# Designing GMAIL (Email Messaging System)

# Few clarifications on the assumptions that I am taking before designing such system:

* Core focus is to design email functionality only
* Requirements:
  + Send emails
  + Receive emails
  + Search emails (Either in sent emails or received emails).
  + Create labels for classification
  + Marking emails with certain tags like staring, important, to-do
  + Performing actions like: Snoozing, Scheduling, Archiving, Moving emails from one section to another
  + Real time updates for cross device synchronization

# Few clarifications on system scaling:

* Considering the popularity of the GMAIL application, I am assuming the daily active count should be in the range of billions.
* Also, this application will be used worldwide i.e., the geographical location doesn’t matter.

# Flow diagrams explaining the design of send and receive emails are attached on the following link:

* Overall flow diagram
* System design flow explaining the architecture for sending & receiving emails

# Flow Tech stack overview:

* Java + Spring Boot + Cloud services for distributed system
* Apache Kafka for higher reliability & assurance of sending message between 2 parties
* MongoDB for storing email metadata
* Google Cloud Storage/buckets for storing files/attachments
* Elastic search Lucene for faster global search
* Actuator service for health check-ups
* API Gateway: An entry point for user authorization and authentication and will be using cookies with expiration time to save user sessions on the browser

# List of APIs in the Real time service:

* Dashboard List API (Invoking Elastic search)
  + Endpoint: /{version}/messages/
  + Query Params:
    - By Date Range
    - By Keyword
    - By text
    - From given email addresses or to given email addresses
    - Having attachments etc.
* fetchEmail
  + Endpoint: /{version}/message/{id}
* Search Emails Globally (Invoking Elastic search)
  + Endpoint: /version/message/{text}
* Fetch Folder Details
  + Endpoint: /{version}/folder/{id}

Few Important Notes:

* **I am not listing all the basic API details other than above, since I am more focusing on the core functionality, those are:**
  + **Tagging an email as SPAM**
  + **Marking an email with STAR**
  + **Moving email from one folder to another and other related tasks**
* **I haven’t provided any details on the DB schema part due to time constraints since it will take some time to design a schema that is open for extension**