**FOIL 1**

Our paper was about the improvement of the users experience when contacting the uber customer service via a ticket system.

**FOIL 2**

At an early stage, Support Tickets at Uber had several attributes which were used in the routing process to direct the ticket to specific support agents.

Such attributes are: Country, Language and Ticket Type like Payment questions or lost items.

The tickets are categorized on these attributes and queued until an appropriate agent is available to handle them.

The simplest routing must consider all combinations of attributes and assign the ticket which has been waiting the longest to an agent.

Additional prioritization rules complicate this routing. For example are tickets from new driver partners prioritized but not all agents who support driver partner question do support new driver partner problems.

**FOIL 3**

This routing logic became comlex over time to satisfy all requirements.

Uber also grew and got into new businesses and expanded into more and more countries.

The result was a logic that became hard to follow and harder to modify.

All the rules, prioritizations of segments and combinations of attributes lead to fragmentation which also made it hard to follow and less testable and more difficult to monitor.

When writing a new feature, this process fragmentation made it difficult to find and re-use existing code, resulting in code duplication. And when bugs were introduced, they were difficult to identify and fix.

**FOIL 4**

To overcome this, a flexible but stable routing platform is needed which makes it able for developers to build own routing rules to support new products and modalities.

To address the problems, the routing logic is designed as a workflow, using Cadence, Uber’s open source orchestration engine.

With Cadence the the control logic got separated from the business logic and also the business logic got modularized.

After the re-design of the routing process as a workflow, the control logic became much easier to understand and the core routing logic got reformulated.

These steps are:

1. Prioritize categories
   1. group and sort the categories which an agent is trained to handle by priority
2. Retrieve candidate tickets
   1. retrieve all of the tickets available to an agent from the queue
3. Score candidate tickets
   1. Assign a score to each of the candidate tickets.this score is mostly based on the age of the ticket.
4. Assign top ticket
   1. The ticket with the highest score is assigned to the agent.

The new workflow helped uber to make the system easier to understand and improved the organization of the code.

**FOIL 5**

The major stakeholder identified are:

The End User, who creates a ticket and is waiting for response as well as the agent who answers to the ticket

Secondly the Maintainer who maintain the system and fixes bugs

The Implementer, who is responsible for taking individual and new components and integrating them

The Tester who actual test and verifies the system on requirements and the architecture

**FOIL 6**

Work is queued in a task list and once a workflow starts and every time execution passes from the workflow to an activity or vice versa.

This design supports durable, highly available and distributed workflow execution, but it also introduces activity to activity latencies on the order of 100 milliseconds.

This is acceptable but faster performance was needed from the routing workflow since it handles real time requests from users. Therefore, Cadence’S local activity functionality is used.

The local activity allows short-lived tasks to be scheduled and run in the workflow’s worker process instead of queueing the work in a task list.

The average performance is comparable to the original routing code