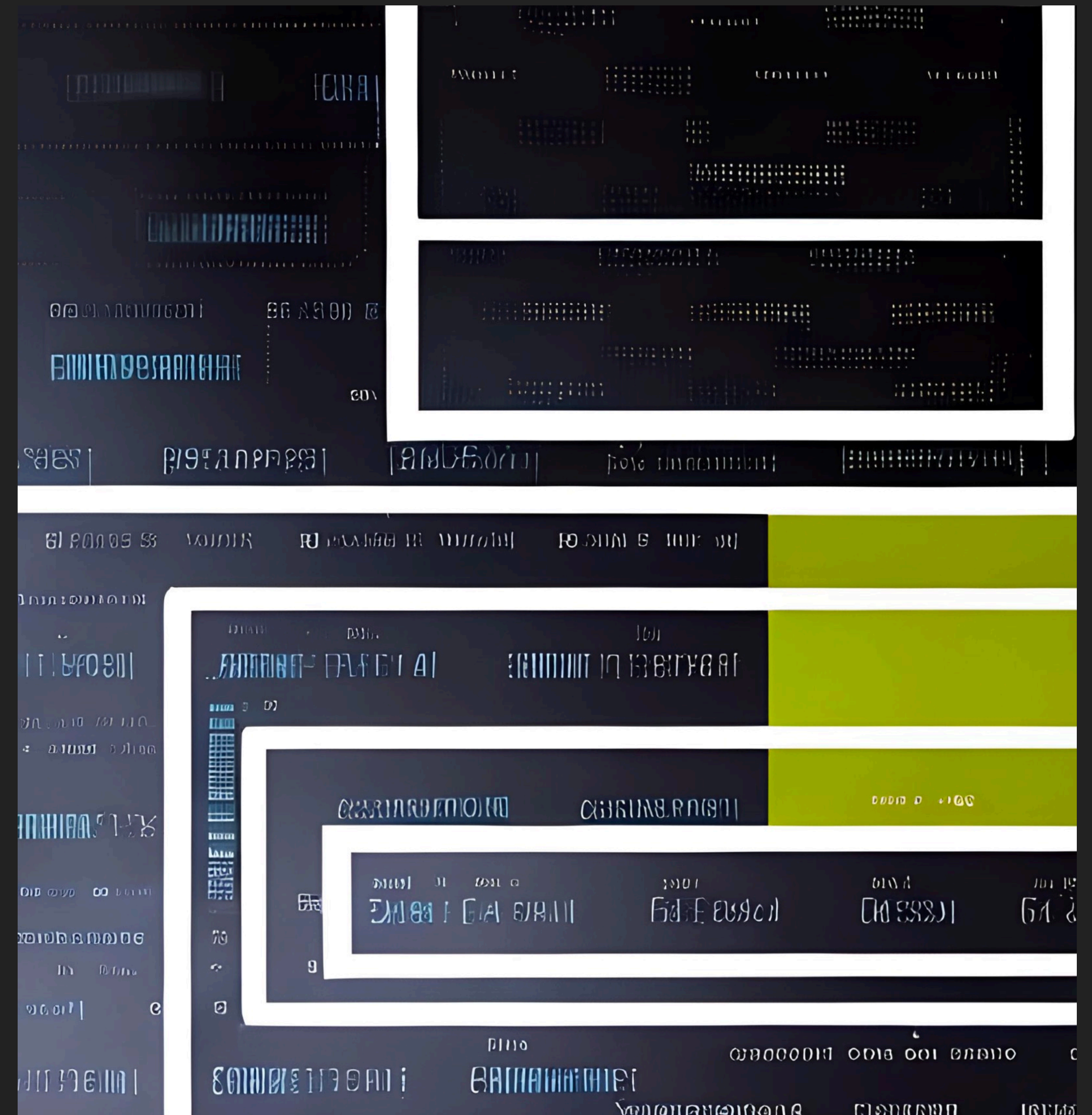




CREATING A SOLUTION: HOLDER-BASED ELISION

HOLDER-BASED ELISION

- ▶ Data-filled credentials shouldn't be out in wild.
- ▶ Let the holder redact information as they see fit.
- ▶ Potentially discriminatory items can be removed.
- ▶ Unnecessary information can be removed.
- ▶ But holder still has the full credential when needed.
(Signatures still verify!)
- ▶ Question of data retention/deletion becomes one for holder – not institute.





IMPLEMENTING THE SOLUTION: HASH-BASED ELISION

WHAT IS A HASH?

- ▶ Like a “data fingerprint”.
- ▶ The smallest change to the data *entirely* changes the hash.
- ▶ They are a fixed size, no matter the size of the input data.
- ▶ Hashes are *one-way*: you can't recover the original data from the hash.
- ▶ They are a long series of numbers, but can be made visual and easily distinguishable with tools like *LifeHash*.

Input:

Hello, world!

Hash:

```
315f5bdb76d078c4
3b8ac0064e4a0164
612b1fce77c86934
5bfc94c75894edd3
```

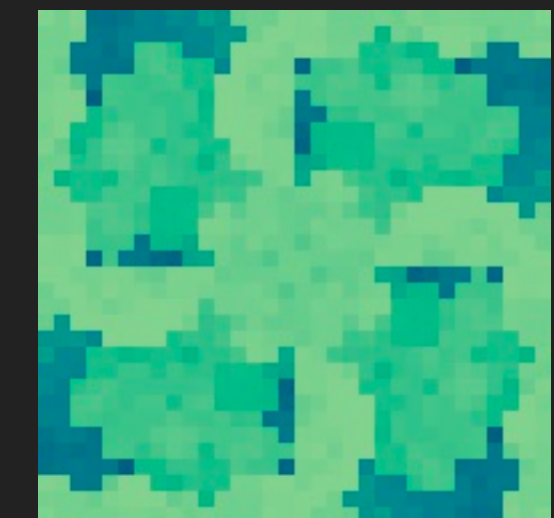


Input:

Hello, Wörld!

Hash:

```
fc759582a2659fd4
a5b4a69be4ada5b2
bb6050c91d9e646b
0b1184abca26bd82
```



HASH-BASED ELISION

- ▶ If you sign a document then remove the data, you can no longer verify the signature.
- ▶ How do we allow the holder to remove data without invalidating signatures?
- ▶ Solution: don't sign the *data*: hash the data and then sign the *hash*!
- ▶ When the data is removed, the hash remains in the document.
- ▶ If the data is restored, verify that its hash matches the hash in the document.

Input:

Hello, world! ← **DON'T SIGN THIS**

Hash:

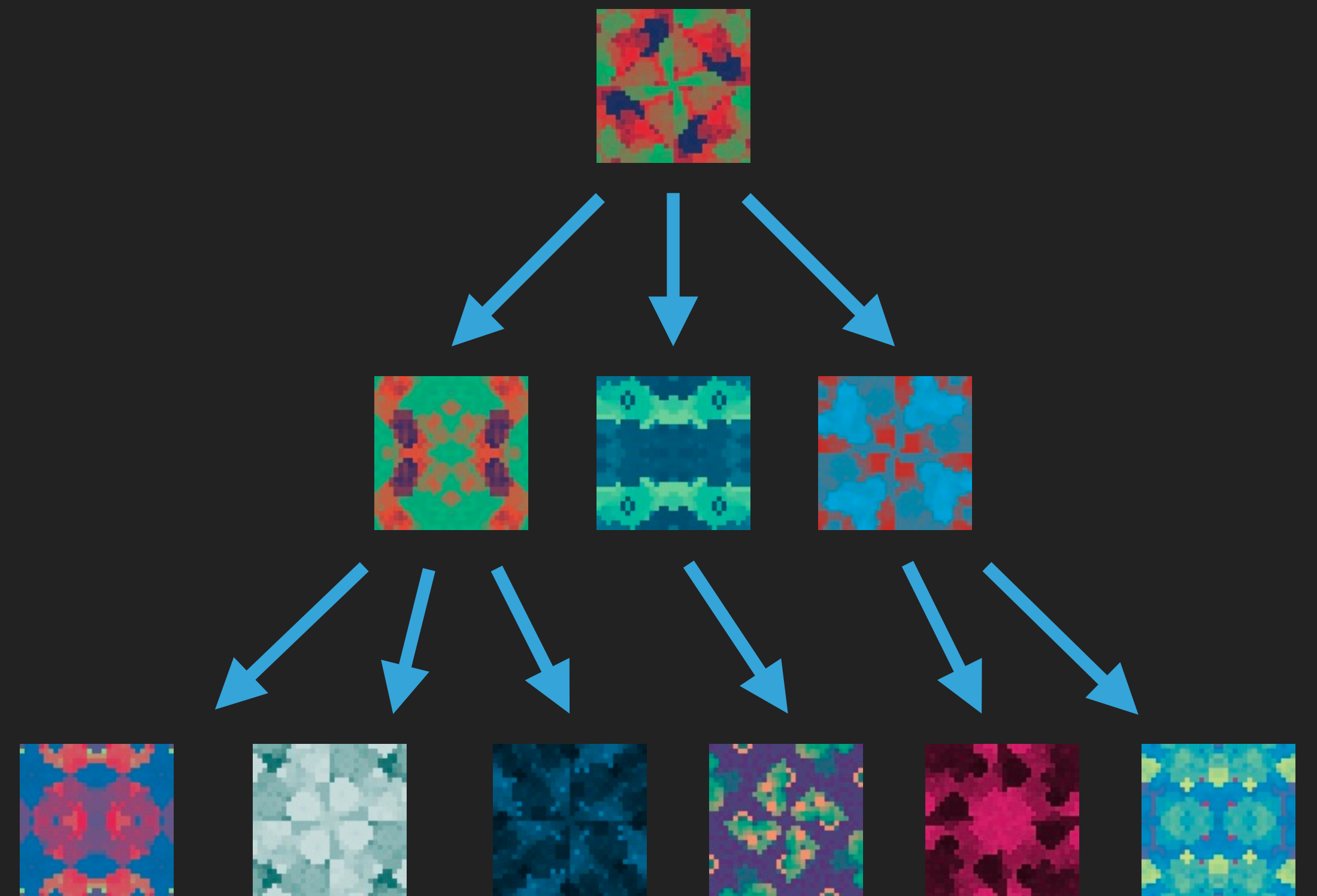
```
315f5bdb76d078c4
3b8ac0064e4a0164
612b1fce77c86934
5bfc94c75894edd3
```



↑
SIGN THIS!

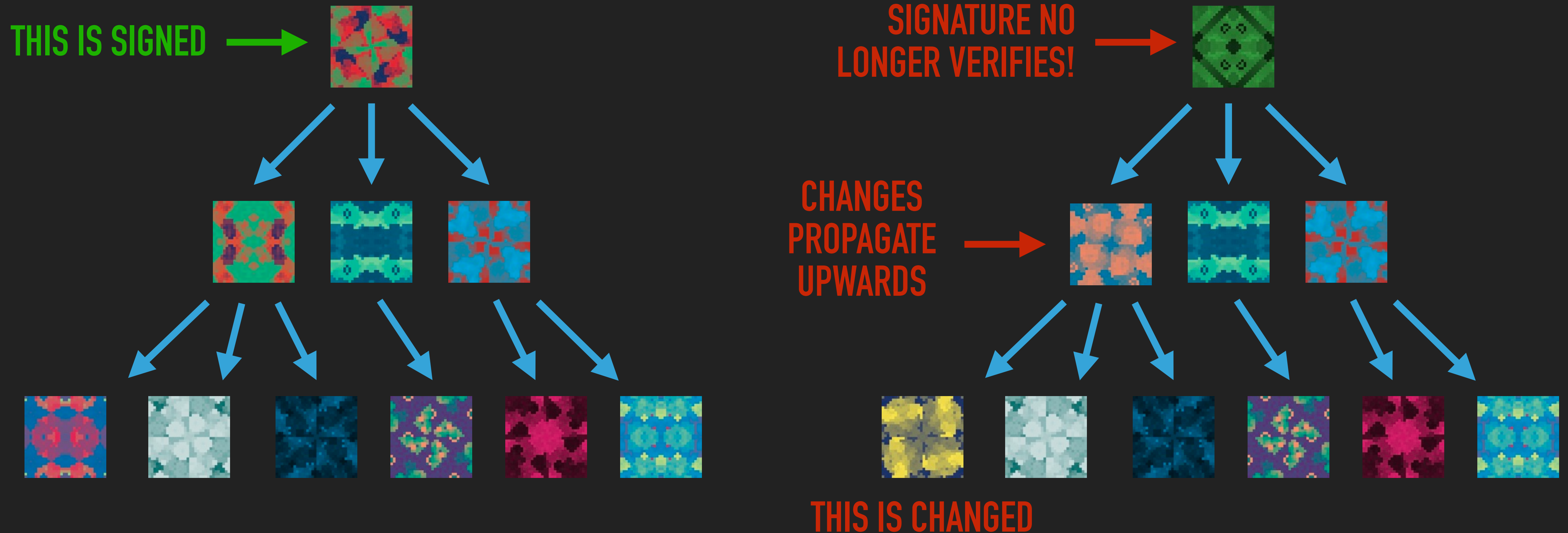
WHAT IS A TREE OF HASHES?

- ▶ Data can be arranged into a tree.
- ▶ All similar data is kept in the same "branch".
- ▶ For a credential, all of a student's Personally Identifiable Information might be in one branch, all of their qualifications in another.
- ▶ The organization continues down from there. This makes it easy to elide specific types of info.
- ▶ The hash tree then matches the structure of the data: each bit of data has its own hash and those hashes combine as data comes together into branches and into the final tree (producing a "root" hash for everything!).
- ▶ Mature technology (*Merkle Tree invented 1979*)



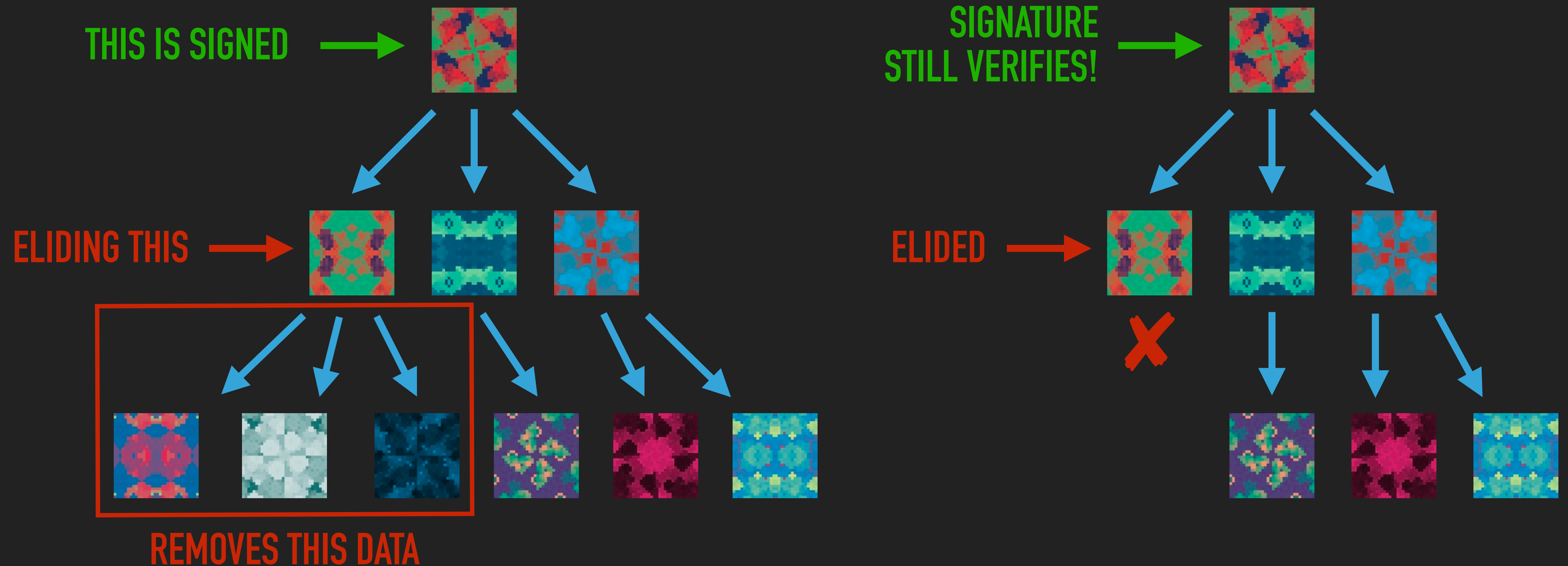
HASH-BASED ELISION

- ▶ If the document is a *tree of hashes*, then any tiny change anywhere in the tree changes the hashes all the way up to the root, invalidating any signatures.



