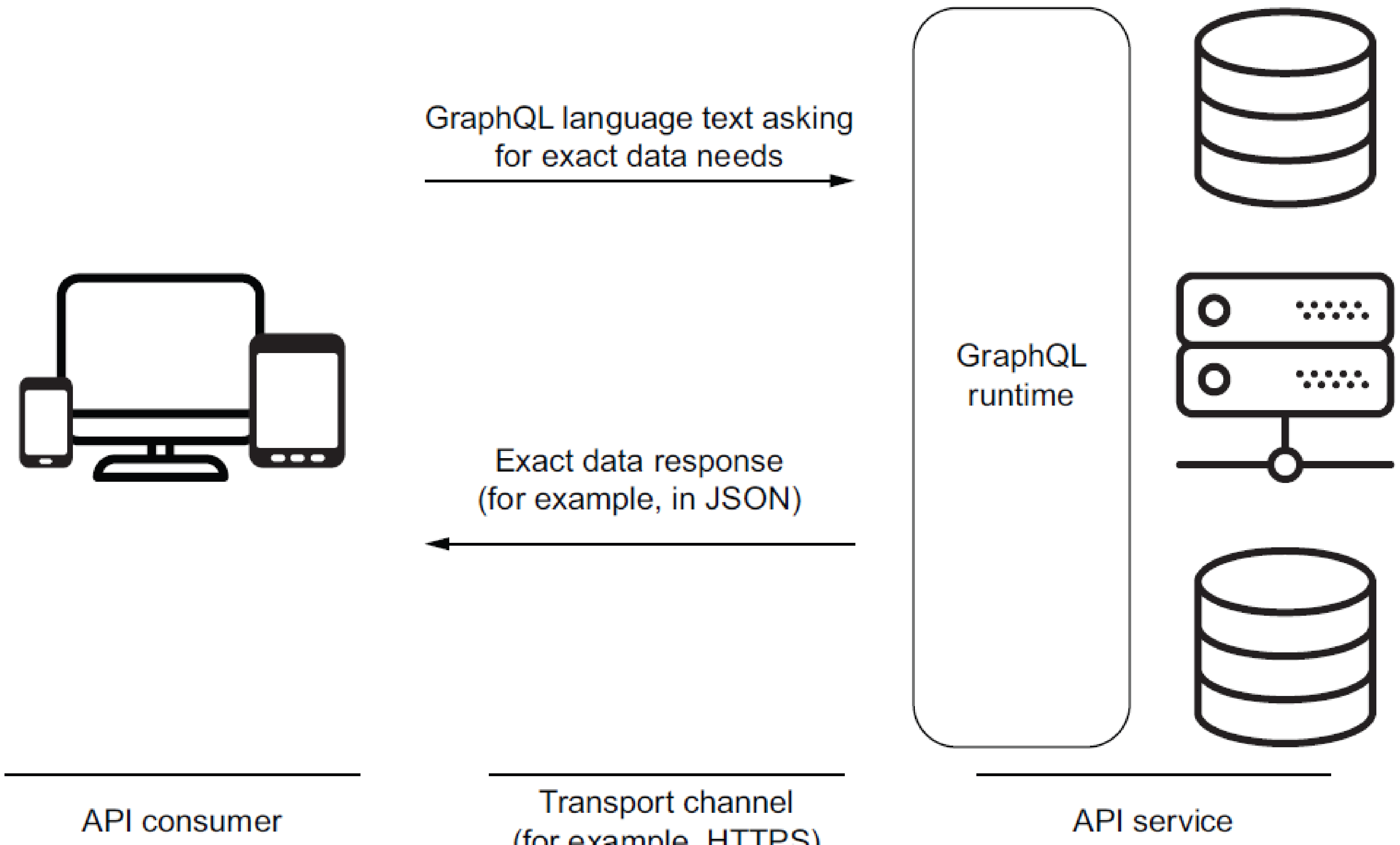
GraphQL

What we will cover

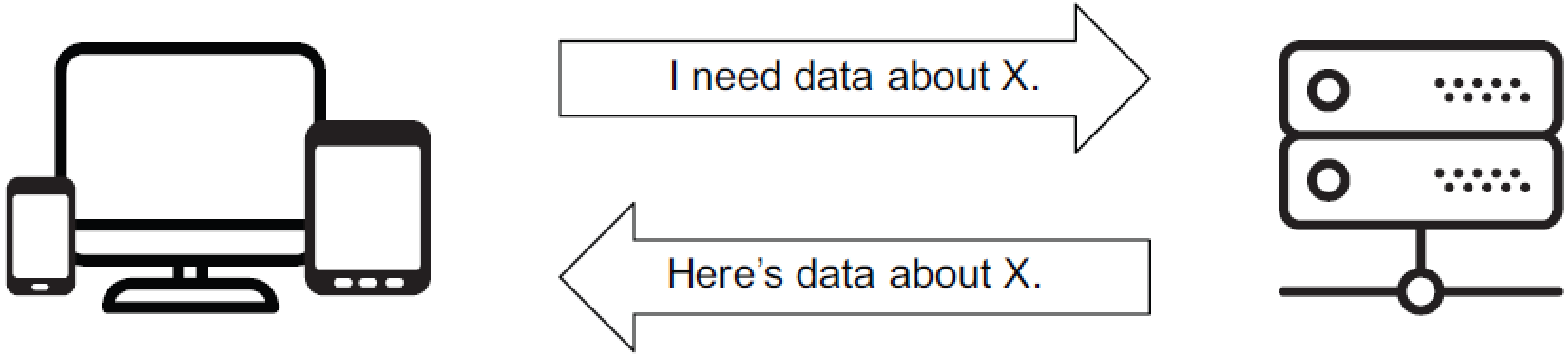
- Understanding GraphQL and the design conceptsbehind it
- How GraphQL differs from alternatives like REST APIs
- Understanding the language used by GraphQL clients and services
- Understanding the advantages and disadvantages of GraphQL


GraphQL language text asking for exact data needs
Exact data response (for example, in JSON)
GraphQL runtime
API consumer
Transport channel (for example, HTTPS)
API service

GraphQL and Libs

GraphQL is ***NOT*** specific to any libraries or frameworks.

The Big Picture



```
SELECT id, first_name, last_name, email, birth_date, hire_date
FROM employees
WHERE department = 'ENGINEERING'
```

```sql
INSERT INTO employees (first_name, last_name, email, birth_date, hire_date)
VALUES ('Jane', 'Doe', 'jane@doe.name', '01/01/1990', '01/01/2020')
```

```json
{
  "data": {
    "employee":{
      "id": 42,
      "name": "Jane Doe",
      "email": "jane@doe.name",
      "birthDate": "01/01/1990",
      "hireDate": "01/01/2020"
    }
  }
}
```

```
{
  "select": {
    "fields": ["name", "email", "birthDate", "hireDate"],
    "from": "employees",
    "where": {


        "id": {
          "equals": 42
        }
      }
    }
  }
```

```
{
  employee(id: 42) {
    name
    email
    birthDate
    hireDate
  }
}
```

GraphQL is a Spec

You can see the current version of this document by navigating to az.dev/graphql-spec

GraphQL is a Language

GraphQL Operations

- Queries represent READ operations.
- Mutations represent WRITE-then-READ operations.

You can think of mutations as queries that have side effects.

- GraphQL also supports a third request type called a subscription, used for real-time data monitoring requests.

John: "Hey Jane, how long does it take sunlight to reach planet Earth?"

Jane: "A bit over 8 minutes."

John: "How about the light from the moon?"

