



NF
25

CONNECTING
THE WORLD



NIAGARA FORUM APAC | 9-10 SEPTEMBER 2025

BIM INTEGRATION





Tong Zhong

Sr Appl/Sys Sales Engineer

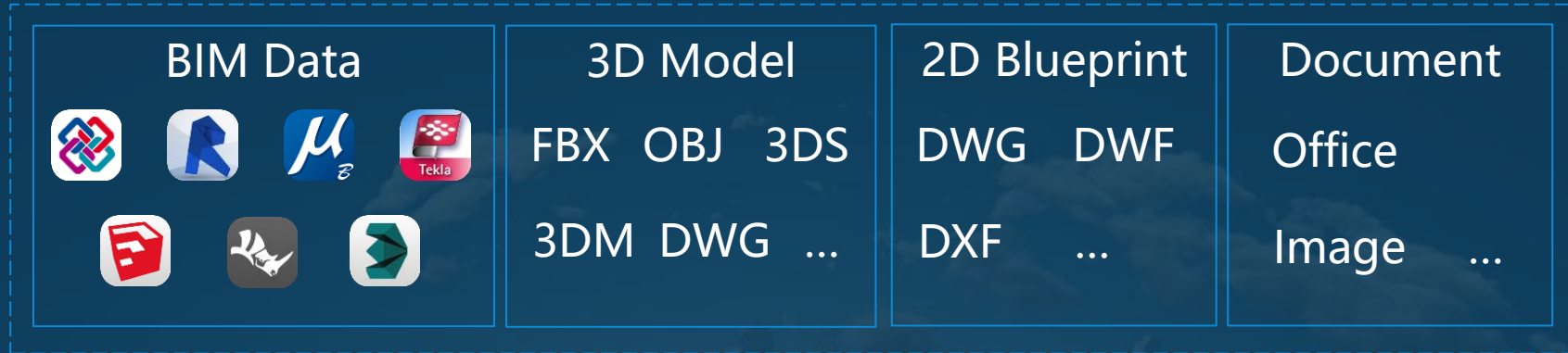
Tridium

NF
25

BIM INTEGRATION

BIM

Building Information Modeling



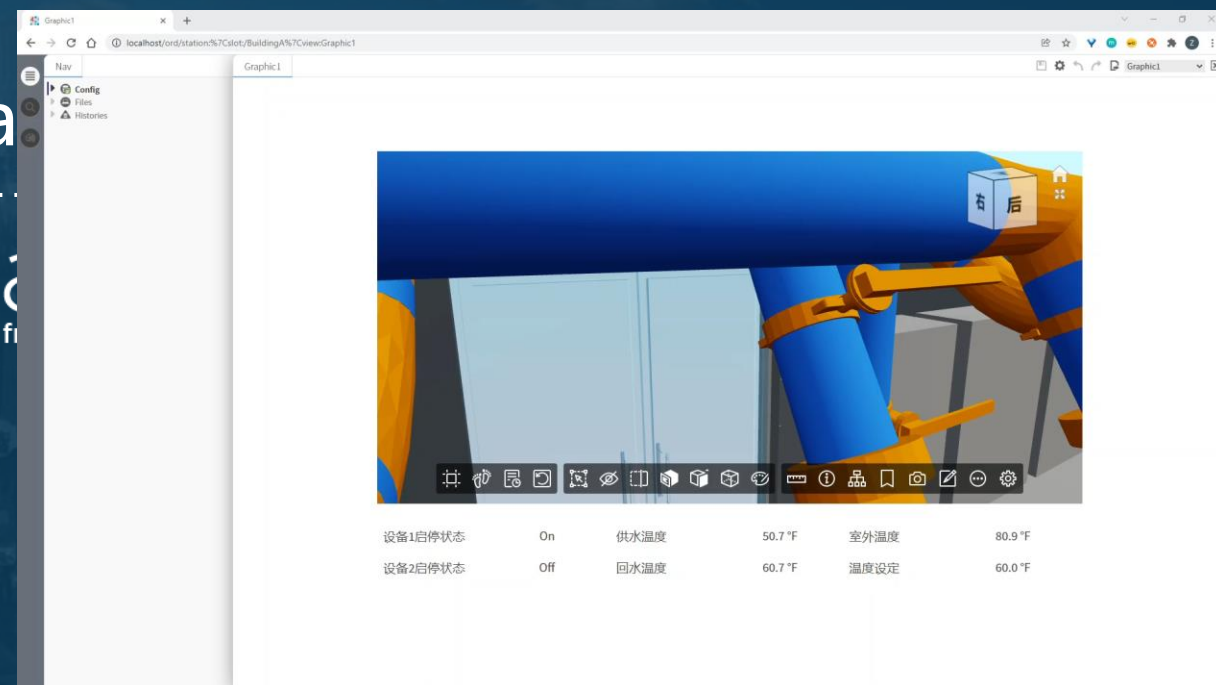
