Toloka

Toloka

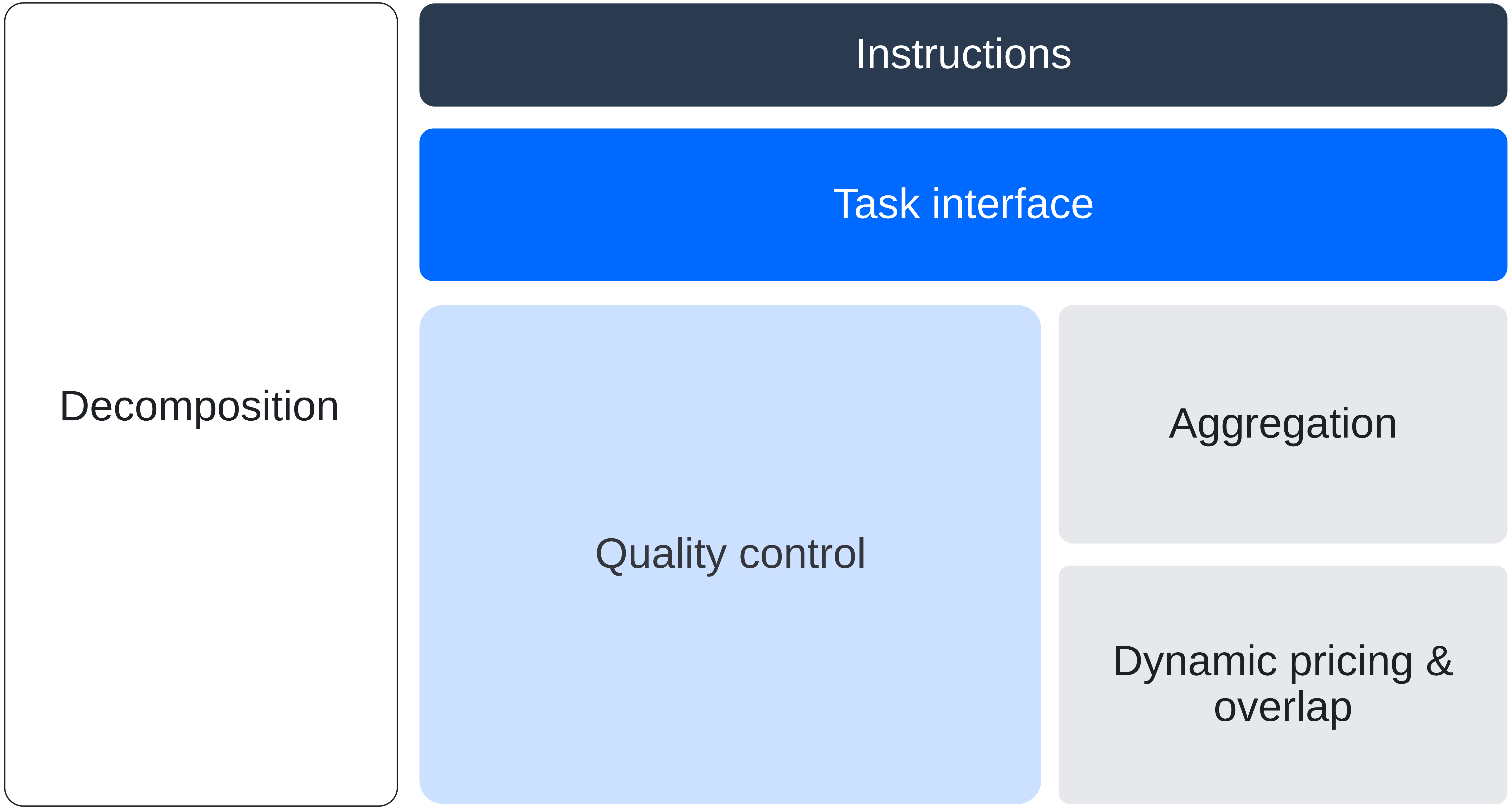
Crowdsourcing for Information Retrieval

Dmitry Ustalov, Natalia Fedorova, Nikita Pavlichenko, Alisa Smirnova, Daniil Likhobaba

Toloka

Part II
Crowdsourcing Essentials

Alisa Smirnova, Head of Research at Toloka


Instructions
Task interface
Decomposition
Quality control
Aggregation
Dynamic pricing & overlap

Decomposition

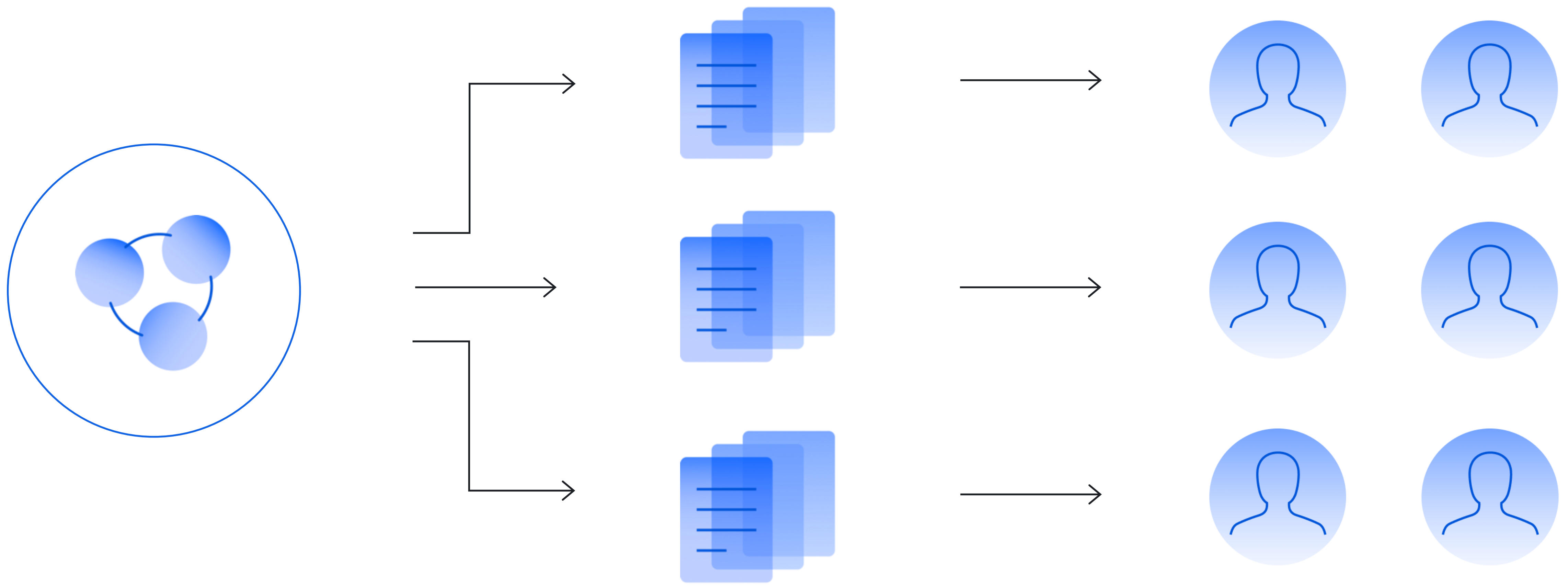
Decomposition: why?

- Annotators are usually non-experts in your specific task
- A simpler task means that:
  - More people can handle your task
  - Instructions are easier to write and follow
  - You get better results
- This is an effective way to:
  - Separate tasks of different difficulty levels
  - Control and optimize pricing
  - Control quality with manual review

Decomposition: when?

- If
  - Your task has more than 3-5 answer options to choose from
  - Your task has long instructions that are hard to read
- Then your task requires decomposition

Decomposition

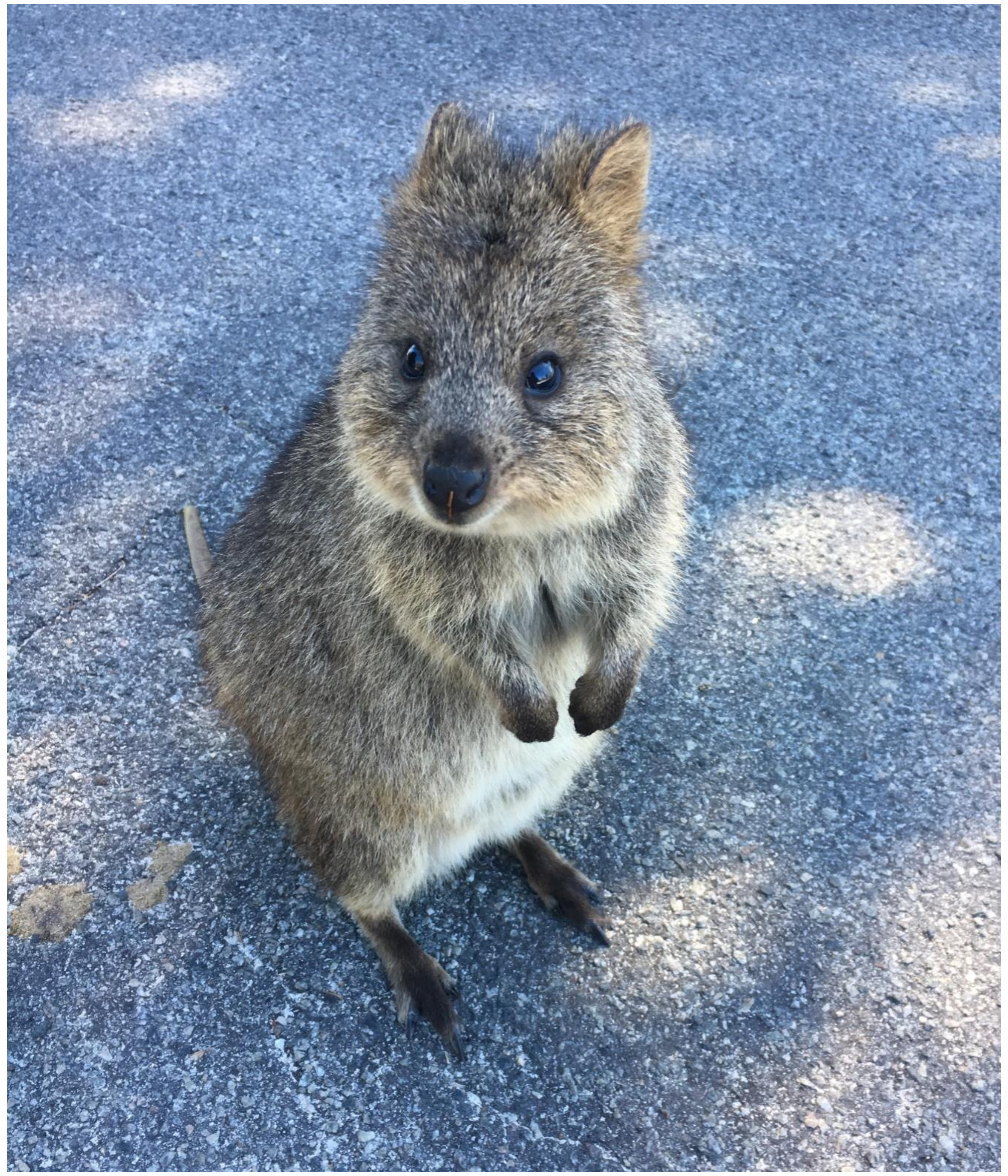


A complex task

Projects with different types of micro tasks

Crowd

Case for decomposition: a lot of questions



Bad practice: All questions in one task

What animal is in the photo?
- Cat
- Rabbit
- Bear
- Whale
- Koala
- None of the above

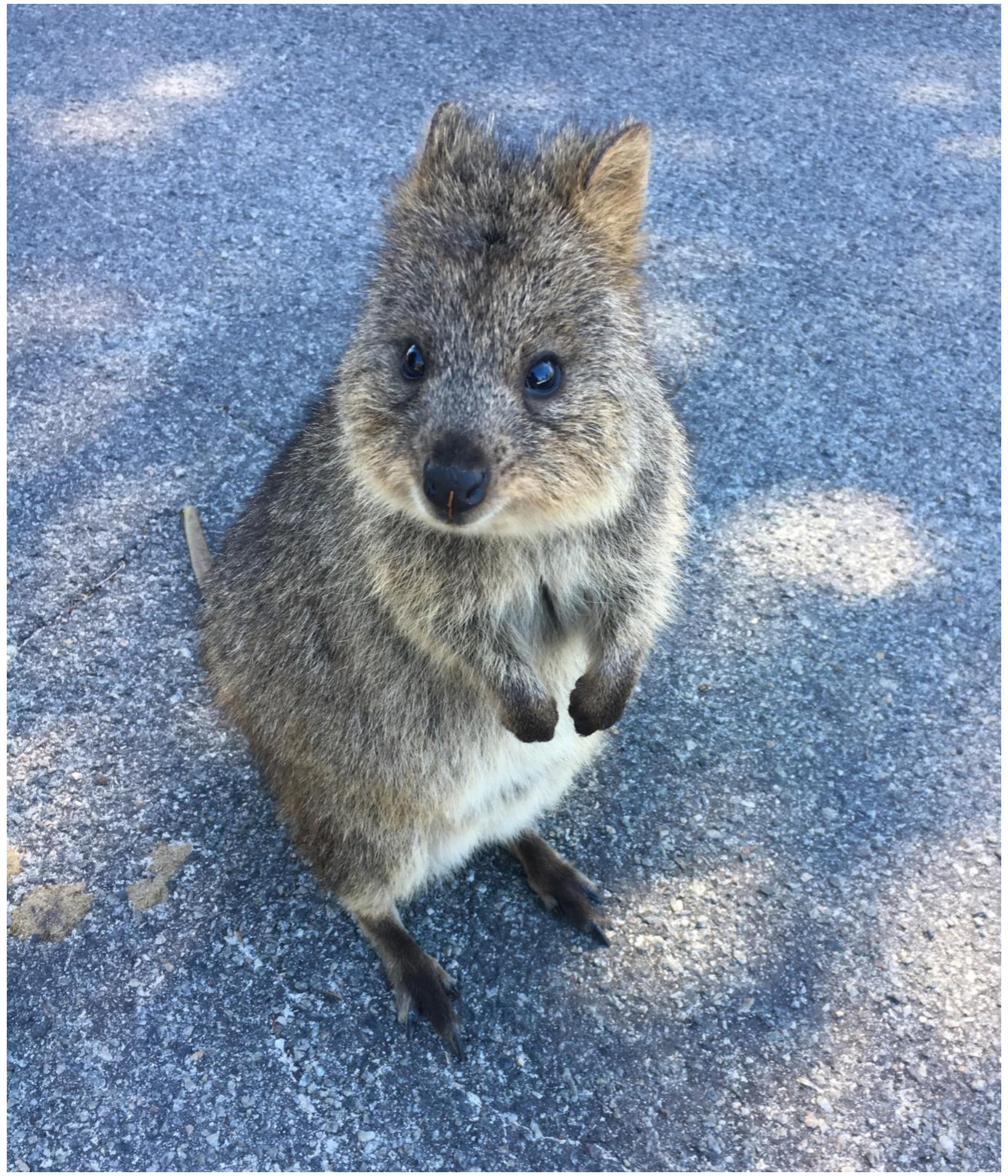
Is its tail visible?
- Yes
- No

Is it running?
- Yes
- No

What color is it?
- White
- Black
- Brown
- Red
- Other

Where is it?
- On the grass
- On a tree
- On a road
- It is flying
- None of the above

Case for decomposition: a lot of questions



Good practice: Each question in a separate task

What animal is in the photo?
- Cat
- Rabbit
- Bear
- Whale
- Koala
- None of the above

