Blockchain

How Bitcoin works?

Content

A class in data structures:

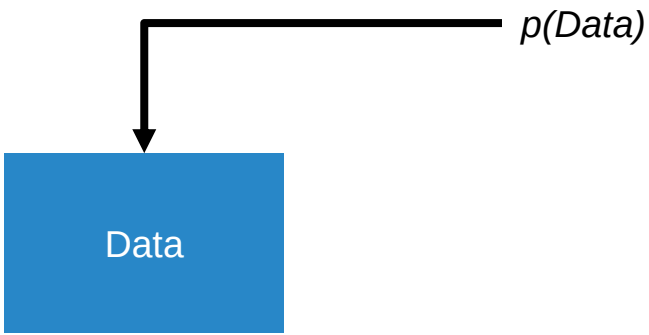
- **Hash pointers**
- Blockchain

A normal pointer

A normal pointer

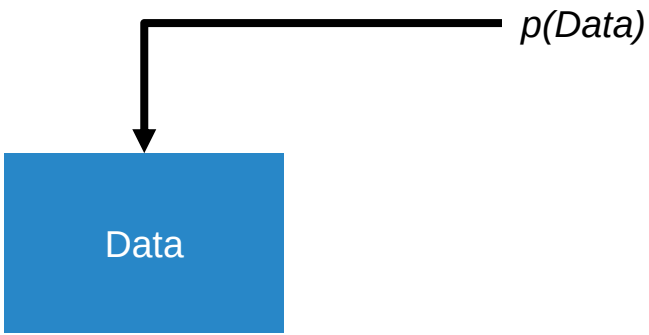
p(Data)

A normal pointer



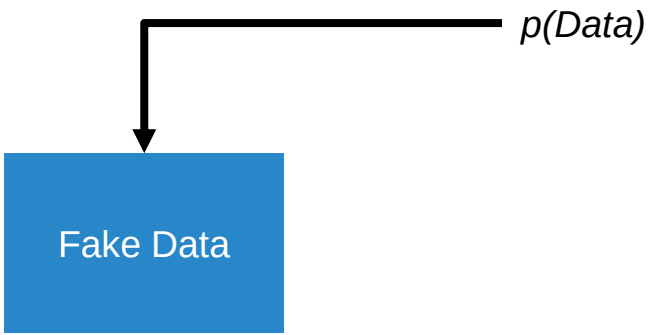
A normal pointer

What happens if the data changes?



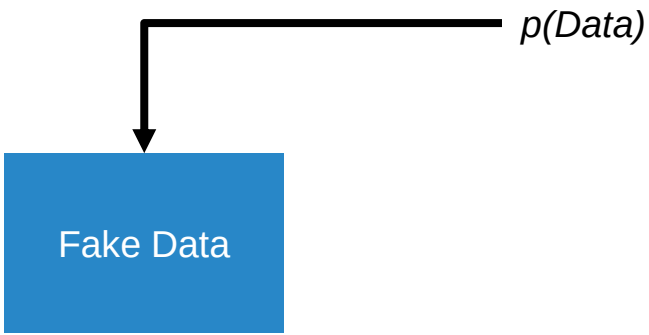
A normal pointer

What happens if the data changes?



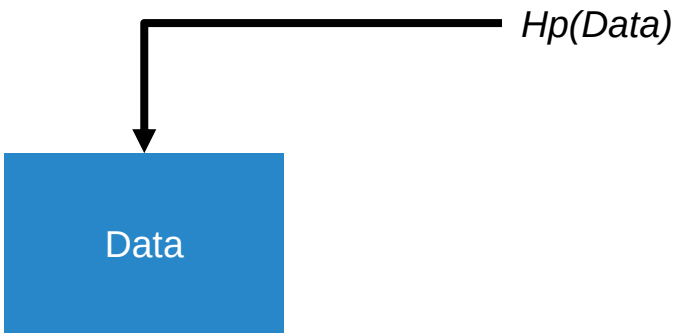
A normal pointer

What happens if the data changes?

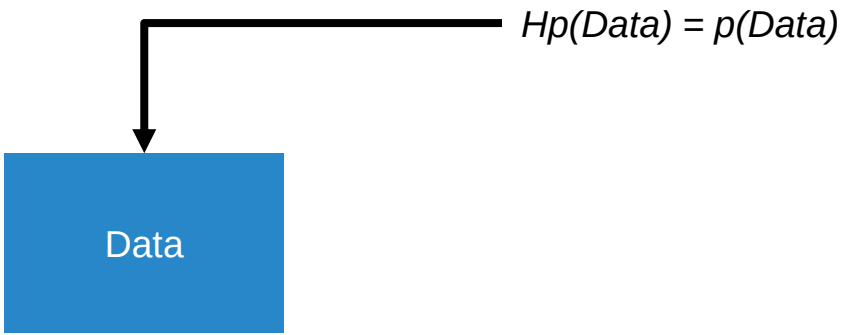


p does not reflect the change!!!

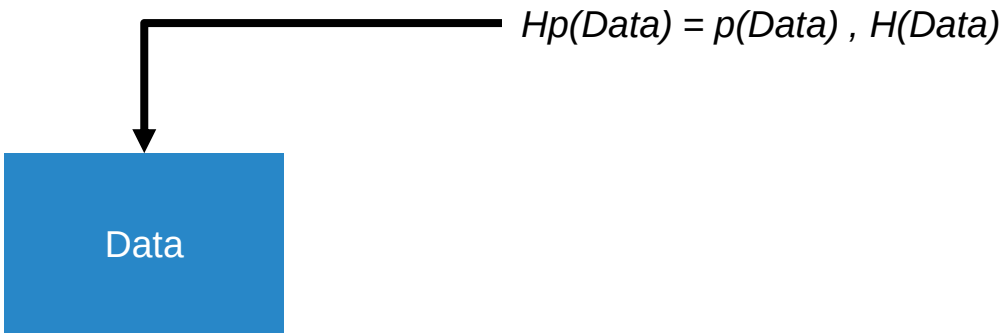
Hash pointer



Hash pointer

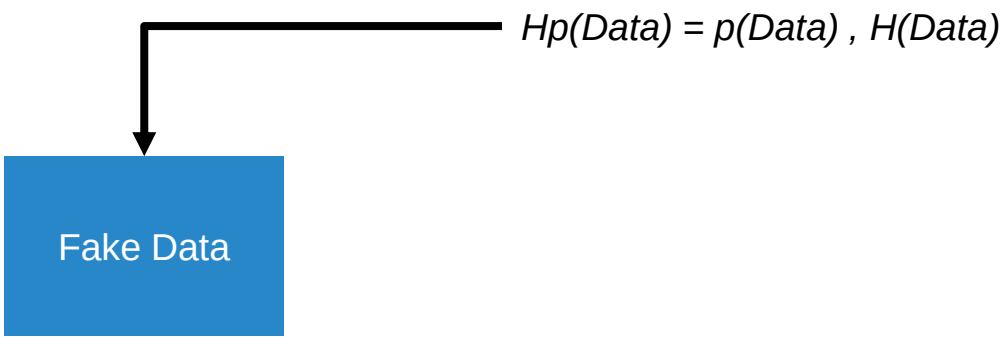


Hash pointer



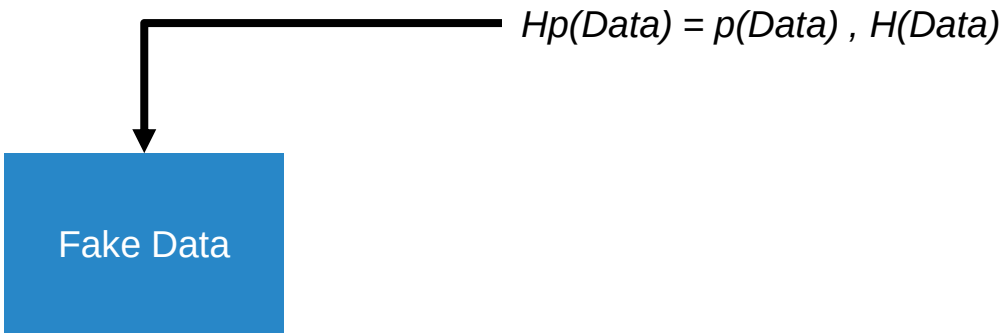
Hash pointer

What happens if the data changes?



Hash pointer

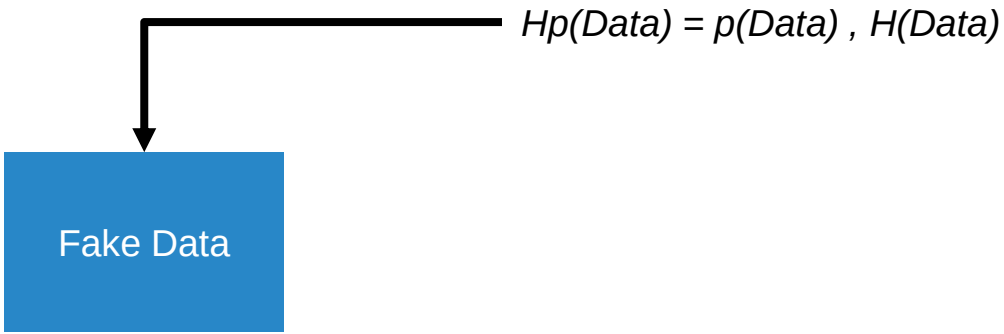
What happens if the data changes?



Hp(Data) does not point to Fake Data

Hash pointer

What happens if the data changes?



Hp(Data) does not point to Fake Data

H(Fake Data) ≠ H(Data)

Hash pointers

Examples of hash pointers:

- If I have a variable
- If my data is in an array
- If my data is in a dictionary (key-value) ← show this

Hash pointers

Use of hash pointers:

- In any data structure that uses pointers
- Linked lists = blockchain
- Binary trees = Merkle Trees

Content

A class in data structures:

