ClickHouse at Roblox Safety

bc Wong | SF ClickHouse Meetup | 2024-09-05

ROBLOX

- Safety @ Roblox
- ClickHouse usage
- Deep dive:
  A counting problem
- Future plans

ClickHouse usage

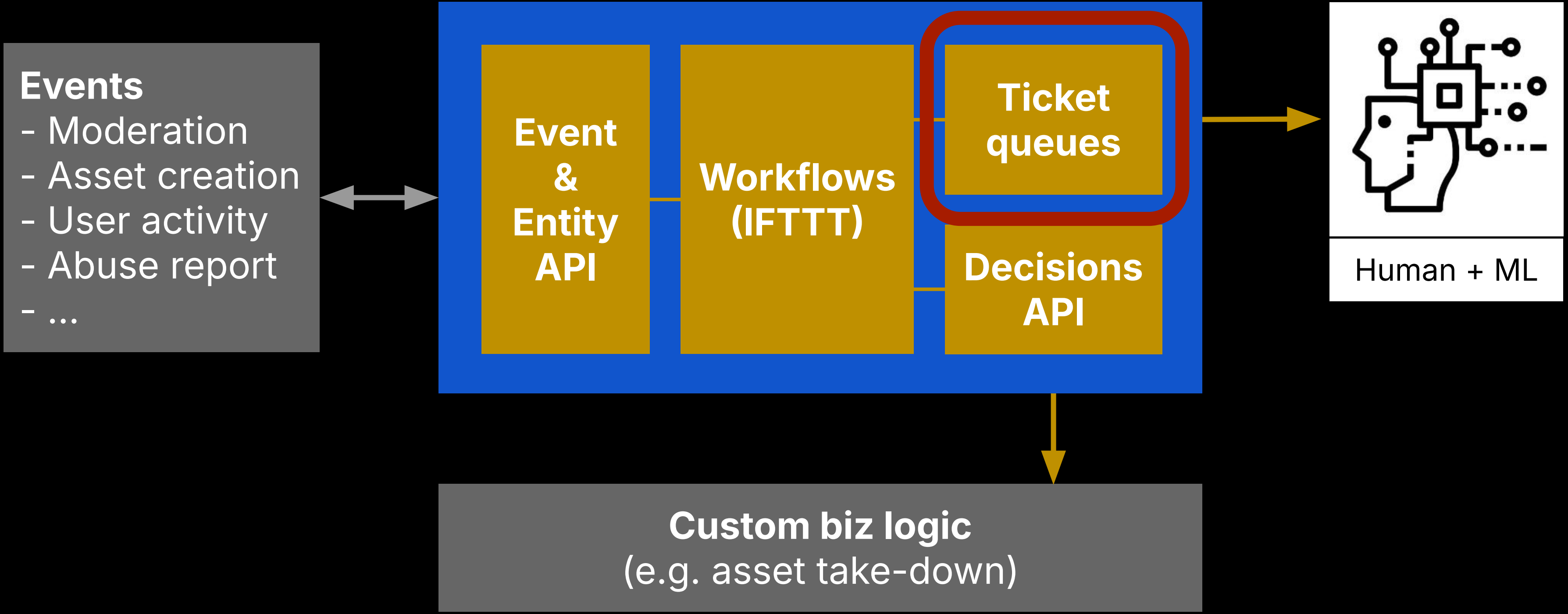
Safety is very real-time. Our usual OKR metrics are prevalence & **exposure**.

