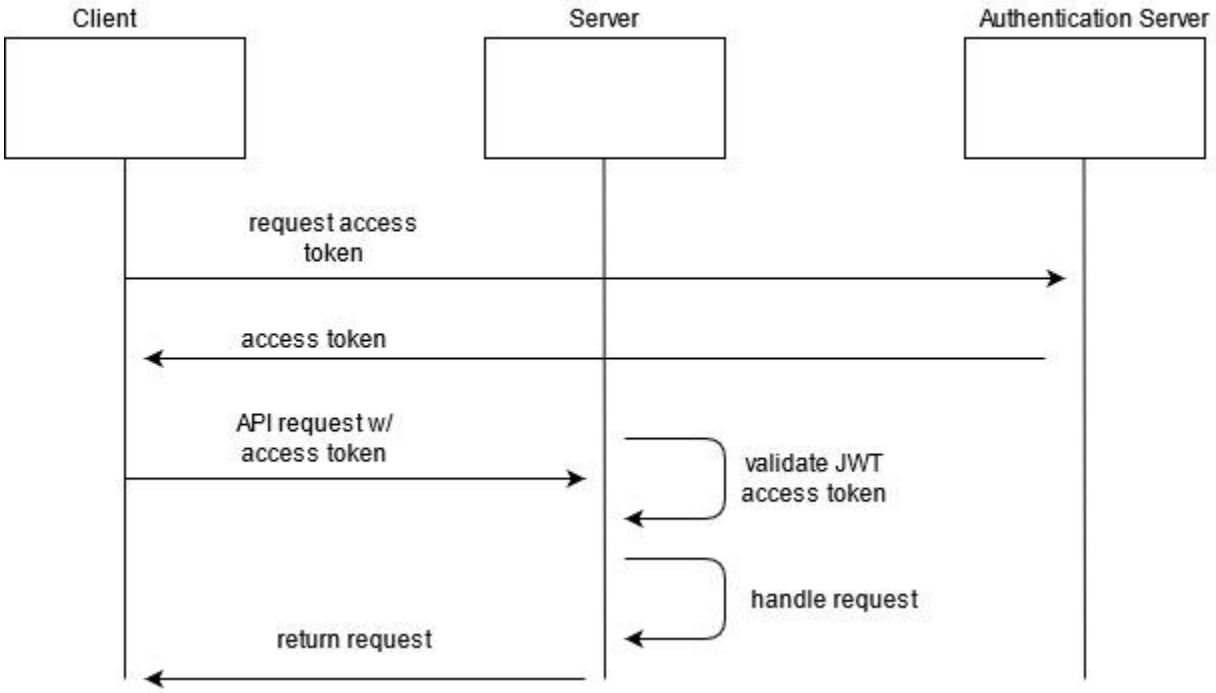
Team members: **Kayahan Kaya, Chivit Chhoeun, Amirabbas Sherafatian**