Jane: "A bit under 2 seconds."

```
{
  timeLightNeedsToTravel(toPlanet: "Earth") {
    fromTheSun: from(star: "Sun")
    fromTheMoon: from(moon: "Moon")
  }
}
```

GraphQL as a Service

Structure

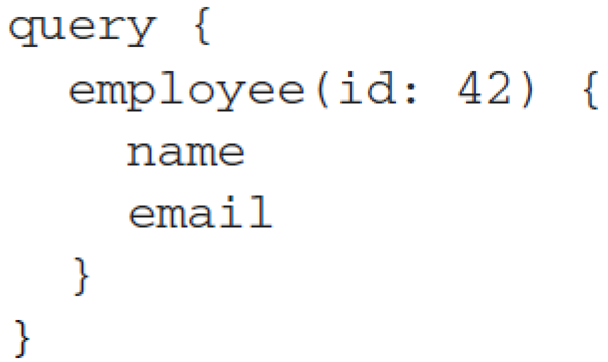
Behavior

Resolver Functions

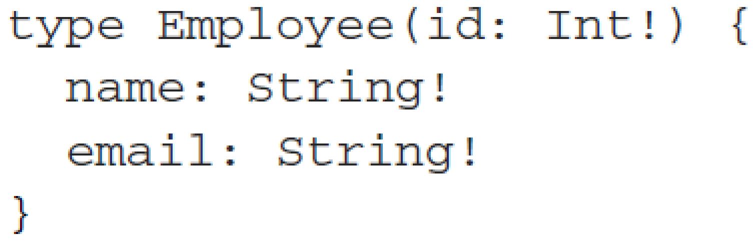
GraphQL as a Restaurant

```
order {
    steak(doneness: MEDIUMWELL)
}
```

Resolvers

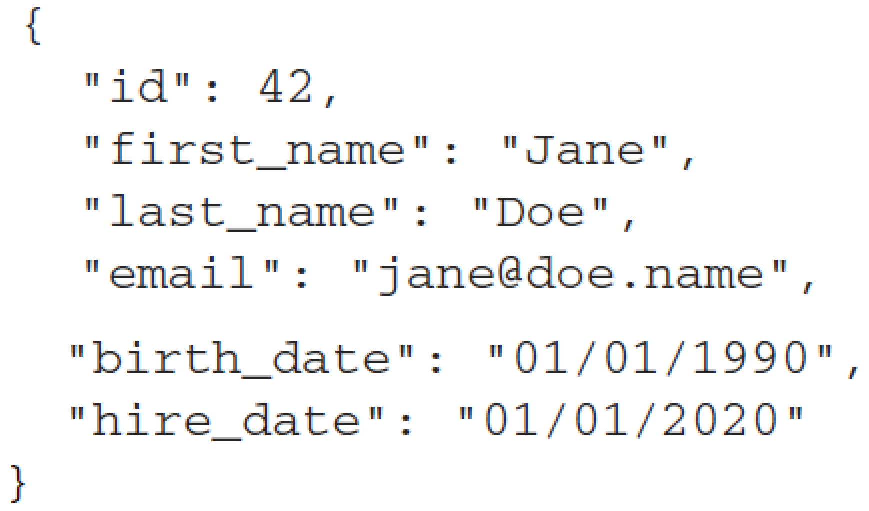


```
query {
  employee(id: 42) {
    name
    email
  }
}
```



```
type Employee(id: Int!) {
  name: String!
  email: String!
}
```