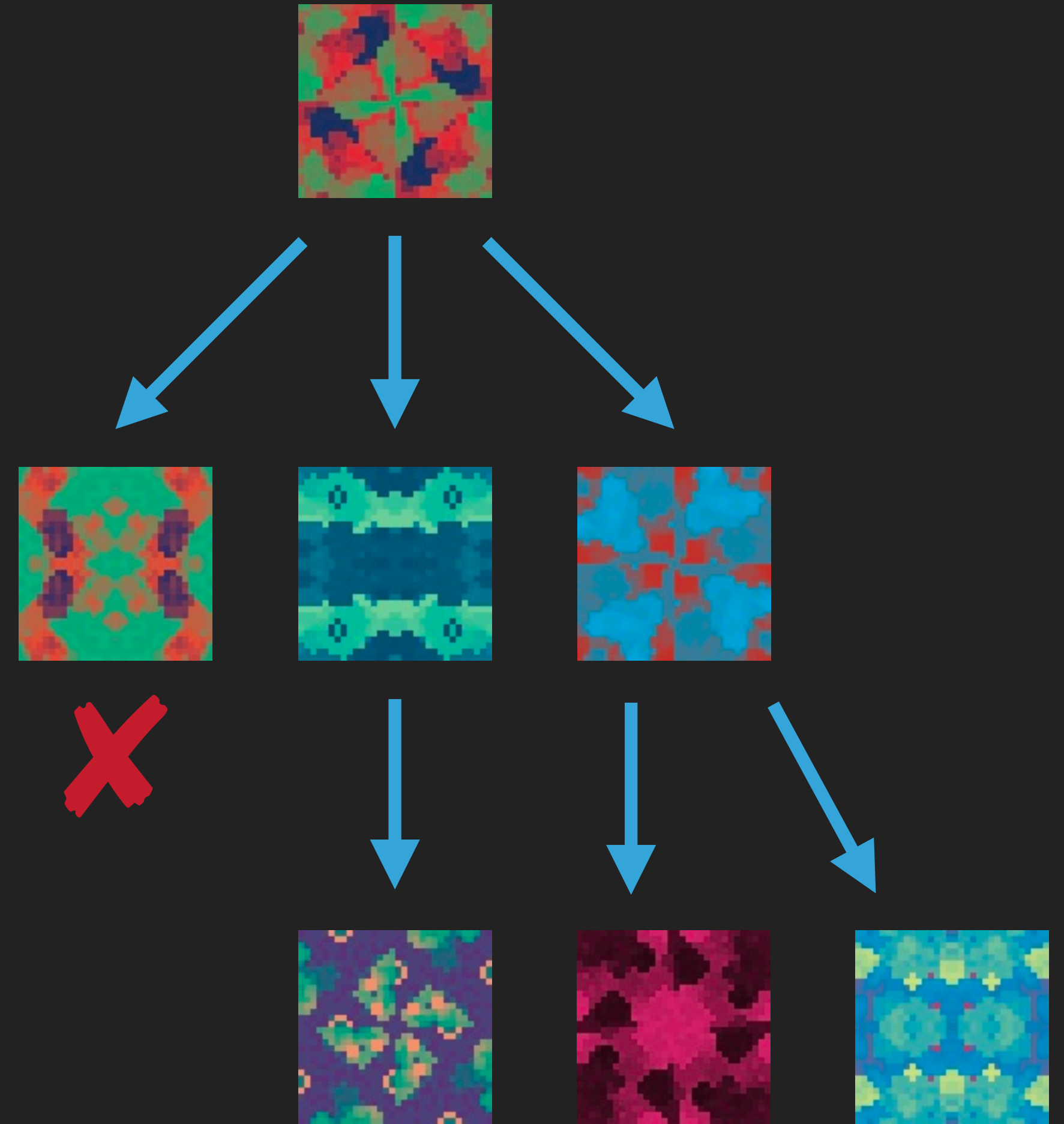
HASH-BASED ELISION

- ▶ If the document is a *tree of hashes*, then any branch can be removed while still verifying all the higher-level signatures.



