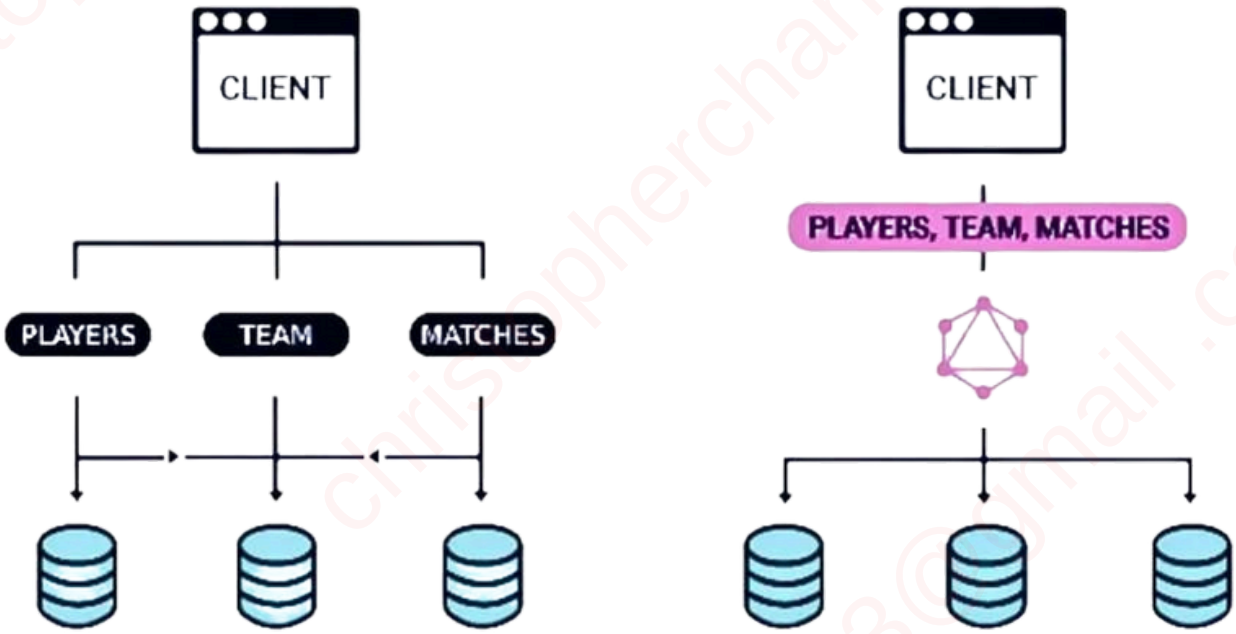
GraphQL vs REST API

(With Examples)



匠人学院®
JR ACADEMY

Data Fetching

REST: Uses fixed endpoints (/users, /posts) to retrieve data.