Is its tail visible?
- Yes
- No

Is it running?
- Yes
- No

What color is it?
- White
- Black
- Brown
- Red
- Other

Where is it?
- On the grass
- On a tree
- On a road
- It is flying
- None of the above

Case for decomposition: need to verify answers

The task: Outline all koalas in the photo

Problem: Outlining can be done in different ways

It is difficult to use:

— Comparison with control answers

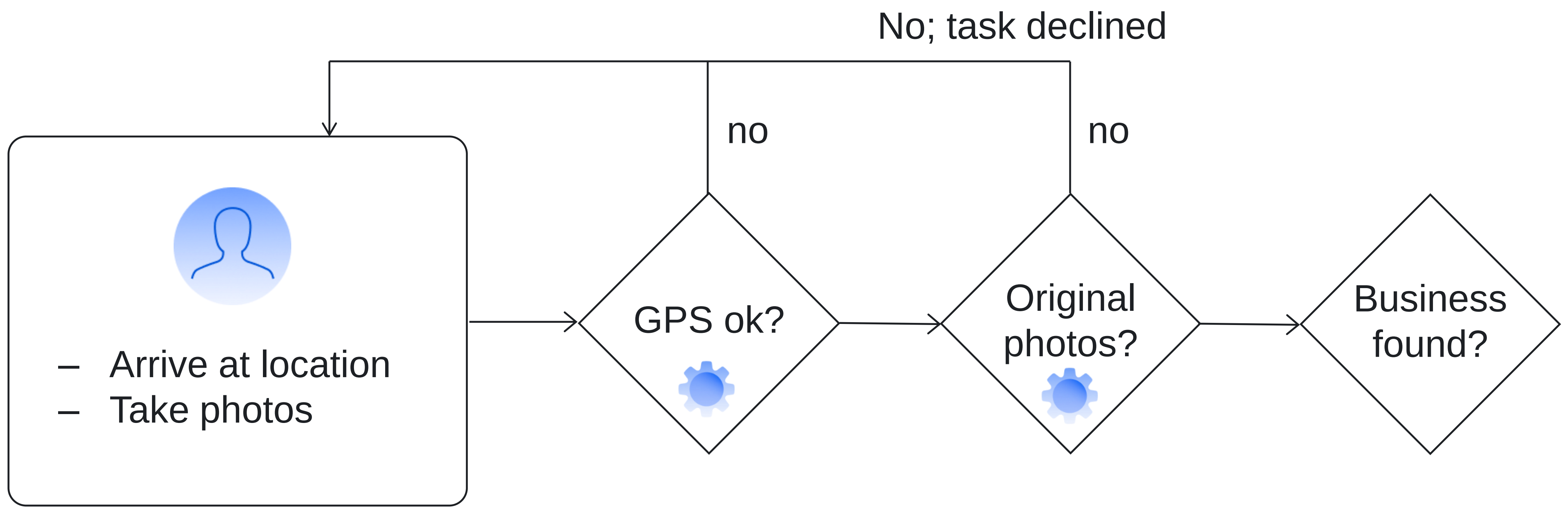
— Aggregation of answers from multiple annotators

Good solution
Create a verification task for other annotators to do:
Are all the koalas outlined correctly?

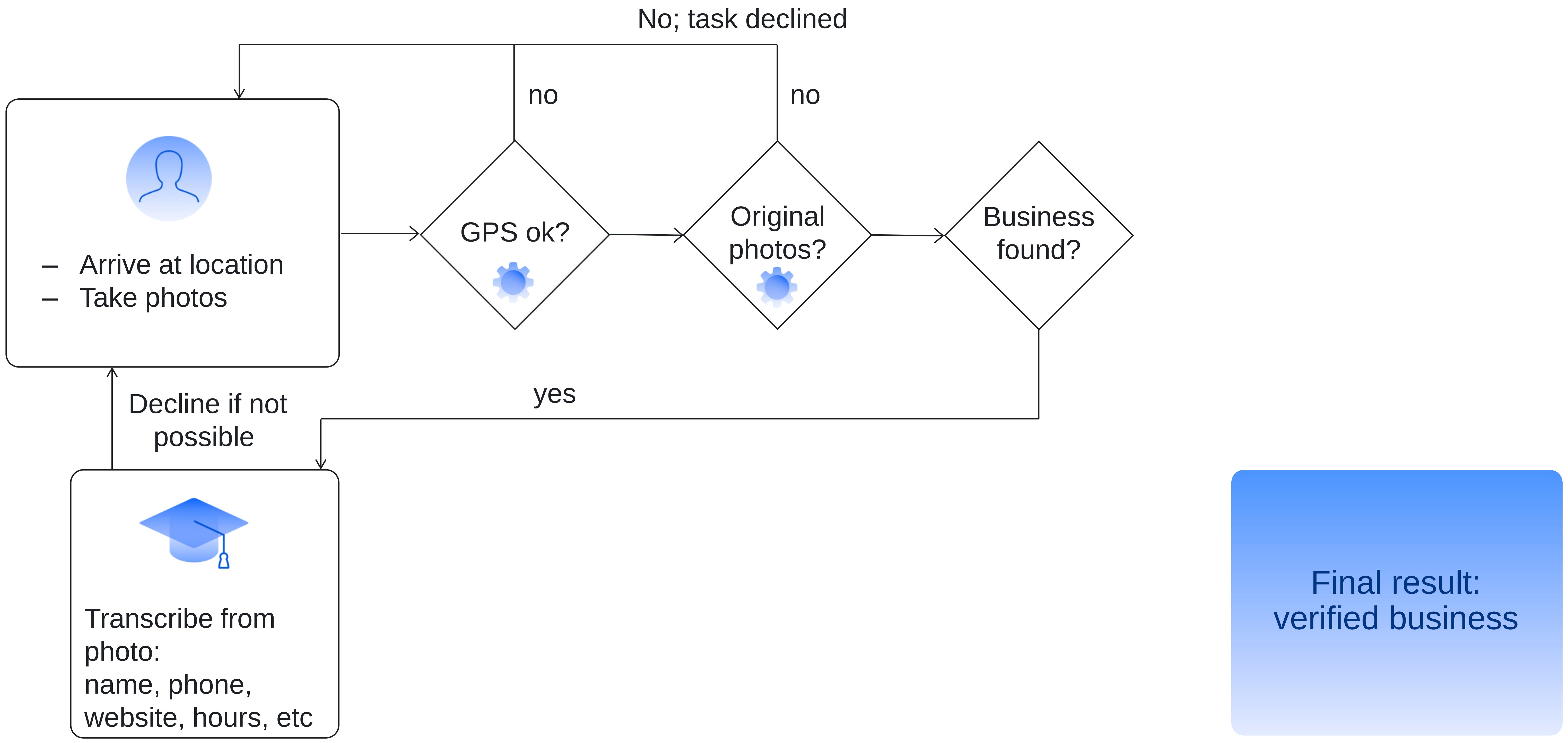
Real example: decomposition for an offline data collection task

