Your Architectural Kata is...

Going Going Gone!

bidder

view live video stream
view live bid stream
place a bid

auctioneer

receive online bid
enter live bids into system
mark item as sold

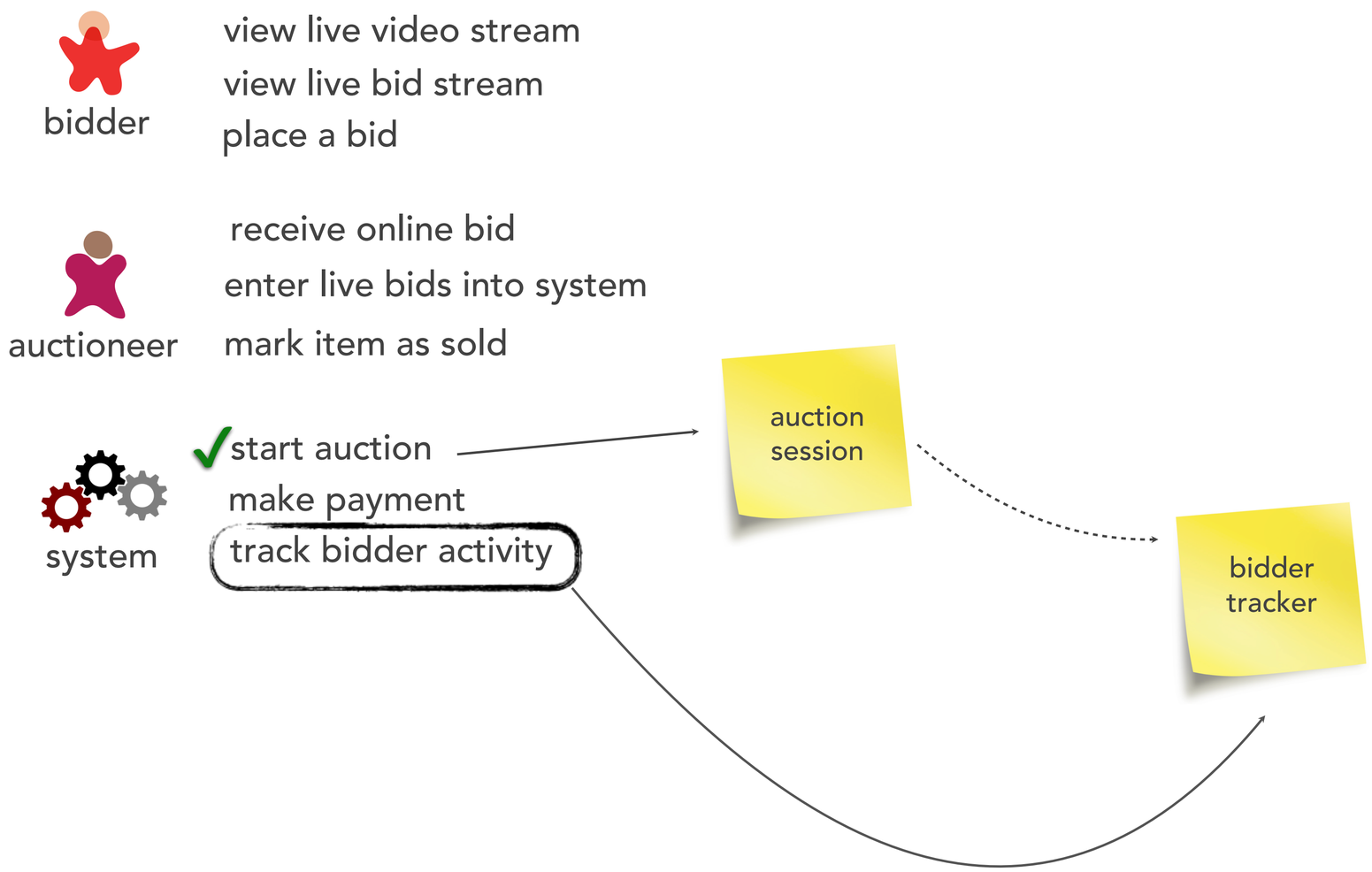
system

✓start auction
make payment
track bidder activity

auction session

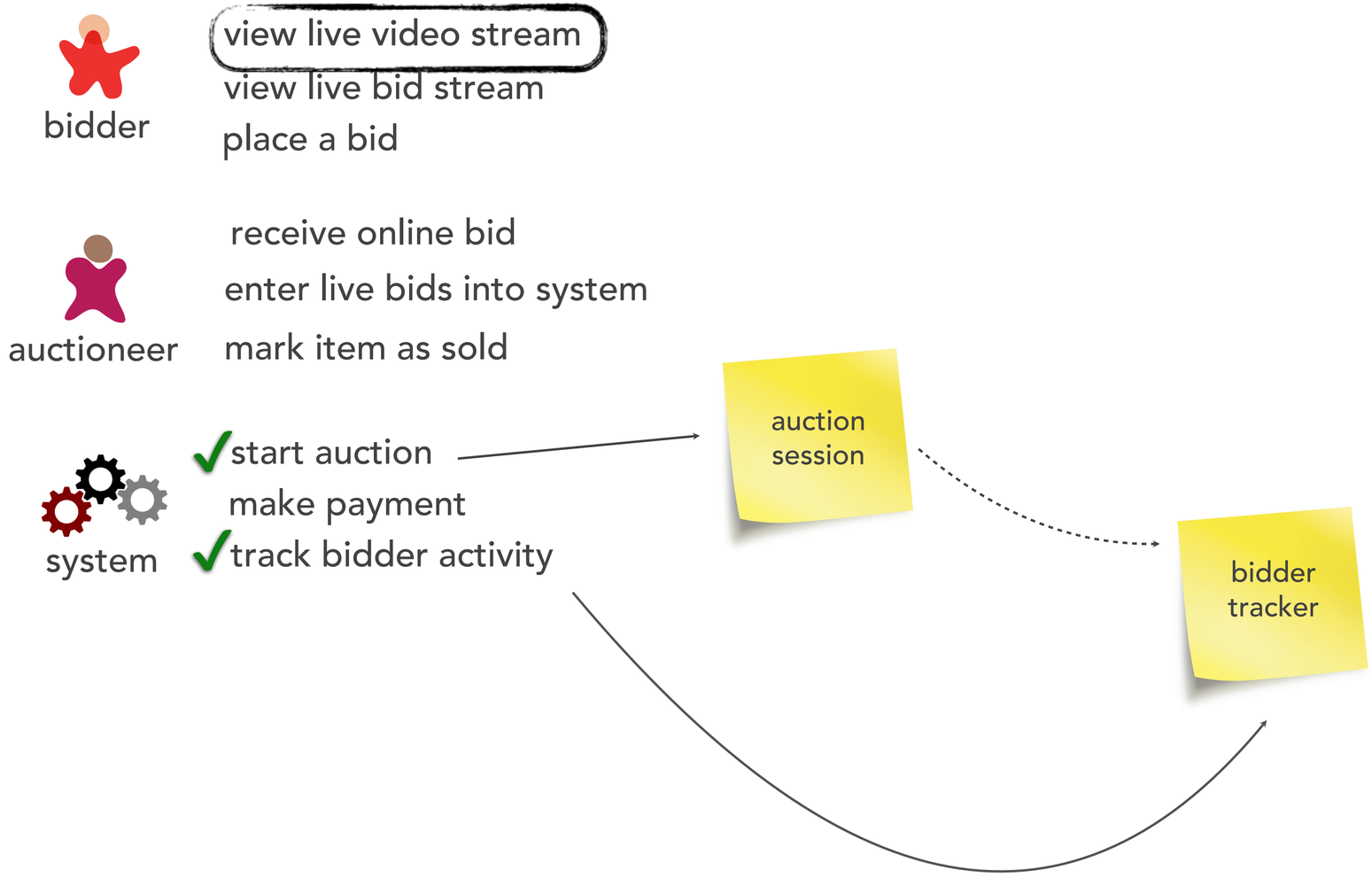
Your Architectural Kata is...

Going Going Gone!



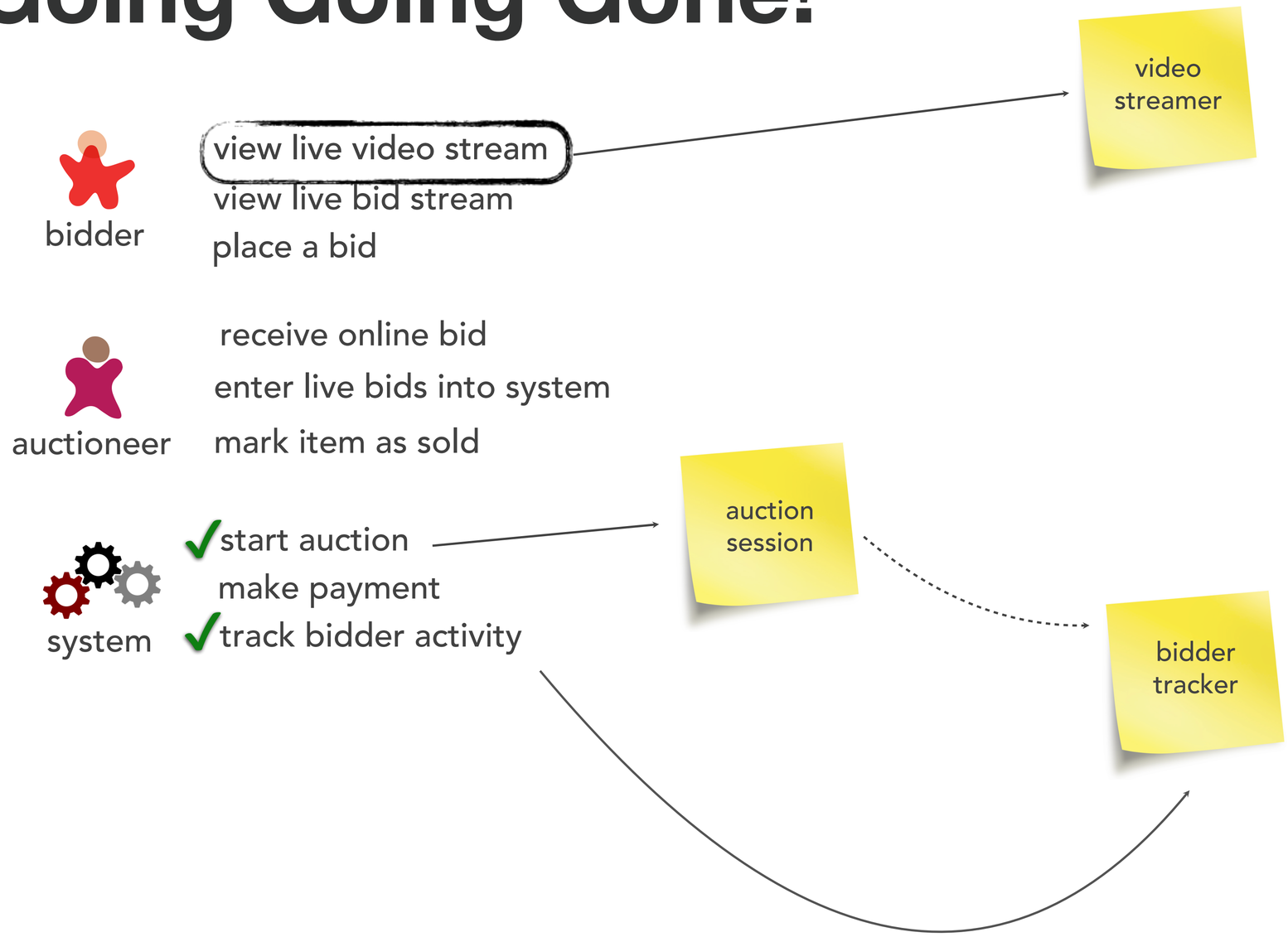
Your Architectural Kata is...

Going Going Gone!