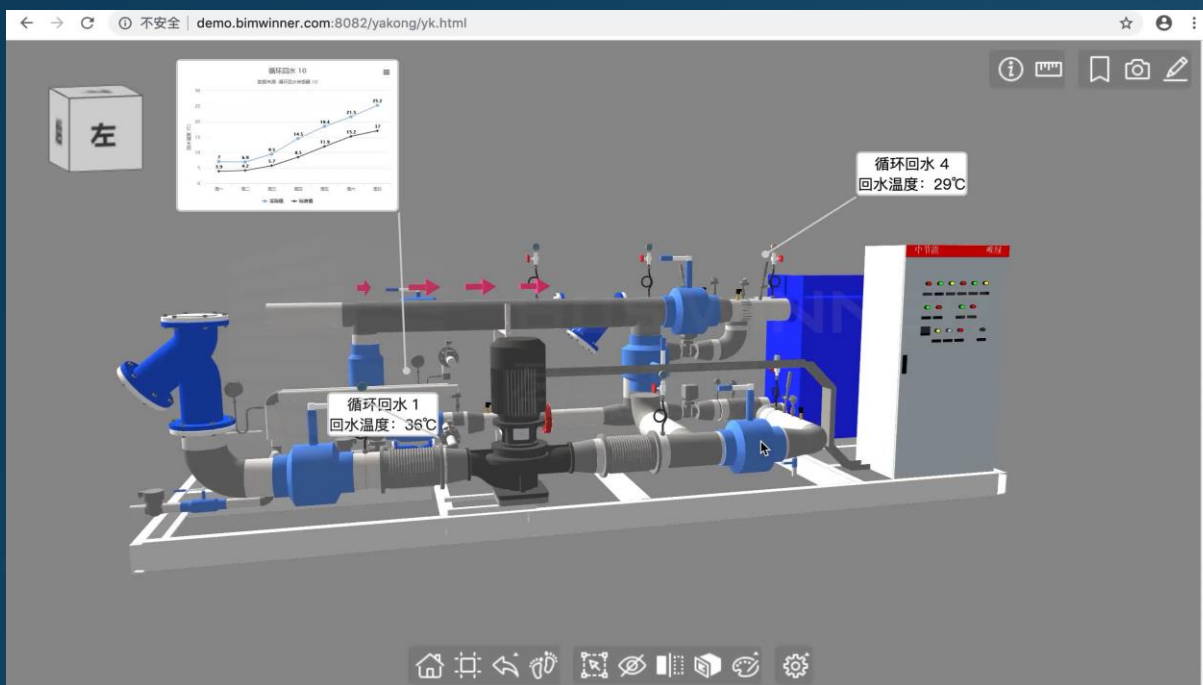
BIM Platform

niagara
framework®

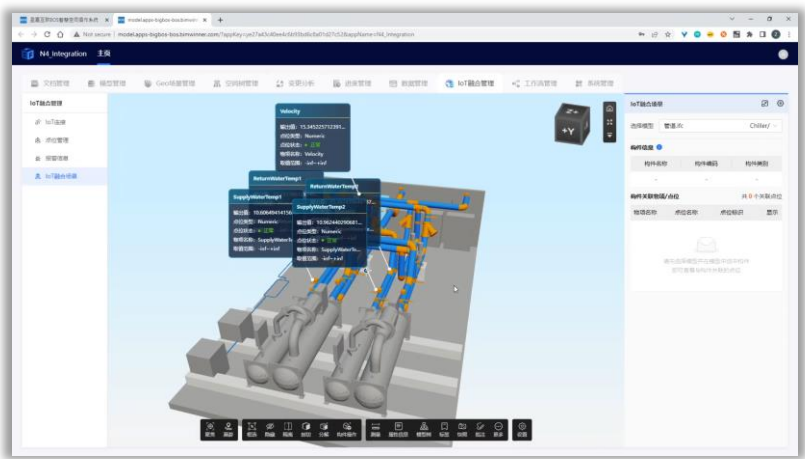
- Protocol: Ethernet/IP, oBIX, OPC UA, BACnet, Modbus, ONVIF, DALI, Profinet, SNMP, ..., & Private Protocols
- Fieldbus: RS485/RS232/LON/CAN/Ethernet/EIB/ZigBee/ZWave/LoRa/...