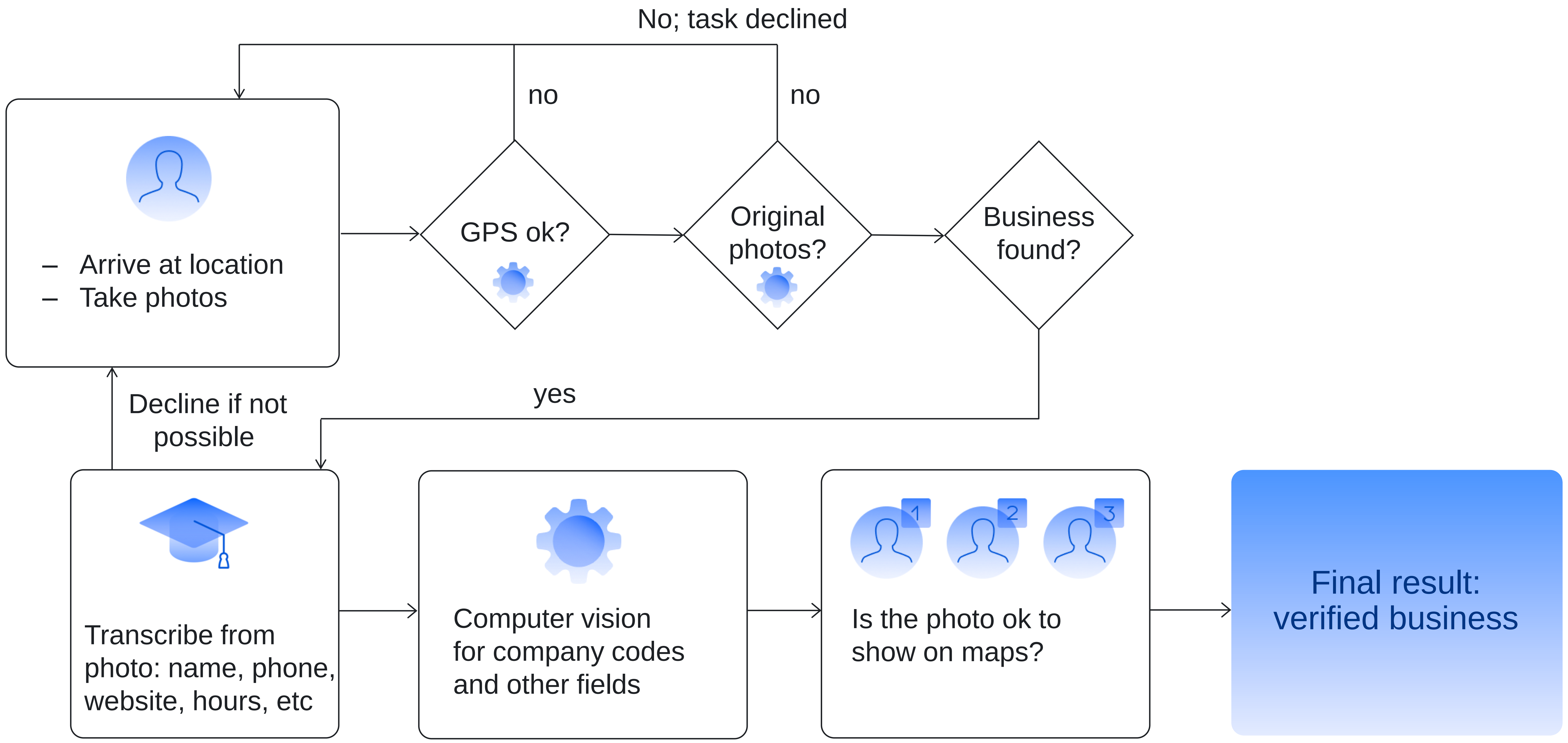
- Arrive at location
- Take photos

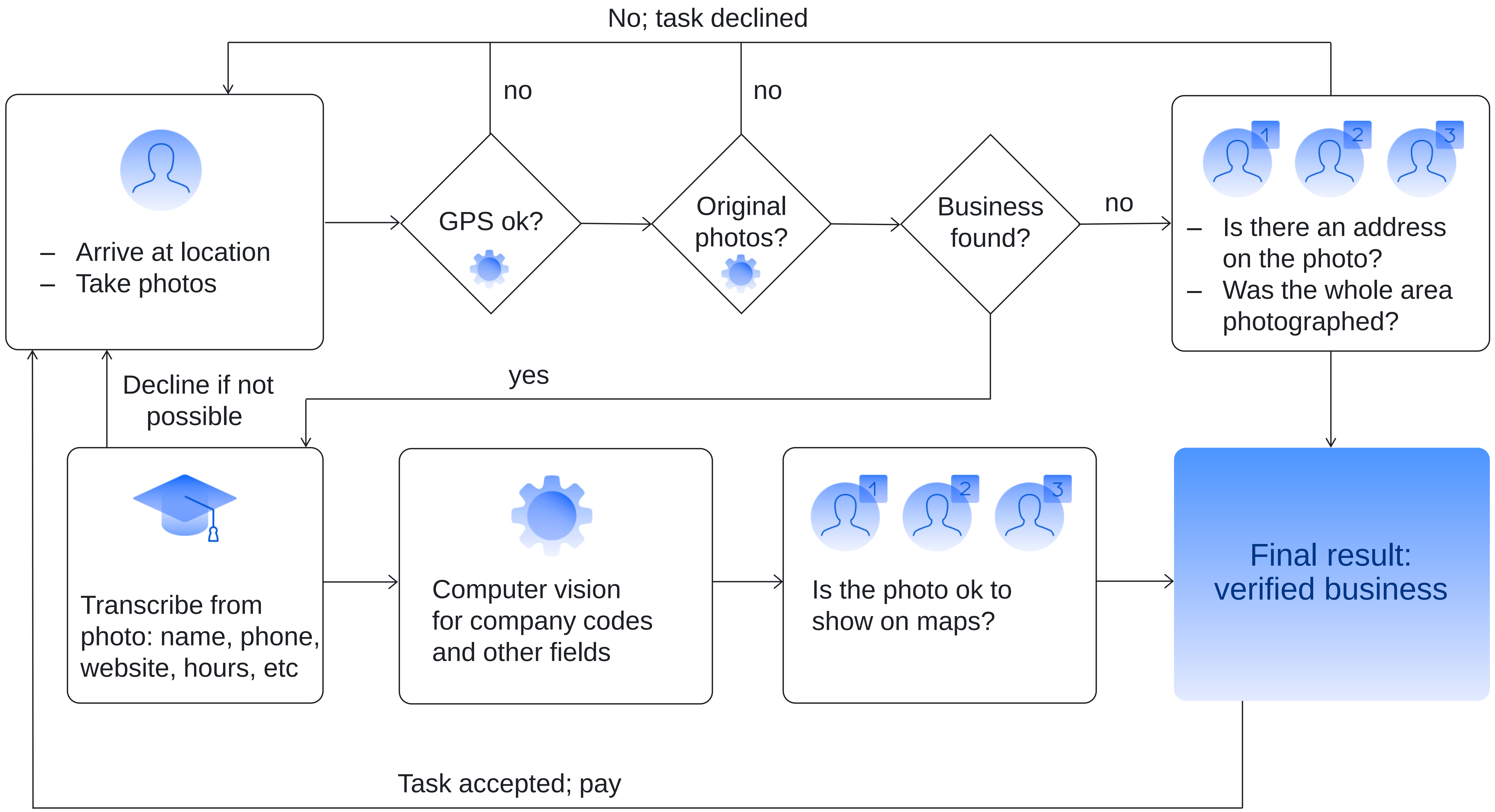
Final result:
verified business



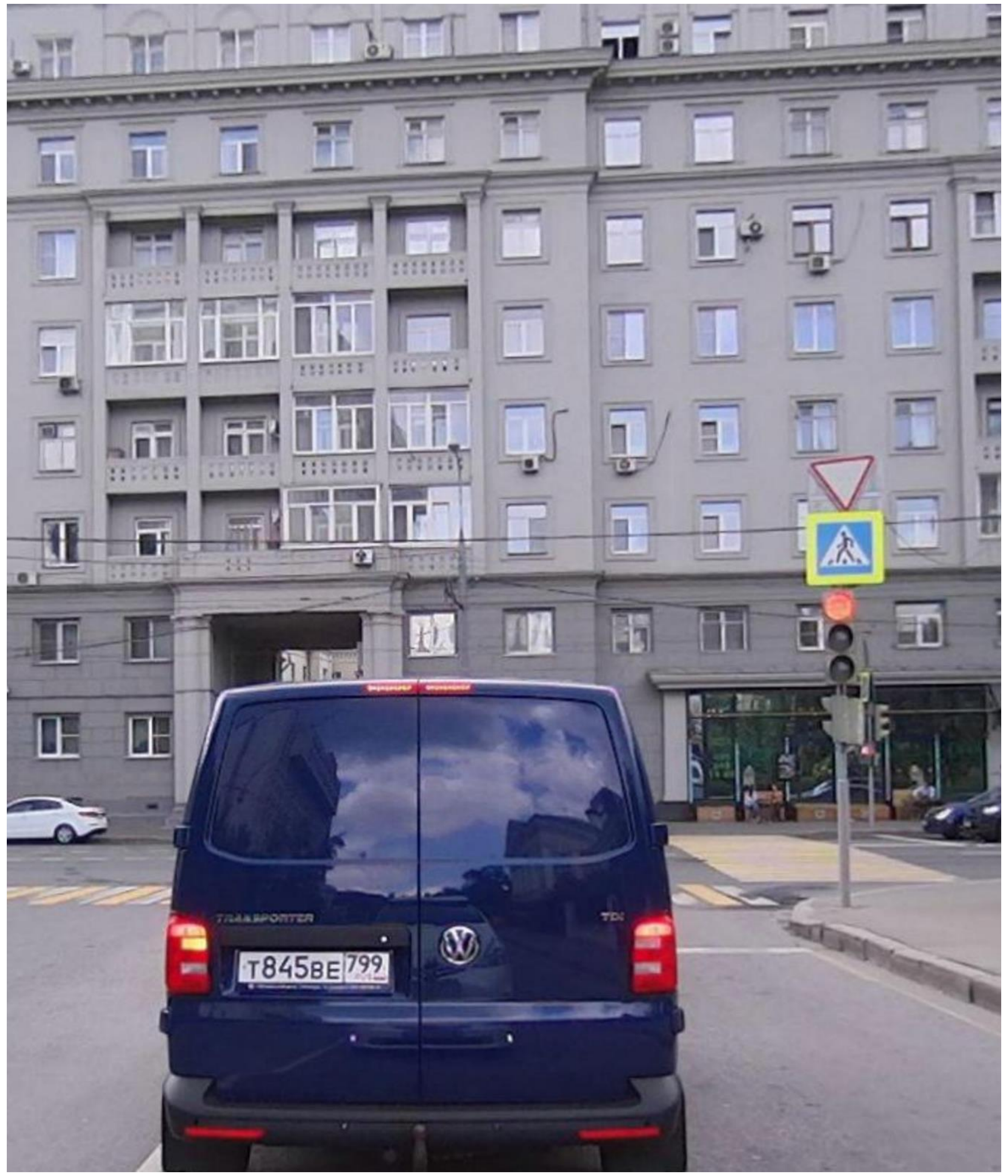
Final result:
verified business


No; task declined
no
no
– Arrive at location
– Take photos
GPS ok?
Original photos?
Business found?
yes
Decline if not possible
Transcribe from photo: name, phone, website, hours, etc
Final result: verified business


No; task declined
no
no
– Arrive at location
– Take photos
GPS ok?
Original photos?
Business found?
yes
Decline if not possible
Transcribe from photo: name, phone, website, hours, etc
Computer vision for company codes and other fields
1
2
3
Is the photo ok to show on maps?
Final result: verified business


No; task declined
no
no
– Arrive at location
– Take photos
GPS ok?
Original photos?
Business found?
no
1
2
3
– Is there an address on the photo?
– Was the whole area photographed?
Decline if not possible
yes
Transcribe from photo: name, phone, website, hours, etc
Computer vision for company codes and other fields
1
2
3
Is the photo ok to show on maps?
Final result: verified business
Task accepted; pay

Case for decomposition: a lot of questions



Bad practice: All questions in one task

What is the vehicle type?
- Car
- Bus
- Truck
- Motorcycle
- Bike
- Tractor
- None of the above

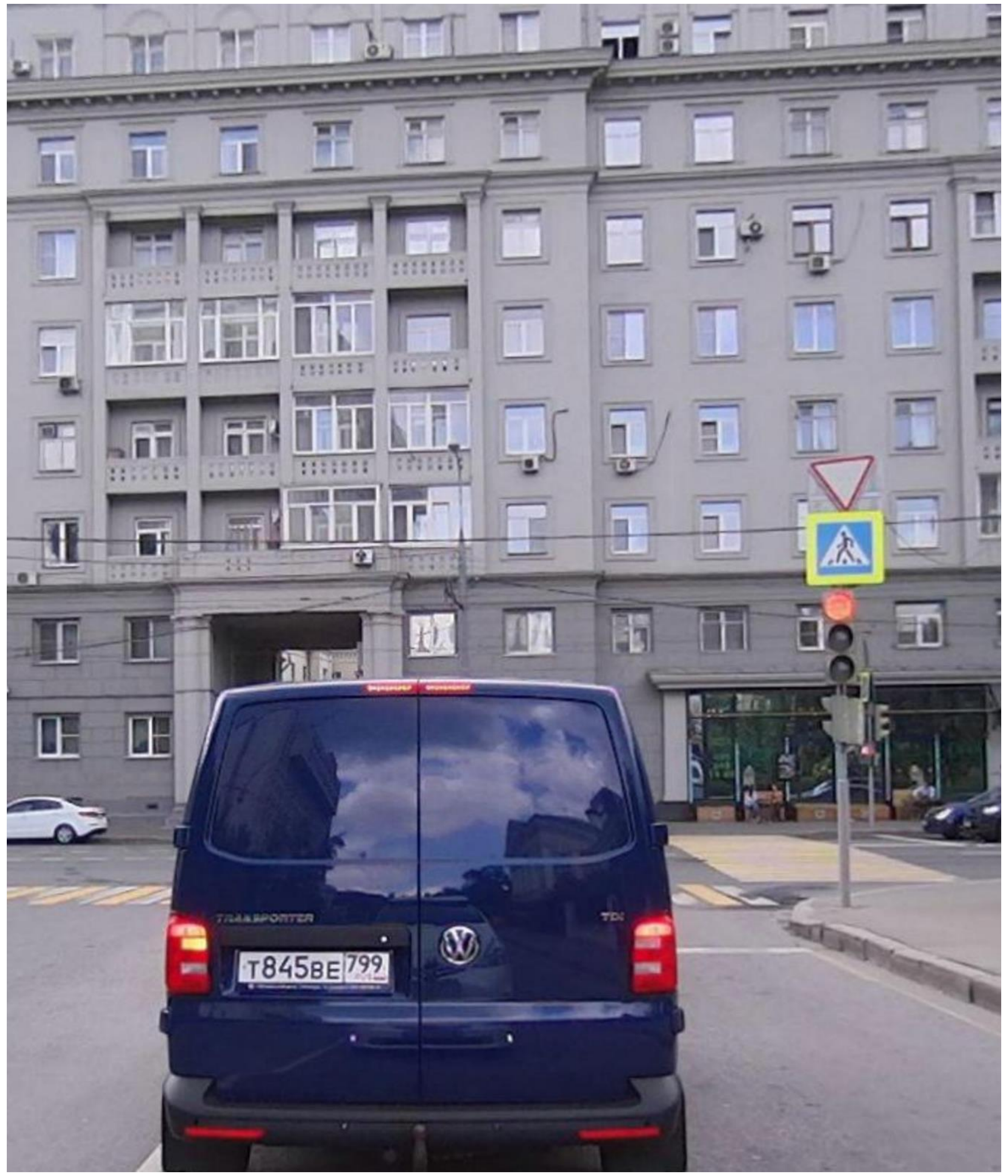
Is there a pedestrian?
- Yes
- No

Is there a traffic light?
- Yes
- No

What color is the vehicle?
- White
- Black
- Brown
- Red
- Other

Where is it?
- On the grass
- On a sidewalk
- On a road
- It is flying
- None of the above

Case for decomposition: a lot of questions



Good practice: Each question in a separate task

What is the vehicle type?
- Car
- Bus
- Truck
- Motorcycle
- Bike
- Tractor
- None of the above

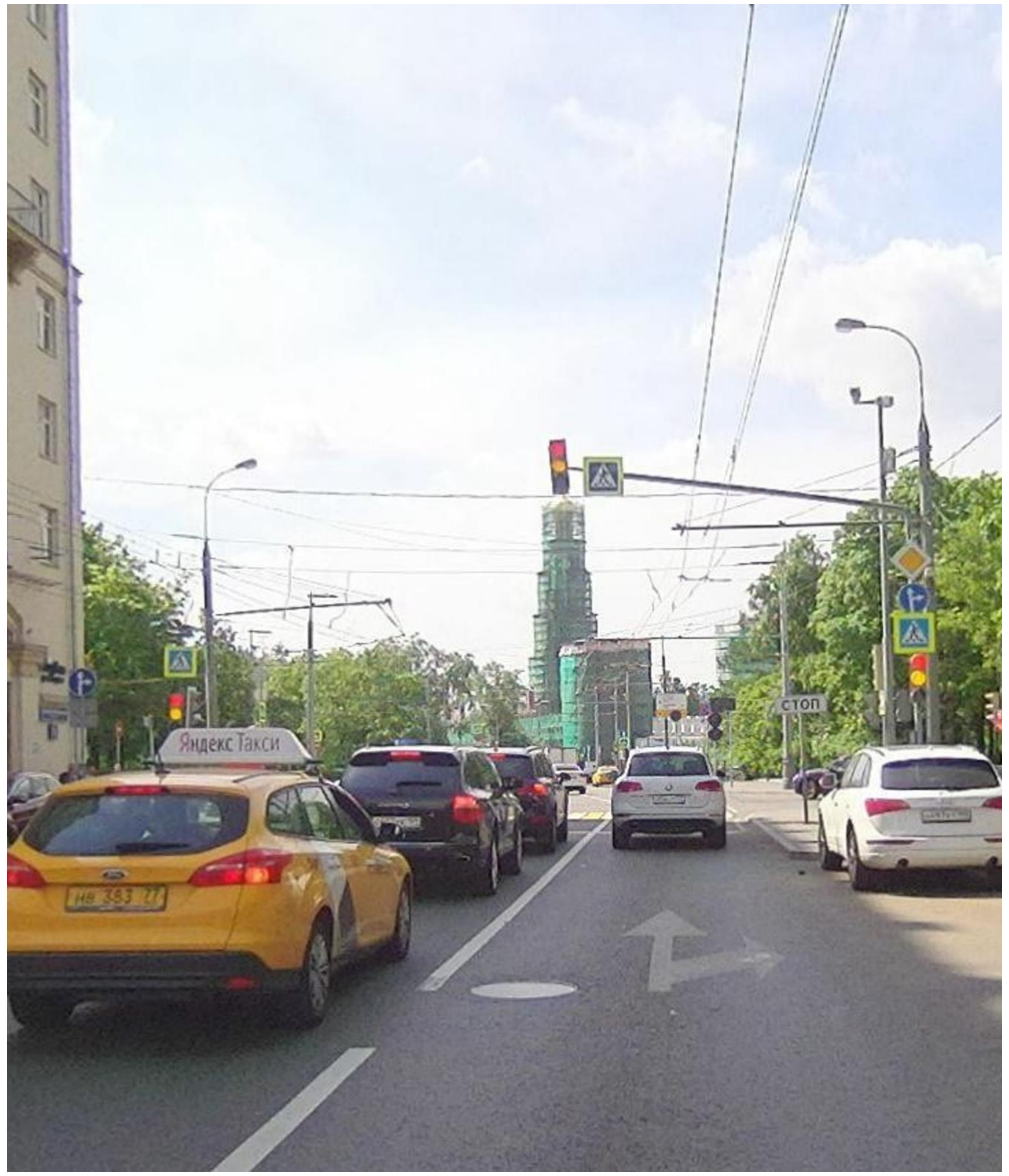
Is there a pedestrian?
- Yes
- No

Is there a traffic light?
- Yes
- No

What color is the vehicle?
- White
- Black
- Brown
- Red
- Other

Where is it?
- On the grass
- On a sidewalk
- On a road
- It is flying
- None of the above

Case for decomposition: need to verify answers



The task: Outline all cars on the photo

Problem: Outlining can be done in different ways

It is difficult to use:

- Comparison with control answers
- Aggregation of answers from multiple annotators

Good solution
Create a verification task for other annotators to do: Are all the cars outlined correctly?

Instructions

Instructions: what to include

- Goal of the task
- Interface description
- Steps to follow
- Examples of good and bad answers
- Examples of unusual cases and how to handle them correctly
- Reference materials

$\uparrow$

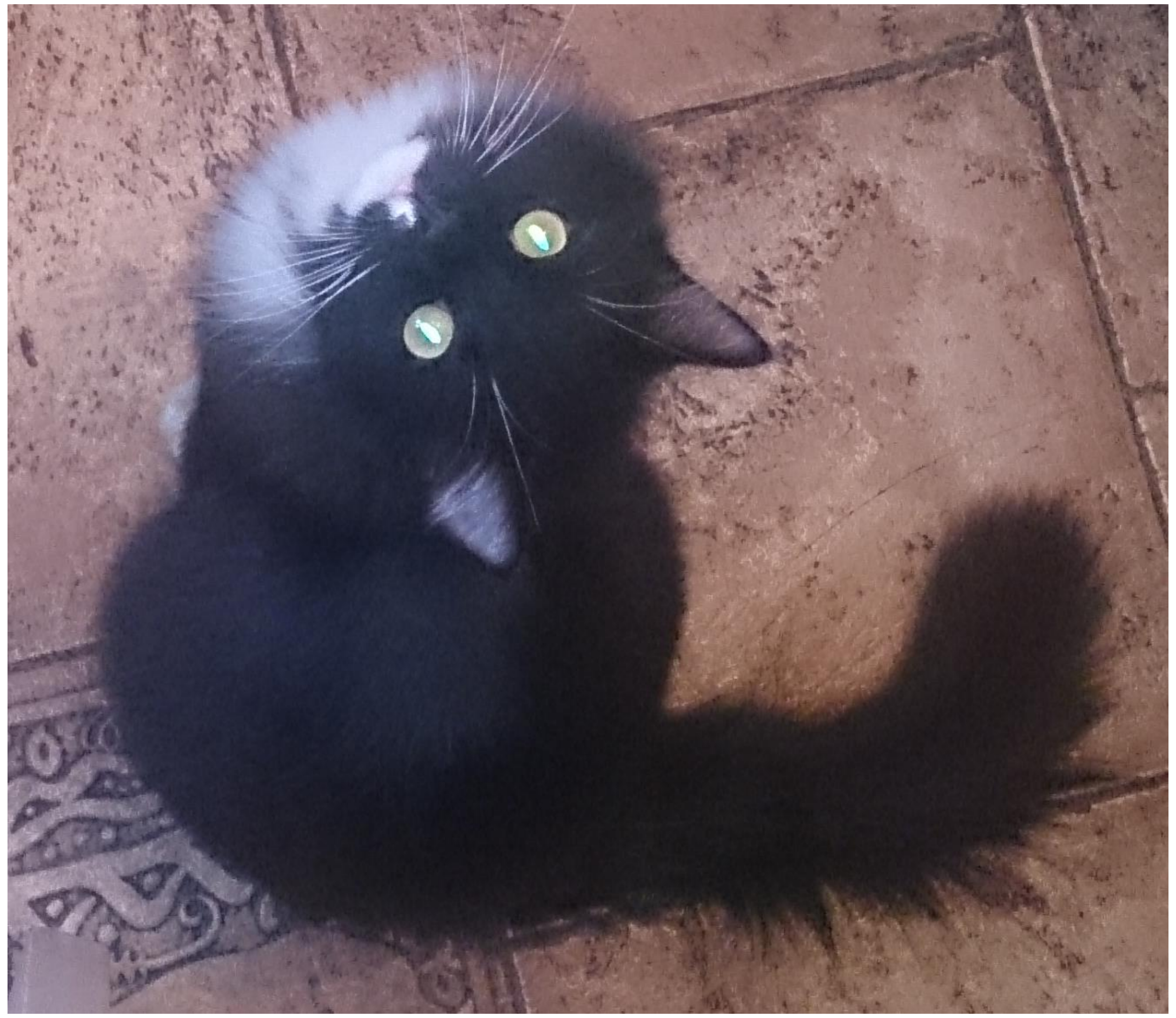
Most pitfalls are here

Ambiguity in instructions: example

Is this cat white?

