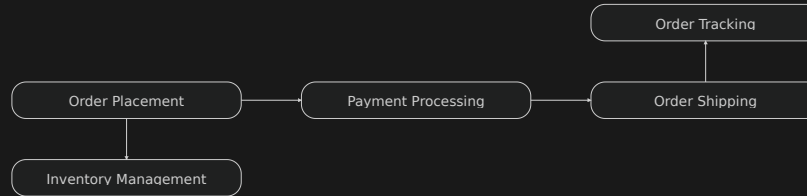


# LOGISCHE COMPONENTEN

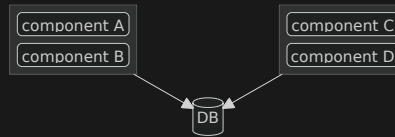
# DIAGRAMVOORSTELLING



# ORGANISATIE BRONCODE

```
.
├── inventory
│   └── management
├── order
│   ├── placement
│   ├── shipping
│   └── tracking
│       ├── source_code_file_1.txt
│       └── source_code_file_2.txt
├── payment
│   └── processing
```

# RELATIE TOT SERVICES

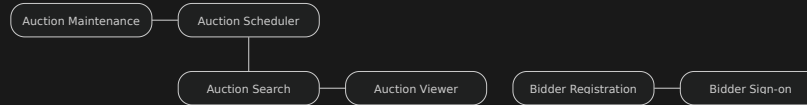


# **CASE: ADVENTUROUS AUCTIONS**

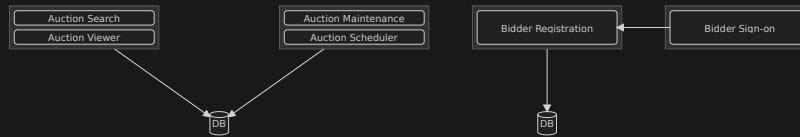
# REQUIREMENTS

- schalen tot mogelijk duizenden users per veiling
- in-person en online biedingen
- registratie (credit card info a priori vereist)
- videostreams van lopende veilingen
- overzicht van reeds gemaakte boden
- automatisch bepalen "winnaar" (of bij conflict overlaten aan veilingmeester)
- automatisch betaling regelen

# VOORBEELDCOMPONENTEN



# MOGELIJKE MAPPING







# STAP 1: INITIËLE KERNCOMPONENTEN ACHTERHALEN

- 2 technieken
  - workflow
  - actor/agent
- mogen beide toepassen

# WORKFLOW APPROACH

- perspectief van een gebruiker
- conceptuele stappen van één "user journey"
- verschillende stappen  $\Rightarrow$  verschillende componenten
  - niet nodig voor sterk verwante stappen

"Meedoen en winnen"

