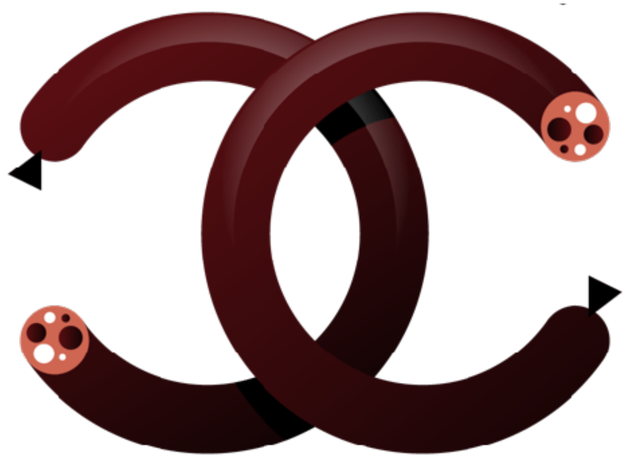


GRAPH TEAM MEMBER

CARL HAGERLING

GRAPH TEAM MEMBER

NENA DJAJA

GRAPH TEAM MEMBER

DAVID KAJPUST

GREAT GREAT GREAT GRANDFATHER

ERLING WENNERSTEN

GREAT GREAT GRANDFATHER

HUGO WENNERSTEN

GREAT GRANDFATHER

UNO WENNERSTEN

ONE PROBLEM

THEY WERE ALL

ROBBED

HOW COULD WE AVOID THE ROBBING OF SAUSAGE SALESMEN?

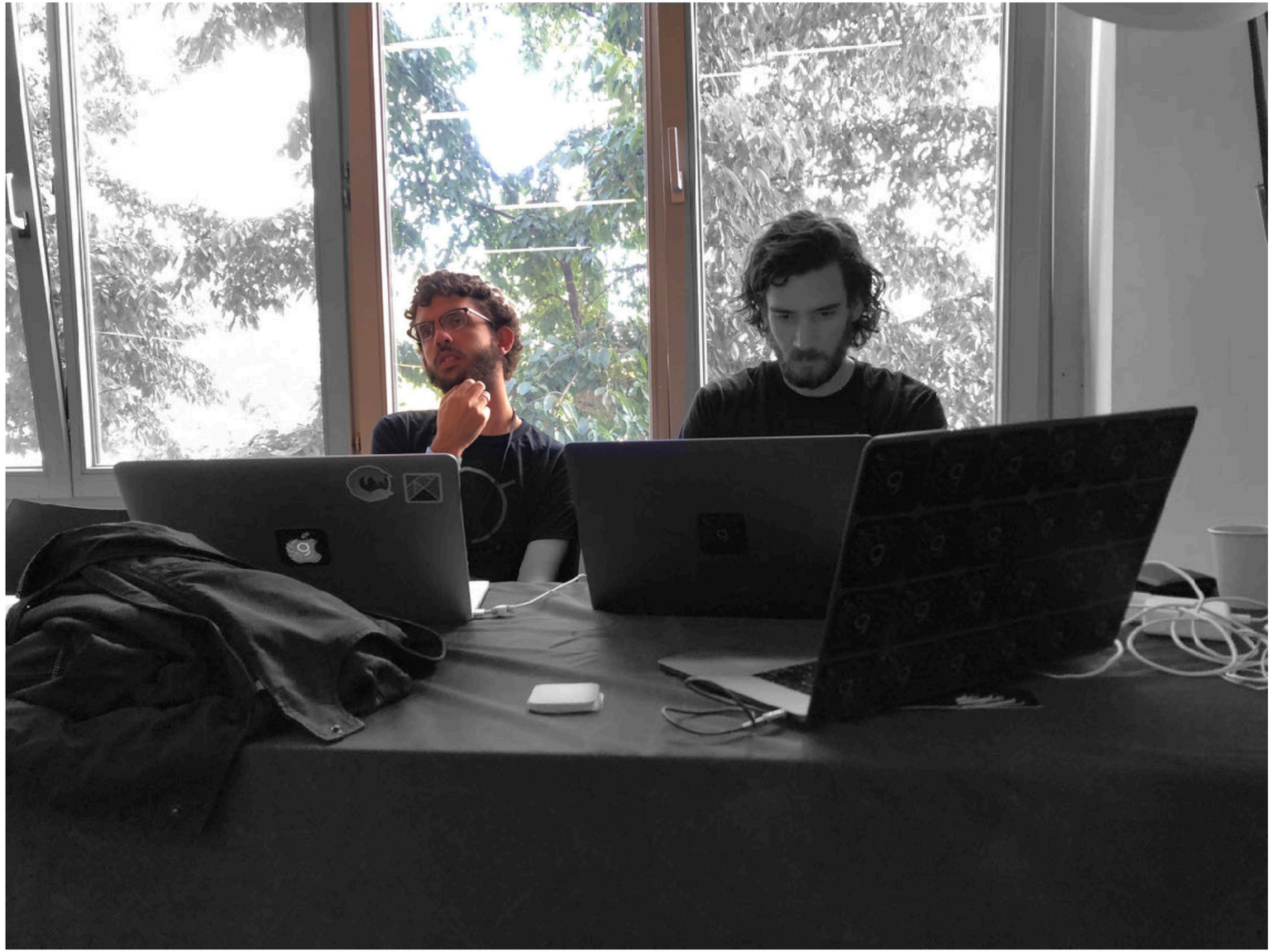
(AND SALESWOMEN)

