

Controler Application Communication (CAC) Framework Reference Document

Javascript API

C# .NET API

v1.0

Contents

1	WebSocket Communication Protocol.....	3
2	Structures and enumerations	3
3	CAC framework commands (WebSocket).....	4
3.1	Register a client to the session	4
3.2	Subscribe / Unsubscribe to a device	4
3.3	Register to receive compressed or raw data (obsolete).....	5
3.4	Publish / stream information as a device	5
3.5	Data Packet Reception	5
4	JSON structures.....	5
4.1	Skeleton structure.....	5
4.1.1	Containing one skeleton	6
4.1.2	Containing two skeletons.....	7
4.2	RGB Image structure	9
4.3	Android sensors structure.....	10
4.4	Wii BalanceBoard and Wii Remote control structure.....	10
4.5	Mindwave structure.....	11
5	Javascript API	11
5.1	Include the Javascript source files	11
5.2	Define SessionID and Server address.....	11
5.3	Add a handler to process incoming stream frames	12
5.4	Subscribing/Unsubscribing to specific streaming sources.....	13
5.5	Javascript Example	13
6	C# .NET API.....	14
6.1	C# Example.....	14
7	Programming Enumerations	16

1 WebSocket Communication Protocol

Web Sockets is a next-generation bidirectional communication technology for web applications which operates over a single socket and is exposed via a JavaScript interface in HTML 5 compliant browsers.

Once you get a Web Socket connection with the web server, you can send data from browser to server by calling a **send()** method, and receive data from server to browser by an **onmessage** event handler.

```
var ws = new WebSocket('ws://localhost:8083');
ws.onopen = function()
{
    // Web Socket is connected, send data using send()
    ws.send("SessionID=demo");
};
ws.onmessage = function (evt)
{
    var received_msg = evt.data;
    var F = jQuery.parseJSON(evt.data);
};
ws.onclose = function()
{
    // websocket is closed.
    alert("Connection is closed...");
};
websocket.onerror = function(evt) {
    onError(evt);
};
```

2 Structures and enumerations

Its streaming packet comes with a Device object

```
{ "Device": { "ID": 0, "DeviceID": "", "PublicIP": null, "LanIP": null, "GUID": null, "DeviceType": 1, "LastUpdateDate Time": "\Date(-62135596800000)\", "SocketID": null, "SessionID": "demo", "DeviceExceptionCmd": 0, "ObligatoryTransmission": false }, "
```

Device	
DeviceID	The device hardware unique id
DeviceType	1: Skeleton, 2: Wiimote, 3:BalanceBoard, 4: Mindwave, 5: RgbColorImage, 6: DepthVideo, 7: AndroidSensor, 8: Epoc
LastUpdateDateTime	The date and time of the last device capture
SessionID	The session ID the device belongs to.

3 CAC framework commands (WebSocket)

3.1 Register a client to the session

It's the first command that needs to be sent when the client connects to the CAC server. It registers the client to the specified session. The server enables, by default, the streaming of all the available packets except for the RGB Images (this can be enabled or disabled later on). In the following example "demo" is the name of the session.

```
SessionID=demo
```

3.2 Subscribe / Unsubscribe to a device

The client subscribes/unsubscribes to the devices' information based on filters. For instance, a client can unsubscribe either from any Kinect skeleton or just from a specific Kinect by defining its DeviceId. The value set in the DeviceExceptionCmd parameter defines if the client sending the command subscribes or unsubscribes to the information channel specified by the rest parameters. The common usage is to subscribe/unsubscribe to certain DeviceTypes by configuring the DeviceType.

","DeviceExceptionCmd":1	Unsubscribe
","DeviceExceptionCmd":2	Subscribe
","DeviceExceptionCmd":3	Subscribe to all the available channels regardless of the parameters specified

```
{"ID":0,"DeviceID":"123456789","PublicIP":null,"LanIP":null,"GUID":null,"DeviceType":3,"LastUpdateDateTime":"\\Date(-62135596800000)\\","SocketID":null,"SessionID":"0e87ac96-aa59-4fc1-ad2a-5e8b44ffd45d","DeviceExceptionCmd":1,"ObligatoryTransmission":false}
```

Examples:

Subscribe to the wii balanceboard:

```
{"ID":0,"DeviceID":"","PublicIP":null,"LanIP":null,"GUID":null,"DeviceType":3,"SocketID":null,"SessionID":"demo","DeviceExceptionCmd":2,"ObligatoryTransmission":true}
```

Unsubscribe to the Kinect skeleton:

```
{"ID":0,"DeviceID":"","PublicIP":null,"LanIP":null,"GUID":null,"DeviceType":1,"SocketID":null,"SessionID":"demo","DeviceExceptionCmd":1,"ObligatoryTransmission":true}
```

