etcd

BDNR Project - Milestone 3

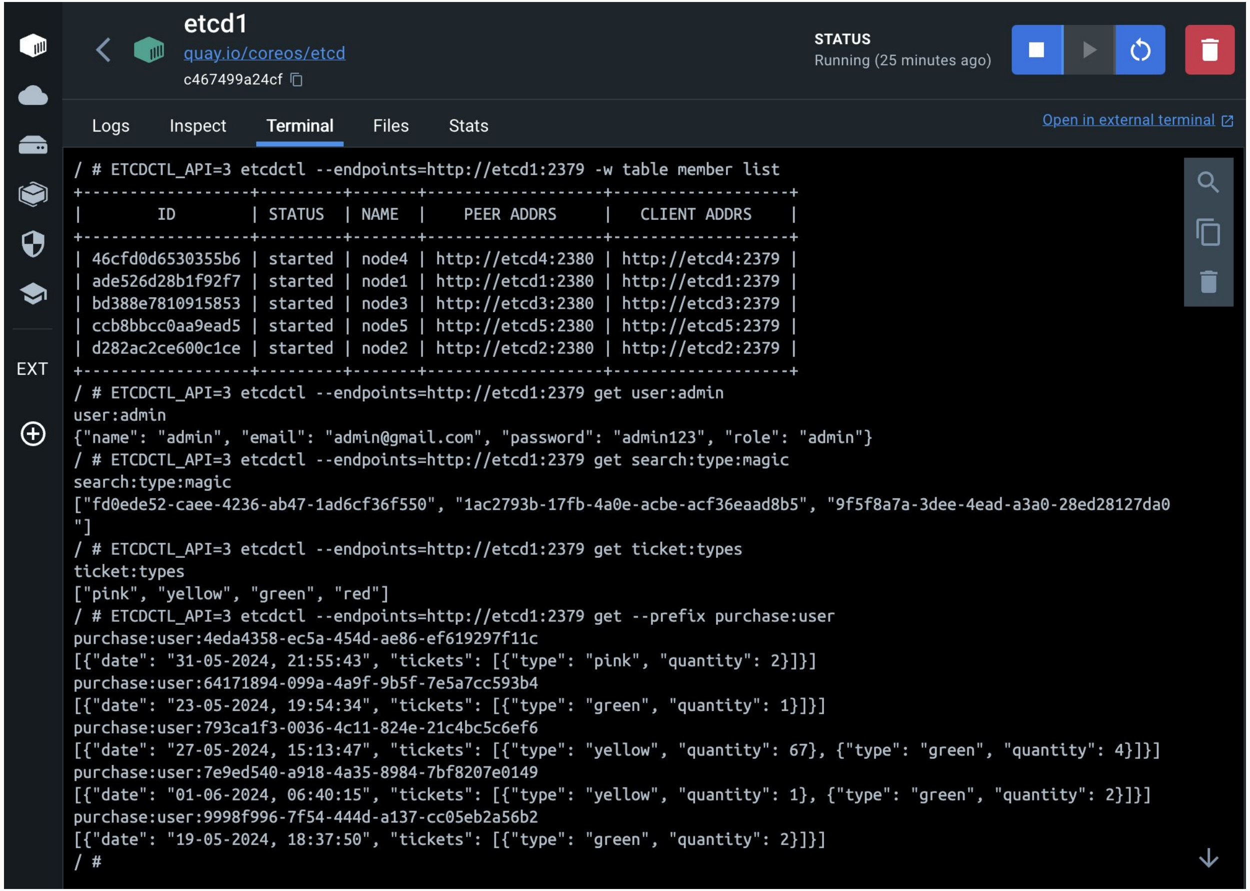
Group 5:

Fábio Sá - up202007658
Inês Gaspar - up202007210
José Gaspar - up202008561

Overview of the technology

- Key-Value database (5th key-value DB in popularity in db-engines)
- Developed by CoreOS
- Name
  - etc - related to /etc folder
  - d - from distributed
- Widely used in distributed systems
  - Configuration management and coordination of systems
- Strong consistency, fault-tolerant
- Use cases:
  - Kubernetes
  - Container Linux by CoreOS

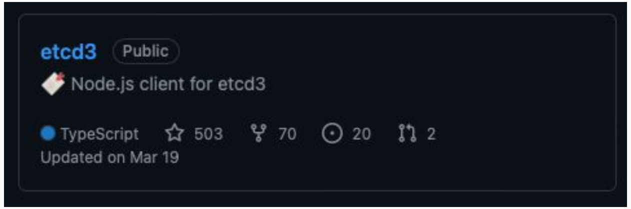
Administration overview & Client libraries



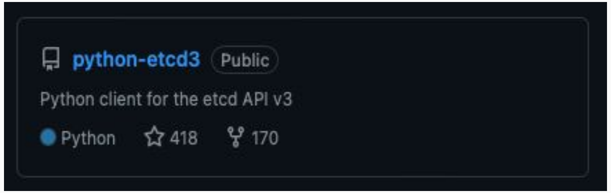
```
/ # ETCDCTL_API=3 etcdctl --endpoints=http://etcd1:2379 -w table member list
+------------------+---------+-------+-------------------+-------------------+
|        ID        | STATUS  | NAME  |    PEER ADDRS     |   CLIENT ADDRS    |
+------------------+---------+-------+-------------------+-------------------+
| 46cfd0d6530355b6 | started | node4 | http://etcd4:2380 | http://etcd4:2379 |
| ade526d28b1f92f7 | started | node1 | http://etcd1:2380 | http://etcd1:2379 |
| bd388e7810915853 | started | node3 | http://etcd3:2380 | http://etcd3:2379 |
| ccb8bbcc0aa9ead5 | started | node5 | http://etcd5:2380 | http://etcd5:2379 |
| d282ac2ce600c1ce | started | node2 | http://etcd2:2380 | http://etcd2:2379 |
+------------------+---------+-------+-------------------+-------------------+
/ # ETCDCTL_API=3 etcdctl --endpoints=http://etcd1:2379 get user:admin
user:admin
{"name": "admin", "email": "admin@gmail.com", "password": "admin123", "role": "admin"}
/ # ETCDCTL_API=3 etcdctl --endpoints=http://etcd1:2379 get search:type:magic
search:type:magic
["fd0ede52-caee-4236-ab47-1ad6cf36f550", "1ac2793b-17fb-4a0e-acbe-acf36eaad8b5", "9f5f8a7a-3dee-4ead-a3a0-28ed28127da0
"]
/ # ETCDCTL_API=3 etcdctl --endpoints=http://etcd1:2379 get ticket:types
ticket:types
["pink", "yellow", "green", "red"]
/ # ETCDCTL_API=3 etcdctl --endpoints=http://etcd1:2379 get --prefix purchase:user
purchase:user:4eda4358-ec5a-454d-ae86-ef619297f11c
[{"date": "31-05-2024, 21:55:43", "tickets": [{"type": "pink", "quantity": 2}]}]
purchase:user:64171894-099a-4a9f-9b5f-7e5a7cc593b4
[{"date": "23-05-2024, 19:54:34", "tickets": [{"type": "green", "quantity": 1}]}]
purchase:user:793ca1f3-0036-4c11-824e-21c4bc5c6ef6
[{"date": "27-05-2024, 15:13:47", "tickets": [{"type": "yellow", "quantity": 67}, {"type": "green", "quantity": 4}]}]
purchase:user:7e9ed540-a918-4a35-8984-7bf8207e0149
[{"date": "01-06-2024, 06:40:15", "tickets": [{"type": "yellow", "quantity": 1}, {"type": "green", "quantity": 2}]}]
purchase:user:9998f996-7f54-444d-a137-cc05eb2a56b2
[{"date": "19-05-2024, 18:37:50", "tickets": [{"type": "green", "quantity": 2}]}]
/ #
```

There are no official client libraries, but the etcd community recommends some:

For JS/TS:



For Python:



Data Model

Logical View

- Flat binary key space
- Multiple revisions / versions over key-value pairs
- Creating a key increments its version
- Deleting a key resets its version to 0 - tombstone

Physical View

- Persistent B+tree
  - Ordered lexically for fast ranged lookups over revision deltas
- Each revision containing only the delta from the previous
  - Very efficient for range queries over deltas

Data Operations

etcd provides a HTTP/JSON API

- GET (one or several keys)
  - single key
  - range of queries by prefix
- PUT (one key)
- DELETE (one key)
- Watcher
  - generation of watchers - used to monitor a value of a given key
  - it is possible to see previous versions of the key-value pair

Features

Replication and node communication features

- Database works in a distributed way mainly (if nodes > 1)
- Number of nodes is preferible odd
  - etcd works with quorums of size (n / 2) + 1
- Uses Raft Consensus algorithm
  - leader election
- The leader node is responsible for
  - ensuring data replication
  - load balance of requests
- FULL-REPLICATION
- FAULT-TOLERANT

Consistency features

- Sequential consistency
  - all nodes reads same events in the same order - stronger form of consistency

Eventual consistency is not enough!

⟶ can lead to problems in critical systems

Nodes do not need to be physically together!

⟶ latency tends to increase

Features

Watcher feature

- Used to monitor a given value of a certain key over time based on the operations executed over that key.
- PUT, GET or both operations can be monitored
- Useful in configuration systems

Data processing features

Functions like count, average, sum DO NOT exist!

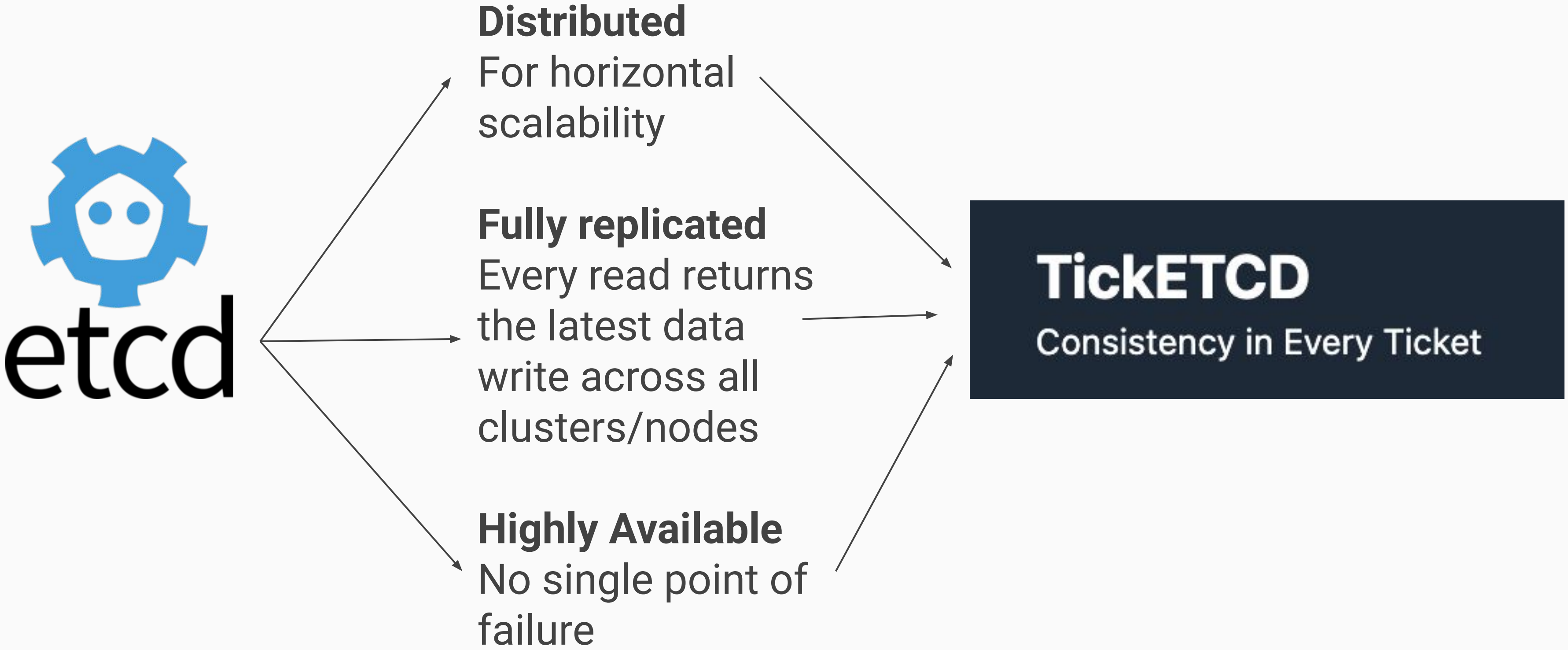
Regarding data types - etcd is limited due to its creation purpose

