MINI-COURSE

Modular Monoliths: CQRS With MongoDB

MJ Tech

Intro

Modular monolith benefits.

Data isolation and polyglot persistence.

Event-driven architecture, CQRS.

Document databases, MongoDB.

Modular Monolith

MJ Tech

Modular Monolith Benefits

- High cohesion
- Loose coupling
- Simplified deployment
- Improved performance
- Lower operational complexity
- Easier transition to microservices

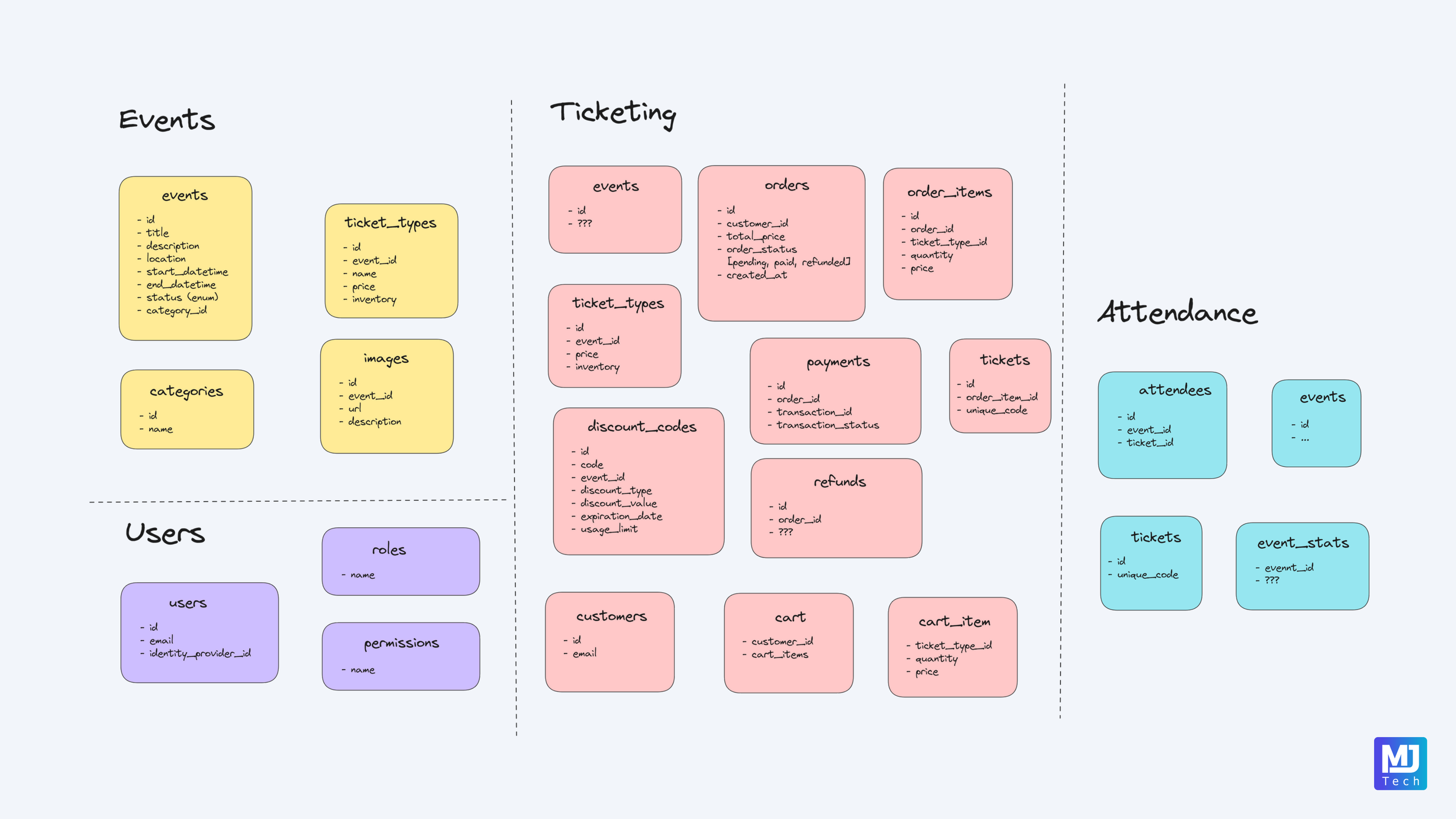
How do we define modules?

Defining Module Boundaries

Domain-Driven Design.

Bounded contexts.

Event Storming.


Events
events
- id
- title
- description
- location
- start_datetime
- end_datetime
- status (enum)
- category_id
ticket_types
- id
- event_id
- name
- price
- inventory
images
- id
- event_id
- url
- description
categories
- id
- name
Users
roles
- name
users
- id
- email
- identity_provider_id
permissions
- name
Ticketing
events
- id
- ???
orders
- id
- customer_id
- total_price
- order_status
[pending, paid, refunded]
- created_at
order_items
- id
- order_id
- ticket_type_id
- quantity
- price
ticket_types
- id
- event_id
- price
- inventory
payments
- id
- order_id
- transaction_id
- transaction_status
tickets
- id
- order_item_id
- unique_code
discount_codes
- id
- code
- event_id
- discount_type
- discount_value
- expiration_date
- usage_limit
refunds
- id
- order_id
- ???
customers
- id
- email
cart
- customer_id
- cart_items
cart_item
- ticket_type_id
- quantity
- price
Attendance
attendees
- id
- event_id
- ticket_id
events
- id
- ...
tickets
- id
- unique_code
event_stats
- evennt_id
- ???