Subscribe to the Kinect skeleton:

```
{"ID":0,"DeviceID":"","PublicIP":null,"LanIP":null,"GUID":null,"DeviceType":1,"SocketID":null,"SessionID":"demo","DeviceExceptionCmd":2,"ObligatoryTransmission":true}
```

3.3 Register to receive compressed or raw data (obsolete)

```
ReceiveCompressedData=1 or ReceiveCompressedData=0
```

3.4 Publish / stream information as a device

The CAC-framework is a JSON messages exchanging protocol. Given that, publishing device's information is done by simply send() the corresponding JSON string message (following the device's structure) from a client that has previously registered to the session. On every publication the CAC server answers with "OK" to acknowledge that the packet has been received and that it will be forward.

```
ws.send("{\"Device\":{\"ID\":\"0\",\"DeviceID\":\"\",\"PublicIP .....\"});
```

3.5 Data Packet Reception

Each client registered to a session receives the JSON packets it is subscribed to. If the client enable the synchronous communication of the packets, it has to acknowledge each packet reception informing the server that is ready to receive the next available packet. The synchronounous communication can be enabled and disabled as follows:

```
send('SYNCON=1'); or send('SYNCOFF=1');, where 1 corresponds to the DeviceType (see Structures and Enumeration)
```

When the synchronous communication is enabled, the client has to send back the acknowledgement on each packet reception.

```
websocket.send('NEXT=1');, where 1 corresponds to the DeviceType of the packet received (see Structures and Enumeration)
```

4 JSON structures

4.1 Skeleton structure

Skeleton contains an array of Skeletons. Each of the object of the array include the arrays of Joints, the Position and the TrackingState. The TrackingState represents the tracking status of an object according to Kinect (0: NotTracked, 1: PositionOnly, 2: Tracked). The array of Joints contains 25 Joints each of them representing a Join of the body. Each Joint implements a Position and a TrackingState. The corresponding number of the Joints is presents in.

Table 1 Skeleton's joints

0: SpineBase (HipCenter)	1: SpineMid (Spine)	2: Neck (ShoulderCenter)	3: Head
4: ShoulderLeft	5: ElbowLeft	6: WristLeft	7: HandLeft
8: ShoulderRight	9: ElbowRight	10: WristRight	11: HandRight
12: HipLeft	13: KneeLeft	14: AnkleLeft	15: FootLeft
16: HipRight	17: KneeRight	18: AnkleRight	19: FootRight
20: SpineShoulder	21: HandTipLeft	22: ThumbLeft	23: HandTipRight
24: ThumbRight			