Bieder

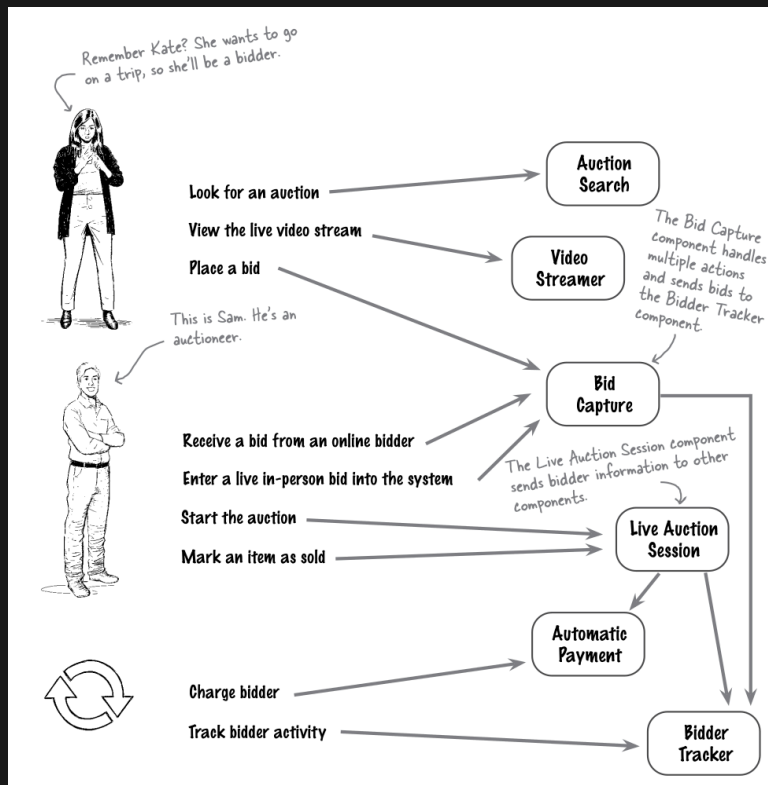


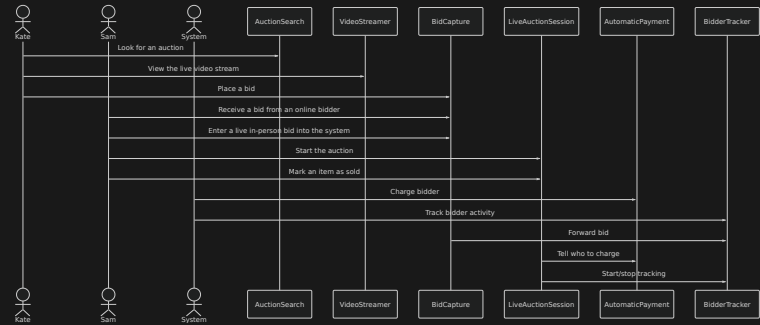
## NAMEN

- Auction Registration
- Live Auction Session
- Video Streamer
- Bid Capture
- Automatic Payment

## ACTOR/ACTION APPROACH

- handig bij meerdere soorten gebruikers
- sommen **belangrijkste** handelingen op
- verbinden deze met componenten
- verbinden ook componenten onderling
- "system" actor voor automatische acties







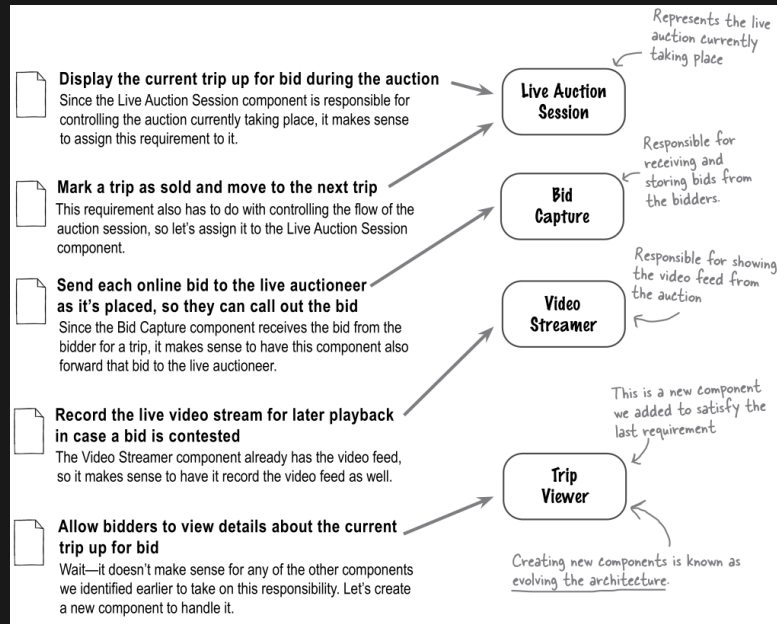
## ENTITY TRAP

- component met twee verantwoordelijkheden
- extra risico bij vage naamgeving
  - "supervisor"
  - "manager"
  - "control center"
  - ...