SO WE DID SOME THINKING...

...AND SOME MORE THINKING

...WE ALMOST GAVE UP

…BUT THEN, WE HAD AN EPIPHANY



THE SOLUTION

WHAT IF...

FIVE YEARS FROM NOW, EVERY SAUSAGE IN GERMANY WILL BE PURCHASED WITH BITSAUSAGE

BITSAUSAGE

THE WURST CRYPTOCURRENCY IN THE UNIWURST


BITSAUSAGE
AUCTION
WURST
CRYPTOCURRENCY

COMPONENTS

SMART CONTRACT

- A new BitSausage NFT-token is minted every X seconds, based on stats about annual sausage consumption in Germany
- Each BitSausage Token is of a different type — currywurst bratwurst, etc.

SUBGRAPH BACKEND

- The Subgraph sources all the Auction Events, which allows for a smooth developer experience for creating a live up-to-date Dapp

BITSAUSAGE DAPP

- The Bit Auction Dapp gives German's the ability to purchase a BitSausage Token
- They can then exchange it for a real-life, edible sausage

HOW IT WORKS

PRODUCED
GENERATED

HOW IT WORKS

BUY ONE GET ONE

THE SCHEMA

```
type Auction {
  id: ID! @unique
  seller: User
  bids: [Bid!]! @relation(name: "AuctionOnBids")
  currentBid: Bid @relation(name: "AuctionOnCurrentBid")
  previousBid: Bid @relation(name: "AuctionOnPreviousBid")
  expirationTime: DateTime
  item: SausageToken
}

type Bid {
  id: ID! @unique
  amount: Int!
  timestamp: DateTime
  bidder: User
  auction: Auction @relation(name: "AuctionOnBids")
}

type SausageToken {
  id: ID! @unique
  type: BitSausageType
}

type User {
  id: ID! @unique
  name: String
  address: String
  bid: Bid
}

enum BitSausageType {
  CurryWurst
  Bratwurst
  Weisswurst
  Frankfurter
  Knackwurst
  Leberwurst
  Blutwurst
}
```

THE CONTRACT

```
pragma solidity ^0.4.23;
contract Auction {
  address public manager;
  address public seller;
  uint public latestBid;
  address public latestBidder;
  uint public auctionSecondsLeft;
  address public erc721TokenAddr;
  bool public auctionLive;

  constructor( address sausageTokenAddr) public {
    manager = msg.sender;
    erc721TokenAddr = sausageTokenAddr;
  }

  event LogNewBid(
    address _latestBidder,
    address _amount,
    uint256 _timeLeft,
    string  _name;
  );

  event LogAuctionOpen(
    string _symbol, //CURRY = currywurst , GOLD = goldwurst , etc.
    address _tokenAddress,
    uint256 _startingTime,
    string _uniqueID,

  );

  event LogAuctionClosed(
    string _symbol, //CURRY = currywurst , GOLD = goldwurst , etc.
    address _tokenAddress,
    uint256 _startingTime,
    string _uniqueID,
    address _winner,
    uint256 _finalBid,
  )
```