+



BIM Software UI



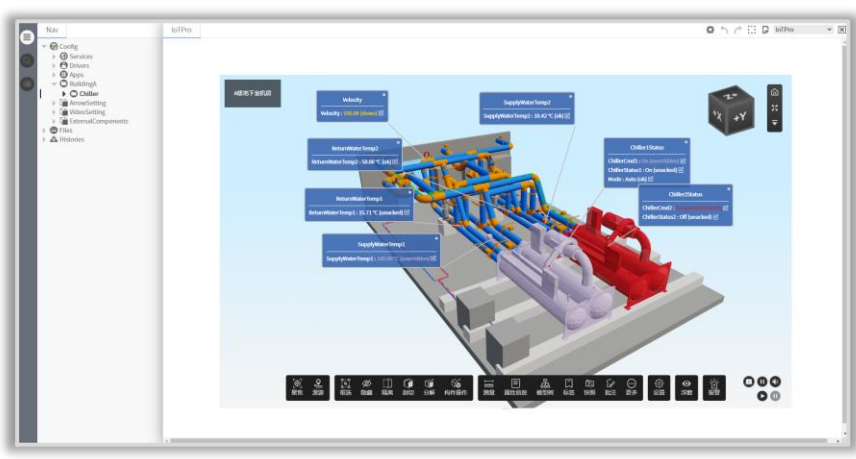
BIM Platform

Device Info (MQTT, oBIX, etc.)



niagara
framework®

Niagara BAJAUX Widget



niagara
framework®

Model Info (HTTPs)



