



Toloka

Web Engineering with Human-in-the-Loop

Dmitry Ustalov, Nikita Pavlichenko, Boris Tseytlin,
Daria Baidakova and Alexey Drutsa



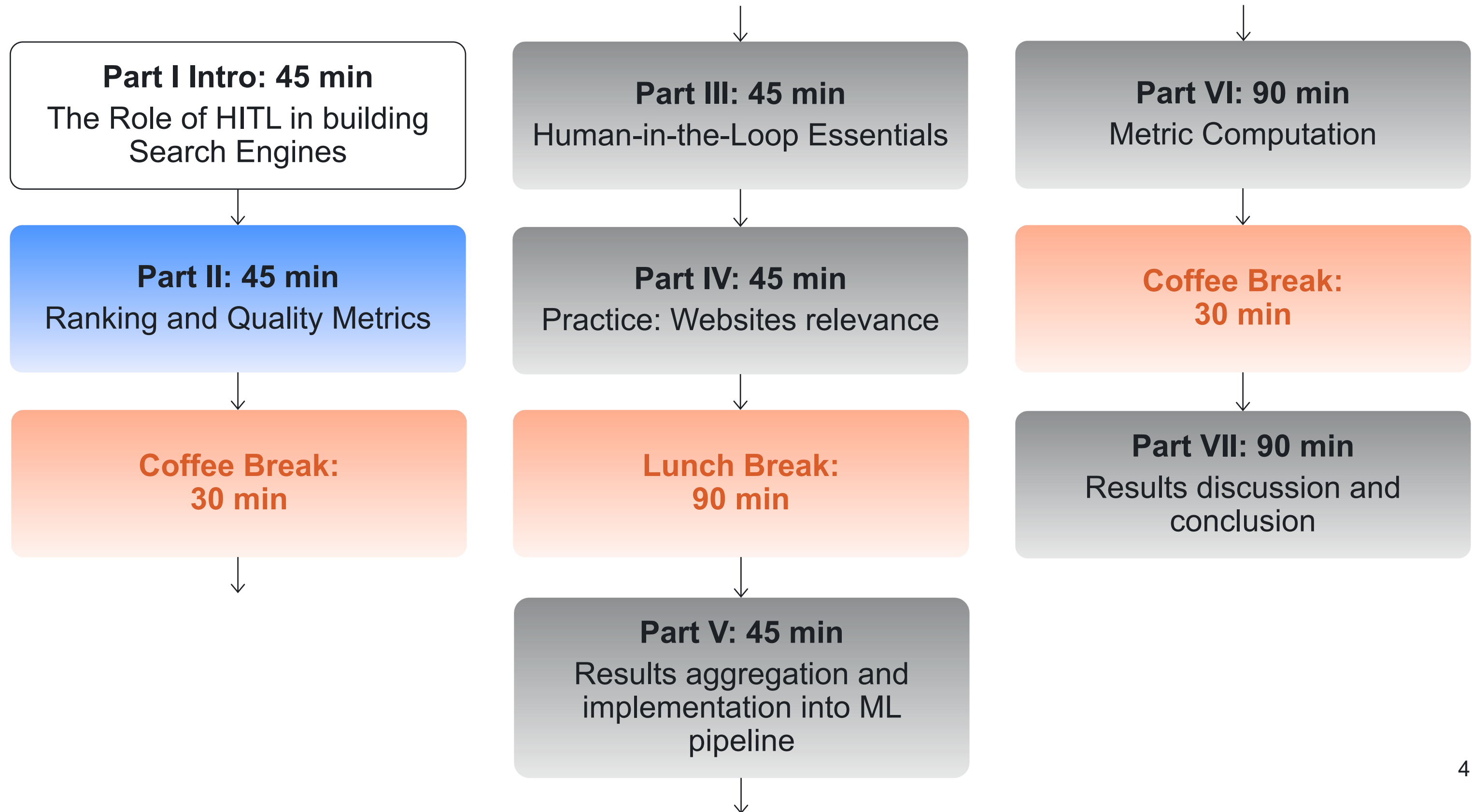
Part III

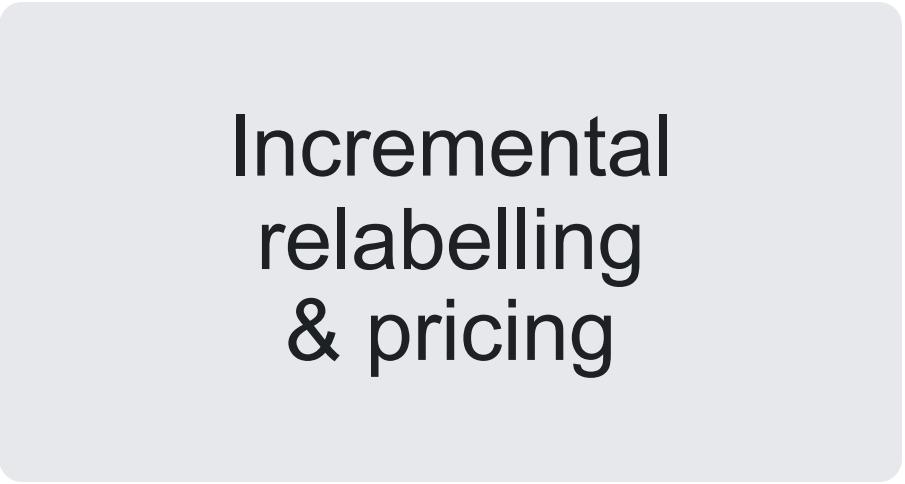
Human-in-the-Loop

Essentials

Nikita Pavlichenko,
Researcher

Tutorial Schedule

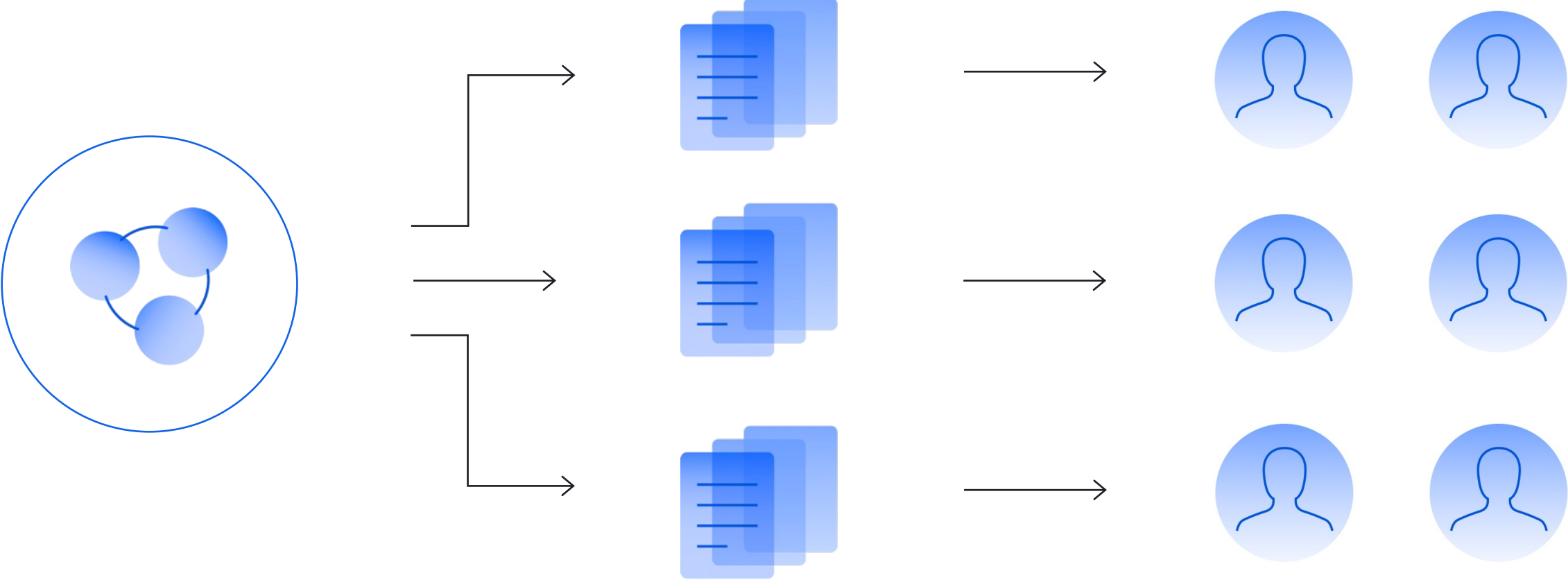




Decomposition

The background features a series of overlapping, curved, blue shapes that create a sense of depth and movement, resembling a stylized sunburst or a series of concentric, curved bands. The colors range from a deep navy blue to a bright, vibrant blue.

Decomposition



A big task

Projects with microtasks
of different type

Cloud
of performers

Decomposition: why?

- ▶ Performers are usually non-specialists in your specific task
- ▶ The simpler a single task is:
 - The more people can perform your task
 - The easier its instruction
 - The better quality of performance
- ▶ A way to:
 - Distinguish tasks of different difficulty levels
 - Control and optimize pricing
