III Workshop 2024 - EMBM-VD

CONTEXT:

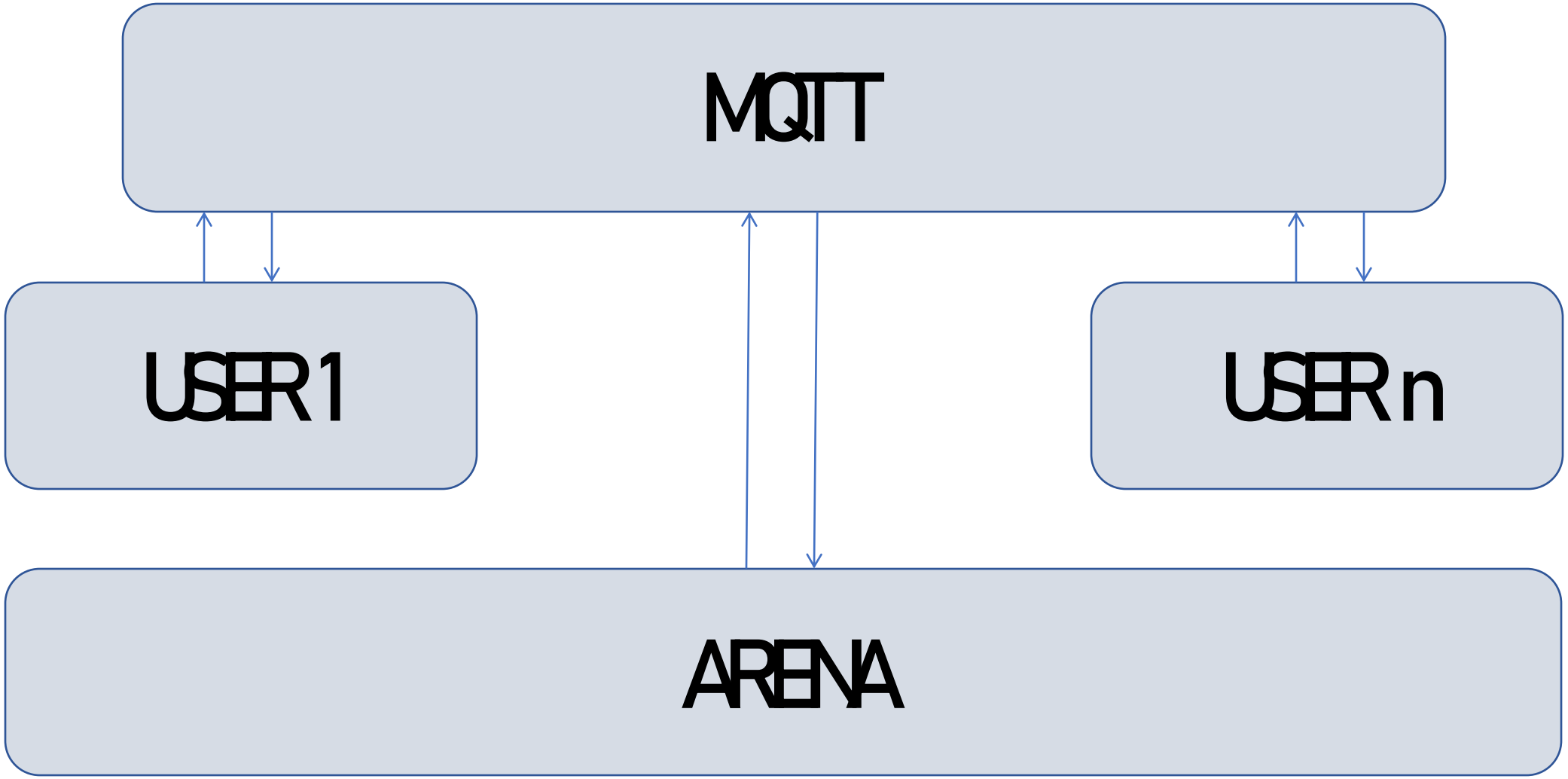
- What is ARENA
- Connection
- Examples

ARENA

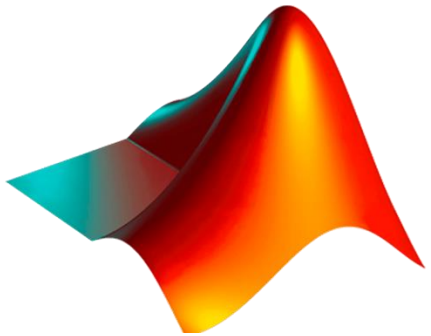
What is Conceptio ARENA?

