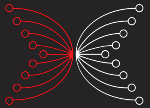
# Smart Contracts

## Lightning

Lars Brünjes



INPUT | OUTPUT

January 9 2020

# Introduction

# Problem

- Problem
  - In current cryptocurrencies, each transaction has to be stored on each node.

# Problem

- Problem
  - In current cryptocurrencies, each transaction has to be stored on each node.
  - The consensus algorithm restricts the maximum rate at which new blocks can be created, and blocks are furthermore subject to size restrictions.

# Problem

- Problem
  - In current cryptocurrencies, each transaction has to be stored on each node.
  - The consensus algorithm restricts the maximum rate at which new blocks can be created, and blocks are furthermore subject to size restrictions.
  - This leads to rather modest transaction rates in comparison to traditional, centralized payment networks like Visa.

# Problem

- Problem
  - In current cryptocurrencies, each transaction has to be stored on each node.
  - The consensus algorithm restricts the maximum rate at which new blocks can be created, and blocks are furthermore subject to size restrictions.
  - This leads to rather modest transaction rates in comparison to traditional, centralized payment networks like Visa.
- Lightning
  - The Lightning-Network is one way to solve this problem.

# Problem

- Problem
  - In current cryptocurrencies, each transaction has to be stored on each node.
  - The consensus algorithm restricts the maximum rate at which new blocks can be created, and blocks are furthermore subject to size restrictions.
  - This leads to rather modest transaction rates in comparison to traditional, centralized payment networks like Visa.
- Lightning
  - The Lightning-Network is one way to solve this problem.
  - It is initially planned for Bitcoin, but its idea is generic enough to apply to many other cryptocurrencies (including Cardano).

# Problem

- Problem
  - In current cryptocurrencies, each transaction has to be stored on each node.
  - The consensus algorithm restricts the maximum rate at which new blocks can be created, and blocks are furthermore subject to size restrictions.
  - This leads to rather modest transaction rates in comparison to traditional, centralized payment networks like Visa.
- Lightning
  - The Lightning-Network is one way to solve this problem.
  - It is initially planned for Bitcoin, but its idea is generic enough to apply to many other cryptocurrencies (including Cardano).
  - The basic idea is to offload work from nodes by creating parallel side channels, which can process the bulk of all transactions.

# Idea

- The basic idea is for two parties who want to exchange funds to create a payment channel between themselves, which is independent of the Bitcoin network.

# Idea

- The basic idea is for two parties who want to exchange funds to create a payment channel between themselves, which is independent of the Bitcoin network.
- This channel can then process arbitrarily many transactions in a very fast and cheap manner.

# Idea

- The basic idea is for two parties who want to exchange funds to create a payment channel between themselves, which is independent of the Bitcoin network.
- This channel can then process arbitrarily many transactions in a very fast and cheap manner.
- In the normal case, when both parties play by the rules, only *two* transactions will ever have to be sent to the Bitcoin network, one to open the channel, one to close it again when it is no longer needed.

# Idea

- The basic idea is for two parties who want to exchange funds to create a payment channel between themselves, which is independent of the Bitcoin network.
- This channel can then process arbitrarily many transactions in a very fast and cheap manner.
- In the normal case, when both parties play by the rules, only *two* transactions will ever have to be sent to the Bitcoin network, one to open the channel, one to close it again when it is no longer needed.
- In spite of this, all payments using the channel are secure and are guaranteed by the blockchain.

# Idea

- The basic idea is for two parties who want to exchange funds to create a payment channel between themselves, which is independent of the Bitcoin network.
- This channel can then process arbitrarily many transactions in a very fast and cheap manner.
- In the normal case, when both parties play by the rules, only *two* transactions will ever have to be sent to the Bitcoin network, one to open the channel, one to close it again when it is no longer needed.
- In spite of this, all payments using the channel are secure and are guaranteed by the blockchain.
- After we will have understood how such a channel between two parties works, we will see how that system can be extended to allow payments between parties who do not possess a direct channel between each other.