 - Control quality by post verification

Decomposition: when?

- ▶ If
 - Your task requires an answer selected among more than 3–5 options
 - Your task has long instructions that are hard to read
- ▶ Then your task requires decomposition

Case of decomposition: a lot of questions



Bad practice: All questions in one task

What animal is on 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 situated?

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

Case of decomposition: a lot of questions



Good practice: Each question in a separate task

What animal is on 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 situated?

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

Case of decomposition: need to verify answers



The task: Highlight all koalas on the photo

Problem: highlighting can be done in different ways

Hence, it is difficult to:

- Compare with control answers
- Aggregate answers from different performers

A good solution

A task for another performer:

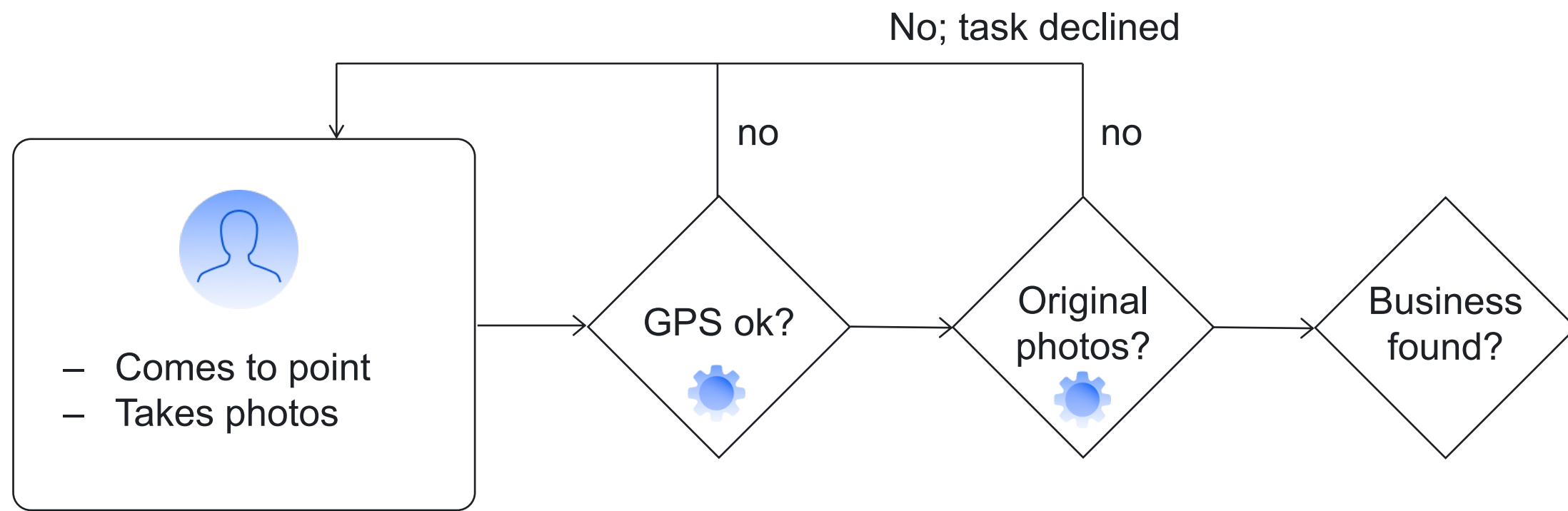
Have the koalas been highlighted correctly?

