DeCloud: Truthful Decentralized Double Auction for Edge Clouds

A. Zavodovski, S. Bayhan, N. Mohan, P. Zhou, W. Wong and J. Kangasharju

ICDCS 2019, Dallas, Texas

Motivation

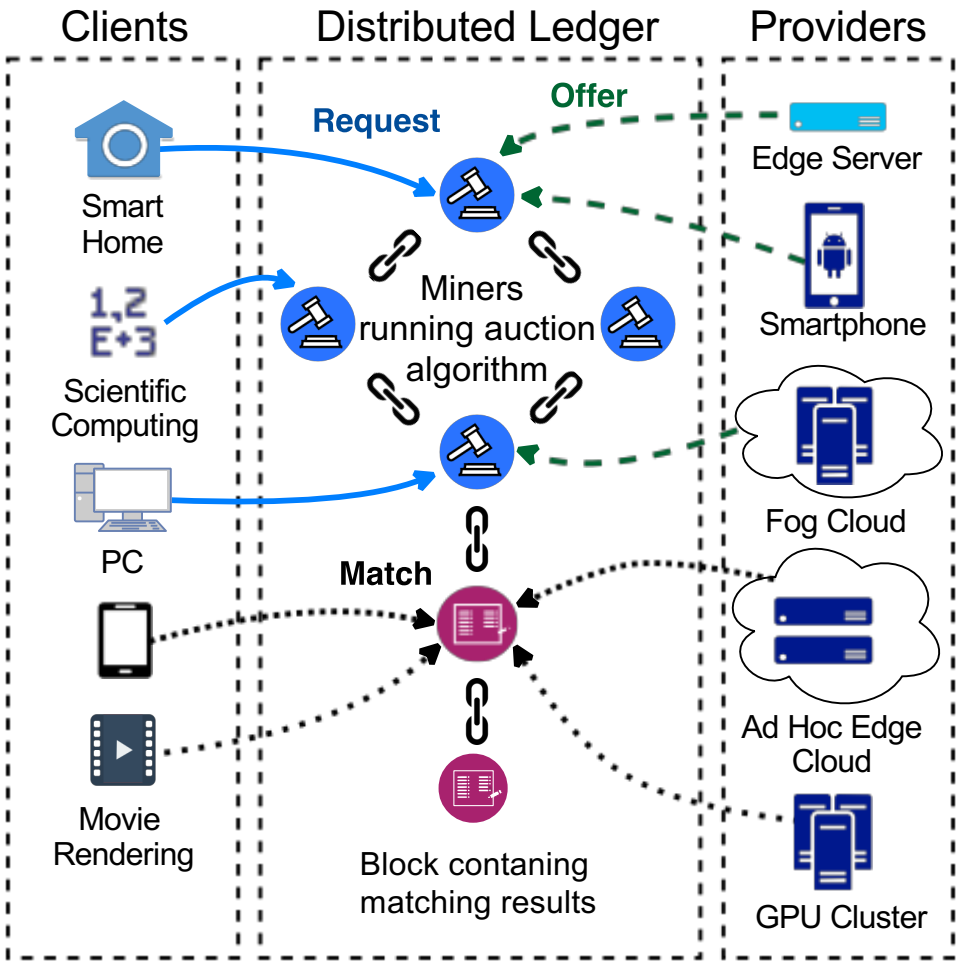
- Growing demand for edge resources, making edge pervasive
- Seeking an alternative to big cloud providers, tackling monopoly and ossification
- Current crowdsourced systems (e.g., iExec, Golem) lack market model
- Devise a market model that would eliminate the need for complex strategizing

Our Proposal – DeCloud

- Provides the market model where rational participants achieve best payoff by following dominant strategy
- Incorporates custom heuristics for matching highly heterogenous resources with diverse demands
- Runs on top of distributed ledger
  - Requires no central authority
- Enables to use consumer, crowdsourced or any other devices for edge purposes, i.e. anyone can become an edge service provider (ESP) and get compensation
  - Crowdsourced devices are generally underutilized and located exactly where they are most needed for edge computing – at the edge of the network

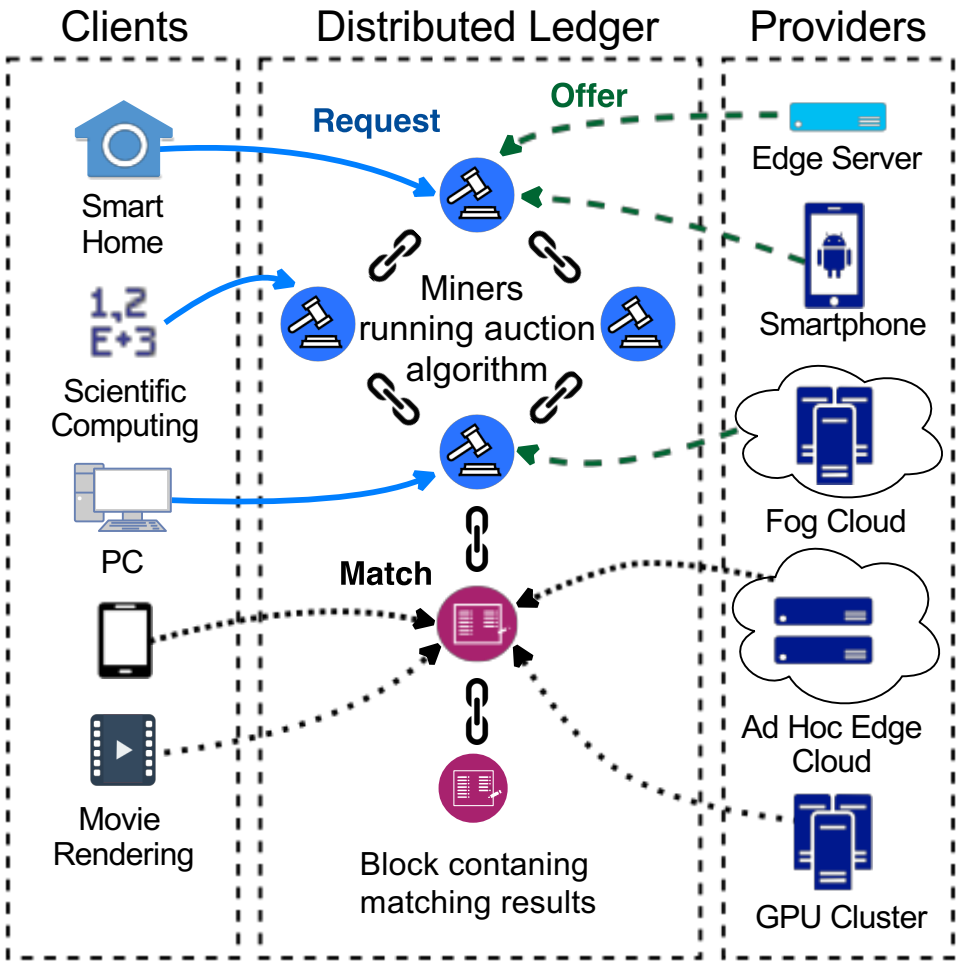
Overview of DeCloud

- Clients and providers submit their bids: requests and offers
- P2P network of miners aggregates bids in block candidates (similarly to transactions in any other blockchain system)
- Miner which discovers a block also computes allocation, i.e. match between clients and providers



Challenges

1. P2P network is open, truthful auction needs sealed bids
   - ✓**Two-phase bid expose protocol**
2. High heterogeneity of resources and demand
   - ✓**Custom matching heuristics**
3. Finding optimal market behavior is complex
   - ✓**Truthful auction – bid your privately known valuation**

