

:-)

# .about

- Java/Scala dev at @SoftwareMill
- ~ 1.5 at company
- Tesco

## .about ++

- Java/Scala dev at @SoftwareMill
  - ~ 1.5 at company
  - Tesco
- 
- Robimy Blockchain
  - Teraz Hyperledger Fabric, czyli prywatny BC
  - ETH -> ICO

# .talk rules

- raczej będzie luźno ;)

# .talk rules

- raczej będzie luźno ;)
- poor context switching - postaram się robić przerwy na pytania

# .talk rules

- raczej będzie luźno ;)
- poor context switching - postaram się robić przerwy na pytania
- Blockchain to temat obszerny - postaram się odpowiedzieć na wszystkie pytania

# .talk rules

- raczej będzie luźno ;)
- poor context switching - postaram się robić przerwy na pytania
- Blockchain to temat obszerny - postaram się odpowiedzieć na wszystkie pytania
- 4 części

# Prywatny Blockchain

...

na co to komu ?



# .why ?

- bo to dobry buzzword ;)

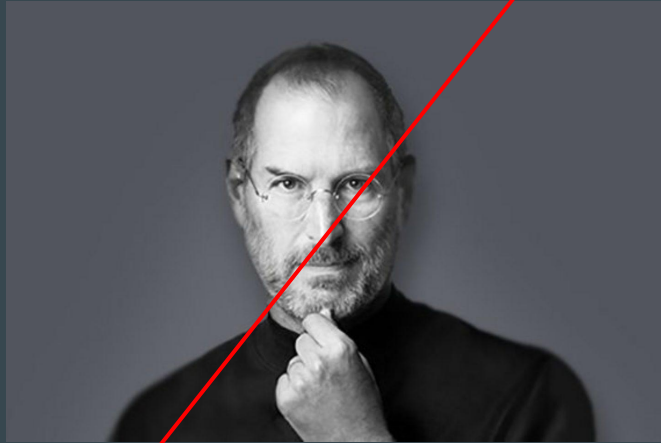
# .why ?

- bo to dobry buzzword ;)
- a buzzwordy trudno jest osadzić w rzeczywistości
  - ciężko managerom => \$\$\$
  - ciężko nam => CV
  - Takie coś prowadzi Nas do głównej maksymy w IT...

.why ?



.why ?



.why ?



.why ?



