SER 321 A Session

SI Session

Sunday, February 23rd 2025

7:00 pm - 8:00 pm MST

Agenda

- Distributed Properties Review
- Consensus
  - Types and Algorithms
  - RAFT Example
- Peer to Peer Communication
- Assignment 5 Example Tracing

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
Connections

Connections - Review!

Check out the recording for our gameplay, or click the link to play yourself!

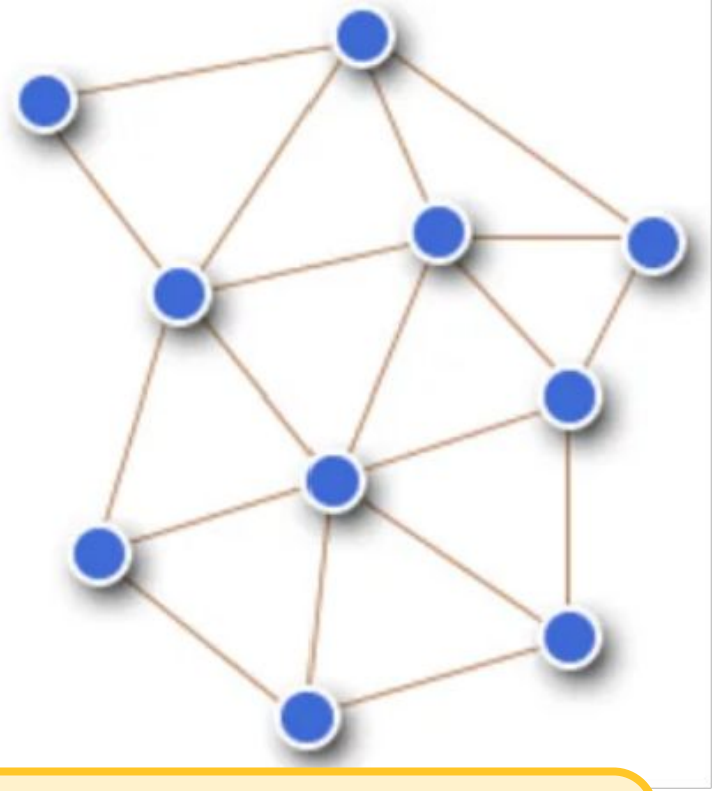
SER 321
Connections

Connections!

Check out the recording for our gameplay, or click the link to play yourself!

SER 321
Distributed Systems

Check out the recording for our gameplay, or click the link to play yourself!



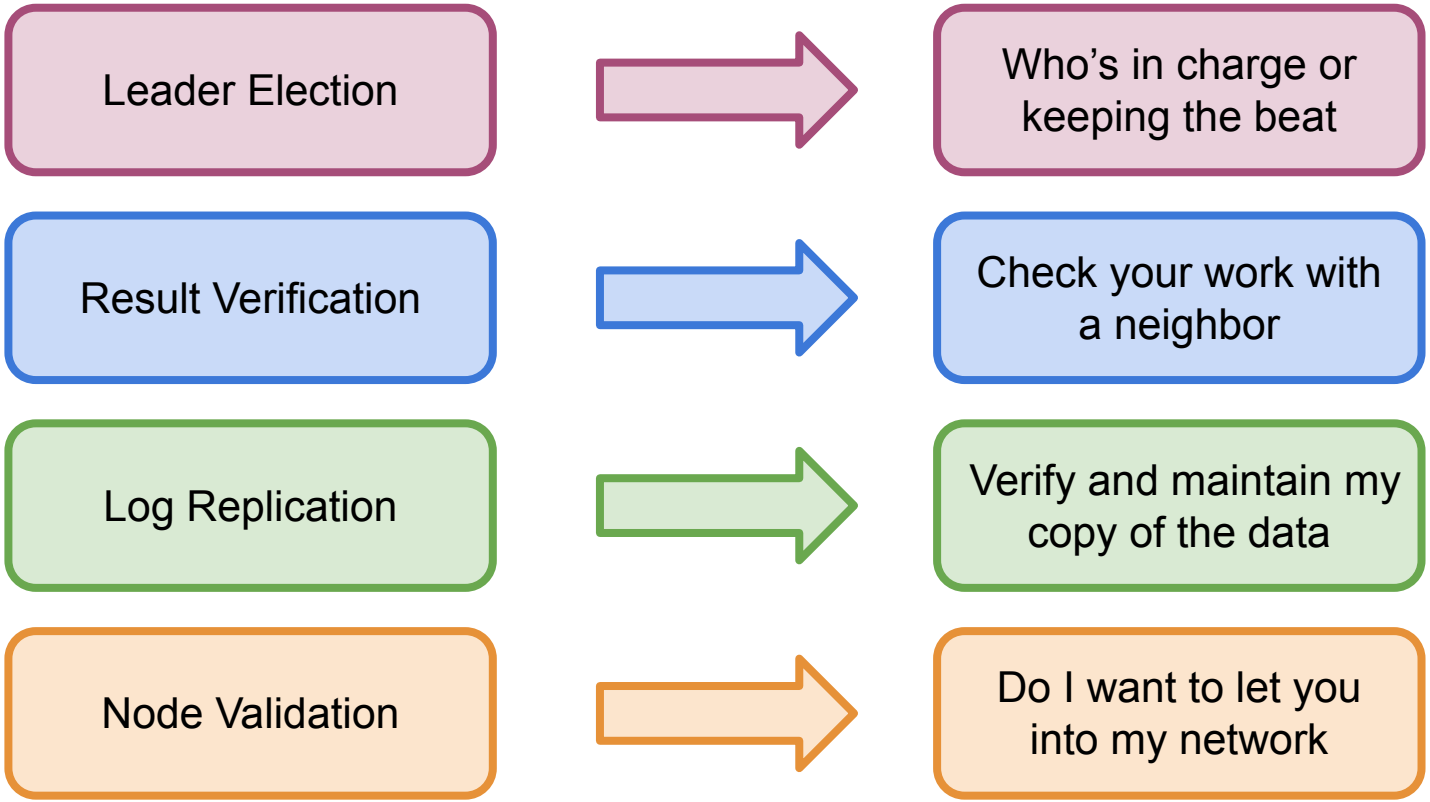
- *No* global clock
- Nodes *will* fail
- Web of nodes *will constantly* change
- Network is not *always* reliable
- Latency is *always present*
- The path traversed *changes*
- Some resources *must be shared*
- *You* need to prevent the pitfalls!
  - No deadlocks
  - No starvation
  - No error states

Word Scramble
or
Raw Recall?

SER 321
Consensus

Check out the recording for the discussion!

Types of Consensus?



SER 321
Consensus

Match the Consensus Algorithm to its Description!

2-Phase Commit

Blockchain

Proof of Work

RAFT

Check out the recording for the solution!

If you solve this resource-intensive problem, you may make a request

Leader Election and Log Replication coordinate transactions

Transaction Coordinator approves and orchestrates transactions

Distributed Ledger used to determine if a transaction is valid

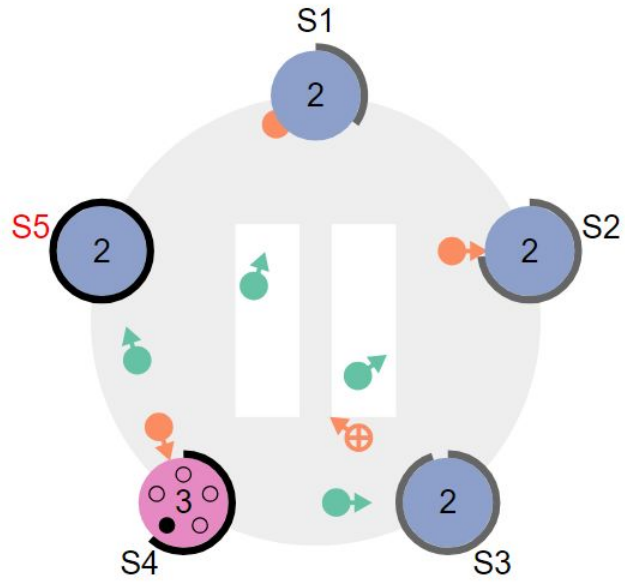
SER 321

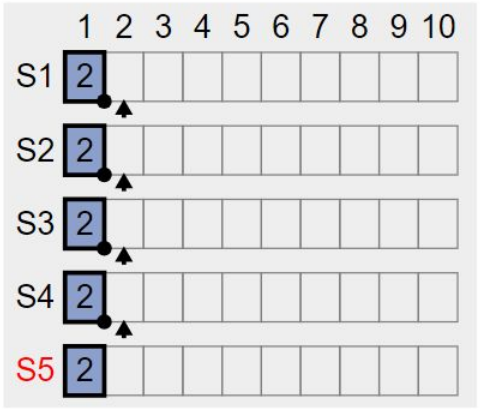
RAFT

RAFT is a great consensus example!

Leader Election

Log Replication

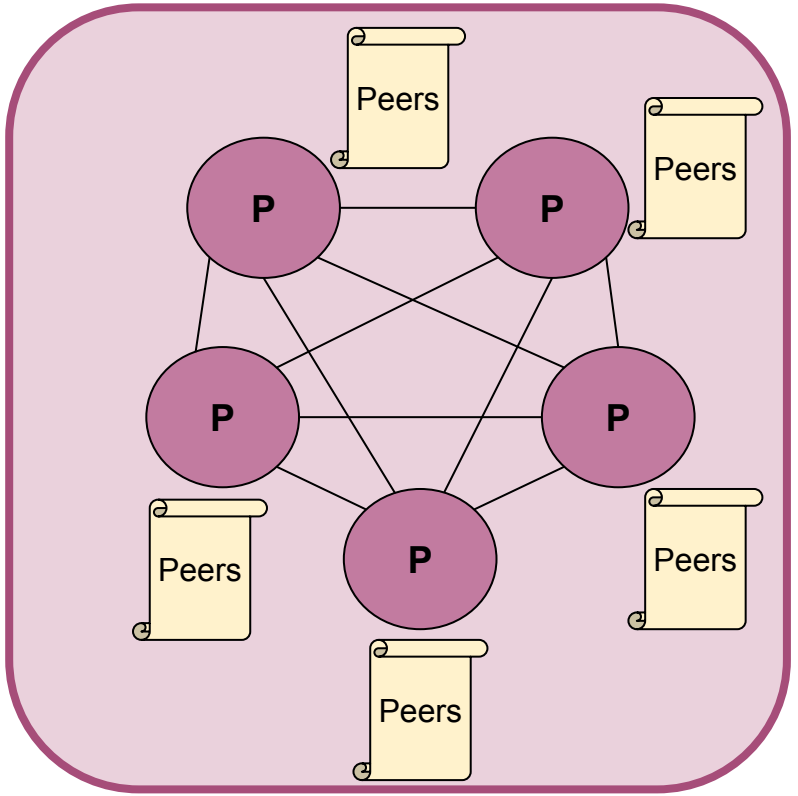




The Secret Lives of Data is a different visualization of Raft. It's more guided and less interactive, so it may be a gentler starting point.

SER 321
Communication

What about ***adding*** a Peer to the Cluster?

