



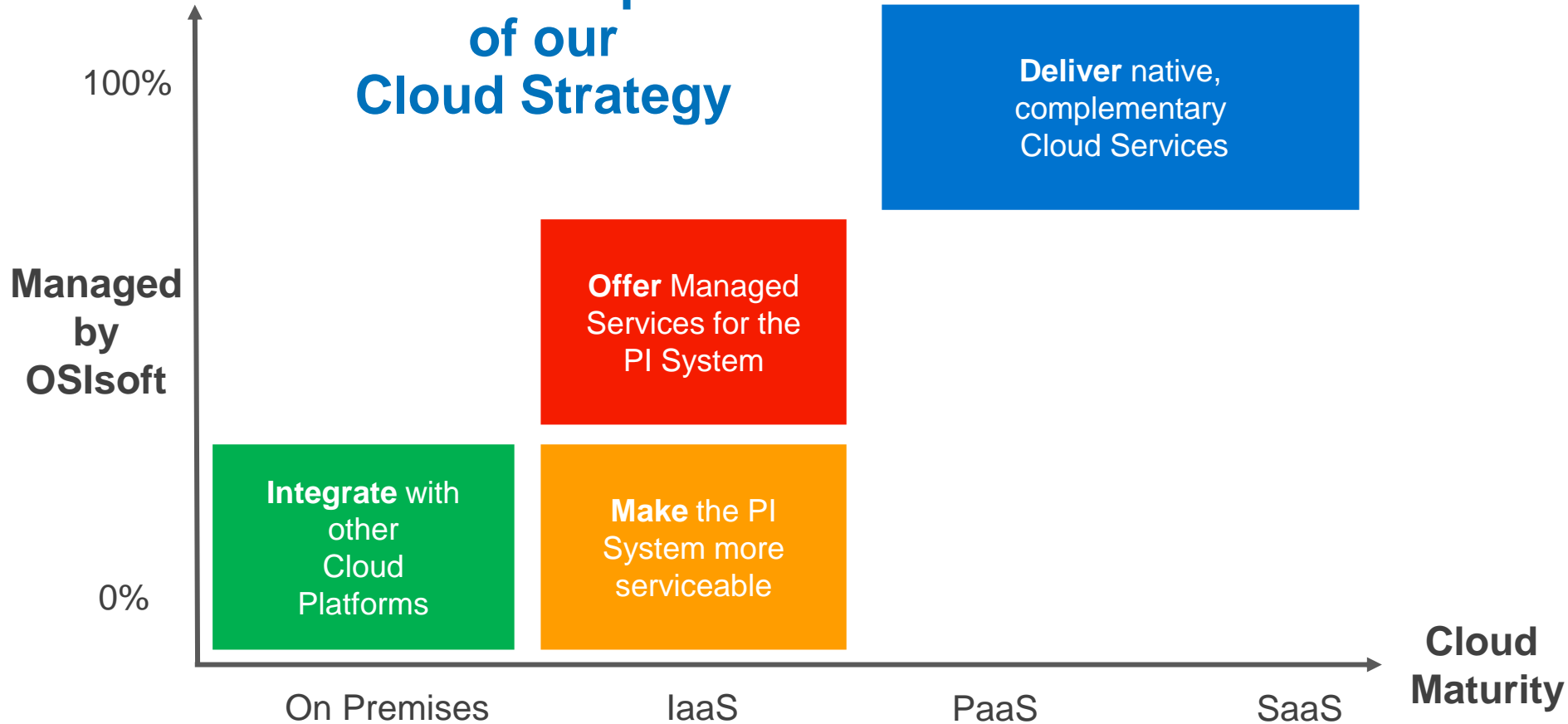
# OSIsoft Cloud Services Core Infrastructure for Developing Partner Applications

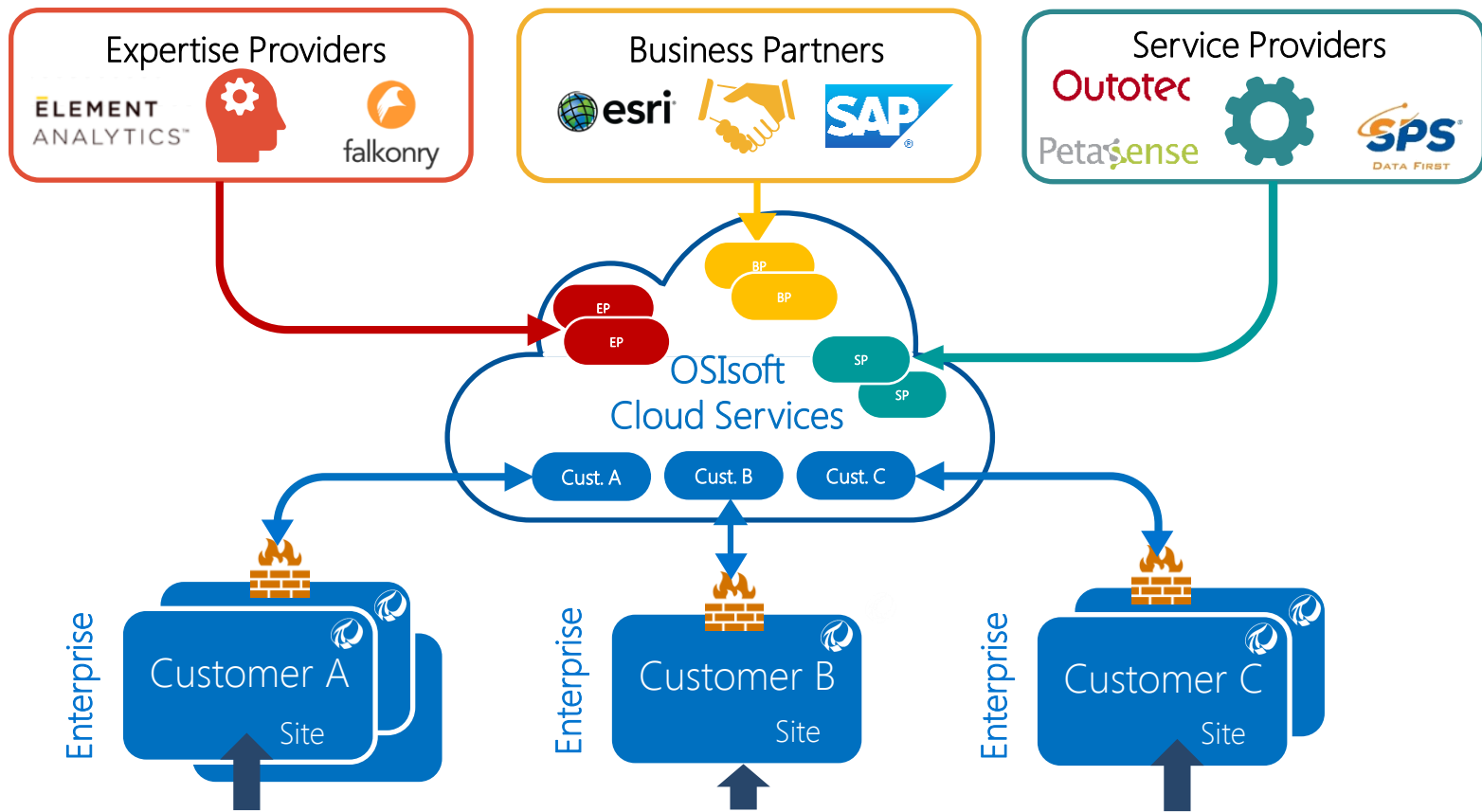
Presented by  
Laurent Garrigues, Gregg Le Blanc, Paul Kaiser