# 01 .meet blockchain

# 01 .meet blockchain - what is it ?

to taka próba odwrócenia ról



# 01 .meet blockchain - what is it ?



Alice



Bob

# 01 .meet blockchain - what is it ?



# 01 .meet blockchain - what is it ?



# 01 .meet blockchain - what is it ?



- double spend
- ograniczone zaufanie

# 01 .meet blockchain - what is it ?



# 01 .meet blockchain - what is it ?



# 01 .meet blockchain - what is it ?

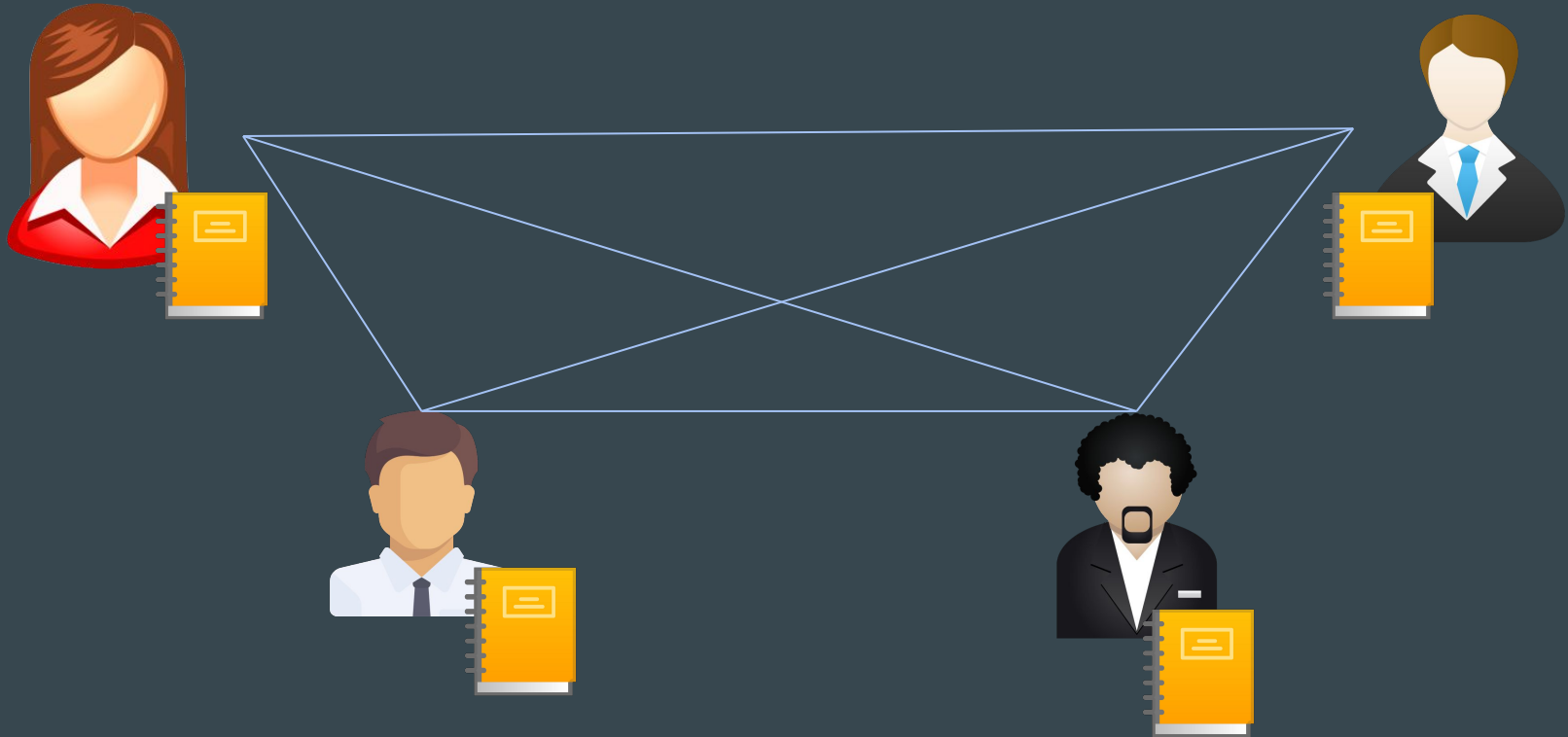


# 01 .meet blockchain - what is it ?

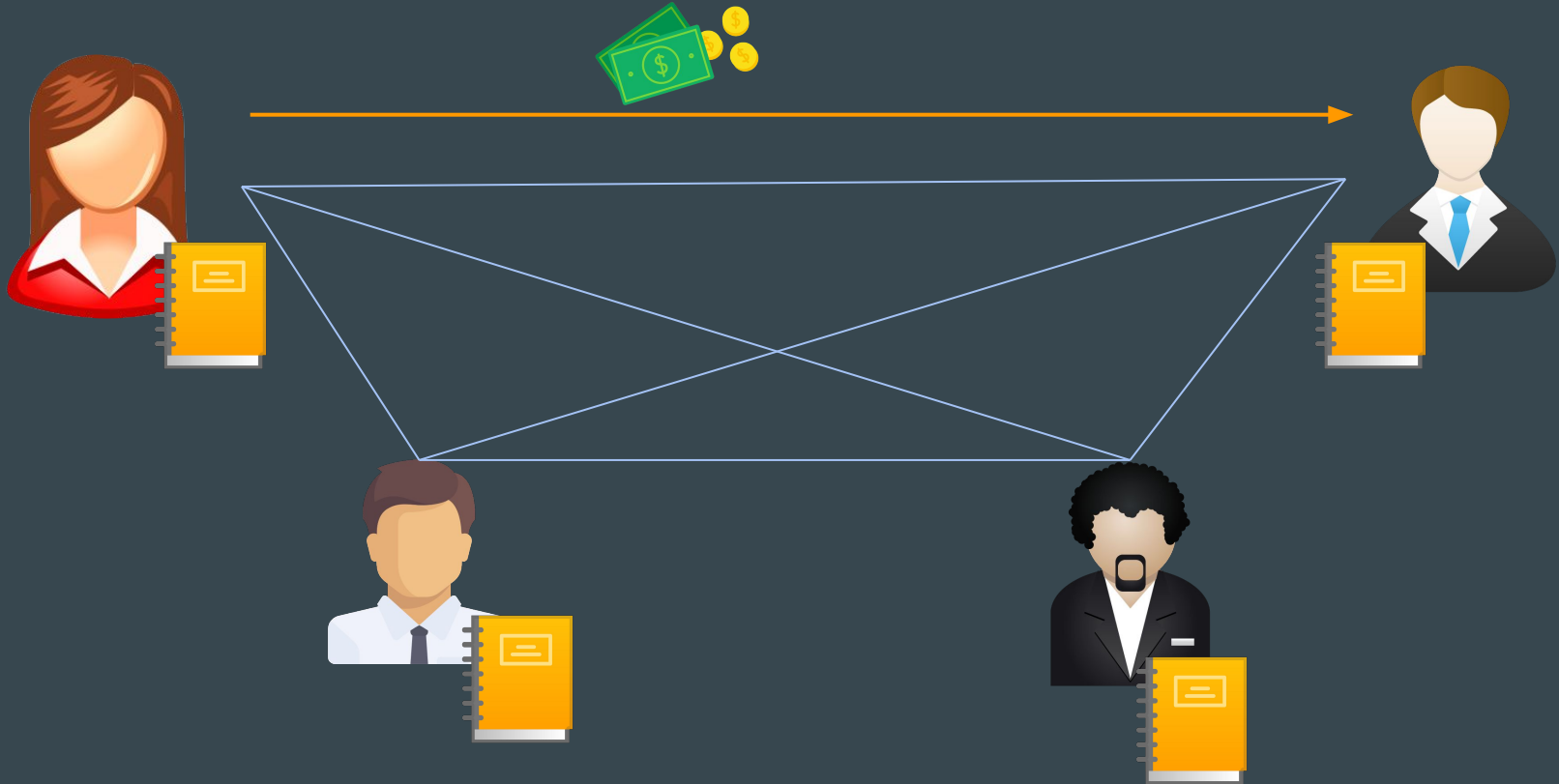




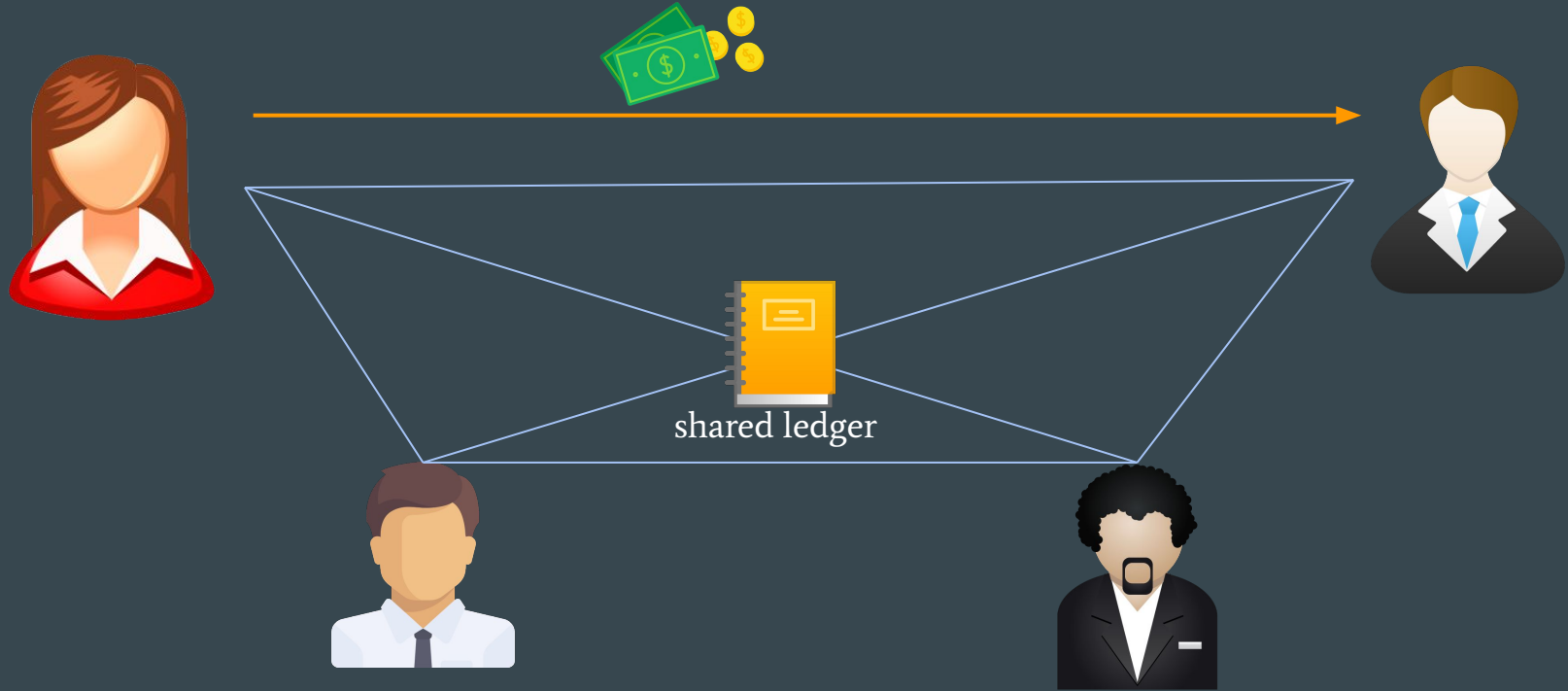
# 01 .meet blockchain - what is it ?



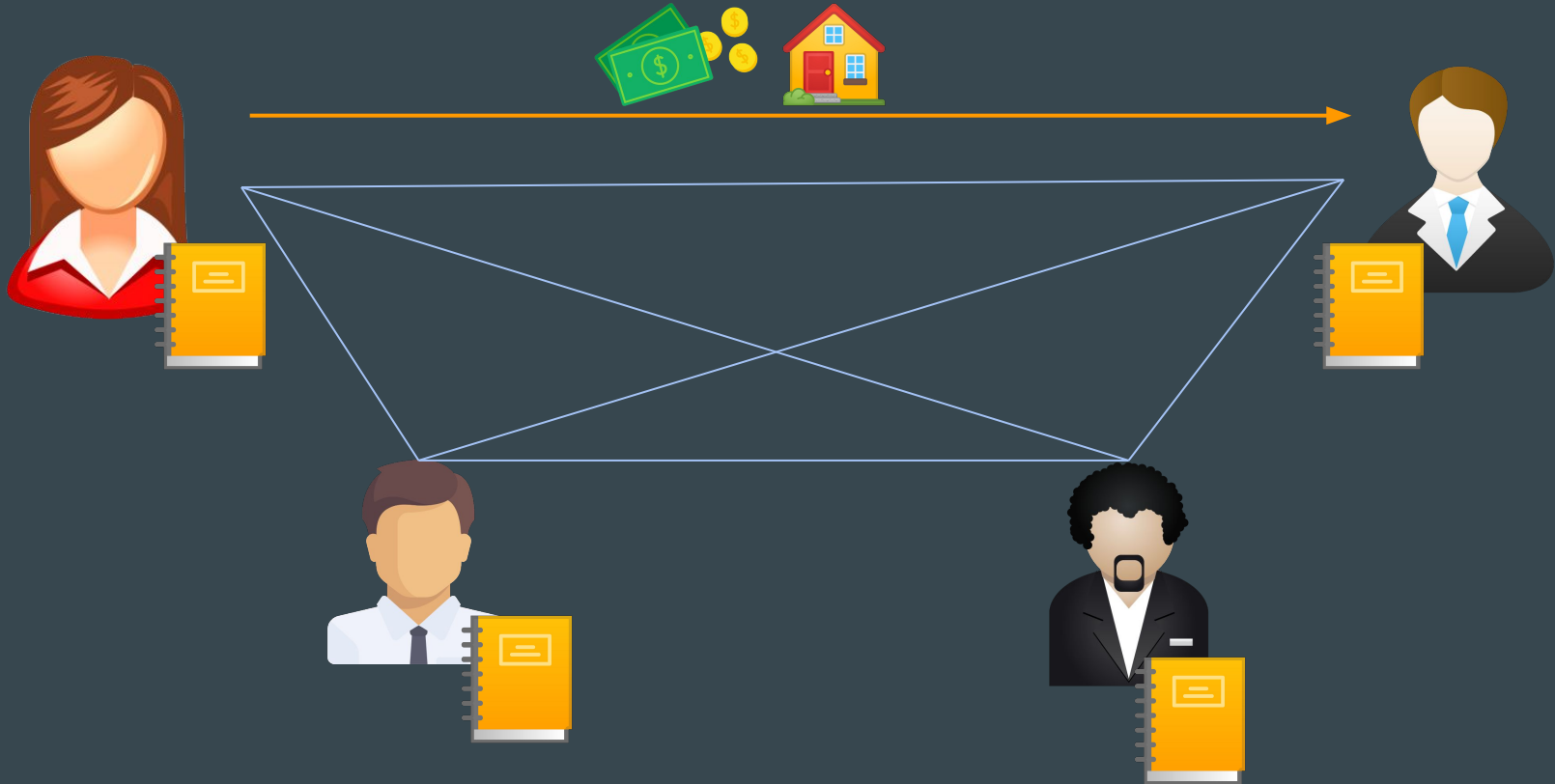
# 01 .meet blockchain - what is it ?



