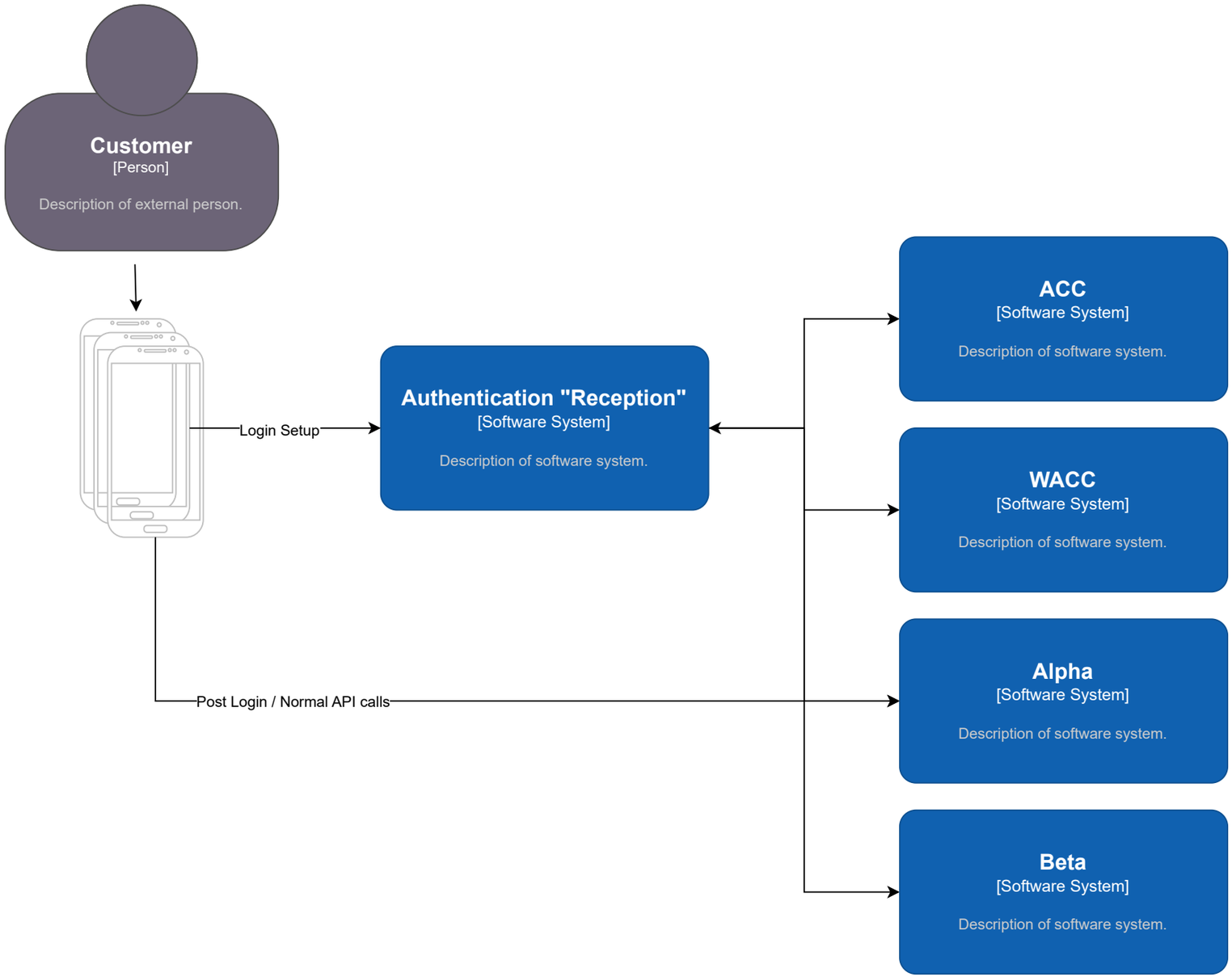
For questions, comments

https://docs.google.com/spreadsheets/d/1PRX9atdX-mb7hC9T7dQERcweiBIRs4i6M59dGRpeJsc/edit?usp=sharing



Login Flow

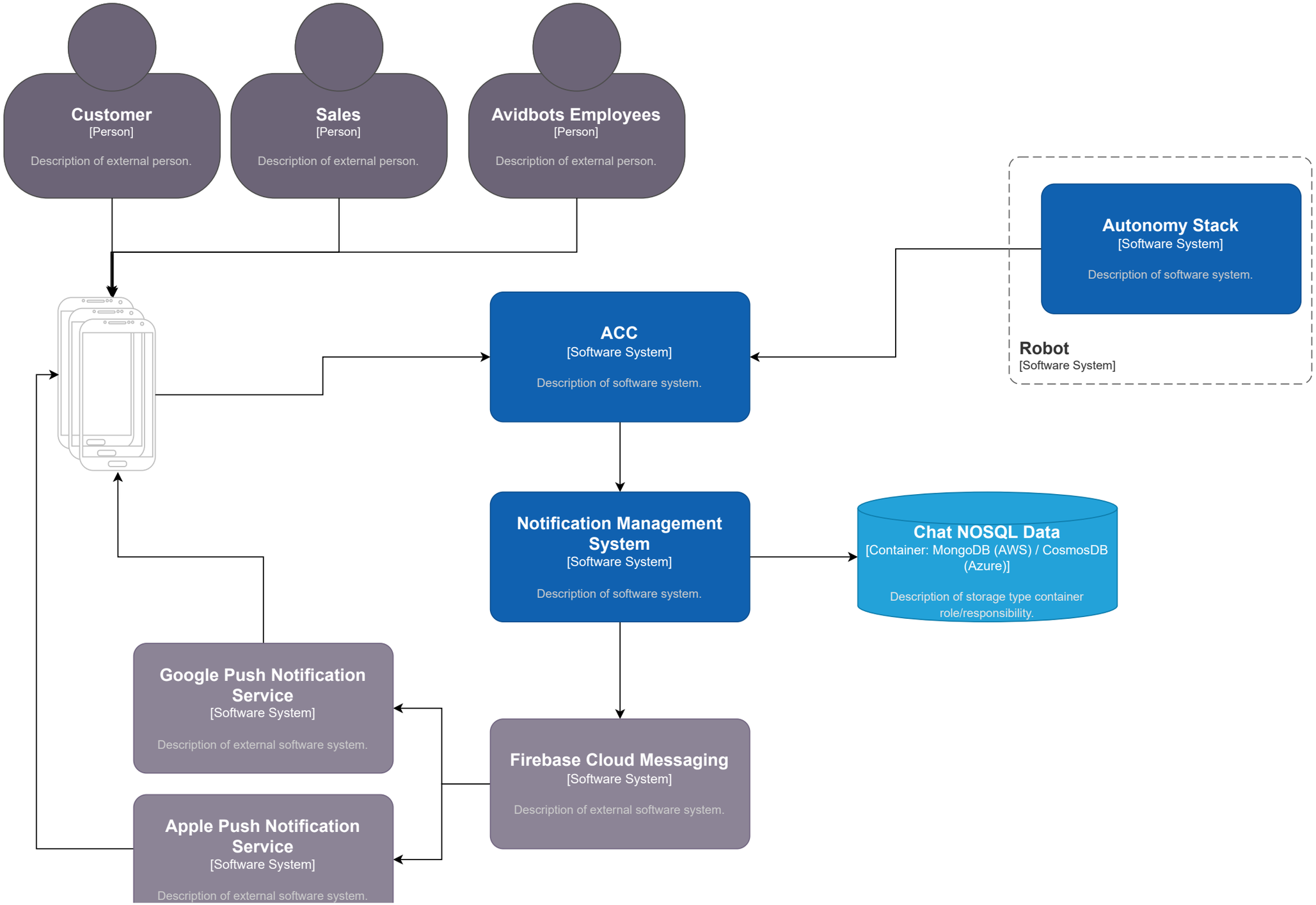
After entering a login ID/email address into the login pane of the phone, the phone will make a call to authentication reception that will list all the endpoints we support.

It asks each if the login is known.

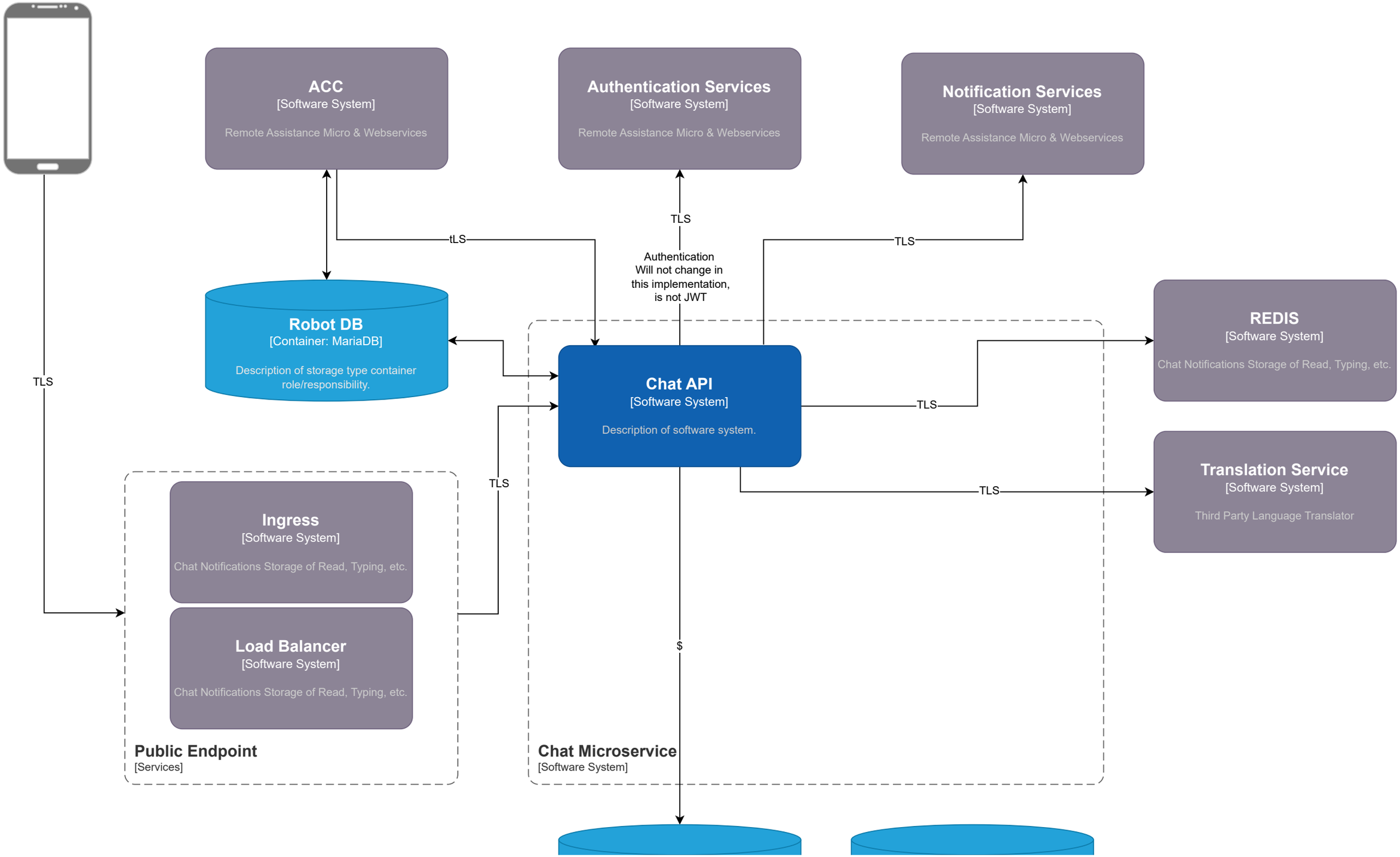
If the ACC site knows of a user with that name, it returns it's name and URI for subsequent access.

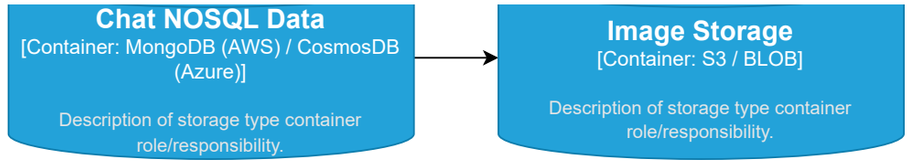
The phone will display a list of connection points if there are more than one, or use the sole access point (most customers will fall into this category)

Past that, the phone utilizes the connection point stored for all API access.


Customer
[Person]
Description of external person.
Sales
[Person]
Description of external person.
Avidbots Employees
[Person]
Description of external person.
Autonomy Stack
[Software System]
Description of software system.
Robot
[Software System]
ACC
[Software System]
Description of software system.
Notification Management System
[Software System]
Description of software system.
Chat NOSQL Data
[Container: MongoDB (AWS) / CosmosDB (Azure)]
Description of storage type container role/responsibility.
Google Push Notification Service
[Software System]
Description of external software system.
Firebase Cloud Messaging
[Software System]
Description of external software system.
Apple Push Notification Service
[Software System]
Description of external software system.