# Agenda

- Overview
- Platform Tour
- Demo
- Partner Preview Program
- Q&A

# The Four Components of our Cloud Strategy





Process Equipment



Control Hardware



IT Hardware



Other Operational Data

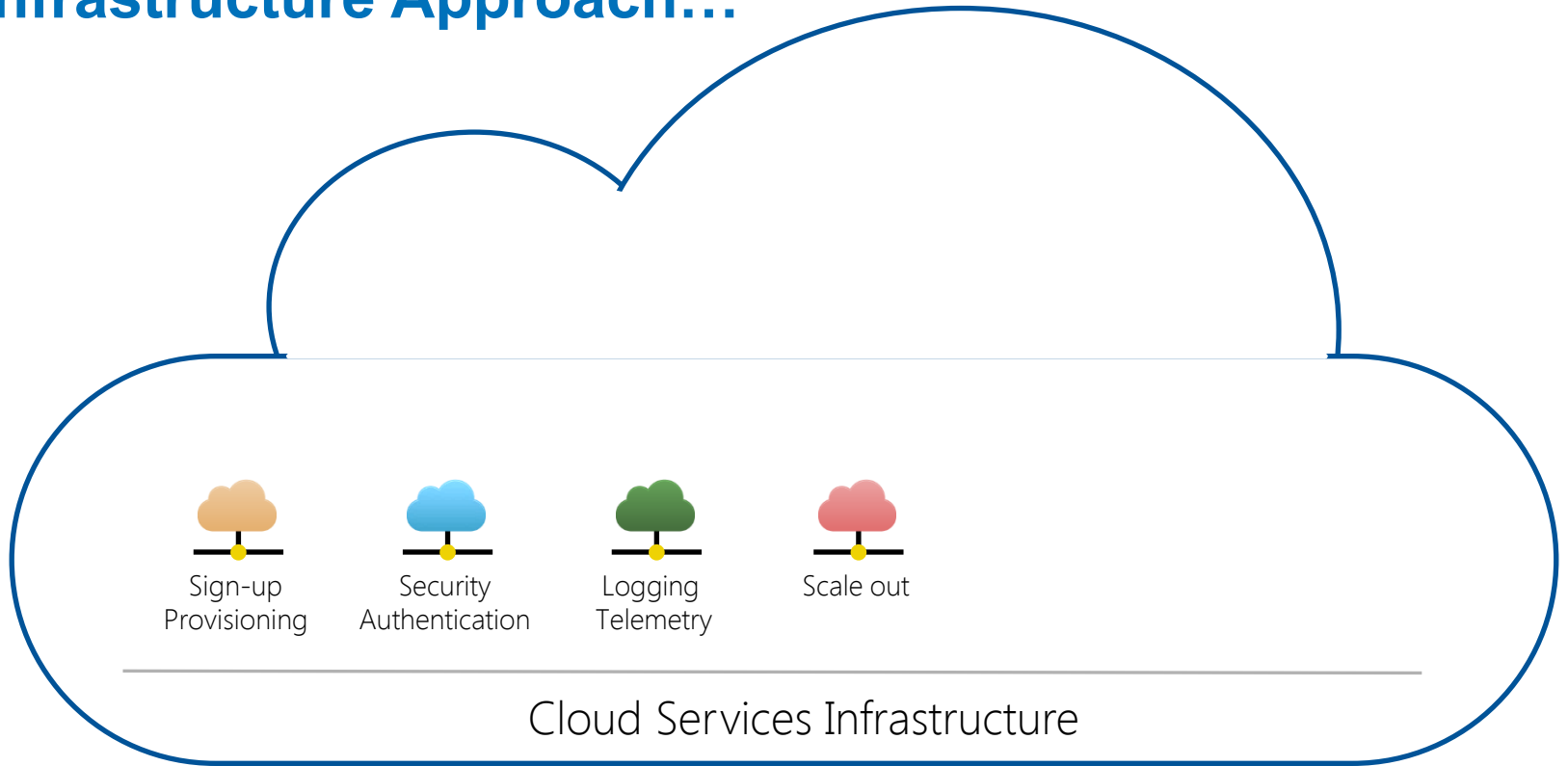
- Facilities
- Logistics

# OSIsoft Cloud Services (OCS) Overview

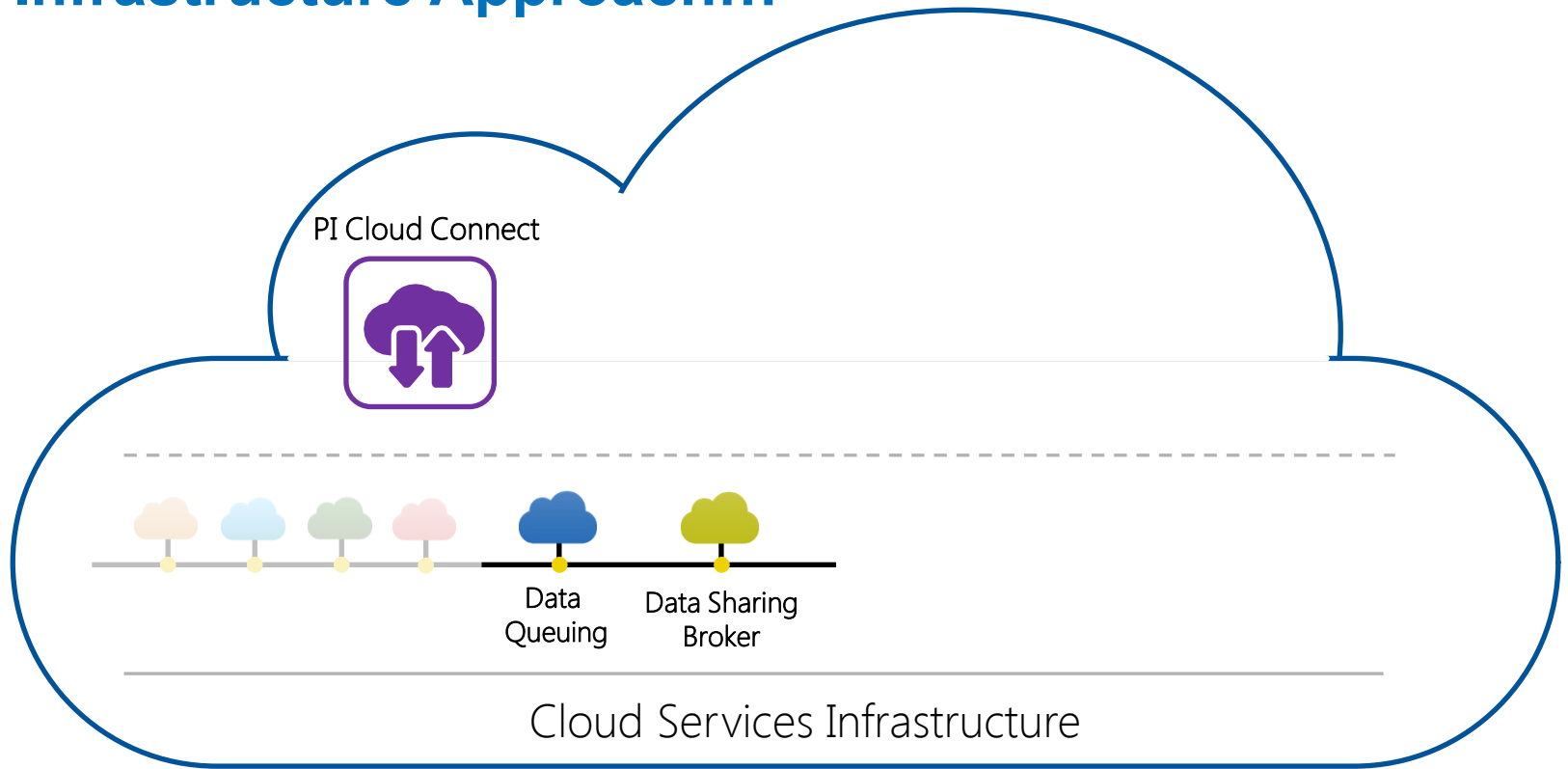
- Managed, secure, data platform
  - Multi-tenant
  - PaaS & SaaS
- Built on Microsoft Azure
- High speed, scalable, data ingress
- Flexible, resilient, data storage
- Modern, secure APIs
- and more...



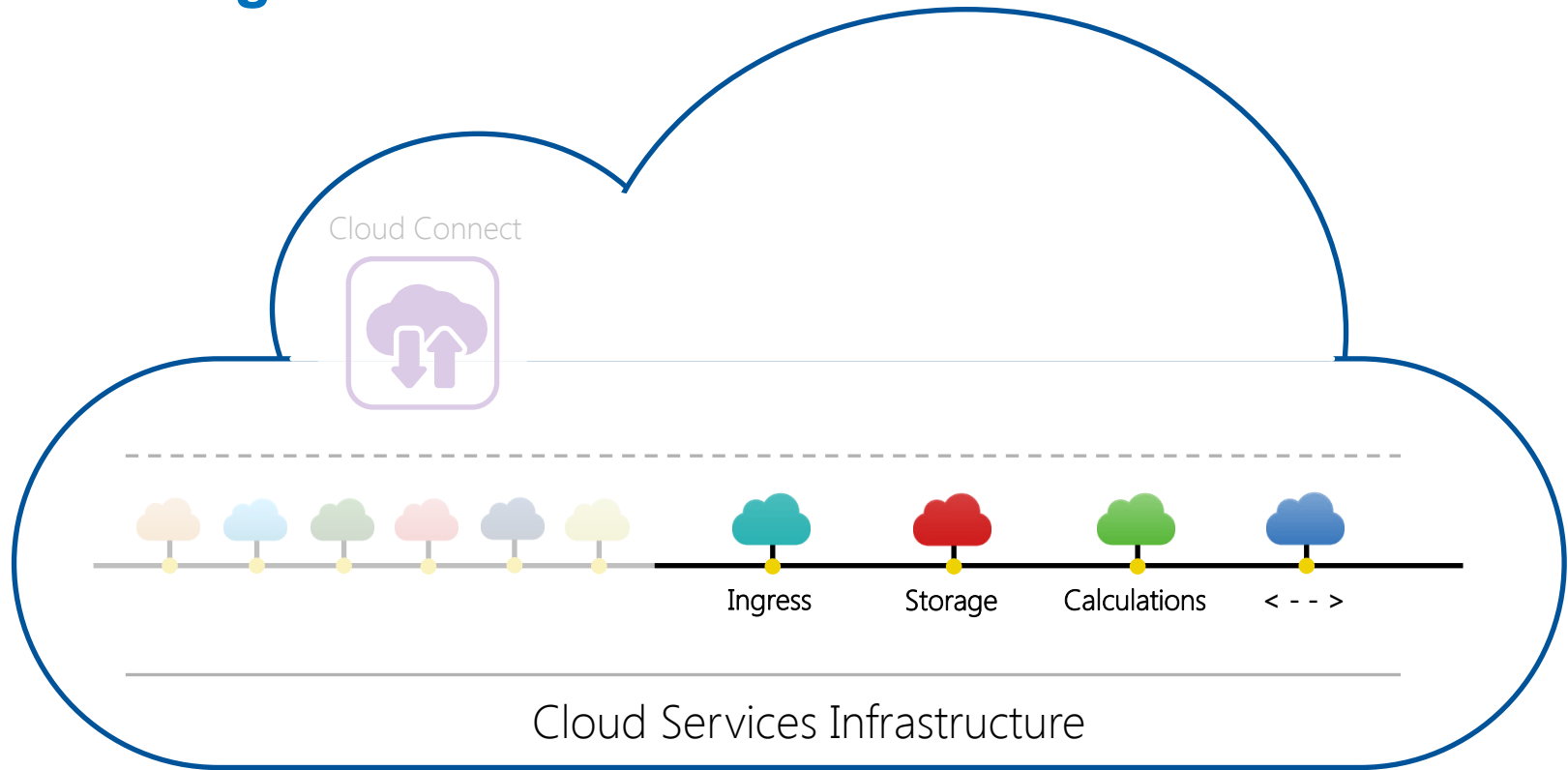
# An Infrastructure Approach...