- Hash pointers
- **Blockchain**

Blockchain

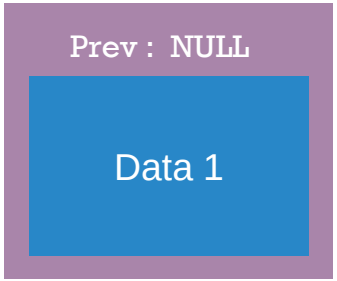
The data structure

Blockchain

The data structure

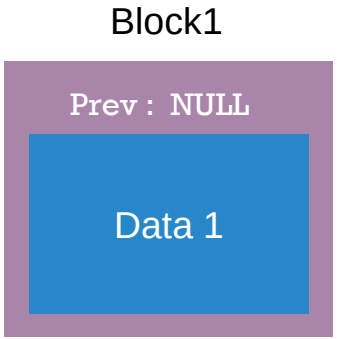
Blockchain

The data structure



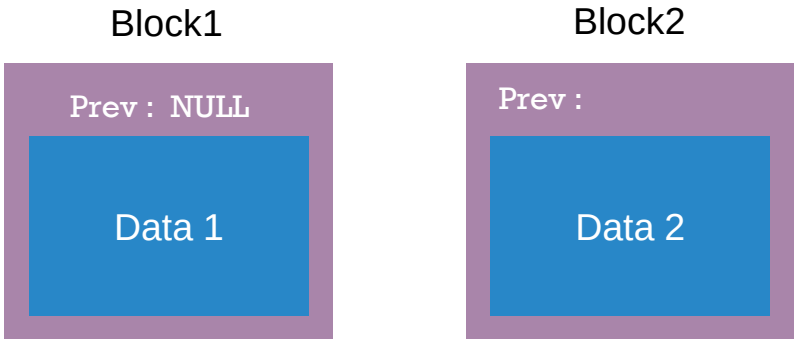
Blockchain

The data structure



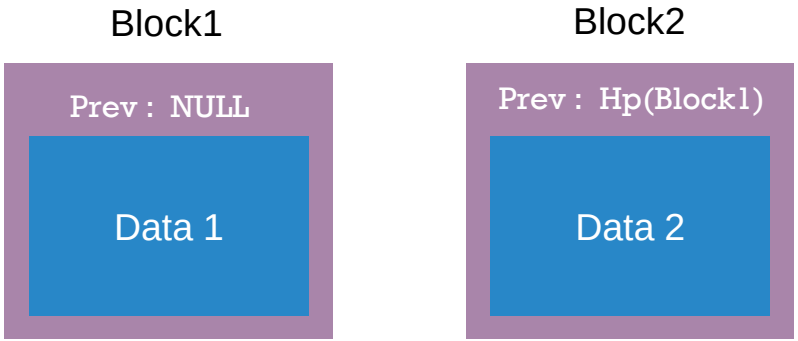
Blockchain

The data structure



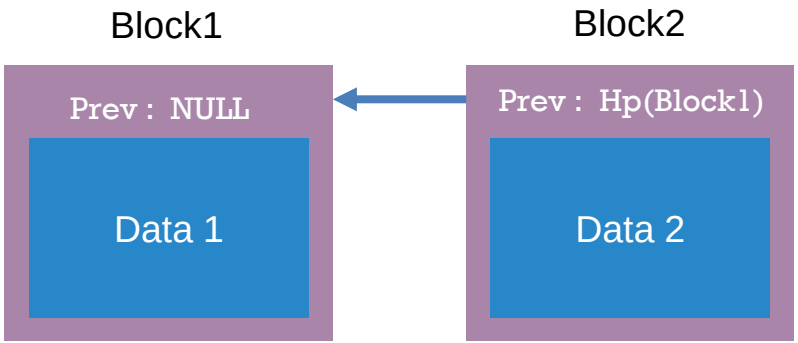
Blockchain

The data structure



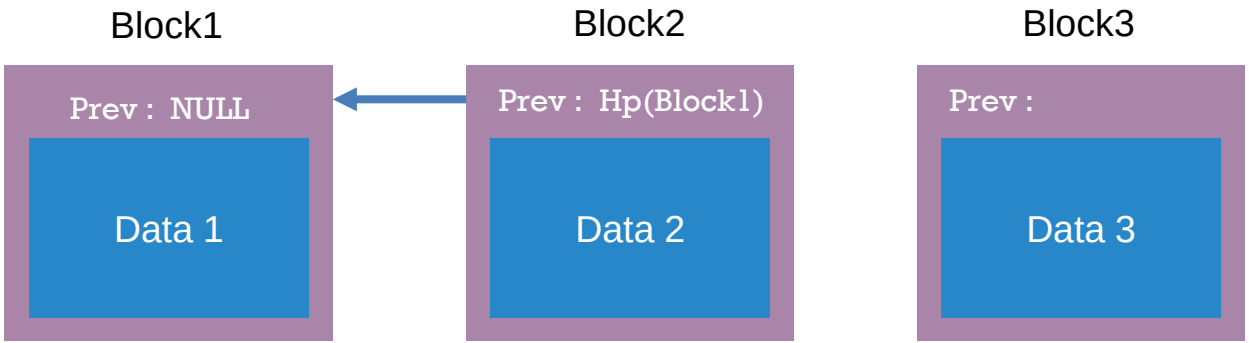
Blockchain

The data structure



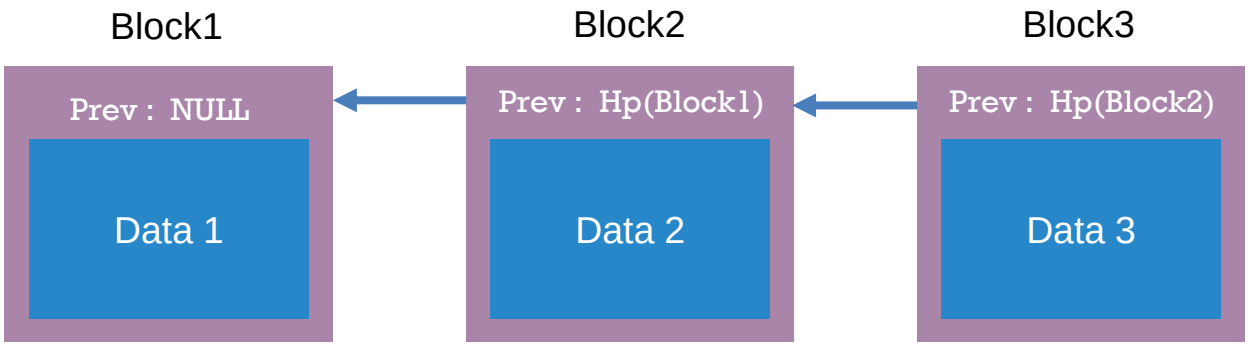
Blockchain

The data structure



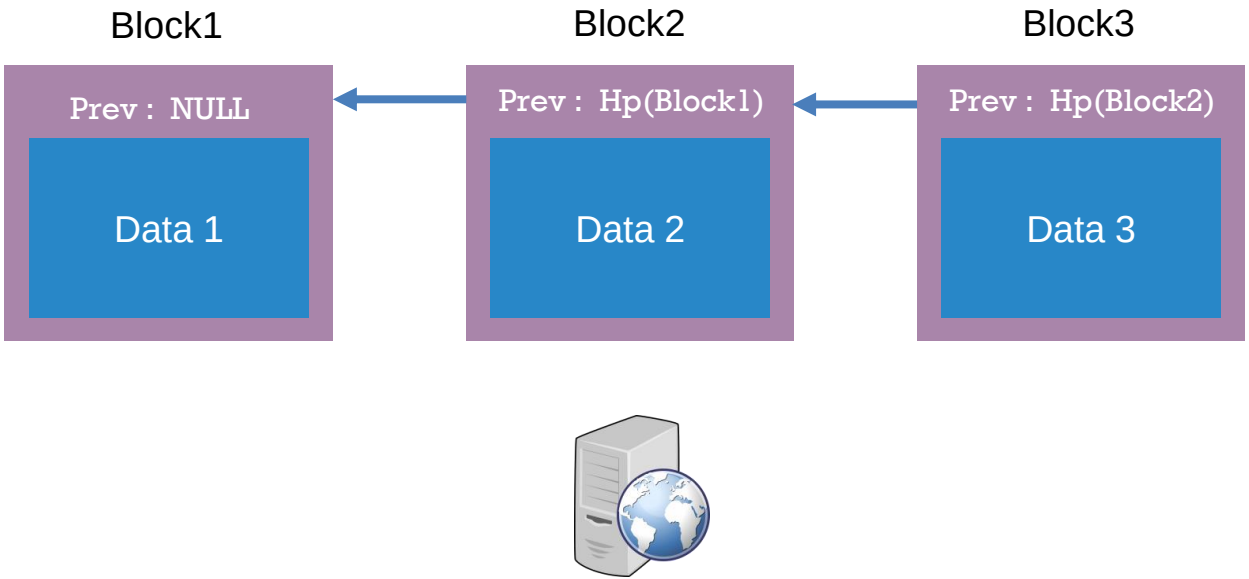
Blockchain

The data structure



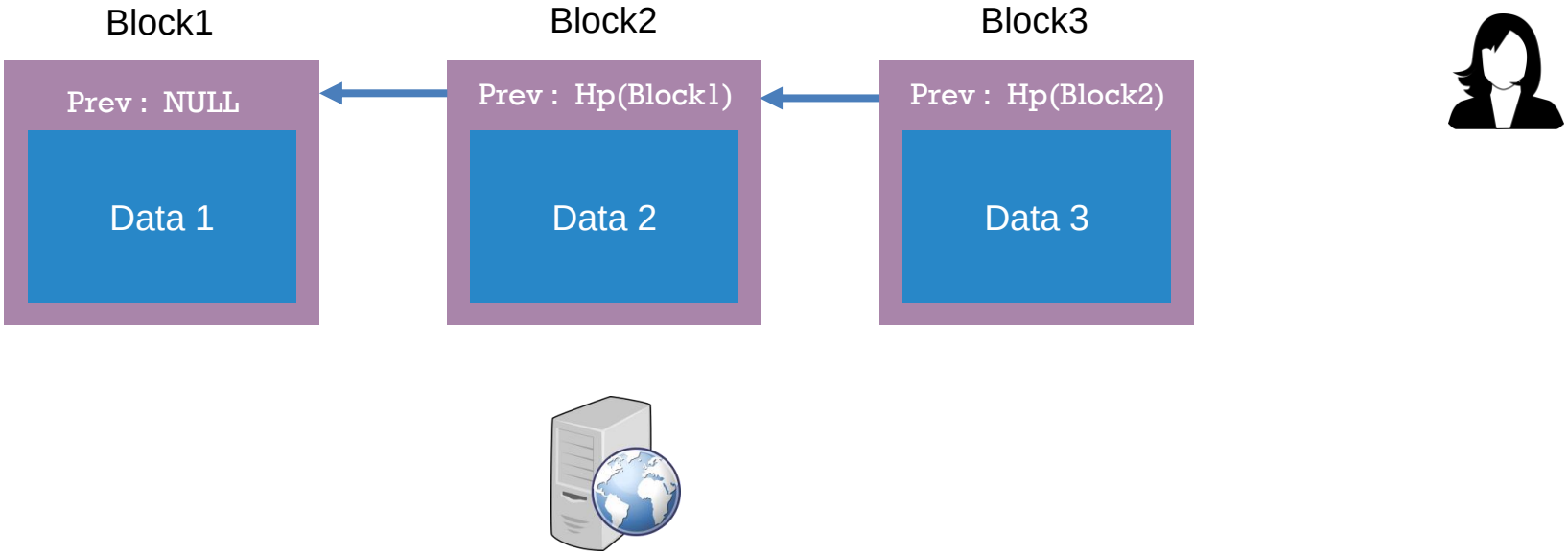
Use of Blockchain

Tamper-evident log



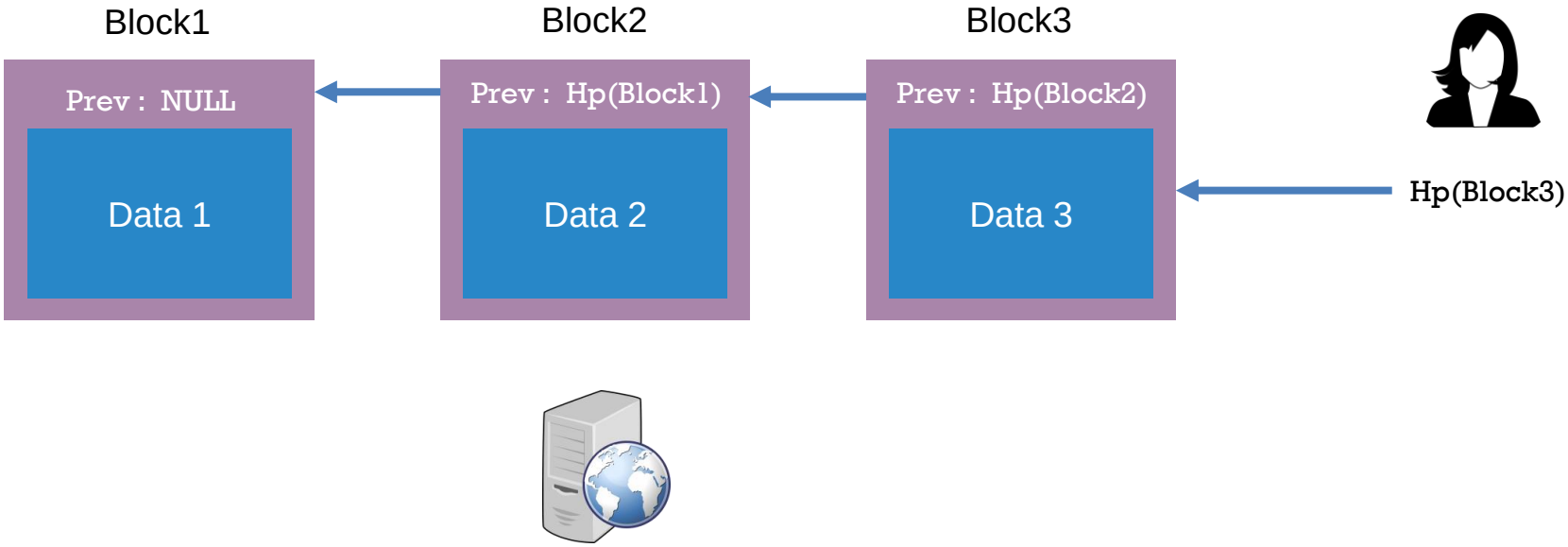
Use of Blockchain

Tamper-evident log



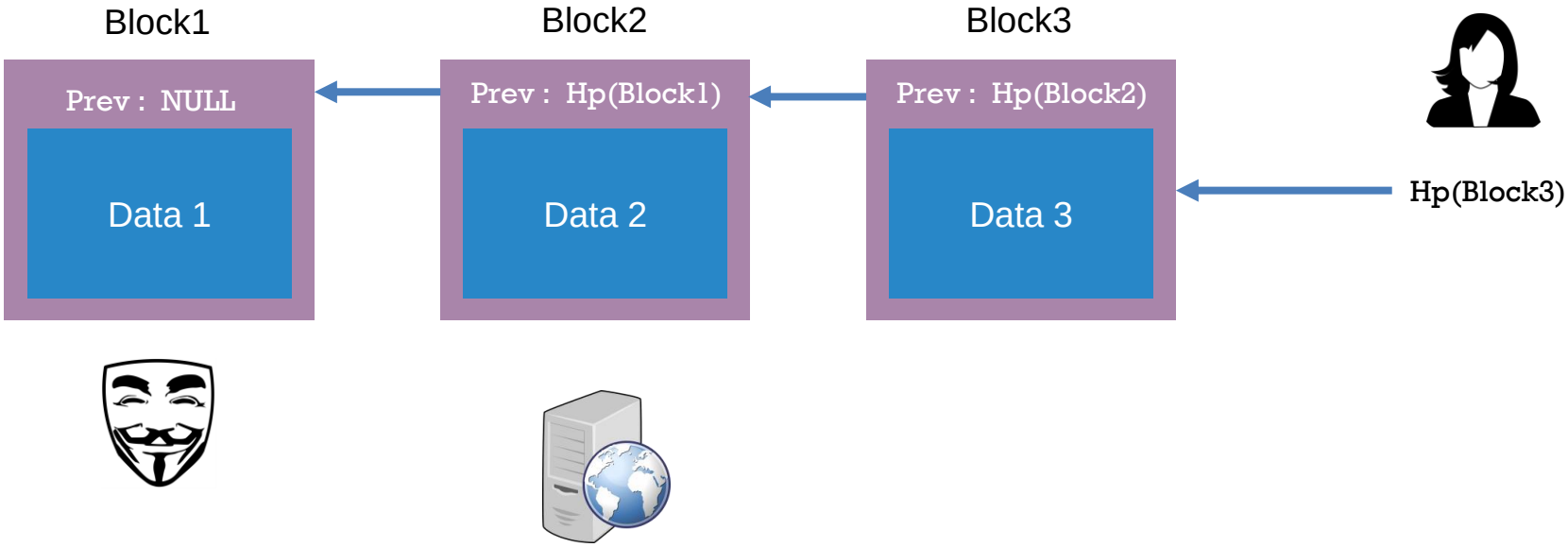
Use of Blockchain

Tamper-evident log



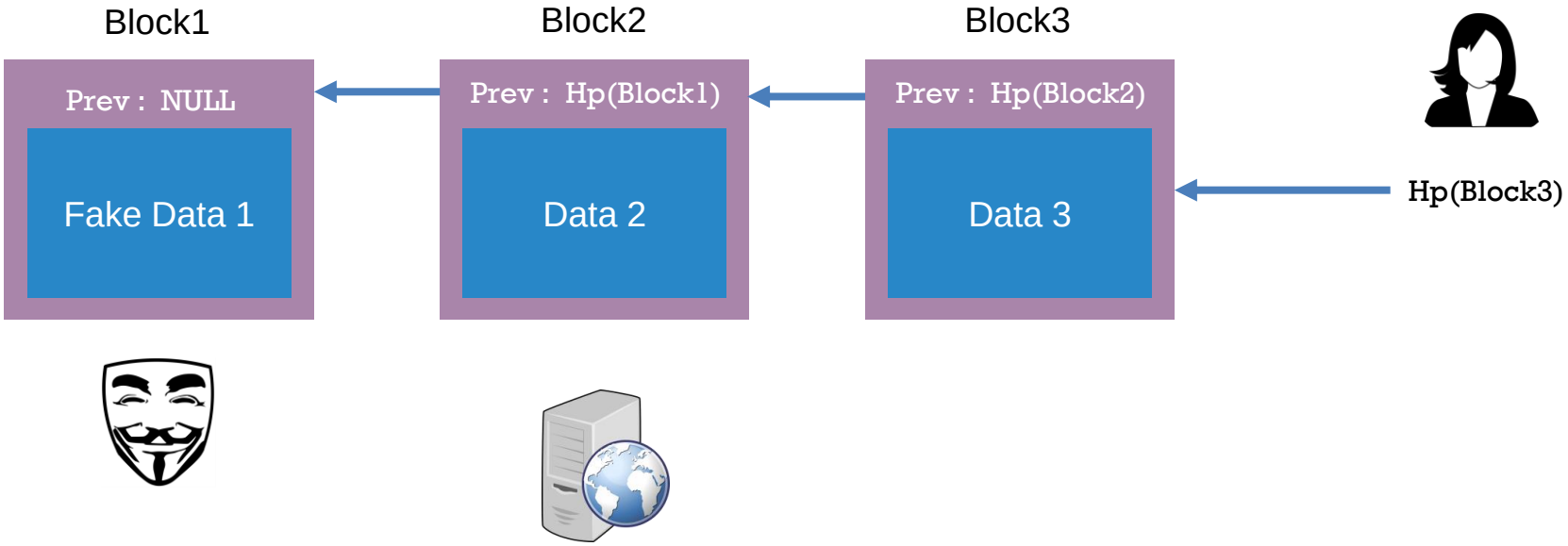
Use of Blockchain

Tamper-evident log



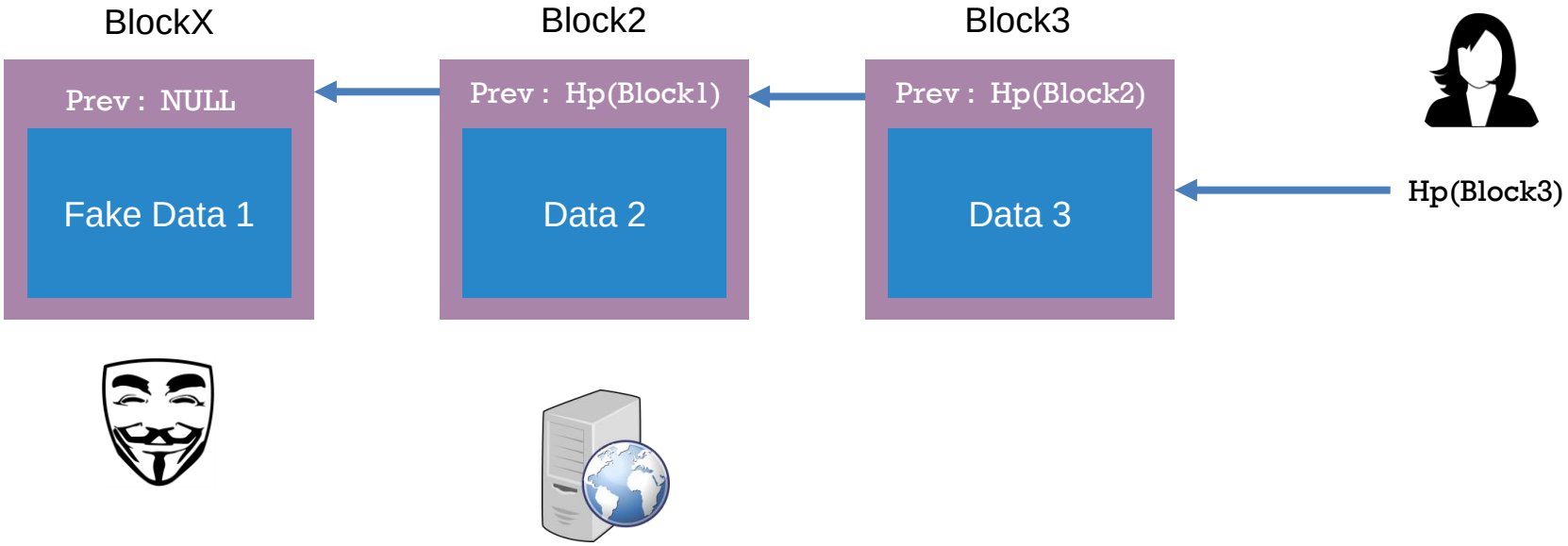
Use of Blockchain

Tamper-evident log



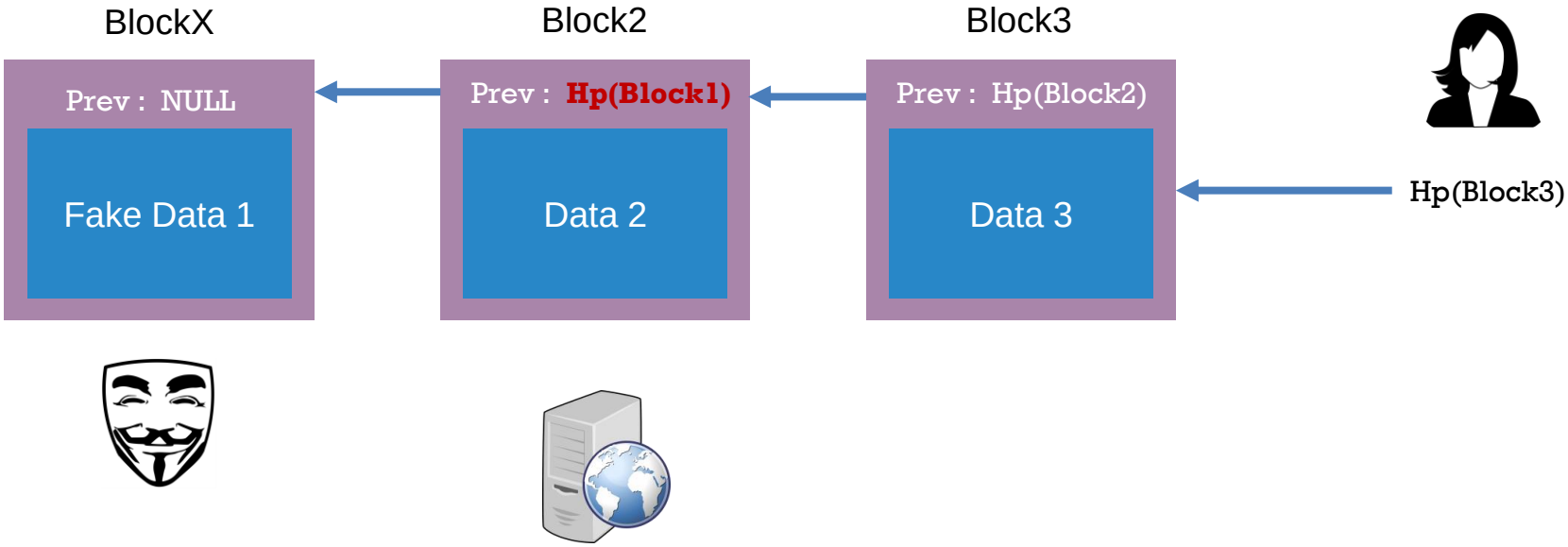
Use of Blockchain

Tamper-evident log



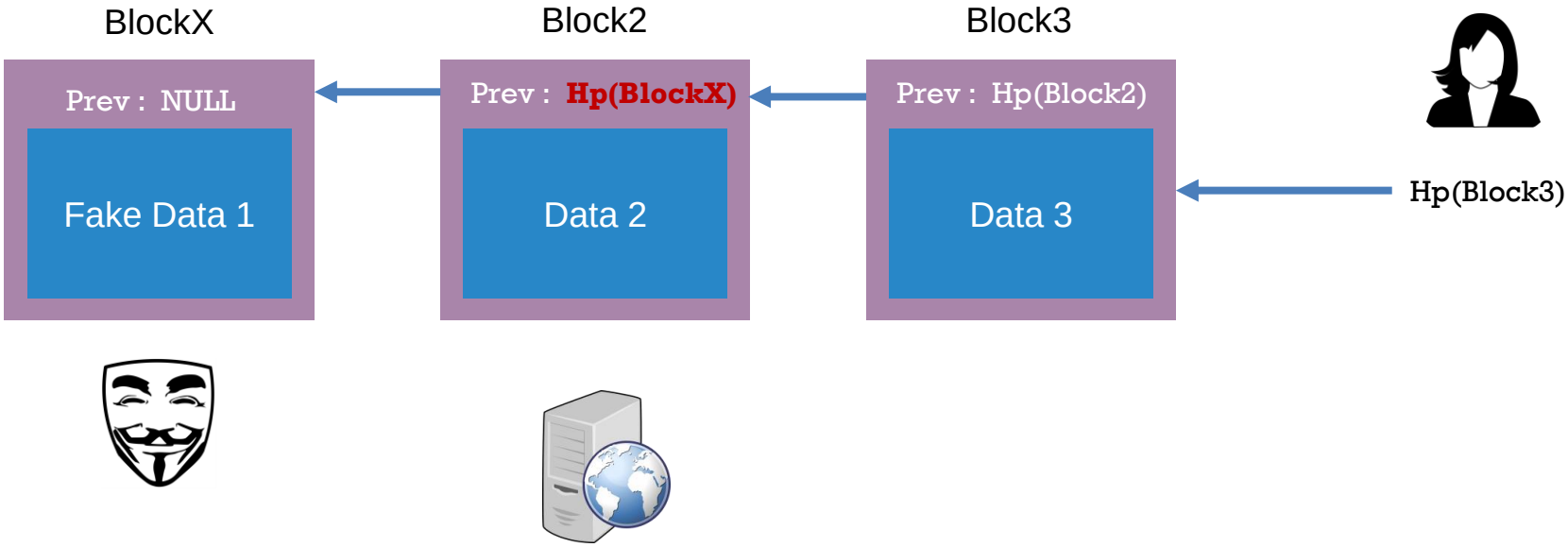
Use of Blockchain

Tamper-evident log



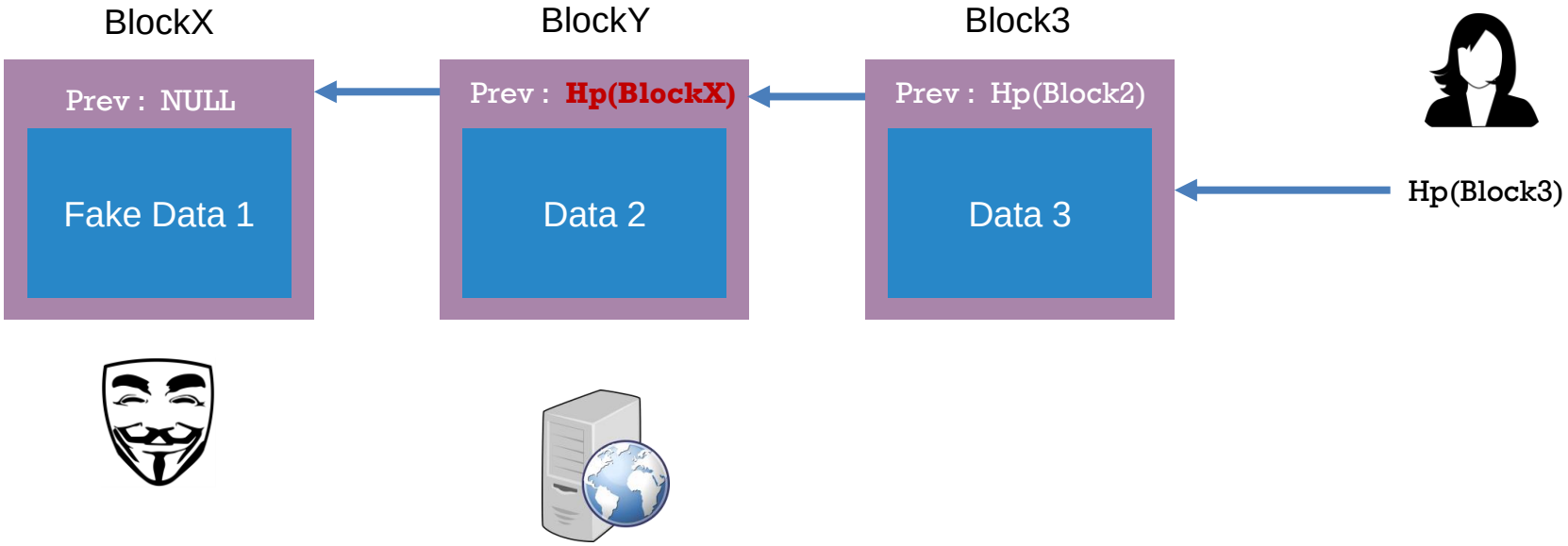
Use of Blockchain

Tamper-evident log



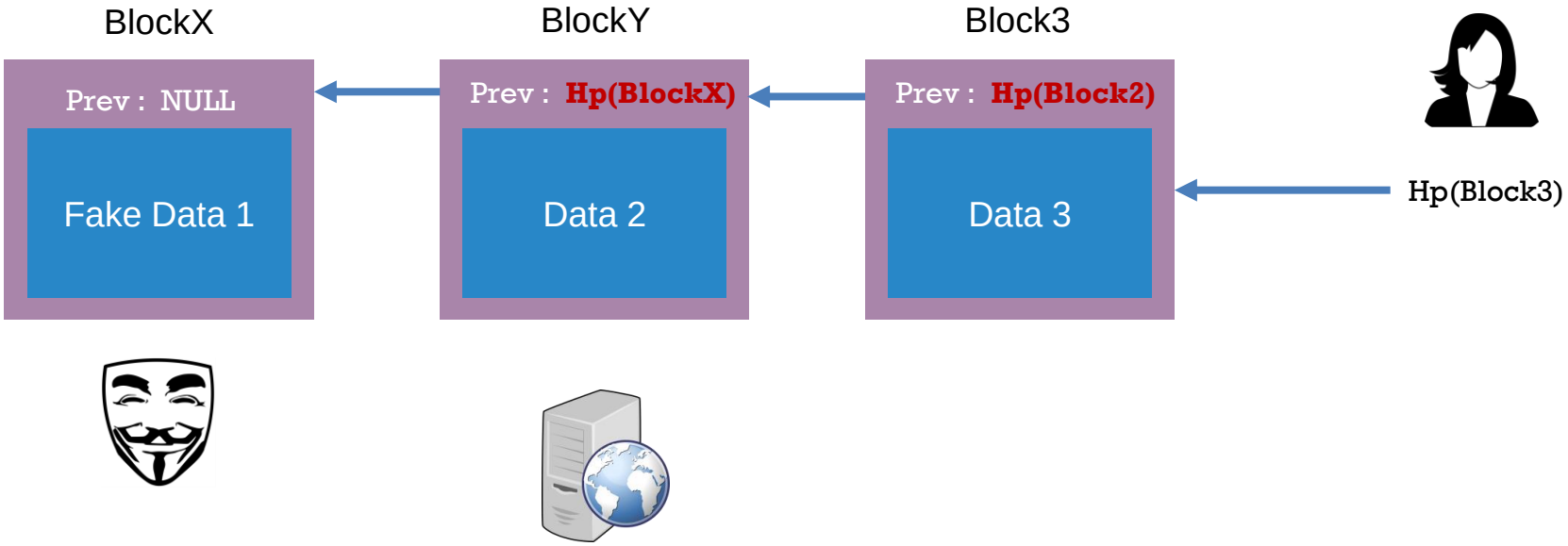
Use of Blockchain

Tamper-evident log



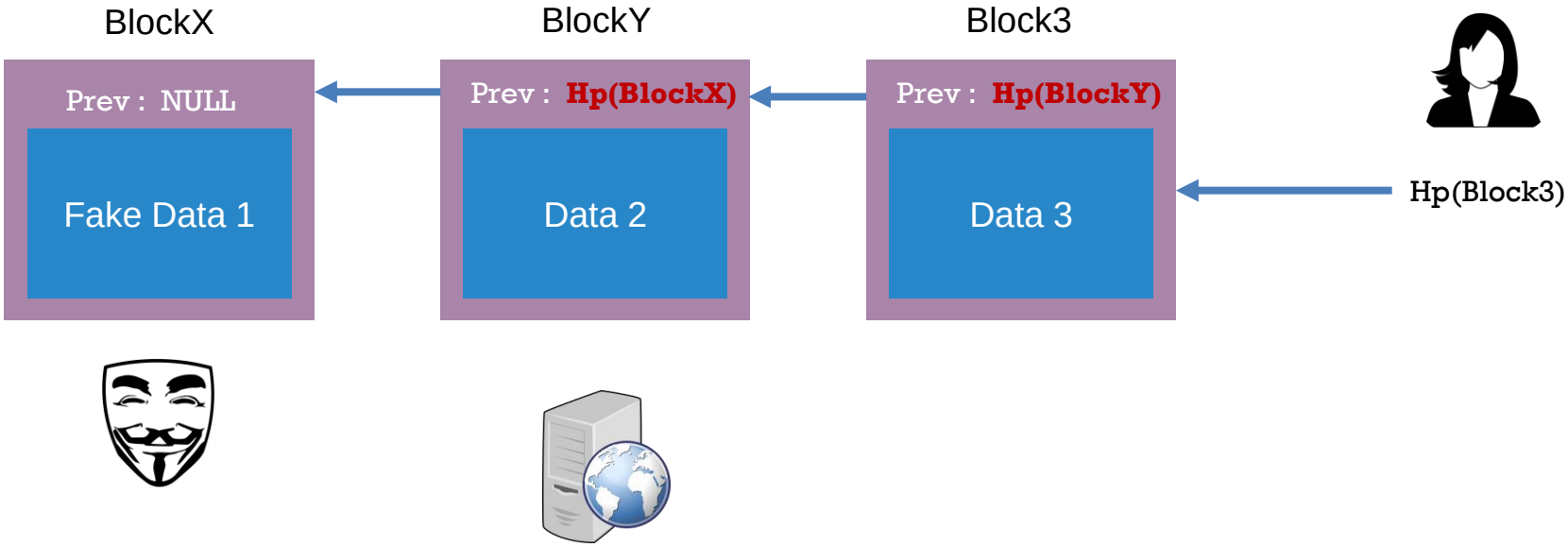
Use of Blockchain

Tamper-evident log



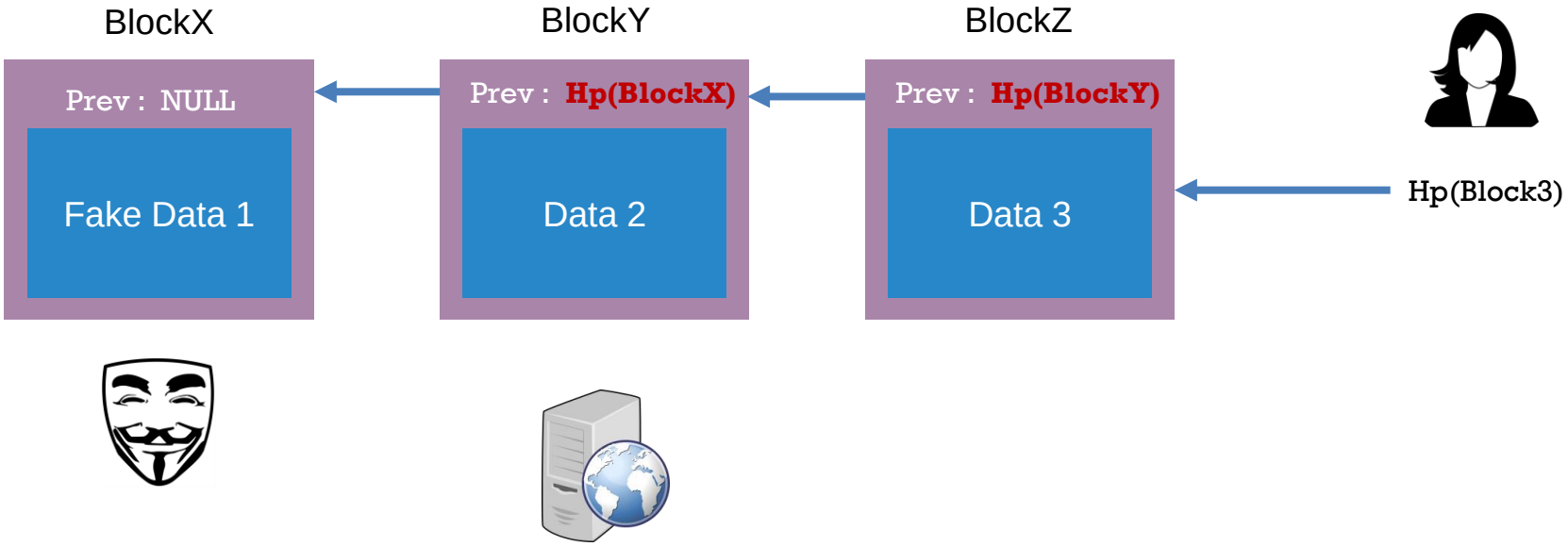
Use of Blockchain

Tamper-evident log



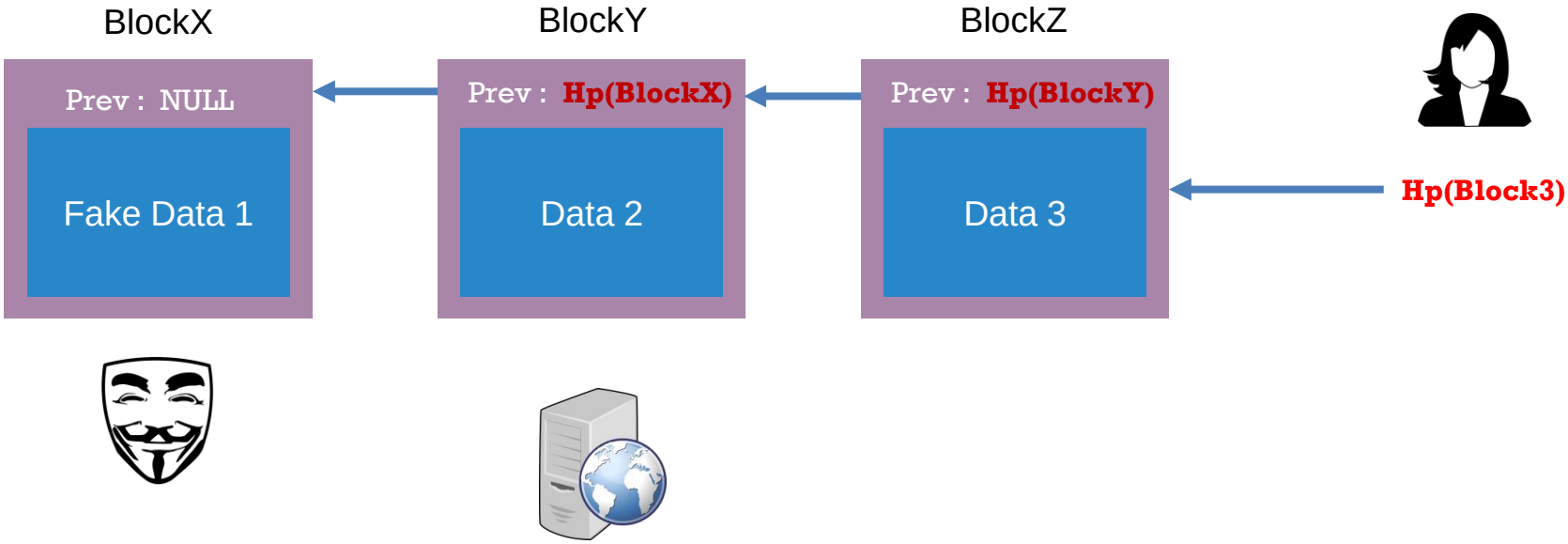
Use of Blockchain

Tamper-evident log



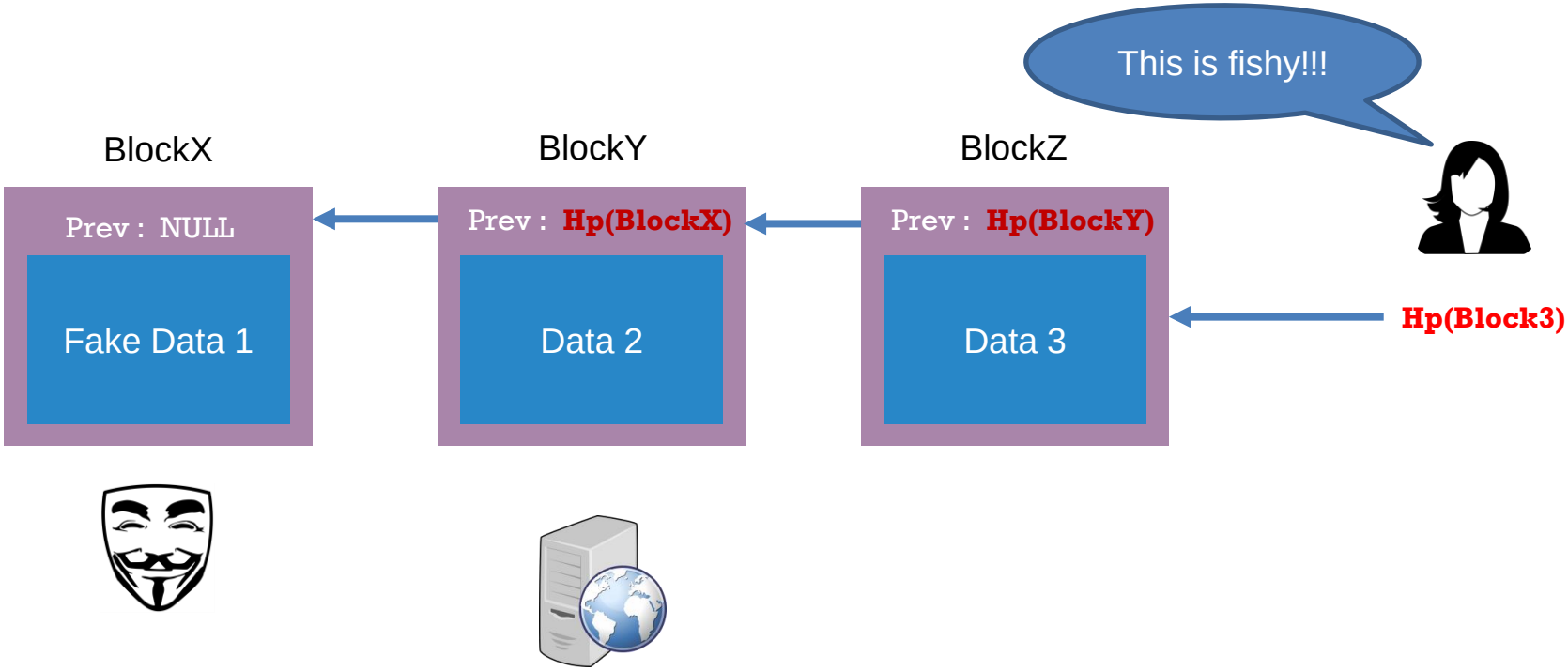
Use of Blockchain

Tamper-evident log



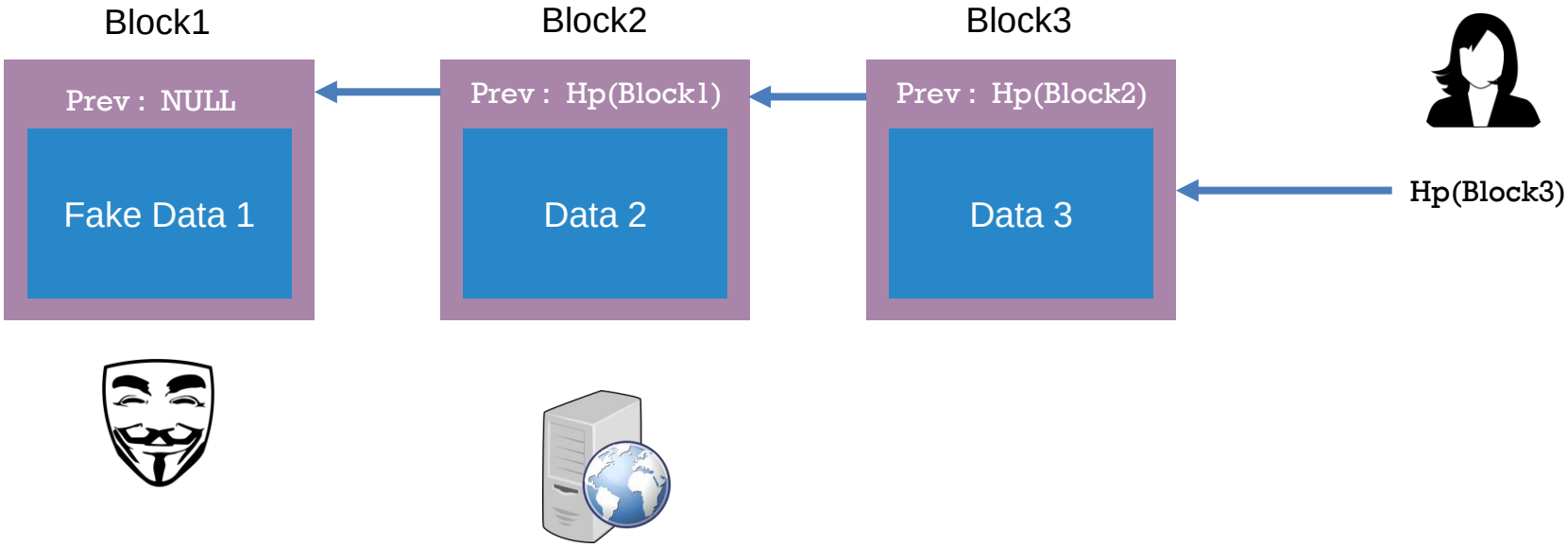