Response from the DB

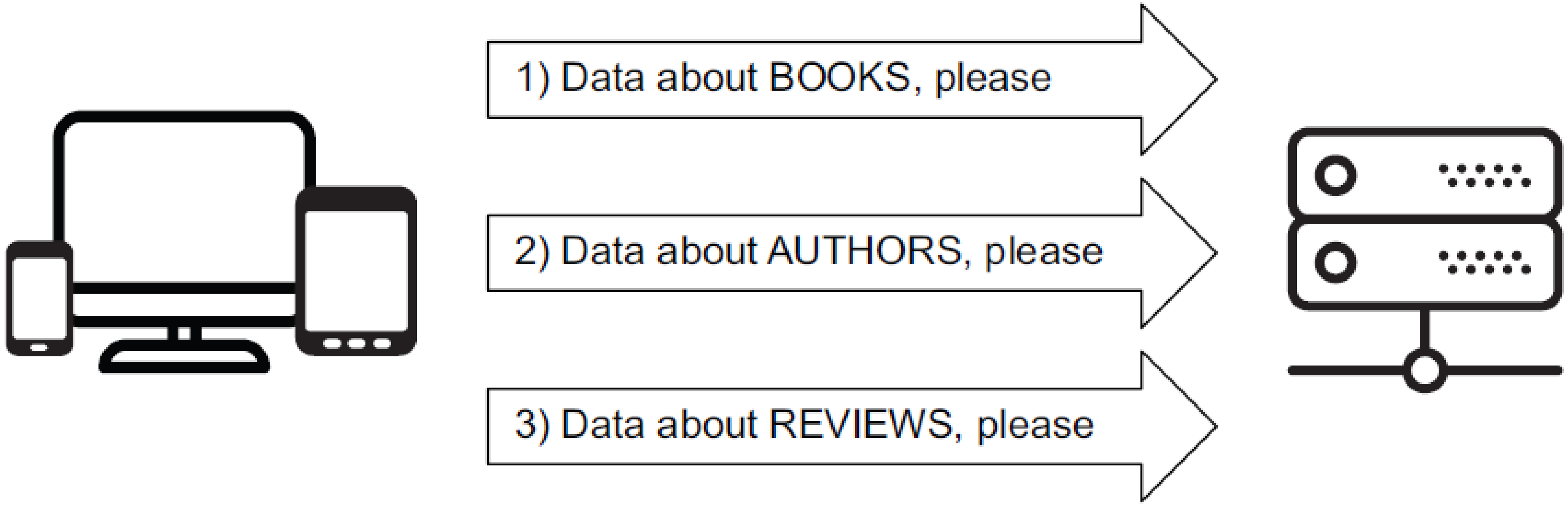


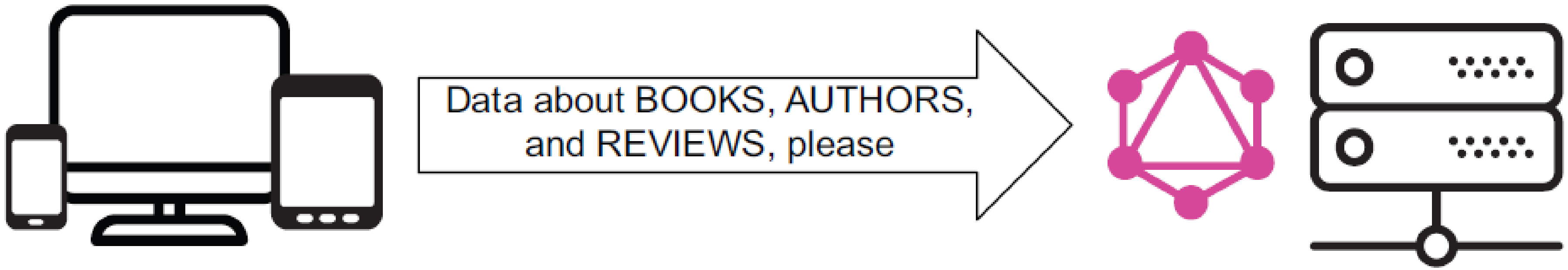
```
{
   "id": 42,
   "first_name": "Jane",
   "last_name": "Doe",
   "email": "jane@doe.name",
  "birth_date": "01/01/1990",
  "hire_date": "01/01/2020"
}
```

