

ZERO-KNOWLEDGE PROOFS IN WEB3

Janno Siim



MY BACKGROUND

1

2

3

2

MY BACKGROUND

1

Full academia

Phd, Postdoc, now Lecturer (assistant professor)

2

3

MY BACKGROUND

1

Full academia

Phd, Postdoc, now Lecturer (assistant professor)

2

Research in (practical) ZKPs and SNARKs

3

MY BACKGROUND

1

Full academia

Phd, Postdoc, now Lecturer (assistant professor)

2

Research in (practical) ZKPs and SNARKs

- Around 10 years

3

MY BACKGROUND

1

Full academia

Phd, Postdoc, now Lecturer (assistant professor)

2

Research in (practical) ZKPs and SNARKs

- Around 10 years
- **Focus:** security assumptions, security properties, protocol design

3

MY BACKGROUND

1

Full academia

Phd, Postdoc, now Lecturer (assistant professor)

2

Research in (practical) ZKPs and SNARKs

- Around 10 years
- **Focus:** security assumptions, security properties, protocol design

3

Applications in Web3

2

MY BACKGROUND

1

Full academia

Phd, Postdoc, now Lecturer (assistant professor)

2

Research in (practical) ZKPs and SNARKs

- Around 10 years
- **Focus:** security assumptions, security properties, protocol design

3

Applications in Web3 ²

- Never worked in Web3 industry

MY BACKGROUND

1

Full academia

Phd, Postdoc, now Lecturer (assistant professor)

2

Research in (practical) ZKPs and SNARKs

- Around 10 years
- **Focus:** security assumptions, security properties, protocol design

3

Applications in Web3 ²

- Never worked in Web3 industry
- Important for research relevance



WHAT THE HECK IS WEB3?

3



WHAT THE HECK IS WEB3?

3

Is it just a buzzword?

WHAT THE HECK IS WEB3?

3

Is it just a buzzword?



WHAT THE HECK IS WEB3?

3

Is it just a buzzword?



Partly, yes

WHAT THE HECK IS WEB3?

3

Is it just a buzzword?



Partly, yes

Is there promising tech?

WHAT THE HECK IS WEB3?

3

Is it just a buzzword?



Partly, yes

Is there promising tech?



WHAT THE HECK IS WEB3?

3

Is it just a buzzword?



Partly, yes

Is there promising tech?



Also, yes

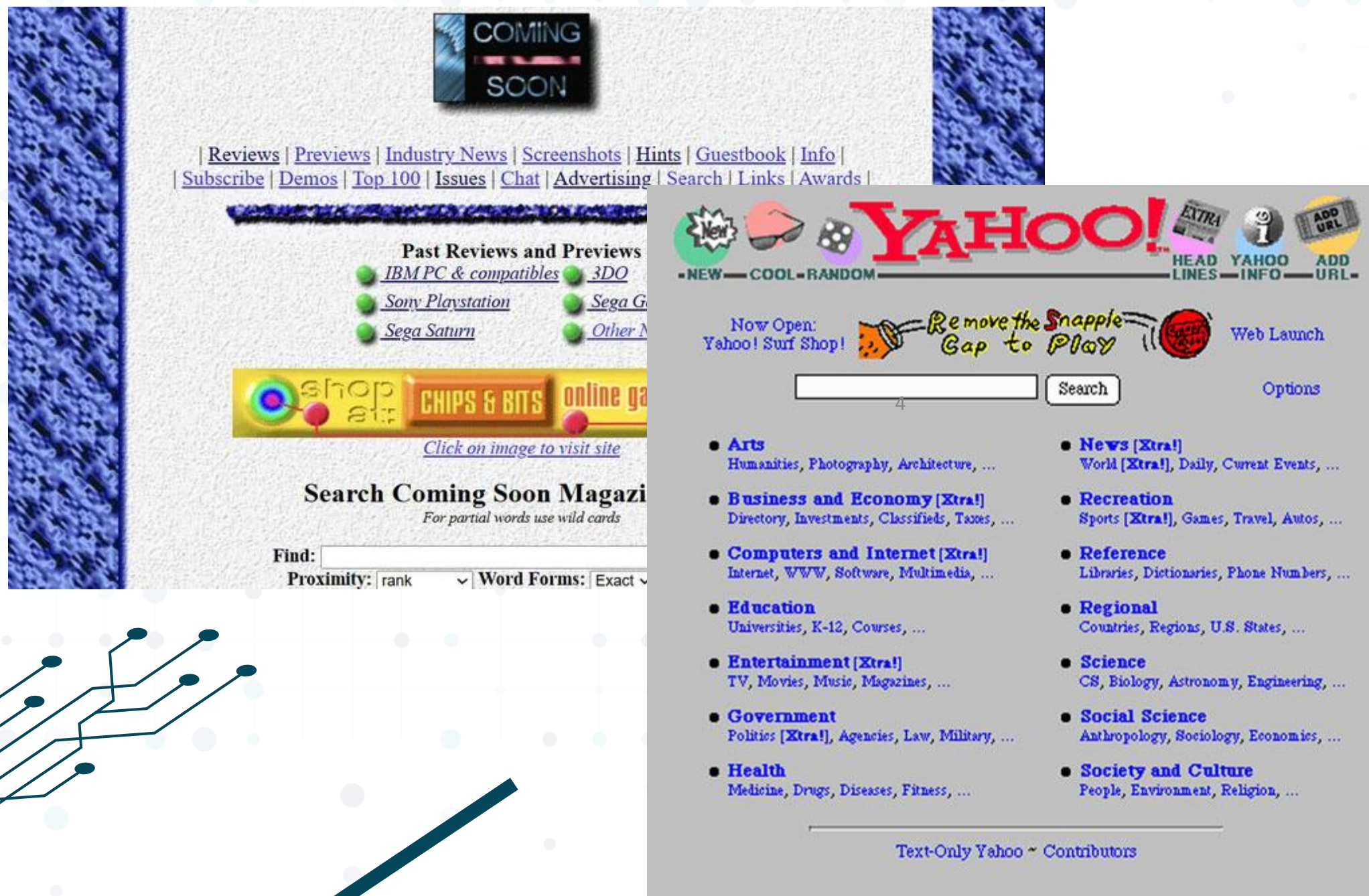
WEB 1.0: 90S, EARLY 2000S

4

WEB 1.0: 90S, EARLY 2000S



WEB 1.0: 90S, EARLY 2000S



WEB 1.0: 90S, EARLY 2000S

COMING SOON

[Reviews](#) | [Previews](#) | [Industry News](#) | [Screenshots](#) | [Hints](#) | [Guestbook](#) | [Info](#) |
[Subscribe](#) | [Demos](#) | [Top 100](#) | [Issues](#) | [Chat](#) | [Advertising](#) | [Search](#) | [Links](#) | [Awards](#)

Past Reviews and Previews
[IBM PC & compatibles](#) | [3DO](#)
[Sony Playstation](#) | [Sega Game Gear](#)
[Sega Saturn](#) | [Other N64](#)

shop at CHIPS & BITS online game store
Click on image to visit site

Search Coming Soon Magazine
For partial words use wild cards

Find:
Proximity: rank Word Forms: Exact

YAHOO!

Now Open: Yahoo! Surf Shop!

Remove the Snapple Gap to Play

Search

- Arts
Humanities, Photography, Architecture, ...
- Business and Economy [Xtra!]
Directory, Investments, Classifieds, Taxes, ...
- Computers and Internet [Xtra!]
Internet, WWW, Software, Multimedia, ...
- Education
Universities, K-12, Courses, ...
- Entertainment [Xtra!]
TV, Movies, Music, Magazines, ...
- Government
Politics [Xtra!], Agencies, Law, Military, ...
- Health
Medicine, Drugs, Diseases, Fitness, ...
- News [Xtra!]
World [Xtra!], Daily, Current Events, ...
- Recreation
Sports [Xtra!], Games, Travel, ...
- Reference
Libraries, Dictionaries, Phone Numbers, ...
- Regional
Countries, Regions, U.S. States, ...
- Science
CS, Biology, Astronomy, Engineering, ...
- Social Science
Anthropology, Sociology, Economics, ...
- Society and Culture
People, Environment, Religion, ...

Text-Only Yahoo ~ Contributors

Abi Reklaam Puu Uued Webi uudised Lisa URL

NETI

Ära kunagi lase arvutil aru saada, et sul on kiire

TALLINN

Kataloog

- Riik ja Ühiskond
Õigus, Ministeeriumid, Mittetulundus-, Regioonid, President ja Riigikogu, ...
- Haridus ja Kultuur
Haridus, Kõrgkoolid, Kirjandus, Teater, Õppematerjalid, Ajalugu, ...
- Teadus
Õigusaktid, Ilm, Sõnastikud, Psühholoogia, Meditsiin, Bioloogia, ...
- Äri
Rahandus, @pank, Kinnisvara, Ehitus, Personaliotsing, Turism, Kaardid, Autod, ...
- Meedia
Ajalehed, Ajakirjad, Televisioon, SAT-TV, Raadio, Uudisteagentuurid, ...
- Arvutid ja Internet
Kataloogid, WWW abi, Tarkvara, Riistvara, Interneti ülevaated, ...
- Meelelahutus ja Hobid
Mängud, Jututoad, Täiskasvanutele, Toitlustus, Sport, Muusika, Horoskoop, ...
- Varia
Kuulutused, Erasisikud, Kirjasõbrad, Aadressid ja Telefonid, Küstlused, ...

Klaver Serverid Puu Üllatus Top100 Uued Webi uudised Lisa URL

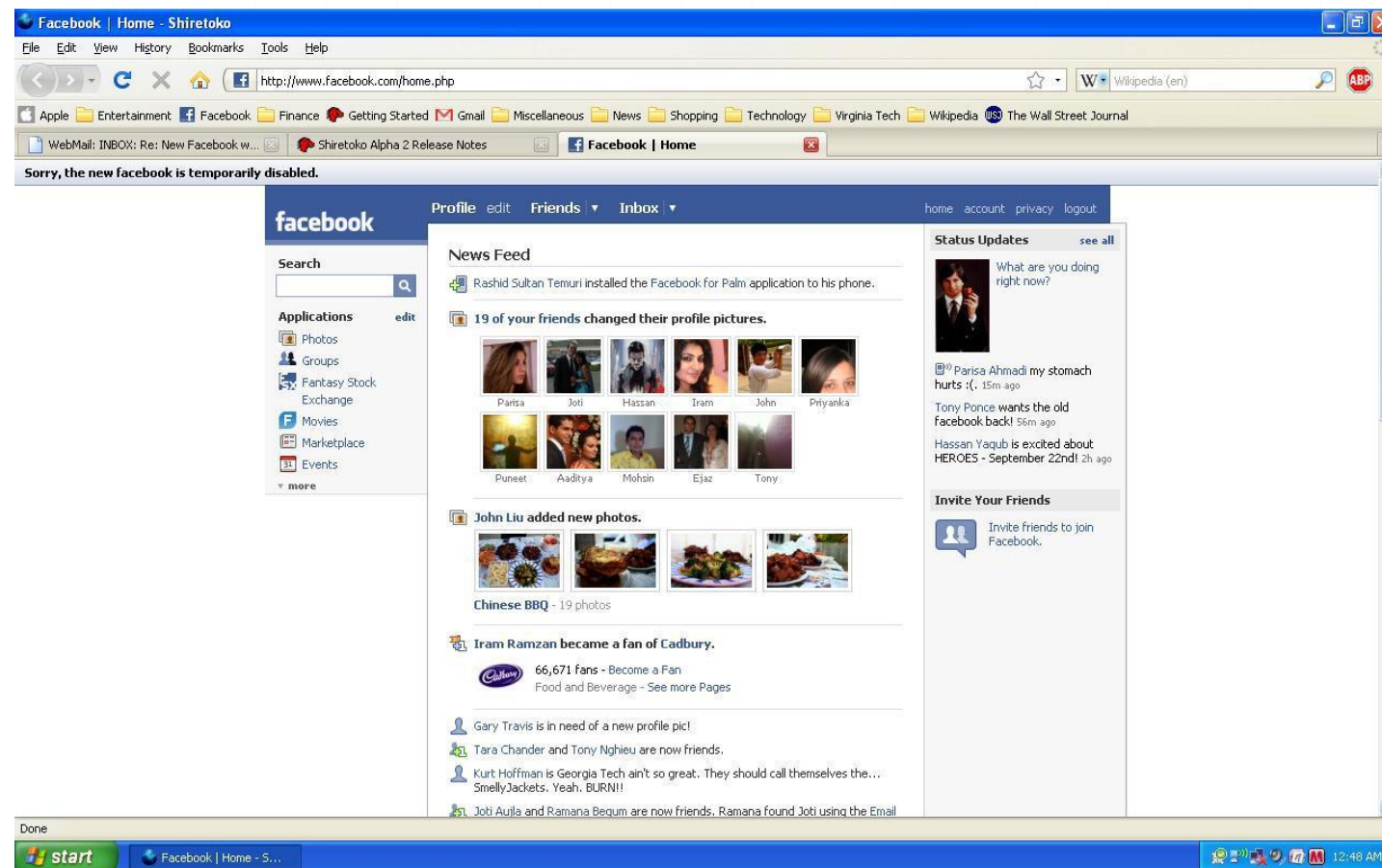
Copyright © 1996 - 2003 Elion

reklaam | webmaster

WEB 2.0: MID 2000S TO NOW

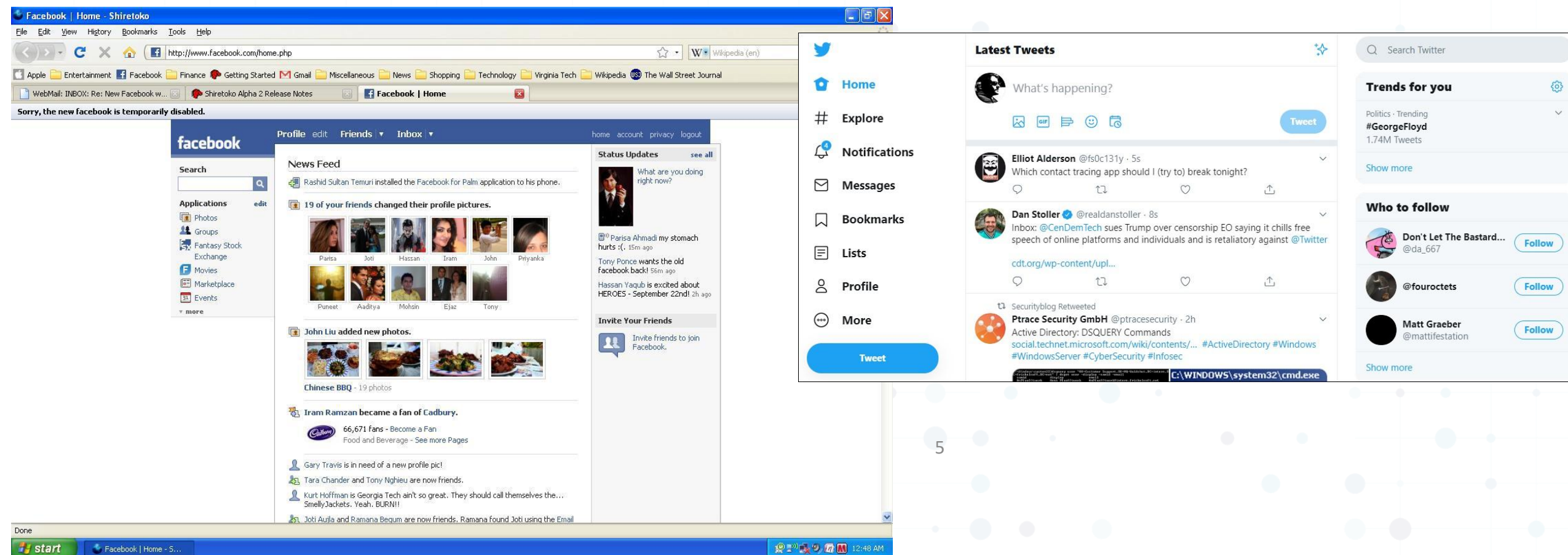
5

WEB 2.0: MID 2000S TO NOW



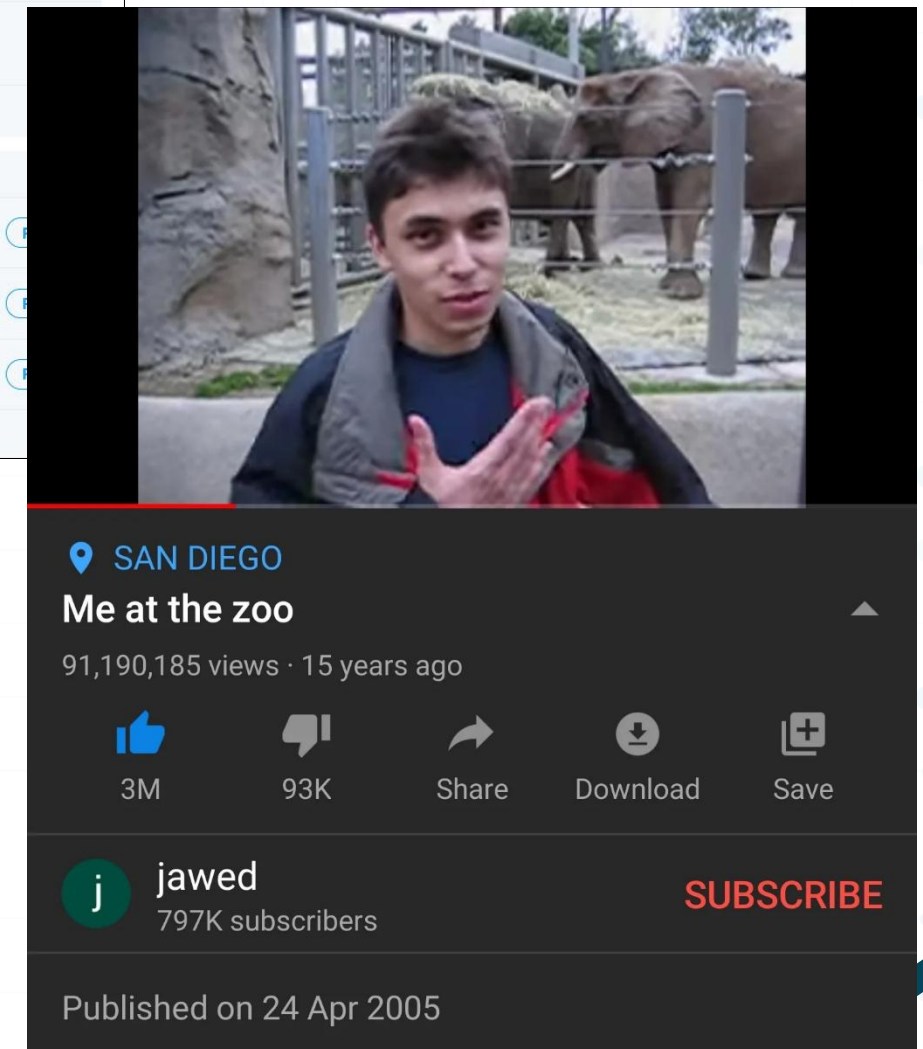
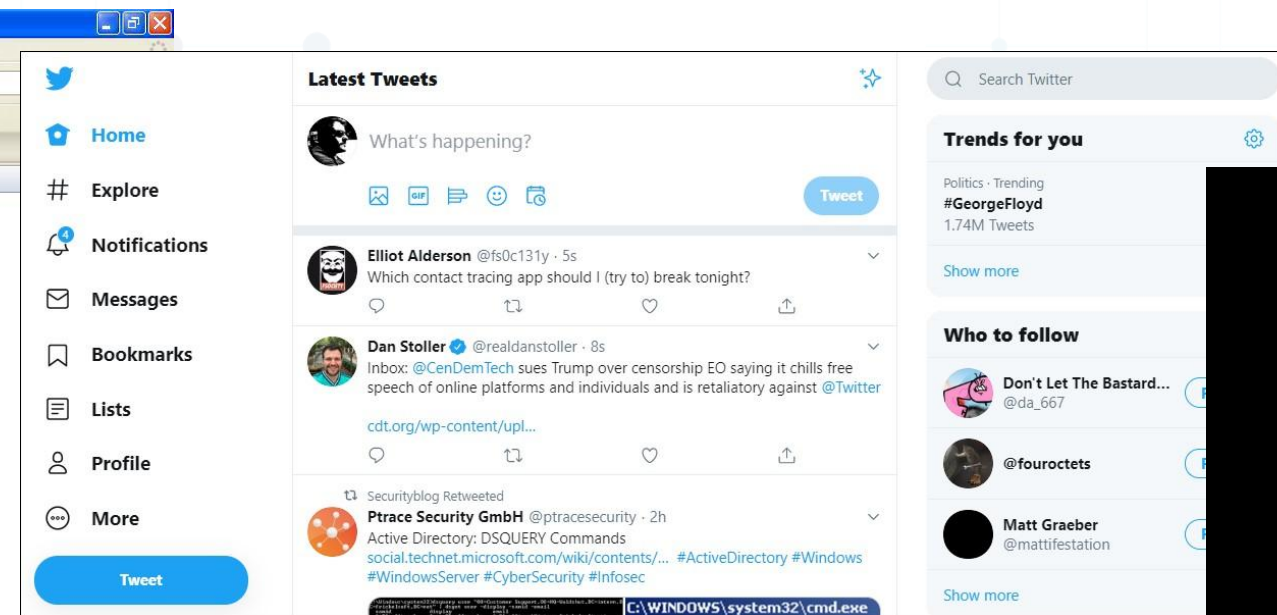
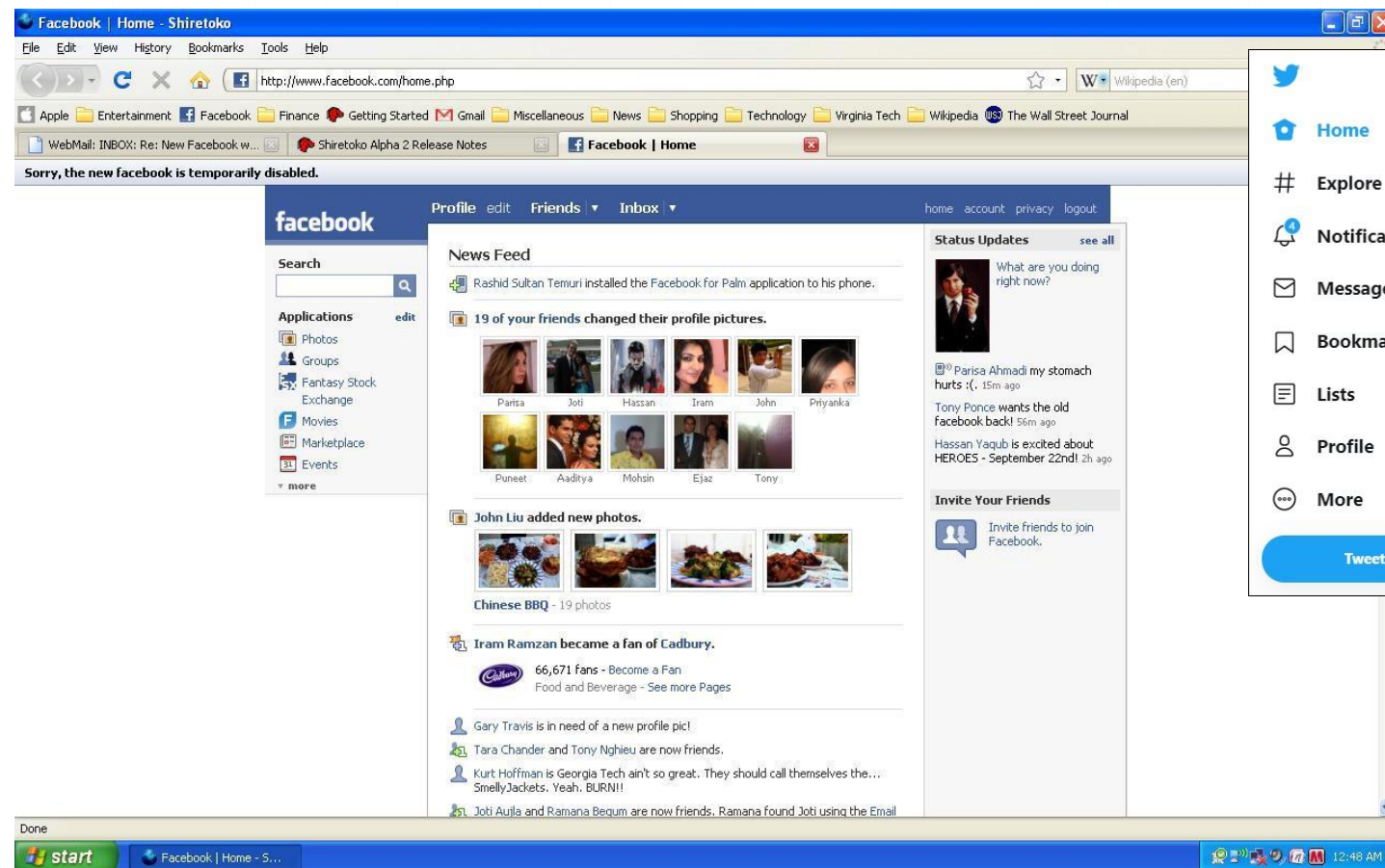
5