Modules shouldn’t be coupled at the database level.

Module Data Isolation

There are four levels of data isolation:

- Level 1 - Separate tables
- Level 2 - Separate schemas
- Level 3 - Separate databases
- Level 4 - Different databases

Module Data Isolation

There are four levels of data isolation:

- Level 1 - Separate tables
- Level 2 - Separate schemas
- Level 3 - Separate databases
- Level 4 - Different databases

Module Data Isolation

There are four levels of data isolation:

- Level 1 - Separate tables
- **Level 2 - Separate schemas**
- Level 3 - Separate databases
- Level 4 - Different databases

Module Data Isolation

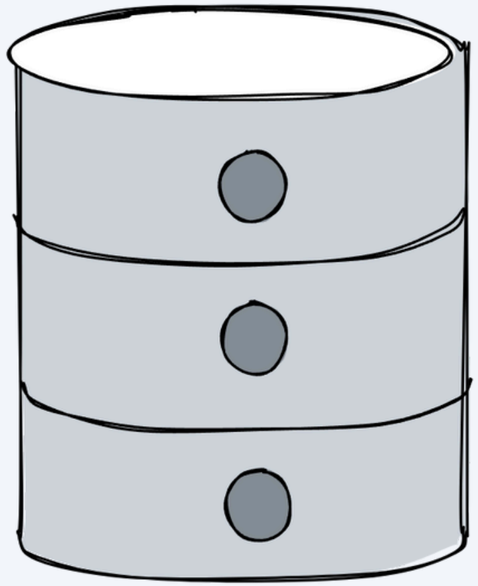
There are four levels of data isolation:

- Level 1 - Separate tables
- Level 2 - Separate schemas
- **Level 3 - Separate databases**
- Level 4 - Different databases

Module Data Isolation

There are four levels of data isolation:

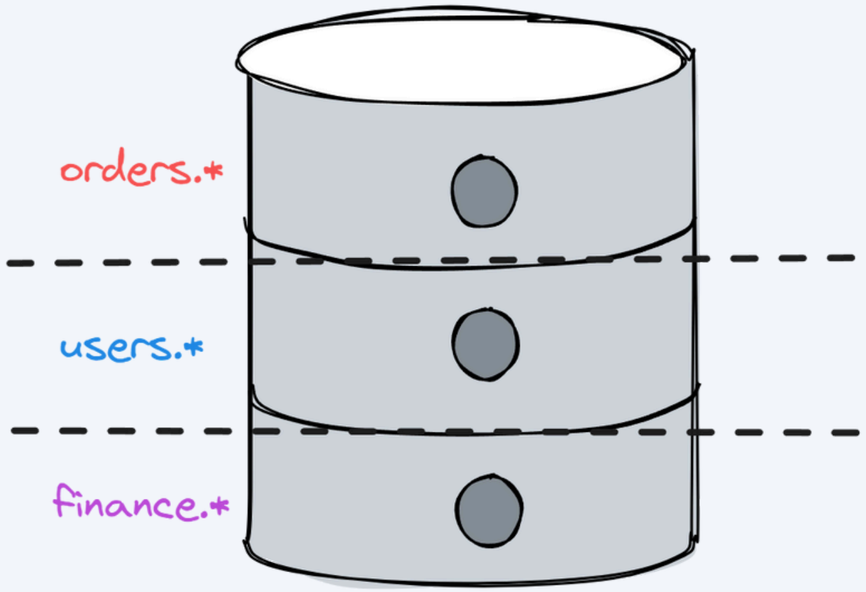
- Level 1 - Separate tables
- Level 2 - Separate schemas
- Level 3 - Separate databases
- Level 4 - Different databases