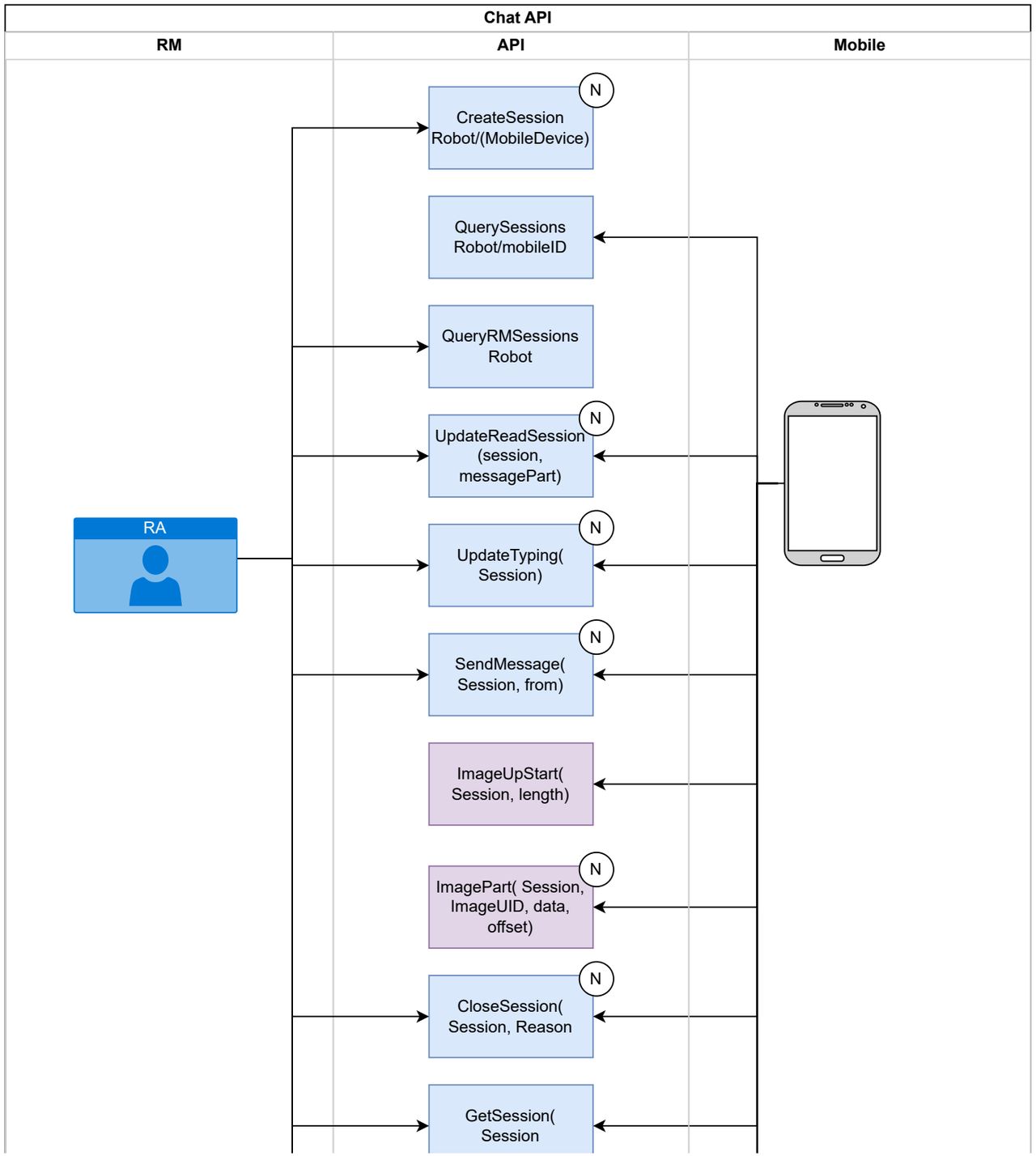
Push Messaging Flow


ACC
[Software System]
Remote Assistance Micro & Webservices
Authentication Services
[Software System]
Remote Assistance Micro & Webservices
Notification Services
[Software System]
Remote Assistance Micro & Webservices
TLS
tLS
TLS
Authentication
Will not change in
this implementation,
is not JWT
Robot DB
[Container: MariaDB]
Description of storage type container
role/responsibility.
REDIS
[Software System]
Chat Notifications Storage of Read, Typing, etc.
TLS
Chat API
[Software System]
Description of software system.
TLS
TLS
Translation Service
[Software System]
Third Party Language Translator
TLS
Ingress
[Software System]
Chat Notifications Storage of Read, Typing, etc.
Load Balancer
[Software System]
Chat Notifications Storage of Read, Typing, etc.
Public Endpoint
[Services]
$
Chat Microservice
[Software System]

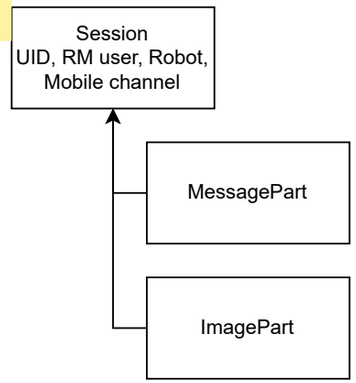


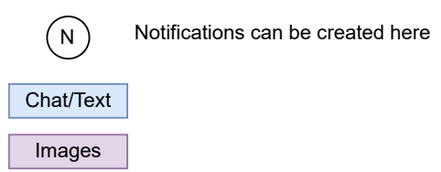
Secure connection terminates at load balancer. All the microservices talk to each other ober http

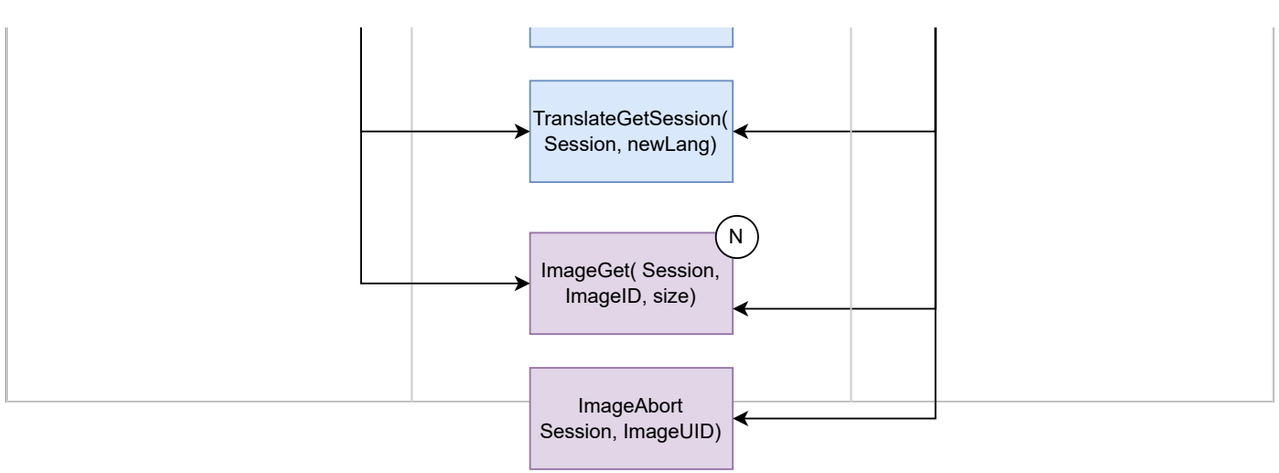
We need Data service to access S3 bucket



If we are sending notification to multiple devices, how do we decide which mobile device to be in the session?

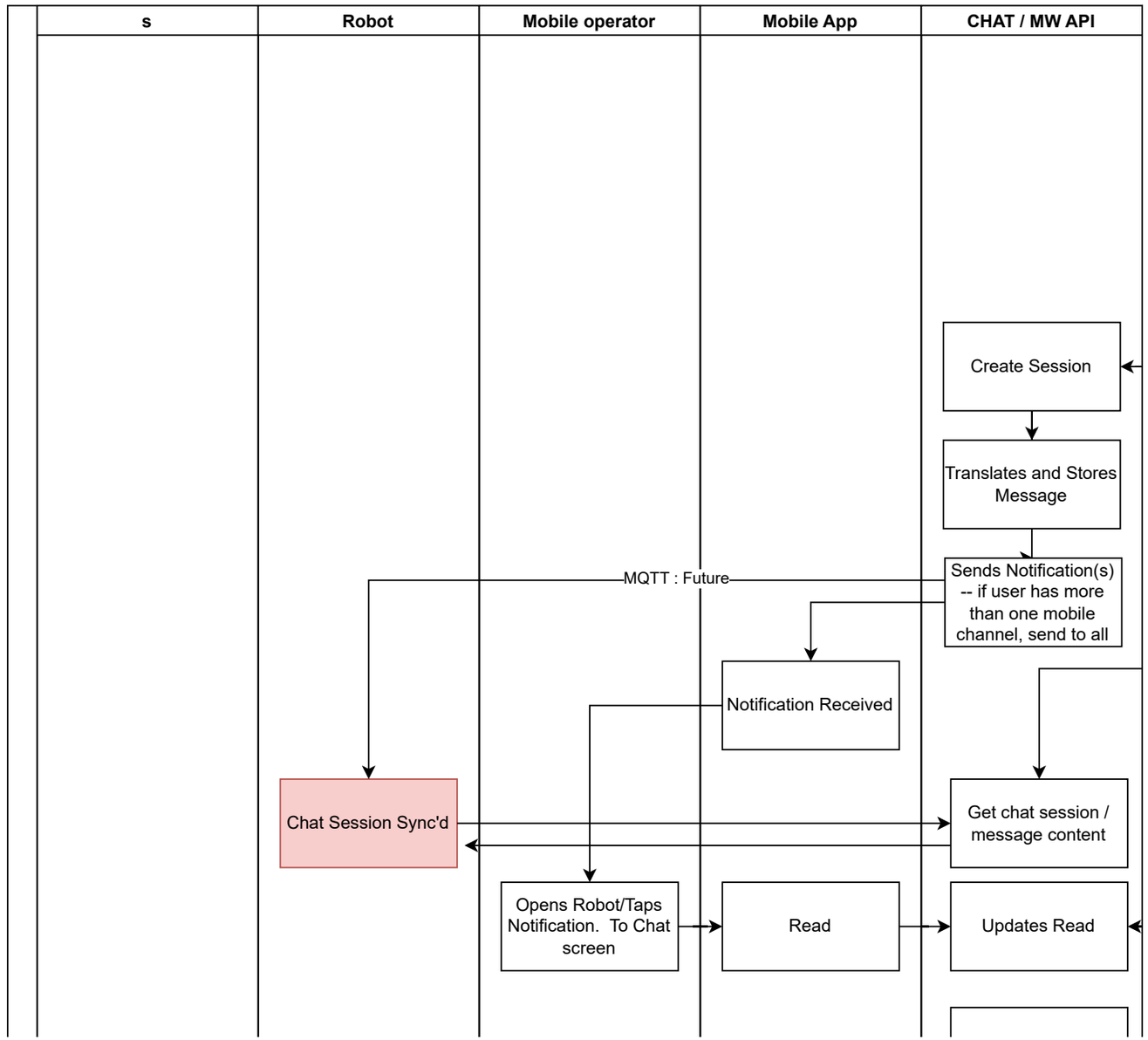


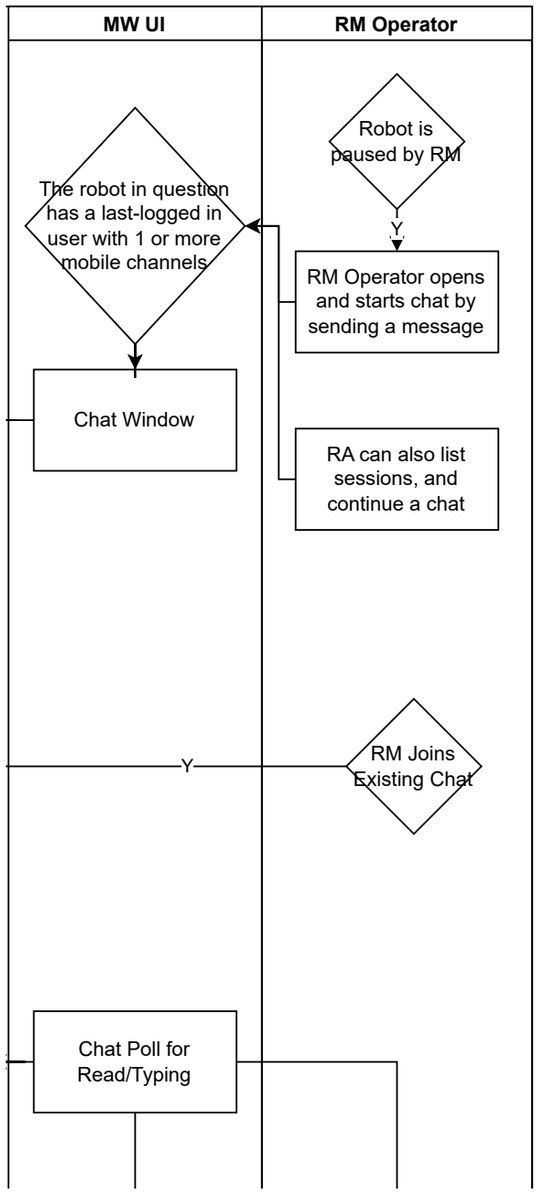



TranslateGetSession(
Session, newLang)
N
ImageGet(Session,
ImageID, size)
ImageAbort
Session, ImageUID)

Where do we add the translation part?

I thought Product requirement for robot side was unchanged. Which means Robot will continue to receive the canned messages.