Use of Blockchain

Tamper-evident log



Use of Blockchain

What is this is the only change?



Problem 3 for e-Kuna

Consistent historic data

Ledger
Alice pays Bob $50
Alice pays Charlie $20

Alice

Bob

Charlie

Problem 3 for e-Kuna

Consistent historic data

Ledger
Alice pays Bob $50
Alice pays Charlie $20
Bob pays Charlie $100
Charlie pays Alice $40
. . .

Alice

Bob

Charlie

Problem 3 for e-Kuna

Consistent historic data

Ledger 1
Alice pays Bob $50
Alice pays Charlie $20
Bob pays Charlie $100
Charlie pays Alice $40
. . .

Ledger 2

Alice

Bob

Charlie

Problem 3 for e-Kuna

Consistent historic data

Ledger 1
Alice pays Bob $50
Alice pays Charlie $20
Bob pays Charlie $100
Charlie pays Alice $40
. . .

Ledger 2
Bob pays Charlie $250

Alice

Bob

Charlie

Problem 3 for e-Kuna

Consistent historic data

Ledger 1
Alice pays Bob $50
Alice pays Charlie $20
Bob pays Charlie $100
Charlie pays Alice $40
. . .

Ledger 2
Bob pays Charlie $250
Bob pays Alice $120

Alice

Bob

Charlie

Problem 3 for e-Kuna

Consistent historic data

Ledger 1
Alice pays Bob $50
Alice pays Charlie $20
Bob pays Charlie $100
Charlie pays Alice $40
. . .

Ledger 2
Bob pays Charlie $250
Bob pays Alice $120
Charlie pays Alice $80
Alice pays Bob $20
Alice pays Charlie $10

Alice Bob Charlie

Problem 3 for e-Kuna

Consistent historic data

Ledger 1
Alice pays Bob $50
Alice pays Charlie $20
Bob pays Charlie $100
Charlie pays Alice $40
. . .

Ledger 2
Bob pays Charlie $250
Bob pays Alice $120
Charlie pays Alice $80
Alice pays Bob $20
Alice pays Charlie $10

Ledger 3

Alice

Bob

Charlie

Problem 3 for e-Kuna

Consistent historic data

Ledger 1
Alice pays Bob $50
Alice pays Charlie $20
Bob pays Charlie $100
Charlie pays Alice $40
. . .