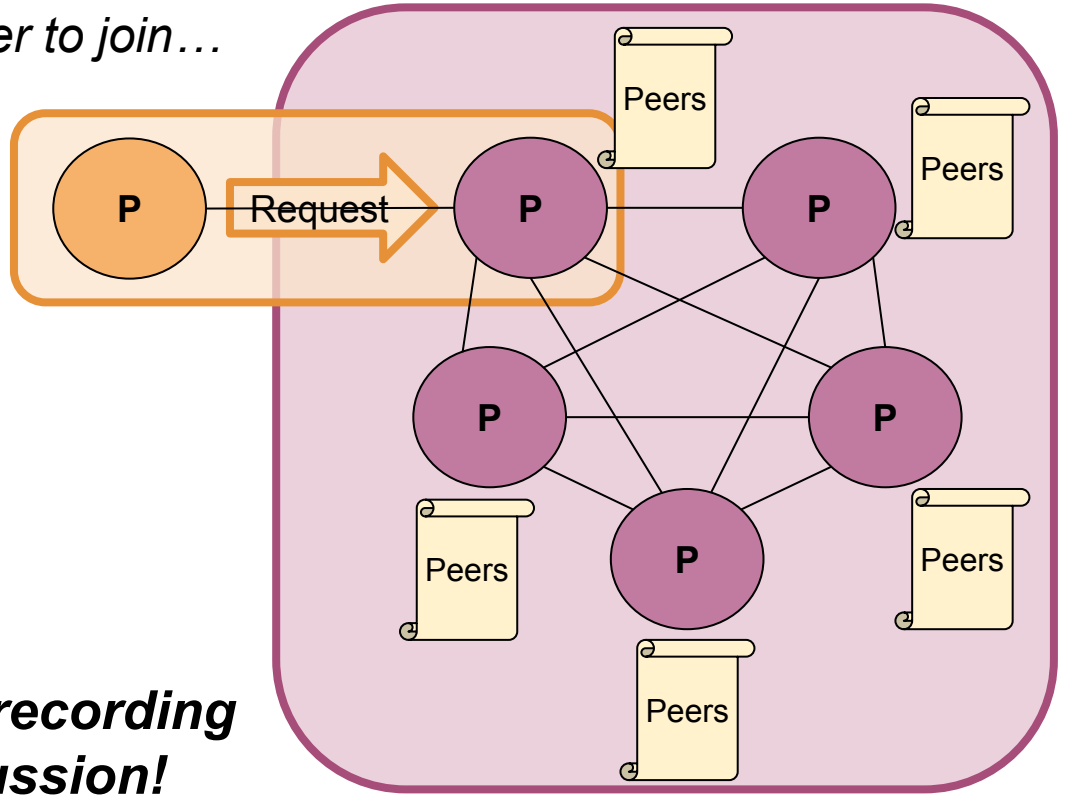


Check out the recording for the discussion!

SER 321
Communication

What about ***adding*** a Peer to the Cluster?

Assuming we want to allow the peer to join…



Is that all?

Check out the recording for the discussion!

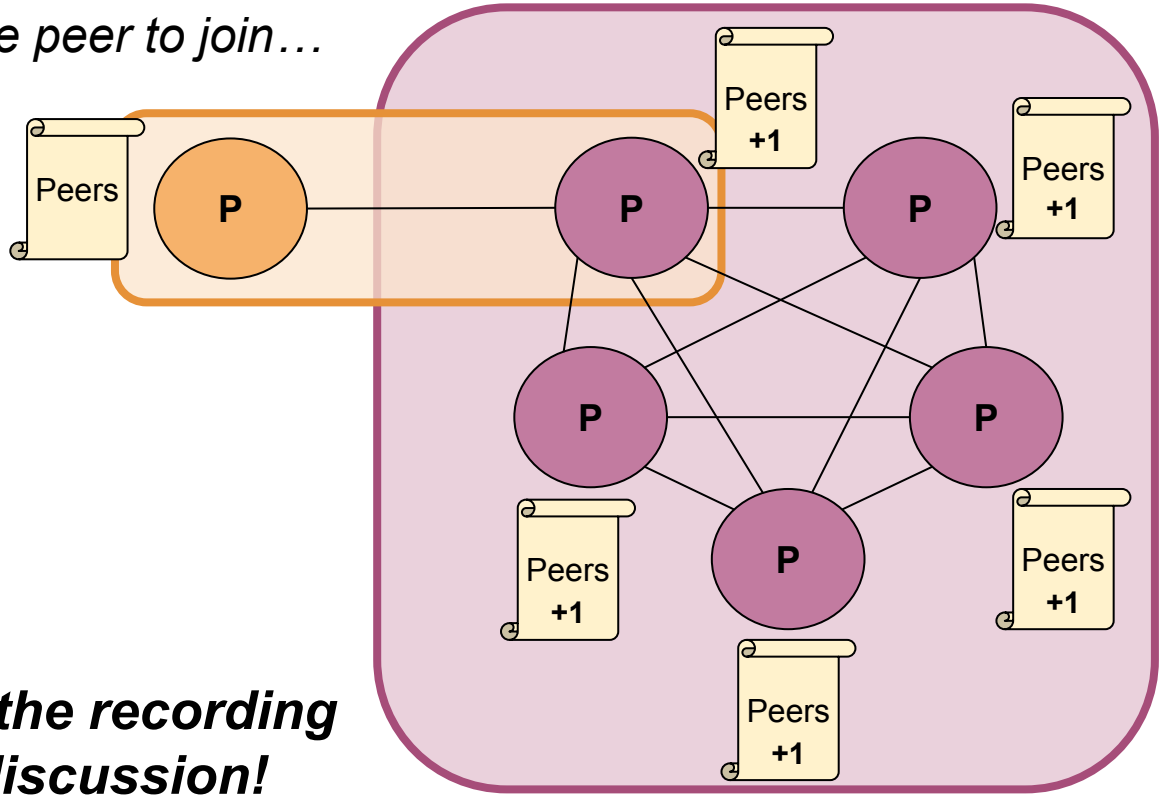
SER 321
Communication

What about ***adding*** a Peer to the Cluster?

Assuming we want to allow the peer to join…

Three Additional Steps:

1.
2.
3.



Check out the recording for the discussion!

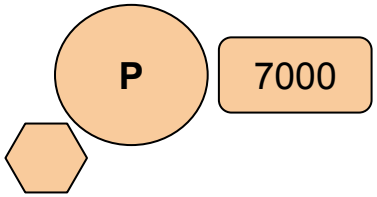
SimplePeerToPeer

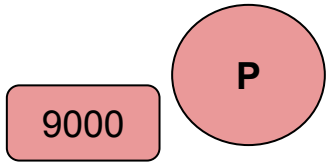
SER 321
Communication

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

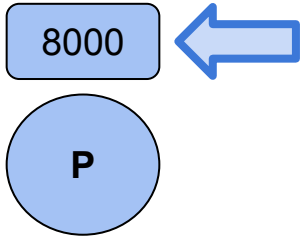
Run With:

```
gradle runPeer --args "Name Port"
```





We are going to take a closer look at the code in a moment!




```
gradle runPeer --args "Peer8000 8000"
```

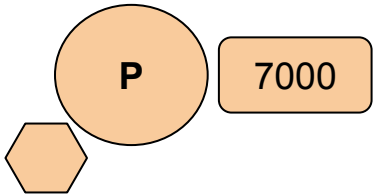
Check out the recording for the discussion!

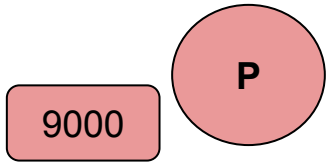
SimplePeerToPeer

SER 321
Communication

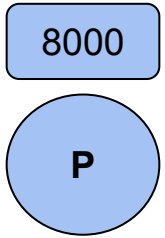
```
gradle runPeer --args "Peer7000 7000"
```

```
> Task :runPeer
Hello Peer7000 and welcome! Your port will be 7000
> Who do you want to listen to? Enter host:port
<==========----> 75% EXECUTING [21s]
> :runPeer
```





Check out the recording for the discussion!




```
gradle runPeer --args "Peer8000 8000"
```

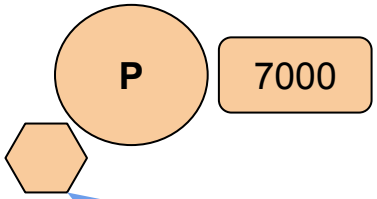
```
> Task :runPeer
Hello Peer8000 and welcome! Your port will be 8000
> Who do you want to listen to? Enter host:port
<==========----> 75% EXECUTING [21s]
> :runPeer
```

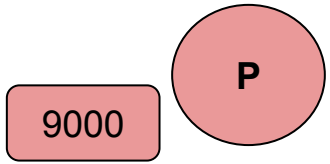
SimplePeerToPeer

SER 321
Communication

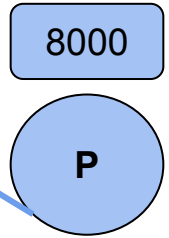
```
gradle runPeer --args "Peer7000 7000"
```

```
> Task :runPeer
Hello Peer7000 and welcome! Your port will be 7000
> Who do you want to listen to? Enter host:port
<==========----> 75% EXECUTING [21s]
> :runPeer
```





Check out the recording for the discussion!




```
> Task :runPeer
Hello Peer8000 and welcome! Your port will be 8000
> Who do you want to listen to? Enter host:port
<<===<==<<==<=<==========----> 75% EXECUTING [1m 56s]
> You can now start chatting (exit to exit)
<==========----> 75% EXECUTING [2m 3s]
> :runPeer
```

SimplePeerToPeer

SER 321
Communication

What will happen?

```
> Task :runPeer
Hello Peer7000 and welcome! Your port will be 7000
> Who do you want to listen to? Enter host:port
<==========----> 75% EXECUTING [21s]
> :runPeer
```

```
> Task :runPeer
Hello Peer8000 and welcome! Your port will be 8000
> Who do you want to listen to? Enter host:port
<<===<==<<==<=<==========----> 75% EXECUTING [1m 56s]
> You can now start chatting (exit to exit)
<<<==<=<==========----> 75% EXECUTING [3m 33s]
<==========----> 75% EXECUTING [3m 37s]
hi 7000
```

```
PS C:\ASU\SER321\examples_repo\ser321examples\Sockets\S
Starting a Gradle Daemon, 1 busy and 1 stopped Daemons
```

```
> Task :runPeer
Hello Peer7000 and welcome! Your port will be 7000
> Who do you want to listen to? Enter host:port
<==========----> 75% EXECUTING [2m 48s]
> :runPeer
```

Why?

9000

P

Check out the recording for the discussion!