Chat button only shows IF there's a mobile app connected to the last physically logged in user at the robot (we'll use the authentication log)

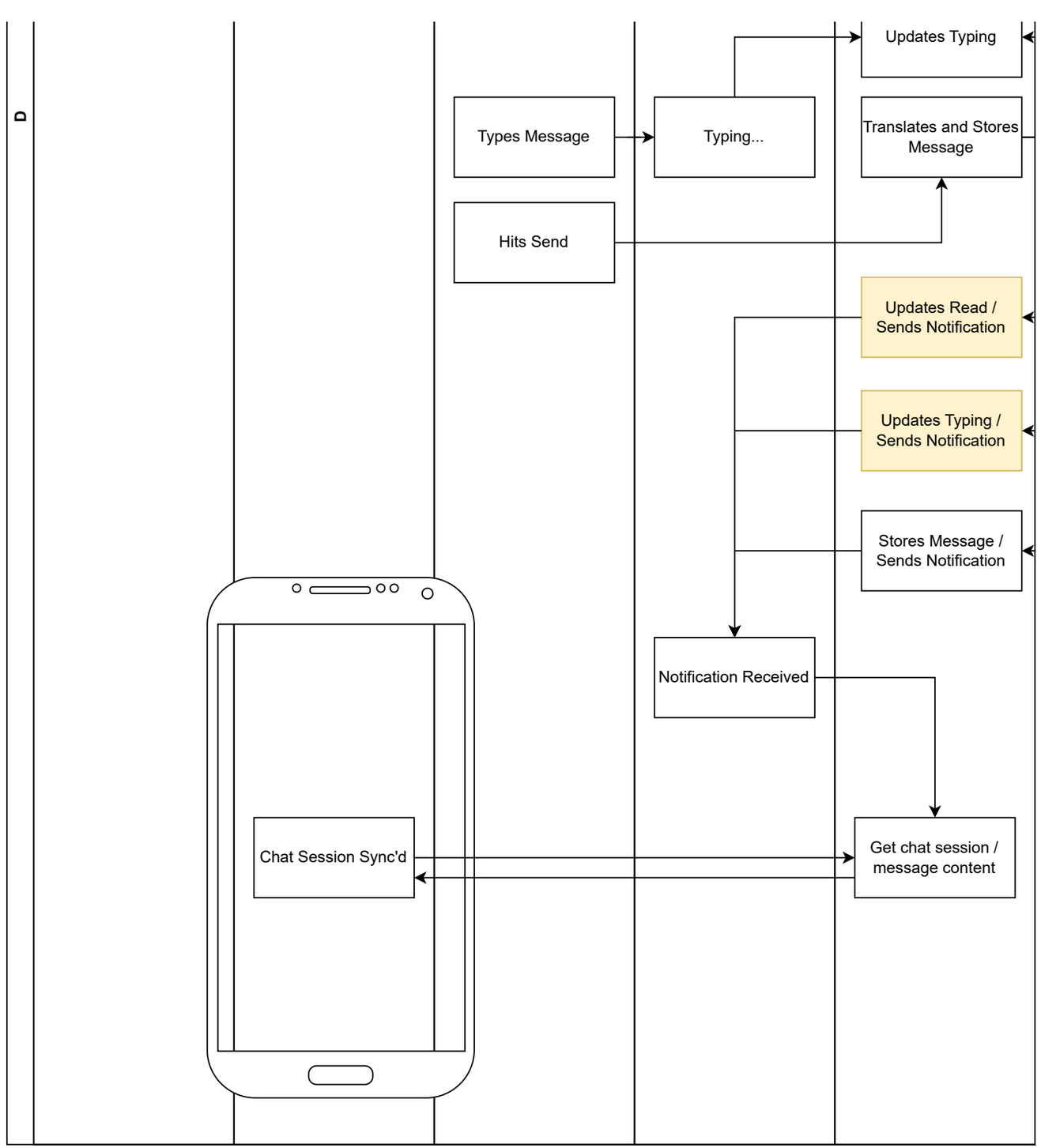
Q: Do we need to store the sessions in a database for tracking purpose? or Can we use redis cache to store the session?
Note :We can keep the session id in the chat history table

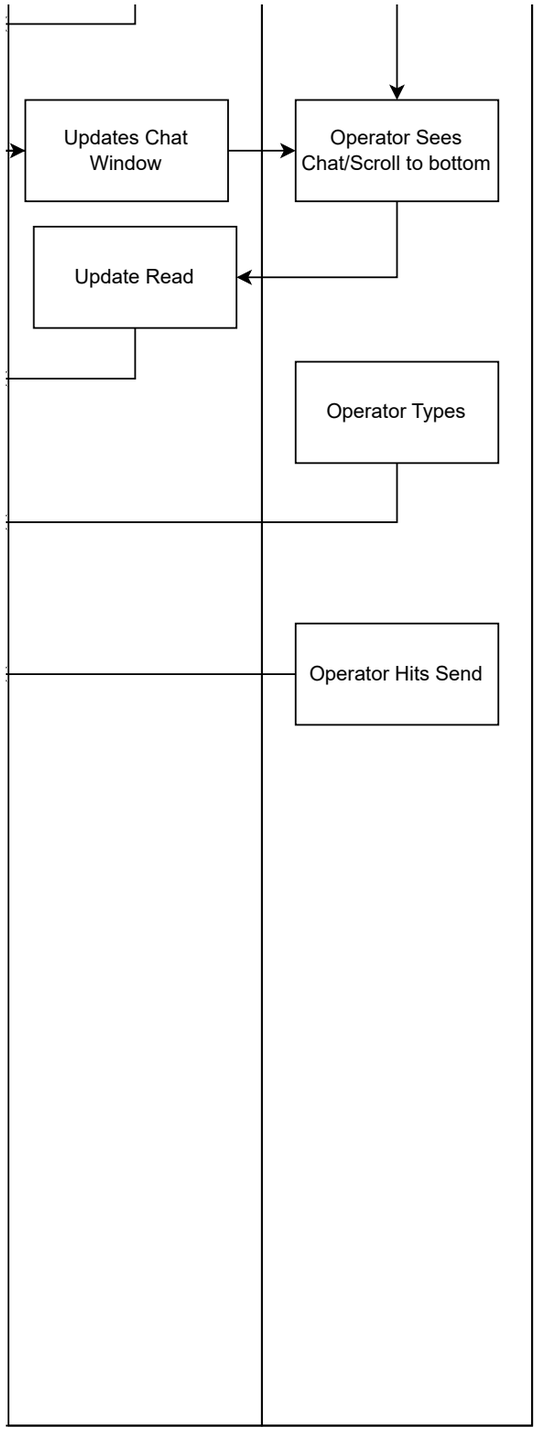
Q :How do we send push notifications to multiple device?
A: We could use the notification service to pull list of devices active for that user and send notification to all.

Future - Do NOT IMPLEMENT

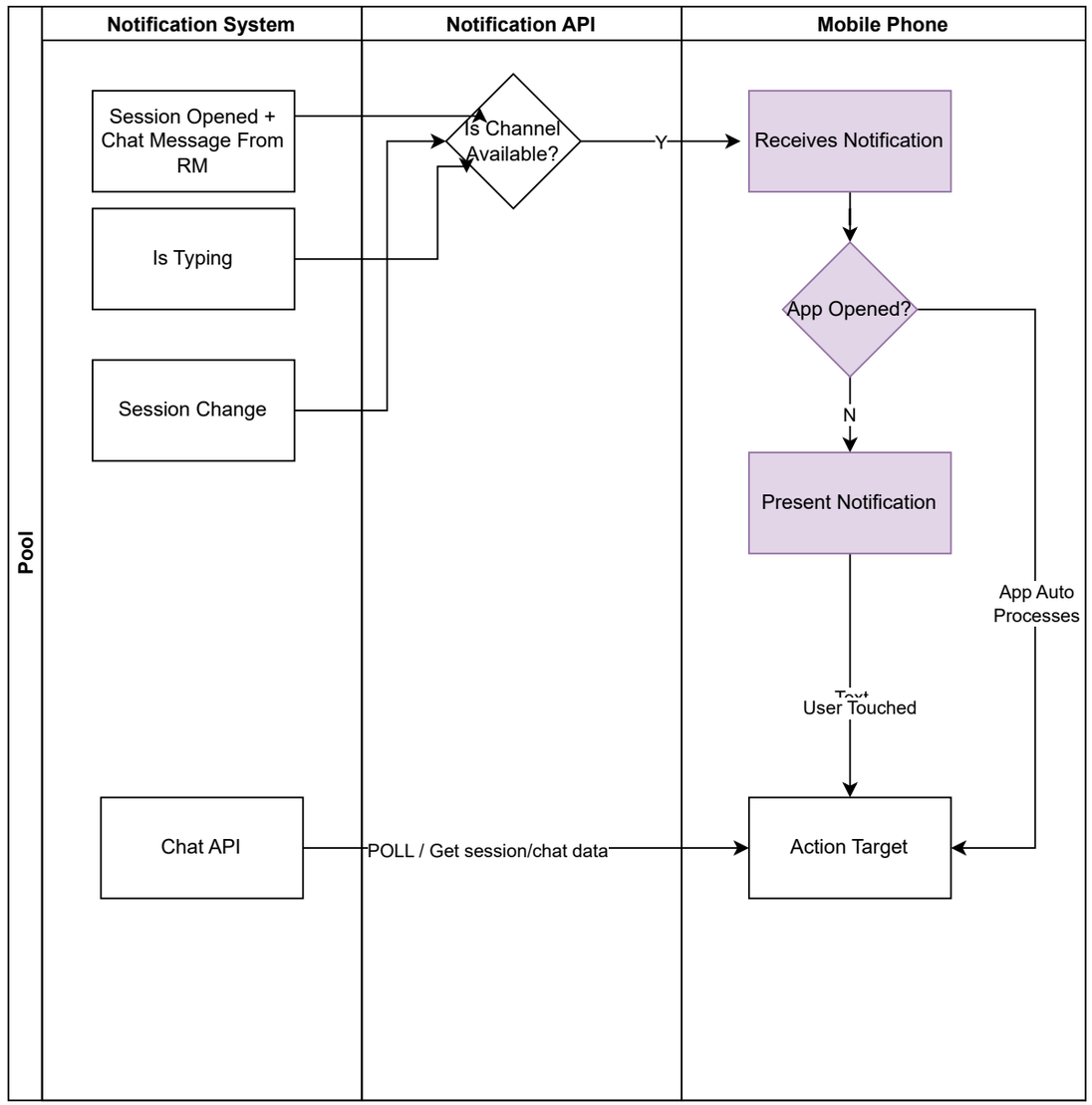
Could be optional

For real time statues what can we use? Can we use web sockets?

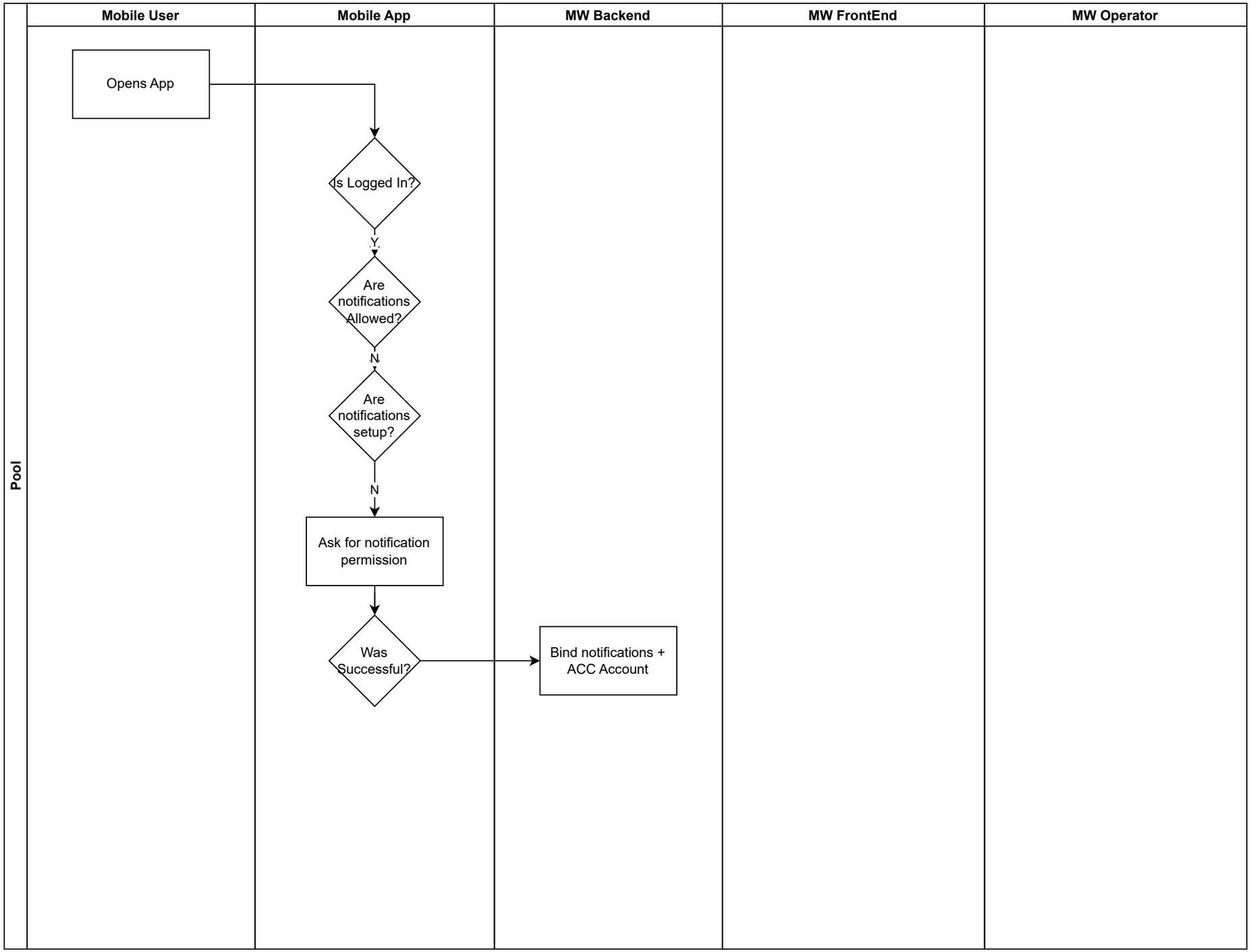

D
Updates Typing
Types Message
Typing...
Translates and Stores Message
Hits Send
Updates Read / Sends Notification
Updates Typing / Sends Notification
Stores Message / Sends Notification
Notification Received
Chat Session Sync'd
Get chat session / message content