## COMBINEREN

- identificeer actors en primaire acties zoals in actor/action
- doorloop dan acties zoals in workflow
- kunnen zo meer in detail gaan indien zinvol

# STAP 2: REQUIREMENTS TOEWIJZEN AAN COMPONENTEN



# STAP 3: ROL EN VERANTWOORDELIJKHEDEN ANALYSEREN

- per component vraag stellen: "welke taken?"
- interne samenhang van een component = "cohesie"
  - hoe duidelijker het takenpakket, hoe hoger de cohesie
- moet opgevolgd worden naarmate systeem groeit
  - kan introductie nieuwe componenten noodzakelijk maken
    - kan bijhouden in een ADR!
- onthoud: **mik op sterke cohesie**

# STAP 4: ARCHITECTURALE KARAKTERISTIEKEN ANALYSEREN

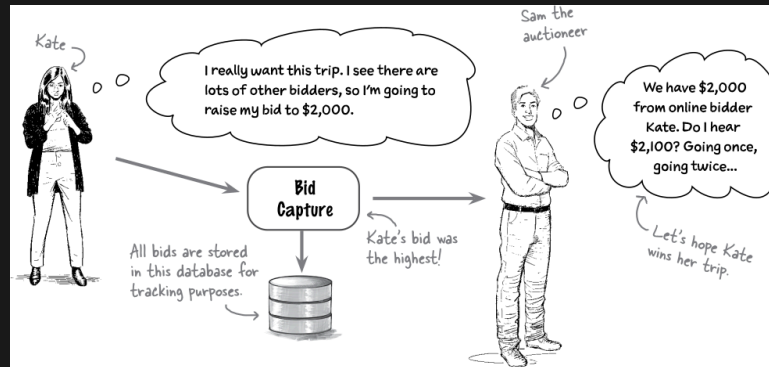
- kijken hier naar de "driving characteristics"
- bekijken of de takenverdeling deze ondersteunt
- hier wel enige redenering over fysieke kenmerken vereist

## VOORBEELD (1)

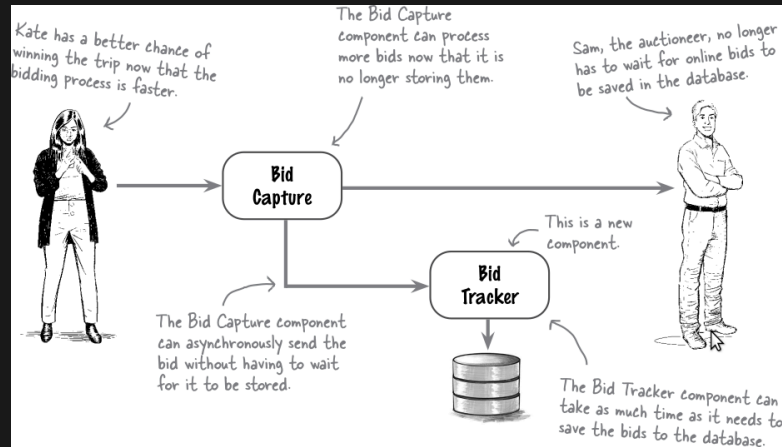
driving characteristics:

- scalability
- availability
- performance

## VOORBEELD (2)



# VOORBEELD (3)





# INTERACTIE TUSSEN COMPONENTEN

# **AFFERENTE KOPPELING**

"Hoe hoger, hoe meer hier van afhangt."

