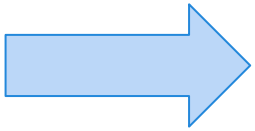
# Advertising on the Web

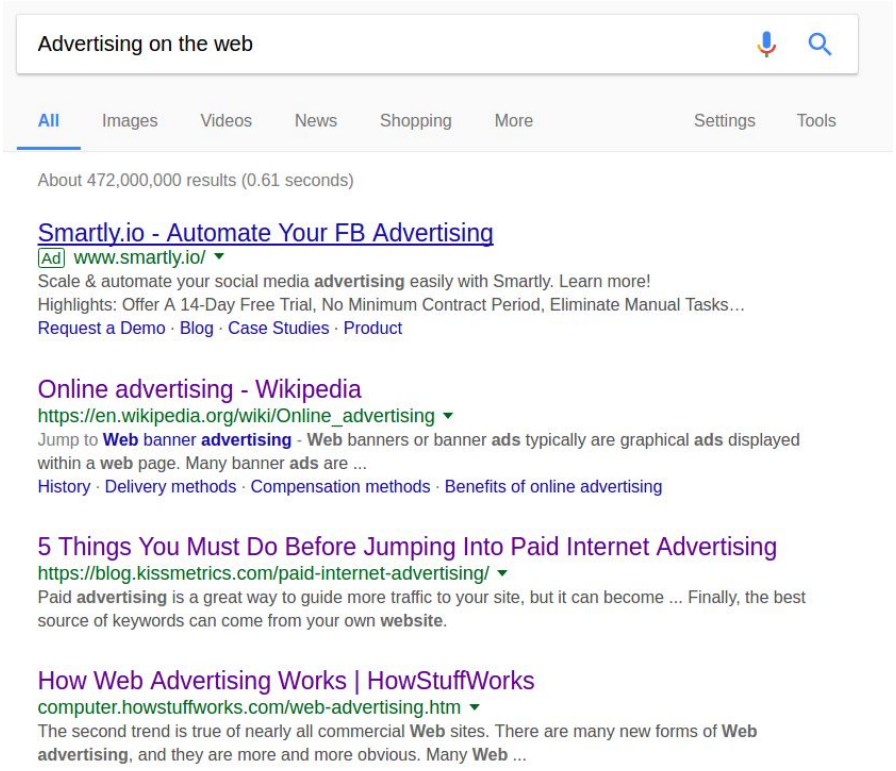
2017-03-18

# Advertisement

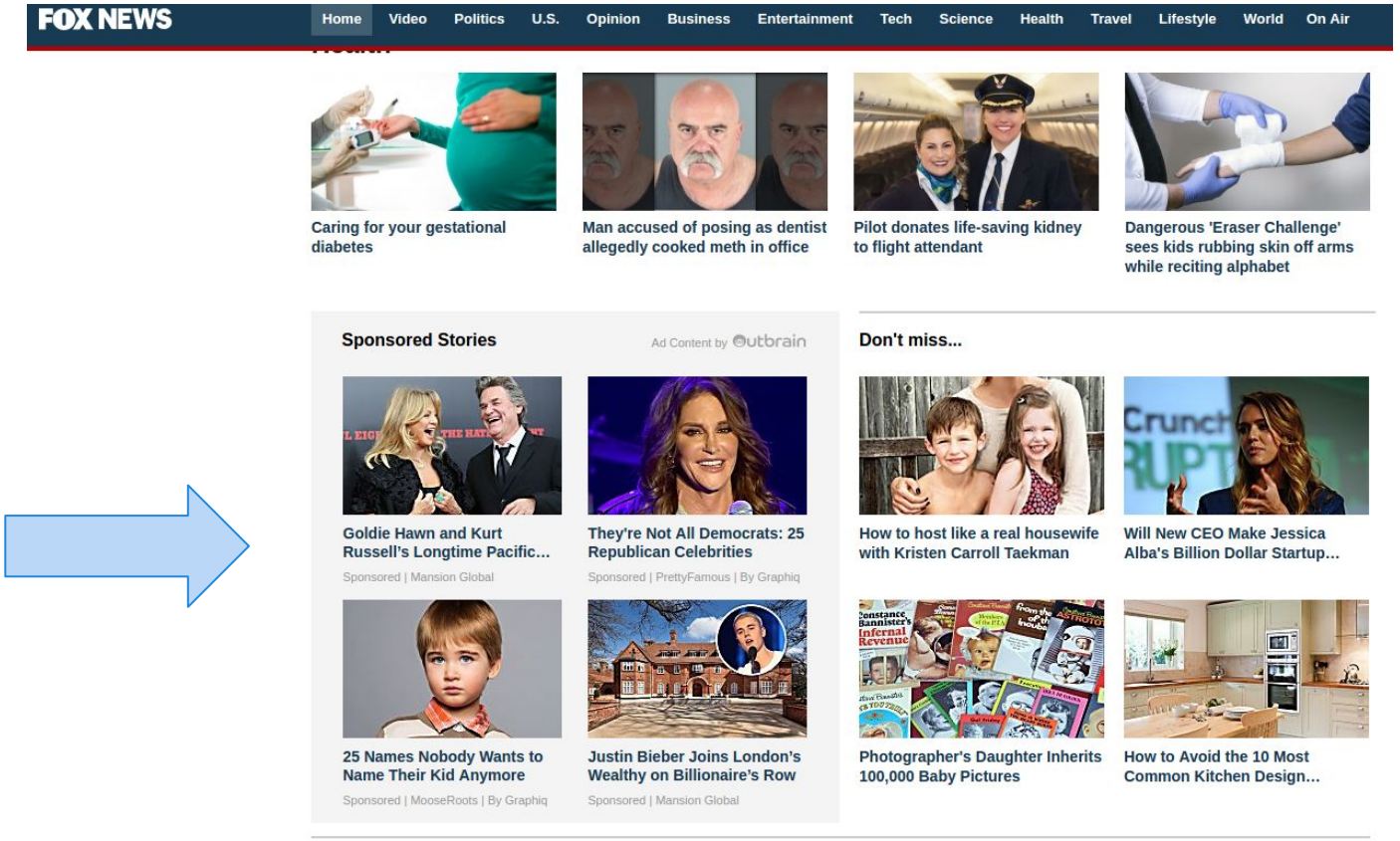
*“a notice or announcement in a public medium promoting a product, service, or event or publicizing a job vacancy.”*

# Where can I see it?

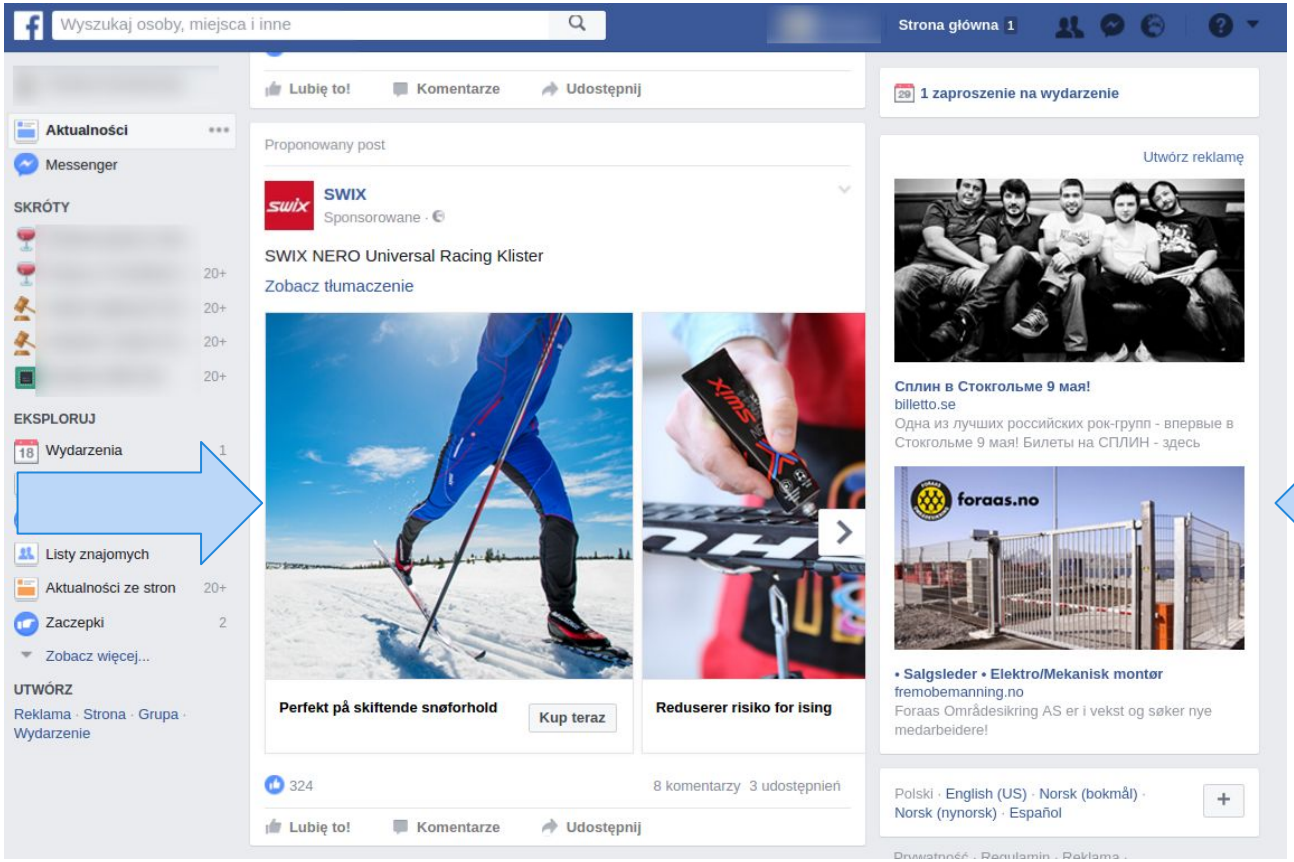




# Where can I see it?



# Where can I see it?



# Where can I see it?

- does not matter what do you do
- in what language
- on which site

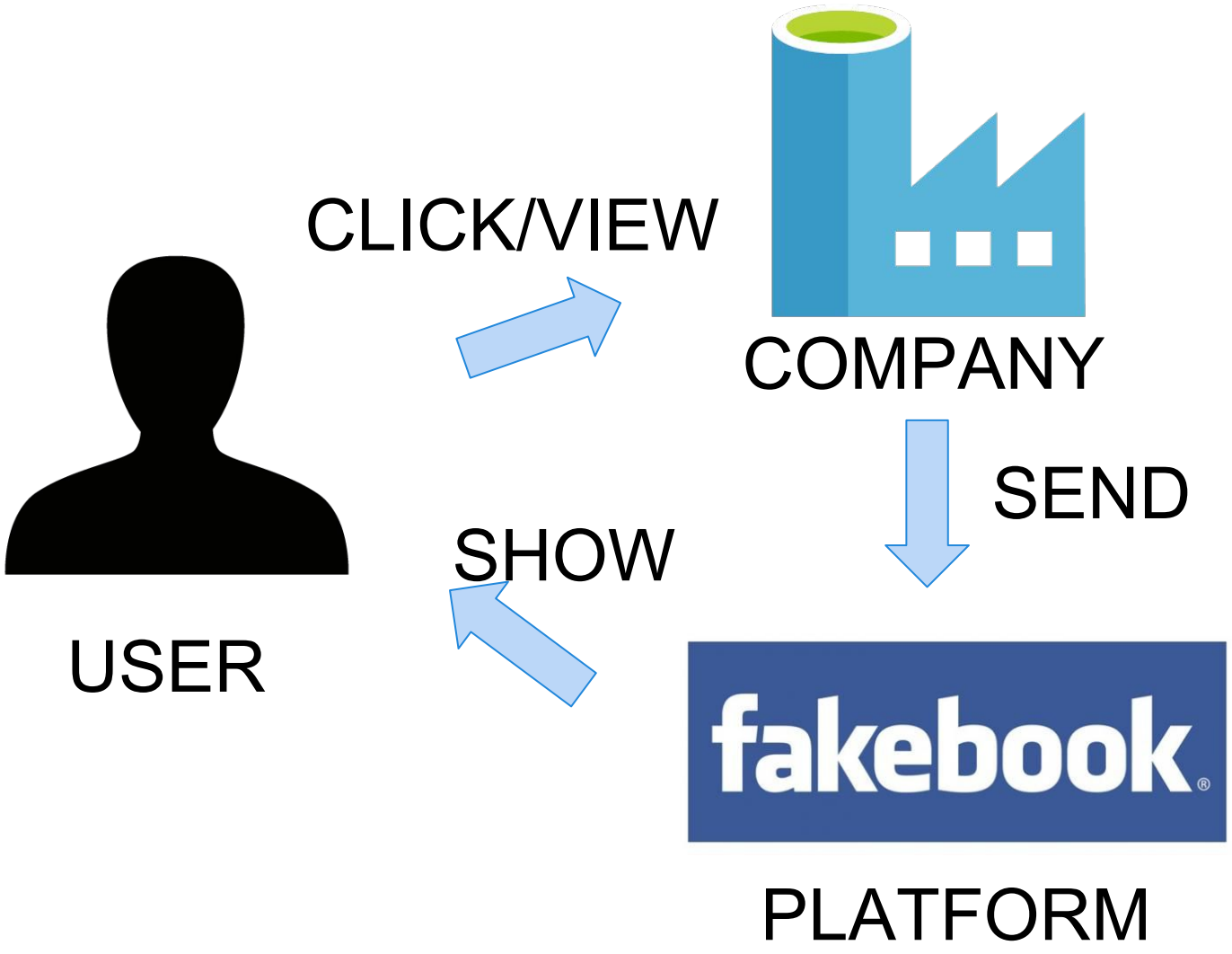
ads are

**EVERYWHERE**

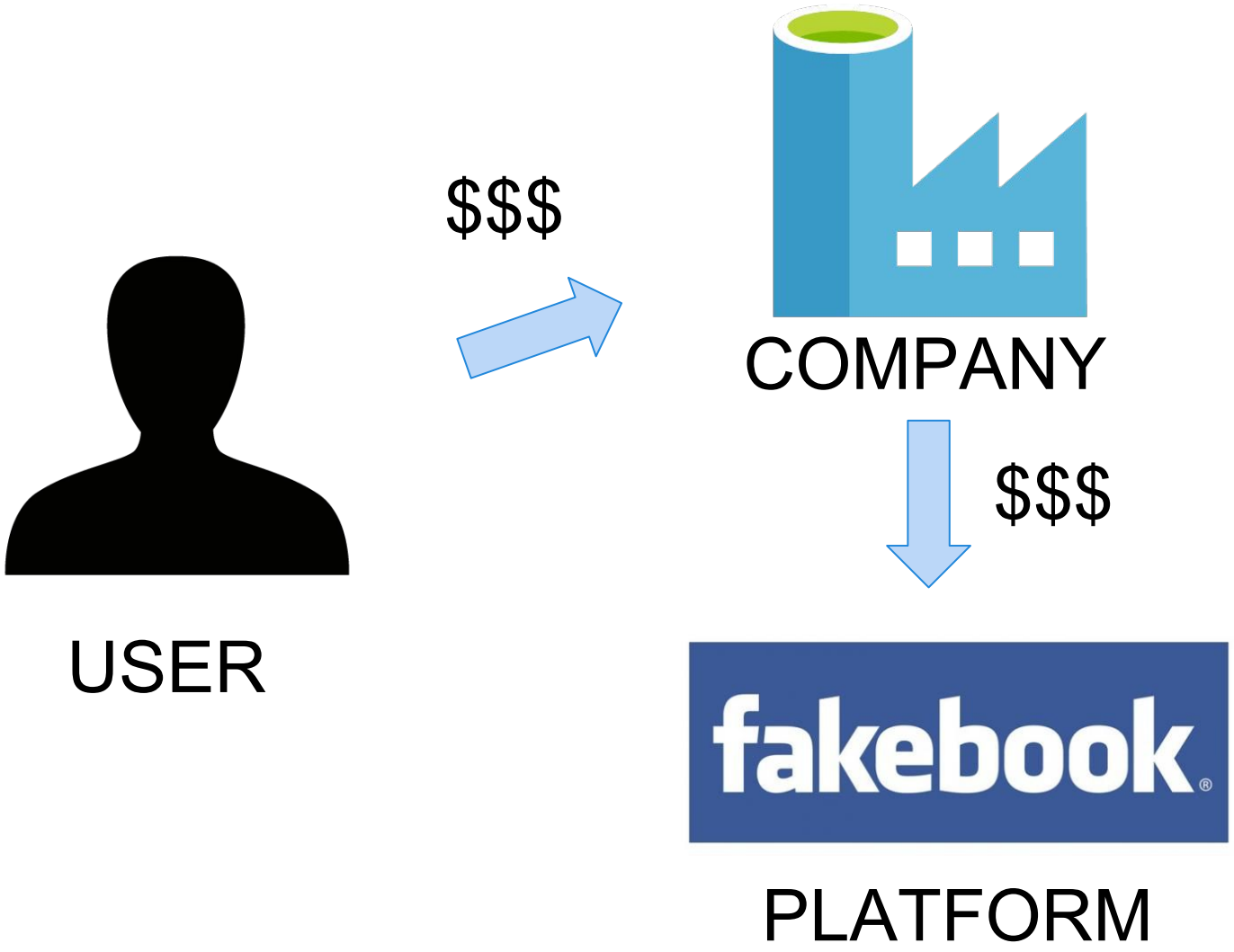
# Why?

