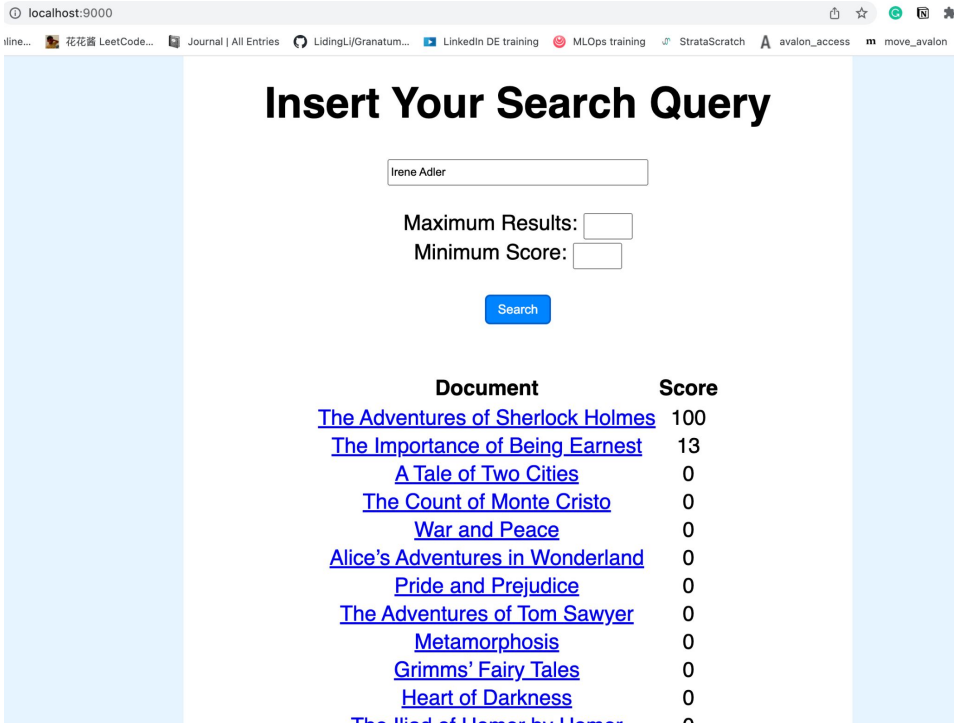
System Design of Distributed Search System in Java

(Notes from Udemy course)