# Direct Payment Channels

# Opening a Payment Channel

- Let us assume that Alice and Bob want to open a payment channel between each other, because they frequently exchange Bitcoin.

# Opening a Payment Channel

- Let us assume that Alice and Bob want to open a payment channel between each other, because they frequently exchange Bitcoin.
- The open the channel by each depositing a certain amount of Bitcoin on the blockchain (for example 5 ฿ each). (If Alice expects that in future, she will send more money to Bob than she will receive from him, she can also deposit more than he does.)
- They proceed as follows:

# Opening a Payment Channel

- Let us assume that Alice and Bob want to open a payment channel between each other, because they frequently exchange Bitcoin.
- The open the channel by each depositing a certain amount of Bitcoin on the blockchain (for example 5 ฿ each). (If Alice expects that in future, she will send more money to Bob than she will receive from him, she can also deposit more than he does.)
- They proceed as follows:
  1. They create a 2-of-2-multisig-address and prepare a transaction, which will send their deposit to this address, but they do not yet sign that transaction.

# Opening a Payment Channel

- Let us assume that Alice and Bob want to open a payment channel between each other, because they frequently exchange Bitcoin.
- The open the channel by each depositing a certain amount of Bitcoin on the blockchain (for example 5 ₿ each). (If Alice expects that in future, she will send more money to Bob than she will receive from him, she can also deposit more than he does.)
- They proceed as follows:
  1. They create a 2-of-2-multisig-address and prepare a transaction, which will send their deposit to this address, but they do not yet sign that transaction.
  2. Each picks a random number, a secret, and sends its hash to the other.

# Opening a Payment Channel

- Let us assume that Alice and Bob want to open a payment channel between each other, because they frequently exchange Bitcoin.
- The open the channel by each depositing a certain amount of Bitcoin on the blockchain (for example 5 ₿ each). (If Alice expects that in future, she will send more money to Bob than she will receive from him, she can also deposit more than he does.)
- They proceed as follows:
  1. They create a 2-of-2-multisig-address and prepare a transaction, which will send their deposit to this address, but they do not yet sign that transaction.
  2. Each picks a random number, a secret, and sends its hash to the other.
  3. Alice creates and signs a new transaction, which has the multisig-address as input and two outputs, her deposit to herself, Bob's deposit to a new multisig-address, which can either be unlocked by Bob after 1000 blocks or immediately by Alice, if she knows Bob's secret.
  4. Bob creates and signs a new transaction, which has the multisig-address as input and two outputs, his deposit to himself, Alice's deposit to a new multisig-address, which can either be unlocked by Alice after 1000 blocks or immediately by Bob, if he knows Alice's secret.