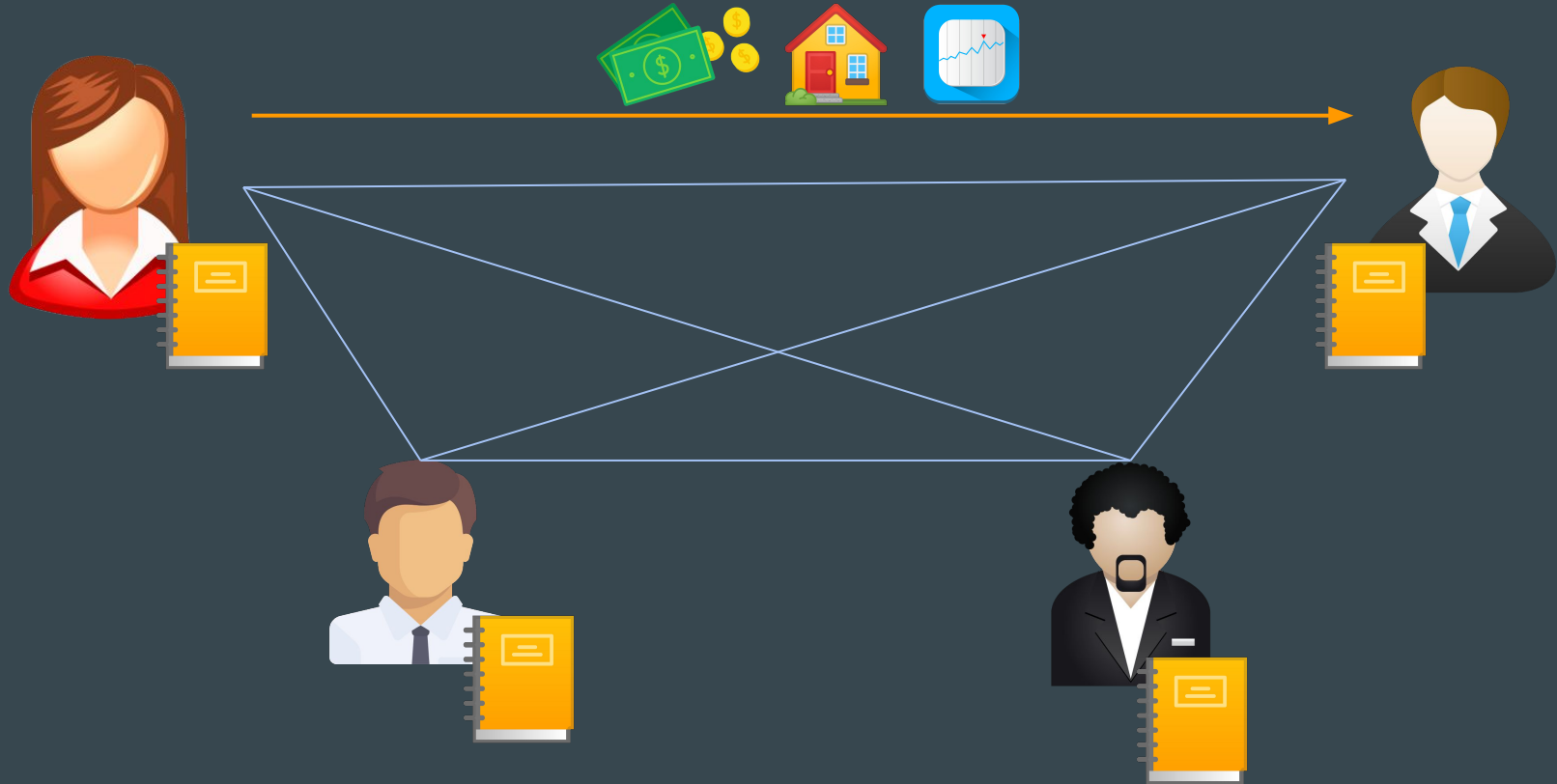
# 01 .meet blockchain - what is it ?



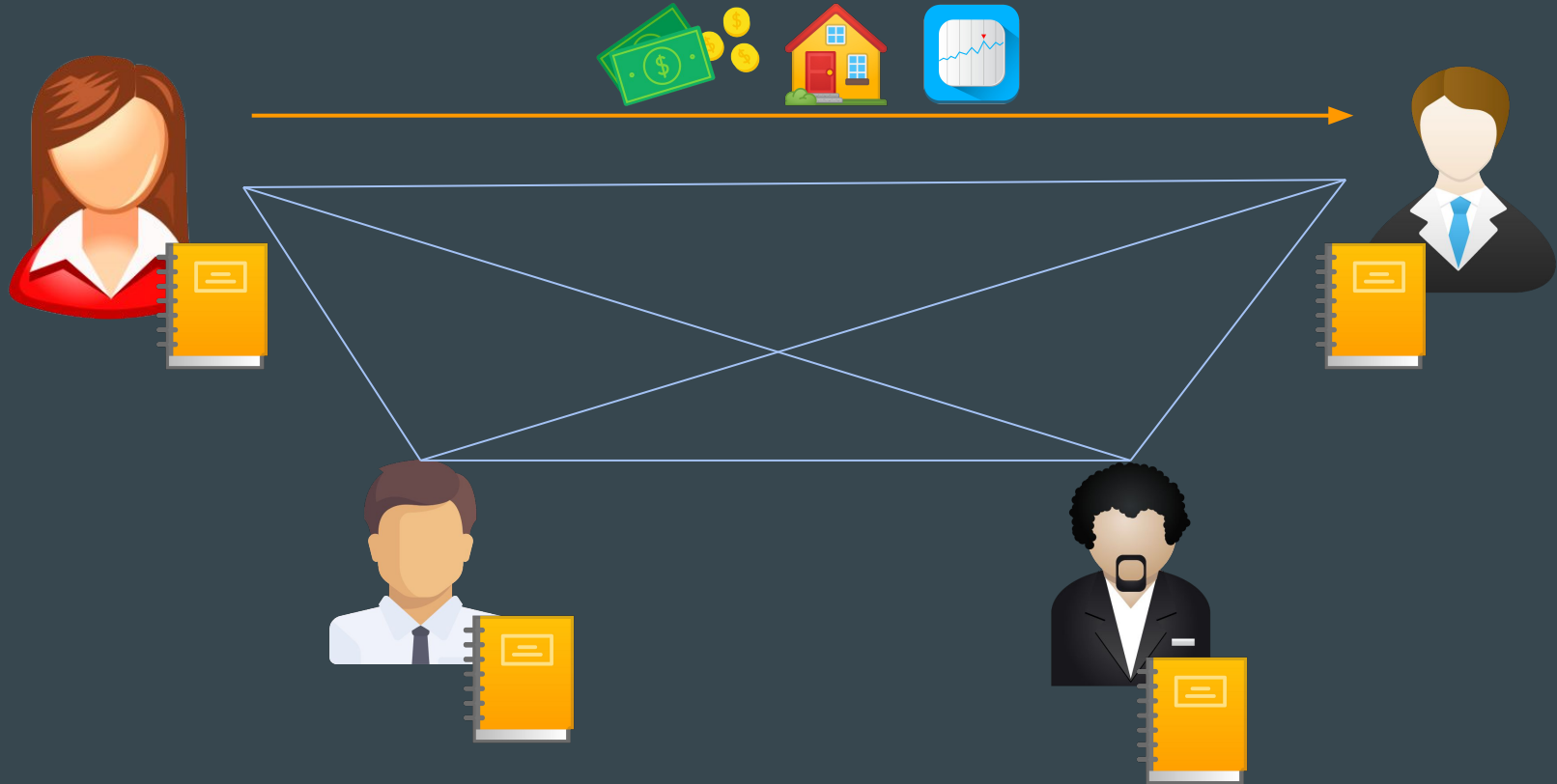
# 01 .meet blockchain - what is it ?



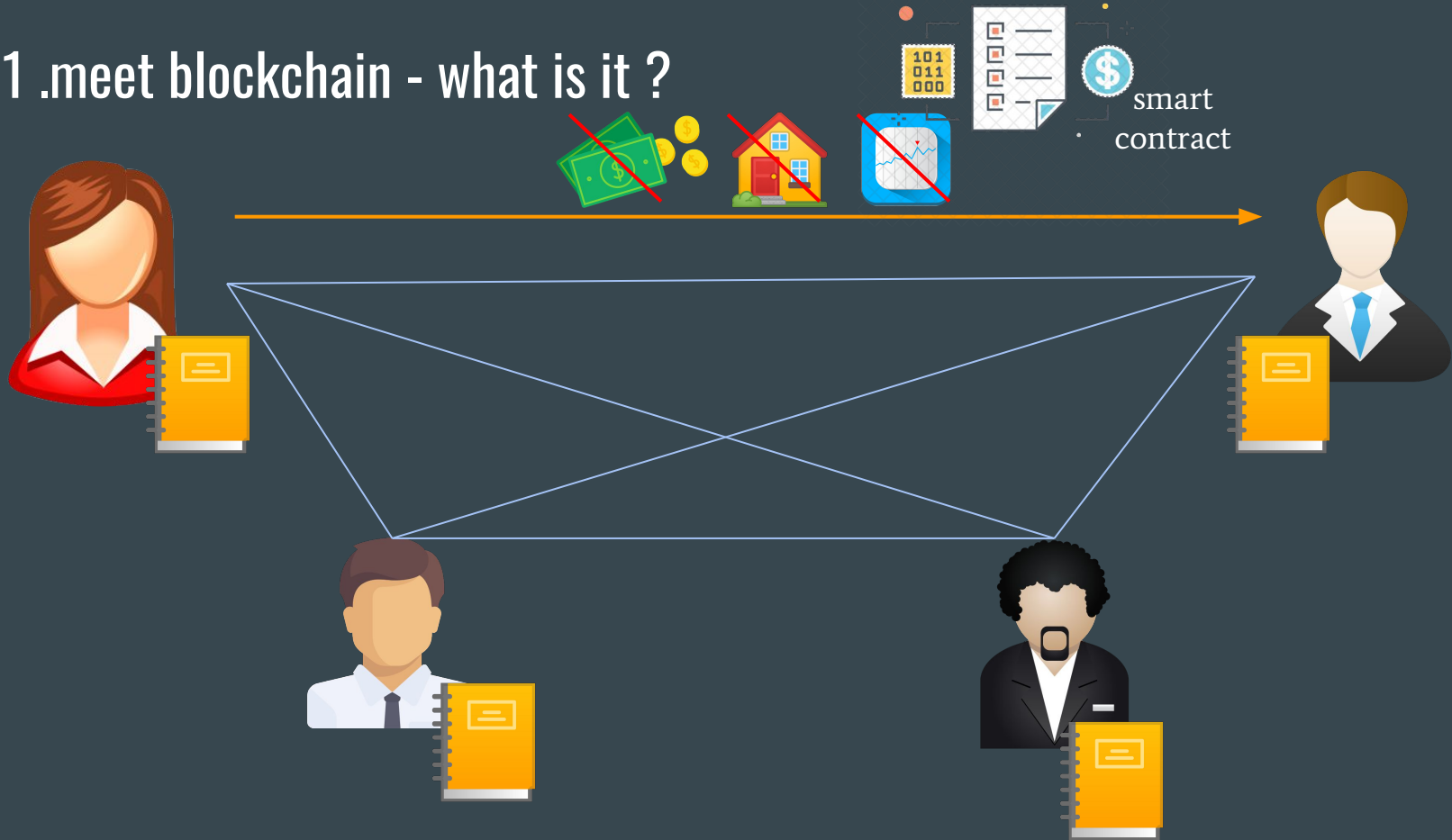
# 01 .meet blockchain - what is it ?



