etcd

BDNR Project - Milestone 2

Group 5:

Fábio Sá
Inês Gaspar
José Gaspar

Recap

- Designed for large scale distributed systems
  - Systems that do not tolerate split-brain operations and sacrifice availability to achieve that

- Consistent key-value storage for configuration management, service discovery and coordinating distributed work
  - Consistent and fault-tolerant
  - Stability, reliability, scalability and performance

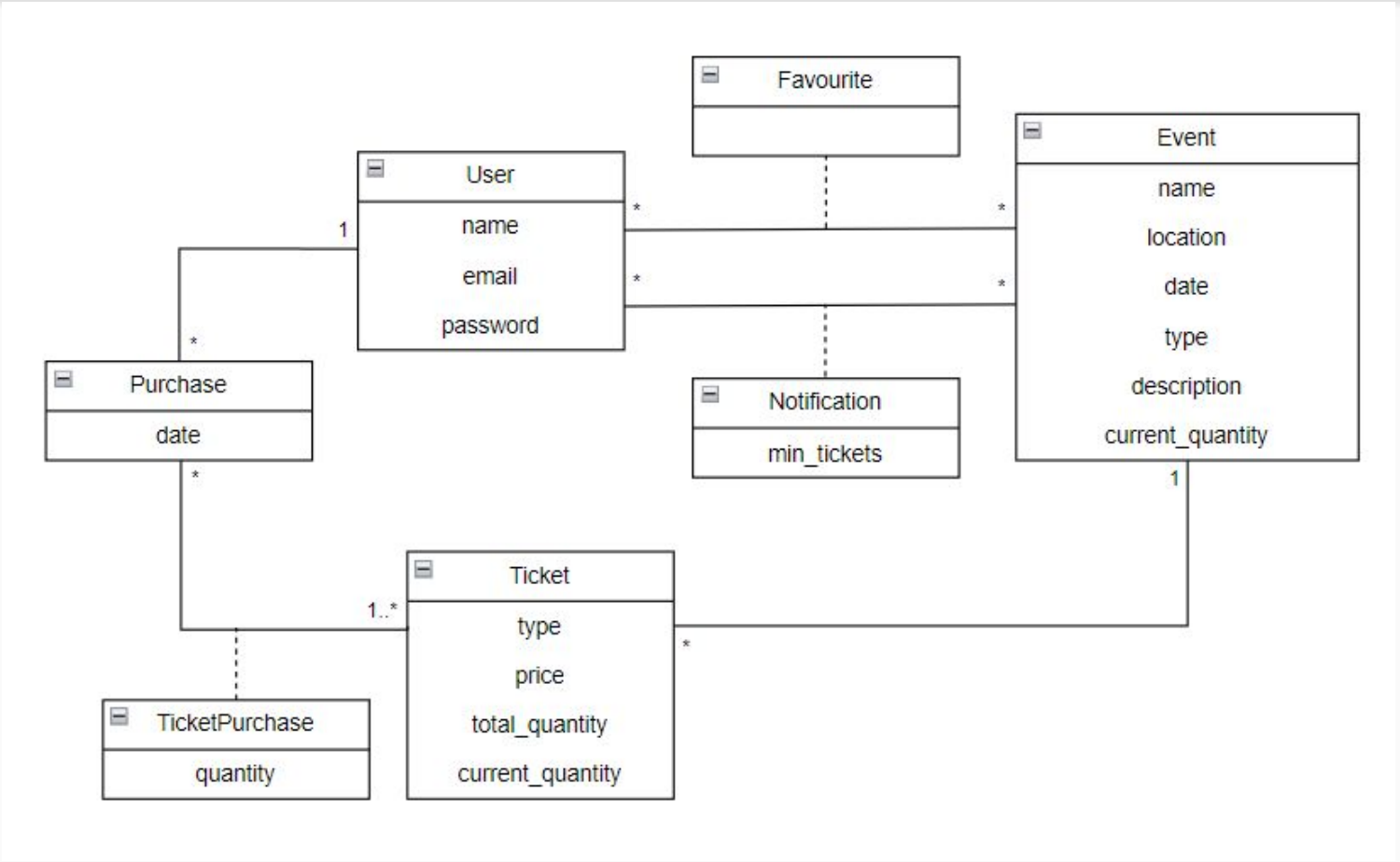
Project Description

TickETCD:

- The project consists of a ticket purchasing system for events.

- With this topic, we intend to demonstrate the advantages of this technology for distributed systems where it is essential to maintain consistency regarding ticket numbers.

Conceptual Model



Query 1

View all ticket purchases made.

```
await db.getAll().prefix(`purchase:${userID}:`).json();
```

Query 2

Determine the total revenue per event type.

```
const event_types = await db.getAll().prefix('search:type:').keys()
                            .map(type => type.split(":")[2]);

for (const event_type of event_types) {
    const events = await db.get(`search:type:${event_type}`).json();
    let total = 0;
    for (const event_id in events) {

        const ticket_types = await db.getAll().prefix(`ticket:${event_id}:`).keys()
                                    .map(type => type.split(":")[2]);

        for (const ticket_type in ticket_types) {
            const details = await db.get(`ticket:${event_id}:${ticket_type}`).json();
            total += details.price * (details.total_quantity - details.current_quantity);
        }
    }
    console.log(`${event_type} event: ${total} $`)
}
```

Query 3

Buy tickets for a particular event.

```
alltickets = {
    "date": "2024-04-08",
    "tickets": [
        {
            "type": "pink",
            "quantity": "2"
        },
        {
            "type": "yellow",
            "quantity": "1"
        }
    ]
}
```

```
await db.put(`purchase:${username}:${eventID}`).value(JSON.stringify(alltickets));
```

Query 4

Search for events based on filters (location and type) and / or textual information.

```
const matches = await db.getAll().prefix(`search:text:${input}`).json();
```

```
const matches = await db.get(`search:type:${type}`).json();
```

```
const matches = await db.get(`search:location:${location}`).json();
```

Aggregates

We have several types of aggregates, namely for the following entities:


```
"user:wendythompson": {
  "name": "Cheryl Mcdaniel",
  "email": "johnsontracy@example.com",
  "password": "DcfyEeJMel",
  "role": "user"
}
```


User


```
"event:df61e141-15d5-4d52-a9fe-bc65c21e66d2": {
  "name": "List majority yes learn suggest",
  "description": "Stop bill painting treatment differen
  "location": "Belfast",
  "type": "theater",
  "date": "20-04-2024 14:52",
  "current_quantity": 382
},
```


Event

Aggregates

We have several types of aggregates, namely for the following entities:


```
"ticket:830e5688-2cd7-4b83-a67c-0547943bb463:yellow": {
  "total_quantity": 86,
  "current_quantity": 2,
  "price": 263.83
}
```


Ticket


```
"purchase:wmoore:9d457250-14f5-40ae-aa05-a516e68b48f5": [
  {
    "date": "30-04-2024 04:16",
    "tickets": [
      {
        "type": "pink",
        "quantity": 53
      },
      {
        "type": "green",
        "quantity": 323
      }
    ]
  },
  {
    "date": "01-05-2024 06:00",
    "tickets": [
      {
        "type": "yellow",
        "quantity": 52
      },
      {
        "type": "green",
        "quantity": 6
      }
    ]
  }
```


Purchase