Real example: decomposition
for an offline data collection task

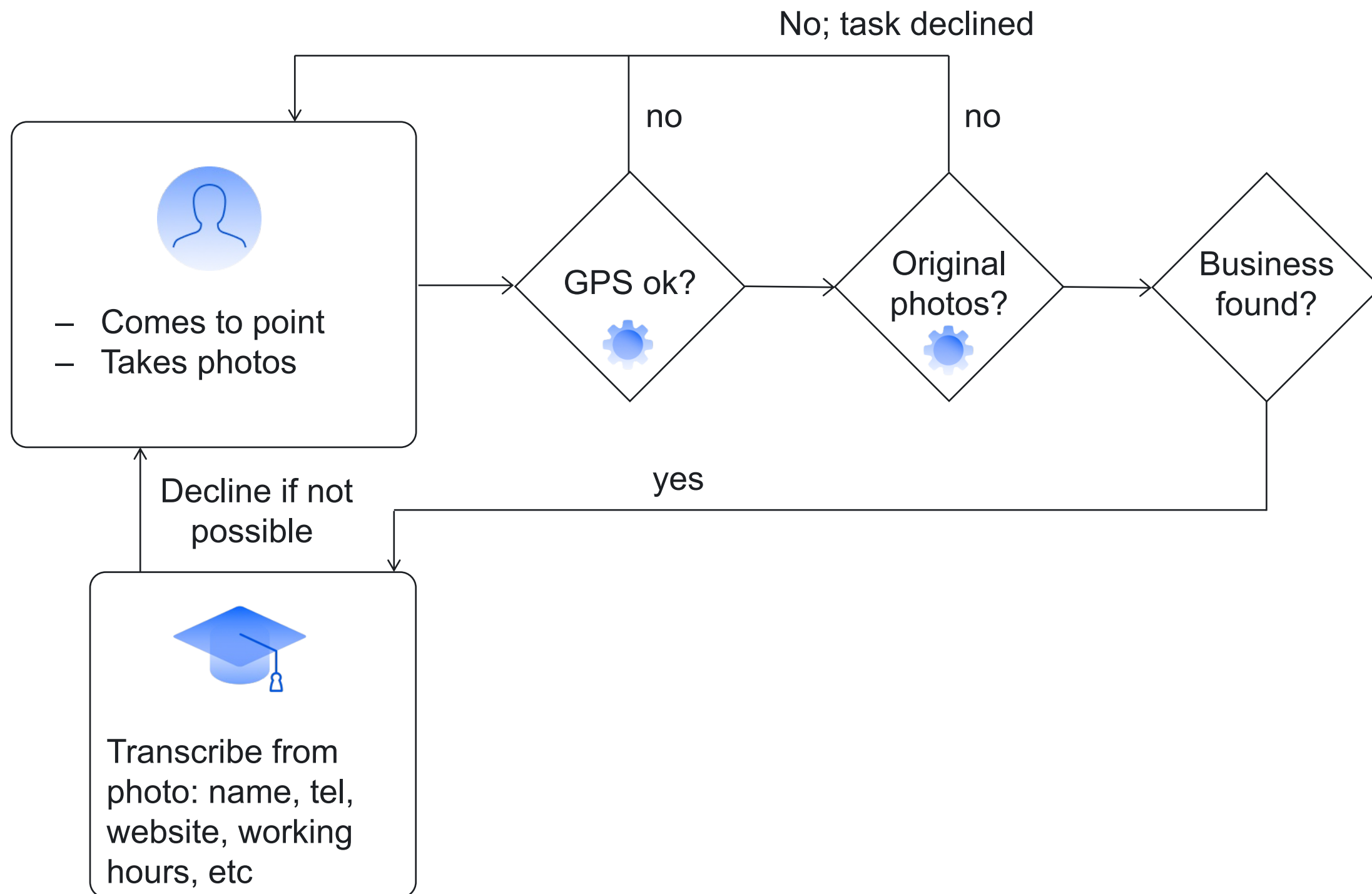


- Comes to point
- Takes photos

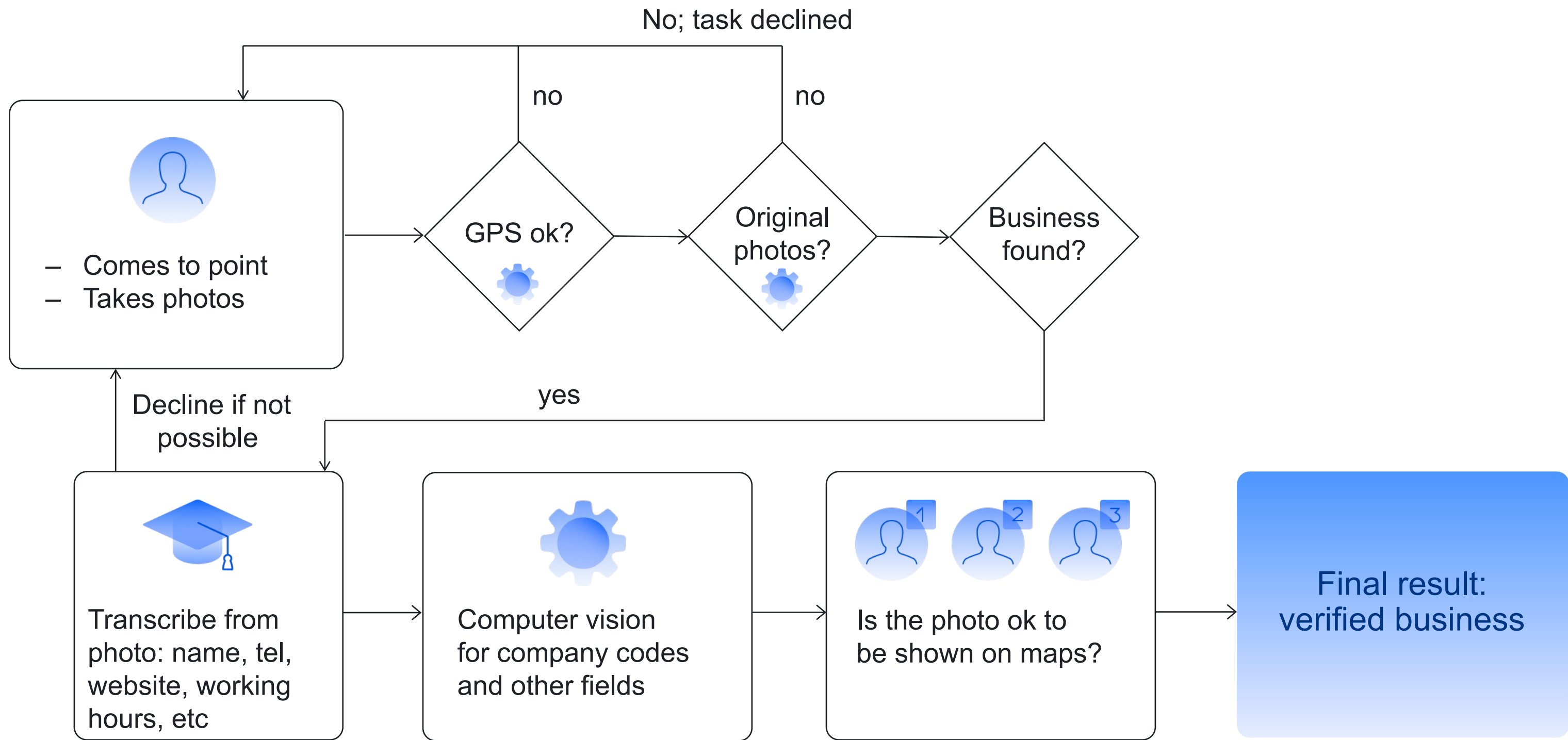
Final result:
verified business

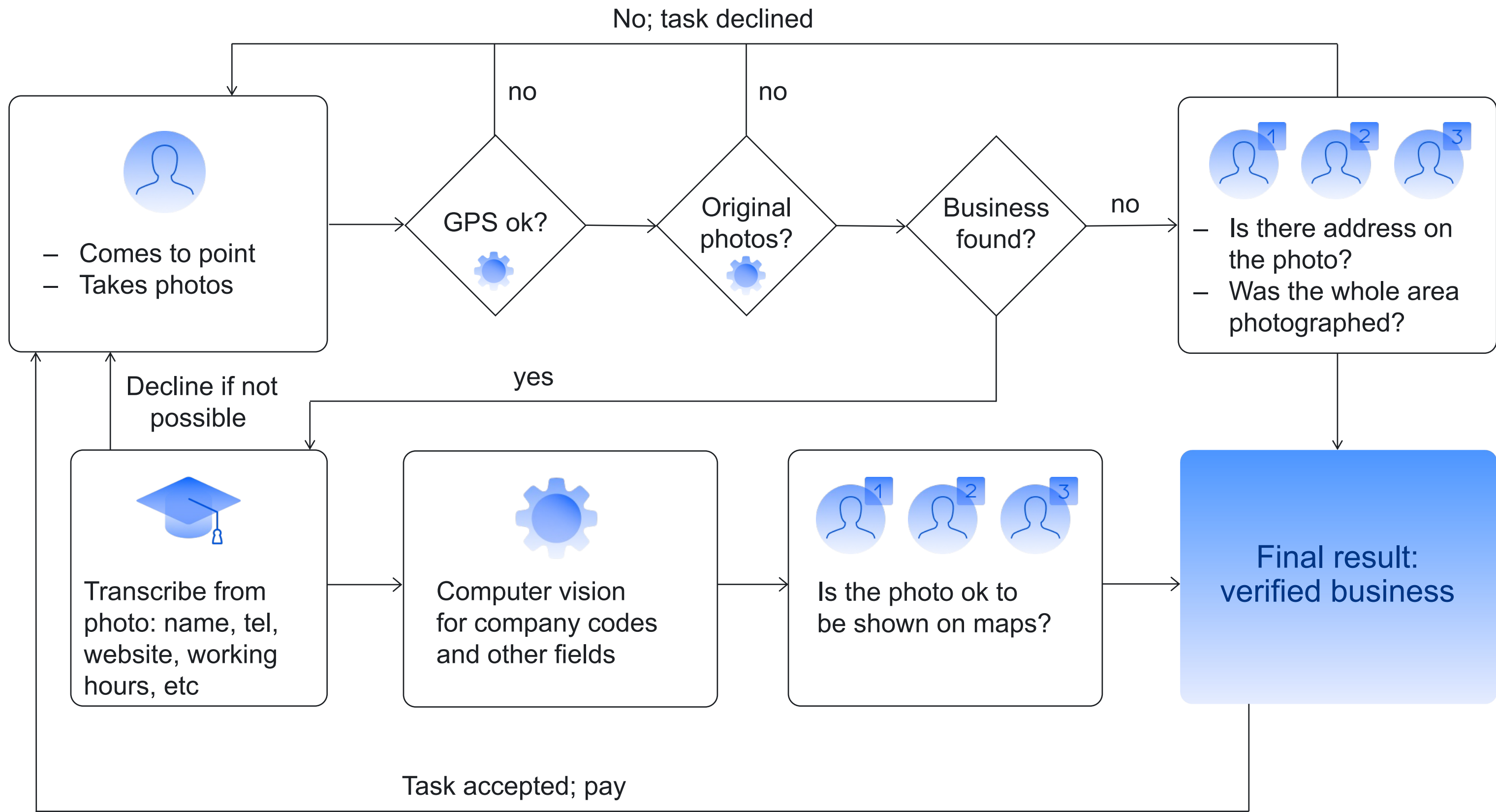


Final result:
verified business



Final result:
verified business





The background features a series of overlapping, curved, blue shapes that create a sense of depth and movement. The colors range from a deep navy blue to a bright, vibrant blue. The shapes are layered, with some appearing to recede into the background while others are more prominent in the foreground.

Instruction

Instruction: a typical structure

- ▶ Goal of the task to be done
- ▶ Interface description
- ▶ Algorithm of required actions
- ▶ Examples of good and bad answers
- ▶ Algorithm and examples for rare cases
- ▶ Reference materials

Most pitfalls are here



Instruction ambiguity for a rare case: example

Is this cat white?

Yes

No



OK: the answer and the task seem clear

Instruction ambiguity for a rare case: example

Is this cat white?

Yes

No



What is the correct answer?

Instruction ambiguity for a rare case: example

Is this cat white?

Yes

No



How to fix

- In the instruction: clarify what you mean under «a white cat»

Instruction ambiguity for a rare case: example

Is this cat white?

