SER 321 A Session

Exam Review Session

Thursday, February 27th 2025

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
- 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

110 minutes

Similar to the quizzes

Make sure to look at the Study Guide!

Exam Info Page

Opens: Tuesday March 3rd @ 12:00 AM

Closes: Tuesday March 4th @ 11:59:59 PM

SER 321
Review Requests

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

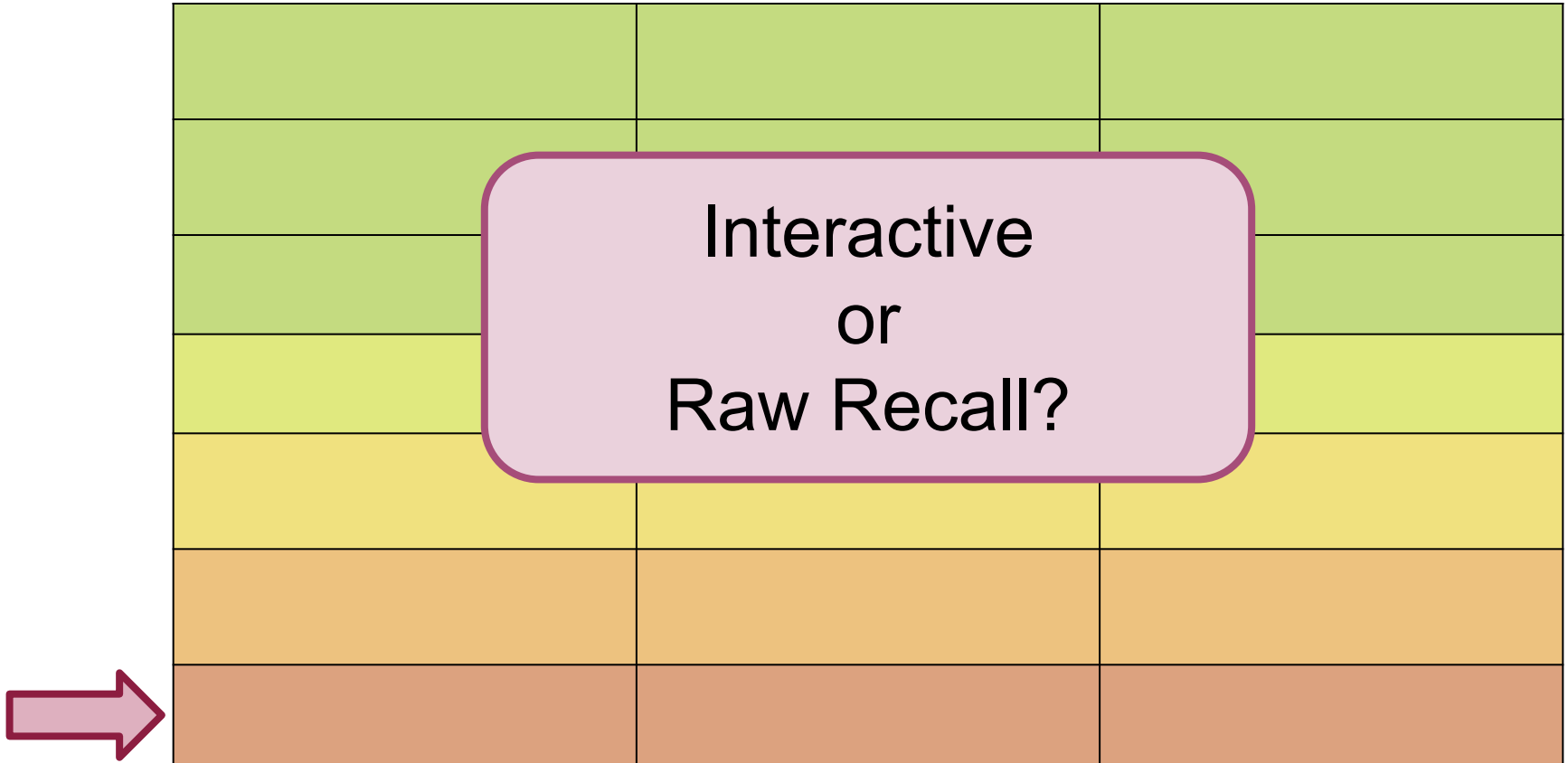
Drop a concept in the chat and we can cover it next!

SER 321
OSI Model

Unit

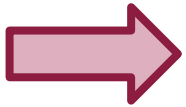
Layer

What we are *really* talking about



SER 321

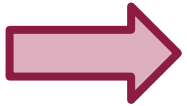
OSI Model



Unit	Layer	What we are *really* talking about
Bits	Physical	Signal, Binary transmission

SER 321

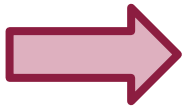
OSI Model



Unit	Layer	What we are *really* talking about
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

SER 321

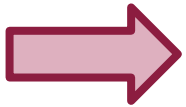
OSI Model



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

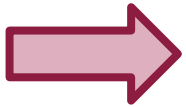
OSI Model



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

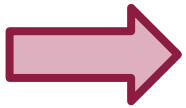
OSI Model



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

SER 321

TCP & UDP

What are the main differences?

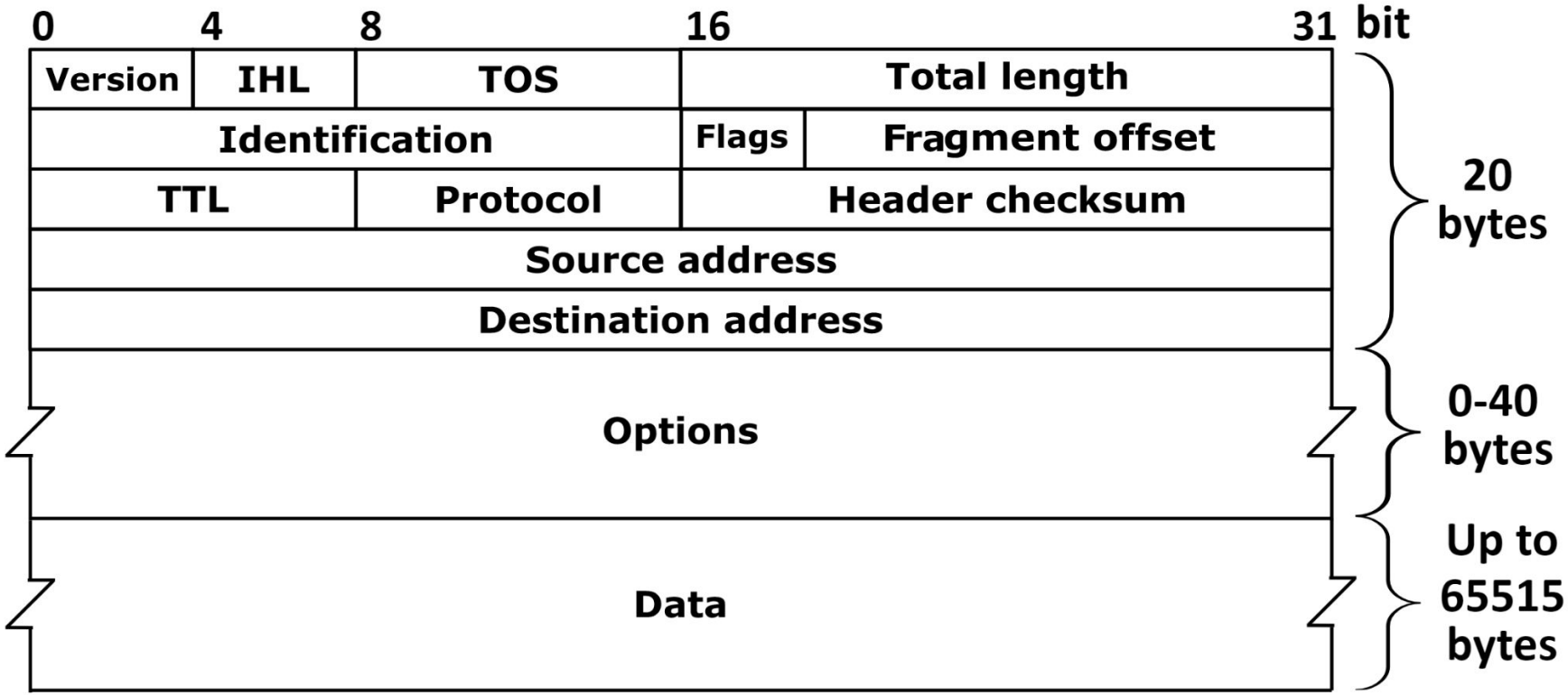
TCP	UDP

SER 321

Headers

What type of header is this?

IP Header



SER 321
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				CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if *data offset* > 5. Padded at the end with "0" bits if necessary.)																															
⋮	⋮																																
56	448																																

SER 321
Headers

TCP or UDP Header?

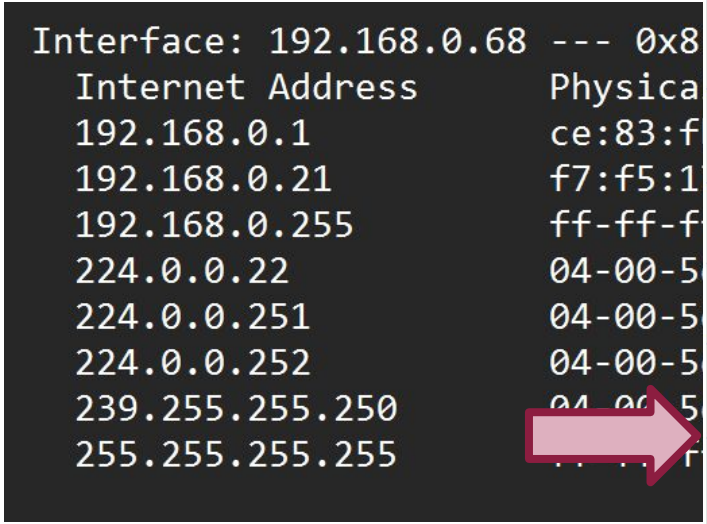
Offset	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Source Port*																*Destination Port*															
4	32	*Length*																*Checksum*															
8	64	*Data*																															
12	96																																
⋮	⋮																																

SER 321

Network Concept Checks

What is an ARP table?

Where does it fit in the OSI model?



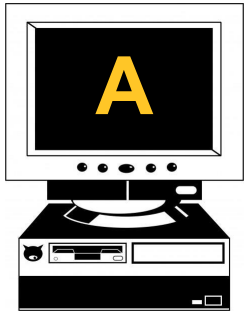
Data	Application	HTTP(s), SMTP, FTP, IMAP, POP, etc.
Data	Presentation	Translation, compression, encryption
Data	Session	AuthN, authZ, session mgmt
Segment	Transport	TCP/UDP
Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

Note - numbers have been modified

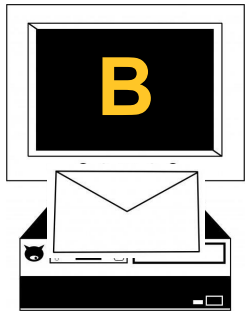
SER 321

Network Concept Checks

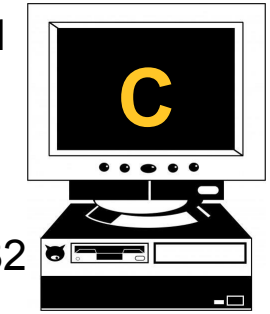
What is NAT?



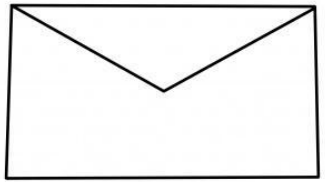
196.122.10.80



196.122.10.81



196.122.10.82

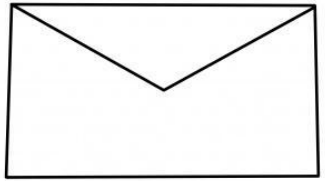


196.122.10.81

128.148.36.179

Thinking…

TO: Device B



128.148.36.179

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

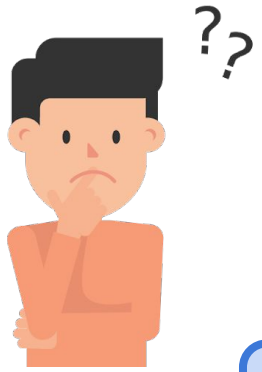
Given the following URL, identify the…

Host

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

SER 321

HTTP(s)



Stateful OR Stateless

Synchronous OR Asynchronous

SER 321
HTTP Requests

MDN HTTP Request Methods

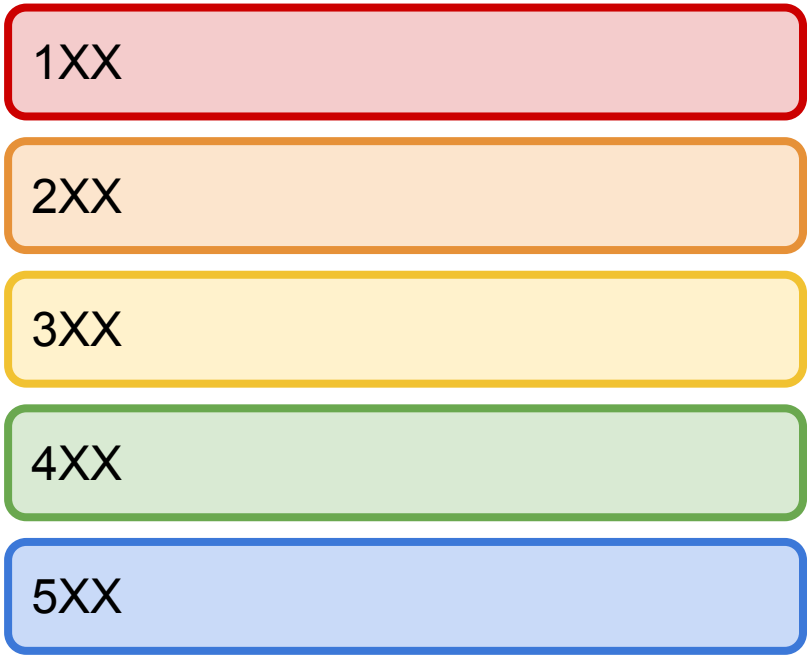
What's the difference?

1. GET
2. POST
3. PUT
4. DELETE

SER 321

HTTP Responses

Status Codes →



SER 321

Socket Properties

Sockets allow our client and server to communicate!