Typical OLAP use cases:

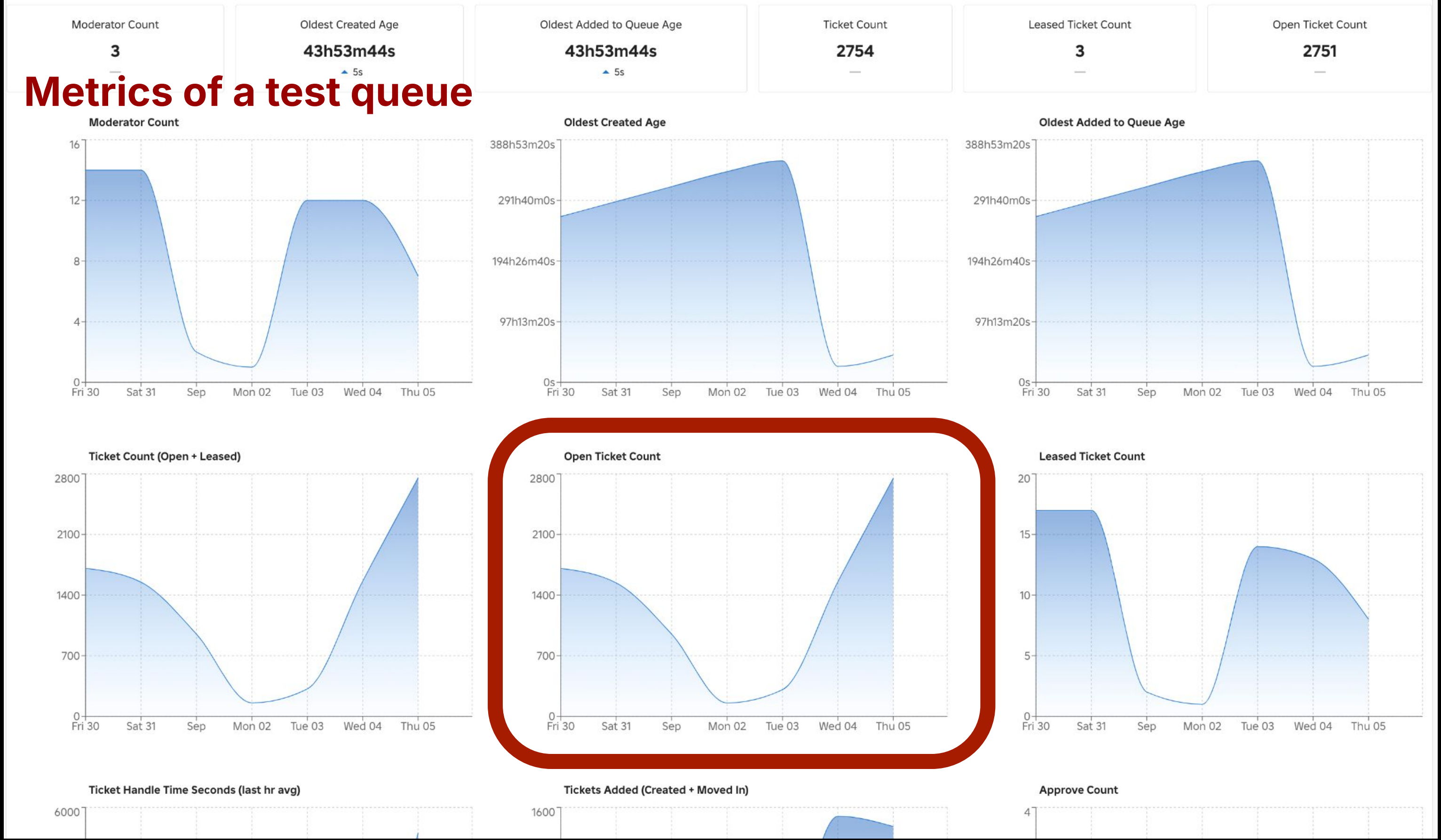
- ~100M safety-related events per day
- Debugging, especially ML decisions
- Real-time monitoring of model accuracy
- ***Operational metrics***

Deep dive: A counting problem

Roblox's platform for safety operations



Operations run on metrics



Moderator Count	Oldest Created Age	Oldest Added to Queue Age	Ticket Count	Leased Ticket Count	Open Ticket Count
3	43h53m44s	43h53m44s	2754	3	2751
—	▲ 5s	▲ 5s	—	—	—

Metrics of a test queue

Number of open tickets on each queue?