**Google's total advertising revenues of USD $43.7 billion in 2012**

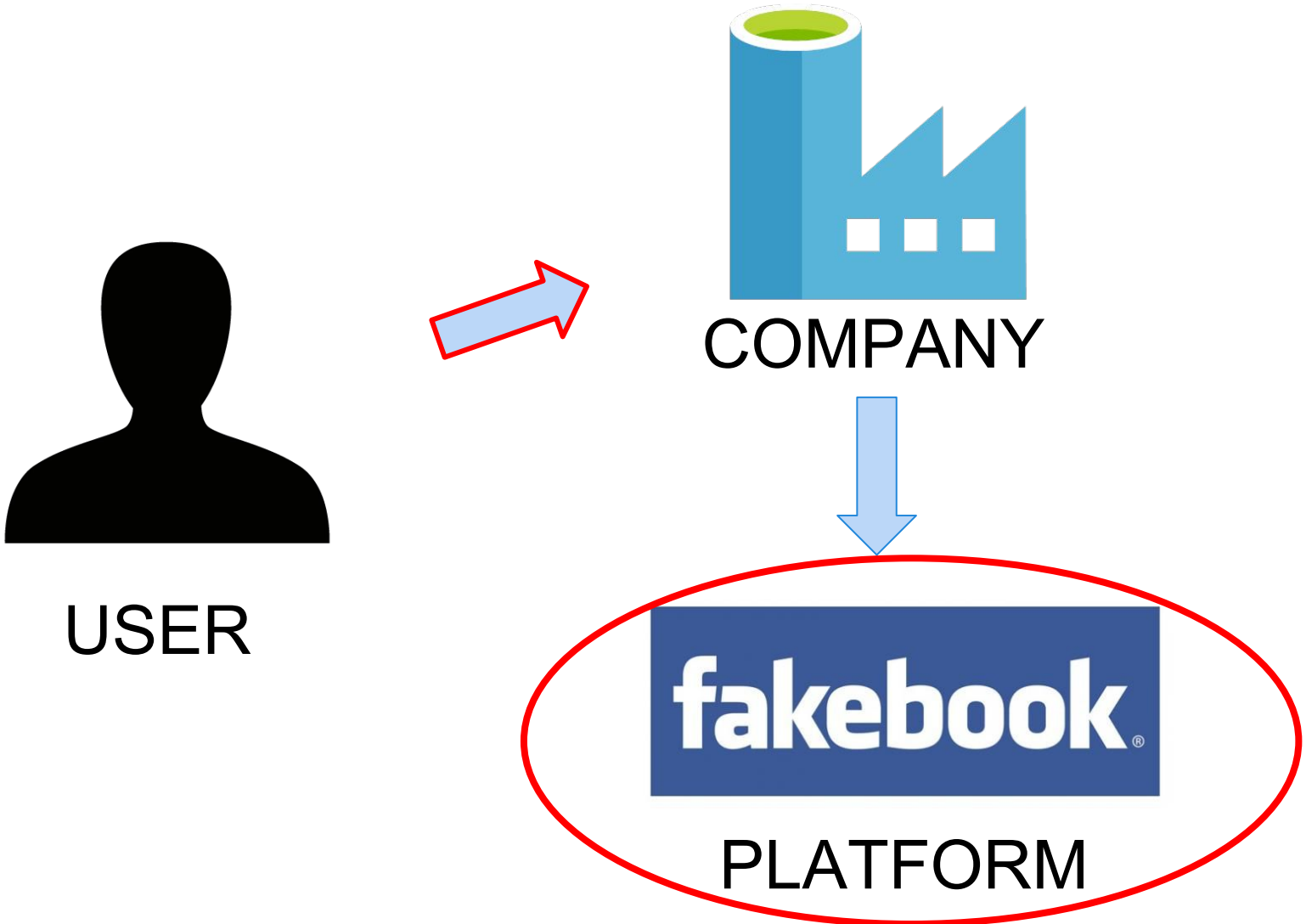
# Setting: three sides - advert flow



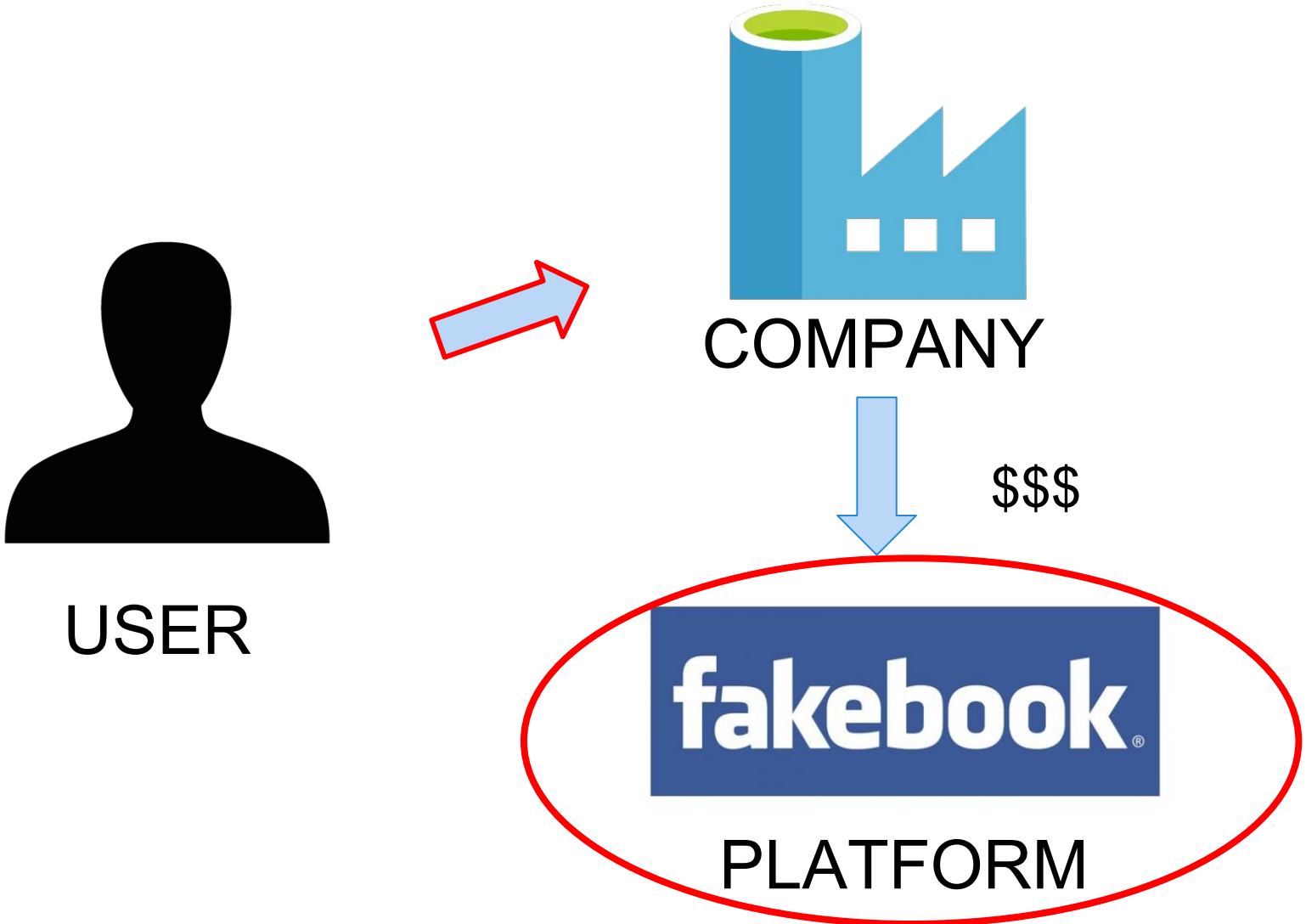
# Setting: three sides - money flow



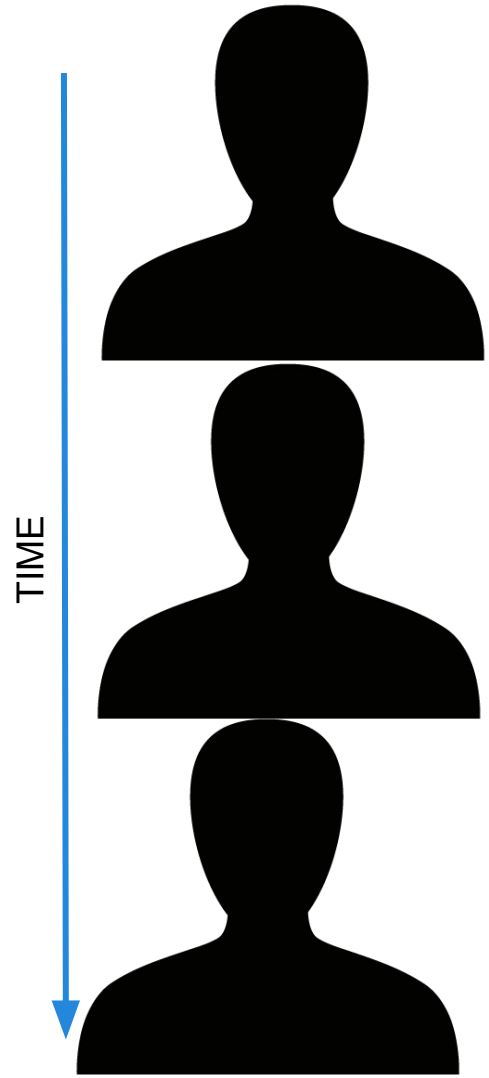
# Setting: Where are we?

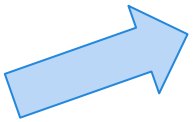


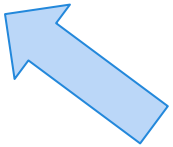
# Setting: What do we want?



# Setting: We forgot about TIME





fakebook®

PLATFORM

# Revenue

**How much do we (as a platform) earn from our matching?:**

- **per impression**
- **per click**

# Adwords model

- **Given:**
  - **1.** A set of bids by advertisers for search queries
  - **2.** A click-through rate for each advertiser-query pair
  - **3.** A budget for each advertiser (say for 1 month)
  - **4.** A limit on the number of ads to be displayed with each search query
- **Respond to each search query with a set of advertisers such that:**
  - **1.** The size of the set is no larger than the limit on the number of ads per query
  - **2.** Each advertiser has bid on the search query
  - **3.** Each advertiser has enough budget left to pay for the ad if it is clicked upon

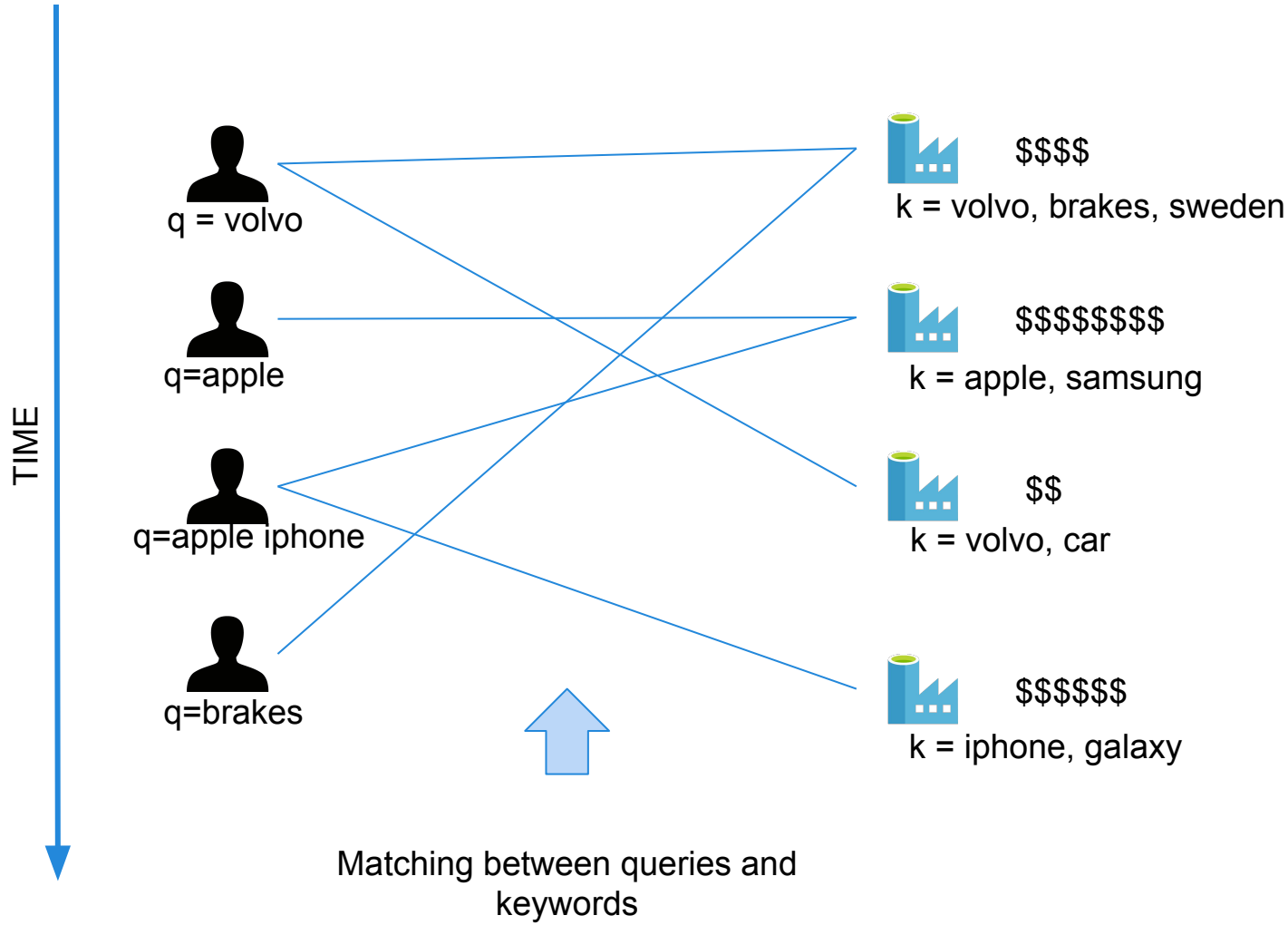
# Only ~1 impression per 100 is clicked: CTR

| Advertiser | Bid | CTR | Bid * CTR |
|---|---|---|---|
| A | $1.00 | 1% | 1 cent |
| B | $0.75 | 2% | 1.5 cents |
| C | $0.50 | 2.5% | 1.125 cents |
| | | Click through rate | Expected revenue |

# Simplifications

- Each user has one advert slot
  - If he has more we can just repeat the ad selection for each of them sequentially
- Every advert has the same expected revenue