Need to define **3 properties** before usage

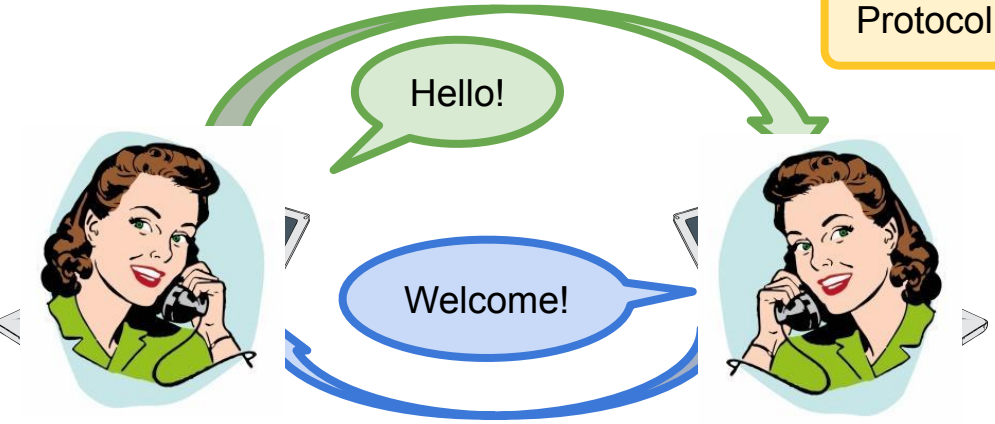
Location	IP or DNS	142.251.46.206	www.google.com
Connection Semantics	TCP or UDP	Connection Oriented	Connectionless
Message Format	Protocol Specs	Synchronous	Asynchronous
		Stateless	Stateful
		Binary	Text
		Headers	No Headers

SER 321
Socket Properties

Sockets allow our client and server to communicate!

d to define **3 properties** before usage

Person	IP or DNS	142.251.46.206	www.google.com
Conversation Flow	TCP or UDP	Connection Oriented	Connectionless
Conversation Content	Protocol Specs	Synchronous	Asynchronous
		Stateless	Stateful
		Binary	Text
		Headers	No Headers



SER 321

Socket Protocol Types

Two Main Conversation Models

1. Client Request

2. Server Response

Pull/Polling Model

Client *pulls* info from the server

Client *polls* the server for info

SER 321

Socket Protocol Types

Two Main Conversation Models

1. Server sends update

2. Client acknowledges

Push Model

Server *pushes* info to client

Push notifications

SER 321

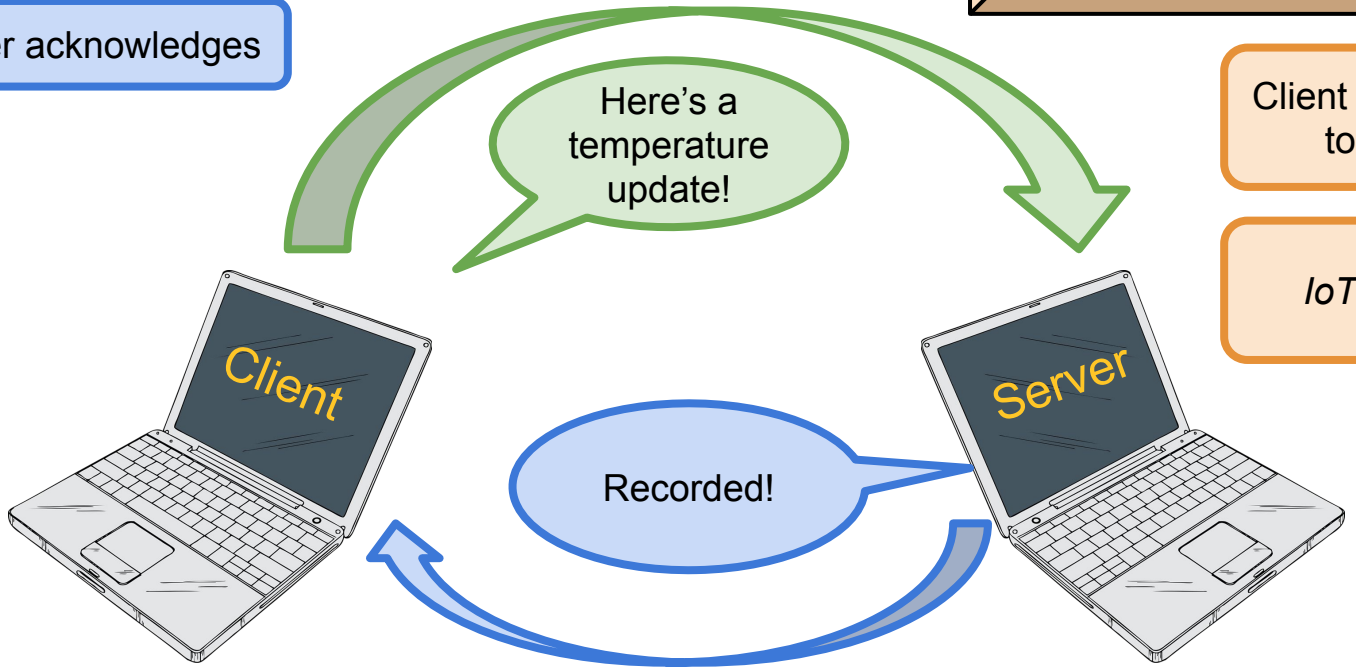
Socket Protocol Types

Two Main Conversation Models

Client Push Model

1. Client sends update

2. Server acknowledges



Client *pushes* info to Server

IoT sensors

SER 321
Client Socket

Steps for the **Client Socket**

1.
2.
3.
4.
5.
6.
7.
8.

Interactive or Raw Recall?

SER 321
Server Socket

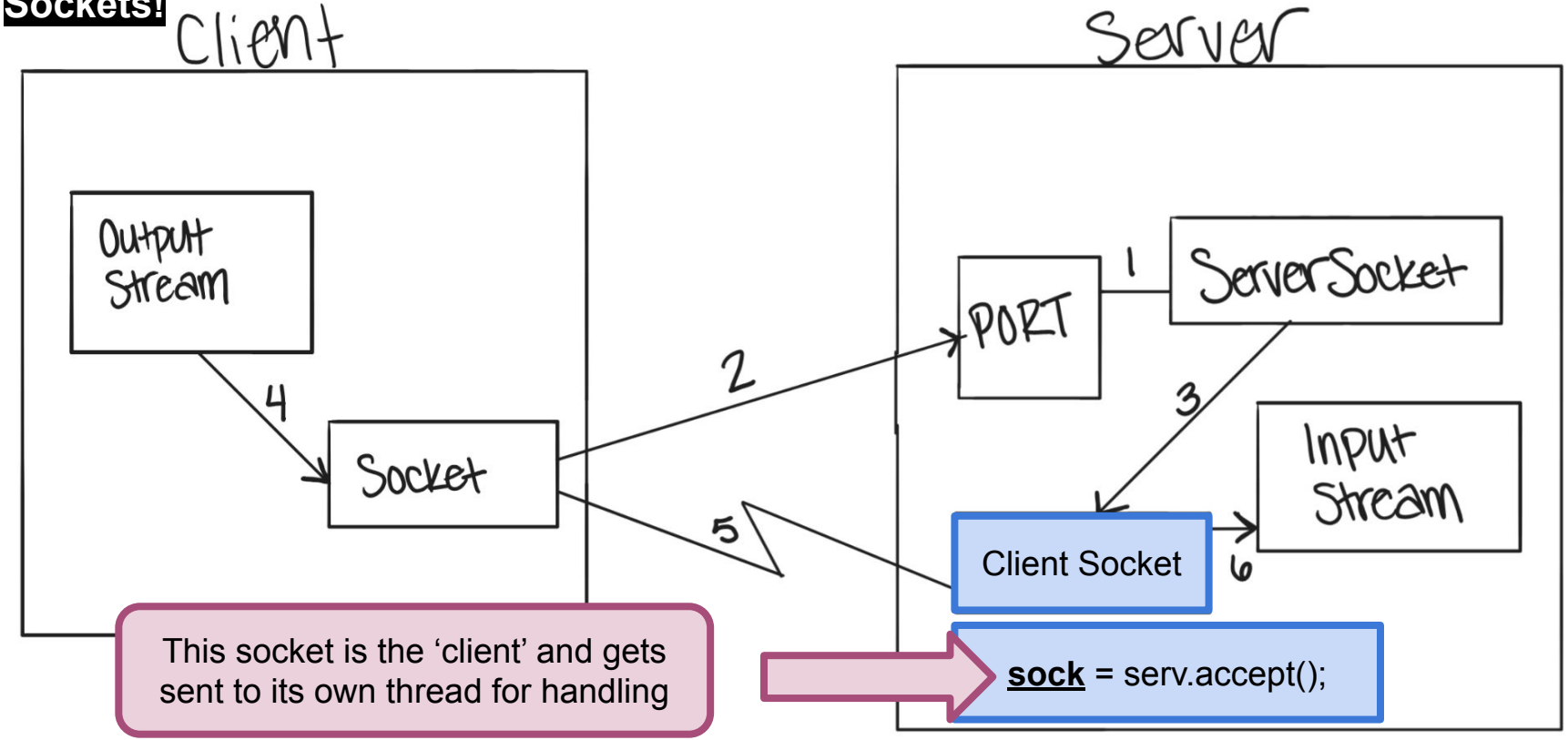
Steps for the **Server Socket**

1.
2.
3.
4.
5.
6.
7.
8.
9.

SER 321

Sockets!

Why do we keep reviewing this?

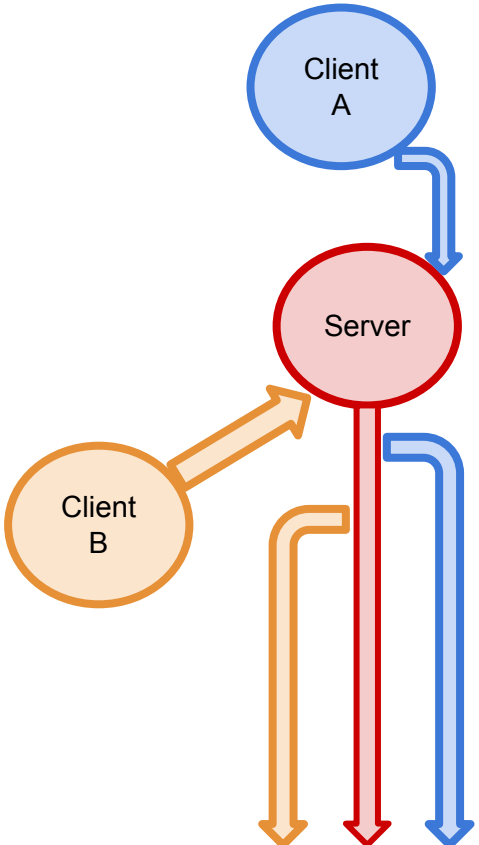


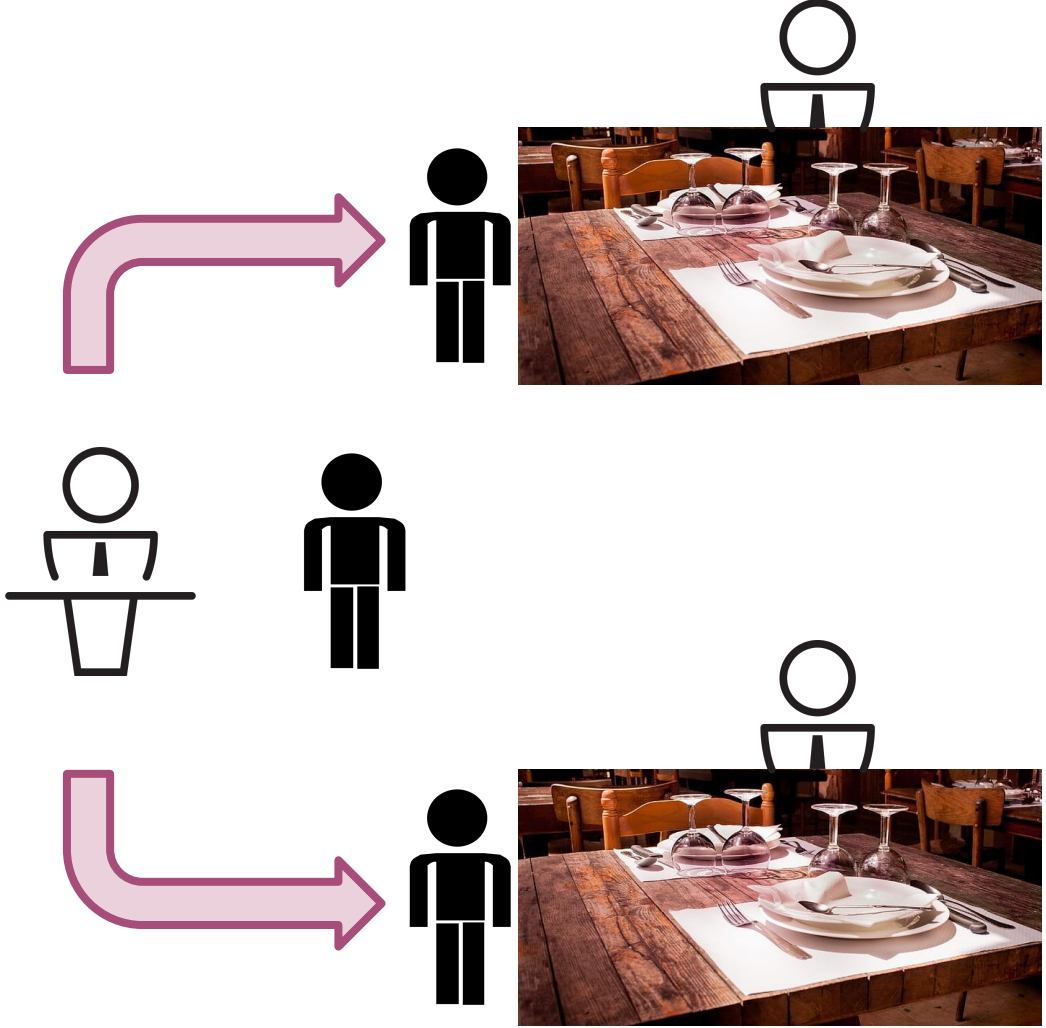
This socket is the ‘client’ and gets sent to its own thread for handling

sock = serv.accept();