8000

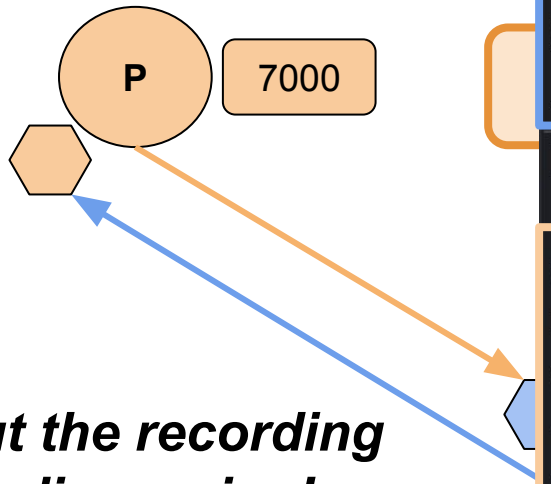
P

```
> Task :runPeer
Hello Peer8000 and welcome! Your port will be 8000
> Who do you want to listen to? Enter host:port
<<===<==<<==<=<==========----> 75% EXECUTING [1m 56s]
> You can now start chatting (exit to exit)
<<<==<=<==========----> 75% EXECUTING [3m 13s]
> :runPeer
hi 7000
```

SER 321
Communication

```
> Task :runPeer
Hello Peer7000 a
> Who do you wan
<=<====<==<===<=
> :runPeer
localhost:8000
```

```
> Task :runPeer
Hello Peer8000 and welcome! Your port will be 8000
> Who do you want to listen to? Enter host:port
<===<<====<=<=<==========----> 75% EXECUTING [3m 4s]
> You can now start chatting (exit to exit)
[Peer7000]: Hi Peer8000!
<==========----> 75% EXECUTING [4m 4s]
> :runPeer
```




```
PS C:\ASU\SER321\examples_repo\ser321examples\Sockets\
```

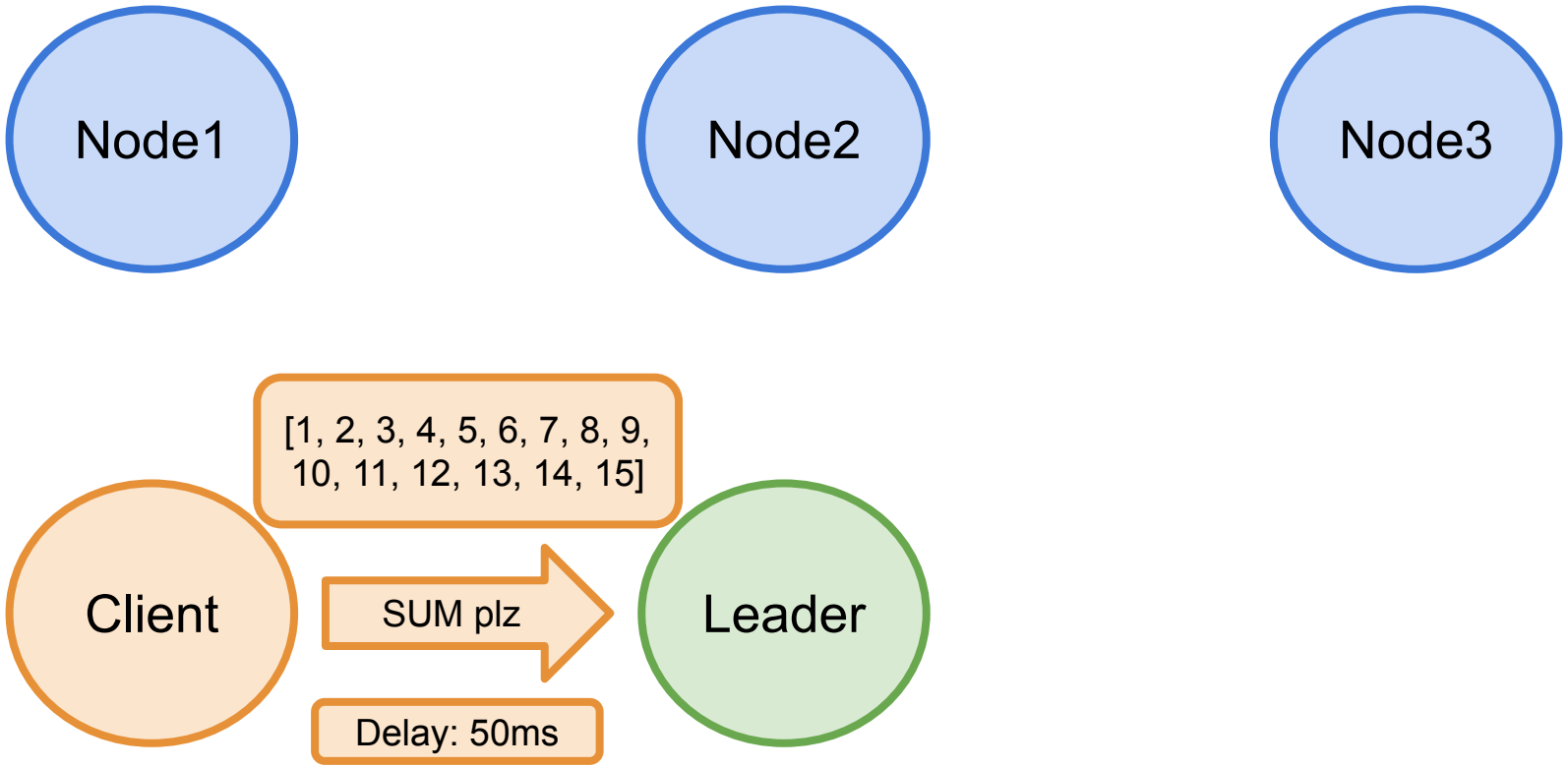
```
> Task :runPeer
Hello Peer7000 and welcome! Your port will be 7000
> Who do you want to listen to? Enter host:port
<=<====<==<===<==========----> 75% EXECUTING [3m 43s]
> You can now start chatting (exit to exit)
<<<=<===<==<<==========----> 75% EXECUTING [3m 58s]
<==========----> 75% EXECUTING [4m 1s]
Hi Peer8000!
```

Check out the recording for the discussion!