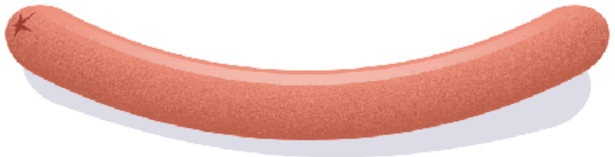
BID FOR BITSBAUSAGES

COLLECT YOUR FAVORITES

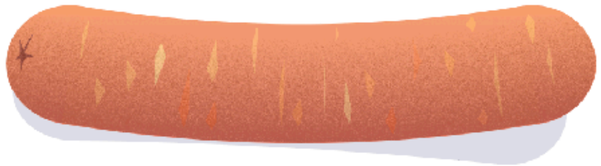
CURRYWURST

BRATWURST

WEISSWURST



FRANKFURTER

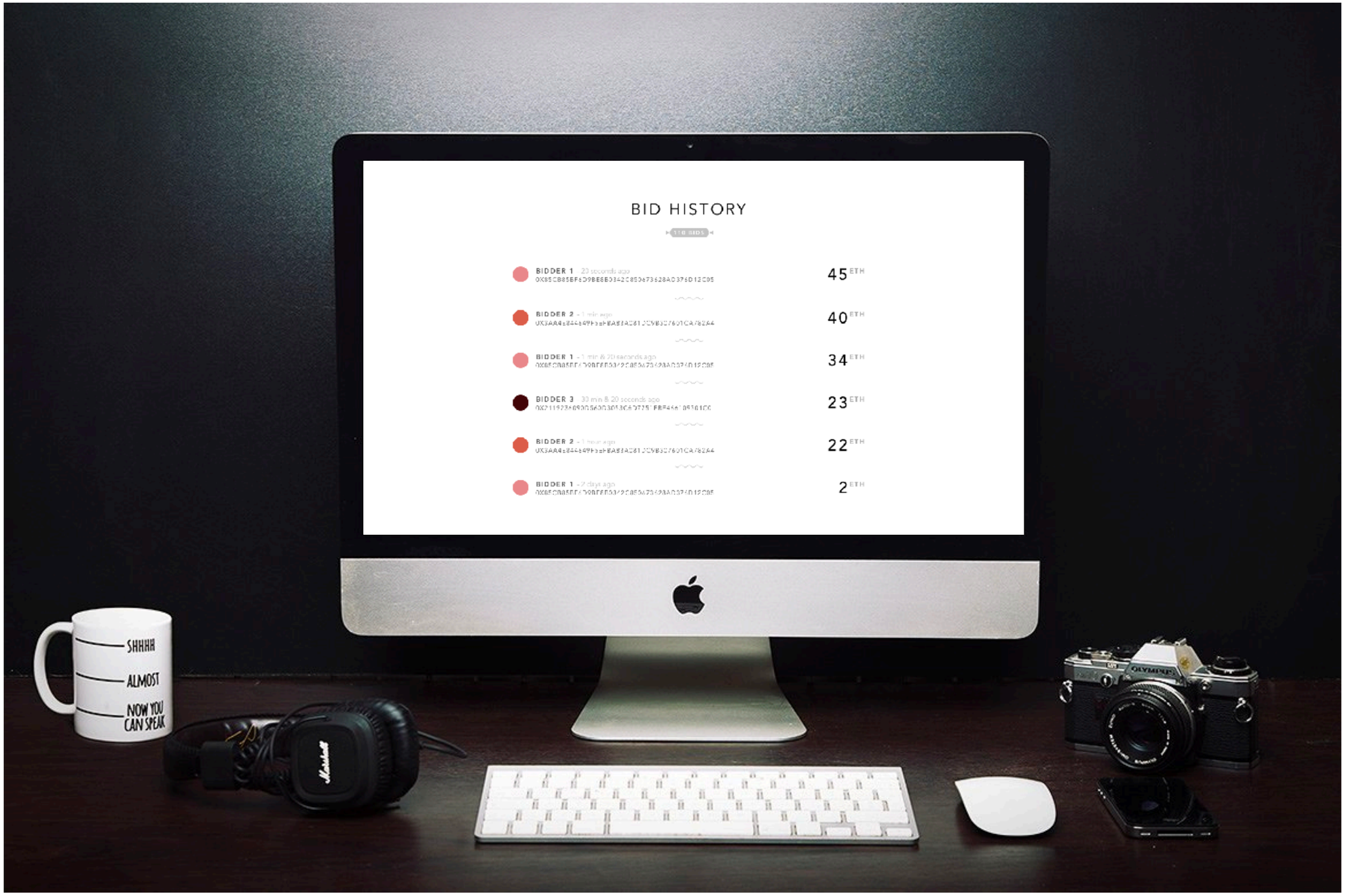