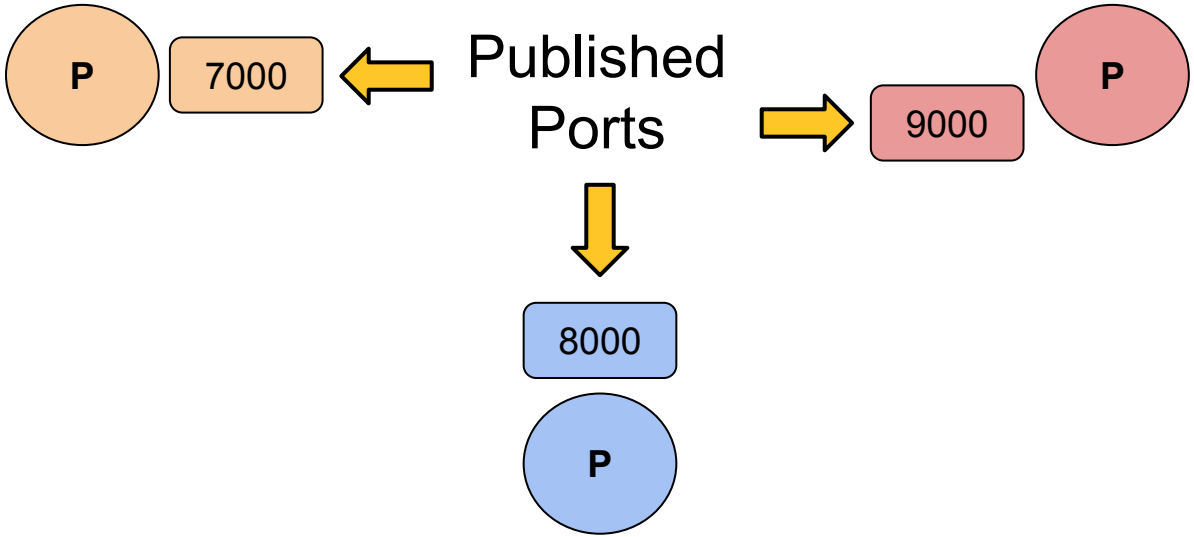
SER 321

Threaded Sockets





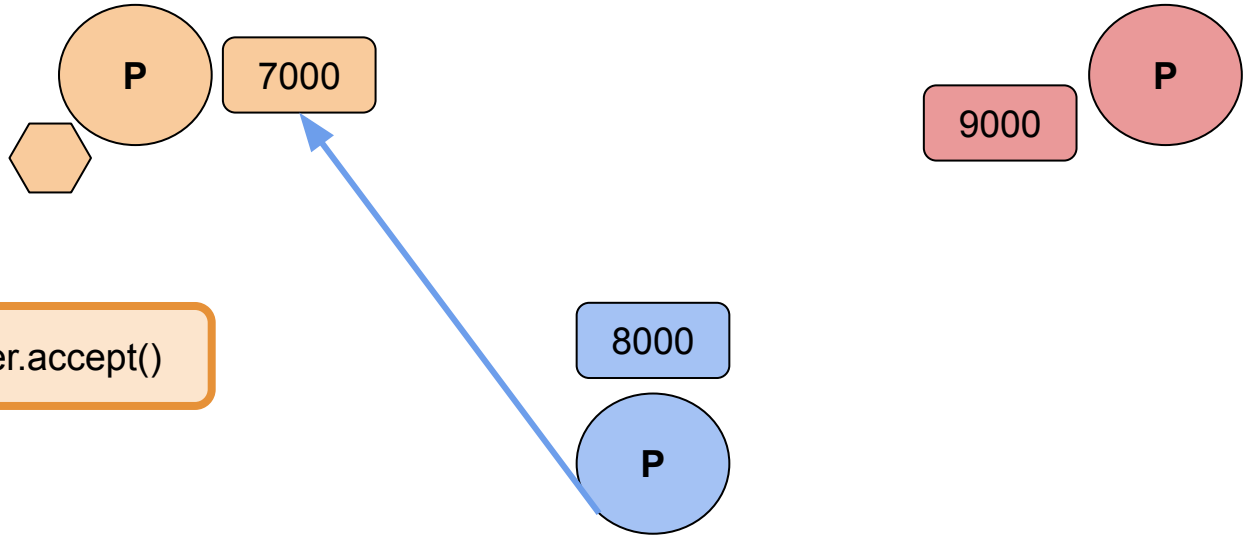
Remember that the OS allocates a new port for the client socket!



SER 321

P2P Communication

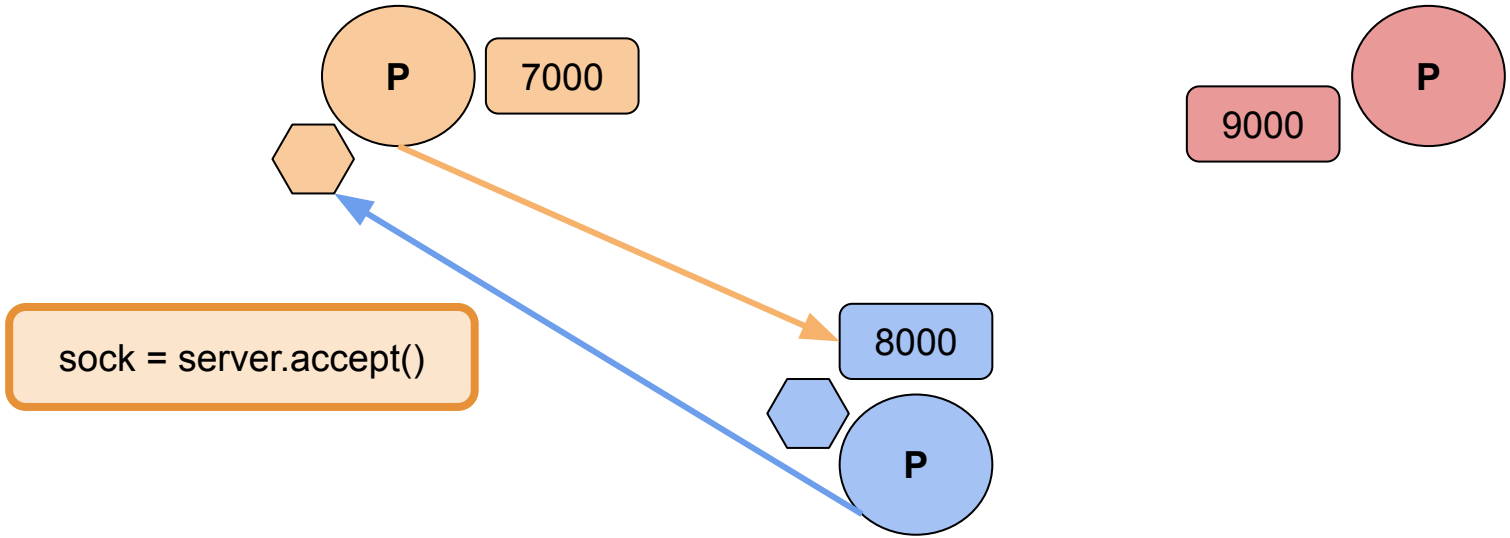
Remember that the OS allocates a new port for the client socket!



SER 321

P2P Communication

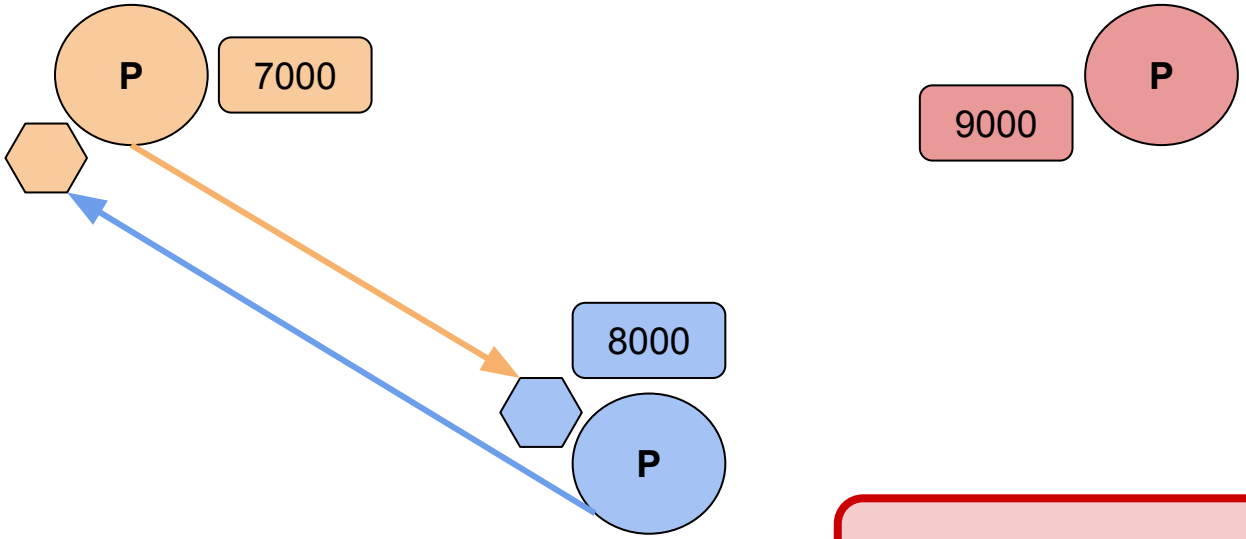
Remember that the OS allocates a new port for the client socket!



SER 321

P2P Communication

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

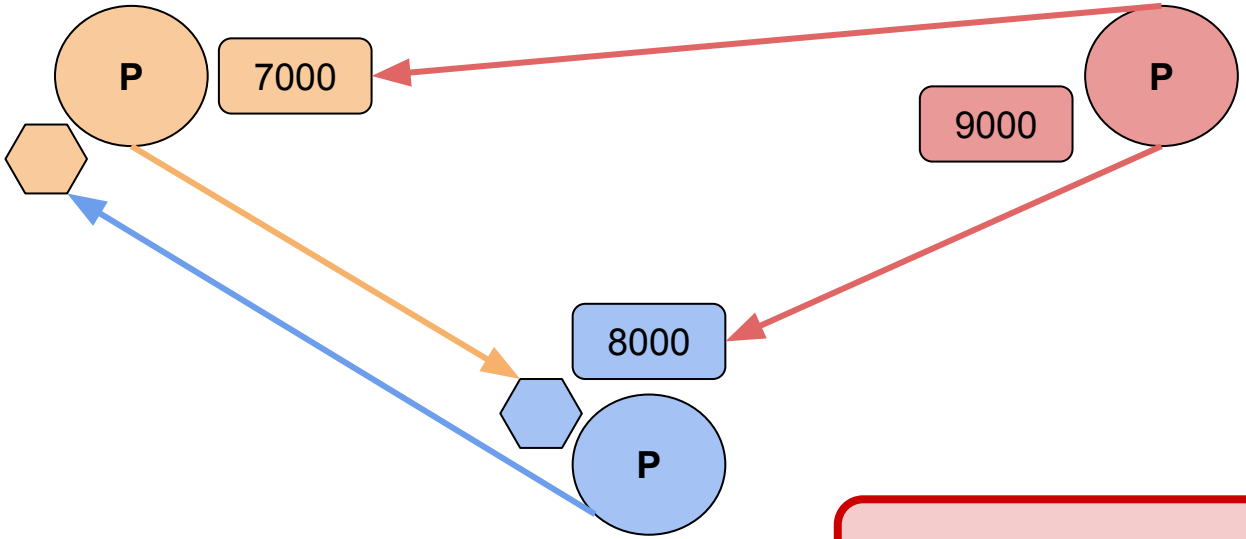


What about Peer 9000?

SER 321

P2P Communication

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

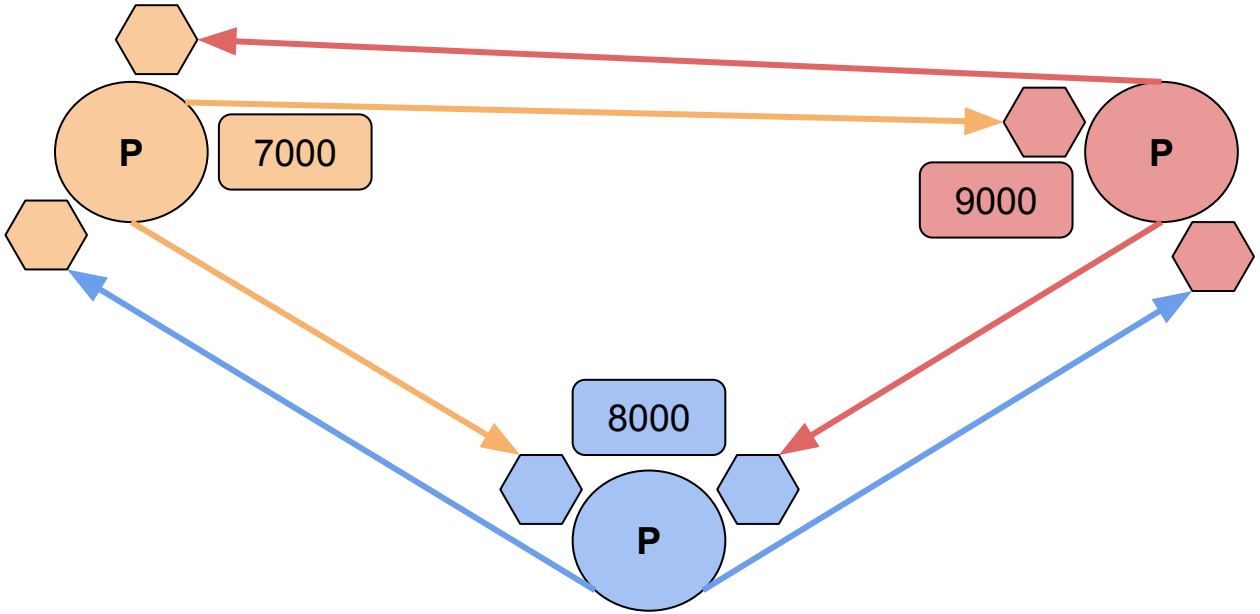


What about Peer 9000?

SER 321

P2P Communication

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



SER 321
Threaded Pitfalls

Starvation	More than one thread accesses a single resource at the same time
Deadlock	A thread never gains access to the resource it needs
Race Condition	A thread is only able to acquire some of the resources it needs

SER 321
Threaded Pitfalls

Starvation	More than one thread accesses a single resource at the same time
Deadlock	A thread never gains access to the resource it needs
Race Condition	A thread is only able to acquire some of the resources it needs

SER 321
Threaded Pitfalls

What's the difference?