- only numbers and strings (all stored in binary)
- use of (de)serialization functions to bypass this limitation

Limitations of etcd

- Lack of data processing features
- Prefix search is allowed but not suffix
  - Require extra carefulness while designing aggregates
- Requests are limited to 1.5Mib and ideally DB should not have more than 8GiB (ideal size is <= 2GiB)
  - Not appropriate to handle large amounts of data
- Writes are a bottleneck and limit database scalability
  - Needing at least 50 sequential IOPS (e.g., from a 7200 RPM disk) and ideally 500 sequential IOPS (e.g., from a local SSD or high-performance virtualized block device) for heavily loaded clusters
- Total replication is also a drawback in terms of performance

Prototype TickETCD - Overview

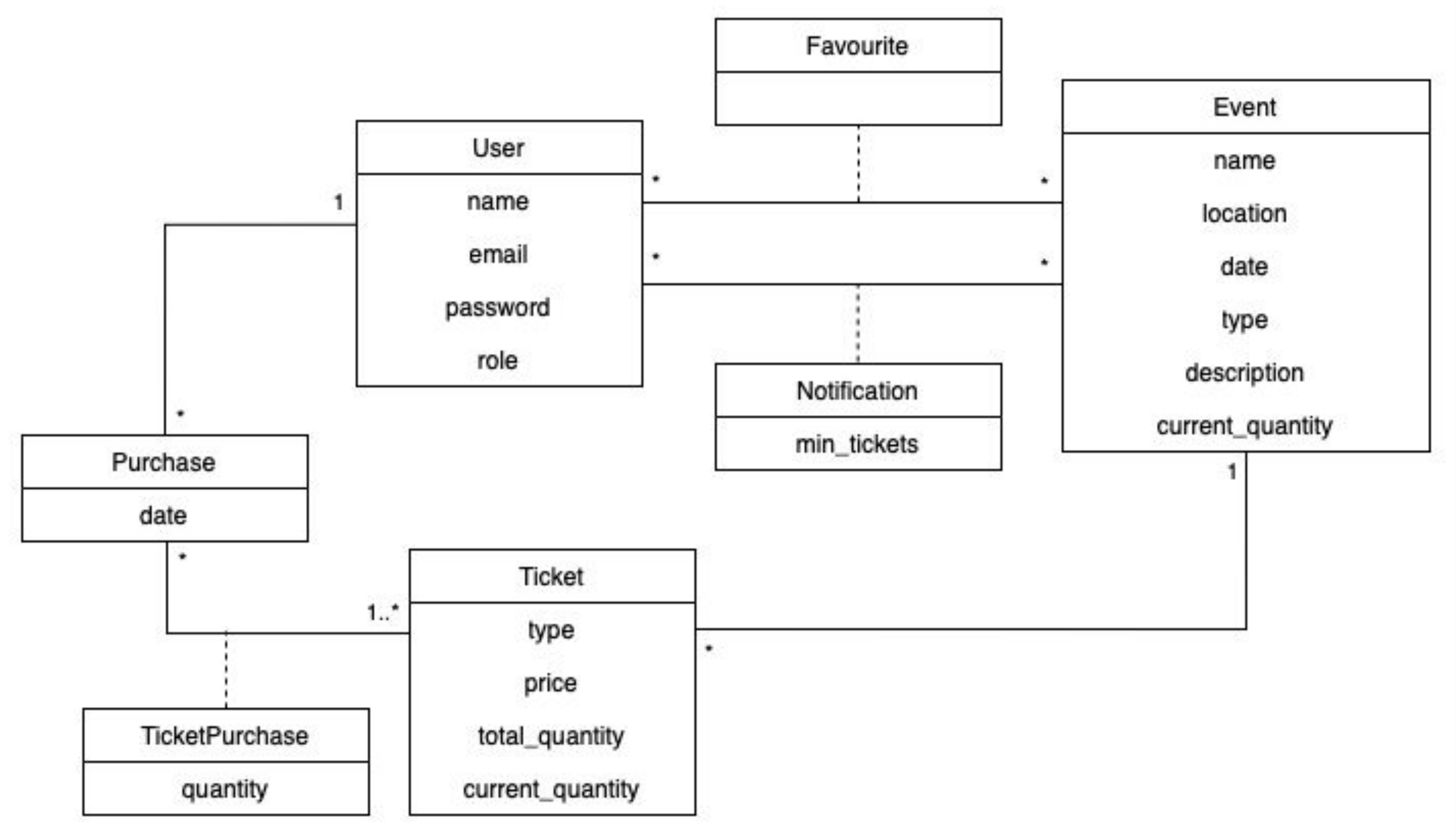


Distributed
For horizontal scalability

Fully replicated
Every read returns the latest data write across all clusters/nodes

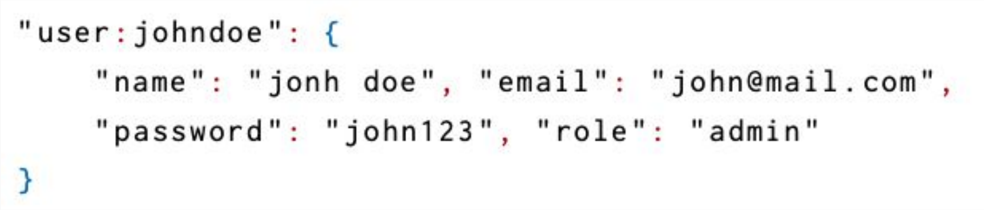
Highly Available
No single point of failure

Prototype TickETCD - Conceptual Model



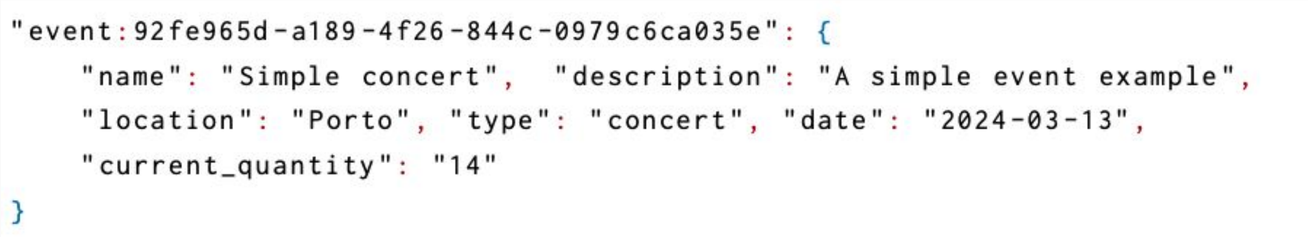
Prototype TickETCD - Physical Model (1/3)

user:<USERNAME>



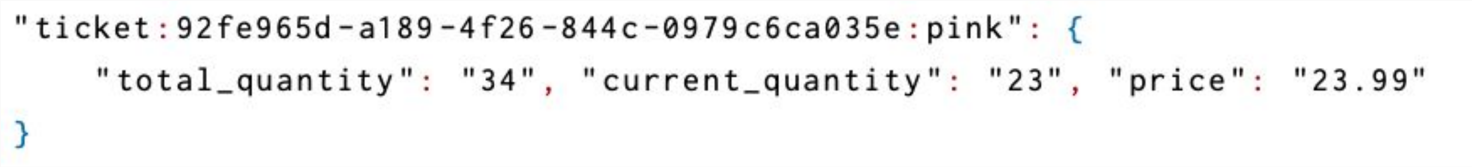
```
"user:johndoe": {
    "name": "jonh doe", "email": "john@mail.com",
    "password": "john123", "role": "admin"
}
```

event:<EVENT_ID>



```
"event:92fe965d-a189-4f26-844c-0979c6ca035e": {
    "name": "Simple concert",  "description": "A simple event example",
    "location": "Porto", "type": "concert", "date": "2024-03-13",
    "current_quantity": "14"
}
```

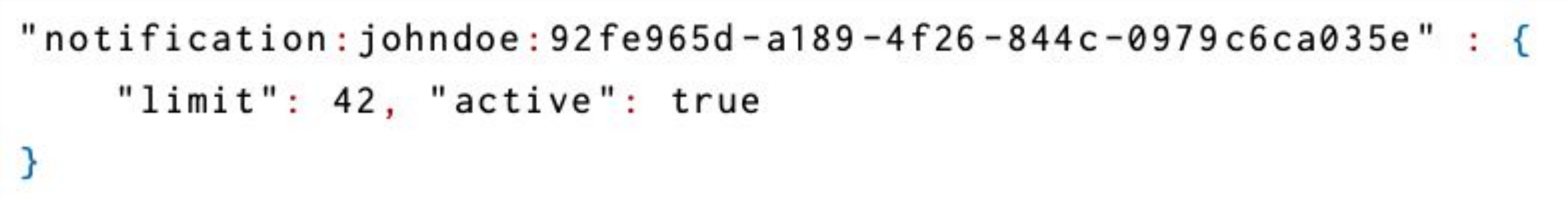
ticket:<EVENT_ID>:<TYPE>



```
"ticket:92fe965d-a189-4f26-844c-0979c6ca035e:pink": {
    "total_quantity": "34", "current_quantity": "23", "price": "23.99"
}
```