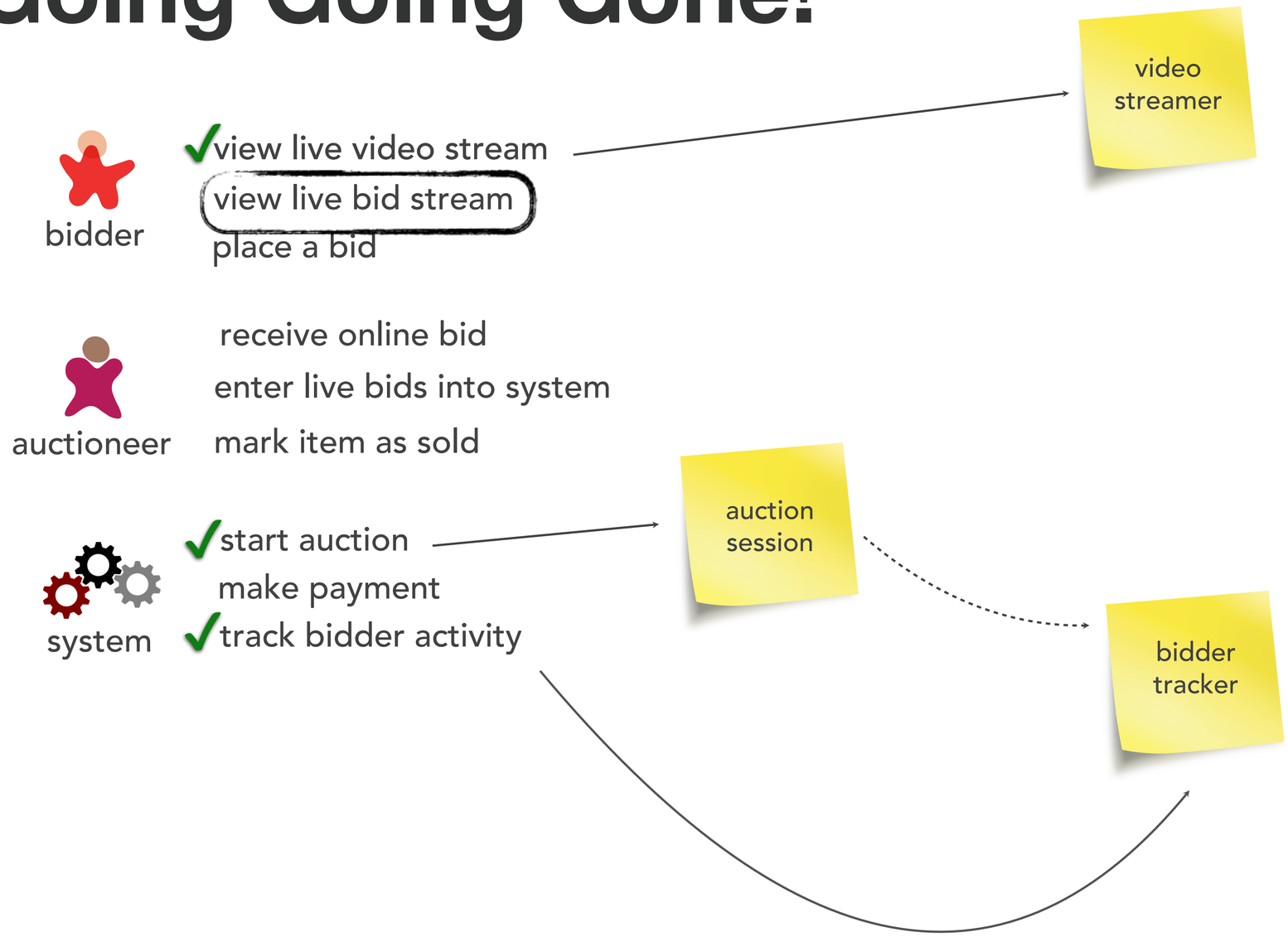
Your Architectural Kata is...

Going Going Gone!



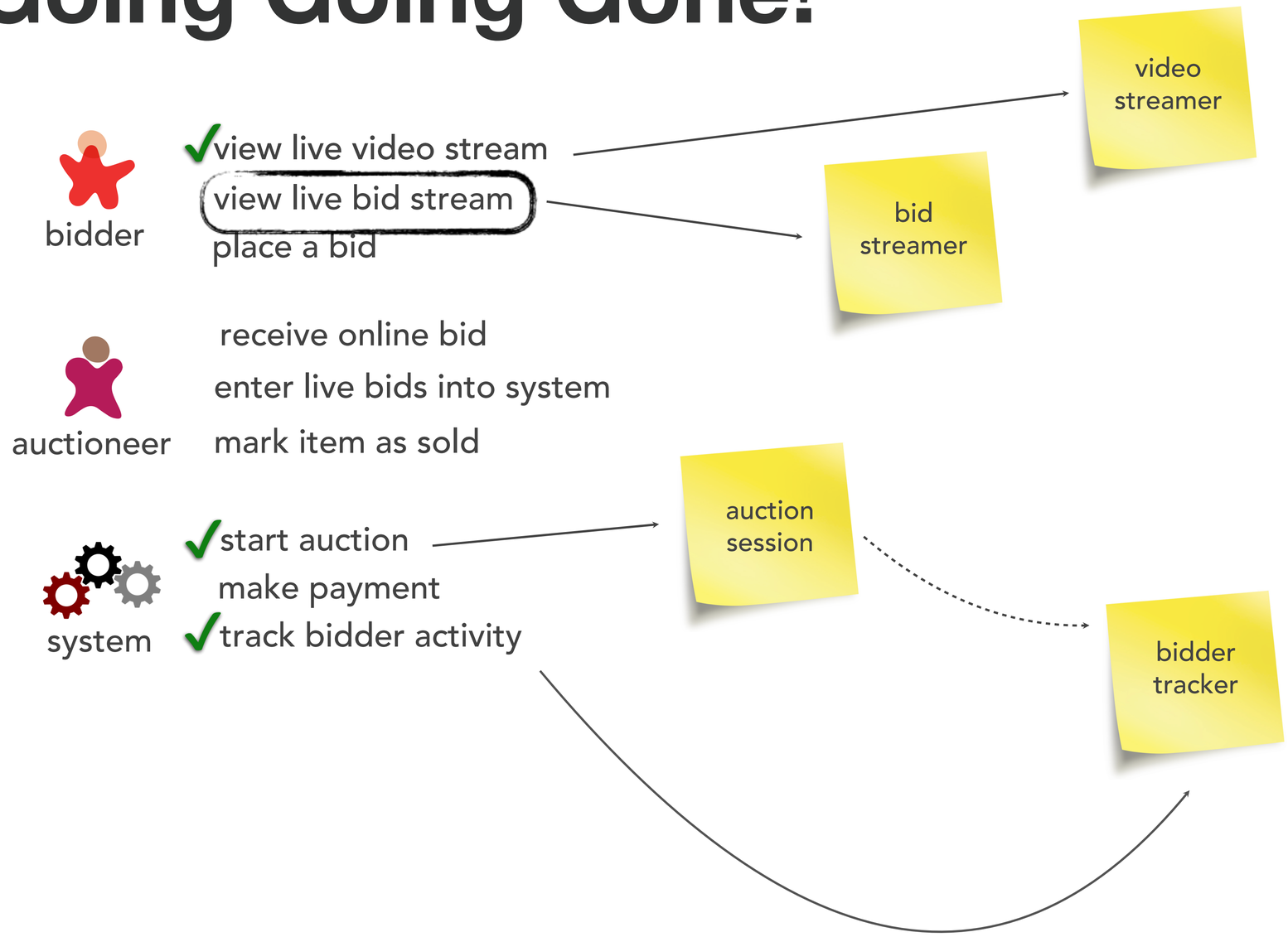
Your Architectural Kata is...

Going Going Gone!



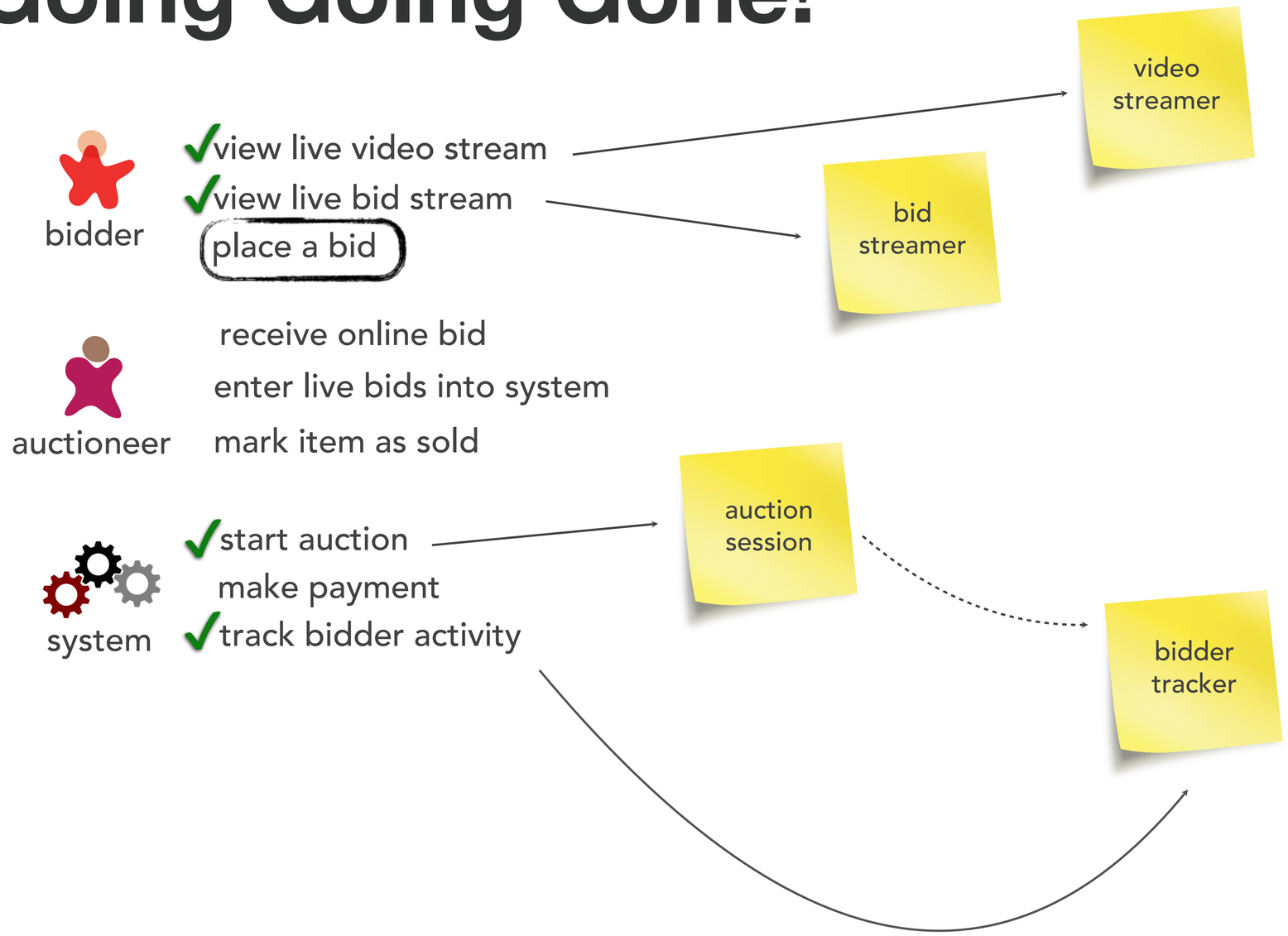
Your Architectural Kata is...

Going Going Gone!



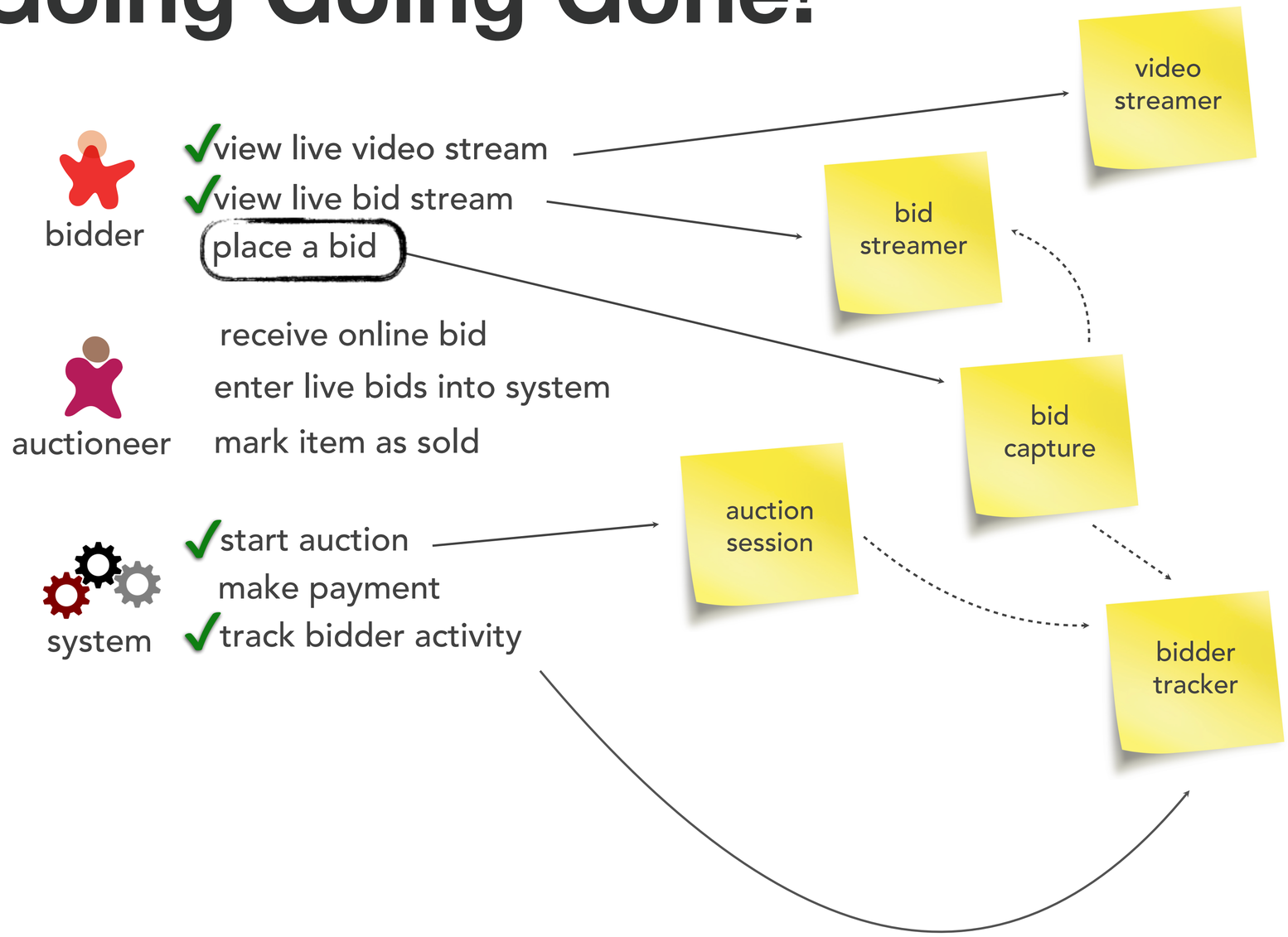
Your Architectural Kata is...

Going Going Gone!