KNACKWURST

LEBERWURST

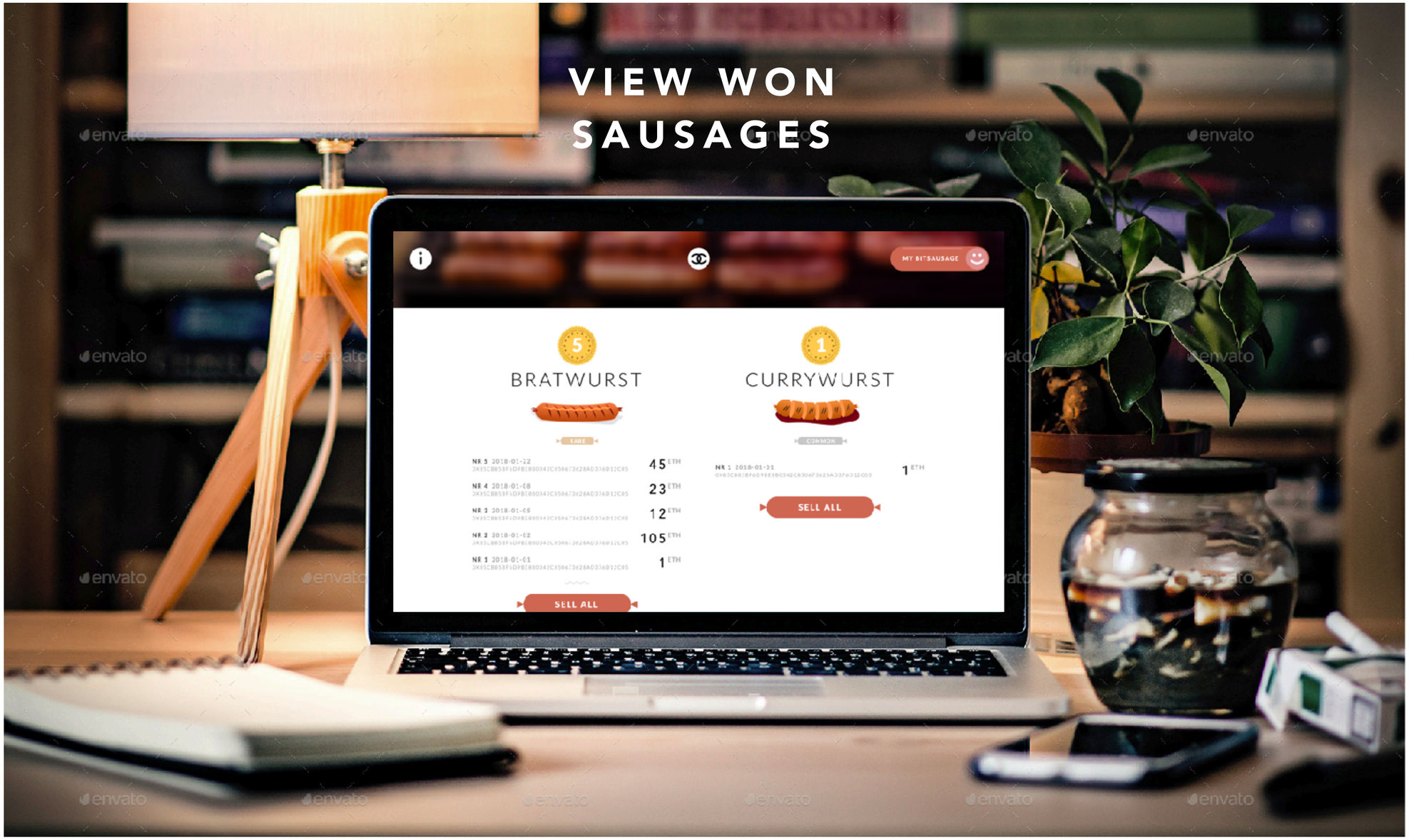
BLUTWURST

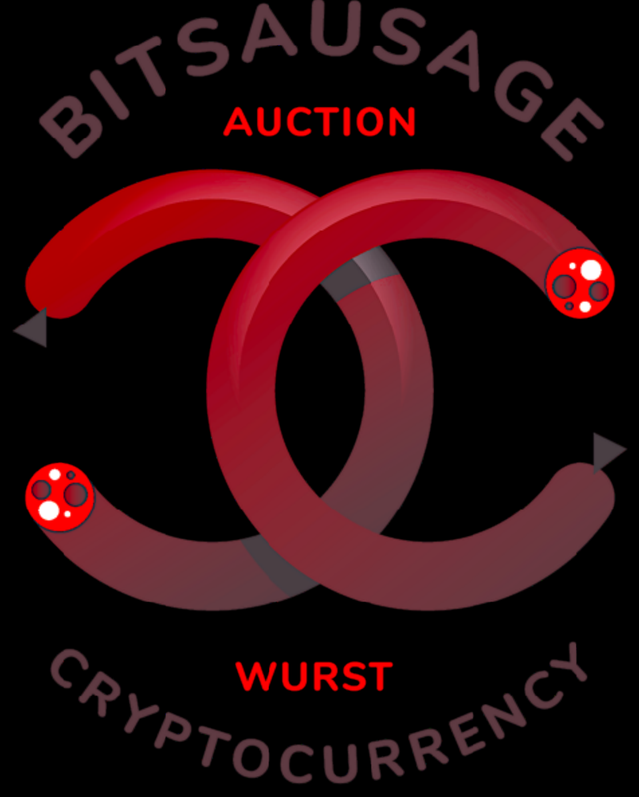
WORSTWURST



VIEW HISTORY

VIEW WON SAUSAGES




BITSAUSAGE
AUCTION
WURST
CRYPTOCURRENCY