WEB 2.0: MID 2000S TO NOW



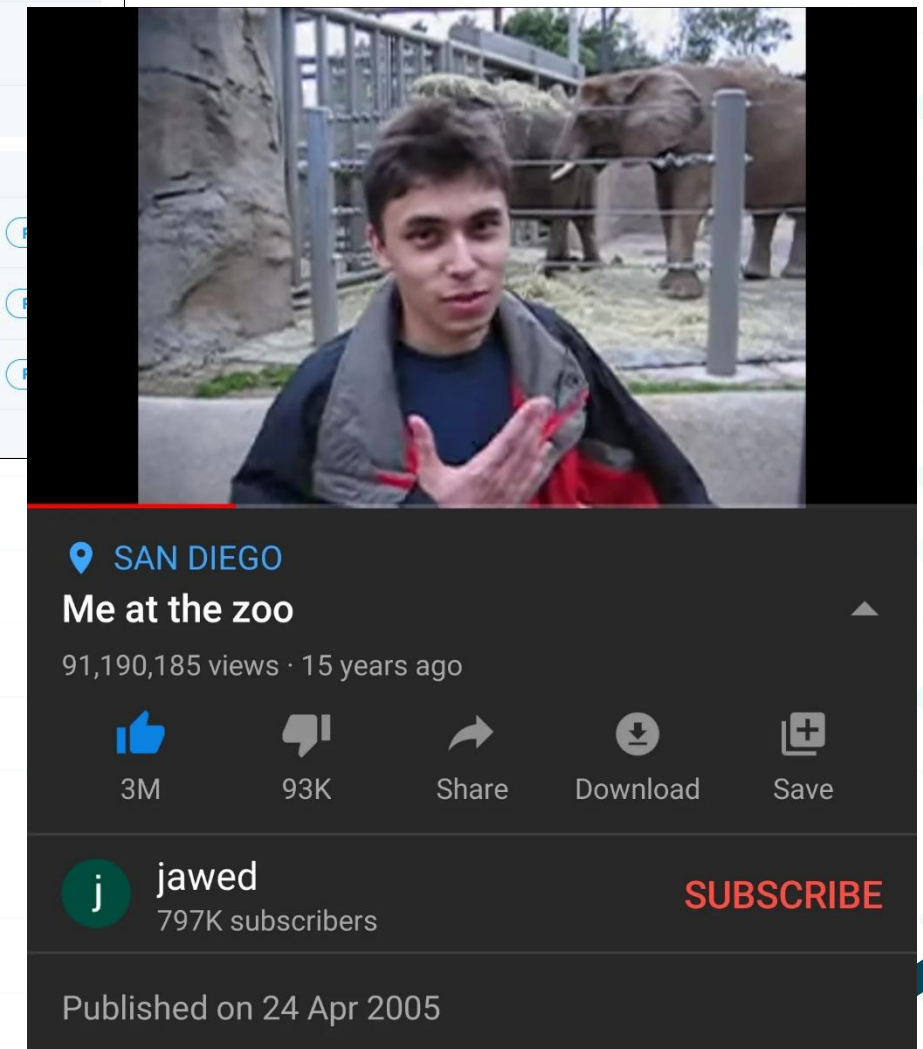
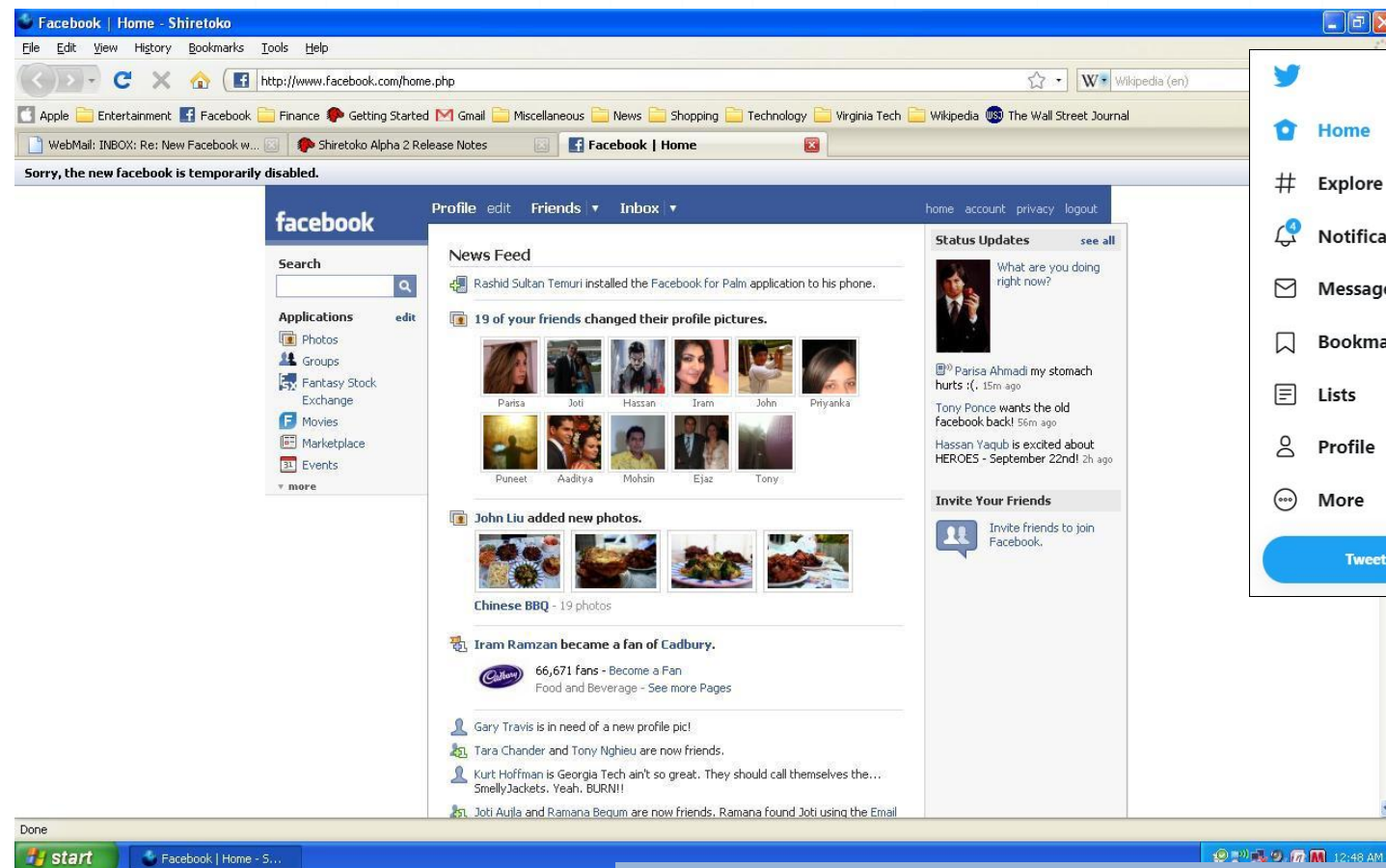
5

WEB 2.0: MID 2000S TO NOW



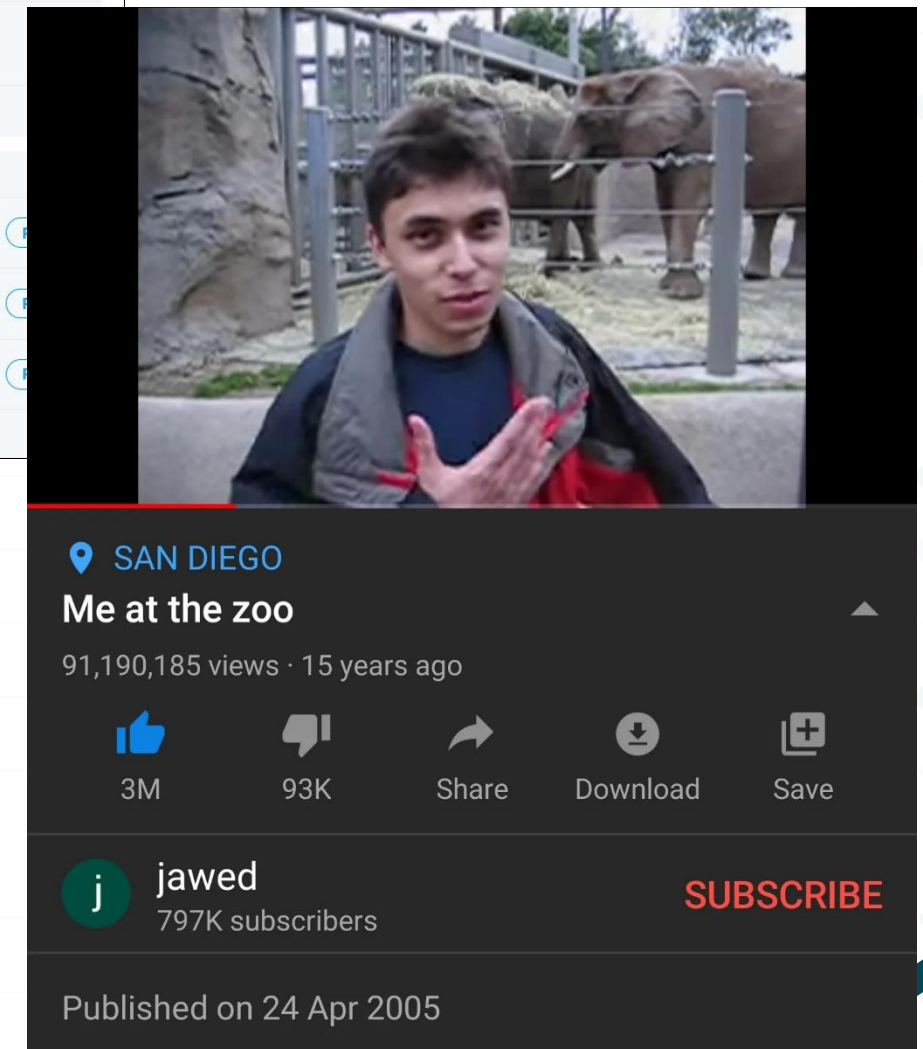
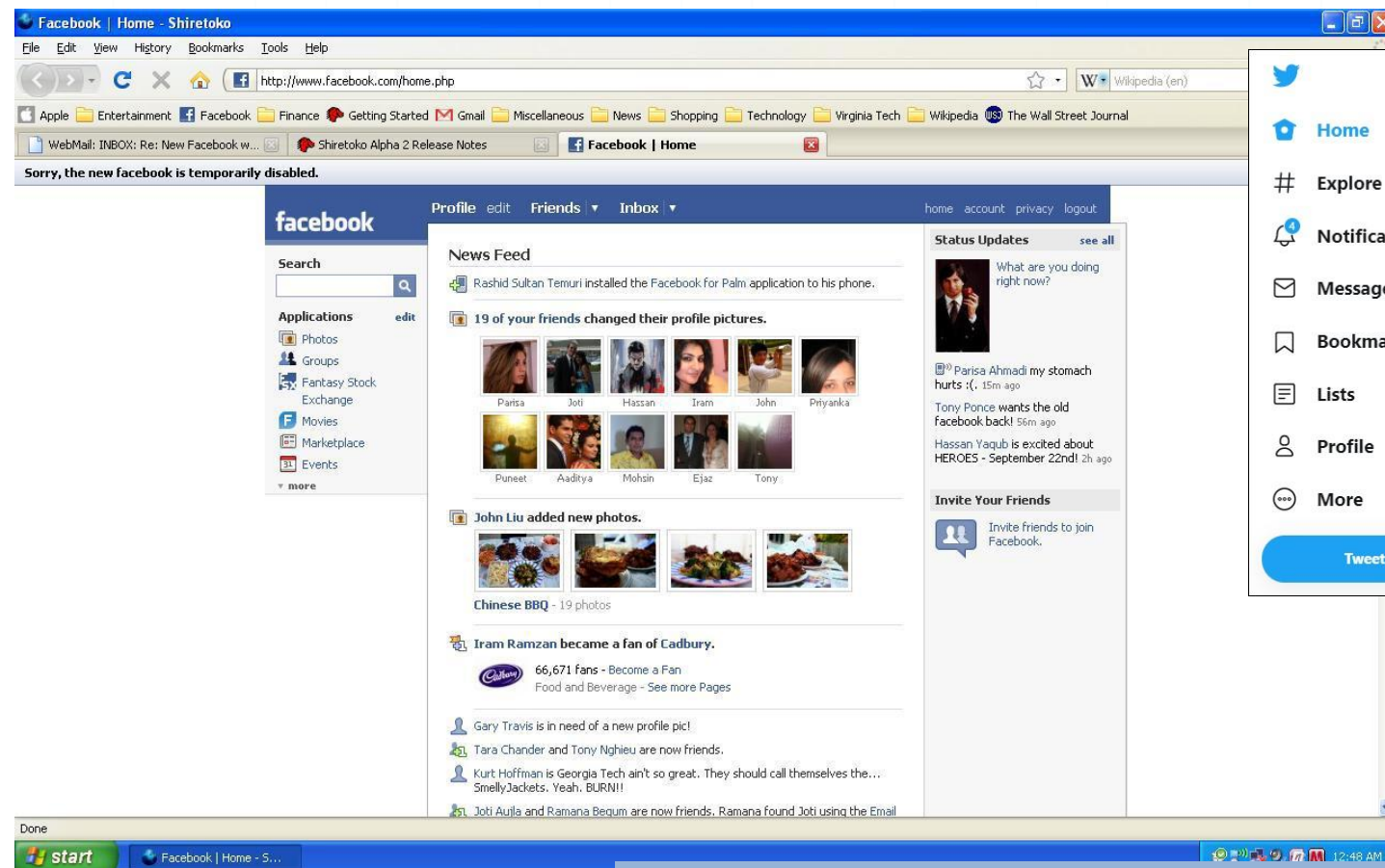
5

WEB 2.0: MID 2000S TO NOW



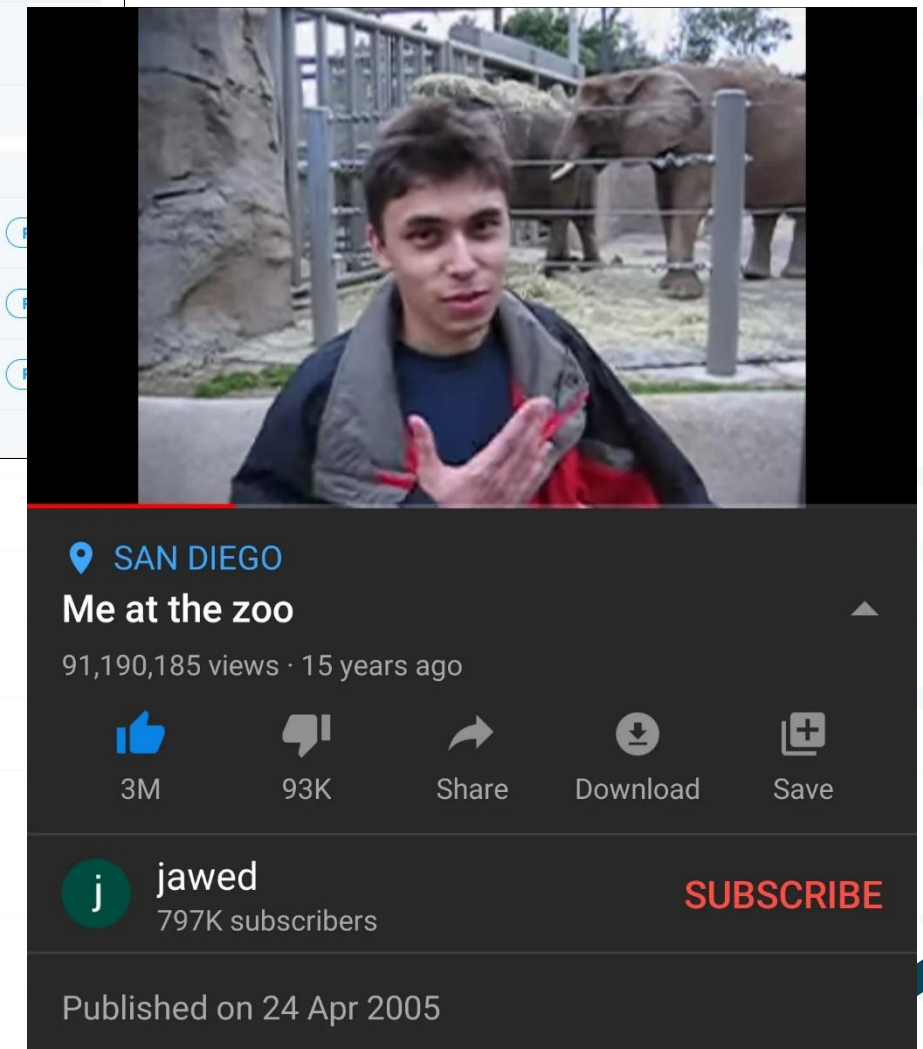
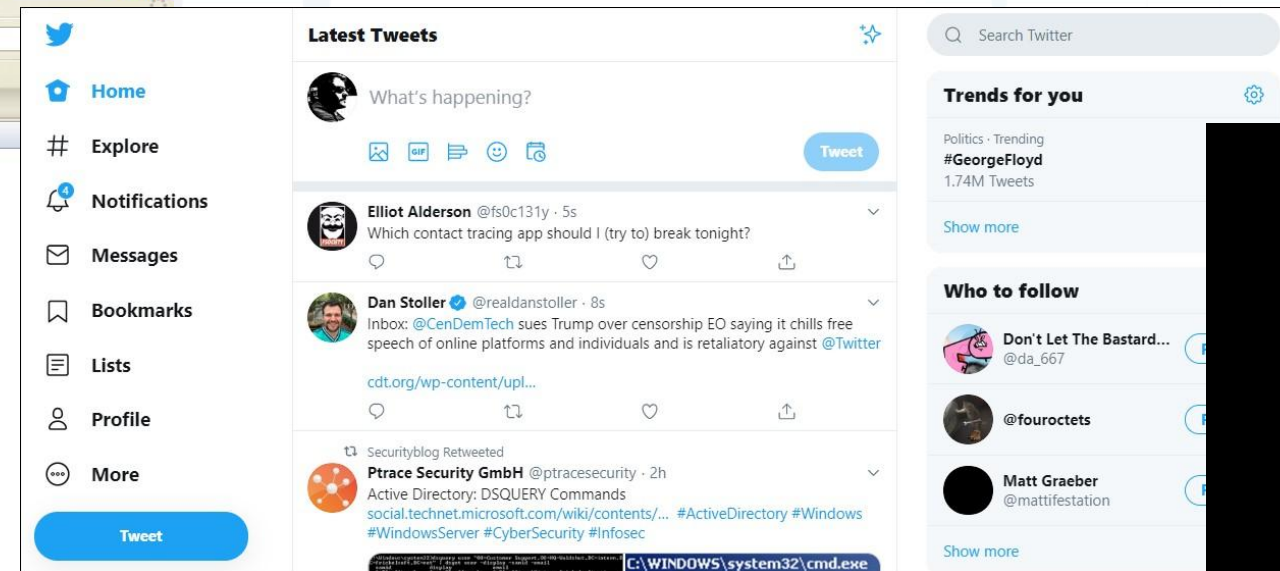
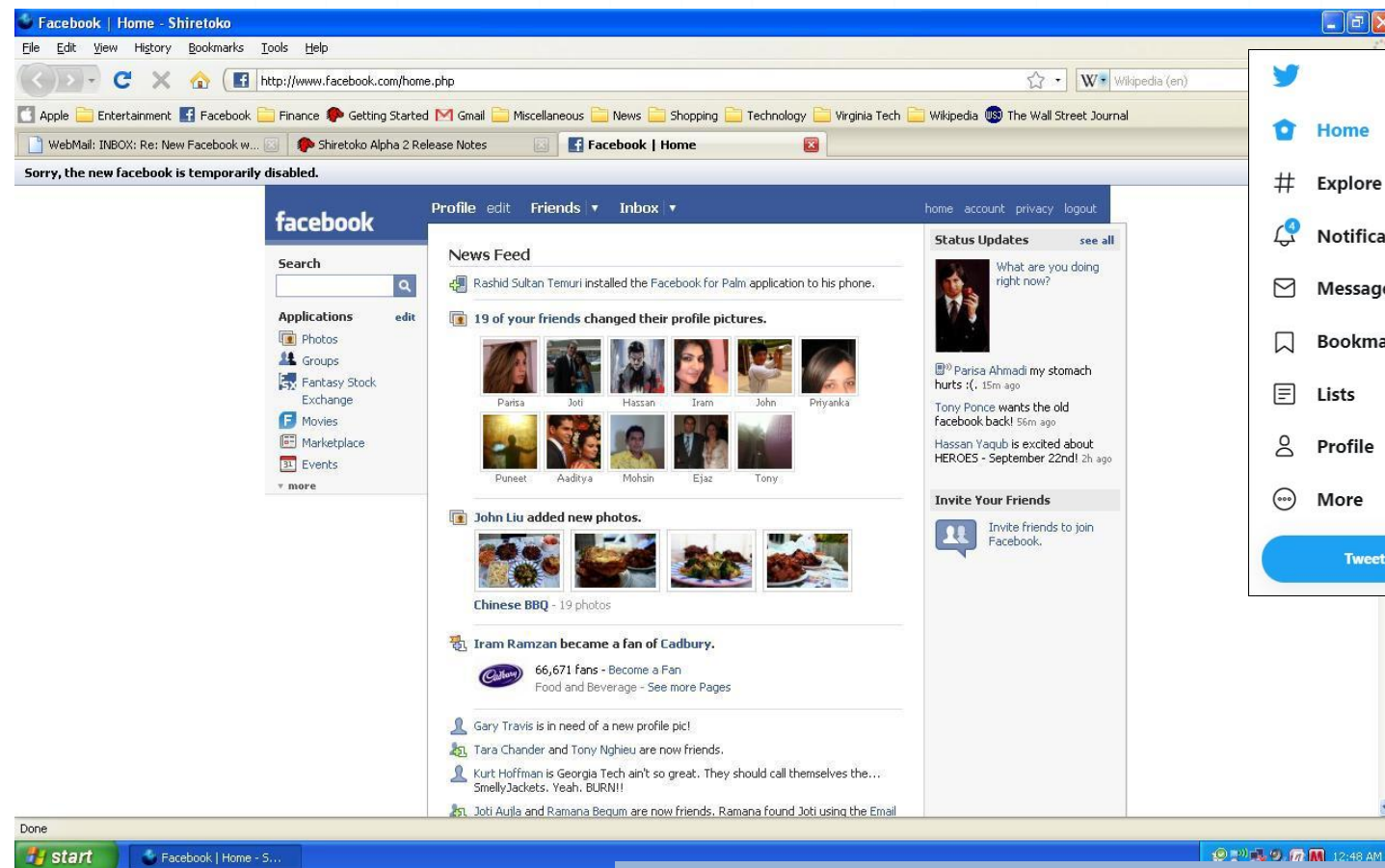
CENTRALIZATION:

WEB 2.0: MID 2000S TO NOW



CENTRALIZATION: Google

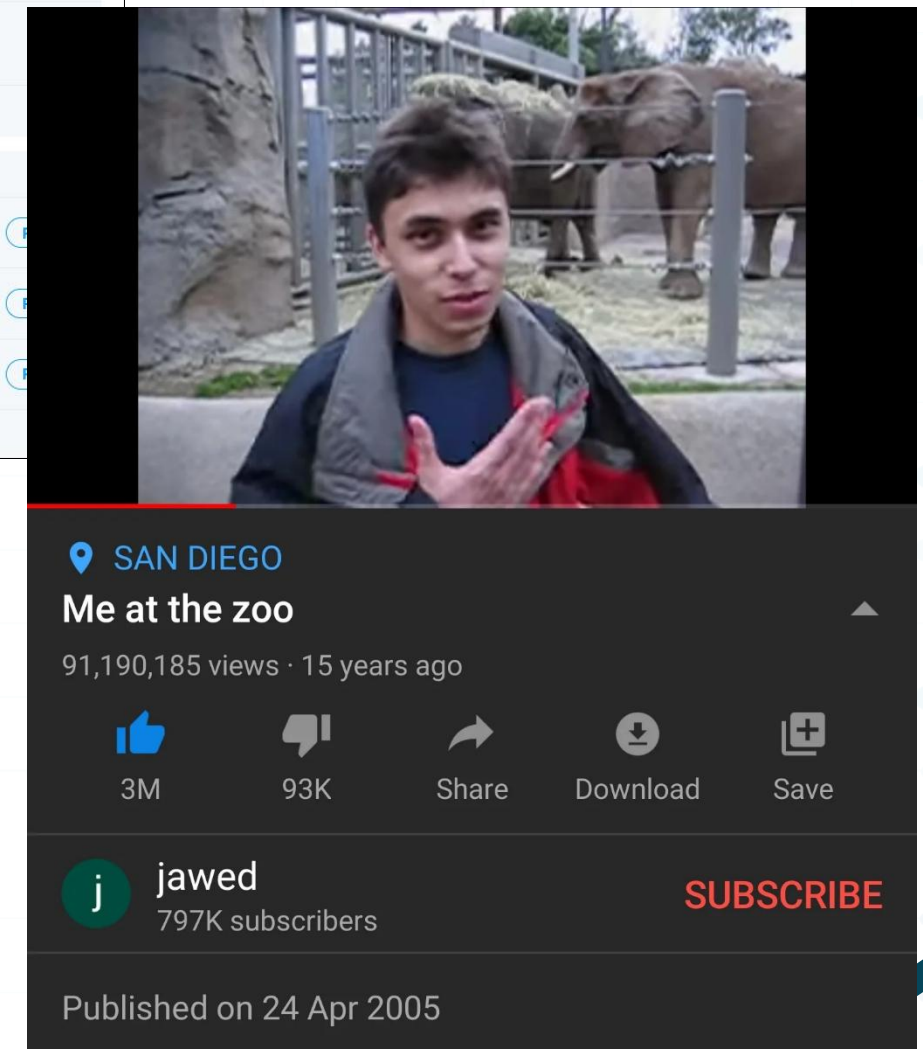
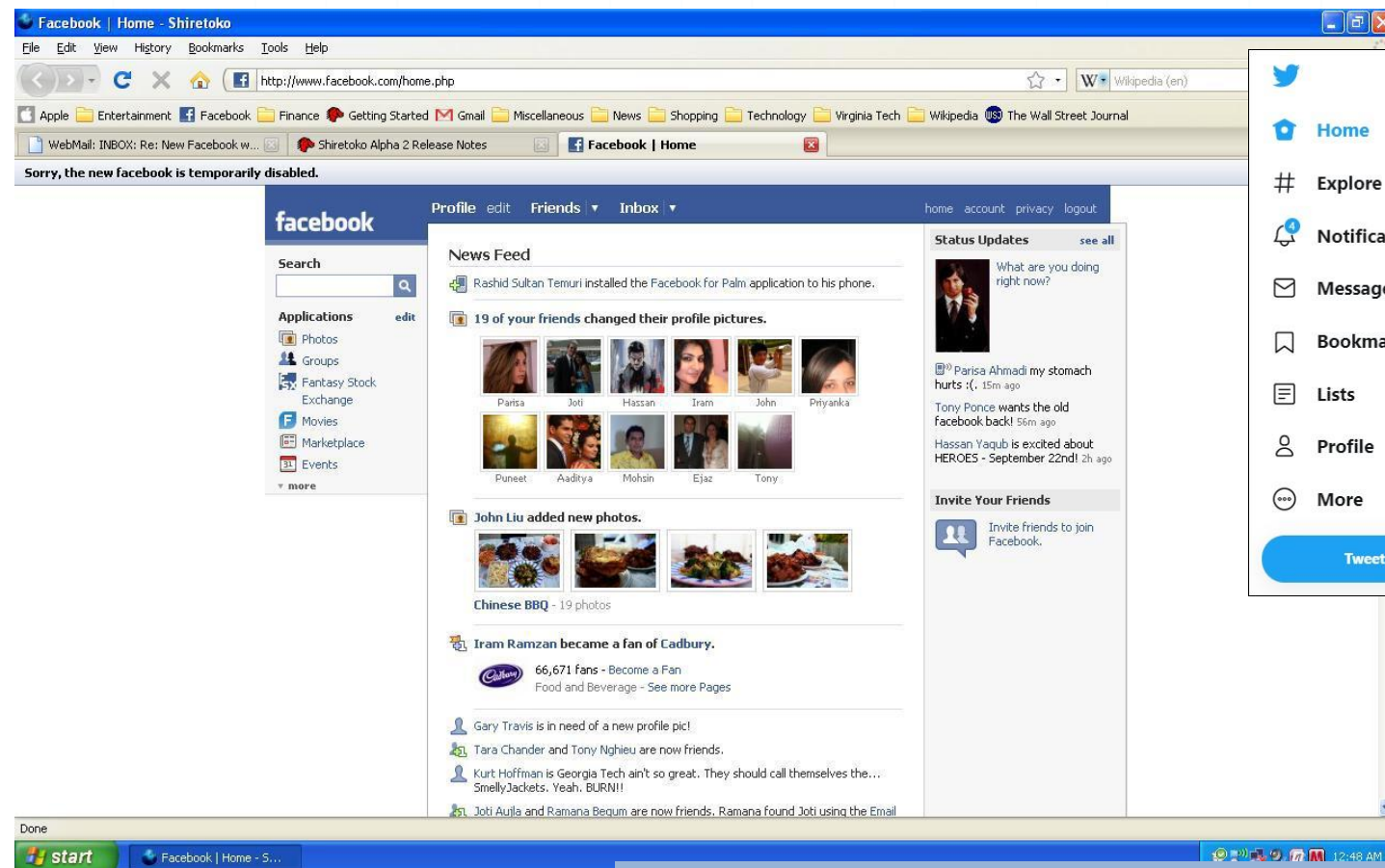
WEB 2.0: MID 2000S TO NOW



CENTRALIZATION:



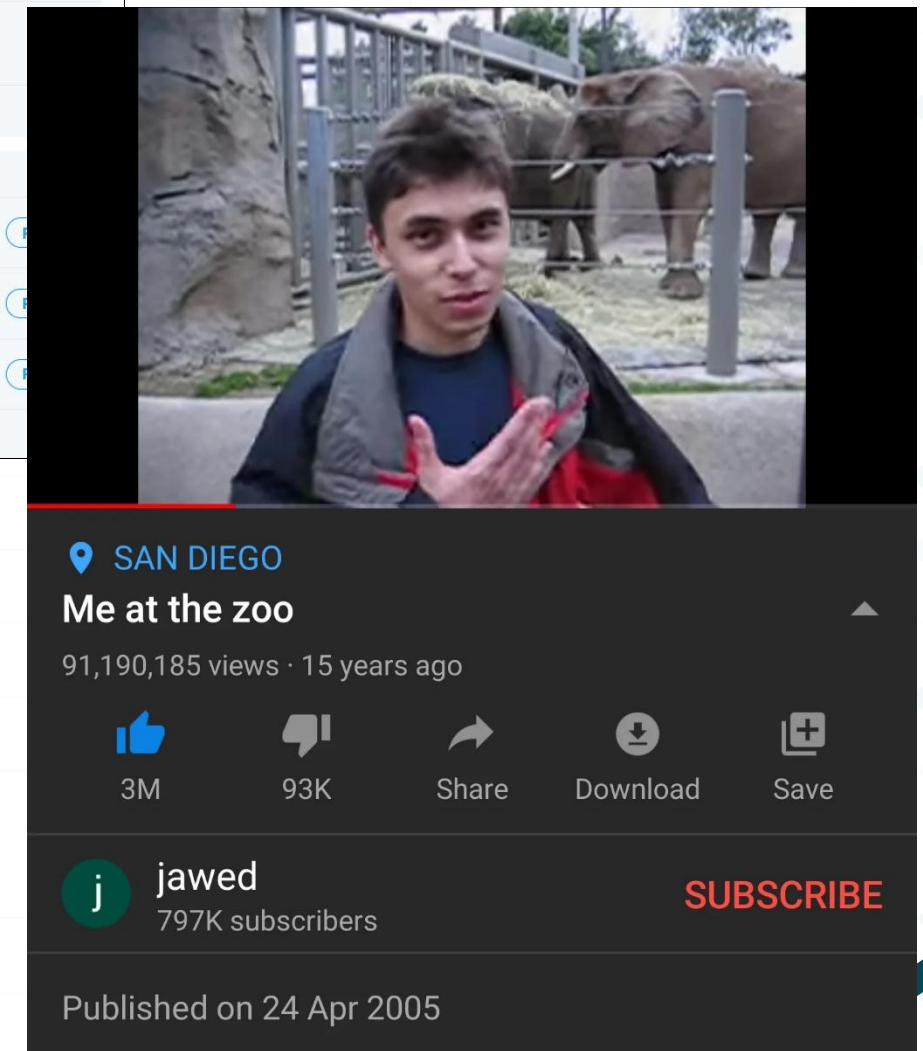
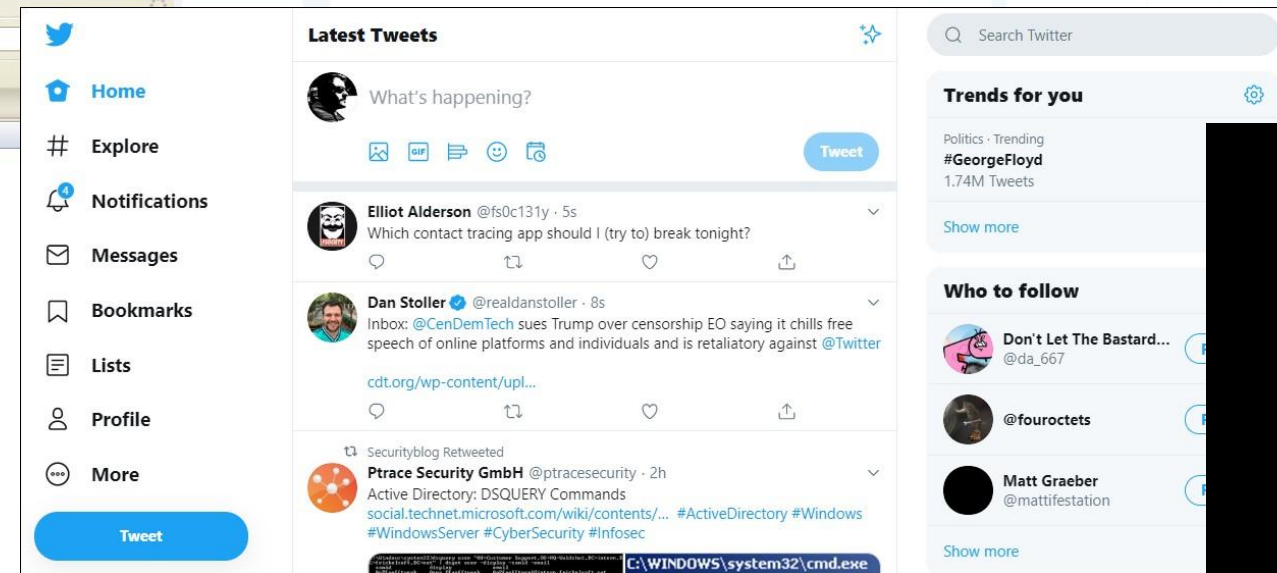
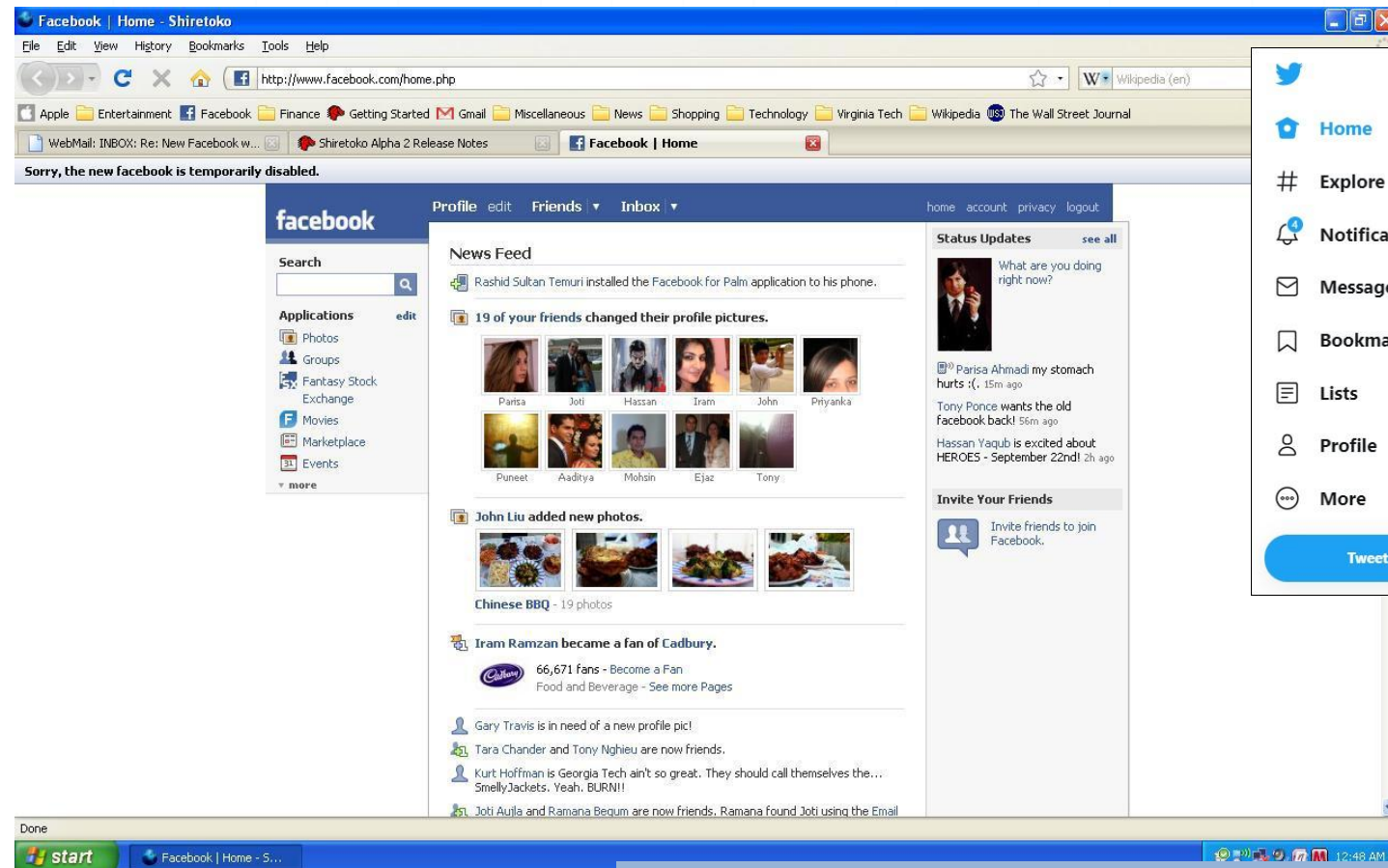
WEB 2.0: MID 2000S TO NOW



CENTRALIZATION:



WEB 2.0: MID 2000S TO NOW



CENTRALIZATION:

Google

Microsoft

Meta

amazon

WEB3: FUTURE?



Distributed Internet

6

WEB3: FUTURE?



Distributed Internet

- Decentralization

6

WEB3: FUTURE?



Distributed Internet

- Decentralization
- Blockchain technology

6

WEB3: FUTURE?



Distributed Internet

- Decentralization
- Blockchain technology
- Cryptocurrencies

6

WEB3: FUTURE?



Distributed Internet

- Decentralization
- Blockchain technology
- Cryptocurrencies
- Smart contracts⁶

WEB3: FUTURE?



Distributed Internet

- Decentralization
- Blockchain technology
- Cryptocurrencies
- Smart contracts⁶
- Privacy-preserving cryptography
(Zero-knowledge proofs)

WEB3: FUTURE?



Distributed Internet

- Decentralization
- Blockchain technology
- Cryptocurrencies
- Smart contracts⁶
- Privacy-preserving cryptography
(Zero-knowledge proofs)

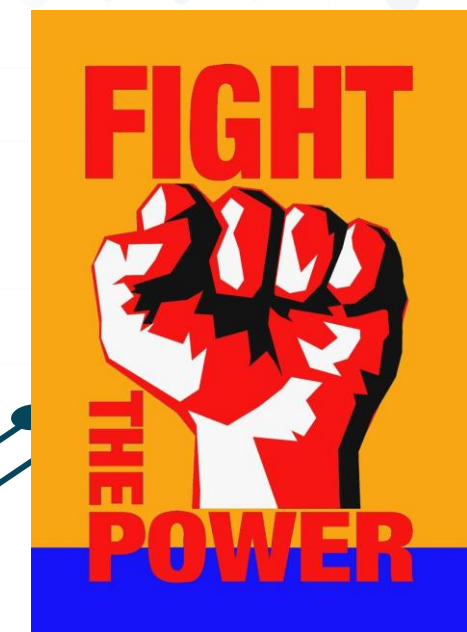


WEB3: FUTURE?



Distributed Internet

- Decentralization
- Blockchain technology
- Cryptocurrencies
- Smart contracts⁶
- Privacy-preserving cryptography
(Zero-knowledge proofs)



CORE: BLOCKCHAIN TECH

Immutable database

7

CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)

CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal

CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal

7



Users

CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal

7



Miners



Users



CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal

7



Miners



Users

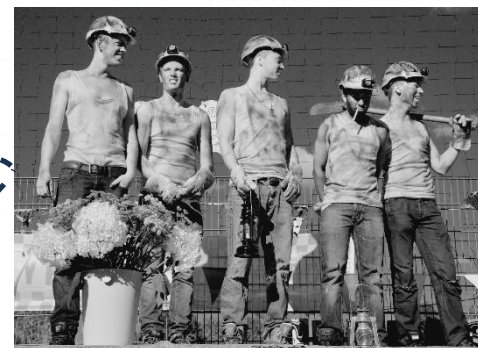
CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal

7

- Check data



Miners



Users

CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal

7

- 
- Check data
 - Mine blocks



Miners



Users

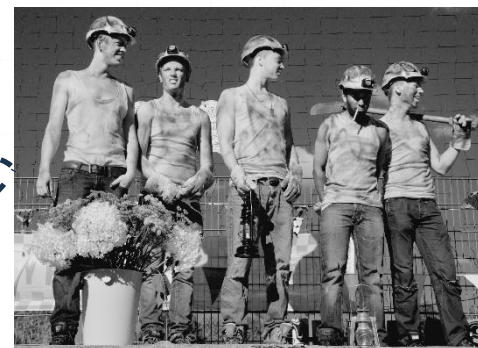
CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal

7

- Check data
- Mine blocks
- Get reward



Miners



Users

CORE: BLOCKCHAIN TECH

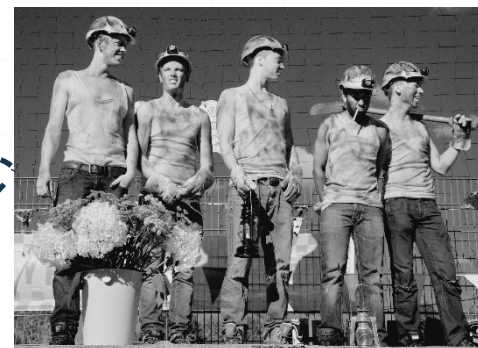
Immutable database

- Add data (under some constraints)
- No removal



7

- Check data
- Mine blocks
- Get reward



Miners



Users



CORE: BLOCKCHAIN TECH

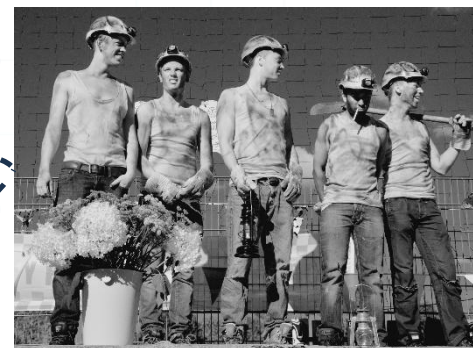
Immutable database

- Add data (under some constraints)
- No removal



7

- Check data
- Mine blocks
- Get reward



Miners



Users

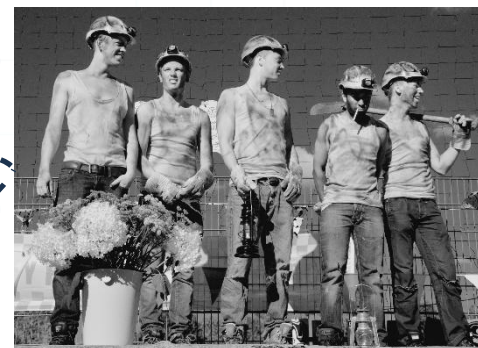
CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal



