UNIVERSITETI I EJL
ЈИЕ УНИВЕРЗИТЕТ
SEE UNIVERSITY

# The impact of System Architecture on the Availability and Performance

## The case: Open Learning and Education System

Advanced Topics in Software Engineering

Nuhi Besimi, Prof. Dr. Betim Çiço, Prof. Dr. Klaus Bothe

DAAD: 16th Workshop "Software Engineering Education and Reverse Engineering"

# Content

- Project Objectives Statement
- Availability
- Performance
- System Architecture
- Data Management
- Experiments
- Conclusion

# Project Objectives Statement

- This is an **open source** project for integrating knowledge and education systems from all over the world. Our idea is to use the technology and contribute to the world.

- Open Learning and Education System
  - Integrate Universities and Students in one place to offer free education environment.
- Free open courses
- Homework
- Standalone and group projects/exams
- Consultation
- Evaluation

# Non-Functional Requirements

- **Availability**
- **Performance**
- **Scalability**
- Reliability
- Security
- Maintainability
- Extensibility
- Usability

# Non-Functional Requirements

- **Availability**
  - 90% of life is just showing up (Woody Allen).
  - Availability refers to a property of software that it is there and ready to carry out its task when you need it to be.

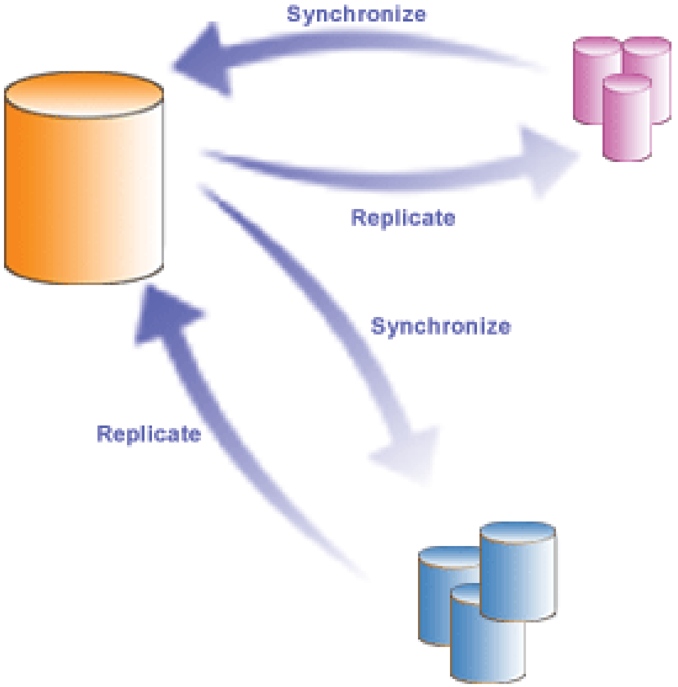
| Availability | Downtime/90 Days | Downtime/Year |
|---|---|---|
| 99.0% | 21 hours, 36 minutes | 3 days, 15.6 hours |
| 99.9% | 2 hours, 10 minutes | 8 hours, 46 seconds |
| 99.99% | 12 minutes, 58 seconds | 52 minutes, 34 seconds |
| 99.999% | 1 minute, 18 seconds | 5 minutes, 15 seconds |
| 99.9999% | 8 seconds | 32 seconds |

# Availability

- Faults in Distributed Systems
    - Node loses power
    - Hard disk fails
    - Administrator accidentally erases data
    - Administrator configures node incorrectly
    - Software bug triggers
    - Network overloaded, drops lots of packets
    - Hacker breaks into some of the nodes
    - Disgruntled employee manipulates node
    - Fire breaks out in data center where node resides
    - Police confiscates node because of illegal activity

# Availability

- Detect Faults
  - Ping/echo
  - Monitor
  - Heartbeat
  - Condition monitoring
  - **Replication**
  - System exceptions



# Availability

- Recover from Faults
  - Why we need luck when we have **redundancy** (Cisco Books).

  - Rollback
  - Software upgrade
  - Ignore faulty behavior
  - Reconfiguration

1+1 Redundancy

# Performance

- It's all about time

- It is about time and the software system's ability to meet timing requirements.

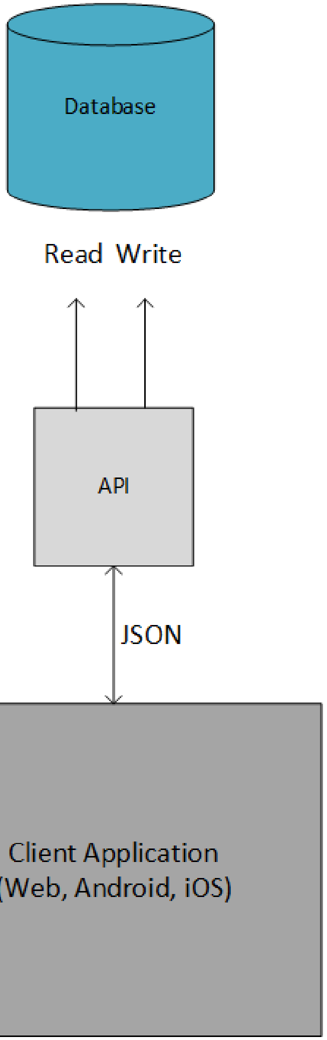
# Performance

- Processing Time
  - Processing consumes resources, which takes time. Hardware resources include CPU, data stores, network communication bandwidth, and memory.
- Blocked Time
  - A computation can be blocked because of contention for some needed resource.
  - Contention for resources
    - Single client at a time
  - Availability of resources
    - Resource is unavailable
  - Dependency on another computation