Starvation	vs.	Deadlock
A thread never gains access to the resource it needs		A thread is only able to acquire some of the resources it needs
Waiting to access the ***CPU***		Waiting to access another ***resource***
Ready to go; never gets a chance		***Not*** ready to go

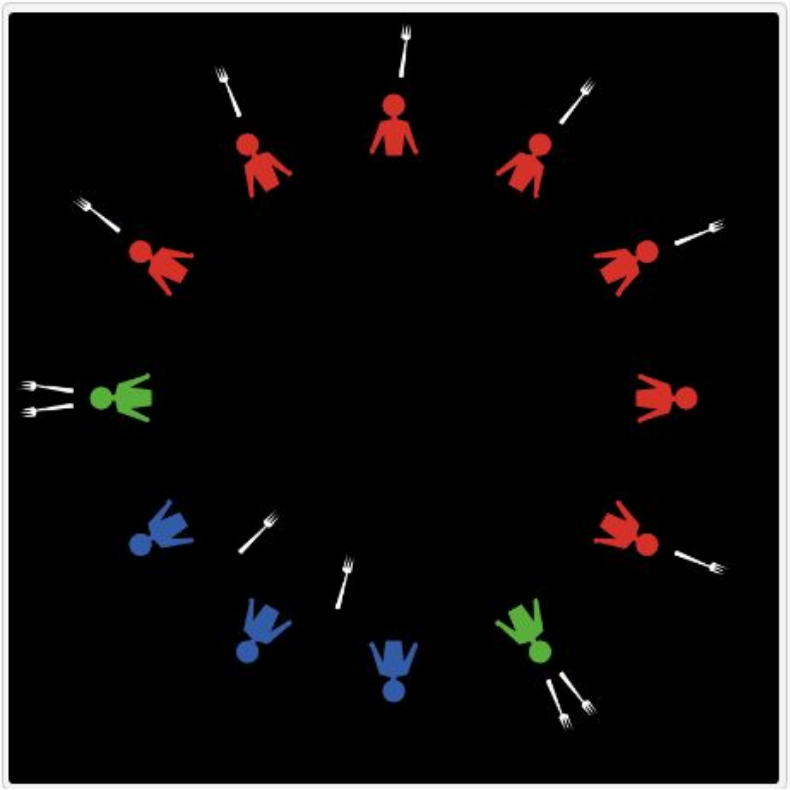
SER 321
Dining Philosophers

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?



org.json Docs

JSON Guide

SER 321

JSON Structure

Data is stored in…

Name:Value pairs → AKA → Members

"Katie"

"student" : "Katie"

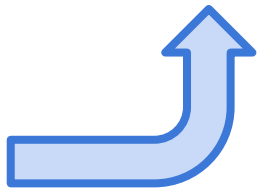
What uses curly braces?

Objects

{ }

What do Objects contain?

Members



org.json Docs

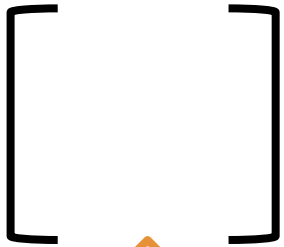
JSON Guide

SER 321

JSON Structure

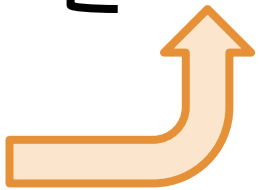
What uses brackets?

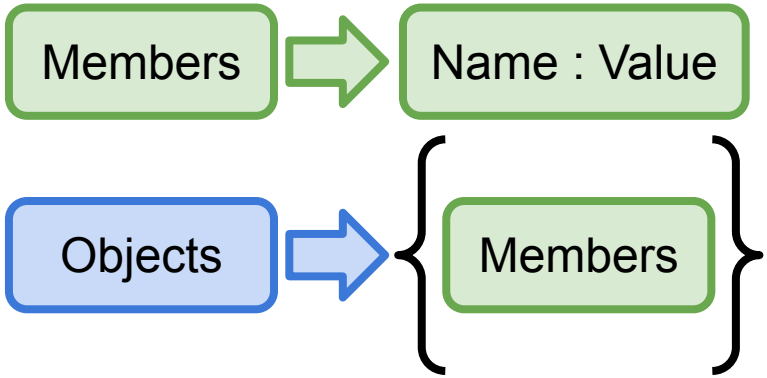
Arrays



What do Arrays contain?

Any ***Valid*** Value





org.json Docs

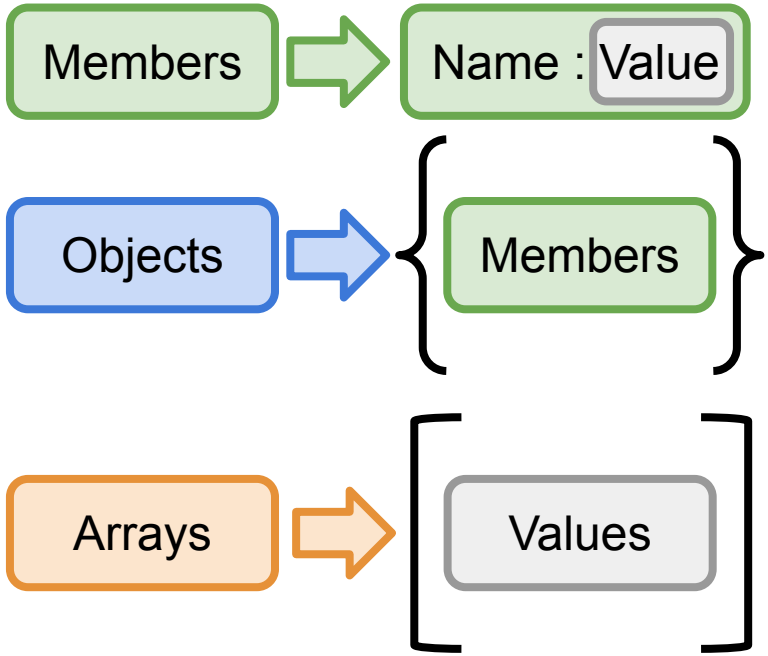
JSON Guide

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

SER 321
Serialization

Binary

Text

What does this imply?

Who uses ***BINARY***?

Text JSON

Binary Java Object Serialization

Binary Protocol Buffers

Text XML

SER 321
Serialization

Streams and their types

Generic Superclass

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

Buffered Stream

Bytes

Data Stream

Primitive DATA Types

Object Stream

Java Objects

org.json Docs

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"
      }
    ]
  }
}
```


JSON Guide

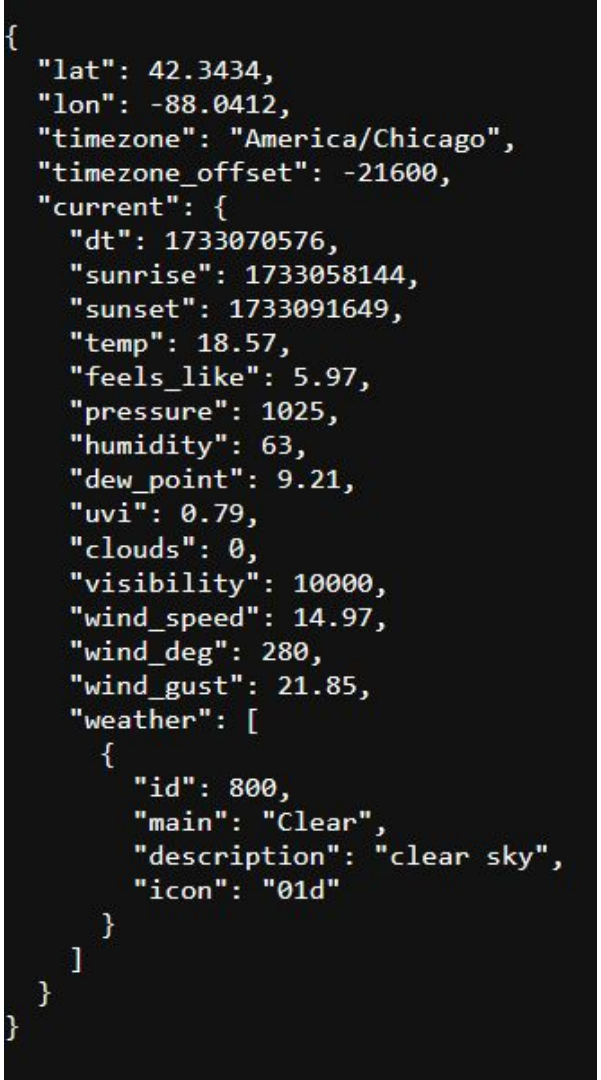
org.json Docs

JSON Guide

SER 321

JSON Practice

```
JSONObject json =
```



```
{
  "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"
      }
    ]
  }
}
```

How would we…

Check for the timezone member?

```
boolean hasTimezone =
```

Get the timezone?

```
String timezone =
```

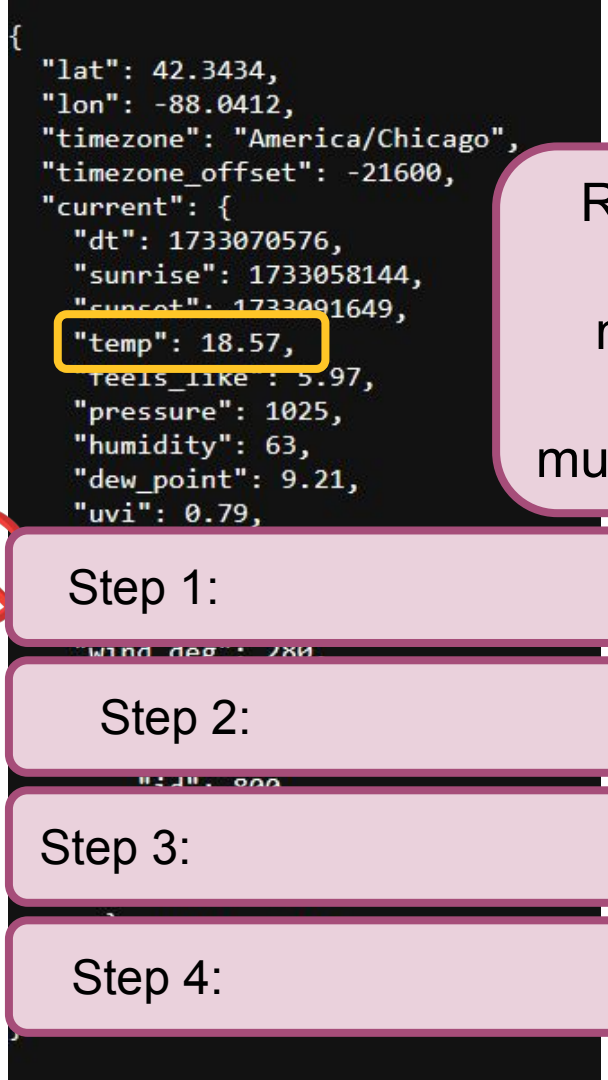
org.json Docs

JSON Guide

SER 321

JSON Practice

```
JSONObject json =
```



How would we…

Obtain the temp value?

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

Recall that nested members require multiple steps!

Step 1:

Step 2:

Step 3:

Step 4:

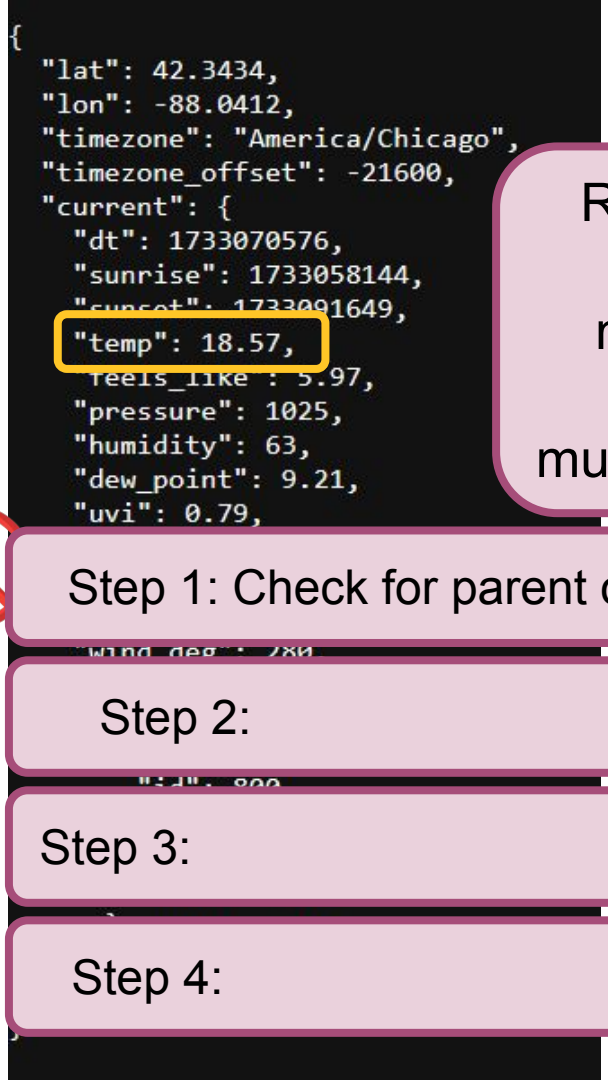
org.json Docs

JSON Guide

SER 321

JSON Practice

```
JSONObject json =
```



How would we…

Obtain the temp value?

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

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

}
```

Recall that nested members require multiple steps!