- Check data
- Mine blocks
- Get reward



Miners

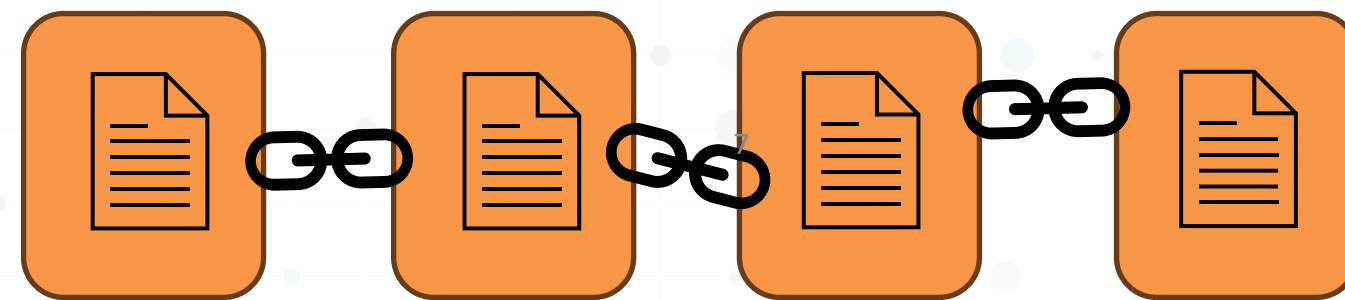


Users

CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal



- Check data
- Mine blocks
- Get reward



Miners



Users

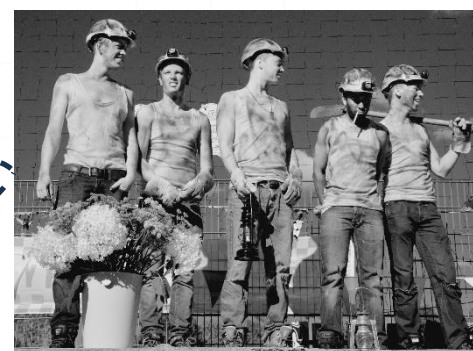
CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal



- Check data
- Mine blocks
- Get reward



Miners

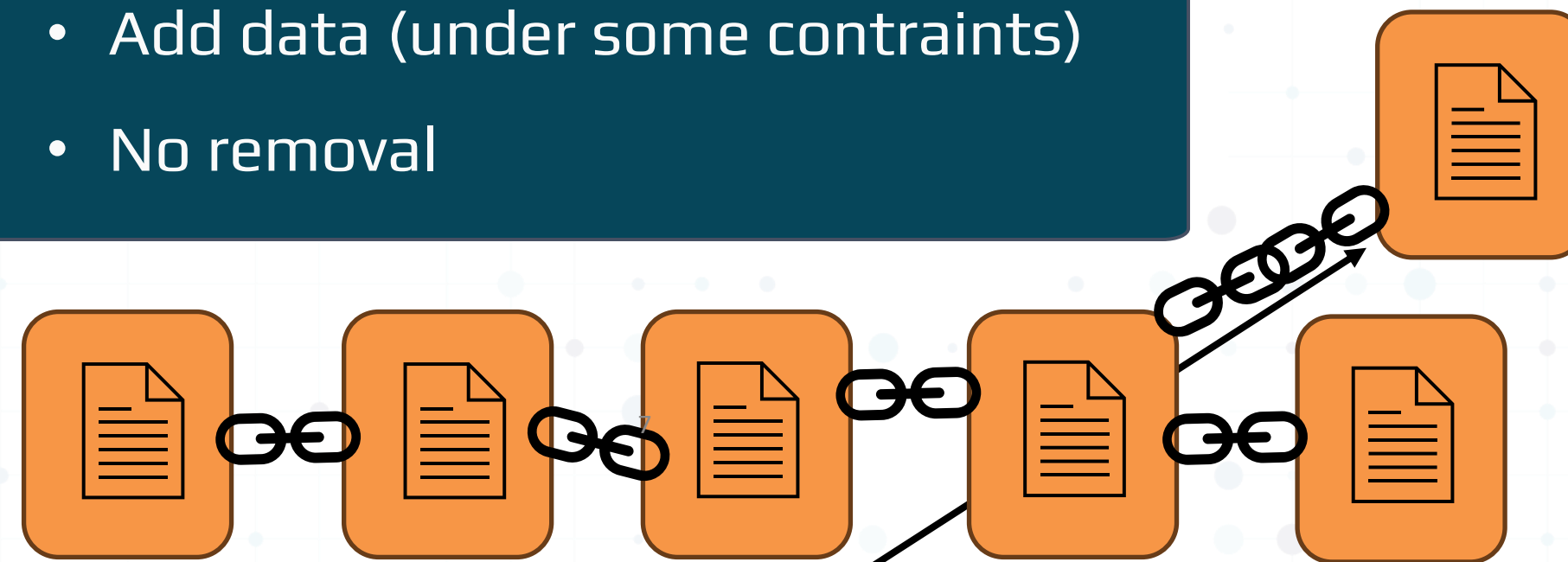


Users

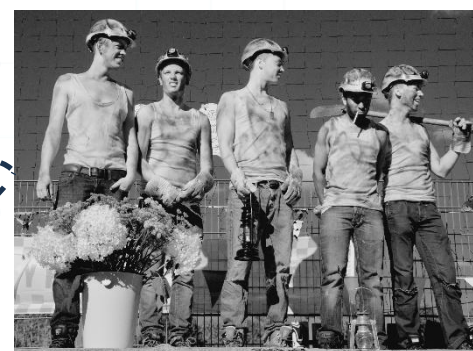
CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal



- Check data
- Mine blocks
- Get reward



Miners

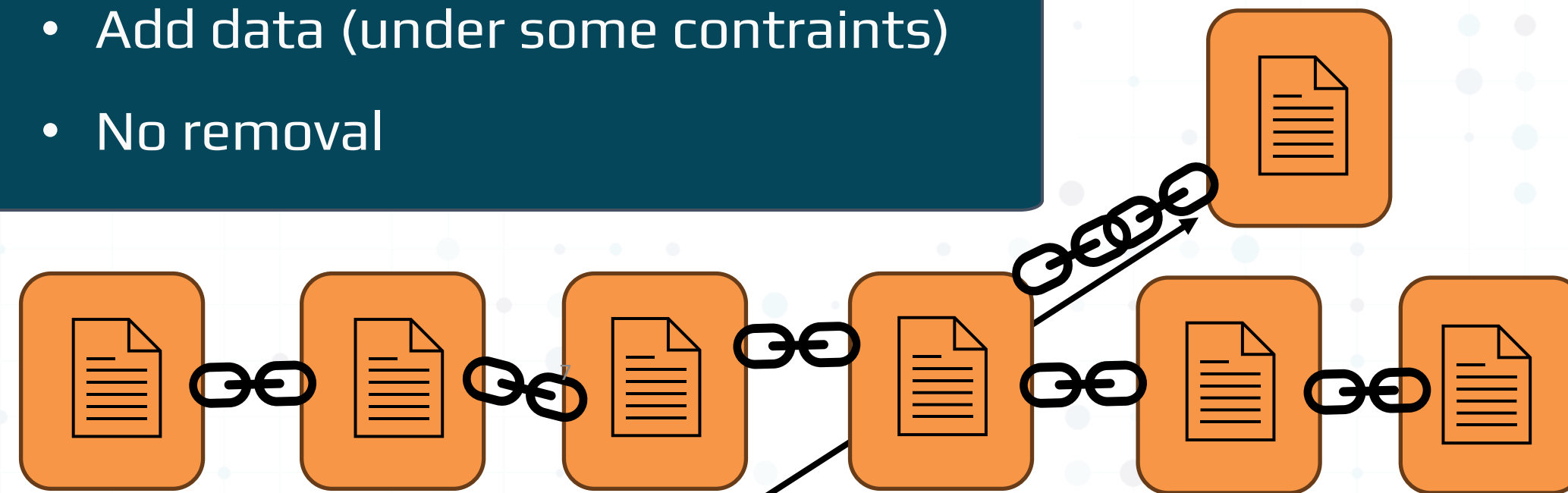


Users

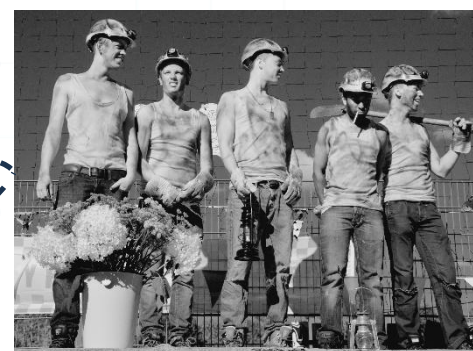
CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal



- Check data
- Mine blocks
- Get reward



Miners

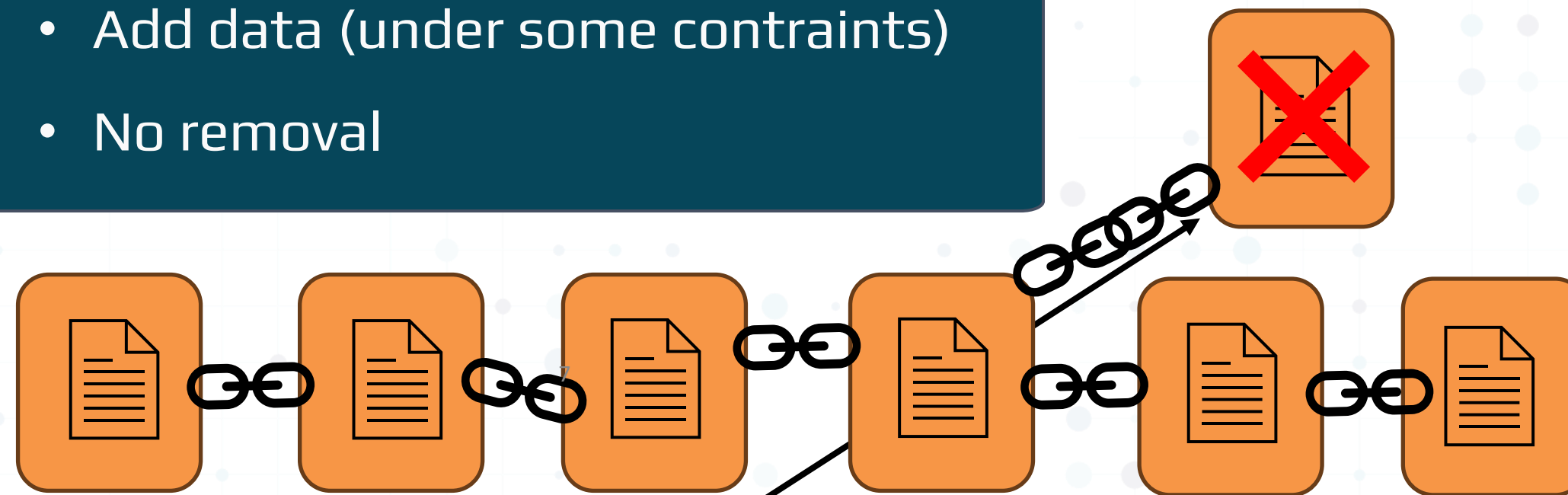


Users

CORE: BLOCKCHAIN TECH

Immutable database

- Add data (under some constraints)
- No removal



- Check data
- Mine blocks
- Get reward



Miners



Users

CRYPTO CURRENCIES



Immutable database



Miners



Users

8

CRYPTO CURRENCIES



Immutable database

From



Miners



Users

8

CRYPTO CURRENCIES



Immutable database

From To



Miners



Users

8

CRYPTO CURRENCIES



Immutable database

From	To	Amount
------	----	--------



Miners



Users

8

CRYPTO CURRENCIES



Immutable database

From	To	Amount	Signature
------	----	--------	-----------



Miners



Users

8

CRYPTO CURRENCIES



Immutable database

From	To	Amount	Signature
------	----	--------	-----------

Alice			
-------	--	--	--

8



Miners



Users

CRYPTO CURRENCIES



Immutable database

From	To	Amount	Signature
Alice	Bob		



Miners



Users

8

CRYPTO CURRENCIES



Immutable database

From	To	Amount	Signature
Alice	Bob	50 euro	



Miners



Users

CRYPTO CURRENCIES



Immutable database

From	To	Amount	Signature
Alice	Bob	50 euro	<i>Alice</i>



Miners



Users

8

CRYPTO CURRENCIES



Immutable database

From	To	Amount	Signature
Alice	Bob	50 euro	<i>Alice</i>
Bob	Eve	20 euros	<i>Bobby</i>



Miners



Users

CRYPTO CURRENCIES



Immutable database

From	To	Amount	Signature
Alice	Bob	50 euro	<i>Alice</i>
Bob	Eve	20 euros	<i>Bobby</i>



Miners



Users



CRYPTO CURRENCIES



Immutable database

From	To	Amount	Signature
Alice	Bob	50 euro	<i>Alice</i>
Bob	Eve	20 euros	<i>Bobby</i>

- Check that no double spending



Miners



Users

CRYPTO CURRENCIES



Immutable database

From	To	Amount	Signature
Alice	Bob	50 euro	<i>Alice</i>
Bob	Eve	20 euros	<i>Bobby</i>

- Check that no double spending
- Signature valid



Miners



Users

CRYPTO CURRENCIES



Immutable database

From	To	Amount	Signature
Alice	Bob	50 euro	<i>Alice</i>
Bob	Eve	20 euros	<i>Bobby</i>

- Check that no double spending
- Signature valid



Miners



Users

CRYPTO CURRENCIES



Immutable database

From	To	Amount	Signature
Alice	Bob	50 euro	<i>Alice</i>
Bob	Eve	20 euros	<i>Bobby</i>
Alice	Eve	10 euro	<i>Alice</i>

- Check that no double spending
- Signature valid



Miners



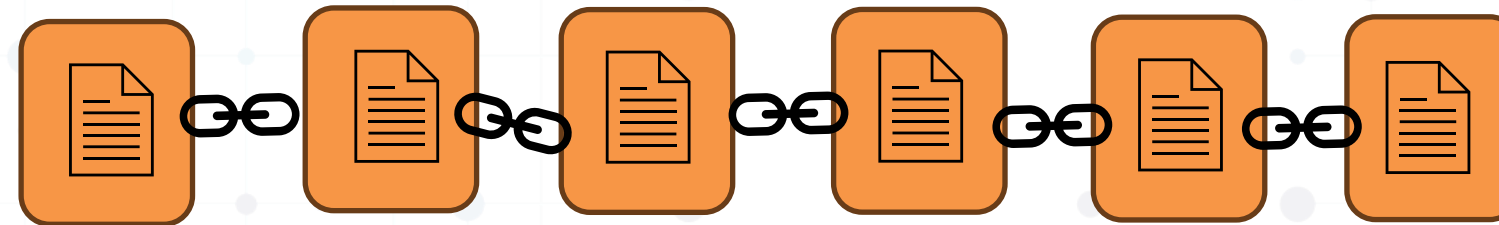
Users

SMART CONTRACTS

1

Why just data on blockchain?

9



Miners



Users

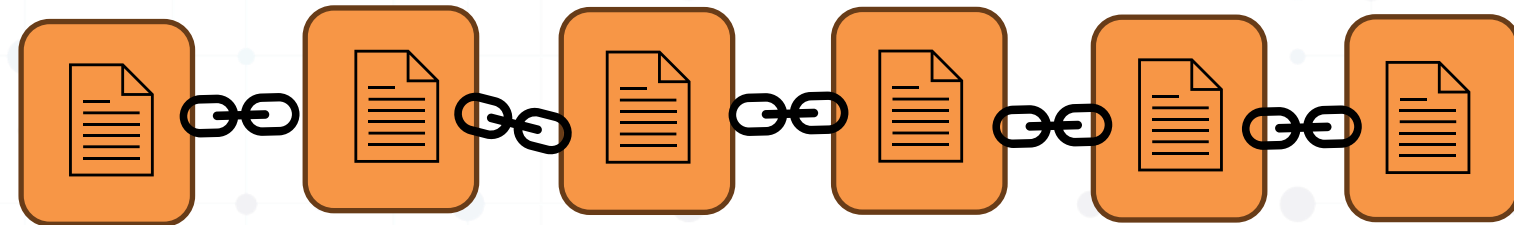


SMART CONTRACTS

1

Why just data on blockchain?

9



Miners



Users



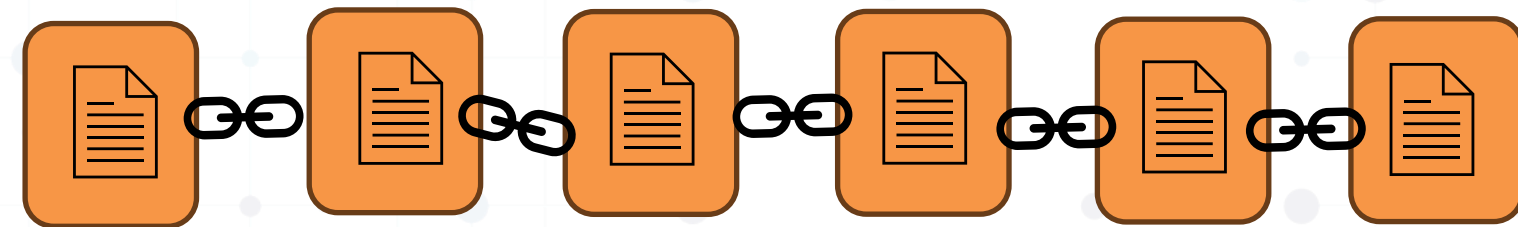
SMART CONTRACTS

1

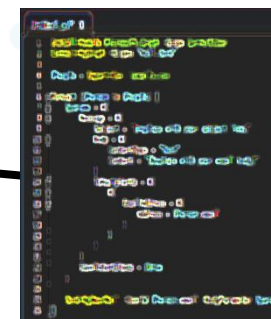
Why just data on blockchain?

- Let's run computer programs on blockchain

9



Miners



Users

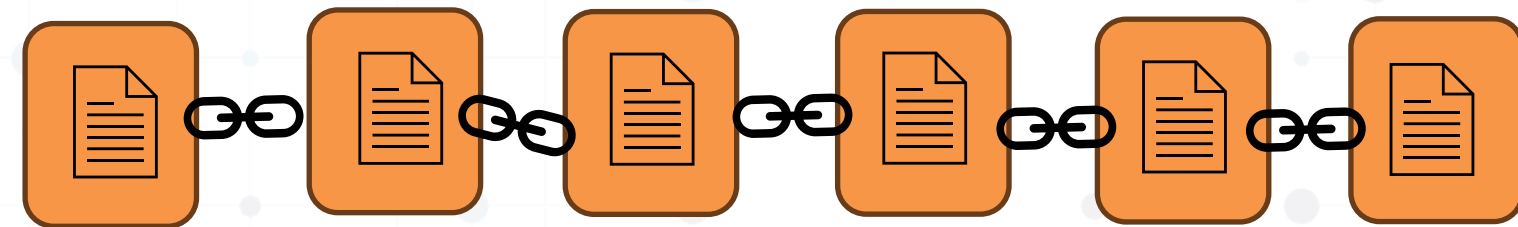
SMART CONTRACTS

1

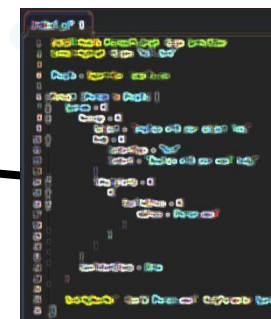
Why just data on blockchain?

- Let's run computer programs on blockchain
- On input X do Y

9



Miners



Users

SMART CONTRACTS

1

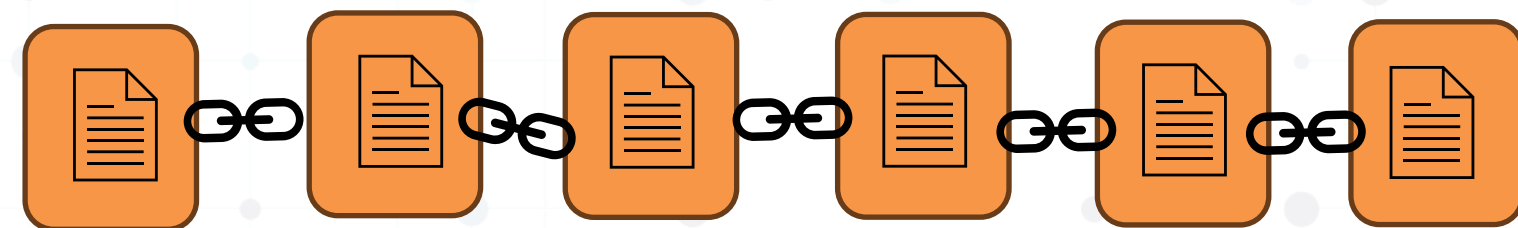
Why just data on blockchain?

- Let's run computer programs on blockchain
- On input X do Y

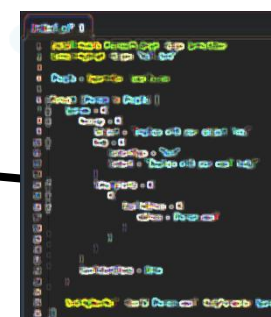
2

Features:

9



Miners



Users

SMART CONTRACTS

1

Why just data on blockchain?

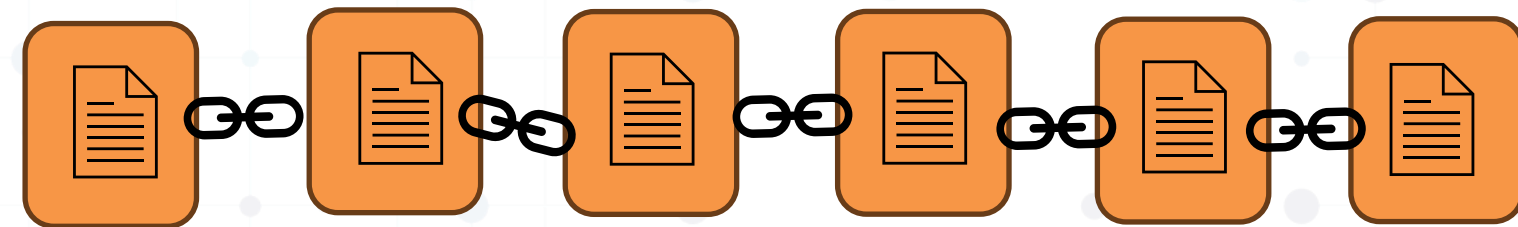
- Let's run computer programs on blockchain
- On input X do Y

2

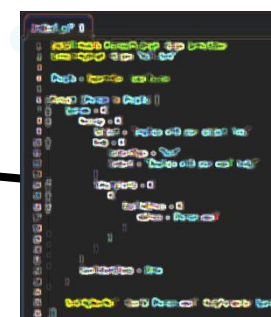
Features:

- Automatic execution (no trusted party)

9



Miners



Users

SMART CONTRACTS

1

Why just data on blockchain?