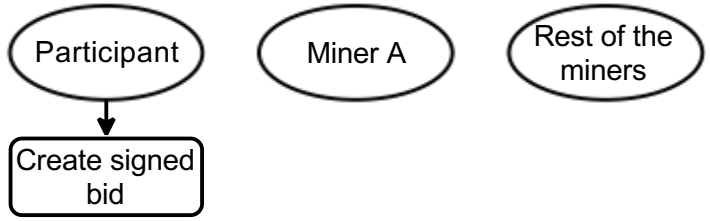


Challenge #1: Sealed Bids on a Blockchain

- Since offers and requests are propagated as transactions across P2P blockchain network, we need to encrypt them and establish two major phases:
  - Bids are sealed
  - Bids are open and allocation (matching) can be computed
- Main idea is to tie bids to cryptographically secured block
  - Set of bids may not be altered after block is generated
  - It is possible for other miners to verify the correctness of allocation algorithm execution
  - Bids are decrypted only after they are tied to valid block

Two-phase bid expose protocol

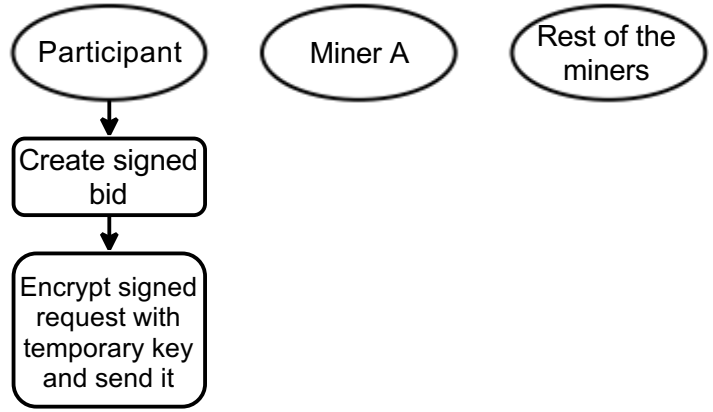
- Participants encrypt bids with temporary keys



Phase I: bidding

Two-phase bid expose protocol

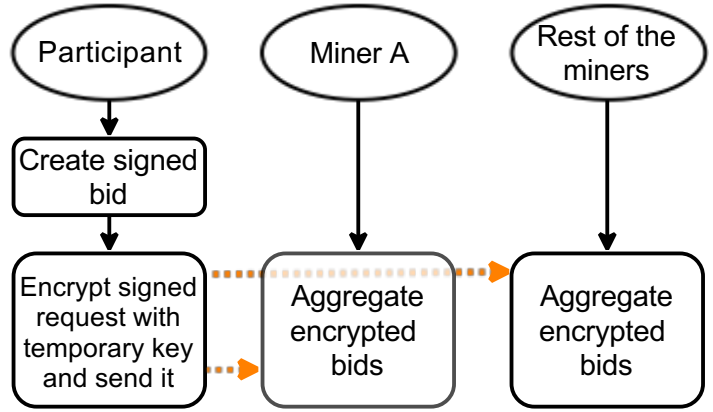
- Participants encrypt bids with temporary keys



Phase I: bidding

Two-phase bid expose protocol

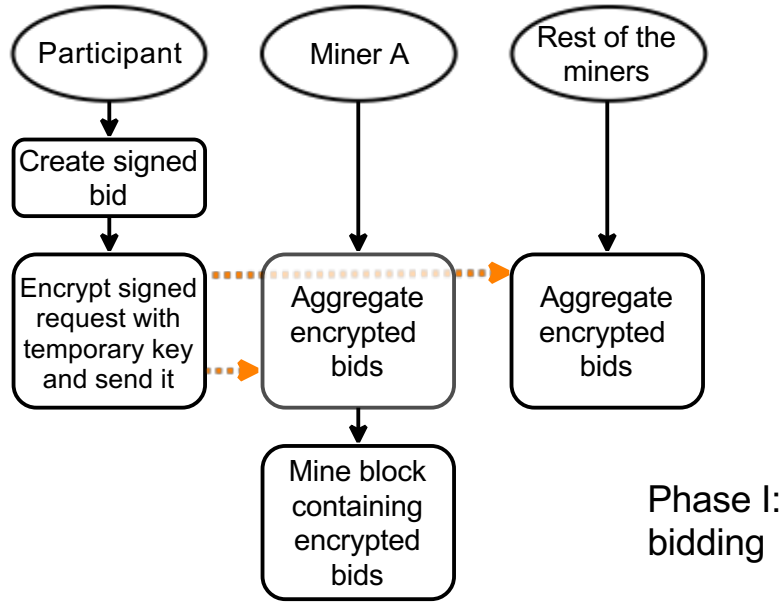
- Participants encrypt bids with temporary keys



Phase I: bidding

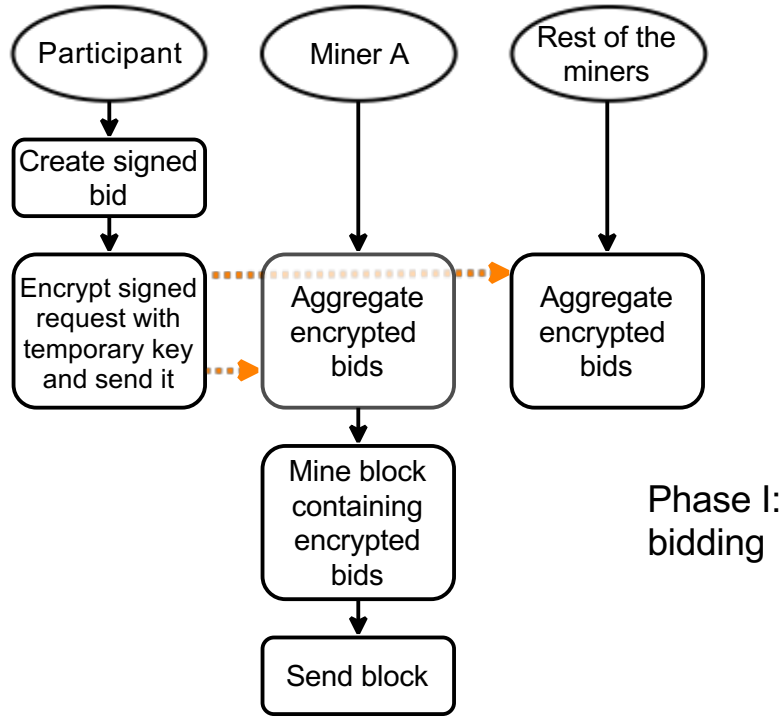
Two-phase bid expose protocol

- Participants encrypt bids with temporary keys
- When block containing encrypted bids is mined, it is broadcasted to the network