Yes

No



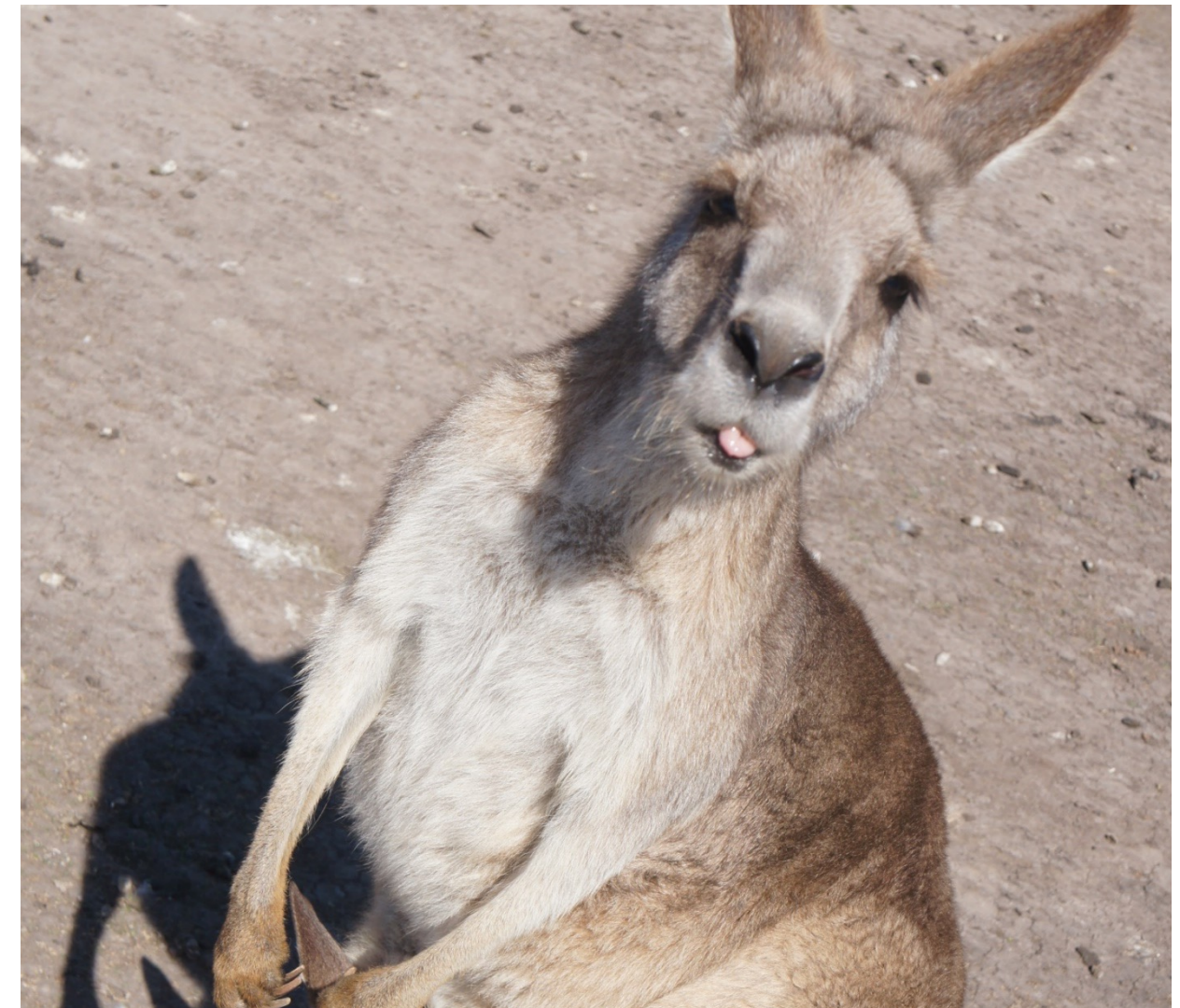
Rare case: many cats

Instruction ambiguity for a rare case: example

Is this cat white?

Yes

No



Rare case: not a cat

Instruction ambiguity for a rare case: example

Is this cat white?

Yes

No

404: Cannot download the image

Rare case: image has not been shown

Instruction ambiguity for a rare case: example

Is this cat white?

Yes

No



404: Cannot download the image

- It is difficult to predict situations of any kind, but you can:
- In the instruction: clarify what should be done in a non-standard situation
 - In the interface: add a text field to allow a performer to report the case

Task interface

Put yourself in the performer's shoes

- ▶ Many tasks at a time
- ▶ Earnings depend on the amount of tasks done
- ▶ Monotonous tasks
- ▶ Concentration required

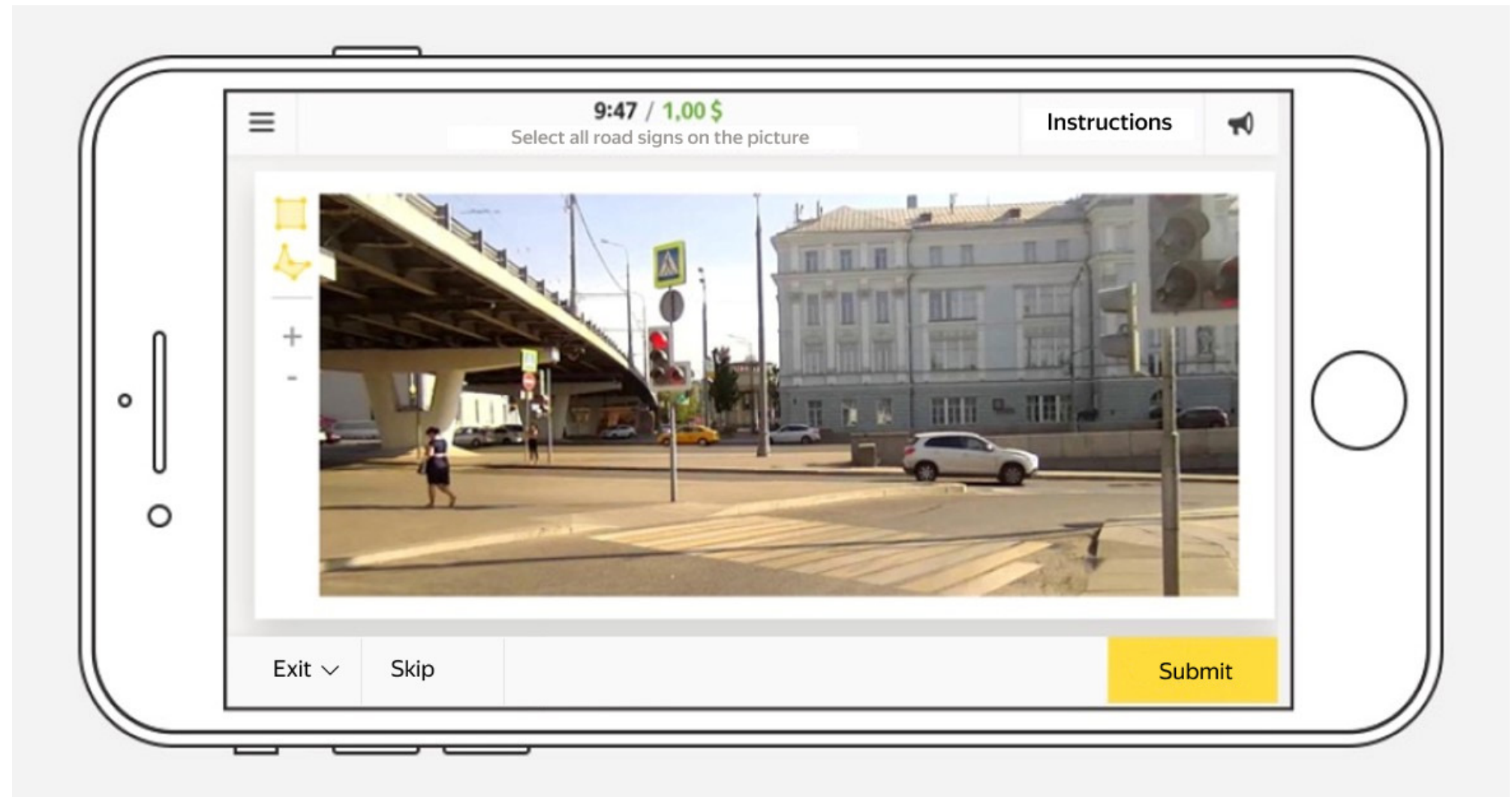


Industry Tech Travel

Additionally, we have developed market-leading on-demand transportation services, navigation products, and other mobile applications for millions of consumers across the globe. Yandex, which has 30 offices worldwide, has been