Relational DB

Separate Table

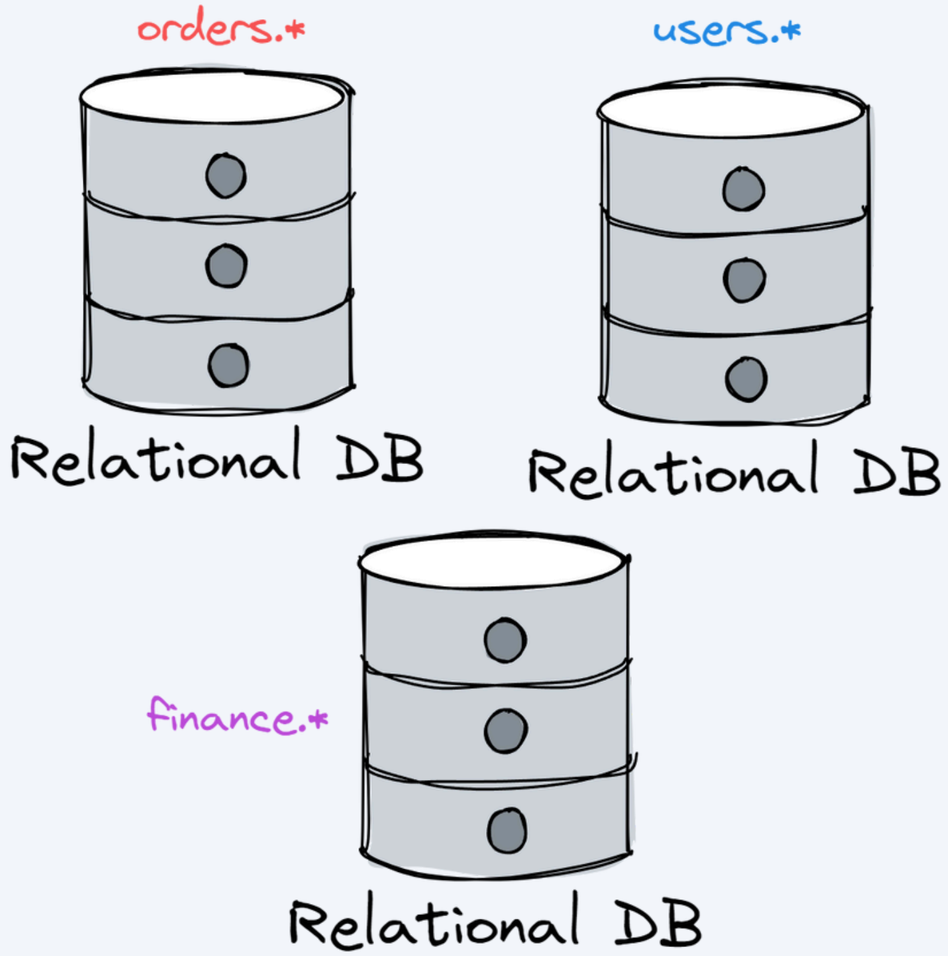
- Same schema
- Same database
- Same DB type



Relational DB

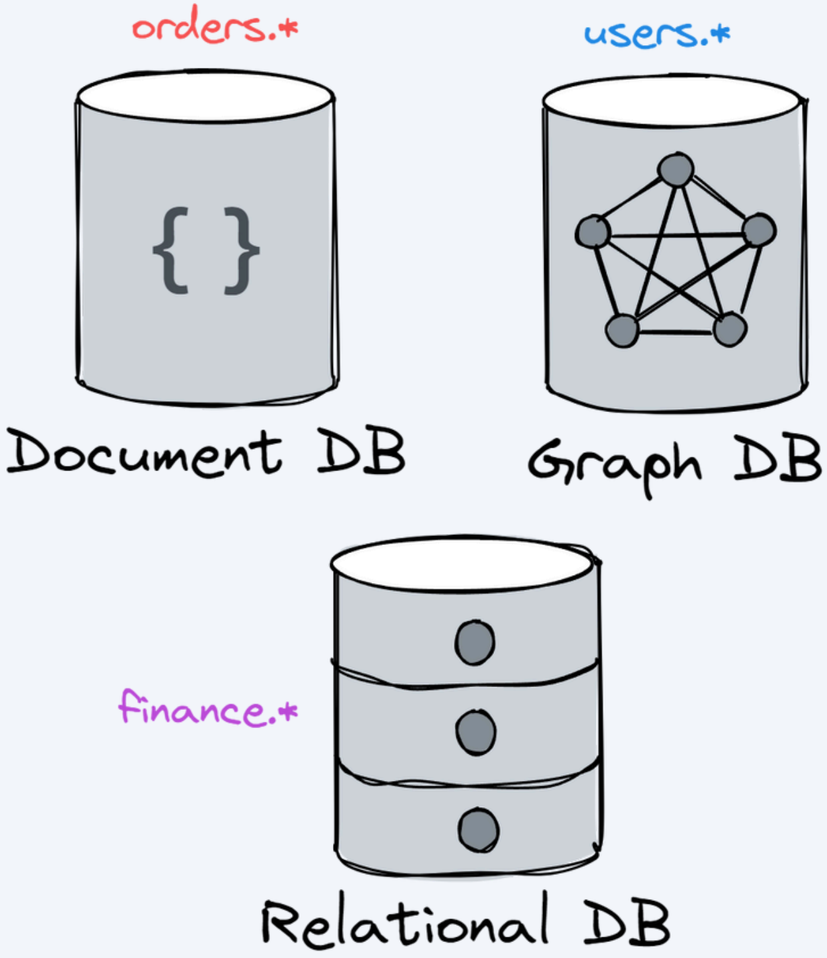
Separate Schema

- Different schema
- Same database
- Same DB type



Separate Database

- Different schema
- Different database
- Same DB type



Different Persistence

- Different schema
- Different database
- Different DB type

Events

events
- id
- title
- description
- location
- start_datetime
- end_datetime
- status (enum)
- category_id

ticket_types
- id
- event_id
- name
- price
- inventory

images
- id
- event_id
- url
- description

categories
- id
- name

Users

users
- id
- email
- identity_provider_id

roles
- name

permissions
- name

Ticketing

events
- id
- ???