4.1.1 Containing one skeleton

```
{
  "Device": {"ID": 0, "DeviceID": "", "PublicIP": null, "LanIP": null, "GUID": null, "DeviceType": 1, "LastUpdateDate": "2023-05-15T10:00:00"},
  "SocketID": null,
  "SessionID": "demo",
  "DeviceExceptionCmd": 0,
  "ObligatoryTransmission": false,
  "Frame": {"ms": 1455279527074, "FrameNumber": 0, "Timestamp": 0, "TrackingMode": 0, "Skeletons": [
    {
      "TrackingId": "72057594038094412",
      "Position": {"X": 0.12804858386516571, "Y": 0.083439305424690247, "Z": 2.5759196281433105},
      "Joints": [
        {
          "JointType": 0,
          "Position": {"X": 0.12804858386516571, "Y": 0.083439305424690247, "Z": 2.5759196281433105},
          "TrackingState": 2,
          "JointType": 1,
          "Position": {"X": 0.16518628597259521, "Y": 0.26960933208465576, "Z": 2.5416672229766846},
          "TrackingState": 2,
          "JointType": 2,
          "Position": {"X": 0.19209247827529907, "Y": 0.45609858632087708, "Z": 2.5062270164489746},
          "TrackingState": 2,
          "JointType": 3,
          "Position": {"X": 0.20525363087654114, "Y": 0.561426043510437, "Z": 2.4933815002441406},
          "TrackingState": 2,
          "JointType": 4,
          "Position": {"X": 0.04683632031083107, "Y": 0.37765863537788391, "Z": 2.5495748519897461},
          "TrackingState": 2,
          "JointType": 5,
          "Position": {"X": 0.02210547961294651, "Y": 0.18943406641483307, "Z": 2.5738632678985596},
          "TrackingState": 2,
          "JointType": 6,
          "Position": {"X": -0.0428231842815876, "Y": -0.011323975399136543, "Z": 2.5725471973419189},
          "TrackingState": 2,
          "JointType": 7,
          "Position": {"X": -0.05731523409485817, "Y": -0.059283334761857986, "Z": 2.5817110538482666},
          "TrackingState": 2,
          "JointType": 8,
          "Position": {"X": 0.30558571219444275, "Y": 0.37409582734107971, "Z": 2.5310385227203369},
          "TrackingState": 2,
          "JointType": 9,
          "Position": {"X": 0.3519568145275116, "Y": 0.23873086273670197, "Z": 2.4745922088623047},
          "TrackingState": 2,
          "JointType": 10,
          "Position": {"X": 0.27481323480606079, "Y": 0.11076648533344269, "Z": 2.4805705547332764},
          "TrackingState": 2,
          "JointType": 11,
          "Position": {"X": 0.2548041045665741, "Y": 0.0810927152633667, "Z": 2.4756999015808105},
          "TrackingState": 2,
          "JointType": 12,
          "Position": {"X": 0.069401763379573822, "Y": 0.08919043093919754, "Z": 2.5551402568817139},
          "TrackingState": 2,
          "JointType": 13,
          "Position": {"X": 0.12404599785804749, "Y": -0.11175079643726349, "Z": 2.509774923324585},
          "TrackingState": 2,
          "JointType": 14,
          "Position": {"X": -0.061535261571407318, "Y": -0.23893290758132935, "Z": 2.5661725997924805},
          "TrackingState": 2,
          "JointType": 15,
          "Position": {"X": -0.0069664567708969116, "Y": -0.19452634453773499, "Z": 2.5516724586486816},
          "TrackingState": 2,
          "JointType": 16,
          "Position": {"X": 0.18416032195091248, "Y": 0.075887702405452728, "Z": 2.5494735240936279},
          "TrackingState": 2,
          "JointType": 17,
          "Position": {"X": 0.16640481352806091, "Y": -0.12964719533920288, "Z": 2.4884281158447266},