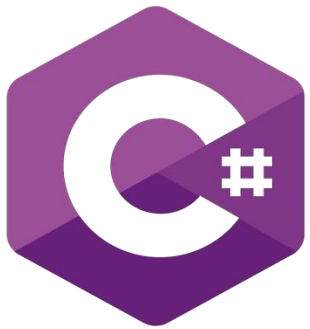
Fonte: ARMA III. Acessed in 20/08, https://kriegsimulation.blogspot.com/2013/05/arma-3-alpha-vidales-agia-marina-and.html

And how it works?



How can I use ARENA?





Connection example

- Paho-mqtt 2.1.0

```python
# Configuração MQTT e Objeto
entity_name = "Golf_3" #Nomenclatura do objeto
domain = "air" #Definição do domínio
owner = "python" #Definição do usuário
visual_definition = "1:1:2:29:2:1:1:2" #Definição estética\física do objeto

broker_address = "161.24.23.15" #Porta MQTT
broker_port = 1883  #Conexões MQTT sem criptografia.

# Verificação da Conecção
def on_connect(client, userdata, flags, rc):
    print(f"Conectado: Código {rc}")
```

Export example

```python
# Criação do primeiro Heartbeat
def publish_heartbeat(mqtt_client, entity_name, uuid_str, visual_definition, domain, owner):
    message = {
        "uuid": uuid_str,
        "name": entity_name,
        "type": visual_definition,
        "visual": visual_definition,
        "owner": owner,
        "domain": domain
    }
    json_message = json.dumps(message)
    topic_name = f"conceptio/unit/{domain}/virtual/{entity_name}/heartbeat"
    mqtt_client.publish(topic_name, json_message, qos=2)
```

Export example

- Paho-mqtt 2.1.0

```python
def publish_kinematics(mqtt_client, uuid_str, domain, entity_name, kinematic):
    message = {
        "uuid": uuid_str,
        "lat": kinematic['lat'],
        "lon": kinematic['lon'],
        "alt": kinematic['alt'],
        "yaw": kinematic['yaw'],
        "pitch": kinematic['pitch'],
        "roll": kinematic['roll']
    }
    json_message = json.dumps(message)
    topic_name = f"conceptio/unit/{domain}/virtual/{entity_name}/data/kinematics"
    mqtt_client.publish(topic_name, json_message, qos=0)

publish_heartbeat(mqtt_client, entity_name, uuid_str, visual_definition, domain, owner)
time.sleep(5)
```

```python
# Looping de publicação e atualização do objeto
for kinematic_row in range(len(df)):
    data = {
        "lat": diff_lat + df.loc[kinematic_row, "LAT"],
        "lon": diff_long + df.loc[kinematic_row, "LONG"],
        "alt": diff_alt + df.loc[kinematic_row, "ALT"],
        "pitch": df.loc[kinematic_row, "Pitch"],
        "roll": df.loc[kinematic_row, "Roll"],
        "yaw": df.loc[kinematic_row, "True_Head"]
    }
    publish_kinematics(mqtt_client, uuid_str, domain, entity_name, data)
    time.sleep(0.01) #Intervalo de tempo (Velocidade^-1)
```

```python
def on_message(client, userdata, msg):
    global entities

    if "other" in msg.topic:
        return

    try:
        data = json.loads(msg.payload.decode())
        uuid = data.get("uuid")
        lat = data.get("lat")
        lon = data.get("lon")
        name = msg.topic.split('/')[-3]

        if not all([uuid, lat, lon, name]):
            logger.error(f"Incomplete data received: UUID={uuid}, lat={lat}, lon={lon}, name={name}")
            return
```

Final considerations

Questions/Comments?!

Centenaro@ita.br

joao.centenaro.101888@ga.ita.br