# Adwords model



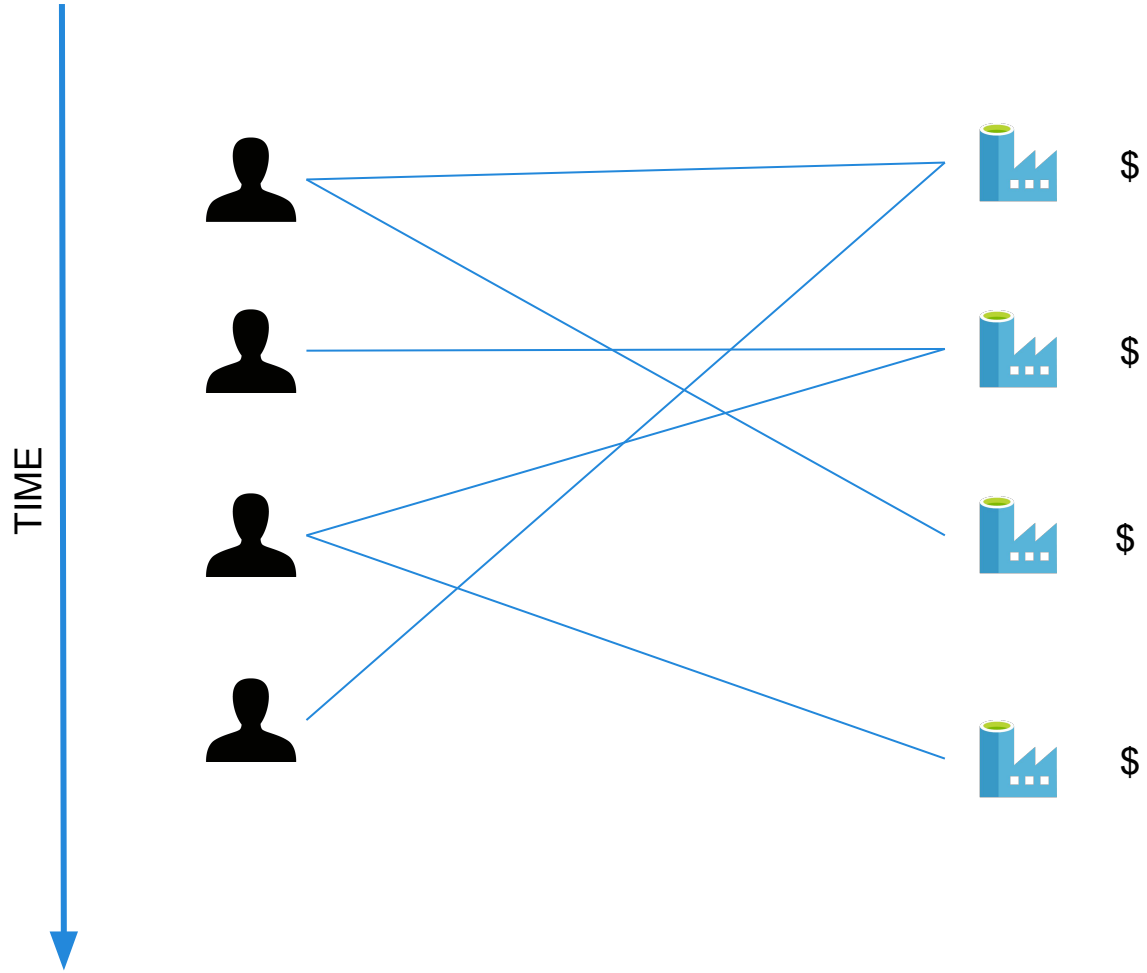
# Greedy algorithm

**For a query pick any advertiser who has bid for that query.**

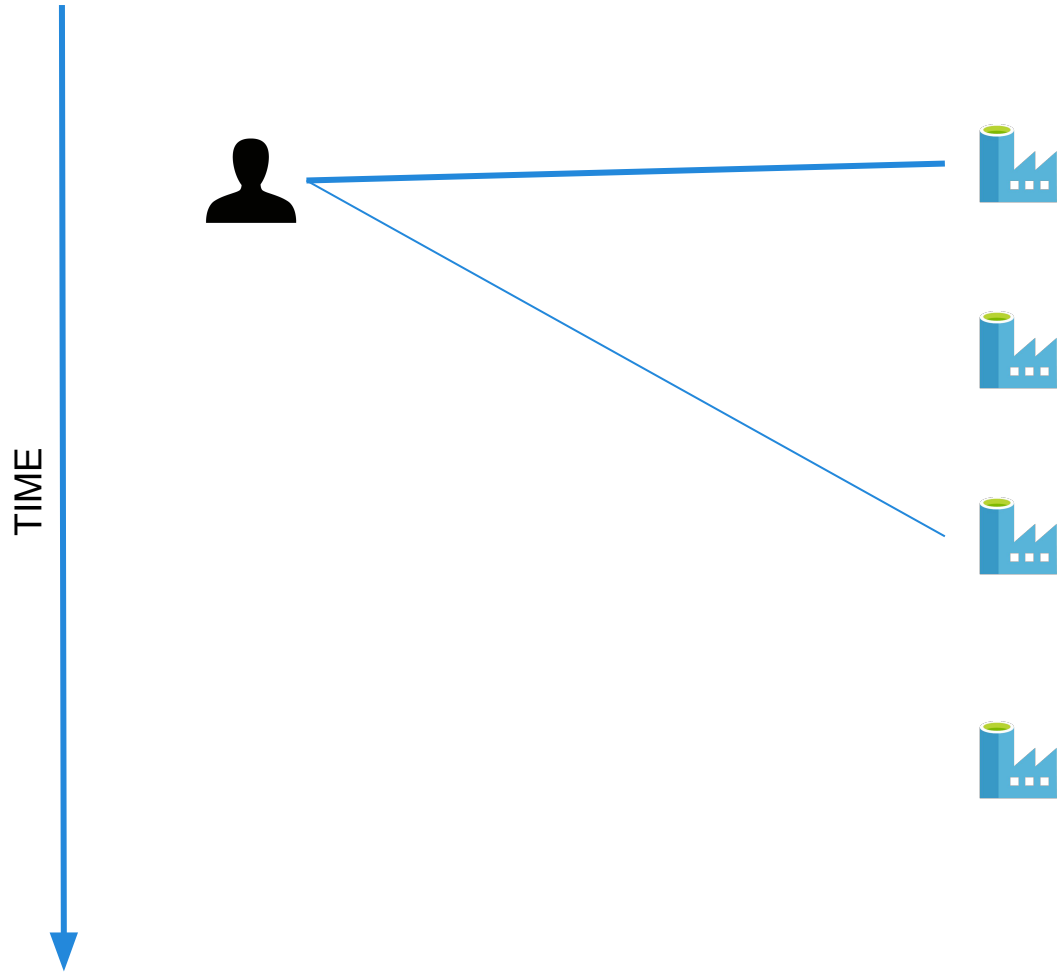
(Few possibilities in when there are several bidders.)

# Simple Greedy with everybody having 1$

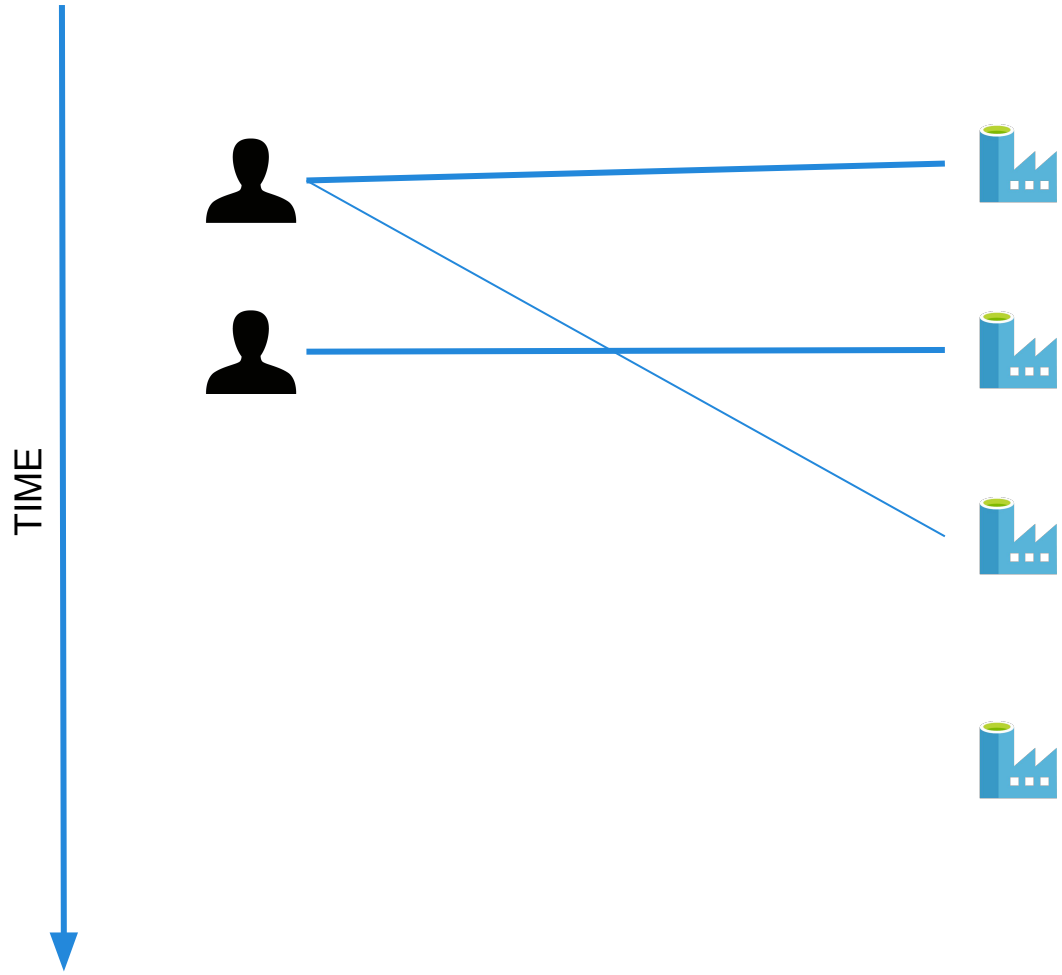
# Matching 1: everybody has 1$



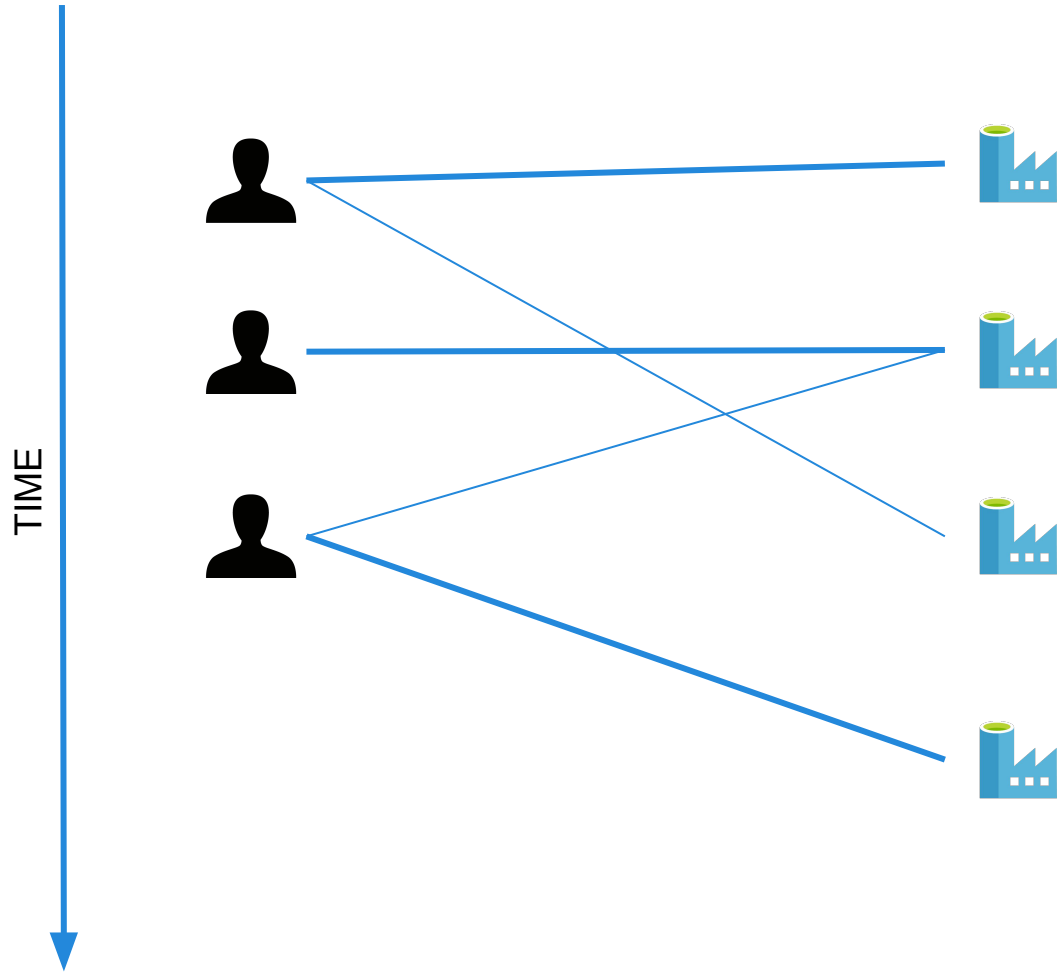
# Online Greedy Matching



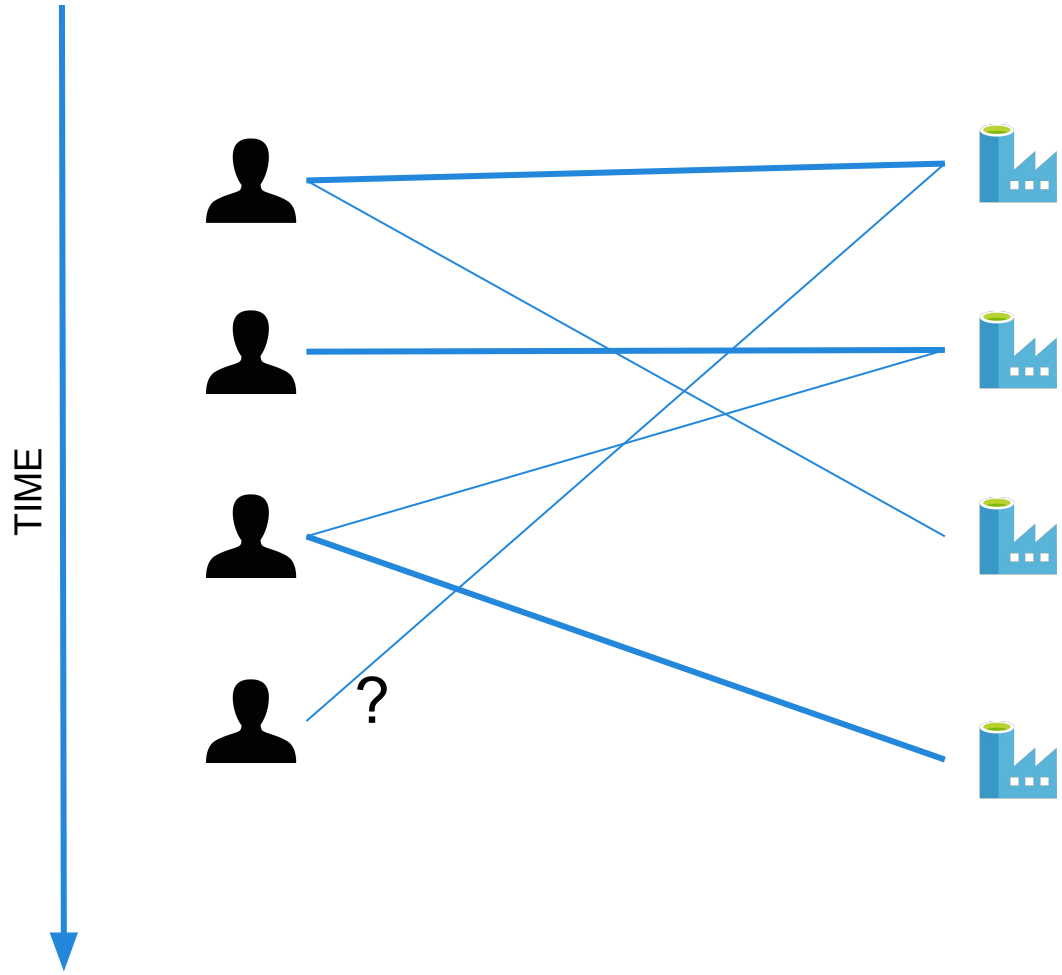
# Online Greedy Matching



# Online Greedy Matching

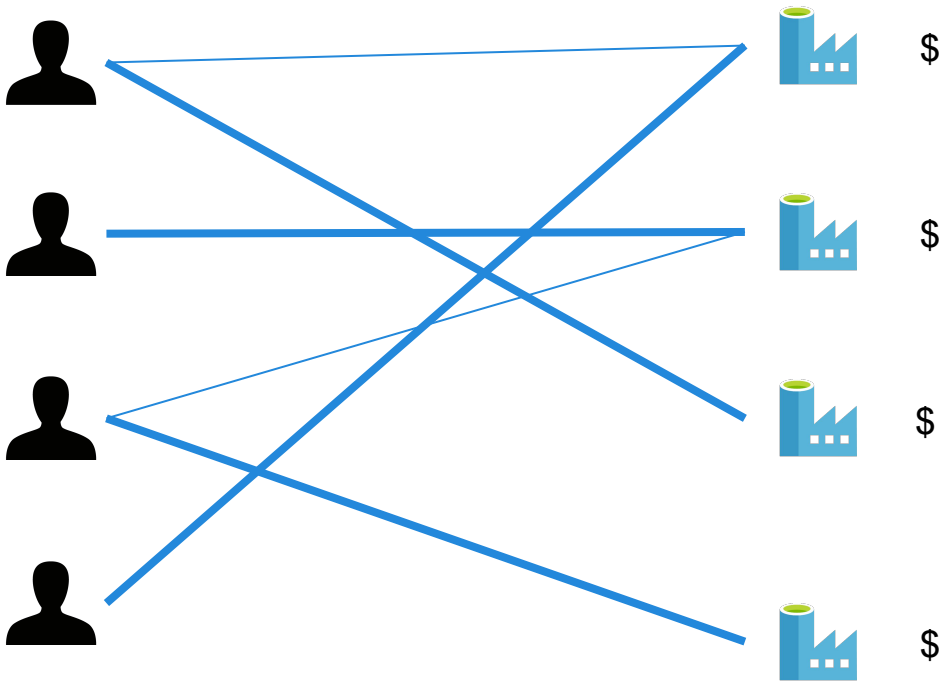


# Online Greedy Matching

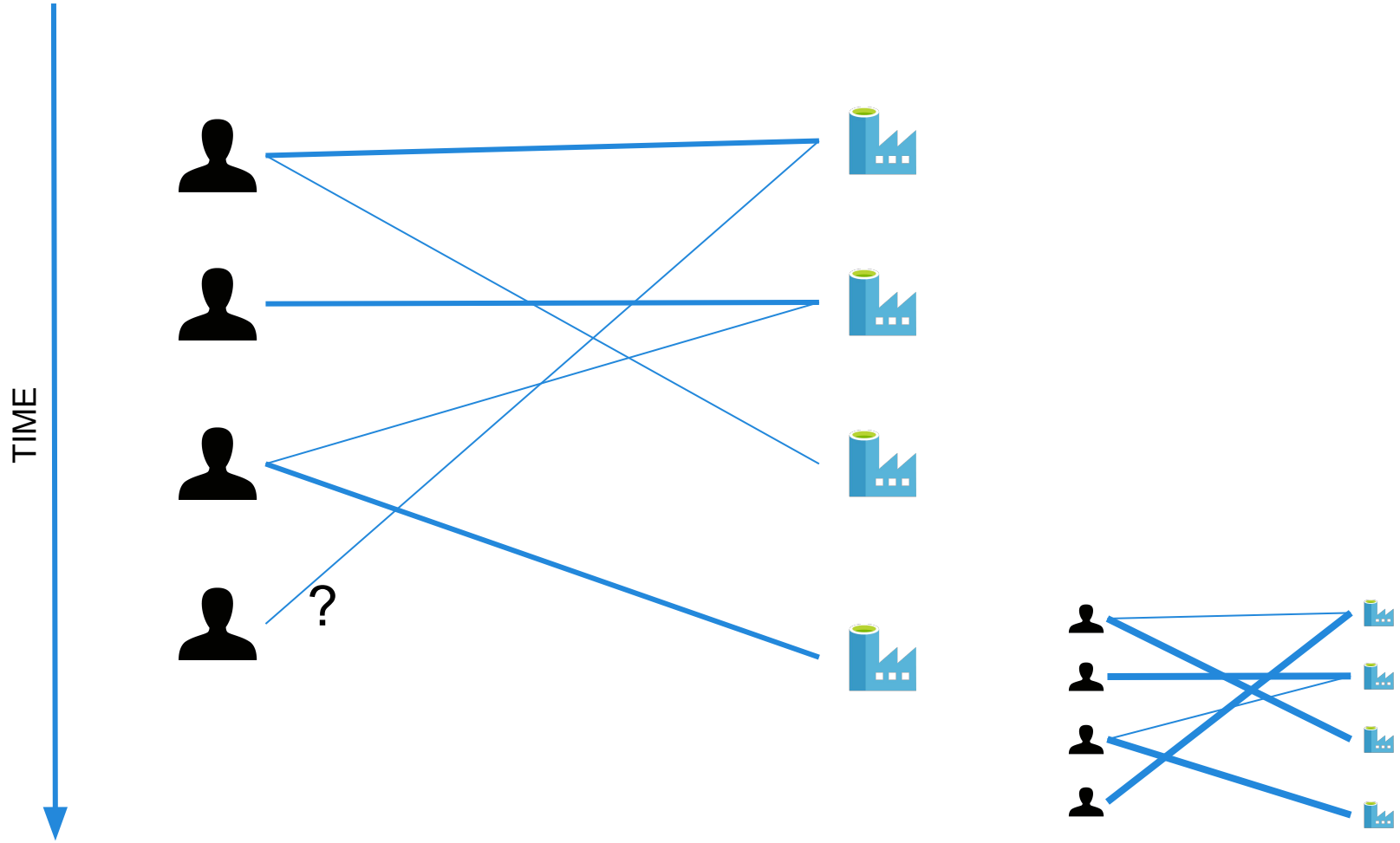


# What’s the perfect matching?

# Perfect Matching



# Online Greedy Matching



# Greedy with 2 companies (bidders) with 3$

Remark: on plots we skip lines representing possible matchings (that can be inferred from keywords overlapping)