SER 321

Assignment 5 Visualization

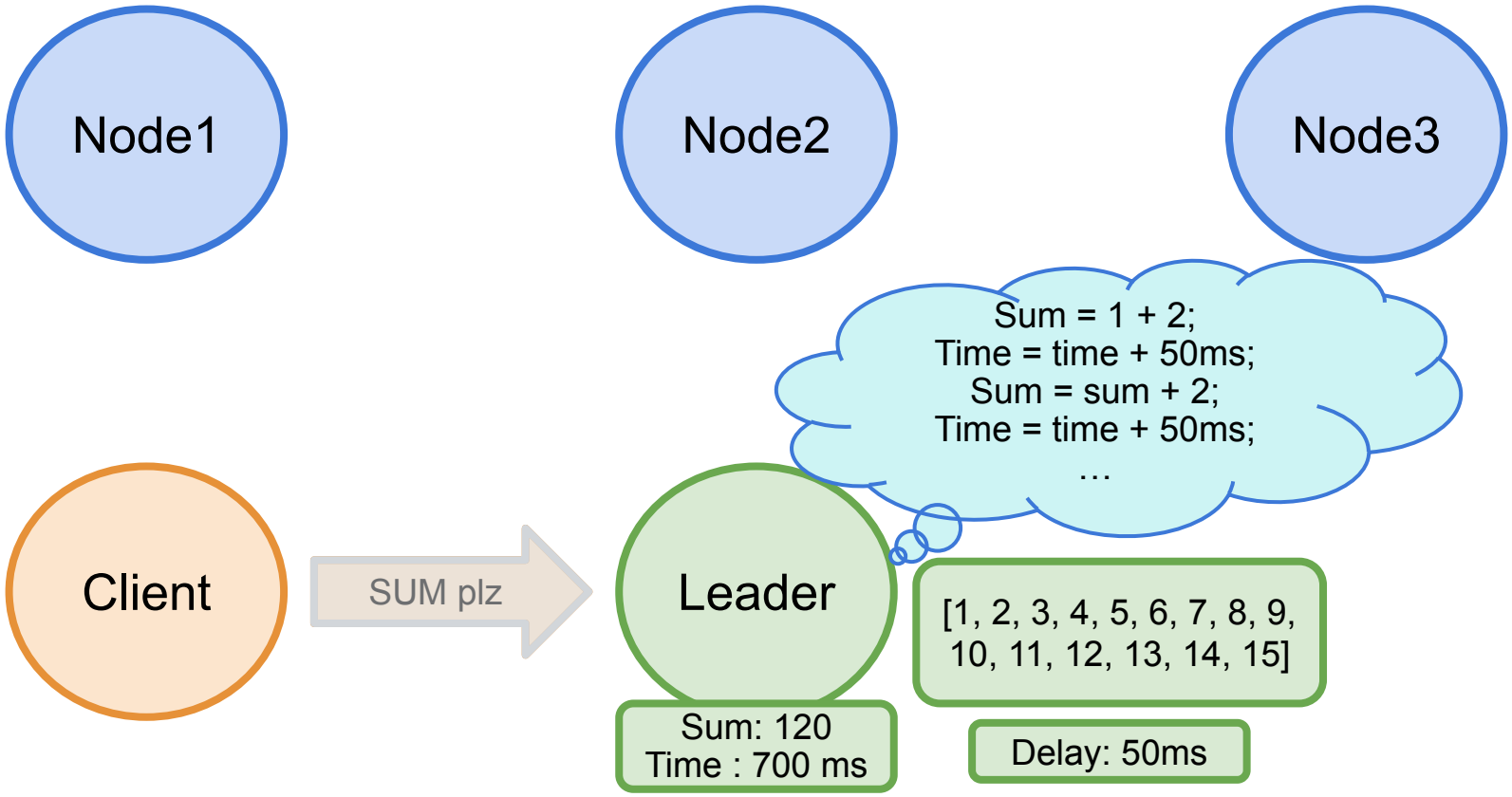
Let's depict the Example…



SER 321

Assignment 5 Visualization

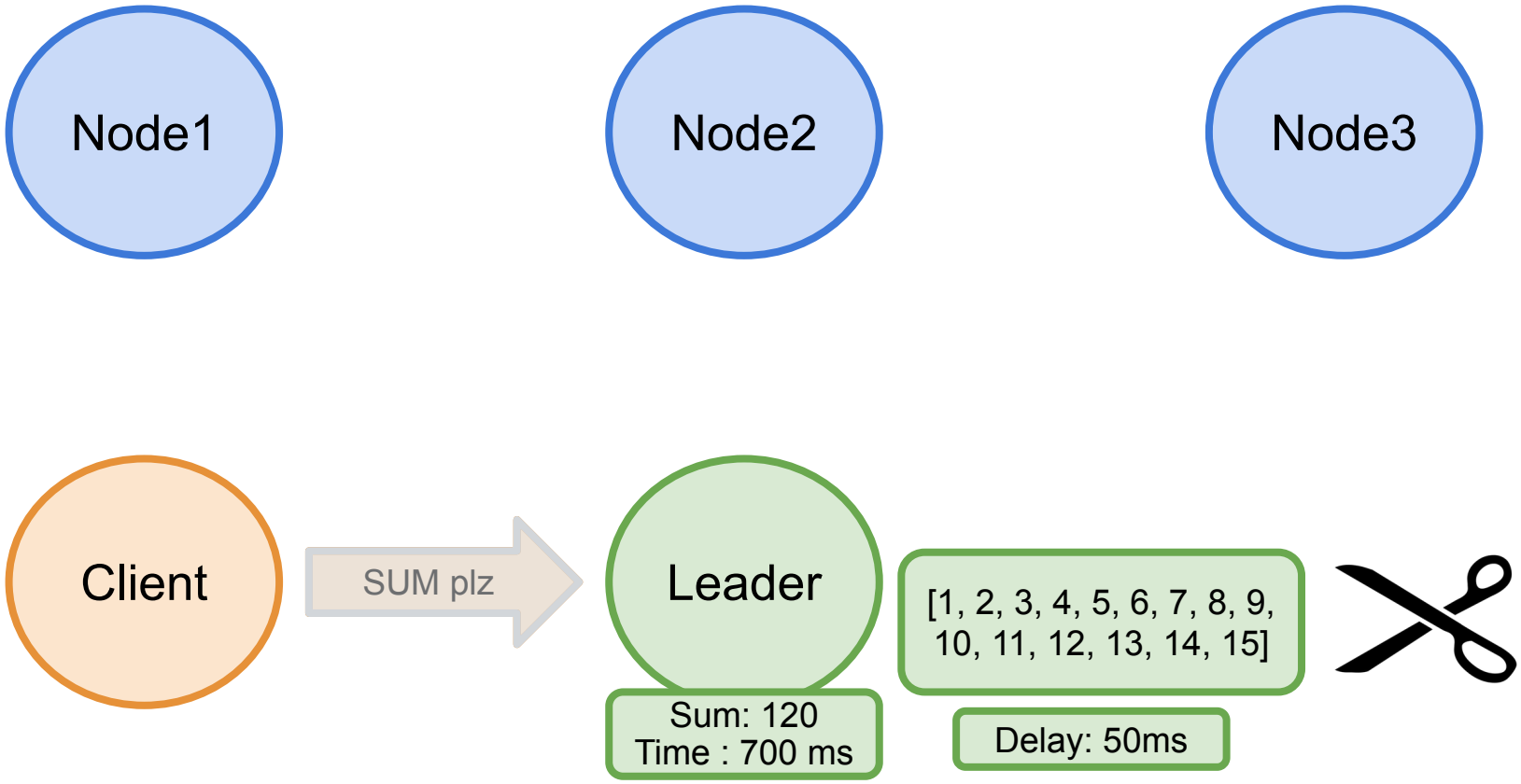
Let's depict the Example…



SER 321

Assignment 5 Visualization

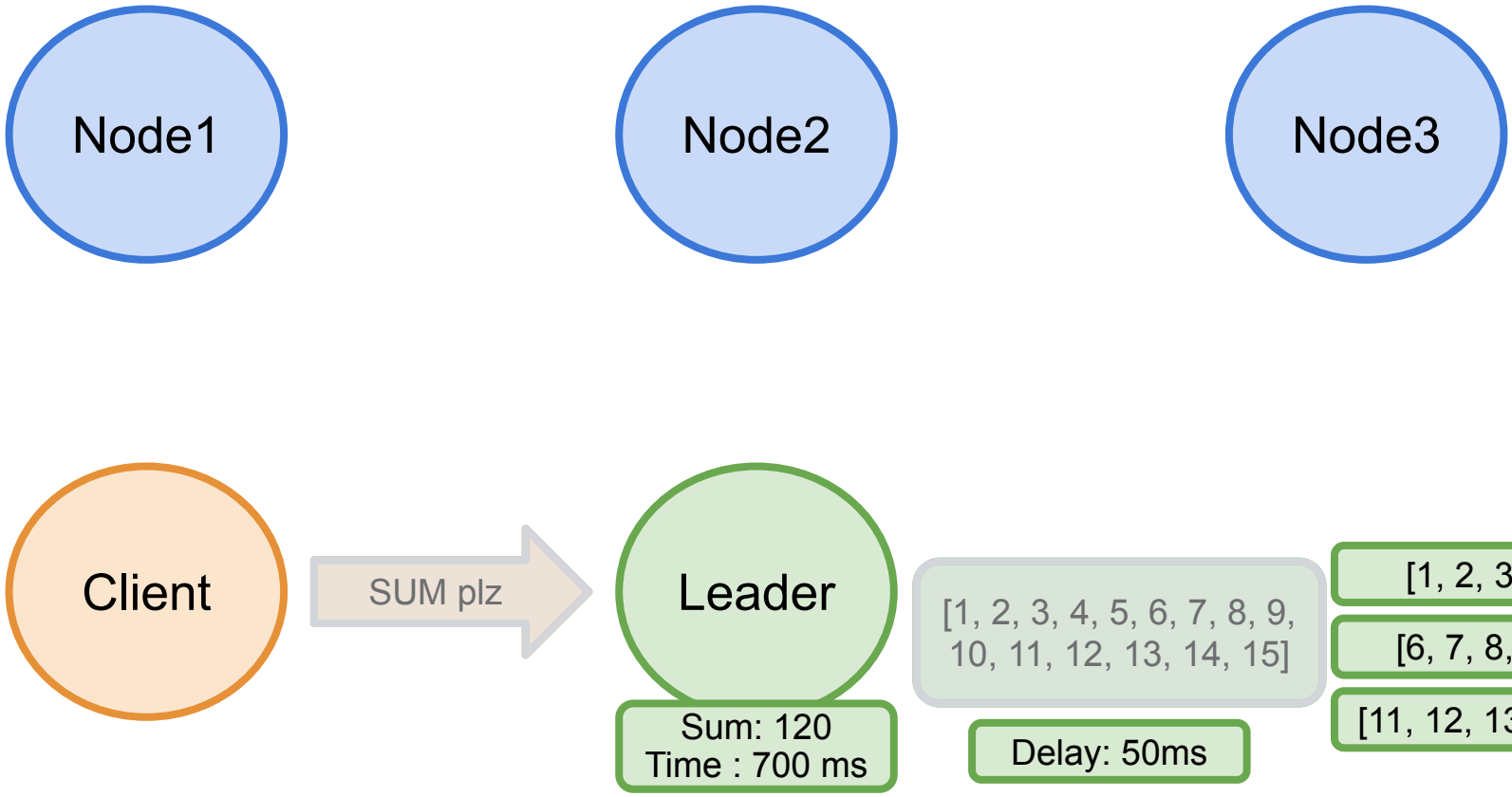
Let's depict the Example…



SER 321

Assignment 5 Visualization

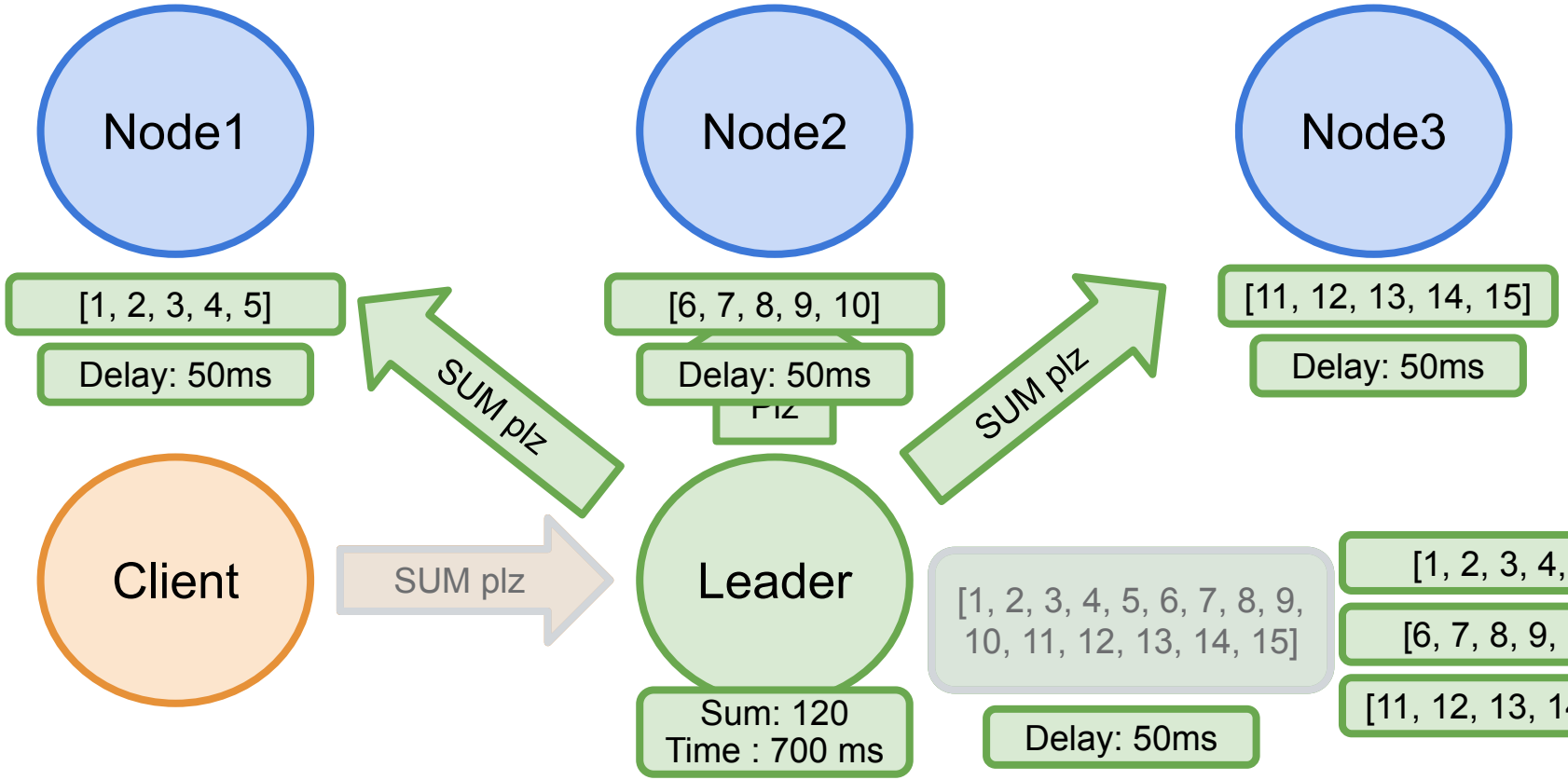
Let's depict the Example…



SER 321

Assignment 5 Visualization

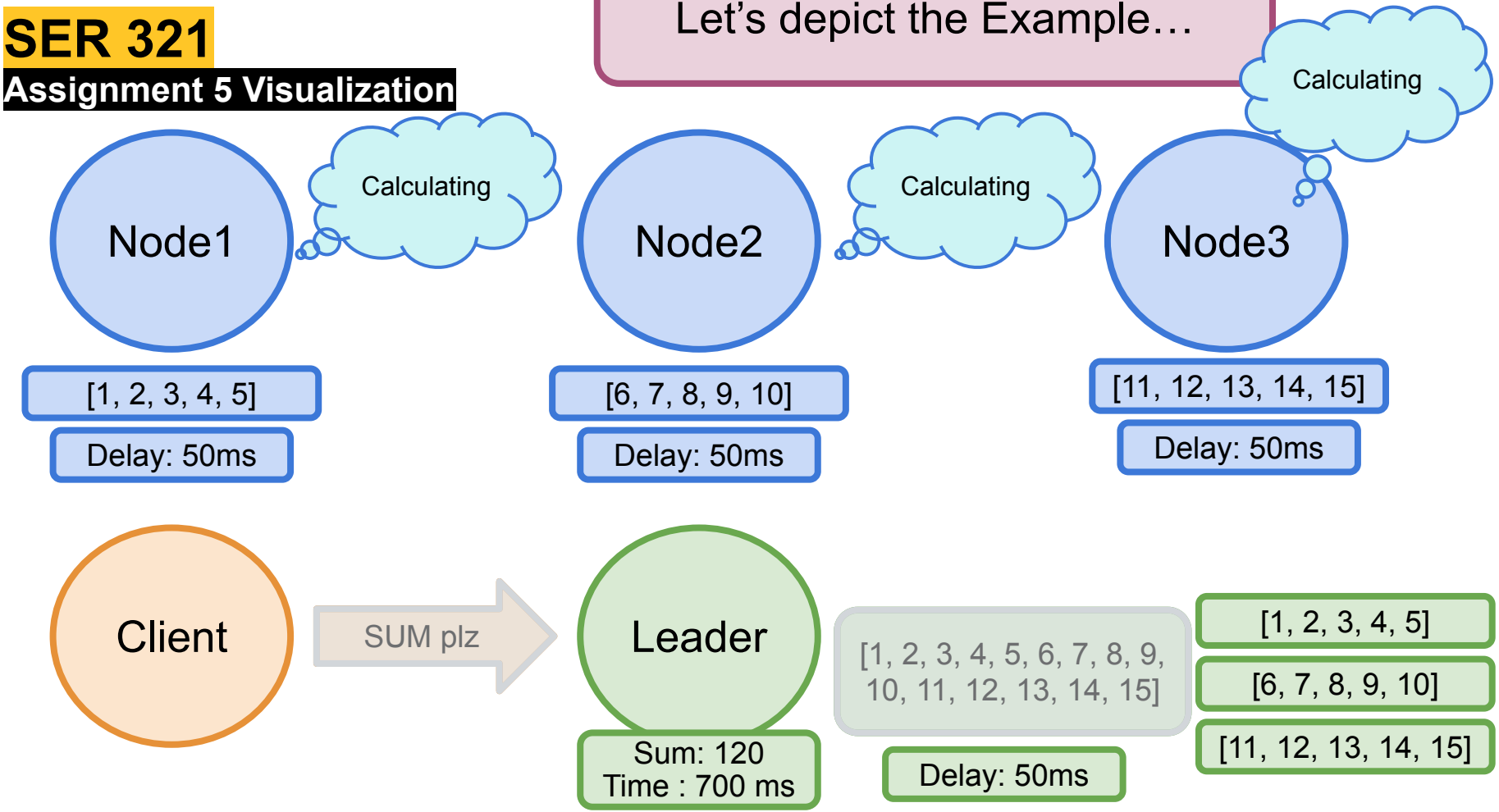
Let's depict the Example…