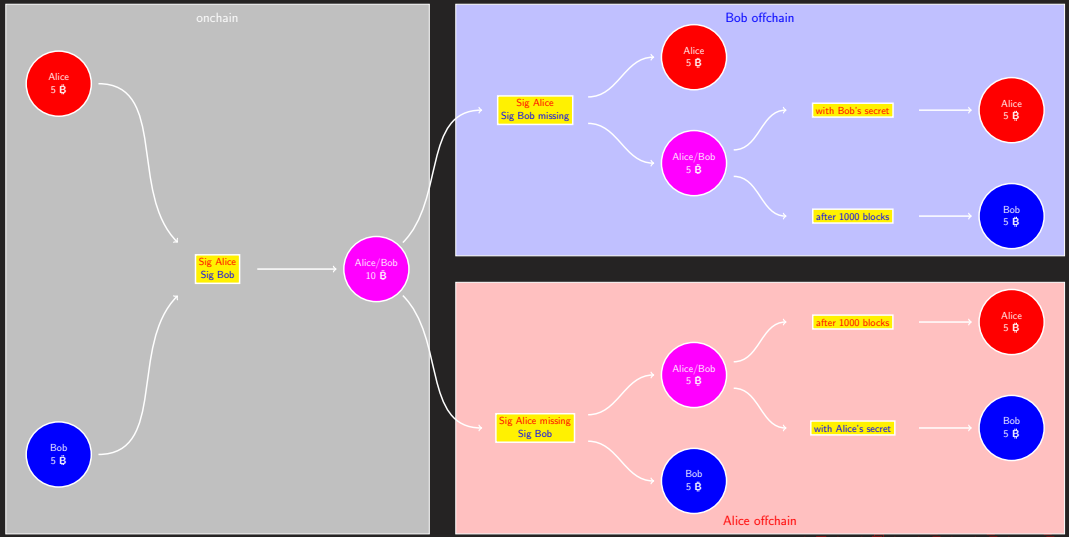
# Opening a Payment Channel

- Let us assume that Alice and Bob want to open a payment channel between each other, because they frequently exchange Bitcoin.
- The open the channel by each depositing a certain amount of Bitcoin on the blockchain (for example 5 ₿ each). (If Alice expects that in future, she will send more money to Bob than she will receive from him, she can also deposit more than he does.)
- They proceed as follows:
  1. They create a 2-of-2-multisig-address and prepare a transaction, which will send their deposit to this address, but they do not yet sign that transaction.
  2. Each picks a random number, a secret, and sends its hash to the other.
  3. Alice creates and signs a new transaction, which has the multisig-address as input and two outputs, her deposit to herself, Bob's deposit to a new multisig-address, which can either be unlocked by Bob after 1000 blocks or immediately by Alice, if she knows Bob's secret.
  4. Bob creates and signs a new transaction, which has the multisig-address as input and two outputs, his deposit to himself, Alice's deposit to a new multisig-address, which can either be unlocked by Alice after 1000 blocks or immediately by Bob, if he knows Alice's secret.
  5. Alice and Bob exchange the two new transactions from steps 3 Und 4.

# Opening a Payment Channel

- Let us assume that Alice and Bob want to open a payment channel between each other, because they frequently exchange Bitcoin.
- The open the channel by each depositing a certain amount of Bitcoin on the blockchain (for example 5 ₿ each). (If Alice expects that in future, she will send more money to Bob than she will receive from him, she can also deposit more than he does.)
- They proceed as follows:
  1. They create a 2-of-2-multisig-address and prepare a transaction, which will send their deposit to this address, but they do not yet sign that transaction.
  2. Each picks a random number, a secret, and sends its hash to the other.
  3. Alice creates and signs a new transaction, which has the multisig-address as input and two outputs, her deposit to herself, Bob's deposit to a new multisig-address, which can either be unlocked by Bob after 1000 blocks or immediately by Alice, if she knows Bob's secret.
  4. Bob creates and signs a new transaction, which has the multisig-address as input and two outputs, his deposit to himself, Alice's deposit to a new multisig-address, which can either be unlocked by Alice after 1000 blocks or immediately by Bob, if he knows Alice's secret.
  5. Alice and Bob exchange the two new transactions from steps 3 Und 4.
  6. Finally, Alice and Bob sign the transaction from step-1 and send it to the blockchain.

# Illustration



# Explanation

- This complicated arrangement guarantees that Alice and Bob will be able to retrieve their deposits, no matter what.

# Explanation

- This complicated arrangement guarantees that Alice and Bob will be able to retrieve their deposits, no matter what.
- Under normal circumstances — when both of them agree — they can send a simple 2-of-2-multisig-transaction to the blockchain, which is signed by both of them and returns their deposits to them.

# Explanation

- This complicated arrangement guarantees that Alice and Bob will be able to retrieve their deposits, no matter what.
- Under normal circumstances — when both of them agree — they can send a simple 2-of-2-multisig-transaction to the blockchain, which is signed by both of them and returns their deposits to them.