orders
- id
- customer_id
- total_price
- order_status
  [pending, paid, refunded]
- created_at

order_items
- id
- order_id
- ticket_type_id
- quantity
- price

ticket_types
- id
- event_id
- price
- inventory

payments
- id
- order_id
- transaction_id
- transaction_status

tickets
- id
- order_item_id
- unique_code

discount_codes
- id
- code
- event_id
- discount_type
- discount_value
- expiration_date
- usage_limit

refunds
- id
- order_id
- ???

customers
- id
- email

cart
- customer_id
- cart_items

cart_item
- ticket_type_id
- quantity
- price

Attendance

attendees
- id
- event_id
- ticket_id

events
- id
- ...

tickets
- id
- unique_code

event_stats
- evennt_id
- ???

MJ Tech

Events

events
- id
- title
- description
- location
- start_datetime
- end_datetime
- status (enum)
- category_id

ticket_types
- id
- event_id
- name
- price
- inventory

images
- id
- event_id
- url
- description

categories
- id
- name

Users

users
- id
- email
- identity_provider_id

roles
- name

permissions
- name

Ticketing

events
- id
- ???

orders
- id
- customer_id
- total_price
- order_status
  [pending, paid, refunded]
- created_at

order_items
- id
- order_id
- ticket_type_id
- quantity
- price

ticket_types
- id
- event_id
- price
- inventory

payments
- id
- order_id
- transaction_id
- transaction_status

tickets
- id
- order_item_id
- unique_code

discount_codes
- id
- code
- event_id
- discount_type
- discount_value
- expiration_date
- usage_limit

refunds
- id
- order_id
- ???

customers
- id
- email

cart
- customer_id
- cart_items

cart_item
- ticket_type_id
- quantity
- price

Attendance

attendees
- id
- event_id
- ticket_id

events
- id
- ...

tickets
- id
- unique_code

event_stats
- evennt_id
- ???

MJ Tech

Events

events
- id
- title
- description
- location
- start_datetime
- end_datetime
- status (enum)
- category_id

ticket_types
- id
- event_id
- name
- price
- inventory

images
- id
- event_id
- url
- description

categories
- id
- name

Users

users
- id
- email
- identity_provider_id

roles
- name

permissions
- name

Ticketing

events
- id
- ???

orders
- id
- customer_id
- total_price
- order_status
  [pending, paid, refunded]
- created_at

order_items
- id
- order_id
- ticket_type_id
- quantity
- price

ticket_types
- id
- event_id
- price
- inventory

payments
- id
- order_id
- transaction_id
- transaction_status

tickets
- id
- order_item_id
- unique_code

discount_codes
- id
- code
- event_id
- discount_type
- discount_value
- expiration_date
- usage_limit

refunds
- id
- order_id
- ???

customers
- id
- email

cart
- customer_id
- cart_items

cart_item
- ticket_type_id
- quantity
- price

Attendance

attendees
- id
- event_id
- ticket_id

events
- id
- ...

tickets
- id
- unique_code

event_stats
- evennt_id
- ???

Events

events
- id
- title
- description
- location
- start_datetime
- end_datetime
- status (enum)
- category_id

ticket_types
- id
- event_id
- name
- price
- inventory

images
- id
- event_id
- url
- description

categories
- id
- name

Users

users
- id
- email
- identity_provider_id

roles
- name

permissions
- name

Ticketing

events
- id
- ???

orders
- id
- customer_id
- total_price
- order_status
  [pending, paid, refunded]
- created_at

order_items
- id
- order_id
- ticket_type_id
- quantity
- price

ticket_types
- id
- event_id
- price
- inventory

payments
- id
- order_id
- transaction_id
- transaction_status

tickets
- id
- order_item_id
- unique_code

discount_codes
- id
- code
- event_id
- discount_type
- discount_value
- expiration_date
- usage_limit

refunds
- id
- order_id
- ???

customers
- id
- email

cart
- customer_id
- cart_items

cart_item
- ticket_type_id
- quantity
- price

Attendance

attendees
- id
- event_id
- ticket_id

events
- id
- ...

tickets
- id
- unique_code

event_stats
- evennt_id
- ???