          "TrackingState": 2,
          "JointType": 18,
          "Position": {"X": 0.14403526484966278, "Y": -0.33420175313949585, "Z": 2.5904128551483154},
          "TrackingState": 1,
          "JointType": 19,
          "Position": {"X": 0.1
        }
      ]
    }
  ]
}
```

```
7495197057724,"Y":-  
0.24844008684158325,"Z":2.539696216583252},"TrackingState":1},{  
"JointType":20,"Position":{"X":0.18595431745052338,"Y":0.40993168950080872,"Z":2.5158984661102295},  
"TrackingState":2},{  
"JointType":21,"Position":{"X":-0.053055636584758759,"Y":-  
0.11151940375566483,"Z":2.5621547698974609},"TrackingState":2},  
{"JointType":22,"Position":{"X":-  
0.045464679598808289,"Y":-  
0.068596541881561279,"Z":2.5595386028289795},"TrackingState":2},  
{"JointType":23,"Position":{"X":0.22335366904735565,"Y":0.040119614452123642,"Z":2.4622960090637207},  
"TrackingState":2},{"JointType":24,"Position":{"X":0.27659019827842712,"Y":0.064220160245895386,"Z":2.4582667350769043},  
"TrackingState":2}],  
"JointOrientations":null,"TrackingState":2,"Activities":null,"Appearance":null,"ClippedEdges":0,"Engaged":0,"Expressions":null,"HandLeftConfidence":0,"HandLeftState":0,"HandRightConfidence":0,"HandRightState":0,"IsRestricted":false,"IsTracked":false,"Lean":null,"LeanTrackingState":0}}
```

4.1.2 Containing two skeletons

```
{"Device":{"ID":0,"DeviceID":"","PublicIP":null,"LanIP":null,"GUID":null,"DeviceType":1,"LastUpdateDate  
Time":"\Date(-  
6213559680000)\","SocketID":null,"SessionID":"demo","DeviceExceptionCmd":0,"ObligatoryTransmission":false},  
"Frame":{"ms":1455279818140},"FrameNumber":0,"Timestamp":0,"TrackingMode":0,"Skeletons":[{"TrackingId":  
"72057594038094629","Position":{"X":0.24304357171058655,"Y":-  
0.45418128371238708,"Z":1.0710293054580688},"Joints":[{"JointType":0,"Position":{"X":0.2430435717  
1058655,"Y":-  
0.45418128371238708,"Z":1.0710293054580688},"TrackingState":2},{"JointType":1,"Position":{"X":0.26  
48521363735199,"Y":-  
0.24125409126281738,"Z":1.0589051246643066},"TrackingState":2},{"JointType":2,"Position":{"X":0.28  
160792589187622,"Y":-  
0.03557996079325676,"Z":1.0317924022674561},"TrackingState":2},{"JointType":3,"Position":{"X":0.27  
395838499069214,"Y":0.1003895029425621,"Z":1.0200172662734985},"TrackingState":2},{"JointType":  
4,"Position":{"X":0.1250922679901123,"Y":-  
0.061166156083345413,"Z":1.1611778736114502},"TrackingState":2},{"JointType":5,"Position":{"X":-  
0.0864500030875206,"Y":-  
0.14434255659580231,"Z":1.1568388938903809},"TrackingState":2},{"JointType":6,"Position":{"X":-  
0.22196575999259949,"Y":0.016761193051934242,"Z":1.0550642013549805},"TrackingState":2},{"Joint  
Type":7,"Position":{"X":-  
0.24300004541873932,"Y":0.0762510672211647,"Z":1.0299282073974609},"TrackingState":2},{"JointTy  
pe":8,"Position":{"X":0.41427153348922729,"Y":-  
0.11366689950227737,"Z":0.94970500469207764},"TrackingState":2},{"JointType":9,"Position":{"X":0.4  
7881439328193665,"Y":-  
0.2598109245300293,"Z":0.81170684099197388},"TrackingState":2},{"JointType":10,"Position":{"X":0.5  
4633998870849609,"Y":-  
0.40988296270370483,"Z":0.67362296581268311},"TrackingState":1},{"JointType":11,"Position":{"X":0.  
56855475902557373,"Y":-  
0.45666837692260742,"Z":0.63395285606384277},"TrackingState":1},{"JointType":12,"Position":{"X":0.
```

