

# SER 321 B Session

**Exam Review Session**

**Sunday, December 1st 2024**

*7:00 pm - 9:00 pm MST*

# Agenda



Exam Information

Study Guide PSA

Requested Content

General Review!

# SI Session Expectations

Thanks for coming to the **SER 321** SI session. We have a packed agenda and we are going to try to get through as many of our planned example problems as possible. This session will be recorded and shared with others.

- If after this you want to see additional examples, please visit the drop-in tutoring center.
- We will post the link in the chat now and at the end of the session.
  - [tutoring.asu.edu](https://tutoring.asu.edu)
- Please keep in mind we are recording this session and it will be made available for you to review 24-48 hours after this session concludes.
- Finally, please be respectful to each other during the session.

# Interact with us:

## Zoom Features



### Zoom Chat

- Use the chat feature to interact with the presenter and respond to presenter's questions.
- Annotations are encouraged

**SER 321**

**Exam Information**

[Exam Info Page](#)

80 minutes

Very similar to  
the quizzes

Make sure to look at  
the Study Guide!

**Opens:** Wednesday  
December 4th  
@ 12:01 AM

**Closes:** Friday  
December 6th  
@ 11:59 PM

Front and Back!

MUST BE *Handwritten*



# SER 321

## Review Requests

It's ***not*** too late to make a topic request!

Drop a concept in the chat  
and we can cover it next!

First Request:  
Threads and Serialization

# SER 321

## Threaded Pitfalls

Starvation

A thread is only able to acquire some of the resources it needs

Deadlock

More than one thread accesses a single resource at the same time

Race Condition

A thread never gains access to the resource it needs

# SER 321

## Threaded Pitfalls

Starvation

A thread is only able to acquire some of the resources it needs

Deadlock

More than one thread accesses a single resource at the same time

Race Condition

A thread never gains access to the resource it needs



### What's the difference?

#### Starvation

A thread never gains access to the resource it needs

Waiting to access the **CPU**

Ready to go; never gets a chance

vs.

#### Deadlock

A thread is only able to acquire some of the resources it needs

Waiting to access another **resource**

**Not** ready to go

# SER 321

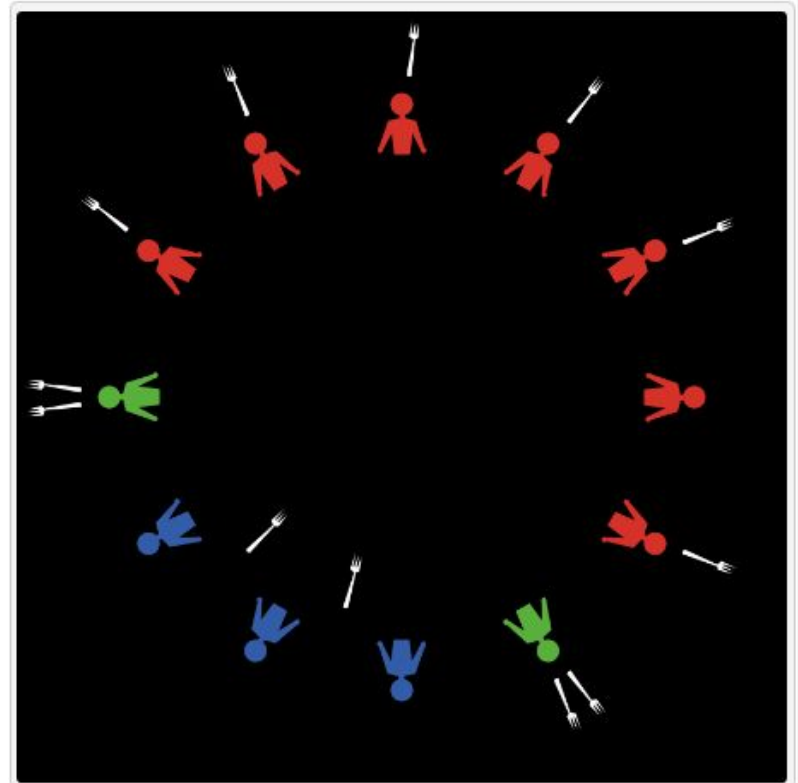
## Dining Philosophers

Can we take a guess at what is happening here?

What are the **BLUE** people doing?

What are the **GREEN** people doing?

What are the **RED** people doing?



# SER 321

## JSON Structure

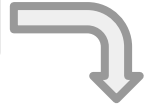
Data is stored in...

Name:Value pairs

AKA

Members

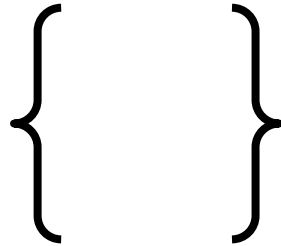
"Katie"



"student" : "Katie"

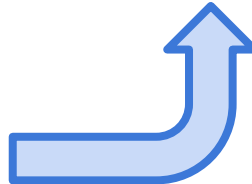
What uses curly braces?

Objects



What do Objects contain?

Members



# SER 321

## JSON Structure

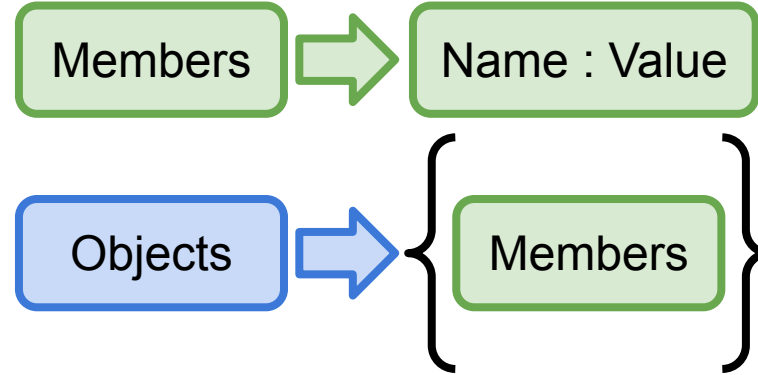
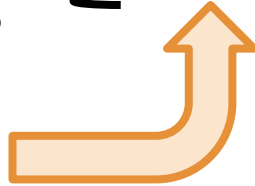
What uses brackets?

Arrays

[ ]

What do Arrays contain?

Any ***Valid*** Value



# SER 321

## JSON Structure

What is a valid value?

Strings

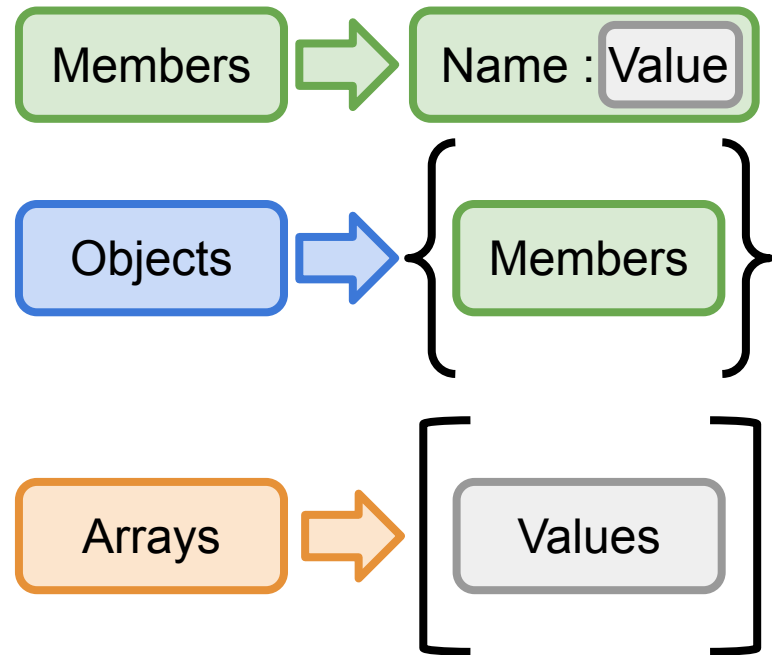
Booleans

Numbers

NULL

Objects

Arrays



**SER 321**

**Serialization**

Can we recall some of the formats?

JSON

Java Object  
Serialization

Protocol Buffers

XML

# SER 321

## Serialization

Binary

Text

Two main  
approaches for  
storing the  
content...

What about the data format?

JSON

Java Object  
Serialization

Protocol Buffers

XML

# SER 321

## Serialization

Binary

Text

Who uses *TEXT*?

Text

JSON

Java Object  
Serialization

Protocol Buffers

Text

XML



Binary

Text

What does  
this imply?

Who uses ***BINARY***?

Text

JSON

Binary

Java Object  
Serialization

Binary

Protocol Buffers

Text

XML

**SER 321**

**Serialization**

Generic  
Superclass

# Streams and their types

```
OutputStream out = sock.getOutputStream();
```

Buffered Stream

Bytes

Data Stream

Primitive DATA Types

Object Stream

Java Objects

# SER 321

## JSON Recognition

How many Objects?

How many Arrays?

How many Members?

```
{
  "lat": 42.3434,
  "lon": -88.0412,
  "timezone": "America/Chicago",
  "timezone_offset": -21600,
  "current": {
    "dt": 1733070576,
    "sunrise": 1733058144,
    "sunset": 1733091649,
    "temp": 18.57,
    "feels_like": 5.97,
    "pressure": 1025,
    "humidity": 63,
    "dew_point": 9.21,
    "uvi": 0.79,
    "clouds": 0,
    "visibility": 10000,
    "wind_speed": 14.97,
    "wind_deg": 280,
    "wind_gust": 21.85,
    "weather": [
      {
        "id": 800,
        "main": "Clear",
        "description": "clear sky",
        "icon": "01d"
      }
    ]
  }
}
```

# SER 321

## JSON Practice

JSONObject json =

How would we...

Check for the timezone member?

boolean hasTimezone =

Get the timezone?