# Adwords (simplified: k=1) model

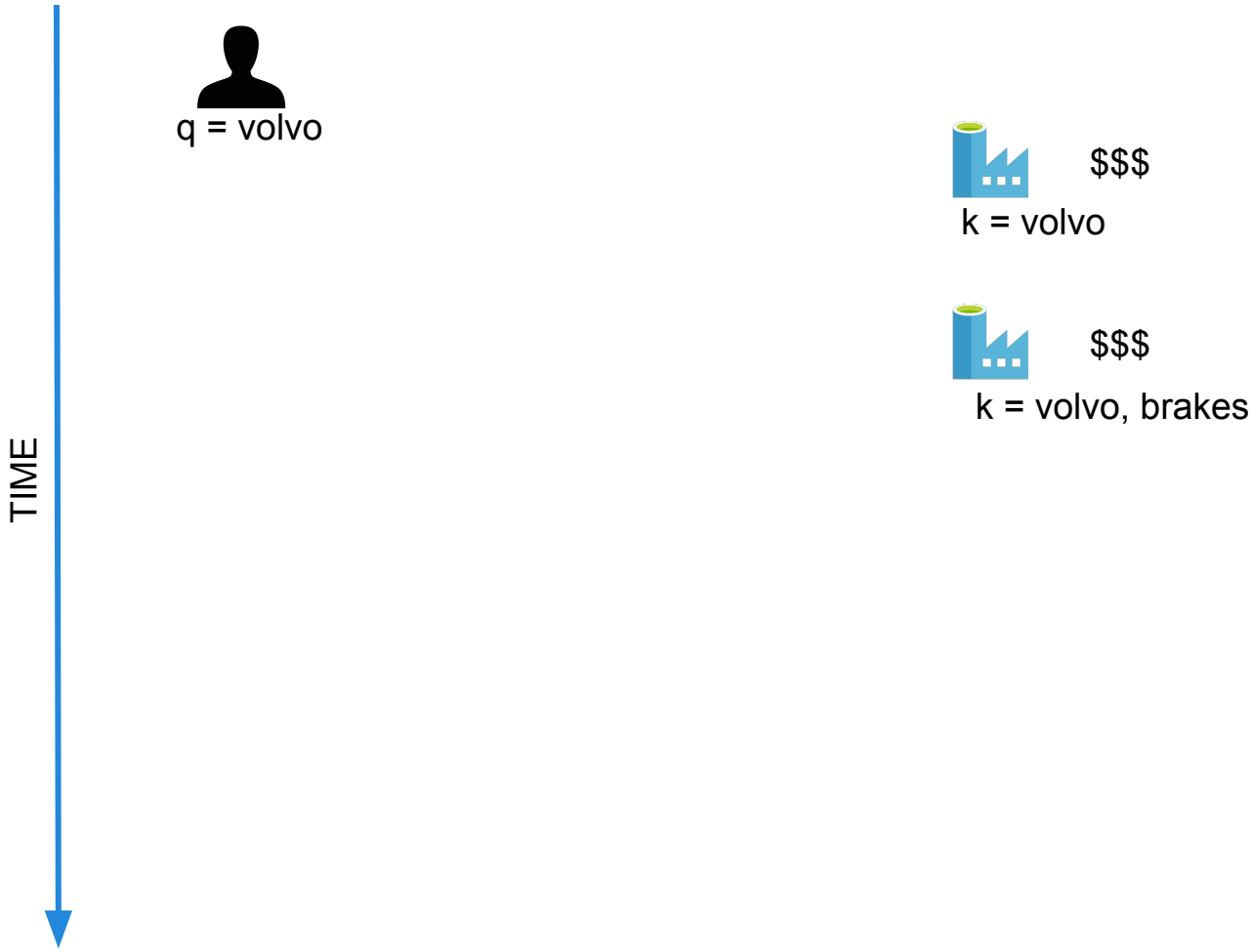

$$$

k = volvo

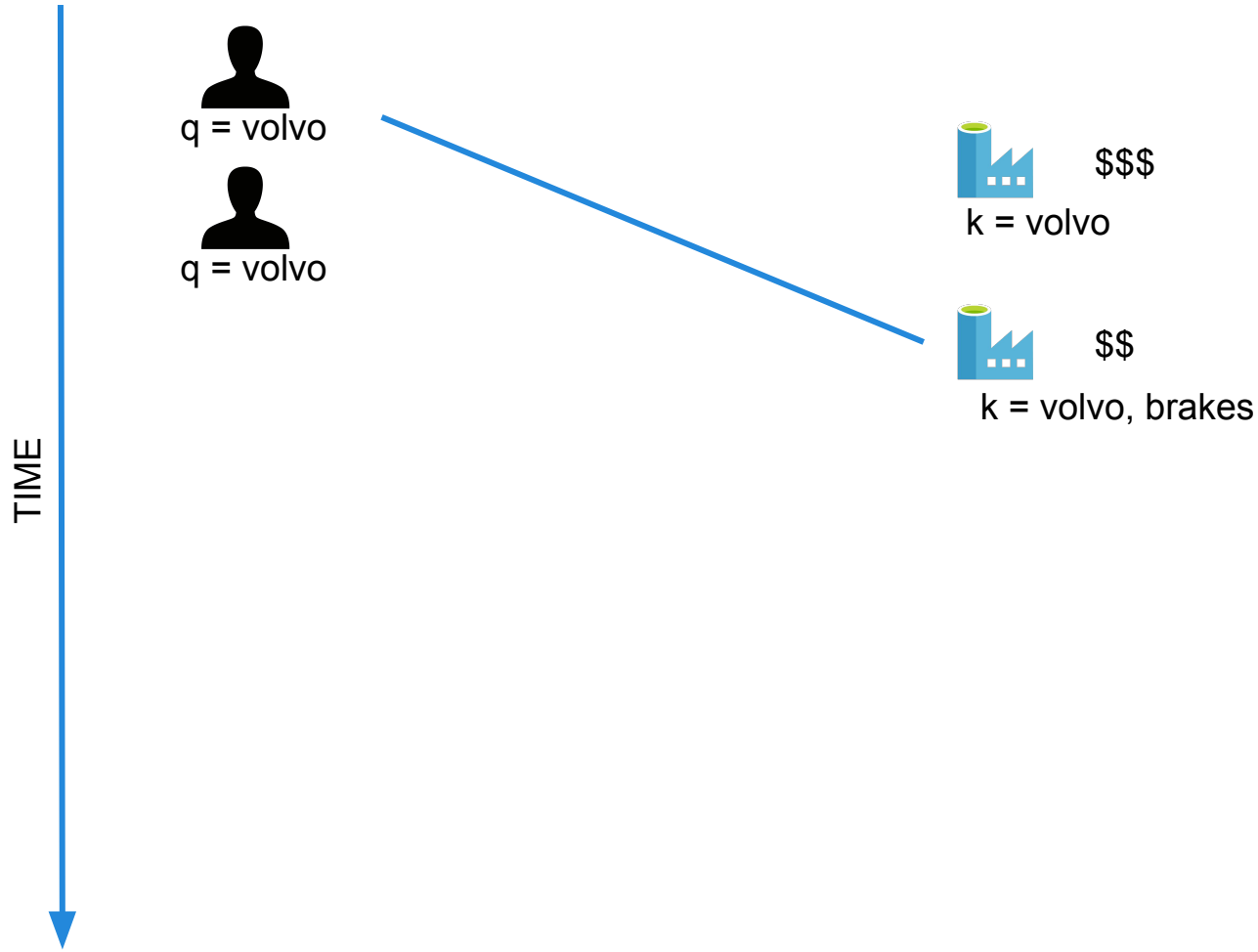
$$$

k = volvo, brakes

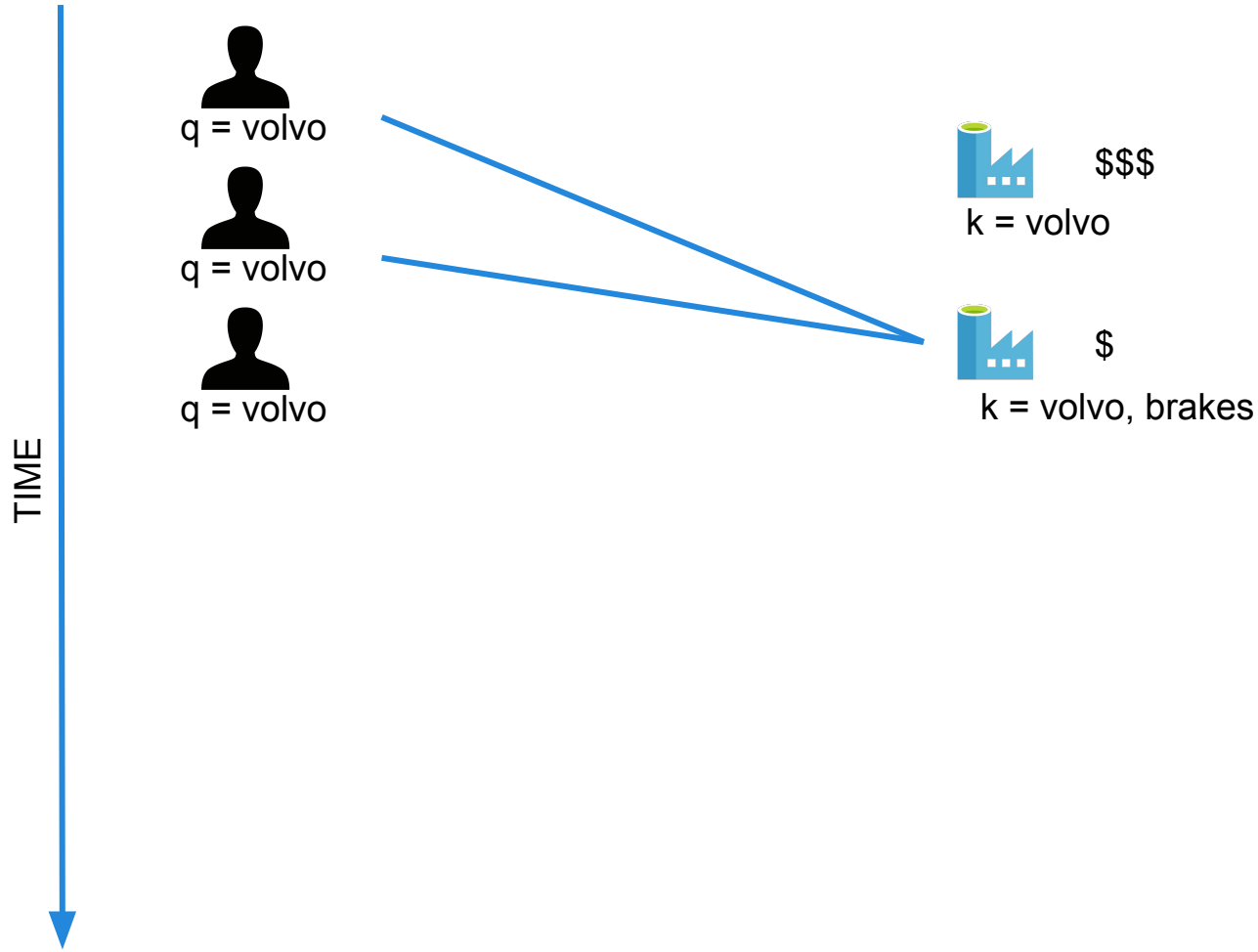
# Adwords (simplified: k=1) model



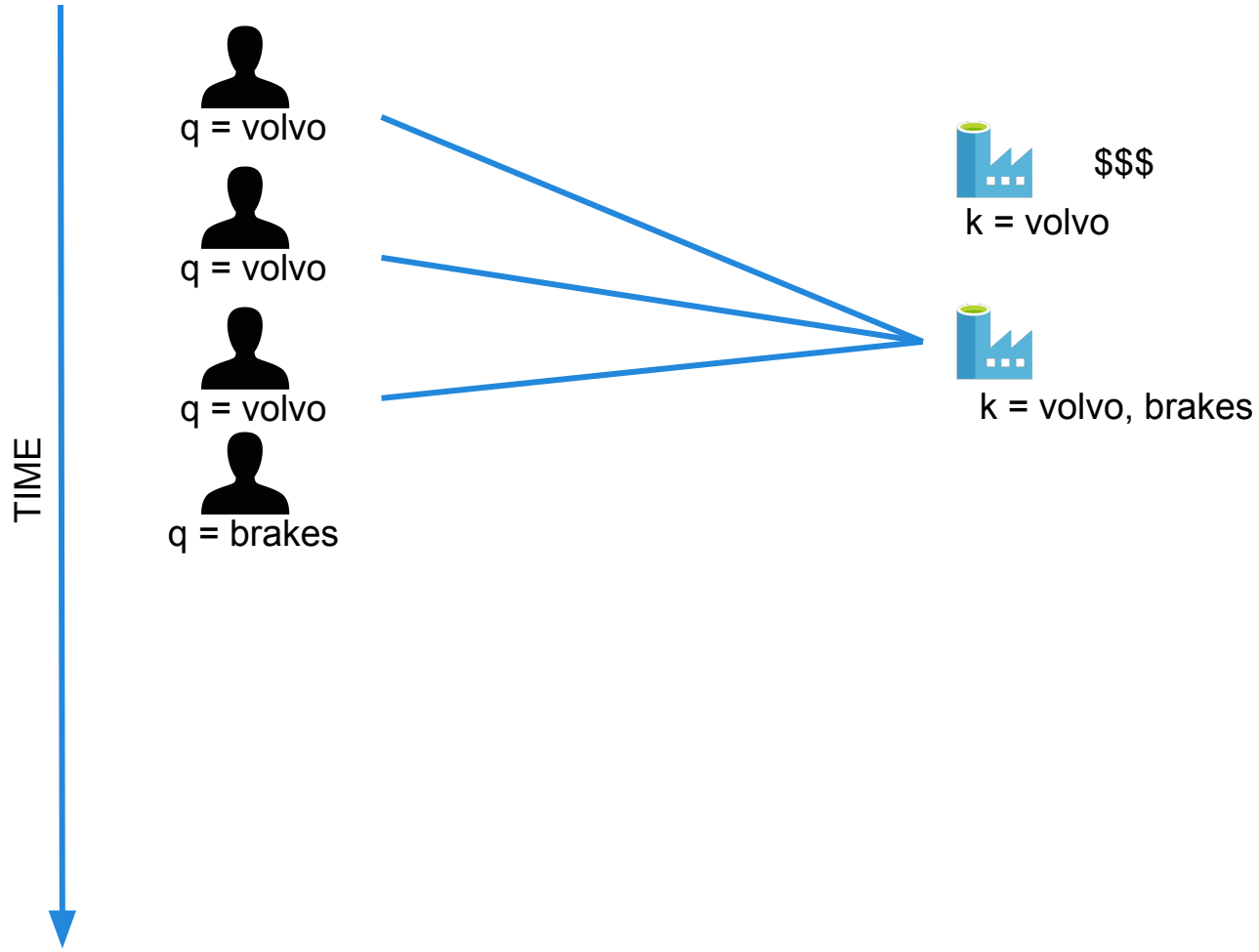
# Adwords (simplified: k=1) model



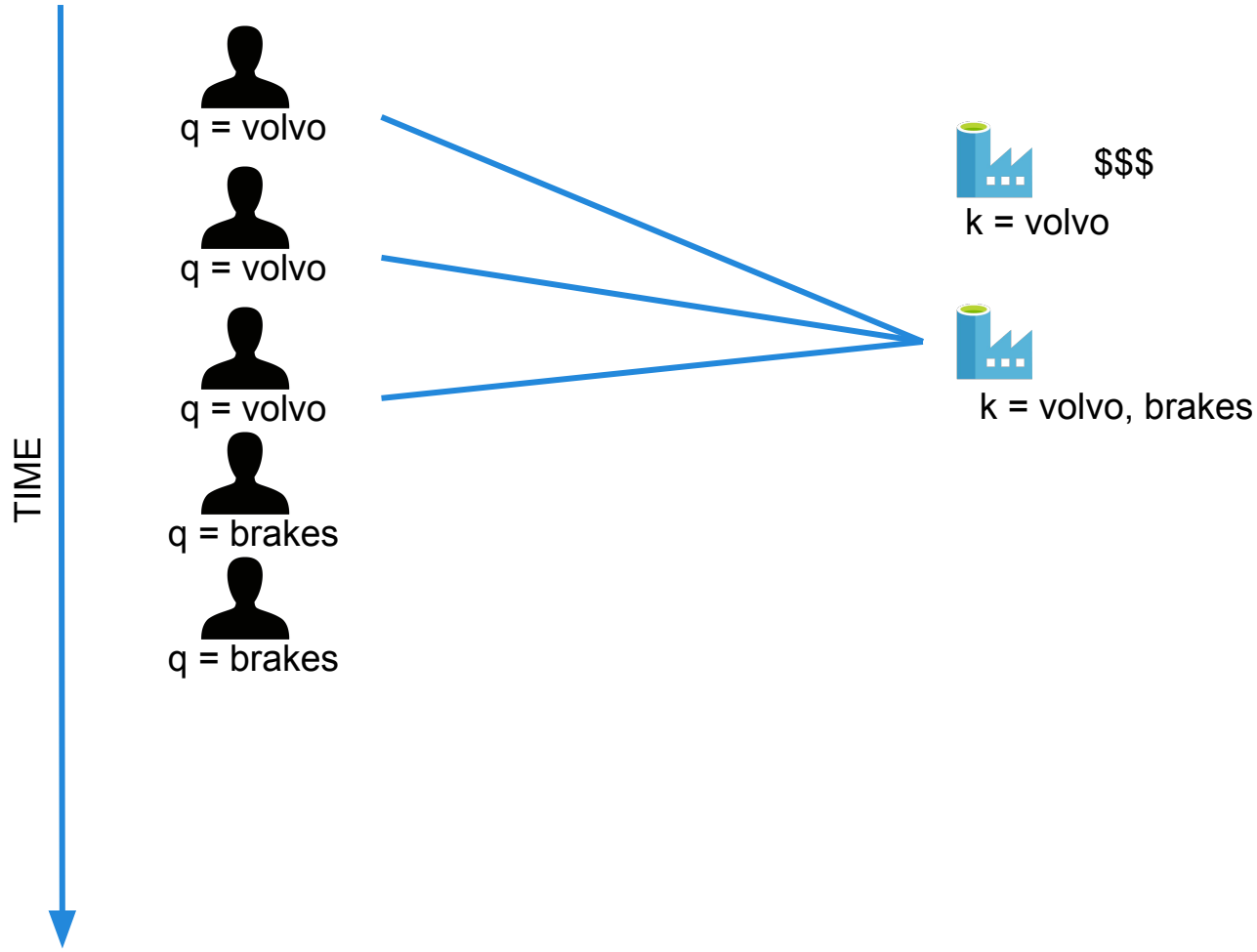
# Adwords (simplified: k=1) model



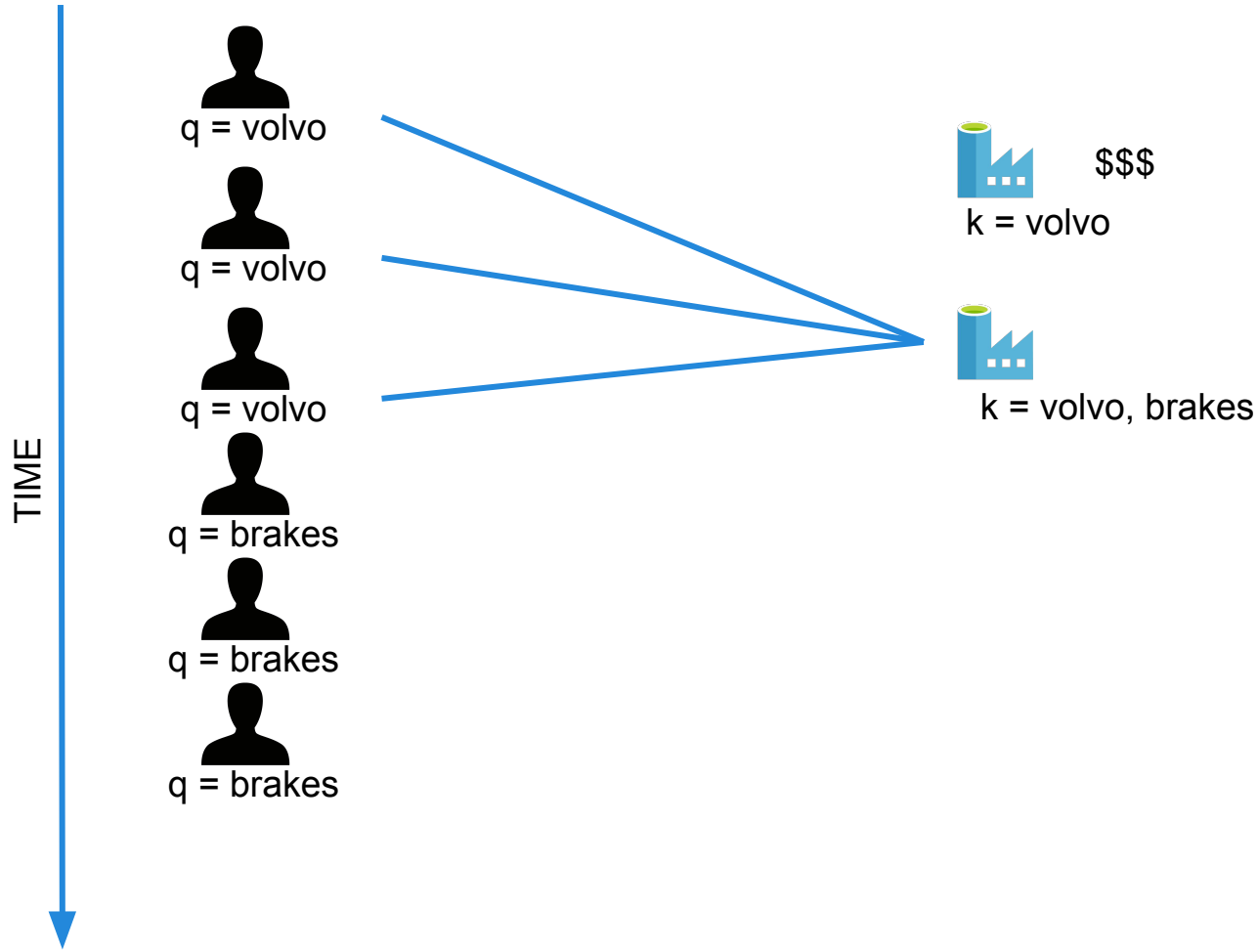
# Adwords (simplified: k=1) model



# Adwords (simplified: k=1) model

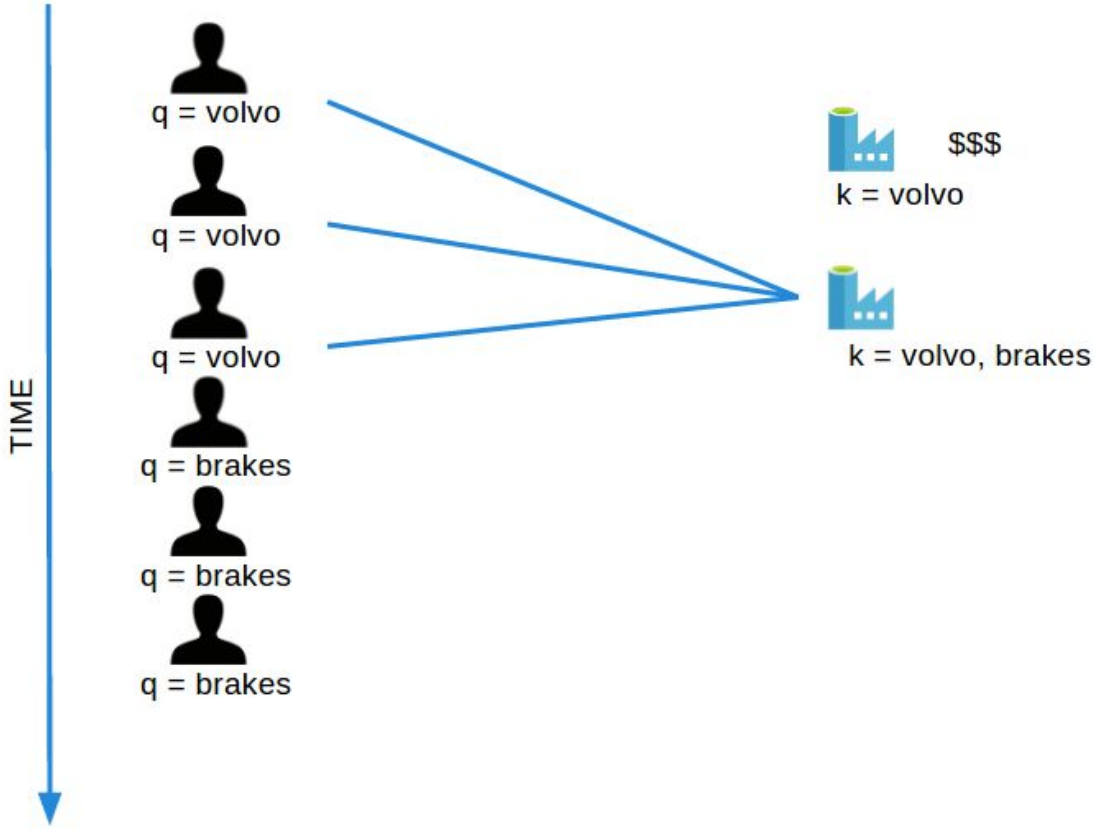


# Adwords (simplified: k=1) model



# Revenue

We could earn 6$ but we earned only 3$