MJ Tech

Events

events
- id
- title
- description
- location
- start_datetime
- end_datetime
- status (enum)
- category_id

ticket_types
- id
- event_id
- name
- price
- inventory

images
- id
- event_id
- url
- description

categories
- id
- name

Users

users
- id
- email
- identity_provider_id

roles
- name

permissions
- name

Ticketing

events
- id
- ???

orders
- id
- customer_id
- total_price
- order_status
  [pending, paid, refunded]
- created_at

order_items
- id
- order_id
- ticket_type_id
- quantity
- price

ticket_types
- id
- event_id
- price
- inventory

payments
- id
- order_id
- transaction_id
- transaction_status

tickets
- id
- order_item_id
- unique_code

discount_codes
- id
- code
- event_id
- discount_type
- discount_value
- expiration_date
- usage_limit

refunds
- id
- order_id
- ???

customers
- id
- email

cart
- customer_id
- cart_items

cart_item
- ticket_type_id
- quantity
- price

Attendance

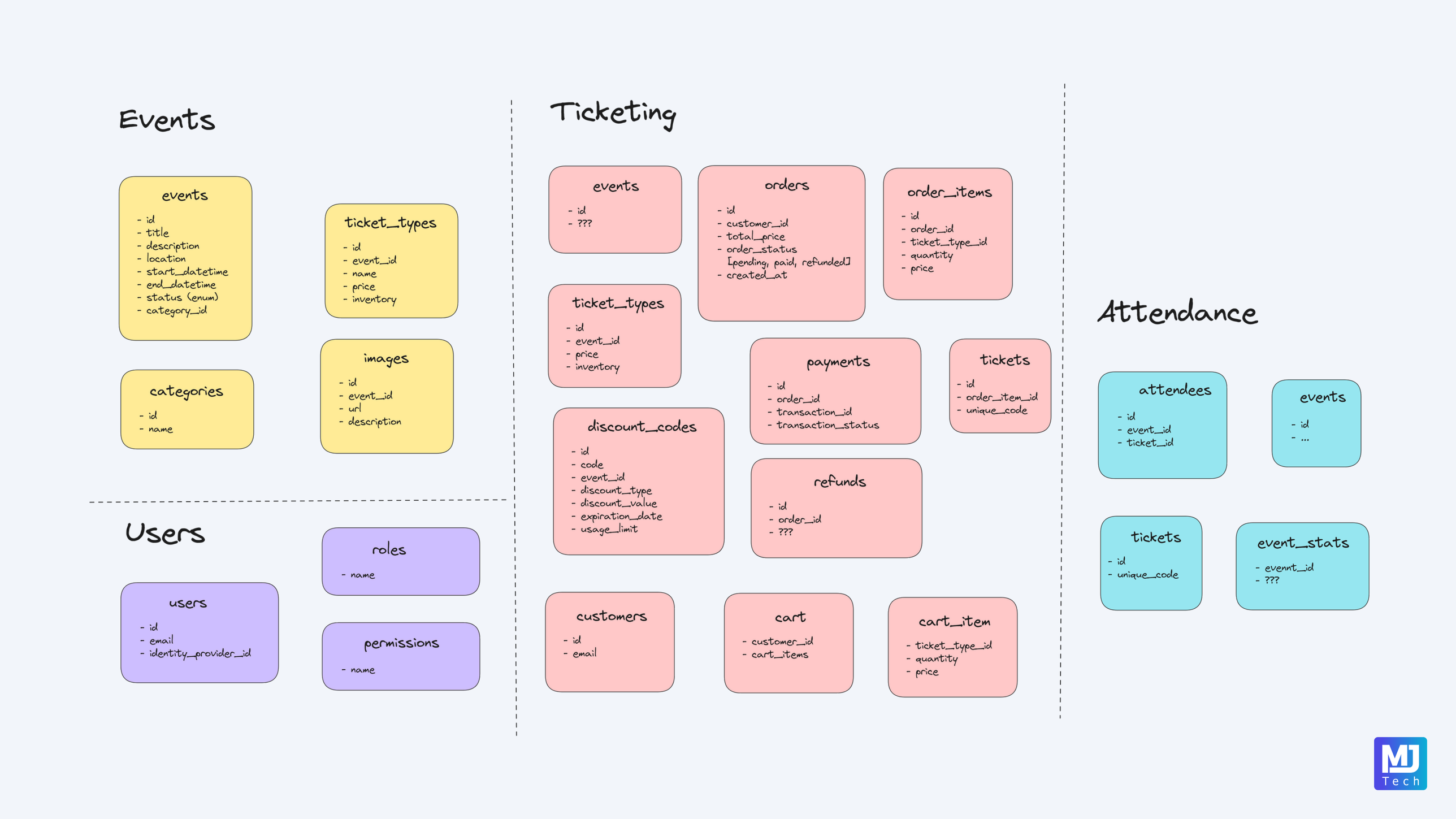
attendees
- id
- event_id
- ticket_id

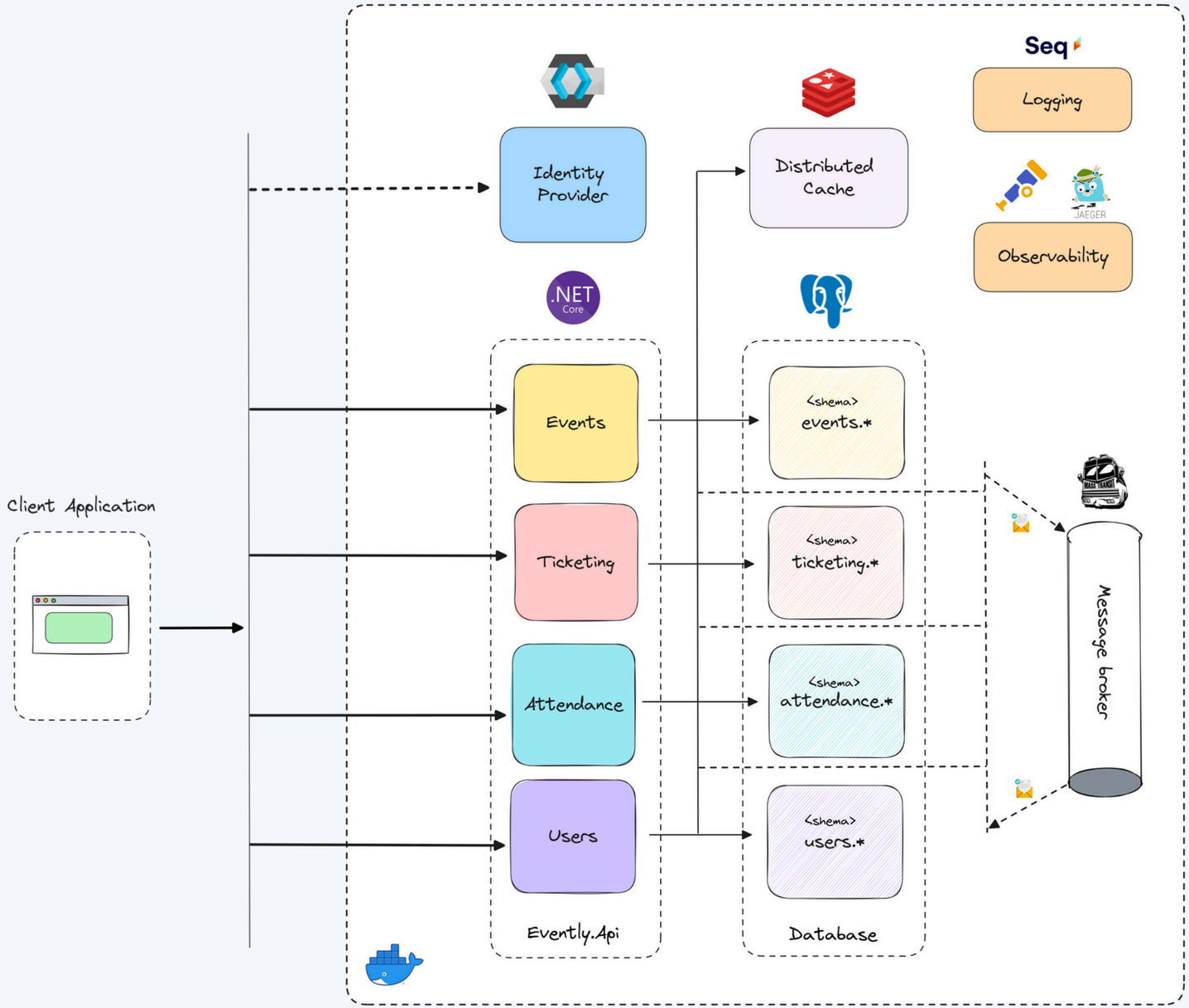
events
- id
- ...

tickets
- id
- unique_code

event_stats
- evennt_id
- ???

MJ Tech


Events
events
- id
- title
- description
- location
- start_datetime
- end_datetime
