





6/01/2021

IOT CIE-3

Name: Sai SrframiV USN: JBMIFCSILD CC: 2015 SPETOT

1) WAMP (Web Application Messaging Protocol) is a sub-protocol

of Websocket which provides publish-subscribe and

remote procedure call (RPC) messaging fatterns.

WAMP enables distributed application architectures

where the application components are distributed on

multiple nodes and communicate with messaging

patterns provided by wamp.

Key to Feature of wamp:

-> Transport: - For wamp, defaut is webrocket.

-> Lession &: - Session is a conversation blu two peers.

> Client: - Clients are peers that can have one or more redes. On publish-lubricite model client can have bollowing pudles roles:
-> Publisher: - Publisher events to topics mountained by the Broker.

-> Subscribes: - He subscribes to the topics and receives the events including the pay load.

In RPC model, Client has two edes: - Delatta:

(i) Caller: - Issues calls to the remote procedures along with call arguments.

ii) Callel: - Executes procedures to which the call as knowed by the caller & returns the results back to the caller.

Router: - Pers that perform generic Call & event houting.

In Publish subcribe model Router has the note of a Broker. and in RPC model, Router has the role role of a Broker.

J. Application Code: - 9+ runs on the Citents (Publishes, Subcriber, Callee of Caller).

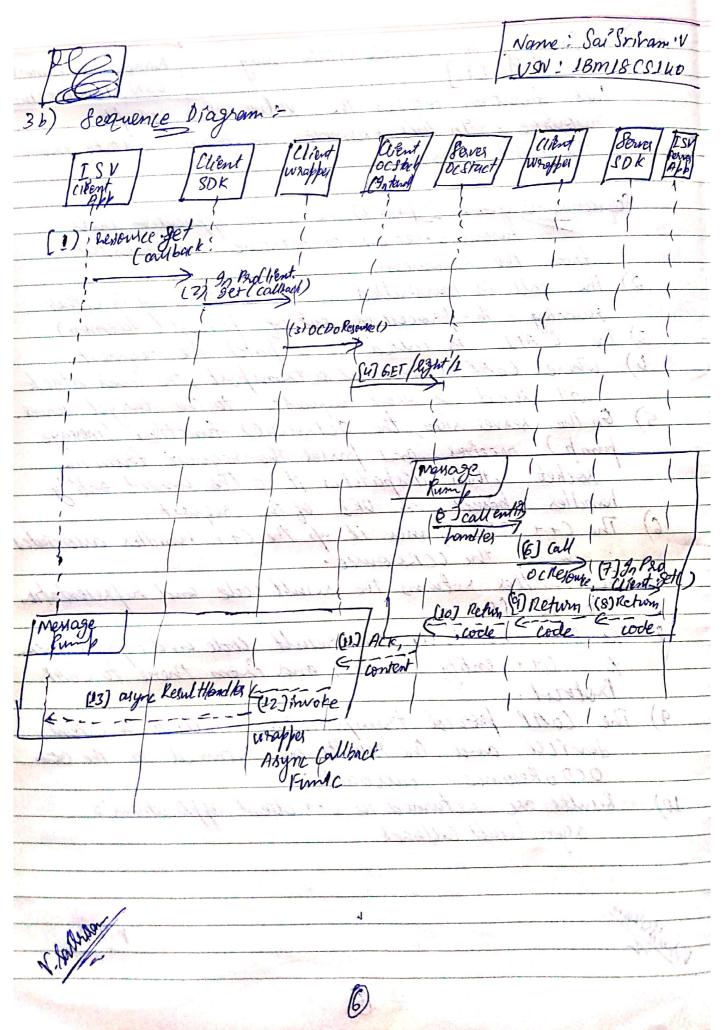
La Ca 2 Comment
Titos Mangiges eseates Groups, Nave, Justinos
20) 26) Things Maneges created Groups, Name: Sal Srivam V Ah Is appropriate member plange USN: 18M18CS140 In the retwork, manages member presence
And appropriate member presence
In the retwork arther easy gt benefits 3nd party
and markes group way:
And appropriate member presence the retwork manages member presence and makes group action easy. It benefits 3nd farty application developers the three way:
s) Application can easily collect things the a specific service by the service characteristics, not by each service by the service characteristics, not by each thing's identification.
1) Apparation con the characteristics, not by each
thing's identification handling tracky of
2) Application does not require handling tracing of monitoring many things.
monitoring many things.
a) all les to does not requise monaging to send
3) Application cois to several things. Also,
monitoring many things. 3) Application does not require managing to send control nexages to several things. Also, configurations and diagnostics of multiple things can be supposted by this service.
to be supported by this service.
Can be supported
his I had som RDS & a web-service that allows you to
a o a stand of the sound of the contracte of the contracte
(1) XOSUPO TO THE PARTY WITH NO , GOVERNO CONTRACTOR
easily set up, operate and scare a recultorial
the cloud. LDS can server as a scalable carastole for IDI
Rustems. With KDS, 101 system developers curion
any amount of data in scalable relational
databases. It was the labour 1974 of
(1) Calles : " 12 y lalls to the bound from order almounts
Call arguments
21) Called or Executed presidences to make the order of the
called a transfirm it will hear to the william
- Kontes T 1812 Int prison Sensis with sent nearthy
The Middle on The Mark Sing merches have the hall the
the Brakes and it here wider it will be the
- 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1
a ban