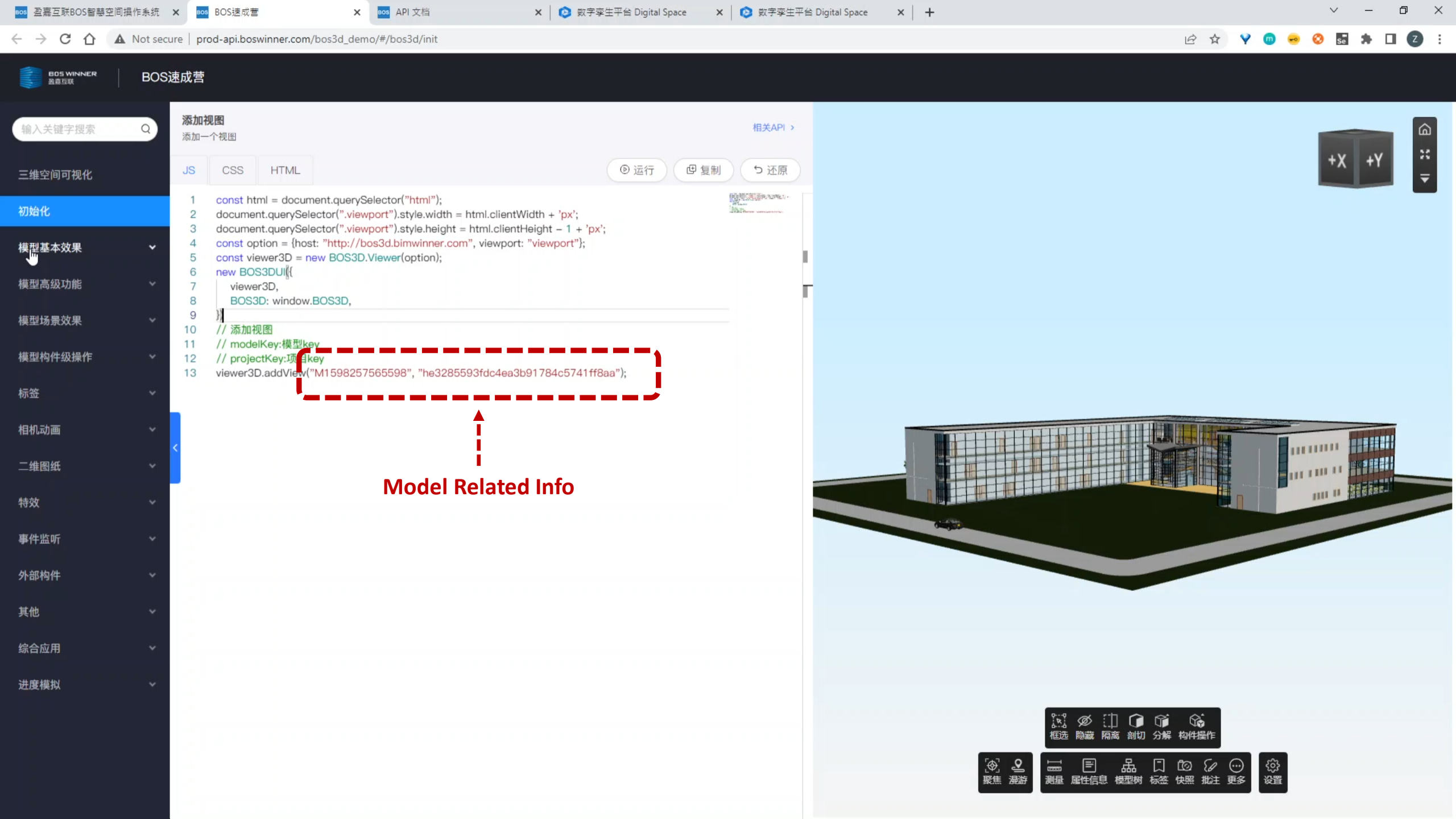
BIM Platform

NIAGARA FORUM
CONNECTING
THE WORLD



EXAMPLE

NIAGARA FORUM
CONNECTING
THE WORLD



Login Service

```
private OkHttpClient createHttpClient()
{
    return AccessController.doPrivileged((PrivilegedAction<OkHttpClient>) () ->
    {
        try
        {
            TrustManagerFactory trustManagerFactory =
                TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
            trustManagerFactory.init((KeyStore) null);
            TrustManager[] trustManagers = trustManagerFactory.getTrustManagers();
            if (trustManagers.length != 1 || !(trustManagers[0] instanceof X509TrustManager))
            {
                throw new IllegalStateException("Unexpected default trust managers:"
                    + Arrays.toString(trustManagers));
            }
            X509TrustManager trustManager = (X509TrustManager) trustManagers[0];

            client = new OkHttpClient.Builder()
                .sslSocketFactory((SSLSocketFactory) HttpsConnection.getDefaultSocketFactory(), trustManager)
                .hostnameVerifier((s, sslSession) -> true)
                .connectTimeout(getConnectTimeout().getSeconds(), TimeUnit.SECONDS)
                .writeTimeout(getWriteTimeout().getSeconds(), TimeUnit.SECONDS)
                .readTimeout(getReadTimeout().getSeconds(), TimeUnit.SECONDS)
                .build();
        }
        catch (Exception e)
        {
            LOGGER.log(Level.WARNING, msg: "Create HTTP client failed: ", e);
        }
        return client;
    });
}
```

```
if (client != null)
{
    String uri = getLoginUri() + "/bospersonnel/service/" + getAppKey() + "/users/login";
    RequestBody body = new MultipartBody.Builder()
        .setType(MultipartBody.FORM)
        .addFormDataPart(name: "name", getUsernameAndPassword().getUsername())
        .addFormDataPart(name: "password", getUsernameAndPassword().getPassword().getValue())
        .build();

    Request request = new Request.Builder()
        .url(uri)
        .post(body)
        .build();

    client.newCall(request).enqueue(new Callback()
    {
        @Override
        public void onFailure(Call call, IOException e)
        {
            String requestUrl = call.request().url().toString();
            LOGGER.log(Level.WARNING, msg: "Request for " + requestUrl + " failed:", e);
        }

        @Override
        public void onResponse(Call call, Response response) throws IOException
        {
            try (ResponseBody responseBody = response.body())
            {
                String requestUrl = call.request().url().toString();
                setLastResponseCode(response.code());
                if (!response.isSuccessful())
                {
                    LOGGER.warning(String.format("%s unexpected code: %s", requestUrl, response));
                }
                else
                {
                    assert responseBody != null;
                    String body = responseBody.string();
                    parseLoginResponse(body);
                    LOGGER.log(Level.FINE, msg: "Received response: " + body);
                }
            }
        }
    });
}
```

Login Service

BosAuthService (Bos Auth Service)

Status	{ok}
Fault Cause	
Enabled	☒ true
Login Uri	https://bosgw.bimwinner.com
App Key	
Username And Password	Username: Password:
Last Login Time	null
Last Response Code	0
Last Response Message	
Bos Platform	☒ true
Interval	30 minutes {Sun Mon Tue Wed Thu Fri Sa...}
Trigger Mode	Interval
Time Of Day	☐ Start Time 12:00:00 AM CST End Time 11:59:59 PM CST
Days Of Week	☒ Sun ☒ Mon ☒ Tue ☒ Wed ☒ Thu ☒ Fri ☒ Sat
Last Trigger	12-Apr-2023 11:03 AM CST
Next Trigger	12-Apr-2023 11:33 AM CST

