AVAR2 Data Model for Service Integration - Complete Reference

HTTP API Endpoint

URL: `http://[device-ip]:8080/avar`

Method: POST

Content-Type: application/json

Root Schema Formats

The API accepts three different root formats:

1. Object Format with Diagram ID

```
{
  "id": 123,                    // Optional: Integer diagram ID for updates
  "elements": [...],            // Array of 3D visualization elements
}
```

2. 2D/RT Layout Format

```
{
  "id": 456,                    // Optional: diagram ID
  "RTelements": [...],          // Array of 2D layout elements
}
```

3. Direct Array Format

```
[...]  // Direct array of ElementDTO objects (no wrapper)
```

Core Element Schema: `ElementDTO`

Standard 3D Elements (`"elements"` key)

```
{
  "id": "node1",                // String/Number: Element identifier
  "type": "element",            // String: Element type (element, edge, camera)
  "model": "Box",               // String: Alternative to type field

  "position": [0, 0, 0],        // Array[Double]: 3D coordinates (X, Y, Z)
  "color": [1.0, 0.5, 0.2, 1.0], // Array[Double]: RGBA color (0.0-1.0)
  "extent": [1.0, 1.0, 1.0],    // Array[Double]: Element size dimensions

  "shape": {                    // Object OR String: Shape definition
    "shapeDescription": "uvSphere", // String: 3D shape type
    "extent": [1.0, 1.0, 1.0],  // Array[Double]: Shape dimensions
    "text": "Label",             // String: Text label
    "color": [1.0, 0.0, 0.0, 1.0], // Array[Double]: Shape color
    "id": "shape1"               // String: Shape identifier
  },

  "from_id": "node1",           // String/Number: Edge source
  "to_id": "node2",             // String/Number: Edge destination
  "from_position": [0, 0, 0],   // Array[Double]: Edge start position
  "to_position": [2, 0, 0],     // Array[Double]: Edge end position
  "interactions": ["tap", "drag"] // Array[String]: Supported interactions
}
```

2D Layout Elements (`"RTelements"` key)

```
{
  "id": 0,                      // Number: Element identifier
  "type": "RTelement",          // String: Always "RTelement" for 2D layouts
  "position": [71.5, 32.5],     // Array[Double]: 2D coordinates (X, Y)
  "color": [0.74, 0.74, 0.74, 1.0], // Array[Double]: RGBA color

  "shape": {
    "shapeDescription": "RTBox", // String: 2D shape type
    "extent": [5.0, 5.0],        // Array[Double]: Width, Height
    "text": "nil"                 // String: Text content (can be "nil")
  }
}
```

Supported Shape Types

3D Shapes (for `"elements"`)

- `"uvSphere"` - UV-mapped sphere
- `"cube"` - 3D cube/box
- `"Box"` - Alternative cube notation
- `"Cylinder"` - 3D cylinder
- `"line"` - Line connector for edges
- Custom shape strings supported

2D Shapes (for `"RTelements"`)

- `"RTBox"` - 2D rectangle/box for layouts
- Custom 2D shape strings supported

Special Element Types

Camera Element

```
{
  "id": "&lt;genid:0&gt;",
  "type": "camera",
  "position": [0.0, 0.0, 5.0]
}
```

Edge/Connection Element

```
{
  "id": "edge1",
  "type": "edge",
  "from_id": "nodeA",
  "to_id": "nodeB",
  "from_position": [0, 0, 0],
  "to_position": [2, 0, 0],
  "shape": {
    "shapeDescription": "line",
    "extent": [2.0, 0.0, 0.0],
    "color": [1.0, 1.0, 0.0, 1.0]
  }
}
```

Example Use Cases

Bar Chart (3D)

```
{
  "elements": [
    {
      "id": "bar1",
      "type": "element",
      "shape": {
        "shapeDescription": "cube",
        "extent": [1.0, 6.1, 1.0]
      },
      "position": [6.0, 3.05, 10.0],
      "color": [1.0, 1.0, 1.0, 1.0]
    }
  ]
}
```

Network Graph with Edges

```
{
  "elements": [
    {
      "id": "A", "type": "element", "shape": "uvSphere",
      "position": [0, 0, 0], "color": [0.0, 0.0, 1.0, 1.0]
    },
    {
      "id": "B", "type": "element", "shape": "uvSphere",
      "position": [2, 0, 0], "color": [0.0, 0.0, 1.0, 1.0]
    },
    {
      "type": "edge", "from_id": "A", "to_id": "B",
      "shape": {"shapeDescription": "line", "extent": [2.0, 0.0, 0.0]},
      "color": [1.0, 1.0, 0.0, 1.0]
    }
  ]
}
```

2D Layout/TreeMap

```
{
  "RTelements": [
    {
      "id": 0, "type": "RTelement",
      "position": [71.5, 32.5],
      "color": [0.74, 0.74, 0.74, 1.0],
      "shape": {
        "shapeDescription": "RTBox",
        "extent": [50.0, 30.0],
        "text": "Category A"
      }
    }
  ]
}
```

Key Features

- **Multi-format support**: 3D elements, 2D layouts, or direct arrays
- **Flexible IDs**: String or numeric identifiers with auto-conversion
- **Coordinate systems**: 3D (X,Y,Z) for elements, 2D (X,Y) for RT layouts
- **Color specification**: RGBA normalized doubles (0.0-1.0)
- **Shape flexibility**: String shortcuts or detailed shape objects
- **Graph connections**: Connect elements via `from_id` / `to_id` with positions
- **Diagram updates**: Include root-level `id` to update existing visualizations
- **Interactive elements**: Define supported interactions per element
- **Cross-platform rendering**: iOS, visionOS, macOS with AR/3D support
- **Real-time collaboration**: Multi-device session sharing capabilities

The app processes all formats and renders them as interactive 3D/AR visualizations with support for collaborative multi-device sessions via MultipeerConnectivity.

cURL Example

```
curl -X POST http://192.168.1.100:8080/avar \
  -H "Content-Type: application/json" \
  -d '{
    "id": "demo1",
    "elements": [
      {
        "id": "sphere1",
        "type": "element",
        "shape": "uvSphere",
        "position": [0, 0, 0],
        "color": [1.0, 0.0, 0.0, 1.0]
      }
    ]
  }'
```