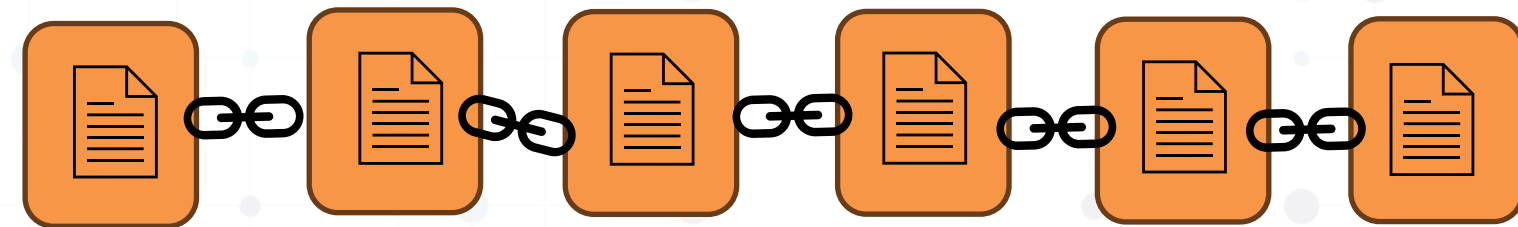
- Let's run computer programs on blockchain
- On input X do Y

2

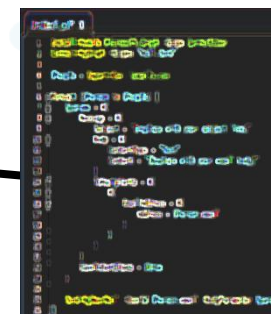
Features:

- Automatic execution (no trusted party)
- Transparency (whatever is in contract happens)

9



Miners



Users

SMART CONTRACTS

1

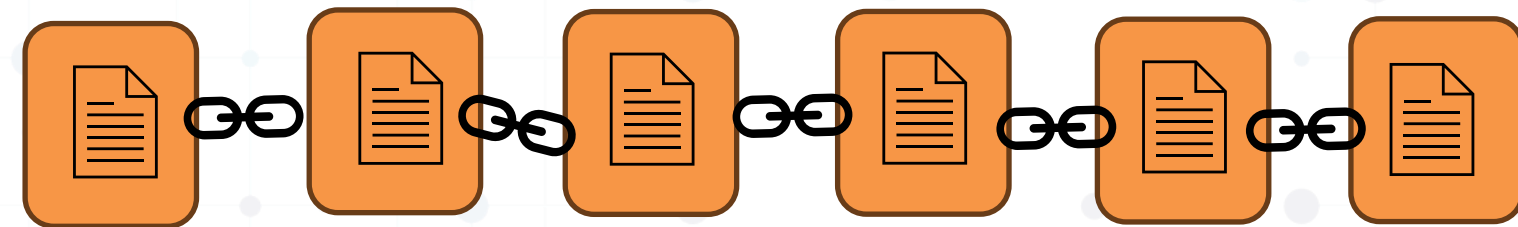
Why just data on blockchain?

- Let's run computer programs on blockchain
- On input X do Y

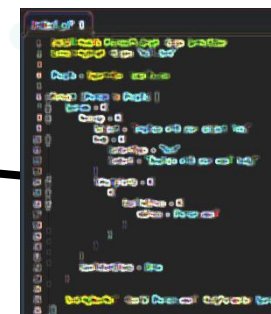
2

Features:

- Automatic execution (no trusted party)
- Transparency (whatever is in contract happens)
- Example: supply chain management (truck's GPS shows location X, release funds)



Miners



Users

SMART CONTRACTS

1

Why just data on blockchain?

- Let's run computer programs on blockchain
- On input X do Y

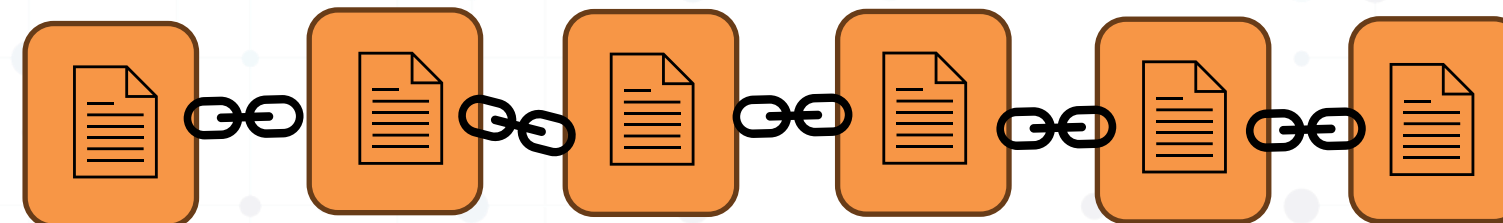
2

Features:

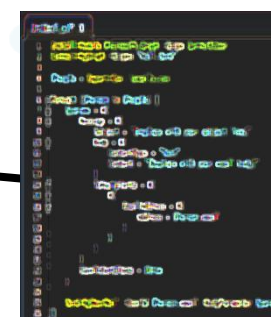
- Automatic execution (no trusted party)
- Transparency (whatever is in contract happens)
- Example: supply chain management (truck's GPS shows location X, release funds)



ethereum



Miners



Users

BIG CHALLENGES IN BLOCKCHAIN

10

BIG CHALLENGES IN BLOCKCHAIN

No Privacy

Transaction are public for everyone.



10



BIG CHALLENGES IN BLOCKCHAIN

No Privacy

Transaction are public for everyone.



Scalability

Bitcoin 3-7tx/sec.
Mastercard 5000 tx/sec.



BIG CHALLENGES IN BLOCKCHAIN

No Privacy

Transaction are public for everyone.

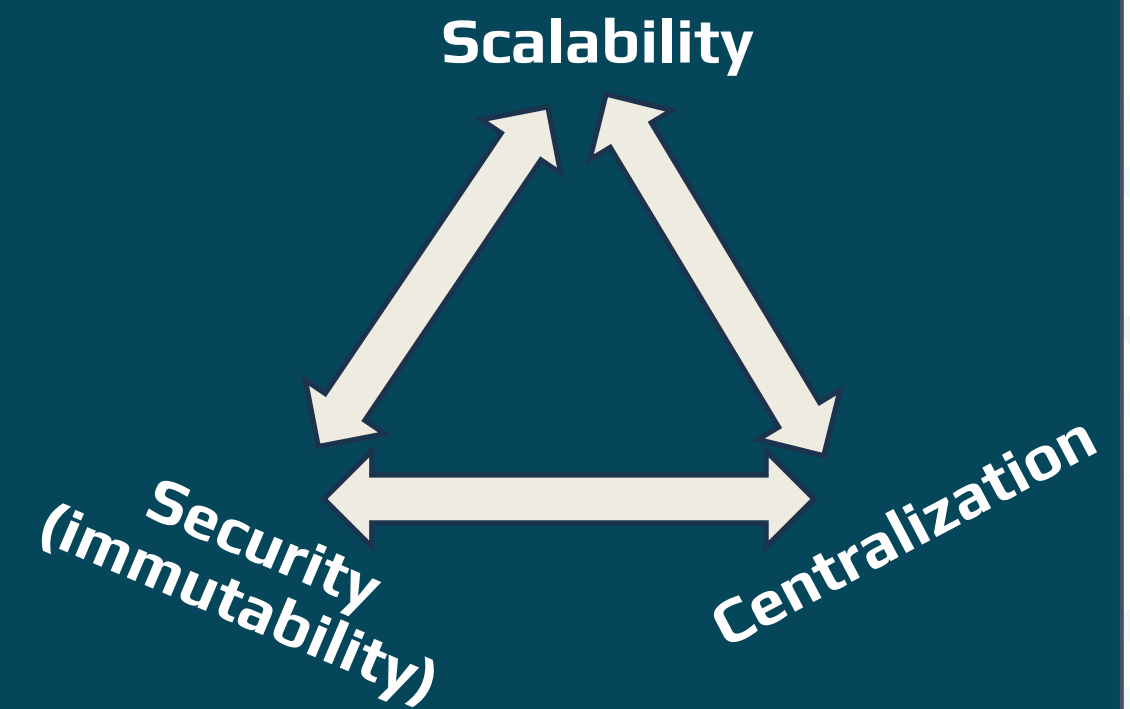


Scalability

Bitcoin 3-7tx/sec.
Mastercard 5000 tx/sec.



Blockchain Trilemma



ZERO-KNOWLEDGE PROOF: PERFECT TOOL FOR BOTH

11

Starkware Valuation Hits \$2B After \$50M Series C Round

The fresh capital will support the deployment of its StarkNet Layer 2 platform, which will allow anyone to build blockchain applications with its technology.

BY JACQUELYN MELINEK / NOVEMBER 16, 2021 02:24 PM



The Proximity Prize

\$1,000,000

in prizes to prove (or disprove!) Reed-Solomon proximity gaps conjectures.

An initiative by the Ethereum Foundation to advance the foundations of modern zkVMs.

ZERO-KNOWLEDGE VIRTUAL MACHINE

ZKVM

12

ZK PROVER

ZERO-KNOWLEDGE VIRTUAL MACHINE

ZKVM

12

ZK PROVER

```
Untitled1.ps1* X
1  Install-Module Microsoft.Graph -Scope CurrentUser
2  Connect-MgGraph -Scopes "Mail.Send"
3
4  $People = Import-Csv .\emails.csv
5
6  foreach ($Person in $People) {
7      $params = @{
8          Message = @{
9              Subject = "[replace with your subject line]"
10             Body = @{
11                 ContentType = "Text"
12                 Content = "[Replace with your email body]"
13             }
14             ToRecipients = @(
15                 @{
16                     EmailAddress = @{
17                         Address = $Person.email
18                     }
19                 }
20             )
21         }
22         SaveToSentItems = $true
23     }
24     Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```

ZERO-KNOWLEDGE VIRTUAL MACHINE

Public data



ZKVM

12

ZK PROVER

```
Untitled1.ps1* X
1  Install-Module Microsoft.Graph -Scope CurrentUser
2  Connect-MgGraph -Scopes "Mail.Send"
3
4  $People = Import-Csv .\emails.csv
5
6  foreach ($Person in $People) {
7      $params = @{
8          Message = @{
9              Subject = "[replace with your subject line]"
10             Body = @{
11                 ContentType = "Text"
12                 Content = "[Replace with your email body]"
13             }
14             ToRecipients = @(
15                 @{
16                     EmailAddress = @{
17                         Address = $Person.email
18                     }
19                 }
20             )
21         }
22         SaveToSentItems = $true
23     }
24     Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```

ZERO-KNOWLEDGE VIRTUAL MACHINE

Public data



Private data



ZKVM

12

ZK PROVER

```
Untitled1.ps1* X
1  Install-Module Microsoft.Graph -Scope CurrentUser
2  Connect-MgGraph -Scopes "Mail.Send"
3
4  $People = Import-Csv .\emails.csv
5
6  foreach ($Person in $People) {
7      $params = @{
8          Message = @{
9              Subject = "[replace with your subject line]"
10             Body = @{
11                 ContentType = "Text"
12                 Content = "[Replace with your email body]"
13             }
14             ToRecipients = @(
15                 @{
16                     EmailAddress = @{
17                         Address = $Person.email
18                     }
19                 }
20             )
21         }
22         SaveToSentItems = $true
23     }
24     Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```

ZERO-KNOWLEDGE VIRTUAL MACHINE

Public data



Private data



ZKVM

```
1 Install-Module Microsoft.Graph -Scope CurrentUser
2 Connect-MgGraph -Scopes "Mail.Send"
3
4 $People = Import-Csv .\emails.csv
5
6 foreach ($Person in $People) {
7     $params = @{
8         Message = @{
9             Subject = "[replace with your subject line]"
10            Body = @{
11                ContentType = "Text"
12                Content = "[Replace with your email body]"
13            }
14            ToRecipients = @(
15                @{
16                    EmailAddress = @{
17                        Address = $Person.email
18                    }
19                }
20            )
21        }
22        SaveToSentItems = $true
23    }
24    Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```

12 ZK PROVER

```
Untitled1.ps1* X
1 Install-Module Microsoft.Graph -Scope CurrentUser
2 Connect-MgGraph -Scopes "Mail.Send"
3
4 $People = Import-Csv .\emails.csv
5
6 foreach ($Person in $People) {
7     $params = @{
8         Message = @{
9             Subject = "[replace with your subject line]"
10            Body = @{
11                ContentType = "Text"
12                Content = "[Replace with your email body]"
13            }
14            ToRecipients = @(
15                @{
16                    EmailAddress = @{
17                        Address = $Person.email
18                    }
19                }
20            )
21        }
22        SaveToSentItems = $true
23    }
24    Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```

ZERO-KNOWLEDGE VIRTUAL MACHINE

Public data



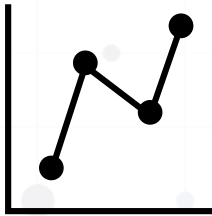
Private data



ZKVM

```
1 Install-Module Microsoft.Graph -Scope CurrentUser
2 Connect-MgGraph -Scopes "Mail.Send"
3
4 $People = Import-Csv .\emails.csv
5
6 foreach ($Person in $People) {
7     $params = @{
8         Message = @{
9             Subject = "[replace with your subject line]"
10            Body = @{
11                ContentType = "Text"
12                Content = "[Replace with your email body]"
13            }
14            ToRecipients = @(
15                @{
16                    EmailAddress = @{
17                        Address = $Person.email
18                    }
19                }
20            )
21        }
22        SaveToSentItems = $true
23    }
24    Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```

Program output



ZK PROVER

12

```
Untitled1.ps1* X
1 Install-Module Microsoft.Graph -Scope CurrentUser
2 Connect-MgGraph -Scopes "Mail.Send"
3
4 $People = Import-Csv .\emails.csv
5
6 foreach ($Person in $People) {
7     $params = @{
8         Message = @{
9             Subject = "[replace with your subject line]"
10            Body = @{
11                ContentType = "Text"
12                Content = "[Replace with your email body]"
13            }
14            ToRecipients = @(
15                @{
16                    EmailAddress = @{
17                        Address = $Person.email
18                    }
19                }
20            )
21        }
22        SaveToSentItems = $true
23    }
24    Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```

ZERO-KNOWLEDGE VIRTUAL MACHINE

Public data



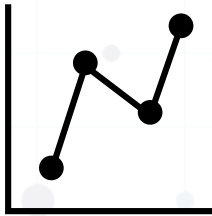
Private data



ZKVM

```
1 Install-Module Microsoft.Graph -Scope CurrentUser
2 Connect-MgGraph -Scopes "Mail.Send"
3
4 $People = Import-Csv .\emails.csv
5
6 foreach ($Person in $People) {
7     $params = @{
8         Message = @{
9             Subject = "[replace with your subject line]"
10            Body = @{
11                ContentType = "Text"
12                Content = "[Replace with your email body]"
13            }
14            ToRecipients = @(
15                @{
16                    EmailAddress = @{
17                        Address = $Person.email
18                    }
19                }
20            )
21        }
22        SaveToSentItems = $true
23    }
24    Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```

Program output



ZK PROVER

ZK proof



```
Untitled1.ps1* X
1 Install-Module Microsoft.Graph -Scope CurrentUser
2 Connect-MgGraph -Scopes "Mail.Send"
3
4 $People = Import-Csv .\emails.csv
5
6 foreach ($Person in $People) {
7     $params = @{
8         Message = @{
9             Subject = "[replace with your subject line]"
10            Body = @{
11                ContentType = "Text"
12                Content = "[Replace with your email body]"
13            }
14            ToRecipients = @(
15                @{
16                    EmailAddress = @{
17                        Address = $Person.email
18                    }
19                }
20            )
21        }
22        SaveToSentItems = $true
23    }
24    Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```

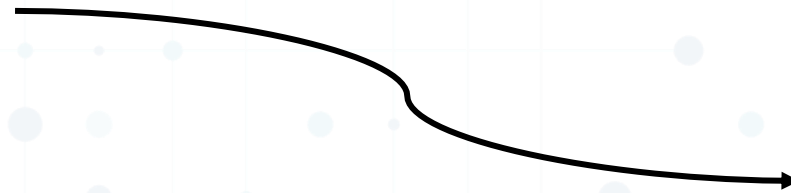
VERIFICATION

ZK VERIFIER

13

VERIFICATION

Public data



ZK VERIFIER

VERIFICATION

Public data



ZK VERIFIER

```
Untitled1.ps1 X
1  Install-Module Microsoft.Graph -Scope CurrentUser
2  Connect-MgGraph -Scopes "Mail.Send"
3
4  $People = Import-Csv .\emails.csv
5
6  foreach ($Person in $People) {
7      $params = @{
8          Message = @{
9              Subject = "[replace with your subject line]"
10             Body = @{
11                 ContentType = "Text"
12                 Content = "[Replace with your email body]"
13             }
14             ToRecipients = @(
15                 @{
16                     EmailAddress = @{
17                         Address = $Person.email
18                     }
19                 }
20             )
21         }
22         SaveToSentItems = $true
23     }
24     Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```

13

VERIFICATION

Public data



ZK VERIFIER

```
Untitled1.ps1* X
1  Install-Module Microsoft.Graph -Scope CurrentUser
2  Connect-MgGraph -Scopes "Mail.Send"
3
4  $People = Import-Csv .\emails.csv
5
6  foreach ($Person in $People) {
7      $params = @{
8          Message = @{
9              Subject = "[replace with your subject line]"
10             Body = @{
11                 ContentType = "Text"
12                 Content = "[Replace with your email body]"
13             }
14             ToRecipients = @(
15                 @{
16                     EmailAddress = @{
17                         Address = $Person.email
18                     }
19                 }
20             )
21         }
22         SaveToSentItems = $true
23     }
24     Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```

13



ZK Proof

VERIFICATION

Public data



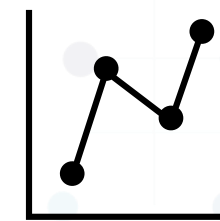
ZK VERIFIER

```
Untitled1.ps1* X
1  Install-Module Microsoft.Graph -Scope CurrentUser
2  Connect-MgGraph -Scopes "Mail.Send"
3
4  $People = Import-Csv .\emails.csv
5
6  foreach ($Person in $People) {
7      $params = @{
8          Message = @{
9              Subject = "[replace with your subject line]"
10             Body = @{
11                 ContentType = "Text"
12                 Content = "[Replace with your email body]"
13             }
14             ToRecipients = @(
15                 @{
16                     EmailAddress = @{
17                         Address = $Person.email
18                     }
19                 }
20             )
21         }
22         SaveToSentItems = $true
23     }
24     Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```

13



ZK Proof



Program output

VERIFICATION

Public data



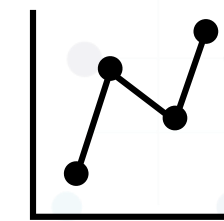
ZK VERIFIER

Output correct

```
Untitled1.ps1* X
1  Install-Module Microsoft.Graph -Scope CurrentUser
2  Connect-MgGraph -Scopes "Mail.Send"
3
4  $People = Import-Csv .\emails.csv
5
6  foreach ($Person in $People) {
7      $params = @{
8          Message = @{
9              Subject = "[replace with your subject line]"
10             Body = @{
11                 ContentType = "Text"
12                 Content = "[Replace with your email body]"
13             }
14             ToRecipients = @(
15                 @{
16                     EmailAddress = @{
17                         Address = $Person.email
18                     }
19                 }
20             )
21         }
22         SaveToSentItems = $true
23     }
24     Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```



ZK Proof



Program output

13

VERIFICATION

Public data



Private data



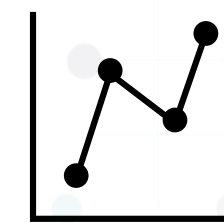
ZK VERIFIER

Output correct

```
Untitled1.ps1* X
1  Install-Module Microsoft.Graph -Scope CurrentUser
2  Connect-MgGraph -Scopes "Mail.Send"
3
4  $People = Import-Csv .\emails.csv
5
6  foreach ($Person in $People) {
7      $params = @{
8          Message = @{
9              Subject = "[replace with your subject line]"
10             Body = @{
11                 ContentType = "Text"
12                 Content = "[Replace with your email body]"
13             }
14             ToRecipients = @(
15                 @{
16                     EmailAddress = @{
17                         Address = $Person.email
18                     }
19                 }
20             )
21         }
22         SaveToSentItems = $true
23     }
24     Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```



ZK Proof



Program output

VERIFICATION

Public data



Private data



ZK VERIFIER

```
Untitled1.ps1* X
1  Install-Module Microsoft.Graph -Scope CurrentUser
2  Connect-MgGraph -Scopes "Mail.Send"
3
4  $People = Import-Csv .\emails.csv
5
6  foreach ($Person in $People) {
7      $params = @{
8          Message = @{
9              Subject = "[replace with your subject line]"
10             Body = @{
11                 ContentType = "Text"
12                 Content = "[Replace with your email body]"
13             }
14             ToRecipients = @(
15                 @{
16                     EmailAddress = @{
17                         Address = $Person.email
18                     }
19                 }
20             )
21         }
22         SaveToSentItems = $true
23     }
24     Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```



ZK Proof



Wrong output

VERIFICATION

Public data



Private data



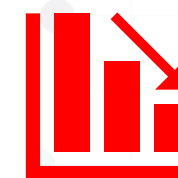
ZK VERIFIER

Output incorrect