# An Infrastructure Approach...

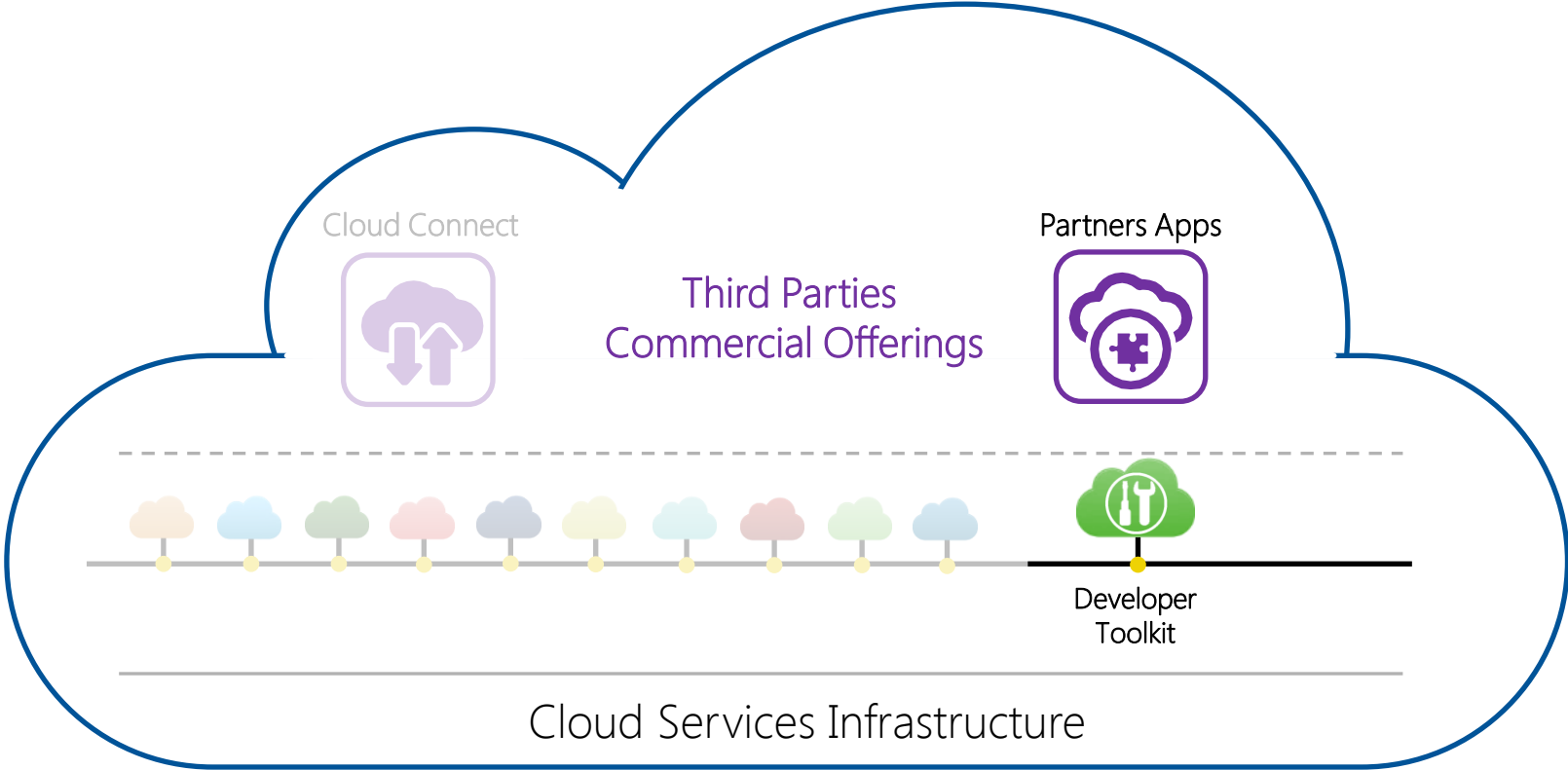


# Extending The Platform...

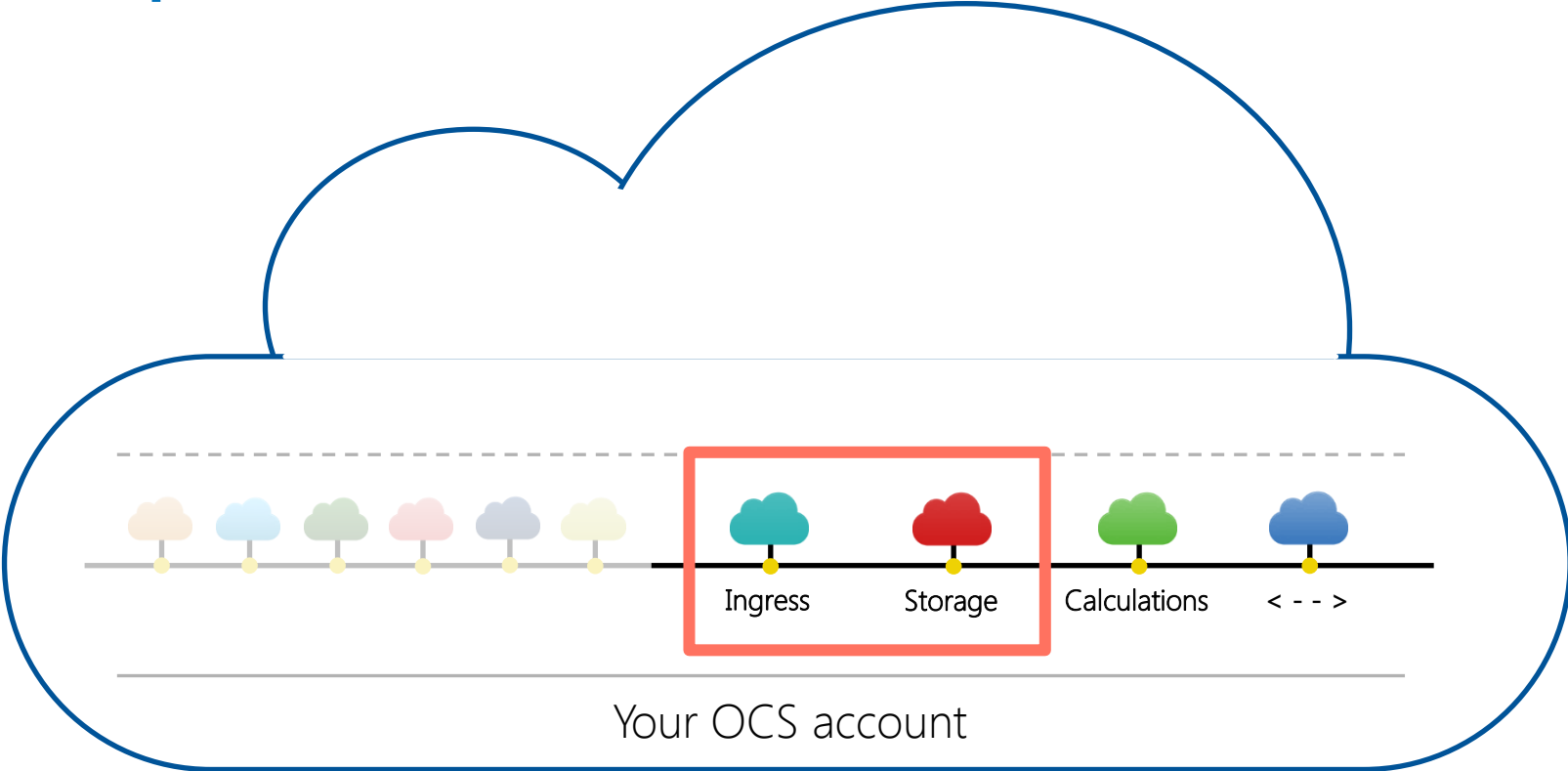




# Enable Partners & App Providers...



# Let's Explore these Services



# Organizing data... Namespaces / Types / Streams / Events

Account: **Annual Analytics**

Namespace: **Drilling Data**

Type: **Simple measurement**

Stream: **Platform 709-G**

Stream: **Platform 432-N**

New event

Namespace: **Meters**

Type

Type

Type

Type

## Ingress Overview

- RESTful ingress
- OSIssoft Message Format
  - OMF v1.0
- High throughput
- Low maintenance
- See OMF Talk later today

## Storage Overview

- Distributed storage layer
  - CRUD
  - “Open Access”
    - Almost any language
- Security based on WAAD
- Redundancy
- High fidelity
- Scalable & elastic

# Ingress & Storage: How to Represent Data?

If you like...

Individual sensors

```
{ "latitude": "29.7817",  
  "timestamp": "2016-04-..." }  
{ "longitude": "-95.6112",  
  "timestamp": "2016-04-..." }  
{ "heading": "42",  
  "timestamp": "2016-04-..." }  
  
{ "heart-rate": "143",  
  "timestamp": "2016-04-..." }  
  
{ "temperature": "30.43",  
  "timestamp": "2016-04-..." }  
{ "pressure": "101.5",  
  "timestamp": "2016-04-..." }
```

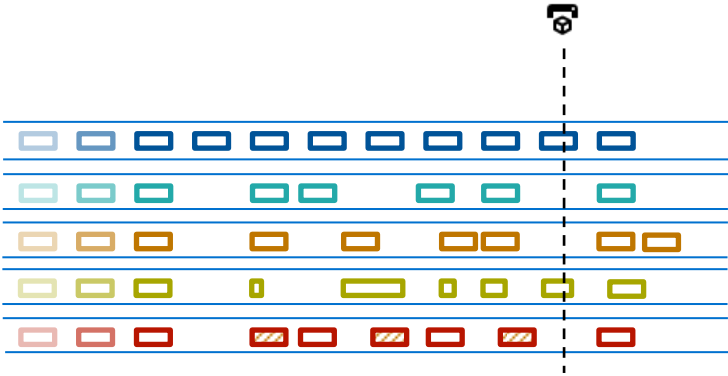


You might also enjoy...

"Fitness band"

```
{ "latitude": "29.7817",  
  "longitude": "-95.6112",  
  "heading": "42",  
  "heart-rate": "143",  
  "temperature": "30.43",  
  "pressure": "101.5",  
  "timestamp": "2016-04-..." }
```

# Data is organized around a primary index



Note that these can be “sparse” events...

# OCS Data Storage is a (Time) Series Database

# Data Structure: Types – Simple Cases

## Type: Simple Double

 Timestamp: DateTime

Value: Double

Quality info: Boolean

## Type: Simple Integer

 Timestamp: DateTime

Value: Integer

Quality info: Boolean



# Data Structure: Types – Complex Cases

## Type: Platform pressure



Timestamp: DateTime

Pressure: Double

Depth: Double

Area code: Integer

Quality info: String

Edited: DateTime?

## Type: Batch measurement



Lot number: Integer

pH: Single

Color: String

Weight: Double

Quality info: String

Tested: DateTime?

# Data Storage: Streams (instances of a Type)

Type: Platform pressure

- Timestamp: DateTime
- Pressure: Double
- Depth: Double
- Area code: Integer
- Quality info: String
- Edited: DateTime?



Stream: Platform 434-E

Stream: Platform 709-G

Stream: Platform 432-N

- Type: Platform pressure
- Name: Platform 432-N
- Description: Fronts...
- Tags: Drilling, EU,...
- Indexes: Depth, Time...

# Data Ingress: You have two choices

## OCS Ingress Service

- OSISoft Message Format (OMF)
  - <http://omf-docs.osisoft.com/>
- Construct messages
  - Type
  - Container
  - Data
- POST to Ingress endpoint

## OCS Storage API

- Documentation:
  - OCS Storage API
  - <https://cloud.osisoft.com/documentation>
- Construct:
  - Type(s)
  - Streams
- POST Data
  - C# Libraries & Sample Code
  - Sample Code for
    - Python
    - JavaScript
      - Node.js & Angular
    - Java

# Data Ingress: OMF Type

- JSON structure
- Sent with headers:
  - messagetype = type
  - action = create
- Has:
  - classification = dynamic

```
{  
  "id": "TankMeasurement",  
  "version": "1.0.0.0",  
  "type": "object",  
  "classification": "dynamic",  
  "properties": {  
    "Time": {  
      "format": "date-time",  
      "type": "string",  
      "isindex": true  
    },  
    "Pressure": {  
      "type": "number",  
      "name": "Tank Pressure",  
      "description": "Tank Pressure in Pa"  
    },  
    "Temperature": {  
      "type": "number",  
      "name": "Tank Temperature",  
      "description": "Tank Temperature in K"  
    }  
  }  
}
```

# Data Ingress: OMF Container

- A “container” for OCS is a “Stream”
- JSON structure
- Sent with headers:
  - messagetype = container
  - action = create

```
[{  
  "id": "Tank1Measurements",  
  "typeid": "TankMeasurement",  
  "typeVersion": "1.0.0.0"  
}, {  
  "id": "Tank2Measurements",  
  "typeid": "TankMeasurement",  
  "typeVersion": "1.0.0.0"  
}]
```