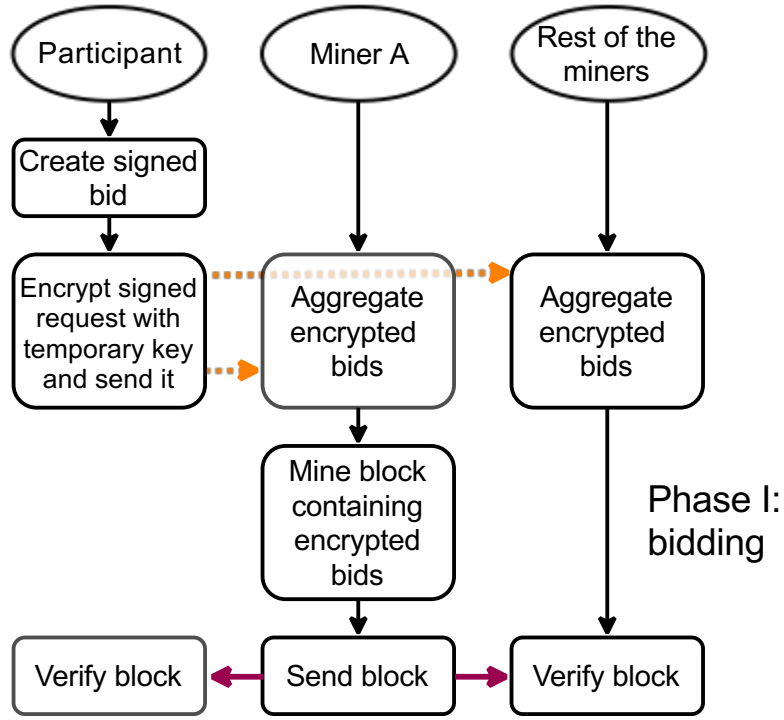
Two-phase bid expose protocol

- Participants encrypt bids with temporary keys
- When block containing encrypted bids is mined, it is broadcasted to the network



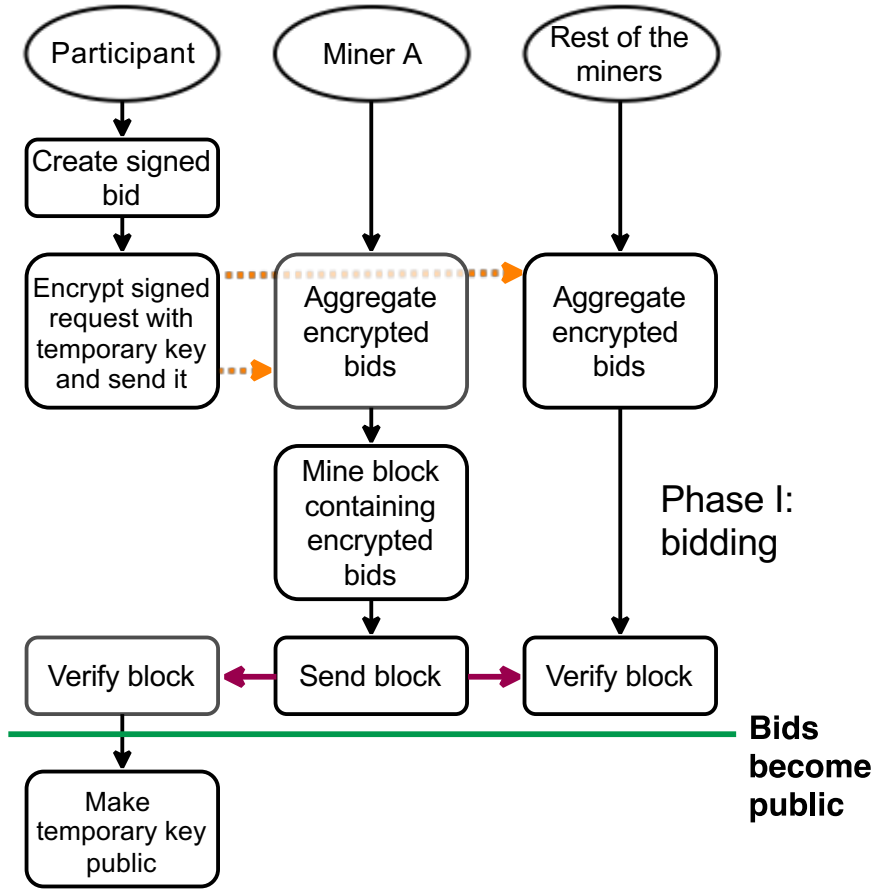
Two-phase bid expose protocol

- Participants encrypt bids with temporary keys
- When block containing encrypted bids is mined, it is broadcasted to the network



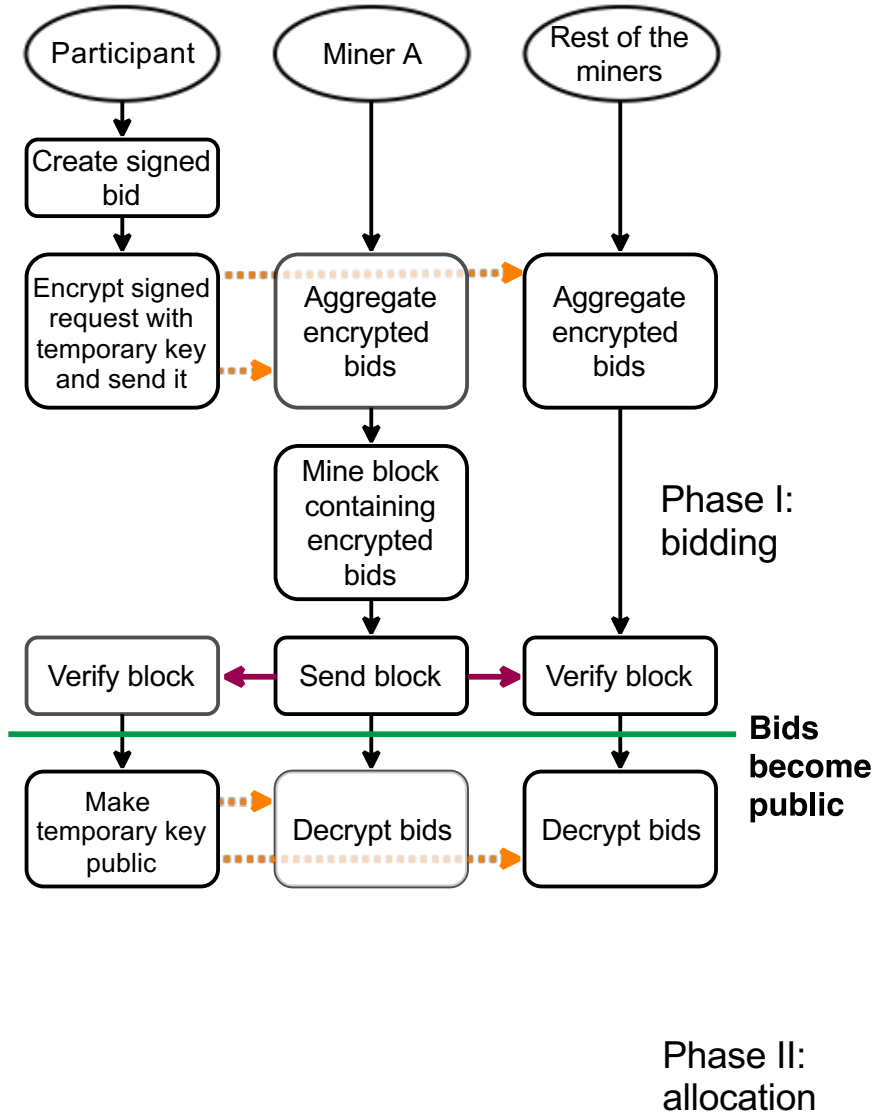
Two-phase bid expose protocol

- Participants encrypt bids with temporary keys
- When block containing encrypted bids is mined, it is broadcasted to the network
- As a reply, participants broadcast their temporary keys if their bid is in the block