# Revenue

**Better algorithm *M* ->**
**More matches ->**
**Higher Revenue**

# Competitiveness

- For input $I$, suppose greedy produces matching $M_{greedy}$ while an optimal matching is $M_{opt}$

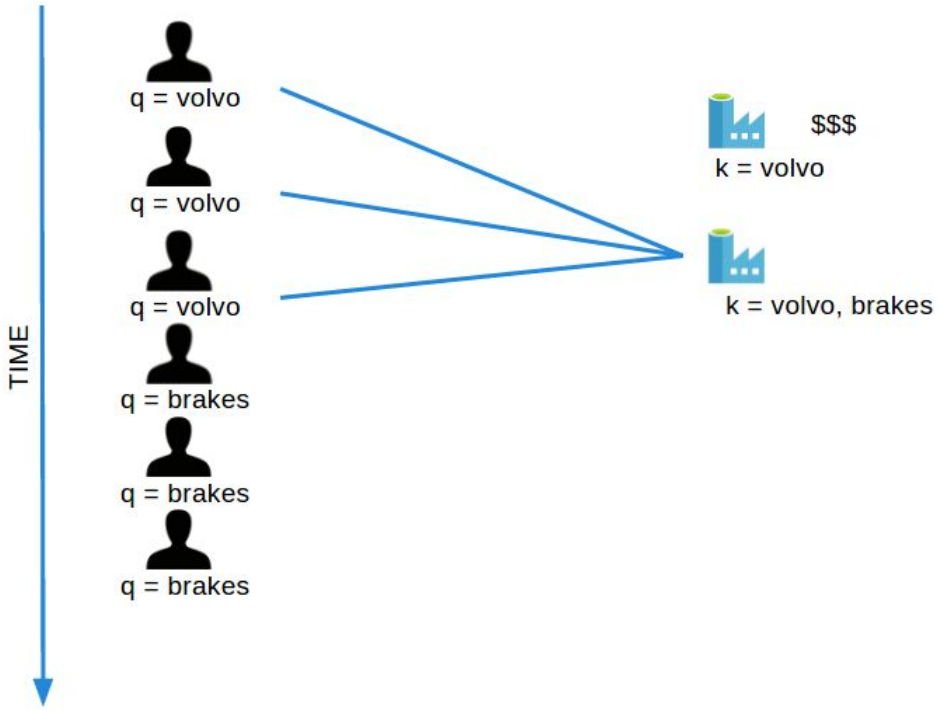
**Competitive ratio =**

$$min_{\text{all possible inputs } I} (|M_{greedy}|/|M_{opt}|)$$

(what is greedy's worst performance over all possible inputs $I$)

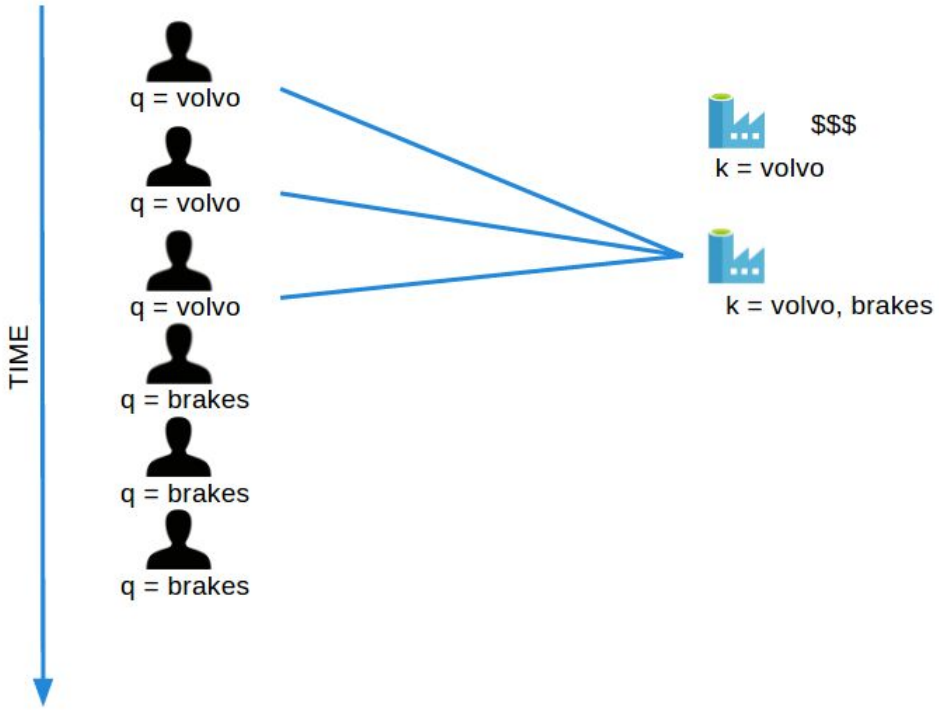
# Competitiveness

For this data we matched 3/6 = 1/2

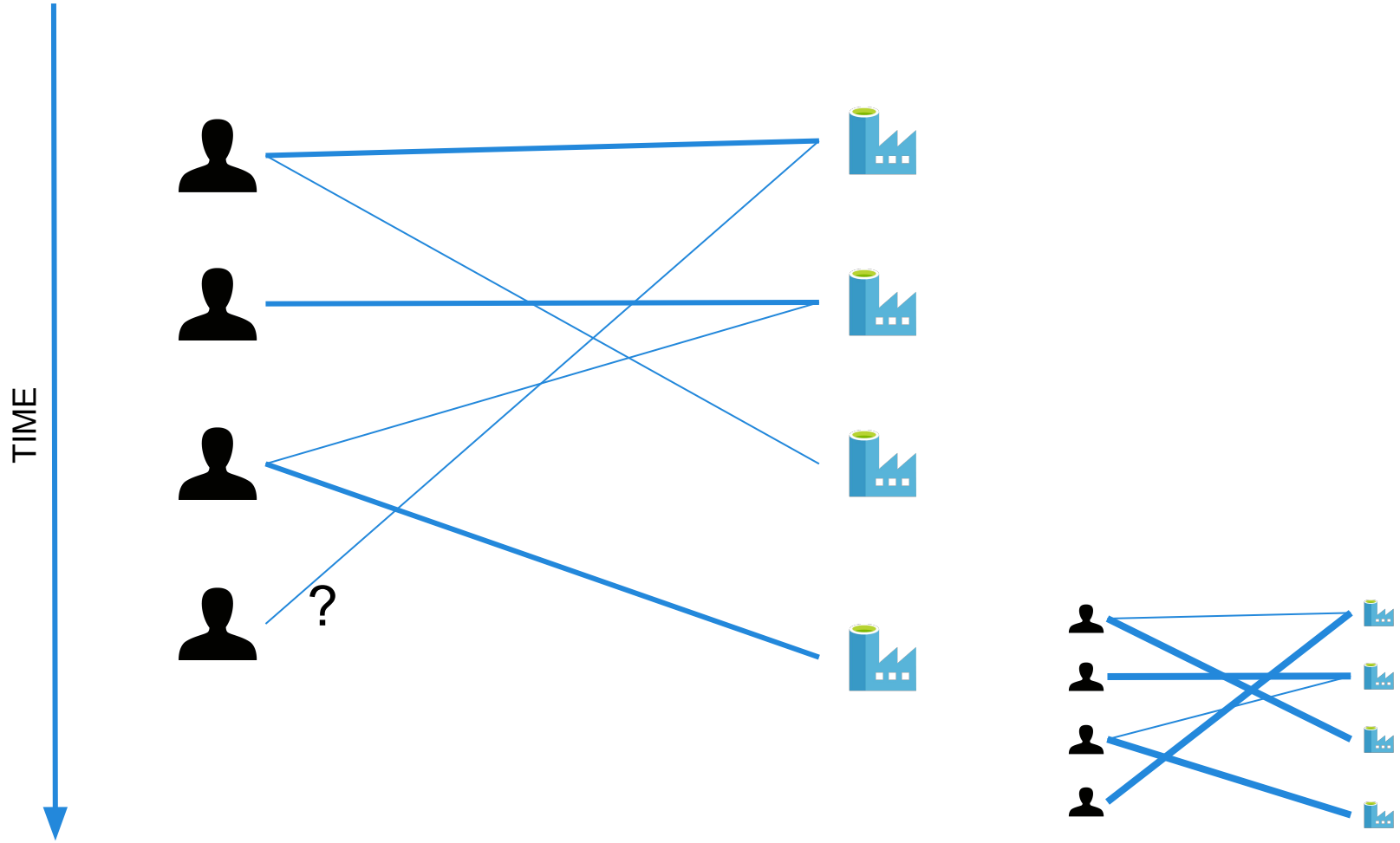


# Competitiveness

Actually it's the worst scenario: (under our simplifications) competitive ratio for the greedy algorithm(s) is 1/2 (in  general, not only here)