- If Alice wants to retrieve her deposit without Bob's help, she signs the transaction she received from Bob and sends it to the blockchain.
  - Bob gets his deposit immediately.
  - Alice must wait for 1000 blocks until she gets her deposit.
  - If Bob manages to learn Alice's secret in the meantime, he can get his hands on Alice's deposit before the 1000 blocks are over. (We will see later what this is good for.)

# Explanation

- This complicated arrangement guarantees that Alice and Bob will be able to retrieve their deposits, no matter what.
- Under normal circumstances — when both of them agree — they can send a simple 2-of-2-multisig-transaction to the blockchain, which is signed by both of them and returns their deposits to them.

- If Bob wants to retrieve his deposit without Alice's help, he signs the transaction he received from Alice and sends it to the blockchain.
  - Alice gets her deposit immediately.
  - Bob must wait for 1000 blocks until he gets his deposit.
  - If Alice manages to learn Bob's secret in the meantime, she can get her hands on Bob's deposit before the 1000 blocks are over. (We will see later what this is good for.)

# Using a Payment Channel

- After all this effort, Alice and Bob want to use their shining new payment channel to send Bitcoin!

# Using a Payment Channel

- After all this effort, Alice and Bob want to use their shining new payment channel to send Bitcoin!
- If Alice wants to send 1 ₿ to Bob, they proceed as follows:

# Using a Payment Channel

- After all this effort, Alice and Bob want to use their shining new payment channel to send Bitcoin!
- If Alice wants to send 1 ₿ to Bob, they proceed as follows:
  1. Each chooses a new secret and sends its hash to the other.

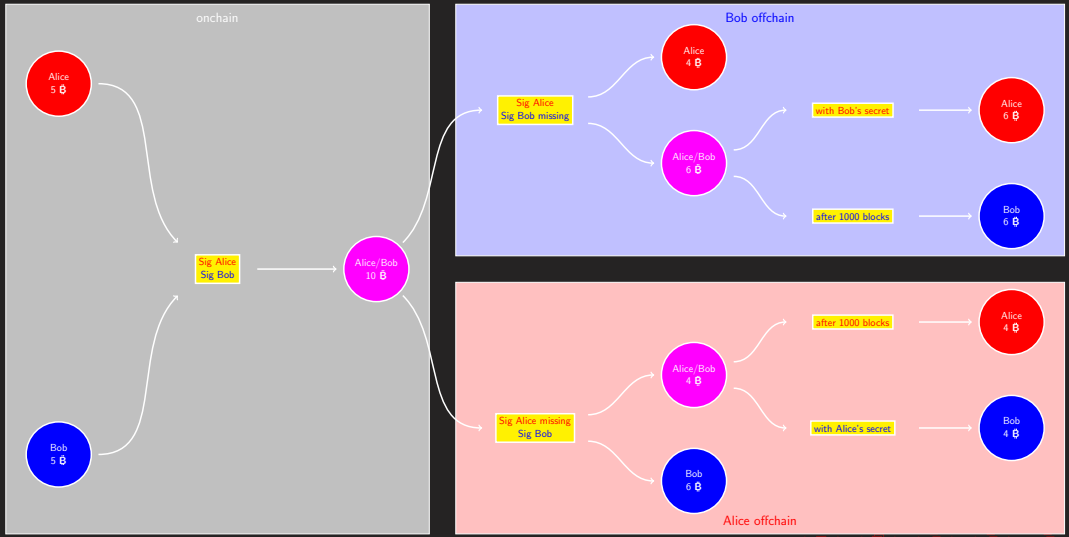
# Using a Payment Channel

- After all this effort, Alice and Bob want to use their shining new payment channel to send Bitcoin!
- If Alice wants to send 1 ₿ to Bob, they proceed as follows:
  1. Each chooses a new secret and sends its hash to the other.
  2. Alice creates, signs and sends Bob a new transaction with the multisig-address as input and two outputs — 4 ₿ to herself, 6 ₿ to a new multisig-address, which can be unlocked either after 1000 blocks by Bob or immediately by Alice, provided she has learned Bob's new secret.
  3. Bob creates, signs and sends Alice a new transaction with the multisig-address as input and two outputs — 6 ₿ to himself, 4 ₿ to a new multisig-address, which can be unlocked either after 1000 blocks by Alice or immediately by Bob, provided he has learned Alice's new secret.