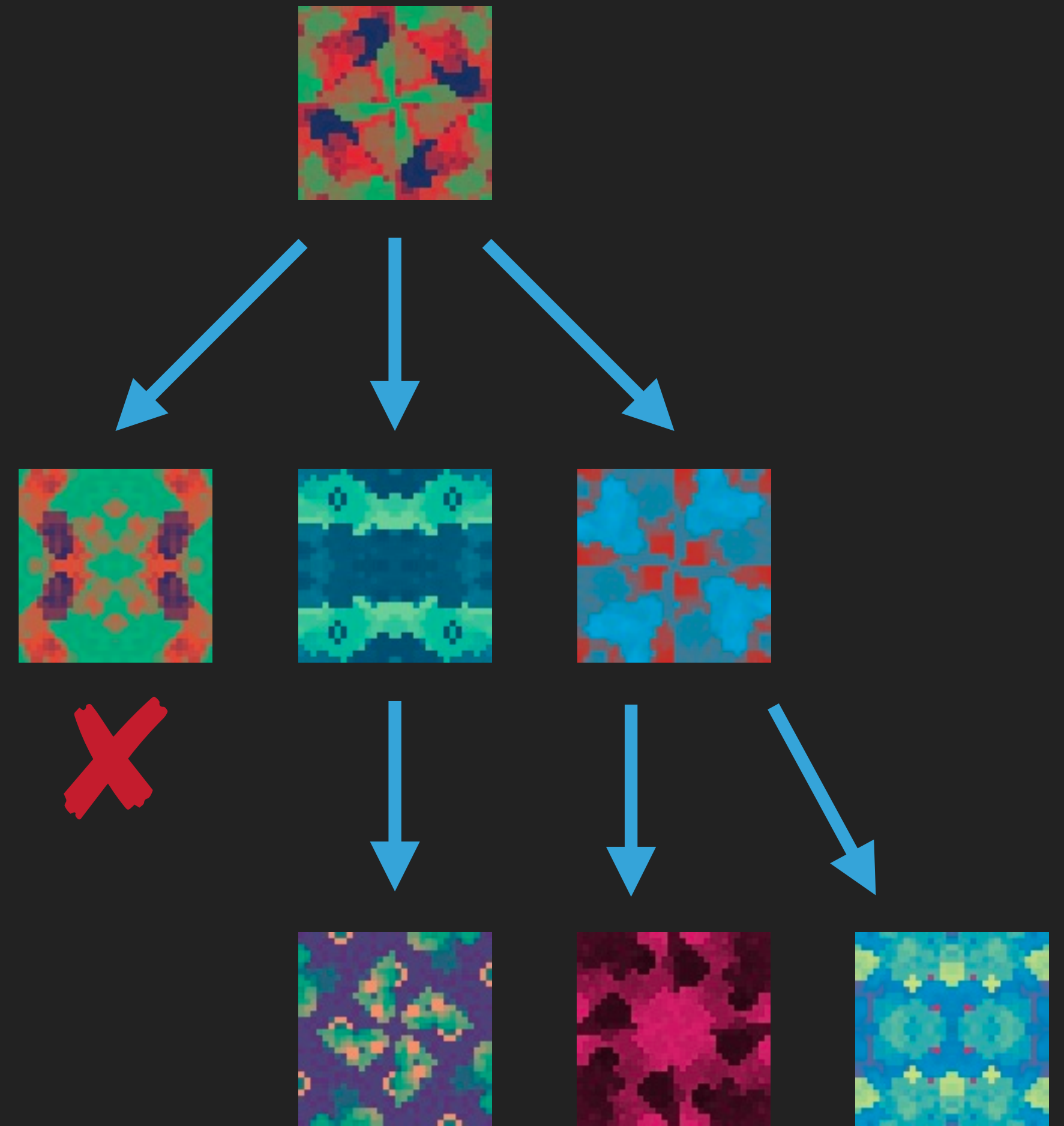
DATA MINIMIZATION: A CORNERSTONE OF PRIVACY

- ▶ The basic rule: reveal what is needed, no more!
- ▶ Requires system of selective disclosure.
- ▶ Holder-based, hash-based elision lets students make all the decisions.



DATA MINIMIZATION: WHY DO WE CARE?

- ▶ We want meaningful credentials BUT ...
- ▶ We want to protect students & their future.
- ▶ We want to protect vulnerable populations.
 - ▶ Students are particularly vulnerable!
 - ▶ Young, away from home & support systems.
- ▶ We value diversity & want to protect it.



DATA MINIMIZATION: HOW IT HELPS INSTITUTIONS

- ▶ They don't have admin of eliding credentials.
- ▶ They don't have liability of overfull credentials.
- ▶ They don't have responsibility for GDPR, et al.
- ▶ Responsibility is transferred to holder.



DATA MINIMIZATION: INSTITUTIONAL COMPLIANCE

- ▶ Elision can protect institutions from violating laws!
 - ▶ **FERPA:** Prohibits transmitting student PII in US, with wide exceptions.
 - ▶ **PPRA:** Defines protected data areas (e.g., religion, income, etc.) that could be compromised in credentials.
 - ▶ **GDPR:** European law with stringent rules about data collection & distribution.
 - ▶ **CCPA:** Californian equivalent of GDPR, with some variations.
- ▶ It's a lot! Data minimization can provide compliance for multiple rules & regulations.



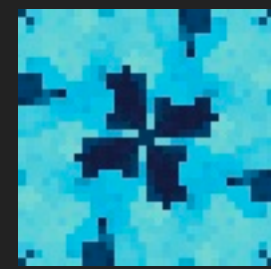
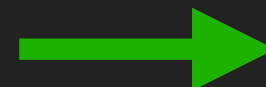


THERE'S MORE...

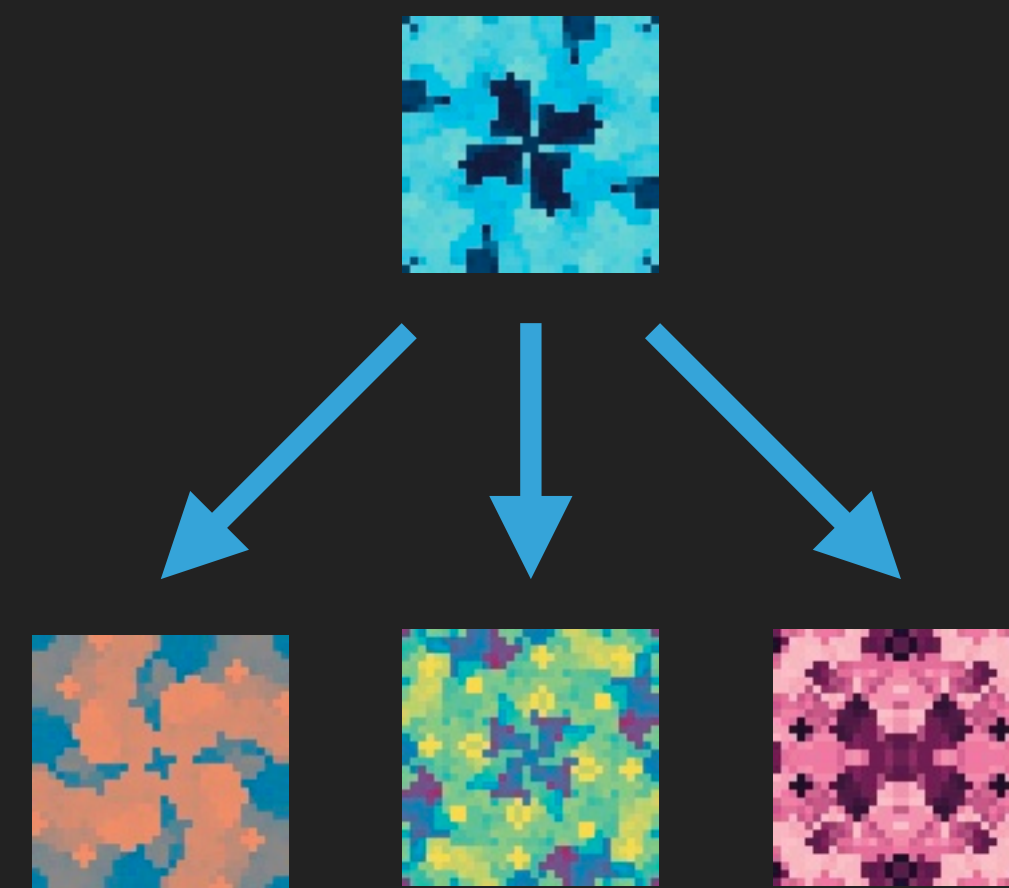
PROOF OF INCLUSION

- ▶ The institution can publish just a signed root hash with no other information.
- ▶ Later they can prove certain information exists in the document by providing just the necessary hashes.