- status (enum)
- category_id
ticket_types
- id
- event_id
- name
- price
- inventory
images
- id
- event_id
- url
- description
categories
- id
- name
Users
roles
- name
users
- id
- email
- identity_provider_id
permissions
- name
Ticketing
events
- id
- ???
orders
- id
- customer_id
- total_price
- order_status
[pending, paid, refunded]
- created_at
order_items
- id
- order_id
- ticket_type_id
- quantity
- price
ticket_types
- id
- event_id
- price
- inventory
payments
- id
- order_id
- transaction_id
- transaction_status
tickets
- id
- order_item_id
- unique_code
discount_codes
- id
- code
- event_id
- discount_type
- discount_value
- expiration_date
- usage_limit
refunds
- id
- order_id
- ???
customers
- id
- email
cart
- customer_id
- cart_items
cart_item
- ticket_type_id
- quantity
- price
Attendance
attendees
- id
- event_id
- ticket_id
events
- id
- ...
tickets
- id
- unique_code
event_stats
- evennt_id
- ???


Seq
Logging
Identity Provider
Distributed Cache
JAEGER
Observability
.NET Core
Client Application
Events
Ticketing
Attendance
Users
<shema> events.*
<shema> ticketing.*
<shema> attendance.*
<shema> users.*
Message broker
Evently.Api
Database

Event-driven Architecture

What does it mean to be event-driven?