# 01 .meet blockchain - what is it ?



# 01 .meet blockchain - what is it ?



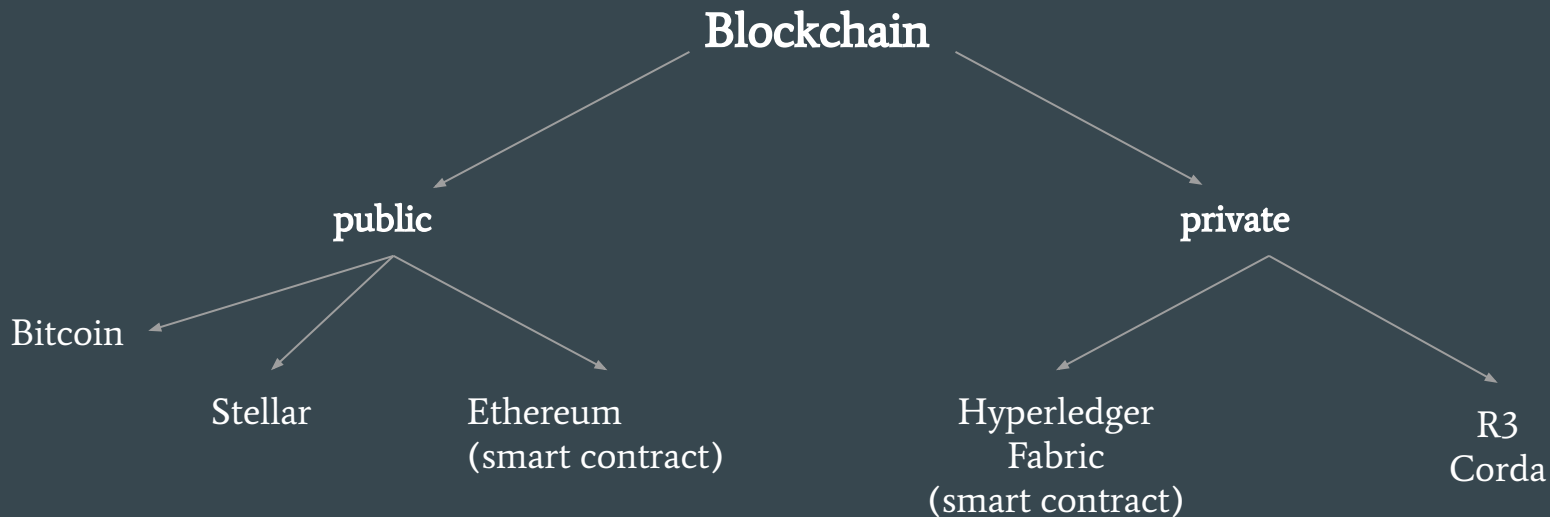
# 01 .meet blockchain - implementations

- nie tylko Bitcoin, dużo więcej



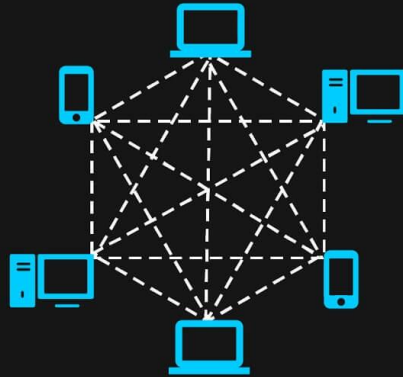
# 01 .meet blockchain - implementations *(simplified)*

- nie tylko Bitcoin, dużo więcej



# 01 .meet blockchain - public vs private

## Public vs Private Blockchain Network



**Public Blockchain: Permissionless**  
An open network system where all the devices can freely access without any kind of permission. The ledger is shared and transparent.



**Private Blockchain: Permissioned**  
A user has to be permitted by the blockchain authority before he/she could access the network. The user might join only if he/she gets an invitation.

# 01 .meet blockchain - public vs private

## public

- każdy może dołączyć
- duże sieci
- długo trwający consensus
- probabilistic consensus

## private

- na zaproszenie
- mniejsze sieci
- szybciej
- final consensus

01 .meet blockchain - part 1 END

## 02 .blockchain facts & doubts

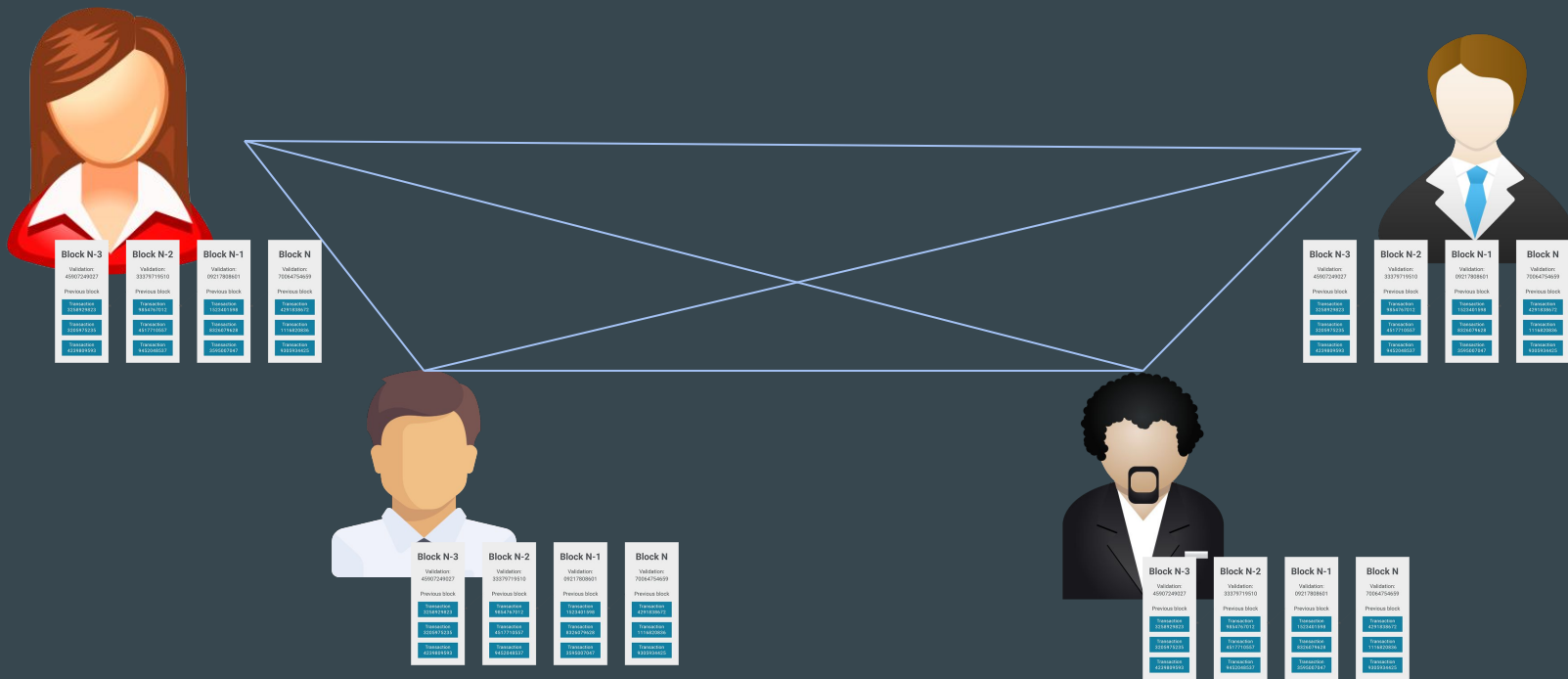
## 02 .blockchain facts & doubts - po pierwsze: immutable ledger

- po pierwsze niemutowalność, czyli LinkedList na sterydach
- cecha wspólna wszystkich BC
- append only jest fajne, bo nie możemy zrobić delete/update



## 02 .blockchain facts & doubts - po drugie: immutable & shared ledger

- każdy uczestnik posiada tą samą kopię danych



## 02 .blockchain facts & doubts - po trzecie: smart contract

- same dane to nie wszystko
- walidujemy pewne reguły biznesowe
- każdy może brać udział w procesie dodawania, walidowania transakcji





## 02 .blockchain facts & doubts - does anyone even need it?

- Czy potrzebujemy tych 3 rzeczy naraz ?
- “Buzzwordy” są fajne, bo są głośne.
  - Managerowie i biznes mają szansę zakumać coś z nowoczesnego IT
  - Nasza rola w tym to doświadczenie

## 02 .blockchain facts & doubts - does anyone even need it?

- Czy potrzebujemy tych 3 rzeczy naraz ?
- “Buzzwordy” są fajne, bo są głośne.
  - Managerowie i biznes mają szansę zakumać coś z nowoczesnego IT
  - Nasza rola w tym to doświadczenie
  - Poróbmy za konsultanta

## 02 .blockchain facts & doubts - does anyone even need it?

Case 1:

- “Potrzebuję głównie replikacji danych, raczej nie będą się zmieniać”
- “szukam rozwiązania tylko dla mojej, 1 organizacji”