```
Untitled1.ps1* X
1  Install-Module Microsoft.Graph -Scope CurrentUser
2  Connect-MgGraph -Scopes "Mail.Send"
3
4  $People = Import-Csv .\emails.csv
5
6  foreach ($Person in $People) {
7      $params = @{
8          Message = @{
9              Subject = "[replace with your subject line]"
10             Body = @{
11                 ContentType = "Text"
12                 Content = "[Replace with your email body]"
13             }
14             ToRecipients = @(
15                 @{
16                     EmailAddress = @{
17                         Address = $Person.email
18                     }
19                 }
20             )
21         }
22         SaveToSentItems = $true
23     }
24     Send-MgUserMail -UserId $Person.email -BodyParameter $params
25 }
26 }
```



ZK Proof



Wrong output

WHY CARE?



Privacy

WHY CARE?



Privacy

- Private data stays private.

WHY CARE?



Privacy

- Private data stays private.
- Proof leaks **ZERO** unnecessary information

WHY CARE?



Privacy

- Private data stays private.
- Proof leaks **ZERO** unnecessary information



15

Succinctness (SNARK)

WHY CARE?



Privacy

- Private data stays private.
- Proof leaks **ZERO** unnecessary information



15

Succinctness (SNARK)

- Private data \gg proof size

WHY CARE?



Privacy

- Private data stays private.
- Proof leaks **ZERO** unnecessary information



15

Succinctness (SNARK)

- Private data \gg proof size
 - 1GB private data \rightarrow Few KB proof size

WHY CARE?



Privacy

- Private data stays private.
- Proof leaks **ZERO** unnecessary information



15

Succinctness (SNARK)

- Private data \gg proof size
 - 1GB private data \rightarrow Few KB proof size
- Slow program, fast proof verification

WHY CARE?



Privacy

- Private data stays private.
- Proof leaks **ZERO** unnecessary information



15

Succinctness (SNARK)

- Private data \gg proof size
 - 1GB private data \rightarrow Few KB proof size
- Slow program, fast proof verification
 - 1 hour \rightarrow less than 1 second

INTUITION

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

INTUITION

SLOW TO SOLVE

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

INTUITION

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

INTUITION

FAST TO VERIFY

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

INTUITION

FAST TO VERIFY
NON-SUCCINCT PROOF

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

INTUITION

SUCCINCT PROOF

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

INTUITION

SUCCINCT PROOF



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



HOW IS IT EVEN POSSIBLE?

19

HOW IS IT EVEN POSSIBLE?

19

Firstly, if $K \leq n + d + 1$, then $\deg(L) \leq K - 1 \leq n + d$, contradicting the claim that $\deg(L) > n + d$. Thus, we assume in the following that $K \geq n + d + 2$.

Let $S_1 = \{\alpha_i\}_{i=1}^{n+d+1}$ and $L_1(X) := \text{Inter}(\{\alpha_i, y_i\}_{i=1}^{n+d+1})$ be the interpolation polynomial over the respective points. Then, L_1 is a unique polynomial of degree at most $n + d$ such that $L_1(\alpha_i) = y_i$ for all $i \in [1..n + d + 1]$. There exists j^* such that $L(\alpha_{j^*}) \neq L_1(\alpha_{j^*})$ (otherwise $L = L_1$ and $\deg(L) \leq n + d$). Let $L_2(X) := \text{Inter}(\{\alpha_i, y_i\}_{i=1}^{n+d} \cup \{\alpha_{j^*}, y_{j^*}\})$. It also holds that $L_1(X) \neq L_2(X)$ because $L_1(\alpha_{j^*}) \neq L_2(\alpha_{j^*})$.

Let $\ell_i^{S_1}(X) \in \mathbb{F}^{\leq n+d}[X]$ be the i th Lagrange polynomial over the set S_1 (see Eq. (1)) for $i \in [1..n + d + 1]$. Then, $L_1(X) = \sum_{i=1}^{n+d+1} y_i \ell_i^{S_1}(X)$. Recall also that

$$\ell_i^{S_1}(X) = d_i^{S_1} Z_{S_1}(X) / (X - \alpha_i) , \quad (3)$$

where $d_i^{S_1} = 1 / \prod_{j \in [1..n+d+1] \setminus \{i\}} (\alpha_i - \alpha_j)$. Now, observe that

$$\begin{aligned} \sum_{j=1}^m g_j(x) c_j - L_1(x) &= \\ \sum_{j=1}^m \left(\sum_{i=1}^{n+d+1} g_j(\alpha_i) \ell_i^{S_1}(x) \right) c_j - \sum_{i=1}^{n+d+1} y_i \ell_i^{S_1}(x) &= \\ \sum_{i=1}^{n+d+1} \ell_i^{S_1}(x) \left(\sum_{j=1}^m g_j(\alpha_i) c_j - y_i \right) &\stackrel{(2)}{=} \sum_{i=1}^{n+d+1} \ell_i^{S_1}(x) \pi_i(x - \alpha_i) \stackrel{(3)}{=} \\ \left(\sum_{i=1}^{n+d+1} d_i^{S_1} \pi_i \right) Z_{S_1}(x) &= \hat{\pi}_1 Z_{S_1}(x) . \end{aligned}$$

where $\hat{\pi}_1 = \sum_{i=1}^{n+d+1} d_i^{S_1} \pi_i$. On the first step, we used that $\deg(g_j) \leq d$ and thus $g_j(X) = \sum_{i=1}^{n+d+1} g_j(\alpha_i) \ell_i^{S_1}(X)$ and similarly $L_1(X) = \sum_{i=1}^{n+d+1} y_i \ell_i^{S_1}(X)$.

By defining $S_2 = \{\alpha_i\}_{i=1}^{n+d} \cup \{\alpha_{j^*}\}$, we can apply the same derivations to S_2 and $L_2(X)$. Thus, we obtain that for $b \in \{1, 2\}$,

$$\sum_{j=1}^m g_j(x) c_j - L_b(x) = \hat{\pi}_b Z_{S_b}(x) . \quad (4)$$

LOT OF COMPLICATED MATH

HOW IS IT EVEN POSSIBLE?

UNIMPORTANT TO UNDERSTAND APPLICATIONS

19

Firstly, if $K \leq n + d + 1$, then $\deg(L) \leq K - 1 \leq n + d$, contradicting the claim that $\deg(L) > n + d$. Thus, we assume in the following that $K \geq n + d + 2$.

Let $S_1 = \{\alpha_i\}_{i=1}^{n+d+1}$ and $L_1(X) := \text{Inter}(\{\alpha_i, y_i\}_{i=1}^{n+d+1})$ be the interpolation polynomial over the respective points. Then, L_1 is a unique polynomial of degree at most $n + d$ such that $L_1(\alpha_i) = y_i$ for all $i \in [1..n + d + 1]$. There exists j^* such that $L(\alpha_{j^*}) \neq L_1(\alpha_{j^*})$ (otherwise $L = L_1$ and $\deg(L) \leq n + d$). Let $L_2(X) := \text{Inter}(\{\alpha_i, y_i\}_{i=1}^{n+d} \cup \{\alpha_{j^*}, y_{j^*}\})$. It also holds that $L_1(X) \neq L_2(X)$ because $L_1(\alpha_{j^*}) \neq L_2(\alpha_{j^*})$.