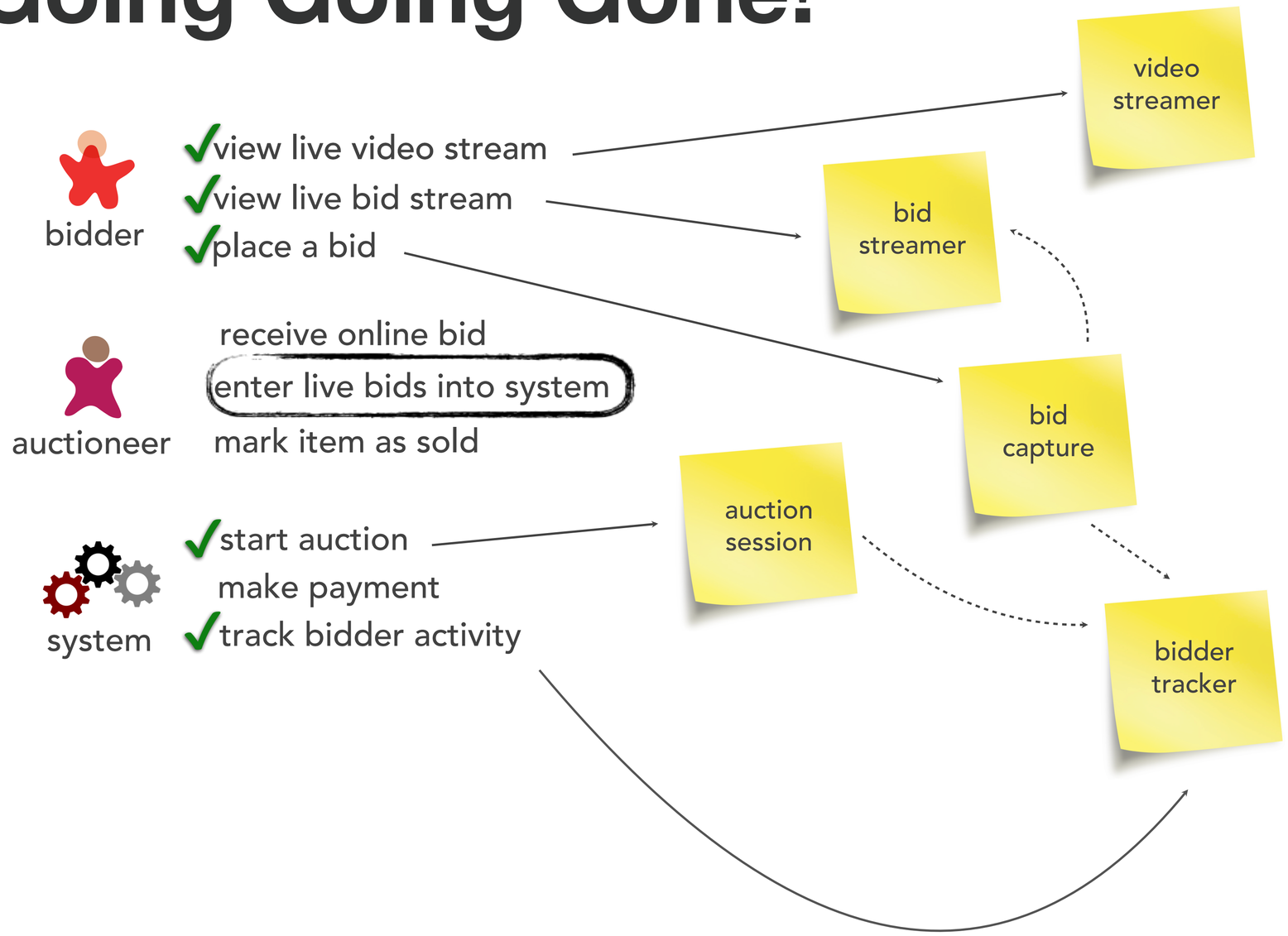
Your Architectural Kata is...

Going Going Gone!



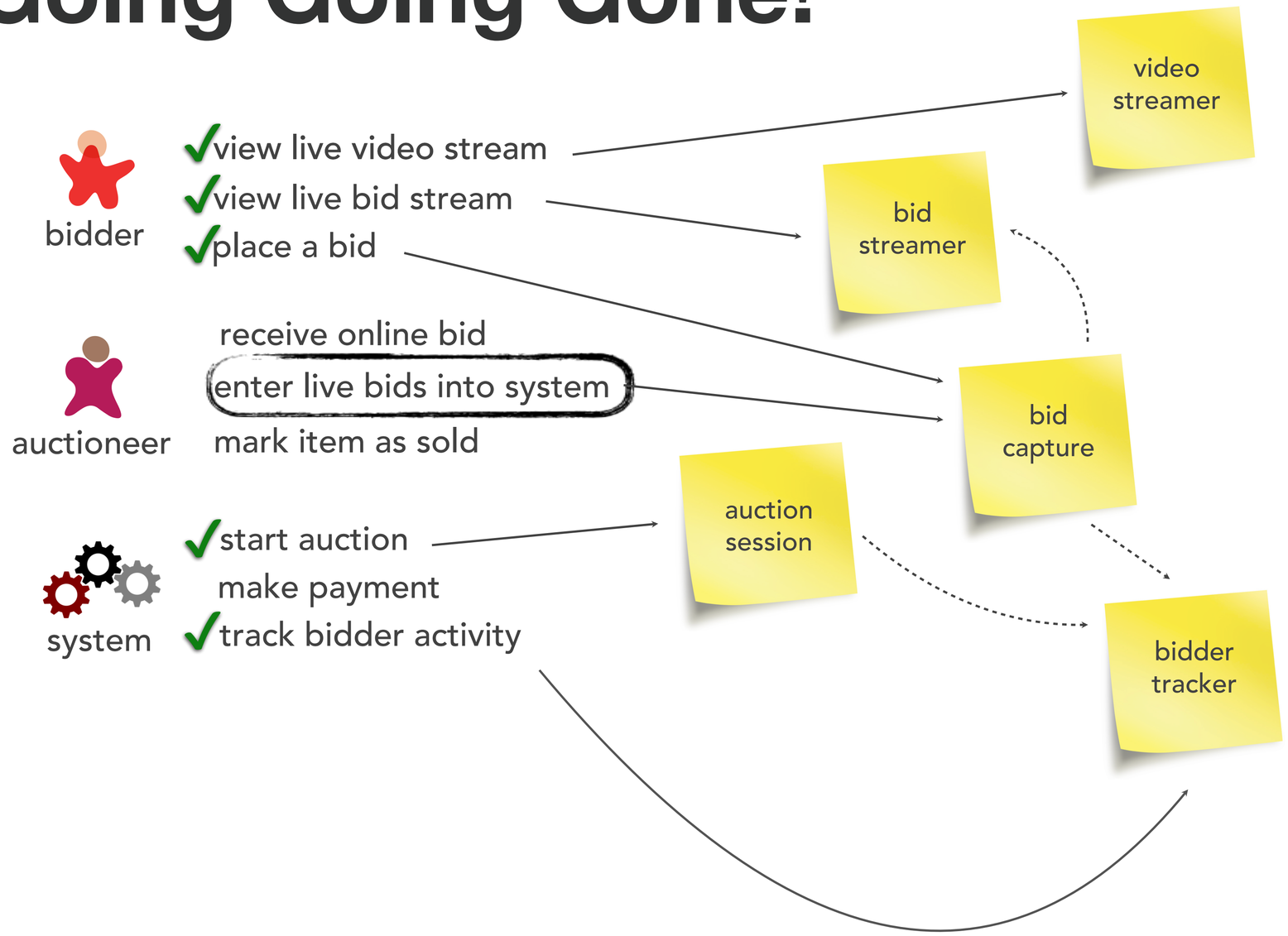
Your Architectural Kata is...

Going Going Gone!



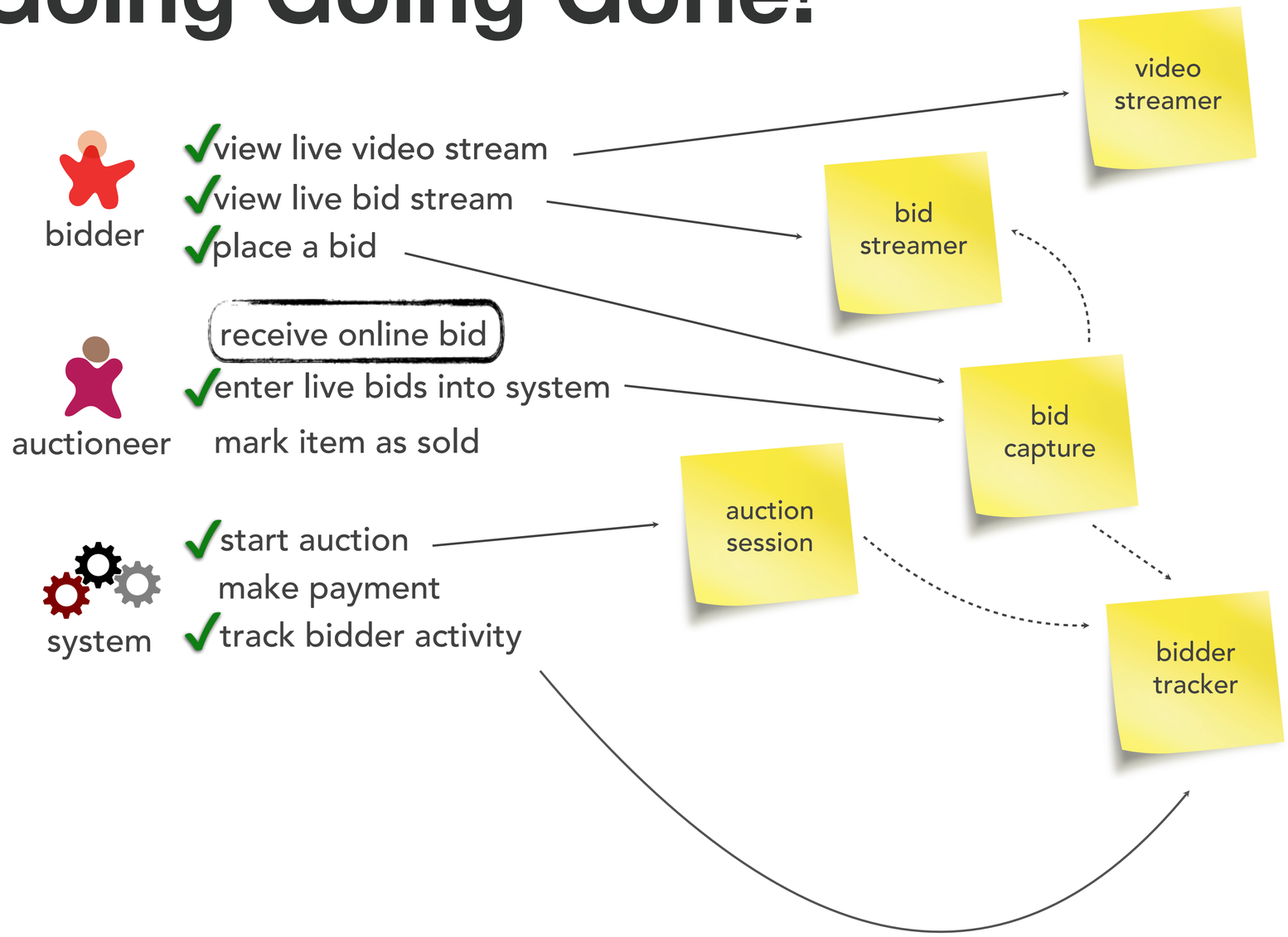
Your Architectural Kata is...

Going Going Gone!



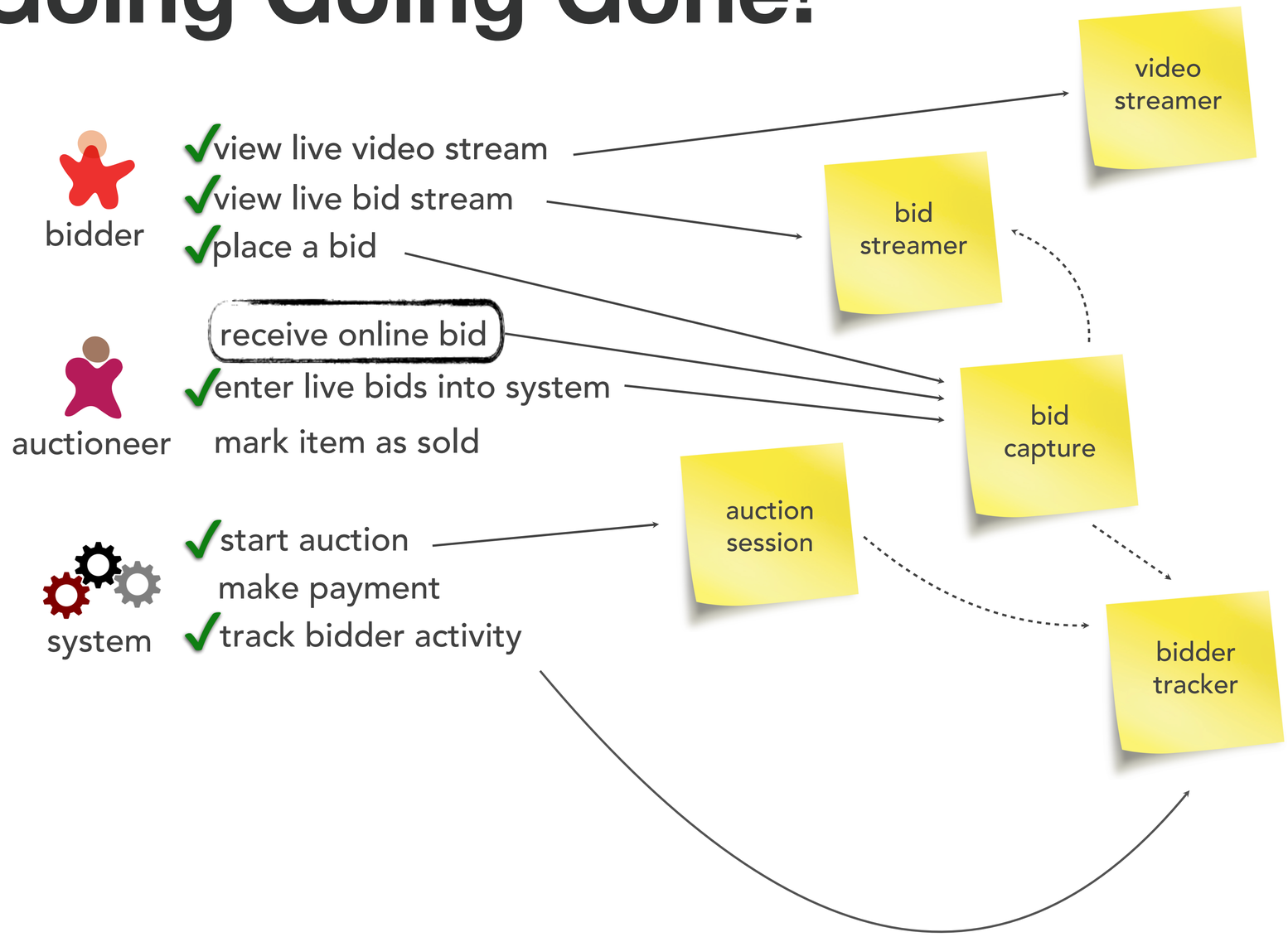
Your Architectural Kata is...