# Performance – Design checklist

- Allocation of Responsibilities
  - Determine the system's responsibilities that will involve heavy loading, heavy time-critical response requirements, are heavily used. Identify the **processing** requirements for these operations.
- Data Model
  - Determine the portions of the data model that will be heavily loaded, have time-critical response requirements or are heavily used.
  - Data replication, data partitioning, schema, processing requirements.

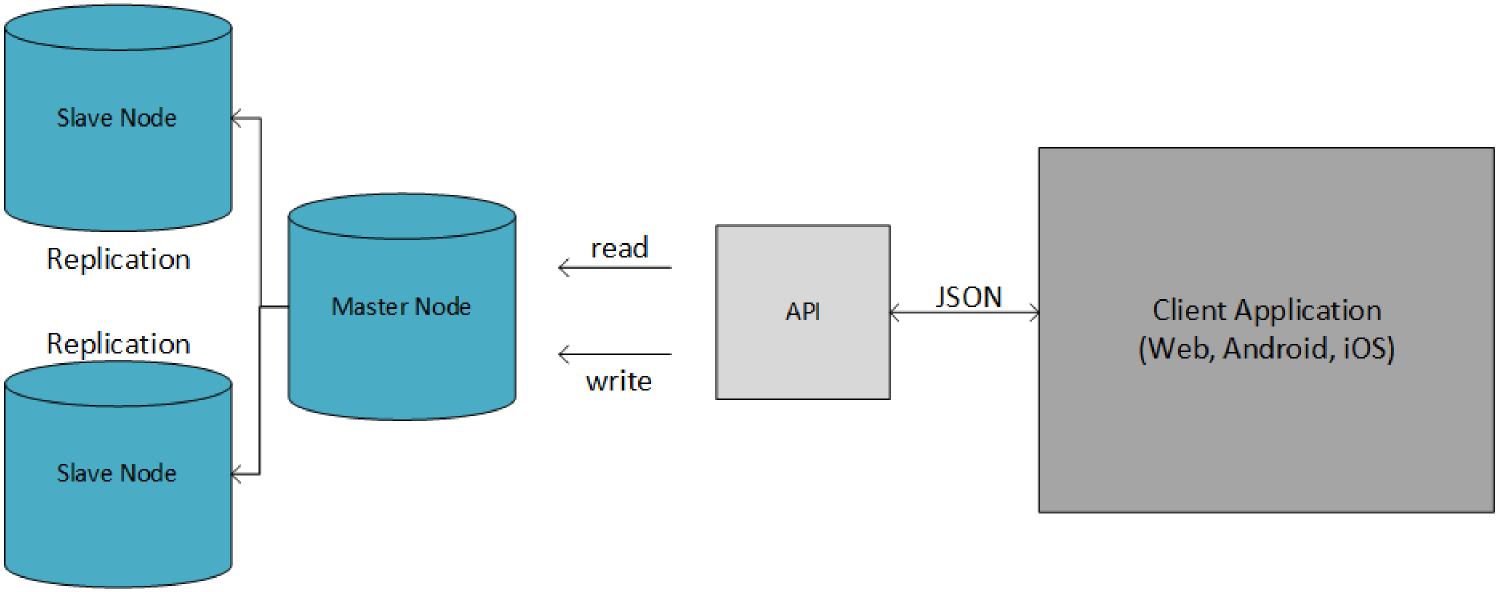
# System Architecture



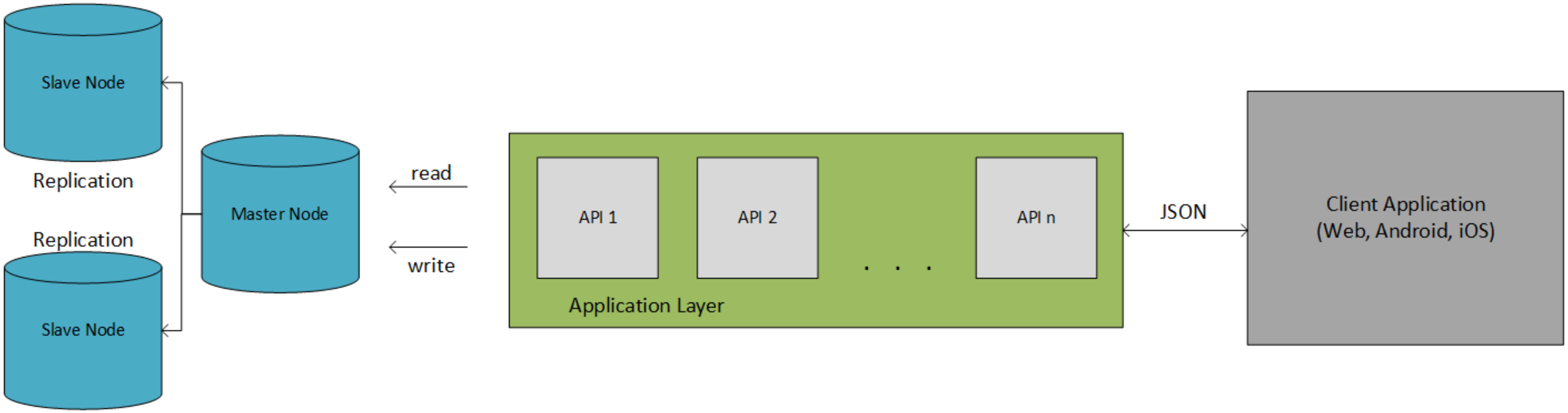
# Data Replication

- Database replication is the capability of a database to automatically maintain a copy of itself at a separate location.
- Real-time replication
- Master – Slave Architecture
- Database Replication can be used to improve:
  - Availability (backup & restore)
  - Performance (distribute INSERT/UPDATE/DELETE and SELECT operations on master/slave)

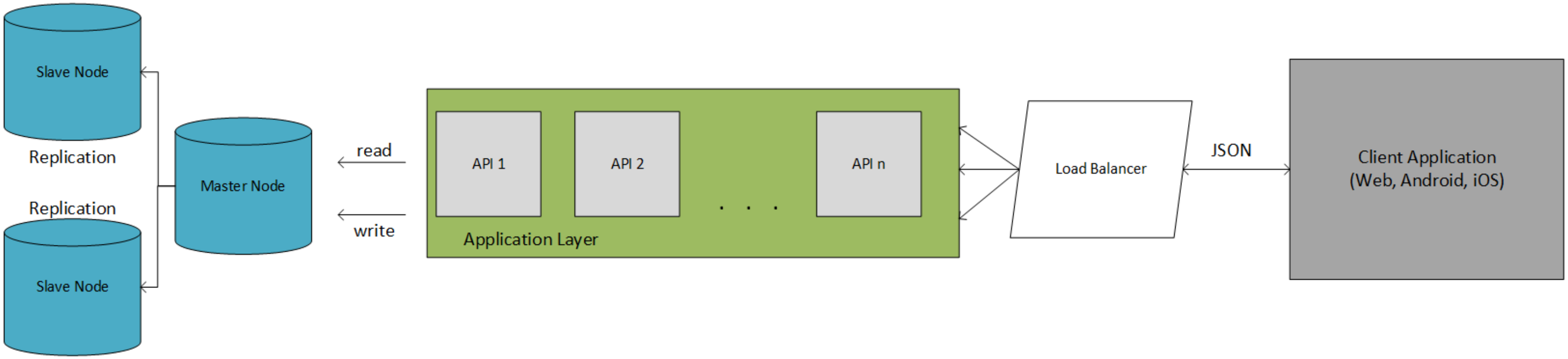
# System Architecture



# System Architecture



# System Architecture



# Data Management

- Unstructured Data
  - Data can be of any type.
  - It does not follow any format or sequence
  - It does not follow any rule, therefore it is not predictable.
- Bitmap Objects
  - Images, videos, or audio files.
- Textual Objects
  - Documents, E-mails, Spreadsheets
  - Web?!

# Data Management

- Structured Data
  - Data is organized in entities
  - Similar entities are grouped together
  - Entities on the same group have the same description
  - Same format
  - A predefined length
  - All present
  - The same order

# Data Management

- Semi-Structured Data
  - Organized in semantic entities
