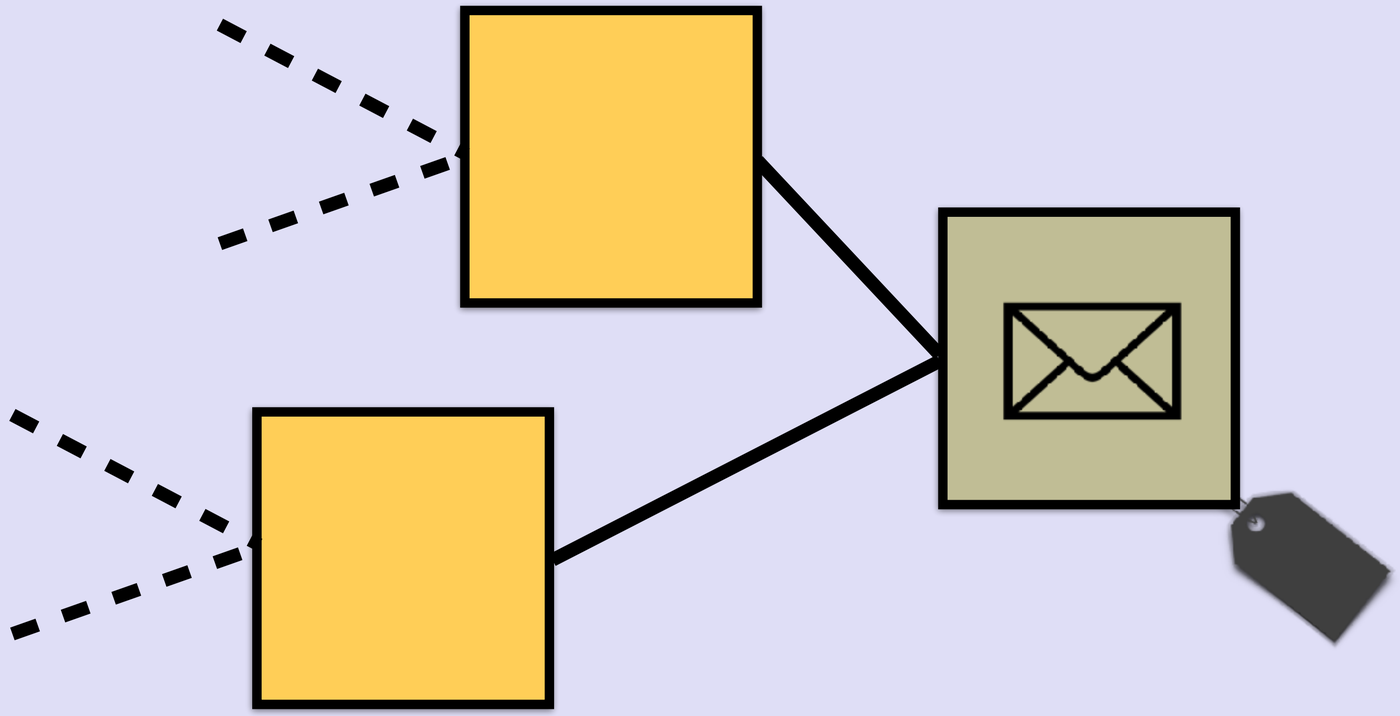
IOTA TUTORIAL 14

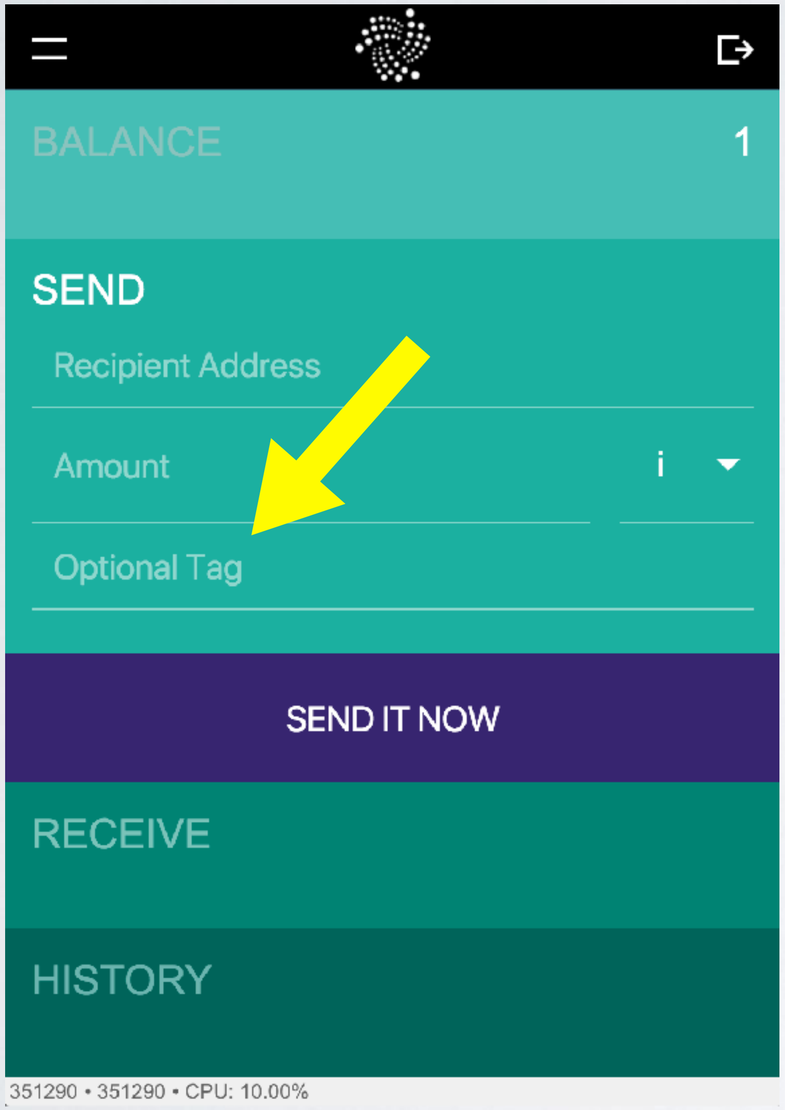
Message & Tag



INTRO

- In this video I will explain what the purpose is of a tag and how to store a message in a transaction.

TAG



IOTA Light wallet
When you create a transaction, you can enter a tag.
For example: "IOTA"

TAG


```
"hash": “BSOE...9999”,
"signatureMessageFragment": "9999...9999",
"address": "NIRU...HLMB9",
"value": 500000,
"obsoleteTag": "IOTA99999999999999999999999",
"timestamp": 1518220488,
"currentIndex": 3,
"lastIndex": 3,
"bundle": "QHCW...YK9C",
"trunkTransaction": "AJFW...9999",
"branchTransaction": "ROKZ...9999",
"tag": "IOTA99999999999999999999999",
"attachmentTimestamp": 1518220491486,
"attachmentTimestampLowerBound": 0,
"attachmentTimestampUpperBound": 3812798742493,
"nonce": "YA999RYA99999999999999999"
```


The transaction object tag field has the value “IOTA”

TAG

- The transaction object tag field is used to search transactions for a specific tag value.
- A value transaction example with the tag "MFISHTAG":
  https://www.mobilefish.com/download/iota/value_transaction_example.txt
- The tag is used in all transactions within the transaction bundle. If a transaction is reattached, the new created transaction will have the same tag.
- Use the Tangle explorer https://thetangle.org and search for the tag "MFISHTAG"
- Not all Tangle explorers can search old tags.
  The above mentioned Tangle explorer stores the full Tangle history.

TAG

- The transaction object tag field always contains 27 trytes (A-Z9).
- Create a tag by using only the characters A-Z9.
- Spaces in the tag are now allowed: "~~HELLO WORLD~~"