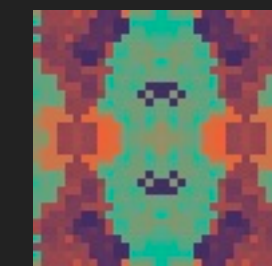
HASH AND SIGNATURE
MADE PUBLIC



ONLY NECESSARY
HASHES REVEALED
FOR PROOF

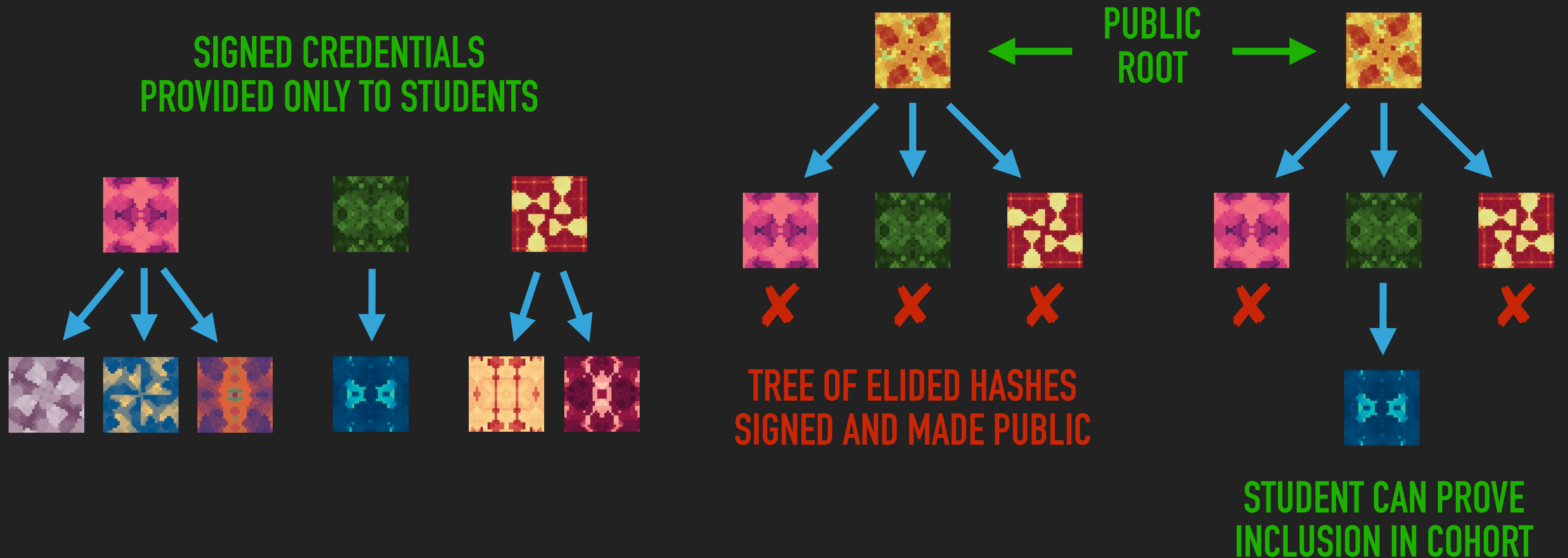


THE REVEALED
DOCUMENT



HERD PRIVACY

- ▶ The institution can give each student their credential, and publish a tree of elided hashes, one for each credential in the cohort.
- ▶ Provides proof that the student graduated with their cohort.

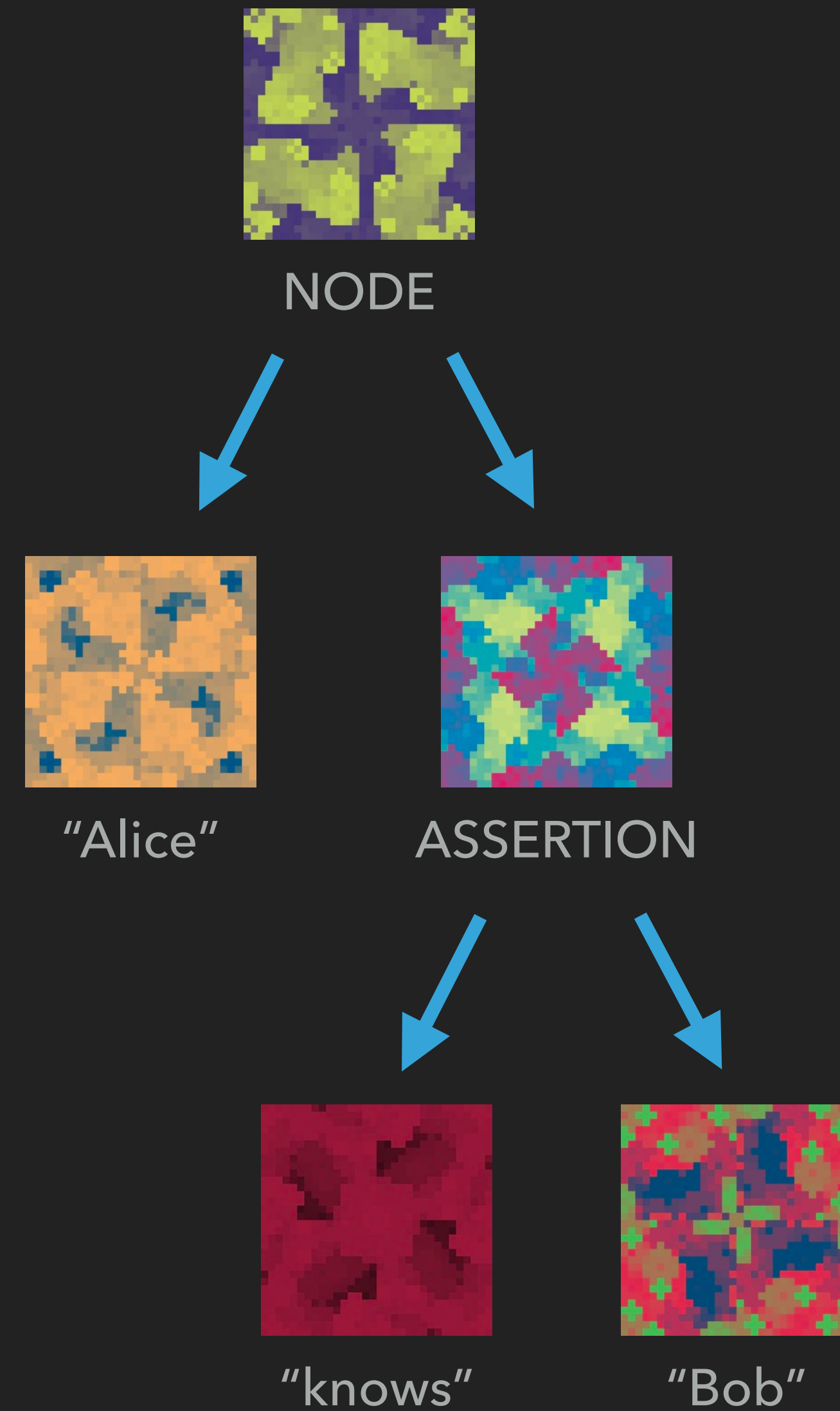




HASH-BASED ELISION WITH GORDIAN ENVELOPE

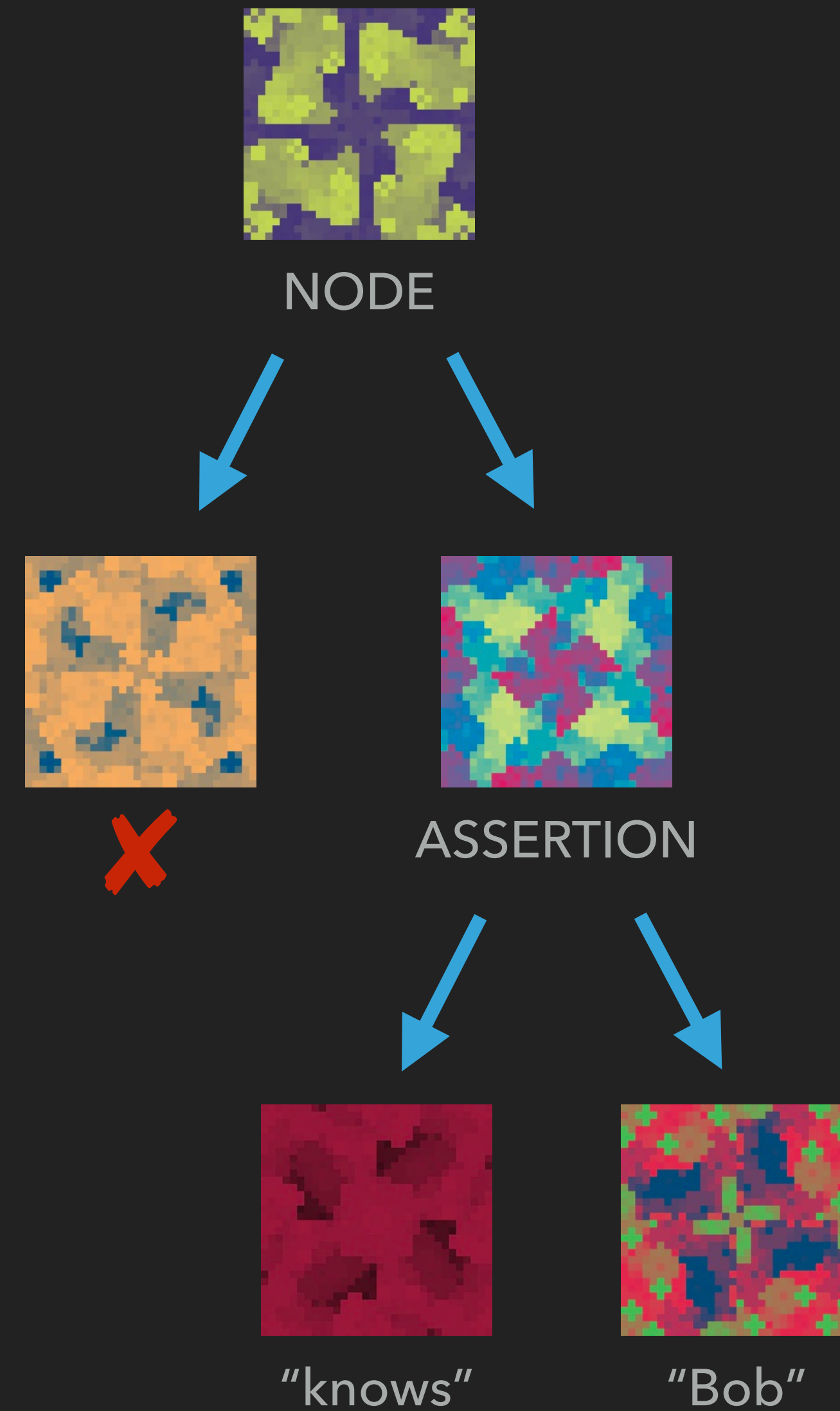
GORDIAN ENVELOPE

```
"Alice" [  
  "knows": "Bob"  
]
```