Step 1: Check for parent object

Step 2:

Step 3:

Step 4:

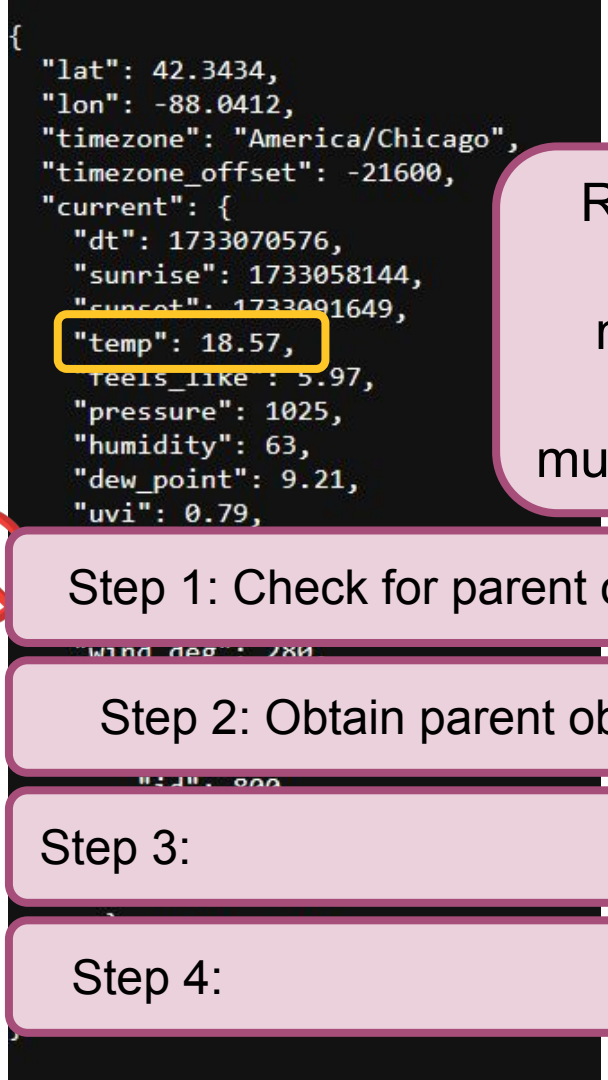
org.json Docs

JSON Guide

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.getObject("current");


 }
```

Recall that nested members require multiple steps!

Step 1: Check for parent object

Step 2: Obtain parent object

Step 3:

Step 4:

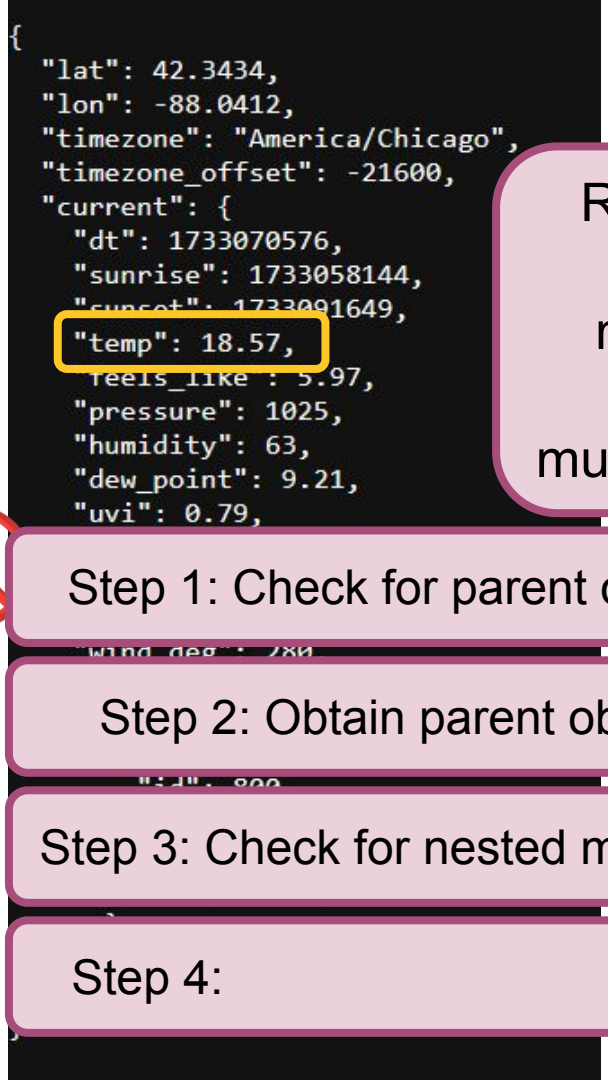
org.json Docs

JSON Guide

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.getObject("current");
    if (current.has("temp")) {
       temp = current.getString("temp");
}
```

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:

org.json Docs

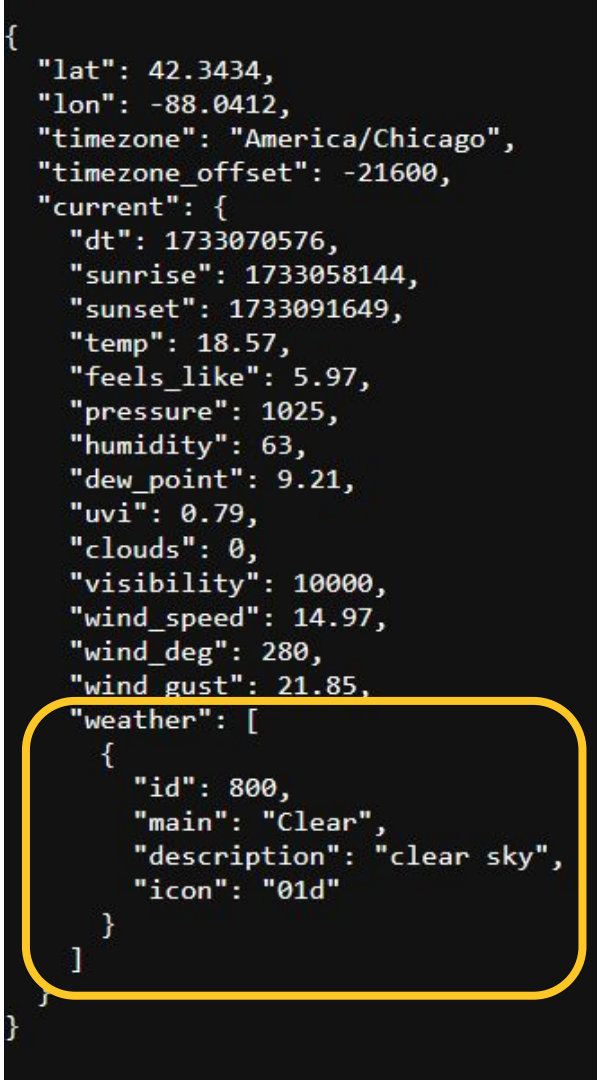
JSON Guide

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"
      }
    ]
  }
}
```

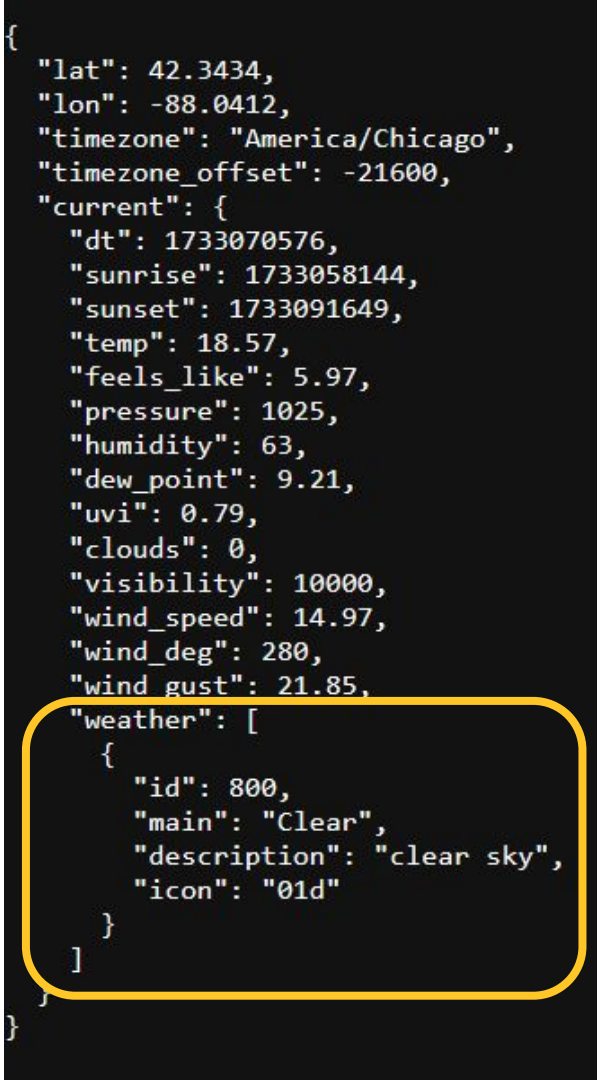
org.json Docs

JSON Guide

SER 321

JSON Practice

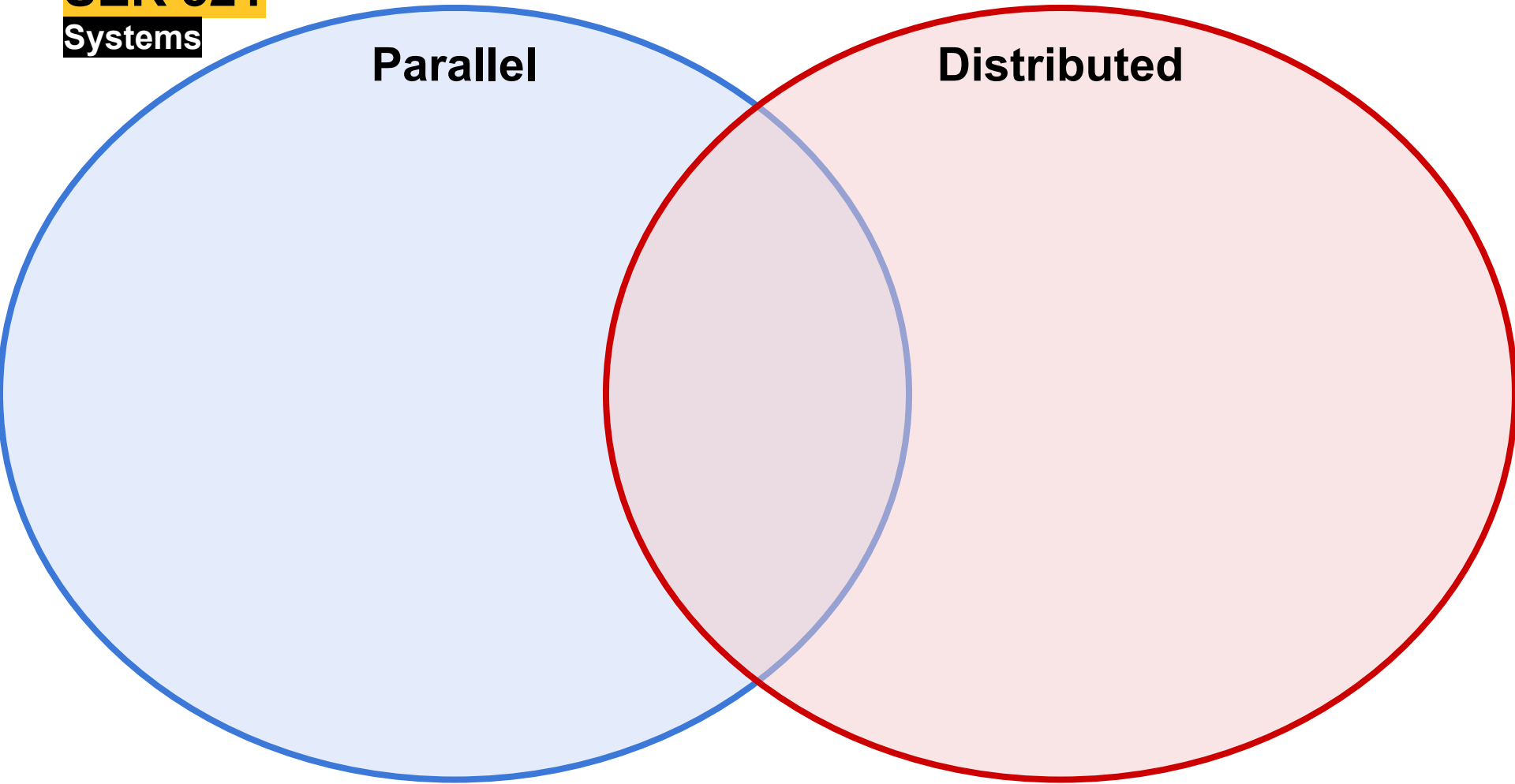
JSONObject json =



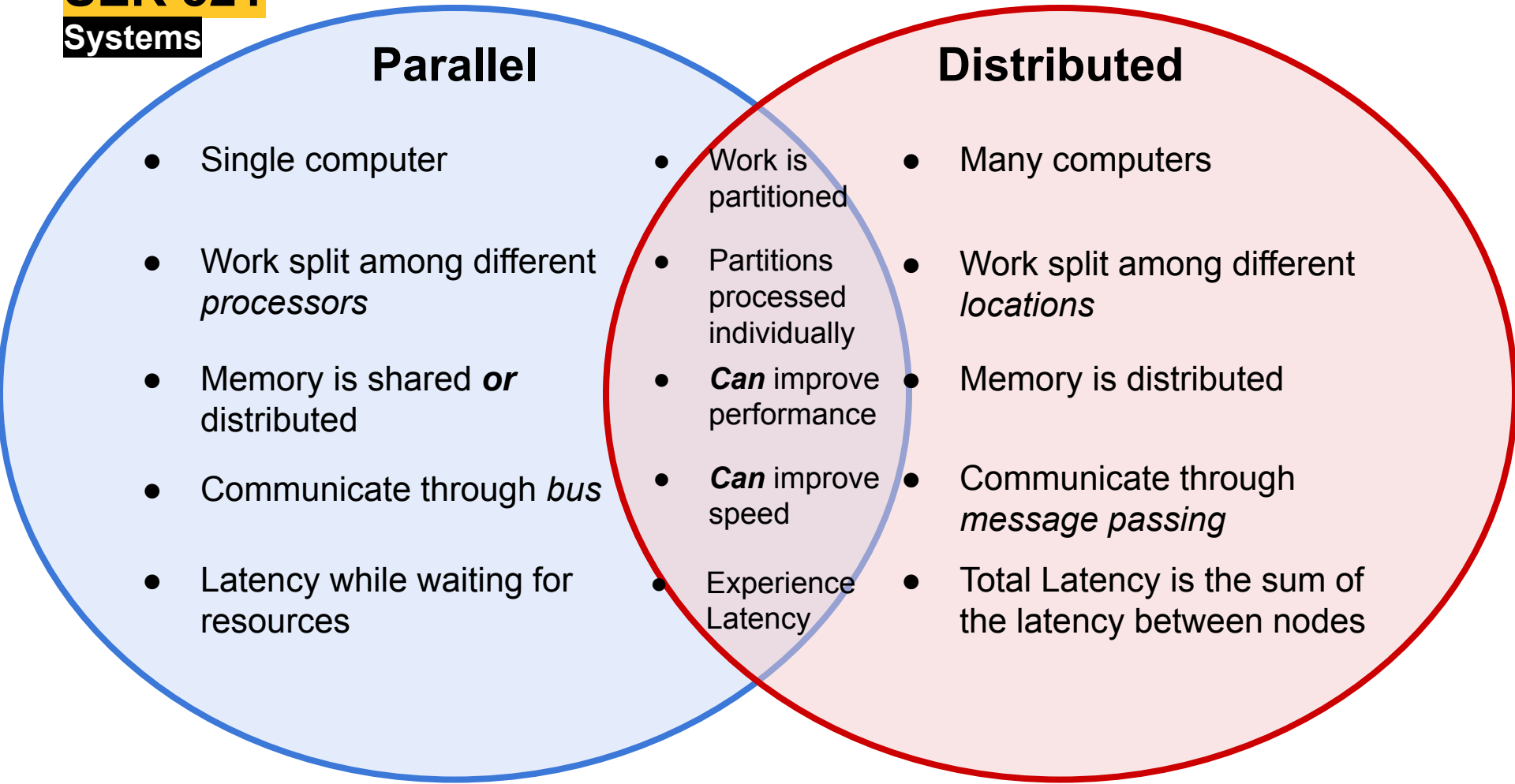
How would we create the "weather" object?