# Data Ingress: OMF Data

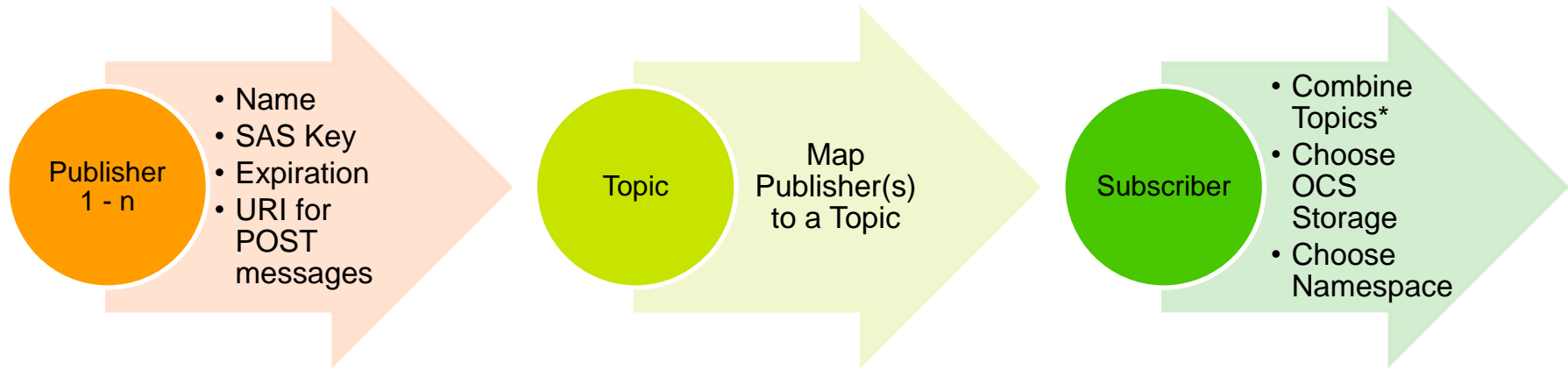
- A “data message” for OCS
- JSON structure
- Sent with headers:
  - messagetype = data
  - action = create

```
{
  "containerid": "Tank1Measurements",
  "values": [{
    "Time": "2017-01-11T22:23:23.430Z",
    "Pressure": "12.0",
    "Temperature": "100.1"
  }, {
    "Time": "2017-01-11T22:24:23.430Z",
    "Pressure": "11.5",
    "Temperature": "101.2"
  }
], {
  "containerid": "Tank2Measurements",
  "values": [{
    "Time": "2017-01-11T22:23:23.430Z",
    "Pressure": "14.0",
    "Temperature": "90.1"
  }, {
    "Time": "2017-01-11T22:24:23.430Z",
    "Pressure": "15.1",
    "Temperature": "91.2"
  }
]
}
```

# Data Ingress: Connecting the dots

- To use OCS Ingress
  - Create a Publisher Token
  - Create a Topic
  - Create a Subscription
    - Choose OCS Storage
- To use the OCS API
  - Create Client ID / Secrets
  - Code
- Visit the OMF Talk later!

# Using OCS Ingress for OMF Messages





# Data Ingress: Streaming data

Stream: Platform 432-N

Type: Platform pressure

Name: Platform 432-N

Description: Fronts...

Tags: Drilling, EU,...

Indexes: Depth, Time...



New event

Stream: Platform 432-N

Timestamp: 24-Oct-2017...

Pressure: 134.0254

Depth: -256.12

Area code: 254

Quality info: Quest...

Edited: null

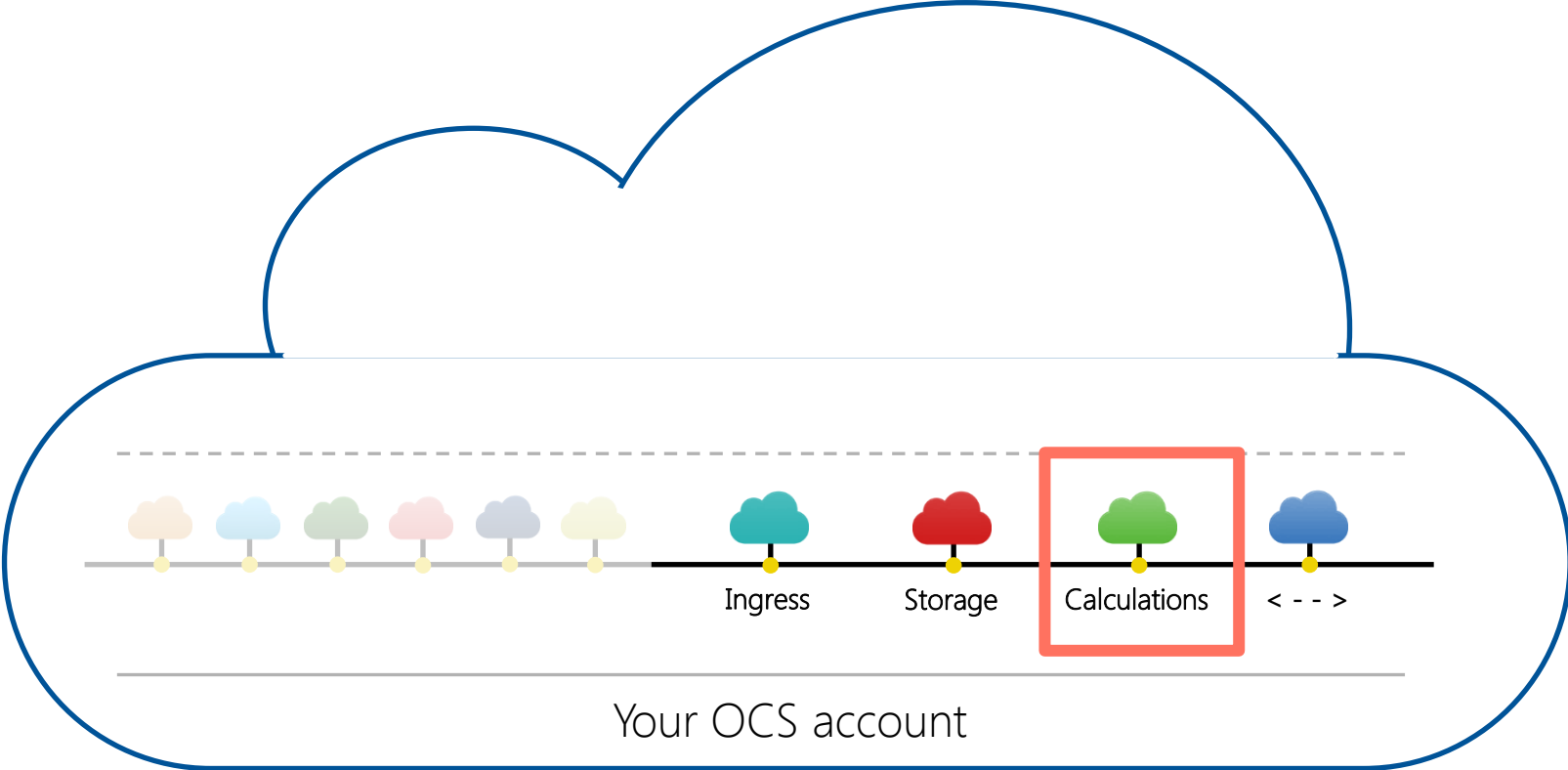
# Data Egress: Example Streaming Data

	Timeld	SummaryKey	Hourly	HourlyDuration	Event	Log	Voltage	DeliveredEnergy	ReceivedEnergy	DeliveredPower	ReceivedPower
1	2016-09-20T00:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":88814,"Next Boundary":88879 }	116.7762	2.4888	2.4882	149.3295	149.2936
2	2016-09-20T01:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":88880,"Next Boundary":88944 }	116.8792	2.7183	2.7107	163.0999	162.6403
3	2016-09-20T02:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":88945,"Next Boundary":89009 }	116.9929	0.7580	0.7571	45.4813	45.4243
4	2016-09-20T03:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":89010,"Next Boundary":89074 }	116.5049	0.6534	0.6545	39.2058	39.2724
5	2016-09-20T04:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":89075,"Next Boundary":89139 }	116.7412	0.7516	0.7478	45.0942	44.8663
6	2016-09-20T05:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": Overpopulated}	{"Average Data":89140,"Next Boundary":89204 }	116.7858	0.6297	0.6333	37.7844	38.0003
7	2016-09-20T06:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": Bad Interval}	{"Average Data":89205,"Next Boundary":89269 }	116.8296	0.6743	0.6751	40.4599	40.5081
8	2016-09-20T07:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":89270,"Next Boundary":89334 }	116.9141	0.9662	0.9666	57.9706	57.9944
9	2016-09-20T08:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":89335,"Next Boundary":89399 }	116.7721	1.2398	1.2378	74.3910	74.2656
10	2016-09-20T09:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":89400,"Next Boundary":89464 }	116.7630	1.0696	1.0722	64.1778	64.3307
11	2016-09-20T10:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":89465,"Next Boundary":89529 }	116.7310	0.9888	0.9872	59.3289	59.2327
12	2016-09-20T11:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":89530,"Next Boundary":89594 }	116.7406	1.1694	1.1745	70.1663	70.4715
13	2016-09-20T12:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":89595,"Next Boundary":89659 }	116.9835	1.0993	1.1041	65.9583	66.2468
14	2016-09-20T13:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":89660,"Next Boundary":89724 }	116.7235	0.6373	0.6371	38.2404	38.2257
15	2016-09-20T14:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":89725,"Next Boundary":89789 }	116.8064	1.0486	1.0507	62.9175	63.0424
16	2016-09-20T15:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":89790,"Next Boundary":89854 }	117.0565	0.8006	0.7992	48.0373	47.9507
17	2016-09-20T16:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": Bad Interval}	{"Average Data":89855,"Next Boundary":89919 }	116.9124	2.2826	2.2688	136.9544	136.1292
18	2016-09-20T17:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":89920,"Next Boundary":89984 }	116.8097	2.5728	2.5703	154.3676	154.2214
19	2016-09-20T18:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": Bad Interval}	{"Average Data":89985,"Next Boundary":90049 }	116.7855	2.3065	2.3068	138.3918	138.4101
20	2016-09-20T19:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":90050,"Next Boundary":90114 }	116.9230	2.3750	2.3734	142.5003	142.4056
21	2016-09-20T20:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":90115,"Next Boundary":90179 }	116.8745	2.3631	2.3631	141.7887	141.7837
22	2016-09-20T21:59:54.3479057+00:00	2	TRUE	00:59:54	{"Current Status": OK}	{"Average Data":0,"Next Boundary":19 }	117.2508	2.5051	2.4873	150.3088	149.2372