Yes

No



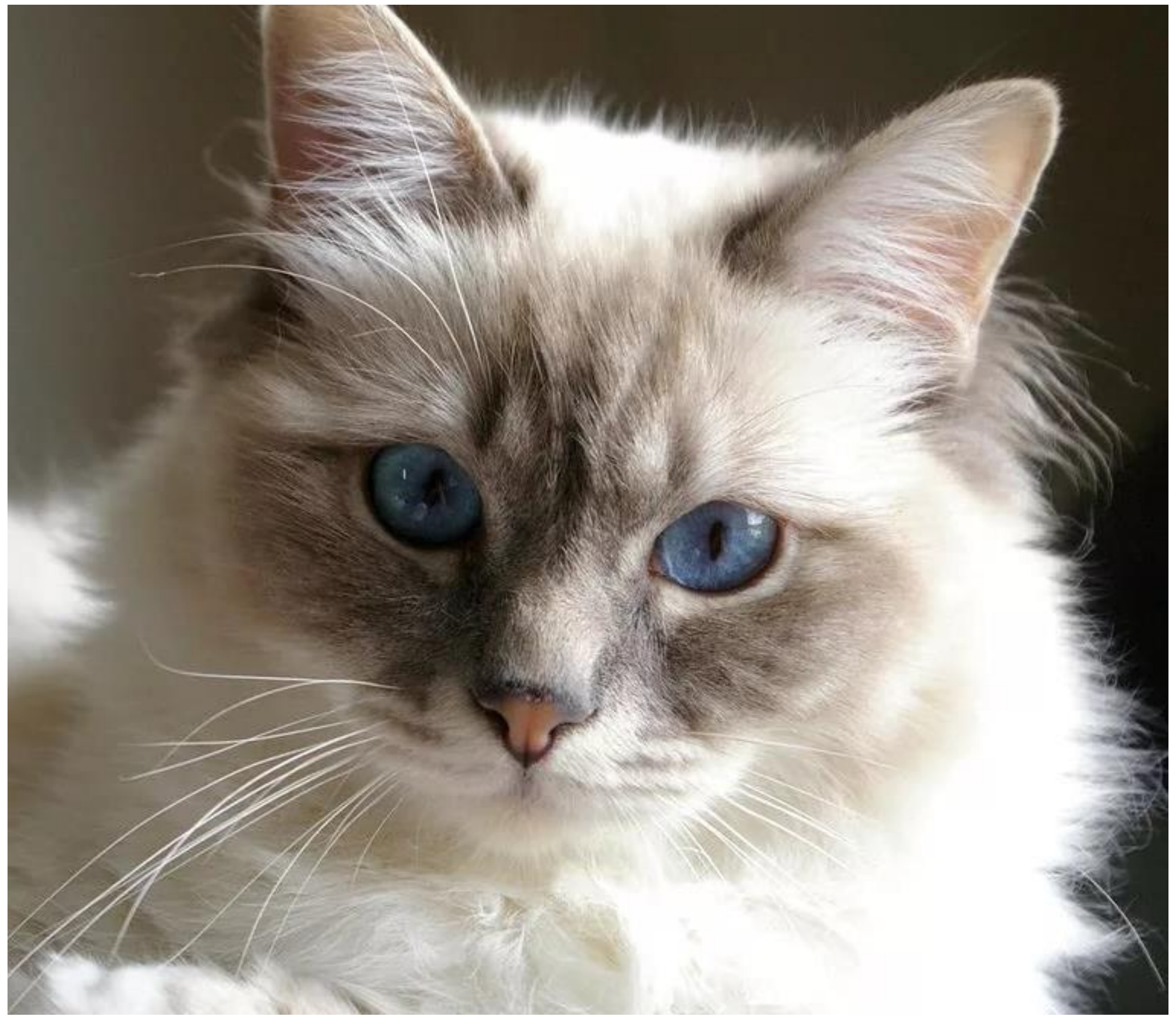
OK: the answer and the task seem clear

Ambiguity in instructions: example

Is this cat white?

Yes

No



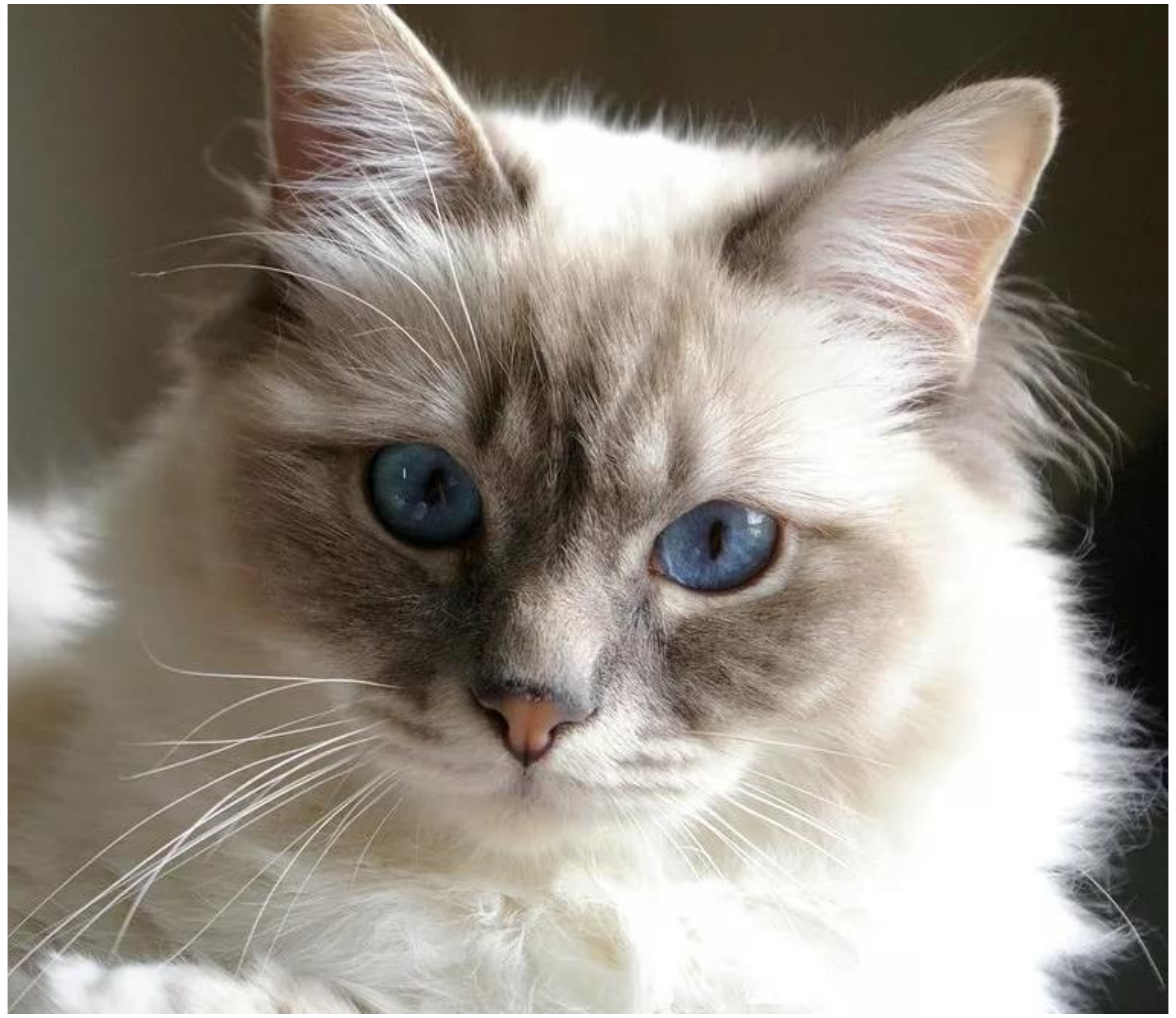
What is the correct answer?

Ambiguity in instructions: example

Is this cat white?

Yes

No



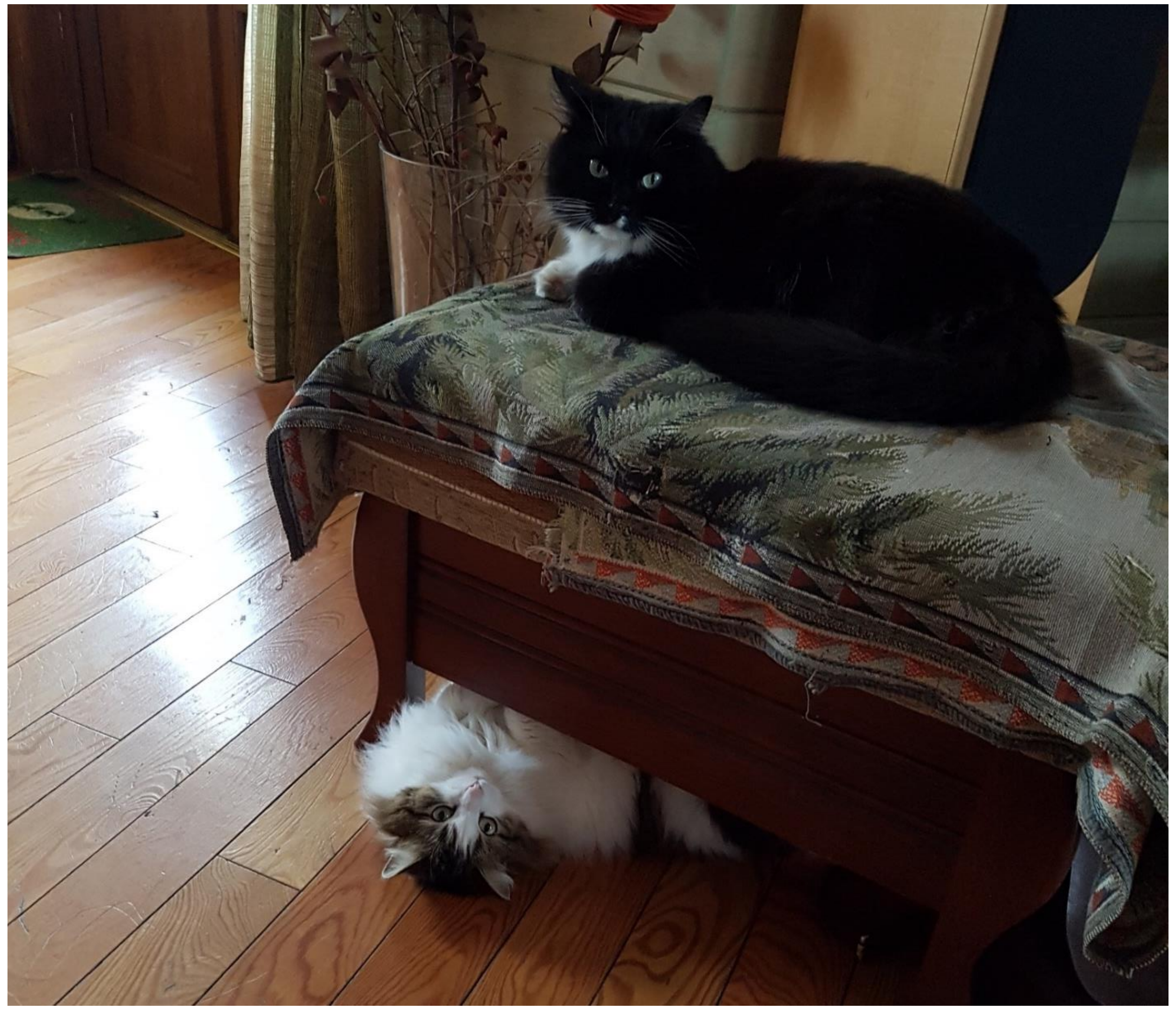
How to fix it In the instructions, clarify what you mean by “a white cat”

Ambiguity in instructions: example

Is this cat white?

Yes

No



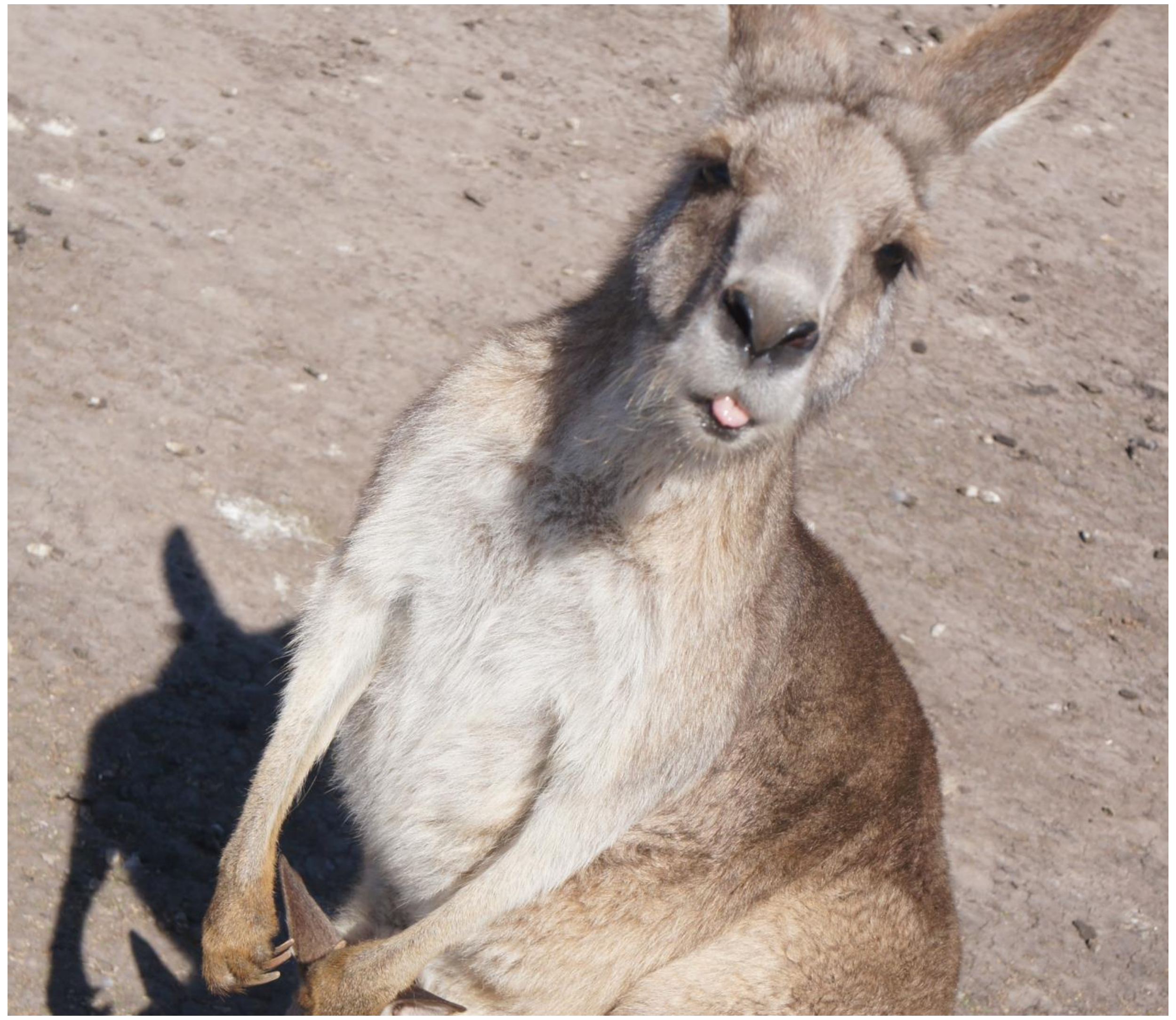
Rare case: multiple cats

Ambiguity in instructions: example

Is this cat white?