```
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());
```

SER 321
Systems



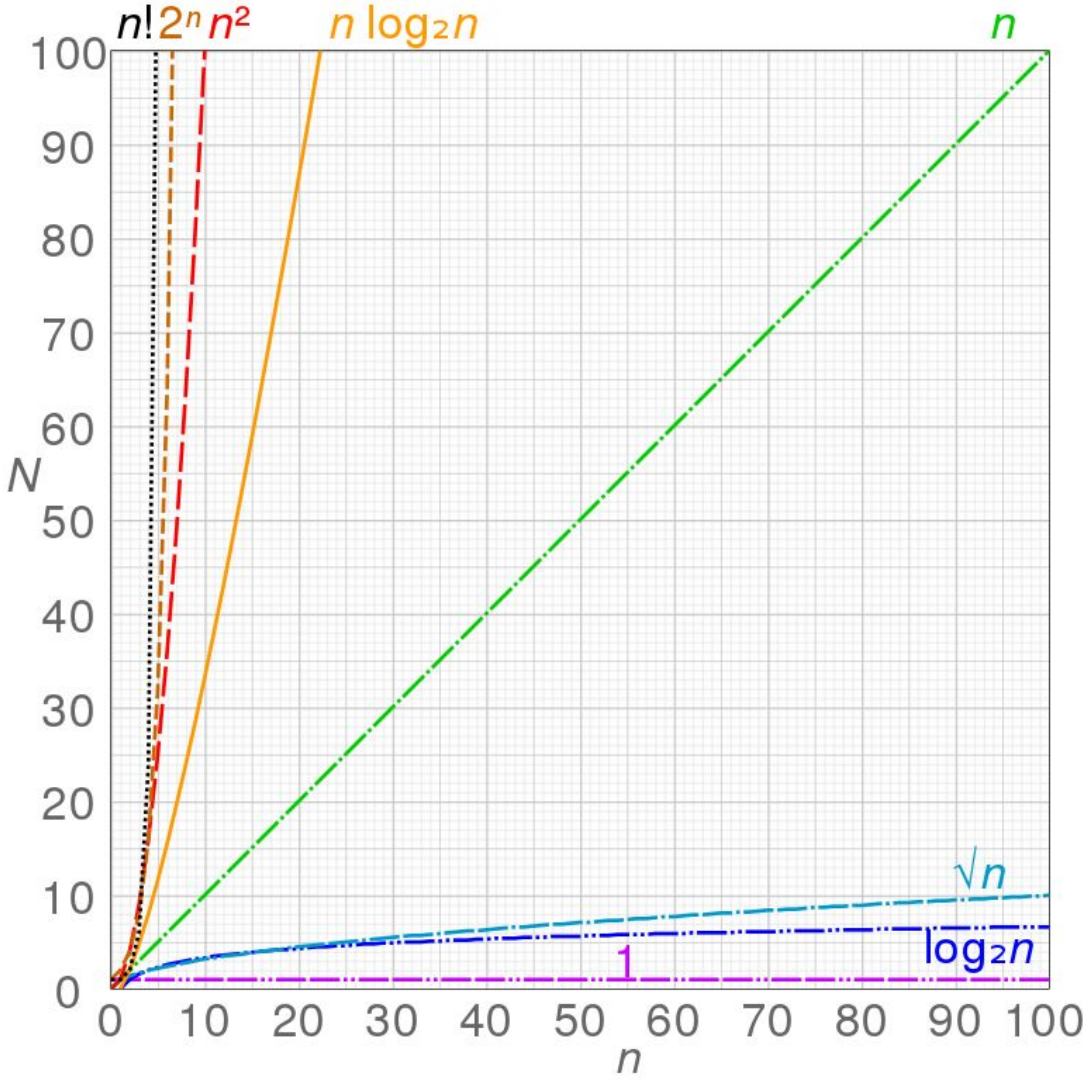
SER 321
Systems



SER 321

When to Distribute

When should we *consider* distributing?

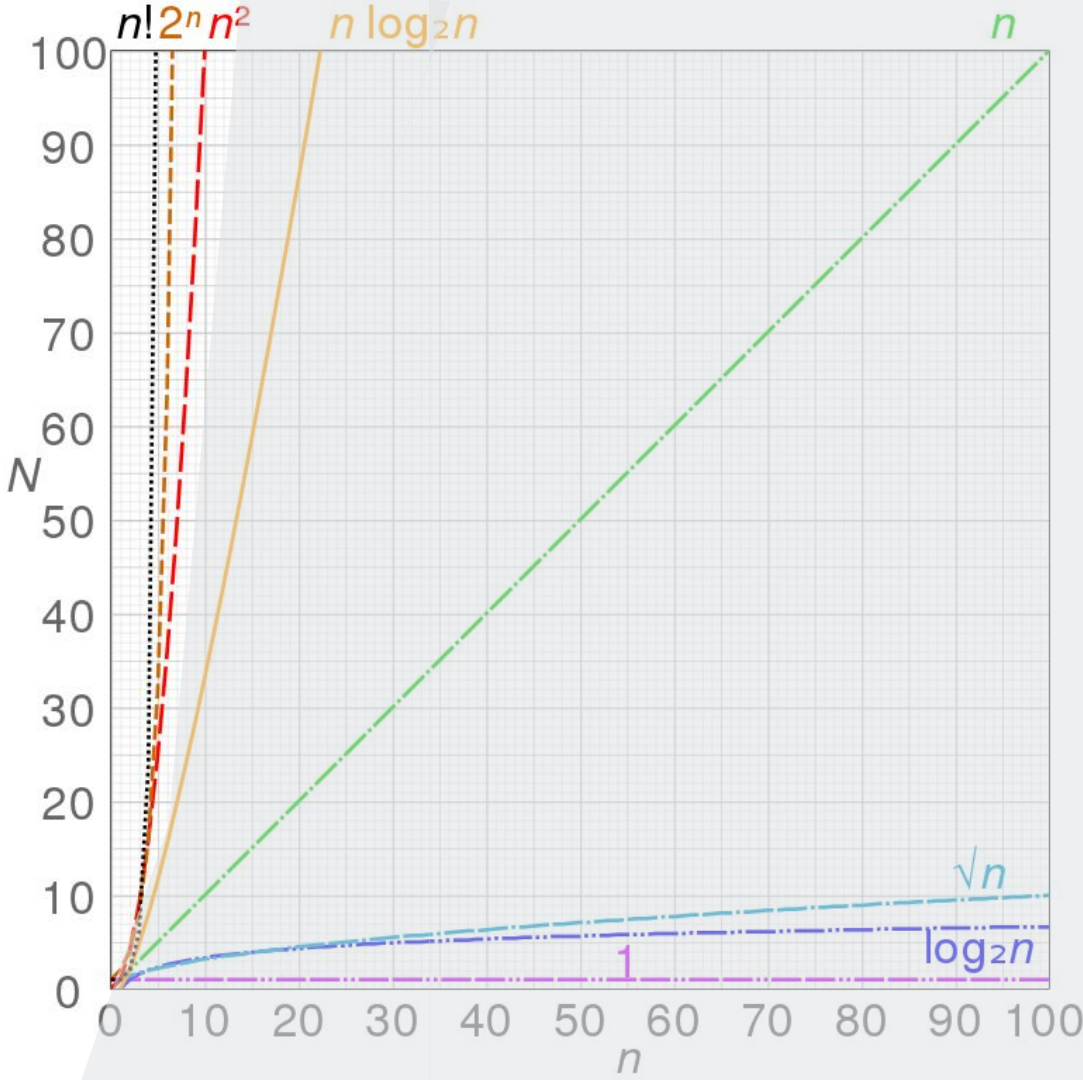


SER 321

When to Distribute

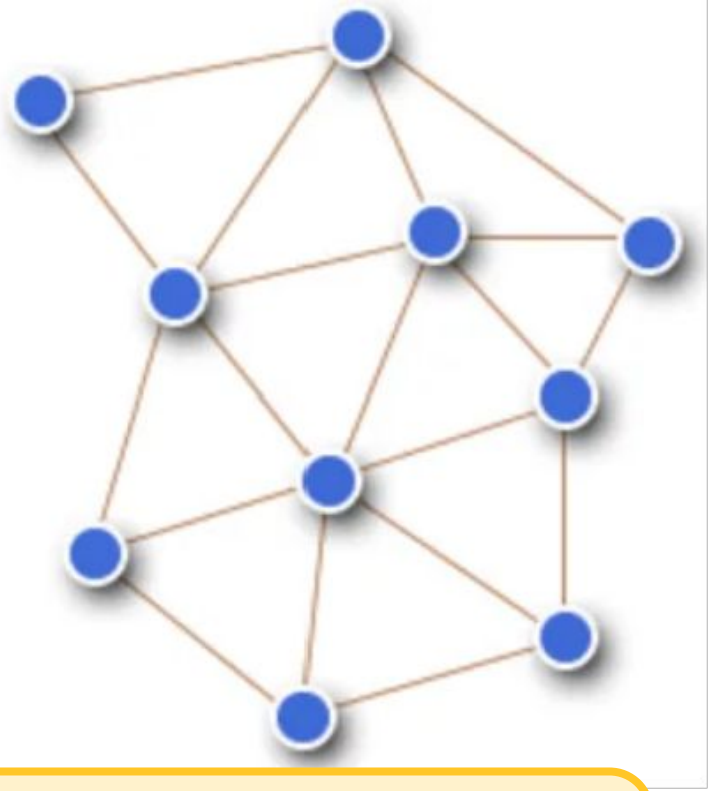
When should we *consider* distributing?

Super Duper Extra Extra Large Orders of Magnitude!



SER 321
Distributed Issues

Remember that we are operating in ***reality***



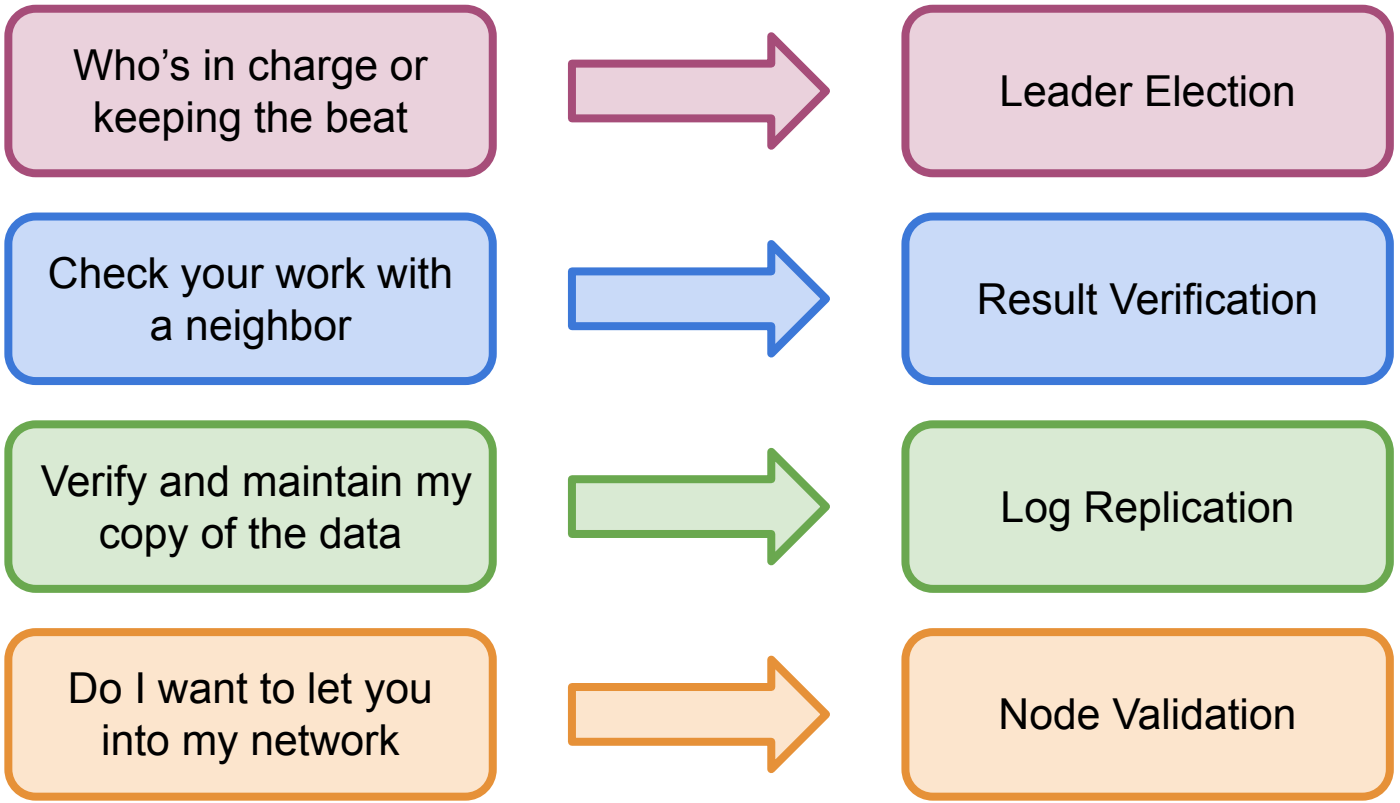
- No global clock
- Nodes *will* fail
- Web of nodes *will constantly* change
- Network is not *always* reliable
- Latency is *always present*
- The path traversed *changes*
- Some resources *must be shared*
- *You* need to prevent the pitfalls!
  - No deadlocks
  - No starvation
  - No error states

Do we remember the distributed properties?

SER 321
Consensus

What is Consensus?

"General agreement or trust amongst a group"



SER 321
Middleware

We have been:

- Serializing Messages
- Sending Messages
- Parsing Messages
- Handle Messages

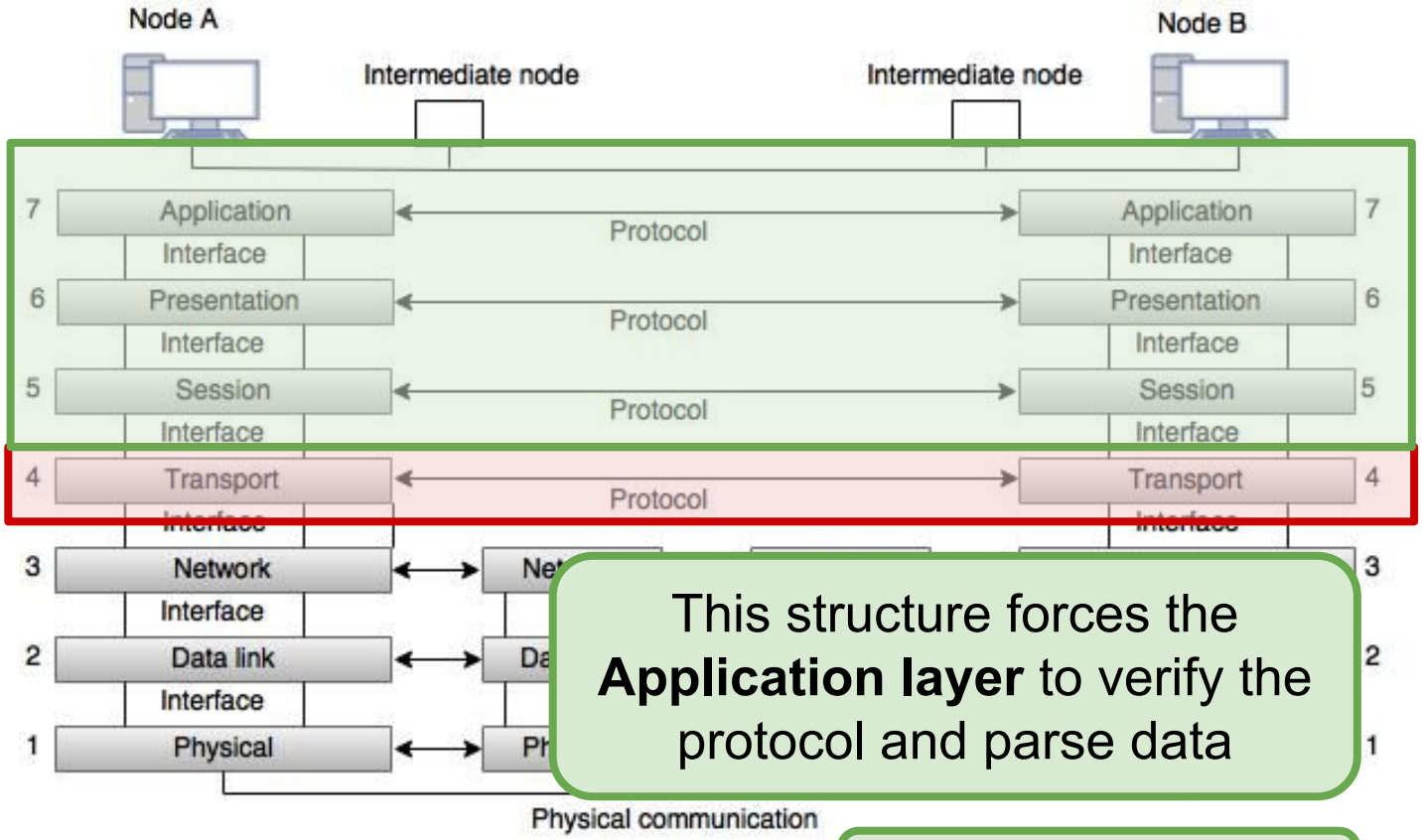


Fig: OSI Model

This structure forces the **Application layer** to verify the protocol and parse data

Not really its job…