Slot Sheet						
Slot	#	Name	Display Name	Definition	Flags	Type
☐ Property	0	status	Status	Frozen	rt	baja:Status
☐ Property	1	faultCause	Fault Cause	Frozen	rt	baja:String
☐ Property	2	enabled	Enabled	Frozen		baja:Boolean
☐ Property	3	loginUri	loginUri	Frozen		baja:String
☐ Property	4	appKey	appKey	Frozen		baja:String
☐ Property	5	usernameAndPassword	usernameAndPassword	Frozen		baja:UsernameAndPassword
☐ Property	6	token	token	Frozen	rh	baja:String
☐ Property	7	modelDb	modelDb	Frozen	rh	baja:String
☐ Property	8	lastLoginTime	lastLoginTime	Frozen	r	baja:AbsTime
☐ Property	9	lastResponseCode	lastResponseCode	Frozen	r	baja:Integer
☐ Property	10	lastResponseMessage	lastResponseMessage	Frozen	r	baja:String
☐ Property	11	connectTimeout	connectTimeout	Frozen	h	baja:RelTime
☐ Property	12	writeTimeout	writeTimeout	Frozen	h	baja:RelTime
☐ Property	13	readTimeout	readTimeout	Frozen	h	baja:RelTime
☐ Property	14	worker	worker	Frozen	h	dkBos:BosAuthWorker
☐ Property	15	bosPlatform	bosPlatform	Frozen		baja:Boolean
☒ Action	16	login	login	Frozen	a	void (void)
☐ Property	17	Interval	Interval	Dynamic		control:TimeTrigger
☐ Property	18	Link	Link	Dynamic		baja:Link

Widget

```
@NiagaraType
@NiagaraSingleton
public class BBos3DWidget extends BSingleton implements BIJavaScript, BIFormFactorMax
{
    /*+ ----- BEGIN BAJA AUTO GENERATED CODE ----- +*/
    /*@ $com.tridium.bos.ux.BBos3DWidget(3933907701)1.0$ @*/
    /* Generated Wed Dec 01 17:00:17 CST 2021 by Slot-o-Matic (c) Tridium, Inc. 2012 */

    public static final BBos3DWidget INSTANCE = new BBos3DWidget();

    //////////////////////////////////////
    // Type
    //////////////////////////////////////

    @Override
    public Type getType() { return TYPE; }
    public static final Type TYPE = Sys.loadType(BBos3DWidget.class);

    /*+ ----- END BAJA AUTO GENERATED CODE ----- +*/

    private static final JsInfo JS_INFO =
        JsInfo.make(
            BOrd.make(s: "module://bos/rc/Bos3DWidget.js"),
            BBosJsBuild.TYPE
        );

    @Override
    public JsInfo getJsInfo(Context cx) { return JS_INFO; }
}
```

Widget

```
define(['bajaux/Widget',  
  'bajaux/mixin/subscriberMixin',  
  'Promise',  
  'jquery',  
  'nmodule/bos/rc/BOS3DUI.min',  
  'nmodule/bos/rc/BOS3D.min',  
  'css!nmodule/bos/rc/BOS3DUI.min'], Function (  
  Widget,  
  subscriberMixin,  
  Promise,  
  $,  
  BOS3DUI,  
  BOS3D) {
```

```
▼ bos-ux [bos.bos-ux]  
  > build  
  ▼ src [main] sources root  
    > com.tridiumap.bos.ux  
      ▼ rc  
        JS async.js  
        JS BOS3D.min.js  
        JS Bos3DAutoPatrolWidget.js  
        BOS3DUI.min.css  
        JS BOS3DUI.min.js  
        JS Bos3DWidget.js  
        JS RoamComponentMonitor.js
```

```
var Bos3DWidget = function () {  
  var that = this;  
  Widget.apply(that, arguments);  
  
  that  
    .properties()  
    .add('host3d', 'http://building-bos3d.rickricks.com')  
    .add('hostBos', 'http://building-bos.rickricks.com')  
    .add('modelCode', '')  
    .add('modelKey', '')  
    .add('viewerId', uuid());  
  
  subscriberMixin(that);  
};
```

```
function uuid() {  
  var temp_url = URL.createObjectURL(new Blob());  
  var uuid = temp_url.toString();  
  URL.revokeObjectURL(temp_url);  
  return uuid.substr(uuid.lastIndexOf("/") + 1);  
}
```