Some tasks are not suitable for cross-platform use

- ▶ Task is to find and encircle all road signs on the picture



Checking required actions

Game «Let's play»

Go to play   Please, visit the page

Works fine ¹ A problem occurred ²

Select a key that didn't work

Space ^Q Enter ^W Shift ^E

Minimum use of external resources

- ▶ Add screenshots
- ▶ Save data in your storage
- ▶ Check clicks on necessary links

Minimalistic design

Too many nested blocks

Same font everywhere

Phrase job occupation in liverpool
Query liverpool totaljobs

Additional
Ad title Job in Liverpool
Ad text Be the first to find out about new jobs on totaljobs.com

Does the phrase match the query?
[Yes](#) [No](#)

Verdicts look like links

Color contrast too bright

Reasonable space usage

Game «Let's play»

Works fine ¹ A problem occurred ² Does not open ³

Select a key that didn't work

Space ^Q Enter ^W Shift ^E

Only necessary elements

- ▶ Task is to evaluate which translation is better

Phrase **where to cross the street**

Translation 1 **wo sollte man die Straße überqueren**

Translation 2 **wo man die Straße überquert**

Check with translation services

Yandex ¹ Google ² Bing ³ Lingvo ⁴ PROMT ⁵

Translation 1 is better ^Q Translation 2 is better ^W

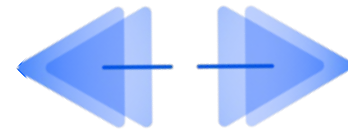


- Multiple tasks on one page help to save time on switching between pages.
- Put as many as can be completed in 5 minutes.

Task suite



Same width
of task blocks



Avoid empty
spaces between
blocks



2–3 task
in a row

— Test your task before launching!

Quality control

Quality control

- ▶ “Before” task performance
 - Selection of performers
 - Well-designed instruction
- ▶ “Within” task performance
 - Golden set (aka honey pots)
 - Well-designed interface
 - Motivation (e.g. performance-based pricing)
 - Tricks to remove bots and cheaters (e.g. quick answers)
- ▶ “After” task performance
 - Post verification (acceptance)
 - Consensus between performers and result aggregation

Selection of performers

- ▶ Filter by static properties (e.g. education, languages, citizenship, etc.)
- ▶ Filter by computed properties (e.g. browser, region by phone/IP, etc.)
- ▶ Filter by skills
 - To select proper specialization
 - To control quality level on your tasks
 - To get performers with best quality on past projects
- ▶ Educate to perform your tasks
 - Use training tasks to show how to perform tasks
 - Use exam tasks to evaluate education level

Golden set (aka honey pots)

- ▶ Tasks with known correct answer shown to performers to evaluate their quality

- Distribution of answers in golden set = distribution in whole set of tasks
- But should contain rare answer variants with higher frequency
- Refresh your set of honey pots regularly to avoid bots and cheating
- Automatic golden set generation via performers:
 - Tasks with answers of high confidence
 - (e.g. aggregation of answers from a large number of performers)

↑
Best practices

Motivation

- ▶ Bonuses for a good quality within a period
- ▶ Gamification (e.g. achievements, leader boards, etc)
- ▶ Price depends on quality