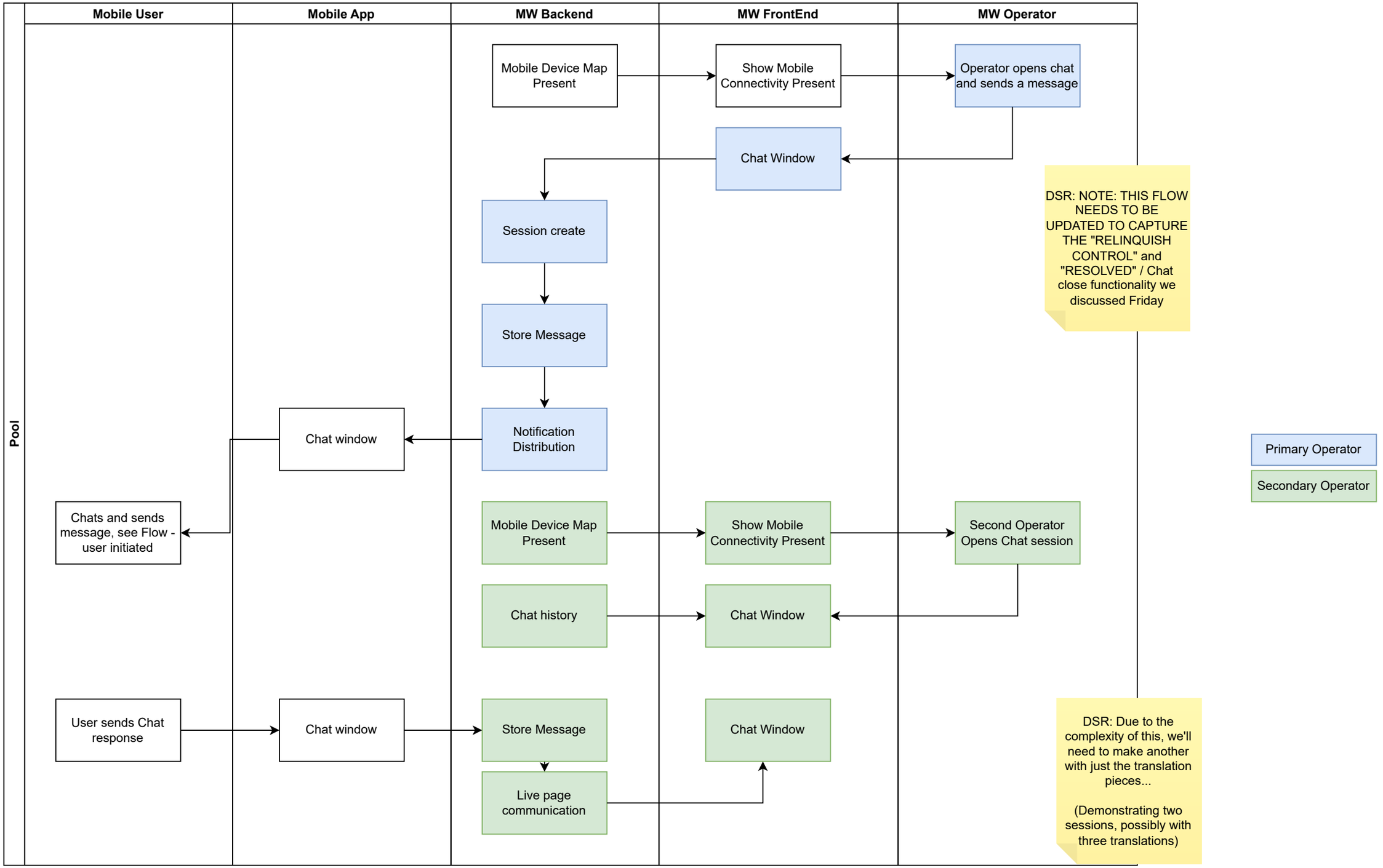


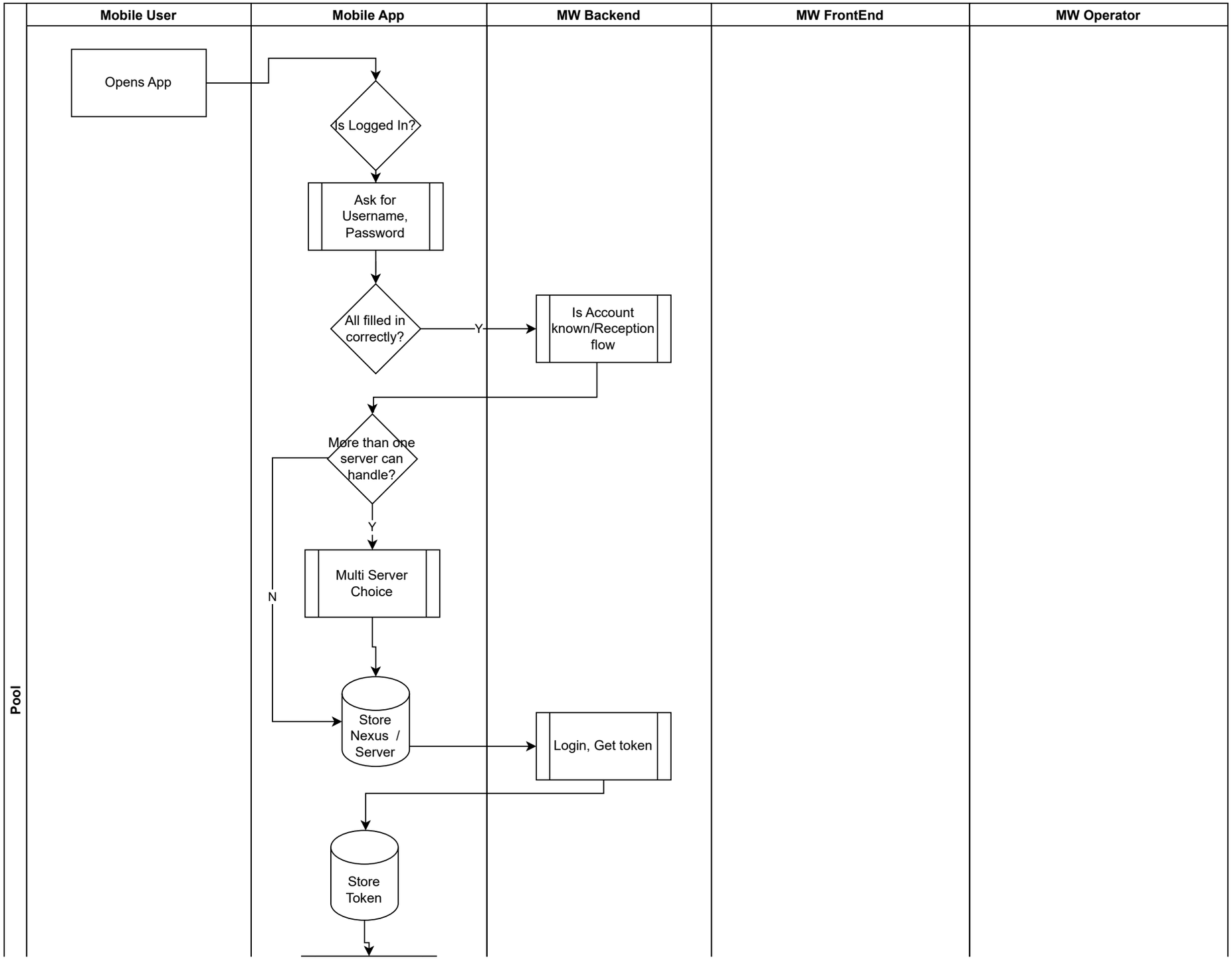
Like Typing / Read/ unread / received

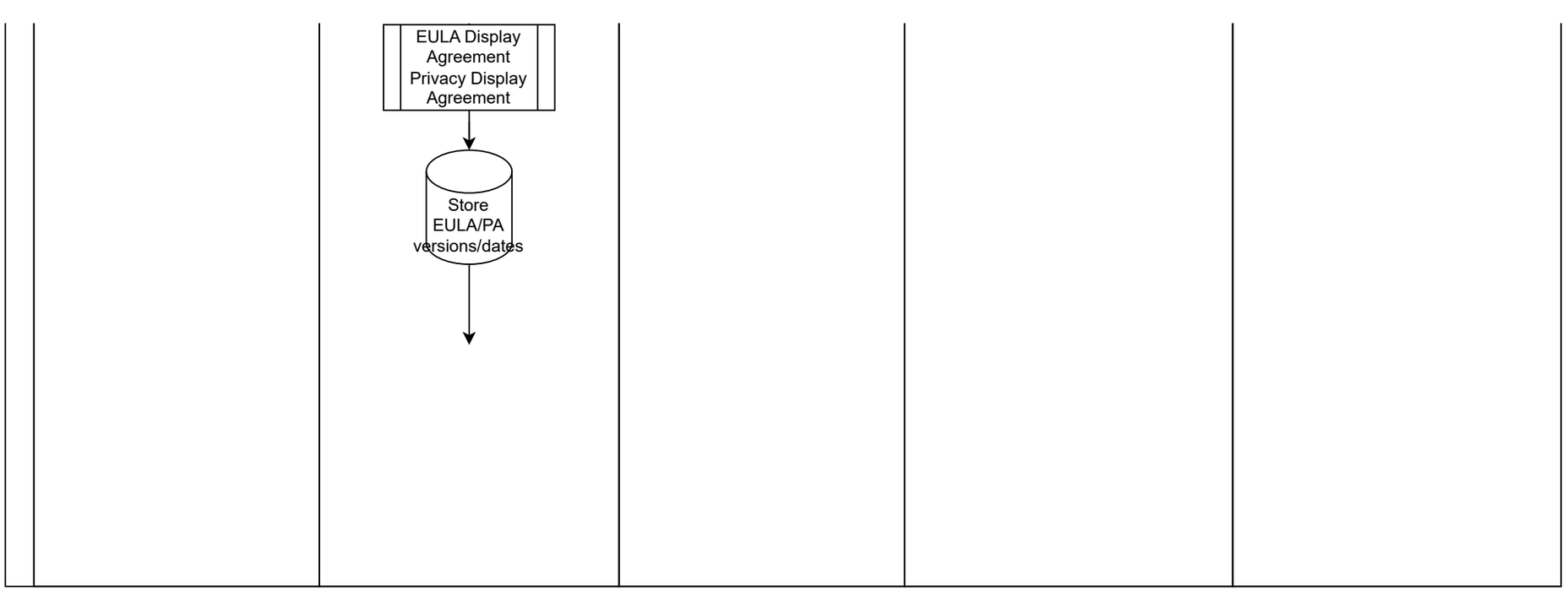


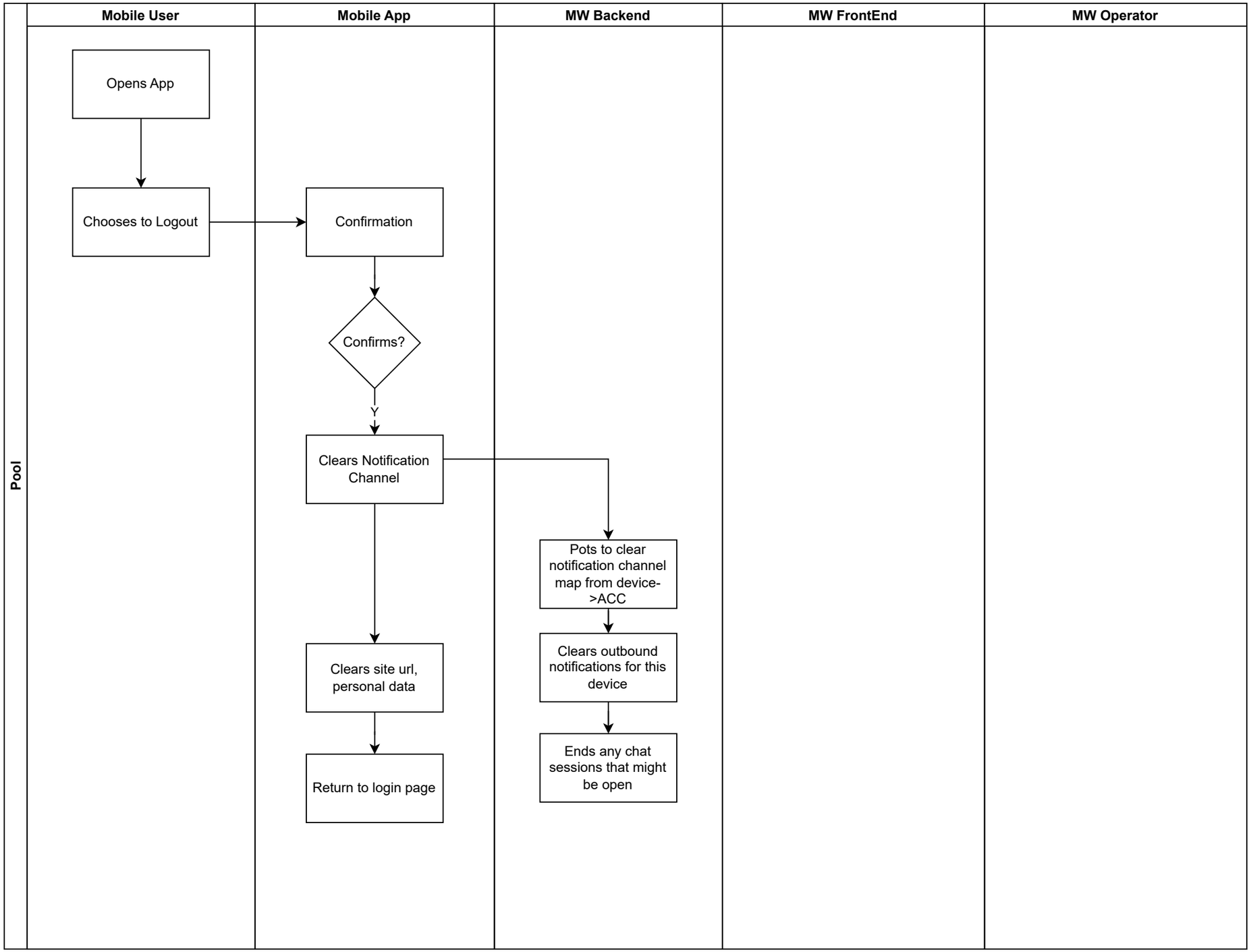
Native Phone Responsibility

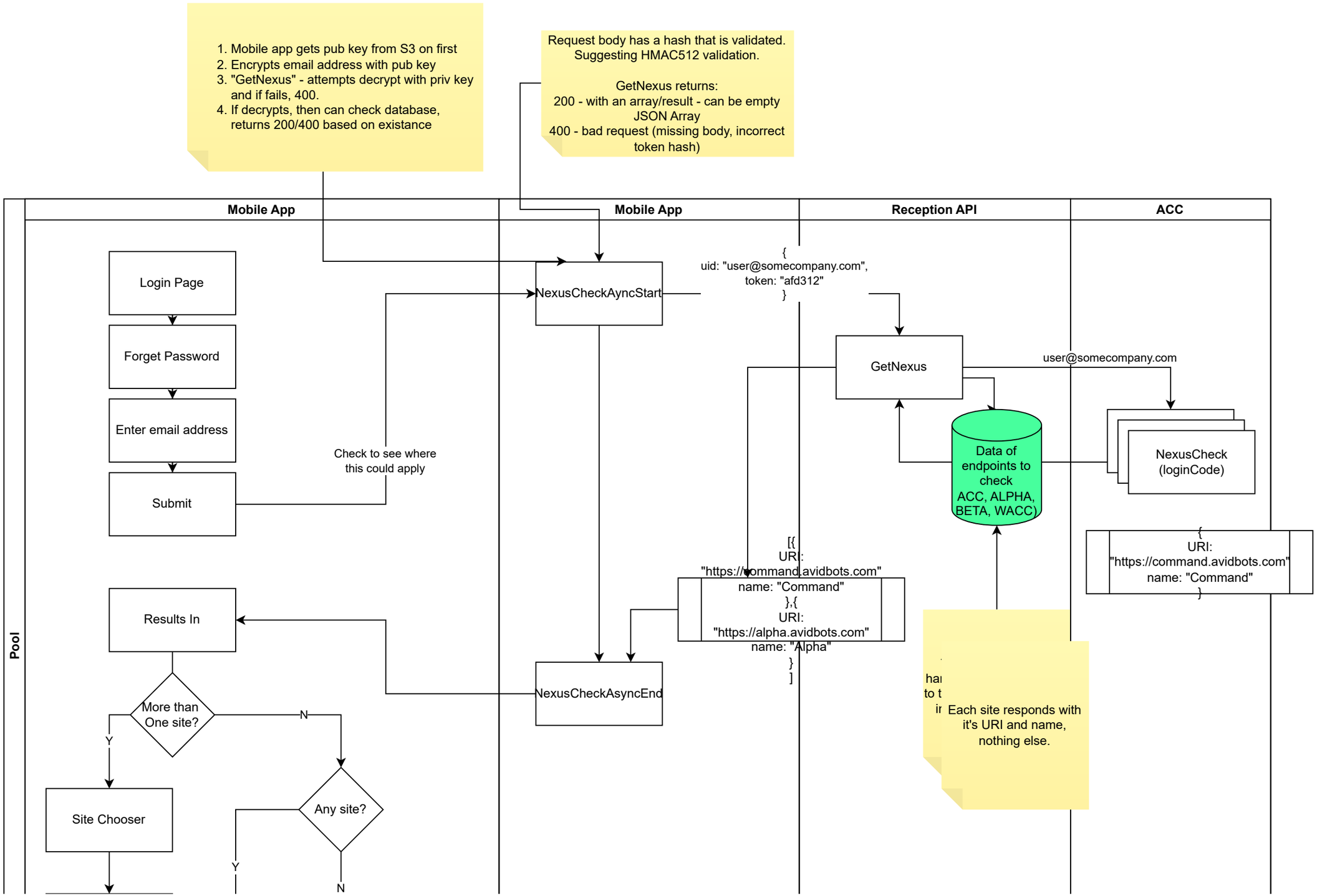

Pool
Mobile User
Mobile App
MW Backend
MW FrontEnd
MW Operator
Opens App
Is Logged In?
Y
Are notifications Allowed?
N
Are notifications setup?
N
Ask for notification permission
Was Successful?
Bind notifications + ACC Account


Pool
Mobile User
Mobile App
MW Backend
MW FrontEnd
MW Operator
Mobile Device Map Present
Show Mobile Connectivity Present
Operator opens chat and sends a message
Chat Window
Session create
Store Message
Notification Distribution
Chat window
Chats and sends message, see Flow - user initiated
DSR: NOTE: THIS FLOW NEEDS TO BE UPDATED TO CAPTURE THE "RELINQUISH CONTROL" and "RESOLVED" / Chat close functionality we discussed Friday
Primary Operator
Secondary Operator
Mobile Device Map Present
Show Mobile Connectivity Present
Second Operator Opens Chat session
Chat history
Chat Window
User sends Chat response
Chat window
Store Message
Chat Window
Live page communication
DSR: Due to the complexity of this, we'll need to make another with just the translation pieces...
(Demonstrating two sessions, possibly with three translations)

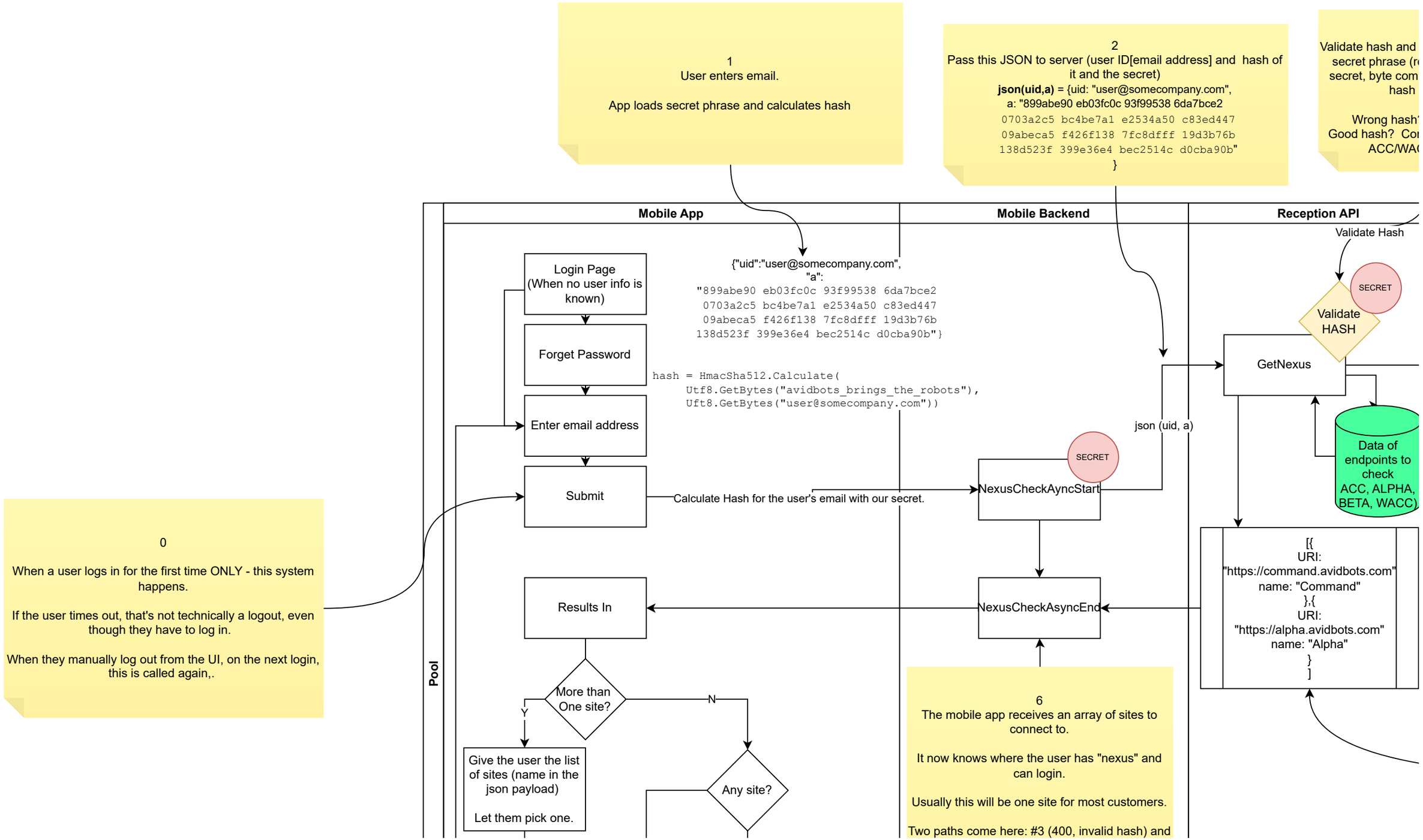

Pool
Mobile User
Mobile App
MW Backend
MW FrontEnd
MW Operator
Opens App
Is Logged In?
Ask for Username, Password
All filled in correctly?
Y
Is Account known/Reception flow
More than one server can handle?
Y
N
Multi Server Choice
Store Nexus / Server
Login, Get token
Store Token


EULA Display Agreement
Privacy Display Agreement
Store EULA/PA versions/dates


Pool
Mobile User
Mobile App
MW Backend
MW FrontEnd
MW Operator
Opens App
Chooses to Logout
Confirmation
Confirms?
Y
Clears Notification Channel
Clears site url, personal data
Return to login page
Pots to clear notification channel map from device->ACC
Clears outbound notifications for this device
Ends any chat sessions that might be open


1. Mobile app gets pub key from S3 on first
2. Encrypts email address with pub key
3. "GetNexus" - attempts decrypt with priv key and if fails, 400.
4. If decrypts, then can check database, returns 200/400 based on existance
Request body has a hash that is validated. Suggesting HMAC512 validation.
GetNexus returns:
200 - with an array/result - can be empty JSON Array
400 - bad request (missing body, incorrect token hash)
Pool
Mobile App
Mobile App
Reception API
ACC
Login Page
Forget Password
Enter email address
Submit