- **Data store:** Distributed KV
- **Data model:**

```
{ticket_id, queue_id, status, ...}
```

- **Common ticket operations:**
  - Create ticket
  - Lease ticket
  - Close ticket
  - Move ticket to a different queue

Credits to Harry Cho & Joshua Lanese for their work on the metrics system

Counting is hard

- Naive `count(...)` doesn't work at our scale
  a. Millions of tickets daily + thousands of queues
  b. Supervisors auto-refresh queue metrics all day long
  c. Alerting rules on the metrics

- CDC from the KV store to ClickHouse and count there.

Ticket Count = Created - Closed + Moved In - Moved Out

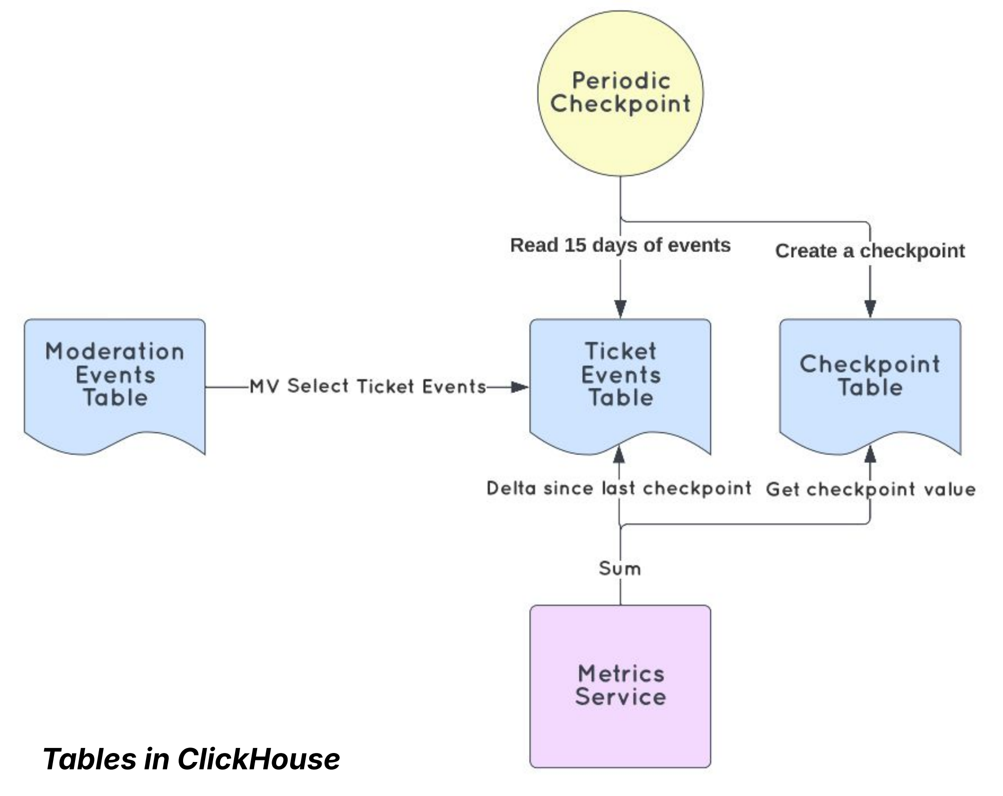
But:

a. How to make the queries performant?
b. How to avoid drift?

Use checkpoints for efficient queries

1. Checkpoint query counts every queue every N minutes.

2. Metrics Service exports minutely counts (and other metrics) to the telemetry system.



Temporary drift is ok

- Drift comes from event drop.
    - Ticket is left “open” forever.

- But all tickets have TTL!
    - Make sure the checkpoint query’s time range > TTL

Ticket Count = Created - Closed + Moved In - Moved Out
+ ***Unmatched Closed*** + ***Unmatched Moved Out***

Future plans

Self-service onboarding!

- Connect Roblox's data warehouse event stream to ClickHouse
  - The disagg architecture of ClickHouse Cloud helps
- Event stream (Kafka) is schematized (protobuf)
  - But ClickHouse isn't great at protobuf schema refresh
  - Naive solution: Convert proto to big json blob first
  - Smarter solution?