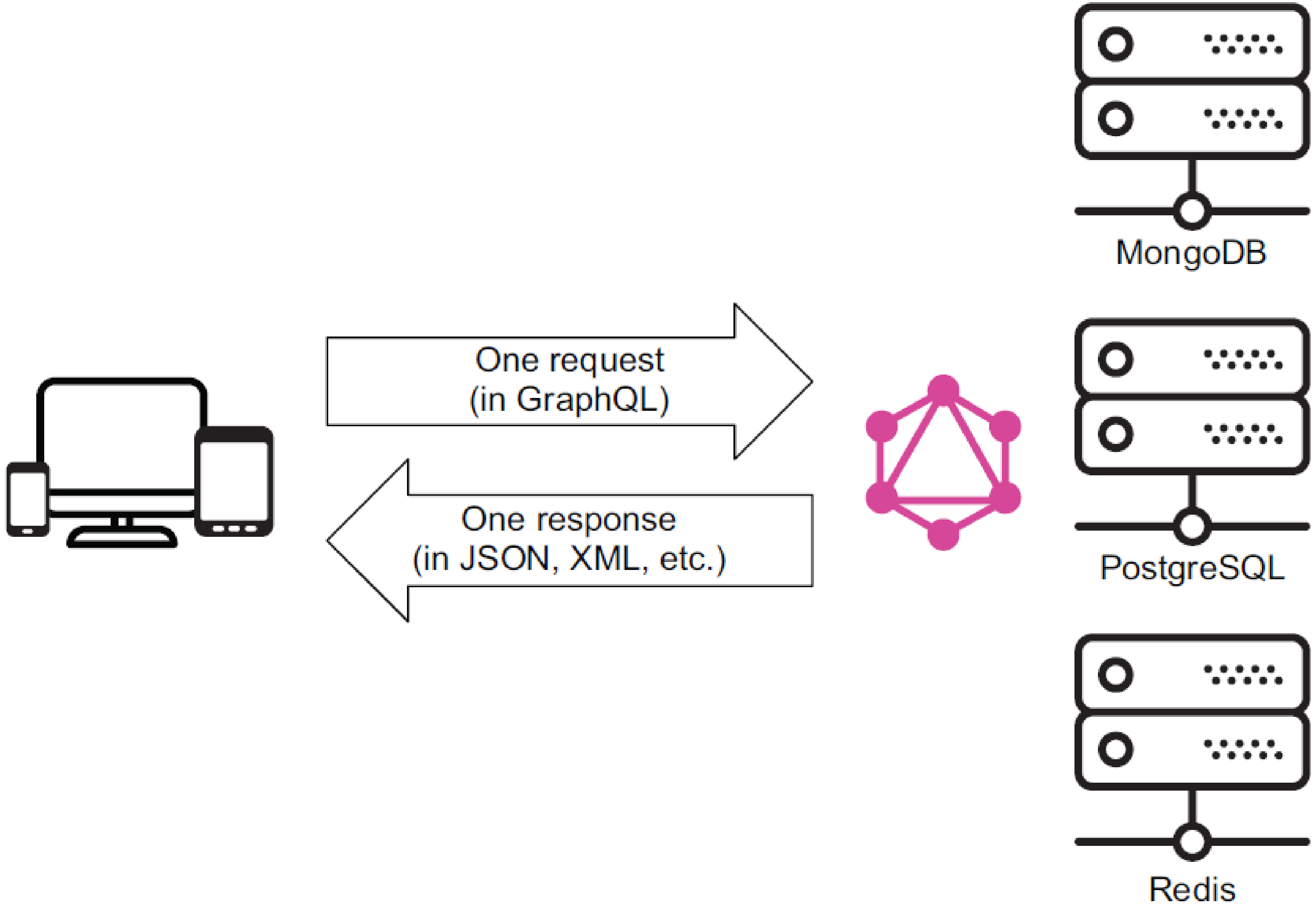
```
// Resolver functions
const name => (source) => `${source.first_name} ${source.last_name}`;
const email => (source) => source.email;
```

```
{
  data: {
    employee: {
      name: 'Jane Doe',
      email: 'jane@doe.name'
    }
  }
}
```