# Using a Payment Channel

- After all this effort, Alice and Bob want to use their shining new payment channel to send Bitcoin!
- If Alice wants to send 1 ₿ to Bob, they proceed as follows:
  1. Each chooses a new secret and sends its hash to the other.
  2. Alice creates, signs and sends Bob a new transaction with the multisig-address as input and two outputs — 4 ₿ to herself, 6 ₿ to a new multisig-address, which can be unlocked either after 1000 blocks by Bob or immediately by Alice, provided she has learned Bob's new secret.
  3. Bob creates, signs and sends Alice a new transaction with the multisig-address as input and two outputs — 6 ₿ to himself, 4 ₿ to a new multisig-address, which can be unlocked either after 1000 blocks by Alice or immediately by Bob, provided he has learned Alice's new secret.
  4. Alice and Bob exchange their old secrets.

# Illustration



# Explanation

- Under normal circumstance, if both agree, they can later send a common 2-of-2-multisig-transaction to the blockchain which is signed by both and gives 4 Ƀ to Alice and 6 Ƀ to Bob.

# Explanation

- Under normal circumstance, if both agree, they can later send a common 2-of-2-multisig-transaction to the blockchain which is signed by both and gives 4 Ƀ to Alice and 6 Ƀ to Bob.

- As before, Alice can retrieve her 4 Ƀ without Bob's help by sending the new transaction she received from Bob to the blockchain.
- If she instead tries to use Bob's old transaction, she will have to wait for 1000 blocks for her 5 Ƀ. But Bob knows her old secret now and can get all the money for himself in this case. This means that Bob's old transaction is now worthless for Alice.

# Explanation

- Under normal circumstance, if both agree, they can later send a common 2-of-2-multisig-transaction to the blockchain which is signed by both and gives 4 ฿ to Alice and 6 ฿ to Bob.

- As before, Bob can retrieve his 6 ฿ without Alice's help by sending the new transaction he received from Alice to the blockchain.
- He has no interest in using Alice's old transaction, because that one only gives him 5 ฿ instead of 6 ฿. In addition to that, Alice could get all the money in this case, because she knows Bob's old secret now. Alice's old transaction is therefore worthless for Bob.

# Closing the Payment Channel

- In this manner Alice and Bob can send arbitrarily many transactions to each other (as long as the balance does not exceed the original deposit made to the blockchain).

# Closing the Payment Channel

- In this manner Alice and Bob can send arbitrarily many transactions to each other (as long as the balance does not exceed the original deposit made to the blockchain).
- As long as both play by the rules, no further transaction is visible on the blockchain. Communication between Alice and Bob is parallel to the blockchain, is "lightning fast" and (as good as) for free. Payments are therefore much faster and cheaper than normal Bitcoin transactions.

# Closing the Payment Channel

- In this manner Alice and Bob can send arbitrarily many transactions to each other (as long as the balance does not exceed the original deposit made to the blockchain).
- As long as both play by the rules, no further transaction is visible on the blockchain. Communication between Alice and Bob is parallel to the blockchain, is "lightning fast" and (as good as) for free. Payments are therefore much faster and cheaper than normal Bitcoin transactions.
- At any point in time, Alice's and Bob's money is safe. They can send the other's most current transaction to the blockchain at any time to retrieve their money (after 1000 blocks).

# Closing the Payment Channel

- In this manner Alice and Bob can send arbitrarily many transactions to each other (as long as the balance does not exceed the original deposit made to the blockchain).
- As long as both play by the rules, no further transaction is visible on the blockchain. Communication between Alice and Bob is parallel to the blockchain, is "lightning fast" and (as good as) for free. Payments are therefore much faster and cheaper than normal Bitcoin transactions.
- At any point in time, Alice's and Bob's money is safe. They can send the other's most current transaction to the blockchain at any time to retrieve their money (after 1000 blocks).
- If both agree to close the channel, they can do so using a common 2-of-2-multisig transaction. This means that under normal circumstances, only two "real" Bitcoin transactions are needed, one to open the channel and one to close it in the end.