Ledger 2
Bob pays Charlie $250
Bob pays Alice $120
Charlie pays Alice $80
Alice pays Bob $20
Alice pays Charlie $10

Ledger 3
Alice pays Bob $20
Alice pays Charlie $10
Bob pays Charlie $100
Charlie pays Alice $40

Alice

Bob

Charlie

Problem 3 for e-Kuna

Consistent historic data

Ledger 1	Ledger 2	Ledger 3
Alice pays Bob $50	Bob pays Charlie $250	Alice pays Bob $20
Alice pays Charlie $20	Bob pays Alice $120	Alice pays Charlie $10
Bob pays Charlie $100	Charlie pays Alice $80	Bob pays Charlie $100
Charlie pays Alice $40	Alice pays Bob $20	Charlie pays Alice $40
. . .	Alice pays Charlie $10	

Alice

Bob

Charlie

Problem 3 for e-Kuna

Consistent historic data

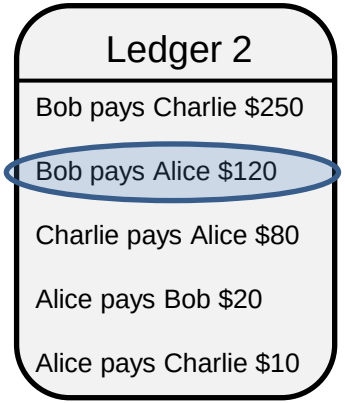
Ledger 1	Ledger 2	Ledger 3
Alice pays Bob $50	Bob pays Charlie $250	Alice pays Bob $20
Alice pays Charlie $20	Bob pays Alice $20	Alice pays Charlie $10
Bob pays Charlie $100	Charlie pays Alice $80	Bob pays Charlie $100
Charlie pays Alice $40	Alice pays Bob $20	Charlie pays Alice $40
. . .	Alice pays Charlie $10	