Let $\ell_i^{S_1}(X) \in \mathbb{F}^{\leq n+d}[X]$ be the i th Lagrange polynomial over the set S_1 (see Eq. (1)) for $i \in [1..n + d + 1]$. Then, $L_1(X) = \sum_{i=1}^{n+d+1} y_i \ell_i^{S_1}(X)$. Recall also that

$$\ell_i^{S_1}(X) = d_i^{S_1} Z_{S_1}(X) / (X - \alpha_i) , \quad (3)$$

where $d_i^{S_1} = 1 / \prod_{j \in [1..n+d+1] \setminus \{i\}} (\alpha_i - \alpha_j)$. Now, observe that

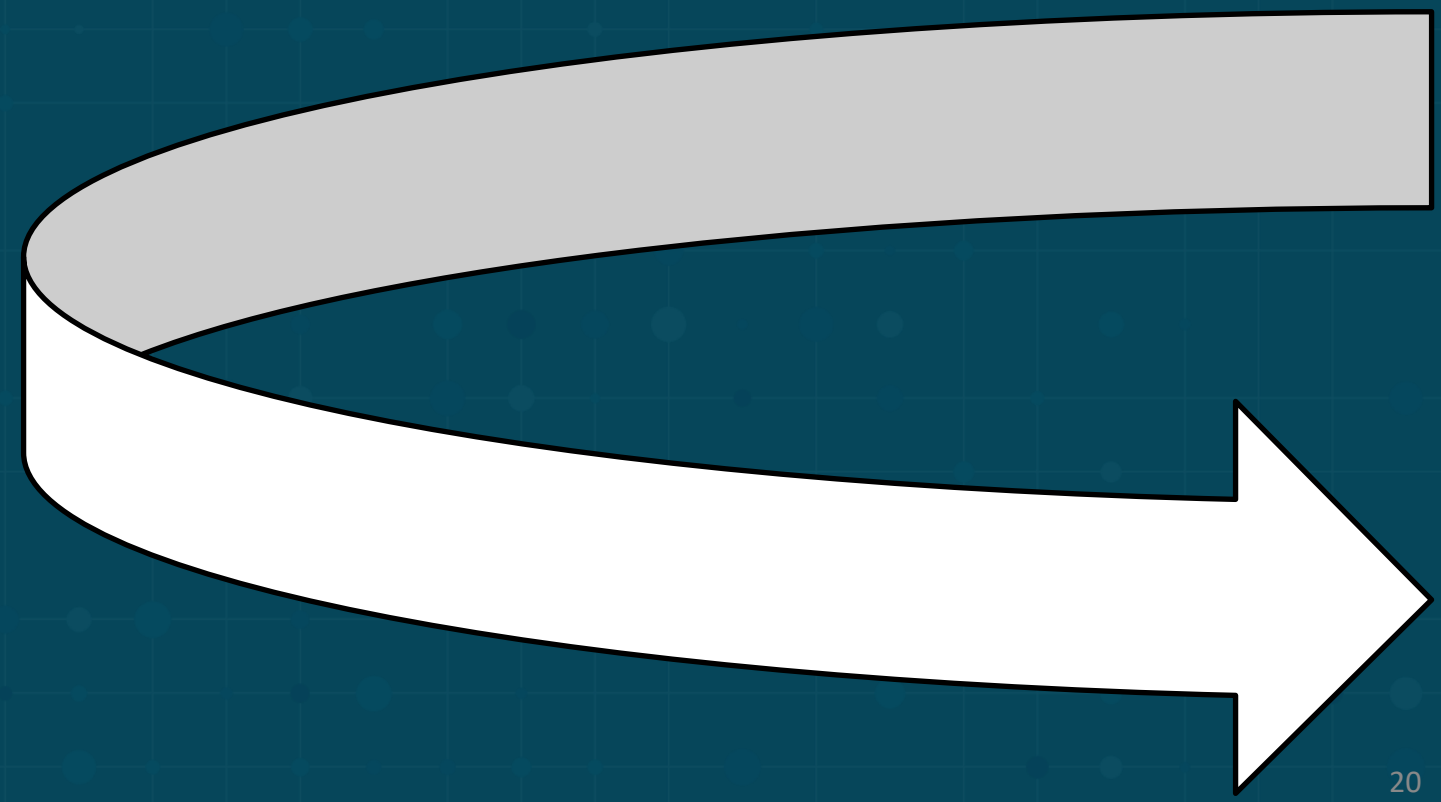
$$\begin{aligned} \sum_{j=1}^m g_j(x) c_j - L_1(x) &= \\ \sum_{j=1}^m \left(\sum_{i=1}^{n+d+1} g_j(\alpha_i) \ell_i^{S_1}(x) \right) c_j - \sum_{i=1}^{n+d+1} y_i \ell_i^{S_1}(x) &= \\ \sum_{i=1}^{n+d+1} \ell_i^{S_1}(x) \left(\sum_{j=1}^m g_j(\alpha_i) c_j - y_i \right) &\stackrel{(2)}{=} \sum_{i=1}^{n+d+1} \ell_i^{S_1}(x) \pi_i(x - \alpha_i) \stackrel{(3)}{=} \\ \left(\sum_{i=1}^{n+d+1} d_i^{S_1} \pi_i \right) Z_{S_1}(x) &= \hat{\pi}_1 Z_{S_1}(x) . \end{aligned}$$

where $\hat{\pi}_1 = \sum_{i=1}^{n+d+1} d_i^{S_1} \pi_i$. On the first step, we used that $\deg(g_j) \leq d$ and thus $g_j(X) = \sum_{i=1}^{n+d+1} g_j(\alpha_i) \ell_i^{S_1}(X)$ and similarly $L_1(X) = \sum_{i=1}^{n+d+1} y_i \ell_i^{S_1}(X)$.

By defining $S_2 = \{\alpha_i\}_{i=1}^{n+d} \cup \{\alpha_{j^*}\}$, we can apply the same derivations to S_2 and $L_2(X)$. Thus, we obtain that for $b \in \{1, 2\}$,

$$\sum_{j=1}^m g_j(x) c_j - L_b(x) = \hat{\pi}_b Z_{S_b}(x) . \quad (4)$$

LOT OF COMPLICATED MATH



20



BACK TO BLOCKCHAIN

ZCASH

21

ZCASH



ZCash

Cryptocurrency launched in 2016

ZCASH



ZCash

Cryptocurrency launched in 2016



Private transaction

Transaction are encrypted

ZCASH



ZCash

Cryptocurrency launched in 2016



Private transaction

Transaction are encrypted



Zero-knowledge proofs

Main tool to avoid double spending

21

ZCASH



ZCash

Cryptocurrency launched in 2016



Private transaction

Transaction are encrypted



Zero-knowledge proofs

Main tool to avoid double spending



Success

Still one of the top cryptocurrencies

MAIN INTUITION



Immutable database

From	To	Amount	Signature
Alice	Bob	50 euro	<i>Alice</i>
Bob	Eve	20 euros	<i>Bobby</i>
Alice	Eve	10 euro	<i>Alice²²</i>

- Check that no double spending
- Signature valid



Miners



Users

MAIN INTUITION



Immutable database

From	To	Amount	Signature
0x43F453	0x23BA2	0xFF231	0xBF9A1
0x1F271	0xF8231	0x5F231	0x4F9A3
0xAF2A1	0xBF9A1	0x3F3A1	0xBD9A1



Miners



Users

MAIN INTUITION



Immutable database

From	To	Amount	Signature
0x43F453	0x23BA2	0xFF231	0xBF9A1
0x1F271	0xF8231	0x5F231	0x4F9A3
0xAF2A1	0xBF9A1	0x3F3A1	0xBD9A1

How to avoid double spending?



Miners



Users

MAIN INTUITION



Immutable database

From	To	Amount	Signature
0x43F453	0x23BA2	0xFF231	0xBF9A1
0x1F271	0xF8231	0x5F231	0x4F9A3
0xAF2A1	0xBF9A1	0x3F3A1	0xBD9A1



Miners



Users

MAIN INTUITION

Immutable database

From	To	Amount	Signature
0x43F453	0x23BA2	0xFF231	0xBF9A1
0x1F271	0xF8231	0x5F231	0x4F9A3
0xAF2A1	0xBF9A1	0x3F3A1	0xBD9A1



Miners



Enc (tx) + ZK proof of correctness



Users



MAIN INTUITION



Immutable database

From	To	Amount	Signature
0x43F453	0x23BA2	0xFF231	0xBF9A1
0x1F271	0xF8231	0x5F231	0x4F9A3
0xAF2A1	0xBF9A1	0x3F3A1	0xBD9A1

Verify proof



Miners

Enc (tx) + ZK proof of correctness



Users

MAIN INTUITION



Immutable database

From	To	Amount	Signature
0x43F453	0x23BA2	0xFF231	0xBF9A1
0x1F271	0xF8231	0x5F231	0x4F9A3
0xAF2A1	0xBF9A1	0x3F3A1	0xBD9A1

Verify proof



Miners

Enc (tx) + ZK proof of correctness



Users

MAIN INTUITION



Immutable database

From	To	Amount	Signature
0x43F453	0x23BA2	0xFF231	0xBF9A1
0x1F271	0xF8231	0x5F231	0x4F9A3
0xAF2A1	0xBF9A1	0x3F3A1	0xBD9A1
0x2F241	0xA8331	0x6F331	0x1A9A3

Verify proof



Miners

Enc (tx) + ZK proof of correctness



Users

MAIN MOTIVATORS

25

MAIN MOTIVATORS

1 Privacy is important



25

MAIN MOTIVATORS

1 Privacy is important

2 Scalability is important



MAIN MOTIVATORS

1 Privacy is important

2 Scalability is importanter

3 Scalability is ²⁵main reason for ZK hype nowadays

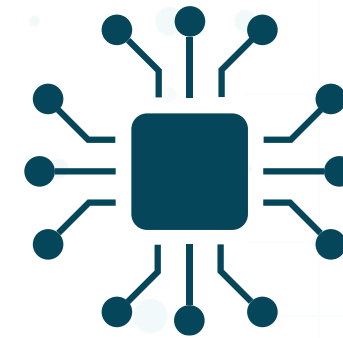


BLOCKCHAIN LAYERS



**Layer 1
blockchains**

26



**Layer 2
blockchains**

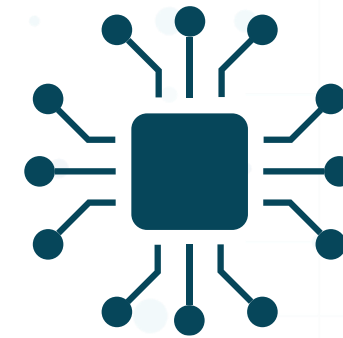
BLOCKCHAIN LAYERS



Layer 1 blockchains

- Actual consensus mechanism

26



Layer 2 blockchains

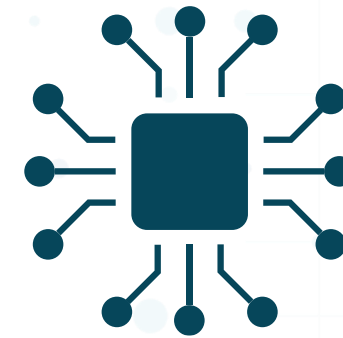
BLOCKCHAIN LAYERS



Layer 1 blockchains

- Actual consensus mechanism
- Secure

26



Layer 2 blockchains

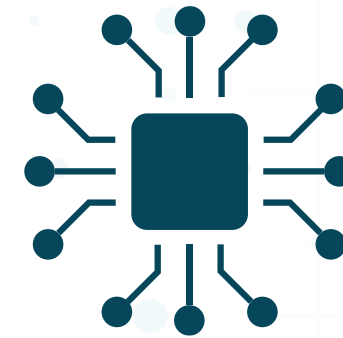
BLOCKCHAIN LAYERS



Layer 1 blockchains

- Actual consensus mechanism
- Secure
- Decentralized

26



Layer 2 blockchains

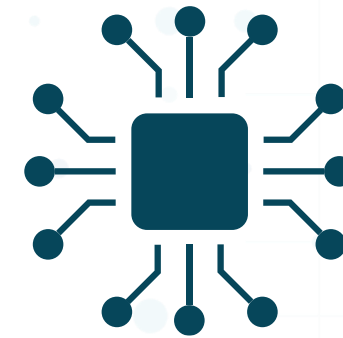
BLOCKCHAIN LAYERS



Layer 1 blockchains

- Actual consensus mechanism
- Secure
- Decentralized
- Slow

26



Layer 2 blockchains

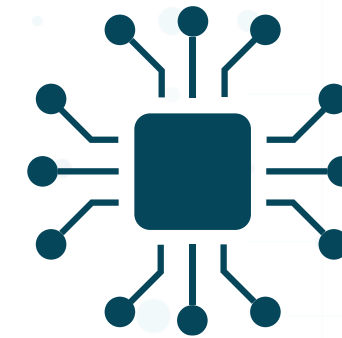
BLOCKCHAIN LAYERS



Layer 1 blockchains

- Actual consensus mechanism
- Secure
- Decentralized
- Slow

26



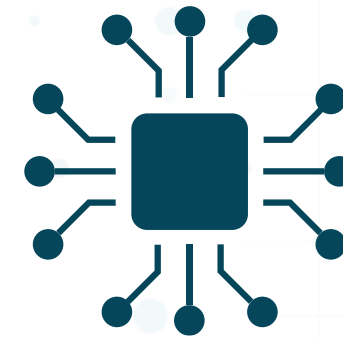
Layer 2 blockchains

BLOCKCHAIN LAYERS



Layer 1 blockchains

- Actual consensus mechanism
- Secure
- Decentralized
- Slow



Layer 2 blockchains

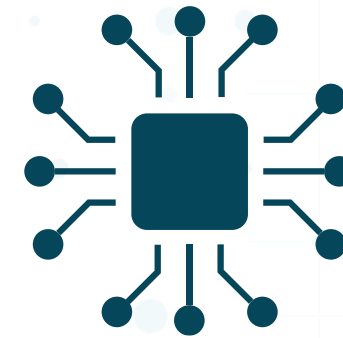
BLOCKCHAIN LAYERS



Layer 1 blockchains

- Actual consensus mechanism
- Secure
- Decentralized
- Slow

26



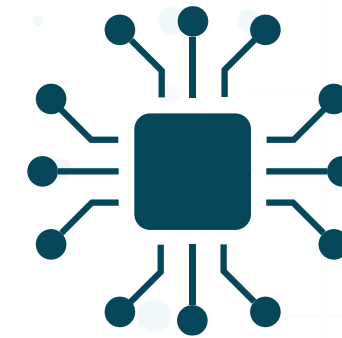
Layer 2 blockchains

BLOCKCHAIN LAYERS



Layer 1 blockchains

- Actual consensus mechanism
- Secure
- Decentralized
- Slow



Layer 2 blockchains

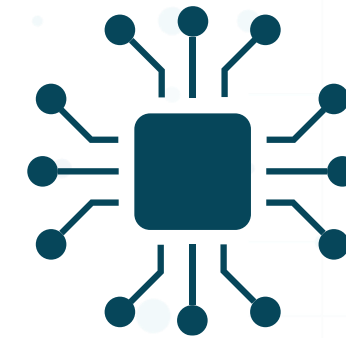
- Depend on layer 1 for security

BLOCKCHAIN LAYERS



Layer 1 blockchains

- Actual consensus mechanism
- Secure
- Decentralized