Yes

No



Rare case: not a cat

Ambiguity in instructions: example

Is this cat white?

Yes

No

404: Cannot download the image

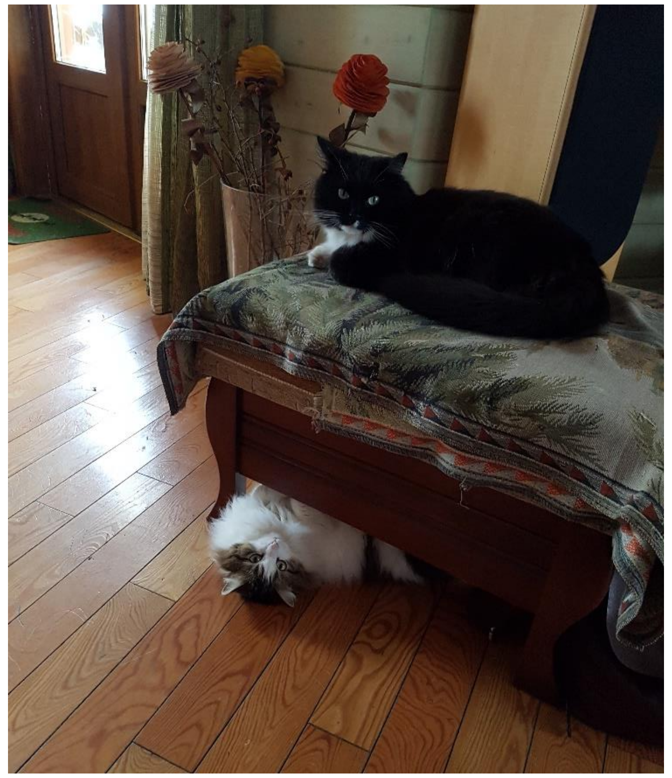
Rare case: image doesn’t load

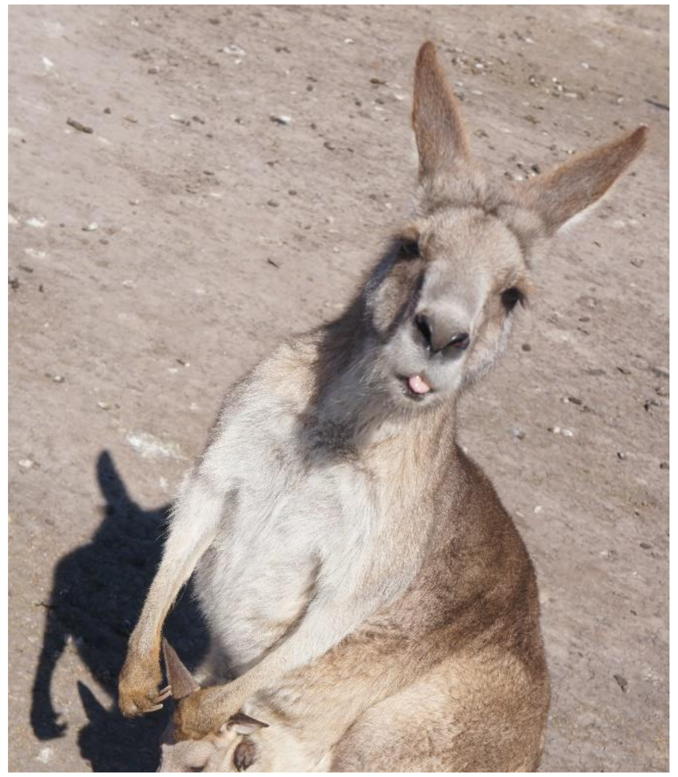
Ambiguity in instructions: example

Is this cat white?

Yes

No





404: Cannot download the image

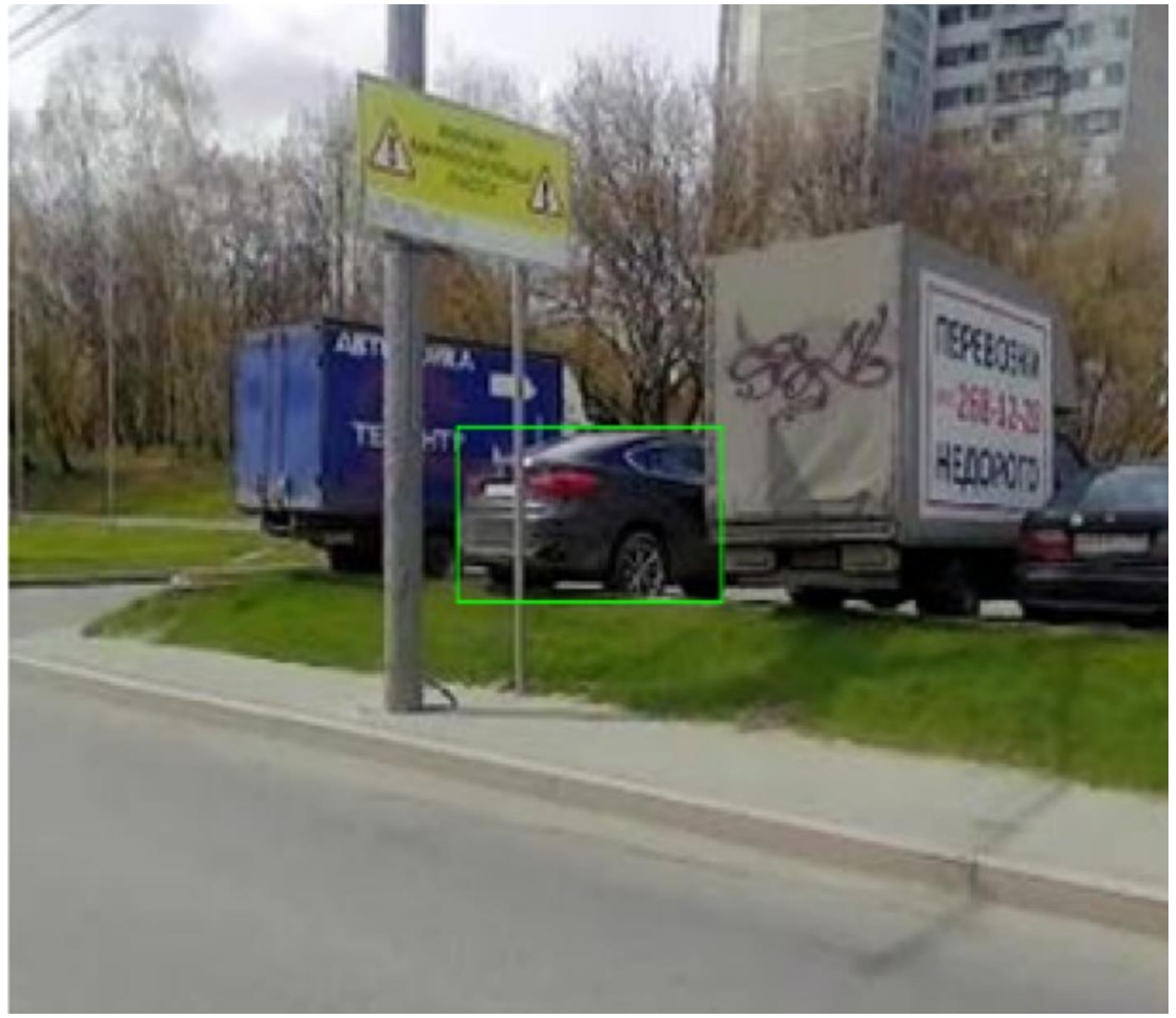
It’s difficult to predict every possible situation, but here’s what you can do:
- In the instructions, explain what to do in a non-standard situation
- In the interface, add a text field for reporting non-standard cases

Ambiguity in instructions: example

Is the outlined object a car?

Yes

No



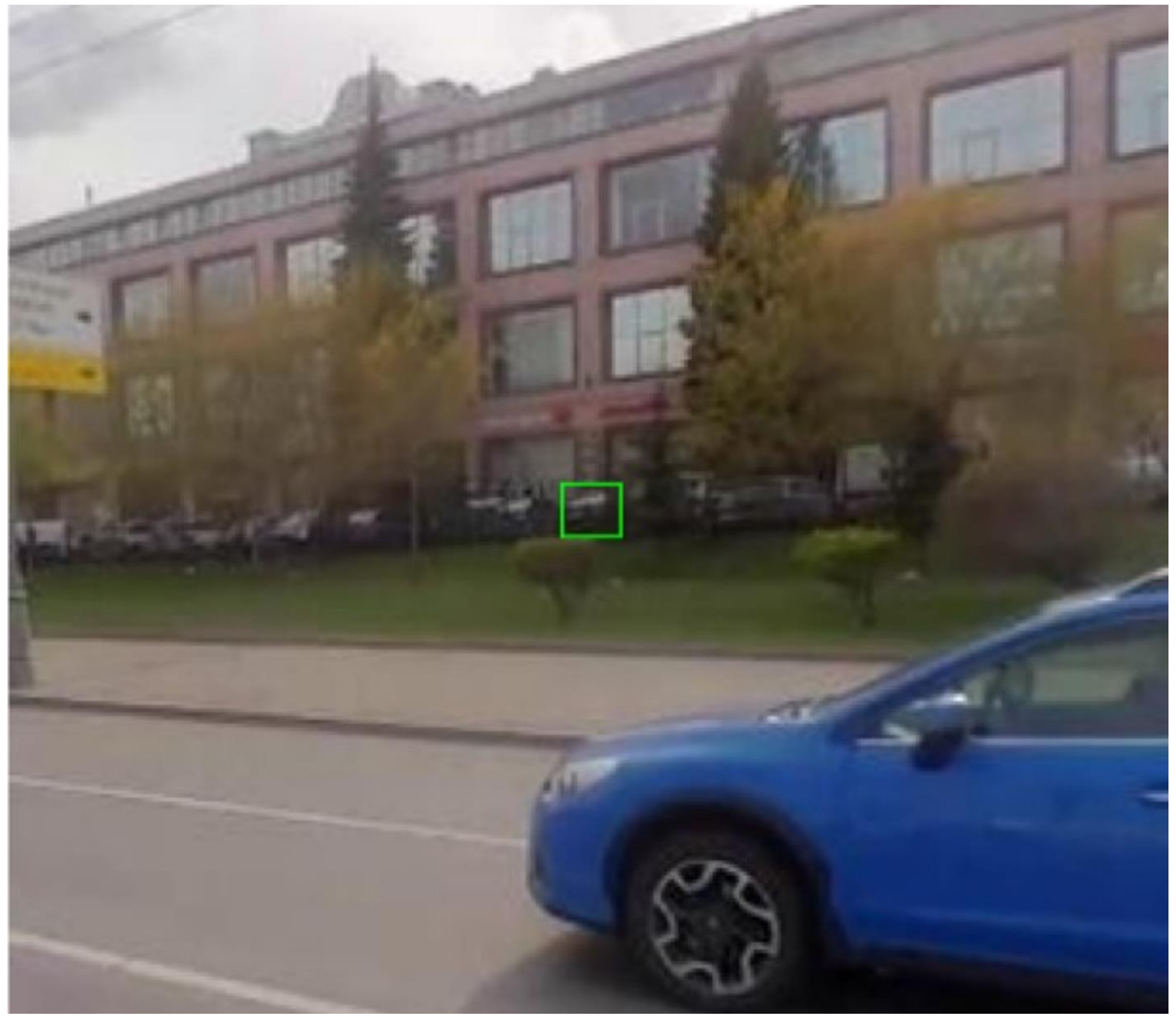
OK: the answer and the task seem clear

Ambiguity in instructions: example

Is the outlined object a car?

Yes

No



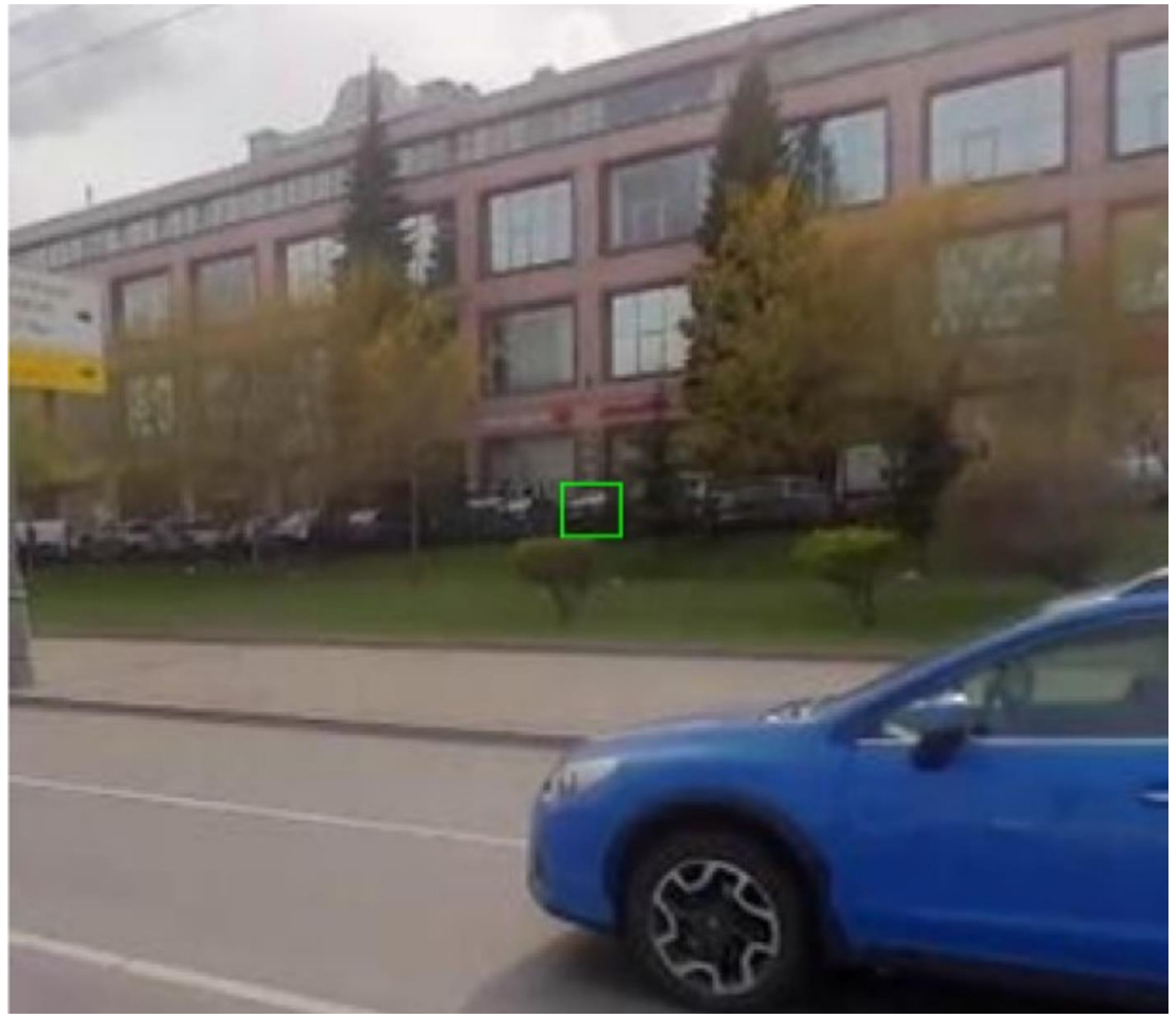
What is the correct answer?

Ambiguity in instructions: example

Is the outlined object a car?