String timezone =

```
{
  "lat": 42.3434,
  "lon": -88.0412,
  "timezone": "America/Chicago",
  "timezone_offset": -21600,
  "current": {
    "dt": 1733070576,
    "sunrise": 1733058144,
    "sunset": 1733091649,
    "temp": 18.57,
    "feels_like": 5.97,
    "pressure": 1025,
    "humidity": 63,
    "dew_point": 9.21,
    "uvi": 0.79,
    "clouds": 0,
    "visibility": 10000,
    "wind_speed": 14.97,
    "wind_deg": 280,
    "wind_gust": 21.85,
    "weather": [
      {
        "id": 800,
        "main": "Clear",
        "description": "clear sky",
        "icon": "01d"
      }
    ]
  }
}
```

# SER 321

## JSON Practice

JSONObject json =

How would we...

Obtain the temp value?

~~String temp = json.getString("temp");~~

```
{
  "lat": 42.3434,
  "lon": -88.0412,
  "timezone": "America/Chicago",
  "timezone_offset": -21600,
  "current": {
    "dt": 1733070576,
    "sunrise": 1733058144,
    "sunset": 1733091649,
    "temp": 18.57,
    "feels_like": 5.97,
    "pressure": 1025,
    "humidity": 63,
    "dew_point": 9.21,
    "uvi": 0.79,
  }
}
```

Recall that nested members require multiple steps!

Step 1:

Step 2:

Step 3:

Step 4:

# SER 321

## JSON Practice

JSONObject json =

How would we...

Obtain the temp value?

~~String temp = json.getString("temp");~~

if (json.has("current")) {

}

```
{
  "lat": 42.3434,
  "lon": -88.0412,
  "timezone": "America/Chicago",
  "timezone_offset": -21600,
  "current": {
    "dt": 1733070576,
    "sunrise": 1733058144,
    "sunset": 1733091649,
    "temp": 18.57,
    "feels_like": 5.97,
    "pressure": 1025,
    "humidity": 63,
    "dew_point": 9.21,
    "uvi": 0.79,
  }
}
```

Recall that  
nested  
members  
require  
multiple steps!

Step 1: Check for parent object

Step 2:

Step 3:

Step 4:

# SER 321

## JSON Practice

JSONObject json =

How would we...

Obtain the temp value?

~~String temp = json.getString("temp");~~

if (json.has("current")) {

JSONObject current =

json.getJSONObject("current");

}

```
{
  "lat": 42.3434,
  "lon": -88.0412,
  "timezone": "America/Chicago",
  "timezone_offset": -21600,
  "current": {
    "dt": 1733070576,
    "sunrise": 1733058144,
    "sunset": 1733091649,
    "temp": 18.57,
    "feels_like": 5.97,
    "pressure": 1025,
    "humidity": 63,
    "dew_point": 9.21,
    "uvi": 0.79,
  }
}
```

Recall that  
nested  
members  
require  
multiple steps!

Step 1: Check for parent object

Step 2: Obtain parent object

Step 3:

Step 4:

# SER 321

## JSON Practice

JSONObject json =

How would we...

Obtain the temp value?

~~String temp = json.getString("temp");~~

if (json.has("current")) {

    JSONObject current =

        json.getJSONObject("current");

    if (current.has("temp")) {

        temp = current.getString("temp");

}

```
{
  "lat": 42.3434,
  "lon": -88.0412,
  "timezone": "America/Chicago",
  "timezone_offset": -21600,
  "current": {
    "dt": 1733070576,
    "sunrise": 1733058144,
    "sunset": 1733091649,
    "temp": 18.57,
    "feels_like": 5.97,
    "pressure": 1025,
    "humidity": 63,
    "dew_point": 9.21,
    "uvi": 0.79,

```

Recall that  
nested  
members  
require  
multiple steps!

Step 1: Check for parent object

Step 2: Obtain parent object

Step 3: Check for nested member

Step 4:



# SER 321

## JSON Practice

JSONObject json =

How would we create the  
“weather” object?

```
{
  "lat": 42.3434,
  "lon": -88.0412,
  "timezone": "America/Chicago",
  "timezone_offset": -21600,
  "current": {
    "dt": 1733070576,
    "sunrise": 1733058144,
    "sunset": 1733091649,
    "temp": 18.57,
    "feels_like": 5.97,
    "pressure": 1025,
    "humidity": 63,
    "dew_point": 9.21,
    "uvi": 0.79,
    "clouds": 0,
    "visibility": 10000,
    "wind_speed": 14.97,
    "wind_deg": 280,
    "wind_gust": 21.85,
    "weather": [
      {
        "id": 800,
        "main": "Clear",
        "description": "clear sky",
        "icon": "01d"
      }
    ]
  }
}
```

# SER 321

## JSON Practice

JSONObject json =

How would we create the  
“weather” object?

```
JSONObject json = new JSONObject();  
JSONObject weather = new JSONArray();  
JSONObject content = new JSONObject();  
content.put("id", 800);  
content.put("main", "Clear");  
content.put("description", "clear sky");  
content.put("icon", "01d");
```

```
weather.put(content.toMap());
```

```
json.put(weather.toMap());
```

```
{  
  "lat": 42.3434,  
  "lon": -88.0412,  
  "timezone": "America/Chicago",  
  "timezone_offset": -21600,  
  "current": {  
    "dt": 1733070576,  
    "sunrise": 1733058144,  
    "sunset": 1733091649,  
    "temp": 18.57,  
    "feels_like": 5.97,  
    "pressure": 1025,  
    "humidity": 63,  
    "dew_point": 9.21,  
    "uvi": 0.79,  
    "clouds": 0,  
    "visibility": 10000,  
    "wind_speed": 14.97,  
    "wind_deg": 280,  
    "wind_gust": 21.85,  
    "weather": [  
      {  
        "id": 800,  
        "main": "Clear",  
        "description": "clear sky",  
        "icon": "01d"  
      }  
    ]  
  }  
}
```

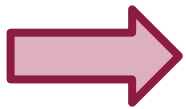
**SER 321**

**OSI Model**

Unit

Layer

What we are *really*  
talking about

**SER 321**

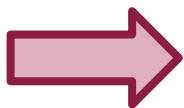
**OSI Model**

Unit

Layer

What we are *really*  
talking about

Bits	Physical	Signal, Binary transmission



**SER 321**

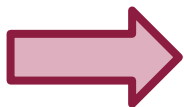
**OSI Model**

Unit

Layer

What we are *really*  
talking about

Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission



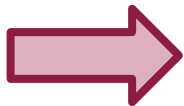
**SER 321**

**OSI Model**

Unit

Layer

What we are *really*  
talking about

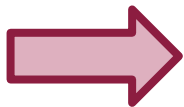


Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

**SER 321****OSI Model**

Unit

Layer

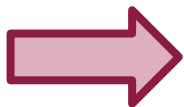
What we are *really*  
talking about

Segment	Transport	TCP/UDP
Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

**SER 321****OSI Model**

Unit

Layer

What we are *really*  
talking about

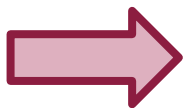
Data	Session	AuthN, authZ, session mgmt
Segment	Transport	TCP/UDP
Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission



**SER 321****OSI Model**

Unit

Layer

What we are *really*  
talking about

Data	Presentation	Translation, compression, encryption
Data	Session	AuthN, authZ, session mgmt
Segment	Transport	TCP/UDP
Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

**SER 321**

**Network Layer - IP**

Given the following IP address, identify the...

Port

128.148.32.110 8080

**SER 321**

**Network Layer - IP**

Given the following IP address, identify the...

Subnet

128.148.32.110:8080

**SER 321**

**Network Layer - IP**

Given the following IP address, identify the...

Network

128.148.32.110:8080

**SER 321**

**Network Layer - IP**

Given the following IP address, identify the...

Host

128.148.32.110:8080

What are the main differences?

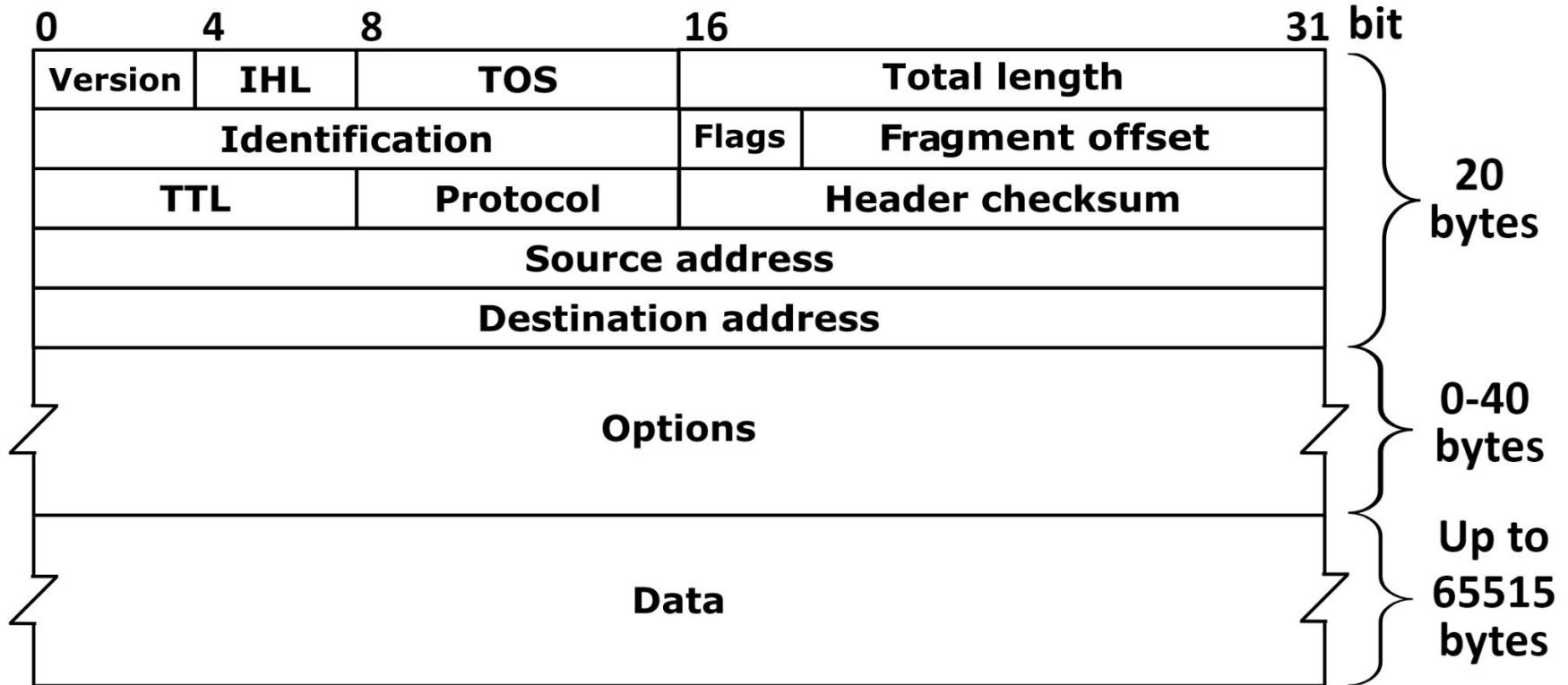
TCP	UDP

**SER 321**

**Headers**

What type of header is this?