↑
Will be discussed in Part V

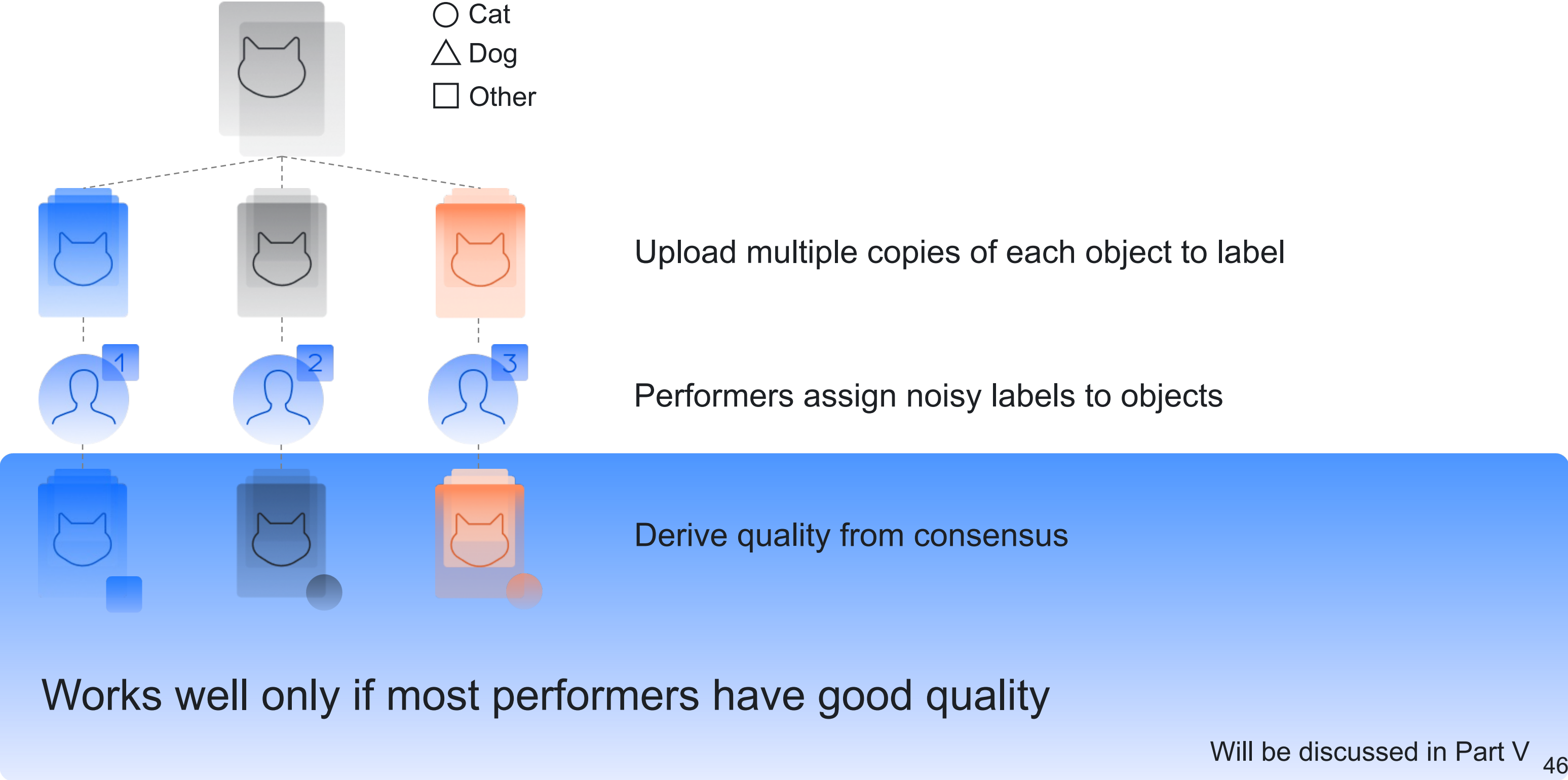
Tricks to remove bots and cheaters

- ▶ Control fast responses
- ▶ Check whether a link has been visited
- ▶ Check whether a video has been played
- ▶ etc

Post verification (acceptance)

- ▶ A performer gets paid only if his answer is accepted
 - Is used when a task is sophisticated (neither golden set nor consensus models work)
 - Can be performed on your own, but
- ▶ You can use other crowd performers via a task of different type
 - Thus, you deal with hierarchy of projects (you apply decomposition)

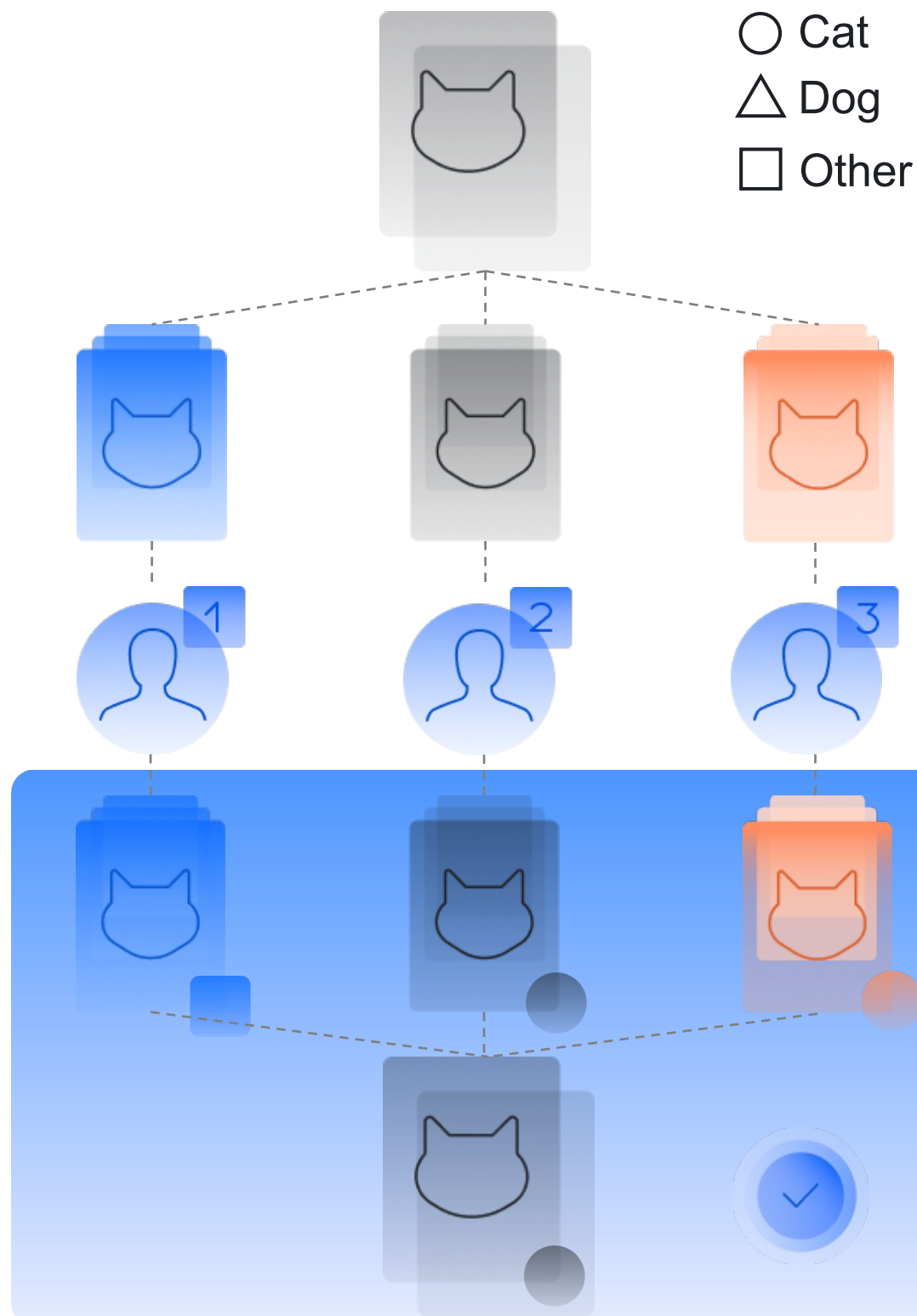
Consensus between performers



Aggregation

The background features a series of overlapping, curved, blue shapes that create a sense of depth and movement, resembling a stylized sunburst or a series of concentric, curved bands. The colors range from a deep navy blue to a bright, vibrant blue.