Check to see where this could apply
NexusCheckAyncStart
{
uid: "user@somecompany.com",
token: "afd312"
}
GetNexus
user@somecompany.com
Data of endpoints to check
ACC, ALPHA, BETA, WACC)
NexusCheck
(loginCode)
{
URI:
"https://command.avidbots.com"
name: "Command"
}
[{
URI:
"https://command.avidbots.com"
name: "Command"
},{
URI:
"https://alpha.avidbots.com"
name: "Alpha"
}
]
Each site responds with it's URI and name, nothing else.
Results In
More than One site?
Y
N
Site Chooser
Any site?
Y
N
NexusCheckAsyncEnd


Redirect to ACC for process continues
LIE: Report that if the email was in fact available, an email was sent.
-- DO NOT SAY WE DON"T HAVE THIS EMAIL ON FILE.


1
User enters email.
App loads secret phrase and calculates hash
2
Pass this JSON to server (user ID[email address] and hash of it and the secret)
json(uid,a) = {uid: "user@somecompany.com",
a: "899abe90 eb03fc0c 93f99538 6da7bce2
0703a2c5 bc4be7a1 e2534a50 c83ed447
09abeca5 f426f138 7fc8dfff 19d3b76b
138d523f 399e36e4 bec2514c d0cba90b"
}
Validate hash and
secret phrase (r
secret, byte com
hash
Wrong hash
Good hash? Co
ACC/WA
Mobile App
Mobile Backend
Reception API
Validate Hash
SECRET
Validate HASH
GetNexus
Data of endpoints to check ACC, ALPHA, BETA, WACC)
{"uid":"user@somecompany.com",
"a":
"899abe90 eb03fc0c 93f99538 6da7bce2
0703a2c5 bc4be7a1 e2534a50 c83ed447
09abeca5 f426f138 7fc8dfff 19d3b76b
138d523f 399e36e4 bec2514c d0cba90b"}
hash = HmacSha512.Calculate(
Utf8.GetBytes("avidbots_brings_the_robots"),
Uft8.GetBytes("user@somecompany.com"))
Login Page
(When no user info is known)
Forget Password
Enter email address
Submit
Calculate Hash for the user's email with our secret.
json (uid, a)
SECRET
NexusCheckAyncStart
NexusCheckAsyncEnd
[{
URI:
"https://command.avidbots.com"
name: "Command"
},{
URI:
"https://alpha.avidbots.com"
name: "Alpha"
}
]
Results In
More than One site?
Y
N
Give the user the list of sites (name in the json payload)
Let them pick one.
Any site?
Pool
0
When a user logs in for the first time ONLY - this system happens.
If the user times out, that's not technically a logout, even though they have to log in.
When they manually log out from the UI, on the next login, this is called again,.
6
The mobile app receives an array of sites to connect to.
It now knows where the user has "nexus" and can login.
Usually this will be one site for most customers.
Two paths come here: #3 (400, invalid hash) and

3
email are valid with the
e-hash the email and
parison should match
passed in.

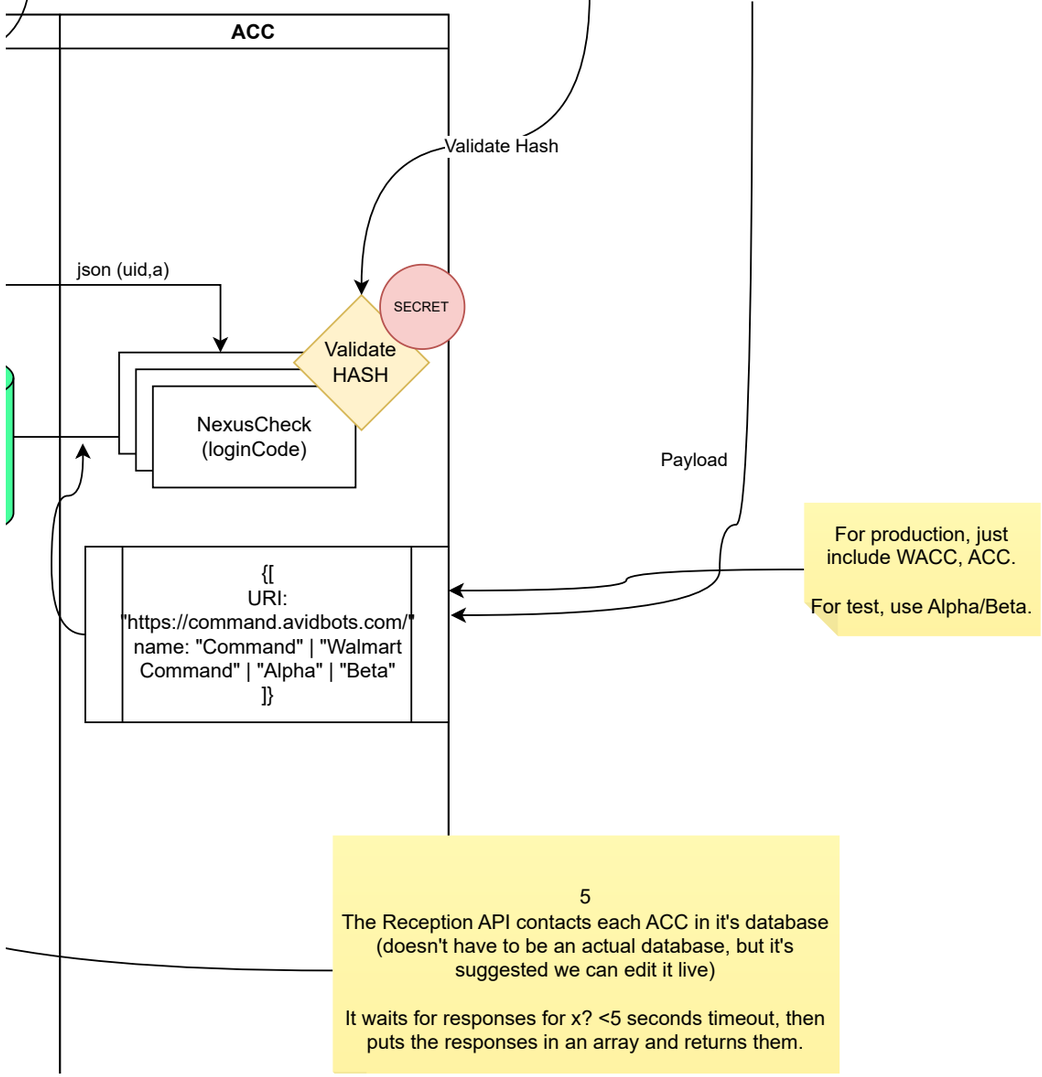
? Http Status 400
ntinue - we can talk to
CC/Alpha/etc..