16675782203674316,"Y":-
0.432873010635376,"Z":1.090928316116333},"TrackingState":2,{"JointType":13,"Position":{"X":0.1263
1167471408844,"Y":-
0.61802405118942261,"Z":0.80581843852996826},"TrackingState":1,{"JointType":14,"Position":{"X":0.
086957812309265137,"Y":-
0.82066285610198975,"Z":0.52681833505630493},"TrackingState":1,{"JointType":15,"Position":{"X":0.
032723352313041687,"Y":-
0.80062651634216309,"Z":0.42318981885910034},"TrackingState":1,{"JointType":16,"Position":{"X":0.
30596056580543518,"Y":-
0.45004075765609741,"Z":0.99058026075363159},"TrackingState":2,{"JointType":17,"Position":{"X":0.
46579280495643616,"Y":-
0.25531503558158875,"Z":0.83216768503189087},"TrackingState":2,{"JointType":18,"Position":{"X":0.
69568538665771484,"Y":0.00025405269116163254,"Z":0.64286160469055176},"TrackingState":1,{"Joi
ntType":19,"Position":{"X":0.64438700675964355,"Y":0.035468954592943192,"Z":0.530734837055206
3},"TrackingState":1,{"JointType":20,"Position":{"X":0.27822861075401306,"Y":-
0.0859438106417656,"Z":1.0409523248672485},"TrackingState":2,{"JointType":21,"Position":{"X":-
0.26832011342048645,"Y":0.14543889462947845,"Z":0.99489408731460571},"TrackingState":2,{"Joi
ntType":22,"Position":{"X":-
0.17374899983406067,"Y":0.076078929007053375,"Z":0.96298336982727051},"TrackingState":2,{"Joi
ntType":23,"Position":{"X":0.565117359161377,"Y":-
0.45319047570228577,"Z":0.59081089496612549},"TrackingState":1,{"JointType":24,"Position":{"X":0.
52729415893554688,"Y":-
0.45829695463180542,"Z":0.62160688638687134},"TrackingState":1},"JointOrientations":null,"Trackin
gState":2,"Activities":null,"Appearance":null,"ClippedEdges":0,"Engaged":0,"Expressions":null,"HandLeft
Confidence":0,"HandLeftState":0,"HandRightConfidence":0,"HandRightState":0,"IsRestricted":false,"IsTr
acked":false,"Lean":null,"LeanTrackingState":0,{"TrackingId":"72057594038094772","Position":{"X":0.0
14705789275467396,"Y":0.00355256674811244,"Z":2.277796745300293},"Joints":[{"JointType":0,"Posi
tion":{"X":0.014705789275467396,"Y":0.00355256674811244,"Z":2.277796745300293},"TrackingState"
:2,{"JointType":1,"Position":{"X":0.024808790534734726,"Y":0.21680520474910736,"Z":2.2743468284
606934},"TrackingState":2,{"JointType":2,"Position":{"X":0.035512905567884445,"Y":0.424840271472
93091,"Z":2.2627747058868408},"TrackingState":2,{"JointType":3,"Position":{"X":0.062010351568460
464,"Y":0.55468958616256714,"Z":2.2414352893829346},"TrackingState":2,{"JointType":4,"Position":{"
"X":-
0.13812632858753204,"Y":0.3422391414642334,"Z":2.25986385345459},"TrackingState":2,{"JointType
":5,"Position":{"X":-
0.31578472256660461,"Y":0.42324674129486084,"Z":2.1827435493469238},"TrackingState":2,{"Joint
Type":6,"Position":{"X":-
0.43576177954673767,"Y":0.564809262752533,"Z":2.0519363880157471},"TrackingState":2,{"JointTyp
e":7,"Position":{"X":-
0.50480037927627563,"Y":0.65111339092254639,"Z":2.0027461051940918},"TrackingState":2,{"Joint
Type":8,"Position":{"X":0.19418415427207947,"Y":0.29758346080780029,"Z":2.2629389762878418},"T
rackingState":2,{"JointType":9,"Position":{"X":0.22706595063209534,"Y":0.10537533462047577,"Z":2.
2729084491729736},"TrackingState":2,{"JointType":10,"Position":{"X":0.24759286642074585,"Y":-
0.052413992583751678,"Z":2.3108251094818115},"TrackingState":1,{"JointType":11,"Position":{"X":0.
254719078540802,"Y":-
0.10486608743667603,"Z":2.3263185024261475},"TrackingState":2,{"JointType":12,"Position":{"X":-
0.05147821456193924,"Y":0.019452504813671112,"Z":2.2496042251586914},"TrackingState":2,{"Joi
ntType":13,"Position":{"X":-0.15421298146247864,"Y":-


```

0.39215552806854248,"Z":2.291001558303833},"TrackingState":2},{ "JointType":14,"Position":{"X":-
0.15681499242782593,"Y":-
0.80109018087387085,"Z":2.37483286857605},"TrackingState":2},{ "JointType":15,"Position":{"X":-
0.074053585529327393,"Y":-
0.77556651830673218,"Z":2.341315746307373},"TrackingState":2},{ "JointType":16,"Position":{"X":0.08
0600611865520477,"Y":-
0.01171950064599514,"Z":2.2534661293029785},"TrackingState":2},{ "JointType":17,"Position":{"X":0.0
71500346064567566,"Y":-
0.43553492426872253,"Z":2.2952649593353271},"TrackingState":2},{ "JointType":18,"Position":{"X":0.0
069591137580573559,"Y":-
0.66796004772186279,"Z":2.3738350868225098},"TrackingState":2},{ "JointType":19,"Position":{"X":-
0.011900357902050018,"Y":-
0.72892856597900391,"Z":2.2806789875030518},"TrackingState":2},{ "JointType":20,"Position":{"X":0.0
32812502235174179,"Y":0.37358862161636353,"Z":2.2672402858734131},"TrackingState":2},{ "JointTy
pe":21,"Position":{"X":-
0.54014325141906738,"Y":0.70758426189422607,"Z":1.9926685094833374},"TrackingState":2},{ "Joint
Type":22,"Position":{"X":-
0.47598287463188171,"Y":0.65755975246429443,"Z":1.9513684511184692},"TrackingState":2},{ "Joint
Type":23,"Position":{"X":0.27139672636985779,"Y":-
0.10755219310522079,"Z":2.3115599155426025},"TrackingState":1},{ "JointType":24,"Position":{"X":0.2
2607968747615814,"Y":-
0.103171706199646,"Z":2.323073148727417},"TrackingState":1}}, "JointOrientations":null,"TrackingStat
e":2,"Activities":null,"Appearance":null,"ClippedEdges":0,"Engaged":0,"Expressions":null,"HandLeftConfi
dence":0,"HandLeftState":0,"HandRightConfidence":0,"HandRightState":0,"IsRestricted":false,"IsTracke
d":false,"Lean":null,"LeanTrackingState":0}}