GORDIAN ENVELOPE

```
ELIDED [
  "knows": "Bob"
]
```

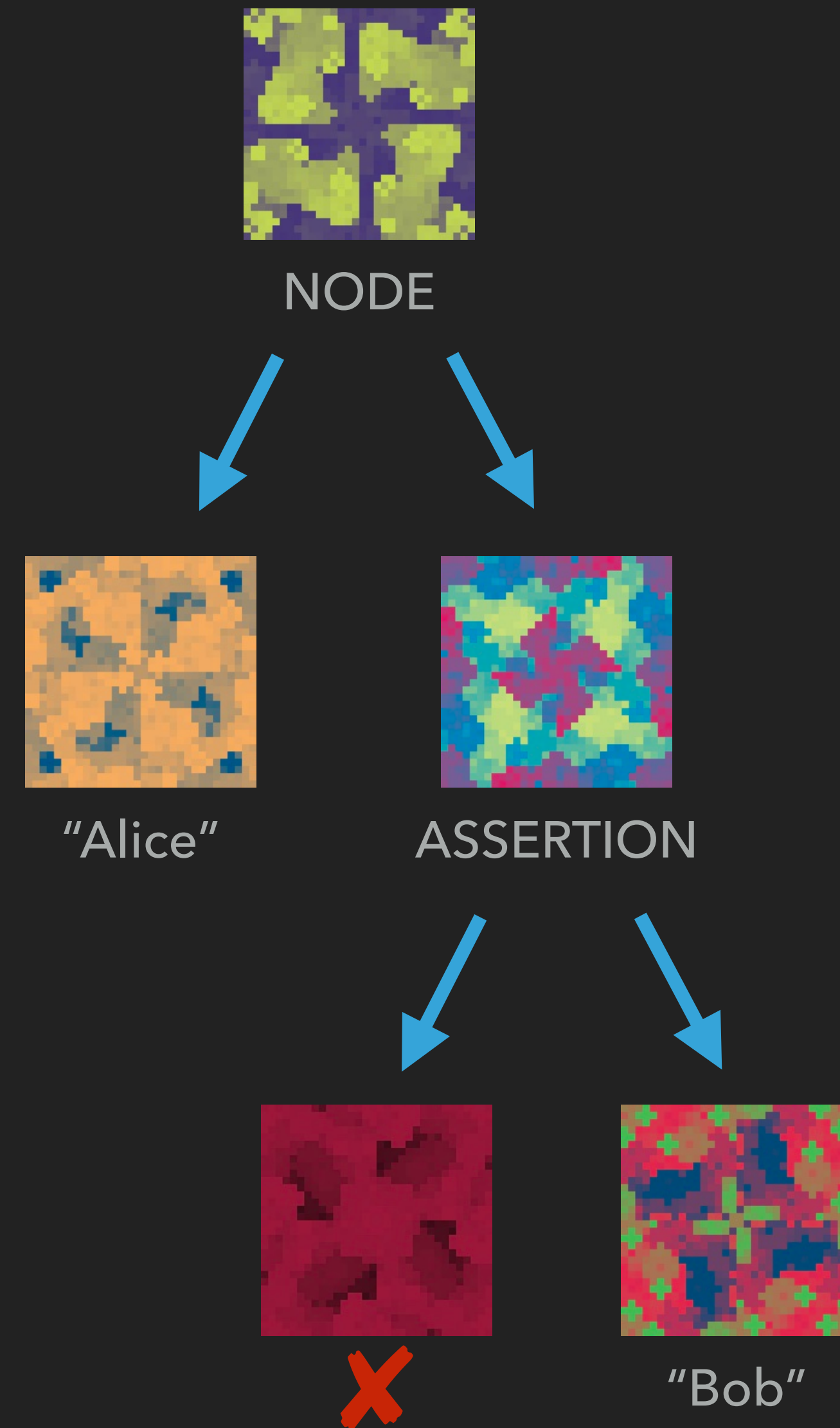


1 - SUBJECT ELIDED



GORDIAN ENVELOPE

```
"Alice" [
  ELIDED: "Bob"
]
```

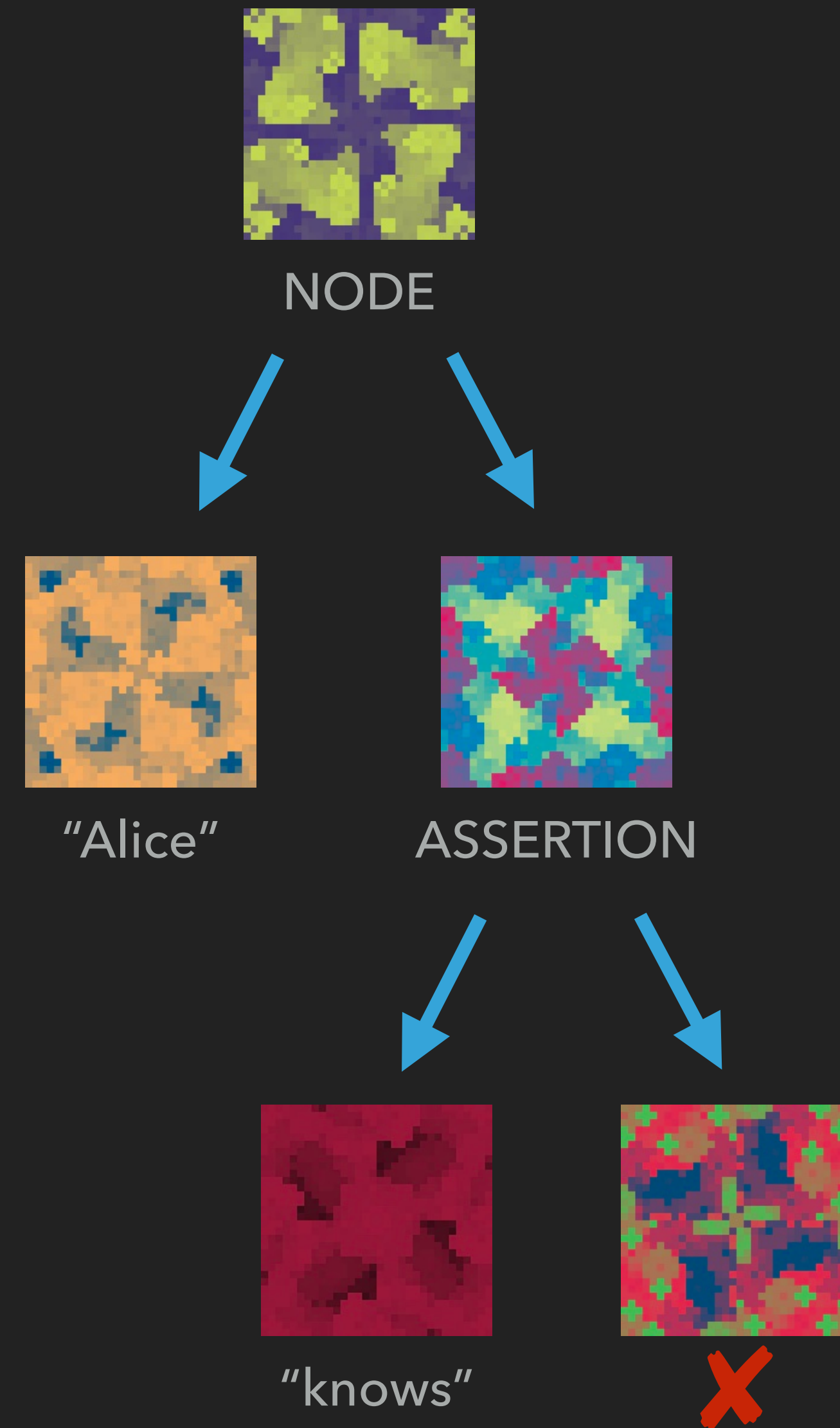


2 - PREDICATE ELIDED



GORDIAN ENVELOPE

```
"Alice" [  
  "knows": ELIDED  
]
```

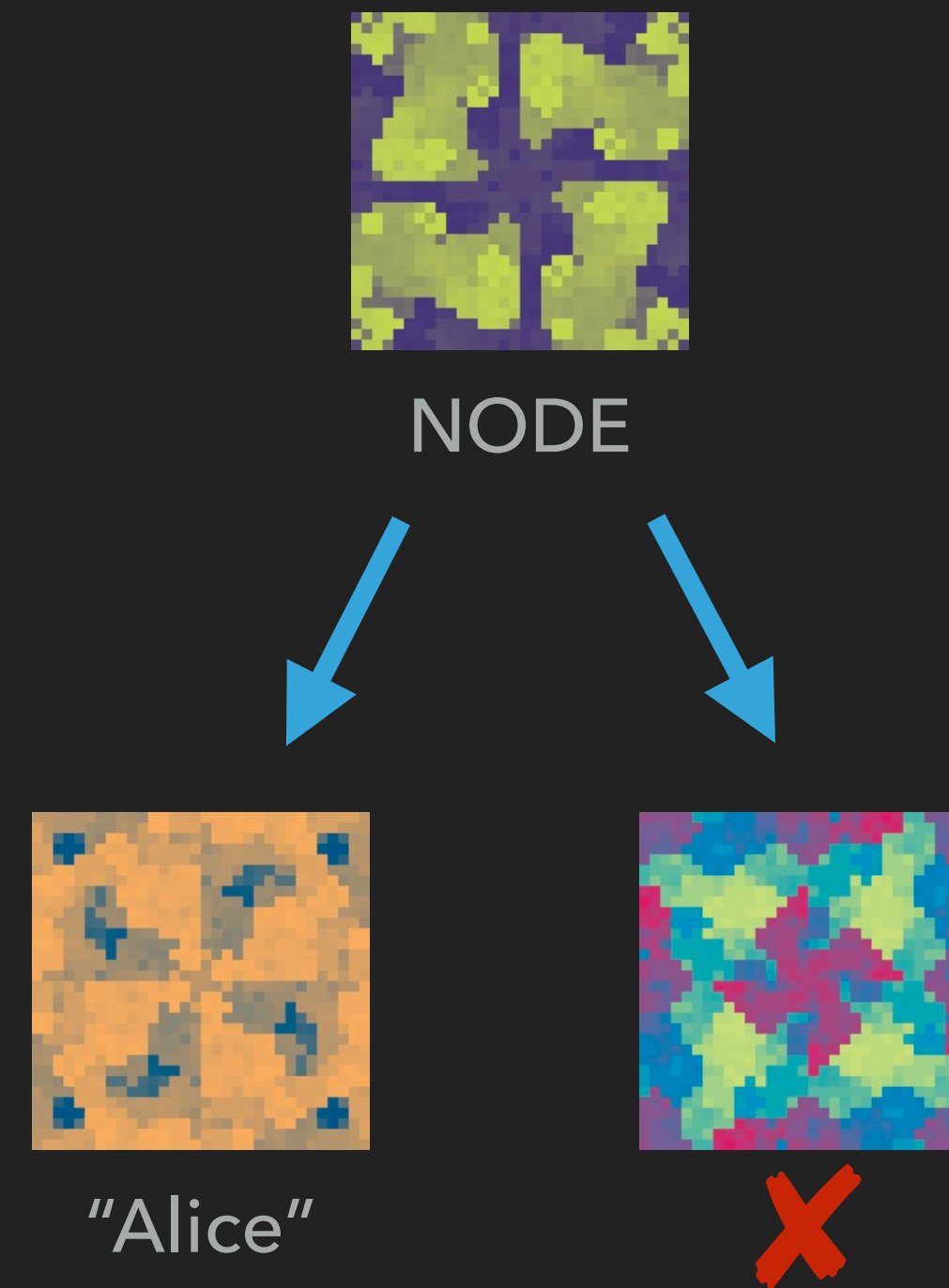


3 - OBJECT ELIDED



GORDIAN ENVELOPE

```
"Alice" [
  ELIDED
]
```



GORDIAN ENVELOPE







ELIDED

5 - ENVELOPE ELIDED



EMERGING ELISION SPECS

Spec		Pros	Cons
SD-JWT (ietf)		Leverage VC-JWT ecosystem, related to ISO mDDL/mDOC; does not require schemas	Hash lists not tree (only elide whole claim from a list); other JWT limitations
LD Merkle Disclosure (w3c)		Leverage VC-JSON-LD ecosystem; node graph data	Hash lists not tree (only elide whole claims from a list); requires node graph structure & schema
Gordian Envelope (ietf)		Data structure agnostic (graphs + lists + schema or no-schema); offers 5-kinds of redaction, inclusion proofs, herd privacy, encryption, compression, secret sharing	Not W3C-VC centric (useful for many other purposes including DIDs and other data); not currently accepted on standards track
BBS+ Signature (ietf)		Powerful anti-correlation of signatures, only offers proof of knowledge of the undisclosed signature	Not hash-based, uses new cryptography (2006); holder-based elision scenarios more complicated