4
All ACC's have a new function much like reception that validate the json(uid,a) and check the hash against the secret.

Same method as in the reception API.

Wrong hash? Http status 400,
Good hash:
a) check db for email address
If found, return 200 with this payload.

If there is no email, the response is 200 and an empty json array.

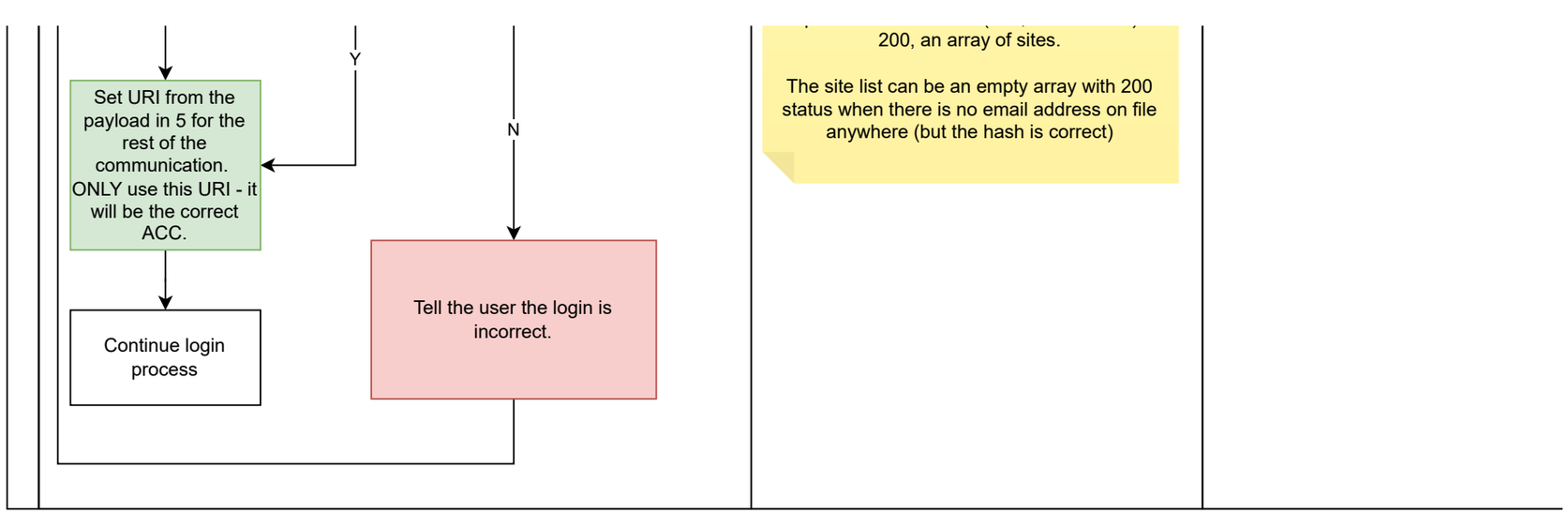


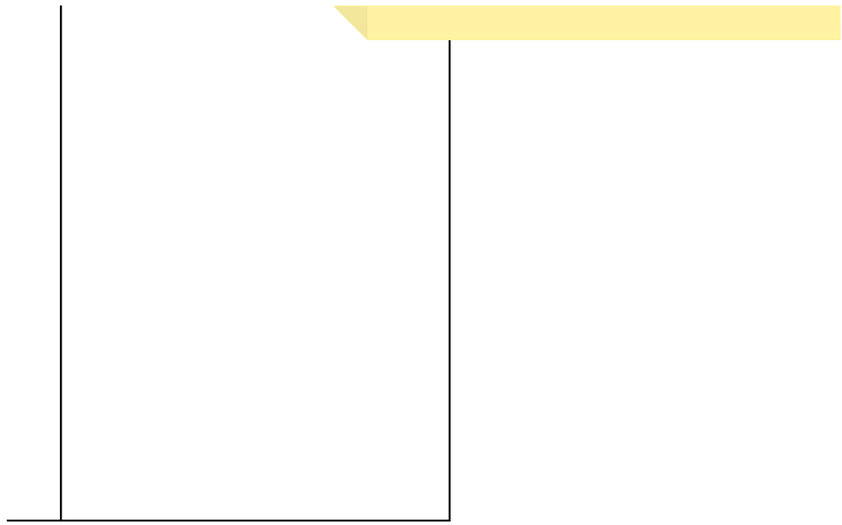
For production, just include WACC, ACC.

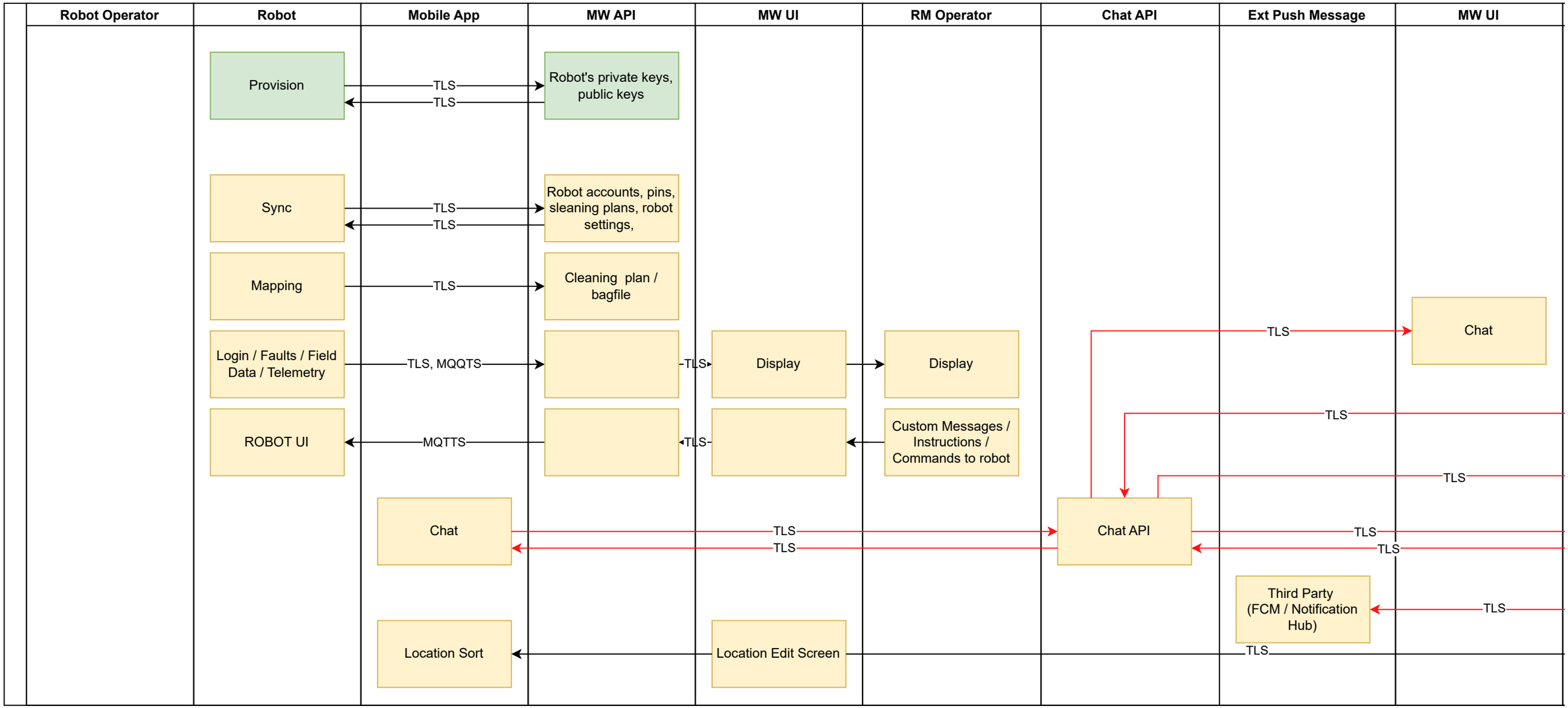
For test, use Alpha/Beta.

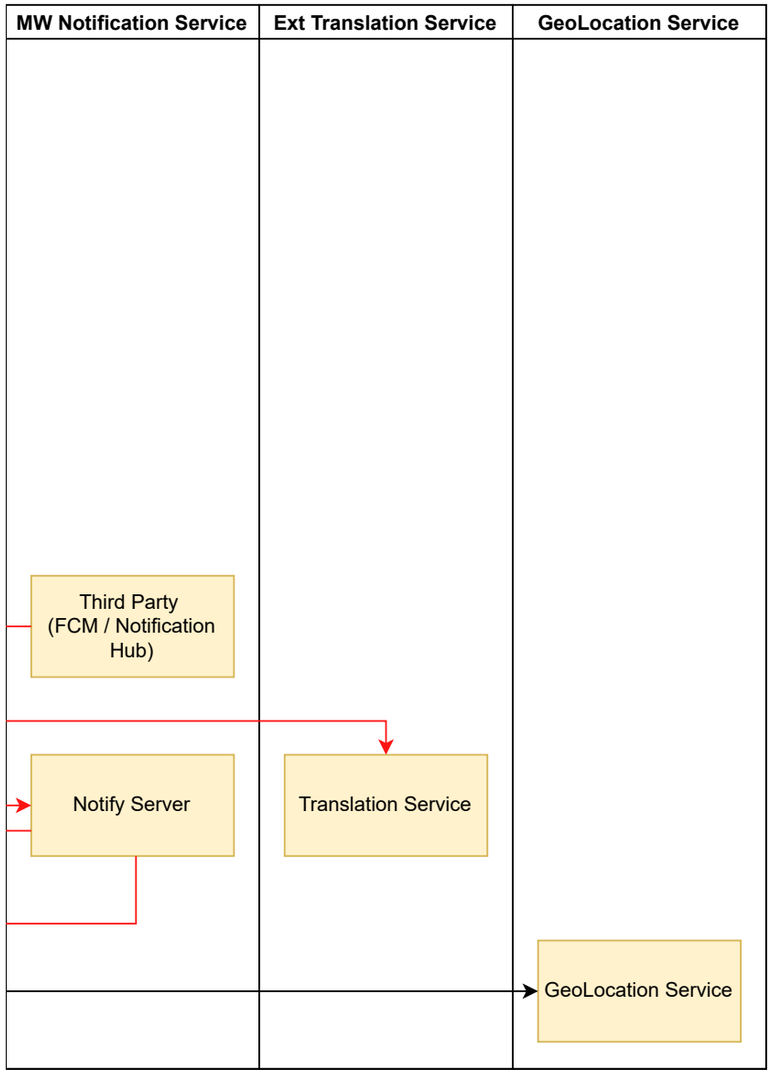
5
The Reception API contacts each ACC in it's database (doesn't have to be an actual database, but it's suggested we can edit it live)

It waits for responses for x? <5 seconds timeout, then puts the responses in an array and returns them.


Y
N
Set URI from the payload in 5 for the rest of the communication. ONLY use this URI - it will be the correct ACC.
Continue login process
Tell the user the login is incorrect.
200, an array of sites.
The site list can be an empty array with 200 status when there is no email address on file anywhere (but the hash is correct)




Robot Operator
Robot
Mobile App
MW API
MW UI
RM Operator
Chat API
Ext Push Message
MW UI
Provision
Robot's private keys, public keys
TLS
TLS
Sync
Robot accounts, pins, sleaning plans, robot settings,
TLS
TLS
Mapping
Cleaning plan / bagfile
TLS
Login / Faults / Field Data / Telemetry
TLS, MQQTS
TLS
Display
Display
ROBOT UI
MQTTS
TLS
Custom Messages / Instructions / Commands to robot
Chat
TLS
TLS
Chat API
Chat
TLS
TLS
TLS
TLS
TLS
TLS
Third Party (FCM / Notification Hub)
TLS
Location Sort
Location Edit Screen
TLS