SER 321

Assignment 5 Visualization

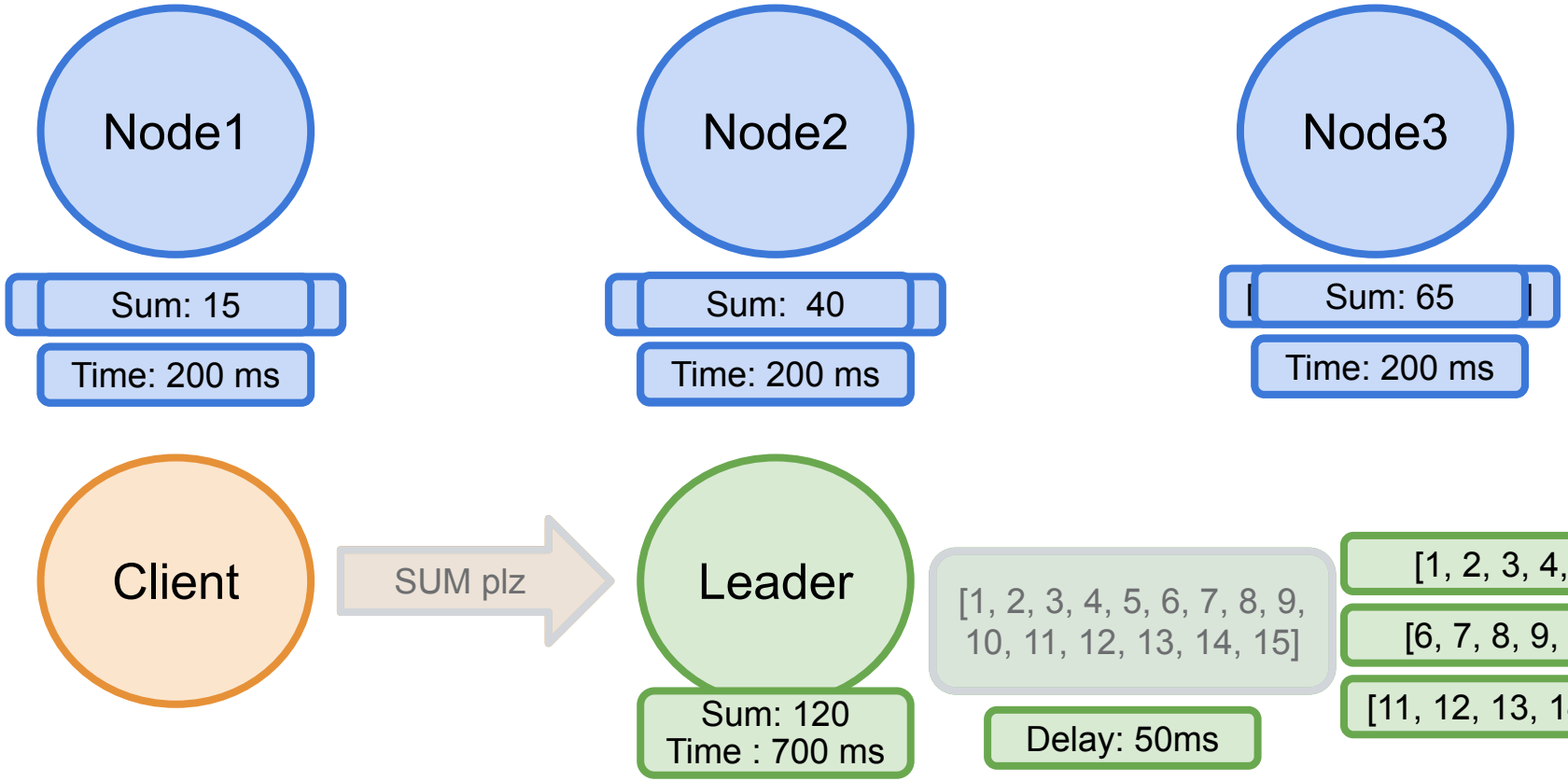
Let's depict the Example…



SER 321

Assignment 5 Visualization

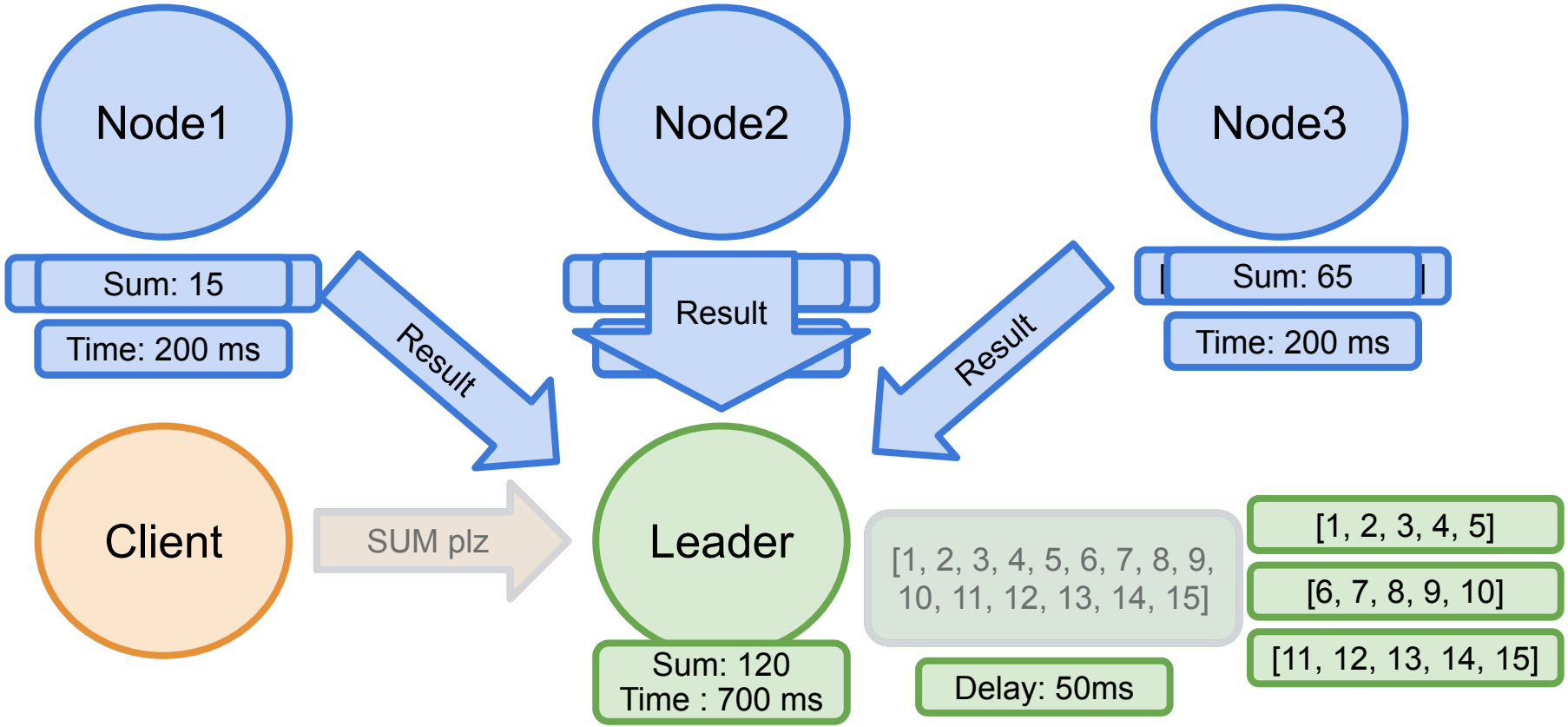
Let's depict the Example…



SER 321

Assignment 5 Visualization

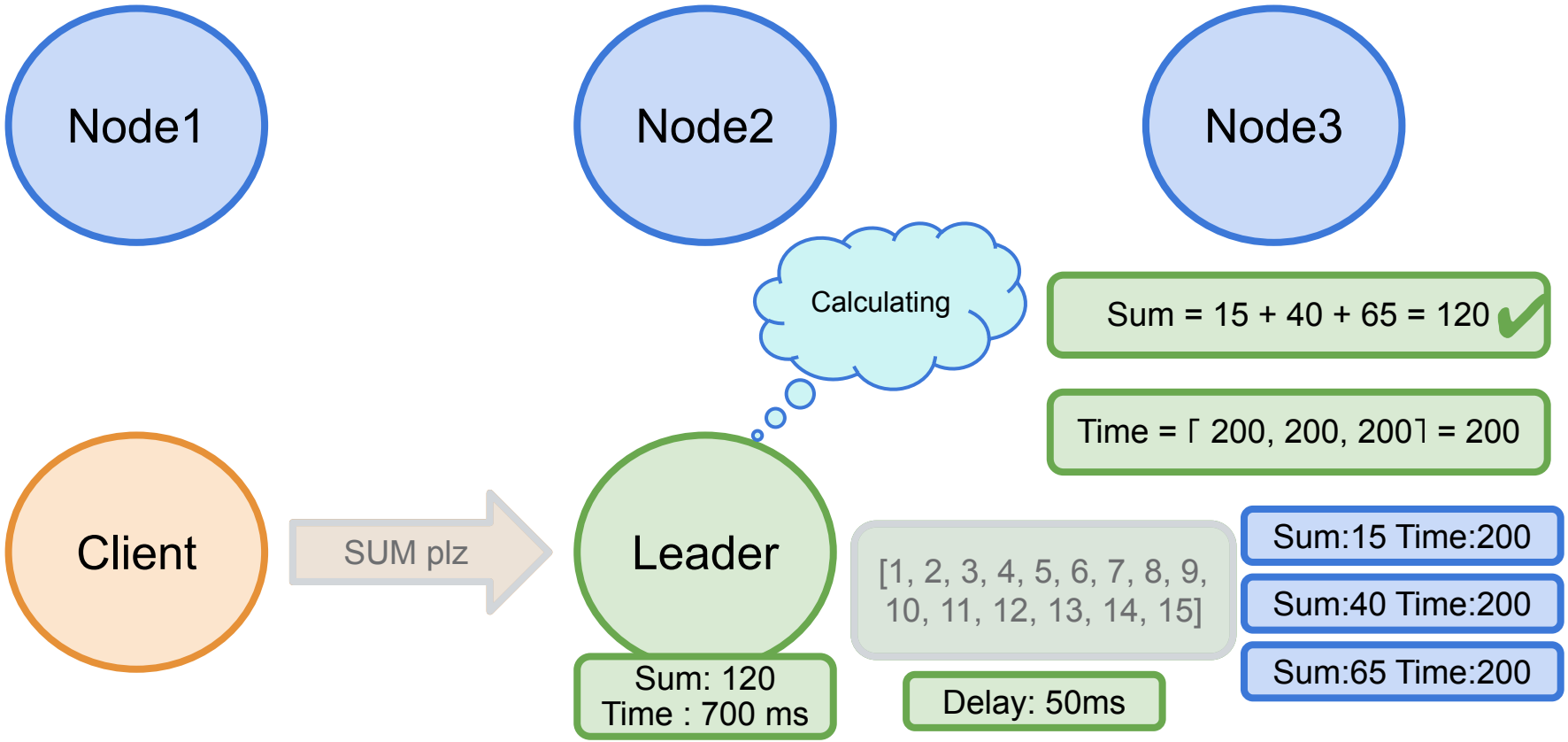
Let's depict the Example…



SER 321

Assignment 5 Visualization

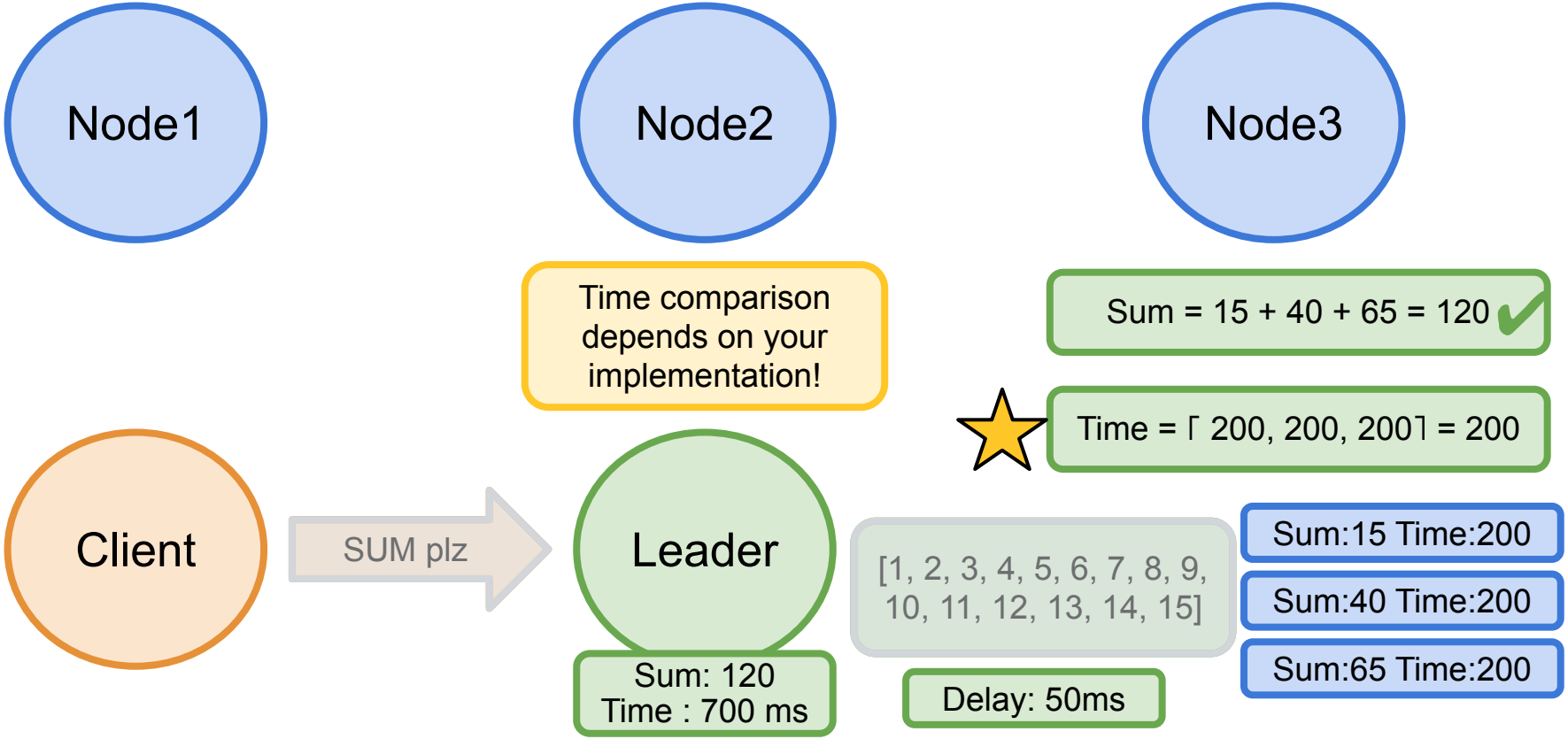
Let's depict the Example…



SER 321

Assignment 5 Visualization

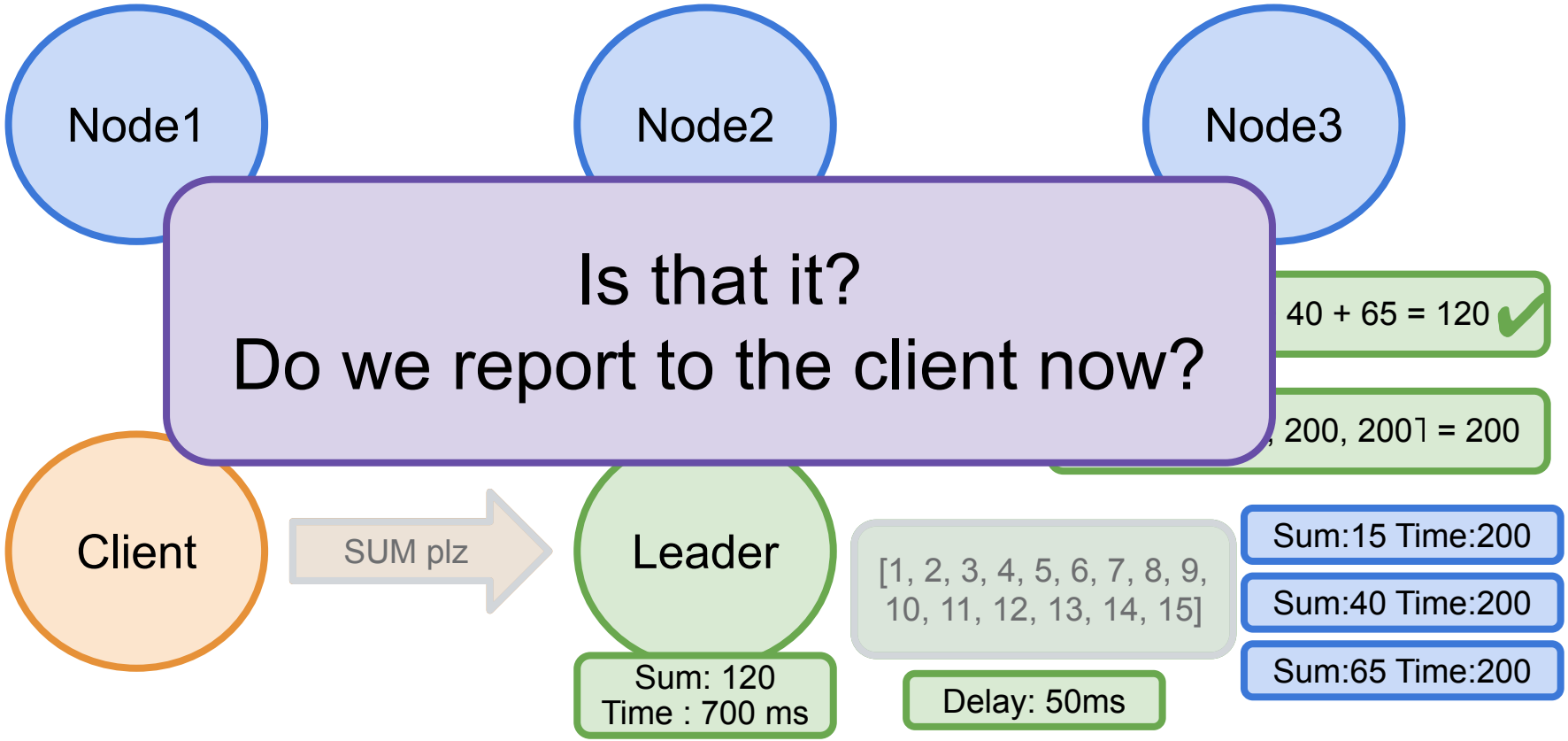