- If the tag is empty, the field contains all 9's: 999999999999999999999999999
- You can not use a Tangle explorer to search transactions for a specific **message**. You can only search for a specific **tag**.
- In Javascript, to search transactions for a specific tag use the findTransactionObjects API call.

MESSAGE

- With the IOTA Light wallet it is not possible to enter a message.

- For **EDUCATIONAL** purpose you can use the Mobilefish.com simple IOTA wallet where you can enter a message.
  https://www.mobilefish.com/services/cryptocurrency/iota_wallet.html

- The message is stored in the transaction object signatureMessageFragment field.
  The signatureMessageFragment field always contains 2187 trytes.

- 2187 trytes converted to bytes:
  bytes = (trytes x 3 x ln(3) / ln(2)) / 8
  bytes = (2187 x 3 x ln(3) / ln(2)) / 8 = ~1300 bytes = ~1.27 kBytes

MESSAGE

- In case of a **value transaction:**
  - In the transaction that spend IOTA where the value < 0, the signature is stored in the signatureMessageFragment field.
  - Subsequent transactions where the value == 0, the signatureMessageFragment fields are part of the signature.
- Value transaction example, the signature is fragmented and stored in two transactions (currentIndex 1 and 2)
  https://www.mobilefish.com/download/iota/value_transaction_example.txt

MESSAGE

- In case of a **data transaction:**
  - A zero value transaction is created (currentIndex = 0) and the message is stored in the signatureMessageFragment field.
  - Subsequent transactions where the value == 0, the signatureMessageFragment fields are part of the message.