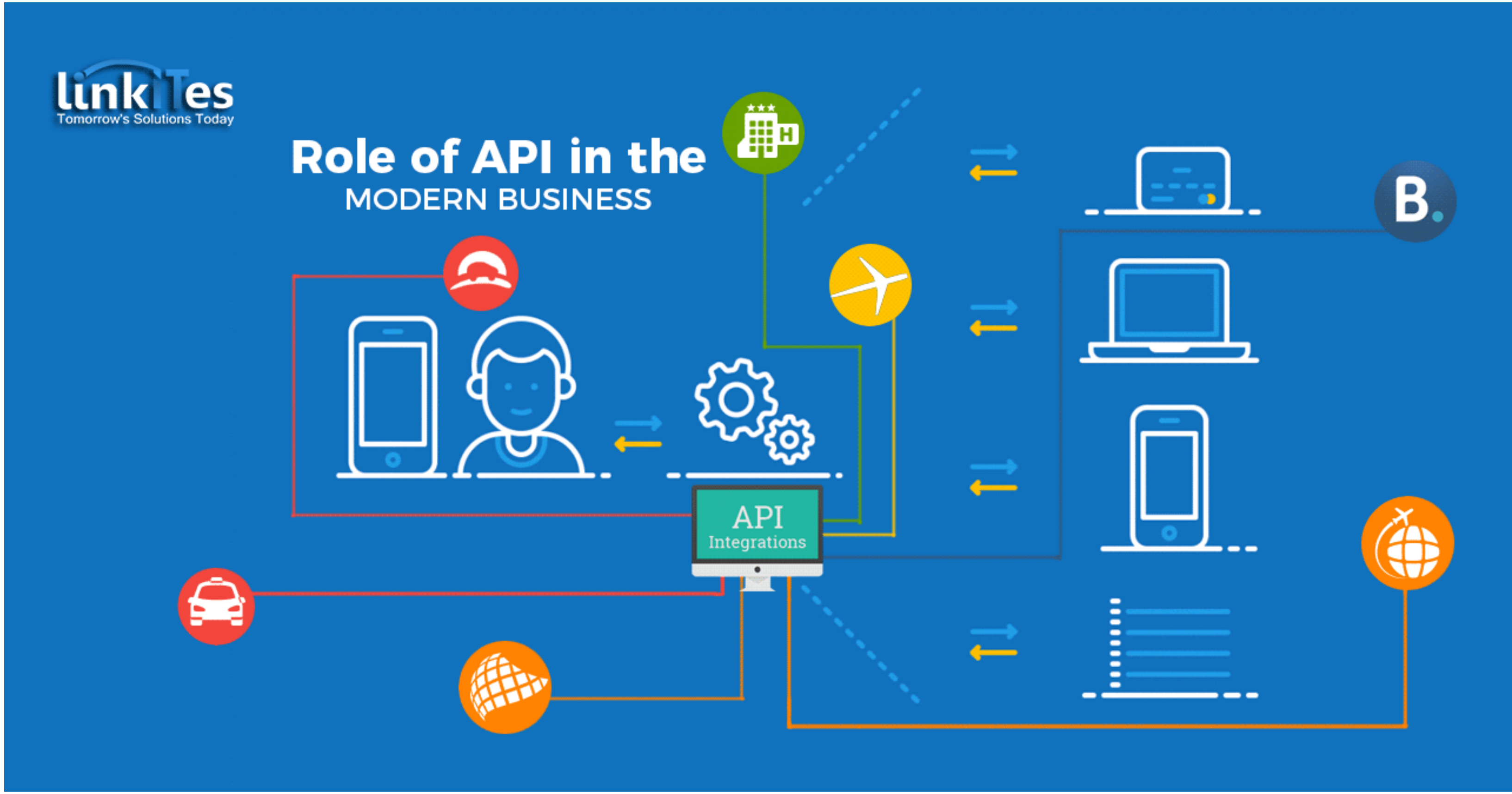
Why GraphQL


1) Data about BOOKS, please
2) Data about AUTHORS, please
3) Data about REVIEWS, please


Data about BOOKS, AUTHORS, and REVIEWS, please


One request
(in GraphQL)
One response
(in JSON, XML, etc.)
MongoDB
PostgreSQL
Redis

Rest APIs


linkITes
Tomorrow's Solutions Today
Role of API in the
MODERN BUSINESS
API
Integrations
B.

GraphQL

Typed Graph Schema

Declarative Language

Single Endpoint and Client Language

Simple Versioning

Does GraphQL replace Rest…

No

GraphQL Problems

Security