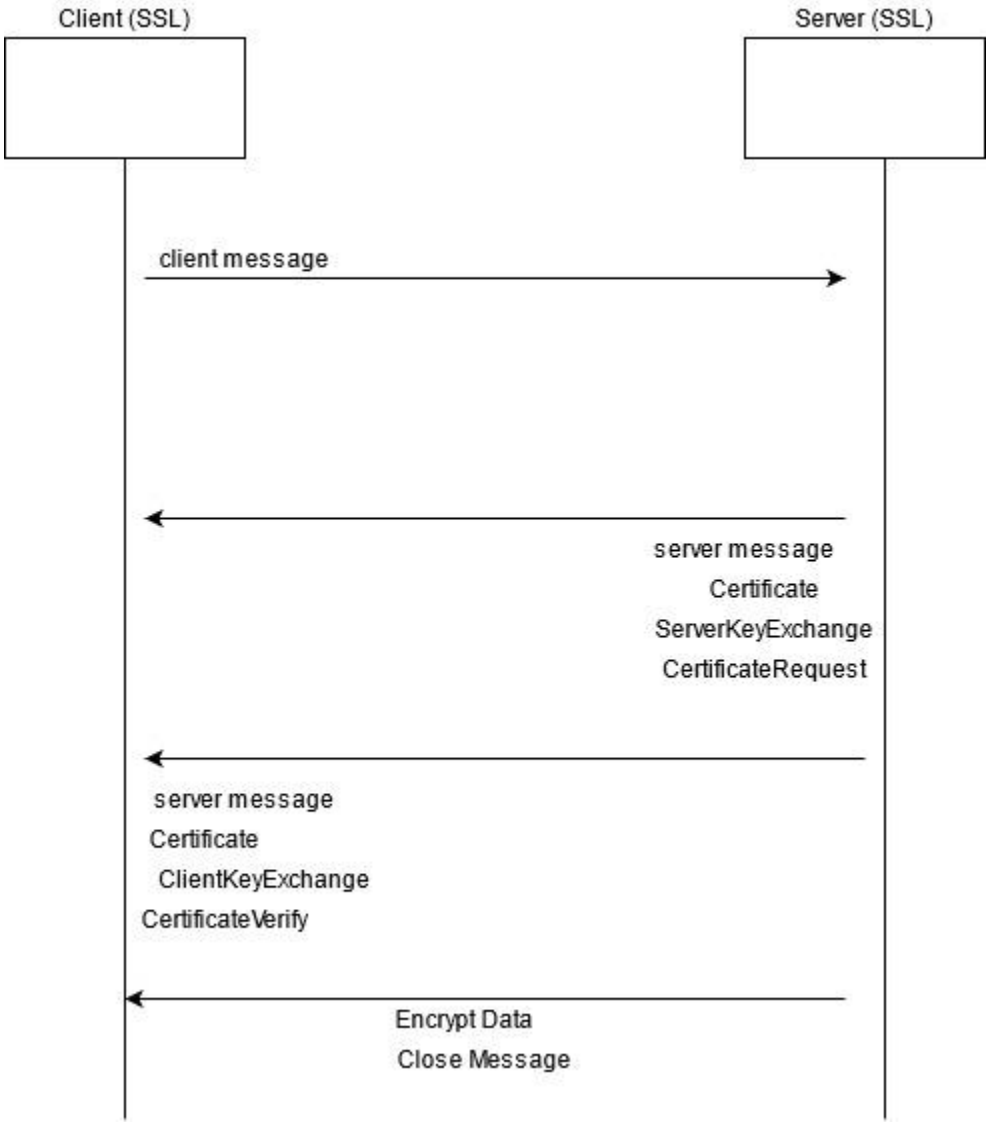
1.

**Authentication server with JWT:** In this new distributed system model, we introduce an authorization server. In this scenario, the resource server does not need to talk to the authorization server at all. The resource server just needs to verify the cryptographic signature in the token and makes sure the expiry date is in the future. And, since the majority of the load in the old model for the resource server was around validating all the requests, this is a huge reduction in the load on the system. JWT access token expiry date can be assigned to 6 months as specified in the requirement, so it is very useful when only a limited amount of time is needed for accessing the system.



Team members: **Kayahan Kaya, Chivit Chhoeun, Amirabbas Sherafatian**

**SSL/TLS Authentication:**




The following diagram represents the SSL/Transport Layer Security (TLS) handshake mechanism between a client and server. This is a two-way SSL authentication. In this concept, both the client and server shake hands by sending the expected and known Domain Name System(DNS)-named certificates (server certificate and client certificate) and confirming the system protocol before exchanging any information. Once the known system encryption is established, they start exchanging the information, which is a secured way of communication. Certificates can be set and set expiry dates depending on the time needed for the access.

Team members: **Kayahan Kaya, Chivit Chhoeun, Amirabbas Sherafatian**

2a.

**Given:**

- 100 clients x 4 I/O = 400 requests
- Think time = 3s = 160 ms
- Disk utilization = 60%
- Disk service time = 30 ms

- (1) $U_i = B_i \div T$, where $U_i$ *is the utilization of the disk resource* $B_i$ *is the total busy time of resource* $i$, and $T$ *is the observation time*
- (2) $S_i = B_i \div C_i$, where $S_i$ *is the disk service time,* $B_i$ *is the total busy time of resource* $i$, and $C_i$ *is the total service completed.*
- $R = (\frac{M}{X_i}) - Z$, where $X_i = C_i \div T$

(1) $B_i = U_i \times T$

(2) $B_i = C_i \times S_i$, therefore $U_i \times T = C_i \times S_i$

$0.6 \times T = 30 \times C_i => \frac{C_i}{T} = 0.02$ , where $C_i \div T = X_i = 0.02$

Therefore, $R = \frac{M}{X_i} - Z => (400 \times 500) - 180 = 199,820\ ms\ or\ 19,982\ s$

**Average response time of the interactive system: 199,820 ms or 19,982 s**

2b. According to the average response time calculated in part (a), every request takes 0.5s. This implies that the web server is fast and responsive to handle each request.