MESSAGE

- Data transaction example, the text message is stored in one transaction (currentIndex = 0):
  https://www.mobilefish.com/download/iota/data_transaction_example.txt

- Data transaction example, the text message is fragmented and stored in two transactions (currentIndex 0 and 1):
  https://www.mobilefish.com/download/iota/data_transaction_example2.txt

- Data transaction example, the JSON message is stored in one transaction (currentIndex 0):
  https://www.mobilefish.com/download/iota/data_transaction_example3.txt

MESSAGE

- In case of a **value + data transaction:**
  - In the transaction that sends IOTA to a recipient (currentIndex = 0) where the value > 0, the message is stored in the signatureMessageFragment field.
  - Subsequent transactions where the value == 0, the signatureMessageFragment fields are part of the message.
  - In the transaction that spend IOTA where the value < 0, the signature is stored in the signatureMessageFragment field.
  - Subsequent transactions where the value == 0, the signatureMessageFragment fields are part of the signature.

MESSAGE

- Value + data transaction examples:
  - Message is stored in one transaction.
    https://www.mobilefish.com/download/iota/value_data_transaction_example.txt
  - Message is stored in two transactions.
    https://www.mobilefish.com/download/iota/value_data_transaction_example2.txt

MESSAGE

- If the signatureMessageFragment is empty the field contains all 9's.

- If the signature or message is longer than 2187 trytes, the signature or message are fragmented and stored over multiple transactions in the transaction bundle.

- But beware that each single transaction inside a transaction bundle consists of 2673 trytes which is ~1.55 kBytes.

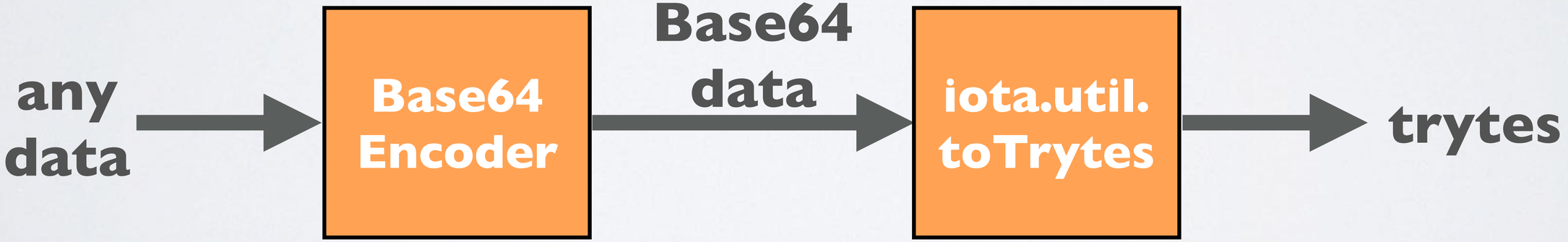
- If you have a message that is 20 kBytes in size, this message is fragmented and stored in 20 / 1.27 = 16 transactions inside the transaction bundle.
  The total transaction bundle size is 16 x 1.55 = ~25 kBytes
  Also for each of these 16 transactions a Proof of Work has to be done.

MESSAGE

- The signatureMessageFragment field can only store tryte values (A-Z9).

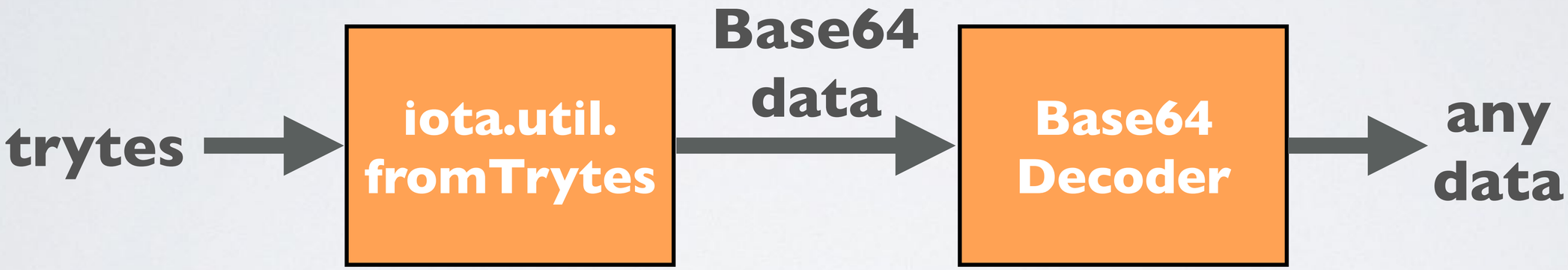
- The IOTA Javascript library has a method called iota.util.toTrytes which converts ASCII characters to trytes.