```

4.2 RGB Image structure

The RGBVideoSourceData contains the RGBVideo object which incorporates the width and height of the image and the base64String of the image data. An image encoded into a base64 string and place it directly withing an HTML image tag or as a CSS background URL.

The base64String can be place whitin an HTML image tag as:

```

image.setAttribute("src","data:image/jpg;base64,"+e.detail.RGBVideoSourceData.RGBVideo.base64String);

```

```

{"Device":{"ID":0,"DeviceID":"A00362901906044A","PublicIP":null,"LanIP":null,"GUID":null,"DeviceType":5,"LastUpdateDateTime":"\Date(-62135596800000)\","SocketID":null,"SessionID":"0e87ac96-aa59-4fc1-ad2a-5e8b44ffd45d","DeviceExceptionCmd":0,"ObligatoryTransmission":false},"RGBVideo":{"Width":640,"Height":480,"FPS":30,"base64String":"/9j/4AAQSkZJRgABAQEAYABgAAD/2wBD.....r3bdVuijmb3Fy22P/9k="}}

```

4.3 Android sensors structure

```
{ "Device": { "ID": 0, "DeviceID": "123456789", "PublicIP": null, "LanIP": null, "GUID": null, "DeviceType": 7, "LastUpdateDateTime": "\Date(-6213559680000)\", "SocketID": null, "SessionID": "demo", "DeviceExceptionCmd": 0, "ObligatoryTransmission": false, "TouchMousePosition": { "MousePoints": { "X": 2.5, "Y": 8.7, "X": 2.5, "Y": 8.7 }, "SingleTap": false, "DoubleTap": false, "LongPress": false, "Fling": { "Point1": { "X": 0, "Y": 0 }, "Point2": { "X": 0, "Y": 0 }, "velocityX": 0, "velocityY": 0 }, "AccelerometerSensor": { "Values": { "X": 1, "Y": 2, "Z": 3 }, "AmbientTemperatureSensor": { "Value": 4 }, "GyroscopeSensor": { "Values": { "X": 1, "Y": 2, "Z": 3 }, "LightSensor": { "Value": 5 }, "MagneticFieldSensor": { "Values": { "X": 11, "Y": 12, "Z": 13 }, "PressureSensor": { "Value": 14 }, "ProximitySensor": { "Value": 15 }, "ButtonState": { "A": true, "B": false, "Down": false, "Home": false, "Right": false, "Up": false }, "Battery": 5 }
```

4.4 Wii BalanceBoard and Wii Remote control structure

The WiiSourceData contains AccelState (in case of wii remote controller) BalanceBoardState and ButtonState. The BalanceBoardState contains Weight in Kg, the CenterOfGravity (x and y axis) and the components of the weight at the 4 corners of the balance board (BottomLeft, BottomRight, TopLeft, TopRight). The ButtonState includes information of the buttons status being BalanceBoard or Wiimote.



Figure 1 Wii Balanceboard device (Nintendo)

```
{ "Device": { "ID": 0, "DeviceID": "123456789", "PublicIP": null, "LanIP": null, "GUID": null, "DeviceType": 3, "LastUpdateDateTime": "\Date(-6213559680000)\", "SocketID": null, "SessionID": "0e87ac96-aa59-4fc1-ad2a-5e8b44ffd45d", "DeviceExceptionCmd": 0, "ObligatoryTransmission": false, "AccelState": null, "BalanceBoardState": { "CenterOfGravity": { "X": 8.404858, "Y": -7.794562 }, "WeightKg": 80, "BottomLeftKg": 85.56, "BottomRightKg": 69.44349, "TopLeftKg": 81.27788, "TopRightKg": 127.349113 }, "Battery": 111.526, "ButtonState": { "A": false, "B": false, "Down": false, "Home": false, "Left": false, "Minus": false, "One": false, "Plus": false, "Right": false, "Two": false, "Up": false, "TriggerL": false, "TriggerR": false, "X": false, "Y": false, "ZL": false, "ZR": false, "StrumDown": false, "StrumUp": false }, "ClassicControllerState": null, "DrumsState": null, "Extension": true, "ExtensionType": 5, "GuitarState": null, "IRState": null, "LEDState": { "LED1": false, "LED2": false, "LED3": false, "LED4": false }, "MotionPlusState": null, "NunchukState": null, "Rumble": false, "TaikoDrumState": null }
```

4.5 Mindwave structure



Figure 2 Mindwave neurosky

The MindwaveSourceData contains the values of the processed EEG power spectrums (Alpha1, Alpha2, Beta1, Beta2, Delta, Gamma1, Gamma2, Theta), output of NeuroSky proprietary eSense meter for Attention, Meditation, and other future meters and signal quality analysis (can be used to detect poor contact and whether the device is off the head)

5 Javascript API

5.1 Include the Javascript source files

Add the required JavaScript source files to your HTML page. The CAC Framework JavaScript API is defined in functionsInputDevice.js. Some of the UI elements such as cursors and buttons require the JQuery API.

```
<script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>  
<script src="http://www.cac-framework.com/js-libs/functionsInputDevice.js"></script>
```

5.2 Define SessionID and Server address

Following the server client architecture, each client has to define the cac framework server address. The default on is ws://kedip16.med.auth.gr:8083.

A group of input devices and clients define a session. Once a device streams information the rest members of the session receive the information. In order to define the session, the developer has to implement the following javascript function before the declaration of the functionsInputDevice.js. The following example register the web application to the Session DEMO for testing purposes. Data and information of all the supported devices is currently streamed (playback) at the session with ID="DEMO"

```

<script>
  function getInputDeviceGuid() {
    inputDeviceGuid = "demo";
    return inputDeviceGuid;
  }

  function getWebSocketsInputDeviceServer() {
    webSocketsInputDeviceServer = "ws://localhost:8083";
    return webSocketsInputDeviceServer;
  }
</script>
<script src="http://www.cac-framework.com/js-lib/functionsInputDevice.js"></script>