Going Going Gone!



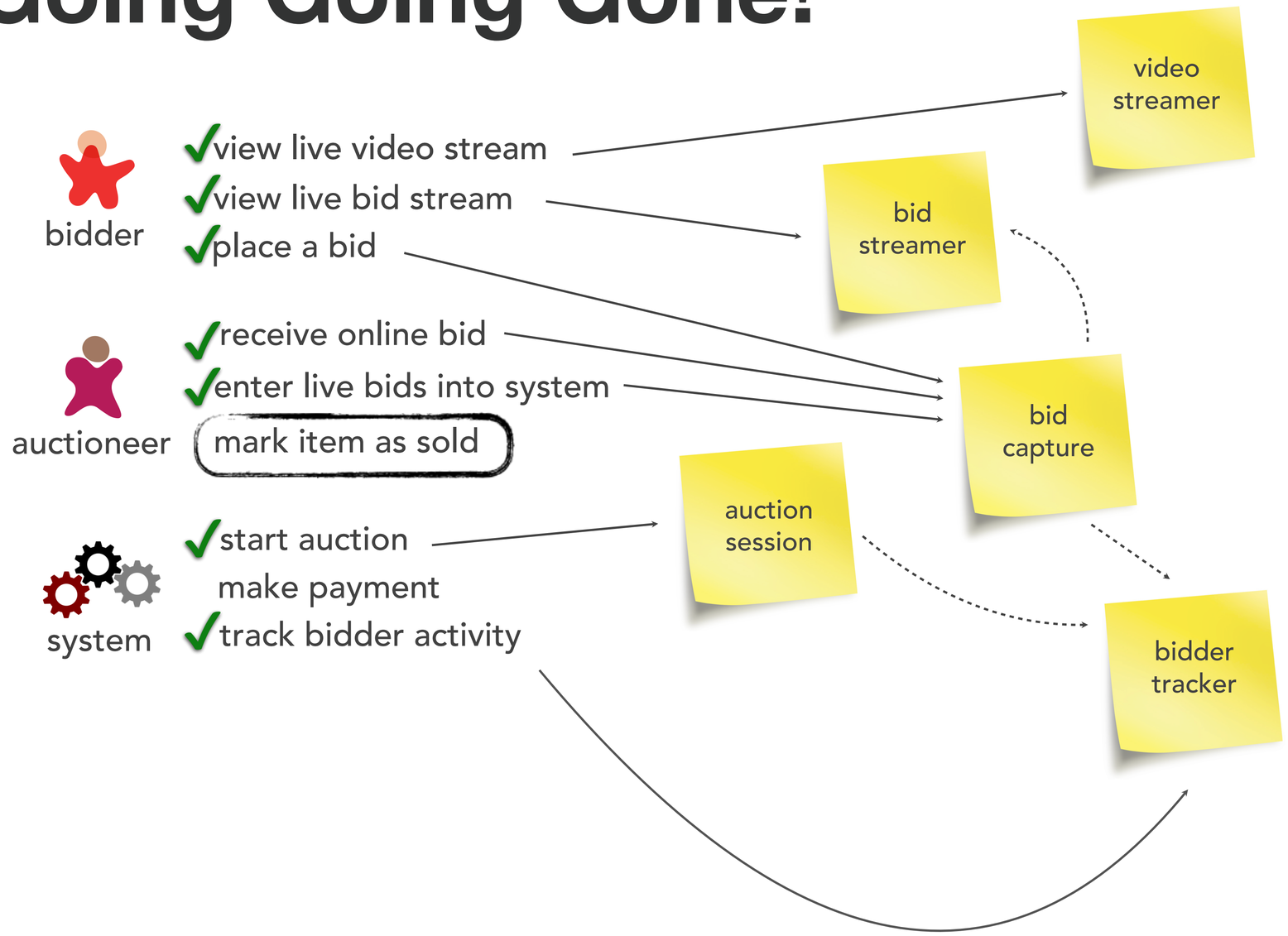
Your Architectural Kata is...

Going Going Gone!



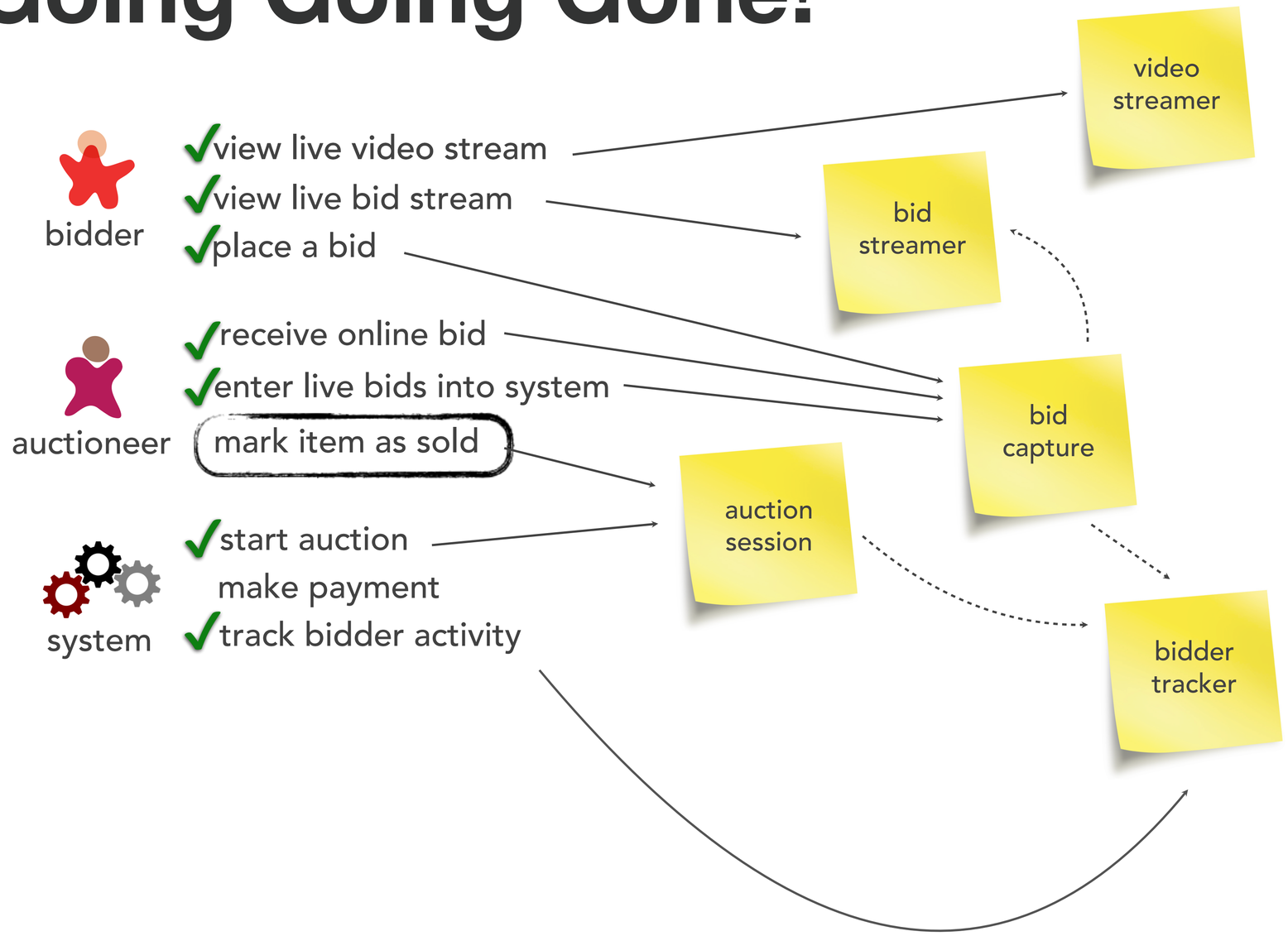
Your Architectural Kata is...

Going Going Gone!



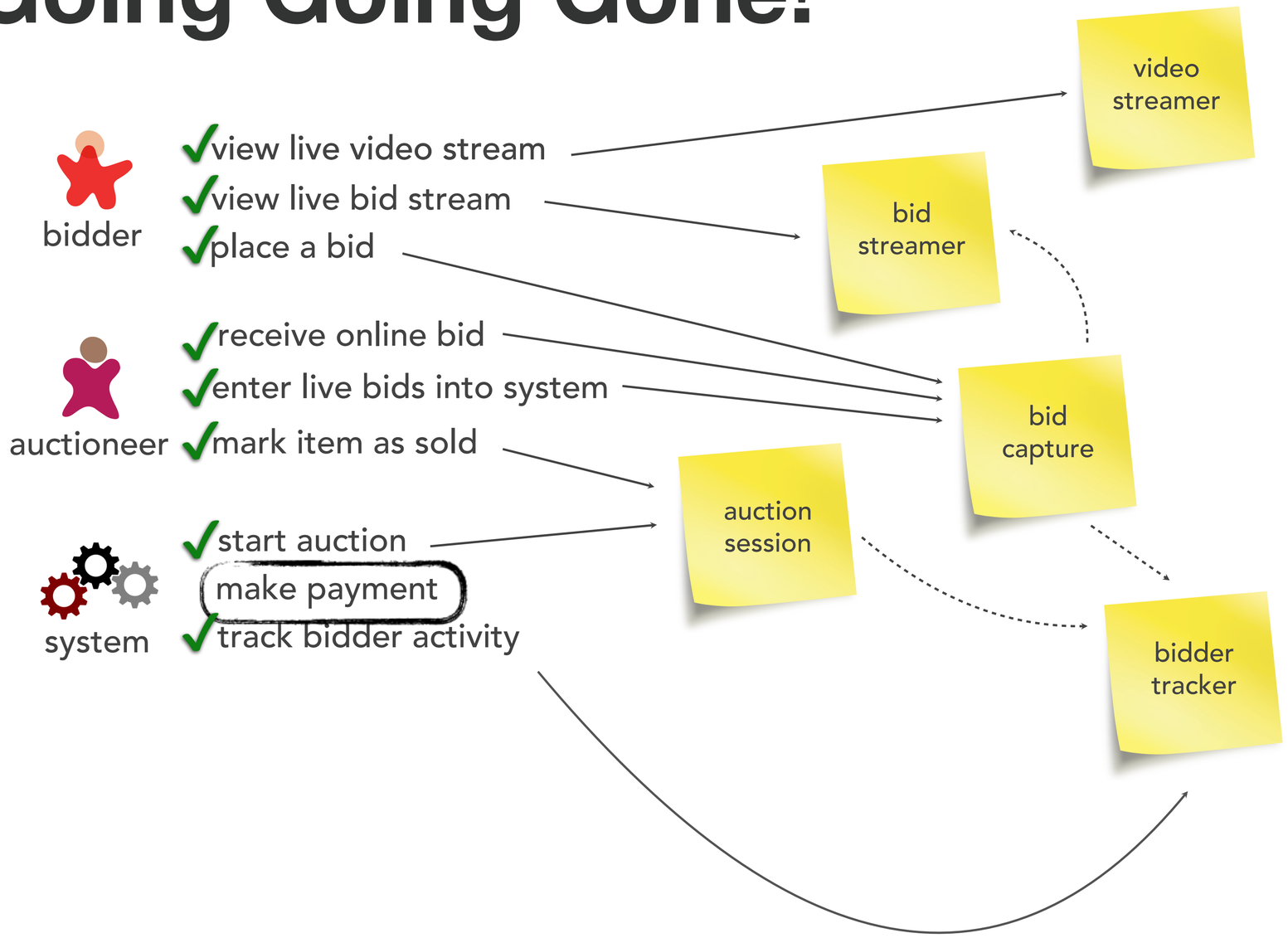
Your Architectural Kata is...

Going Going Gone!