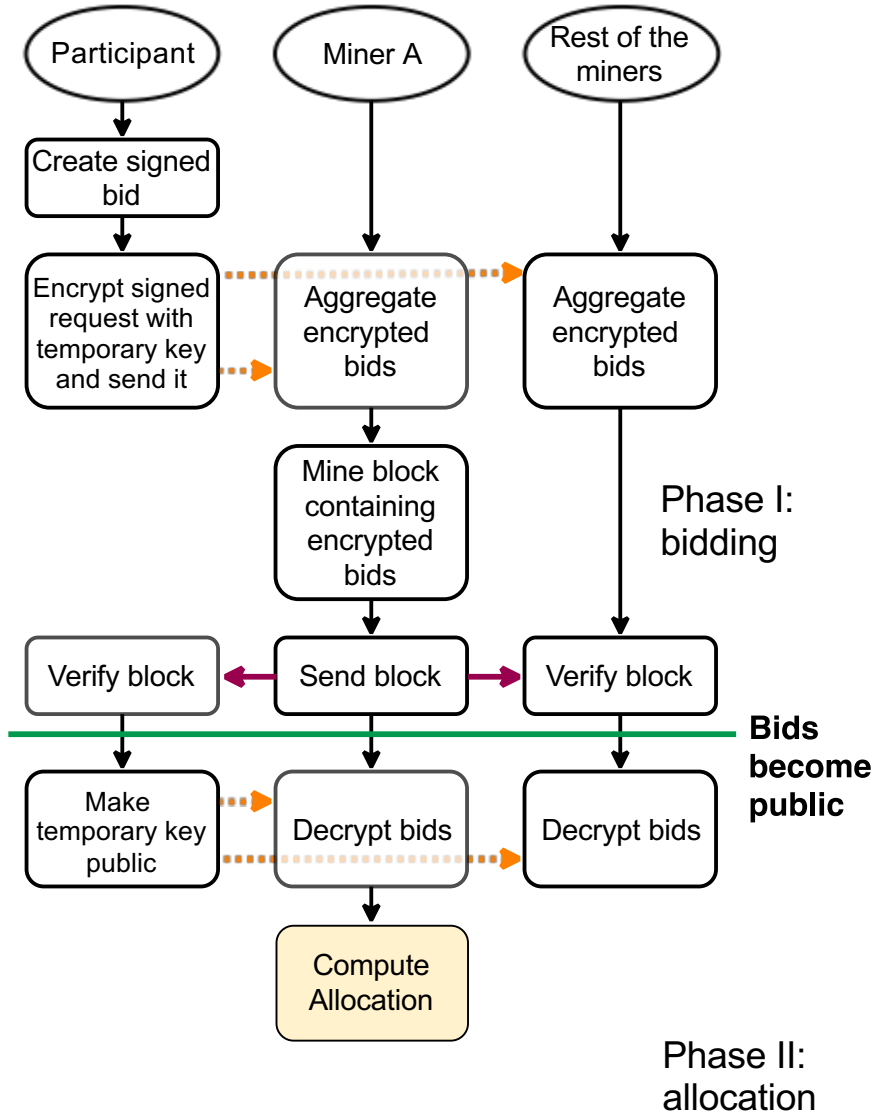
Two-phase bid expose protocol

- Participants encrypt bids with temporary keys
- When block containing encrypted bids is mined, it is broadcasted to the network
- As a reply, participants broadcast their temporary keys if their bid is in the block
- Content of the block is then decrypted and allocation can be computed



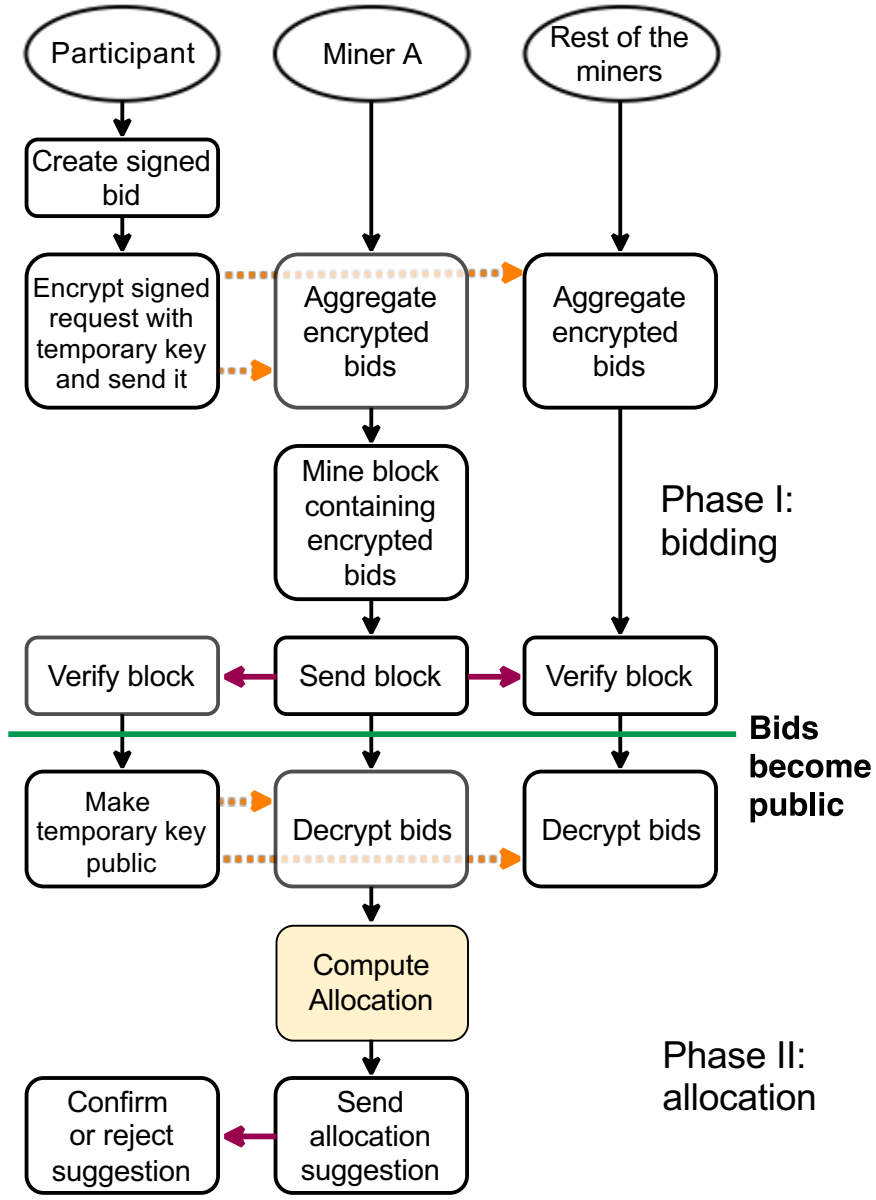
Two-phase bid expose protocol

- Participants encrypt bids with temporary keys
- When block containing encrypted bids is mined, it is broadcasted to the network
- As a reply, participants broadcast their temporary keys if their bid is in the block
- Content of the block is then decrypted and allocation can be computed



