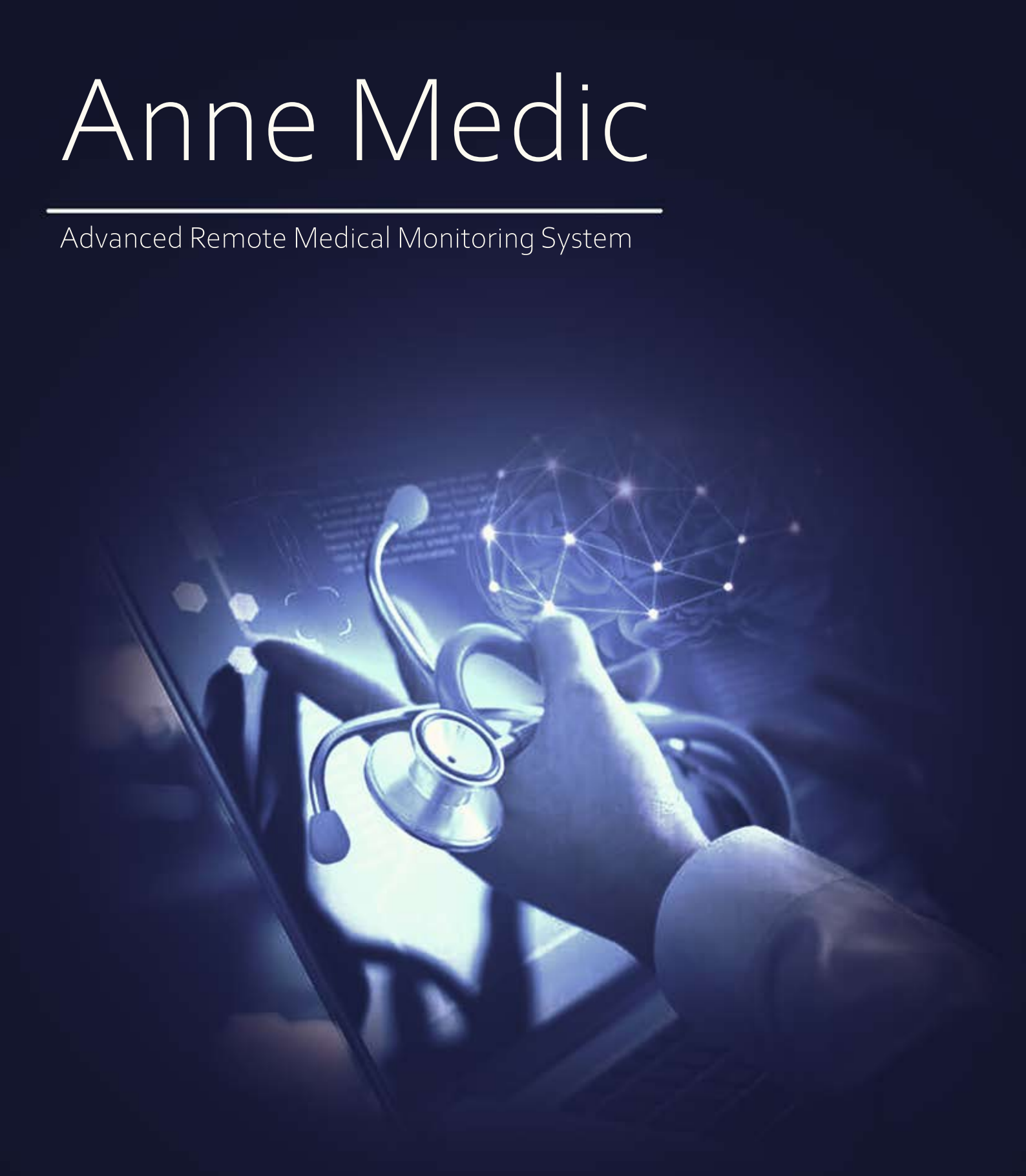
Anne Medic

Advanced Remote Medical Monitoring System

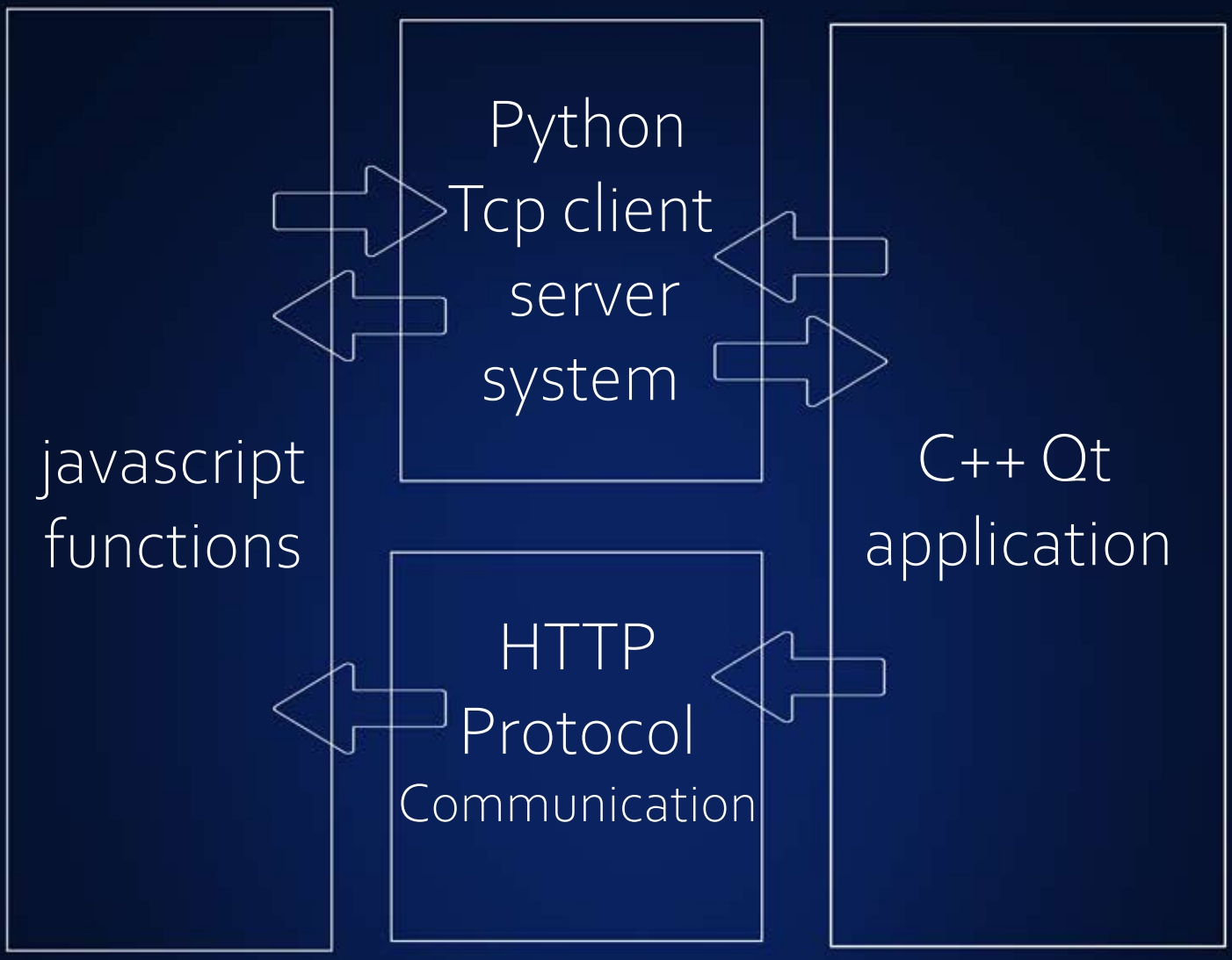


Real Time Rest API

Web Server

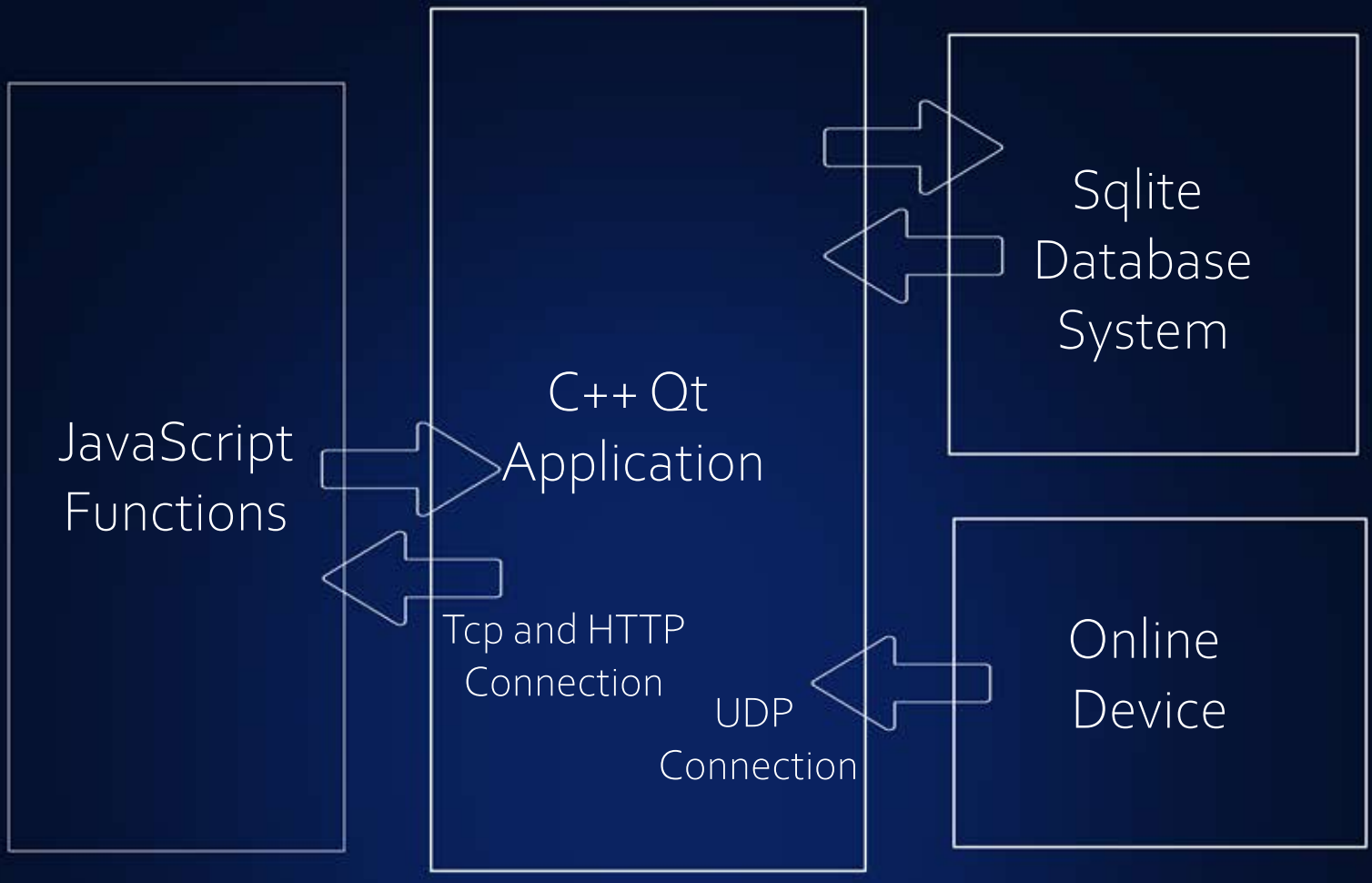
15-dec-2022

System Architecture:



This is how our api working ,the qt app has an http server at port 8080 and indirect tcp server and client also at the same time , the server at port 8081 and the client at port 8082 . so the javascript functions can be used to access the database which can be managed throught the qt app , also this app is a udp server ,

continue system Architecture



The Qt app manages the sqlite database system and also recieve the data throught a sophihestica ted planned protocol to display the graphs and also check the alarms from the device , and to achieve the real time concepts we use the udp protocol which is the fastest serial protocol in computer networks ,then the qt app will store every data and spreed it over tcp or http protocol

How to Use ::

Our Api Has the Following javascript functions:-

1) `function init()`

in this function we initialize the api and should be called only once .