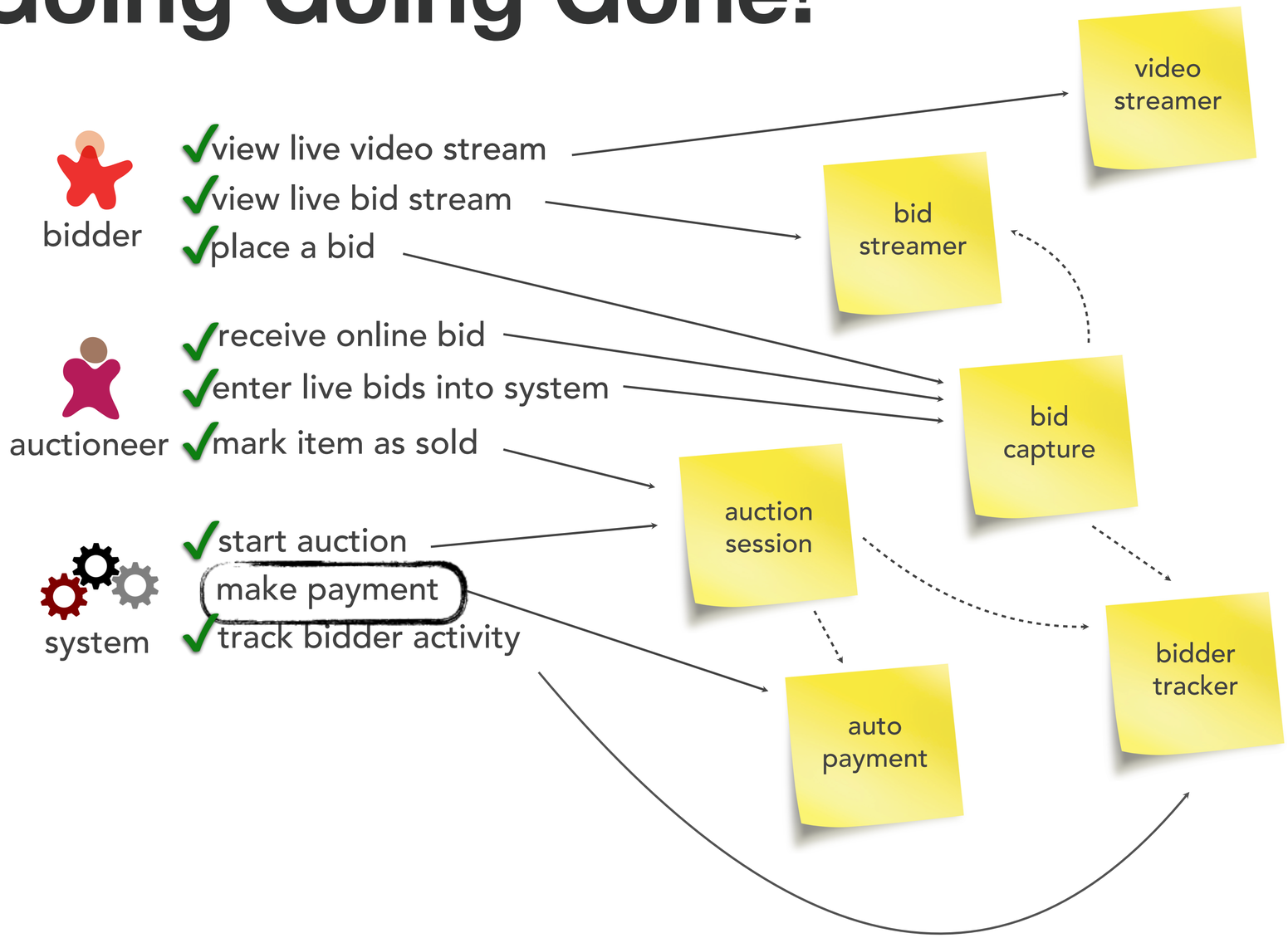
Your Architectural Kata is...

Going Going Gone!



Your Architectural Kata is...

Going Going Gone!

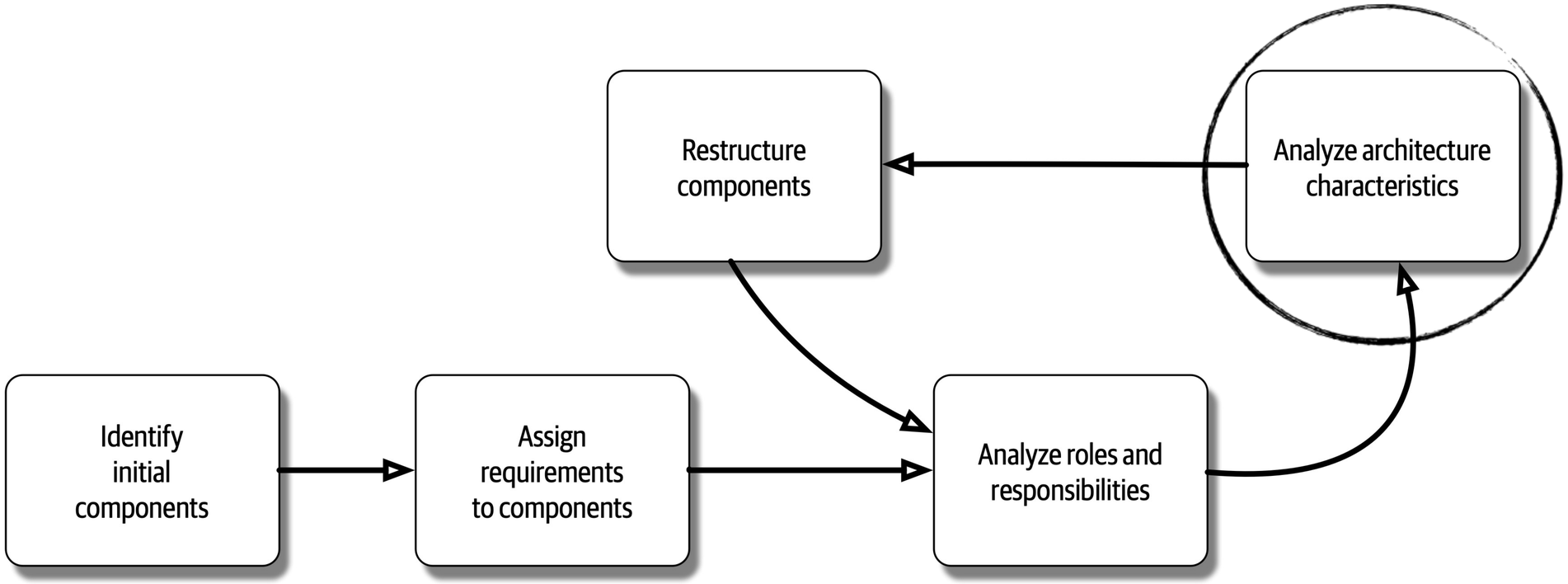


Your Architectural Kata is...

Going Going Gone!

Your Architectural Kata is...

Going Going Gone!

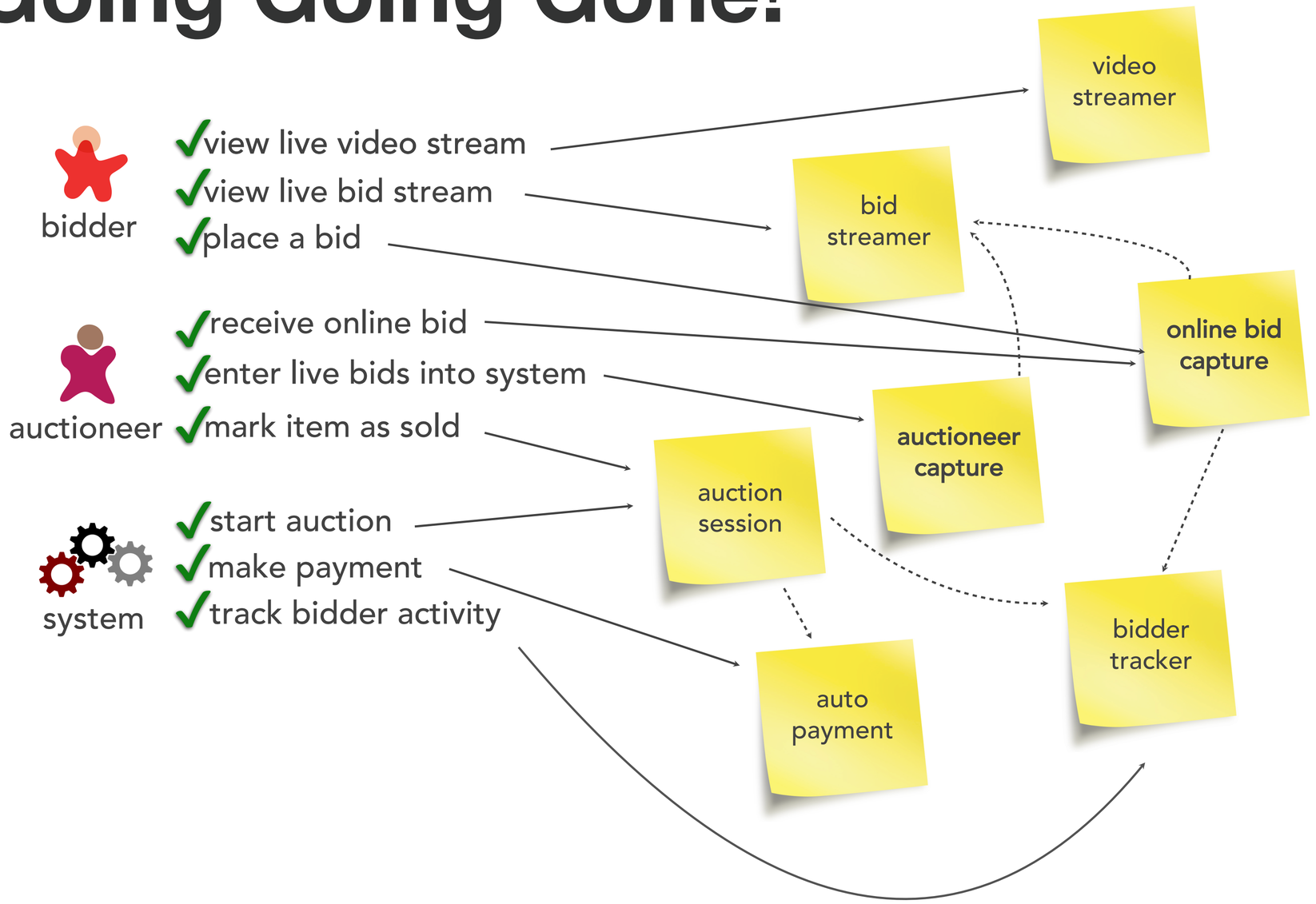


Your Architectural Kata is...

Going Going Gone!

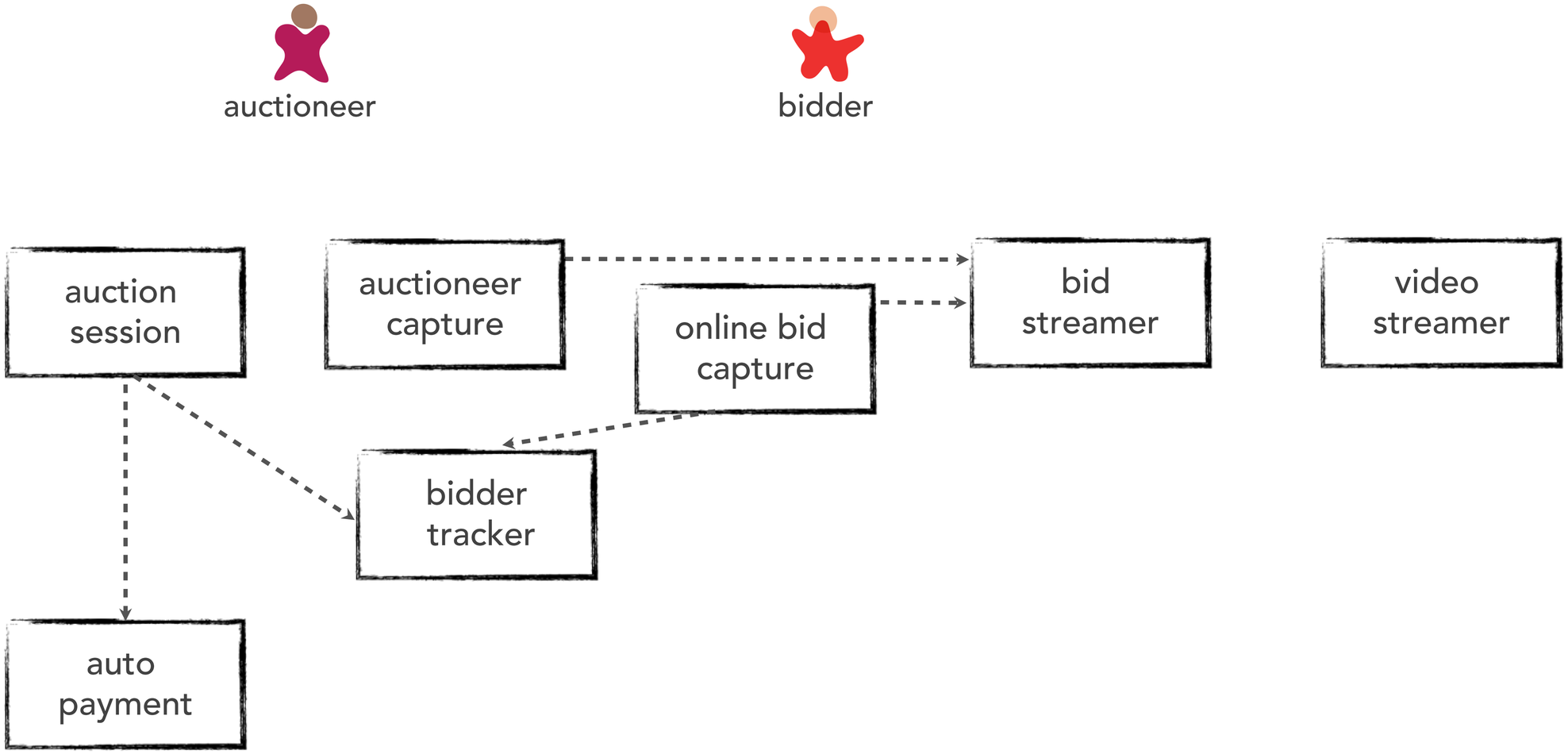
Your Architectural Kata is...