IP Header



## Headers

TCP or UDP Header?

Offsets		0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0 0				C W R	E C E	U R G	A C K	P S H	R S T	S Y N	F I N	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if <i>data offset</i> > 5. Padded at the end with "0" bits if necessary.)																															
:	:																																
56	448																																



**SER 321**

**URLs**

Given the following URL, identify the...

**Query**

<https://www.google.com/search?q=asu>

**SER 321**

**URLs**

Given the following URL, identify the...

**Protocol**

**https://www.google.com/search?q=asu**

**SER 321**

**URLs**

Given the following URL, identify the...

**Path**

<https://www.google.com/search?q=asu>

**SER 321**

**URLs**

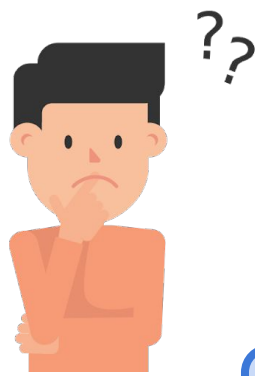
Given the following URL, identify the...

**Host**

<https://www.google.com/search?q=asu>

**SER 321**

**HTTP(s)**



Stateful

OR

Stateless

Synchronous

OR

Asynchronous

What's the difference?

1. GET

2. POST

3. PUT

4. DELETE

# SER 321

## HTTP Responses

Status Codes →

1XX

2XX

3XX

4XX

5XX

# SER 321

## Socket Properties

Sockets allow our client and server to communicate!

Location

Connection  
Semantics

Message Format

Need to define **3 properties** before usage

IP or DNS

142.251.46.206

www.google.com

TCP or UDP

Connection  
Oriented

Connectionless

Protocol Specs

Synchronous

Asynchronous

Stateless

Stateful

Binary

Text

Headers

No Headers





**SER 321**

**Client Socket**

## Steps for the **Client Socket**

1.

2.

3.

4.

5.

6.

7.

8.

**SER 321**

**Server Socket**

# Steps for the **Server Socket**

1.

2.

3.

4.

5.

6.

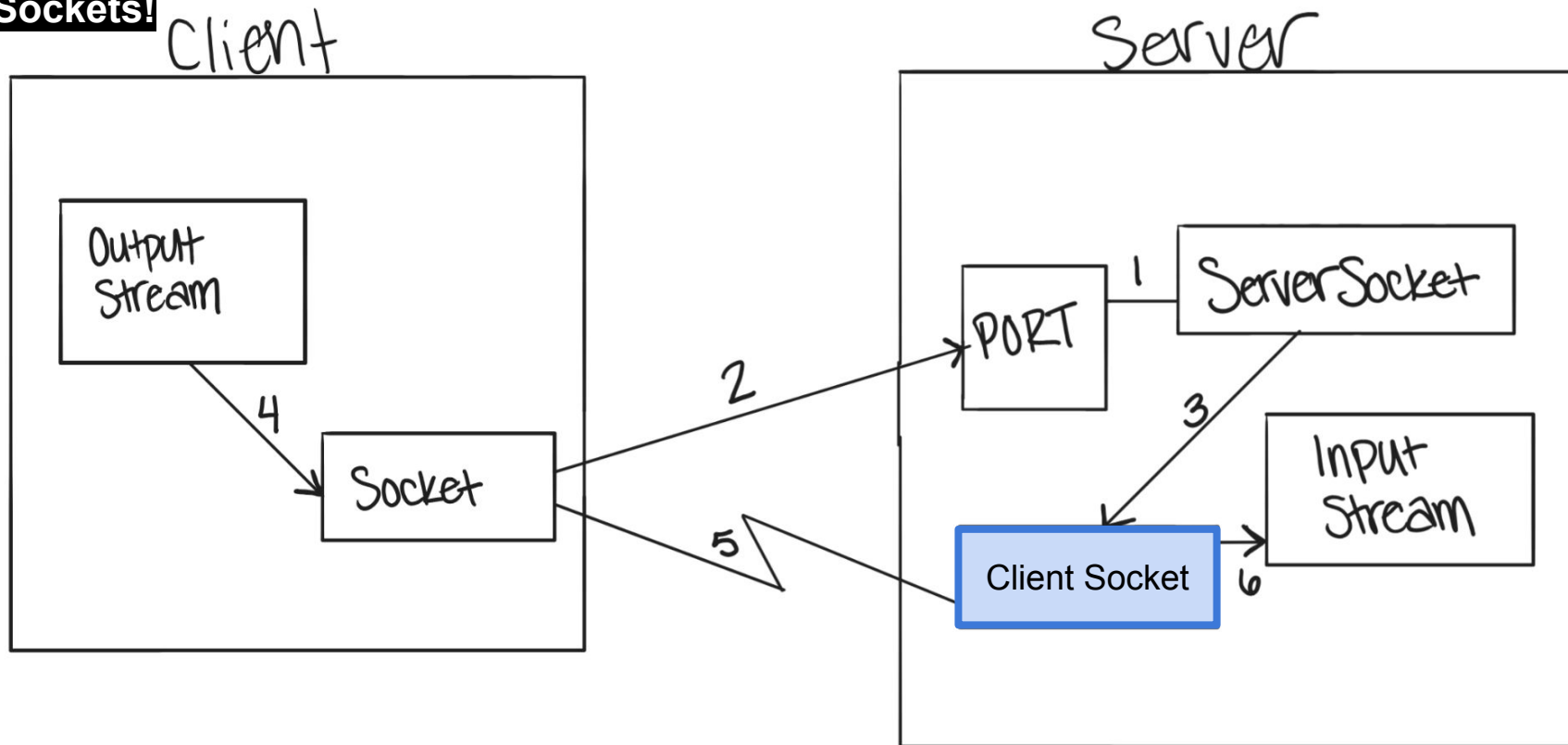
7.

8.

9.

# SER 321

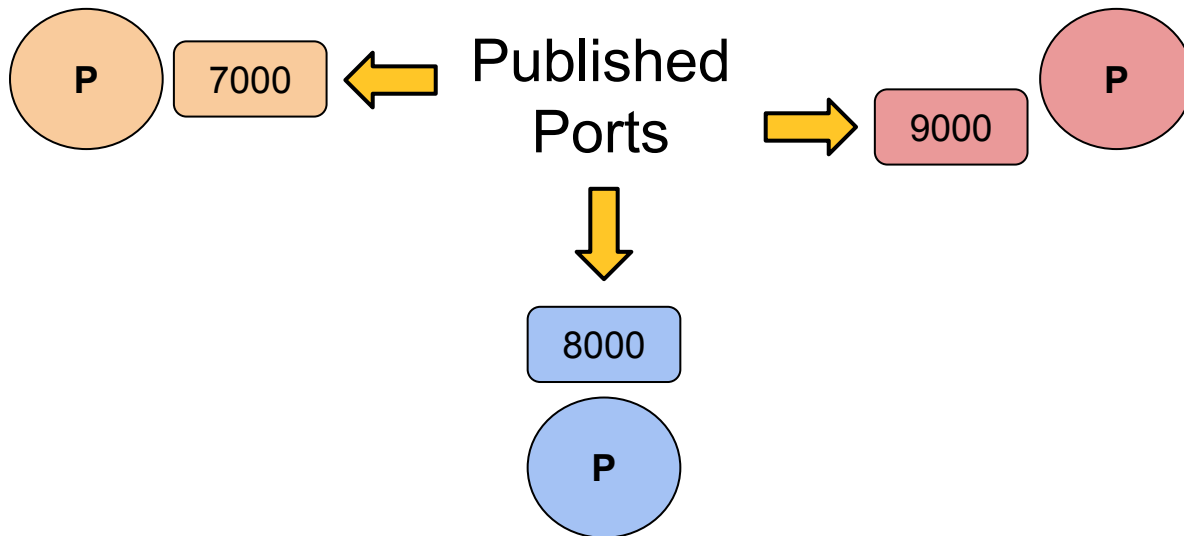
## Sockets!



# SER 321

## Communication

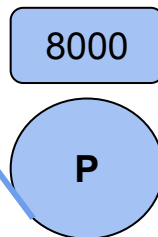
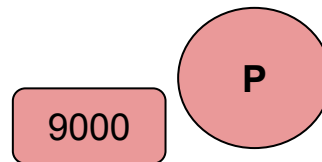
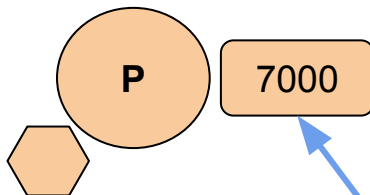
Remember that the OS allocates a new port for the client socket!



# SER 321

## Communication

Remember that the OS allocates a new port for the client socket!