# OSIsoft Cloud Services...

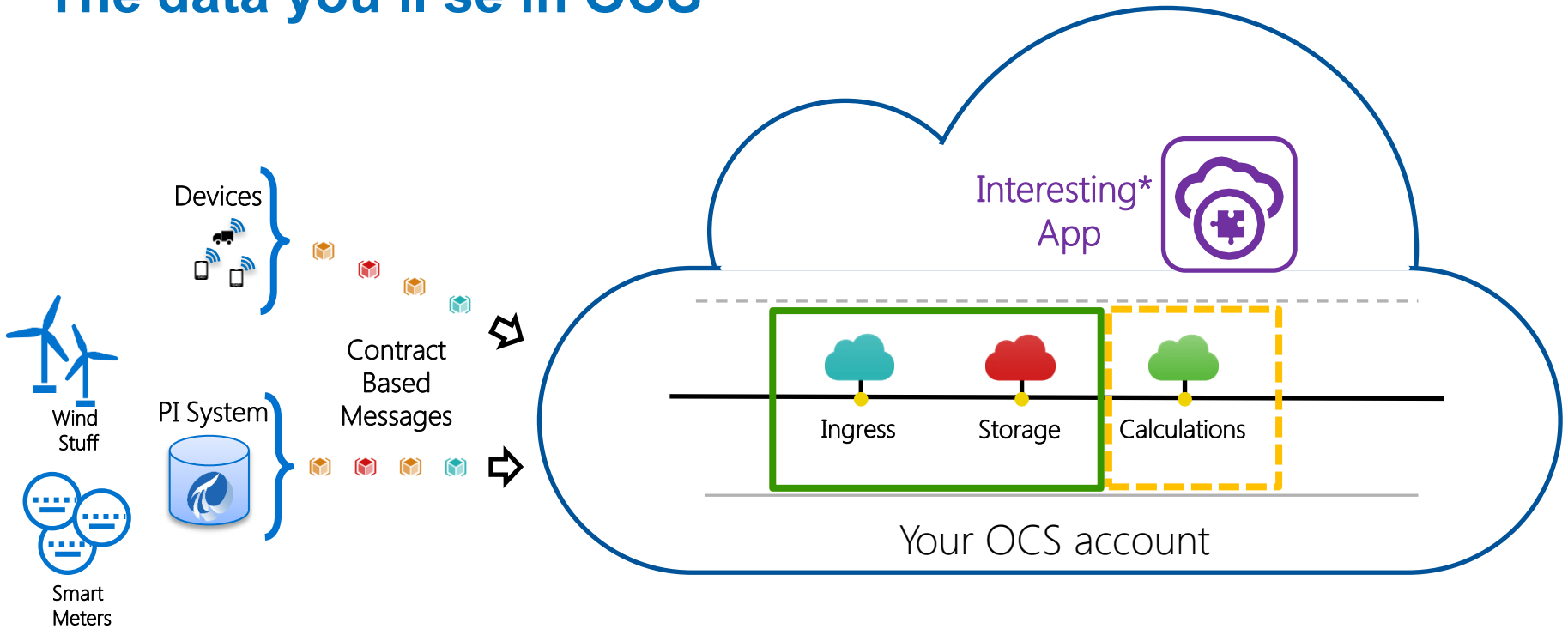
## What else?

# OCS and You... in the fullness of time



# A Lap Around OCS in 10 Minutes

# The data you'll see in OCS



\* App may not be that interesting

# Platform Tour

- <https://cloud.osisoft.com>
- Documentation
- Code samples
- Service status page

>Welcome to OSIsoft Cloud Services

## What is OSIsoft Cloud Services?

### OSIsoft Cloud Services

Industrial digitalization and globalization is generating more data and new opportunities to leverage data. Data that can be shared and used to power new applications that deliver increased operational efficiency and support new business models. At the same time the cloud offers a scalable, elastic and centralized environment to aggregate data for reporting, advanced analytics and third party applications. OSIsoft's Cloud Services are designed to meet today's digital transformation needs by connecting real-time data to people, analytics, and applications, to allow enterprises and communities to realize the value of their data.

For more information, please visit <http://www.osisoft.com/Solutions/OSIsoft-Cloud-Services/>.

## Explore OSIsoft Cloud Services

### Control Your Cloud

The OSIsoft Cloud Services portal is your gateway to learning about Cloud Services features. You will find both detailed API documentation as well as sample code to help you interact programmatically with your data via the Cloud Services API. Additionally, the OSIsoft Cloud Services portal can be used to manage your Cloud Services account.

Use the links below to explore the OSIsoft Cloud Services Portal:

- ▶ [View OSIsoft Cloud Services API Documentation](#)
- ▶ [Explore Code Samples Provided in Multiple Programming Languages](#)
- ▶ [View The OSIsoft Cloud Services Service Blog](#)

You must sign in to access the following features:

- Manage Users And User Access For Your Organization
- Manage Client Keys For Securely Accessing Your Data
- Manage Namespaces to Organize Your Data
- Create And Execute Calculations Against Your Data





Search docs

What's new in Qi?

Introducing Qi

Quick start

Qi Namespace information

Qi Namespace API calls

QiType information

QiType API

QiStream information

QiStream API

QiStreamBehavior information

QiStreamBehavior API

Indexes

Writing data

Write exception handling

Read the Docs

v: latest ▾

Docs » Writing data

## Writing data

The Qi API library includes an assortment of methods that are used to write, replace, and remove data from streams. Methods can be used to operate on one or more data events in a single call. All methods that write or manipulate data values are allowed only by the administrator security accounts.

Reading and writing data with the Qi Client Libraries is performed through the `IQIDataService` interface, which is accessed with the `QiService.GetDataService( )` helper.

### Write exception handling

If a method that acts upon multiple data events encounters a problem while carrying out the operation, an exception is thrown and none of the list of elements is acted upon. For example, when `InsertValuesAsync` is called with a list of events and one of the events uses an index at which data is already present, an exception is thrown and all of the events are rolled back. The event at which the error occurred is identified in the exception and no inserts are performed.

For example:



## Code Samples

## Python

Sample Qi client  
written in Python  
language



## .NET

Sample Qi client  
written in C#



## Java

Sample Qi client  
written in Java



## NodeJs

Sample Qi client  
written in Nodejs



## AngularJs

Sample Qi client  
written in AngularJS



These samples are available in the Qi Samples GitHub repository. The repository can also be accessed directly at <https://github.com/osisoft/Qi-Samples>.



## Service Blog

Welcome to OSIsoft Cloud Services

10/2

*Monday, October 2, 2017 4:23PM***OSIsoft Cloud Services Announcement**

Welcome to OCS. You can service-related announcements on this page. Happy coding!

The **Service Blog** displays bulletins, announcements and reported system-wide issues for the Cloud Historian service. Click on a blog header to view details for that entry.

By default, the last ten entries are displayed. To view additional entries, click **Load More** at the bottom of the list.



The service is operating normally

© 2017 - OSIsoft, LLC.



# Platform Tour

- Logging in
  - You have a WAAD?
  - We have a WAAD for you!
- Users / Roles
- Application keys



chad@annualanalytics.onmicrosoft.com



## Enter password

Back

Sign in

Keep me signed in

[Forgot my password](#)



## Manage Users

+ Add

First Name	Last Name	Email	Sign-In		
Gregg	Le Blanc	gleblanc@osisof...	gregg@AnnualA...		
Chad	Member	chad@osisoft.co...	cmember@Ann...		
Chad	Chisholm	cchisholm@osis...	Chad@AnnualA...		



To add a new user to your directory, click **Add** and fill out the provided **Add User** dialog. The **Sign In** name you enter will be used by the user to sign in to the OSIsoft Cloud Services (OCS) Portal. Once finished, click **Add**. The new user will receive an email at their **Contact Email** address with a temporary password allowing them to log into the OCS Portal.

To view or edit user information, click the **Edit User** button for that user in the table below. To remove a user, click the **Remove User** button. Note that you cannot remove the currently logged-in user.



## Manage Users

First Name	Last Name	Email
Gregg	Le Blanc	gleblanc@osisoft.com
Chad	Member	chad@osisoft.com
Chad	Chisholm	cchisholm@osisoft.com

## Edit User

The **Contact Email** is used to send notifications and password reset information to the user. Assigning a feature **Administrator** role will allow the user to add, edit and remove data for that feature. Click **Save** to save changes to the user, or click **Cancel** to cancel changes to the user.

First Name: Last Name: Sign In: Contact Email: 

Roles:



Tenant Administrator



Tenant Contributor



Tenant Data Steward



Save

Cancel

## API Client Keys

User Role Keys

Administrator Role Keys

Name	Date Created	Value	
chad's key	9/26/2017	*****	 

In order to connect to a cloud historian and access data, your application must authenticate using standard OAuth2 protocol. Below you can generate Client Keys (Client Identity/Client Secret pairs) for use in authentication. Each Client Key can be generated as either a **User** or **Administrator** role. Client Keys with the Administrator role have full read/write access to your data. Client Keys with the User role can only read data. Note that there is currently a limit of 25 Client Keys per each role.

### Explanation and Usage of Client Keys

Security and Authentication in OSIsoft Cloud Services involve the use of Windows Azure Active Directory (WAAD) and WAAD uses a technology called OAuth2. OAuth2 prescribes several different techniques (or "flows") dependent on the nature of the Client and Server applications that are exchanging secured information.

The "Client Credentials Flow" is used when the connection between client and server needs to be long term and when the identity of the user is not important. It is used between rich clients and servers and between servers and servers. The client credentials flow uses a Client ID and a Client Key.





## API Client Keys

Enter a Name...

User Role Keys

Administrator Role Keys

Name	Date Created
------	--------------

chad's key	9/26/2017
------------	-----------

for you to connect to the cloud historian.

## Account Identity

8c6e5ba7-██████████fed██████████d4e

## Client Identity

4f68██████████95fc-██████████71

## Client Secret

██████████v61/hnqHq██████████Xahk=

## Code Usage Examples

.NET (C#)

Node.js

Python

Java

```
import requests, httplib
def mod():
    response = requests.post('https://login.windows.net/8c6e5ba7-0328-4943-8e
    data = { 'grant_type' : 'client_credentials',
            'client_id' : '4f68██████████95fc-██████████1',
            'client_secret' : '██████████v61/hnqHq██████████Xahk=',
            'resource' : 'https://qi-data.osisoft.com:443'
          })
    conn = httplib.HTTPSConnection('https://qi-data.osisoft.com:443')
    conn.request('GET', '/api/tenants/8c6e5ba7-██████████fed██████████4e/na
    headers = {'Authorization' : 'bearer %s' % response.json()['access_to
              'Content-type': 'application/json',
              'Accept': 'text/plain'}}
```

historian and access  
authenticate using  
how you can generate  
ent Secret pairs) for use in  
(can be generated as  
role. Client Keys with  
read/write access to  
User role can only read  
a limit of 25 Client

## of Client Keys

OSIsoft Cloud Services  
ure Active Directory  
nology called OAuth2.  
erent techniques (or  
ure of the Client and  
changing secured

used when the  
server needs to be long  
the user is not  
which clients and servers  
ers. The client credentials  
nt Key.

# Platform Tour

- Data storage
- Namespace
- Types
- Streams

## &gt; Manage Namespaces

Id	Description	Tier Id	Storage	Calcul
Meters		Stan...	100	100
Rese...	Chad...	Stan...	100	100
Wind...		Stan...	100	100

## Add Namespace

The **Namespace Id** must be unique across your account. The **Tier Id** can be selected from the list of options provided and the **Throughput Units**, **Storage Units** and **Calculation Units** will be displayed for the selected **Tier Id**. Once finished, click **Add**. All account administrators will receive an email confirming the creation of the new namespace.