Aggregation



Upload multiple copies of each object to label

Performers assign noisy labels to objects

Aggregate multiple labels into a more reliable one

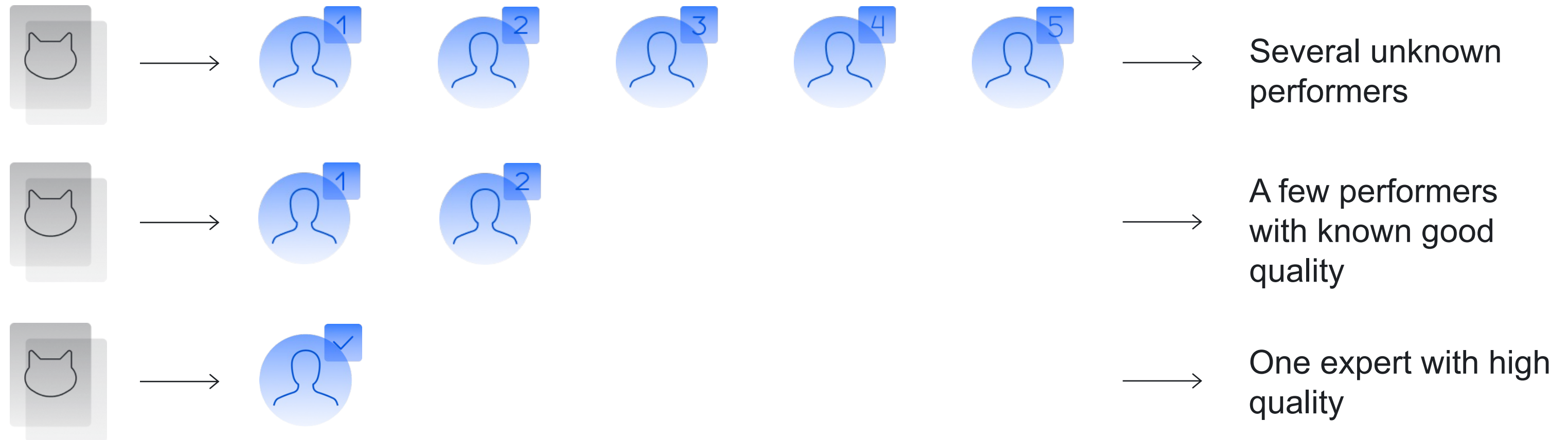
The simplest way:

- Assign the most popular answer (Majority Vote)
- There are more sophisticated methods

Incremental relabeling & pricing

Incremental relabeling

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



Pricing depends on

▶ Task design

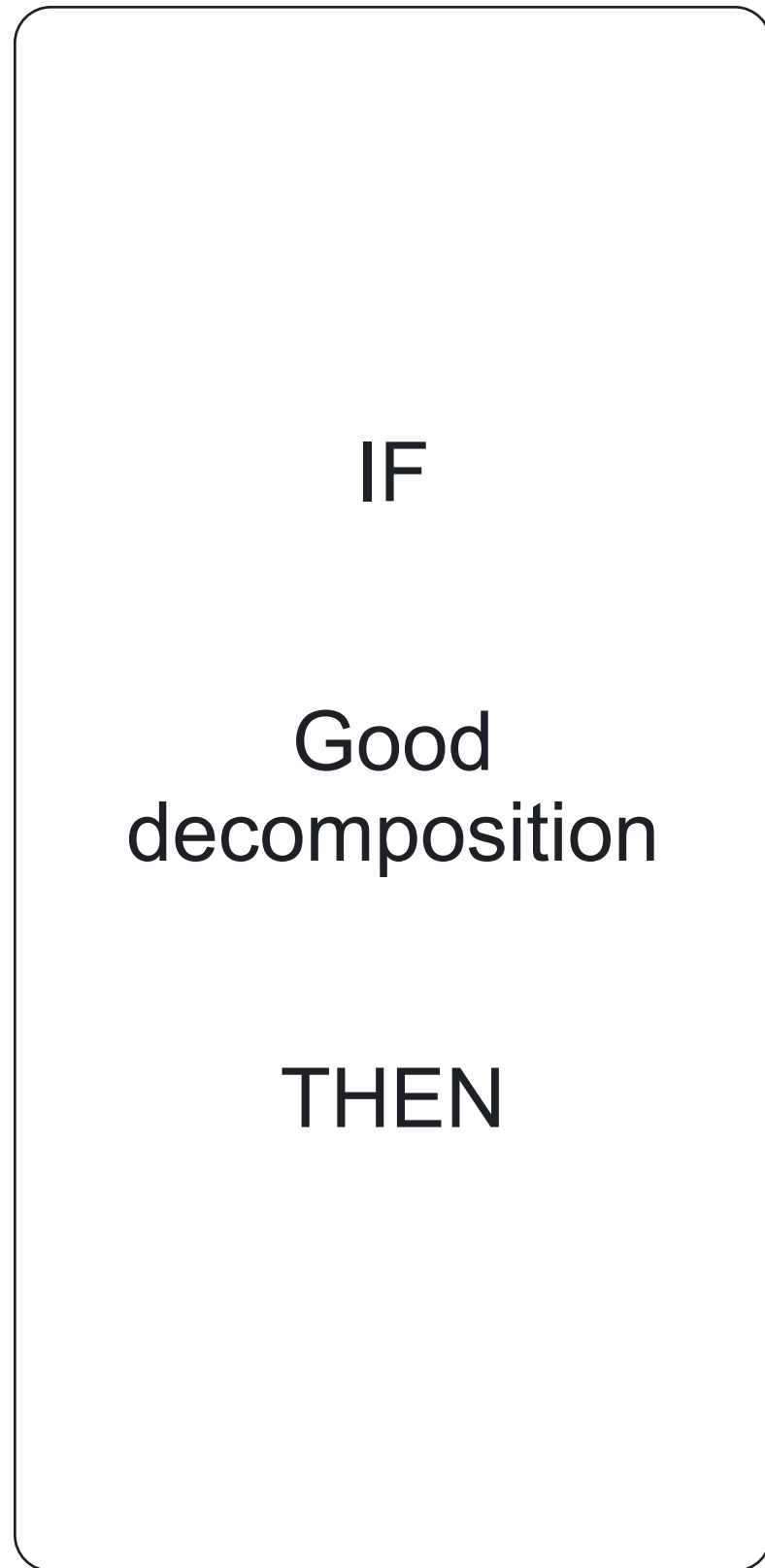
- Payment is made per a batch of microtasks (aka a task suite)
- Time required to perform a task: control hourly wage

▶ Market economy aspects

- The lower supply of performers is (e.g. due to specific skills), the higher price
- How quickly do you need the accomplished tasks (latency)?

▶ Result quality

- Incentivize better performance with a quality-dependent price



Simple instruction

Easy to use task interface

Performers do tasks with better quality

Easy to control quality

Standard aggregation models work well

Easy to control and optimize pricing

Questions?

Join our Slack: icwe_tutorial channel

Nikita Pavlichenko

Researcher



pavlichenko@toloka.ai



<https://toloka.ai/events/icwe-2022/>