## 02 .blockchain facts & doubts - does anyone even need it?

Case 1:

- “Potrzebuję głównie replikacji danych, raczej nie będą się zmieniać”
- “szukam rozwiązania tylko dla mojej, 1 organizacji”



## 02 .blockchain facts & doubts - does anyone even need it?

Case 2:

- “Dane mają być niemutowalne i replikowane”
- “Zależy mi na takich max sprzed tygodnia”
- “szukam rozwiązania tylko dla mojej, 1 organizacji”

## 02 .blockchain facts & doubts - does anyone even need it?

Case 2:

- “Dane mają być niemutowalne i replikowane”
- “Zależy mi na takich max sprzed tygodnia”
- “szukam rozwiązania tylko dla mojej, 1 organizacji”



## 02 .blockchain facts & doubts - does anyone even need it?

Case 3:

- “Dane mają być niemutowalne i replikowane.
- “szukam rozwiązania dzielonego między wieloma organizacjami”
- “ufamy sobie nawzajem”

## 02 .blockchain facts & doubts - does anyone even need it?

Case 3:

- “Dane mają być niemutowalne i replikowane.”
- “szukam rozwiązania dzielonego między wieloma organizacjami”
- “ufamy sobie nawzajem”





## 02 .blockchain facts & doubts - does anyone even need it?

TL;DR :

Blockchain ma sens tylko kiedy:

- jednostki/organizacje nie ufają sobie nawzajem i muszą dojść do porozumienia
- ryzyko przejęcia części sieci

a inaczej to...

- Immutable ledger == immutable data structures
- Shared ledger == replikacja
- Smart contract == walidacja

## 02 .blockchain facts & doubts - part 2 END

(distributed consensus approaching)

## 03 .distributed consensus

## 03 .distributed consensus != distributed consensus

- Konsensus konsensusowi nie równy
- W sumie co to jest consesus ? ...

## 03 .distributed consensus != distributed consensus

*“The consensus problem requires agreement among a number of nodes for a single data value. Some of the processes (agents) may fail or be unreliable in other ways, so consensus protocols must be fault tolerant or resilient.”*

## 03 .distributed consensus != distributed consensus

*“The consensus problem requires agreement among a number of nodes for a single data value. Some of the processes (agents) may fail or be unreliable in other ways, so consensus protocols must be fault tolerant or resilient.”*

[Wikipedia](#) :-D

## 03 .distributed consensus != distributed consensus

*“The consensus problem requires agreement among a number of nodes for a single data value. Some of the processes (agents) may fail or be unreliable in other ways, so consensus protocols must be fault tolerant or resilient.”*

[Wikipedia](#) :-D

- fault tolerant to duży temat
- różne typy konsensusu

## 03 .distributed consensus - Type 1: Crash fault tolerance

- *“In your datacenter, all the nodes are controlled by your organization (so they can hopefully be trusted).”*

Designing Data Intensive Applications  
Martin Kleppmann  
Chapter 8



## 03 .distributed consensus - Type 1: Crash fault tolerance

- *“In your datacenter, all the nodes are controlled by your organization (so they can hopefully be trusted).”*
- *“Weak forms of lying”*
- *“Nodes are faulty but, honest”*

Designing Data Intensive Applications,  
Martin Kleppmann  
Chapter 8

## 03 .distributed consensus - Type 1: Crash fault tolerance



Availability



Consistency

## 03 .distributed consensus - Type 1: Crash fault tolerance



ethereum



**HYPERLEDGER**



Availability



Consistency

## 03 .distributed consensus - Type 2: Byzantine fault tolerance

- Byzantine Generals Problem

“A system is Byzantine fault-tolerant if it continues to operate correctly even if some of the nodes are malfunctioning and not obeying the protocol, or if malicious attackers are interfering with the network”

## 03 .distributed consensus - Type 2: Byzantine fault tolerance

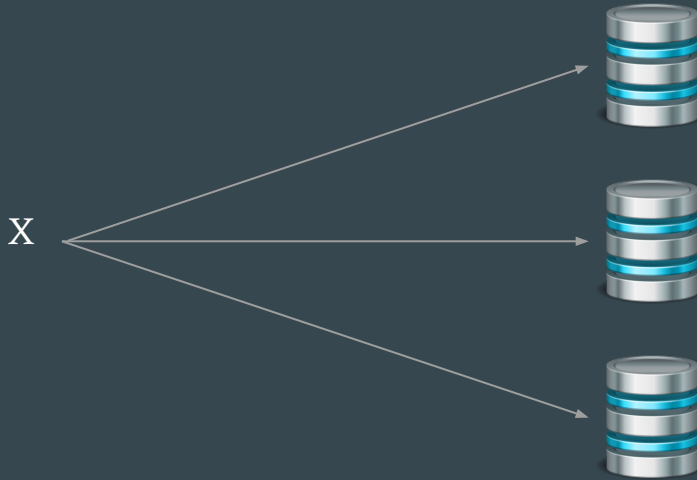
- *“The Byzantine Generals Problem”*

Leslie Lamport, Robert Shostak, and Marshall Pease, 1982

## 03 .distributed consensus - Type 2: Byzantine fault tolerance

- *“The Byzantine Generals Problem”*

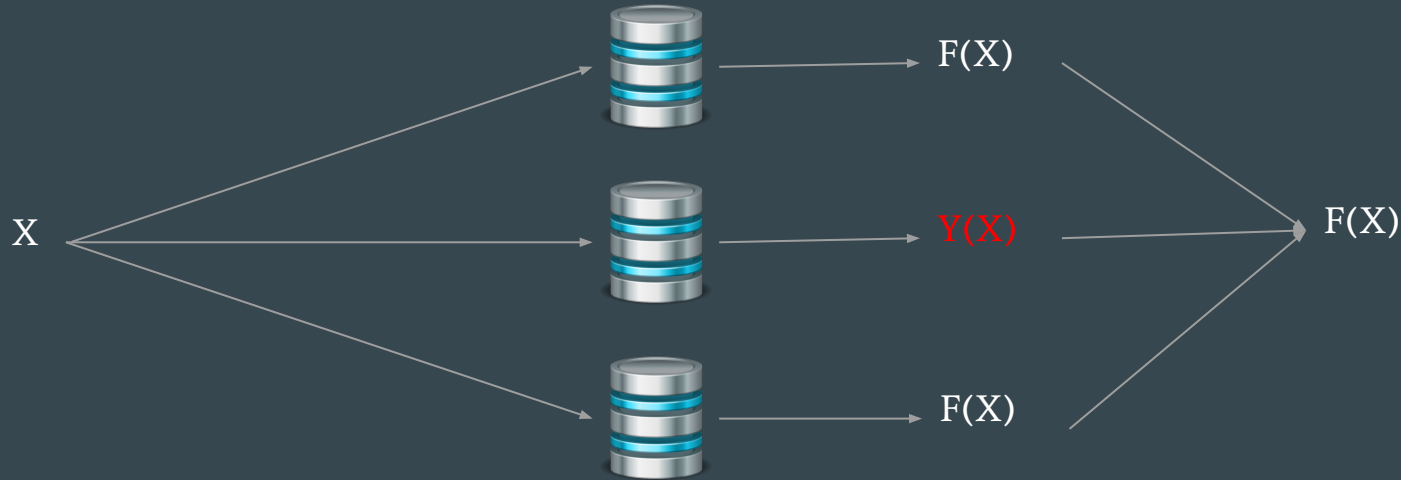
Leslie Lamport, Robert Shostak, and Marshall Pease, 1982