Alice

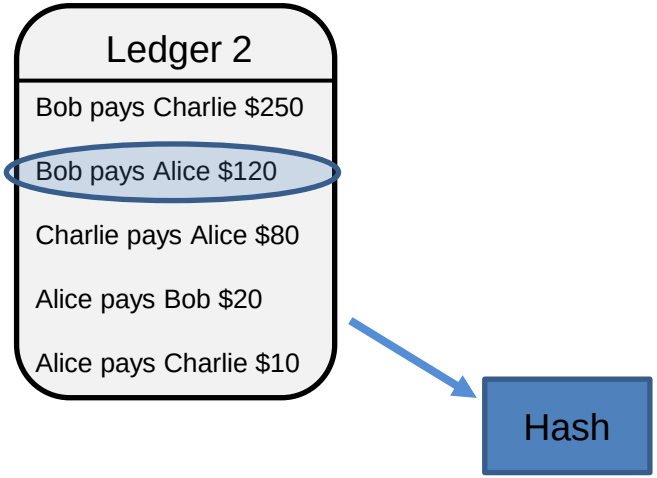
Bob

Charlie

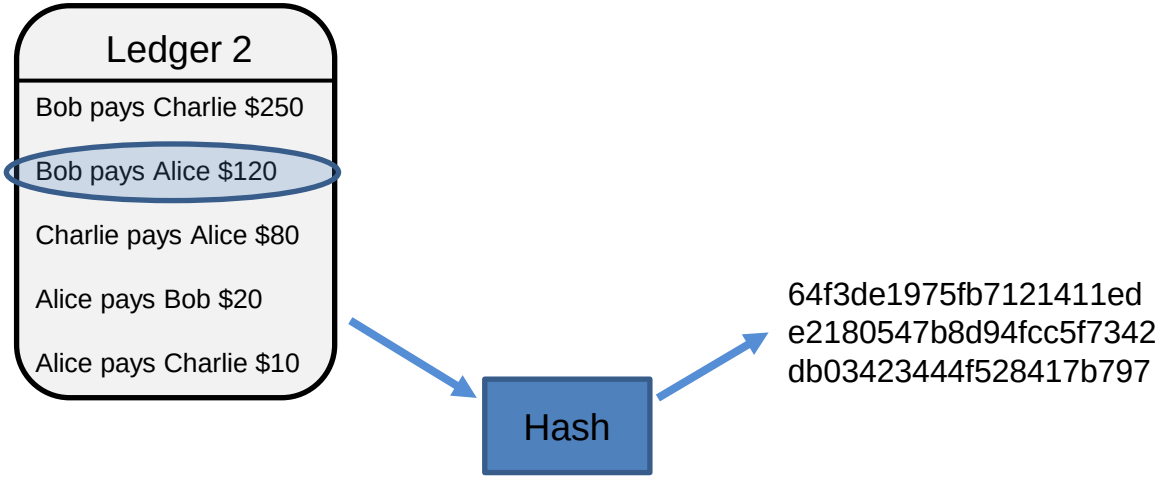
Property of the hash function



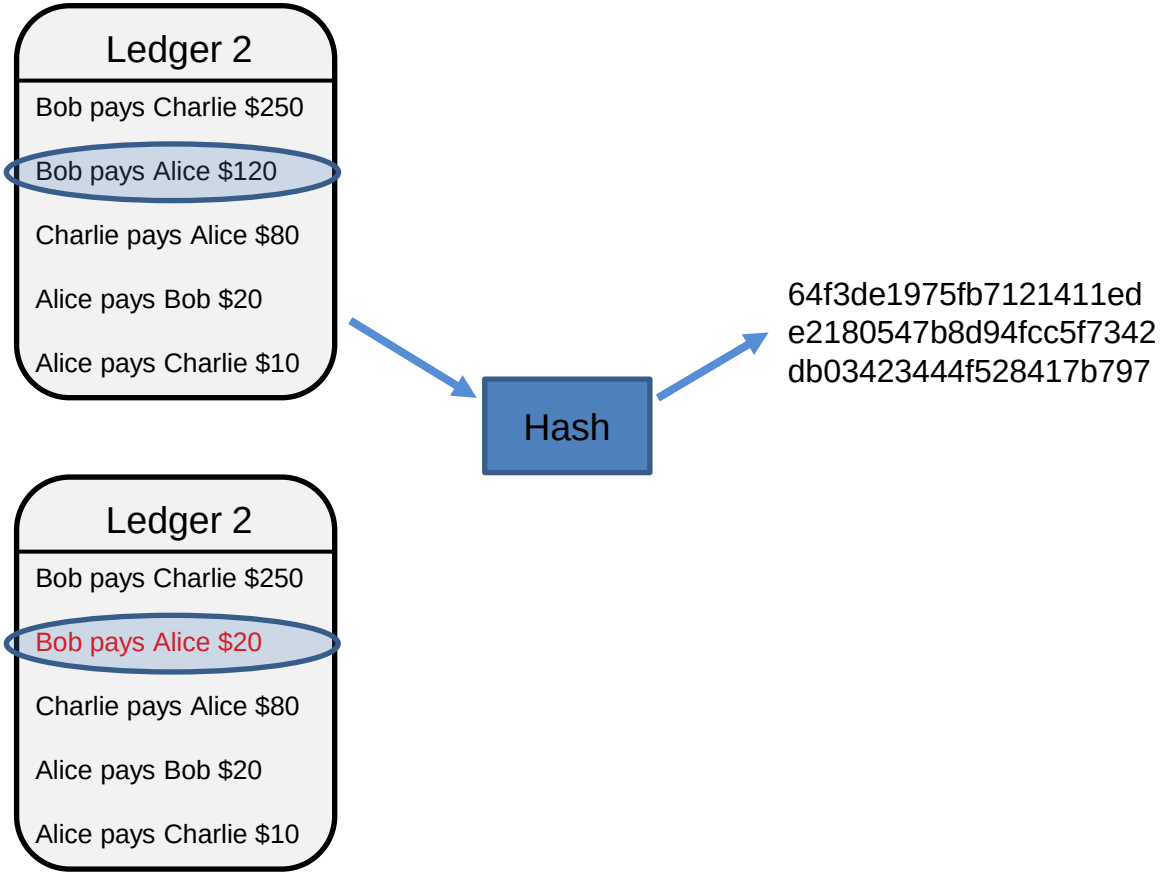
Property of the hash function



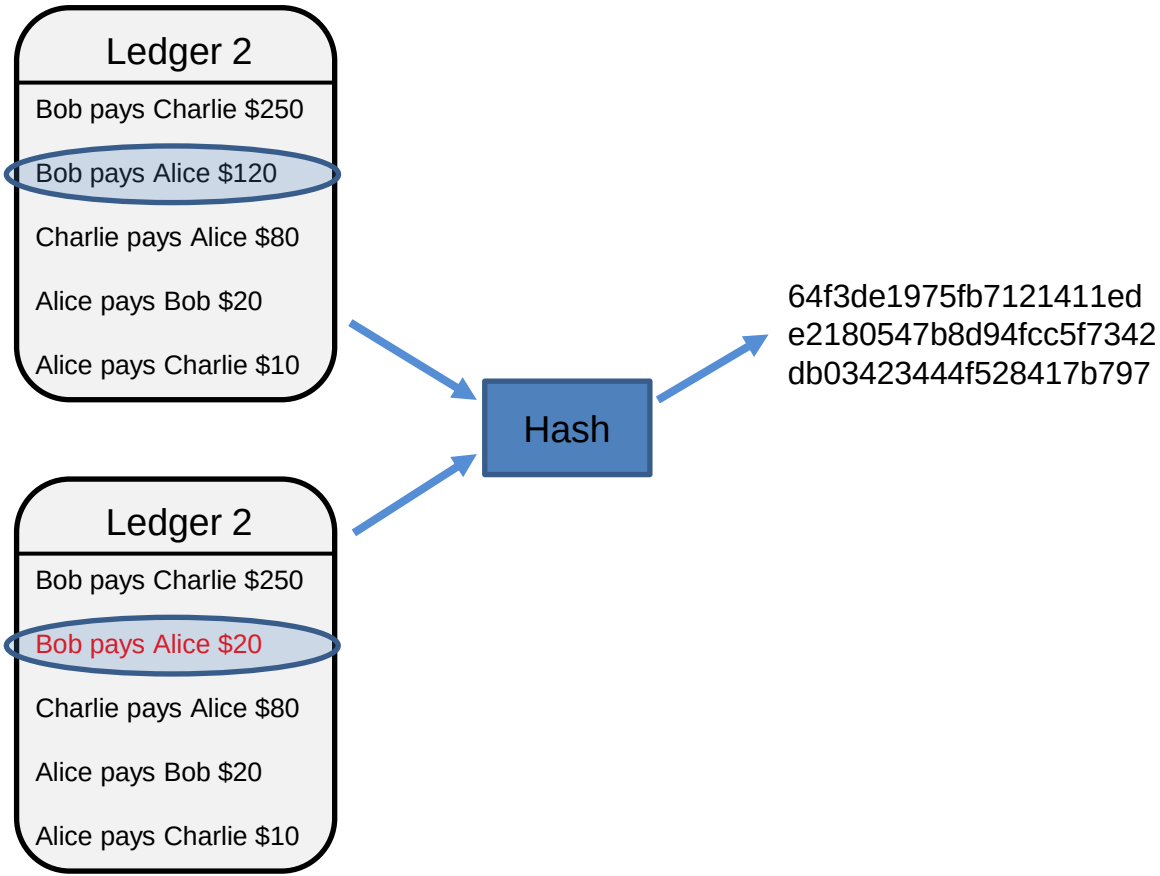
Property of the hash function



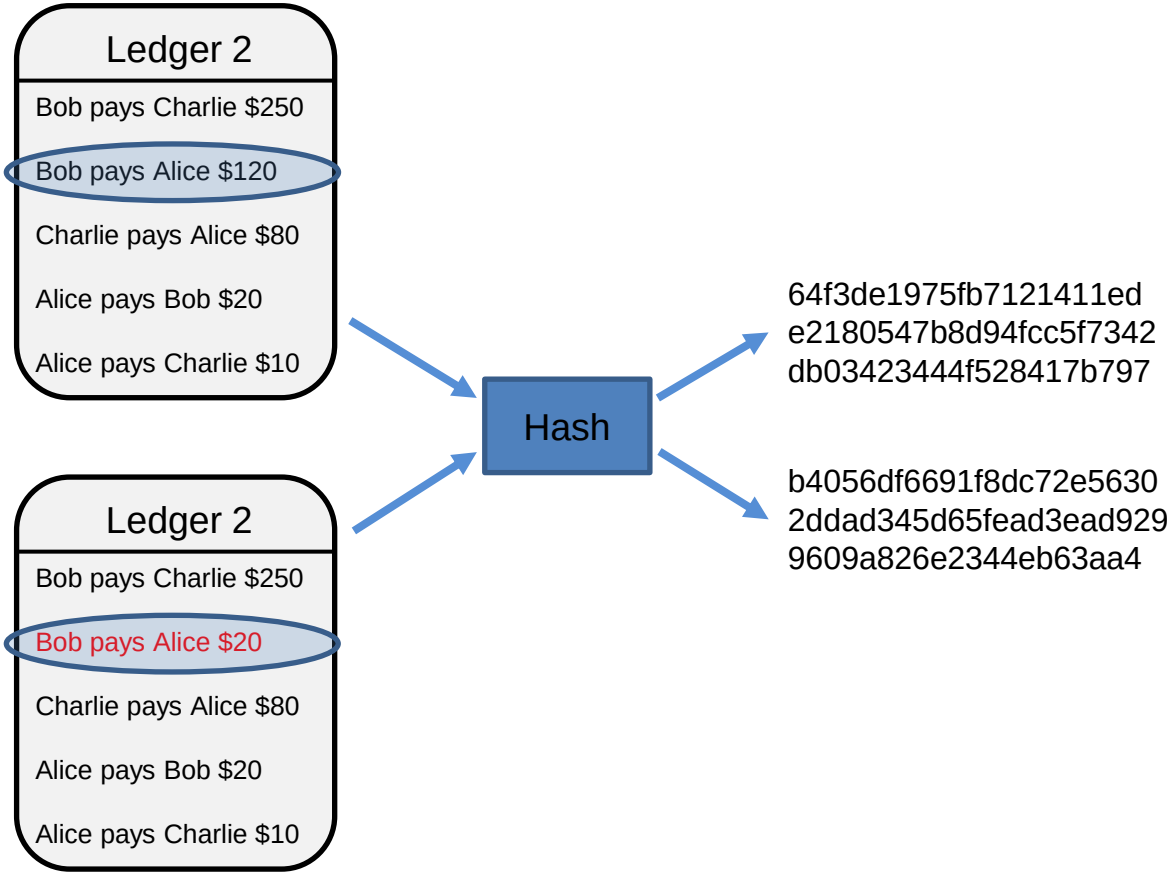
Property of the hash function



Property of the hash function

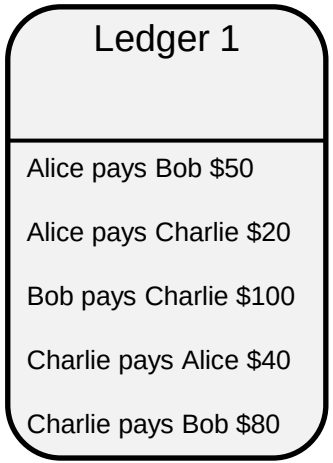


Property of the hash function



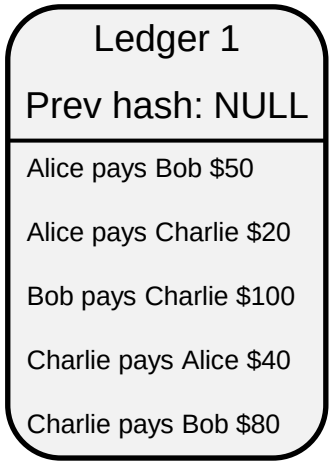
Blockchain

Consistent historic data



Blockchain

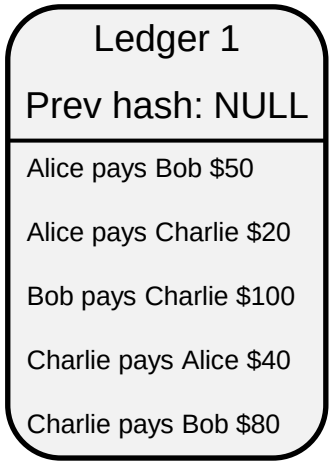
Consistent historic data



Blockchain

Consistent historic data

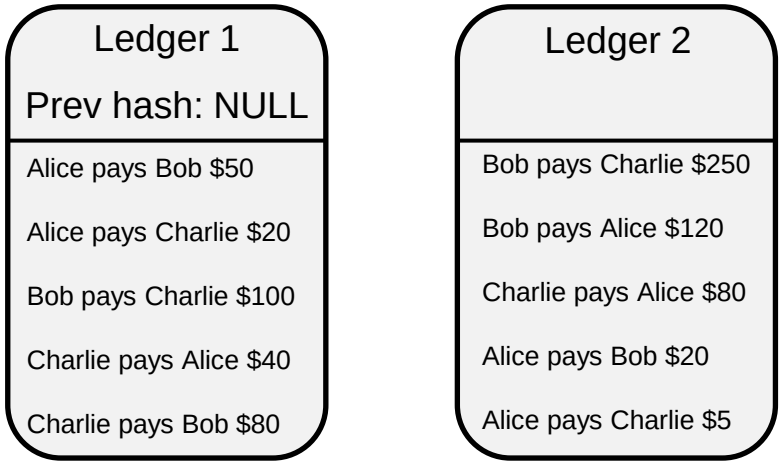
L1



Blockchain

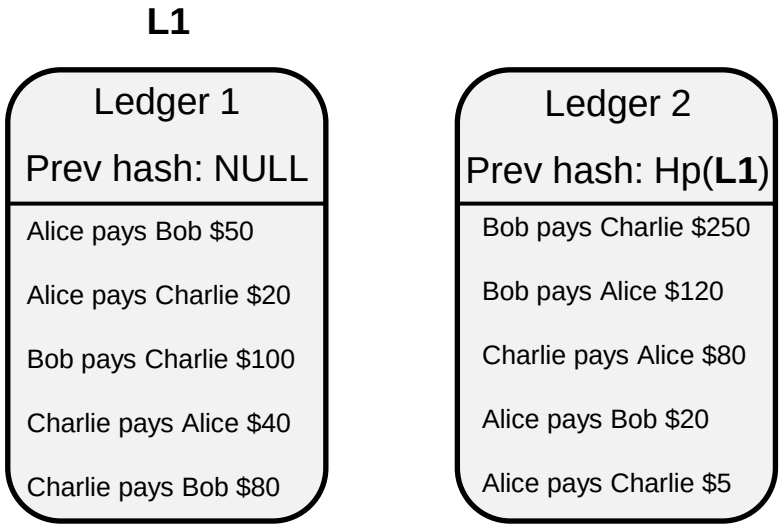
Consistent historic data

L1



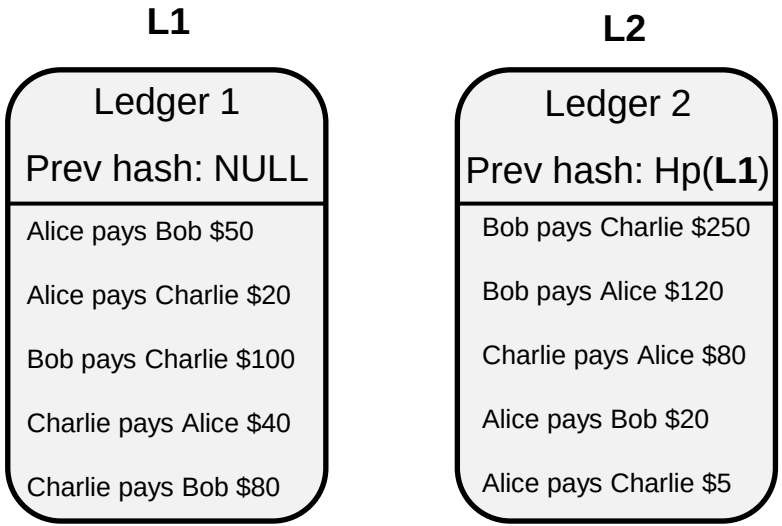
Blockchain

Consistent historic data



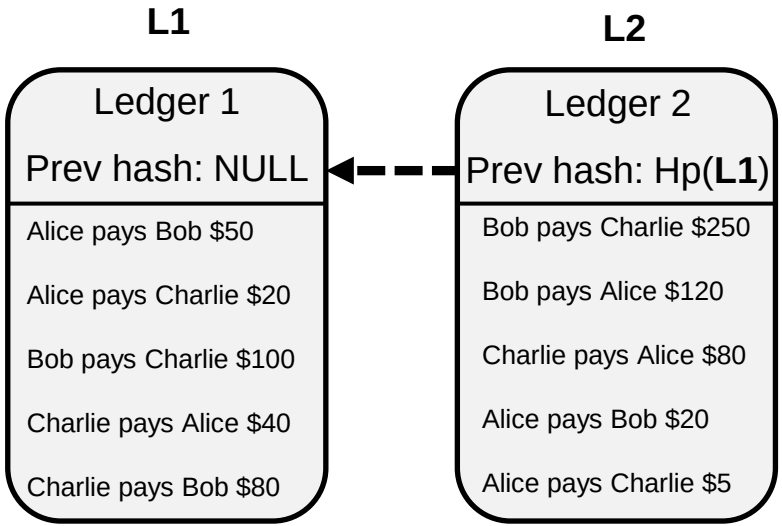
Blockchain

Consistent historic data



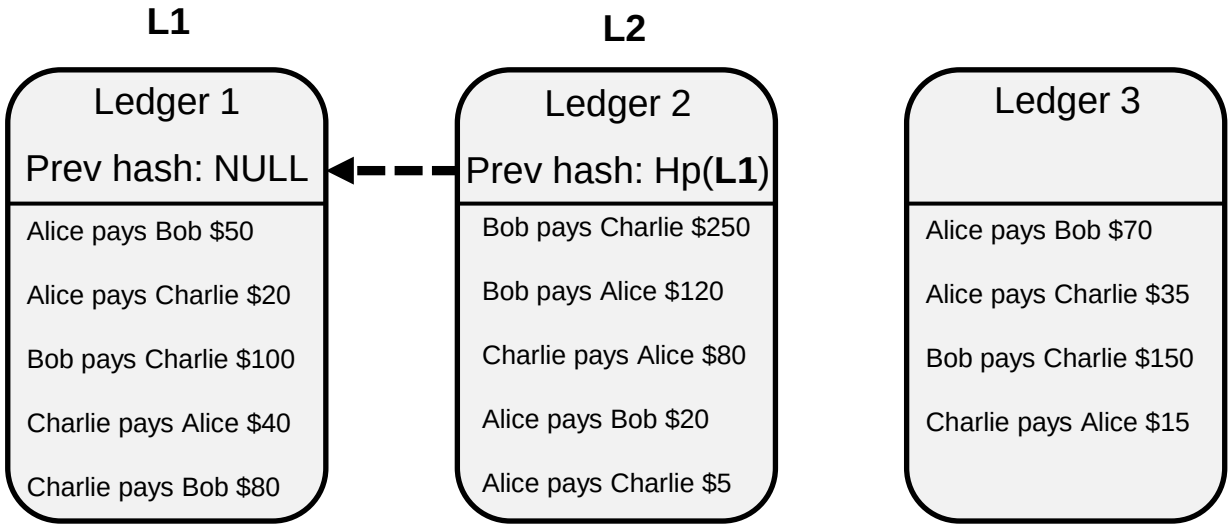
Blockchain

Consistent historic data



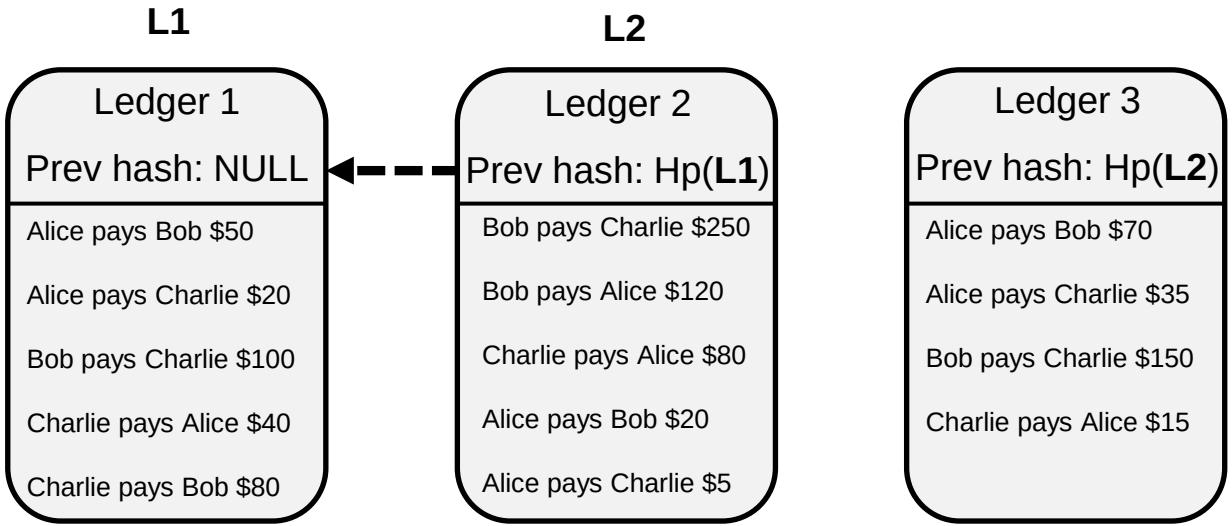
Blockchain

Consistent historic data



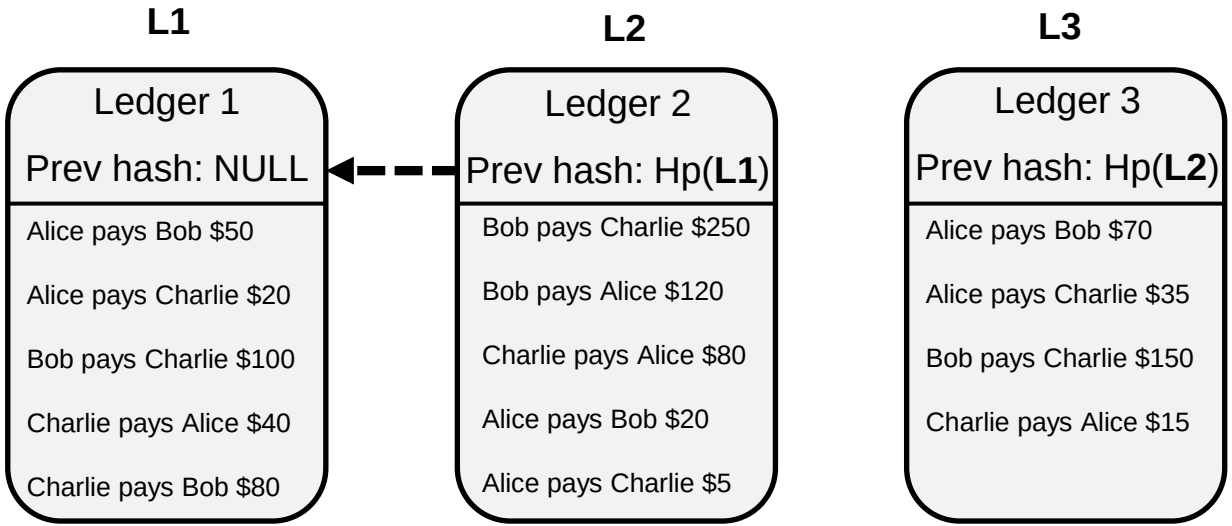
Blockchain

Consistent historic data



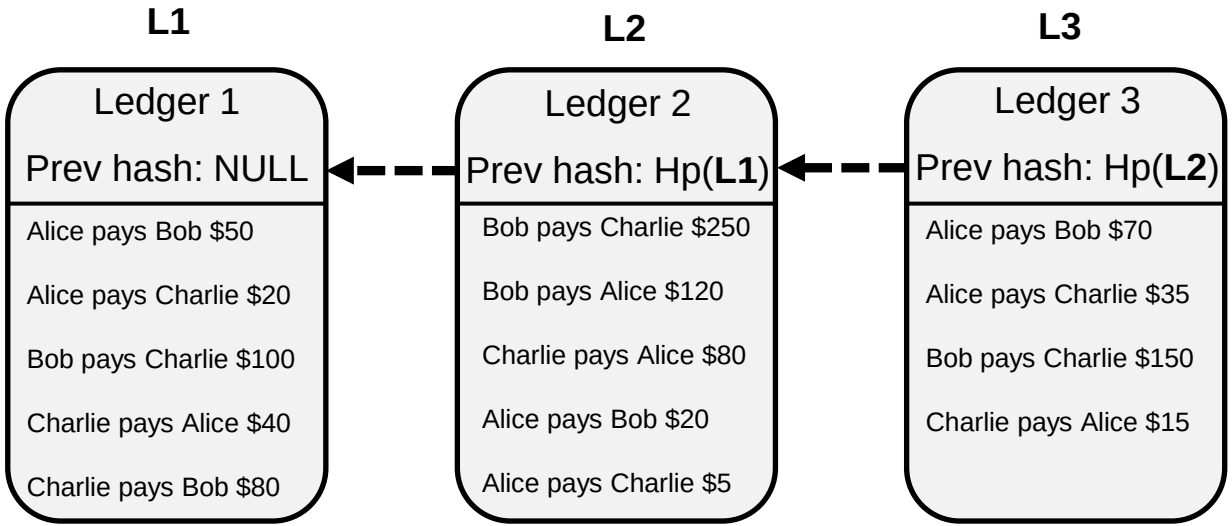
Blockchain

Consistent historic data



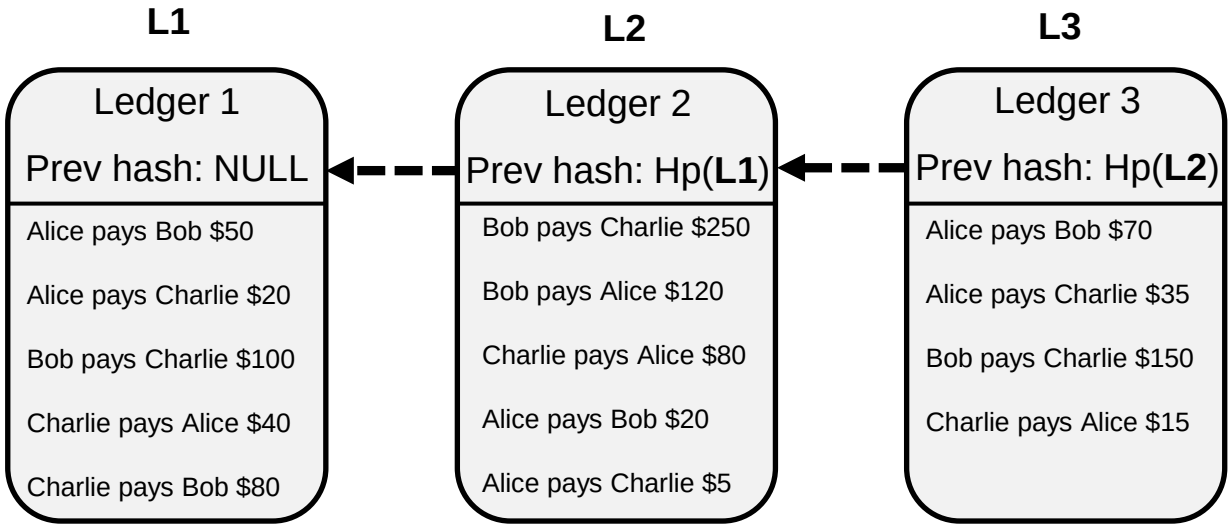
Blockchain

Consistent historic data



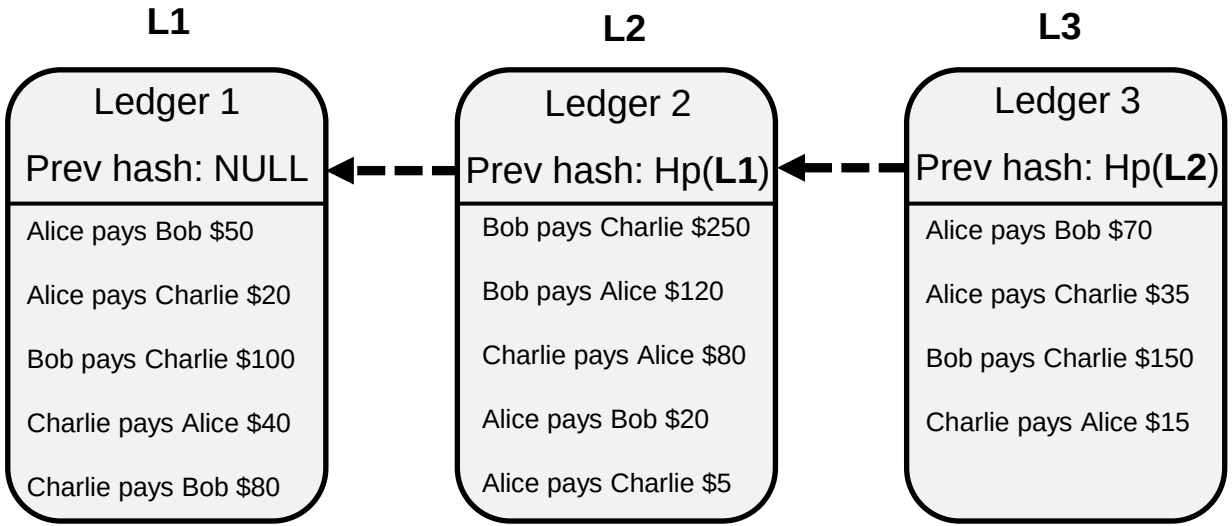
Blockchain

Consistent historic data

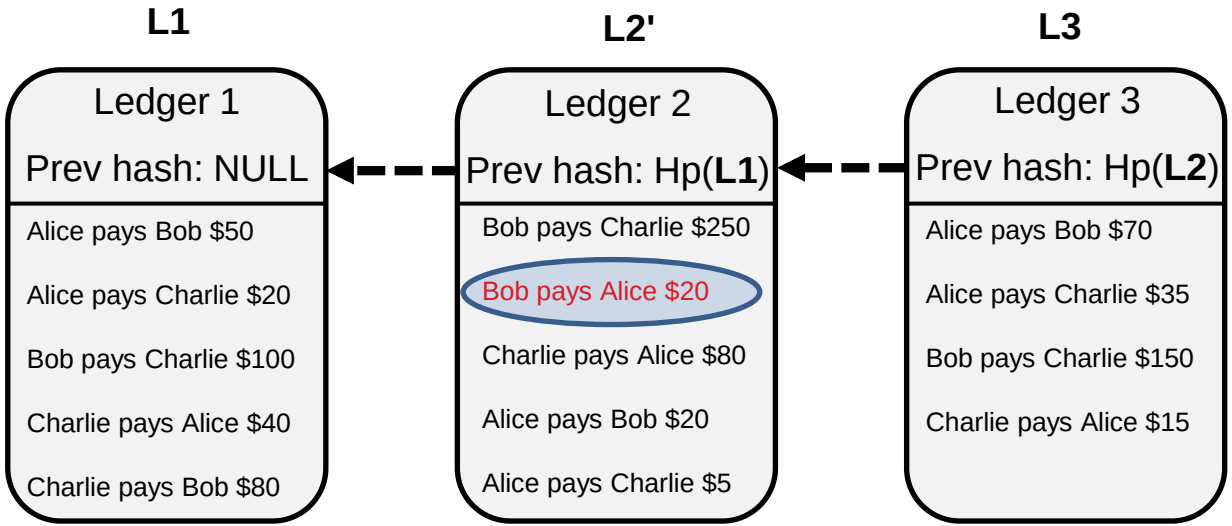


Blockchain