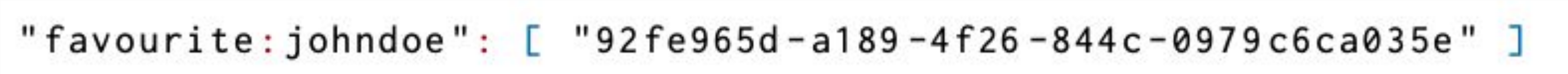
Prototype TickETCD - Physical Model (2/3)

notification:<USERNAME>:<EVENT_ID>



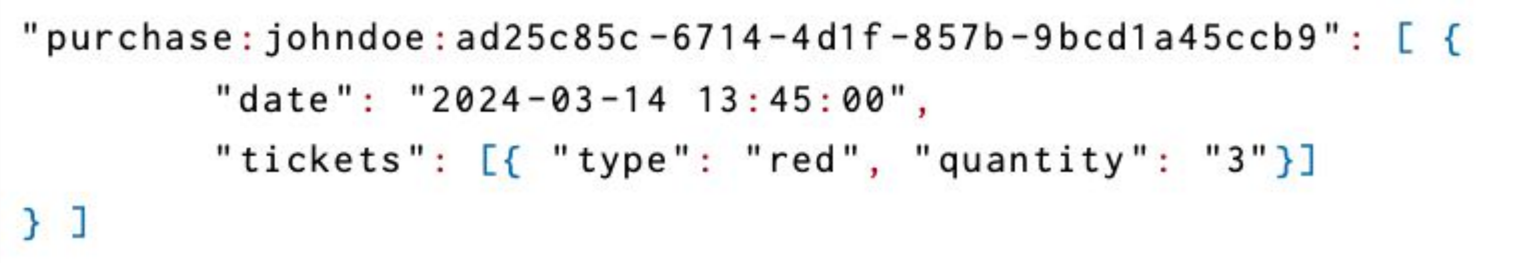
```
"notification:johndoe:92fe965d-a189-4f26-844c-0979c6ca035e" : {
    "limit": 42, "active": true
}
```

favourite:<USERNAME>



```
"favourite:johndoe": [ "92fe965d-a189-4f26-844c-0979c6ca035e" ]
```

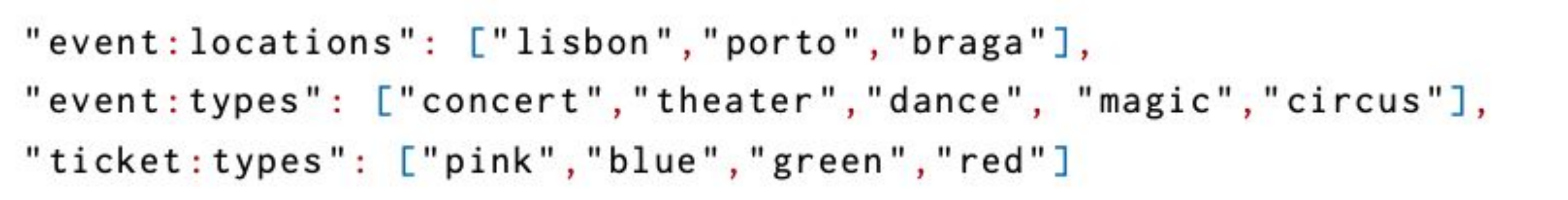
purchase:<USERNAME>:<EVENT_ID>



```
"purchase:johndoe:ad25c85c-6714-4d1f-857b-9bcd1a45ccb9": [ {
        "date": "2024-03-14 13:45:00",
        "tickets": [{ "type": "red", "quantity": "3"}]
} ]
```

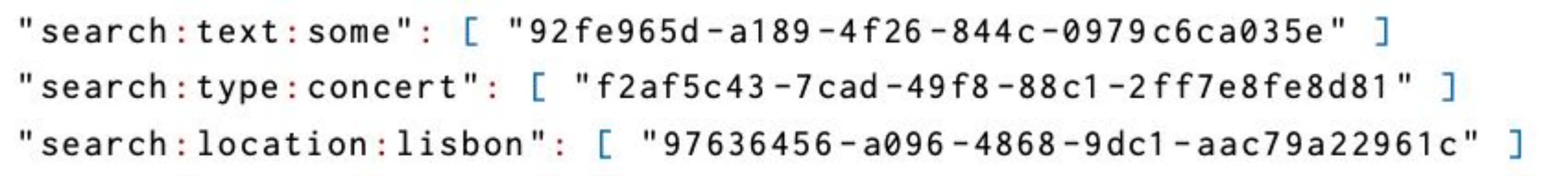
Prototype TickETCD - Physical Model (3/3)

Static data



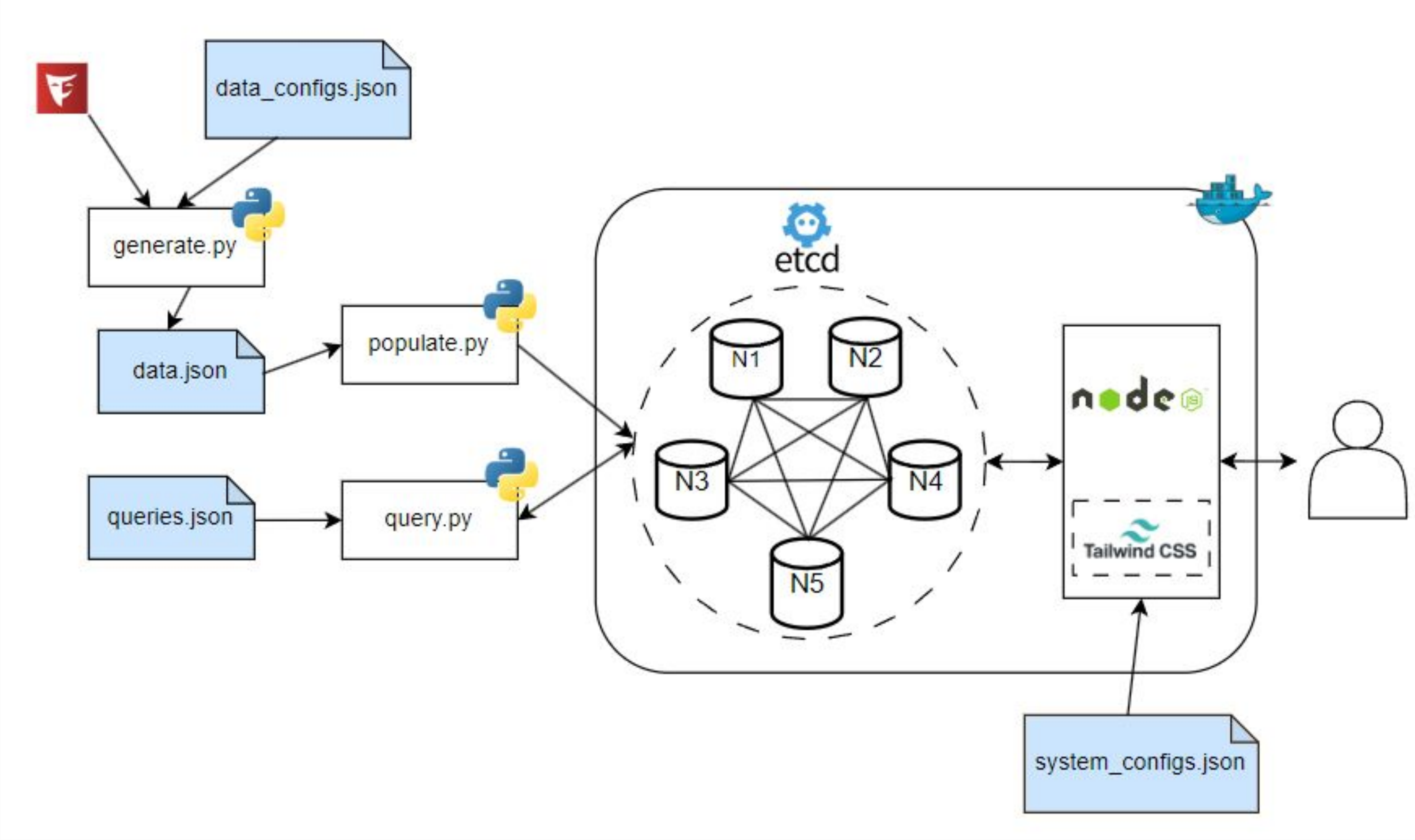
```
"event:locations": ["lisbon","porto","braga"],
"event:types": ["concert","theater","dance", "magic","circus"],
"ticket:types": ["pink","blue","green","red"]
```

search:<TYPE>:<INPUT>

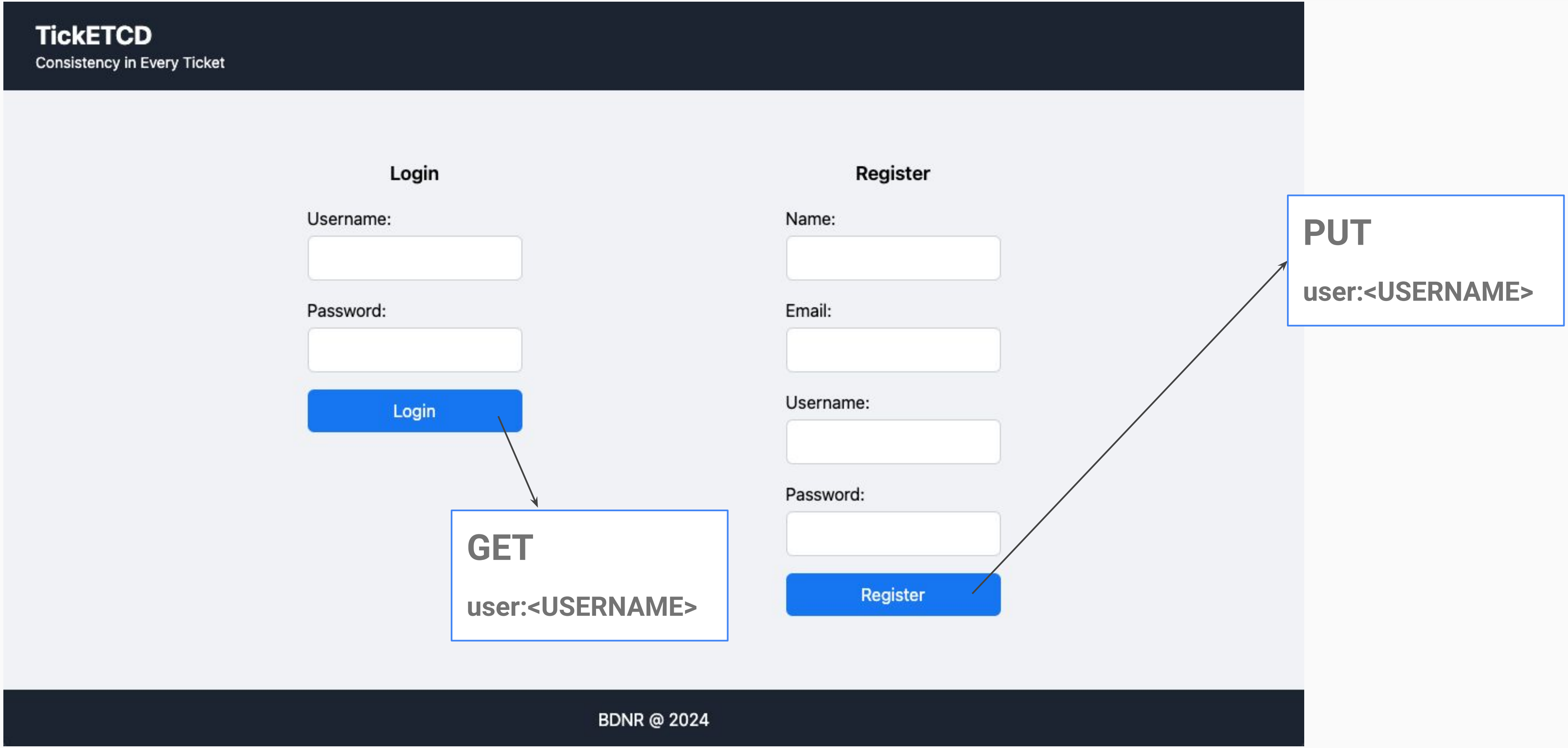


```
"search:text:some": [ "92fe965d-a189-4f26-844c-0979c6ca035e" ]
"search:type:concert": [ "f2af5c43-7cad-49f8-88c1-2ff7e8fe8d81" ]
"search:location:lisbon": [ "97636456-a096-4868-9dc1-aac79a22961c" ]
```

