Project 1: MOOC Communication Backbone

What is a MOOC?
What is a communication backbone?

Technical constraints (the remainder is up to you!):

Language:	Client: Python Lua (optional) Server: Java, Scala, Python, …
Tech Base:	Netty with Protobuf
UI:	None!
Database:	Your choice (if you consider a NoSQL, e.g., MongoDB, justify req. to choice)

Goals:

Your team has been asked to come up with a new approach to storing, finding and doing stuff in a large, decentralized, network of servers. To this goal, you are given the responsibility to design and build the MOOC communication architecture. This should include strategies for scalability and distributed coordination. Secondary consideration is determining health and status of your system.

Overall you should include:

- Collaborative MOOCs require a flexible network to support the entropy (churn) of connectivity. The churn can be found in:
  - Intra-cluster – How does a MOOC scale and coordinate activity?
  - Inter-cluster – How do MOOCs interact?
  - How do you maintain and monitor your cluster? Hint: should include awareness, near-time, of the cluster's state and external relationships (hence an adaptable backbone)
- No centralized coordination (no SPOF!).
  - The ability for clusters of groups to join (discovery) other groups and participate in cooperative problem solving at runtime.
  - Coordination - any cluster can propose problems, interact with other clusters (a.k.a. discussions), and cooperatively reach a solution (consensus gathering).
- While a MOOC provides many UI collaboration features. The constraints of our project time does not allow a full featured network. However, you should be aware and design for the possibility of new functionality to be added.
- **Our goal is to create the comm. infrastructure of MOOC clusters** (a team's PCs represent virtual groups (classrooms) compete within the larger community
- No databases! Each cluster must save state.
  - A cluster cannot directly share data with another cluster.
  - What is the design for fault tolerance of the database?
  - How do you maintain in-memory consistency?

While Web Services (RESTful) seems like a natural method of implementing our MOOC (and is likely a strong candidate), REST is not the learning objective of this project. This project focuses on communication performance, overlay network flexibility, consensus through leader election and voting.

Hints:

- Coordinate, agreement, organize, chunk, and consider how your system handles failure-recovery. What does it mean to `fail fast`?

Note likely we will need to refine these goals as the project progresses