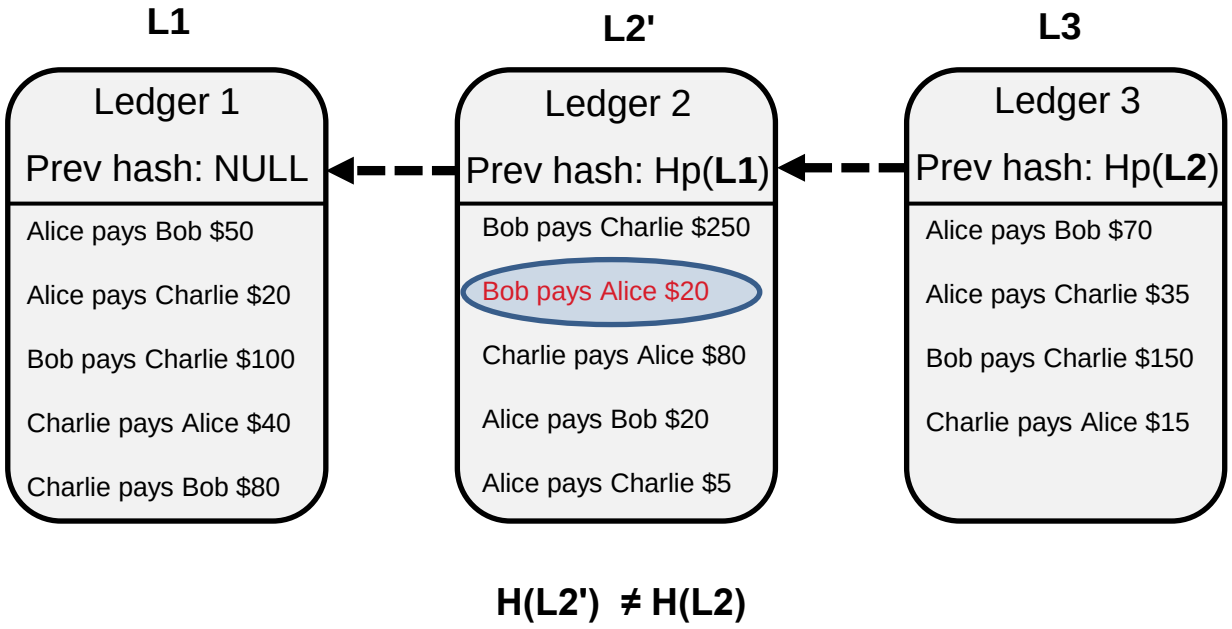
Immutability



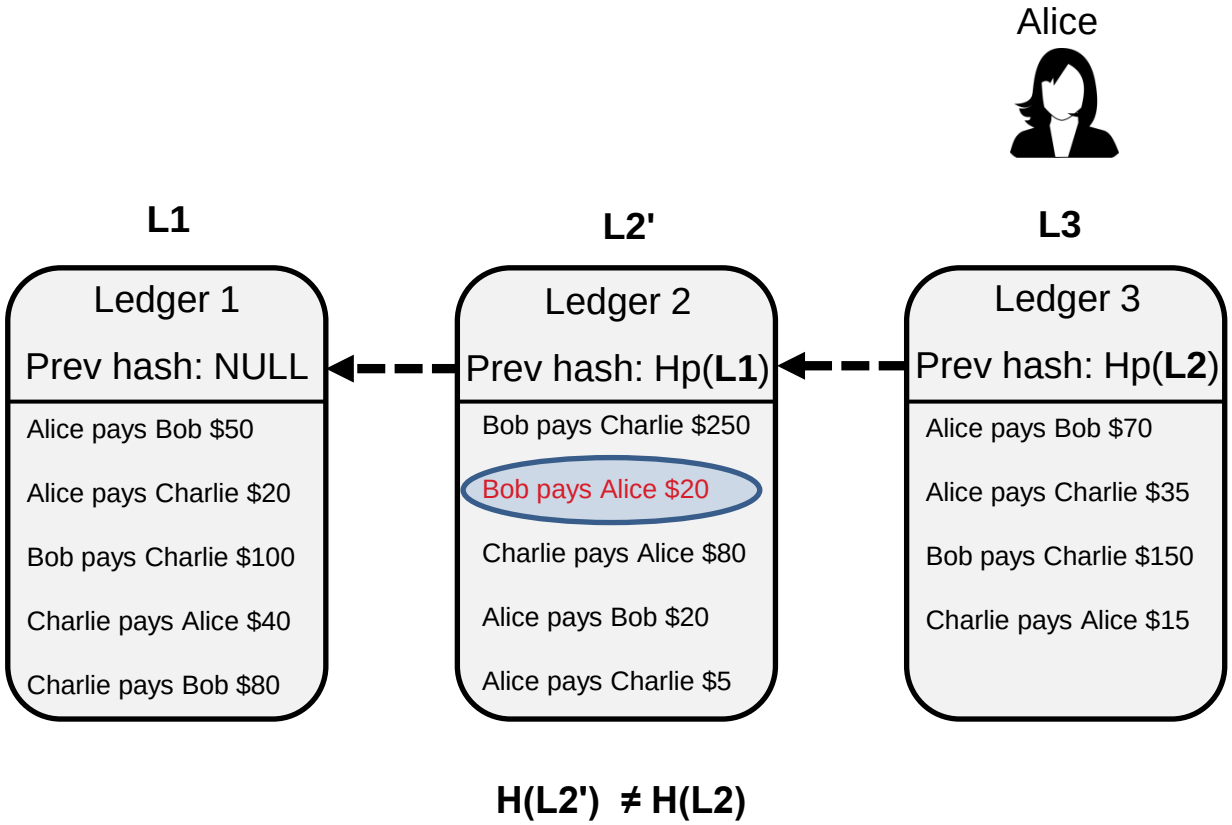
Immutability



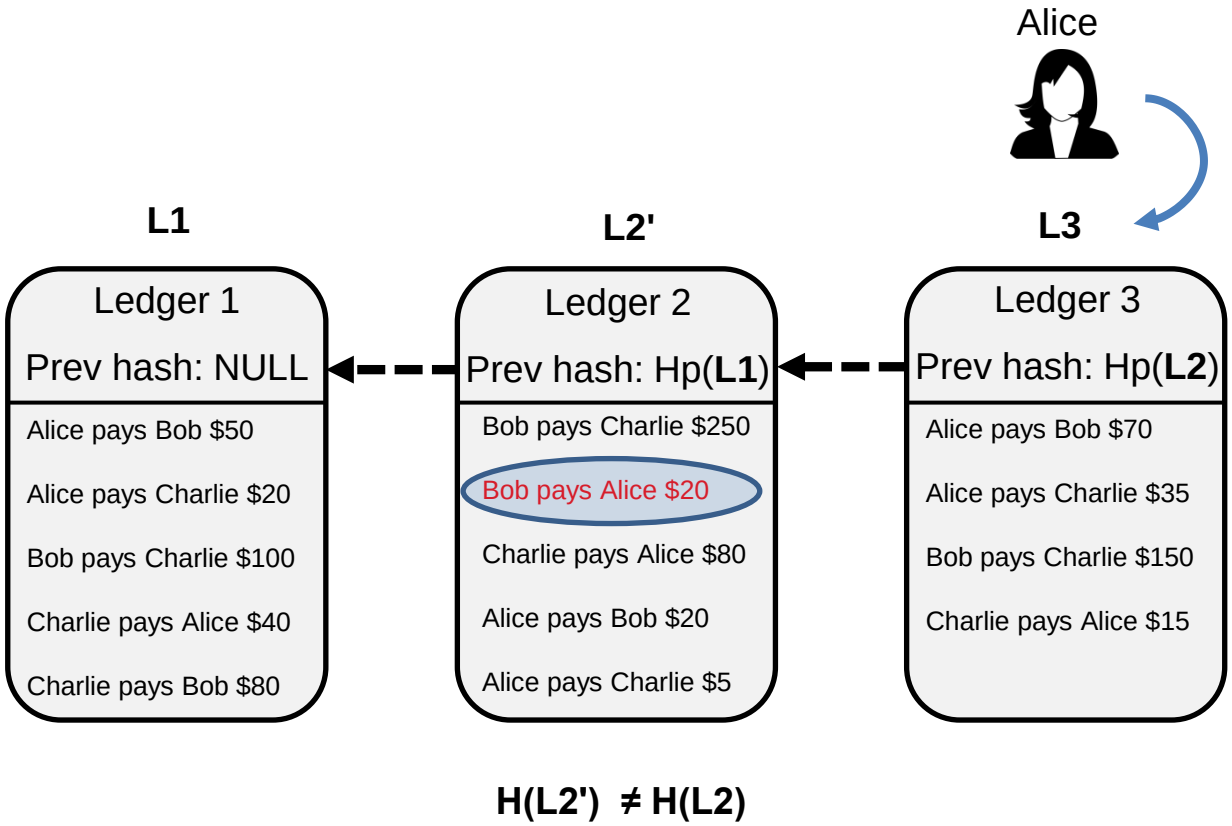
Immutability



Immutability



Immutability



H(L2') ≠ H(L2)

Immutability

Alice

L1

L2'

L3

Ledger 1	Ledger 2	Ledger 3
Prev hash: NULL	Prev hash: Hp(**L1**)	Prev hash: Hp(**L2**)
Alice pays Bob $50	Bob pays Charlie $250	Alice pays Bob $70
Alice pays Charlie $20	Bob pays Alice $20	Alice pays Charlie $35
Bob pays Charlie $100	Charlie pays Alice $80	Bob pays Charlie $150
Charlie pays Alice $40	Alice pays Bob $20	Charlie pays Alice $15
Charlie pays Bob $80	Alice pays Charlie $5	

H(L2') ≠ H(L2)

Immutability

Alice

L1

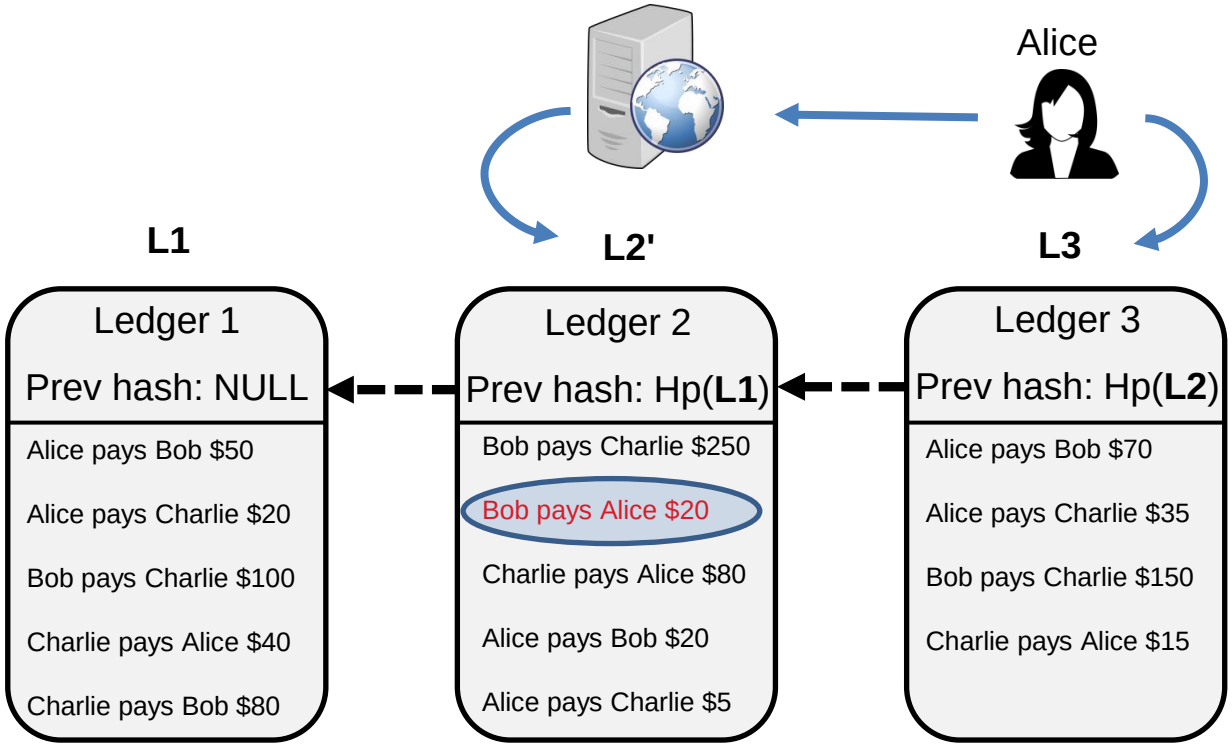
L2'

L3

Ledger 1	Ledger 2	Ledger 3
Prev hash: NULL	Prev hash: Hp(**L1**)	Prev hash: Hp(**L2**)
Alice pays Bob $50	Bob pays Charlie $250	Alice pays Bob $70
Alice pays Charlie $20	Bob pays Alice $20	Alice pays Charlie $35
Bob pays Charlie $100	Charlie pays Alice $80	Bob pays Charlie $150
Charlie pays Alice $40	Alice pays Bob $20	Charlie pays Alice $15
Charlie pays Bob $80	Alice pays Charlie $5	

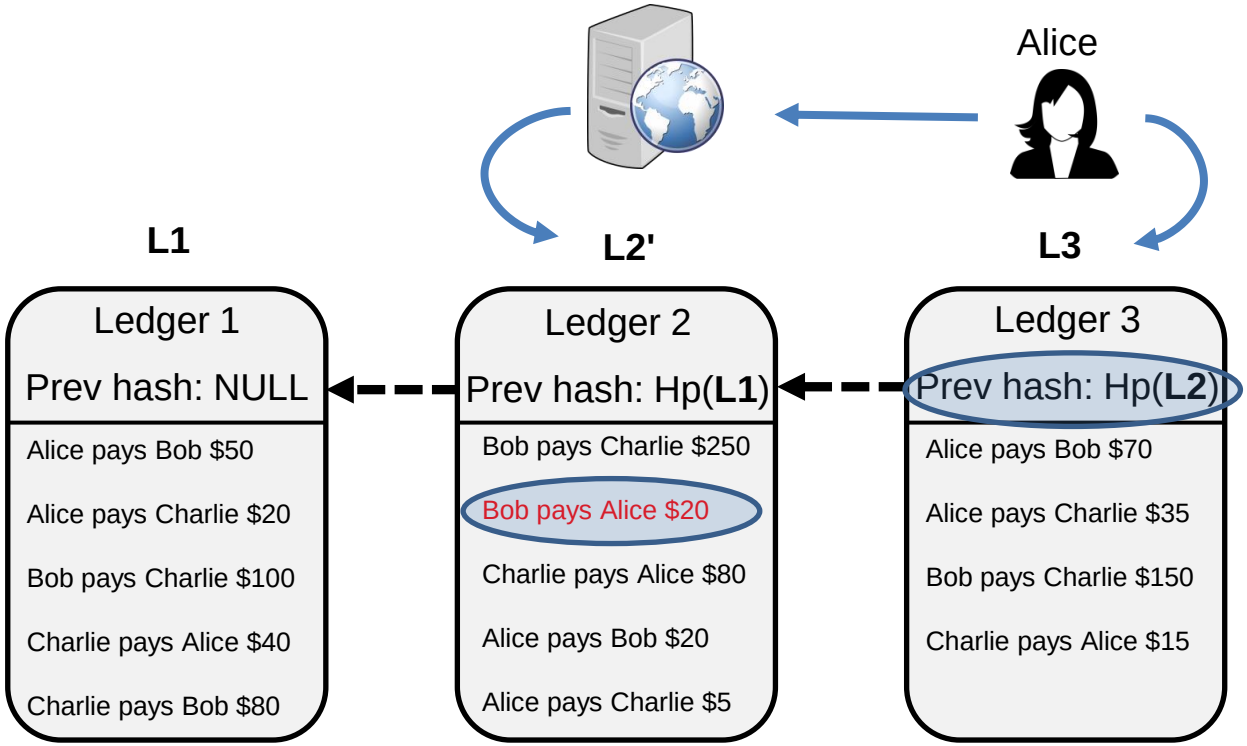
H(L2') ≠ H(L2)

Immutability



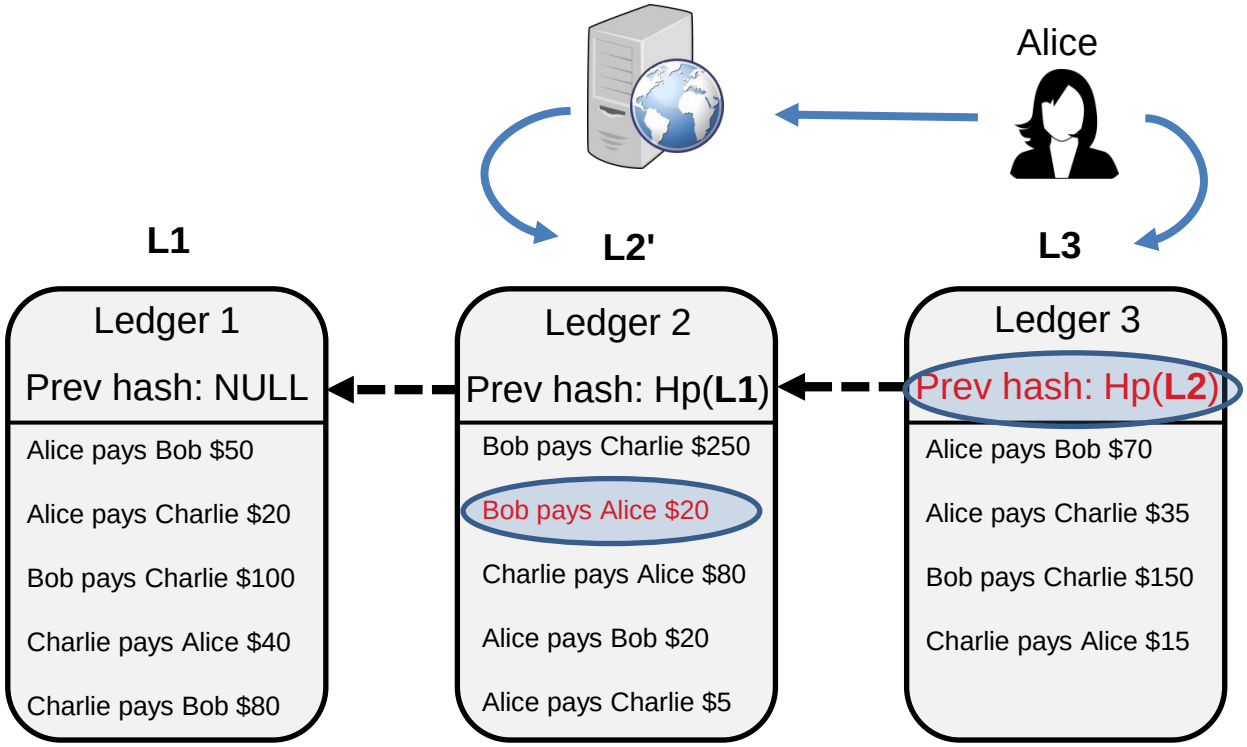
H(L2') ≠ H(L2)

Immutability

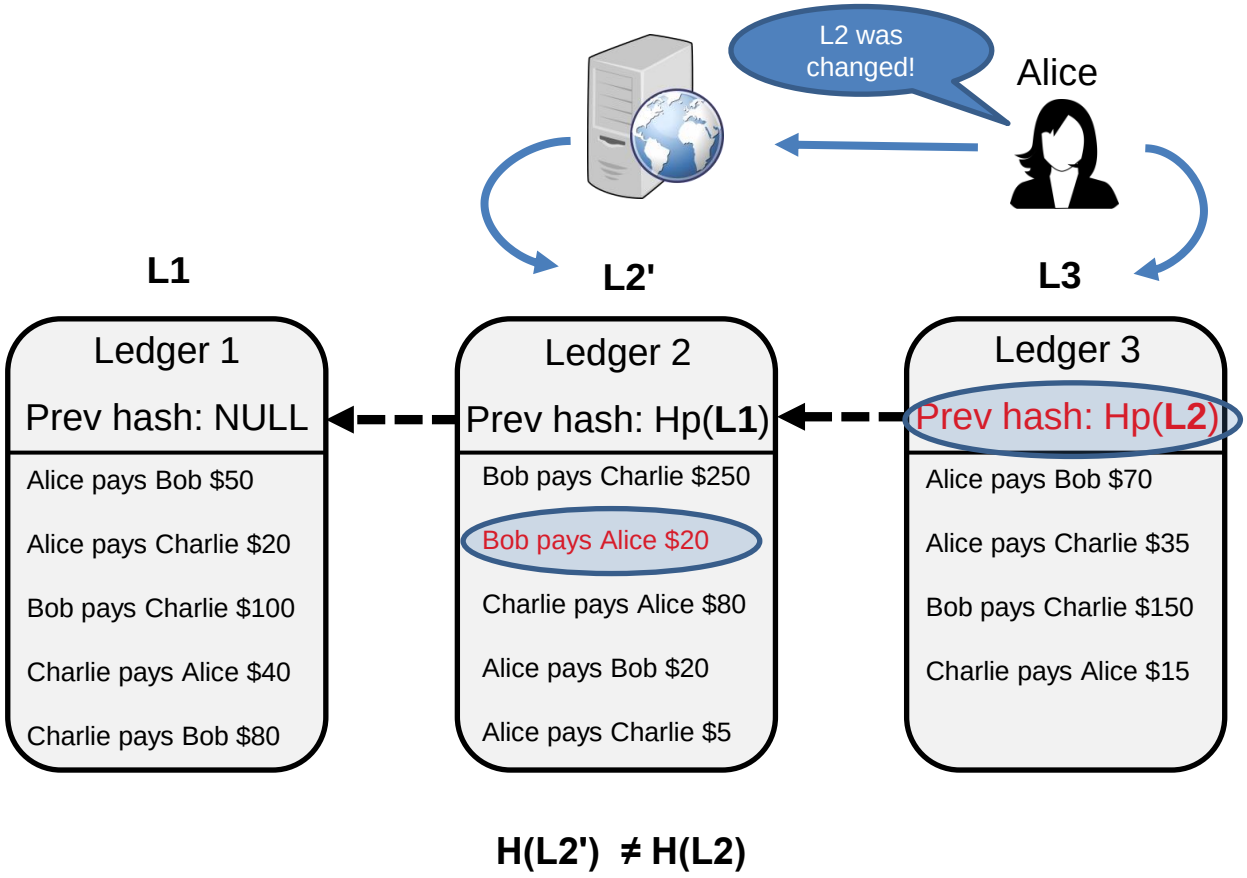


H(L2') ≠ H(L2)

Immutability

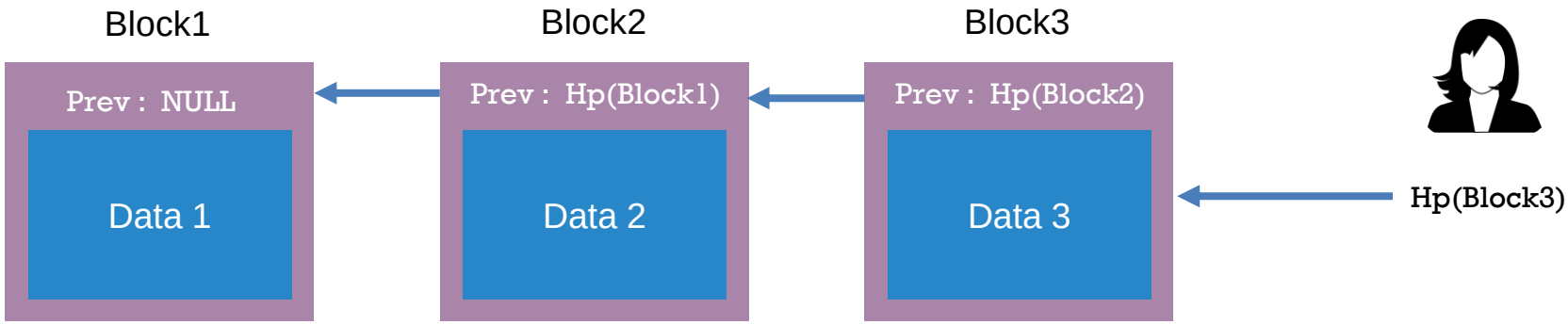


Immutability



Use of Blockchain

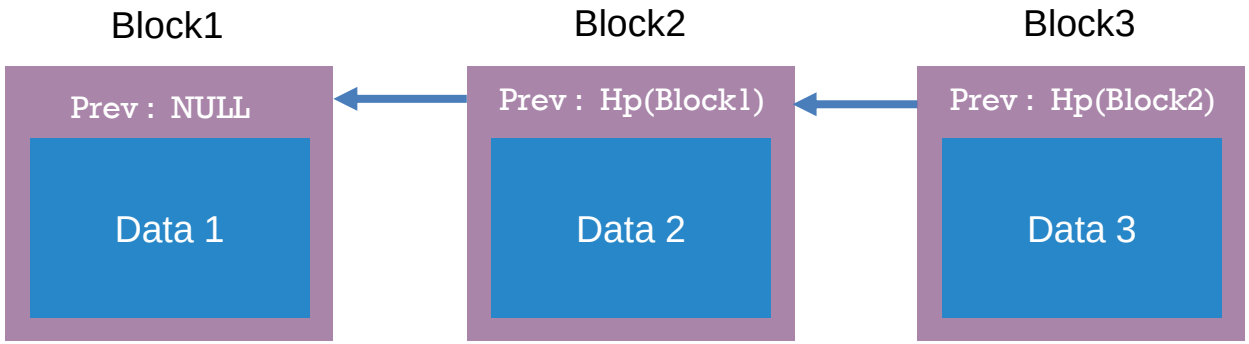
How to detect if there is a change?