Prototype TickETCD - Architecture



Prototype TickETCD - Features (1/10)



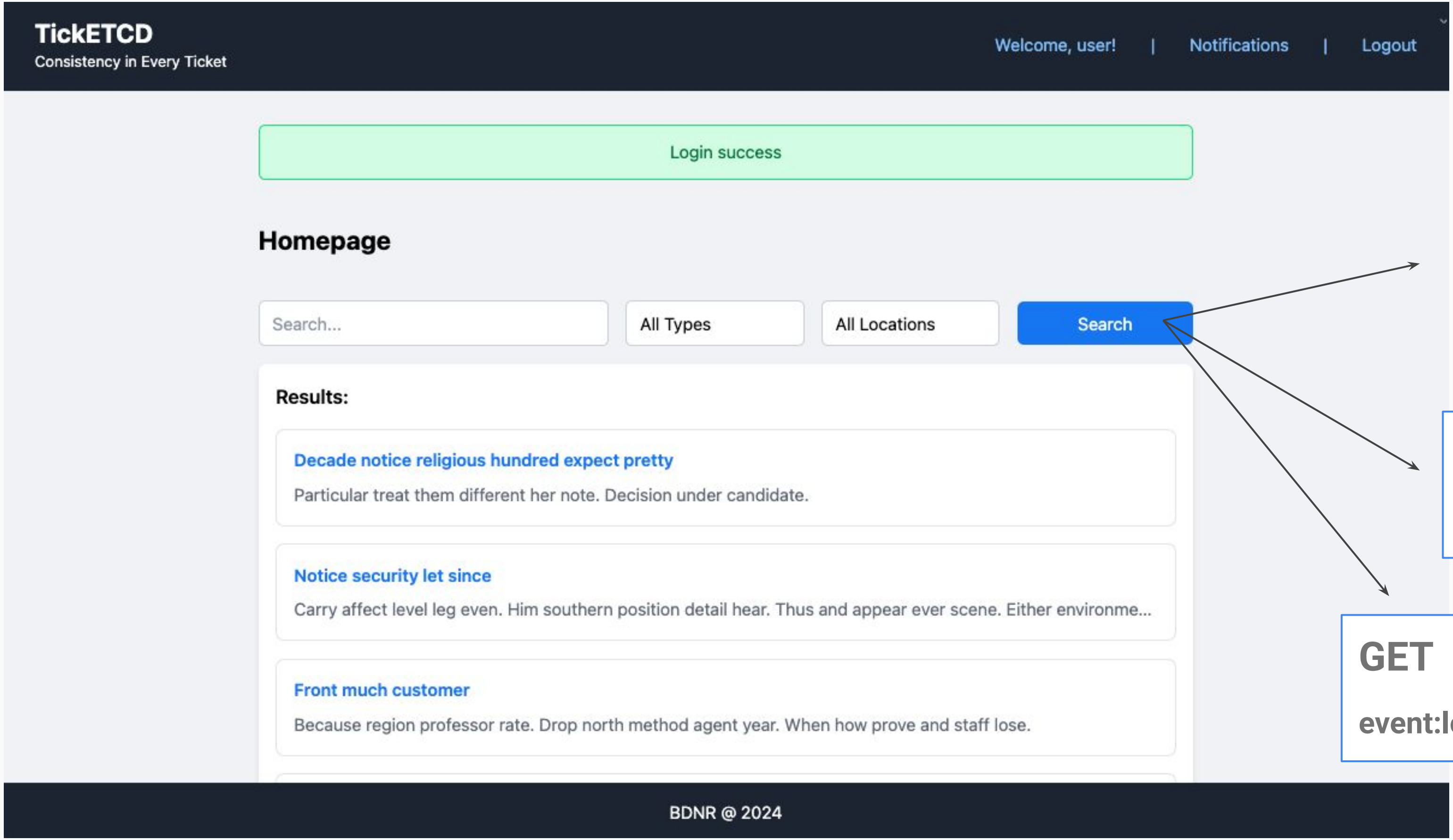
GET

user:<USERNAME>

PUT

user:<USERNAME>

Prototype TickETCD - Features (2/10)



TickETCD
Consistency in Every Ticket
Welcome, user! | Notifications | Logout

Login success

Homepage

Search...
All Types
All Locations
Search

Results:

Decade notice religious hundred expect pretty

Particular treat them different her note. Decision under candidate.

Notice security let since

Carry affect level leg even. Him southern position detail hear. Thus and appear ever scene. Either environme...

Front much customer

Because region professor rate. Drop north method agent year. When how prove and staff lose.

BDNR @ 2024

GET

event:text:<INPUT>

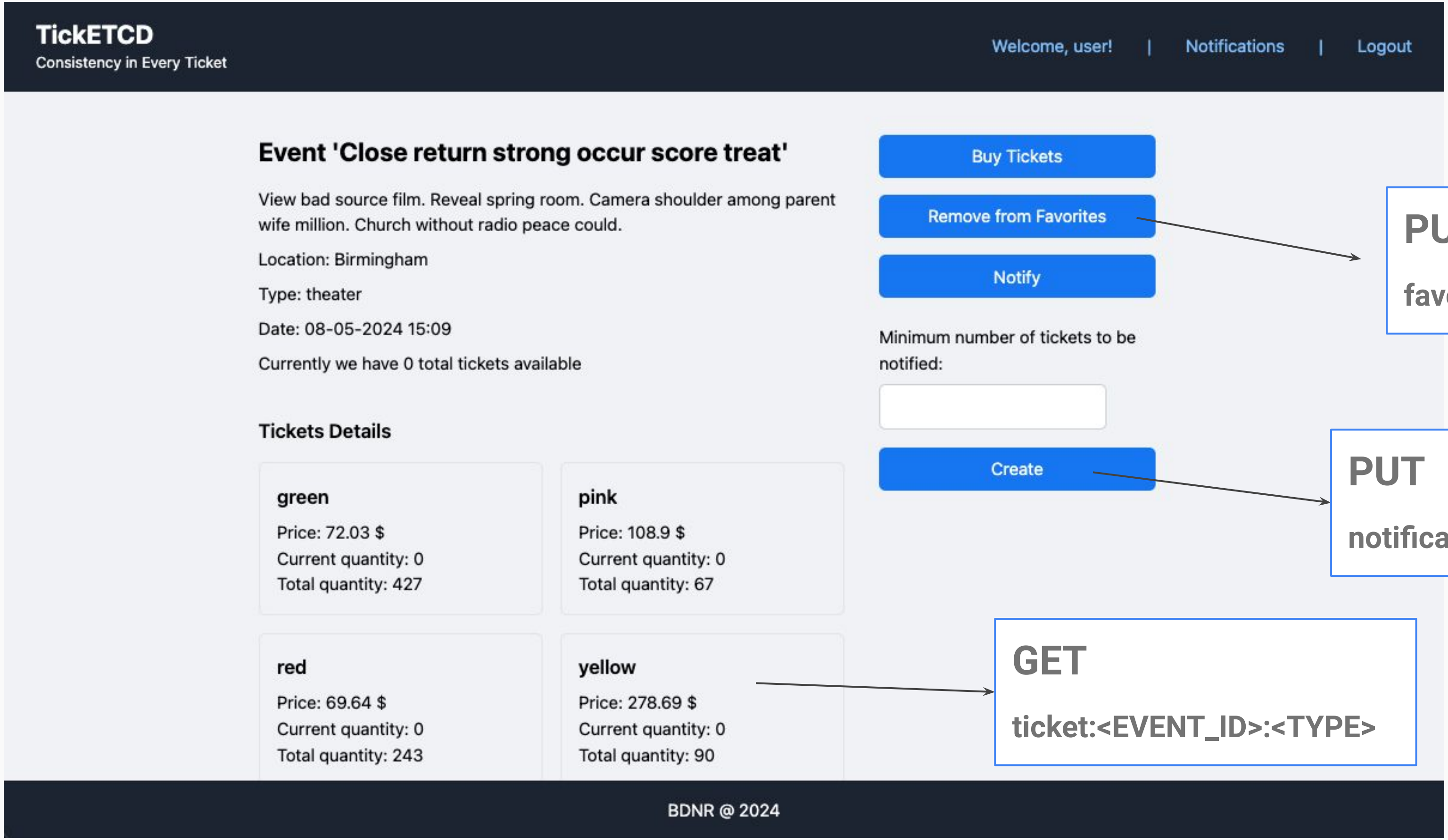
GET

event:type:<TYPE>

GET

event:location:<LOCATION>

Prototype TickETCD - Features (3/10)



TickETCD
Consistency in Every Ticket
Welcome, user! | Notifications | Logout

Event 'Close return strong occur score treat'

View bad source film. Reveal spring room. Camera shoulder among parent wife million. Church without radio peace could.

Location: Birmingham

Type: theater

Date: 08-05-2024 15:09

Currently we have 0 total tickets available

Tickets Details

green
Price: 72.03 $
Current quantity: 0
Total quantity: 427

pink
Price: 108.9 $
Current quantity: 0
Total quantity: 67

red
Price: 69.64 $
Current quantity: 0
Total quantity: 243

yellow
Price: 278.69 $
Current quantity: 0
Total quantity: 90

Minimum number of tickets to be notified:

PUT/DELETE

favourite:<USERNAME>

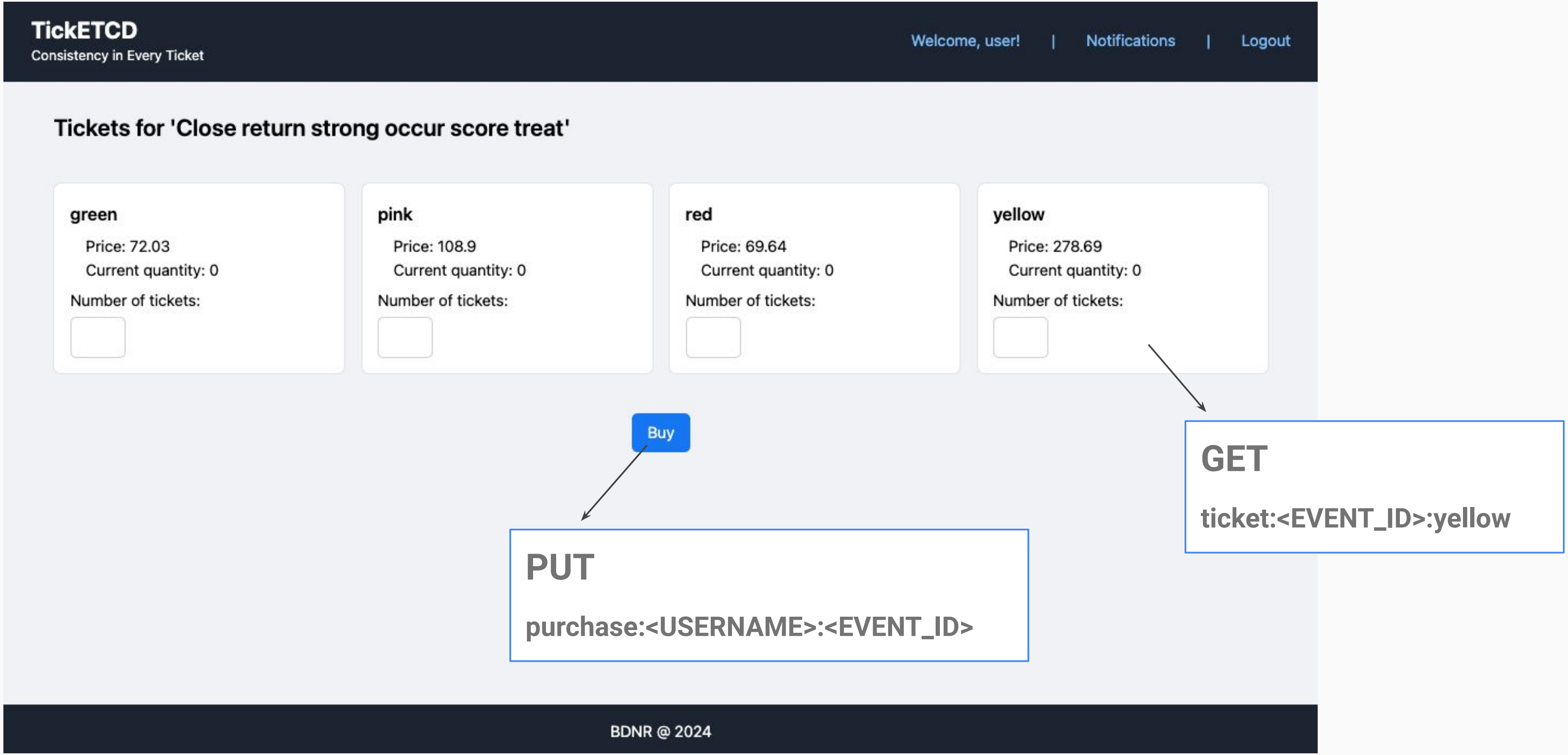
PUT

notification:<USERNAME>

GET

ticket:<EVENT_ID>:<TYPE>

Prototype TickETCD - Features (4/10)



TickETCD
Consistency in Every Ticket

Welcome, user! | Notifications | Logout

Tickets for 'Close return strong occur score treat'

green
Price: 72.03
Current quantity: 0
Number of tickets:

pink
Price: 108.9
Current quantity: 0
Number of tickets:

red
Price: 69.64
Current quantity: 0
Number of tickets:

yellow
Price: 278.69
Current quantity: 0
Number of tickets:

BDNR @ 2024

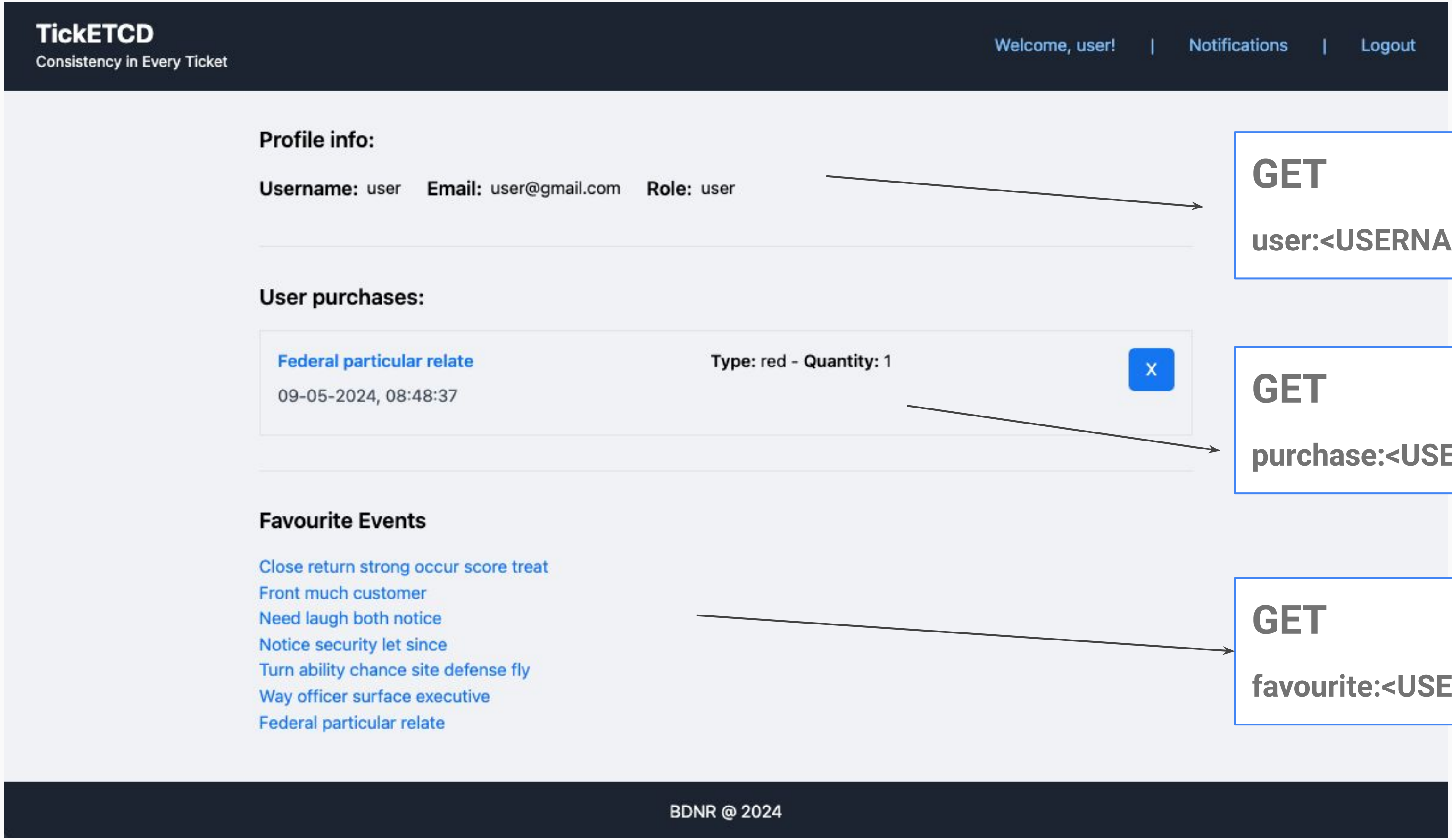
GET

ticket:<EVENT_ID>:yellow

PUT

purchase:<USERNAME>:<EVENT_ID>

Prototype TickETCD - Features (5/10)



TickETCD
Consistency in Every Ticket

Welcome, user! | Notifications | Logout

Profile info:

Username: user **Email:** user@gmail.com **Role:** user

User purchases:

Federal particular relate **Type:** red - **Quantity:** 1

09-05-2024, 08:48:37

Favourite Events

Close return strong occur score treat
Front much customer
Need laugh both notice
Notice security let since
Turn ability chance site defense fly
Way officer surface executive
Federal particular relate

BDNR @ 2024

GET

user:<USERNAME>

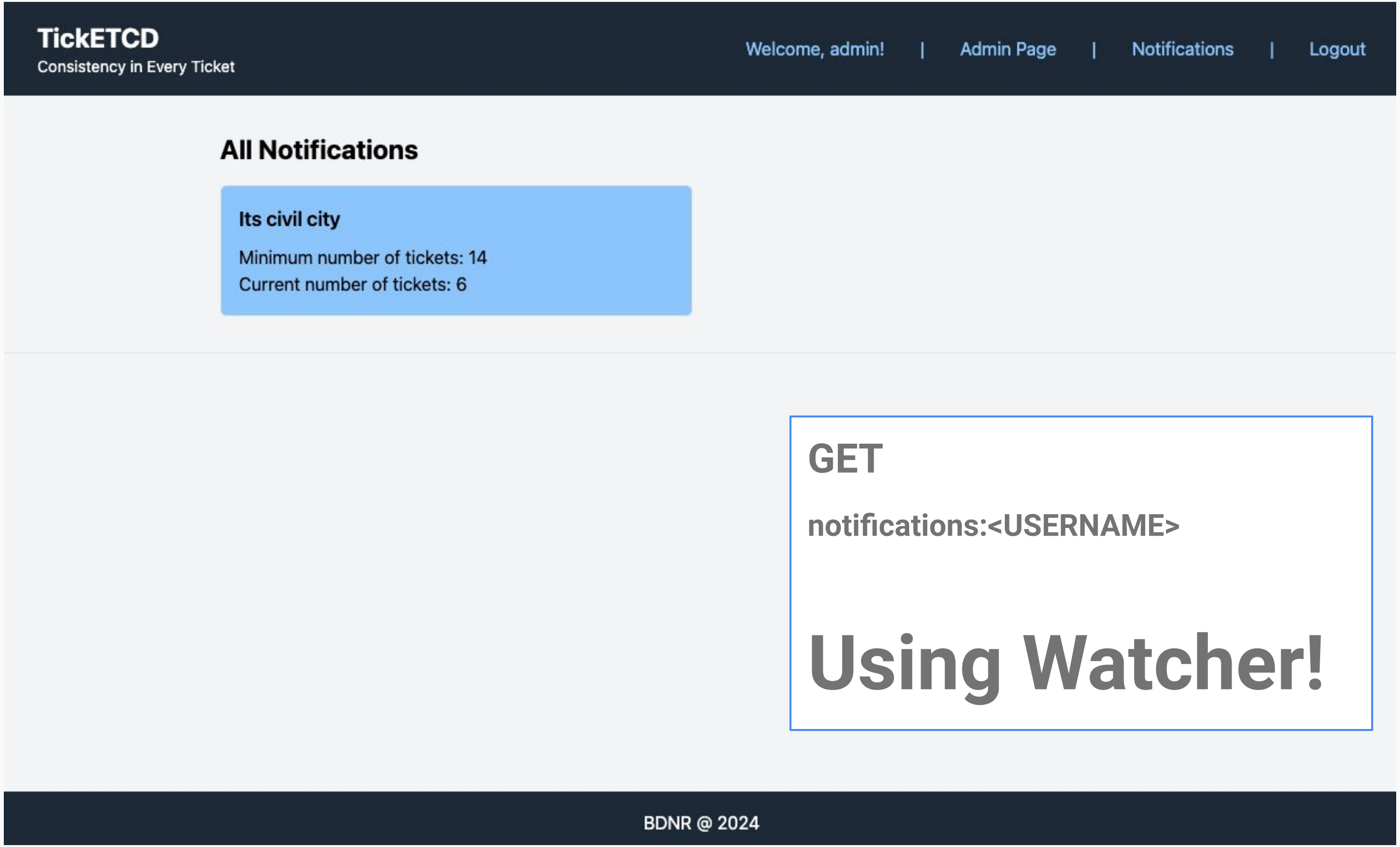
GET

purchase:<USERNAME>

GET

favourite:<USERNAME>

Prototype TickETCD - Features (6/10)