# Indirect Payments

# Payments Without a Direct Channel

- Lightning also allows payments between parties who don't own a direct payment channel between them.

# Payments Without a Direct Channel

- Lightning also allows payments between parties who don't own a direct payment channel between them.
- Let us assume that Alice wants to send 1 ₿ to Charlie without first establishing a payment channel with him. Let us further assume that Bob and Charlie possess a payment channel between each other.

# Payments Without a Direct Channel

- Lightning also allows payments between parties who don't own a direct payment channel between them.
- Let us assume that Alice wants to send 1 ₿ to Charlie without first establishing a payment channel with him. Let us further assume that Bob and Charlie possess a payment channel between each other.
- The idea is to use the two channels between Alice and Bob and Bob and Charlie to enable Alice to pay Charlie.

# Payments Without a Direct Channel

- Lightning also allows payments between parties who don't own a direct payment channel between them.
- Let us assume that Alice wants to send 1 ₿ to Charlie without first establishing a payment channel with him. Let us further assume that Bob and Charlie possess a payment channel between each other.
- The idea is to use the two channels between Alice and Bob and Bob and Charlie to enable Alice to pay Charlie.
- Alice, Bob and Charlie proceed as follows:

# Payments Without a Direct Channel

- Lightning also allows payments between parties who don't own a direct payment channel between them.
- Let us assume that Alice wants to send 1 ₿ to Charlie without first establishing a payment channel with him. Let us further assume that Bob and Charlie possess a payment channel between each other.
- The idea is to use the two channels between Alice and Bob and Bob and Charlie to enable Alice to pay Charlie.
- Alice, Bob and Charlie proceed as follows:
  1. Alice asks Charlie to create a secret X and send her its hash.