Yes

No



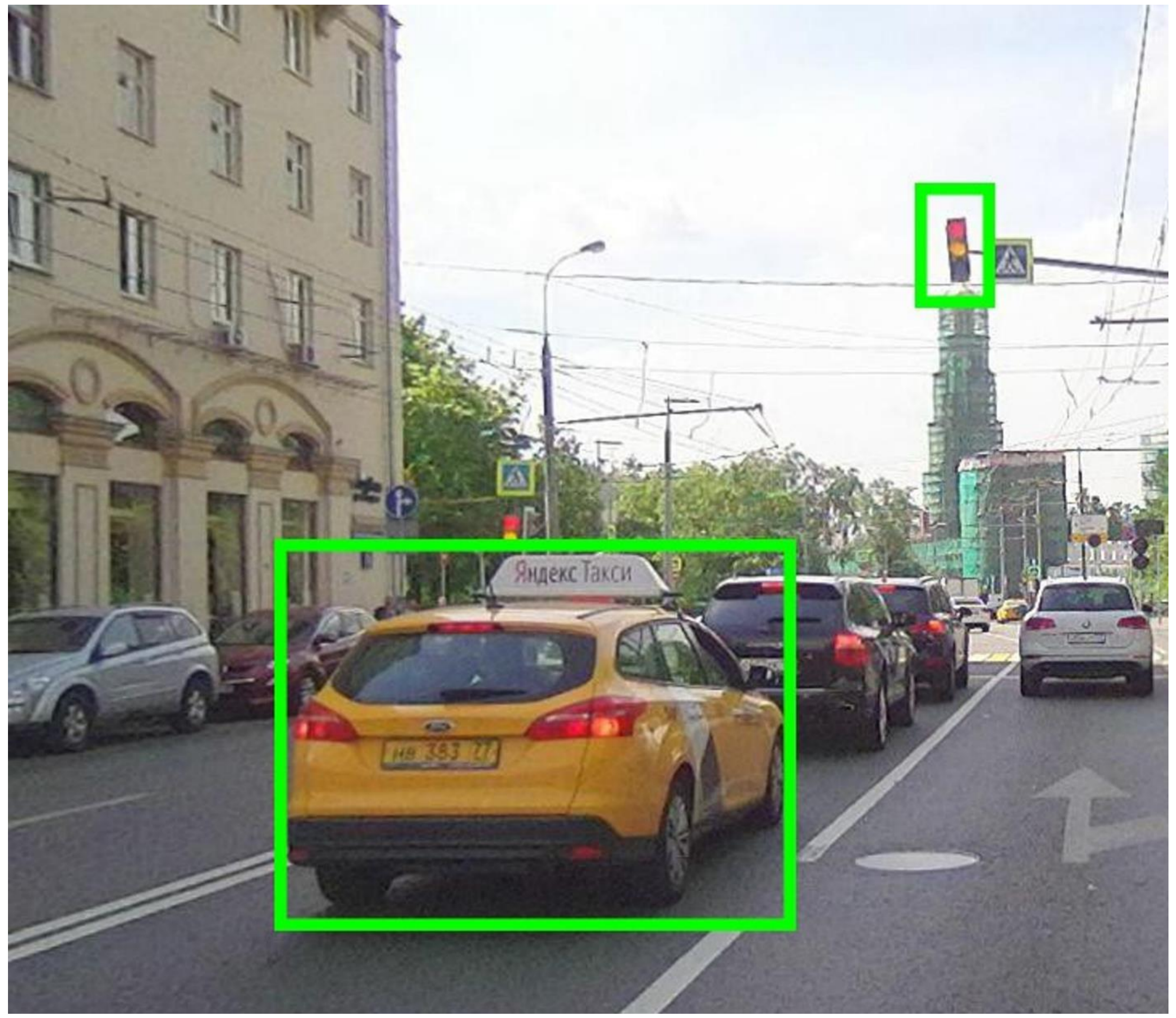
How to fix it In the instructions, clarify what you mean by “a car”

Ambiguity in instructions: example

Is the outlined object a car?

Yes

No



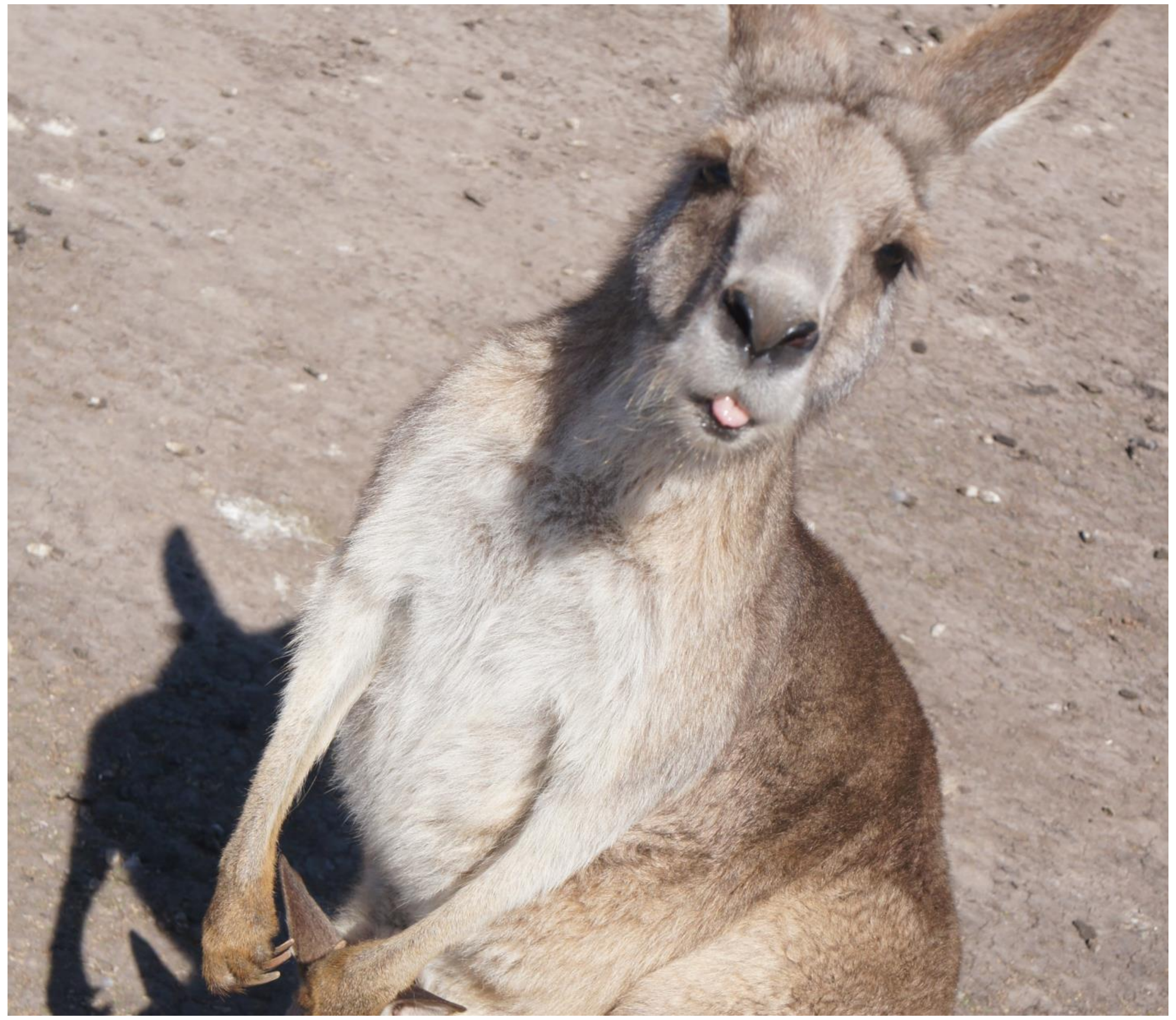
Rare case: multiple objects outlined

Ambiguity in instructions: example

Is the outlined object a car?

Yes

No



Rare case: no objects outlined

Ambiguity in instructions: example

Is the outlined object a car?

Yes

No

404: Cannot download the image

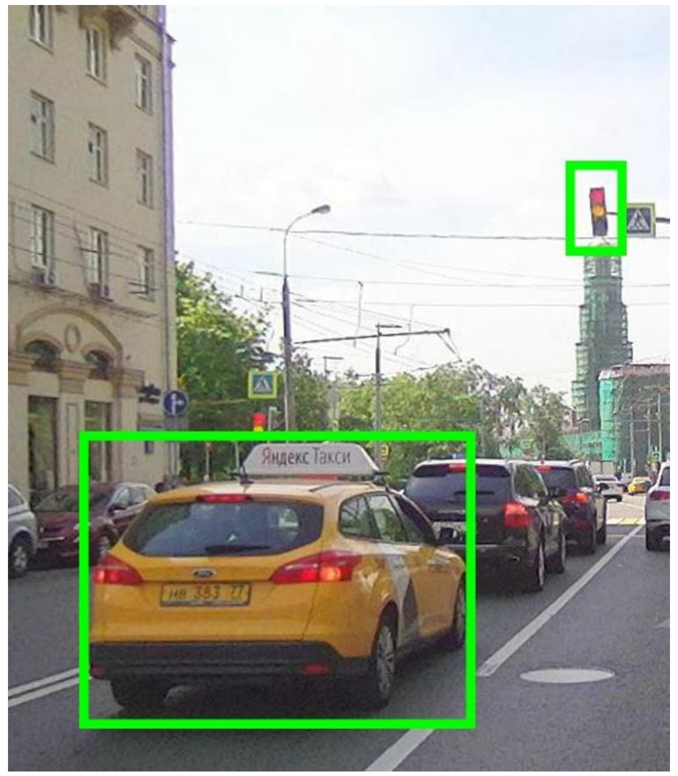
Rare case: image doesn’t load

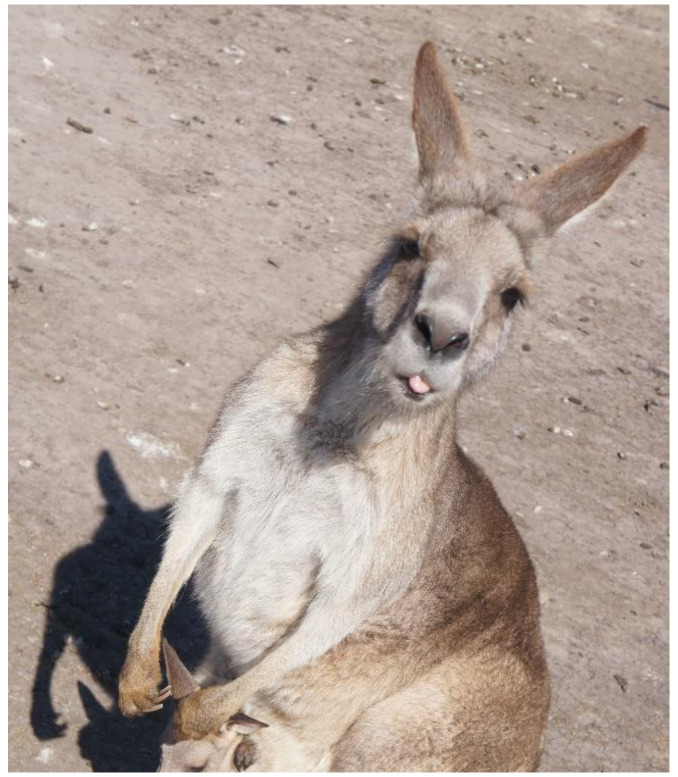
Ambiguity in instructions: example

Is the outlined object a car?

Yes

No





404: Cannot download the image

It's difficult to predict every possible situation, but here's what you can do:

- In the instructions, explain what to do in a non-standard situation
- In the interface, add a text field for reporting non-standard cases

Task interface

Task interface: summary of best practices

- **For faster performance**
  - Assign hot keys for all response options and buttons
  - Reduce navigation to third-party sites
  - Design the interface to be user-friendly
  - Arrange tasks in optimal positions on the page

Task interface: summary of best practices

- **For better quality and fewer errors**
  - Dynamic interface (show/hide input controls depending on user actions)
  - Adaptive interface (good view for any device and screen resolution)
  - Always test your interface (template testing)
  - Dynamic validation of input data (e.g., a text is less than 3 words)

Quality control

Tools for managing the crowd

1. Decomposition of tasks

2. Selecting the right crowd

3. Training and exams

4. Quality control rules

5. Motivation

6. Aggregating results

Quality control

- ► Before starting tasks
  - Crowd selection to match annotators to the task
  - Clear instructions
- ► During tasks
  - Well-designed interface
  - Motivation (quality-based pricing)
  - Control tasks
  - Automated rules to catch bots and cheaters
- ► After tasks are completed
  - Manual review
  - Consensus between annotators and aggregation of results

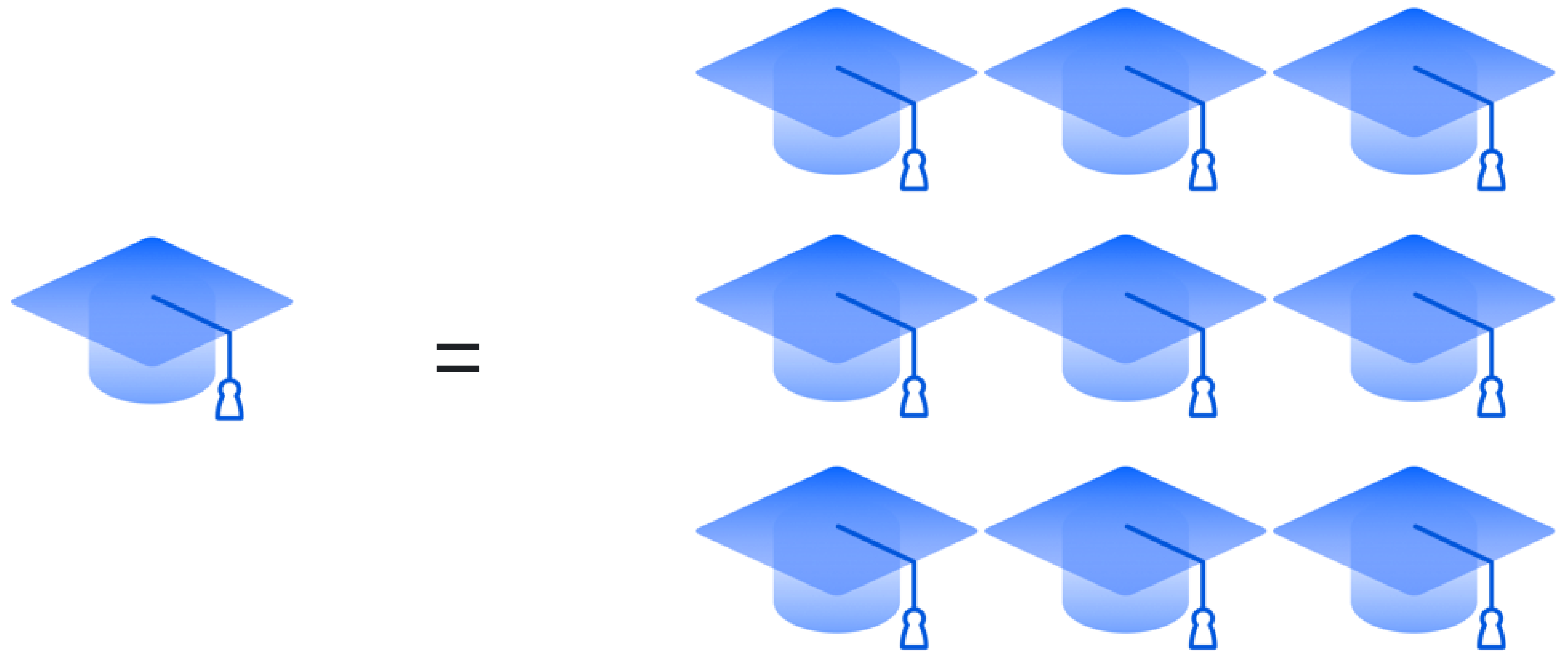
Crowd selection

- Filter by static properties (education, languages, country, etc.)
- Filter by computed properties (browser, region by phone/IP, etc.)