# Online Greedy Matching

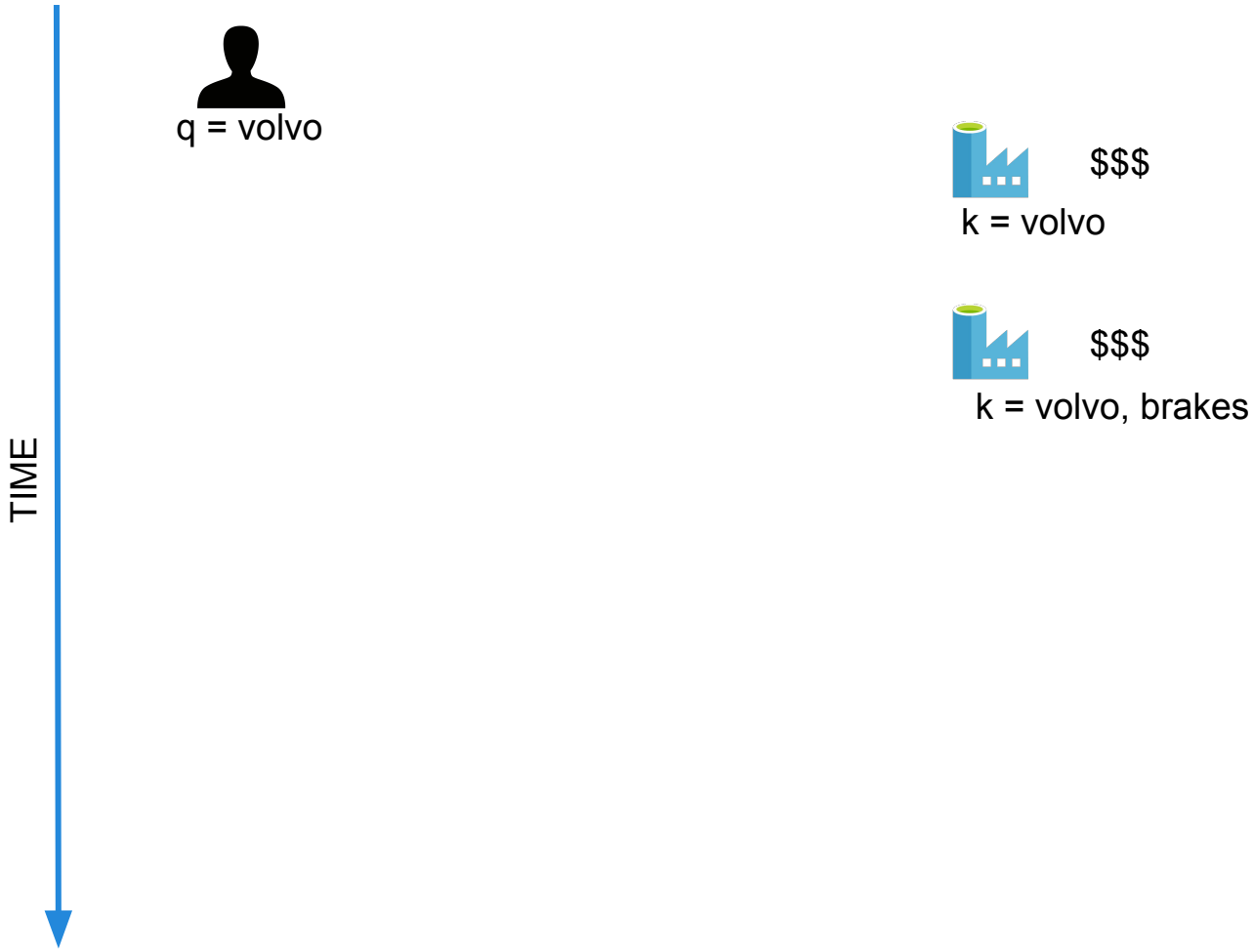


# Balance algorithm

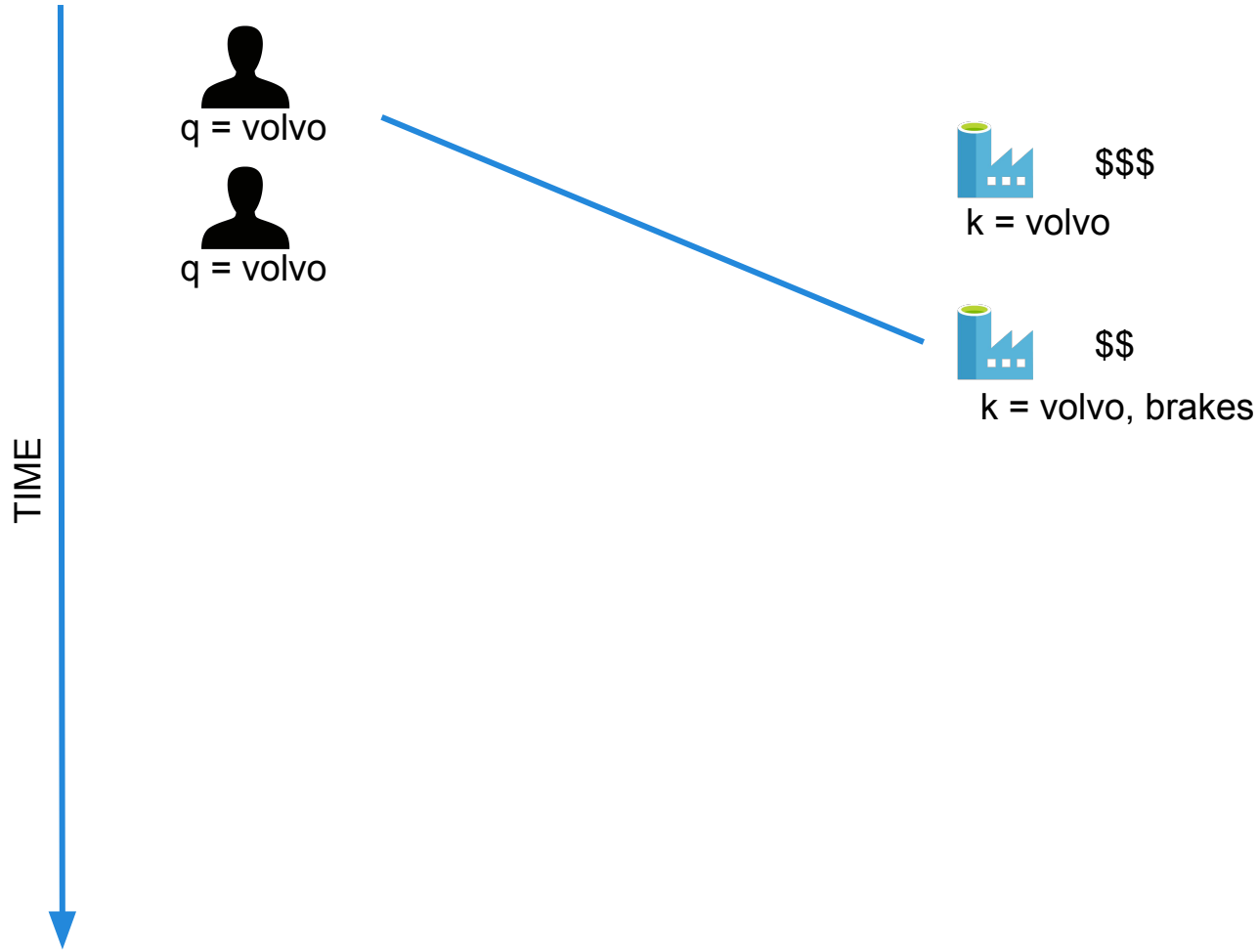
# Balance algorithm

For each query, pick the advertiser with the largest unspent budget.