Widget

```
Bos3DWidget.prototype.doLoad = function (component){  
    var that = this;  
  
    if (component.getType().toString() != "bos:BosAuthService") {  
        console.error("Binding target is " + component.getType().toString() + " but not bos:BosAuthService. ");  
        return;  
    }  
  
    that.$projectKey = $.parseJSON(component.get('modelDb'))[0];  
    that.$token = component.get('token');  
  
    var url = that.properties().getValue('hostBos') +  
        '/bosfoundationservice/' +  
        component.get('appKey') +  
        '/prototype/query/uoModelDocument?noRelation=true';  
  
    // if user already has the model key (when using BOS platform)  
    // use it directly other than to request it by query  
    if (that.properties().getValue('modelKey') == '') {  
        getModelKey(  
            url,  
            that.$token,  
            that.properties().getValue('modelCode')  
        ).done(function (response) {  
            if ('SUCCESS' == response.code && response.data.count > 0) {  
                that.$modelKey = response.data.data[0].modelKey;  
                createView(that);  
            }  
            else {  
                console.error("Get model response format is invalid or search result is empty: ", response);  
            }  
        });  
    }  
    else {  
        that.$modelKey = that.properties().getValue('modelKey');  
        createView(that);  
    }  
};
```

Widget

```
var createView = function (that) {  
  var option = { host: that.properties().getValue('host3d'), viewport: that.properties().getValue('viewerId') };  
  var viewer3D = new BOS3D.Viewer(option);  
  that.$viewer3D = viewer3D;  
  var bos3dui = new window.BOS3DUI({  
    viewer3D: viewer3D,  
    BOS3D: BOS3D  
  });  
  viewer3D.addView(that.$modelKey, that.$projectKey, that.$token);  
  that.$viewAdded(that);  
}
```

```
Bos3DWidget.prototype.$viewAdded = function (that) {  
}
```



Callback for view added event.

Widget Extension

- Auto Patrol

```
define(['bajaux/Widget',  
    'bajaux/mixin/subscriberMixIn',  
    'Promise',  
    'jquery',  
    'nmodule/bos/rc/BOS3DUI.min',  
    'nmodule/bos/rc/BOS3D.min',  
    'nmodule/bos/rc/Bos3DWidget',  
    'nmodule/bos/rc/RoamComponentMonitor',  
    'css!nmodule/bos/rc/BOS3DUI.min'], function (  
    Widget,  
    subscriberMixIn,  
    Promise,  
    $,  
    BOS3DUI,  
    BOS3D,  
    Bos3DWidget,  
    RoamComponentMonitor) {  
  
    'use strict';  
  
    var Bos3DAutoPatrolWidget = function () {  
        var that = this;  
        Bos3DWidget.call(that);  
        that  
            .properties()  
            .add('path', '')  
            .add('loop', true)  
            .add('animatorDuration', 800)  
            .add('loopNum', -1)  
            .add('timeGap', 1000);  
    };  
};
```

Widget Extension - Auto Patrol

```
Bos3DAutoPatrolWidget.prototype.$viewAdded = function (parent) {  
  
    var that = this;  
  
    var viewer3D = that.$viewer3D;  
    var modelKey = that.$modelKey;  
  
    viewer3D.registerModelEventListener(BOS3D.EVENTS.ON_LOAD_COMPLETE, function () {  
        viewer3D.setModelMatrix(modelKey, new BOS3D.THREE.Matrix4().set(  
            1, 0, 0, 0,  
            0, 1, 0, 0,  
            0, 0, 1, 0,  
            0, 0, 0, 1  
        ));  
        viewer3D.render();  
    })  
  
    var customLine = that.properties().getValue('path');  
    let lineInfoArray = customLine.split('@@');  
    let lineObj = formatLineData(lineInfoArray[0], lineInfoArray[1]);  
    let setting = {    //设置参数  
        defaultAnimatorDuration: that.properties().getValue('animatorDuration'),    //3d默认的动画间隔800ms  
        animatorDuration: that.properties().getValue('animatorDuration'),  
        loopSetting: {  
            loop: that.properties().getValue('loop'),    //是否重复循环  
            loopNum: that.properties().getValue('loopNum'),    //循环次数, 该值权重最大, 默认小于1, 即为0或者-1都行  
            timeGap: that.properties().getValue('timeGap')    //下一次循环间隔  
        }  
    };  
  
    window.customRoam = new RoamComponentMonitor(viewer3D, lineObj.keyFrameList, setting);  
    if(lineObj.keyFrameList.length > 0)  
    {  
        var startPoint = lineObj.keyFrameList[0];  
        customRoam.startRoam(() => {  
            viewer3D.linearFlyTo({  
                position: startPoint.position,  
                target: startPoint.target,  
                up: startPoint.up,  
            }, function () {  
                // console.log('回到正视图')  
            });  
        });  
    }  
}
```

A nighttime photograph of a modern building complex. In the center, a large, dark-roofed pavilion-like structure with a wooden interior is visible. A wide staircase leads up to its entrance. To the left, a multi-story building with balconies is partially visible. The foreground is a paved walkway leading towards the central structure, flanked by lush landscaping including palm trees and various shrubs. The scene is illuminated by warm interior and exterior lights, creating a serene atmosphere.