Let's depict the Example…



SER 321

Assignment 5 Visualization

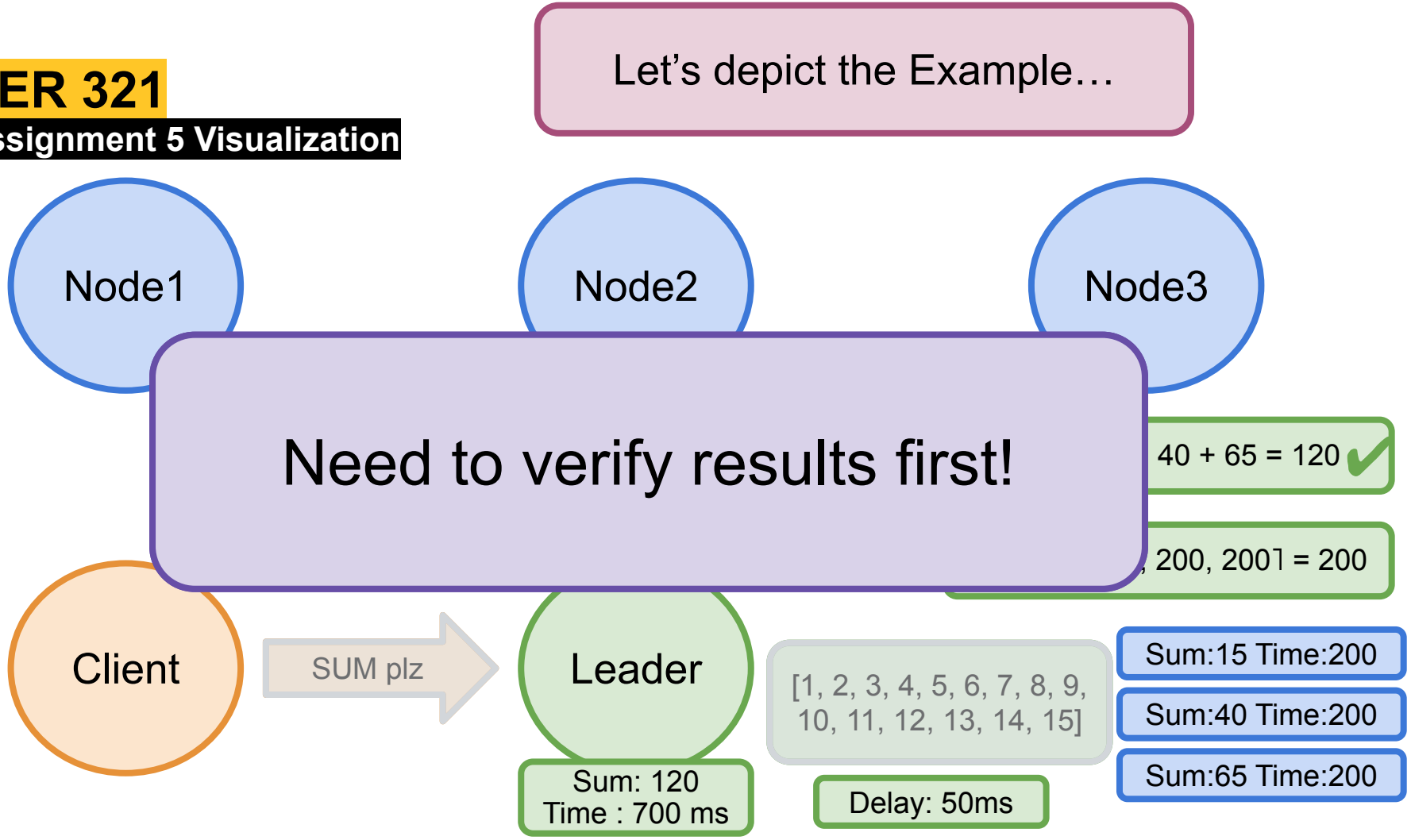
Let's depict the Example…



SER 321

Assignment 5 Visualization

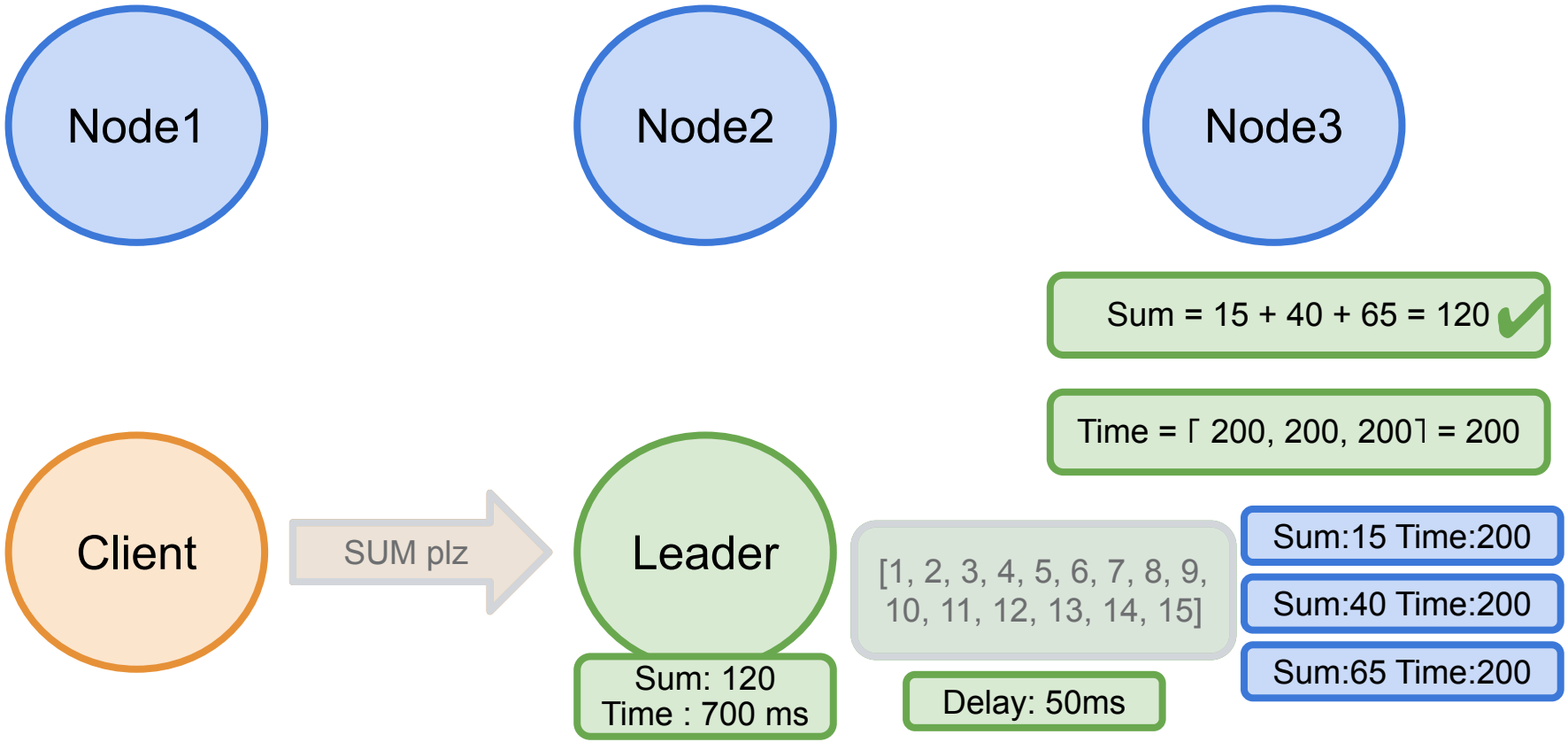
Let's depict the Example…



SER 321

Assignment 5 Visualization

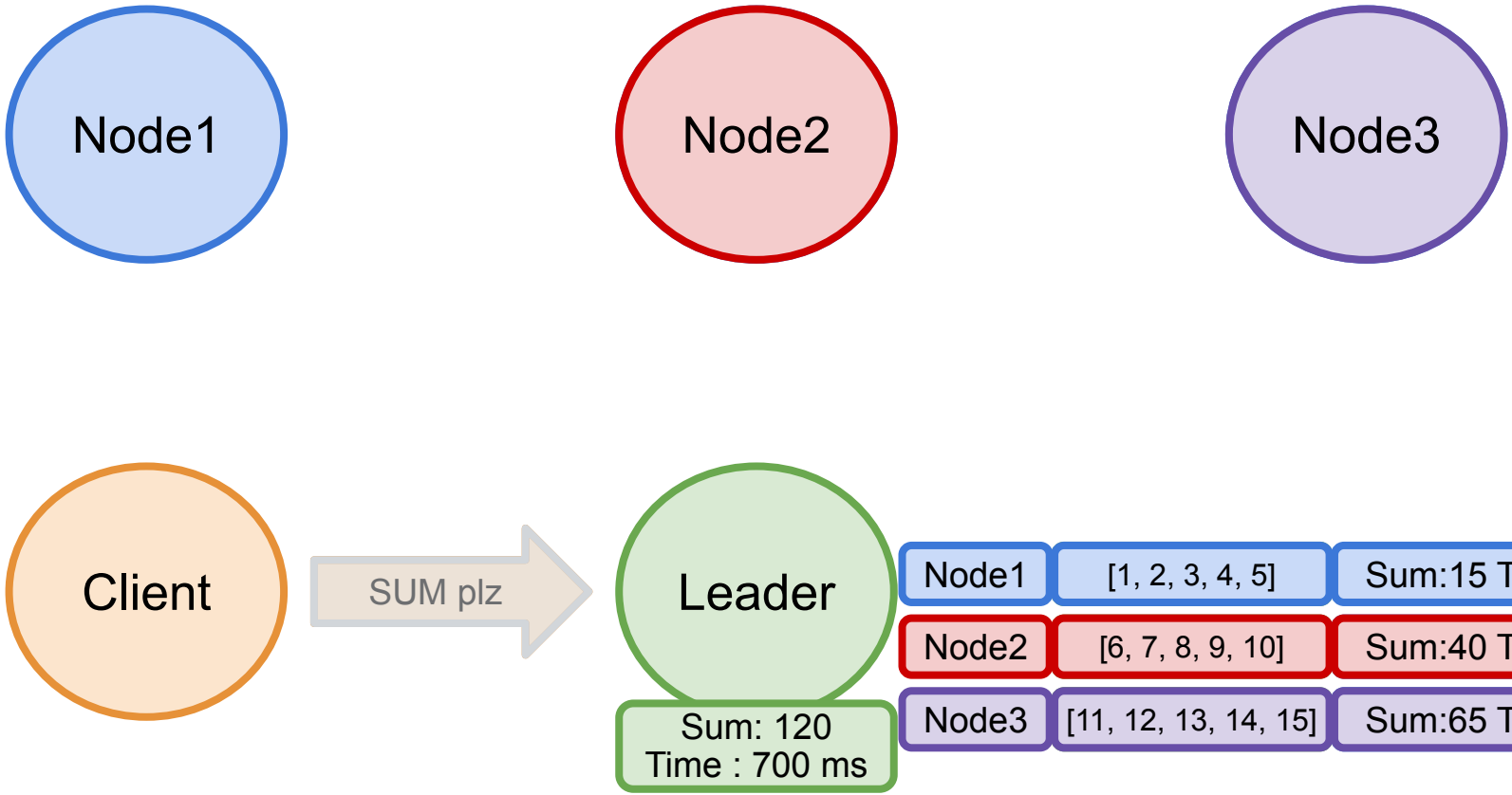
Let's depict the Example…



SER 321

Assignment 5 Visualization

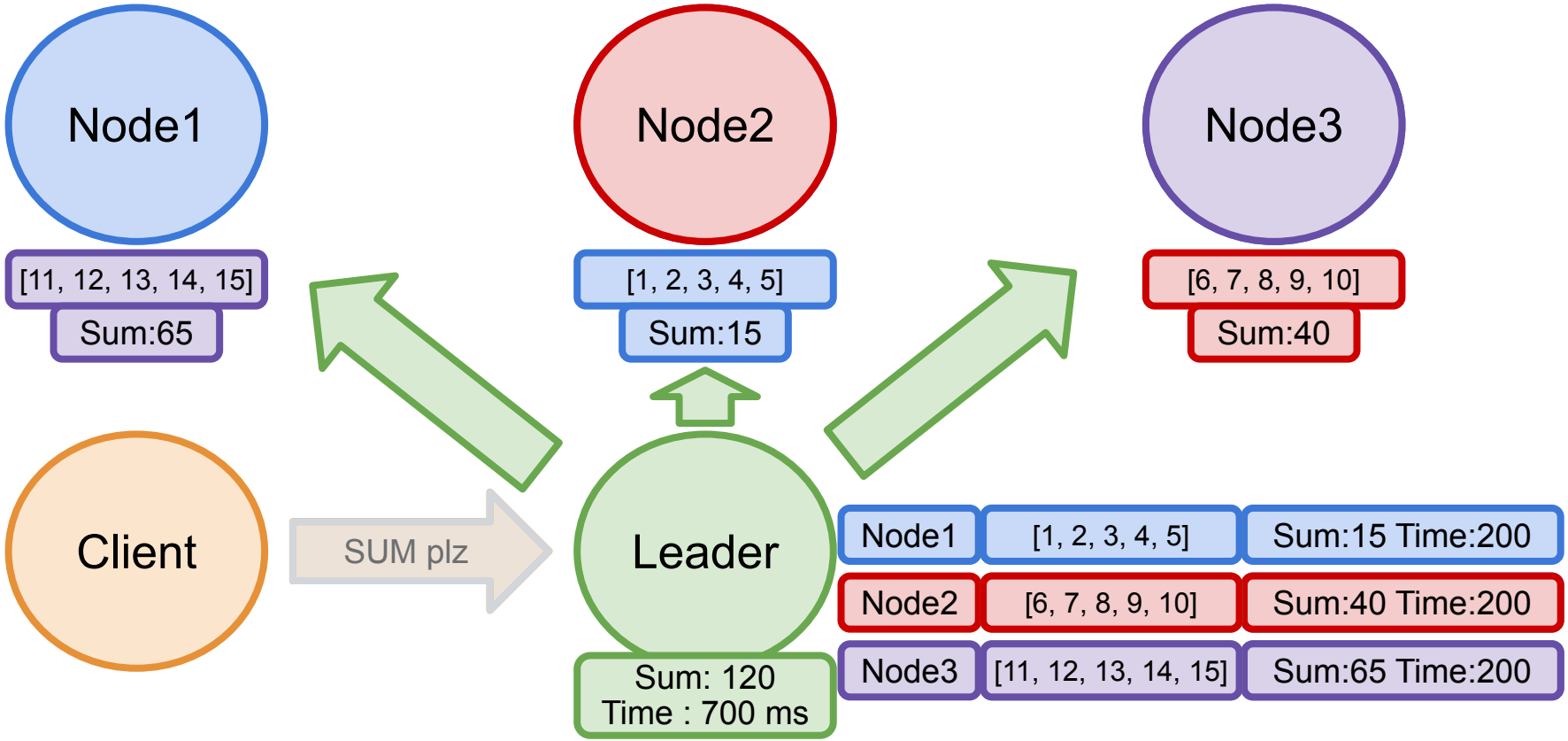
Let's depict the Example…



SER 321

Assignment 5 Visualization

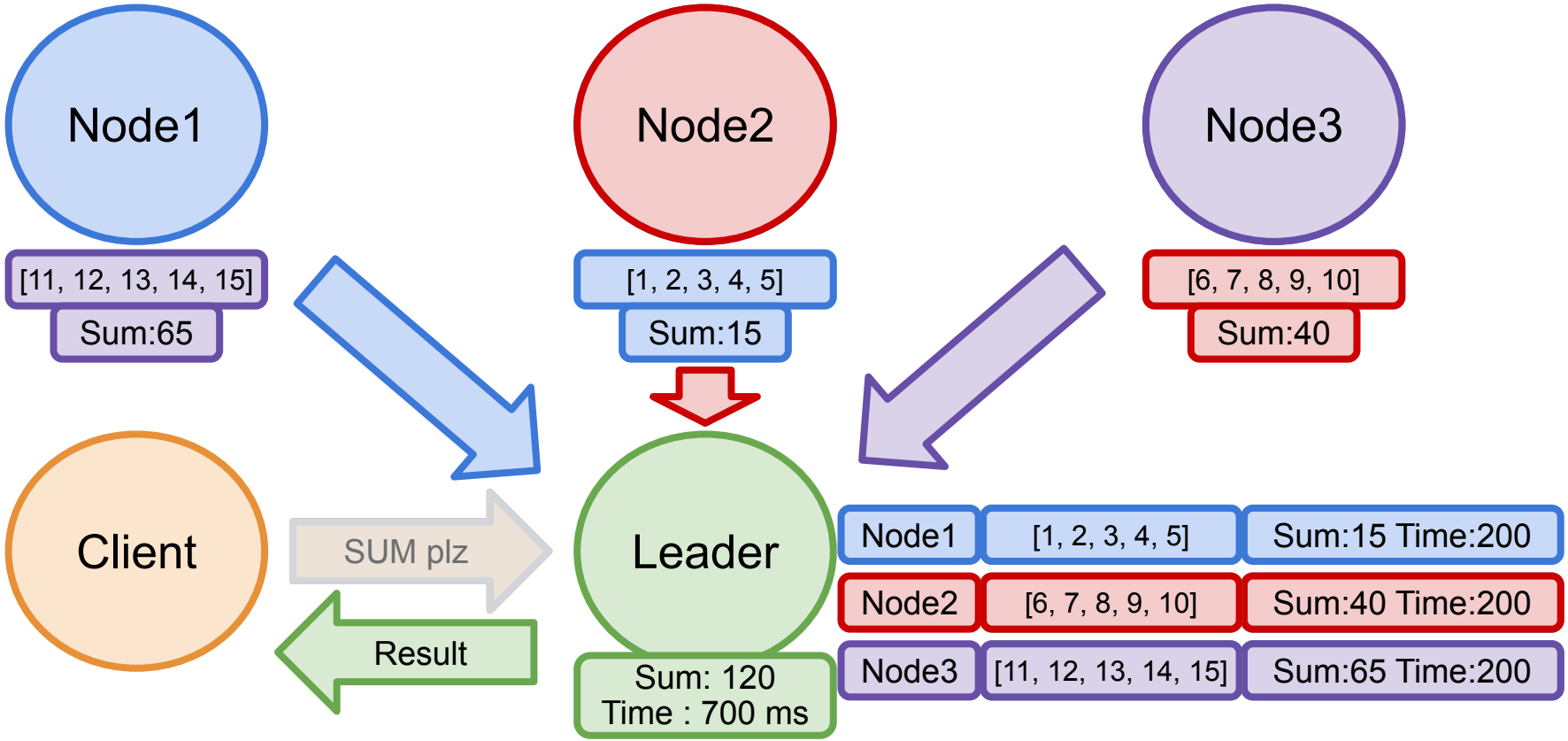