SER 321
Middleware

With Middleware:

- Serializing Messages
- Sending Messages
- Parsing Messages
- Handle Messages

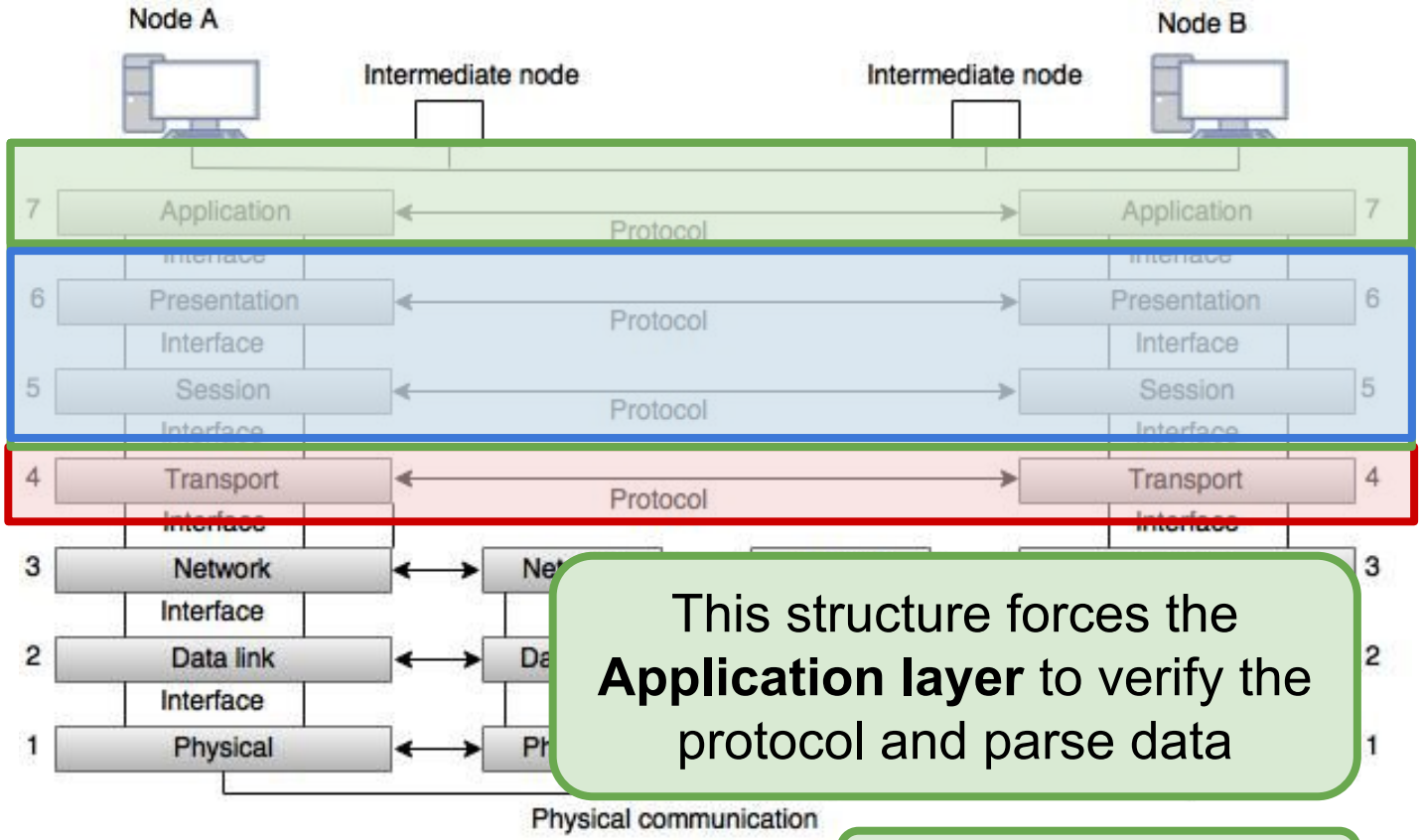


Fig: OSI Model

This structure forces the **Application layer** to verify the protocol and parse data

Not really its job…

SER 321
Middleware

With Middleware:

Serializing Messages

Sending Messages

Parsing Messages

Handle Messages

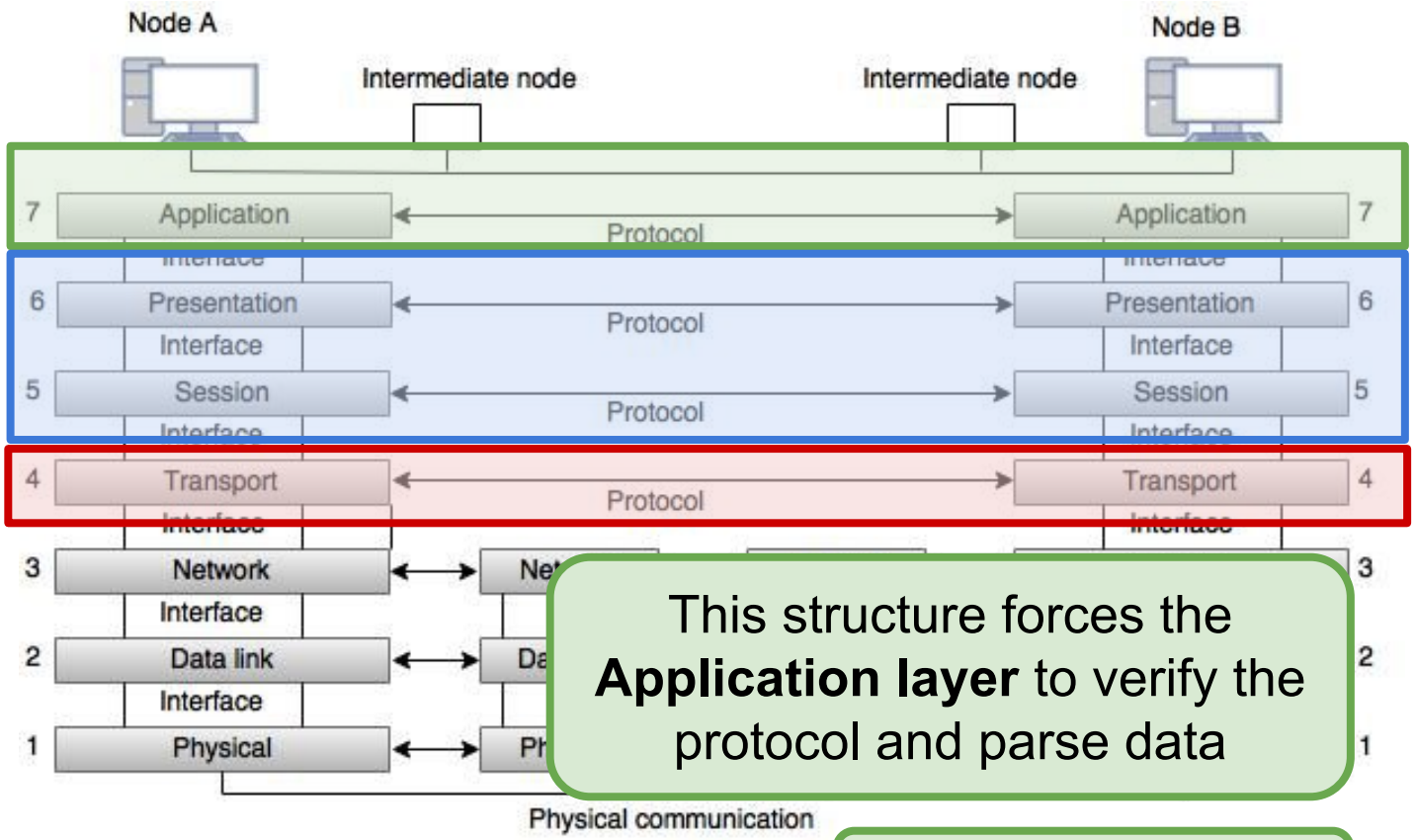


Fig: OSI Model

This structure forces the **Application layer** to verify the protocol and parse data

Not really its job…

SER 321
Middleware

Middleware:

Session Layer Responsibilities:

Authentication

Authorization

Session Management

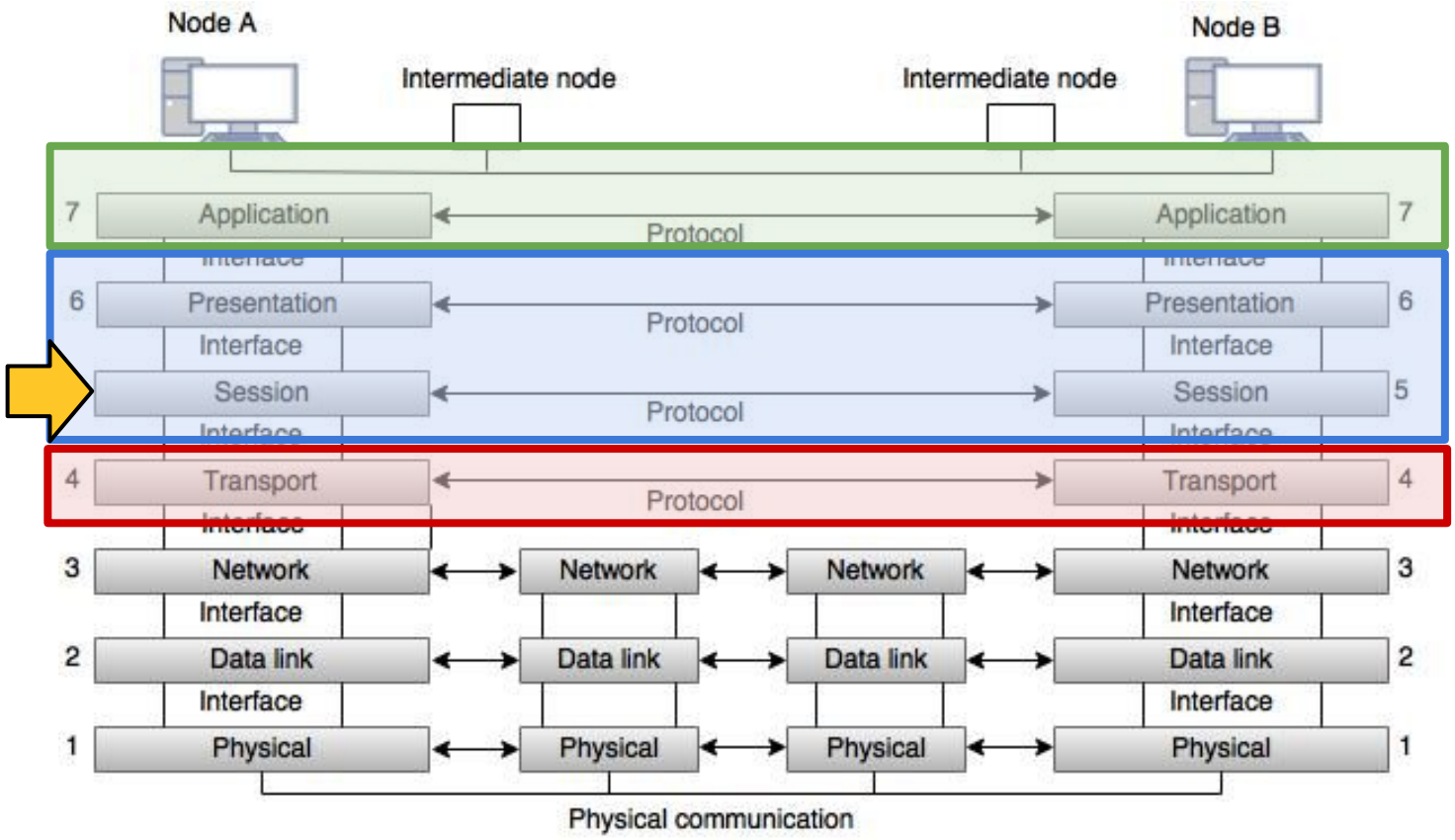


Fig: OSI Model

SER 321
Middleware

Middleware:

Presentation Layer Responsibilities:

Translation

Compression

Encryption

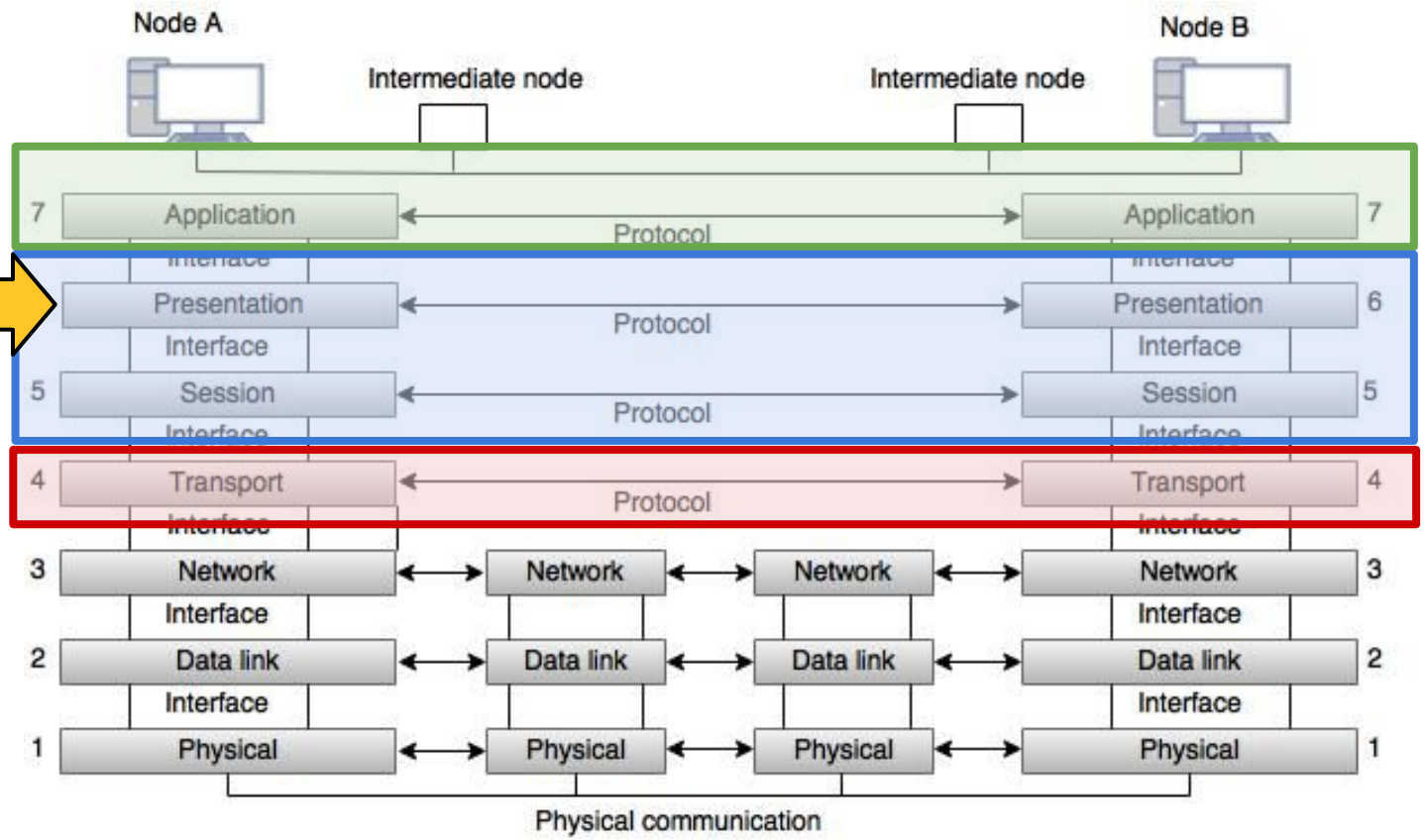


Fig: OSI Model

SER 321

Middleware Examples

Examples?

Message Oriented Middleware (MOM)

Web Frameworks

Remote Procedure Calls (RPC)

App. Programming Interface (API)

SER 321
Middleware Benefits

Why do we care?

Agility

Reusability

Efficiency

Cost Effectiveness

Portability

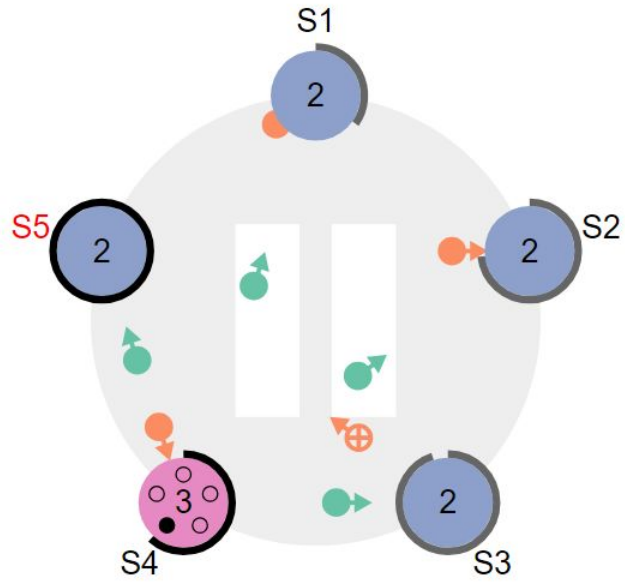
TL;DR → It's the "glue" between the client and server