TickETCD
Consistency in Every Ticket
Welcome, admin! | Admin Page | Notifications | Logout

All Notifications

Its civil city

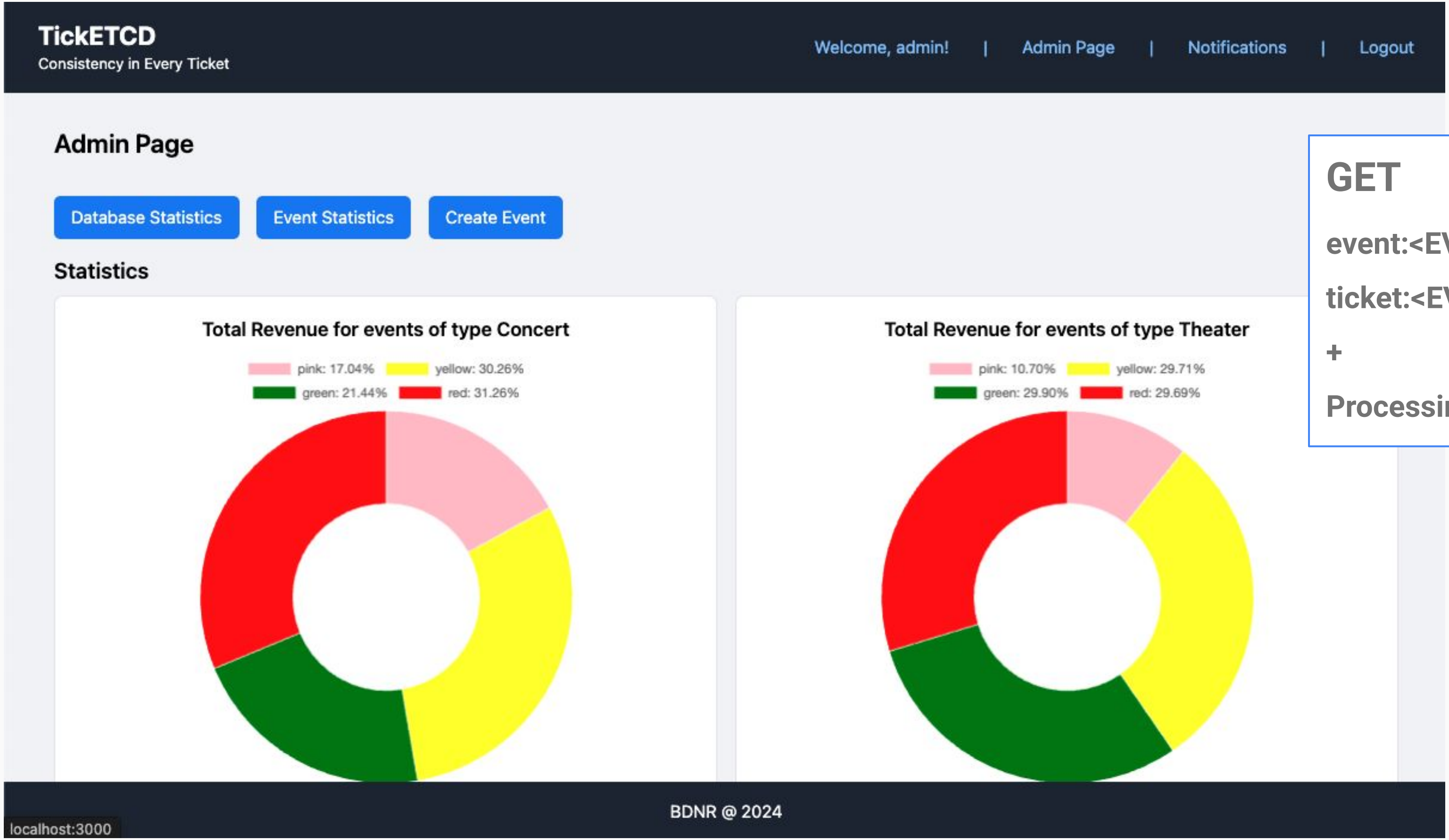
Minimum number of tickets: 14
Current number of tickets: 6

GET

notifications:<USERNAME>

Using Watcher!

Prototype TickETCD - Features (7/10)



Admin Page

Statistics

Total Revenue for events of type Concert

Total Revenue for events of type Theater

GET

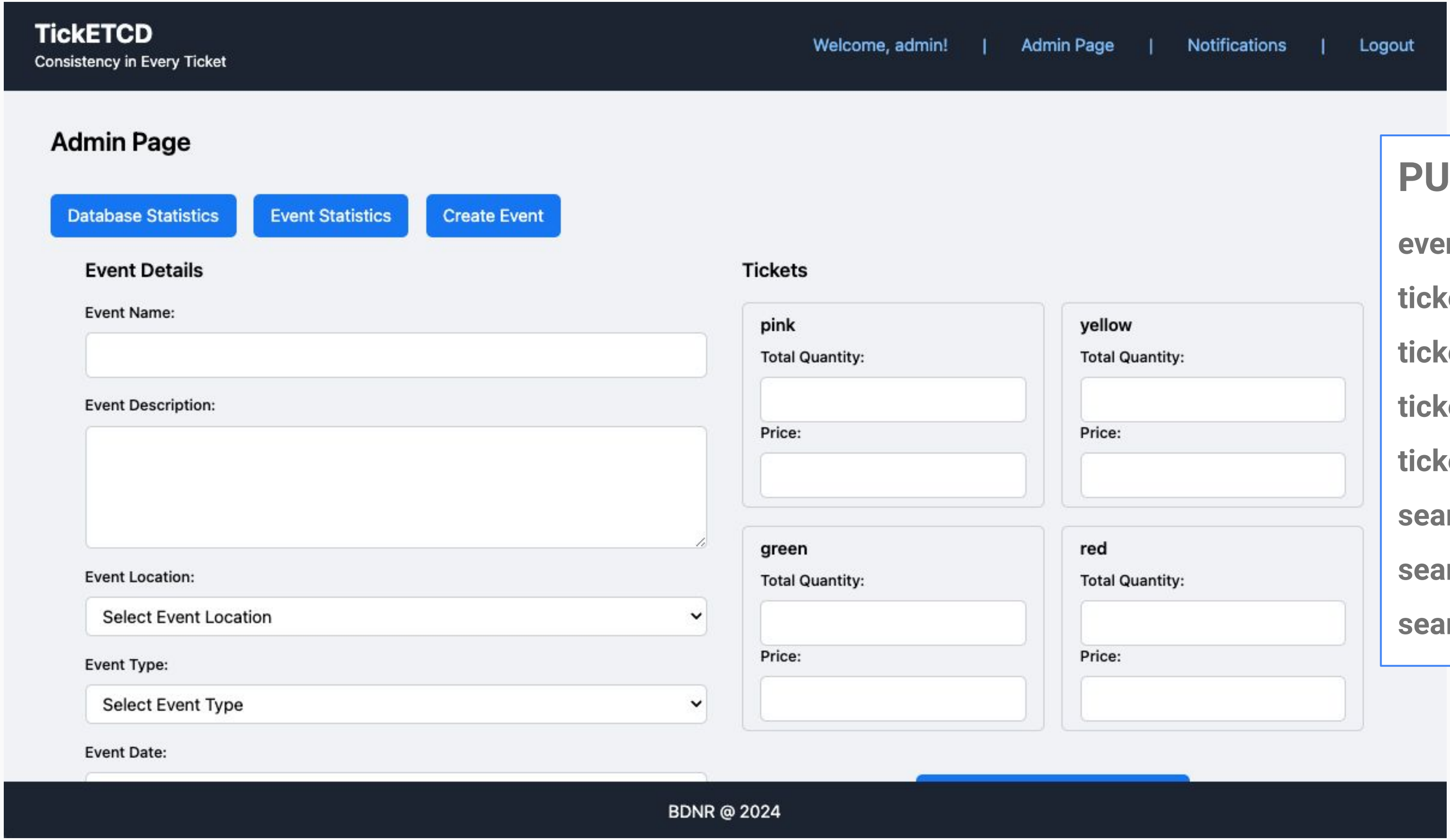
event:<EVENT_ID>

ticket:<EVENT_ID>:<TYPE>

+

Processing

Prototype TickETCD - Features (8/10)



TickETCD

Consistency in Every Ticket

Welcome, admin! | Admin Page | Notifications | Logout

Admin Page

Database Statistics | Event Statistics | Create Event

Event Details

Event Name:

Event Description:

Event Location:

Select Event Location

Event Type:

Select Event Type

Event Date:

Tickets

pink

Total Quantity:

Price:

yellow

Total Quantity:

Price:

green

Total Quantity:

Price:

red

Total Quantity:

Price:

BDNR @ 2024

PUT

event:<EVENT_ID>

ticket:<EVENT_ID>:pink

ticket:<EVENT_ID>:yellow

ticket:<EVENT_ID>:green

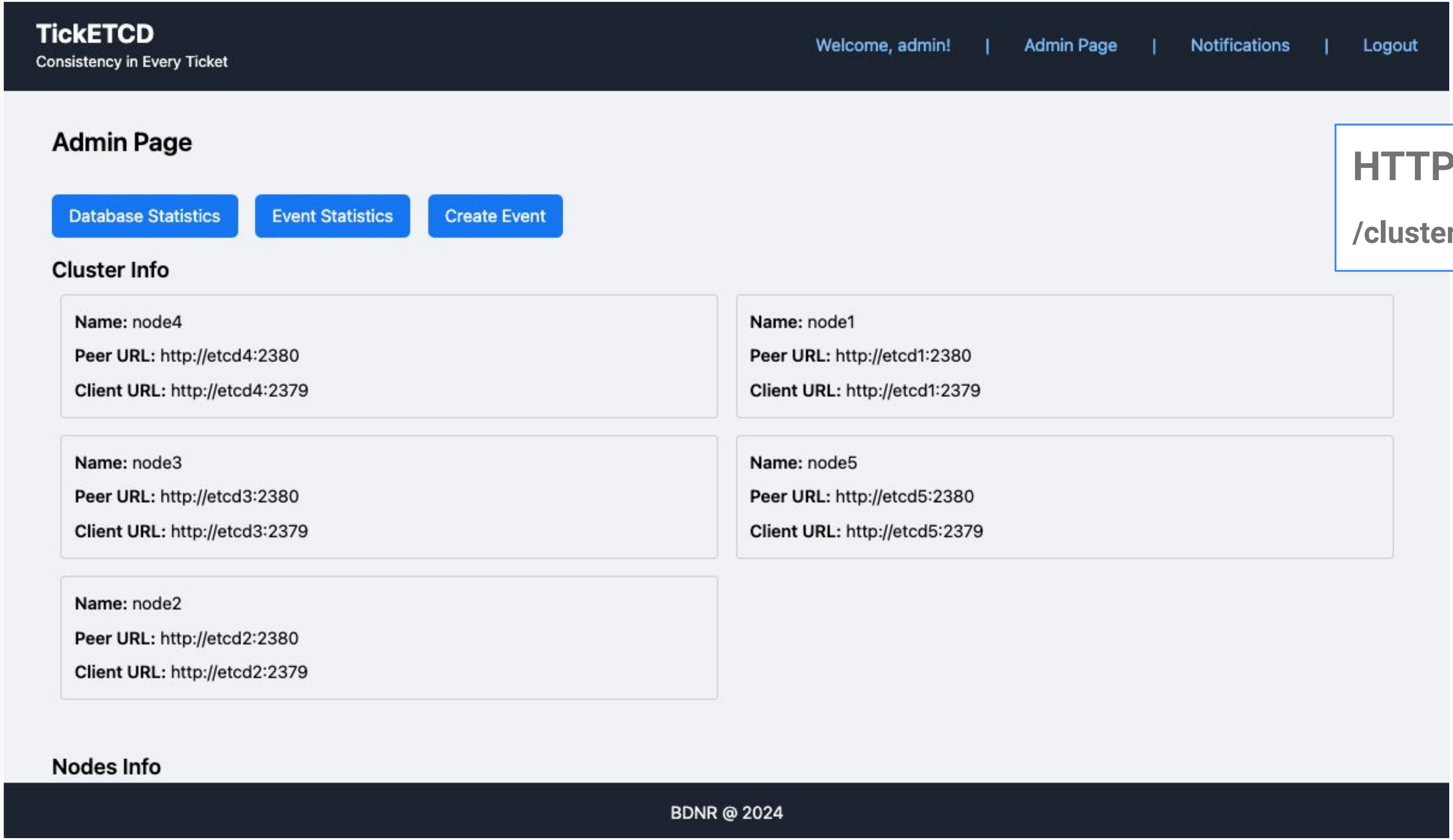
ticket:<EVENT_ID>:red

search:type:<INPUT>

search:location:<INPUT>

search:text:<INPUT>

Prototype TickETCD - Features (9/10)


TickETCD
Consistency in Every Ticket
Welcome, admin! | Admin Page | Notifications | Logout

Admin Page

Database Statistics
Event Statistics
Create Event

Cluster Info

Name: node4
Peer URL: http://etcd4:2380
Client URL: http://etcd4:2379

Name: node1
Peer URL: http://etcd1:2380
Client URL: http://etcd1:2379

Name: node3
Peer URL: http://etcd3:2380
Client URL: http://etcd3:2379

Name: node5
Peer URL: http://etcd5:2380
Client URL: http://etcd5:2379

Name: node2
Peer URL: http://etcd2:2380
Client URL: http://etcd2:2379

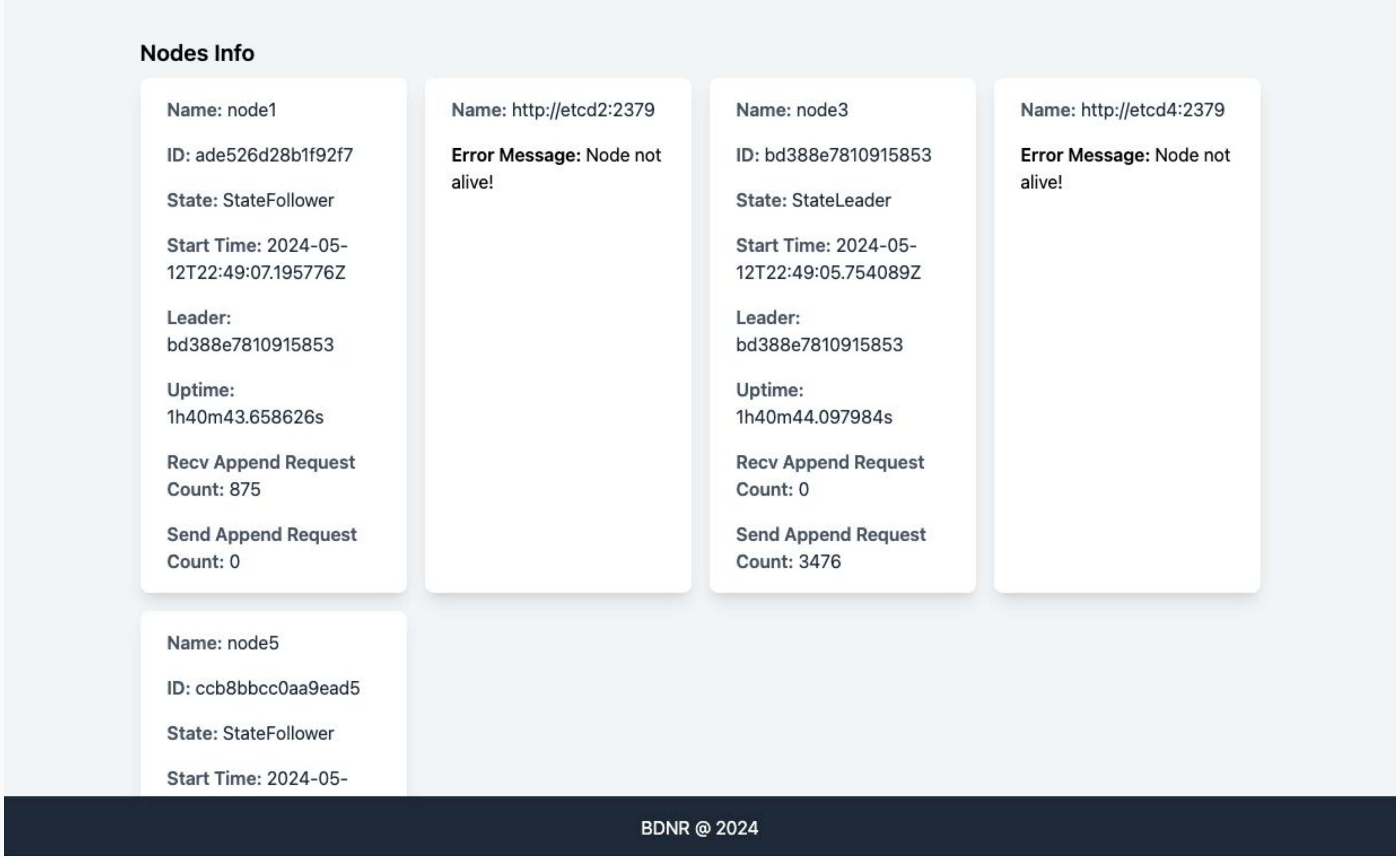
Nodes Info

BDNR @ 2024

HTTP GET

/cluster/members

Prototype TickETCD - Features (10/10)



Nodes Info

Name: node1

ID: ade526d28b1f92f7

State: StateFollower

Start Time: 2024-05-12T22:49:07.195776Z

Leader: bd388e7810915853

Uptime: 1h40m43.658626s

Recv Append Request Count: 875

Send Append Request Count: 0

Name: http://etcd2:2379

Error Message: Node not alive!

Name: node3

ID: bd388e7810915853

State: StateLeader

Start Time: 2024-05-12T22:49:05.754089Z

Leader: bd388e7810915853

Uptime: 1h40m44.097984s

Recv Append Request Count: 0

Send Append Request Count: 3476

Name: http://etcd4:2379

Error Message: Node not alive!

Name: node5

ID: ccb8bbcc0aa9ead5

State: StateFollower

Start Time: 2024-05-

BDNR @ 2024

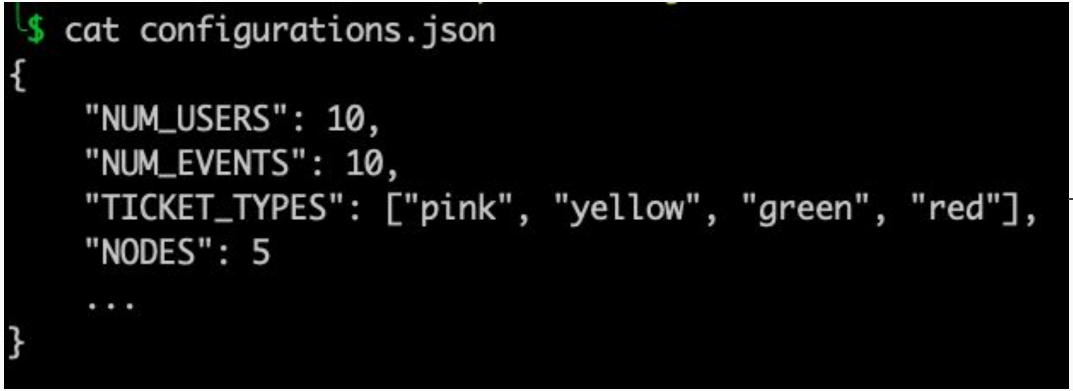
HTTP GET

/node/info

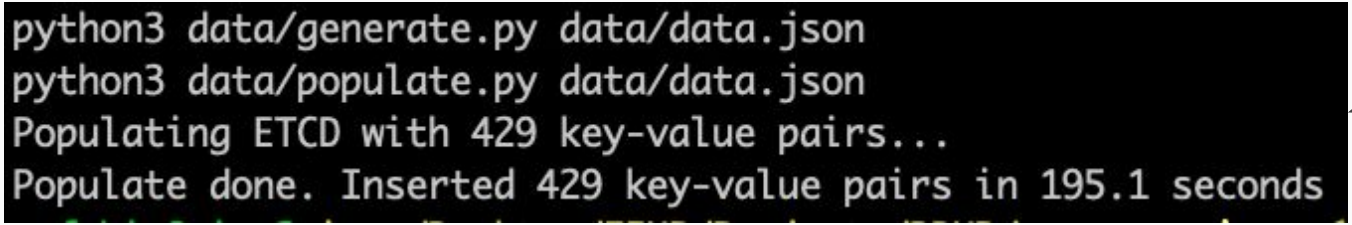
Prototype TickETCD - Demo

Prototype TickETCD - Limitations (1/5)

Redundancy, redundancy, redundancy...



```
$ cat configurations.json
{
    "NUM_USERS": 10,
    "NUM_EVENTS": 10,
    "TICKET_TYPES": ["pink", "yellow", "green", "red"],
    "NODES": 5
    ...
}
```

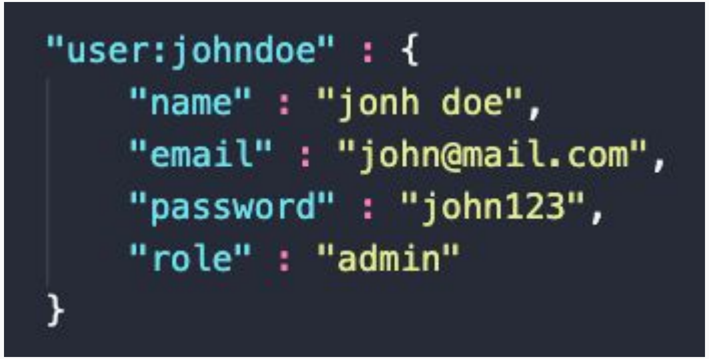


```
python3 data/generate.py data/data.json
python3 data/populate.py data/data.json
Populating ETCD with 429 key-value pairs...
Populate done. Inserted 429 key-value pairs in 195.1 seconds
```

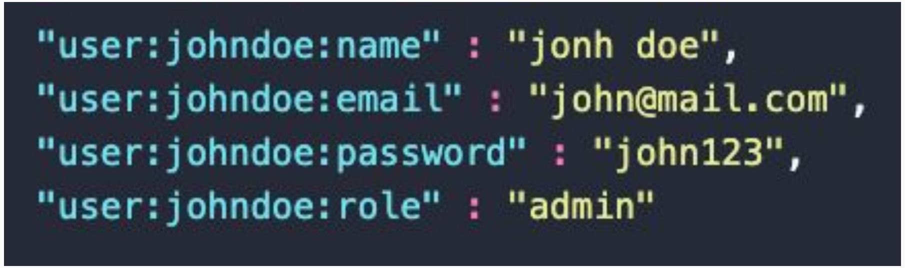
Just 10 users and 10 events... more than 400 key-value pairs and took more than 3 minutes in populate step!