```

5.3 Add a handler to process incoming stream frames

To process incoming data streams from the input devices, call the appropriate `addEventListener` function to set a callback function that is called whenever a data frame is received. The call back function may be fired on event **skeletonEvent** when a skeleton of the Kinect sensor is received, **wiiEvent** when a wiiremote or wii balanceboard frame is received, **MindwaveEvent** when a mindwave frame is received and **RGBVideoEvent** when an image is available. The following examples implement the `addEventListener`s.

```

<script type="text/javascript">
document.addEventListener("skeletonEvent", function (e) {
  //e.detail.SkeletonSourceData.Skeletons[0].Joints[0].Position.X
});
document.addEventListener("wiiEvent", function (e) {
  //e.detail.WiiSourceData.BalanceBoardState.WeightKg;
});
document.addEventListener("MindwaveEvent", function (e) {
  // e.detail.MindwaveSourceData.Attention;
});
document.addEventListener("RGBVideoEvent", function (e) {
  //e.detail.RGBVideoSourceData.RGBVideo.base64String;
});
document.addEventListener("CacFrameworkEvent", function(e) {
  ConfigCacFrameworkSource('RGBVideo', true);
});
</script>

```

5.4 Subscribing/Unsubscribing to specific streaming sources

The javascript API provides the developers with the functionality of enabling / disabling streaming reception of individual sources.

```
ConfigCacFrameworkSource('Skeleton', true);
ConfigCacFrameworkSource('Wii mote', true);
ConfigCacFrameworkSource('WiiBalanceboard', true);
ConfigCacFrameworkSource('Mindwave', true);
ConfigCacFrameworkSource('RGBVideo', true);
```

5.5 Javascript Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Usecase for Developers</title>
    <script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
    <script>
      function getInputDeviceGuid() {
        inputDeviceGuid = "demo";
        return inputDeviceGuid;
      }
      function getWebSocketsInputDeviceServer() {
        webSocketsInputDeviceServer = "ws://localhost:8083";
        return webSocketsInputDeviceServer;
      }
    </script>
    <script src="http://www.cac-framework.com/js-lib/functionsInputDevice.js"></script> </head>
  <body>
    <form>
      <label>Y position of user's HipCenter
        <input id="y-position" type="text" name="y-position" />
      </label><br>
      <label>User's alpha rhythm
        <input id="alpha-rhythm" type="text" name="alpha-rhythm" />
      </label><br>
      <label>User's center of mass (x-axis)
        <input id="center-of-gravity" type="text" name="center-of-gravity" />
      </label>
      <label>Android Phone acceleration on x axis
        <input id="android-x-axis" type="text" name="android-x-axis" />
      </label>
    </form>
    <p>RGB Video</p>
    <img src="" alt="RGB image" id="video">
    <script>
      document.addEventListener("skeletonEvent", function(e) {
        if (e.detail.SkeletonSourceData.Skeletons[0].Joints[0].TrackingState === 2) {
          document.getElementById("y-position").value = e.detail.SkeletonSourceData.Skeletons[0].Joints[0].Position.Y;
        } else {
          document.getElementById("y-position").value = 'Not tracked';
        }
      });
      document.addEventListener("wiiEvent", function(e) {
        document.getElementById("center-of-gravity").value = e.detail.WiiSourceData.BalanceBoardState.WeightKg;
```

```

    });

    document.addEventListener("MindwaveEvent", function(e) {
        document.getElementById("alpha-rhythm").value = e.detail.MindwaveSourceData.Attention;
    });

    document.addEventListener("AndroidSensorEvent", function(e) {
        document.getElementById("android-x-axis").value = e.detail.AndroidSensorSourceData.AccelerometerSensor.Values.X;
    });

    document.addEventListener("RGBVideoEvent", function(e) {
        document.getElementById("video").setAttribute("src", "data:image/jpg;base64," +
e.detail.RGBVideoSourceData.RGBVideo.base64String);
    });

    document.addEventListener("CacFrameworkEvent", function(e) {
        ConfigCacFrameworkSource('RGBVideo', true);
    });

</script>
</body>
</html>

```

6 C# .NET API

The [CACServiceClient](#) in the following example is an instance of the CACServiceClient Library available for download at: http://cac-framework.com/files/CAC_Csharp_Library.zip

6.1 C# Example

```

public CACServiceClient KinectServiceClientSkeleton;
public CACServiceClient KinectServiceClientColor;

public deviceKinect()
{
    PacketsFrequencyTimer = new System.Threading.Timer(PacketsFrequencyTimer_Tick, null, 0, 1000);
    KinectServiceClientSkeleton = new CACServiceClient(StaticValues.WebSocketCACServiceAddress);
    KinectServiceClientSkeleton.ConnectionToServerEstablished +=
KinectServiceClientSkeleton_ConnectionToServerEstablished;
    KinectServiceClientColor = new CACServiceClient(StaticValues.WebSocketCACServiceAddress);
    KinectServiceClientColor.ConnectionToServerEstablished += KinectServiceClientColor_ConnectionToServerEstablished;
    KinectServiceClientColor.DataTransmitCompleted += KinectServiceClientColor_DataTransmitCompleted;
    this.KinectStart();
}

void KinectServiceClientColor_ConnectionToServerEstablished(object sender, EventArgs e)
{
    KinectServiceClientColor.ExceptionMindwave(DeviceExceptionCmdEnum.AddException);
    KinectServiceClientColor.ExceptionRGB(DeviceExceptionCmdEnum.AddException);
    KinectServiceClientColor.ExceptionSkeleton(DeviceExceptionCmdEnum.AddException);
    KinectServiceClientColor.ExceptionWiiBalanceboard(DeviceExceptionCmdEnum.AddException);
    KinectServiceClientColor.ExceptionWiimote(DeviceExceptionCmdEnum.AddException);
}

void KinectServiceClientSkeleton_ConnectionToServerEstablished(object sender, EventArgs e)
{

```