Request

```
GET /users/1
```

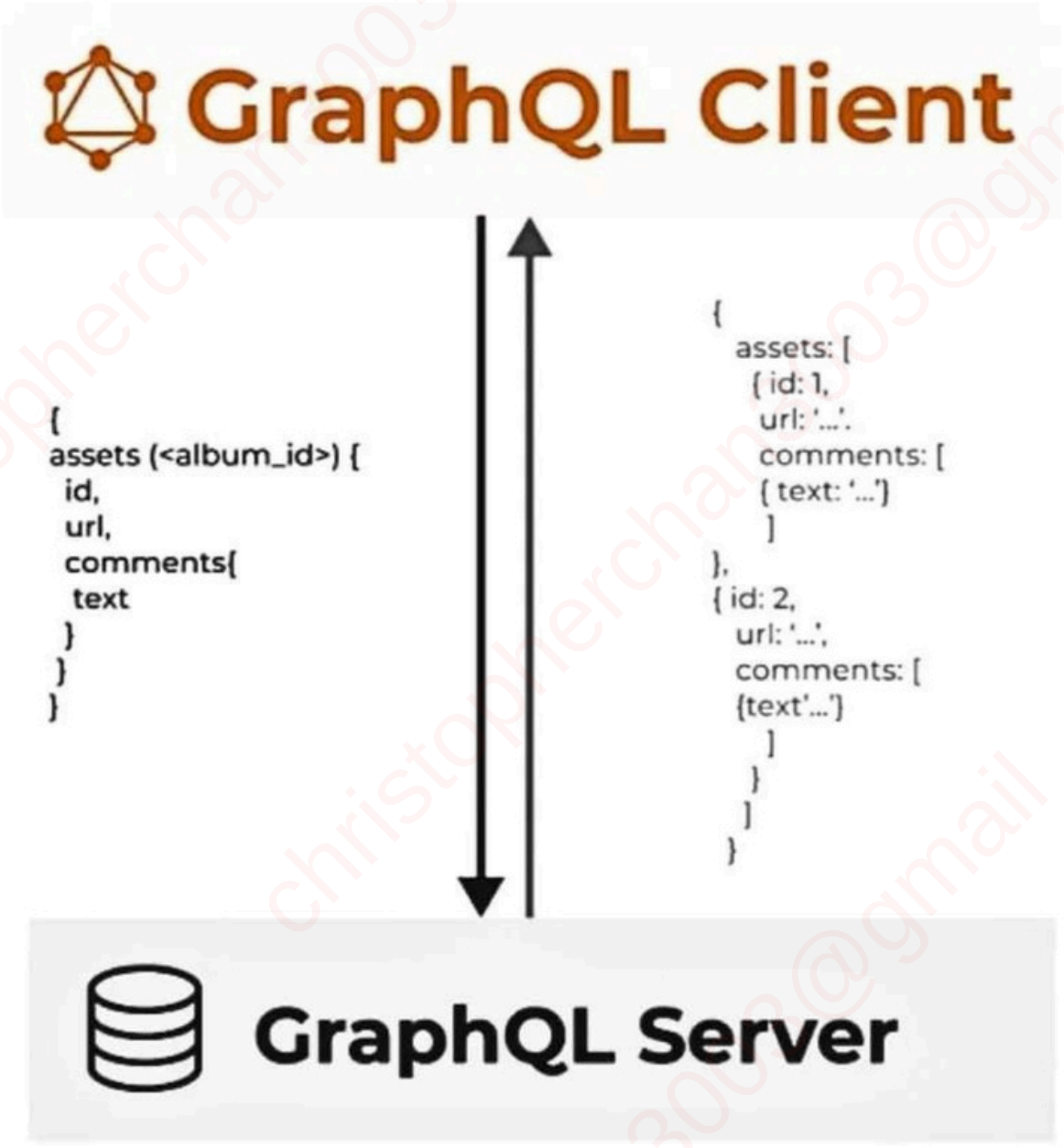
Response

```
{ "id": 1, "name": "John", "email": "john@email.com" }
```