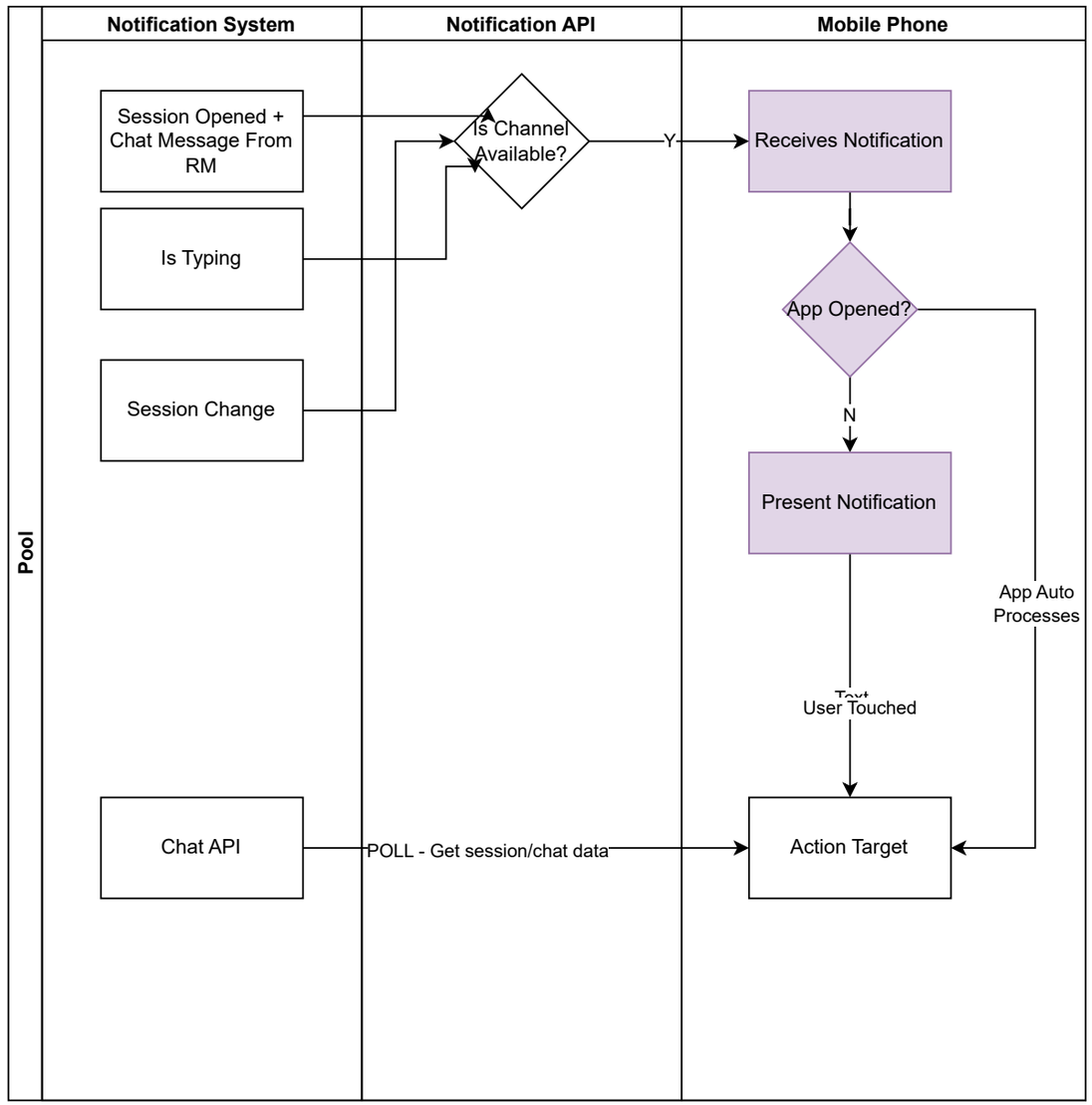


FACTORY

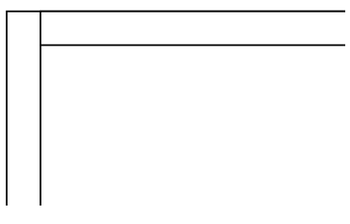
DAY TO DAY

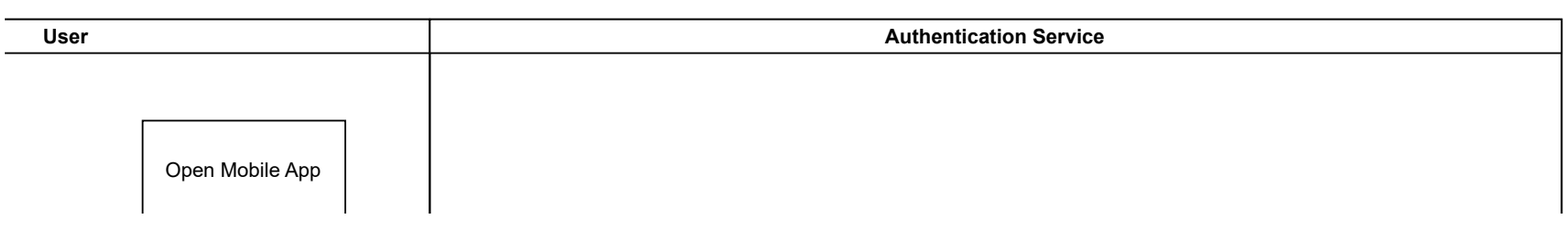
PII

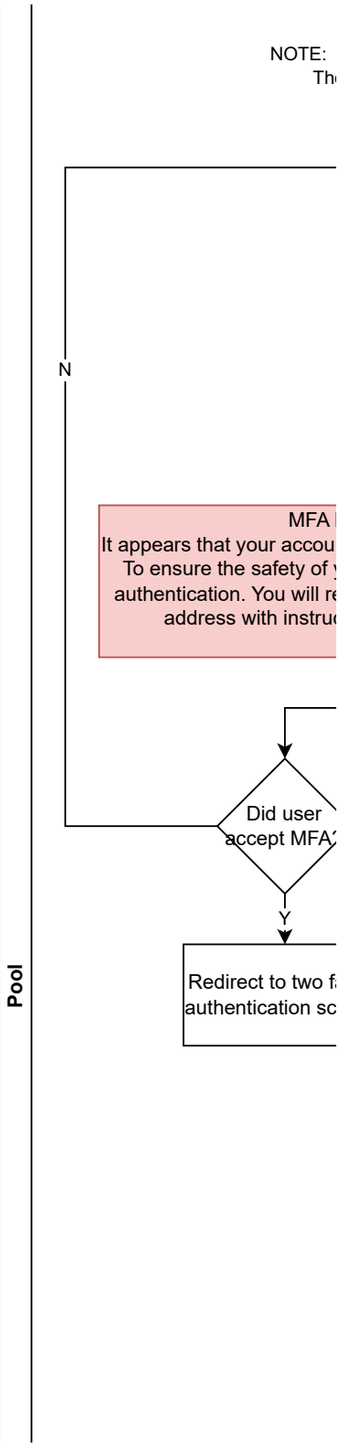
Non-PII

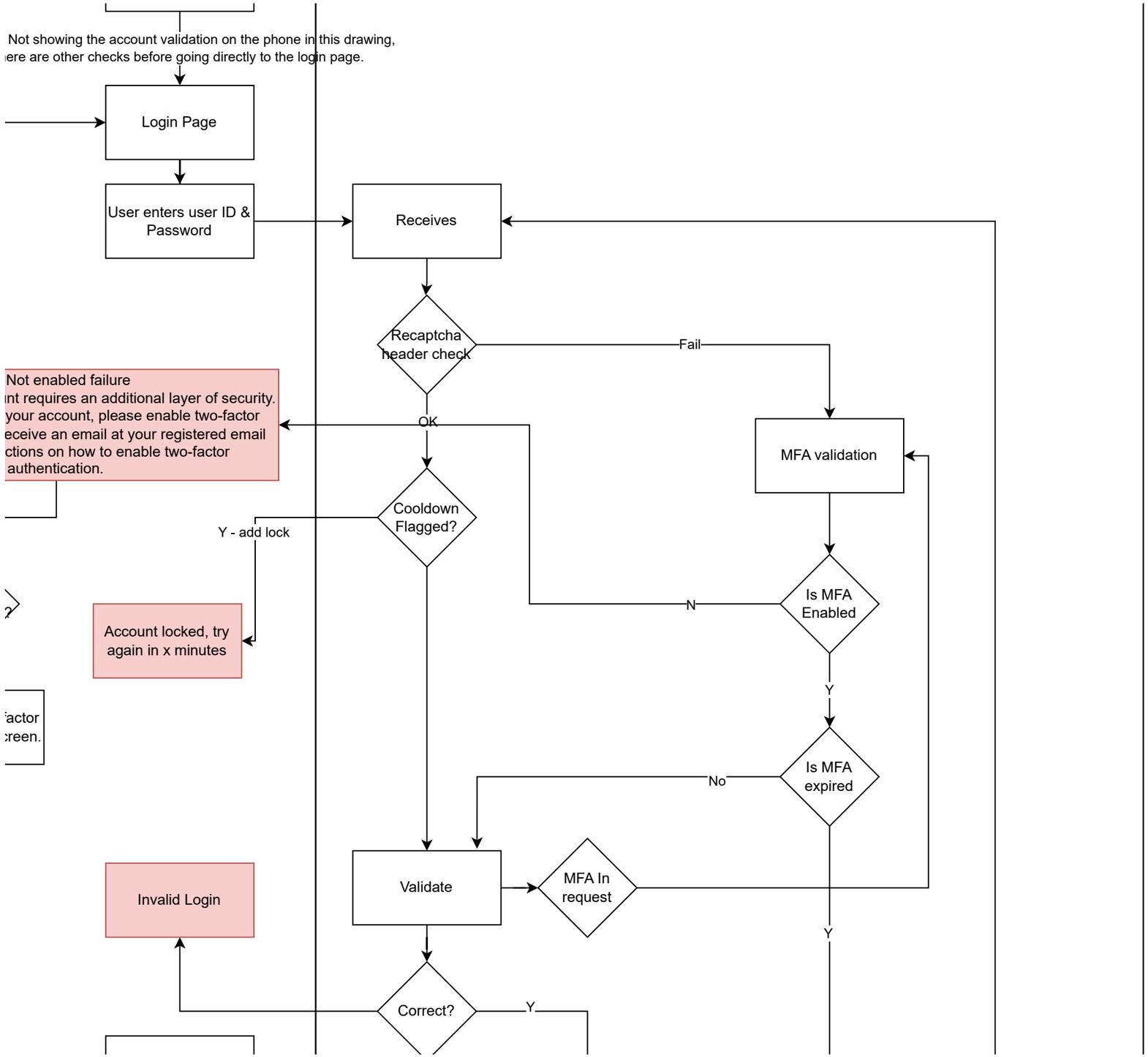


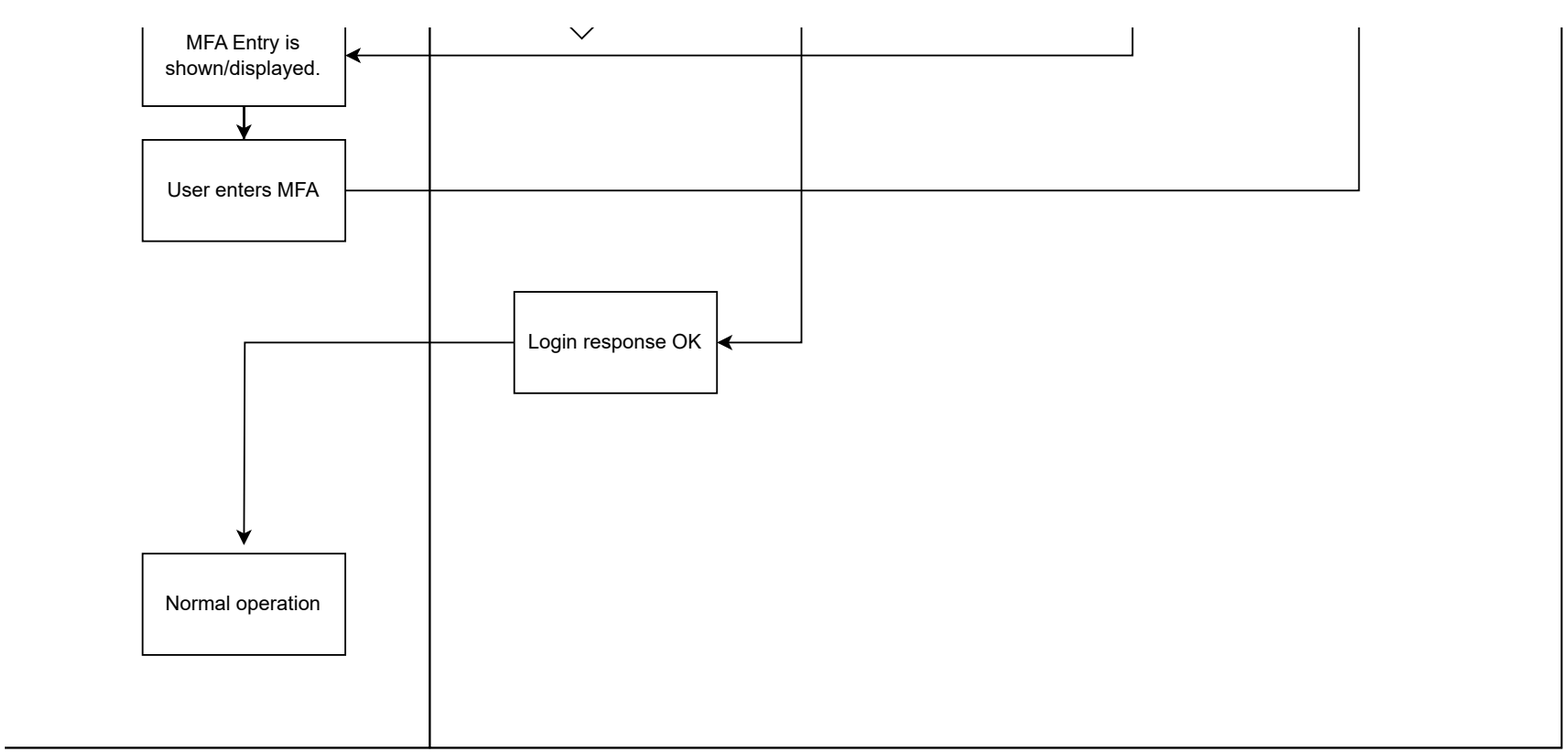
Native Phone Responsibility

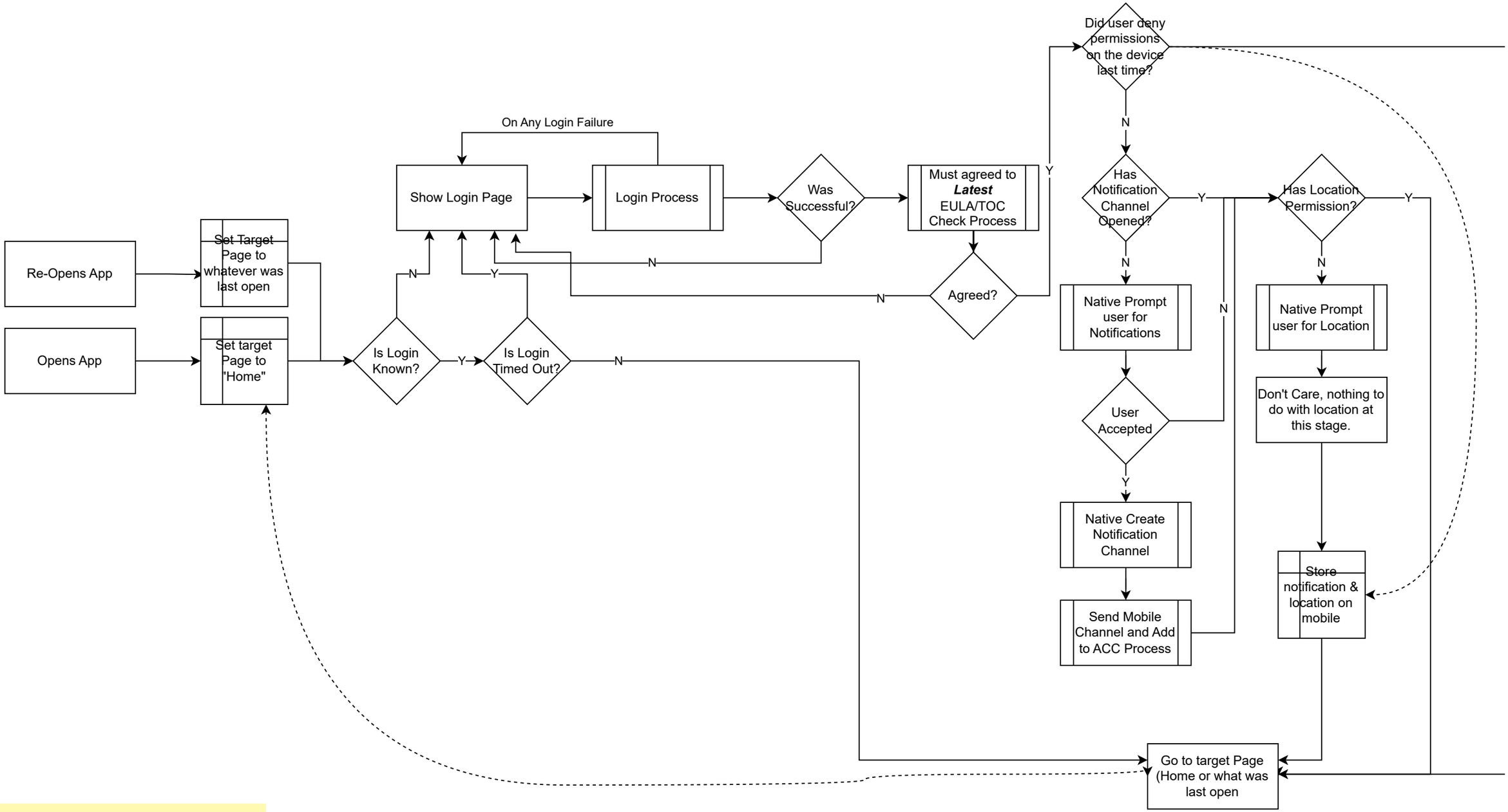



User
Authentication Service
Open Mobile App


Pool
NOTE:
N
MFA
It appears that your accou
To ensure the safety of
authentication. You will re
address with instruc
Did user
accept MFA
Y
Redirect to two f
authentication sc

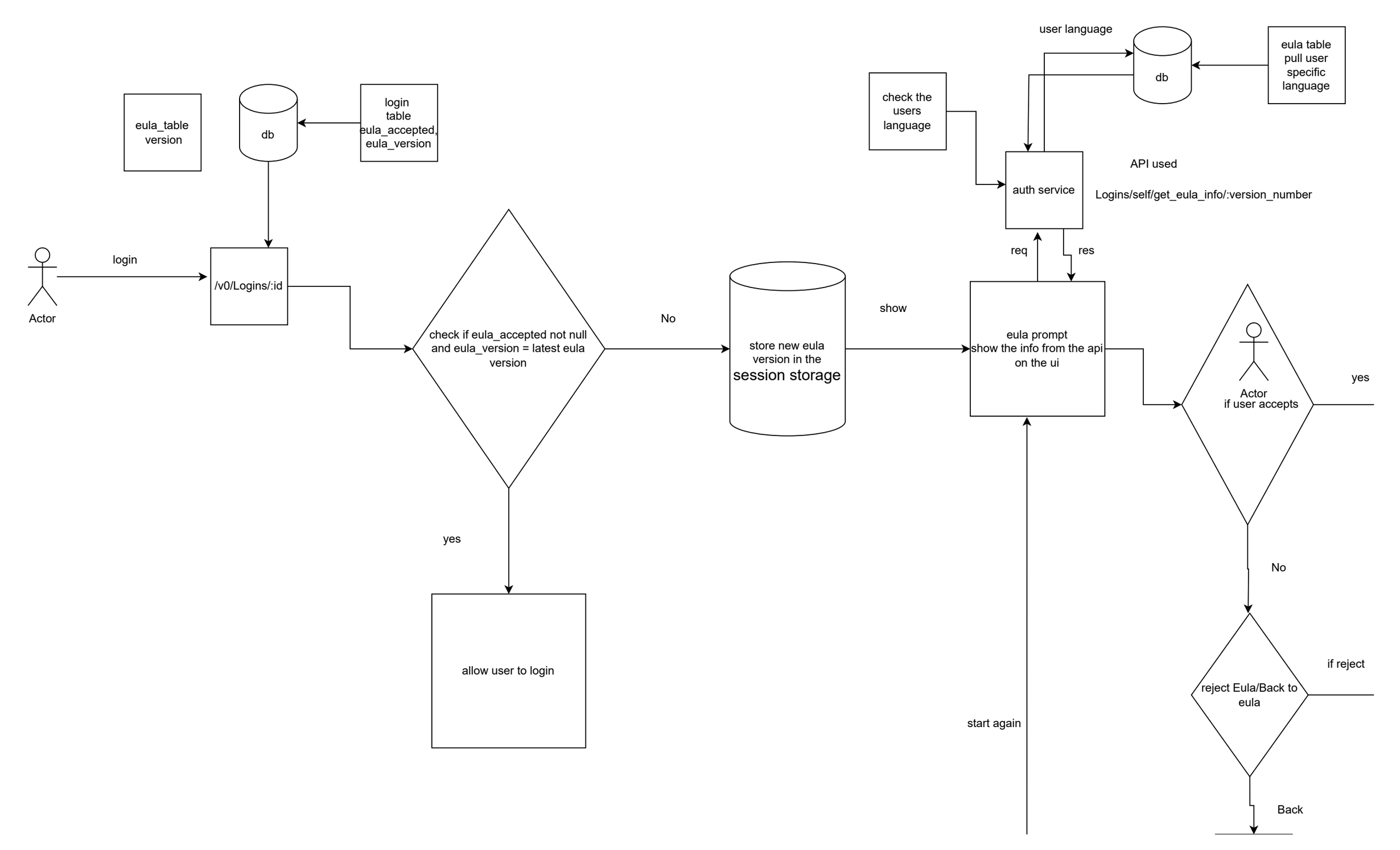

Not showing the account validation on the phone in this drawing,
ere are other checks before going directly to the login page.
Login Page
User enters user ID & Password
Receives
Recaptcha header check
Fail
OK
Not enabled failure
nt requires an additional layer of security.
your account, please enable two-factor
eceive an email at your registered email
ctions on how to enable two-factor
authentication.
MFA validation
Cooldown Flagged?
Y - add lock
Account locked, try again in x minutes
Is MFA Enabled
N
Y
Is MFA expired
No
Validate
MFA In request
Invalid Login
Y
Correct?
Y

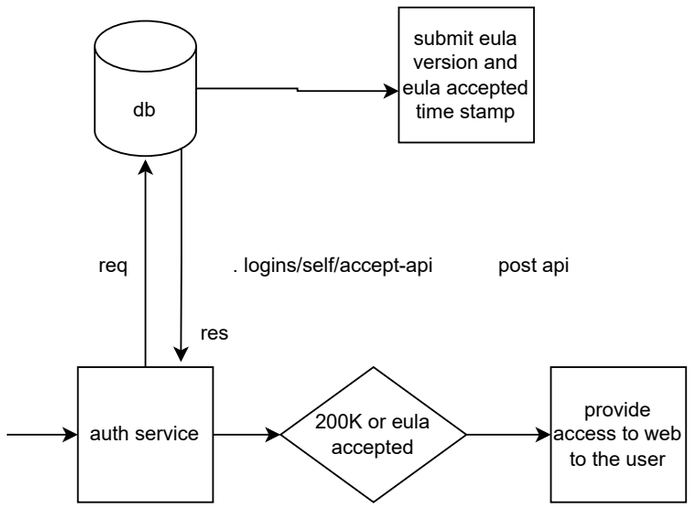

MFA Entry is shown/displayed.
User enters MFA
Login response OK
Normal operation

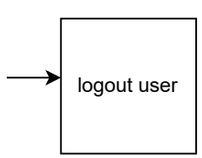


Sid brings up a great comment - in the POC we did permissions first, then login, EULA.

Technically, doesn't matter - so long as pairing with / on ACC is done after EULA.