Account Id: 8c6e5ba7-0328-4943-8e0f-fed182dddd4e

Namespace Id: Description: Tier Id: 

Throughput Units: 100

Storage Units: 100

Calculation Units: 100

Add

Cancel

# Platform Tour

- Data Ingress
- Publications
- Topics
- Subscriptions
- OMF

# Platform Tour

- API Console



URI GET /users-e11dd62e2f81/roles?skip=0&count=10 History ▾

Full Path /api/Tenants/bdfe4b39af88/users/1060d388-89b7-49c2-b0c0-e11dd62e2f81/roles?skip=0&count=10

Send

URI Path GET /api/Tenants/bdfe4b39af88/users/1060d388-89b7-49c2-b0c0-e11dd62e2f81/roles?skip=0&count=10

Status Code: 200 Text: OK

```
Body [
  {
    "Id": "0e3b314b-ed9a-42d4-ae47-0255999887a6",
    "Name": "Tenant Administrator",
    "Description": "Tenant Administrator",
    "RoleScope": 1,
    "TenantId": "b699ae77-0b01-4053-acba-5dfe4b39af88",
    "CommunityId": null,
    "RoleId": "2DC742AB-39EA-4FC0-A39E-2BCB71C26A5F"
  },
  {
    "Id": "5e610e3d-af88-4116-b785-bd72cc343f98",
    "Name": "Tenant Member",
    "Description": "Tenant Member role",
    "RoleScope": 1,
  }
]
```

## &gt; API Console

URI GET mespaces/Meters/Streams/Substation1.Feeder1.Transformer1.6014.Muni Meter/Data/GetFirstValue History ▾

Full Path /api/Tenants/ jf-fed182dddd4e/Namespaces/Meters/Streams/Substation1.Feeder1.Transformer1.6014.Muni Meter/L

Send

URI Path GET /api/Tenants/8c6e5ba7-0328-4943-8e0f-  
/Namespaces/Meters/Streams/Substation1.Feeder1.Transformer1.6014.Muni Meter/Data/GetFirstValue

Status Code: 200 Text: OK

Body {  
    "Time": "2016-10-12T13:27:28Z",  
    "Active": "Inactive",  
    "Daily kWh": 4.876304,  
    "Latitude": 37.72561,  
    "Longitude": -122.17038,  
    "Unit": "kWh",  
    "Value": 4.876304

Copy



# Demo

- GetWindowValues
- Trend vs. Trend
- Dashboard

# Checking out your data

> API Console

URI GET /Namespaces/Meters/Streams/Substation1.Feeder1.Transformer1

History

Params

startIndex 2017-10-14T02:07:14Z

endIndex 2017-10-14T15:07:14Z

boundaryType

filter

URI Path GET /api/Tenants/8c  
fed182ddd4e/Namespaces/Meters/Streams/Substation1.Feeder1.Transformer1.6017.Muni  
Meter/Data/GetWindowValues?startIndex=2017-10-14T02:07:14Z&endIndex=2017-10-  
14T15:07:14Z&selectExpression='Time,Daily kWh,Vrms.Value'

Status Code: 200 Text: OK

Body

[

{

```
"Time": "2017-10-14T02:07:54Z",  
"Active": "Active",  
"Daily kWh": 4.71148968,  
"Latitude": 37.71995,  
"Longitude": -122.174049,  
"Net.kWh": 0.202360332,  
"Offpeak Daily kWh": 1.99063492,  
"Peak Daily kWh": 1.174082,  
"Shoulder Daily kWh": 1.54677284
```

Copy

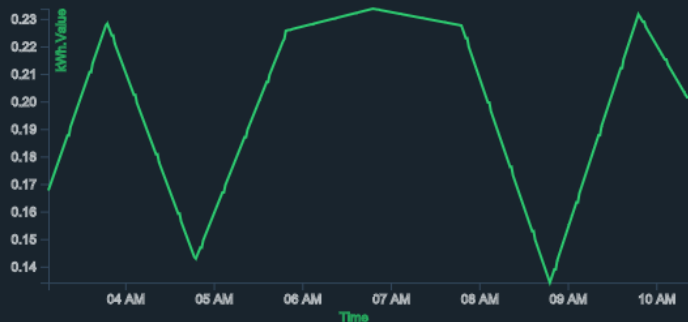
Trend

Substation1.Feeder1.Transformer1.6017.Muni Meter

To display a trend for the response values, select the properties within the response that you wish to use for the X and Y axis of the trend. A blank trend will be displayed if the selected properties contain no distinct values.

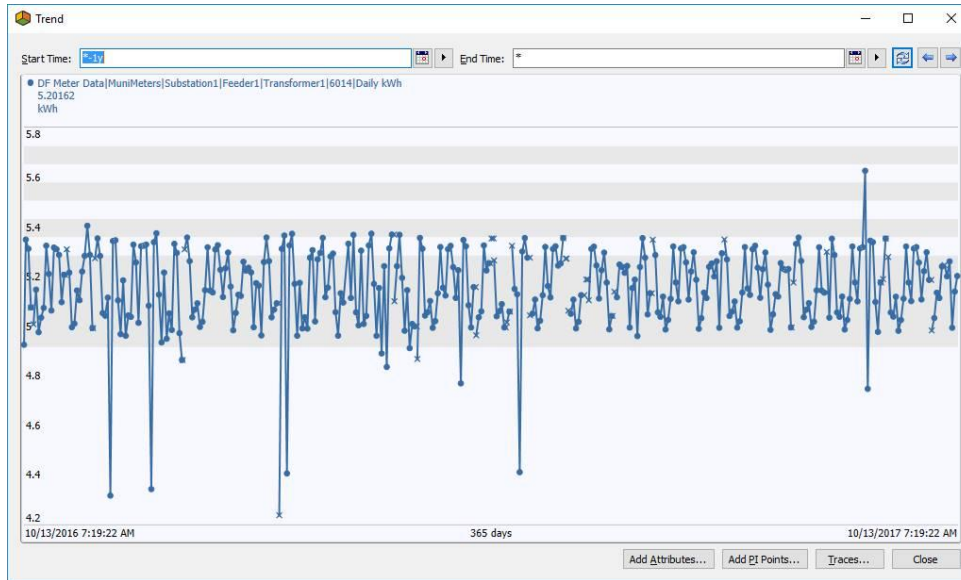
X Axis: Time

Y Axis: kWh.Value



Close

# Trend 1 and Trend 2



OSISOFT Cloud Services

Secure | https://cloud.osisoft.com/apiconsole

Home Learn Manage

Derek Endres  
Annual Analytics

API Console

### Substation1.Feeder1.Transformer1.6014.Muni Meter

To display a trend for the response values, select the properties within the response that you wish to use for the X and Y axis of the trend. A blank trend will be displayed if the selected properties contain no distinct values.

X Axis: Time

Y Axis: Daily kWh

5.5  
5.4  
5.3  
5.2  
5.1  
5.0  
4.9  
4.8  
4.7  
4.6  
4.5  
4.4

Daily kWh

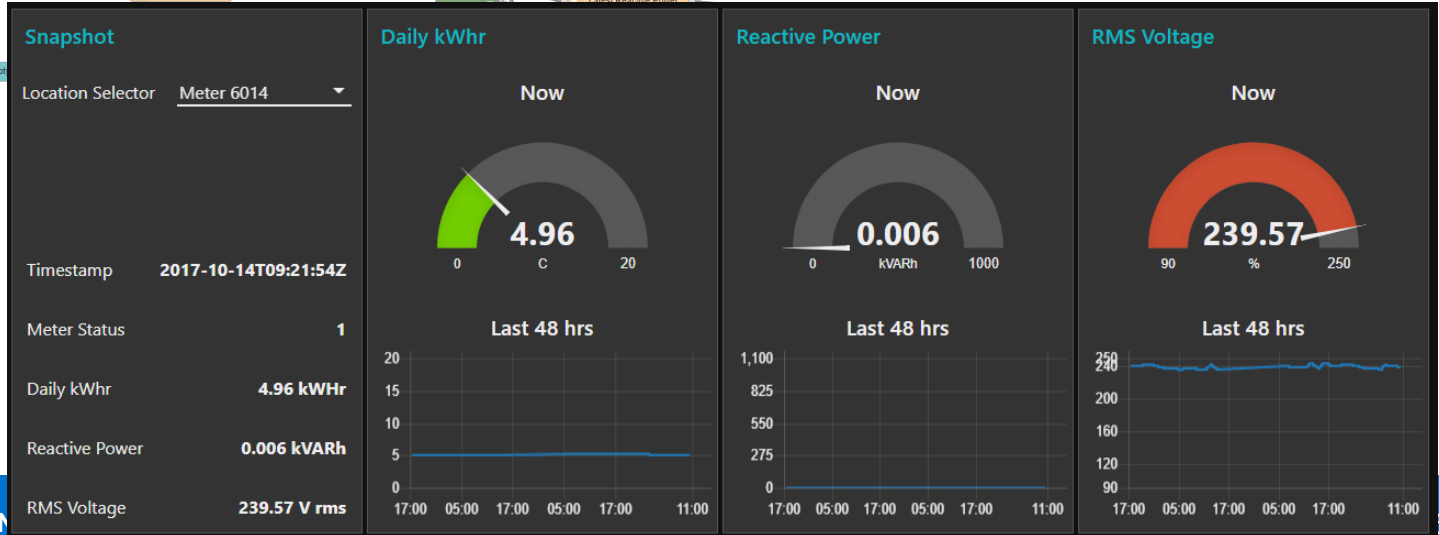
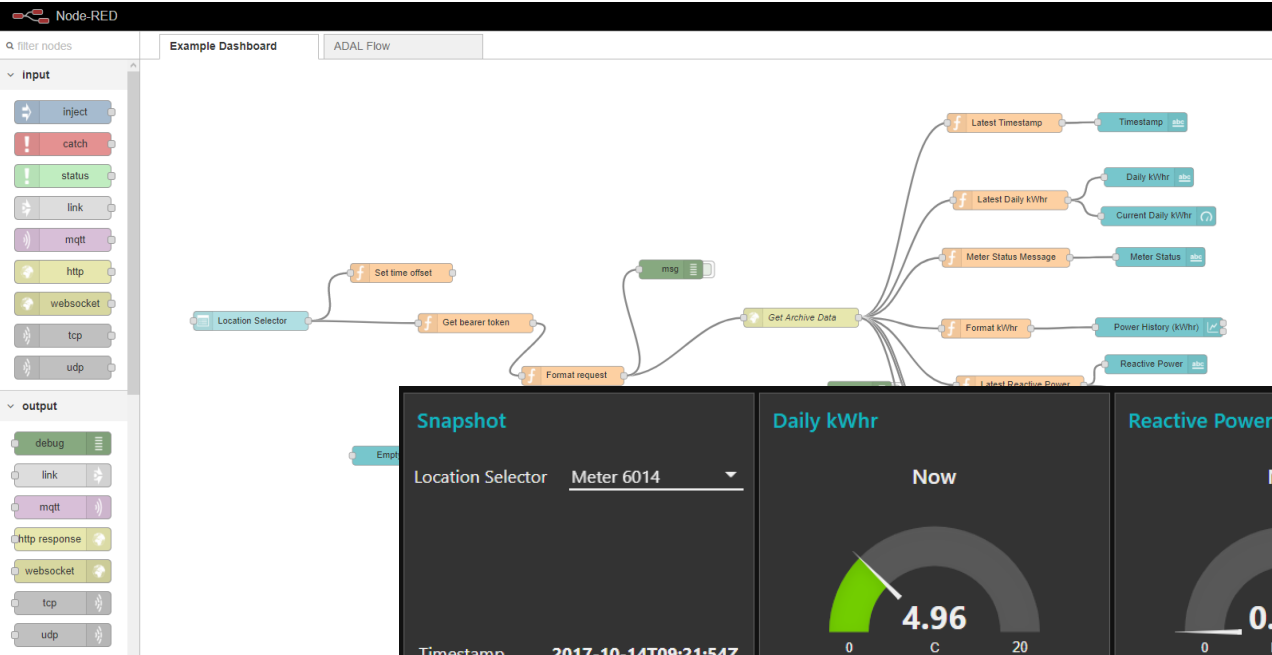
November December 2017 February March April May June July August September October

Time

URI Path GET /api/...