Going Going Gone!



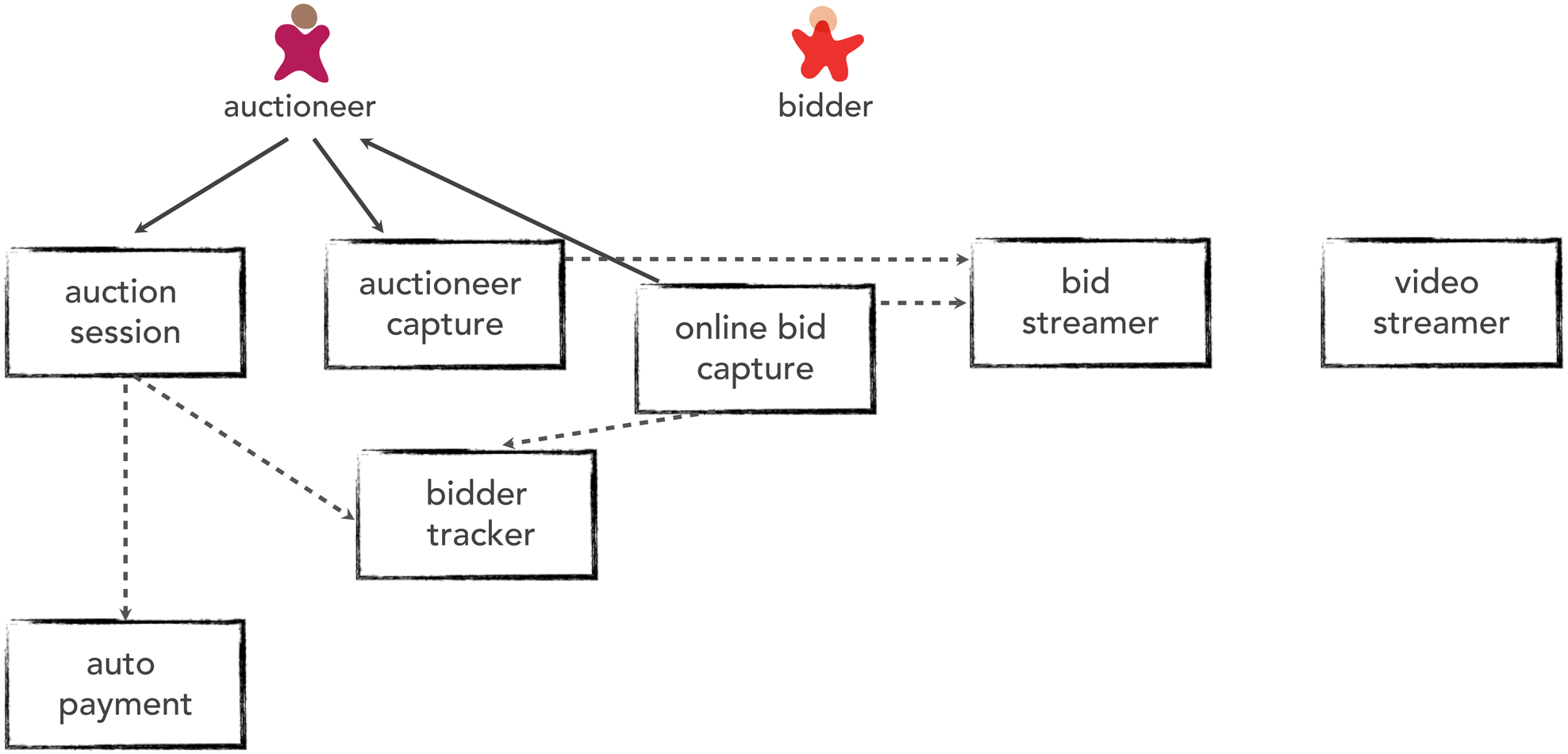
Your Architectural Kata is...

Going Going Gone!



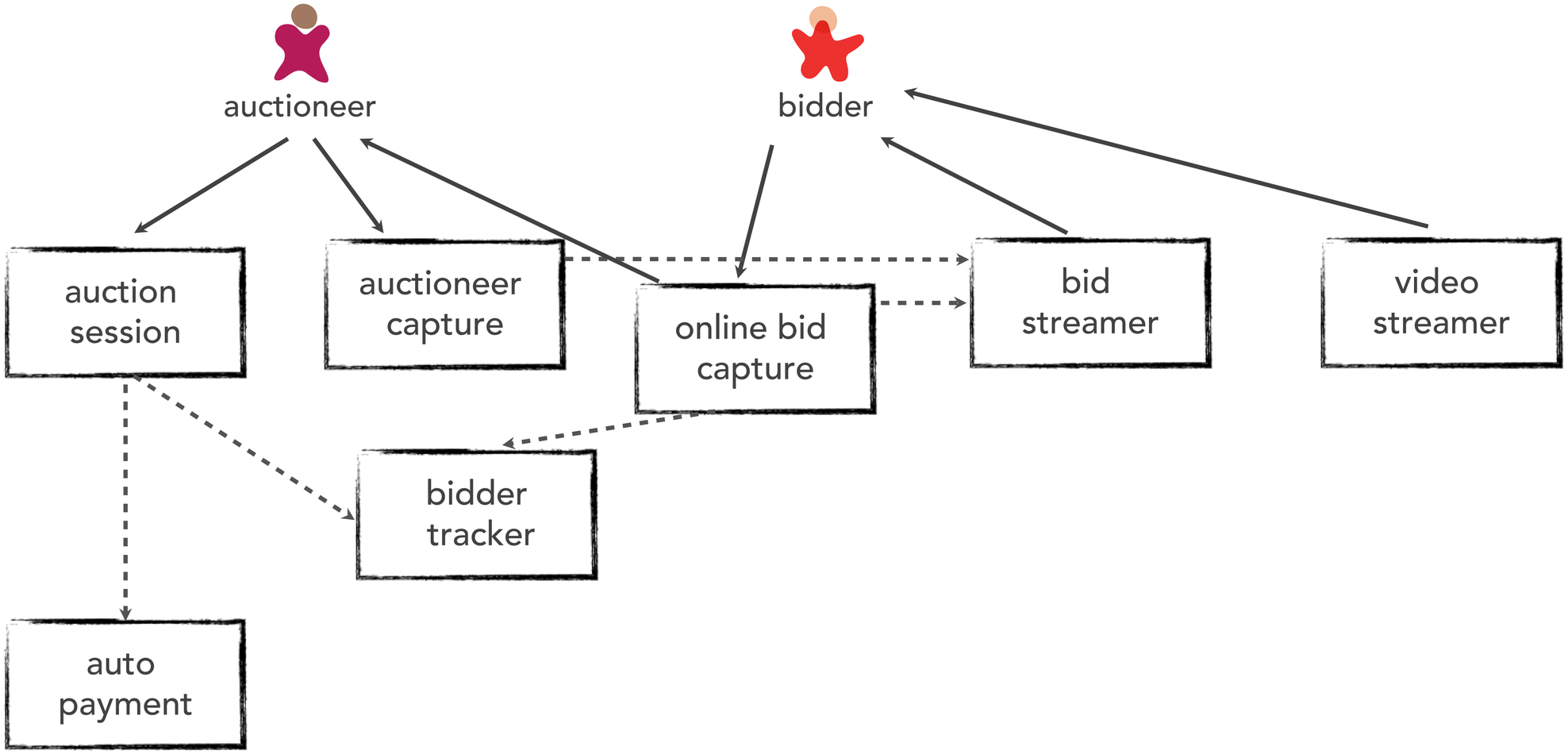
Your Architectural Kata is...

Going Going Gone!



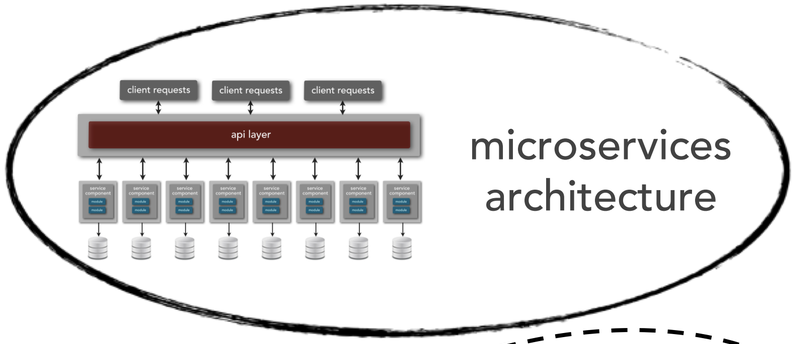
Your Architectural Kata is...

Going Going Gone!

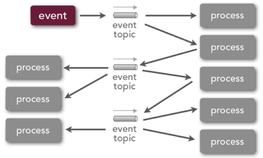


Your Architectural Kata is...

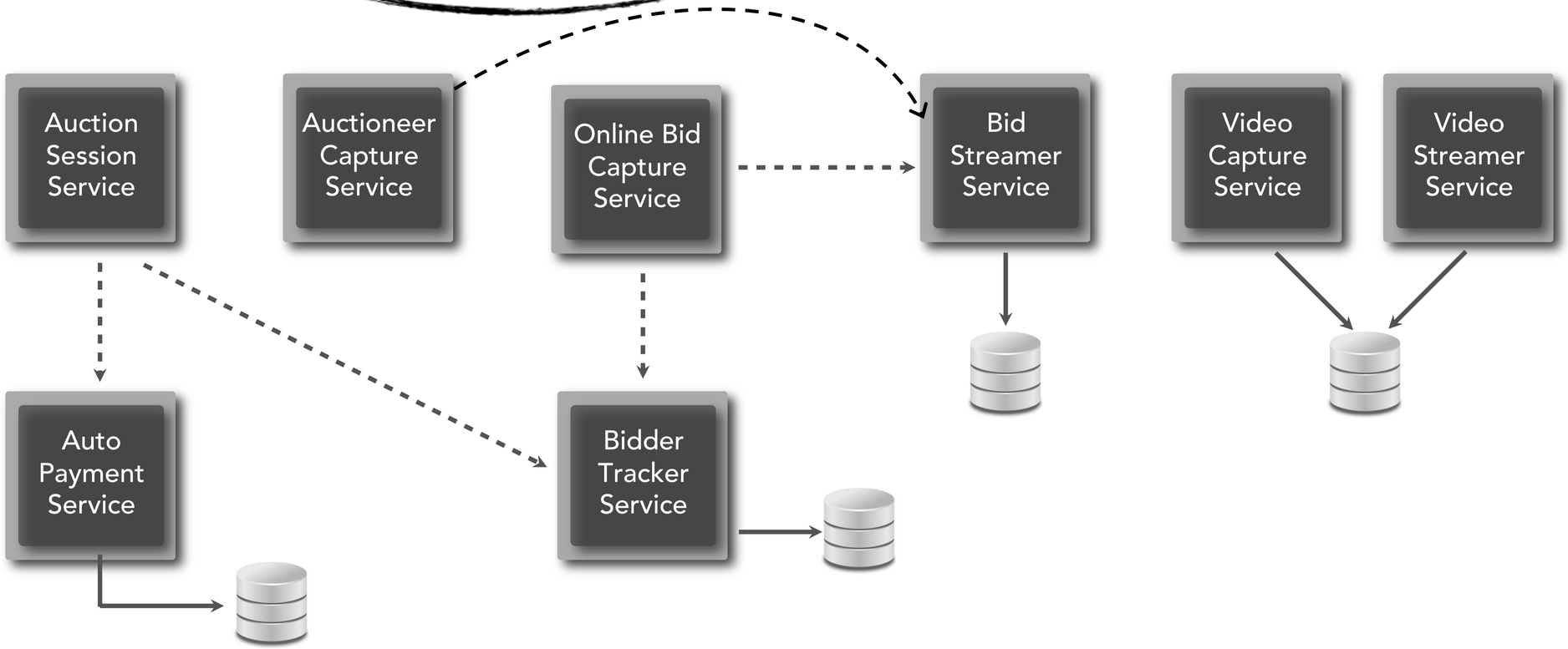
Going Going Gone!