DEMO

NIAGARA FORUM
CONNECTING
THE WORLD

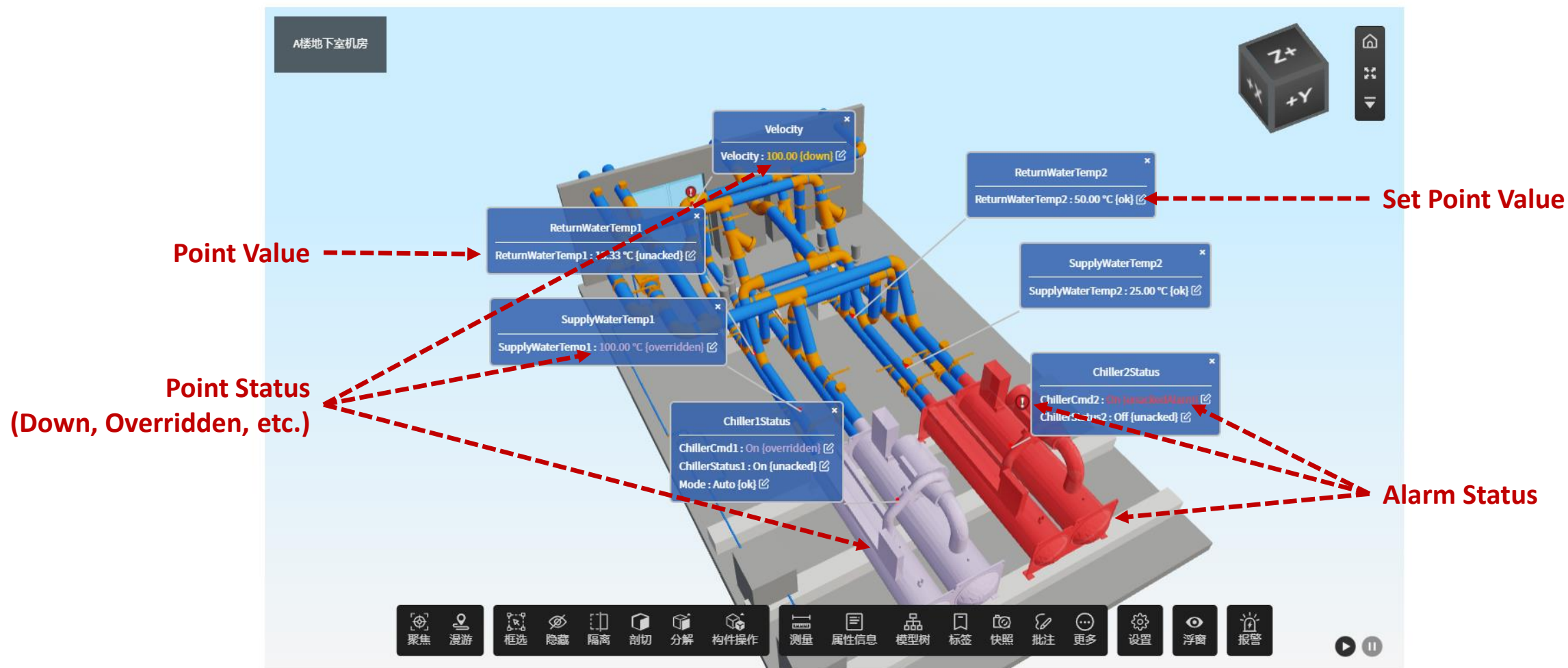
 History

4 TextBlock

6 objects

11 objects

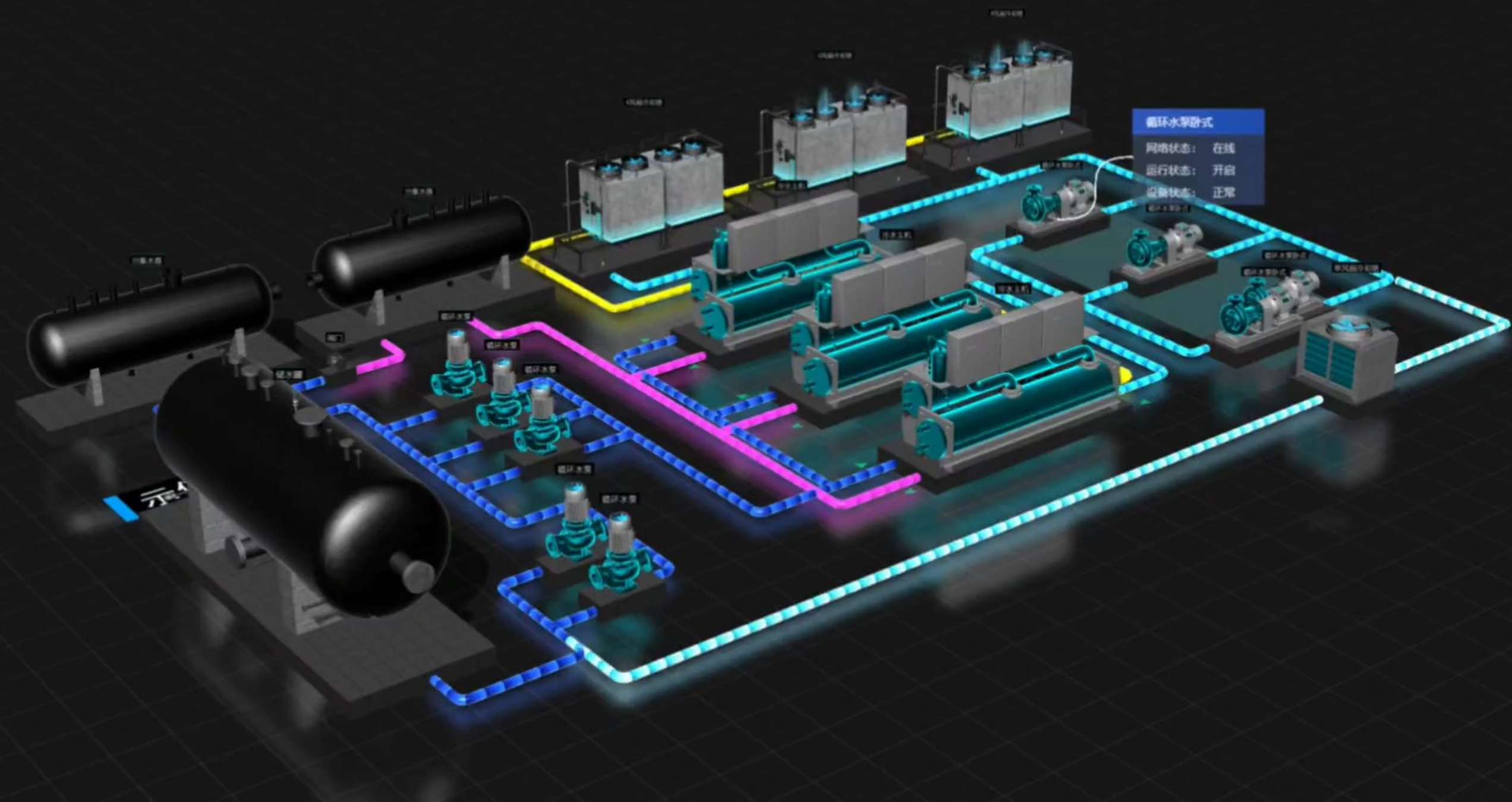
Property	Value
Station Name	bos
Host	/192.168.1.150
Host Model	Workstation
Host Model Version	
Host Product	
Host Id	Win-C0E1-A26B-6B0C-BED8
Niagara Version	4.11.1.16
Java Version	Java HotSpot(TM) 64-Bit Server VM 25.321-b07