The service is operating normally © 2017 - OSISOFT, LLC

# Extra Credit: Dashboards and data in Node-RED



# Series Indices – Getting the data you want

API Console

URI GET ms/Well2016-A\_\_Wellbore2016-A\_\_measureddepth/Data/GetRang History ▾

Params ▲

startIndex	1800
count	100
reversed	
skip	
boundaryType	
filter	

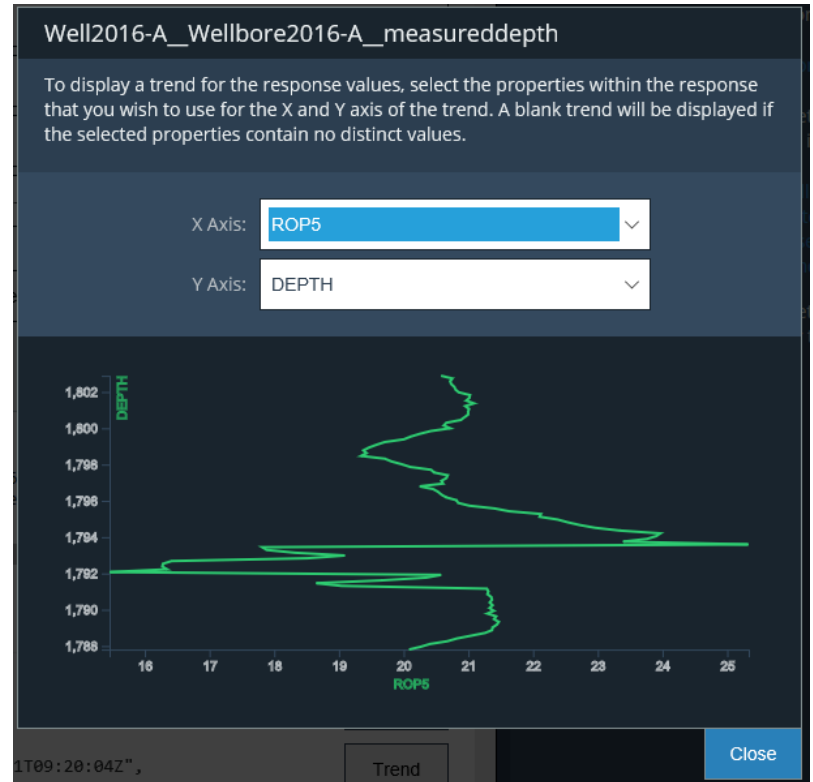
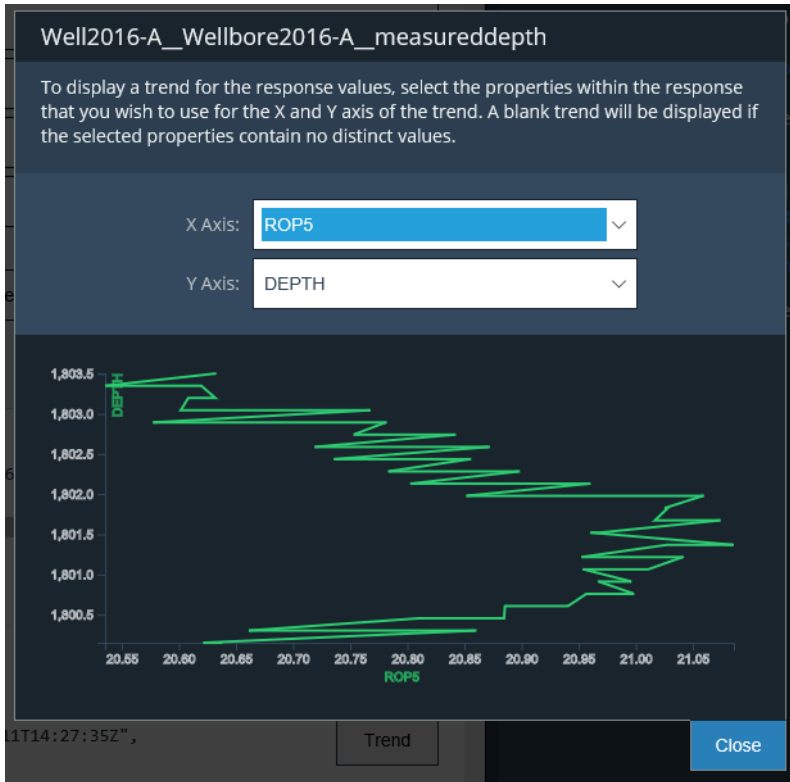
API Console

URI GET ms/Well2016-A\_\_Wellbore2016-A\_\_measureddepth/Data/GetRang History ▾

Params ▲

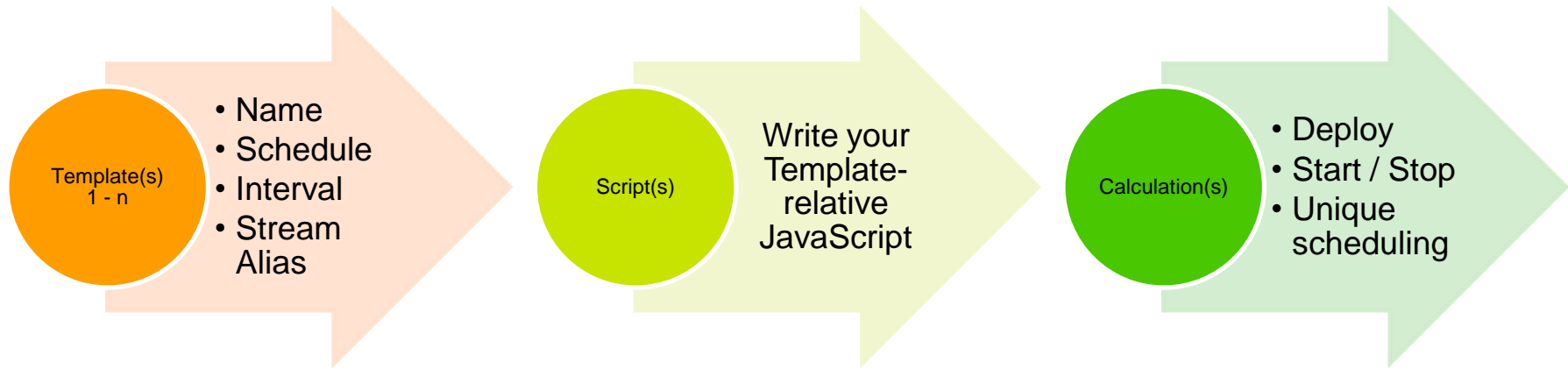
startIndex	1803 2016-03-01T09:20:04Z
count	100
reversed	true
skip	
boundaryType	
filter	CreationDate eq 2016-03-01T09:20:04Z

# How the data type and query changes things



# Did we say Calculations?

# Using Calculations per Namespace in OCS





# Template

Calculation Templates Properties Scripts Calculations Logs Save All

Meters

Filter Templates + Add

- x MuniMeterPickupAverage

Template Attributes

Template Id	e0537dcd-8e5a-6440-3f16-43a837d11380
Name	MuniMeterPickupAverage
Description	A calculated average for Muni Meters

## Default Stream Aliases + Add

- x meter Substation1.Feeder1.Transformer1.6014.Muni Meter
- x meter2 
  - Phase Meter
  - Z Phase.TLM\_0102213301.SDP\_6001208141.Single
  - Phase Meter
  - Z Phase.TLM\_0102221102.SDP\_6001220258.Single

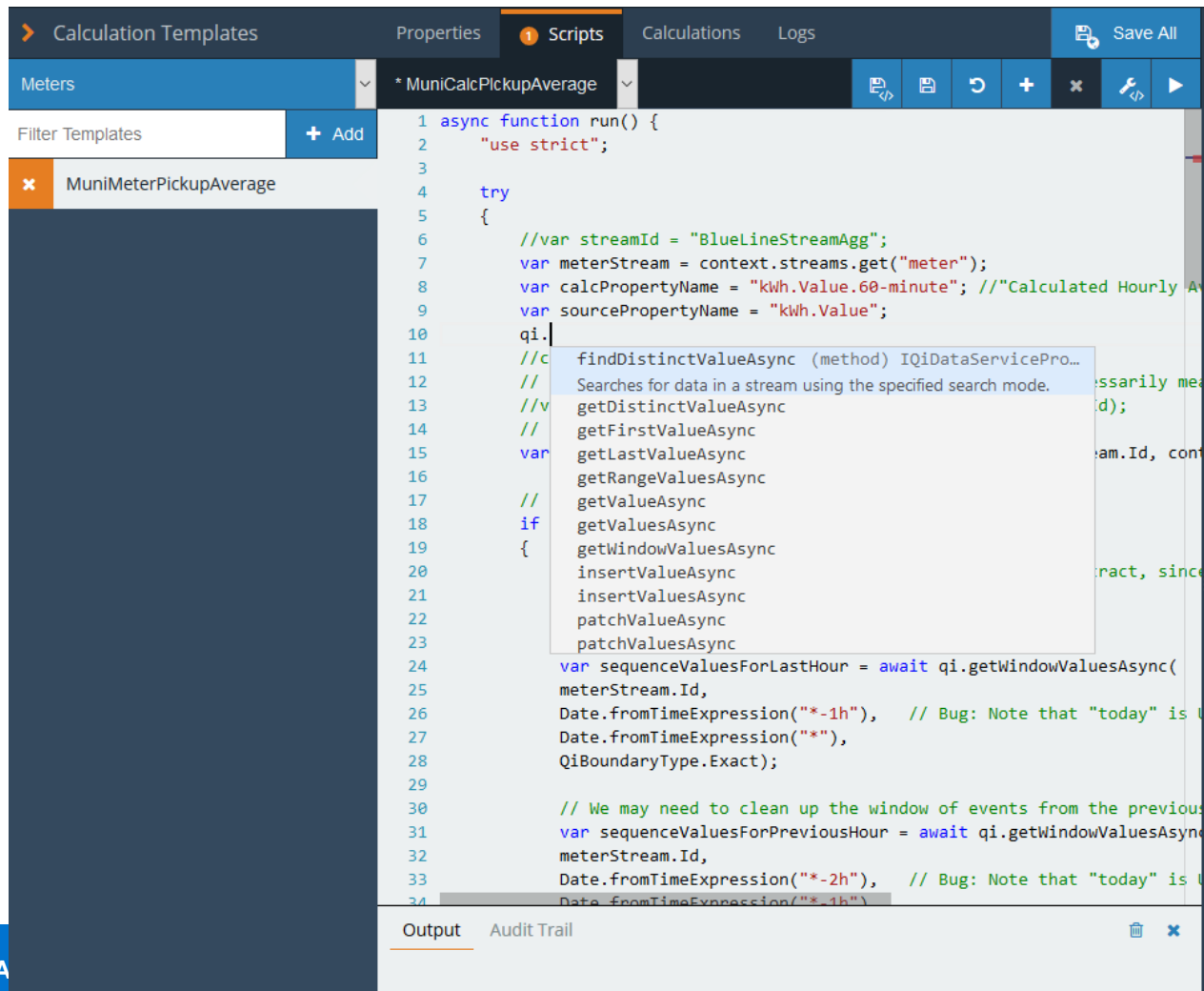
## Default Schedule

Schedule Type	Periodic
Time Zone	(UTC) Coordinated Universal Time
Frequency	TimeInterval
Interval	01 Hours 00 Minutes 00 Seconds
Offset	00 Hours 00 Minutes 00 Seconds