If we only have the head

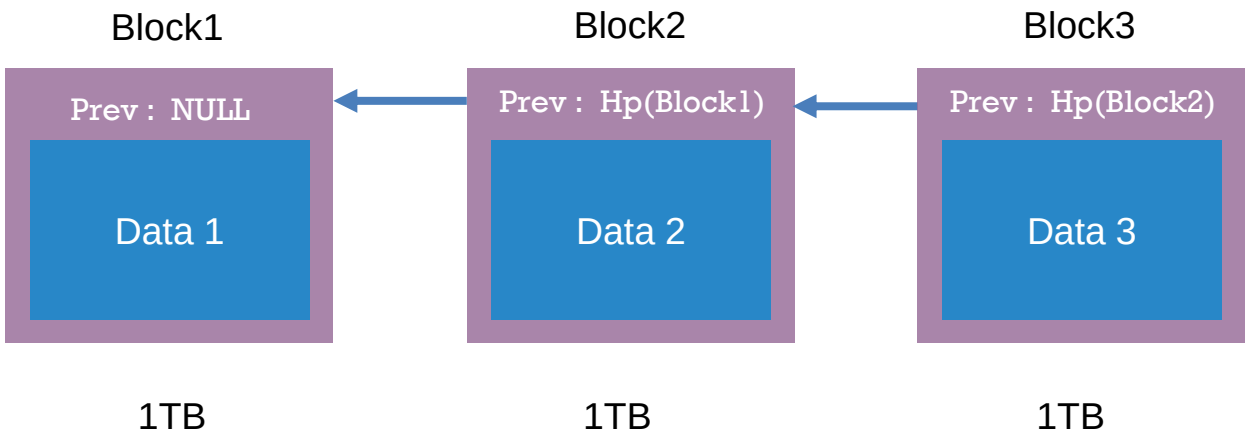
Use of Blockchain

How to detect if there is a change?



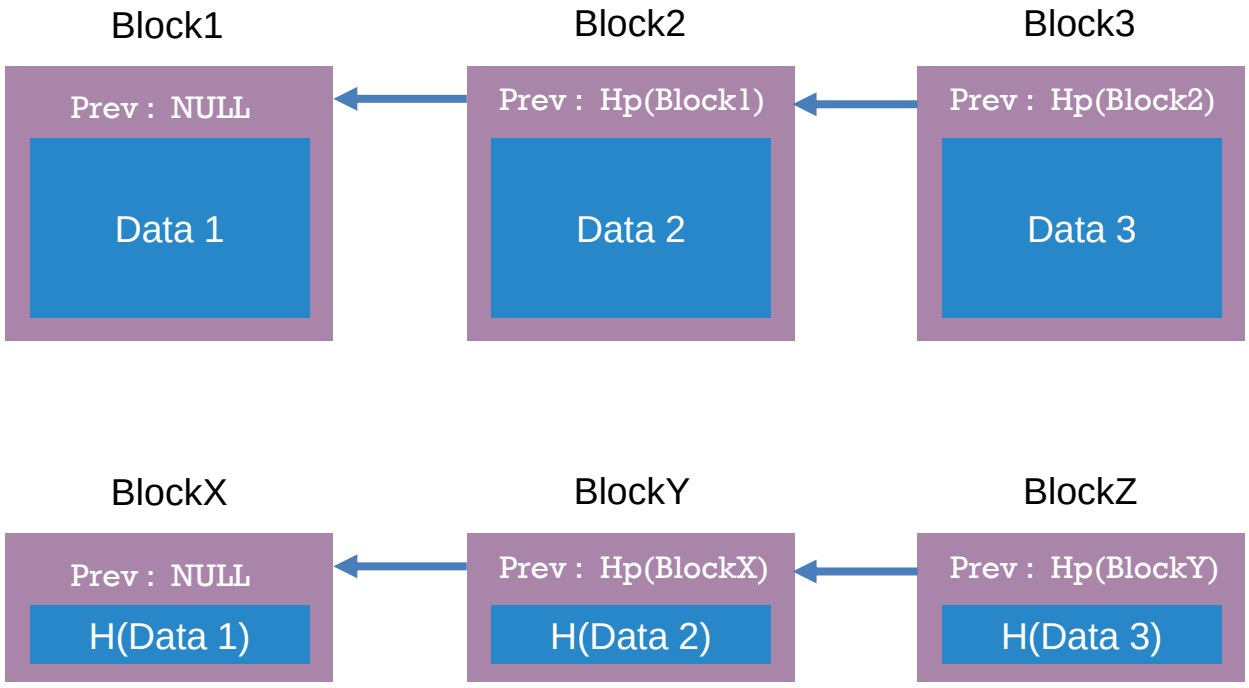
Use of Blockchain

How to detect if there is a change?



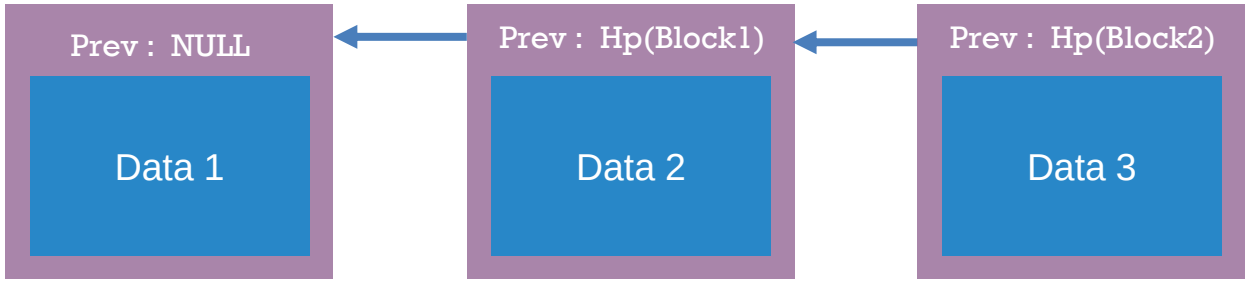
Use of Blockchain

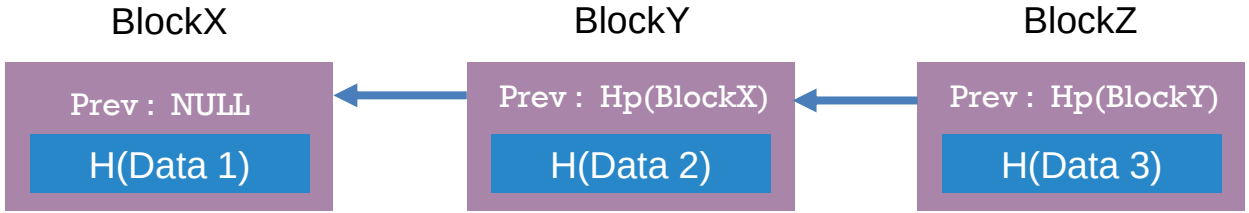
How to detect if there is a change?



Use of Blockchain

How to detect if there is a change?





Use of Blockchain

How to detect if there is a change?

Block1

Prev : NULL

Data 1

Block2

Prev : Hp(Block1)

Fake Data 2

Block3

Prev : Hp(Block2)

Data 3

BlockX

Prev : NULL

H(Data 1)

BlockY

Prev : Hp(BlockX)

H(Data 2)

BlockZ

Prev : Hp(BlockY)

H(Data 3)

Use of Blockchain

How to detect if there is a change?

Block1

Prev : NULL

Data 1

Block2

Prev : Hp(Block1)

Fake Data 2

Block3

Prev : Hp(Block2)

Data 3

BlockX

Prev : NULL

H(Data 1)

BlockY

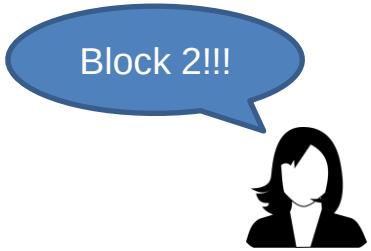
Prev : Hp(BlockX)

H(Data 2)

BlockZ

Prev : Hp(BlockY)

H(Data 3)