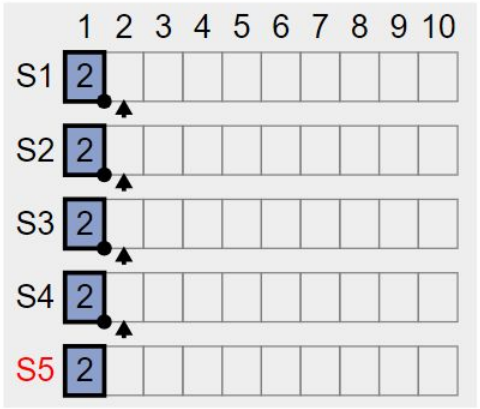
SER 321

RAFT

How do we feel about Consensus?

Do we want to look at RAFT again?





SER 321

Scratch Space

Questions?

Survey:

https://asuasn.info/ASNSurvey

81

Upcoming Events

SI Sessions:

- Sunday, March 2nd at 7:00 pm MST - **Q&A Session**

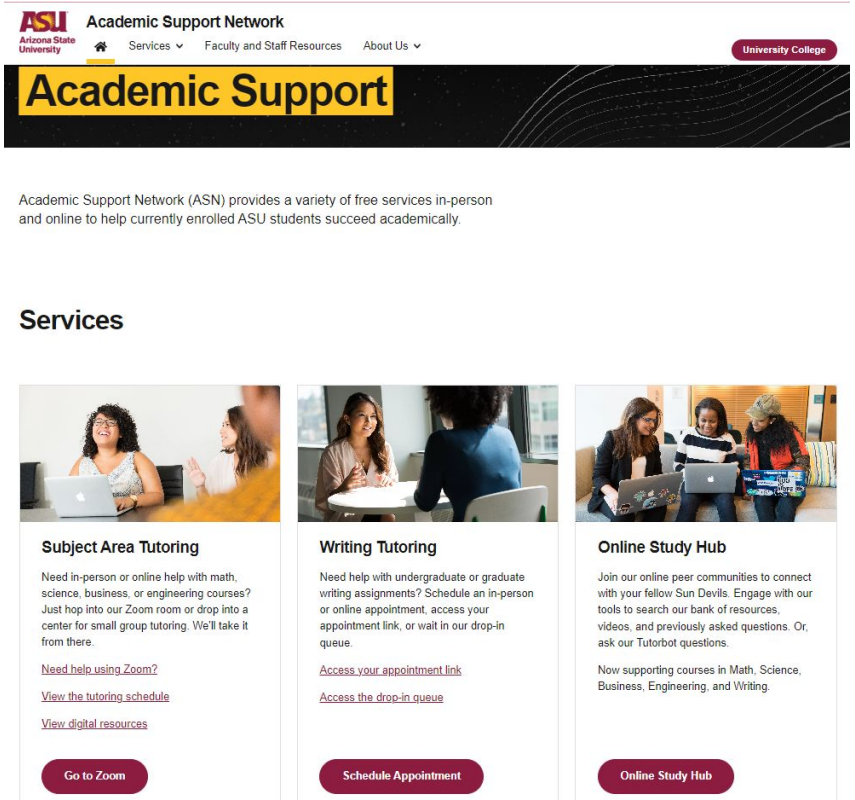
Review Sessions:

- Thursday, February 27th at 7:00 pm MST - **Exam Review Session (2hrs)**
- Sunday, March 2nd at 7:00 pm MST - **Q&A Session**

More Questions?

Check out our other resources!

tutoring.asu.edu

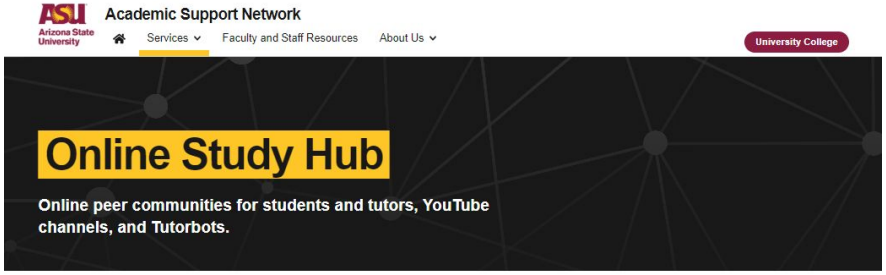


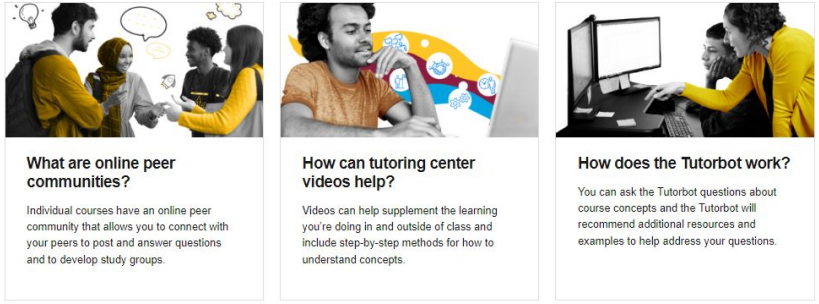
1. **Click on ‘Go to Zoom’ to log onto our Online Tutoring Center.**
2. **Click on ‘View the tutoring schedule’ to see when tutors are available for specific courses.**

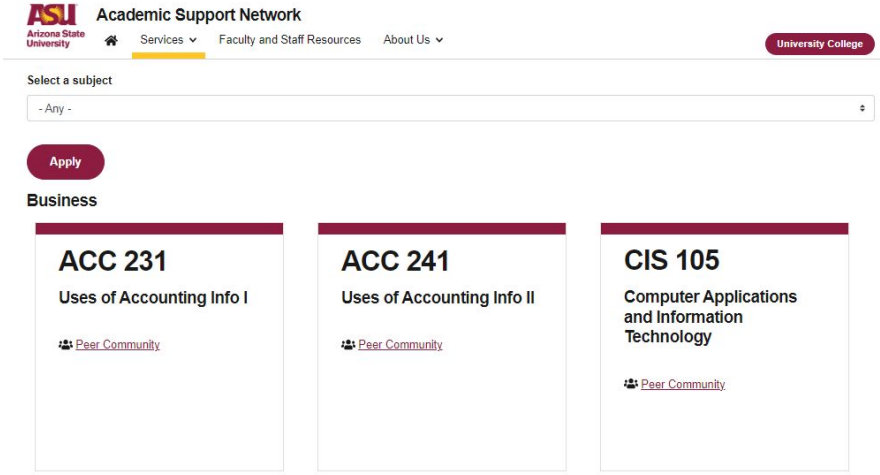
More Questions?

Check out our other resources!

tutoring.asu.edu/online-study-hub



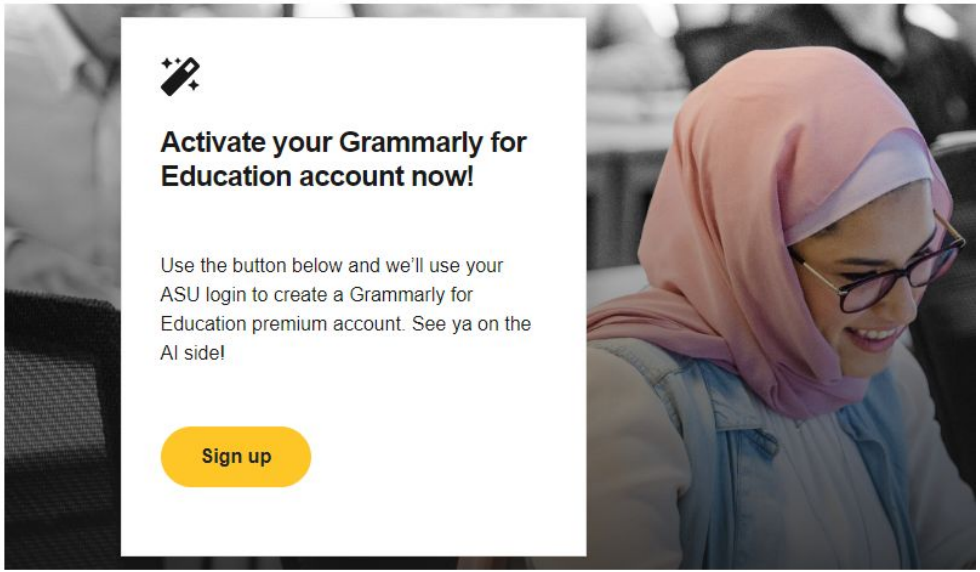




Don't forget to check out the Online Study Hub for additional resources!

Expanded Writing Support Available

Including Grammarly for Education, at no cost!



tutoring.asu.edu/expanded-writing-support

*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