an sos queue Name: Sai Srivam.V USN: 1BM28CS14D impost boto sgs from boto-sgs-message impost Message ACCESS KEX = " < enter access key SECRET KEX = " < enter secret key >" pent " Connecting to SQS" boto sgs · connect_to_region queue name = 'my testqueue! "Connecting to queue: "+ queue name com. get_all_queues (presix = queue the quece: " + str (count) neuge from quene for P in range (count): m= 9 [0]. read() print "mengge "/d "/.5" (i+1, 8+8(m.9) 9[0]·de lete_message (m) part "Read % of messages from queue" 1. (count) writing to an SQS queue? from boto . 898 - message import Topas time SECRET KEY = " us - east -1 P.7-0.

Scanned with CamScanner

brint "Connec they to SQS"	
conn = boto . sqs · connect_ to_region (
REGION,	
aws _ access_key_id = ACCESS_KEY, aws _ secret _ access_key = SEGRET_KE	y)
queue name = 'mytest queue'	
part "Connecting to queue: " + queue name	
print 'Connecting to queue: " + queue name q= conn.get_all_queues (prefix = queue name)	
meg date time = time. asctime Ctime. localtime (time time	·()()
msg = "Test message generated on: "+ meg date the print = "Writing to queue: "+ msg m- Message ()	
point & "Writing to queue: "+meg	16
m- Message 13	
me see bouy (my)	
status = q[o] : wente(m)	- 20
point "message written to que is	151
count = 9 So J. count ()	
pount 'Total messages in queue: "+8tx (count)	100 A
i thought from the	130
m= 2/0).icado	
Them might be still growth from the state of	14
Je Miron	
A 10 T. gr left marcade in	- 1
	-
Street for hour want proposed for front their	
	100
- if their proposes to watery to car so grains	
to the second of	* 110
The wit student from Significant . They were the first the the	
man I was a second of the seco	1 10
Clarker	
	7/2
The state of the s	

-	36)
	This operation forth of which the state with the second
-	167 ET] USN' IBM ISCSIUD
-	This operation fetches the value of a shiple resource. In this example, we fetch the state from the light resource.
1	resource. In this example in both the state form
7	The light resource
-	The state of the s
	Sequence Diagramo steps: 1) The client application calls resource get (2) to retrieve a representation
	resousce cot () to a to a
_	2) The call of resources.
	2) The call is marshalled to the stack which is either
elj-	fruming the backer of a fire stack while is cities
	1) Where COAP is used as a territory the request
	5) On the server side the Ochreway functions (mouses stack
	bump) receives and basen the action (menage
	handles based on 1105 and the world entity
	handles based on URT of the request.
9_	6) The C++ SDK passes it up the C++ handles associated
	F) The handles returns the squitt code
	The handles returns the result code and representation
	1) The SDK marshalls the result code and representation to C++ entity hamdler and from there to COAP 1) The COAP protocol transports the soull is
-	Posters 1
, i	9) The COAP protocol transports the
-	devices and the results are get to the client
	9) The COAP protocol transports the results to the client devices and the results are returned to the client OCD o Resource callback.
	10) Runte are returned to C++ client applications's
-	asynckerult Callback. Cyen copplications's
-	
-	
(has	a som
1000	Na Maria
1	Vhailing P.7.0
	(5)
The same	



2 a) On Analyzing given diagram, Behavious on Resource Model:	Name: Sai SrivamiV
Bolowing given diagram,	USN: IBMISCSILLO
on Resource Model:	USN IBM BESTA
- Finding a rejource	2 440
- Queryty, setting and obse	suly folousce state.
Here, one of the died resource	es on the garage
door opener is the light cont	ed: it has a GET
Tinding a resource Tinding a resource Ducouping, setting and observed, one of the child resource door opener is the light controperation that allows a device light state (on/off).	to get the current
light state (an (of)	
g	2
	The second second
V Jai Shi san	
V Jail	
6	
(/)	