Training and exam

- Train annotators to do your tasks
    - Use training tasks to show how to do tasks correctly
    - Use exams to evaluate skill level after training

Training



Training of one

Training of many

Control tasks (golden sets)

Control annotators performance

- Control tasks are the ones with known correct answers

Training tasks

Train annotators to do your tasks correctly

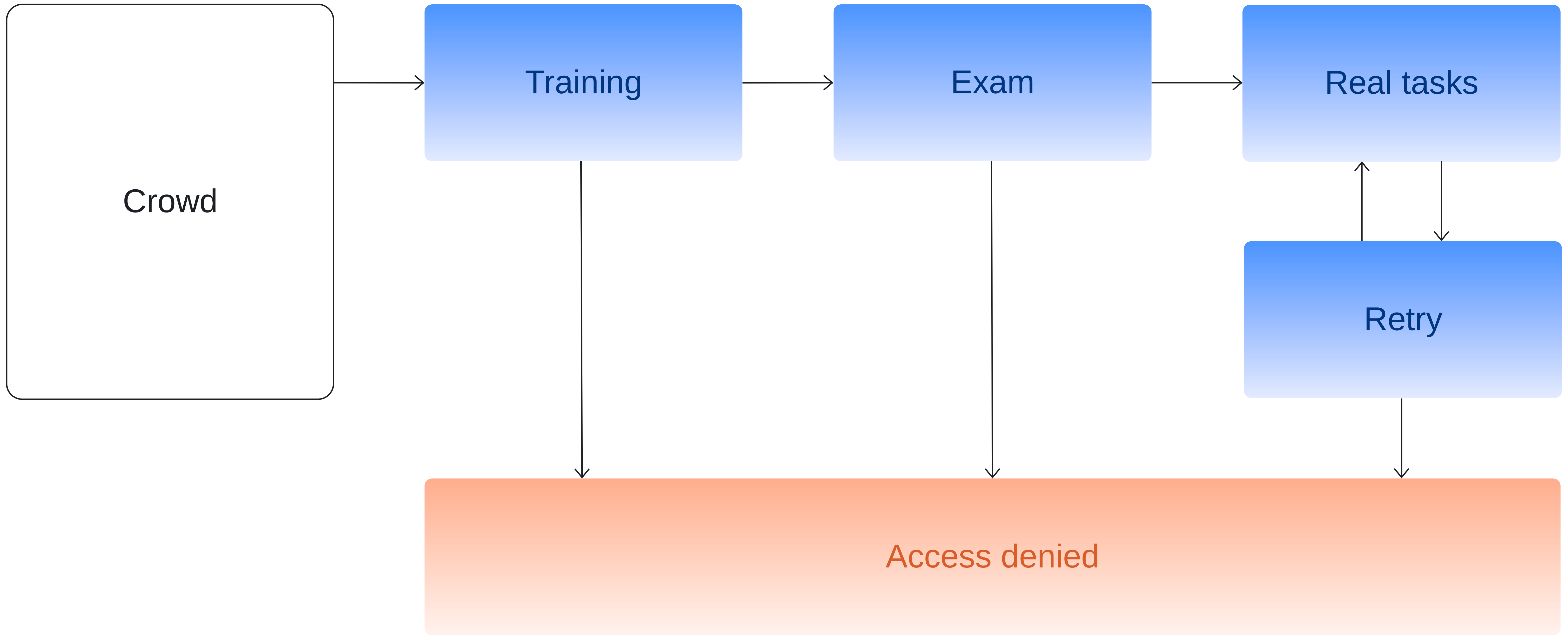
- All tasks are control tasks
- These tasks have additional hints that explain correct answers

Exam tasks

Control the results of training

- ► All tasks are control tasks
- ► No hints or explanations
- ► A good exam should be:
  - Passable
  - Regularly updated
  - Short

Recommended lifecycle of the crowd



Training

Theory + Practice + Exam = Trained crowd

* Footnote text

Quality control: skills

Platform rating

Calculated for each annotator based on overall behavior on all tasks within the platform

Skill is a variable assigned to an annotator

Derived from automatic calculations

- Rate of correct responses (determined by control tasks, consensus, or manual review)
- Behavioral features (submitting answers too fast)
- Binary information about execution of particular projects
- Any combinations of these and other features

Used for automatic decision making

- Control access to certain projects and tasks
- Revoke access to tasks if an annotator’s skill drops too low

Best practices for robust skills

Combine different signals to get a skill robust to gaming

- Combine agreement (consensus) signal with control tasks or manual review
- Add behavioral information: execution time, CAPTCHA, etc.

Use this skill in quality-based pricing

Control tasks (golden tasks)

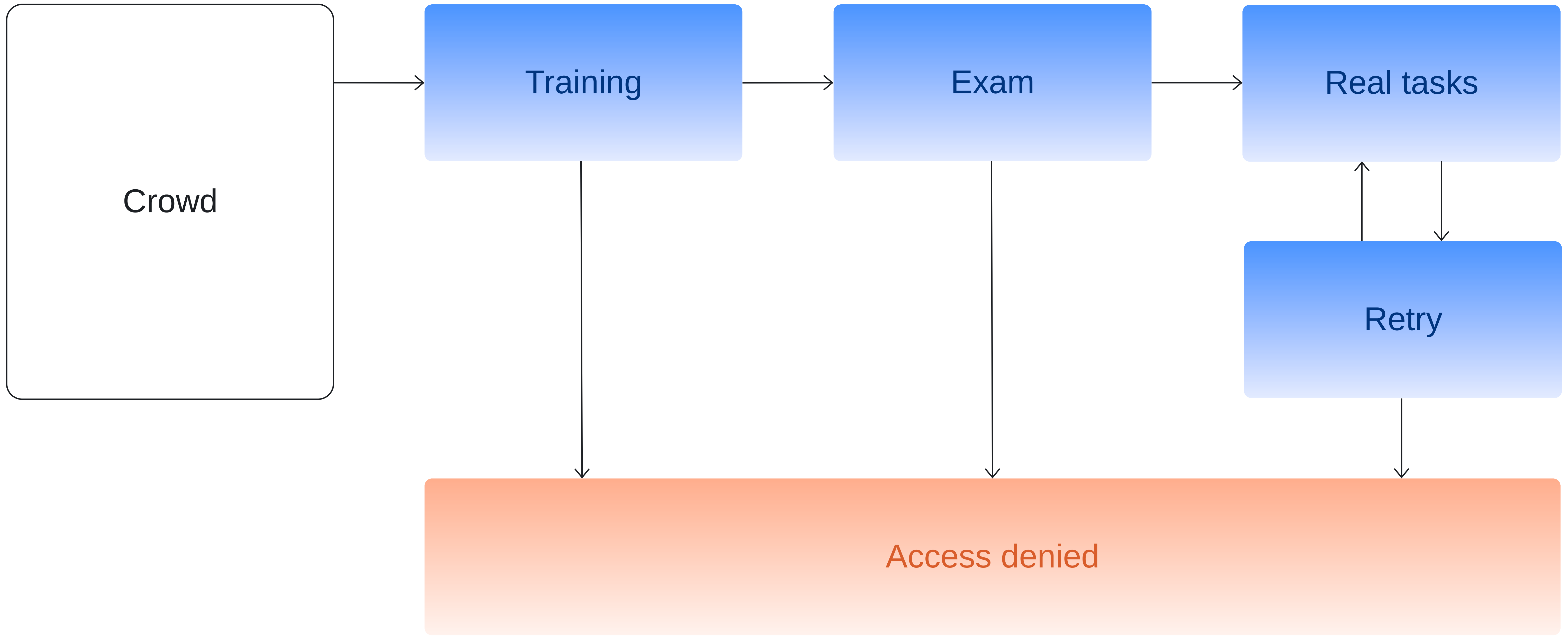
- Tasks with known correct answers assigned to annotators to evaluate their performance

- Distribution of answers in control tasks = distribution in whole set of tasks
- Rare answer variants should have higher frequency
- Refresh your set of control tasks regularly to catch bots and cheaters
- Generate control tasks automatically from submitted answers:
  - Use answers with high confidence level
  - Aggregate answers from a large number of annotators

↑

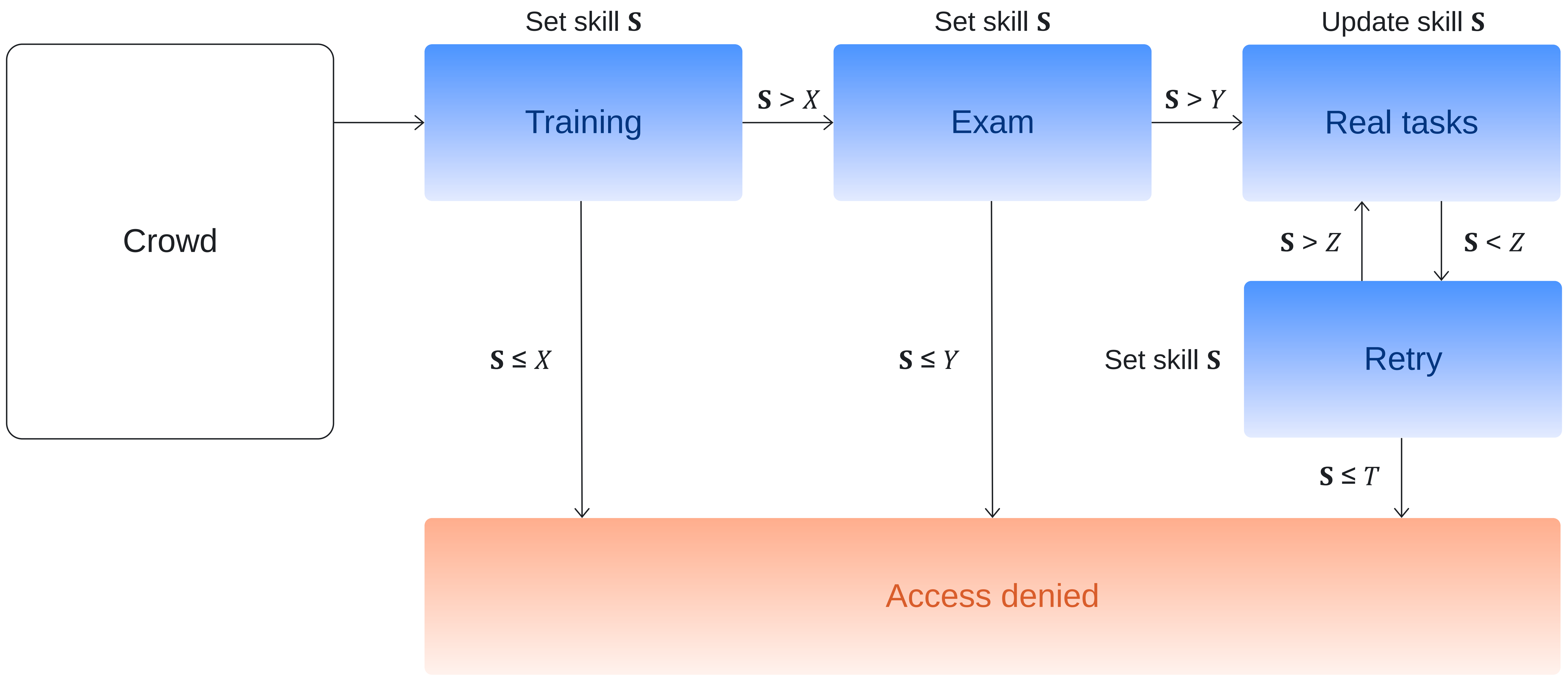
Best practices

Recommended lifecycle of the crowd



Recommended lifecycle of the crowd

Let quality be controlled by means of a skill $\boldsymbol{S}$



Automated ways to catch bots and cheaters

- Control fast responses
- Detect whether links were visited
- Detect whether videos were played
- Use captchas

Motivation

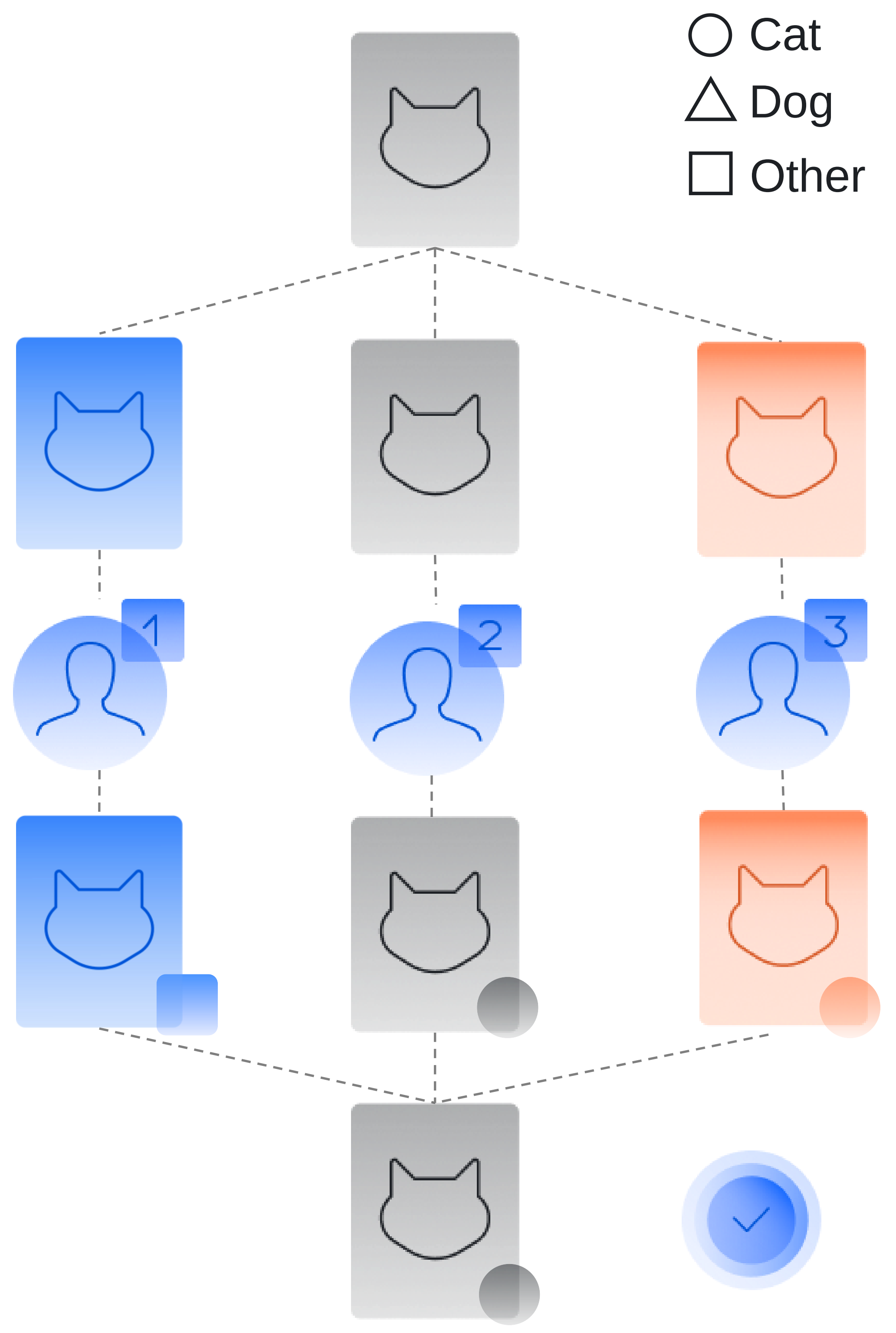
- Bonuses for stable quality over time
- Gamification (level up skills, leader boards, etc.)
- Dynamic pricing (depends on individual quality rating)