Team members: **Kayahan Kaya, Chivit Chhoeun, Amirabbas Sherafatian**

3. A Web server is deployed to handle a shopping Web page. Due to the increasing number of sold items, the Web page has become very unresponsive. To fix the problem, some performance measurements are collected from the server hosting the page (using logs). Measurements taken during one hour from the Web server shows that the CPU utilization and the utilization of the two disks are: UCPU = 0.25, Udisk1 = 0.35, and Udisk2 = 0.30. The Web server log shows that 141,600 requests were processed during the measurement interval. Please elaborate and use illustrations to respond to the following questions/statements.

a. Draw the QN model that is most appropriate to model this situation (qualitative aspects). (2pts)

b. What are the service demands at the CPU and both disks? (3pts)

c. What is the maximum throughput? (2pts)

d. What was the response time of the Web server during the measurement interval? (2pts)

e. What would be your recommendations in this situation? (1pt)

Answers;

Team members: **Kayahan Kaya, Chivit Chhoeun, Amirabbas Sherafatian**

**a)**

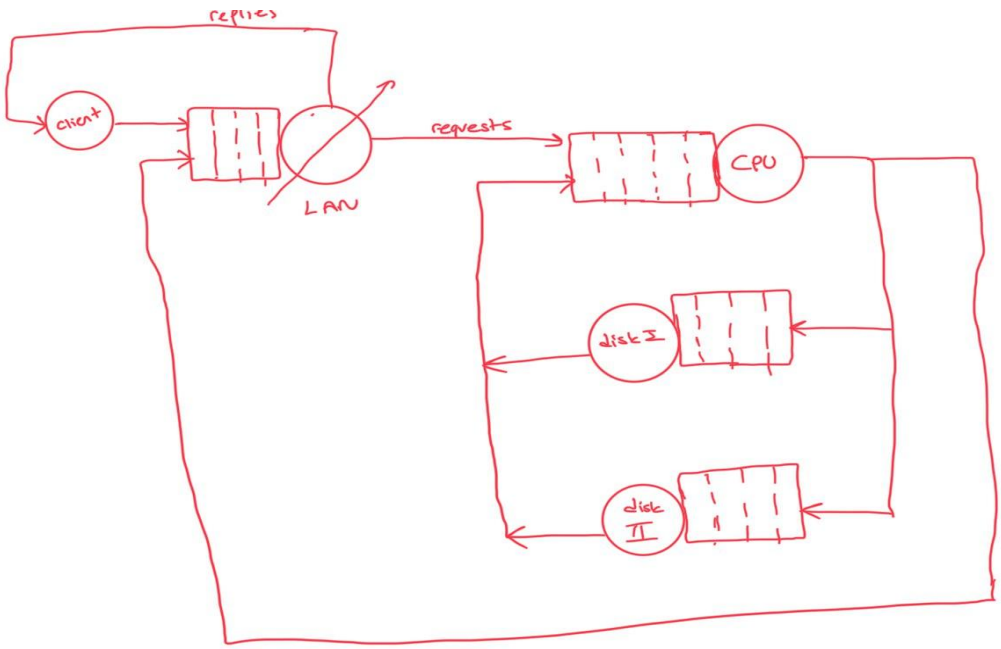



Figure: QN model

**b)**

T=3600 seconds (=60x60)

Web Server Throughput;

$X_0 = 141.600 \div 3600 = 39.3\ requests/seconds$

CPU utilization; $Ucpu$ =0.25

Disk1 utilization; $Udisk_1$=0.35

Disk2 utilization; $Udisk_2$=0.30

Service demand at the CPU

Team members: **Kayahan Kaya, Chivit Chhoeun, Amirabbas Sherafatian**

$$Dcpu = \frac{Ucpu}{Xo} = \frac{0.25}{39.3} = 0.00636 \text{ seconds/requests.}$$

$$Ddisk_1 = \frac{Udisk_1}{Xo} = \frac{0.35}{39.3} = 0.00890 \text{ seconds/requests.}$$

$$Ddisk_2 = \frac{Udisk_2}{Xo} = \frac{0.30}{39.3} = 0.00763 \text{ seconds/requests.}$$

c) Maximum throughput:

**d)**

z=0

$$R = \left(\frac{M}{Xo} - Z\right) = \frac{141.600}{39.3} = \textbf{3603.05 seconds}$$

**e.** Based on the response time of the server, it is implied that the server is not very responsive. During the observed time, resource utilization CPU and disks were not that high.

**4) Explain the key differences between stateful and stateless servers. Use illustrations to support your ideas (8 points)**

Stateful and stateless servers have a mutually dependent relationship when it comes to their definitions. To understand one, you need to understand the other.

**Stateful Server**

Team members: **Kayahan Kaya, Chivit Chhoeun, Amirabbas Sherafatian**

Stateful applications and processes, however, are those that can be returned to again and again, like online banking or email. They're performed with the context of previous transactions and the current transaction may be affected by what happened during previous transactions. For these reasons, stateful apps use the same servers each time they process a request from a user.

