RBC central server
- Registry of which devices contain which layers of the model
etc: [phone1 → layers 1 to 8
phone2 → layers 9 to 16
phonen → layers 72 to 80]

Also, store an 8-bit quantized LLM: 8-bit quantized Deepseek, 80 layers

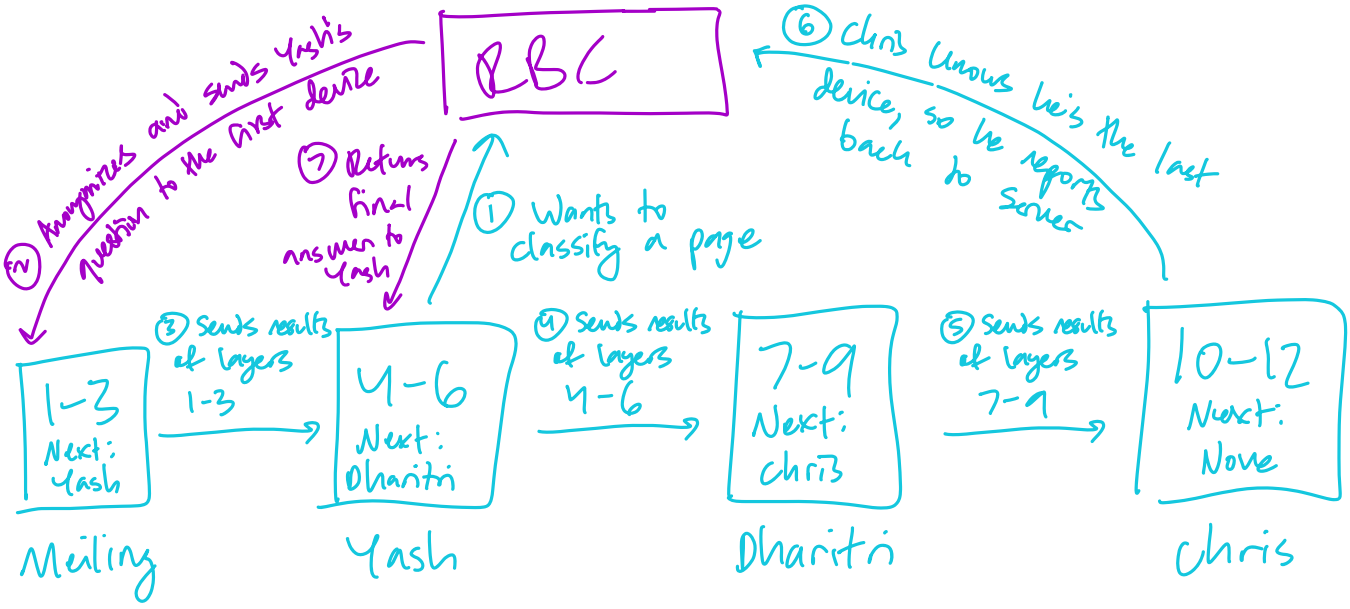
And a "queue" of which layers have the least number of devices storing them

Devices store:
- Their 8-bit quantized layer files
- The device that comes "next" in the chain of layers
  - So phone 1 should know to send its data to phone 2 (or another phone with layers 9 to 16 if the connection to phone 2 fails/ want to give phone 2 a break for load-balancing reasons)

Updates:
- When a phone joins the network (downloads app), assign and download its layers based on the queue
- When a phone leaves the network, update the queue accordingly

Meiling: Layers 1-3
Yash: Layers 4-6
Dharitri: Layers 7-9
Chris: Layers 10-12

Consider a 12-layer LLM distributed like so:



- Devices don't have access to user info or who initiated the request, only RBC has that information
- No API call to a third party-hosted LLM
- Layer and chain assignments can refresh periodically with software updates