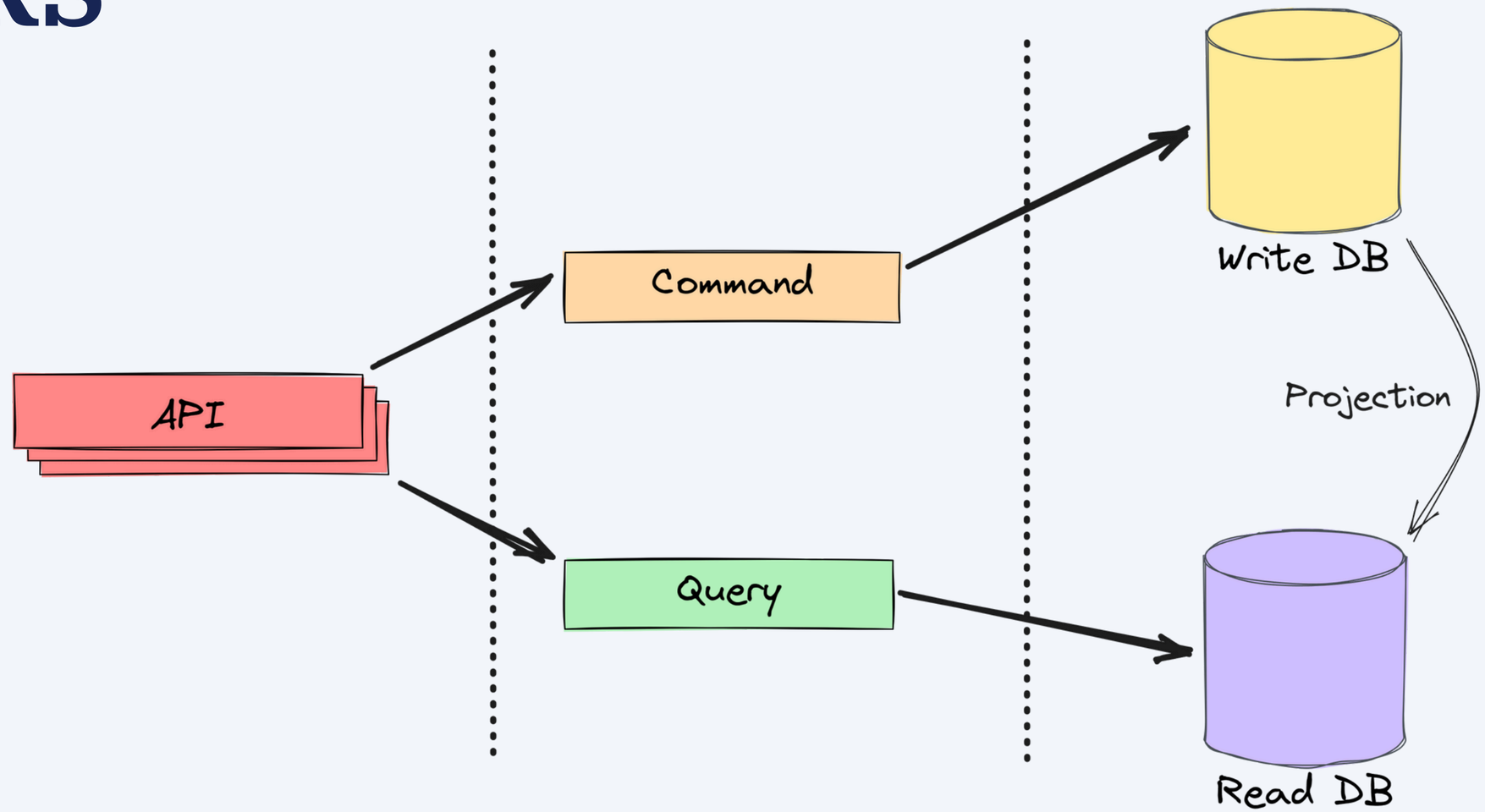
Driven by business events.

Patterns of EDA

You are *event-driven* if you are using at least one of these patterns.

- Event sourcing
- Event notifications
- Event-carried state transfer
- CQRS

CQRS



Embrace eventual consistency

Document Databases

A document database is a database that stores information in documents.

Flexible database schema.

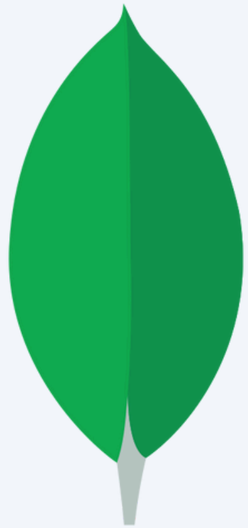
Horizontal scalability.

```
{
  "_id": "5cf0029caff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  }
  "hobbies": ["surfing", "coding"]
}
```

Source: https://www.mongodb.com/document-databases

MongoDB

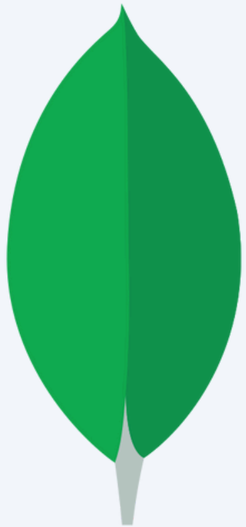
NoSQL document database.



MongoDB

NoSQL document database.

Documents are stored using JSONB.

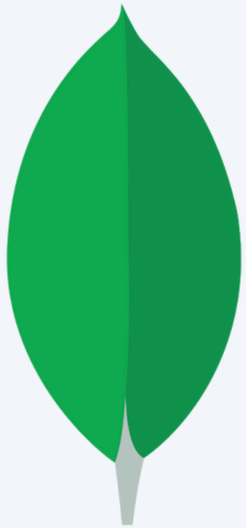


MongoDB

NoSQL document database.