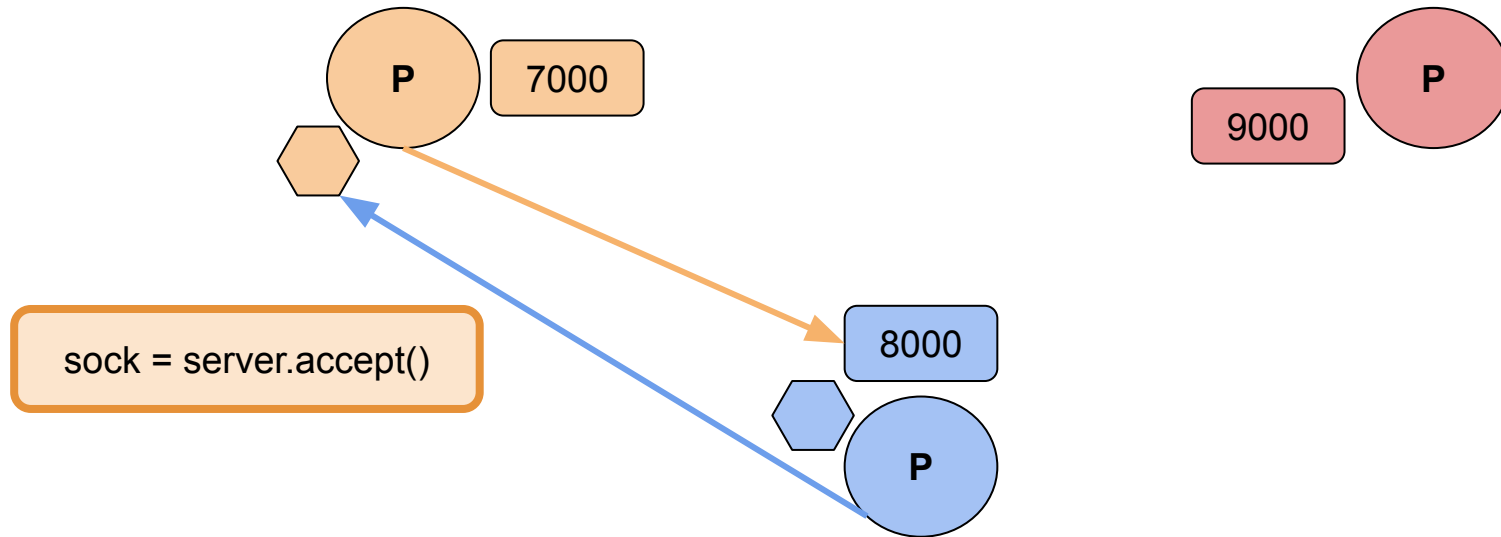


```
sock = server.accept()
```

# SER 321

## Communication

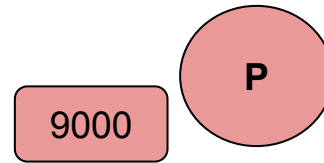
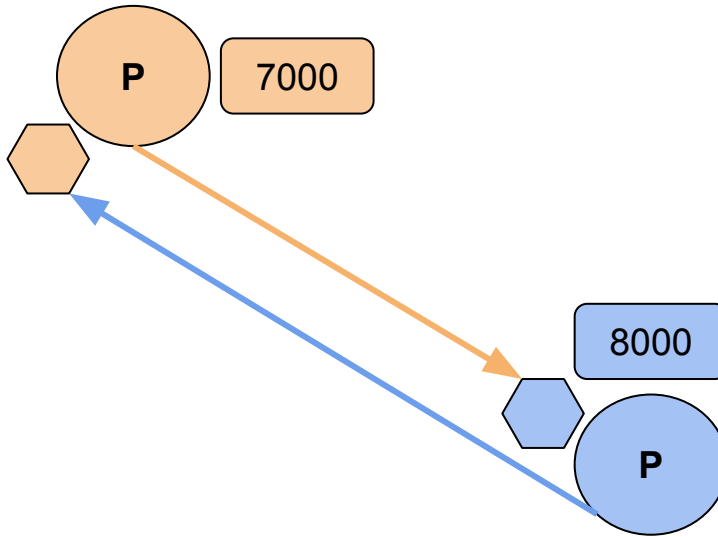
Remember that the OS allocates a new port for the client socket!



**SER 321**

**Communication**

Remember that the OS allocates a new port for the client socket!

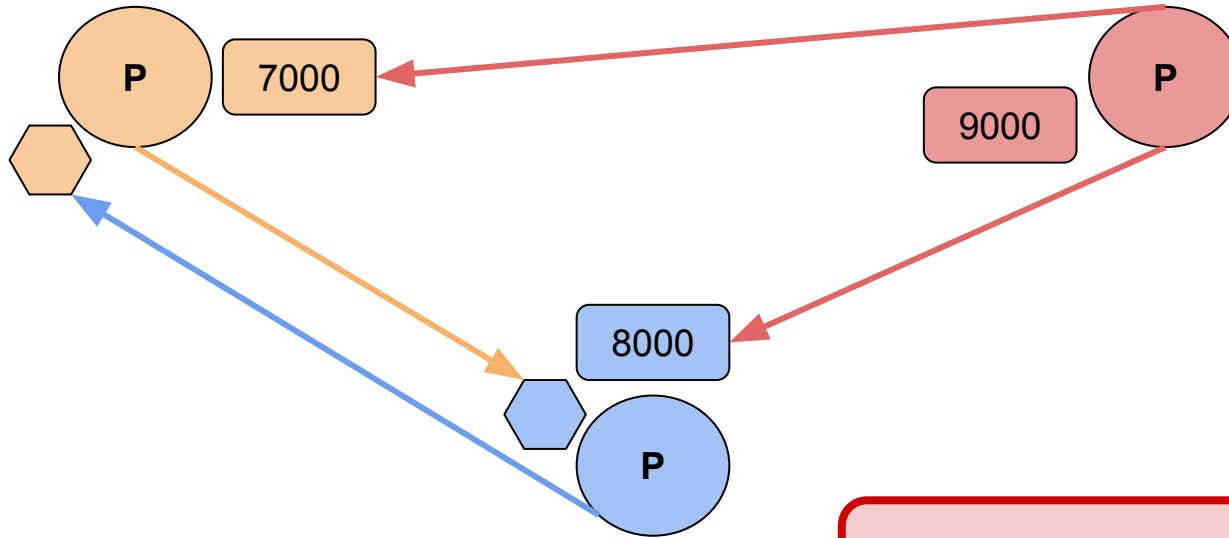


What about Peer 9000?

# SER 321

## Communication

Remember that the OS allocates a new port for the client socket!



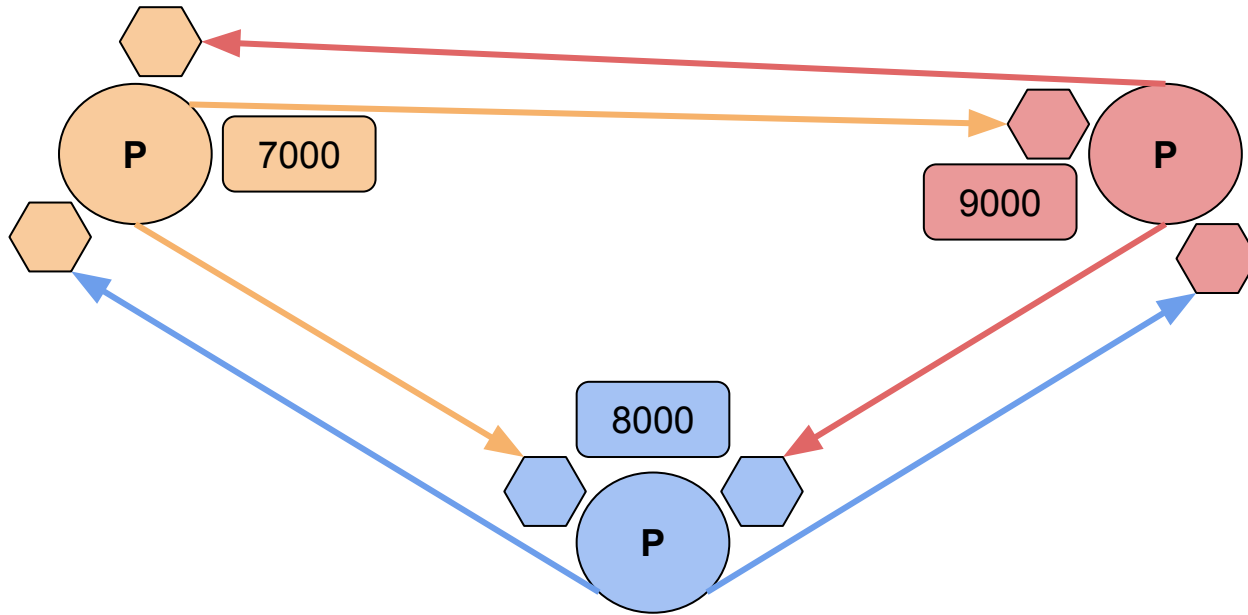
What about Peer 9000?