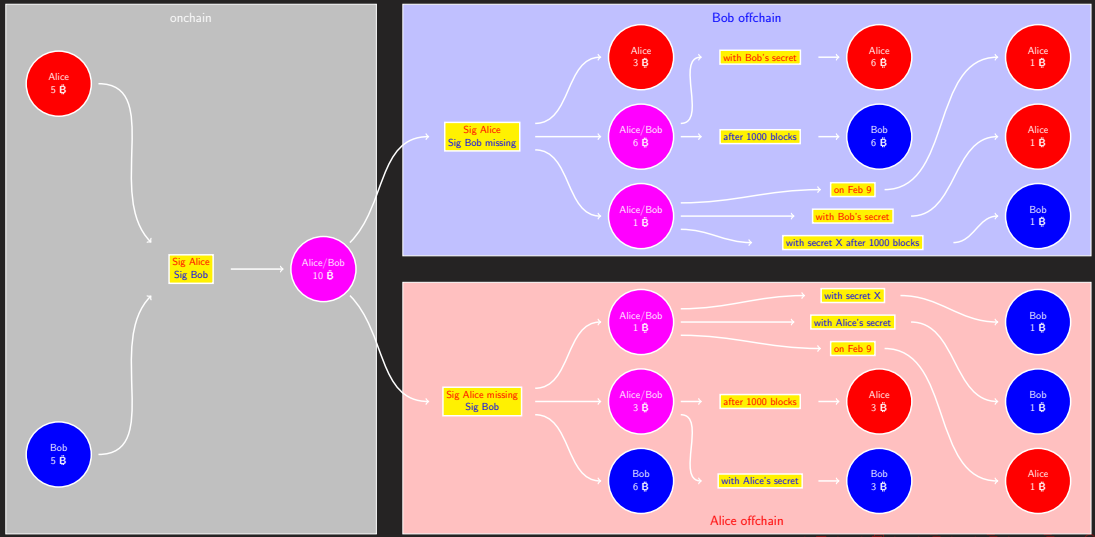
# Payments Without a Direct Channel

- Lightning also allows payments between parties who don't own a direct payment channel between them.
- Let us assume that Alice wants to send 1 ₿ to Charlie without first establishing a payment channel with him. Let us further assume that Bob and Charlie possess a payment channel between each other.
- The idea is to use the two channels between Alice and Bob and Bob and Charlie to enable Alice to pay Charlie.
- Alice, Bob and Charlie proceed as follows:
  1. Alice asks Charlie to create a secret X and send her its hash.
  2. Bob uses his channel to Charlie to pay Charlie 1 ₿ in exchange for secret X.

# Payments Without a Direct Channel

- Lightning also allows payments between parties who don't own a direct payment channel between them.
- Let us assume that Alice wants to send 1 ₿ to Charlie without first establishing a payment channel with him. Let us further assume that Bob and Charlie possess a payment channel between each other.
- The idea is to use the two channels between Alice and Bob and Bob and Charlie to enable Alice to pay Charlie.
- Alice, Bob and Charlie proceed as follows:
  1. Alice asks Charlie to create a secret X and send her its hash.
  2. Bob uses his channel to Charlie to pay Charlie 1 ₿ in exchange for secret X.
  3. Alice uses her channel to Bob to pay Bob 1 ₿ in exchange for secret X.
  4. Similar to direct payments, steps 2 and 3 will use special Hash Time-Locked Contracts (HTLCs), which will make use of absolute time locks instead of relative ones.

# Hash Time-Locked Contracts — Channel between Alice and Bob



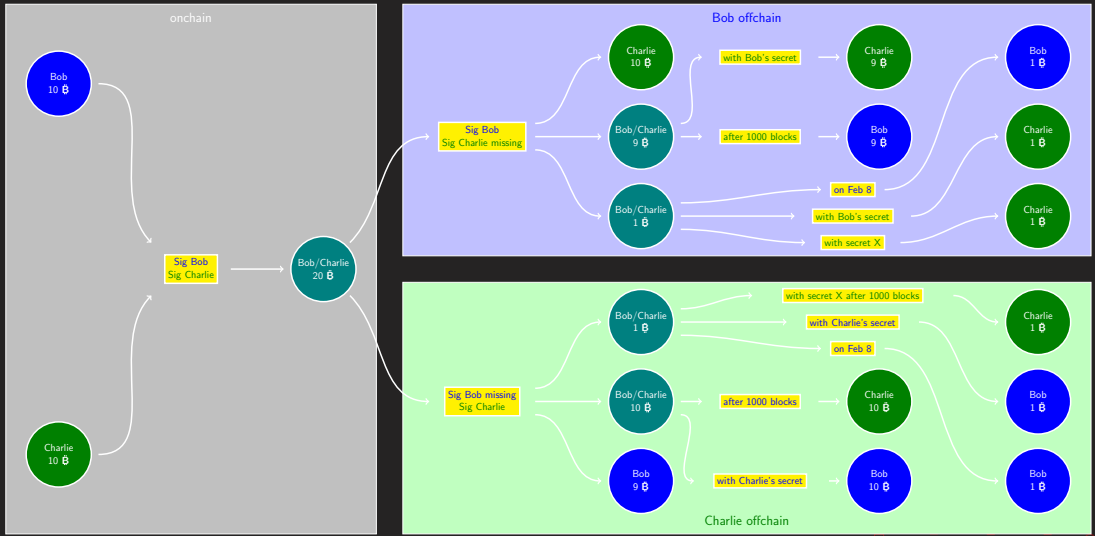
# Explanation

- The parts that are not effected by the payment of 1 ₿ are as before.