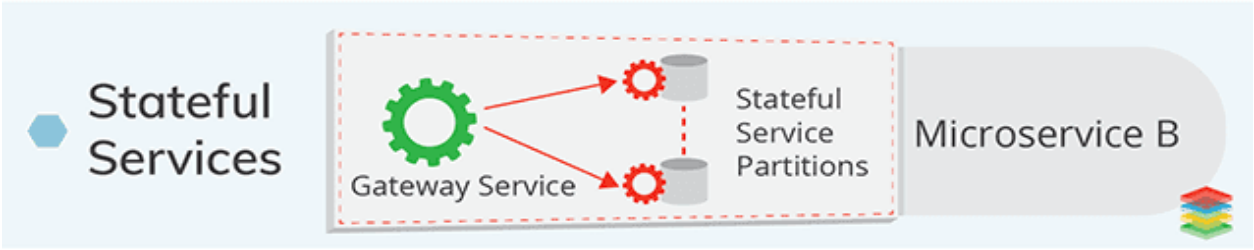



Figure 1: Stateful Services

If a stateful transaction is interrupted, the context and history have been stored so you can more or less pick up where you left off. Stateful apps track things like window location, setting preferences, and recent activity. You can think of stateful transactions as an ongoing periodic conversation with the same person.

The majority of applications we use day to day are stateful, but as technology advances, microservices and containers make it easier to build and deploy applications in the cloud.

**Stateless Server**

A stateless process or application can be understood in isolation. There is no stored knowledge of or reference to past transactions. Each transaction is made as if from scratch for the first time. Stateless applications provide one service or function and use a content delivery network (CDN), web, or print servers to process these short-term requests.

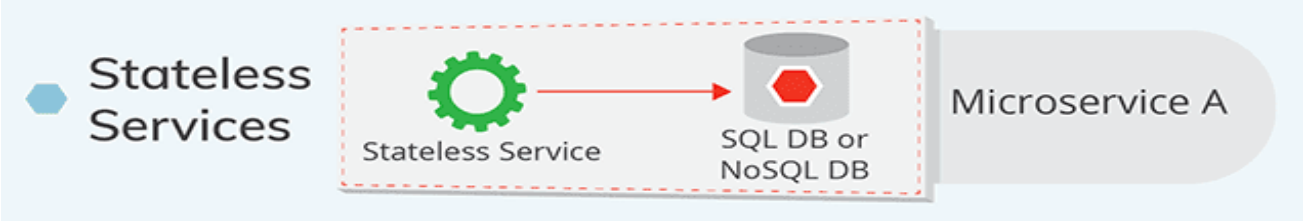



Figure 2: Stateless Services

An example of a stateless transaction would be doing a search online to answer a question you've thought of. You type your question into a search engine and hit enter. If your transaction is interrupted or closed accidentally, you just start a new one. Think of stateless transactions as a vending machine: a single request and a response.

Figure 3: Stateless vs. Stateful

**a) State Information:**

- A Stateful server remembers client data (state) from one request to the next. Stateful servers store session state Therefore, keep track of which clients have opened which files, current read and write pointers for files, which files have been locked by which clients, etc.
- A Stateless server keeps no state information. Stateless file servers do not store any session state. This means that every client request is treated independently, and not as a part of a new or existing session.

**b) Programming:**

- Stateful server is harder to code
- Stateless server is straightforward to code

**c) Crash Recovery:**

- Stateful servers have difficult crash recovery due to loss of information.
- Stateless servers can easily recover from failure because there is no state that must be restored.

**d) Information Transfer:**

- Using a Stateful file server, the client can send less data with each request.
- Using a stateless file server, the client must specify complete file names in each request, specify location for reading or writing and re-authenticate for each request.

5. $Maximum\ possible\ available\ time\ = 3 \times 24 \times 60 = 4,320\ minutes$
Downtime = 10 + 3 + 5 + 5 + 7 + 1 = 31 minutes.
**Availability after downtime = 4320 - 31 = 4280 minutes**

Team members: **Kayahan Kaya, Chivit Chhoeun, Amirabbas Sherafatian**

6.

a. If Disk 1 is replaced with a 50% slower disk, the service demand for each class for disk 1 will increase by 50% because slower disk means slower system.
b. If Disk 1, Disk 2 and Disk 3 are replaced by disks that are 50% faster, the service demand for the whole system will decrease by 50%.
c. The response time will increase by 70%.
d. The response time will increase by 45%
e. The response time will decrease because now there is less load on the system.