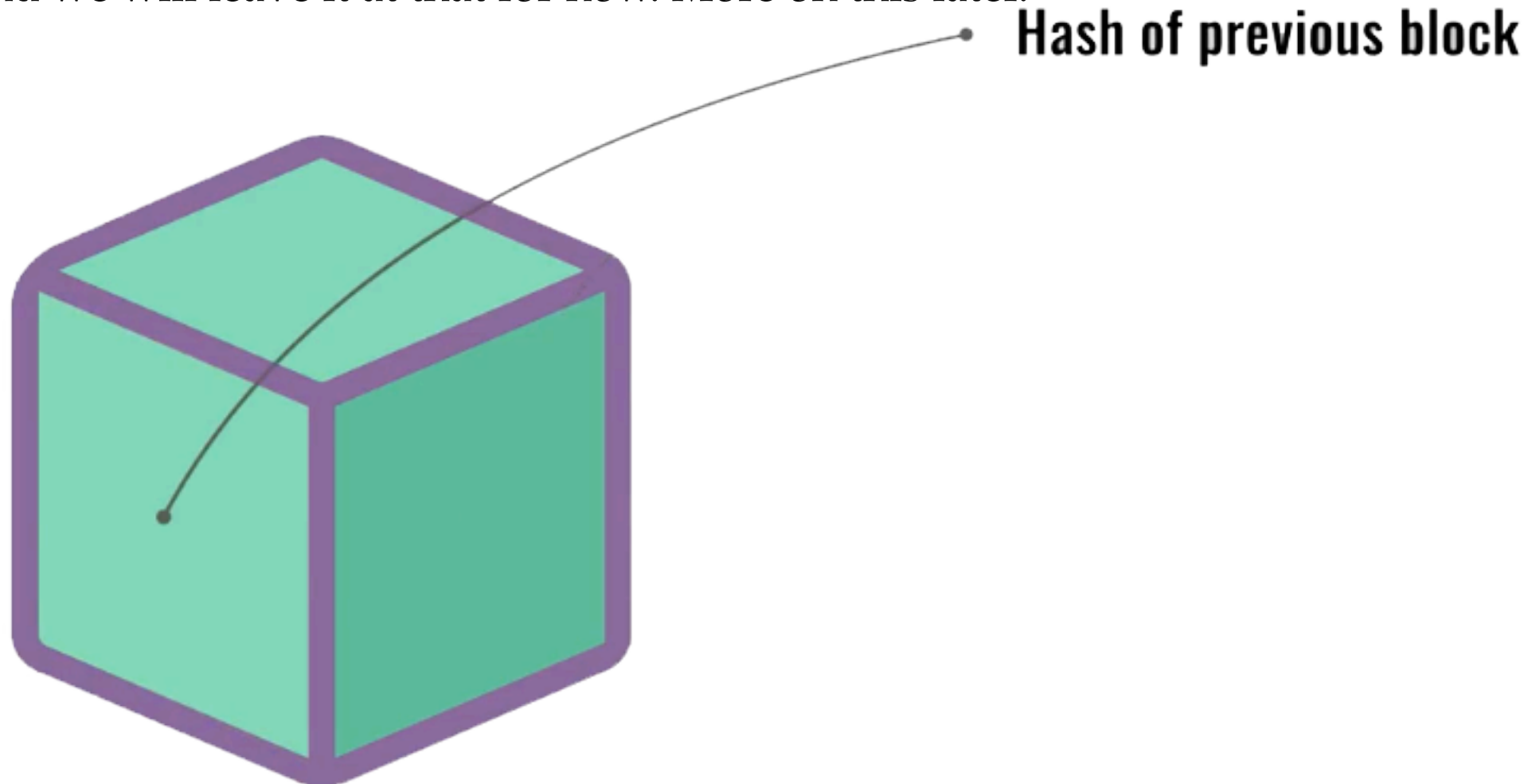


BLOCK HEADER:

The header contains metadata about a block. There are three different sets of metadata:

- The previous block hash. Remember that in a blockchain, every block inherits from the previous block because we use the previous block's hash to create the new block's hash. For every block N, we feed it the hash of the block N-1.
- Mining competition. For a block to be part of the blockchain, it needs to be given a valid hash. This contains the timestamp, the nonce and the difficulty. Mining is another crucial part of the blockchain technology, but it is outside the scope of this article.
- The third part is a merkle tree root. This is a data structure to summarise the transactions in the block. And we will leave it at that for now. More on this later.



BLOCK IDENTIFIERS:

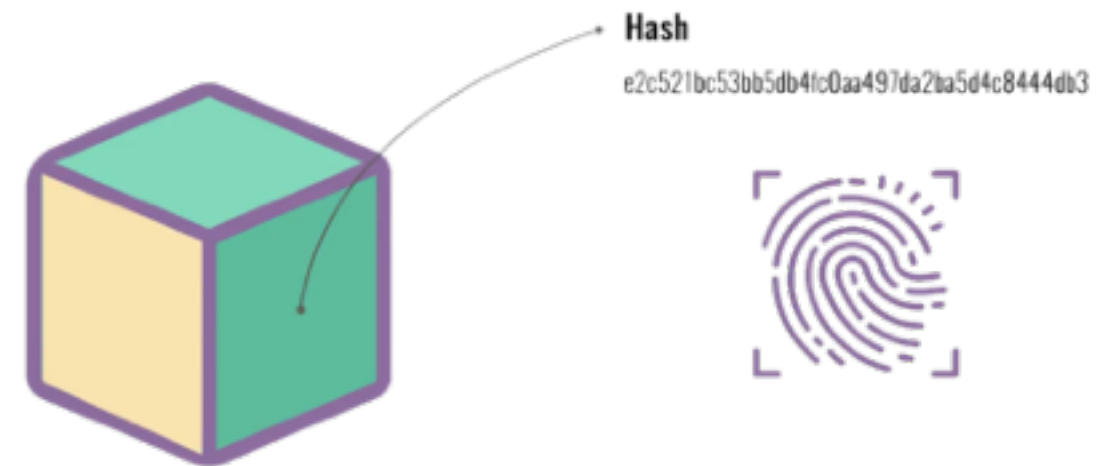
To identify a block, you have a cryptographic hash, a digital signature if you will. This is created by hashing the block header twice with the SHA256 algorithm. For example, this is a block. I will refer to this block as an example for this article.

The block header hash for this particular block is (right column):

```
000000000000000000000000301fcfeb141088a93b77dc0d52571a1185b425256ae2fb
```

We also can see the previous block's hash (left column):

```
000000000000000000004b1ef0105dc1275b3adfd067aed63a43324929bed64fd7
```



Remember that we used the second hash to create the first. Every block uses the previous block's hash to construct its own hash. The block hash is a unique identifier. You won't find two blocks with the same hash. The other way to identify a specific block is the block height. This is the position of the block in the blockchain. Our example's block is in the 500312 position. This means that there are 500311 blocks before this one. Since the creation of the Bitcoin blockchain in 2009, 500312 blocks have been created (at the time of writing obviously). A block height is not unique. Several blocks can compete for the same position in the case of a fork, like Bitcoin Cash for example.