- You can store any data in the signatureMessageFragment field by first Base64 encode the data.
  The Base64 data only contains the characters **A**-**Z**, **a**-**z**, **0**-**9**, **+**, **/** and **=**
  The Base64 data can now be converted to trytes using the iota.util.toTrytes method.



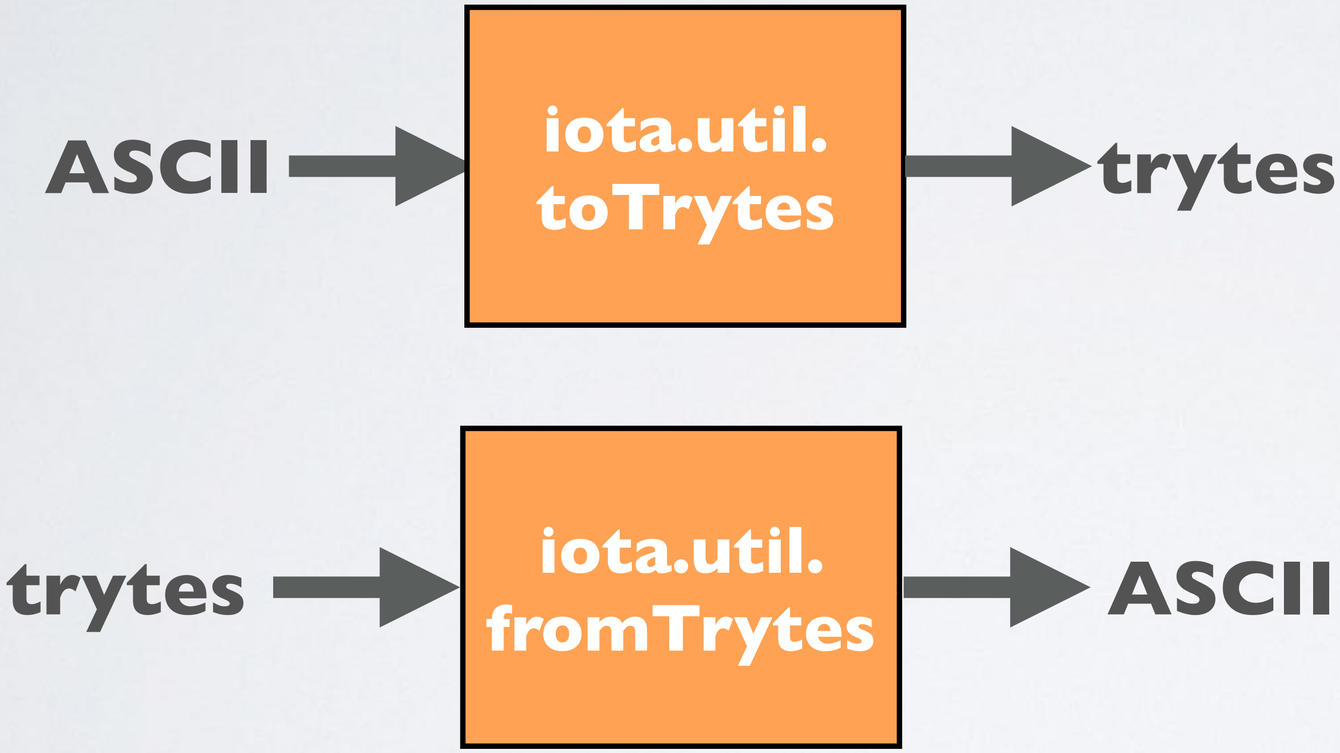
MESSAGE

- To restore the message to its original format use the iota.util.fromTrytes method to convert trytes to its Base64 representation, than use the Base64 decoder to convert the data to it original format.



MESSAGE

- If the original message only contains ASCII characters you can skip the Base64 encoding and decoding.



MESSAGE

- The message stored in the signatureMessageFragment field is not encrypted.
- If the message is fragmented and stored in multiple transactions you can manually restore the raw tryte message by concatenating all fragments, starting with the transaction with currentIndex 0 inside the transaction bundle.
- Just like a value transaction, a data transaction can be confirmed by the Tangle network.
- If you create a data transaction and you want to prove that this data exists on the Tangle or that there are no other conflicting data, than make sure this transaction is confirmed.

PERMANODES

- A snapshot is designed to reduce the size of the Tangle to reduce memory burden on nodes. When full nodes applies snapshots they only store the final account balances. Meta data such as tags and messages are deleted.

- A permanode is a node that stores the entire Tangle history permanently and securely. Permanodes don't apply snapshots.

- The Tangle explorer https://thetangle.org is a "permanode" but the transactions are only searchable thru their website.

- In the future permanodes will be available and probably they be incentivised by pay per query.