Let's depict the Example…



SER 321

Assignment 5 Visualization

Let's depict the Example…



SER 321

Scratch Space

Upcoming Events

SI Sessions:

- Tuesday, February 25th at 11:00 am MST
- Thursday, February 27th at 7:00 pm MST - **Exam Review Session (2hrs)**
- 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**

Questions?

Survey:

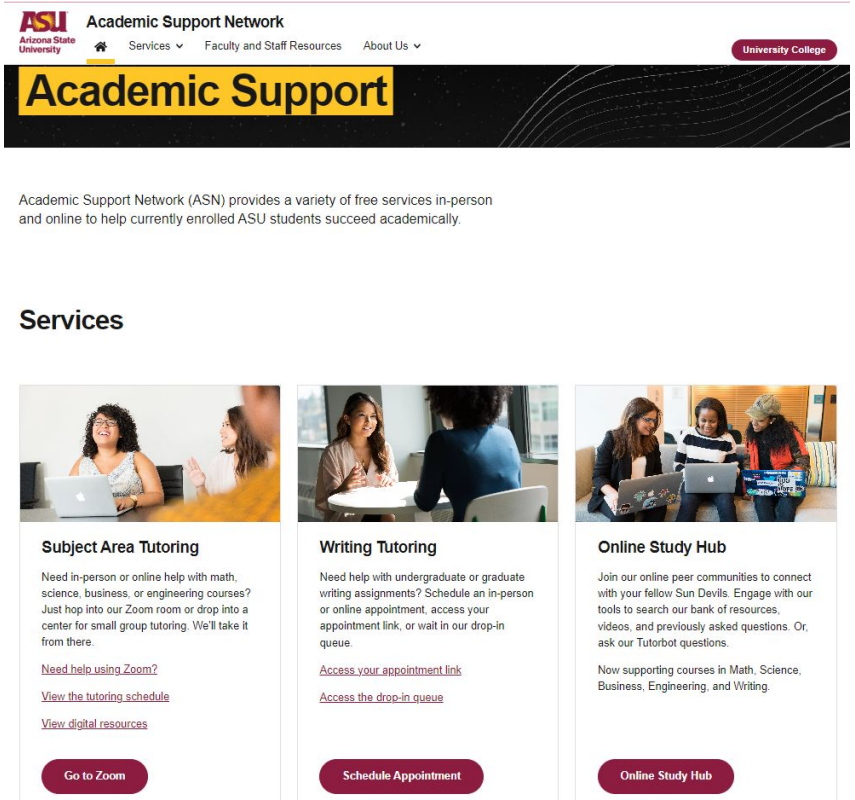
https://asuasn.info/ASNSurvey

37

More Questions?

Check out our other resources!

tutoring.asu.edu

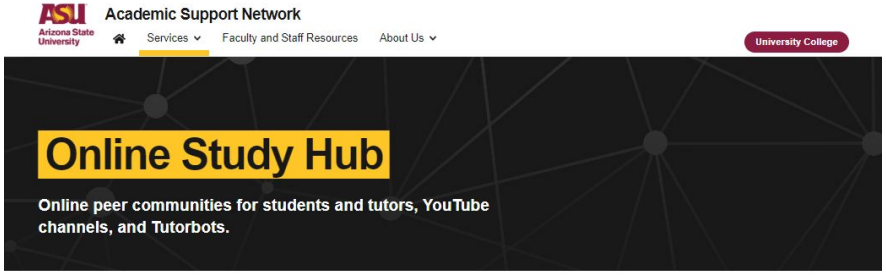