  - Similar entities are grouped together
  - Entities in the same group may **not** have the same attributes
    - Order of attributes not necessarily important
    - Not all attributes are required
    - Size of same attributes in a group may different
    - Type of the same attribute may be different

# SQL vs. NoSQL Databases

- Relational (SQL)
  - **Stuck**. Data now includes rich data types – tweets, videos, podcasts, animated gifs – which are hard, if not impossible, to store in a relational database. Development slows to a crawl, and ops is caught playing whack-a-mole.
  - **Can't Scale.** Your audience is global, in many countries, speaking many languages, accessing content on many devices. Scaling a relational database is not trivial. And it isn't cheap.
  - **$$$$.** Large teams tied up for long periods of time make these applications expensive to build and maintain. Proprietary software and hardware, plus separate databases and file systems needed to manage your content, add to the cost.

# SQL vs. NoSQL Databases

- Non-Relational (NoSQL)
  - **Do the Impossible**. NoSQL can incorporate literally any type of data, while providing all the features needed to build content-rich apps.
  - **Scale Big.** Scaling is built into the database. It is automatic and transparent. You can scale as your audience grows, both within a data center and across regions.
  - **$.** More productive teams, plus commodity hardware, make your projects cost 10% what they would with a relational database.

# MongoDB

- Building on the best of relational with the innovations of NoSQL
- MongoDB is an open-source, document database designed for ease of development and scaling.

# QUESTIONS?

??!