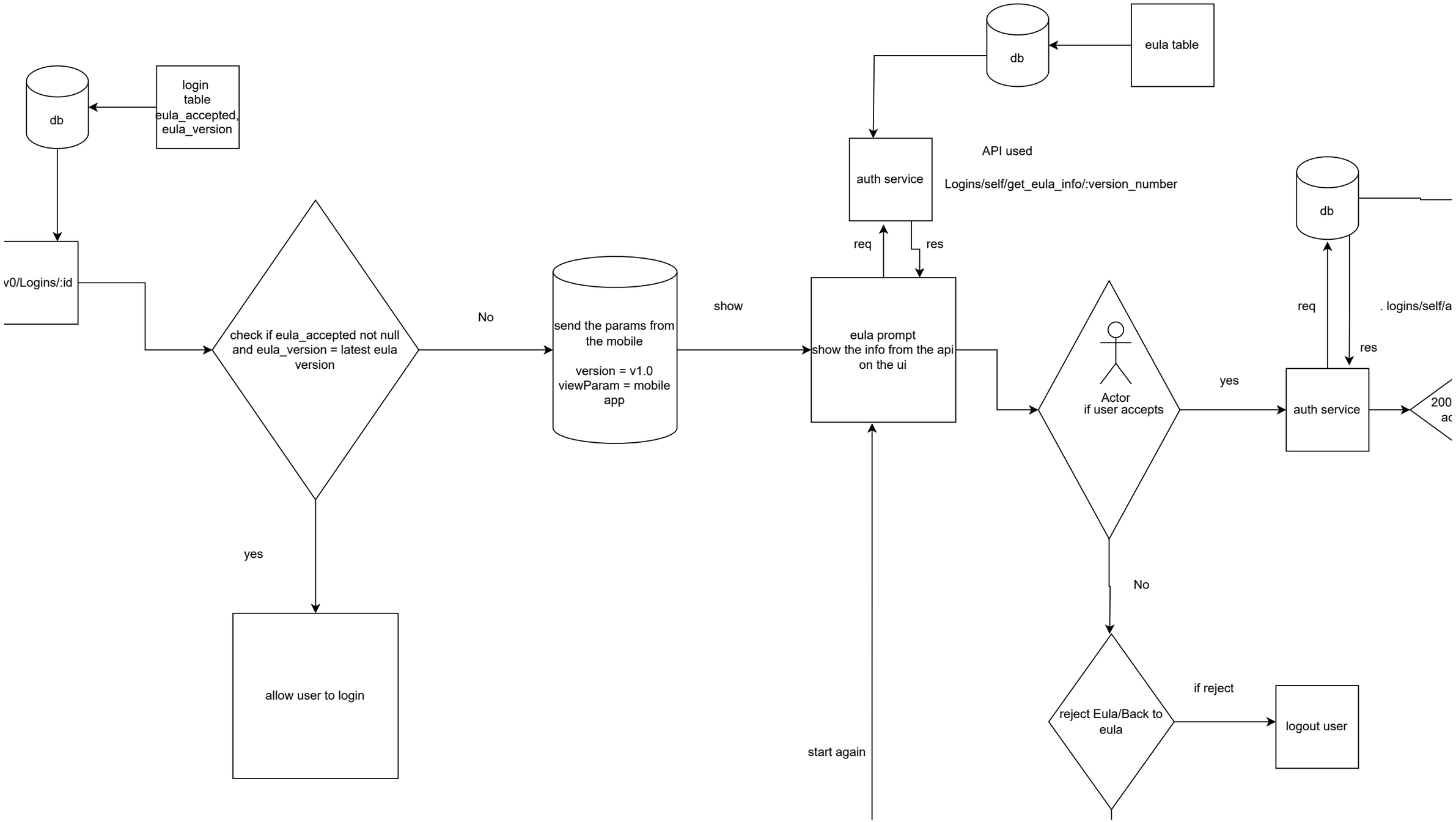

user language
db
eula table
pull user
specific
language
check the
users
language
eula_table
version
db
login
table
eula_accepted,
eula_version
auth service
API used
Logins/self/get_eula_info/:version_number
req
res
login
Actor
/v0/Logins/:id
check if eula_accepted not null
and eula_version = latest eula
version
No
store new eula
version in the
session storage
show
eula prompt
show the info from the api
on the ui
Actor
if user accepts
yes
yes
allow user to login
No
if reject
reject Eula/Back to
eula
start again
Back

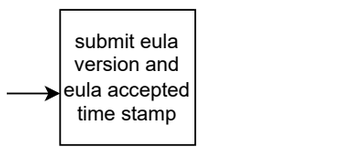

db
submit eula version and eula accepted time stamp
req
. logins/self/accept-api
post api
res
auth service
200K or eula accepted
provide access to web to the user


logout user

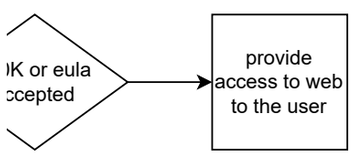

start the eula
accept steps

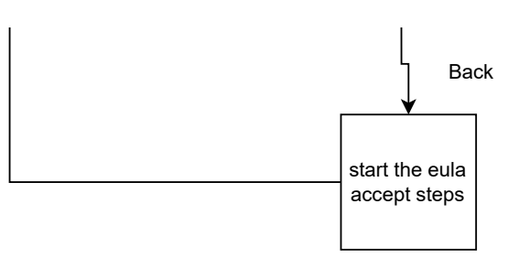

eula_table
version
login
Actor


db
login
table
eula_accepted,
eula_version
v0/Logins/:id
check if eula_accepted not null and eula_version = latest eula version
yes
allow user to login
No
send the params from the mobile
version = v1.0
viewParam = mobile app
show
db
eula table
auth service
API used
Logins/self/get_eula_info/:version_number
req
res
eula prompt
show the info from the api on the ui
start again
Actor
if user accepts
yes
No
reject Eula/Back to eula
if reject
logout user
db
req
res
. logins/self/a
auth service
200
ac



ıccept-api post api




Back
start the eula
accept steps

TODO:

Mobile app process & build/pipelines

Asbuilt