## Default Stream Aliases + Add

- x meter Substation1.Feeder1.Transformer1.6014.Muni Meter

# Code



The screenshot shows a code editor with the following elements:

- Top Bar:** Calculation Templates, Properties, Scripts (active), Calculations, Logs, Save All.
- Left Panel:** Meters (dropdown), Filter Templates (+ Add), MuniMeterPickupAverage (selected).
- Main Editor:** A JavaScript function `run()` with the following code:

```
1 async function run() {
2   "use strict";
3
4   try
5   {
6     //var streamId = "BlueLineStreamAgg";
7     var meterStream = context.streams.get("meter");
8     var calcPropertyName = "kWh.Value.60-minute"; //"Calculated Hourly Average"
9     var sourcePropertyName = "kWh.Value";
10    qi.
11    //c findDistinctValueAsync (method) IQiDataServicePro...
12    // Searches for data in a stream using the specified search mode.
13    //v getDistinctValueAsync
14    // getFirstValueAsync
15    var getLastValueAsync
16    // getRangeValuesAsync
17    // getValueAsync
18    if getValuesAsync
19    { getWindowValuesAsync
20    insertValueAsync
21    insertValuesAsync
22    patchValueAsync
23    patchValuesAsync
24    var sequenceValuesForLastHour = await qi.getWindowValuesAsync(
25      meterStream.Id,
26      Date.fromTimeExpression("*-1h"), // Bug: Note that "today" is T
27      Date.fromTimeExpression("*"),
28      QiBoundaryType.Exact);
29
30    // We may need to clean up the window of events from the previous
31    var sequenceValuesForPreviousHour = await qi.getWindowValuesAsync
32      meterStream.Id,
33      Date.fromTimeExpression("*-2h"), // Bug: Note that "today" is T
34      Date.fromTimeExpression("*-1h")
```
- Bottom Panel:** Output, Audit Trail.

# Calculation Instantiation

The screenshot displays the OSIsoft Cloud Services interface for configuring a calculation. The top navigation bar includes 'Home', 'Learn', and 'Manage' icons. The main menu shows 'Calculation Templates', 'Properties', 'Scripts', 'Calculations', and 'Logs'. The current view is 'Calculations', showing a list of meters with a search filter and an 'Add' button. A specific calculation, 'MuniMeterPickupAverage', is selected, and its configuration is shown on the right. The configuration includes a 'Calculation Attributes' section with fields for Calculation Id, Name, Description, and Status. Below this is the 'Calculation Schedule' section with a checked checkbox for 'Reference Template Default Schedule' and fields for Schedule Type, Time Zone, Frequency, Interval, and Offset. At the bottom is the 'Calculation Stream Aliases' section with a field for 'meter'. A confirmation dialog is overlaid on the left, asking 'Start Calculation?' and 'Do you want to start Calculate Muni Meter Pickup Average?' with 'Start' and 'Cancel' buttons.

**OSIsoft Cloud Services**

Home Learn Manage

Calculation Templates Properties Scripts **Calculations** Logs Save All

Meters Calculate Muni Meter Pickup Average

Filter Templates + Add

**Calculations**

MuniMeterPickupAverage

**Calculation Attributes**

Calculation Id	d109f5fa-f04c-4f9a-5844-c8072059aec1
Name	Calculate Muni Meter Pickup Average
Description	Do the calc
Status	● Running

**Calculation Schedule**  Reference Template Default Schedule

Schedule Type	Periodic
Time Zone	(UTC) Coordinated Universal Time
Frequency	TimeInterval
Interval	01 Hours 00 Minutes 00 Seconds
Offset	00 Hours 00 Minutes 00 Seconds

**Calculation Stream Aliases**

meter	Substation1.Feeder1.Transformer1.6014.Muni Meter
-------	--

Status **Stopped**

Calculation Schedule

Schedule Type

Time Zone (UTC) Coordinated Universal Time

**Start Calculation?**

Do you want to start Calculate Muni Meter Pickup Average?

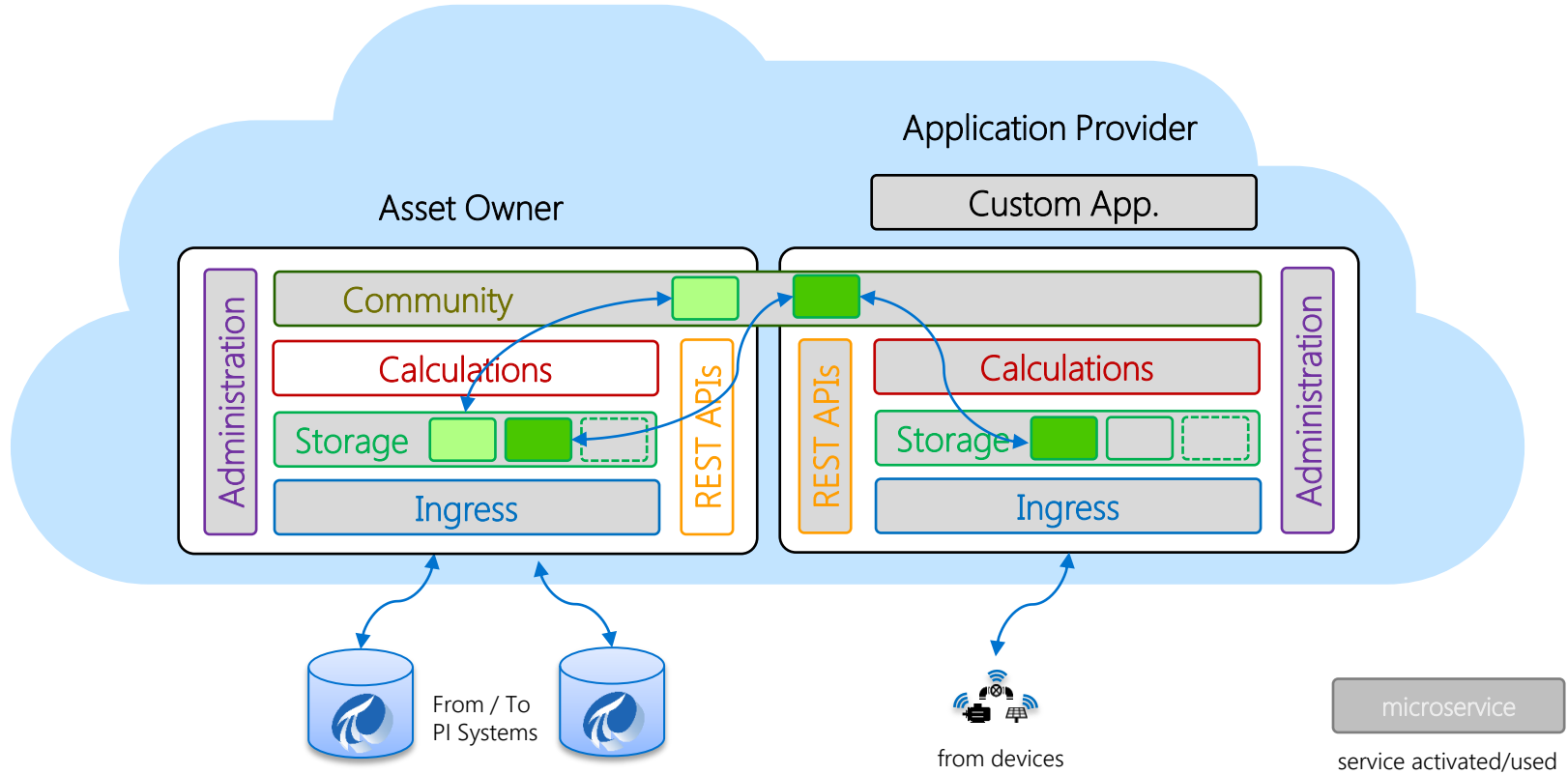
**Start** **Cancel**

# OCS and You

- Ingress
  - OMF
  - Direct API Access
- Storage
  - Time series and more!
  - Complex Types
- Egress
  - Flexible API
- Calculations
  - Template-based

# The OCS Partner Program Preview

# Typical scenario



# Partner Preview Program – Overview

- Kick-off mid-November
  - Gradual onboarding
- Requirements
  - App./Solution concept
  - Development resources
  - Target market / customers
- Resources
  - OCS account, developers, discussion forum



## Cloud Services for Vendors & Service Providers

With OSIsoft Cloud Services, customers can share operations data with you so you have real-time visibility into remote assets and processes to determine if assets and systems are performing as expected.

### Create and share displays with your customers



OSIsoft Cloud Services enables you to create and share displays that are updated in real-time to show KPIs and other key metrics so customers can see the value of your solution.

### Expand your business while optimizing costs



OSIsoft Cloud Services is Software-as-a-Service (SaaS) so you can scale your business without adding additional IT infrastructure and resources.

### Managed and kept up-to-date by OSIsoft



Our scalable and reliable cloud service offering is built on Microsoft Azure and is completely maintained and managed by OSIsoft. You don't need to worry about installing, monitoring, or updating the system.

Interested in Building Apps for OSIsoft Cloud Services?

Request to Join our Partner Preview for the OCS Developer Toolkit.

[Request to Join ▶](#)



**Laurent Garrigues**

[lgarrigues@OSIsoft.com](mailto:lgarrigues@OSIsoft.com)

SaaS Program Manager

OSIsoft, LLC

**Gregg Le Blanc**

[gleblanc@OSIsoft.com](mailto:gleblanc@OSIsoft.com)

Director, Applied Research  
Technology

OSIsoft, LLC

**Paul Kaiser**

[pkaiser@OSIsoft.com](mailto:pkaiser@OSIsoft.com)

Director, Cloud Services

OSIsoft, LLC

## Questions

Please wait for the **microphone** before asking your questions



State your **name & company**

## Please remember to...

Complete the Online Survey for this session



Download the OSISOFT Users Conference 2017 app

**UPDATED VERSION COMING SOON**

Download on the App Store  
GET IT ON Google Play  
HTML

search OSISOFT in the app store

감사합니다

Danke

谢谢

Merci

Gracias

**Thank You**

ありがとう

Спасибо

Obrigado

this.Dispose(true);