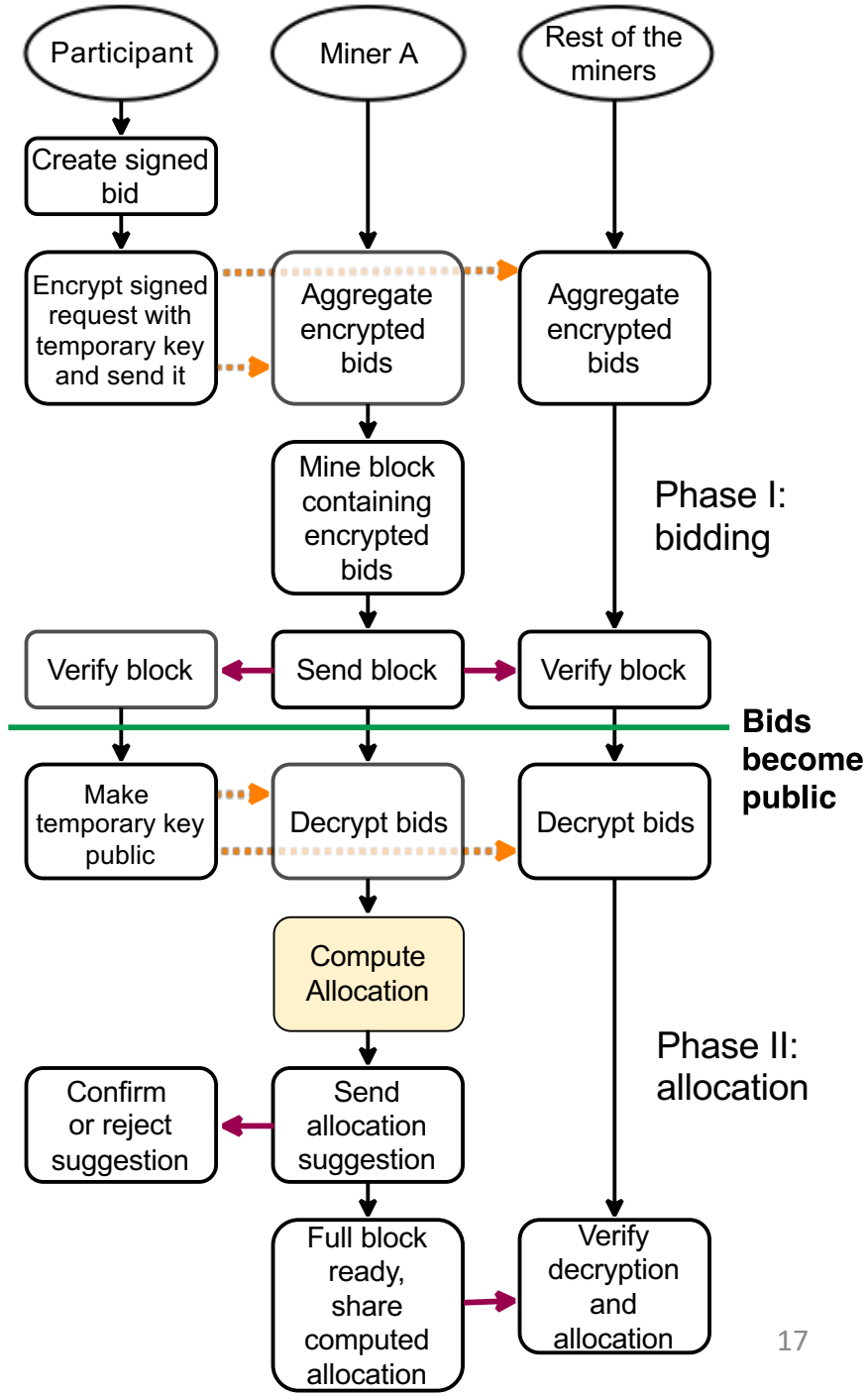
Two-phase bid expose protocol

- Participants encrypt bids with temporary keys
- When block containing encrypted bids is mined, it is broadcasted to the network
- As a reply, participants broadcast their temporary keys if their bid is in the block
- Content of the block is then decrypted and allocation can be computed



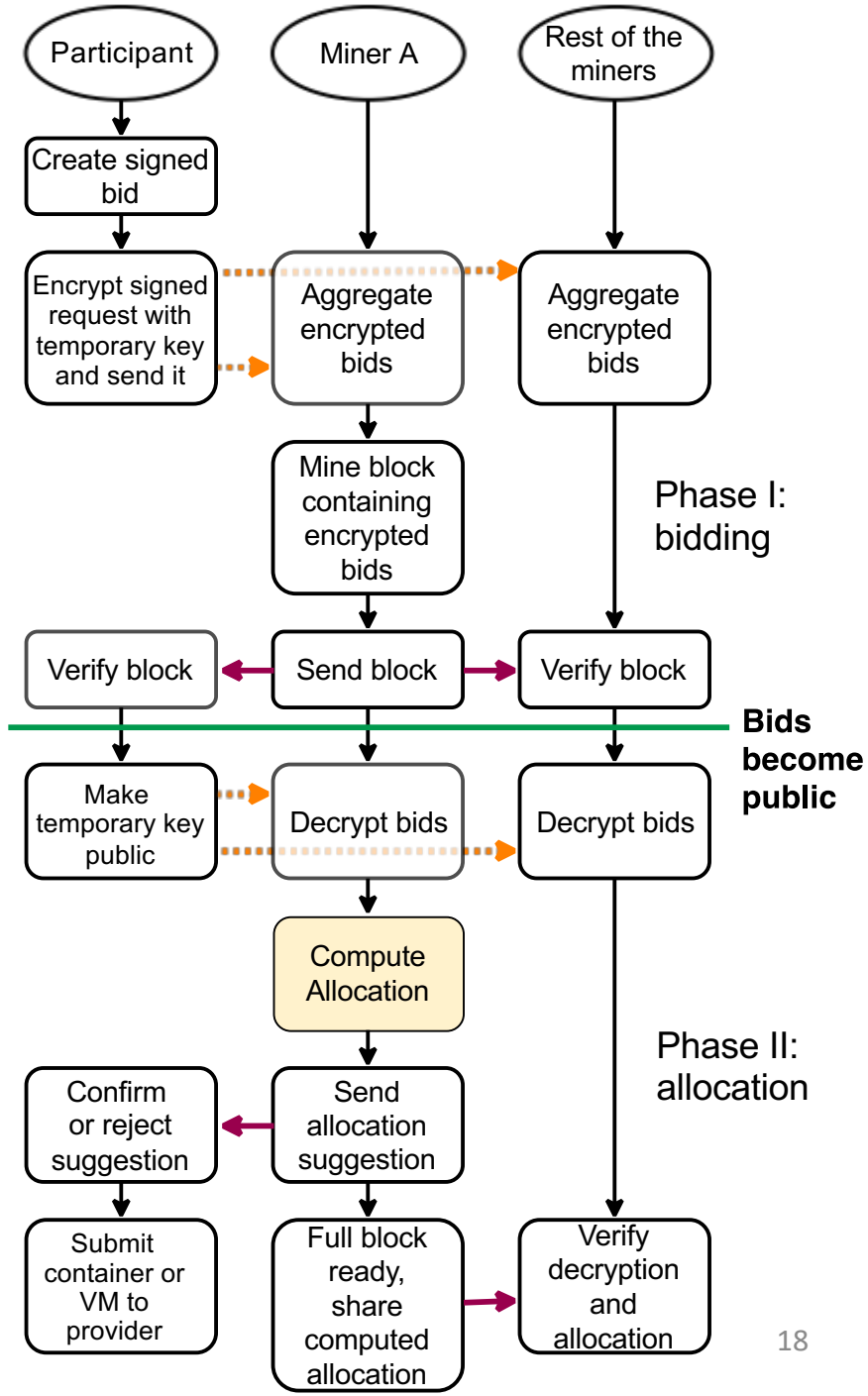
Two-phase bid expose protocol

- Participants encrypt bids with temporary keys
- When block containing encrypted bids is mined, it is broadcasted to the network
- As a reply, participants broadcast their temporary keys if their bid is in the block
- Content of the block is then decrypted and allocation can be computed
- Full valid block consists of two parts, preamble (encrypted bids), and computed allocation (match between requests and offers)