## 03 .distributed consensus - Type 2: Byzantine fault tolerance

- *“The Byzantine Generals Problem”*

Leslie Lamport, Robert Shostak, and Marshall Pease, 1982



## 03 .distributed consensus - Byzantine Faults: unleash the...

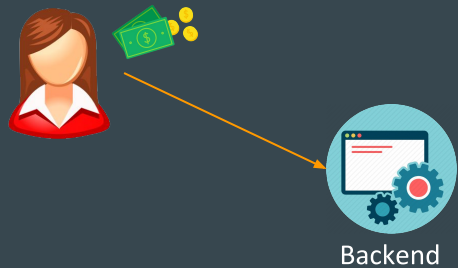


## 03 .distributed consensus - Byzantine Faults: unleash the... turtle

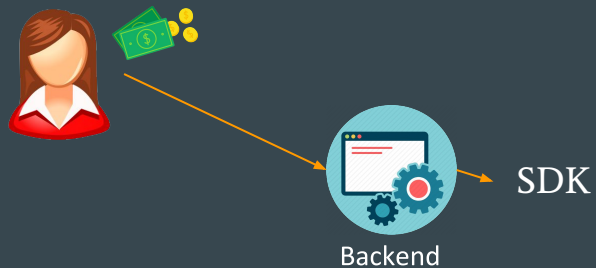


## 04 .hyperledger transaction flow

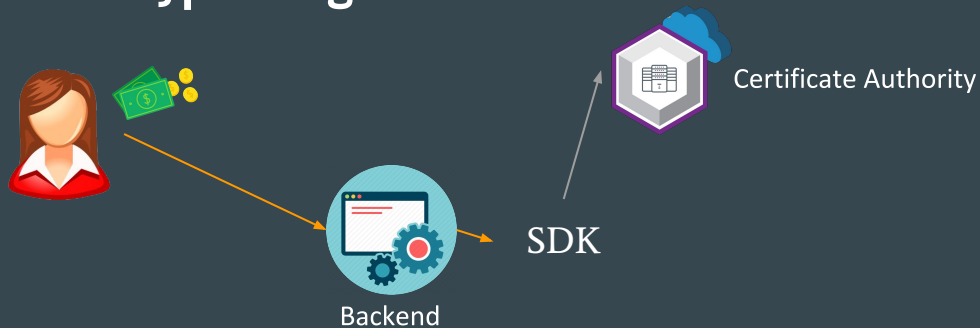
## 04 .hyperledger transaction flow



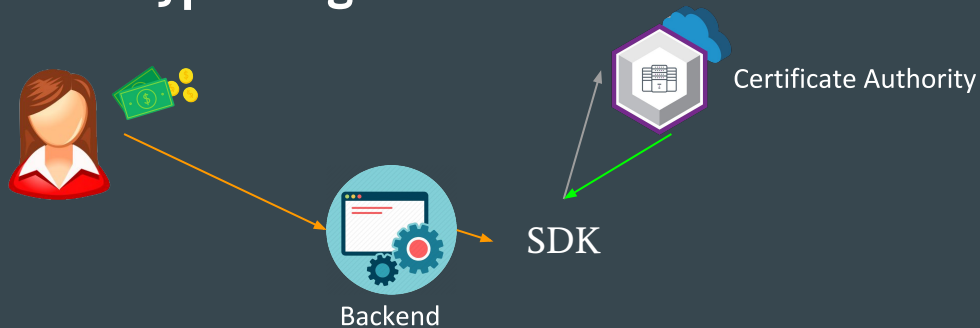
## 04 .hyperledger transaction flow



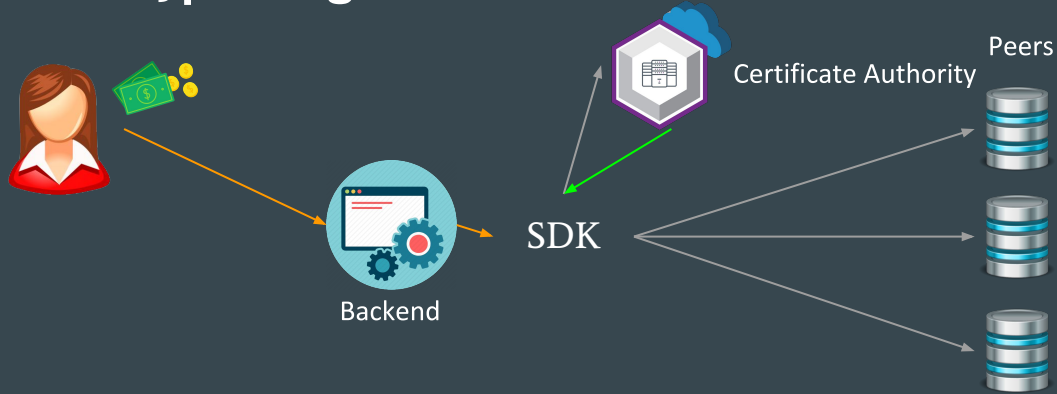
## 04 .hyperledger transaction flow: 1 - Auth



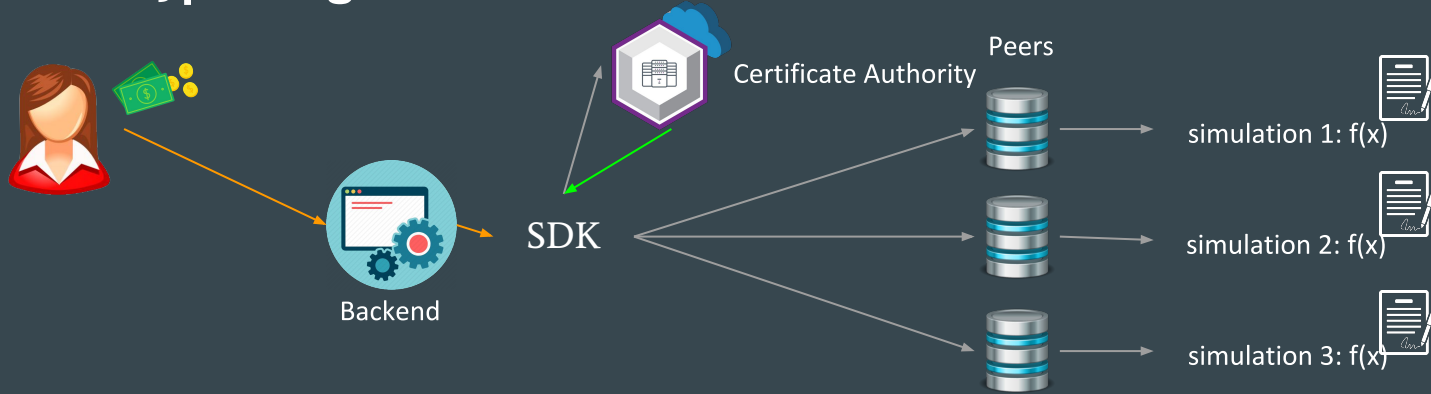
## 04 .hyperledger transaction flow: 1 - Auth