2) `function get_db_conn_status(table_name)`

This will get the connection status of the database use it like this:-

```
init();

/*************************************(tested)
=> test for get_db_conn_status(...)
*************************************/

get_db_conn_status("users");
setTimeout(() => {
console.log("x after is: "+dataBaseConnection)
if(dataBaseConnection)
{
    console.log("connected to database successfully")
}
else
{
    console.log("failed to connect to database")

}
},max_time_out)
```

note: that the init() should be used once

How To Use :: continue

4) `function get_table_lenght(table_name)`

this will get the Number of the table rows
use it like this::

```
/************************************ (tested)
=> test for get_table_lenght(...)
************************************/

get_table_lenght("users");
setTimeout(() => {
console.log("table lenght is : "+table_lenght);

},max_time_out)
```

5) `function get_index_by_value(TableName,Col_name,Col_value)`

will return the index of specified row based on the table name and one colmun name and one colmun value.
use it like this::

```
/********************************** (tested)
=> test for get_index_by_value(...)
**********************************/

get_index_by_value("users","name","ahmed");
setTimeout(() => {
console.log("index of value is : "+index_by_value);
},max_time_out)
```

take care :: you need to check if the row is available or not before using this function or you will get inacurate values.

How To Use :: continue

6) `function read_by_index(TableName,Col_name,index)`

return item in specified TableName and in the colmun name Col_name and at index (the return value is a string) .

```
/************************************ (tested)
=> test for read_by_index(...)
************************************/

read_by_index("users","name",5);
setTimeout(() => {
console.log("content at index : "+read_by_idx);
},max_time_out)
```

if the return value is integer you need to parse it to int

```
read_by_index("users","name",5);
setTimeout(() => {
console.log("content at index : "+parseInt(read_by_idx));
},max_time_out)
```

7) `function read_by_colmun(TableName,Col_name)`

will read the whole colmun valuse available in TableName and the return is an array of string values

```
/************************************ (tested)
=> test for read_by_colmun(...)
************************************/

read_by_colmun("users","name");
setTimeout(() => {
console.log("table colomun is : [ "+read_by_clmn+" ]");
},max_time_out)
```

How To Use :: continue

8) `function read_by_value(TableName,Col_name,Col_value)`

this is a very important function that may be used in Authorization and check if any value in specified colmun at TableName available or not and will return boolean

use it like this::

```
/************************************ (tested)
=> test for read_by_value(...)
************************************/

read_by_value("users","name","ahmed");
setTimeout(() => {
console.log("read by value is : "+read_by_val);
},max_time_out)
```

note that this function return boolean

9)

```
function add_row(TableName,Col_names,Col_values);
function update_row(TableName,Col_names,Col_values);
function delete_row(TableName,Col_names,Col_values);
```

these functions will be used to add or remove or delete rows but be careful that all Col_values should be string

use it like this::

```
let tname="users";
let cnames= ["id","name","password","age","position","hospital"];
let cvalues= ["13","fadl","abcd","35","doctor","elmansoura emergency hospital"];

// test for add_row(...) (tested)
add_row(tname,cnames,cvalues);
// test for update_row(...) (tested)
update_row(tname,cnames,cvalues);
// test for delete_row(...) (tested)
delete_row(tname,cnames,cvalues);
```

database Methodology::

colmun name 0	colmun name 1	colmun name 2	colmun name .....
row index 0	colmun value	colmun value	. . . . .
row index 1	colmun value	colmun value	
row index 2	colmun value	colmun value	
row index 3	colmun value	colmun value	
			

THANK YOU