Documents are stored using JSONB.

Collections instead of tables.



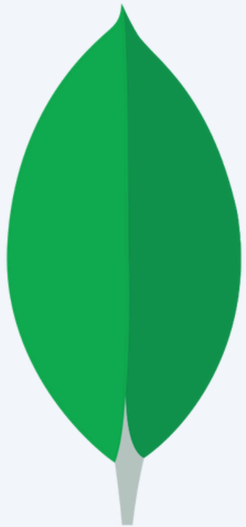
MongoDB

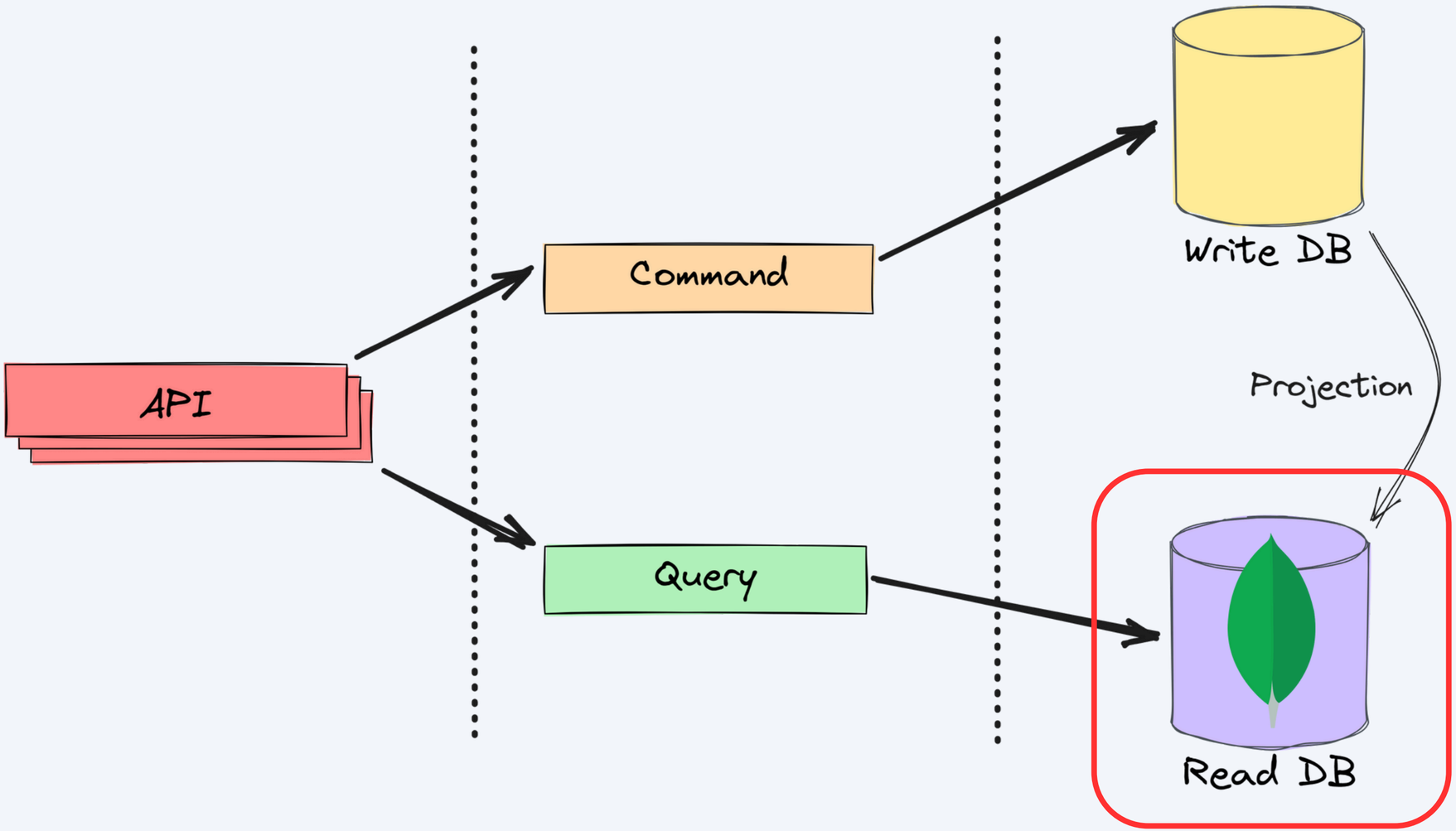
NoSQL document database.

Documents are stored using JSONB.

Collections instead of tables.

Schemaless.




API
Command
Query
Write DB
Projection
Read DB

MINI-COURSE

Modular Monoliths: CQRS With MongoDB

MJ Tech

Intro

Set up MongoDB using Docker.

OpenTelemetry, Health checks.

Use MongoDB to implement projections.

RECAP

Modular Monoliths: CQRS With MongoDB

MJ Tech