## 04 .hyperledger transaction flow: 2 - transaction simulation

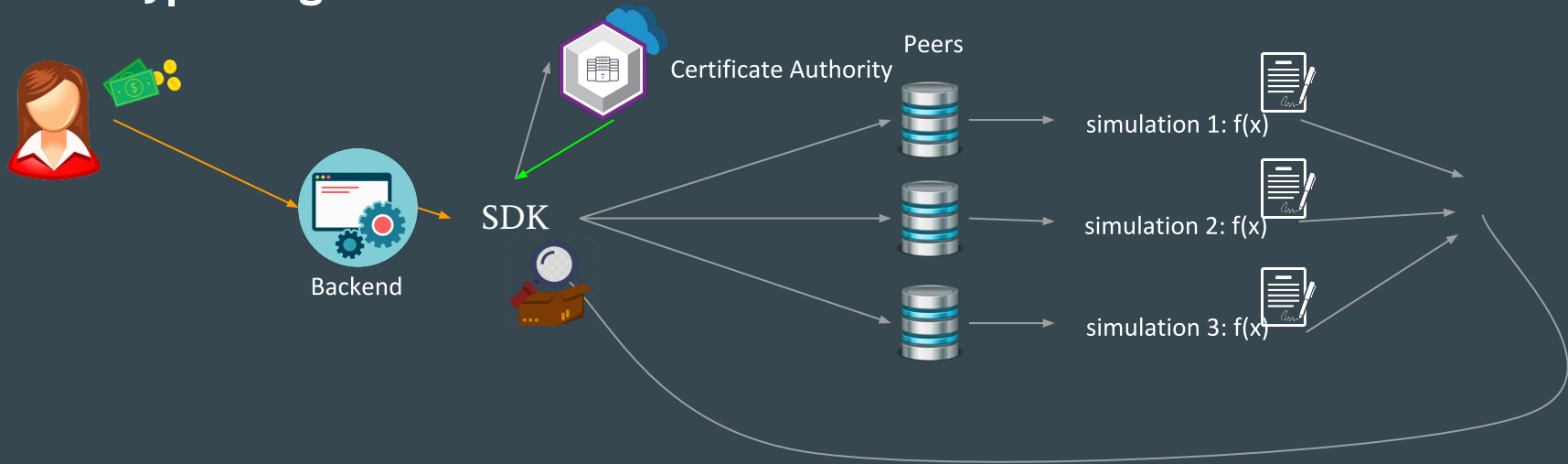


## 04 .hyperledger transaction flow: 2 - transaction simulation

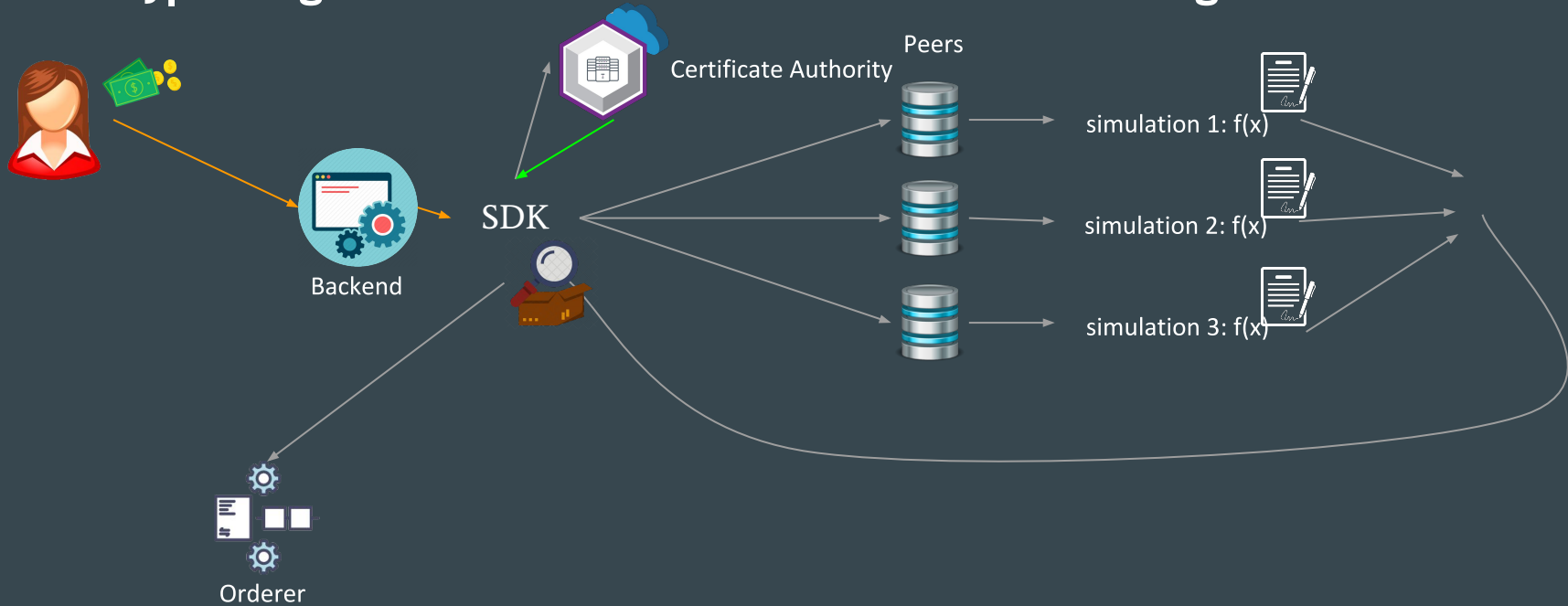




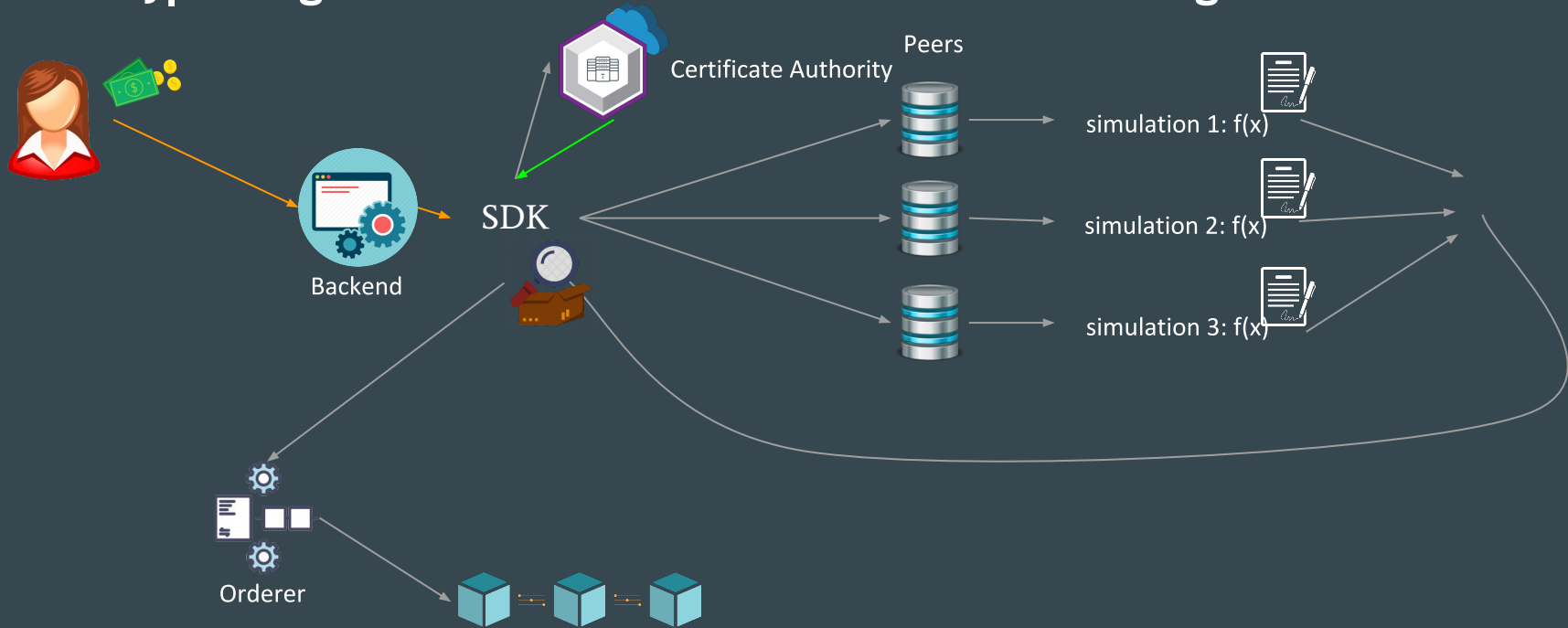
## 04 .hyperledger transaction flow: 2 - transaction simulation



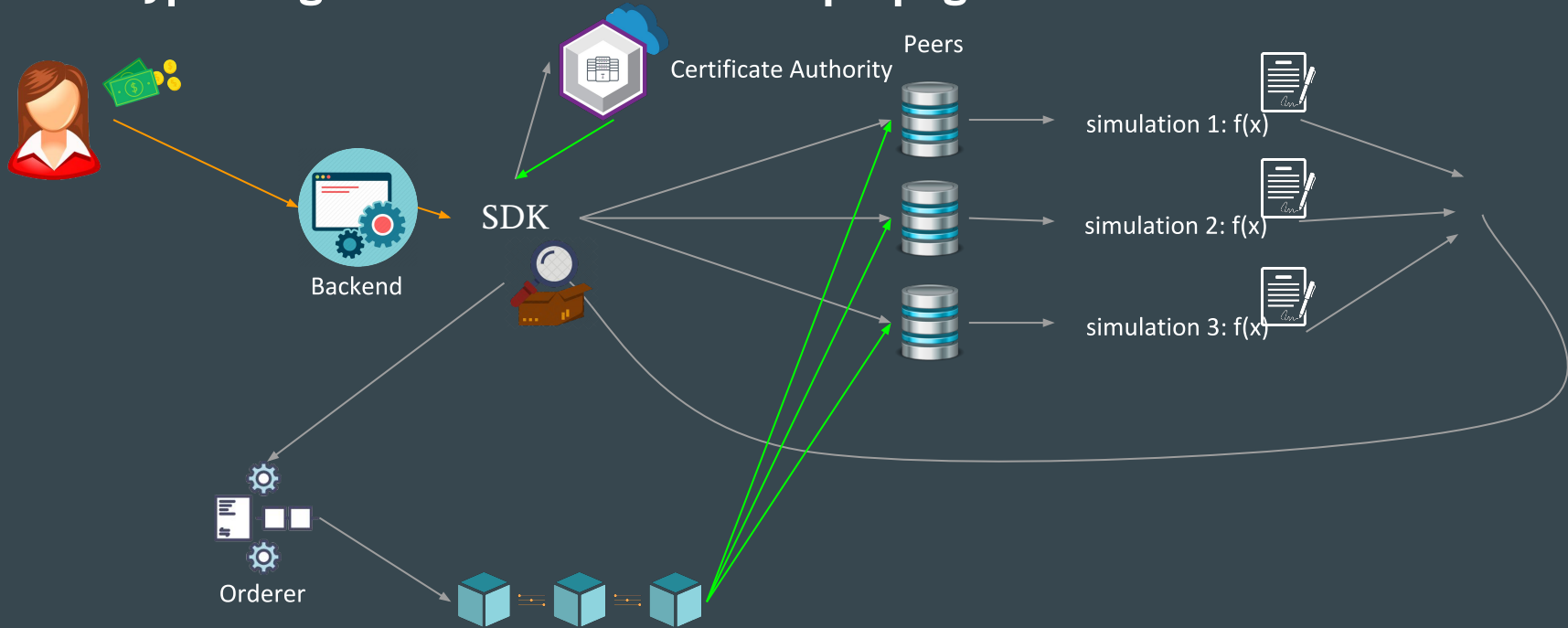
## 04 .hyperledger transaction flow: 3- simulation ordering



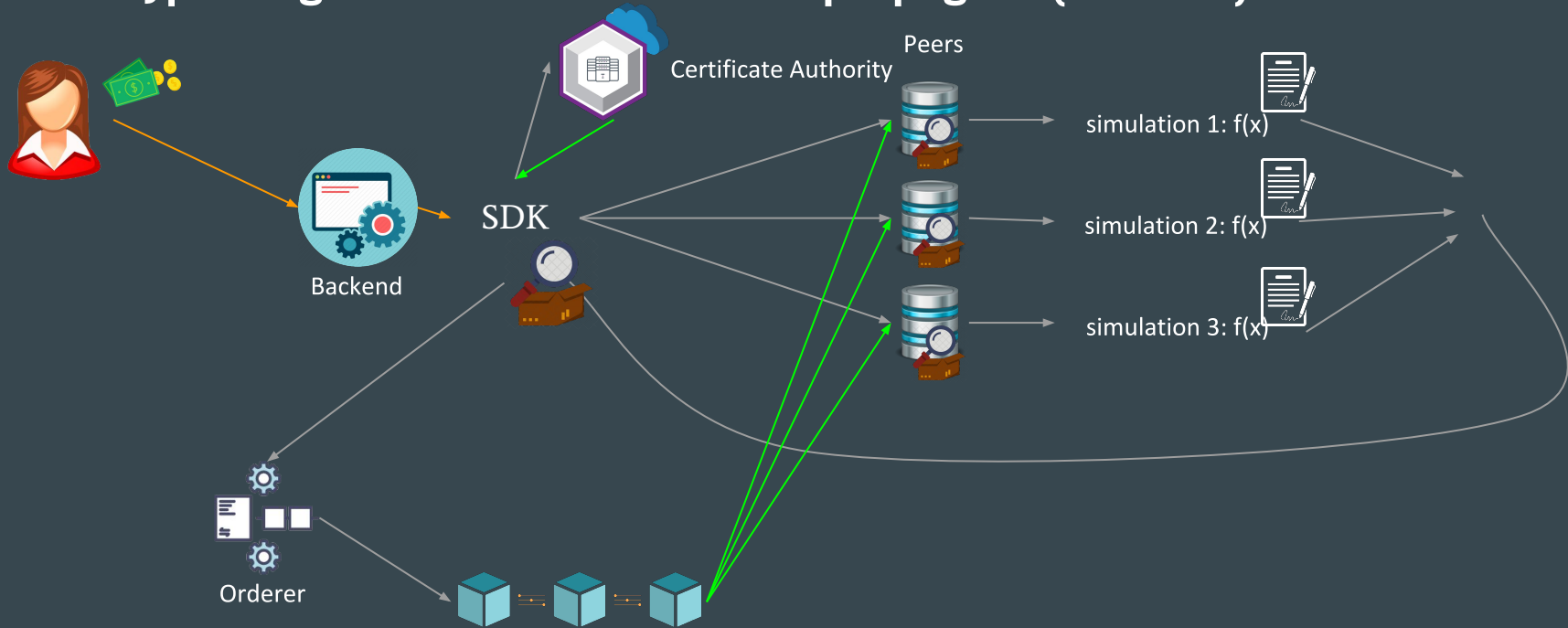
## 04 .hyperledger transaction flow: 3- simulation ordering