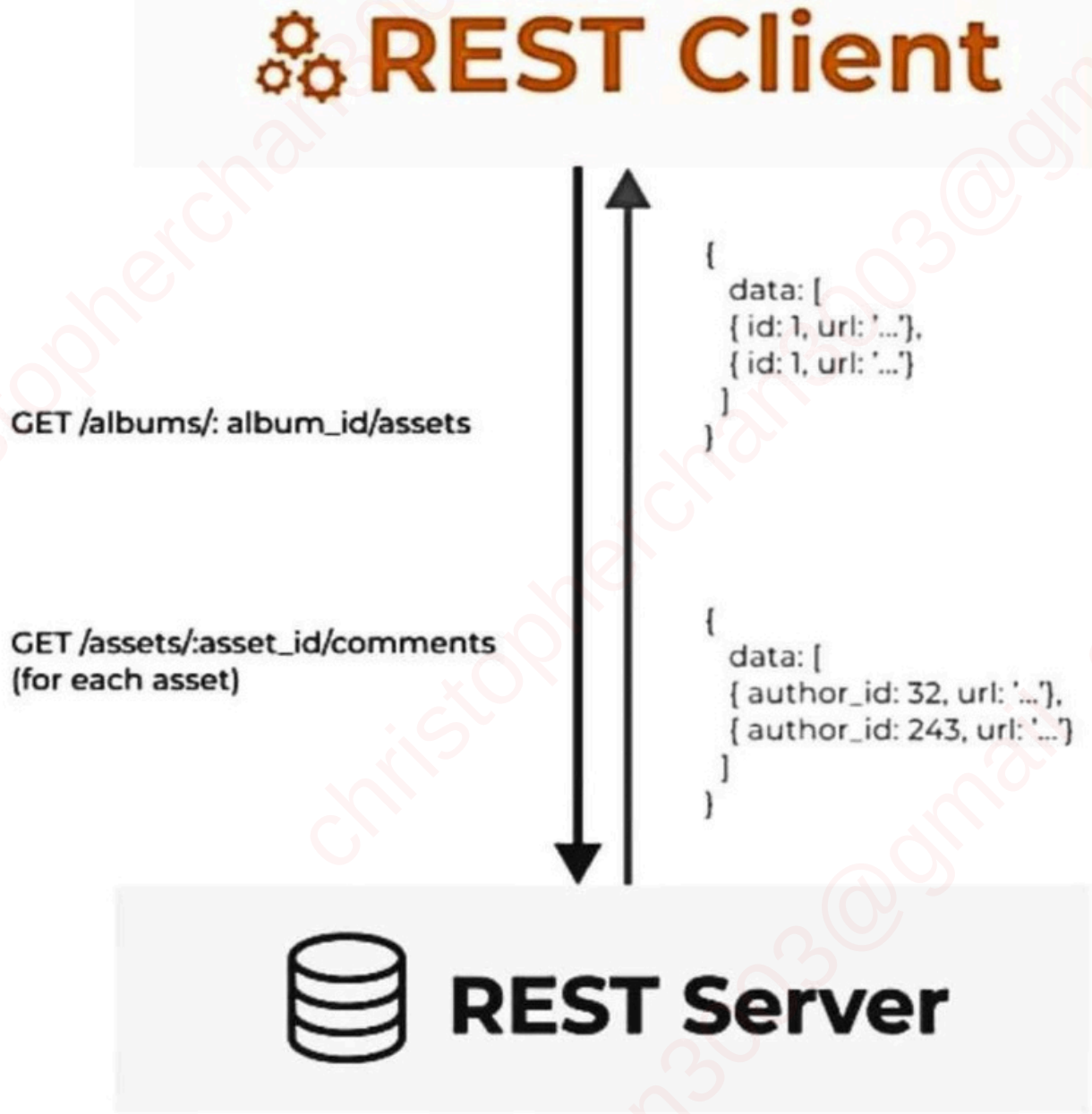
GraphQL: Allows querying specific data fields in a single request.

```
{
  user(id: 1) {
    name
    email
  }
}
```

匠人学院®
JR ACADEMY



匠人学院®
JR ACADEMY



匠人学院®
JR ACADEMY

Efficiency

REST: Often requires multiple requests for related data.

```
GET /users/1
GET /users/1/posts
```

GraphQL: Fetches all necessary data in a single request.

```
{
  user(id: 1) { name email posts { title content } }
}
```

Response Format

REST: Returns a predefined JSON structure.

```
{
  "id": 1,
  "name": "John",
  "email": "john@email.com",
  "age": 30,
  "posts": [
    { "id": 101, "title": "GraphQL Basics", "content": "..." }
  ]
}
```

GraphQL: Returns only the requested data & eliminates unnecessary fields, optimizing responses.

```
{
  "user": {
    "name": "John",
    "email": "john@email.com"
  }
}
```

匠人学院®
JR ACADEMY

Use Cases

REST API: Best for simple, cache-friendly APIs with fixed structures.

Use REST when: Public APIs & caching are needed.

GraphQL: Best for complex, dynamic apps needing flexible data fetching.

Use GraphQL when: Frontend needs customized queries.

匠人学院®
JR ACADEMY