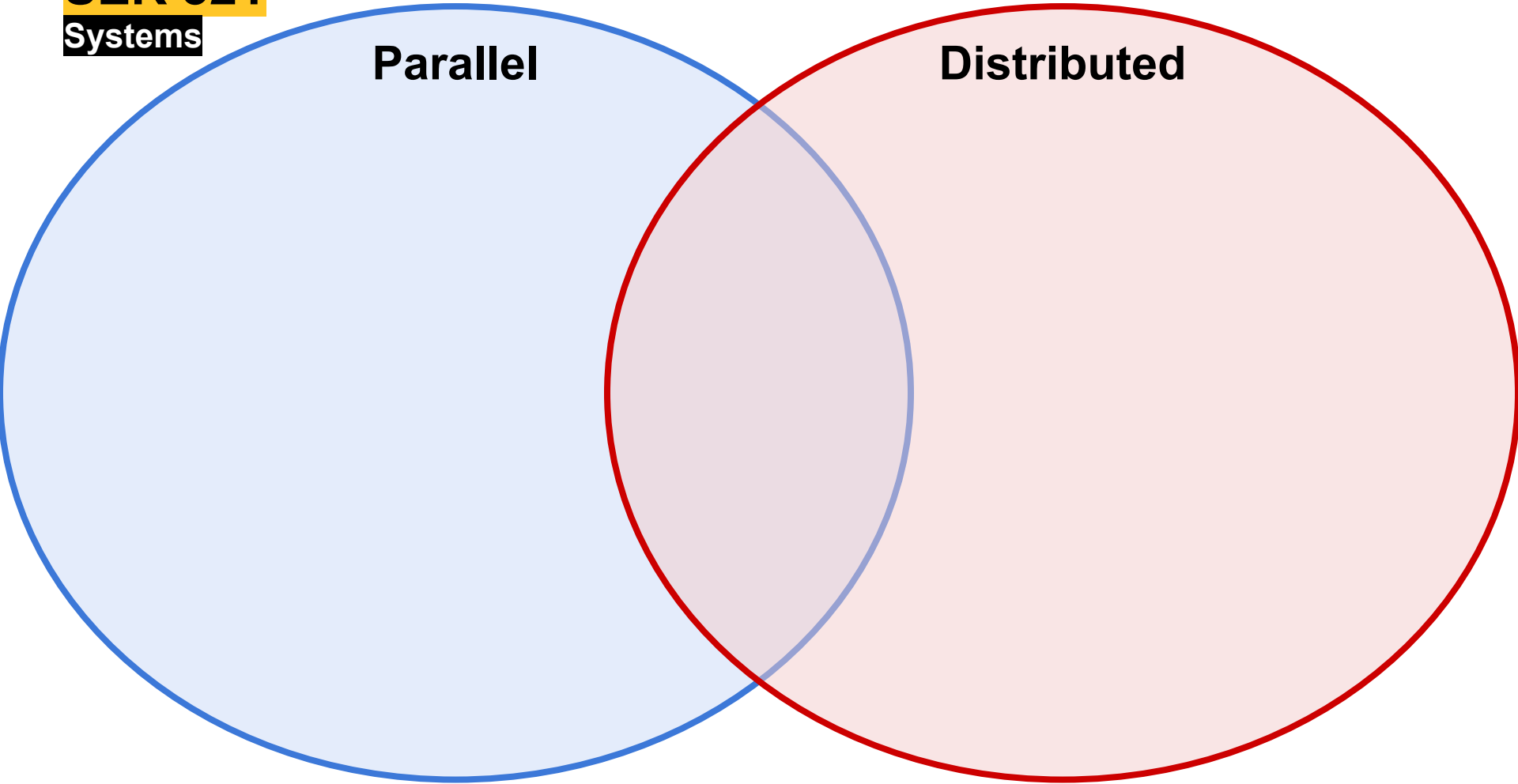


# SER 321

## Communication

Remember that the OS allocates a new port for the client socket!





### Parallel

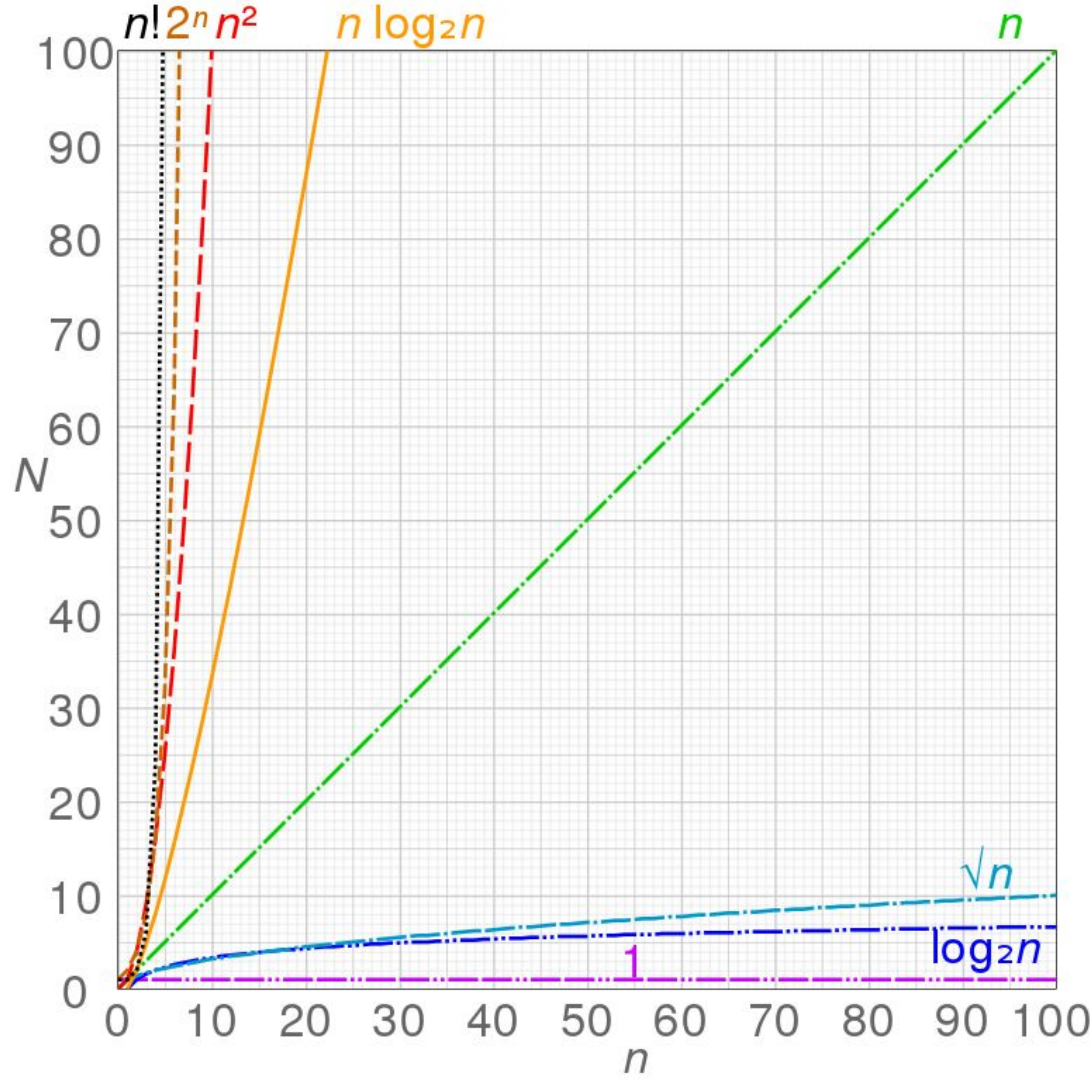
- Single computer
- Work split among different *processors*
- Memory is shared **or** distributed
- Communicate through *bus*

### Distributed

- Work is partitioned
- Partitions processed individually
- **Can** improve performance
- **Can** improve speed
- Many computers
- Work split among different *locations*
- Memory is distributed
- Communicate through *message passing*

## When to Distribute

When should  
we *consider*  
distributing?

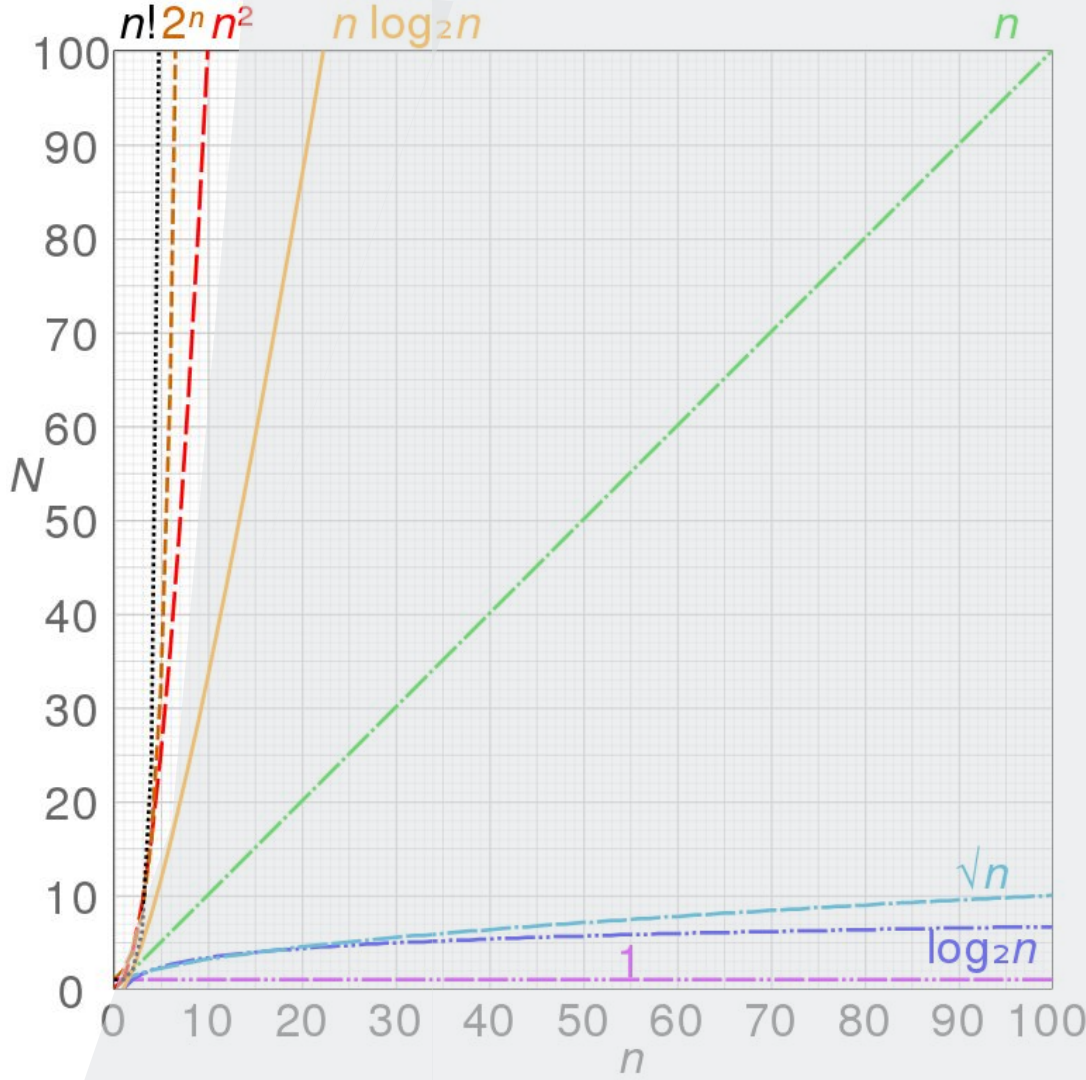


# SER 321

## When to Distribute

When should  
we *consider*  
distributing?

Super Duper Extra Extra  
Large Orders of Magnitude!