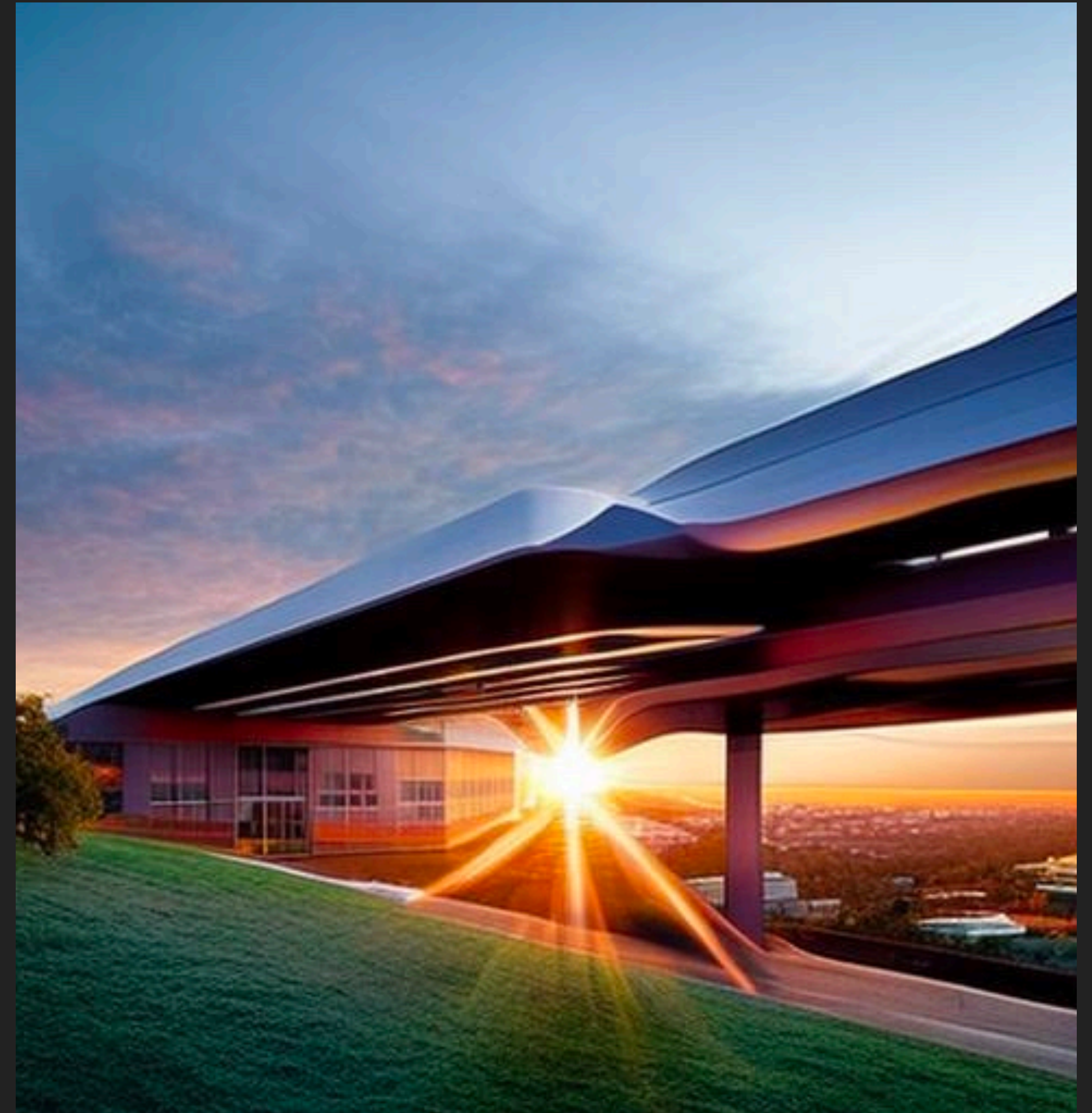
FINAL NOTES

- ▶ Digital credentials are powerful.
- ▶ But simple credentials don't protect privacy.
 - ▶ Holder & issuer both at risk!
 - ▶ Transient, can be lost, too much info!
- ▶ Strong, safe credentials NEED ...
 - ▶ Control by holder.
 - ▶ Ability to elide.
 - ▶ Maintenance of signatures through hashing.
 - ▶ Proofs for further data minimization.



A CALL TO ACTION

- ▶ Holder-based elision is crucial for privacy.
- ▶ We need to turn MAYS & SHOULD be into MUST.
 - ▶ Data Minimization as a REQUIREMENT.
 - ▶ User Control as a REQUIREMENT.
- ▶ We'd like you to use Gordian Envelope
 - ▶ Useful features such as encryption, inclusion proofs & herd privacy, etc.
- ▶ But if not, please use another emerging spec!



FOR MORE ON GORDIAN

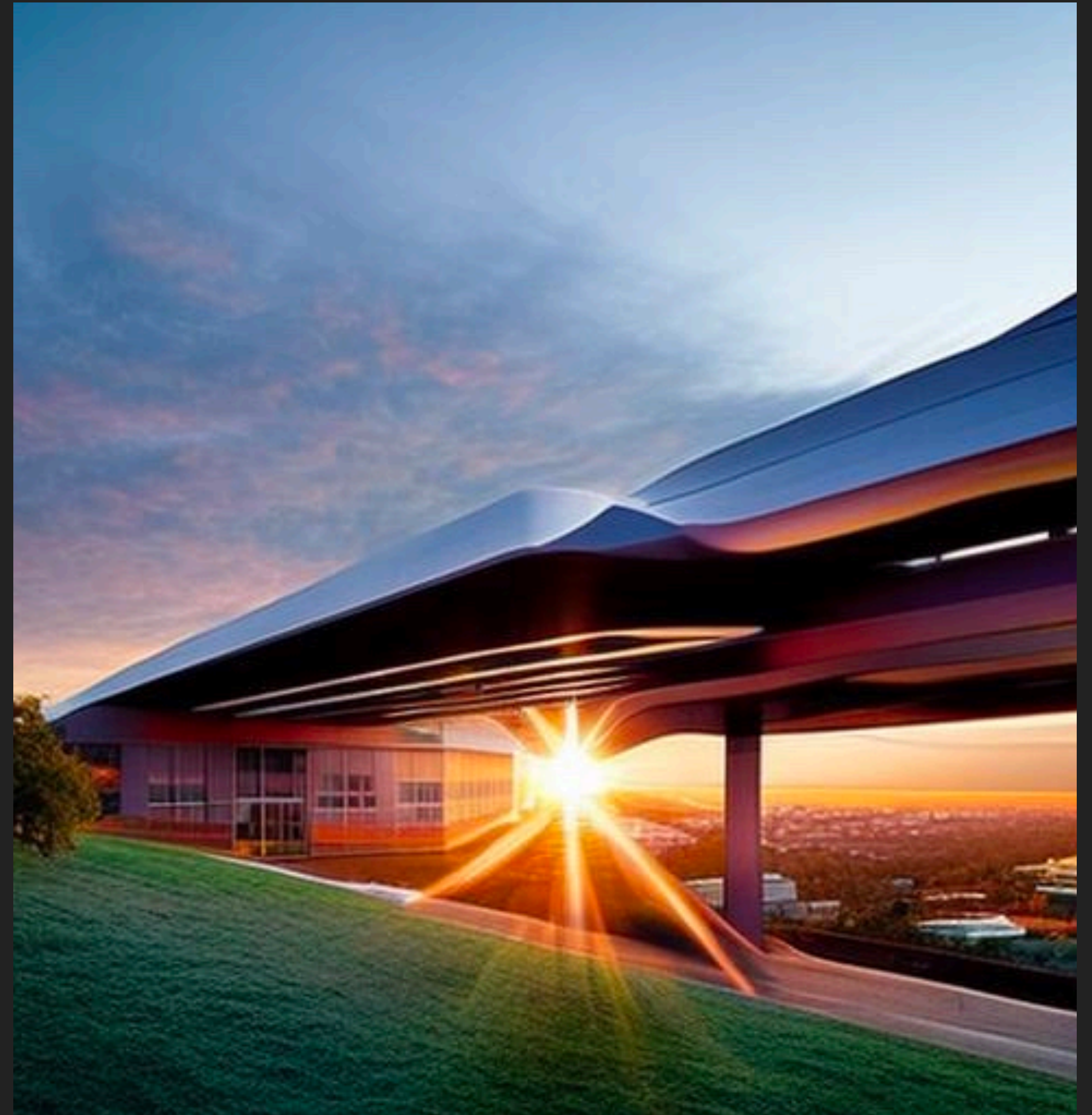
- ▶ Read Gordian Envelopes intro:
- ▶ <https://tinyurl.com/gordian-envelope>



- ▶ Watch Gordon Envelope videos:
- ▶ <https://tinyurl.com/gordian-videos>



- ▶ Read Educational use cases:
- ▶ <https://tinyurl.com/gordian-educational>



CHRISTOPHER ALLEN

christophera@lifewithalacrity.com



@BlockchainComms

WOLF MCNALLY

wolf@wolfmcnally.com



@WolfMcNally