Responsible for allowing  
bidders to sign up for an  
auction

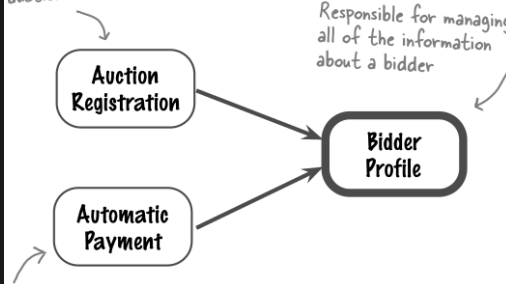
**Auction  
Registration**

Responsible for managing  
all of the information  
about a bidder

**Bidder  
Profile**

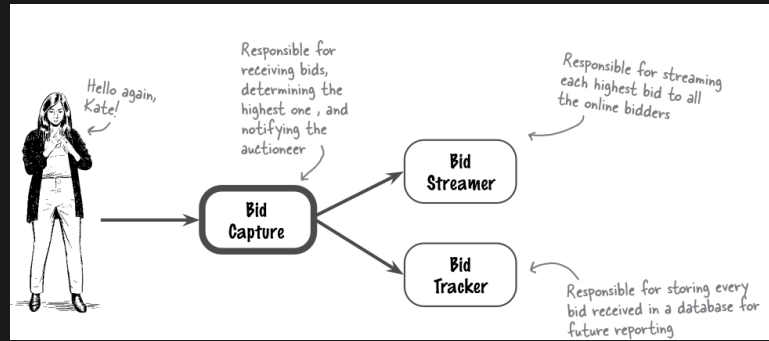
**Automatic  
Payment**

Responsible for charging  
the winning bidder's  
credit card



# EFFERENTE KOPPELING

"Hoe hoger, hoe afhankelijker."



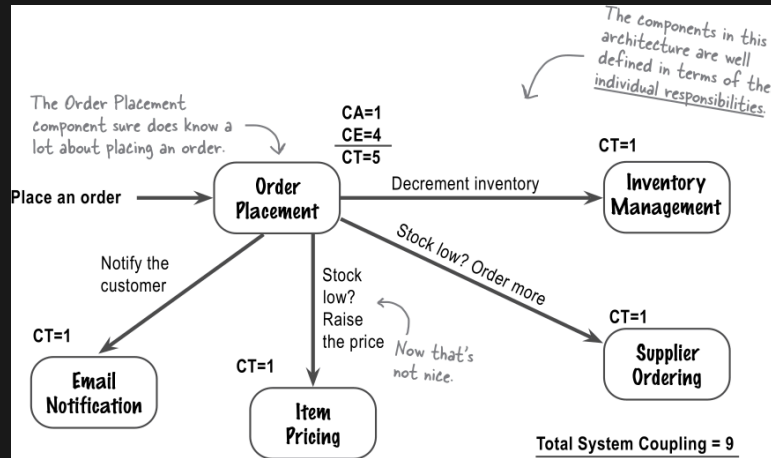
# KOPPELING METEN

- per component afferente, efferente en totale (som) koppeling
- ook totale koppeling van het systeem (som van sommen)

# WET VAN DEMETER

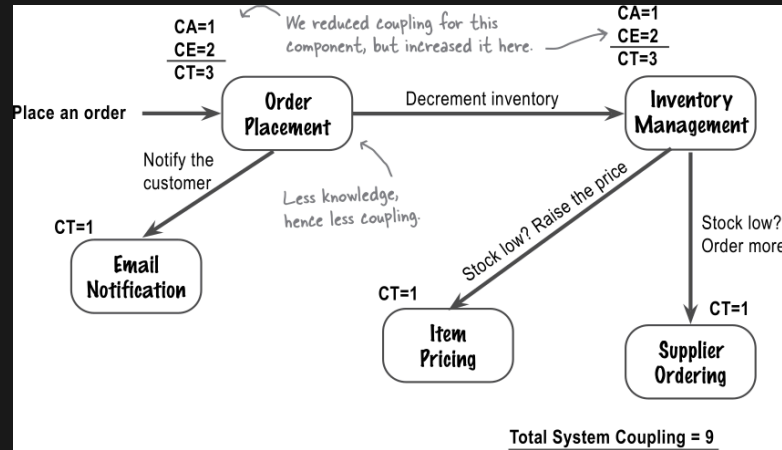


# TOEGEPAST (1)





# TOEGEPAST (2)



# TOEGEPAST (3)