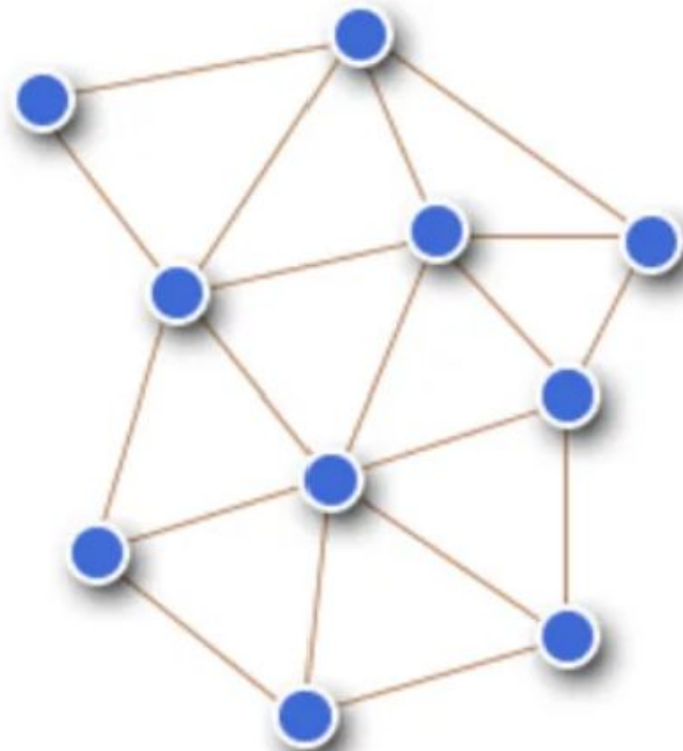


# SER 321

## Distributed Issues

Remember that we are operating in *reality*

- Nodes *will* fail
- Web of nodes *will constantly* change
- Network is not *always* reliable
- Latency is *always present*
- The path traversed *changes*
- Some resources *must be shared*
- *You* need to prevent the pitfalls!
  - No deadlocks
  - No starvation
  - No error states



# SER 321

## Consensus

“General agreement or trust amongst a group”

## What is Consensus?

Who's in charge or keeping the beat



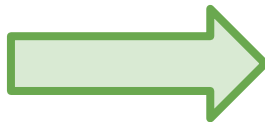
Leader Election

Check your work with a neighbor



Result Verification

Verify and maintain my copy of the data



Log Replication

Do I want to let you into my network



Validate Nodes

# SER 321

## Middleware

We have been:

Serializing  
Messages

Sending  
Messages

Parsing  
Messages

Handle  
Messages

*Check out the recording for the discussion!*

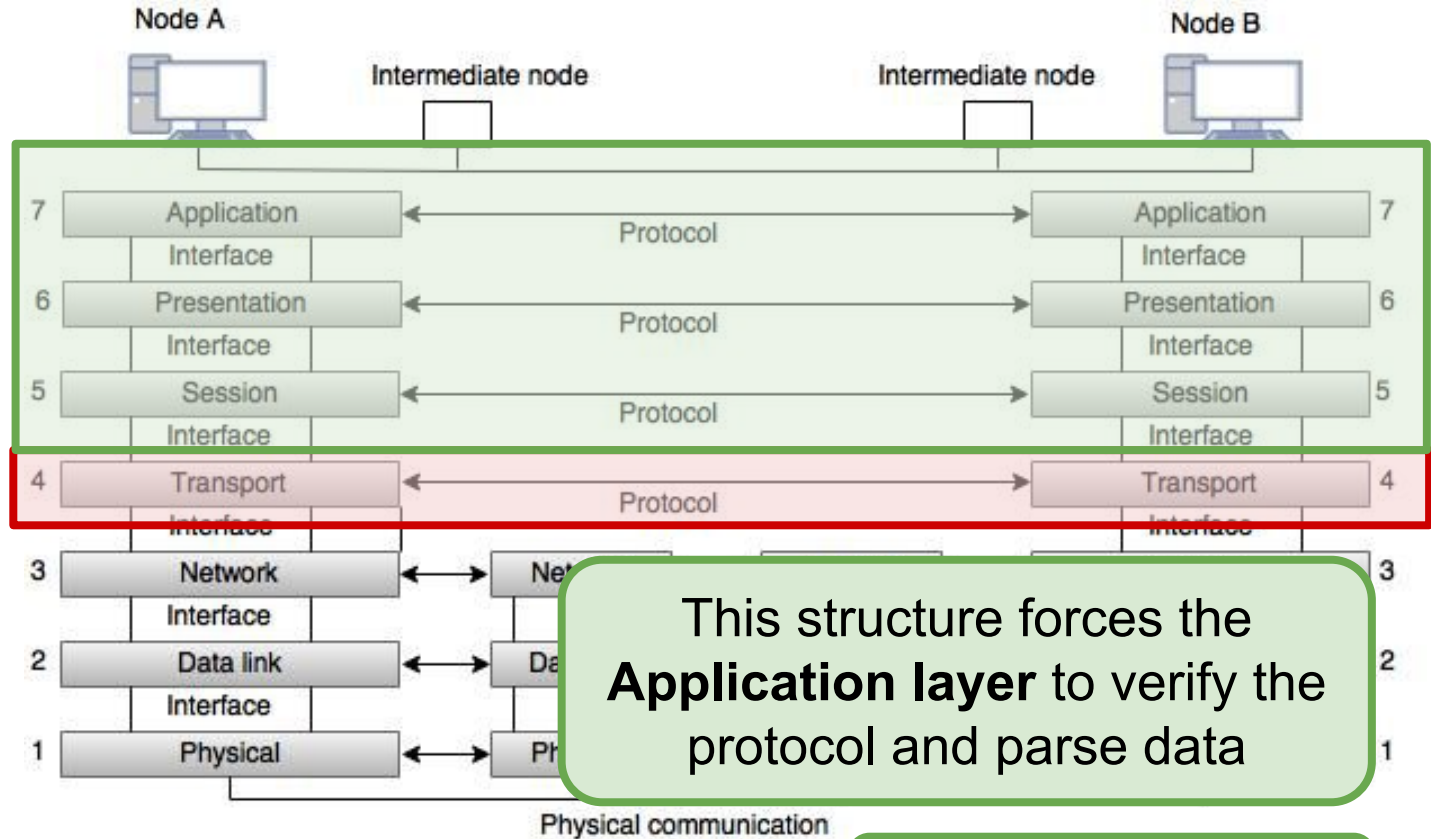


Fig: OSI Model

Not really its job...



*Check out the recording for the discussion!*

## SER 321 Middleware

With Middleware:

Serializing  
Messages

Sending  
Messages

Parsing  
Messages

Handle  
Messages

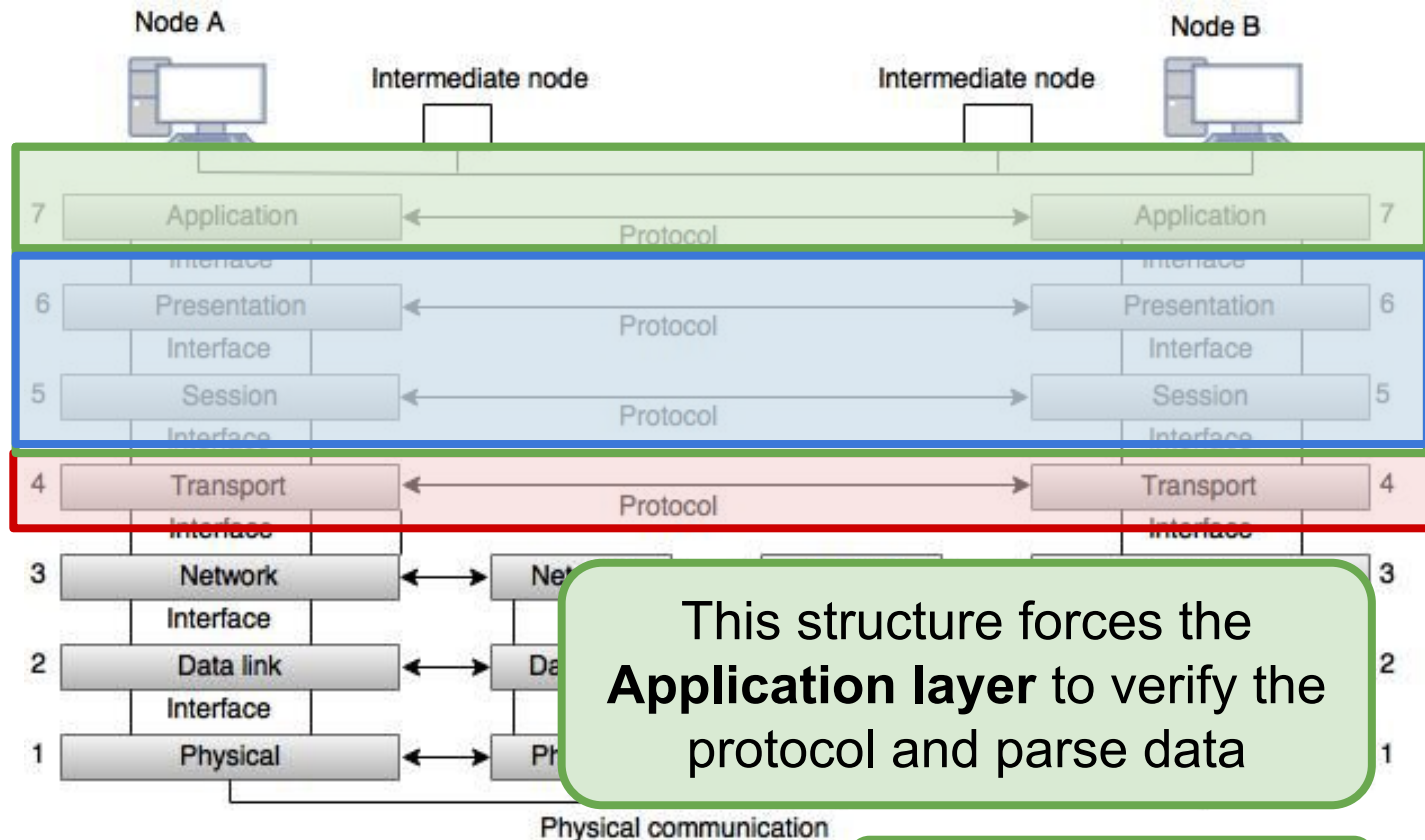


Fig: OSI Model

Not really its job...