Blockchain and commitments

Blockchain and commitments

Day1

Data Alice

Data Bob

Blockchain and commitments

Day1

Data 1

Data Alice

Data Bob

Blockchain and commitments

Day1

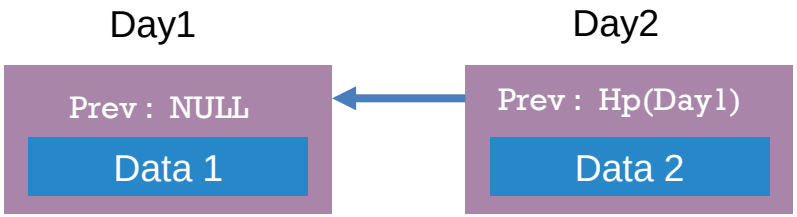
Prev : NULL

Data 1

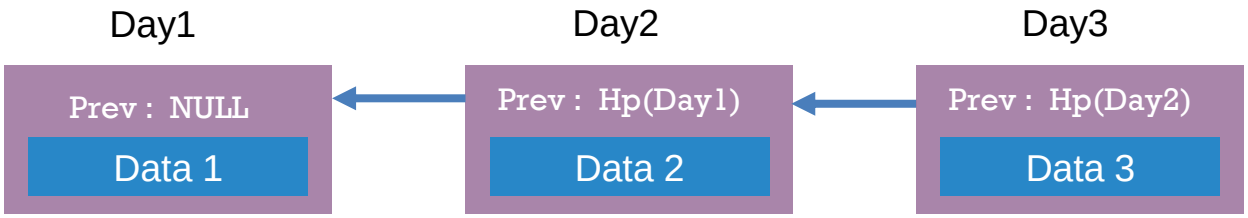
Data Alice

Data Bob

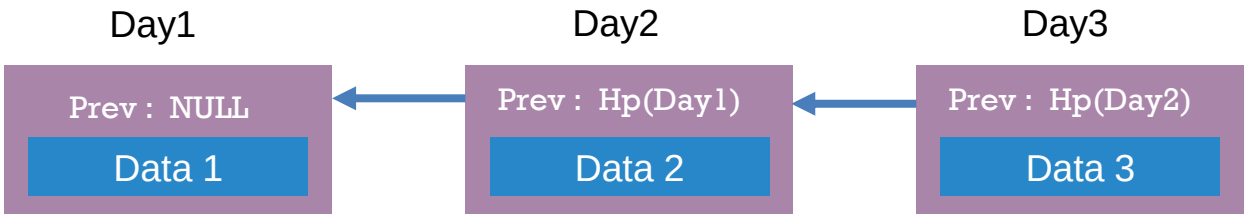
Blockchain and commitments



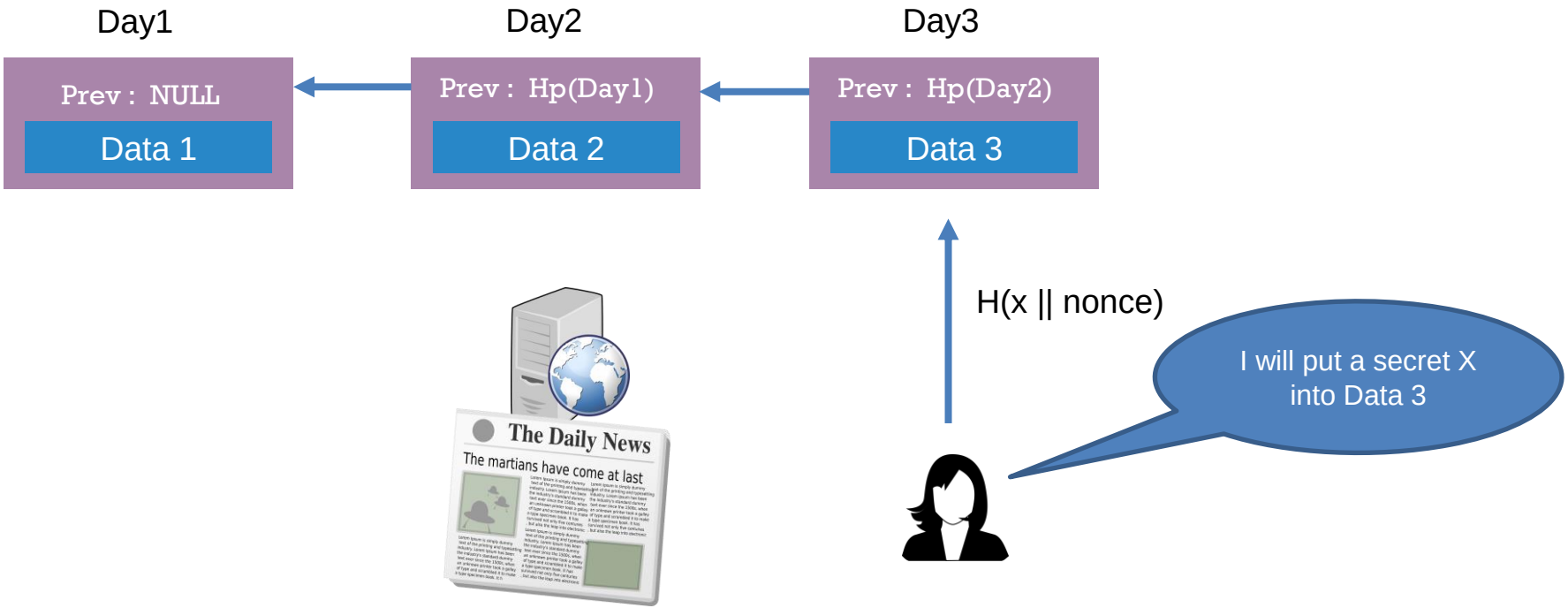
Blockchain and commitments



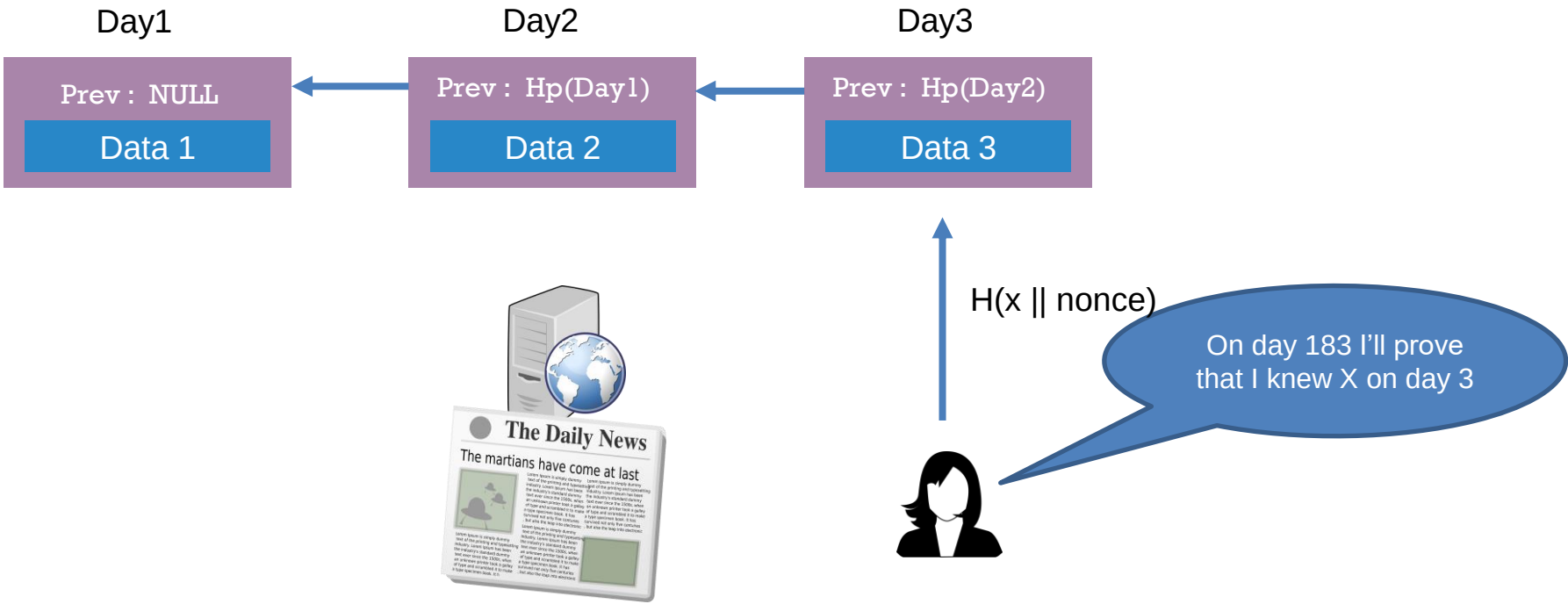
Blockchain and commitments



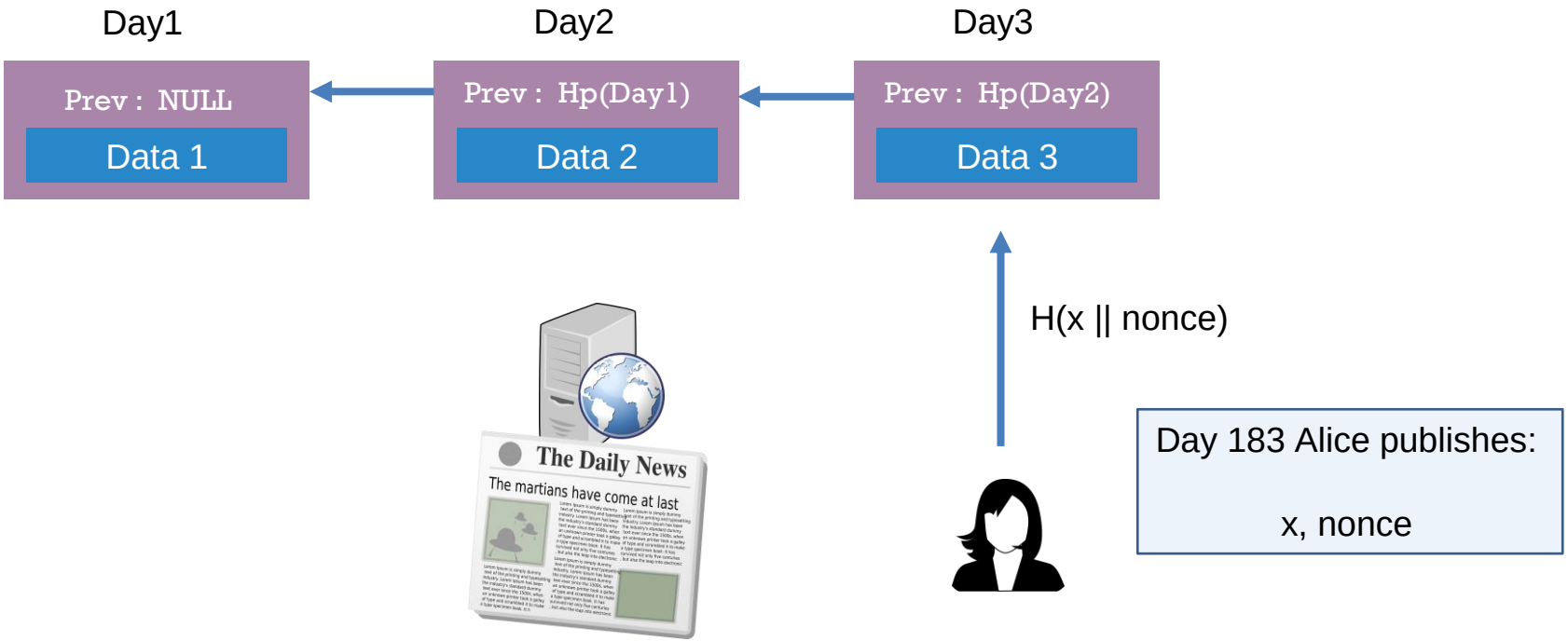
Blockchain and commitments



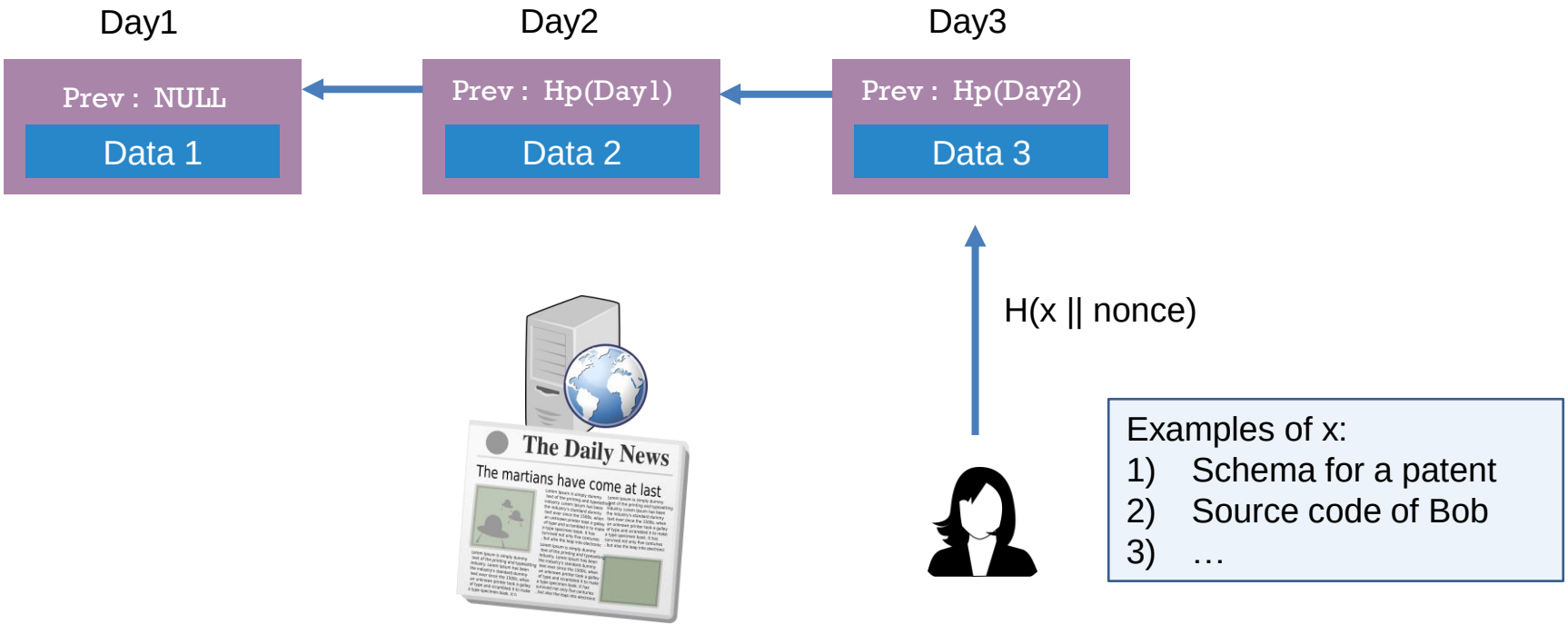
Blockchain and commitments



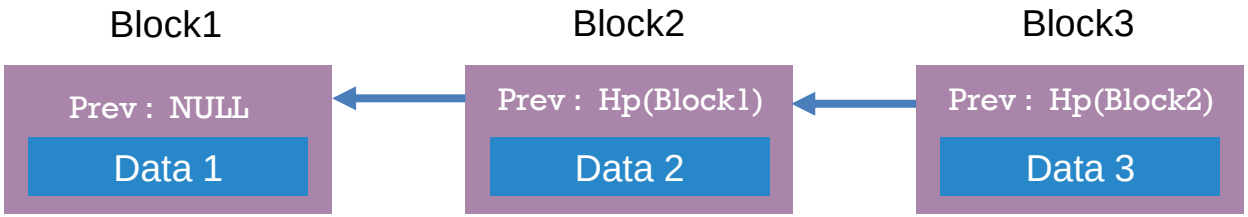
Blockchain and commitments



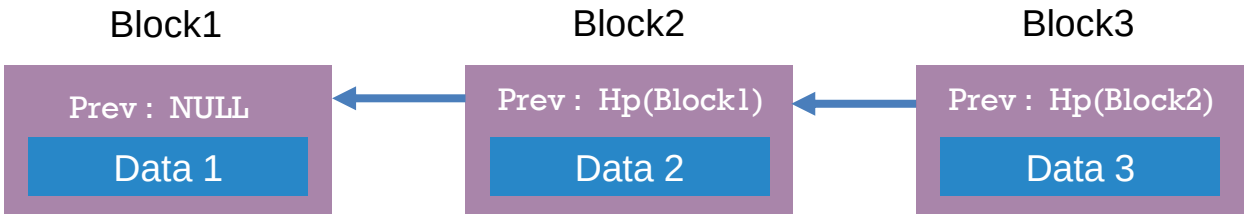
Blockchain and commitments



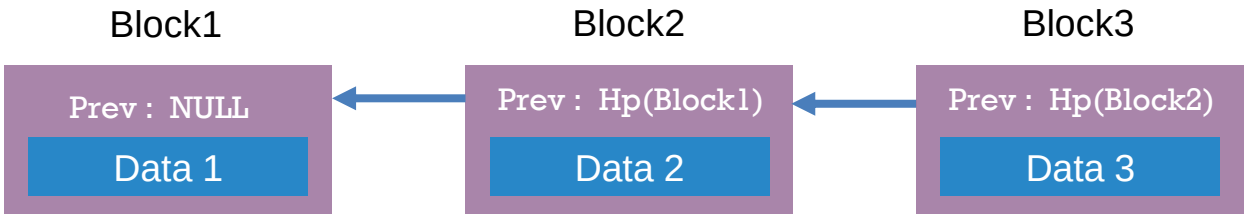
One weakness

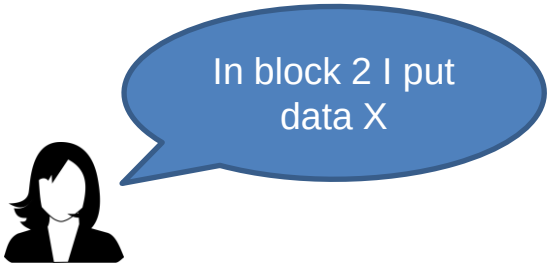


One weakness

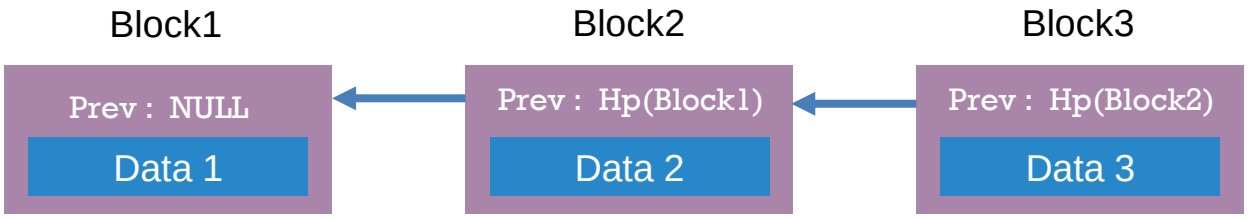


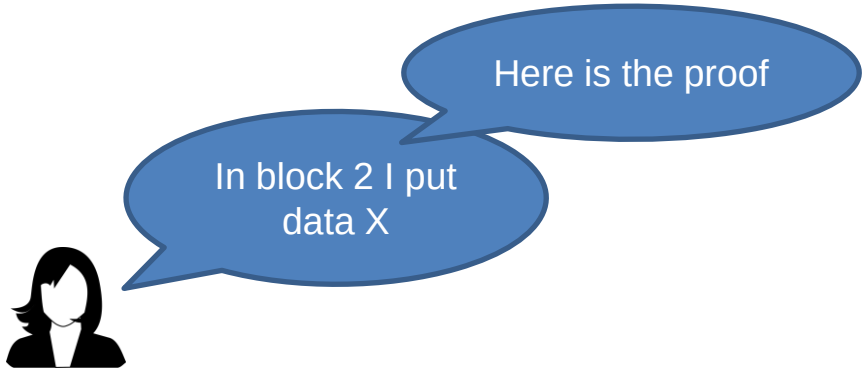
One weakness



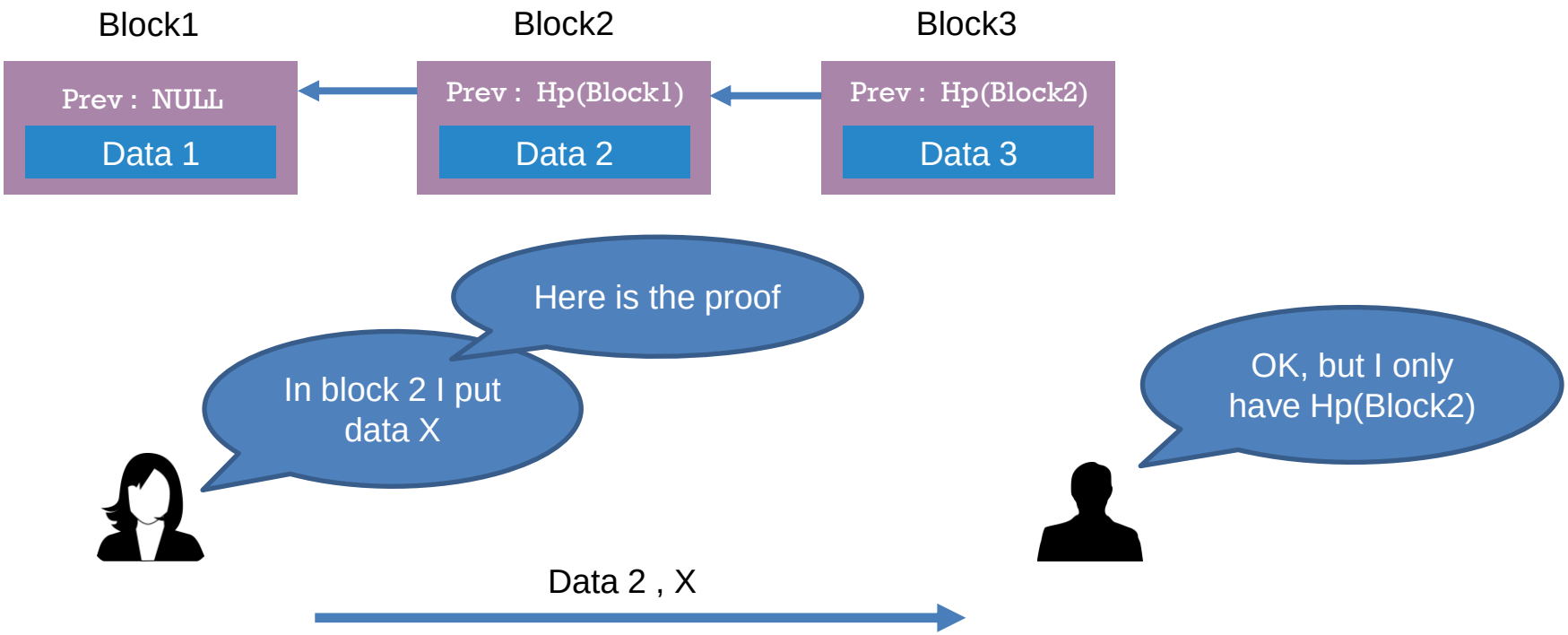


One weakness

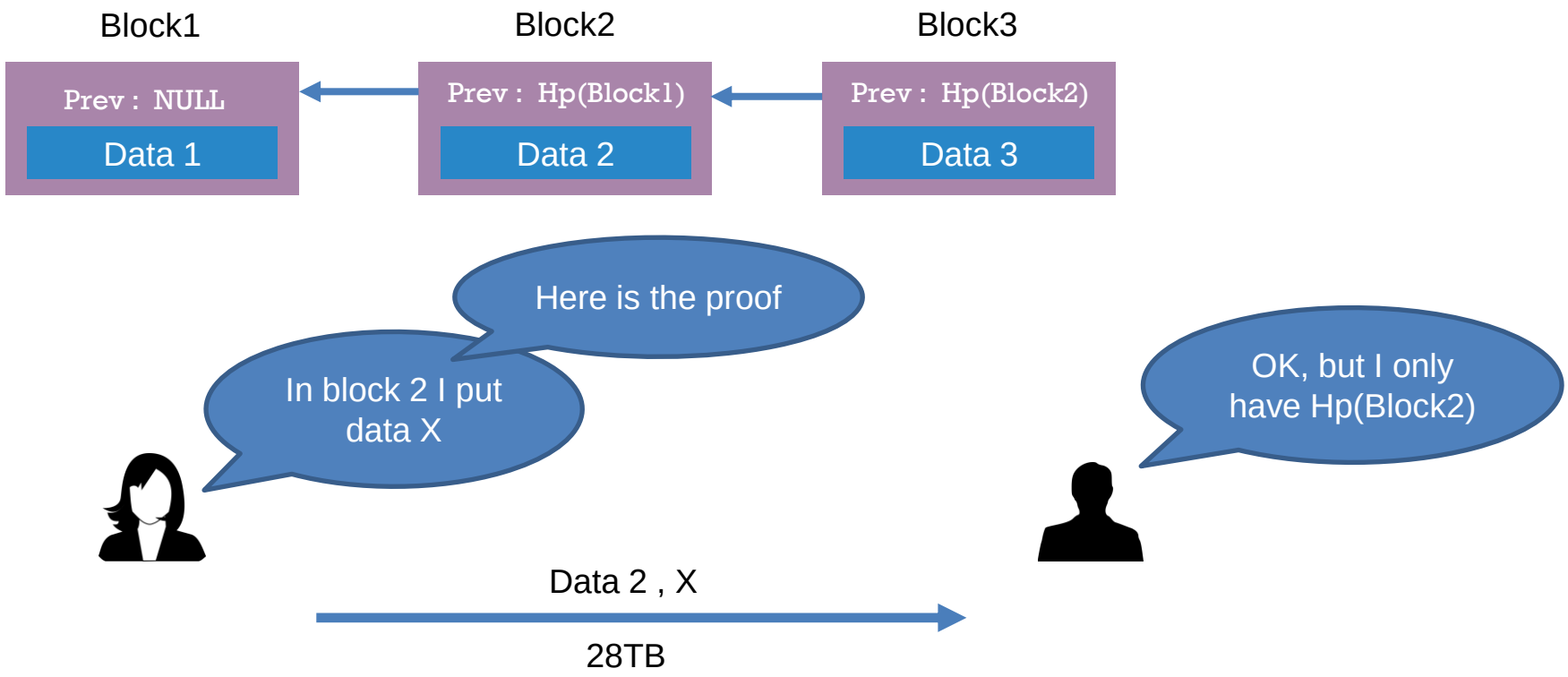




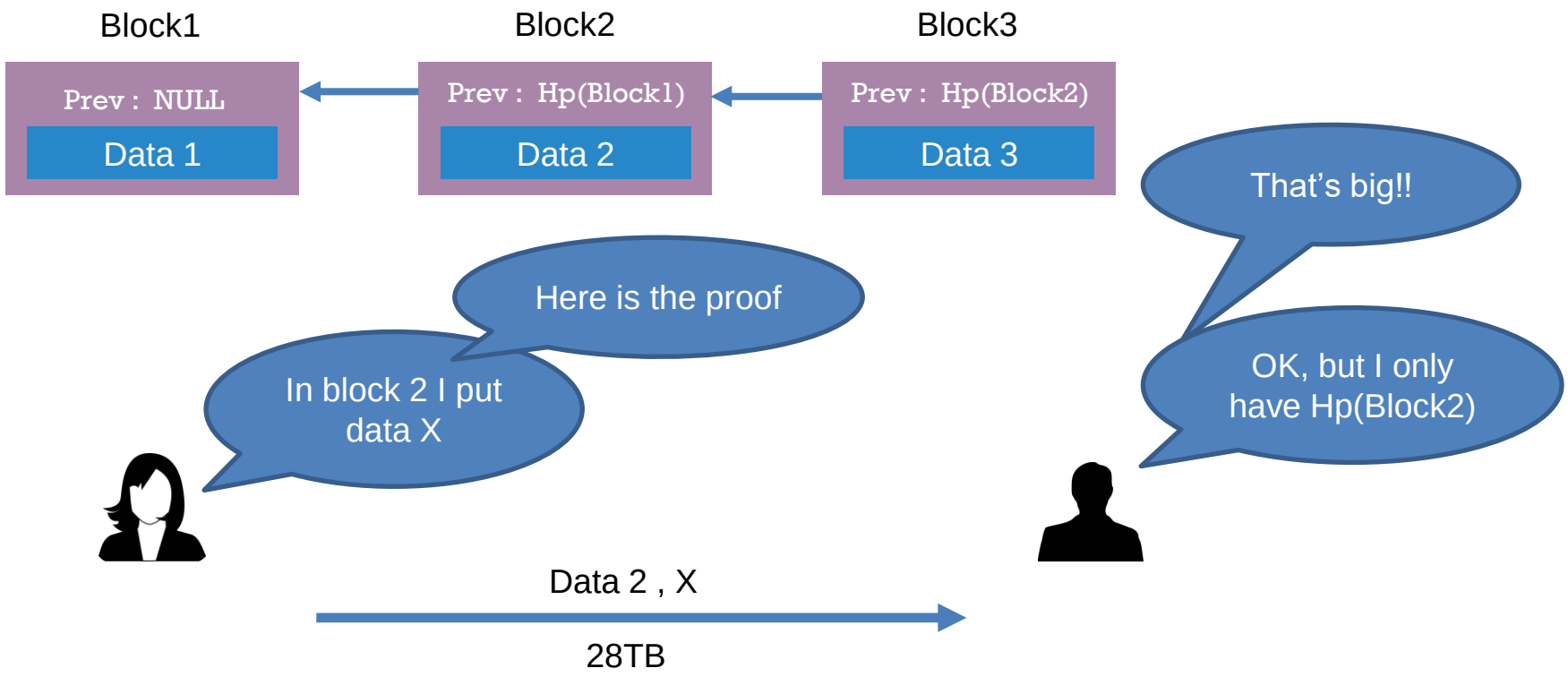
One weakness



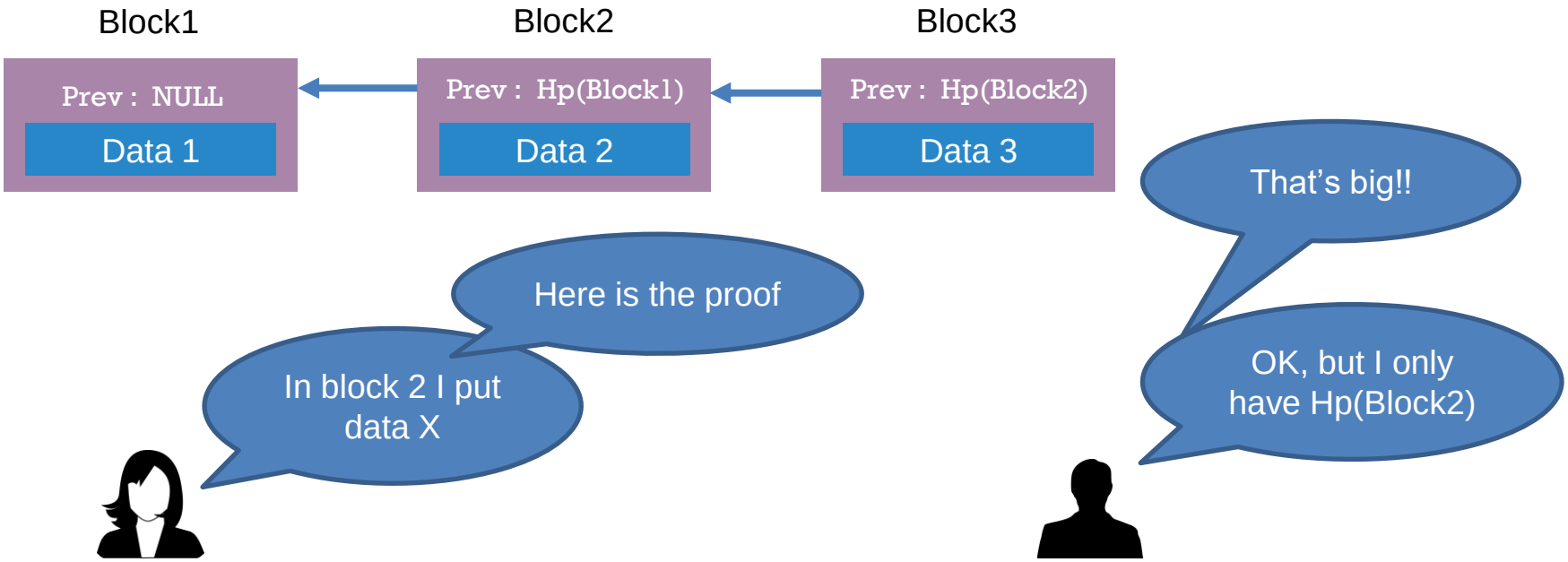
One weakness



One weakness



One weakness



How to make the proof more efficient???

reading

Narayanan et. Al:

- Chapter 1.2
- Chapter 9.1

Practice time!!!

Exercizes!!!

- Implement blockchain
- Run against the test data
- Different ways of implementing this

Let’s see the two class we need to implement!!!