Manual review of submitted tasks

- ► Only pay for correct (accepted) responses
  - Useful for sophisticated tasks when control tasks and consensus models don't work
- ► For small volumes, review submitted tasks yourself
- ► For large volumes, ask the crowd to do it
  - Create a separate verification project to check submitted tasks
  - Design a hierarchy of data labeling projects (apply decomposition)

Aggregation

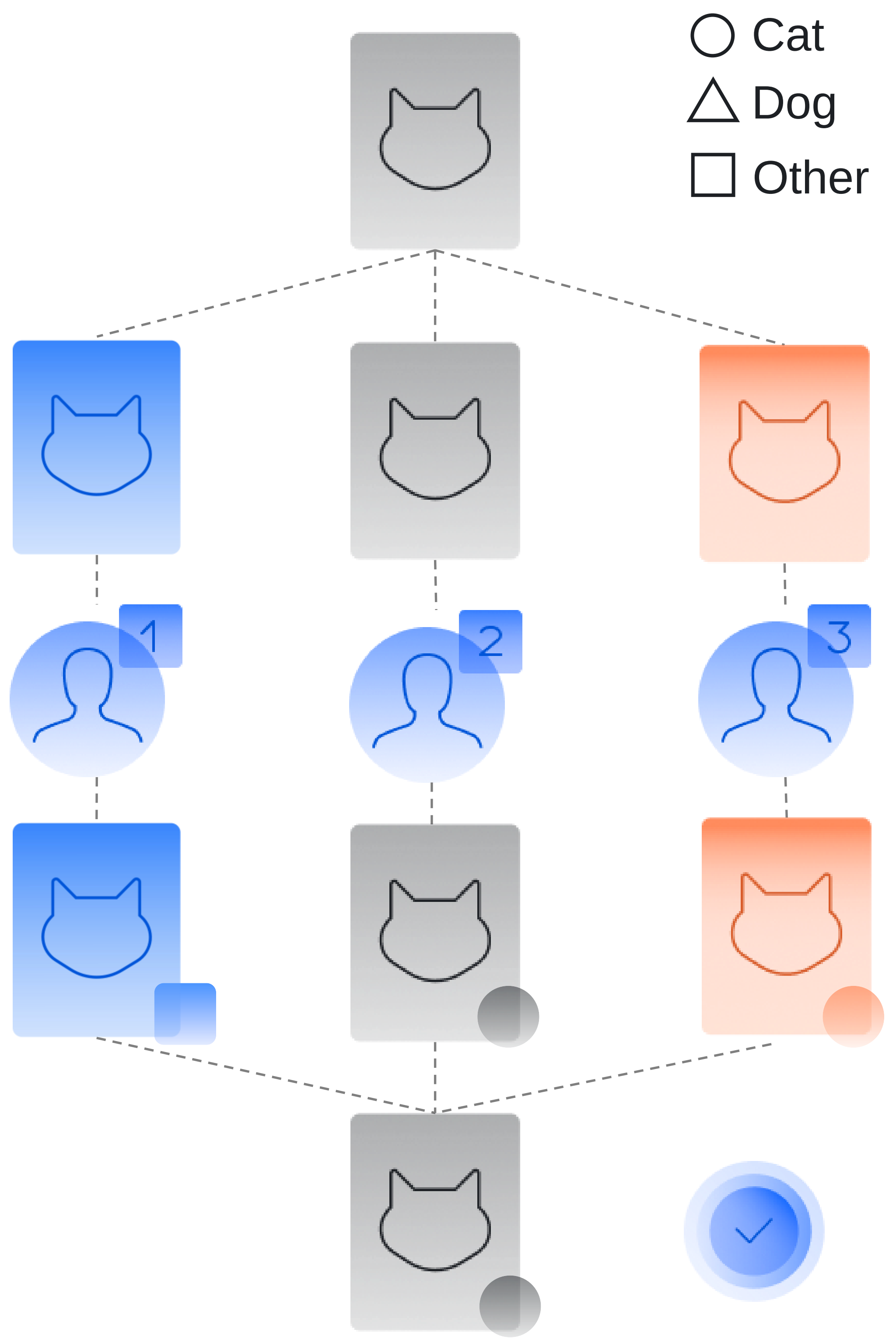
Aggregation



Crowd annotators assign noisy labels to objects

Aggregate multiple labels into a more reliable one

Aggregation



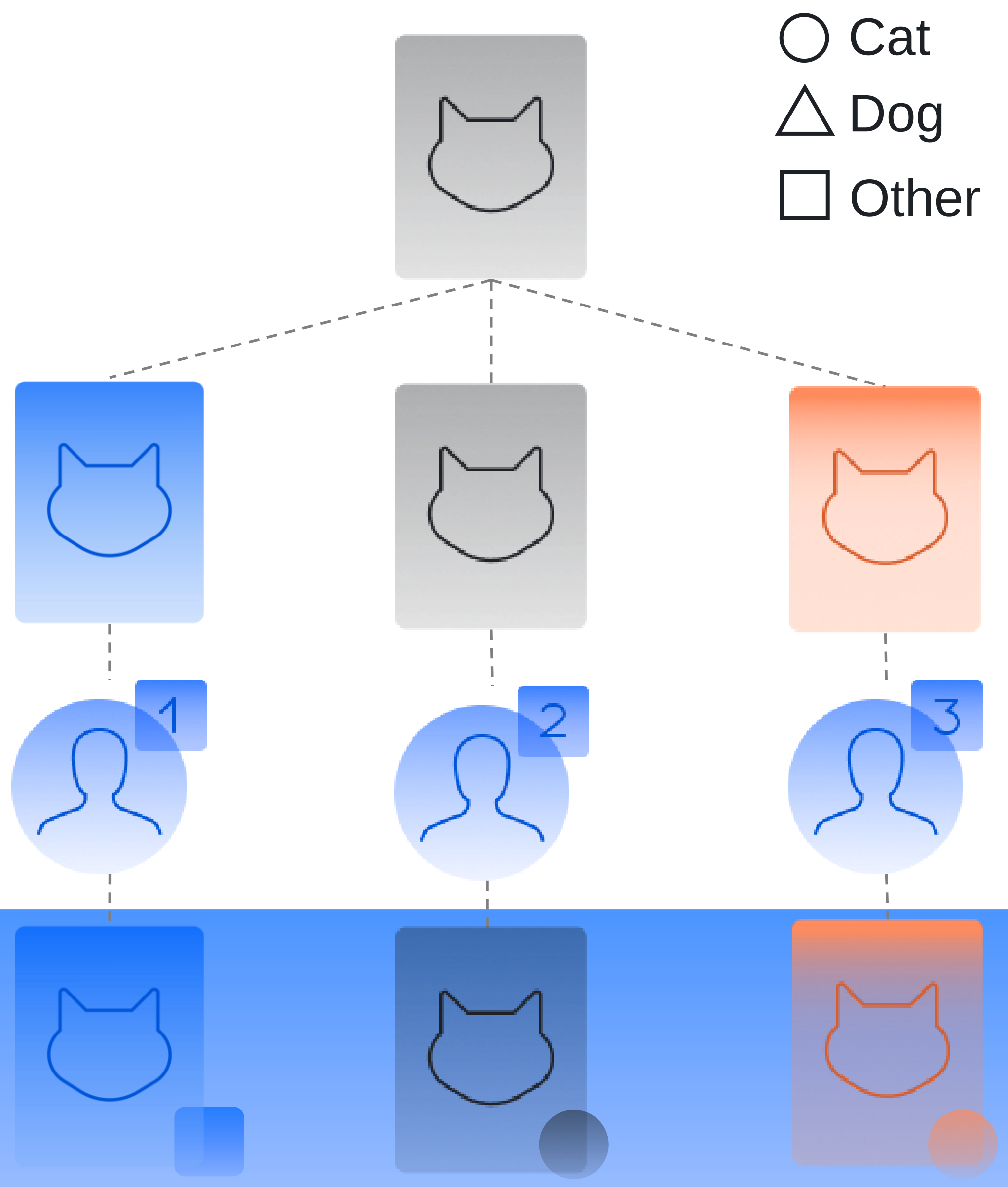
Crowd annotators assign noisy labels to objects

Aggregate multiple labels into a more reliable one

The simplest way: Majority Vote

There are more sophisticated methods which can increase quality by as much as 15%

Consensus between annotators



Upload multiple copies of each object to label

Annotators assign noisy labels to objects

Derive quality from consensus

Works well only if most annotators have good accuracy

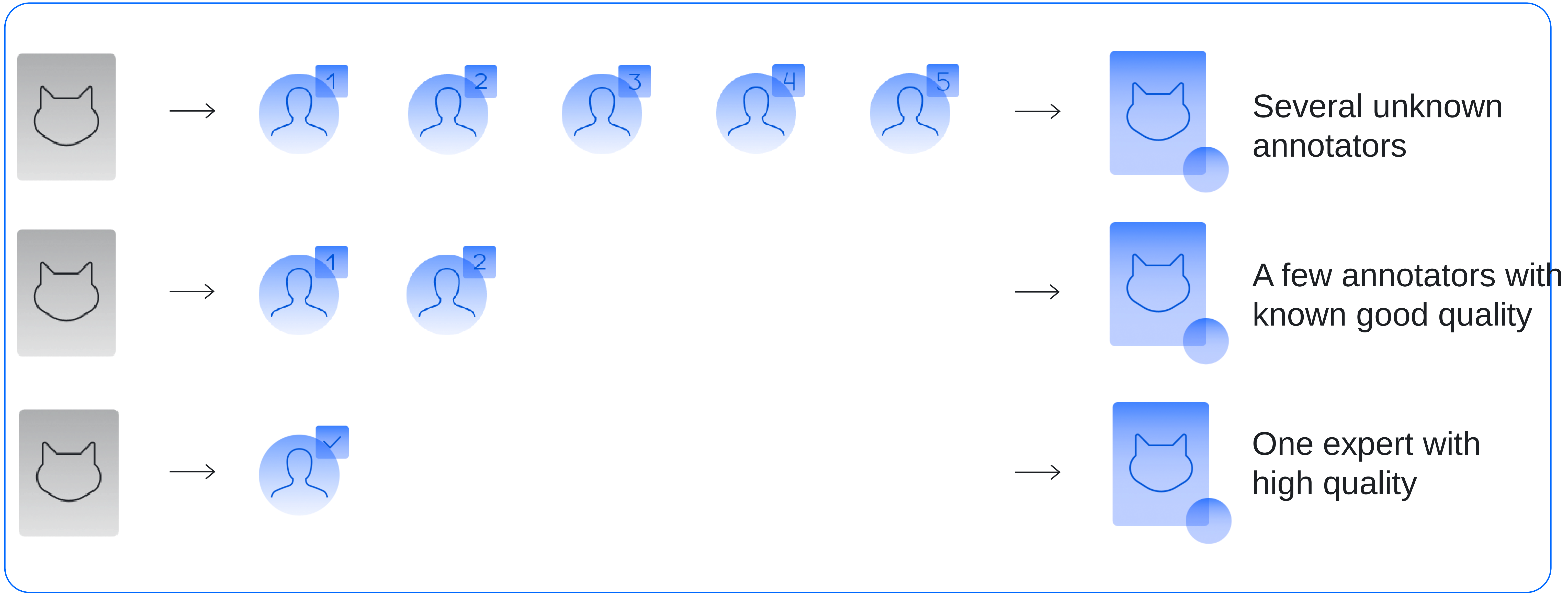
Dynamic pricing & dynamic overlap

Pricing depends on

- Task design
  - Payment is made per page of microtasks (a task suite)
  - Time required to do a task: control hourly wage
- Market economy aspects
  - The fewer annotators available (i.e., specialized skills), the higher the price
  - How soon do you need the tasks done (latency)?
- Result quality
  - Incentivize better performance with a quality-dependent price

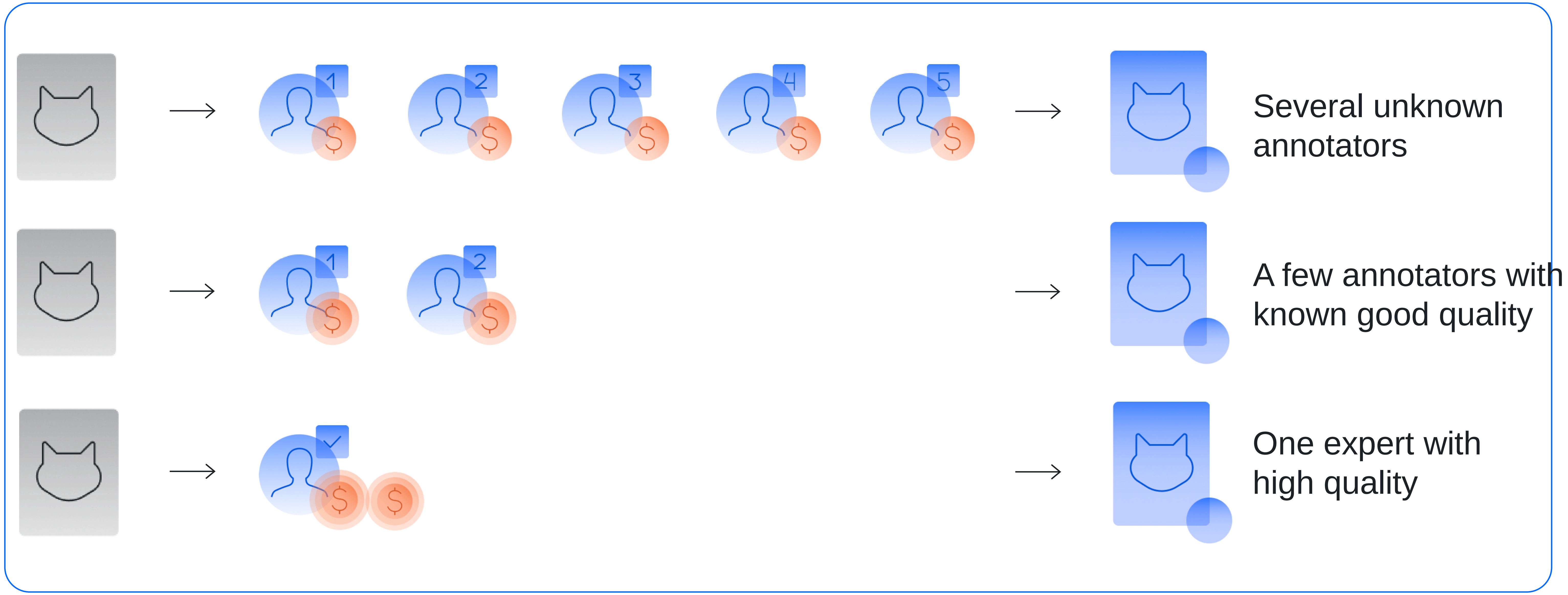
Dynamic overlap

Obtain aggregated labels of a desired quality level using a smaller number of noisy labels


1
2
3
4
5
Several unknown annotators
1
2
A few annotators with known good quality
One expert with high quality

Dynamic overlap

Obtain aggregated labels of a desired quality level using a smaller number of noisy labels


1
2
3
4
5
Several unknown annotators
1
2
A few annotators with known good quality
One expert with high quality

By adjusting overlap based on annotator quality, you can:

→ Incentivize effort (better pay for better quality)

→ Stay within a fixed budget

IF

Simple instructions

Easy-to-use
task interface

Good
decomposition

THEN

Annotators do a better job

Easy to control quality

Standard aggregation
models work well

Easy to control
and optimize pricing

Toloka