microservices architecture

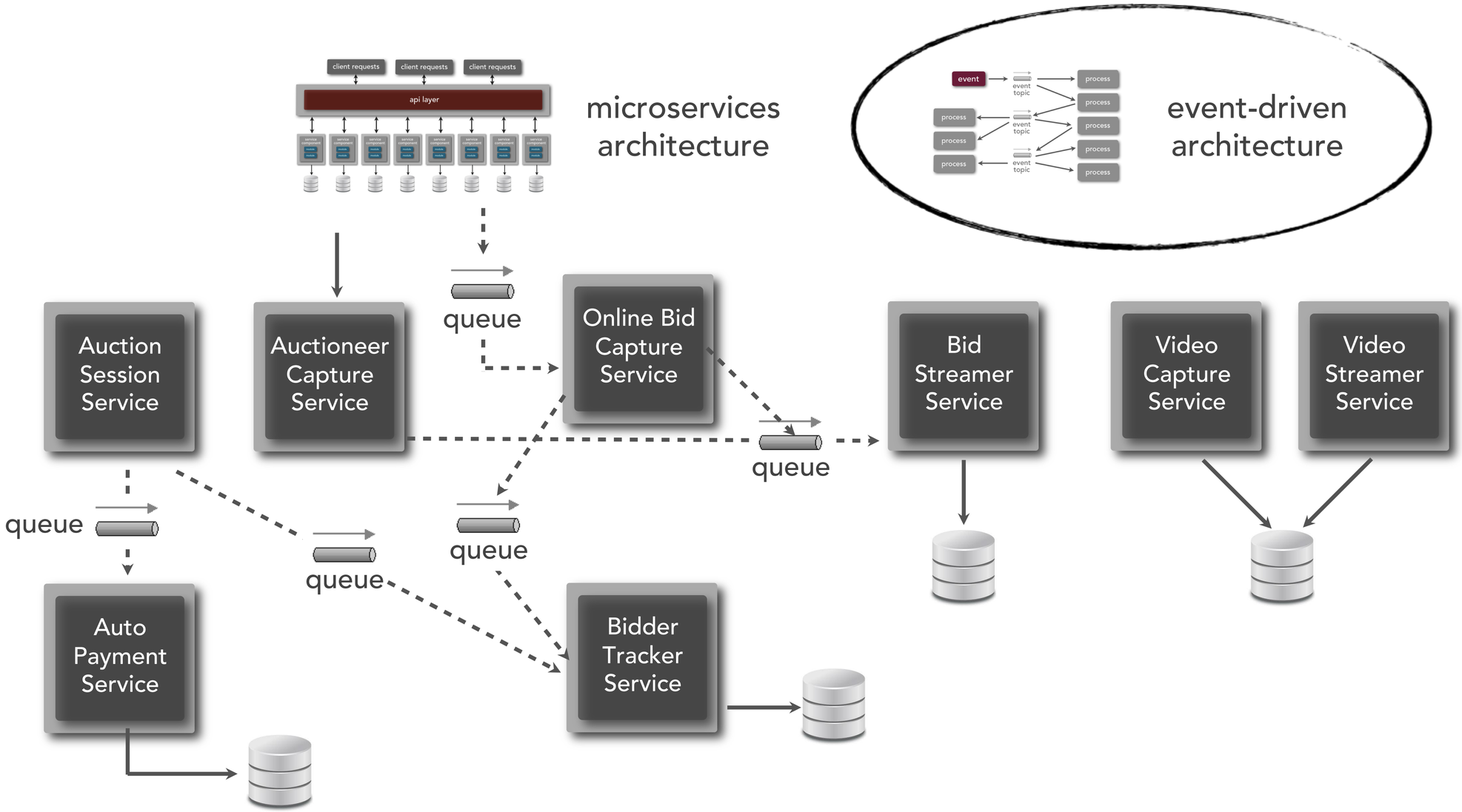


event-driven architecture



Your Architectural Kata is...

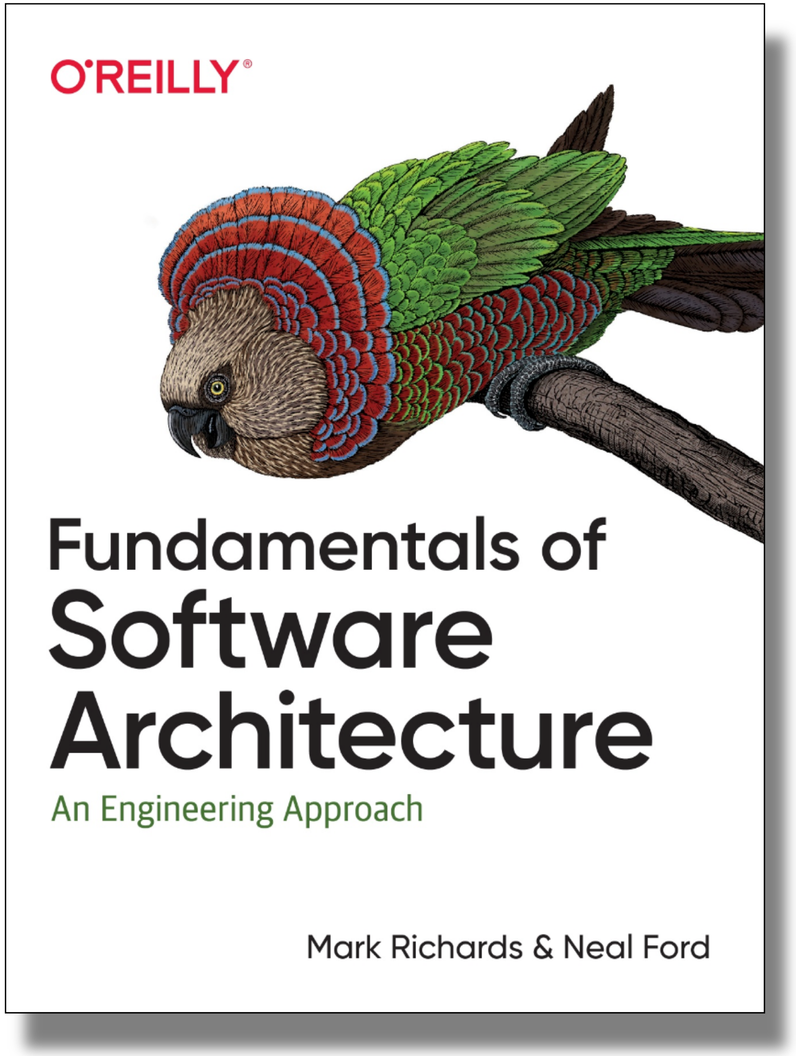
Going Going Gone!



documenting software architecture

documenting software architecture

documenting software architecture



Second Law of Software Architecture

"Why is more important than how"

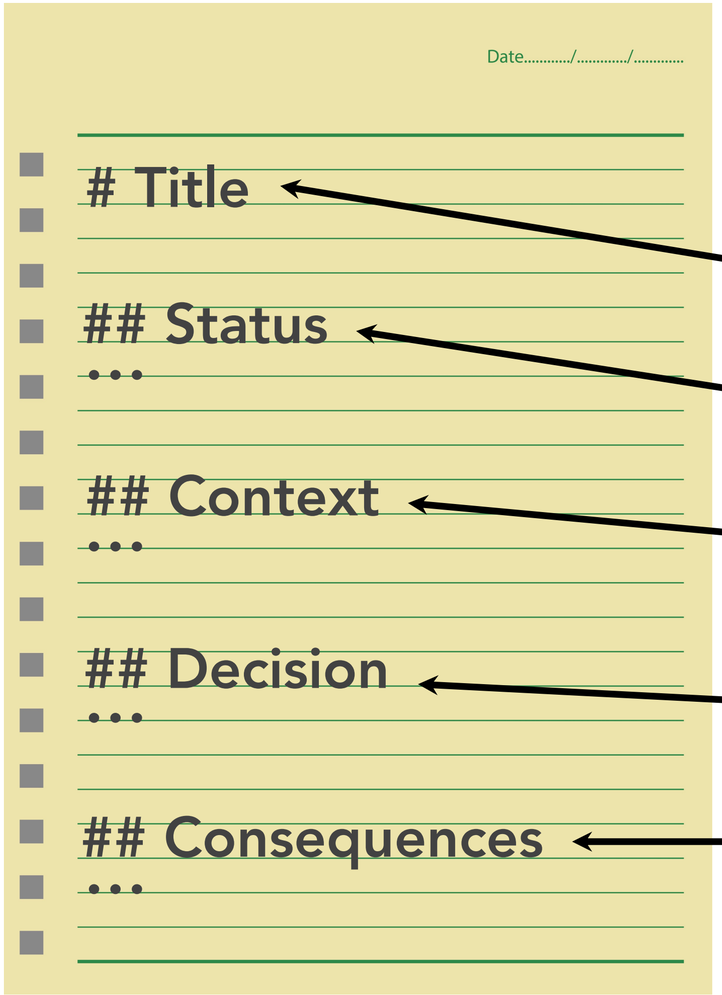
documenting software architecture

"We will keep a collection of records for *architecturally significant decisions*: those that affect the structure, non-functional characteristics, dependencies, interfaces, or construction techniques."

- Michael Nygard

http://thinkrelevance.com/blog/2011/11/15/documenting-architecture-decisions

documenting software architecture



short text file; 1-2 pages long, one file per decision
markdown, textile, asciidoc, plaintext, etc.

short noun phrase

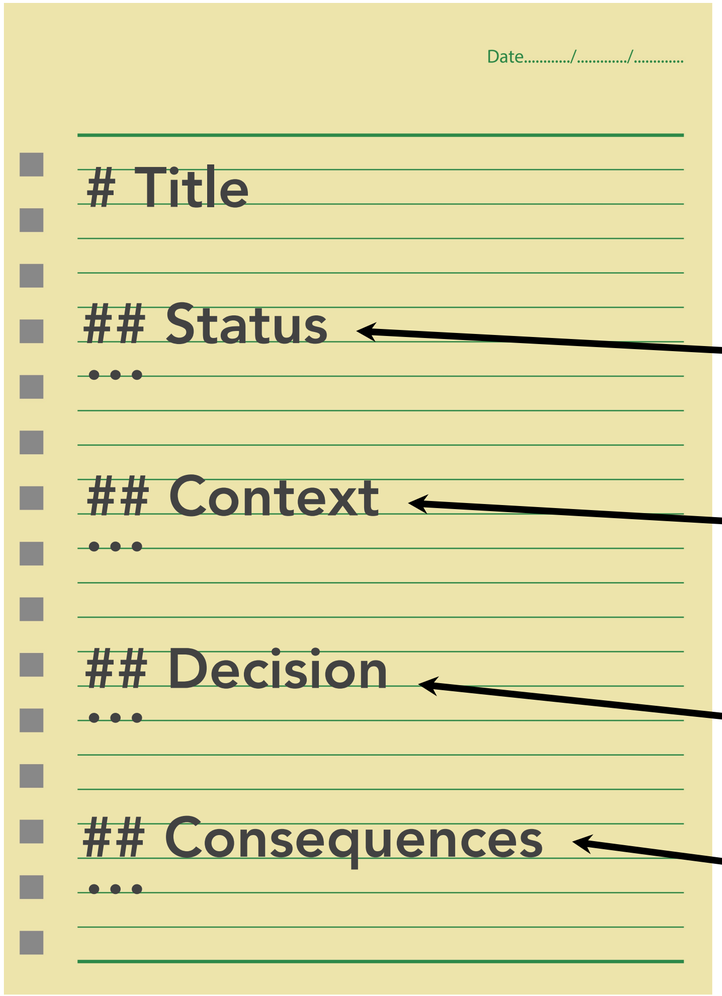
proposed, accepted, superseded

forces at play

response to forces

context after decision is applied

documenting software architecture



short text file; 1-2 pages long, one file per decision
markdown, textile, asciidoc, plaintext, etc.

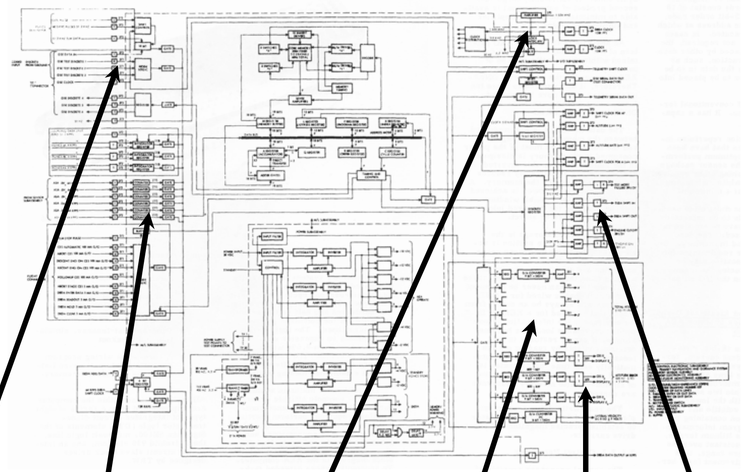
forces criteria for knowing when an architect must seek approval for a decision

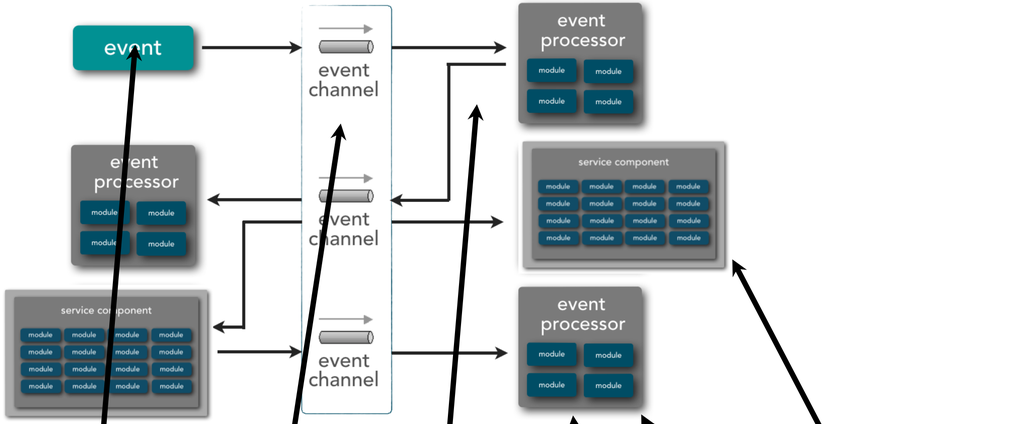
description of the problem and alternative solutions available (documentation)

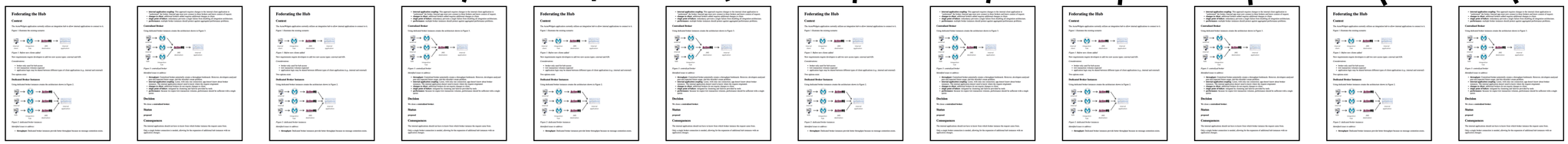
justification (the “why”)

tradeoffs and impact of decision

documenting software architecture







Your Architectural Kata is...