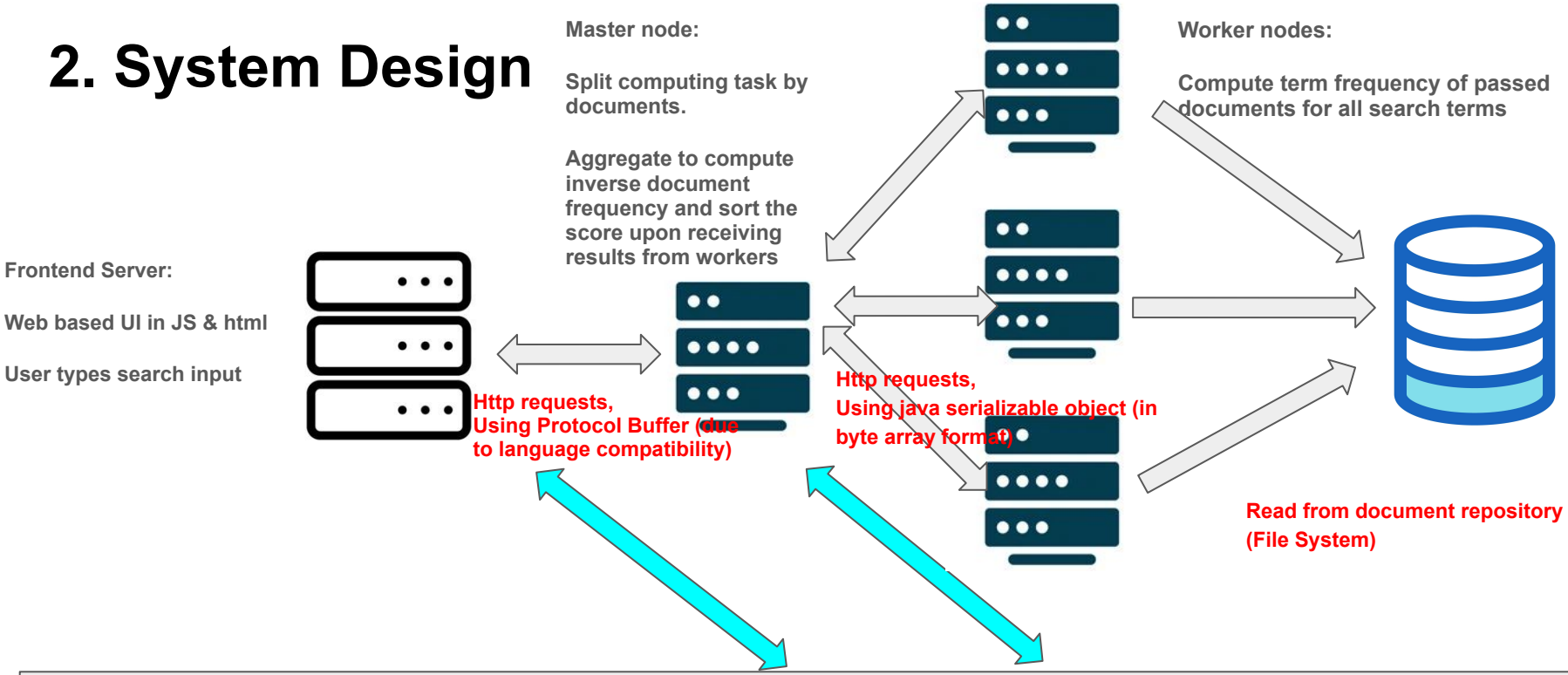
1. Demo



Frontend server (port 9000)

Backend Server:
1 master node on port 8080, 3 worker nodes on port 8081, 8082, 8083

2. System Design

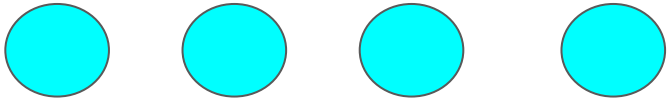


ZooKeeper Level

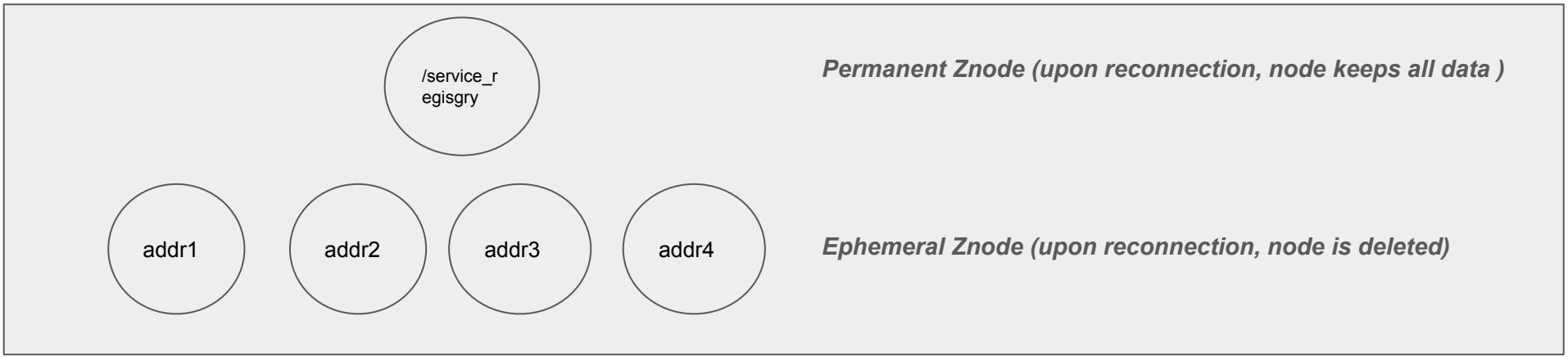
- *Store all registered node address*
- *Reelect Leader once a master node failed*

Coordinators(Master) Service Registry	Workers Service Registry

ZooKeeper Service Registry



Actual Node (communicating thru getChildren, getData NodeChildrenChanged to Znode)



Permanent Znode (upon reconnection, node keeps all data)

Ephemeral Znode (upon reconnection, node is deleted)

3. Side Notes

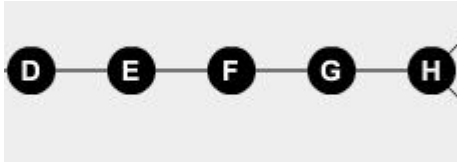
- In real case, the master node and worker nodes are usually machines, rather than a process. This system design simulates the situation where servers communicates through https.
- It is possible that the file system locates at another physical server or any cloud server. The communication would not be simple as retrieving files in local computer. Http request should be implemented.
- In terms of scaling of service, it is possible that numbers of users using the service at the same time, then the design needs load balancer between 1) user and frontend service, and 2) frontend service and backend service.

3. Side Notes
Master-Worker architecture:
Leader Reelection Algorithm

Once a master node fails, the service registry must reelect a new leader.

If all nodes watches all the other nodes, the cost will be huge.

One way is to allow one node to watch one the other node such that once a node fails, the watching node will know and become the leader



3. Side Notes
Network Communication Choices

- Json is most common
  - Human readable but no strict schema
  - Msg is in plain text, so its network overhead is larger than binary array
- Protocol Buffer is another (Google)
  - The benefit is speed and easy communication between different programing languages
- Serializing Object

  - Java Serializable object: Serializable interface

  - Python: pickle package

  …

  - The benefit is having a smaller overhead and clear schema