```
KinectServiceClientSkeleton.ExceptionMindwave(DeviceExceptionCmdEnum.AddException);
KinectServiceClientSkeleton.ExceptionRGB(DeviceExceptionCmdEnum.AddException);
KinectServiceClientSkeleton.ExceptionSkeleton(DeviceExceptionCmdEnum.AddException);
KinectServiceClientSkeleton.ExceptionWiiBalanceboard(DeviceExceptionCmdEnum.AddException);
KinectServiceClientSkeleton.ExceptionWiimote(DeviceExceptionCmdEnum.AddException);
}

private void Reader_FrameArrived(object sender, BodyFrameArrivedEventArgs e)
{
    bool dataReceived = false;
    using (BodyFrame bodyFrame = e.FrameReference.AcquireFrame())
    {
        if (bodyFrame != null)
        {
            Error er = KinectServiceClientSkeleton.SaveSkeletonToServerKinect(bodyFrame);
        }
    }
}

private void Reader_ColorFrameArrived(object sender, ColorFrameArrivedEventArgs e)
{
    if (StreamVideoEnable == true)
    {
        using (ColorFrame colorFrame = e.FrameReference.AcquireFrame())
        {
            if (colorFrame != null)
            {
                Error ErrorColorVideo = KinectServiceClientColor.SaveColorVideoToServerKinect(colorFrame);
            }
        }
    }
}
```

7 Programming Enumerations

```
public enum SkeletonTrackingStateJSON
{
    NotTracked = 0,
    PositionOnly = 1,
    Tracked = 2
}
```

```
public enum SkeletonTrackingModeJSON
{
    Default = 0,
    Seated = 1
}
```

```
public enum TrackingStateJSON
{
    NotTracked = 0,
    Inferred = 1,
    Tracked = 2
}
```

```
public enum JointTypeJSON
{
    HipCenter = 0, //kinect1
    SpineBase = 0, //kinect2
    Spine = 1, //kinect1
    SpineMid = 1, //kinect2
    ShoulderCenter = 2, //kinect1
    Neck = 2, //kinect2
    Head = 3, //kinect1 and Kinect2
    ShoulderLeft = 4, //kinect1 and Kinect2
    ElbowLeft = 5, //kinect1 and Kinect2
    WristLeft = 6, //kinect1 and Kinect2
    HandLeft = 7, //kinect1 and Kinect2
    ShoulderRight = 8, //kinect1 and Kinect2
    ElbowRight = 9, //kinect1 and Kinect2
    WristRight = 10, //kinect1 and Kinect2
    HandRight = 11, //kinect1 and Kinect2
    HipLeft = 12, //kinect1 and Kinect2
    KneeLeft = 13, //kinect1 and Kinect2
    AnkleLeft = 14, //kinect1 and Kinect2
    FootLeft = 15, //kinect1 and Kinect2
    HipRight = 16, //kinect1 and Kinect2
    KneeRight = 17, //kinect1 and Kinect2
    AnkleRight = 18, //kinect1 and Kinect2
    FootRight = 19, //kinect1 and Kinect2
    SpineShoulder = 20, //kinect2
    HandTipLeft = 21, //kinect2
    ThumbLeft = 22, //kinect2
    HandTipRight = 23, //kinect2
    ThumbRight = 24, //kinect2
}
```

```
public enum ActivityJSON
{
    EyeLeftClosed = 0,
    EyeRightClosed = 1,
    MouthOpen = 2,
}
```



```
MouthMoved = 3,  
LookingAway = 4  
}  
  
public enum AppearanceJSON  
{  
    WearingGlasses = 0  
}  
  
public enum Expression  
{  
    Neutral = 0,  
    Happy = 1  
}  
  
public enum DetectionResultJSON  
{  
    Unknown = 0,  
    No = 1,  
    Maybe = 2,  
    Yes = 3  
}  
  
public enum FrameEdgesJSON  
{  
    None = 0,  
    Right = 1,  
    Left = 2,  
    Top = 4,  
    Bottom = 8  
}  
  
public enum TrackingConfidenceJSON  
{  
    Low = 0,  
    High = 1  
}  
  
public enum HandStateJSON  
{  
    Unknown = 0,  
    NotTracked = 1,  
    Open = 2,  
    Closed = 3,  
    Lasso = 4  
}
```