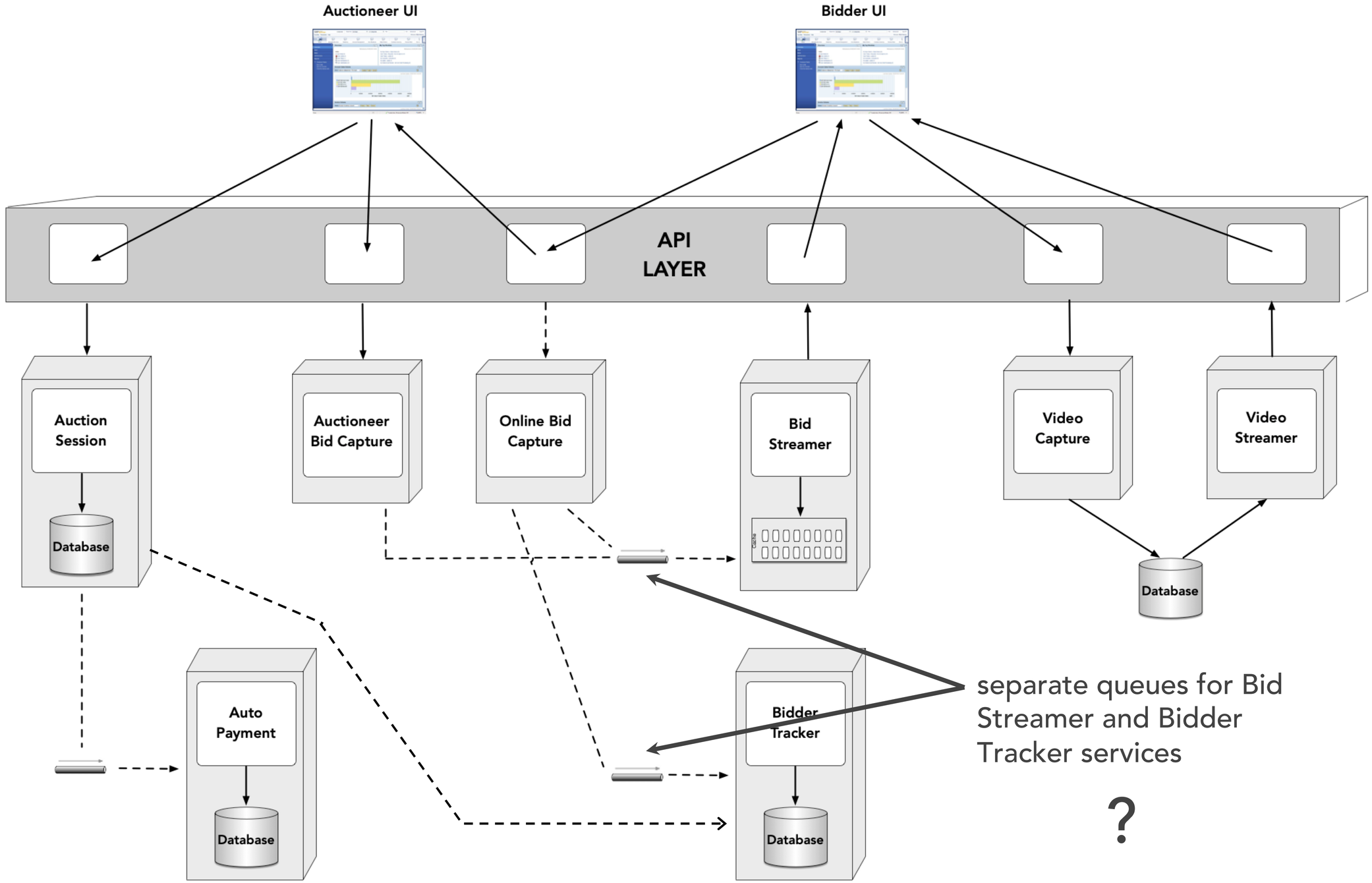
Going Going Gone!

An auction company wants to take their auctions online to a nationwide scale--customers choose the auction to participate in, wait until the auction begins, then bid during the live auction as if they were there in the room, with the auctioneer.

- **Users**: scale up to hundreds of participants (per auction), potentially up to thousands of participants, and as many simultaneous auctions as possible
- **Requirements**:
  - bidders can see a live video stream of the auction and see all bids as they occur
  - auctions must be as real-time as possible
  - both online and live bids must be received in the order in which they are placed
  - bidders register with credit card; system automatically charges card if bidder wins
  - participants must be tracked via a reputation index
- **Additional Context:**
  - auction company is expanding aggressively by merging with smaller competitors
  - if nationwide auction is a success, replicate the model overseas
  - budget is not constrained--this is a strategic direction
  - company just exited a lawsuit where they settled a suit alleging fraud

Your Architectural Kata is...

Going Going Gone!



1. Separate Queues for Bid Streamer and Tracker Services

Status

Accepted

Context

The Bid Capture Services, upon receiving a bid, must forward that bid to the Bid Streamer Service and the Bidder Tracker Service. This could be done using a single topic (pub/sub) or separate queues (p2p) for each service.

Decision

We will use separate queues for the Bid Streamer and Bidder Tracker services.

Multiple bids will come in for the same ask amount. The Streamer service only needs the first bid received for that amount, whereas the Bidder Tracker needs all bids received. Using a topic (pub/sub) would require the Bid Streamer to contain logic to ignore bids that are the same as the prior amount, forcing the Bid Streamer to store shared state between instances.

The Bid Streamer Service stores the bids for an item in an in-memory cache, whereas the Bidder Tracker stored bids in a database. The Bidder Tracker will therefore be slower and might require back pressure. Using a dedicated Bidder Tracker queue provides this dedicated back pressure point.

Consequences

This decision will require the Bid Capture services to send the same information to multiple queues.

Use of Micro-kernel Architecture

Status
PROPOSED

Context
Two key requirements of the system (_promotions_ anc

The current design features a modular monolith archite

![modular monolith](fig1_modular_monolith.jpg)
Figure 1: the current state architecture

Currently, stores must specify custom behavior (produc

All local customizations reside in one service and in on

To allow stores to most easily add and customize local

![microkernel architecture](fig2_microkernel.jpg)
Figure 2: proposed microkernel architecture

The new design allows easy update of global policy (pr

Decision
The architects decided to migrate the current monolith

Consequences
The architects take advantage of the restructuring oppo

The new design also incorporates the BFF patterns, dis

The new design will greatly improve the customization

- the local store plug-in architecture certifies customizat
- promotions within threshold values go live within 15 n
- all stores work with generic workflows via the core sys
  - promotions
  - location exemptions
  - local products

quirements.

e 1_.

FTP. Operations must certify the JAR, leading to dela

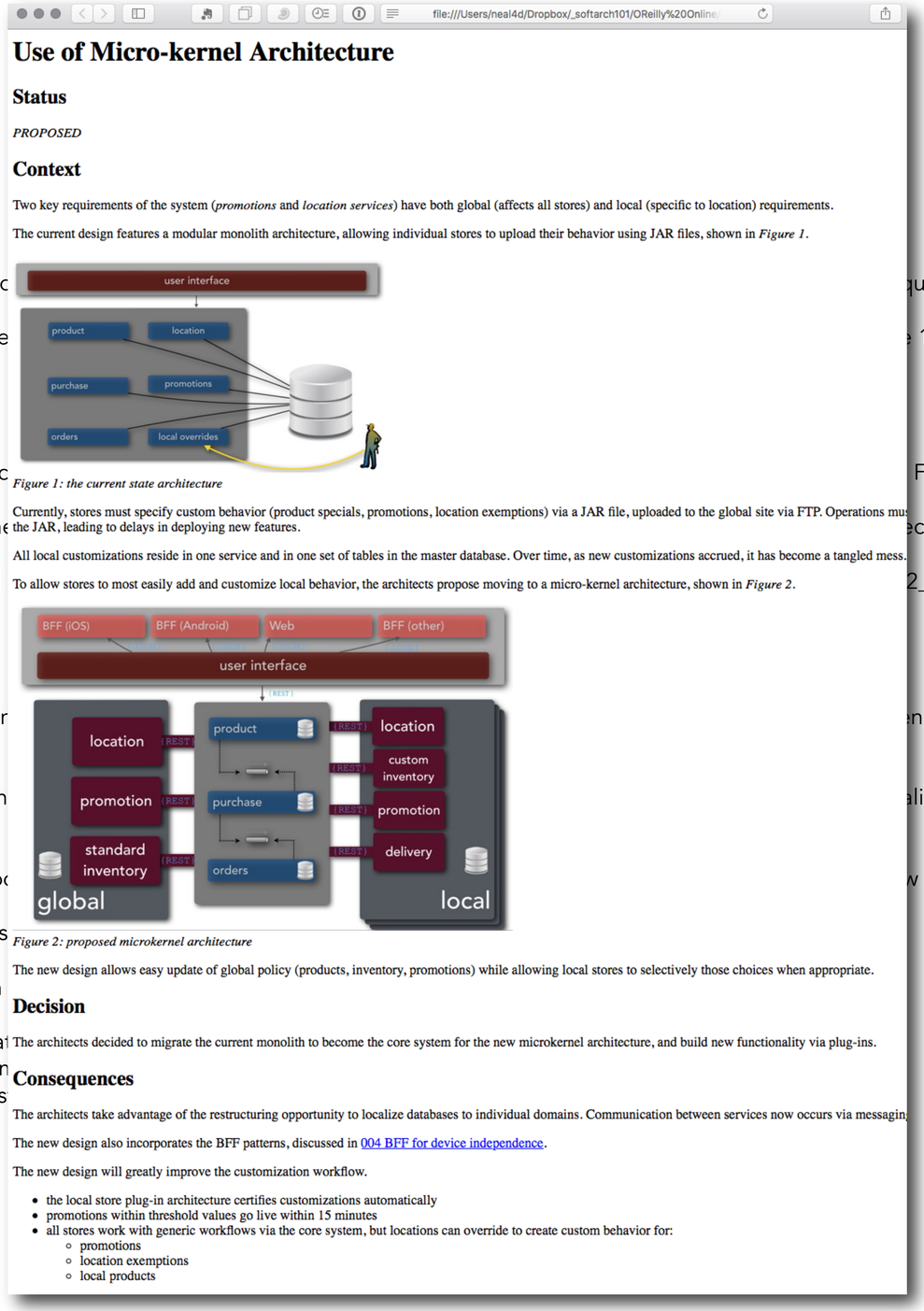
ecome a tangled mess.

2_.

n appropriate.

ality via plug-ins.

w occurs via messaging.



Use c

Statu
_PROP(

Con
Two ke

The cur

![modu
_Figure

Current

All loca

To allo

![microl
_Figure

The ne

Dec
The arc

Con
The arc

The ne

The ne

- the lo
- promo
- all sto
  - prom
  - locat
  - local

file:///Users/neal4d/Dropbox/_softarch101/OReilly%20Online/

Use of Micro-kernel Architecture

Status

PROPOSED

Context

Two key requirements of the system (*promotions* and *location services*) have both global (affects all stores) and local (specific to location) requirements.

The current design features a modular monolith architecture, allowing individual stores to upload their behavior using JAR files, shown in *Figure 1*.

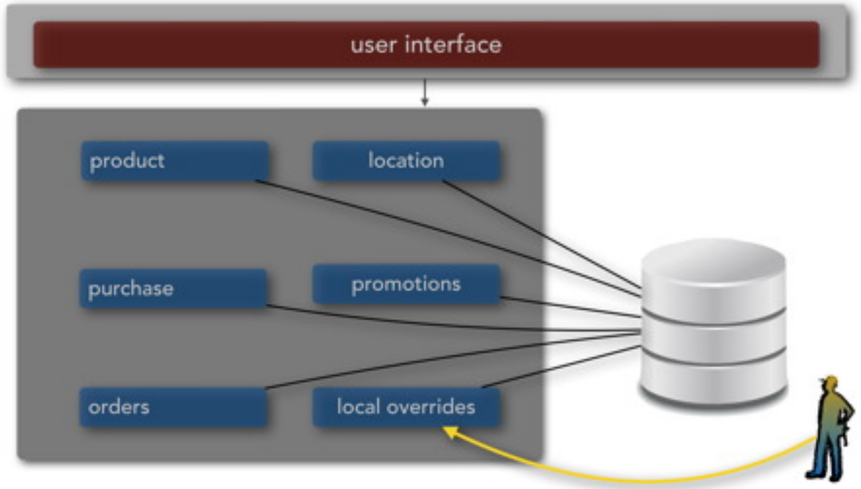


Figure 1: the current state architecture

Currently, stores must specify custom behavior (product specials, promotions, location exemptions) via a JAR file, uploaded to the global site via FTP. Operations n
the JAR, leading to delays in deploying new features.

All local customizations reside in one service and in one set of tables in the master database. Over time, as new customizations accrued, it has become a tangled me

Meet the Judges!

Meet the Client !!

Logistics and details

Q&A

Go forth & do some architecture!