Prototype TickETCD - Limitations (2/5)

Adding more redundancy to the system is good in etcd… or not!



```
"user:johndoe" : {
        "name" : "jonh doe",
        "email" : "john@mail.com",
        "password" : "john123",
        "role" : "admin"
}
```



```
"user:johndoe:name" : "jonh doe",
"user:johndoe:email" : "john@mail.com",
"user:johndoe:password" : "john123",
"user:johndoe:role" : "admin"
```

N attributes = 1 Query

+

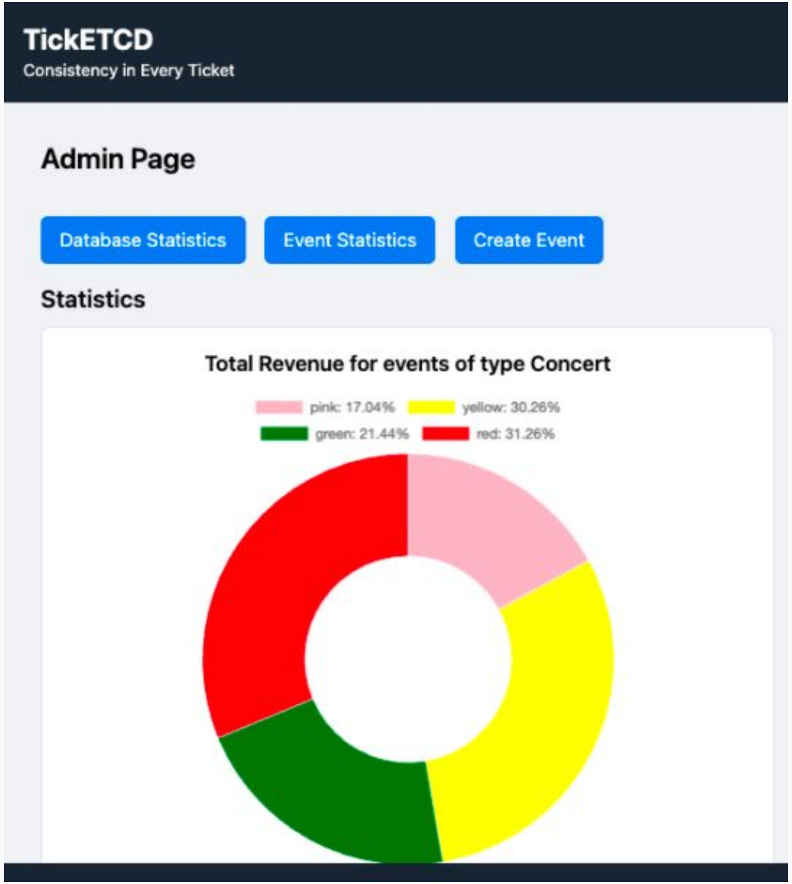
Object serialization/deserialization

N attributes = N Queries (disk!)

+

Not suitable for most TickETCD aggregates

Prototype TickETCD - Limitations (3/5)



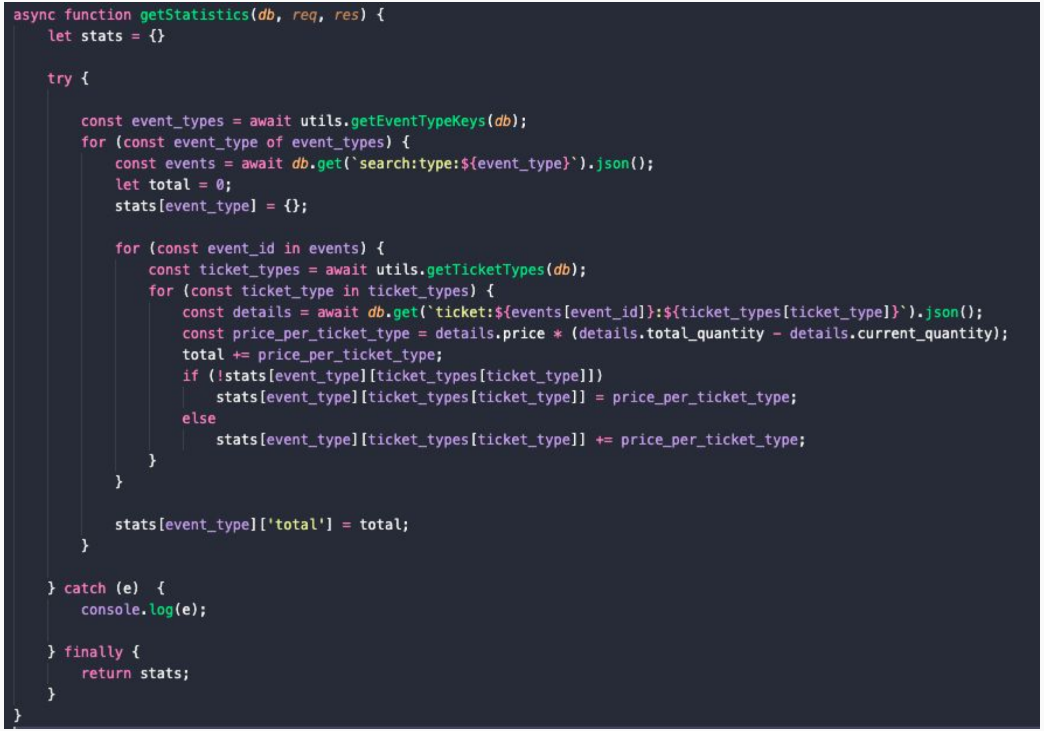
TickETCD

Consistency in Every Ticket

Admin Page

Statistics

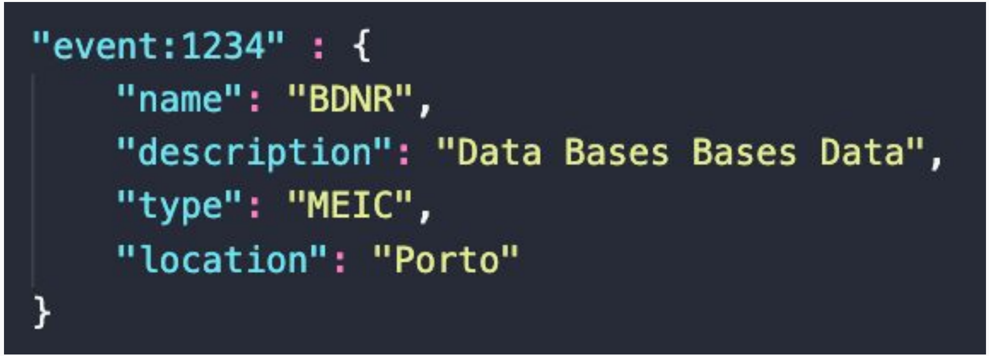
Event statistics is a nice feature, but…



```
async function getStatistics(db, req, res) {
    let stats = {}

    try {

        const event_types = await utils.getEventTypeKeys(db);
        for (const event_type of event_types) {
            const events = await db.get(`search:type:${event_type}`).json();
            let total = 0;
            stats[event_type] = {};

            for (const event_id in events) {
                const ticket_types = await utils.getTicketTypes(db);
                for (const ticket_type in ticket_types) {
                    const details = await db.get(`ticket:${events[event_id]}:${ticket_types[ticket_type]}`).json();
                    const price_per_ticket_type = details.price * (details.total_quantity - details.current_quantity);
                    total += price_per_ticket_type;
                    if (!stats[event_type][ticket_types[ticket_type]])
                        stats[event_type][ticket_types[ticket_type]] = price_per_ticket_type;
                    else
                        stats[event_type][ticket_types[ticket_type]] += price_per_ticket_type;
                }
            }

            stats[event_type]['total'] = total;
        }

    } catch (e)  {
        console.log(e);

    } finally {
        return stats;
    }
}
```

Prototype TickETCD - Limitations (4/5)

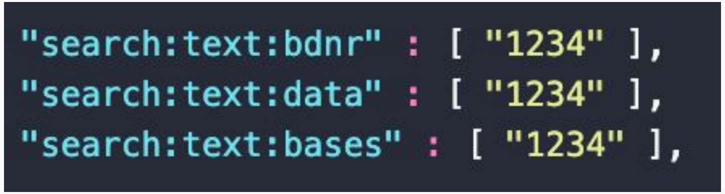
Searching by event attributes requires a **lot of external processing**…



```
"event:1234" : {
    "name": "BDNR",
    "description": "Data Bases Bases Data",
    "type": "MEIC",
    "location": "Porto"
}
```

“BDNR Data Bases Bases Data” =

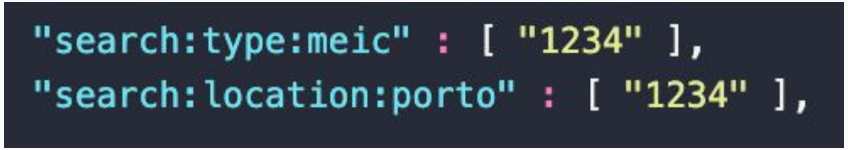
[‘bdnr’, ‘data’, ‘bases’]



```
"search:text:bdnr" : [ "1234" ],
"search:text:data" : [ "1234" ],
"search:text:bases" : [ "1234" ],
```

type = ‘MEIC’ => ‘meic’

location = ‘Porto’ => ‘porto’



```
"search:type:meic" : [ "1234" ],
"search:location:porto" : [ "1234" ],
```

… and makes **updating event attributes unfeasible!**

Prototype TickETCD - Limitations (5/5)

Let’s take a look at the computation flow for purchasing tickets of type **X** and **Y** for event **bdnr** by user **jonhdoe**...

1. Insert an entry in the array **purchase:jonhdoe:bdnr**
2. Update **ticket:bdnr:x** ticket current quantity
3. Update **ticket:bdnr:y** ticket current quantity
4. Update the global ticket quantity in **event:bdnr**

Manipulates 4 aggregates sequentially and **not atomically**!

↓

Transactions don’t work for multiple aggregates at the same time!

Conclusions

- Exploration with more detail of one of the paradigms of non-relational databases, key-value
- Opportunity to study a new technology, etcd, understand its specificities
- The implementation of the prototype made it possible to understand this approach and to apply features such as the scalability and consistency that these technology offer