Two-phase bid expose protocol

- Participants encrypt bids with temporary keys
- When block containing encrypted bids is mined, it is broadcasted to the network
- As a reply, participants broadcast their temporary keys if their bid is in the block
- Content of the block is then decrypted and allocation can be computed
- Full valid block consists of two parts, preamble (encrypted bids), and computed allocation (match between requests and offers)



Challenge #2: Matching Requests and Offers

- The common problems of open crowdsourced environment:
  - heterogeneity of resources
  - diversity of demand
- Edge imposes own requirements, e.g., latency, and location becomes important:
  - Someone may want just a Raspberry Pi but in **specific location**
- DeCloud:
  - Let participants describe exactly what they want and give them the best possible match out of available resources
  - Since exact match is not always possible, let participants express importance of the resources by weights
  - Everything is a resource: location, reputation, etc.

Finding the Best Match

- Offers and requests are represented as normalized vectors
- For example, assume client wants 4 CPU cores
- Distance does not work well:
  - if there are two offers with 2 and 8 cores, then 2 is closest to 4
- Vector dot product does not address the flexibility well:
  - If there are offers with 3 and 8 cores, vector product will match the request with offer having 8 cores
  - For a flexible client 3 cores is likely to be a better match.

Quality of Match

If we assume resource to have a mass, then by slight modification of Newtonian formula we get following definition for quality of match between request r and offer o (as a weighted sum):

$$\sum_{k \in K}^{K} \left(w_{r,k} \frac{\rho_{o,k}}{|\rho_{o,k} - \rho_{r,k}|^2 + 1} \right)$$

where ρ where is amount of resource, k is type of resource, $w_{r,k} \in [0,1]$ defines how important resource of type k is for request r. If $w_{r,k} = 1$, then condition $\rho_{o,k} \geq \rho_{r,k}$ must hold, but not otherwise.

Challenge #3: Truthful Auction

- Dominant strategy incentive compatible (DSIC) auction
  - Dominant strategy – a strategy that provides best payoff no matter what other players do
  - Incentive compatibility – acting according to true preferences, in our context bidding privately known valuation
- Most known example – Vickrey or second price auction
  - Sealed bids submitted to auctioneer for some single indivisible good
  - The highest bid wins
  - The winner pays what second highest bidder has offered for the good

Some Auction Terms and Metrics

- **Payoff or utility** – this is what rational participants want to maximize:
  - Difference between amount paid and privately known (true) valuation
- **Revenue**:
  - What seller(s) receives, Vickrey auction clearly does not optimize for this
- **Welfare**:
  - Giving the goods to those who value them most, i.e. maximizing the sum of valuations, Vickrey auction achieves best possible welfare

Double Auction in DeCloud

- In double auction **both** sellers and buyers submit their bids, forming the market
- Optimizing for **revenue** puts sellers in the privileged position
- Thus, **welfare** is more suitable as a performance metric for DeCloud
- In double auction, welfare is sum of valuations of allocated (winning) participants minus all costs of allocated sellers:

$$w_\beta(X_\beta) = \sum_{r \in R^\beta} \sum_{o \in O^\beta} v_r x^\beta_{(r,o)} - \sum_{o \in O^\beta} \sum_{r \in R^\beta} x^\beta_{(r,o)} \varphi^\beta_{(r,o)} c_o$$

Cost of offer o

For all requests and offers in block beta

Valuation of request r

Allocation vector {0,1}

Fraction of resources allocated to r

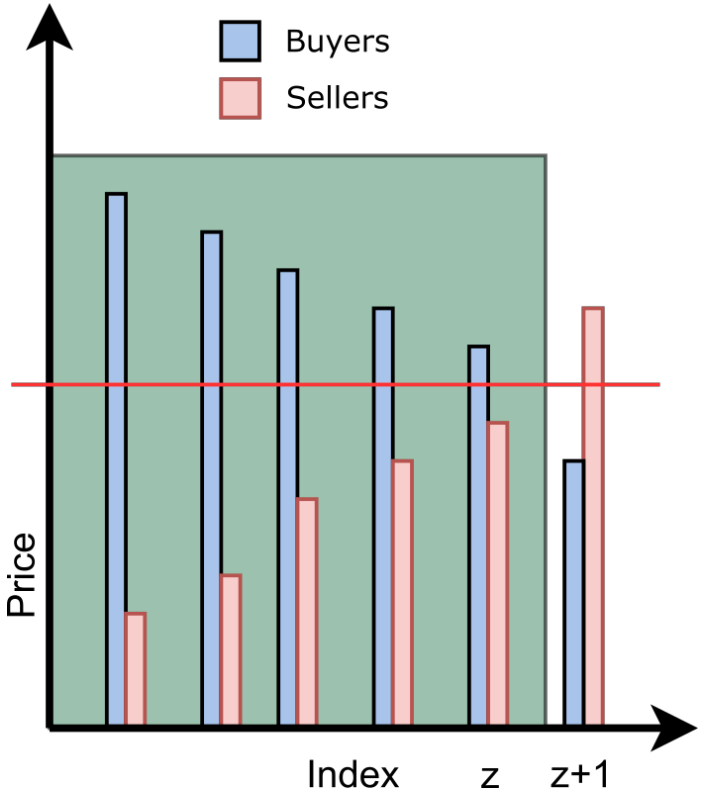
DSIC Double Auction

- McAfee has shown[1)], that for double auction with just one seller and buyer DSIC auction is not possible
- McAfee has offered a double auction mechanism where DSIC property is achieved for more than one pair of participants
- McAfee's solution became known as **trade reduction** mechanism, because the valid seller-buyer pair that determines the trading price must be excluded.

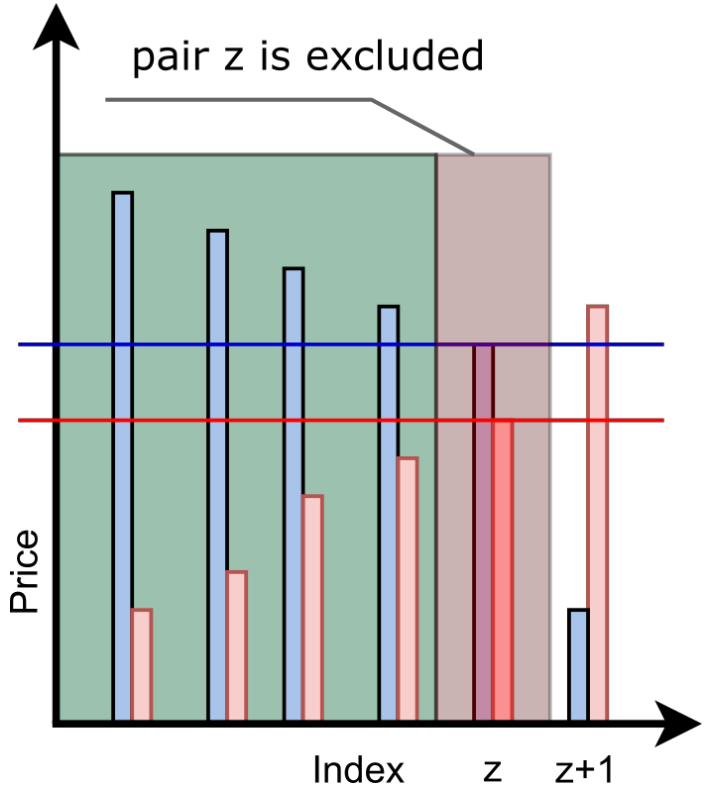
1) R.P. McAfee, A dominant strategy double auction. Journal of economic Theory 56.2 (1992)

