* Hello, Tayeb el naharda hanetkalem 3an el gRPC, eh heya w eh el far2 beinha w bein el REST APIs
* Di el agenda bta3etna lel naharda fa 5alina nebda2
* Abl ma netkalem 3an el gRPC, 5alina n-recap sari3an fekret el APIs .. heya el set of rules eli bet-allow el communication for example bein el client wel server. Zay el example eli I think most of us 3adda 3aleina, aw this is how I learnt it :D eno ka2eno el waiter eli bey-serve el akl men el matba5 lel 7ad eli 3ayez yakol :D  
  How they work b2a? eno el client bey-make el request – b3den el API bey-process el request dah b eno bey-validate el route da – then el server nfso (fel 3orf bta3na howa el BE) beyebda2 eno actually bey-perform el action, for example bey-retrieve el needed data – finally el response beyerga3 tani lel client 3an tari2 el API
* Tayeb eh el REST APIs? Sari3an brdo heya Architectural style based on HTTP bey-define el communication 3al web beykon 3amel ezay. Beyesta5dem HTTP verbs eli homa (GET, POST, PUT, DELETE) – bey-treat kol 7aga 3ala enaha unique URL – w tab3an el mota3araf 3aleh eno el data format beykoun in form of json aw xml
* Keda we recapped sari3an 3an el definition of APIs wel REST APIs .. eh heya el gRPC? Ka definition heya e5tsar le Google Remote Procedure Call, heya open-source high-performance framework bey5alina ne2dar n call functions 3al remote servers ka2enohom mawgodin locally 3andena fel code. Liih shwayet specifications 5alina n2olhom: zay ma olna howa based on RPC y3ni a2dar a-call functions fih directly – (feature mohema gedan) en howa built on HTTP/2 w one of the most important thing f HTTP/2 eno bey-support el multiplexing. Tab eh howa el multiplexing? El awel 5alina n2ol eno HTTP/1 bey-load ay resource one after the other, fa lw one resource makansh loaded, hay-block kol el el 7agat eli b3do. On the other hand, HTTP/2 bey-establish single connection to send multiple streams of data at once, fa mafish ay resource bey-block the other w da howa el multiplexing. – w therefore, da bey5ali el streaming bein el client wel server as-hal – w finally beyesta5dem 7aga esmaha Protocol Buffers lel data serialization  
  Tayeb ana 3amalt frontend analogy for this part, eno lw hanfakar fel REST APIs heya ka2enaha Vanilla JS, laken el gRPC, ka2enaha Typescript (enaha strictly typed)
* Tayeb e7na olna eno el gRPC beyesta5dem Protocol Buffers lel data serialization -> 5alina nshof eh heya el protocol buffers asasan. Heya simply bet-define el shape of the data eli hat-get transferred fel application bt3na.   
  Fa el frontend analogy here haykon ka2eno el protobuf howa interface in typescript, eno bey-define shakl el data 3amel ezay.  
  Da example le shakl el protocol buffer 3amel ezay , fel awel ben-define eno el syntax of the file is “proto3” eli howa protocol buffer v3 – Message di bet-define shakl el single data record fa fel 7ala di law e7na han-get User aw n-create one, han-return User el interface bta3o shaklo kda (gowah ID, Name, Email w hobbies masalan), repeated hena m3naha ka2enaha array of strings (Tayeb side note lw 7ad bey-wonder eh heya =1 w =2 eli mawgodin dol, howa fel protocol buffers, kol field fel message lazem ykon liih unique number, w da mainly beysa3ed fel parsing lama y7sal serialize aw deserialize lel data, el parser beyesta5dem el numbers di 3ashan y-map el binary data back lel correct fields.) – Service hena ka2eni b2a ba define el actual endpoints eli hat-get called, fa in this protobuff, ha-define for example eno I need to getUser masalan w CreateUser
* Tayeb 5alina nshof full example hena. Same example implemented b both REST w gRPC  
  Fel example hena, el REST server hena 3ady ba-allow CORS fel configuration beta3o, w b3den ba define GET endpoint mohemet-ha enaha t-return items  
  Men na7yet el client pretty straight forward, eni ba call el endpoint, w b3den ha7tag a parse el json 3ashan a5od meno el info eli ana 3ayzaha .. Tayeb da fel REST eli kolena 3arfeno, ezay b2a a3mel 7aga zay di fel gRPC  
  Zay ma olna ha7tag a-define shakl el data 3ashan in this case msh hatkon JSON, hateb2a compact binary format eli howa f shakl el protocol buffers. Fa tab3an da asra3 fel parsing wel serialization in comparison lel JSON.  
  Fa hena ben-define eno el syntax protocol buffer version 3, w b3den ben2ol eno el request eli hangib bih el data, msh haykon liih parameters, eli howa Empty da. Bardo fi another data entity hatkoun el Item, eli 3obara 3an ID w Name w Value, b3den han-define el endpoints eli to be called by el clients, eli fel case di hatkoun getData, msh hata5od parameters w hat-return Data Response eli howa 3obara 3an array of Items.  
  5alina nro7 lel server, pretty much the same, bs hena I use el protobuff eli 3amalto 3ashan a5od meno el DataService eli menha h2dar a-call ay endpoint. Fa hena I called el GetData eli defined henak fl protobuff, w menha ba2ol eno return el items lw el function di called  
  Finally, el client, brdo ha load el protobuff, w hena ha call el GetData.
* Tayeb 5alini a2ol 7aga, eno ah el example da momken maybayensh el power of gRPC, leh? La2en el gRPC sor3etha msh betban fel basic CRUD operations, heya mainly betban fel 7agat eli feha streaming zay Netflix masalan aw chat applications aw even large payload data. 3ashan keda numbers hena might not illustrate much.
* Tayeb pros and cons le kol wa7da, 5alina n-highlight keda few ones  
  el gRPC tb3an performance-wise in some cases haykon a7san men el REST APIs w da bsabab el binary encoding w eno built over HTTP/2, tb3an type safety 3ashan el whole code shayef el protocol buffers w heya ka2enaha contract enaha bt-enforce el types, w brdo heya simply bet-support el data streaming, 3ashan heya bet-establish one TCP connection w momken tb3at multiple requests at once w y7sal multiplexing, fa el streaming asra3 w as-hal.  
  On the other hand b2a tb3an el browser support le HTTP/2 msh dayman beykon supported w el learning curve bt3ha sa3b shwaya  
  REST APIs tb3an are simple to use, w browser compatible. Bas brdo on the other hand, momken fel massive data ykon fi performance overhead 3ashan el parsing bta3 el json, w tb3an enaha makanetsh designed lel bidirectional communication, ah we can use websockets, bas msh da el power of REST.
* Tayeb emta a5tar meen? Mainly gRPC fel real-time streaming zy ma olna, w lw fi large data. W tab3an REST lw 3andi simple CRUD operations.
* Finally, da simple decision matrix kda momken ysa3edna n-decide in certain scenarios ne5tar which approach in such cases.