# Explanation

- The parts that are not effected by the payment of 1 ฿ are as before.
- For the payment of 1 ฿ a new multisig-address is created for both transactions, which can be unlocked in three different ways:
  - If Bob knows secret X and signs, he gets the money. However, he has to wait for 1000 blocks to receive it if he is the one that closes the channel. If he chooses this option, secret X will be publicly visible on the blockchain.
  - Whoever closes the channel gets the money if he or she knows the other's secret. As before, this makes outdated transactions useless.
  - Independent of who closes the channel, Alice can get her money back on February 9.

# Hash Time-Locked Contracts — Channel between Bob and Charlie



# Explanation

- The situation is analogous to the one between Alice and Bob.

# Explanation

- The situation is analogous to the one between Alice and Bob.
- However, the date when Bob can get his 1 ₿ back is before the date when Alice can get *her* 1 ₿ back, so that Bob has time to get his money from Alice as soon as Charlie has revealed secret X.

# Explanation

- The situation is analogous to the one between Alice and Bob.
- However, the date when Bob can get his 1 ₿ back is before the date when Alice can get *her* 1 ₿ back, so that Bob has time to get his money from Alice as soon as Charlie has revealed secret X.

**Remark**

Of course it is possible to do transactions in the same way with more than one intermediary. The only thing to keep in mind is to set the time-lock dates in a way that give parties further down the chain enough time to react.

# Explanation

- The situation is analogous to the one between Alice and Bob.
- However, the date when Bob can get his 1 ฿ back is before the date when Alice can get *her* 1 ฿ back, so that Bob has time to get his money from Alice as soon as Charlie has revealed secret X.

**Remark**

As for direct payments, payments using intermediaries normally do not require any transactions to be sent to the blockchain. As long as everybody plays by the rules, everything happens offchain.

# Summary

# Summary

- Scalability is one the biggest challenges facing blockchain technology: Centralized networks like Visa are orders of magnitude faster than current blockchain systems.

# Summary

- Scalability is one the biggest challenges facing blockchain technology: Centralized networks like Visa are orders of magnitude faster than current blockchain systems.
- Bitcoin Lightning is one possible solution to this problem, and its idea is general enough to be applicable to many other blockchain systems.

# Summary

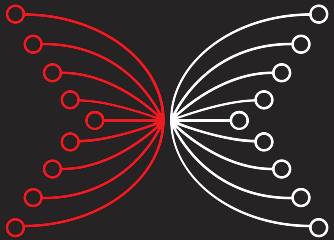
- Scalability is one the biggest challenges facing blockchain technology: Centralized networks like Visa are orders of magnitude faster than current blockchain systems.
- Bitcoin Lightning is one possible solution to this problem, and its idea is general enough to be applicable to many other blockchain systems.
- In Lightning, parties create bidirectional payment channels between each other and can then — using "channel chains" — do payments between parties they do not even share a channel with.

# Summary

- Scalability is one the biggest challenges facing blockchain technology: Centralized networks like Visa are orders of magnitude faster than current blockchain systems.
- Bitcoin Lightning is one possible solution to this problem, and its idea is general enough to be applicable to many other blockchain systems.
- In Lightning, parties create bidirectional payment channels between each other and can then — using "channel chains" — do payments between parties they do not even share a channel with.
- As long as anybody plays by the rules, only two Bitcoin transactions are necessary per channel, one to open the channel, one to close it again. All other transactions can be processed fast and cheap "offchain".

# Summary

- Scalability is one the biggest challenges facing blockchain technology: Centralized networks like Visa are orders of magnitude faster than current blockchain systems.
- Bitcoin Lightning is one possible solution to this problem, and its idea is general enough to be applicable to many other blockchain systems.
- In Lightning, parties create bidirectional payment channels between each other and can then — using "channel chains" — do payments between parties they do not even share a channel with.
- As long as anybody plays by the rules, only two Bitcoin transactions are necessary per channel, one to open the channel, one to close it again. All other transactions can be processed fast and cheap "offchain".
- Bitcoin guarantess the security of the system: If somebody violates the rules, no honest party loses their money.



INPUT | OUTPUT