McAfee's Mechanism

100% of optimal welfare achieved



Trade reduction performed

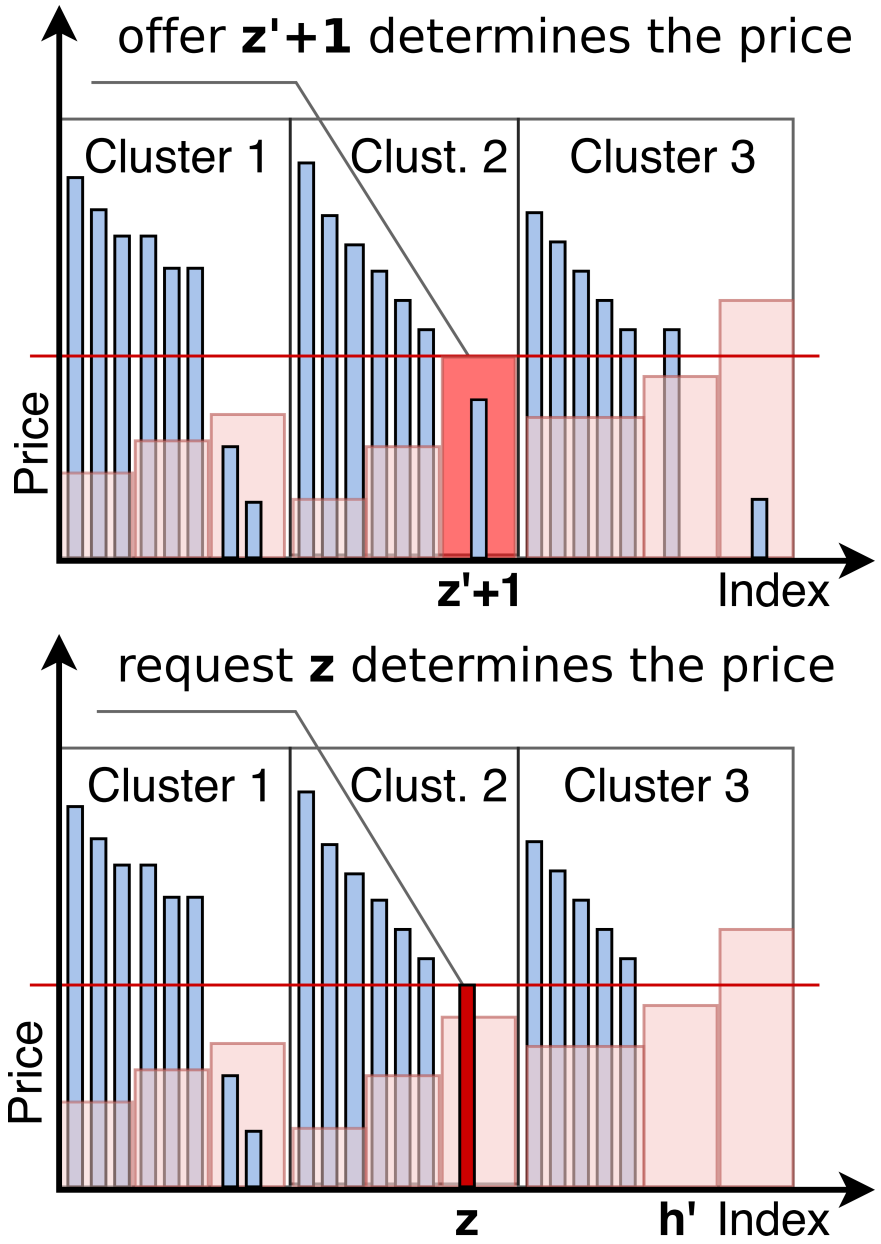


DeCloud Double Action Challenges

- Complex environment
  - In cloud auctions, bidders might misreport not only their valuations, but also hardware requirements or delay bid submission to get better payoff
- Goods are not of single type
  - Moreover, in DeCloud there are no discrete types of goods
- Buyers and sellers do not necessarily form pairs
  - One seller may serve multiple buyers
- How to minimize negative effect of trade reduction?

DeCloud Mechanism Design

- We group offers and requests together into **clusters** using our gravity-like matching heuristics
- In each cluster there are requests and offers which are the best match for each other
- To minimize negative effect of trade reduction we group price-compatible clusters in **mini-auctions**
- In mini-auction, only one cluster determines the price and potentially suffers from trade reduction