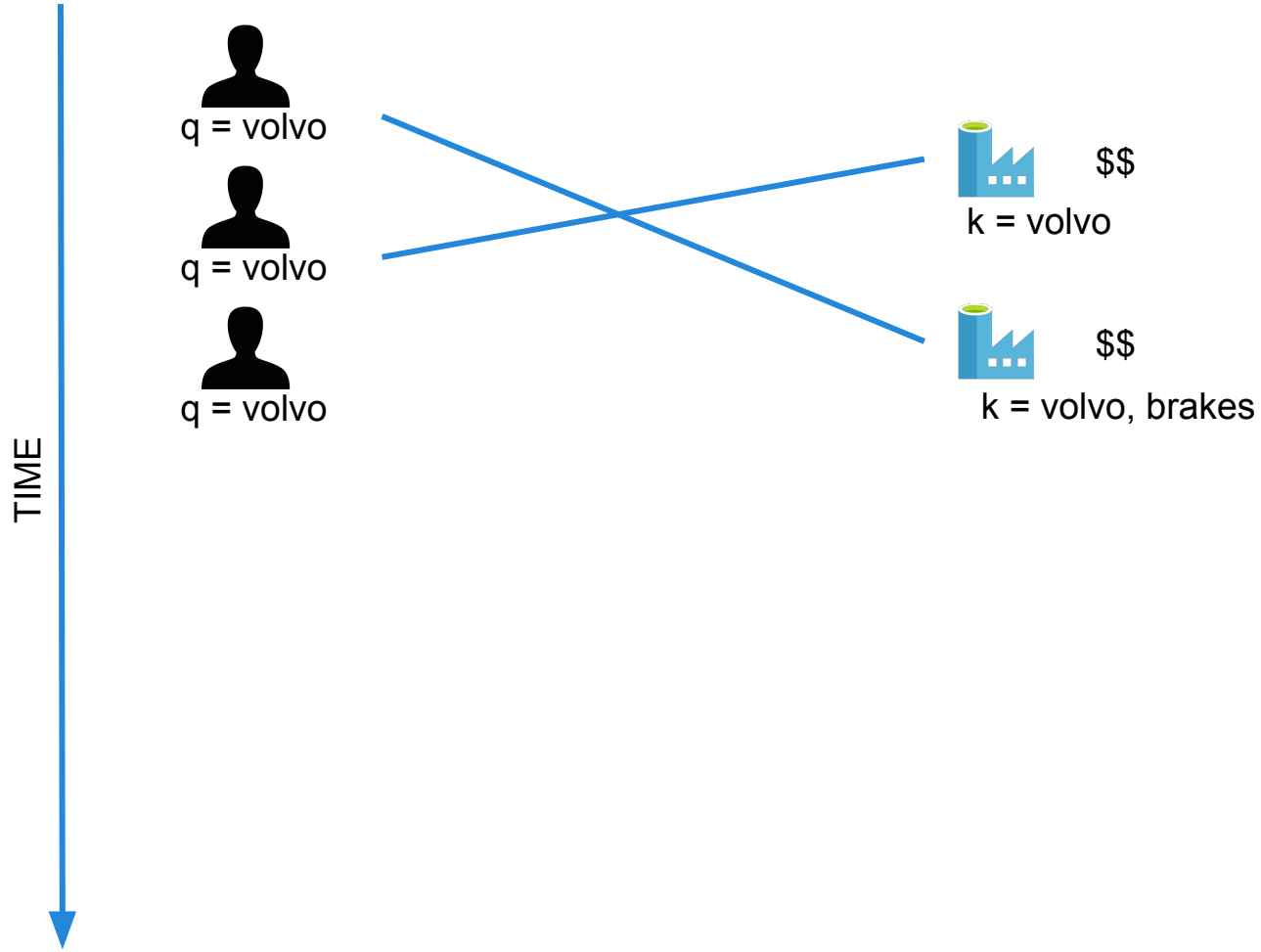
# Adwords (simplified: k=1) model



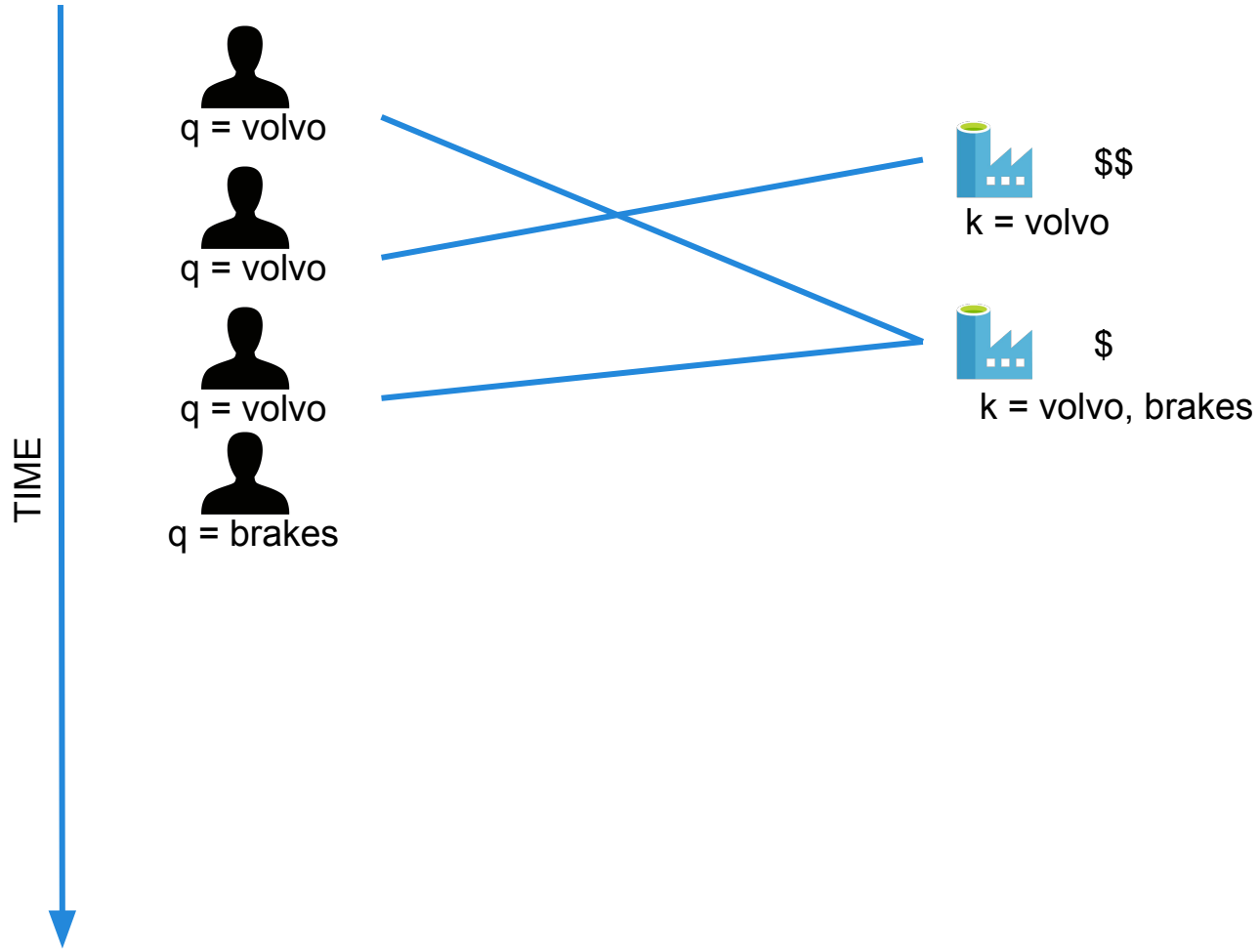
# Adwords (simplified: k=1) model



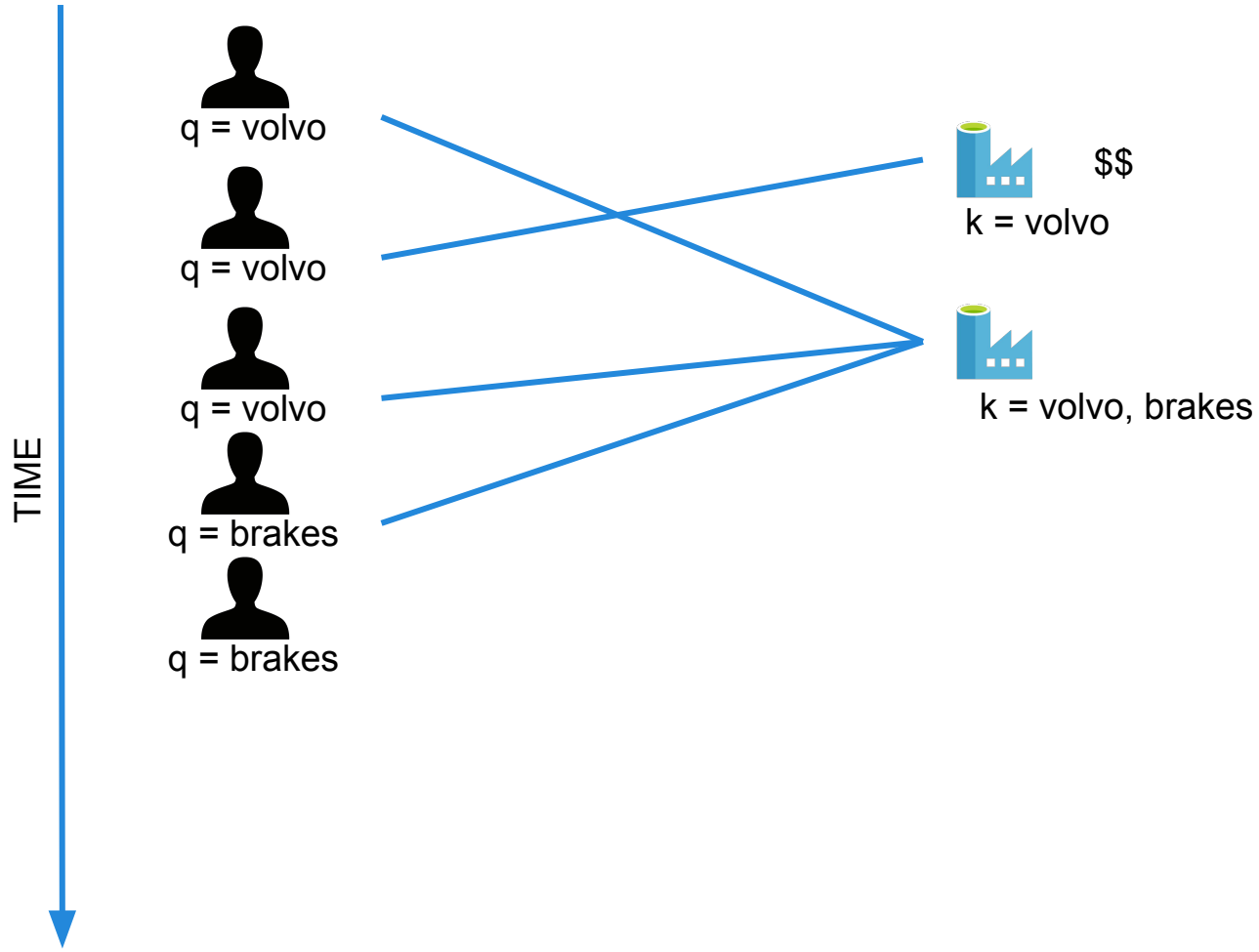
# Adwords (simplified: k=1) model



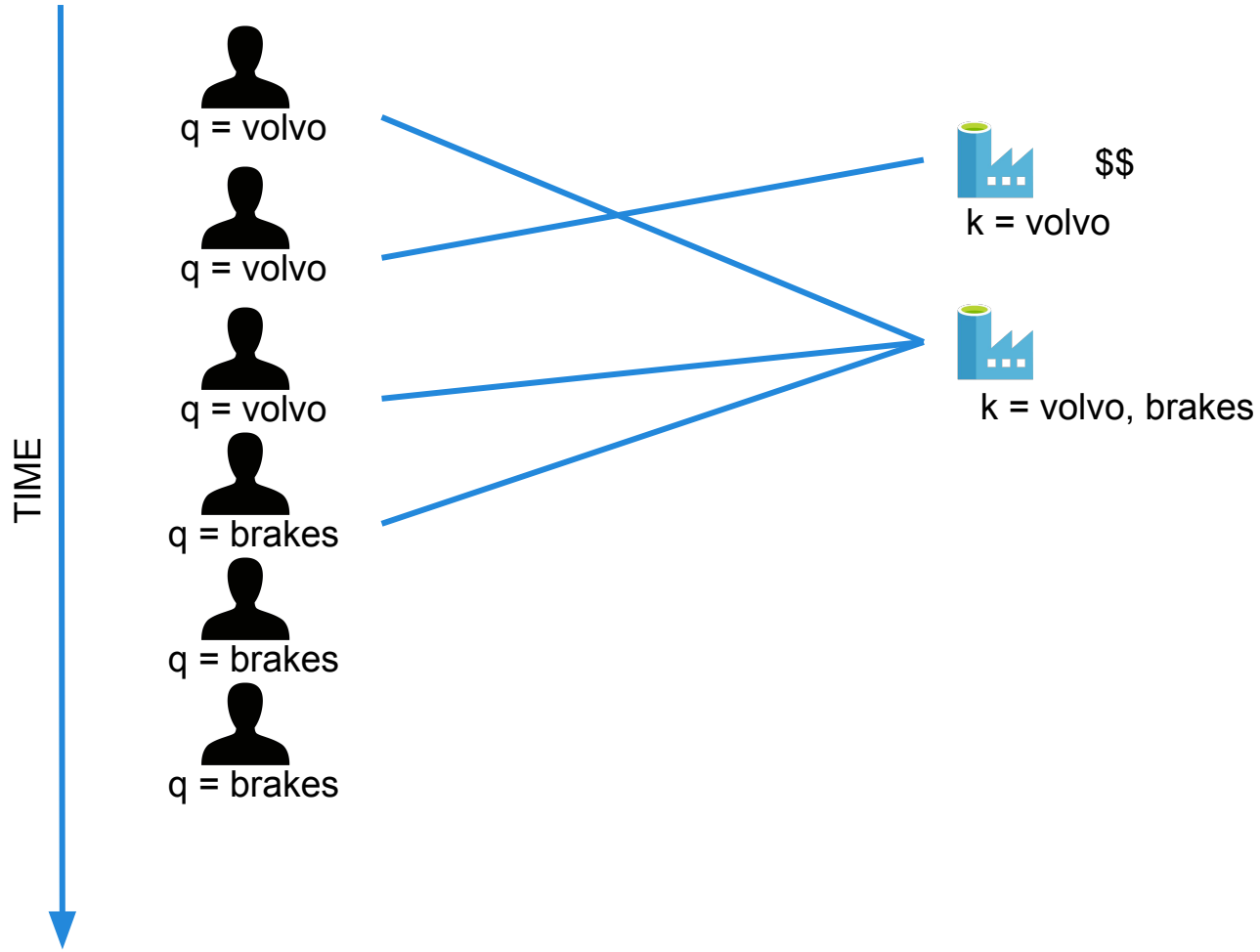
# Adwords (simplified: k=1) model



# Adwords (simplified: k=1) model

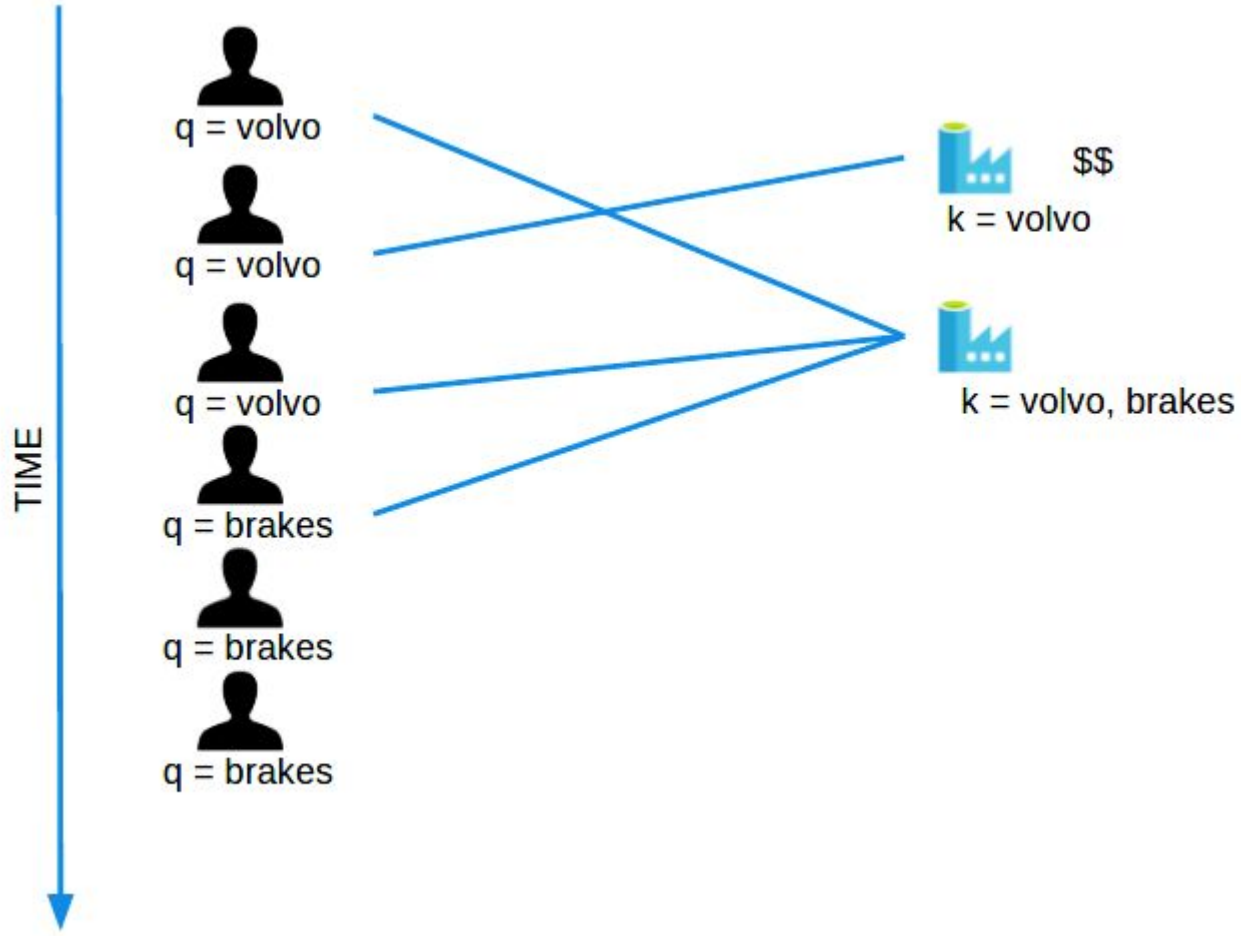


# Adwords (simplified: k=1) model



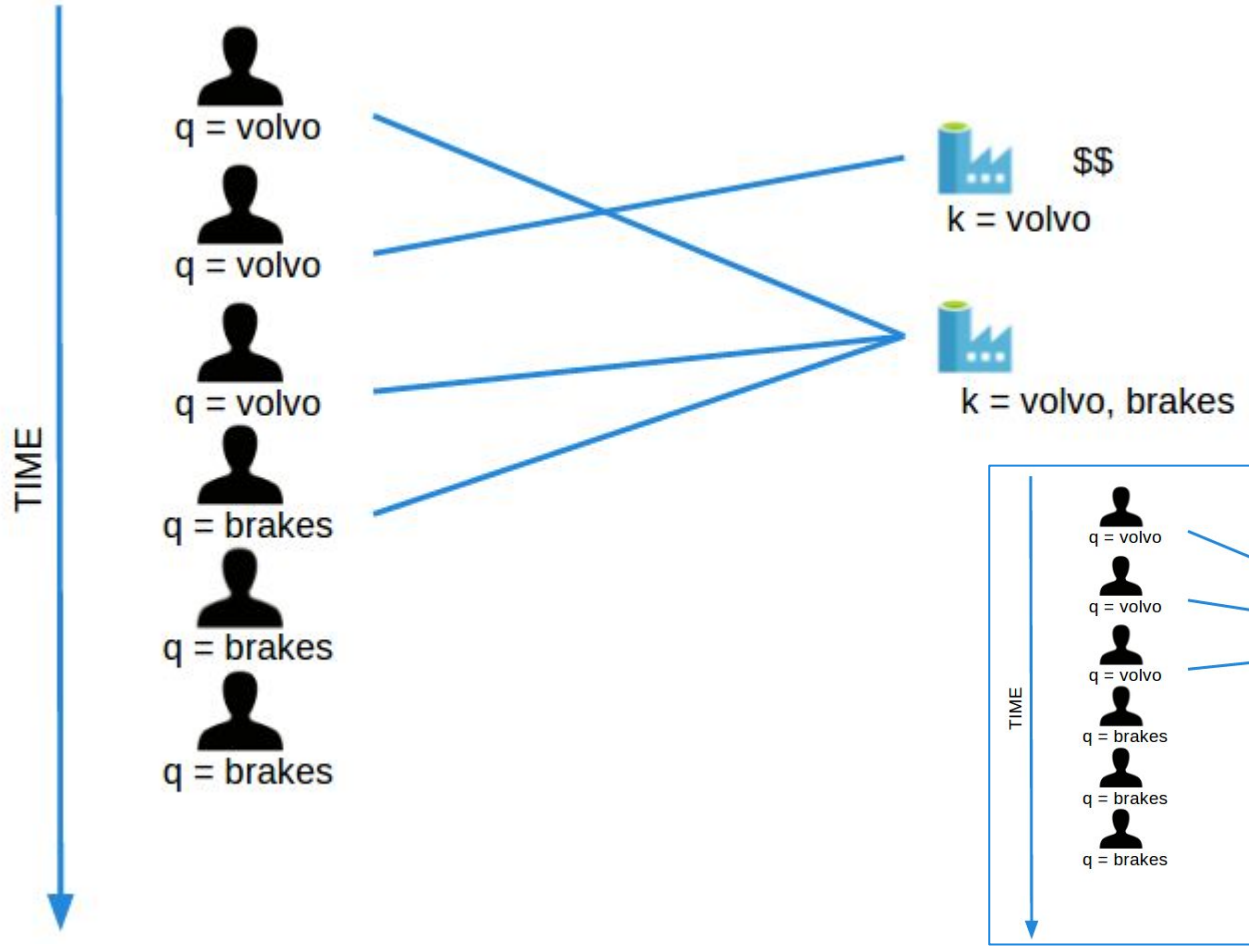
# Competitiveness

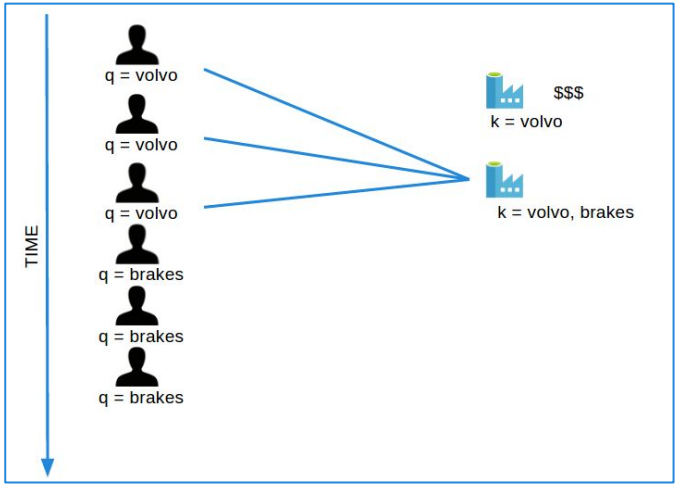
For this data we matched 4/6 = 2/3



# Revenue

4$ instead of optimal 6$ but 1$ more than for greedy



# Balance competitiveness

In the general case competitive ratio of BALANCE is **1–1/e** = approx. 0.63

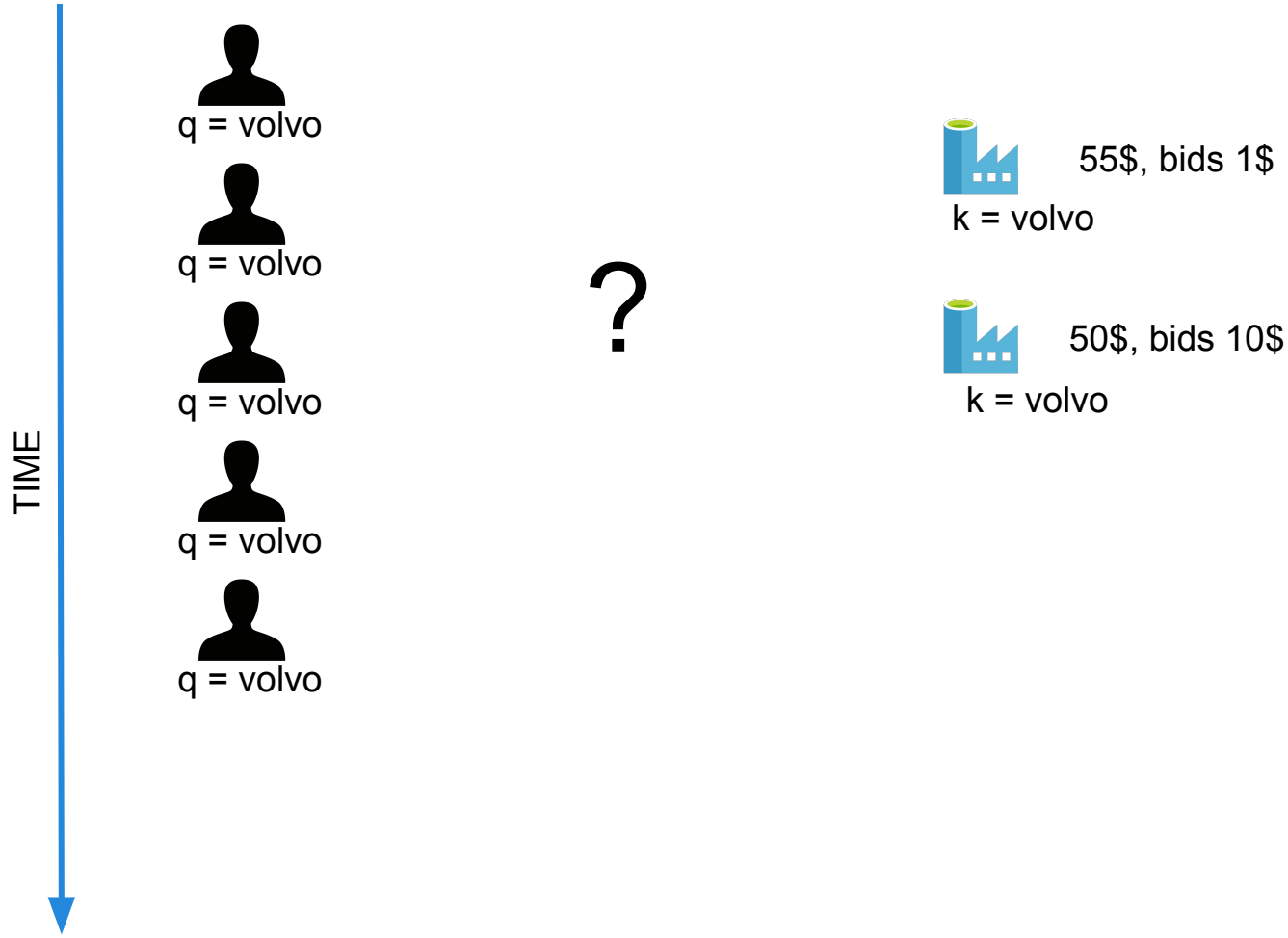
Interestingly, no online algorithm has a better competitive ratio!