- Slow



Layer 2 blockchains

- Depend on layer 1 for security
- Much faster

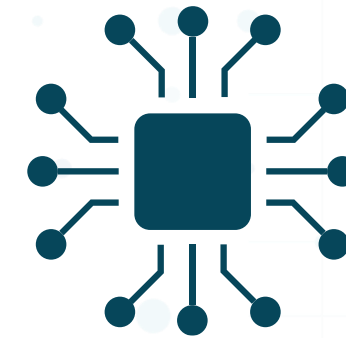
26

BLOCKCHAIN LAYERS



Layer 1 blockchains

- Actual consensus mechanism
- Secure
- Decentralized
- Slow



Layer 2 blockchains

- Depend on layer 1 for security
- Much faster



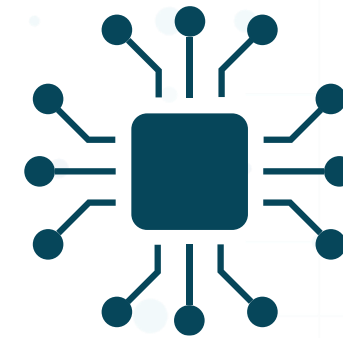
Lightning Network®

BLOCKCHAIN LAYERS



Layer 1 blockchains

- Actual consensus mechanism
- Secure
- Decentralized
- Slow



Layer 2 blockchains

- Depend on layer 1 for security
- Much faster



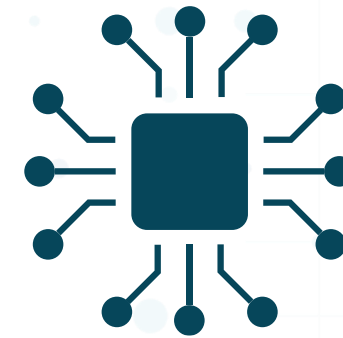
BLOCKCHAIN LAYERS



Layer 1 blockchains

- Actual consensus mechanism
- Secure
- Decentralized
- Slow

26



Layer 2 blockchains

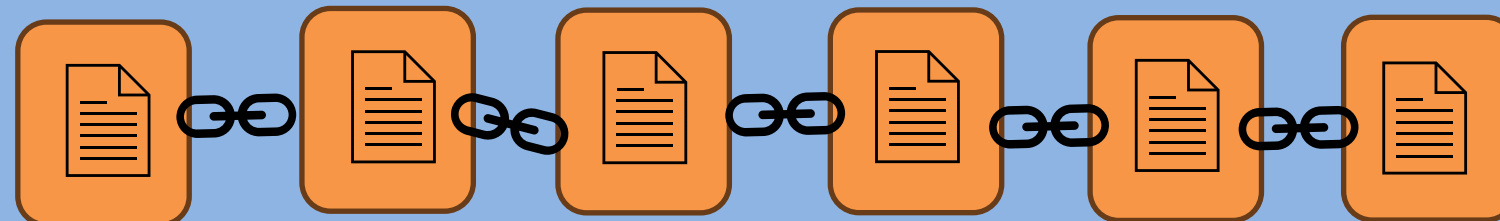
- Depend on layer 1 for security
- Much faster



ZK-ROLLUP

Layer 2
blockchain

Layer 1
blockchain



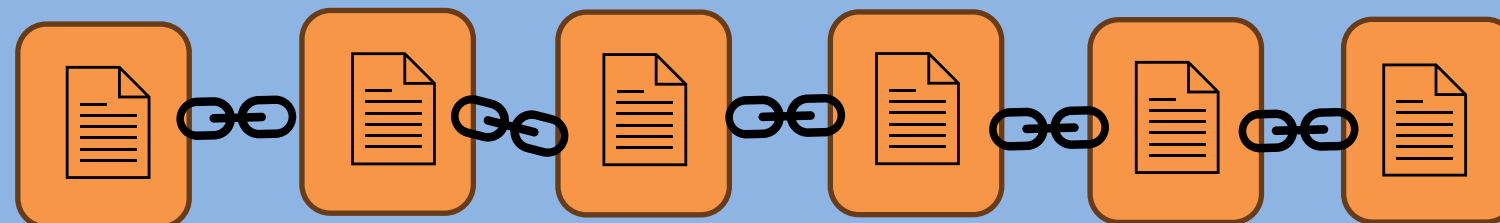
ZK-ROLLUP

Layer 2
blockchain



Super Node

Layer 1
blockchain

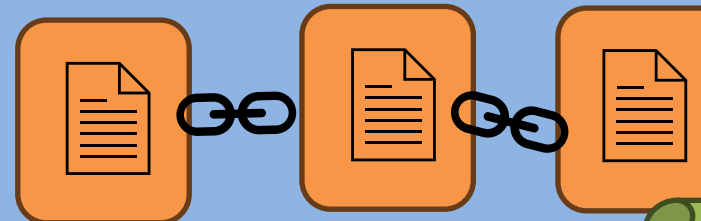


ethereum

ZK-ROLLUP

Layer 2
blockchain

Layer 1
blockchain



New cryptocurrency
smart contract

STATE



Super Node



ethereum

27

ZK-ROLLUP

Layer 2
blockchain



Users



Super Node

Layer 1
blockchain



New cryptocurrency
smart contract

STATE



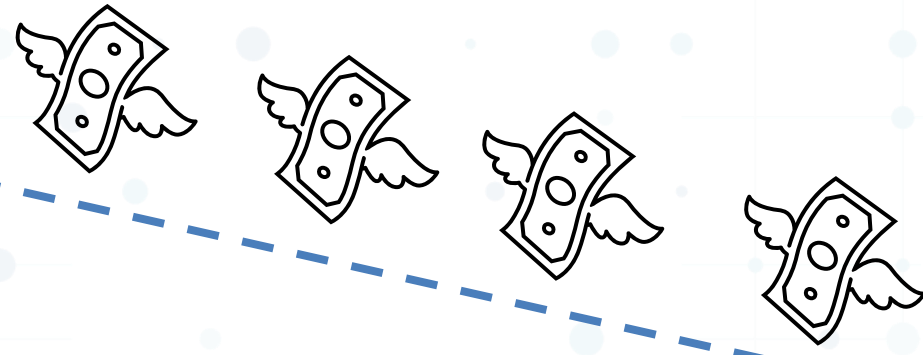
27

ZK-ROLLUP

Layer 2
blockchain



Users



Super Node

Layer 1
blockchain



New cryptocurrency
smart contract

STATE



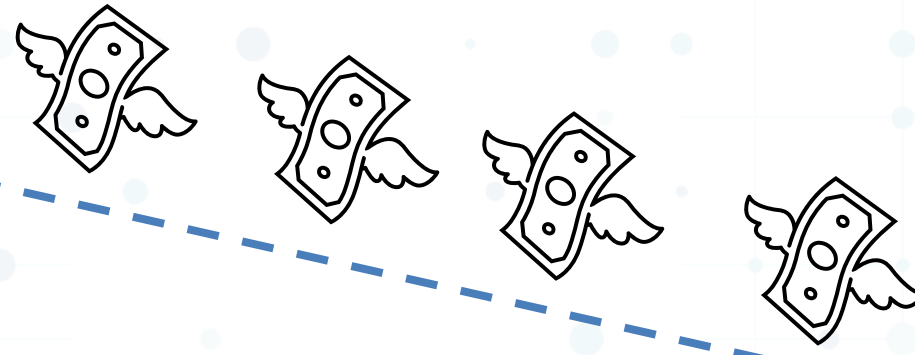
27

ZK-ROLLUP

Layer 2
blockchain



Users



Verify/execute
tx-s

Super Node

Layer 1
blockchain



New cryptocurrency
smart contract

STATE



27

ZK-ROLLUP

Layer 2
blockchain



Users



Compressed tx-s
+
Succinct Proof



Super Node

Verify/execute
tx-s

Layer 1
blockchain



New cryptocurrency
smart contract

STATE



COMPRESSION: EASY EXAMPLE

From	To	Amount	Signature
Alice	Bob	50 euro	<i>Alice</i>
Bob	Eve	20 euros	<i>Bobby</i>
Alice	Eve	10 euro	<i>Alice</i>



Super Node

COMPRESSION: EASY EXAMPLE

From	To	Amount	Signature
Alice	Bob	50 euro	Alice
Bob	Eve	20 euros	Bob
Alice	Eve	10 euro	Alice



Super Node

COMPRESSION: EASY EXAMPLE

From	To	Amount	Signature
Alice	Bob	50 euro	Alice
Bob	Eve	20 euros	Bob
Alice	Eve	10 euro	Alice



Super Node



Succinct proof that signatures verified

ADVANTAGES

29

ADVANTAGES

1

Transactions now take much less space on Layer 1

ADVANTAGES

- 1** Transactions now take much less space on Layer 1
- 2** Layer 1 only verifies proof (fast!)

ADVANTAGES

- 1** Transactions now take much less space on Layer 1
- 2** Layer 1 only verifies proof (fast!)
- 3** More tricks (blobs, external storage)

29

ADVANTAGES

- 1** Transactions now take much less space on Layer 1
- 2** Layer 1 only verifies proof (fast!)
- 3** More tricks (blobs, external storage)
- 4** 10tx/sec → 3000+ tx/sec

29

MORE APPLICATIONS: PROOF OF SOLVENCY



Customers



Crypto bank

30



MORE APPLICATIONS: PROOF OF SOLVENCY



Customers



Crypto bank

30



MORE APPLICATIONS: PROOF OF SOLVENCY



Customers



Crypto bank



MORE APPLICATIONS: PROOF OF SOLVENCY

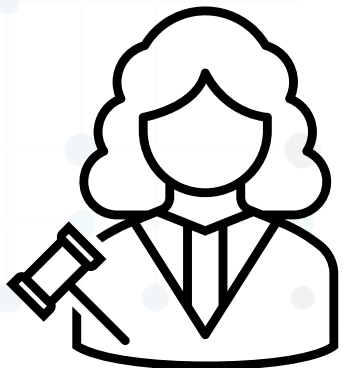


Customers



Crypto bank

30



Auditor

MORE APPLICATIONS: PROOF OF SOLVENCY



Customers



Crypto bank



Are you solvent?



Auditor

30

MORE APPLICATIONS: PROOF OF SOLVENCY



Customers



Crypto bank



Are you solvent?



Auditor

ZK proof of solvency

30

MORE APPLICATIONS: PROOF OF SOLVENCY



Customers



Crypto bank



ZK proof of solvency

Are you solvent?

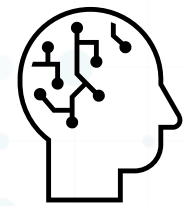
30

Missing Bitcoins: Inside the Mt. Gox Crypto Heist

In 2014, Bitcoins worth 470 million USD vanished from an exchange in Tokyo. Where did they go?

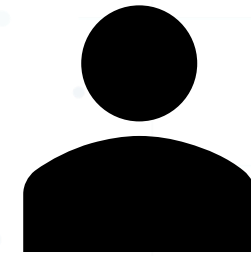
Auditor

MORE APPLICATIONS: ZK MACHINE LEARNING

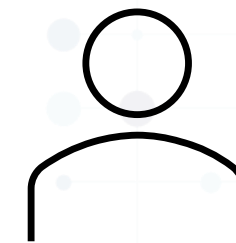


ML Model

31

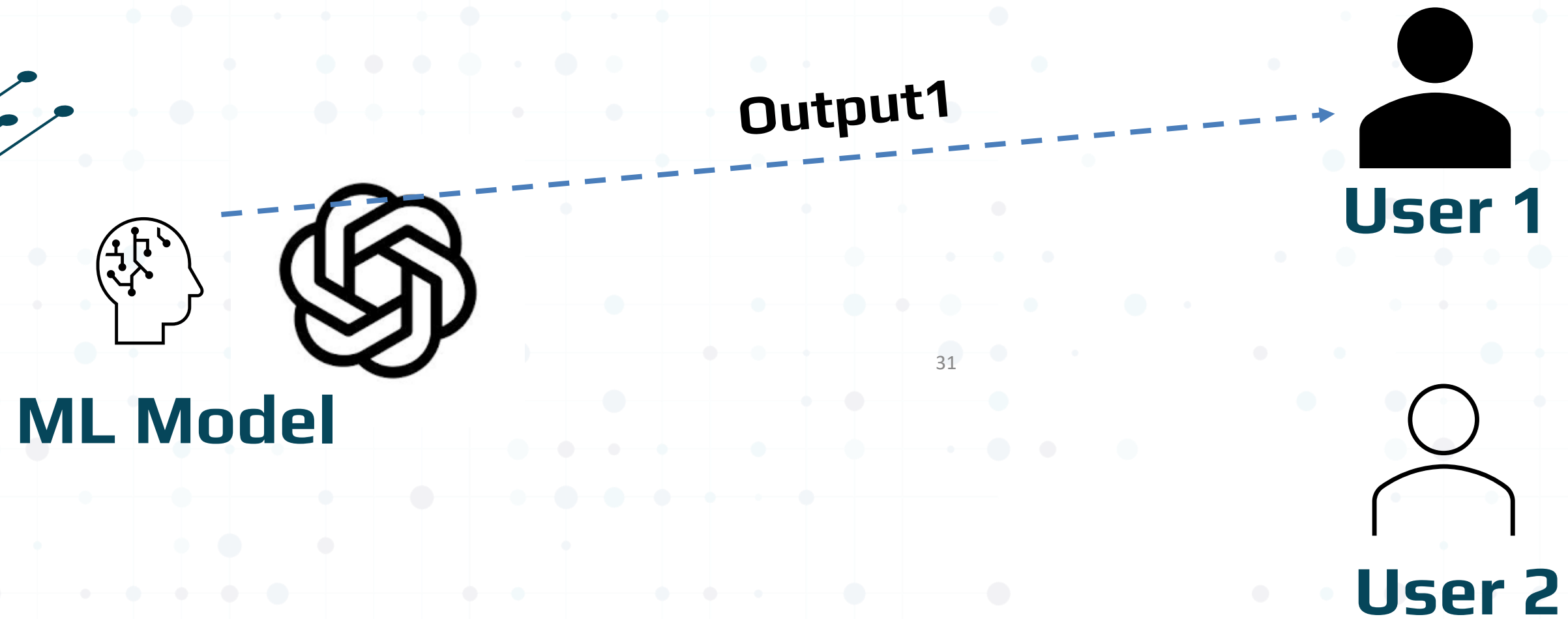


User 1



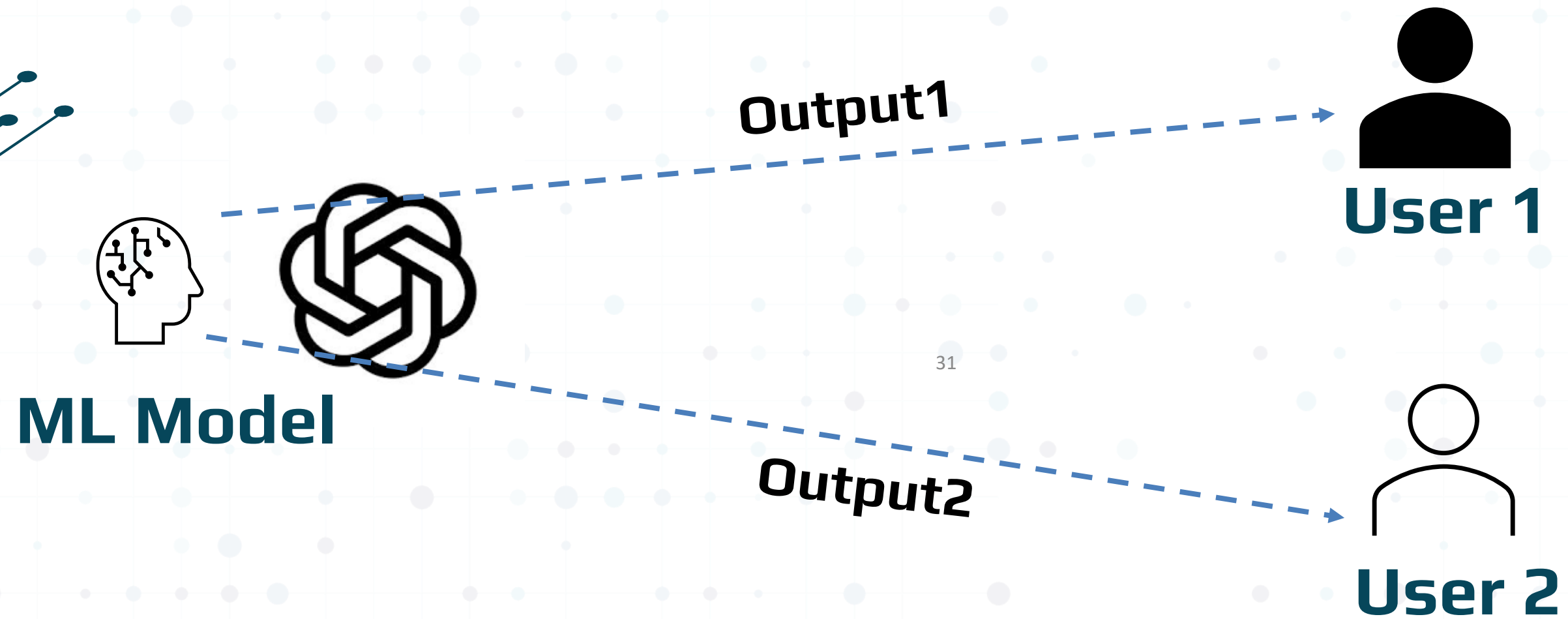
User 2

MORE APPLICATIONS: ZK MACHINE LEARNING



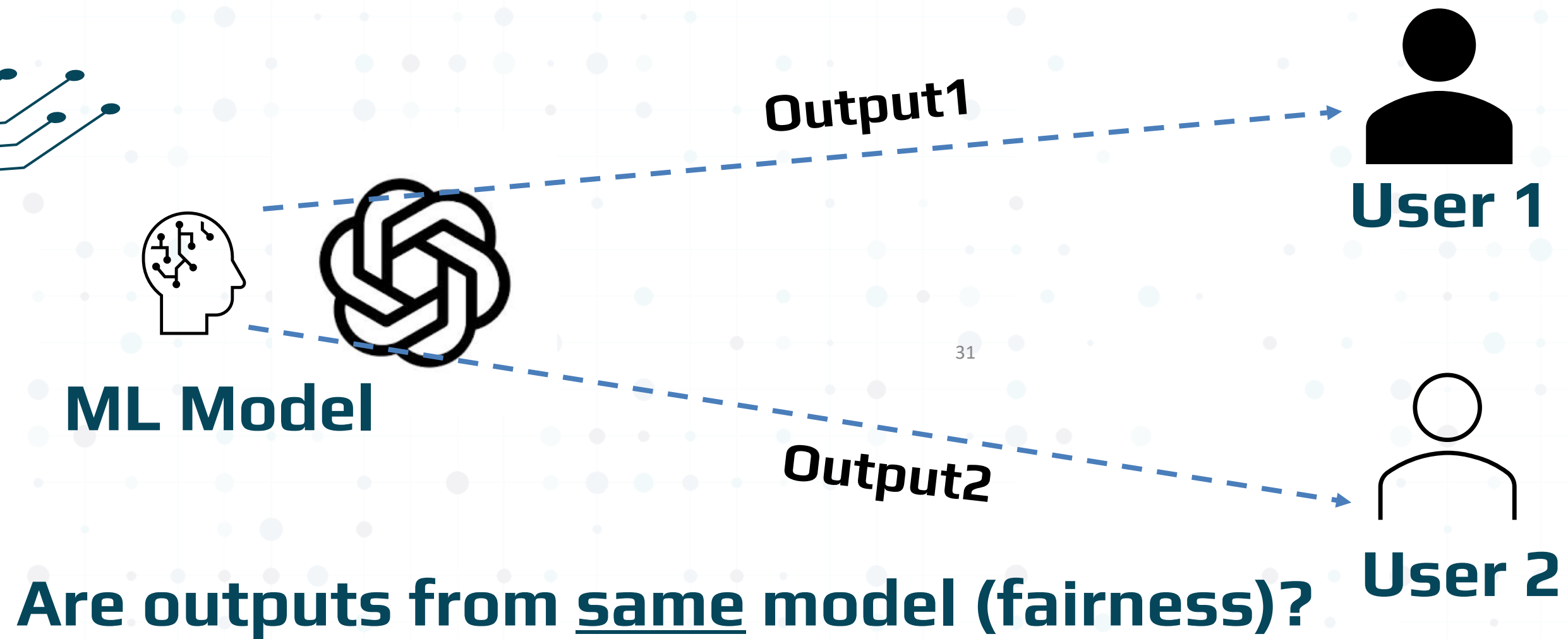
31

MORE APPLICATIONS: ZK MACHINE LEARNING

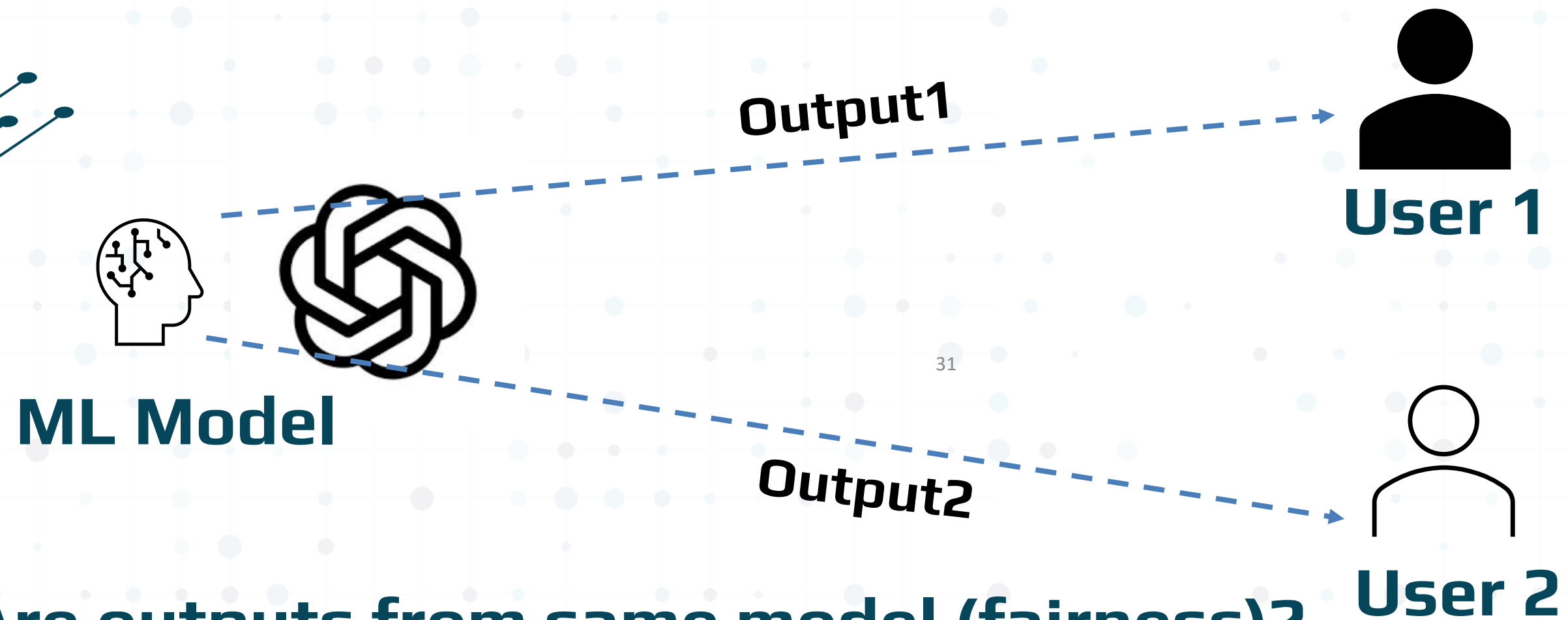


31

MORE APPLICATIONS: ZK MACHINE LEARNING



MORE APPLICATIONS: ZK MACHINE LEARNING

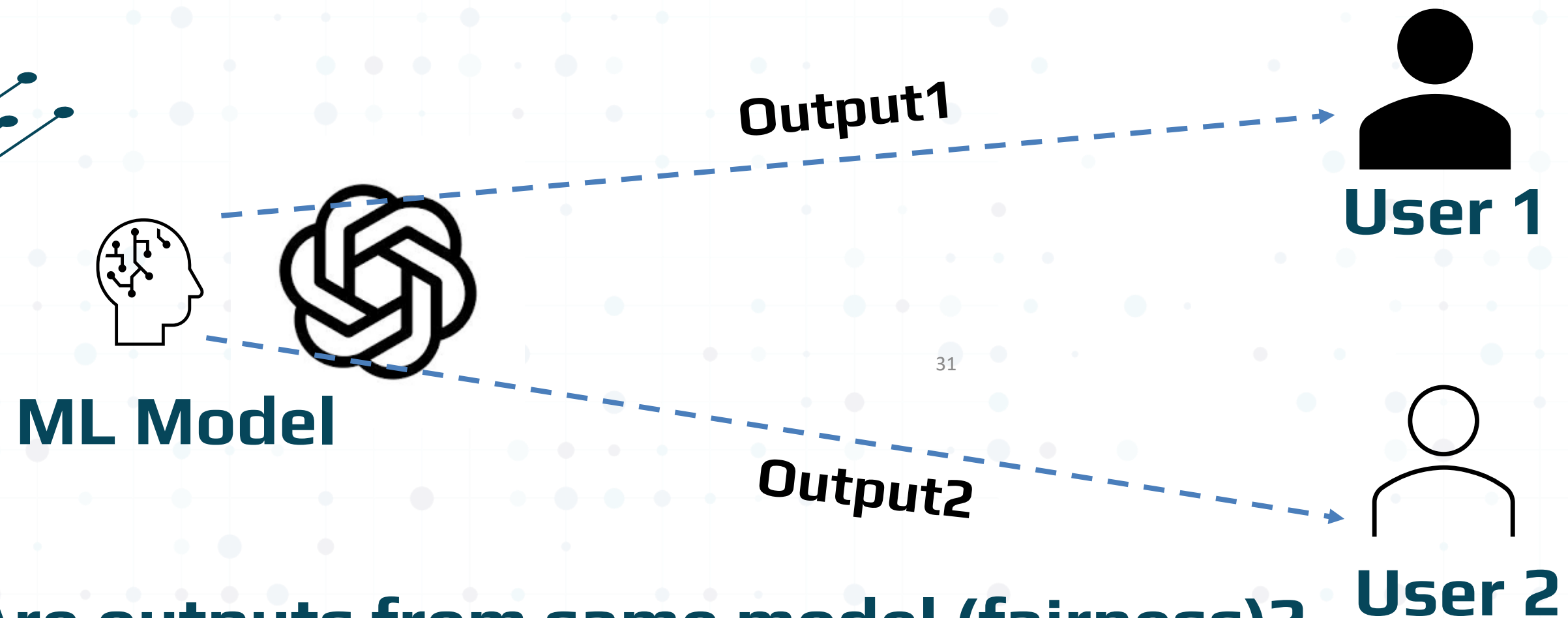


31

Are outputs from same model (fairness)?

- fair loans, fair prices, correct model,...

MORE APPLICATIONS: ZK MACHINE LEARNING



31

- Are outputs from same model (fairness)?
- fair loans, fair prices, correct model,...
 - give a ZK proof

MORE APPLICATIONS: FIGHTING FAKE NEWS

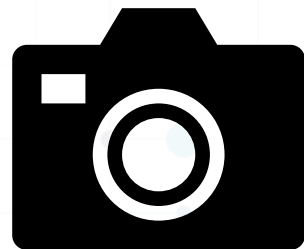
32

MORE APPLICATIONS: FIGHTING FAKE NEWS

How check that photo is authentic?

MORE APPLICATIONS: FIGHTING FAKE NEWS

How check that photo is authentic?



32

MORE APPLICATIONS: FIGHTING FAKE NEWS

How check that photo is authentic?



32

MORE APPLICATIONS: FIGHTING FAKE NEWS

How check that photo is authentic?



MORE APPLICATIONS: FIGHTING FAKE NEWS

How check that photo is authentic?



VISION OF FUTURE

33

Real World

VISION OF FUTURE

33

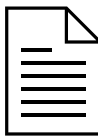
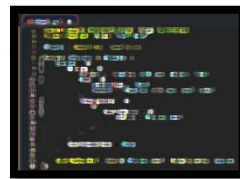


Real World



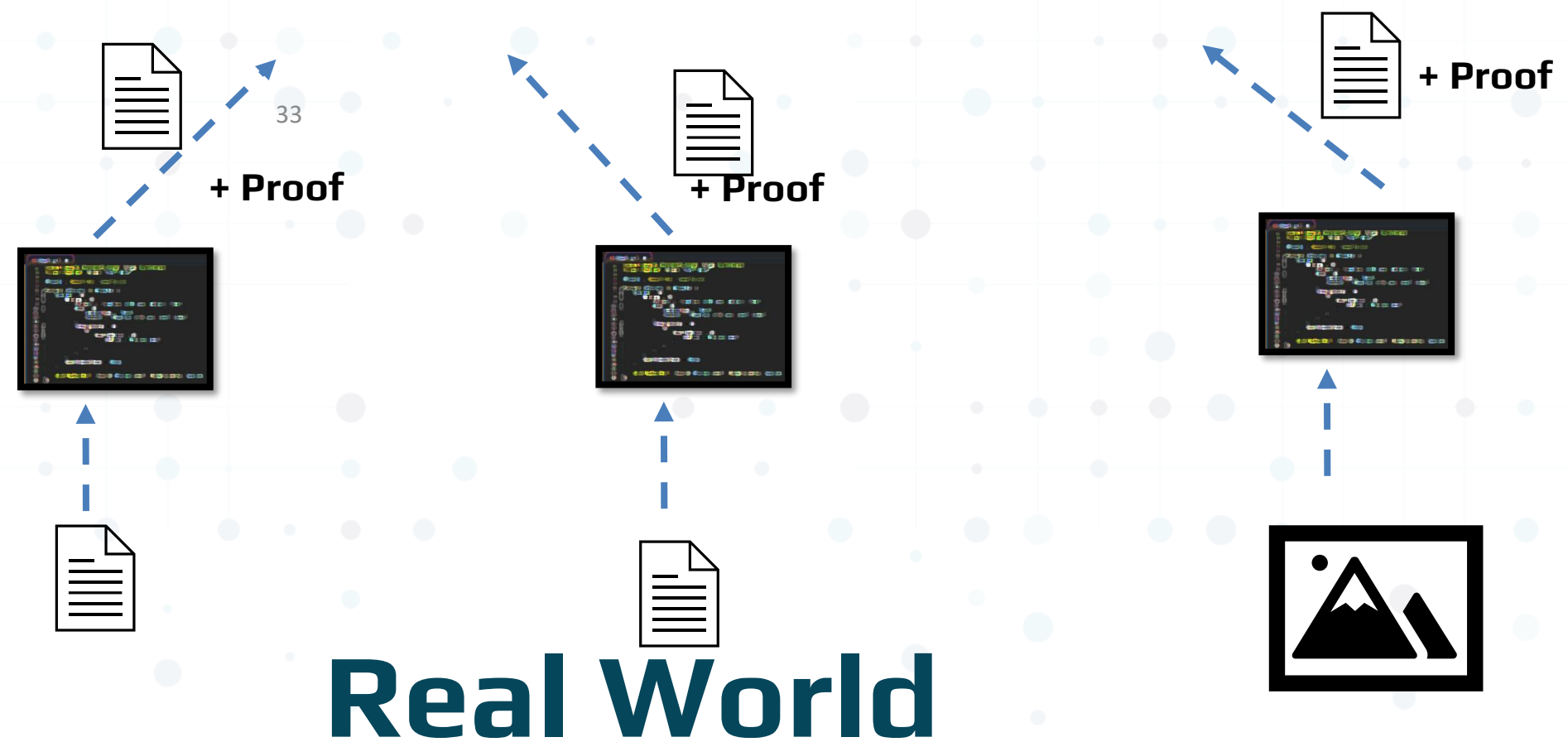
VISION OF FUTURE

33

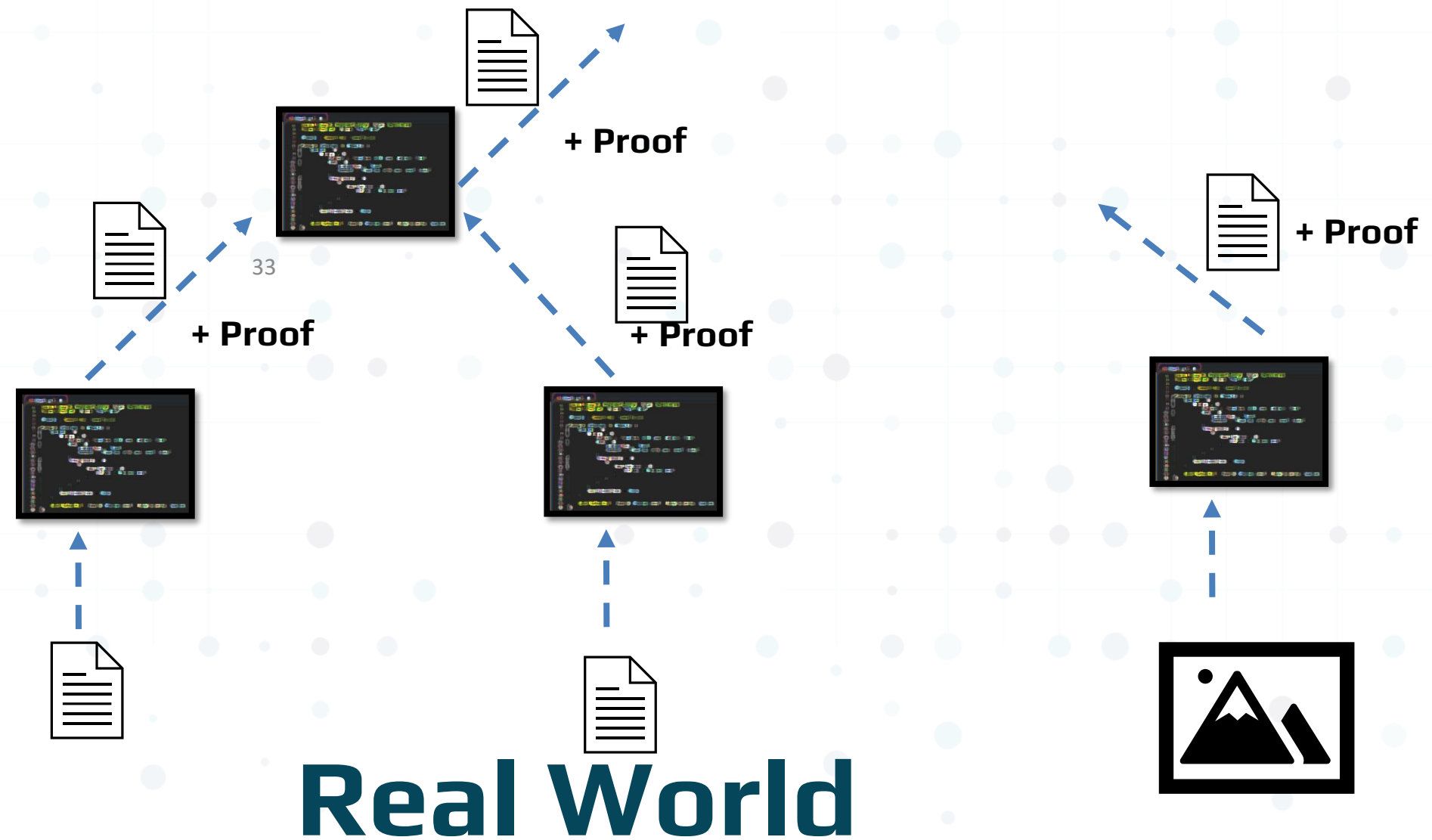


Real World

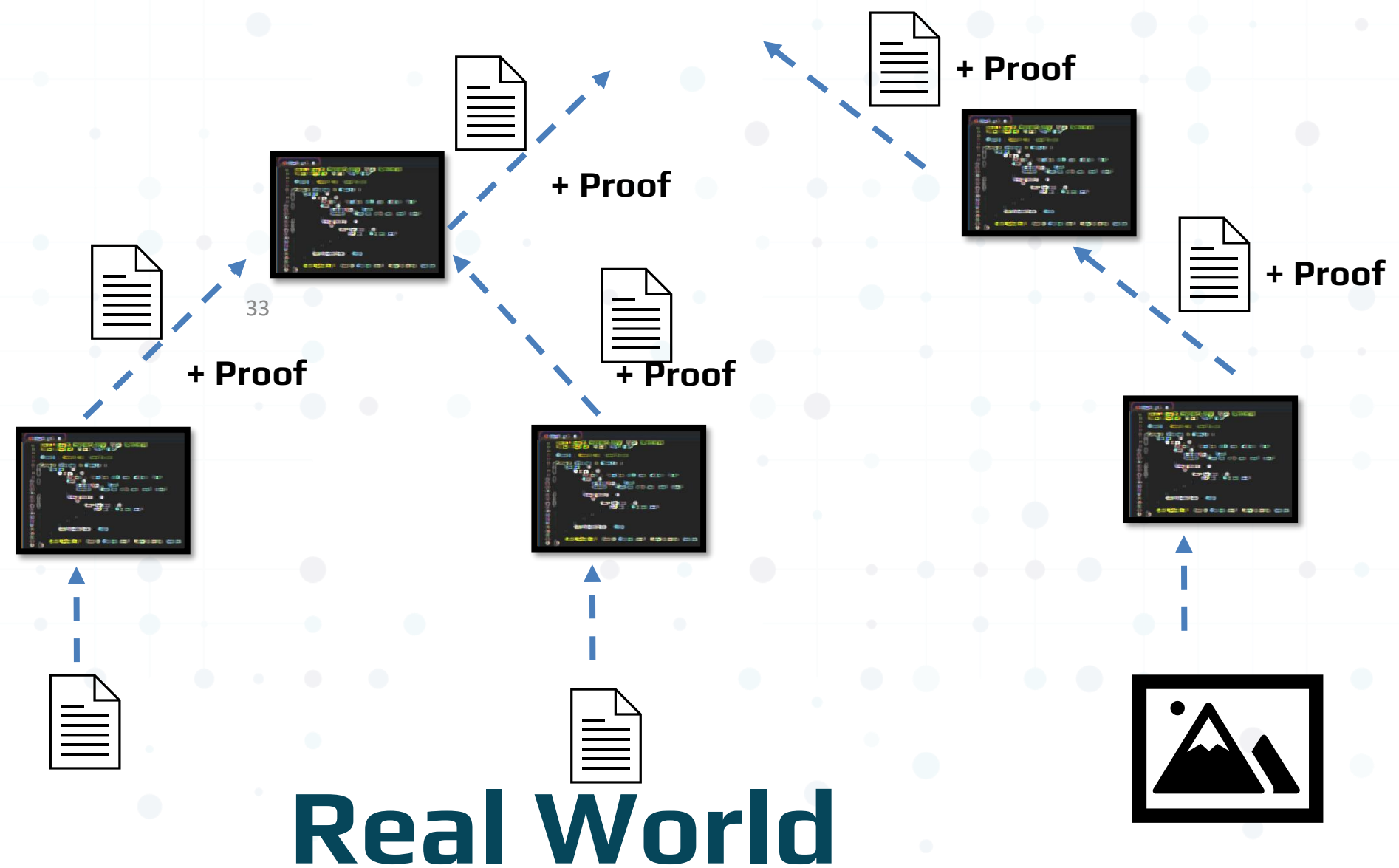
VISION OF FUTURE



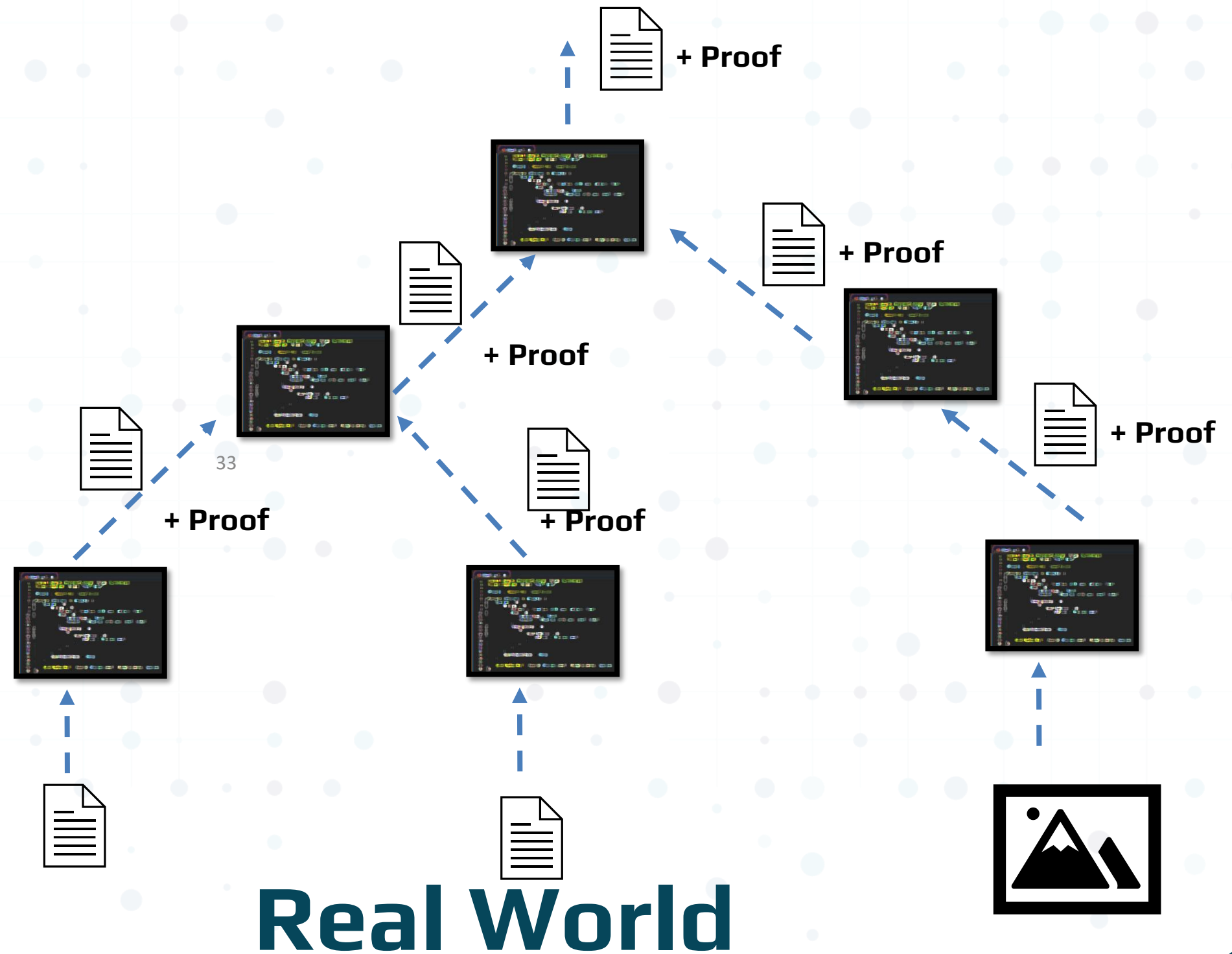
VISION OF FUTURE



VISION OF FUTURE

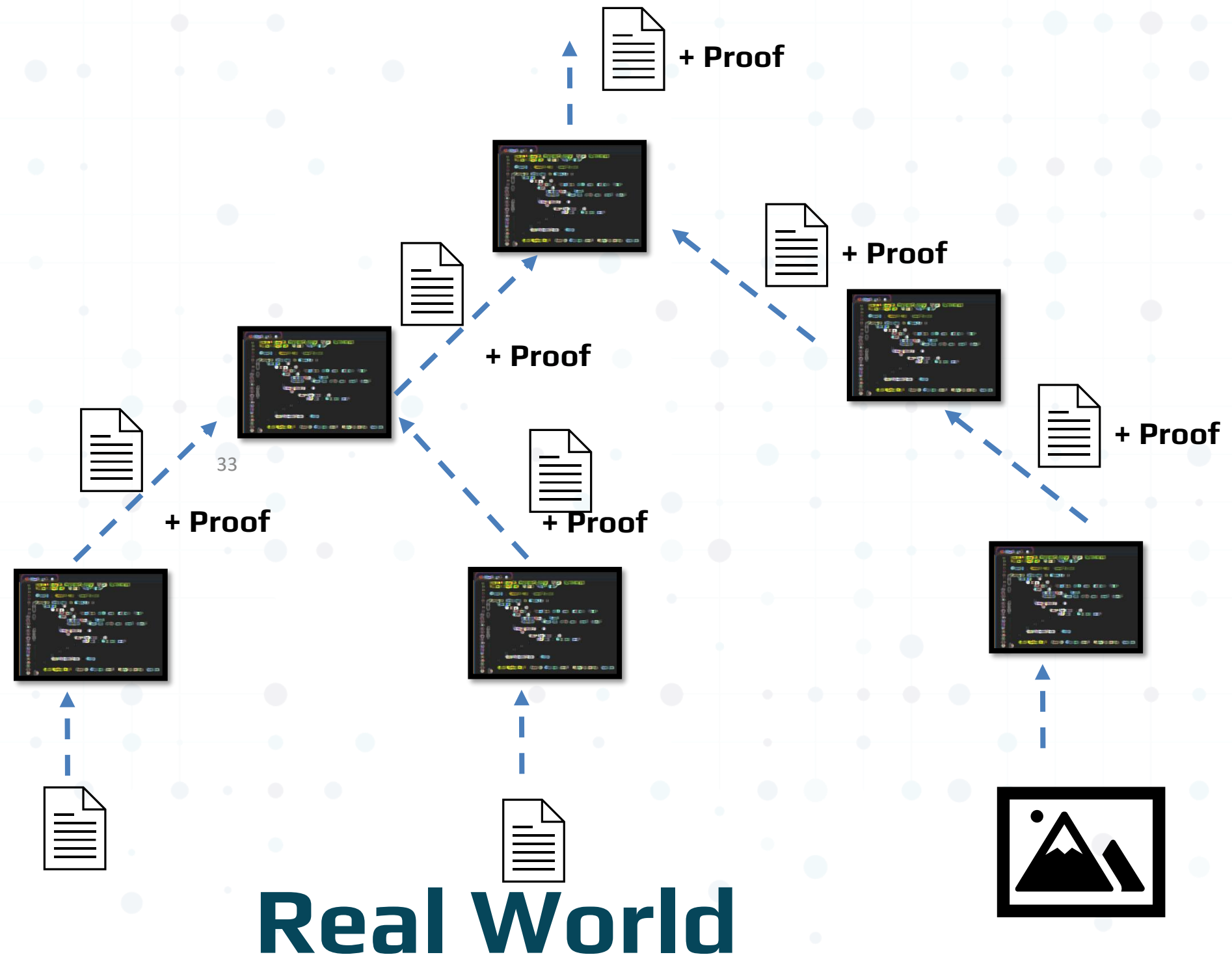


VISION OF FUTURE



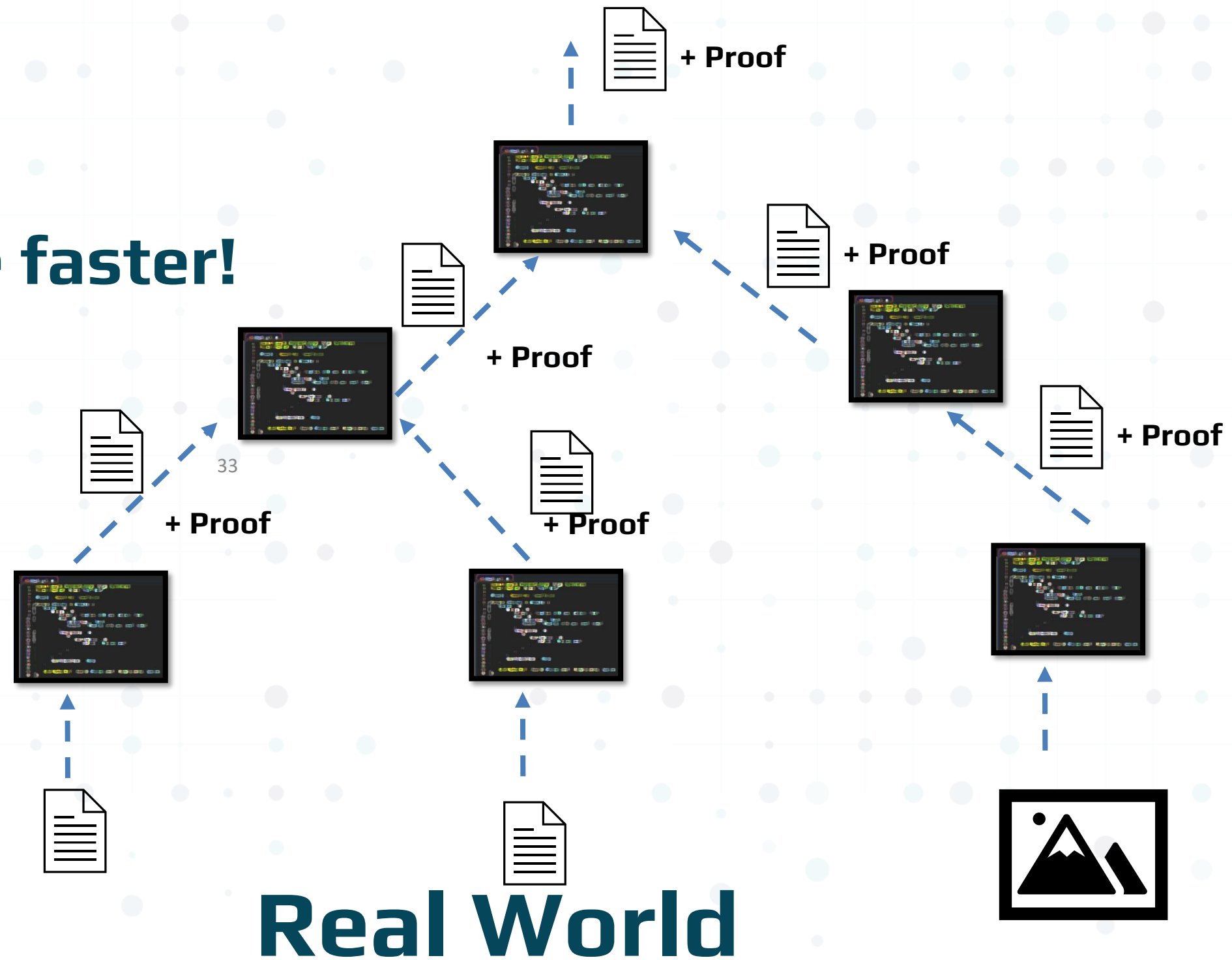
VISION OF FUTURE

- **Proof carrying data**



VISION OF FUTURE

- **Proof carrying data**
- **Proving must become faster!**





**THANK YOU
FOR LISTENING!**