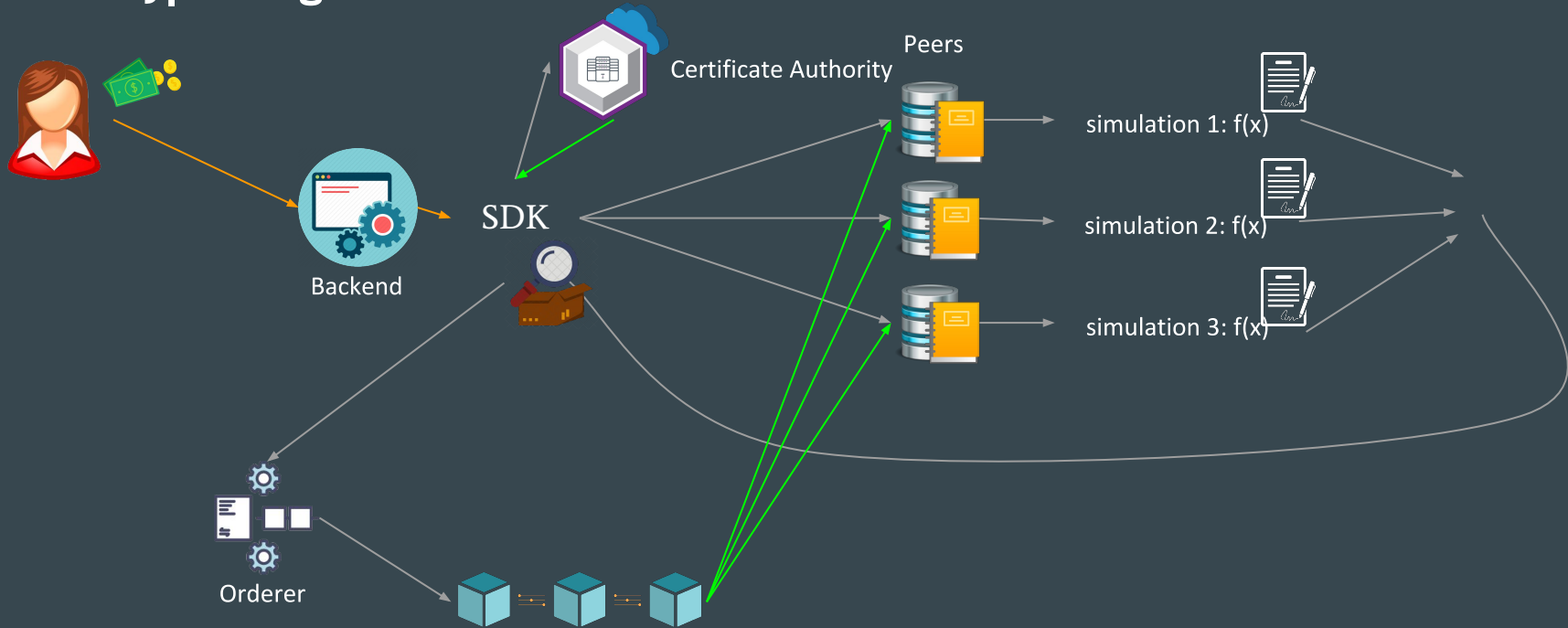
## 04 .hyperledger transaction flow: 4 - propagate



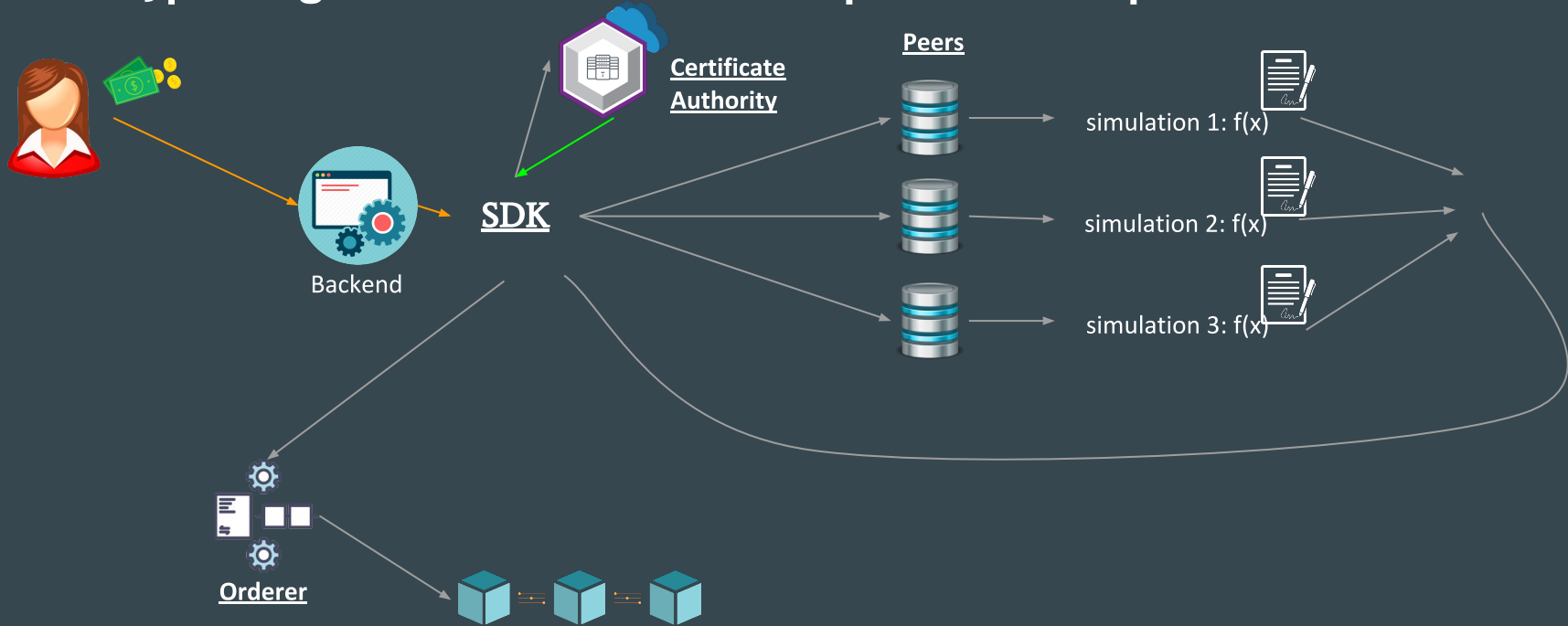
## 04 .hyperledger transaction flow: 4 - propagate (validate)



## 04 .hyperledger transaction flow: 5 - submitted !



## 04 .hyperledger transaction flow: Components Recap



**The End - Tak, zaraz koniec ;)**



# The End - Tak, zaraz koniec ;)

Recap:

- Poznaliśmy podstawy Blockchaina

# The End - Tak, zaraz koniec ;)

Recap:

- Poznaliśmy podstawy Blockchaina
- Wiemy kiedy go nie używać

# The End - Tak, zaraz koniec ;)

Recap:

- Poznaliśmy podstawy Blockchaina
- Wiemy kiedy go nie używać
- Porozmawialiśmy o Konsensusie

# The End - Tak, zaraz koniec ;)

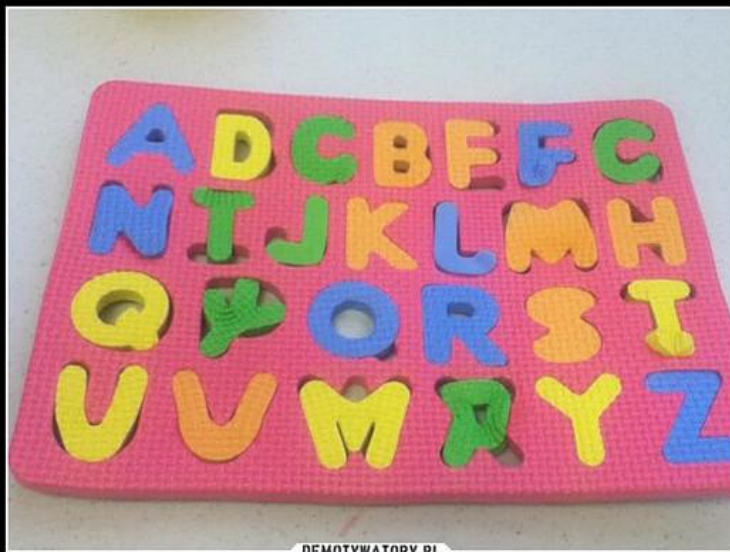
Recap:

- Poznaliśmy podstawy Blockchaina
- Wiemy kiedy go nie używać
- Porozmawialiśmy o Konsensusie
- Zobaczyliśmy Hyperledgera

.the end



.the end



Jasio nie jest zbyt mądry,  
ale za to jest silny

.birthday pizza !!! --->