# Balance competitiveness

OK, how about the more general situation?:

bids can be arbitrary

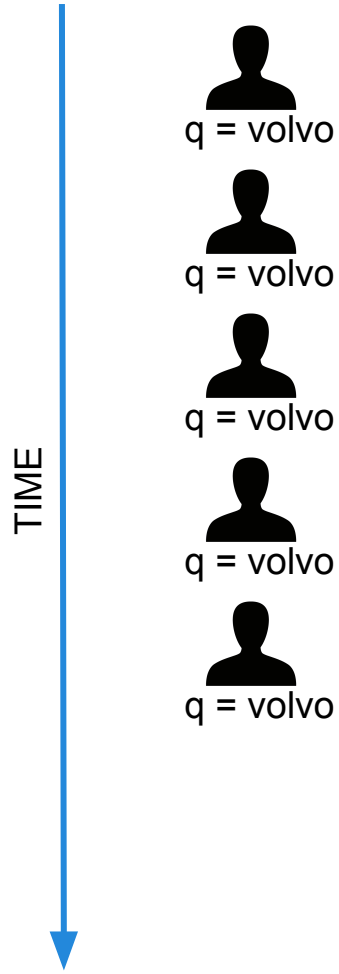
# Balance Failure Example



# Generalized Balance

- **Arbitrary bids:** consider query $q$, bidder $i$
  - Bid = $x_i$
  - Budget = $b_i$
  - Amount spent so far = $m_i$
  - Fraction of budget left over $f_i = 1 - m_i/b_i$
  - Define $\psi_i(q) = x_i(1 - e^{-f_i})$
- Allocate query $q$ to bidder $i$ with largest value of $\psi_i(q)$
- **Same competitive ratio (1-1/e)**

# Generalized Balance (whiteboard)





55$, bids 1$

k = volvo


50$, bids 10$

k = volvo

# Generalized Balance (whiteboard)

1. a=55 x=1, m=0, fi=0.632121
2. a=50 x=10, m=0, fi=6.321206
selecting 2

1. a=55 x=1, m=0, fi=0.632121
2. a=50 x=10, m=10, fi=5.506710
selecting 2

1. a=55 x=1, m=0, fi=0.632121
2. a=50 x=10, m=20, fi=4.511884
selecting 2

1. a=55 x=1, m=0, fi=0.632121
2. a=50 x=10, m=30, fi=3.296800
selecting 2

1. a=55 x=1, m=0, fi=0.632121
2. a=50 x=10, m=40, fi=1.812692
selecting 2

1. a=55 x=1, m=0, fi=0.632121
2. a=50 x=10, m=50, fi=0.000000
selecting 1

1. a=55 x=1, m=1, fi=0.625371
2. a=50 x=10, m=50, fi=0.000000
selecting 1

# Real life

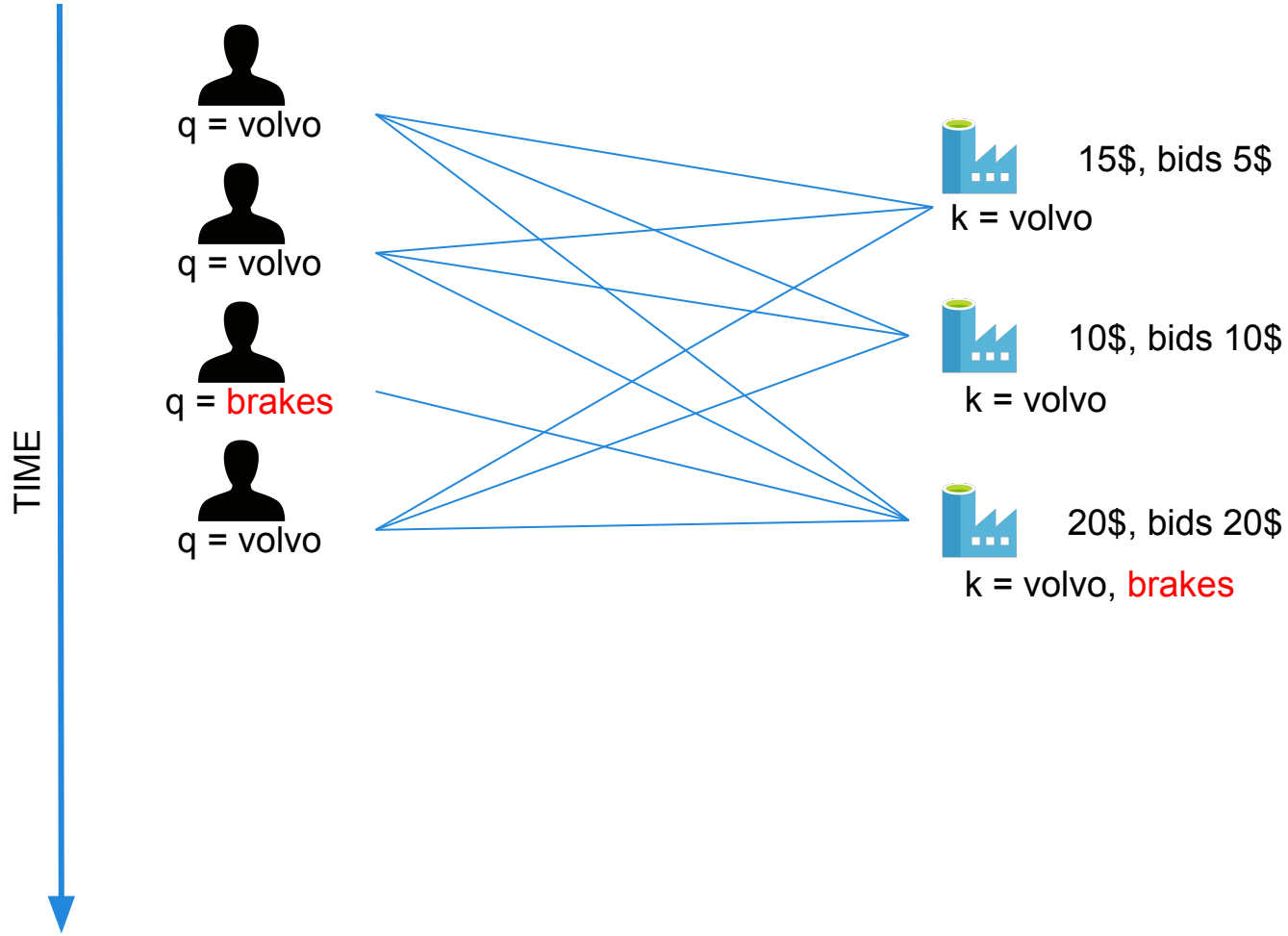
- Budgets differ
- Bids differ
- Bidders (companies) coming and leaving
- Matching is more complicated than keywords

# Assignment

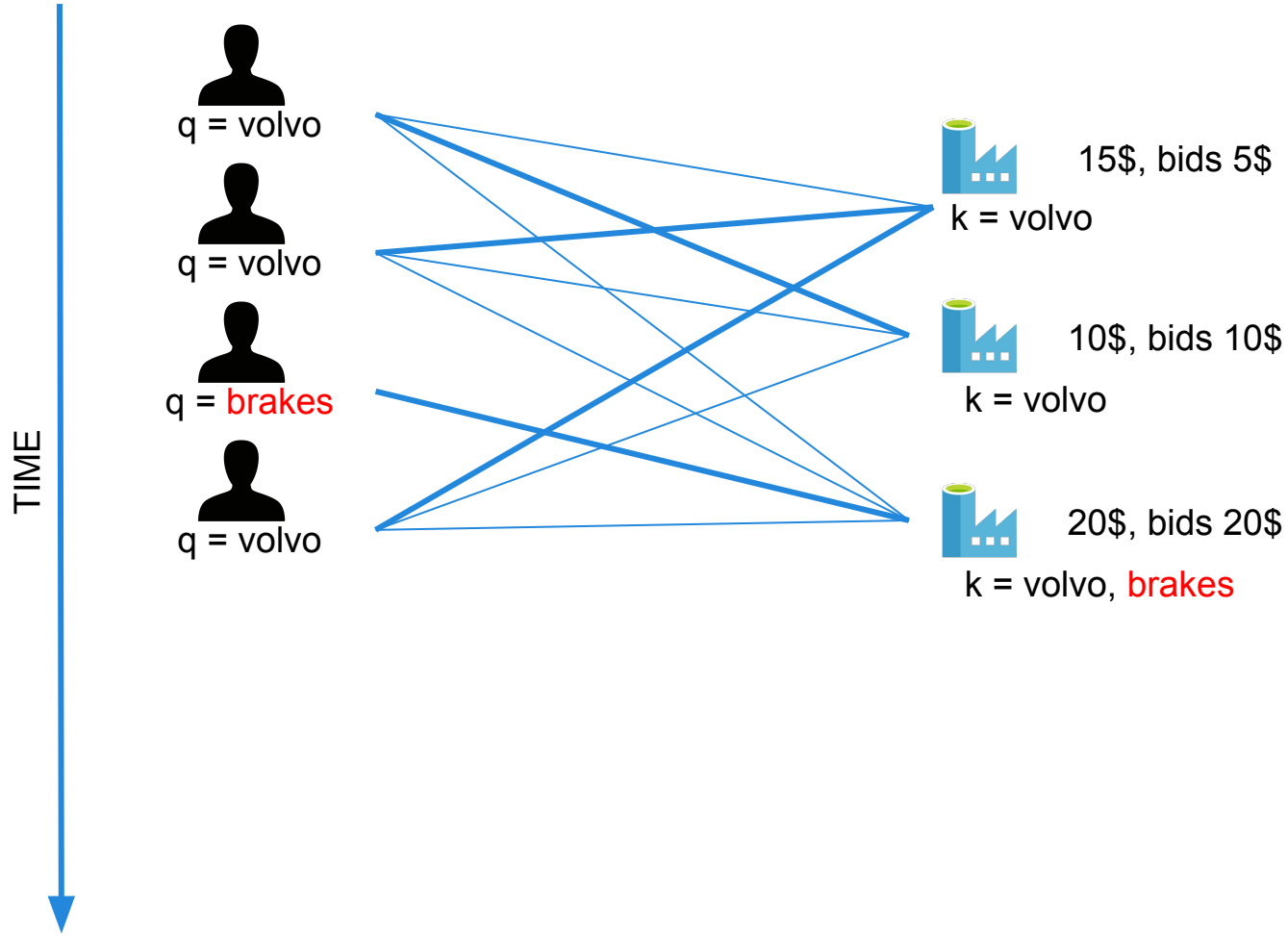
For the given below assignment problem:

- Find the optimal assignment (with max revenue)
- Find the Generalized Balance Algorithm solution.
- Compare matchings and revenues.

# Problem


TIME
q = volvo
q = volvo
q = brakes
q = volvo
15$, bids 5$
k = volvo
10$, bids 10$
k = volvo
20$, bids 20$
k = volvo, brakes

# Solution: sample perfect matching

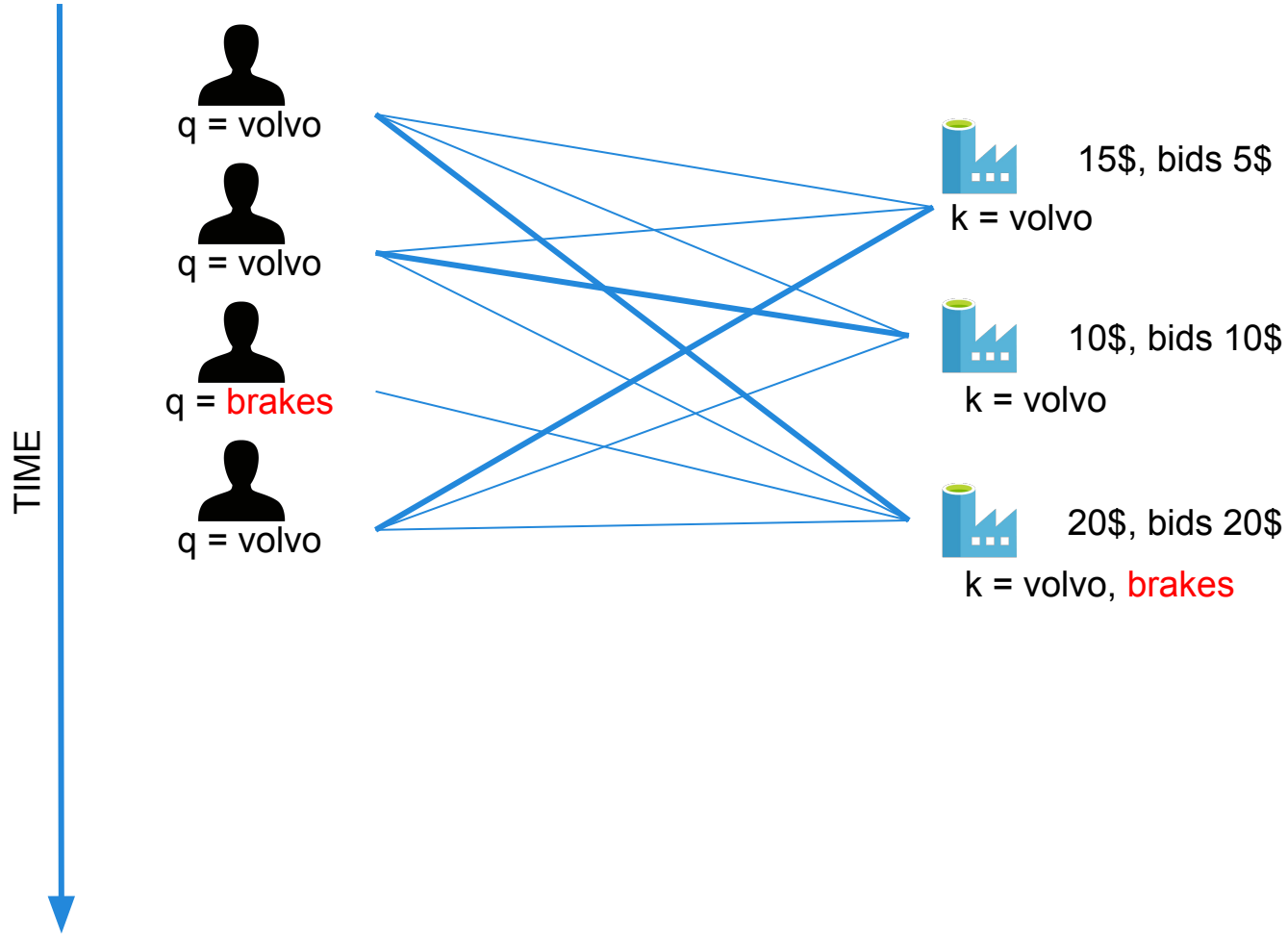


# Solution: Generalized balance

| i | m1 | fi1 | m2 | fi2 | m3 | f3 | Selected |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 3.16 | 0 | 6.32 | 0 | 12.64 | 3 |
| 2 | 0 | 3.16 | 0 | 6.32 | 20 | 0 | 2 |
| 3 | - | - | - | - | 20 | 0 | NONE |
| 4 | 0 | 3.16 | 10 | 0 | 20 | 0 | 1 |

User 3 has matching keywords only with the bidder no 3, but when he arrives the bidder has no more money to spend

# Solution: Generalized balance



# Solution

Revenues:

- Generalized balance: 35$
- Perfect matching: 40$

Matchings:

- Generalized balance: 3
- Perfect matching: 4