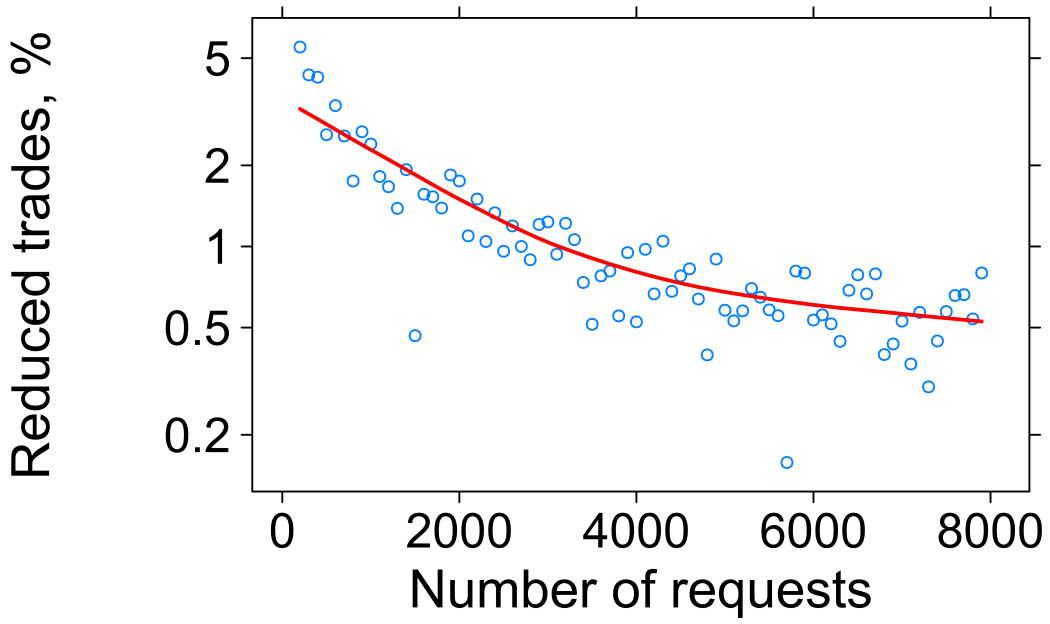


Measuring the Performance

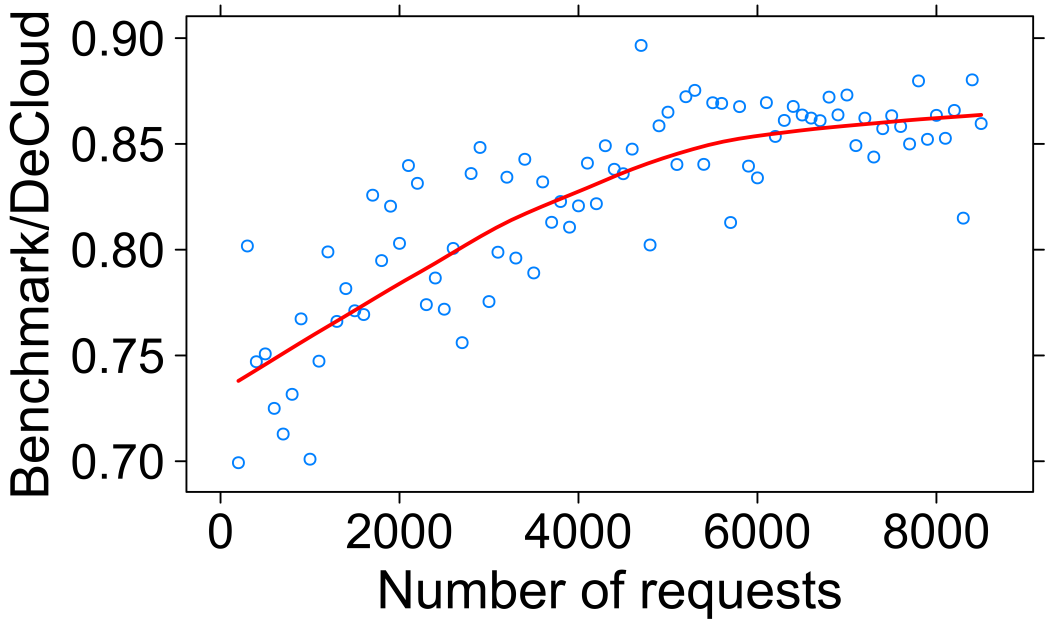
- Google Cluster Data to emulate requests of clients
- Amazon EC2 M5 instance types as providers
- For costs and valuations we used randomized Amazon EC2 costs
- Truthfulness (DSIC property) affects welfare negatively:
  - We exclude requests or offers in the case they define price
  - We use pseudo randomness in the case there are not enough requests to allocate all the valid (by price) offers (or visa versa).
- How much we loose if we would just assume truthful bids and not take any measures to make truthful bidding the dominant strategy?

Results: Welfare

As size of the market grows, the fraction of reduced trades becomes marginal

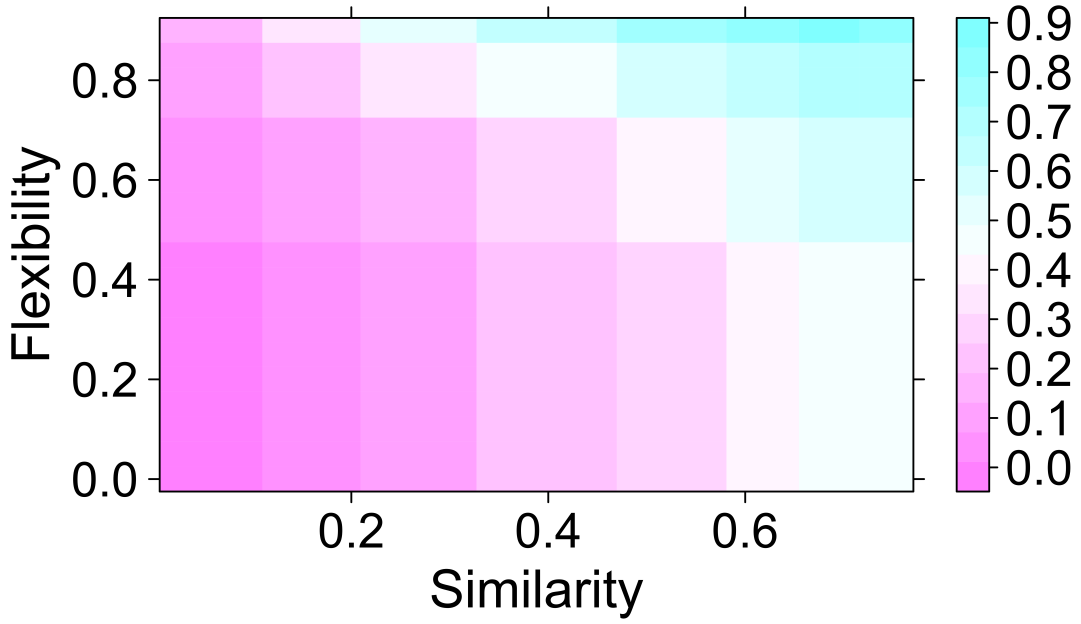


DeCloud approaches its non-DSIC benchmark given enough participants

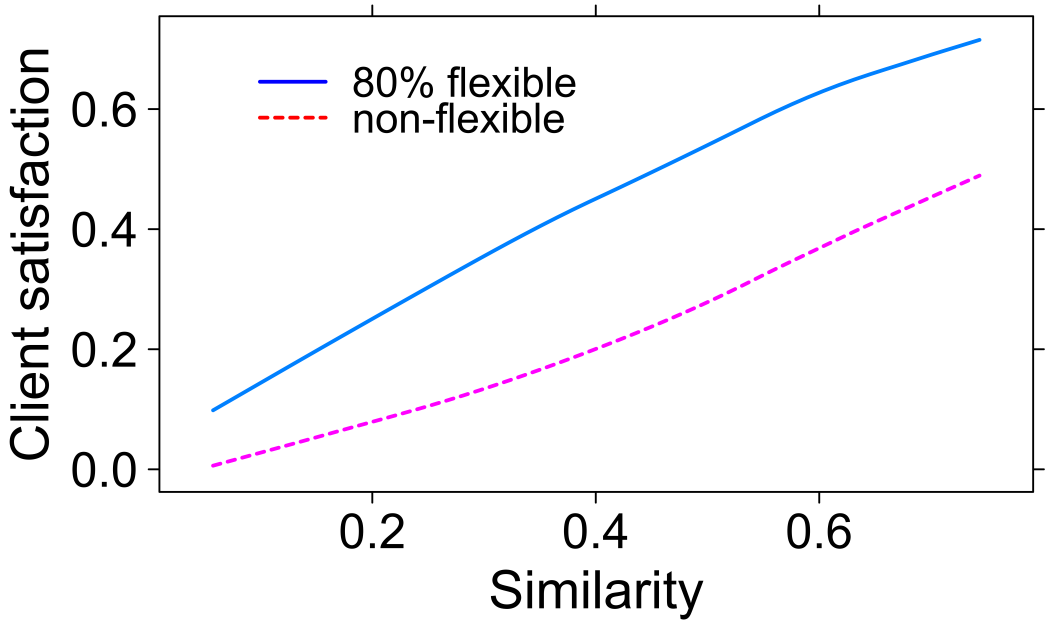


Results: Client Satisfaction

Client satisfaction heatmap



Client satisfaction with fixed flexibility



Client satisfaction indicates the fraction of the accepted clients

Summary and Potential

- Our contribution: DeCloud, truthful double auction with no auctioneer
- Tackling demand for edge resources with open crowdsourced (and not only) environment
- Distributed auctions do not have to be limited to edge/cloud computing
- There is a potential for removing the **middleman** and minimizing the costs also in the other areas, involving crowdsourcing or not

Thank you!

aleksandr.zavodovski@helsinki.fi