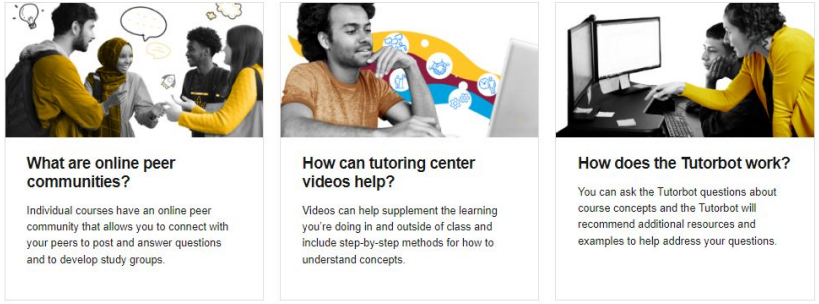
1. **Click on ‘Go to Zoom’ to log onto our Online Tutoring Center.**
2. **Click on ‘View the tutoring schedule’ to see when tutors are available for specific courses.**

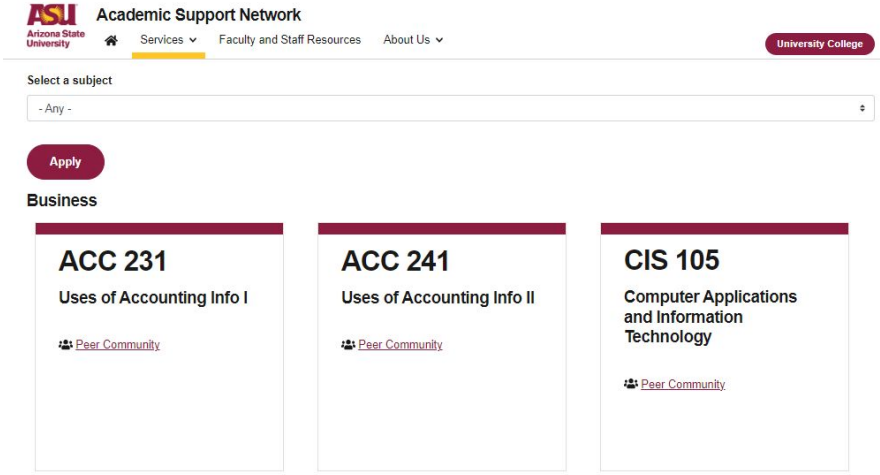
More Questions?

Check out our other resources!

tutoring.asu.edu/online-study-hub



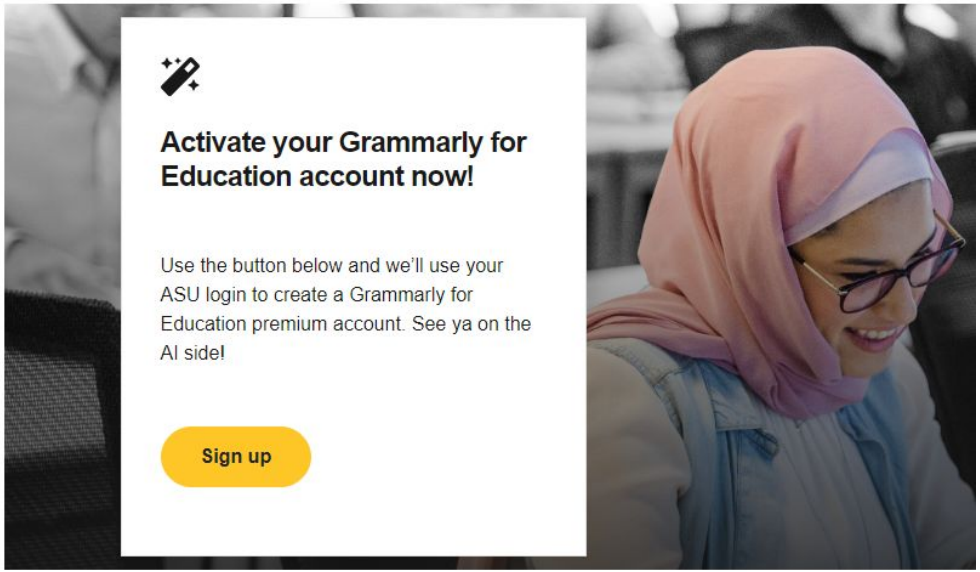




Don't forget to check out the Online Study Hub for additional resources!

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