*Check out the recording for the discussion!*

## SER 321 Middleware

With Middleware:

Serializing  
Messages

Sending  
Messages

Parsing  
Messages

Handle  
Messages

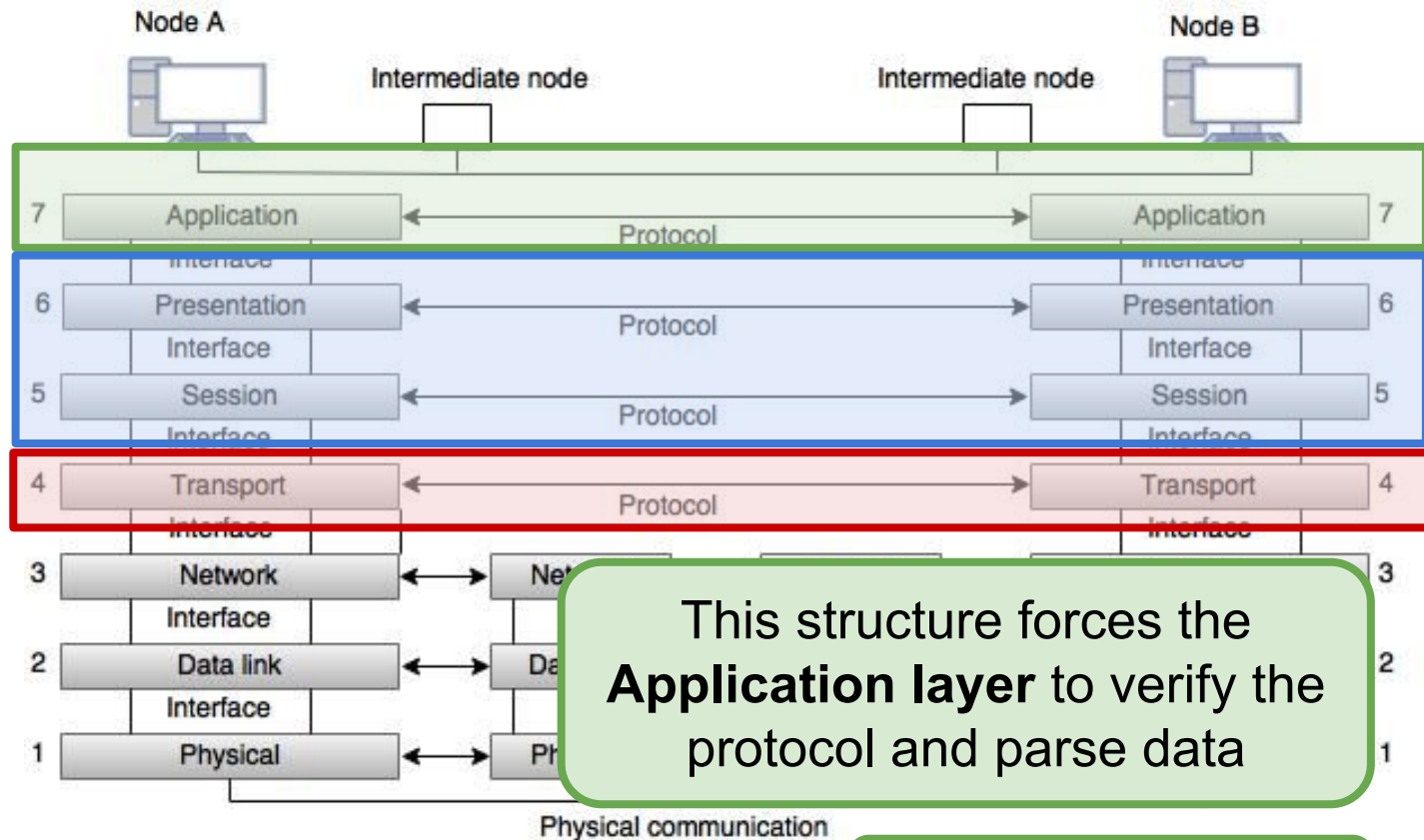


Fig: OSI Model

Not really its job...

*Check out the recording for the discussion!*

## SER 321 Middleware

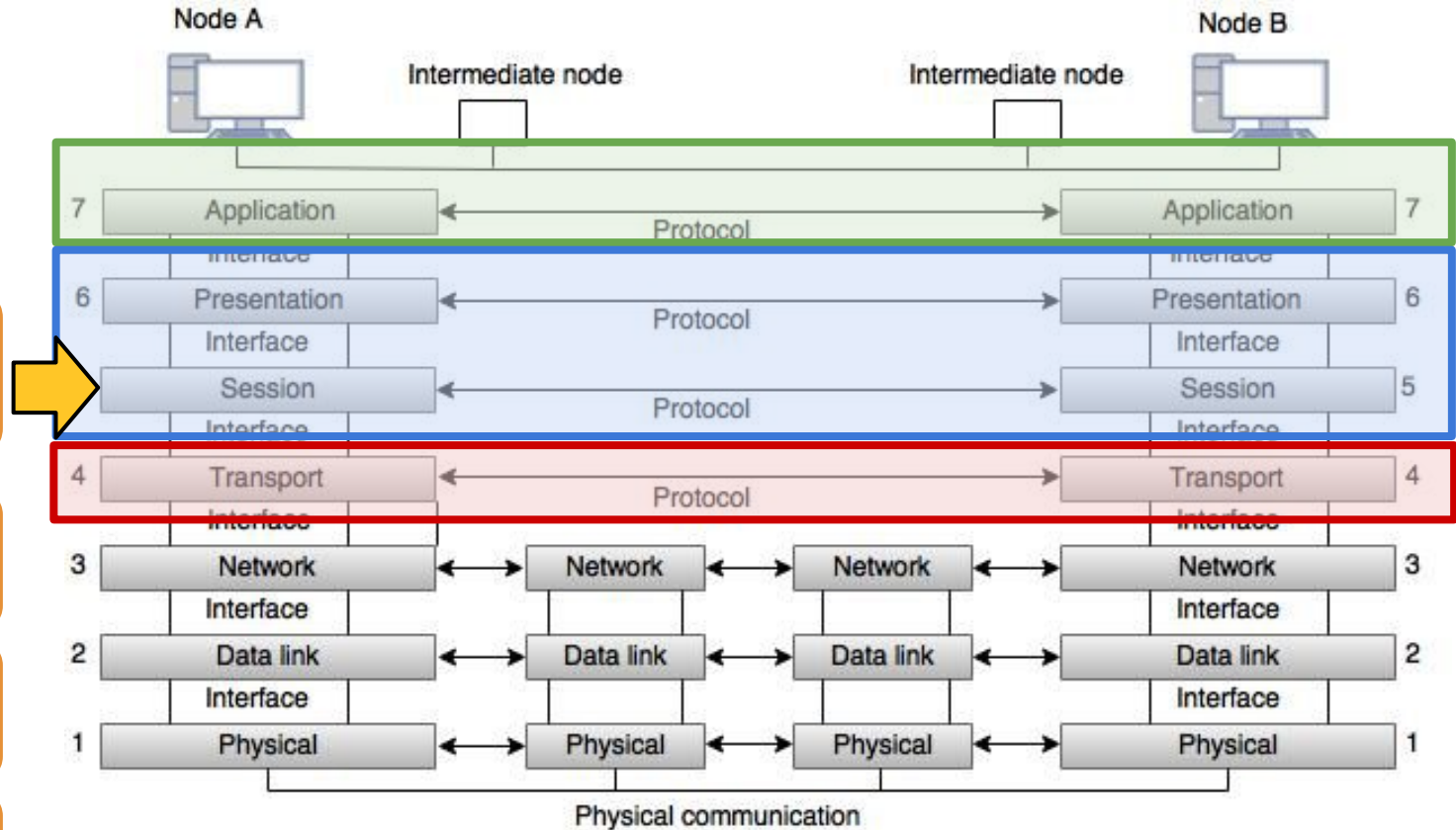
Middleware:

*Session Layer  
Responsibilities:*

Authentication

Authorization

Session  
Management



**Fig: OSI Model**

*Check out the recording for the discussion!*

## SER 321 Middleware

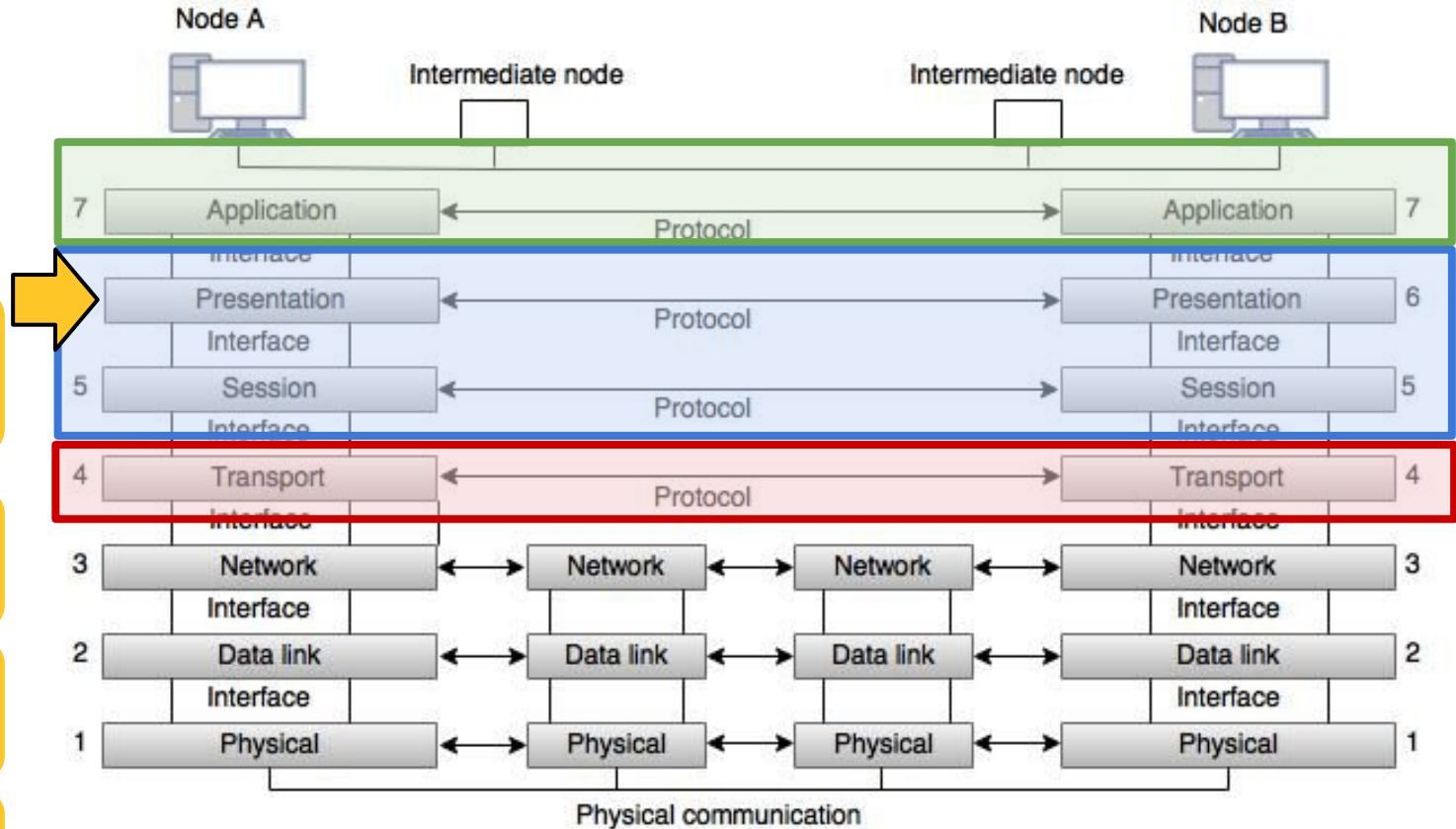
Middleware:

*Presentation  
Layer  
Responsibilities:*

Translation

Compression

Encryption



**Fig: OSI Model**

**SER 321**

**Middleware Benefits**

Why do we care?

Agility

Reusability

Efficiency

Cost  
Effectiveness

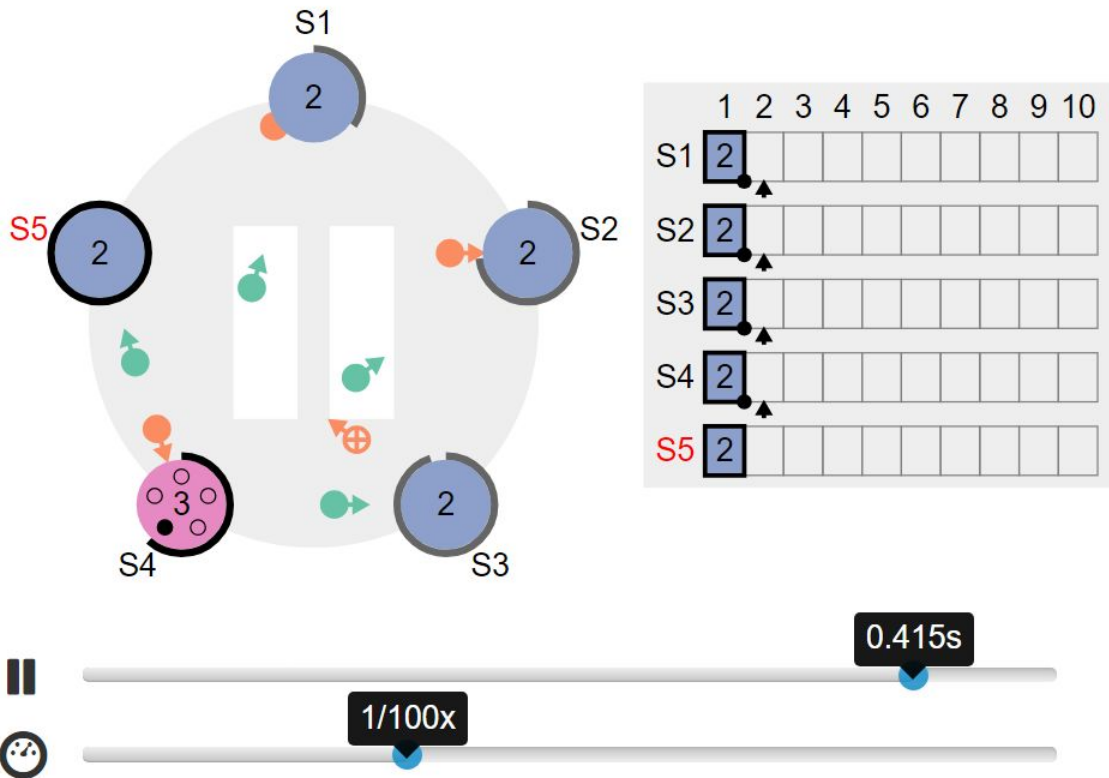
Portability

**SER 321**

**RAFT**

How do we  
feel about  
Consensus?

RAFT



**SER 321**

**Scratch Space**

# Questions?



## Survey:

<http://bit.ly/ASN2324>





## Upcoming Events

### SI Sessions:

- Tuesday, December 3rd at 10:00 am MST - **Q&A Session**

### Review Sessions:

- Sunday, December 1st at 7:00 pm MST - **2 hour Review Session**
- Tuesday, December 3rd at 10:00 am MST - **Q&A Session**

# More Questions?

Check out our other resources!

tutoring.asu.edu



## Academic Support

Academic Support Network (ASN) provides a variety of free services in-person and online to help currently enrolled ASU students succeed academically.

### Services



#### Subject Area Tutoring

Need in-person or online help with math, science, business, or engineering courses? Just hop into our Zoom room or drop into a center for small group tutoring. We'll take it from there.

[Need help using Zoom?](#)

[View the tutoring schedule](#)

[View digital resources](#)

Go to Zoom



#### Writing Tutoring

Need help with undergraduate or graduate writing assignments? Schedule an in-person or online appointment, access your appointment link, or wait in our drop-in queue.

[Access your appointment link](#)

[Access the drop-in queue](#)

Schedule Appointment



#### Online Study Hub

Join our online peer communities to connect with your fellow Sun Devils. Engage with our tools to search our bank of resources, videos, and previously asked questions. Or, ask our Tutorbot questions.

Now supporting courses in Math, Science, Business, Engineering, and Writing.

Online Study Hub

1-

Go to Zoom

2-

[Need help using Zoom?](#)

[View the tutoring schedule](#)

[View digital resources](#)



1. Click on 'Go to Zoom' to log onto our Online Tutoring Center.
2. Click on 'View the tutoring schedule' to see when tutors are available for specific courses.

# More Questions?

## Check out our other resources!

[tutoring.asu.edu/online-study-hub](https://tutoring.asu.edu/online-study-hub)

 **Academic Support Network**

 [Services](#)  [Faculty and Staff Resources](#) [About Us](#) 

[University College](#)

## Online Study Hub

Online peer communities for students and tutors, YouTube channels, and Tutorbots.



### What are online peer communities?

Individual courses have an online peer community that allows you to connect with your peers to post and answer questions and to develop study groups.



### How can tutoring center videos help?

Videos can help supplement the learning you're doing in and outside of class and include step-by-step methods for how to understand concepts.



### How does the Tutorbot work?

You can ask the Tutorbot questions about course concepts and the Tutorbot will recommend additional resources and examples to help address your questions.

Select a subject

- Any -

Apply



Academic Support Network



[Services](#) 

[Faculty and Staff Resources](#)

[About Us](#) 

[University College](#)

Select a subject

- Any -

Apply

Business

### ACC 231

Uses of Accounting Info I

 [Peer Community](#)

### ACC 241

Uses of Accounting Info II

 [Peer Community](#)

### CIS 105

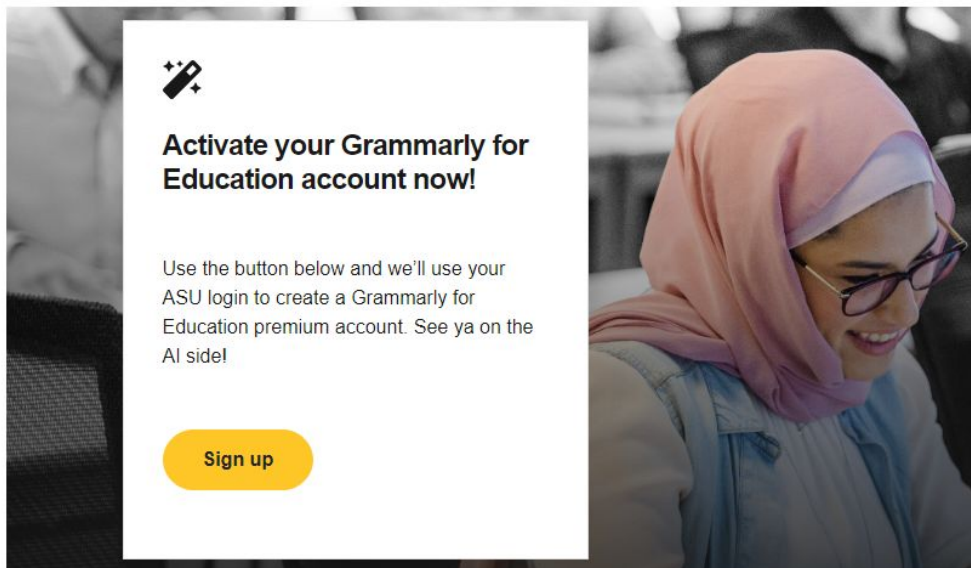
Computer Applications and Information Technology

 [Peer Community](#)

Don't forget to check out the Online Study Hub for additional resources!

# Expanded Writing Support Available

Including Grammarly for Education, at no cost!



[tutoring.asu.edu/expanded-writing-support](https://tutoring.asu.edu/expanded-writing-support)

\*Available slots for this pilot are limited

## Additional Resources

- [Course Repo](#)
- [Gradle Documentation](#)
- [GitHub SSH Help](#)
- [Linux Man Pages](#)
- [OSI Interactive](#)
- [MDN HTTP Docs](#)
  - [Requests](#)
  - [Responses](#)
- [JSON Guide](#)
- [org.json Docs](#)
- [javax.swing package API](#)
- [Swing Tutorials](#)
- [Dining Philosophers Interactive](#)
- [Austin G Walters Traffic Comparison](#)
- [RAFT](#)