OS Version	amd64 Windows 10 10.0
Locale	en_US
Current Time	02-Nov-22 10:06 PM CST





DEMO 2

NIAGARA FORUM
CONNECTING
THE WORLD





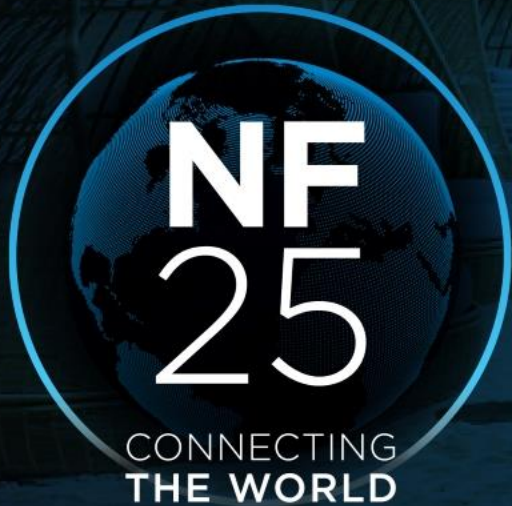
KEY POINTS

NIAGARA FORUM
CONNECTING
THE WORLD

Key Points

- Architecture
- Data Transfer Protocol
- Performance
- Data Binding

THANK YOU



Q & A

NIAGARA FORUM
CONNECTING
THE WORLD