

OCS for Developers

Chad Chisholm, Engineering Program Manager

Gregg Leblanc, VP of Product

Laurent Garrigues, Sr. Strategic Product Manager

Talking Points & Demos

- Sign in & Authentication
- PltoOCS Connections
- SDS Types & Streams
- SDS Storage Explorer
- OMF Connections
 - OMF Message validation
- API Console
- Secondary indexes
- UOMs
- SDS Streams metadata & tags
- Access Controls
- Client Libraries
- Hyperscale demo

• Q&A

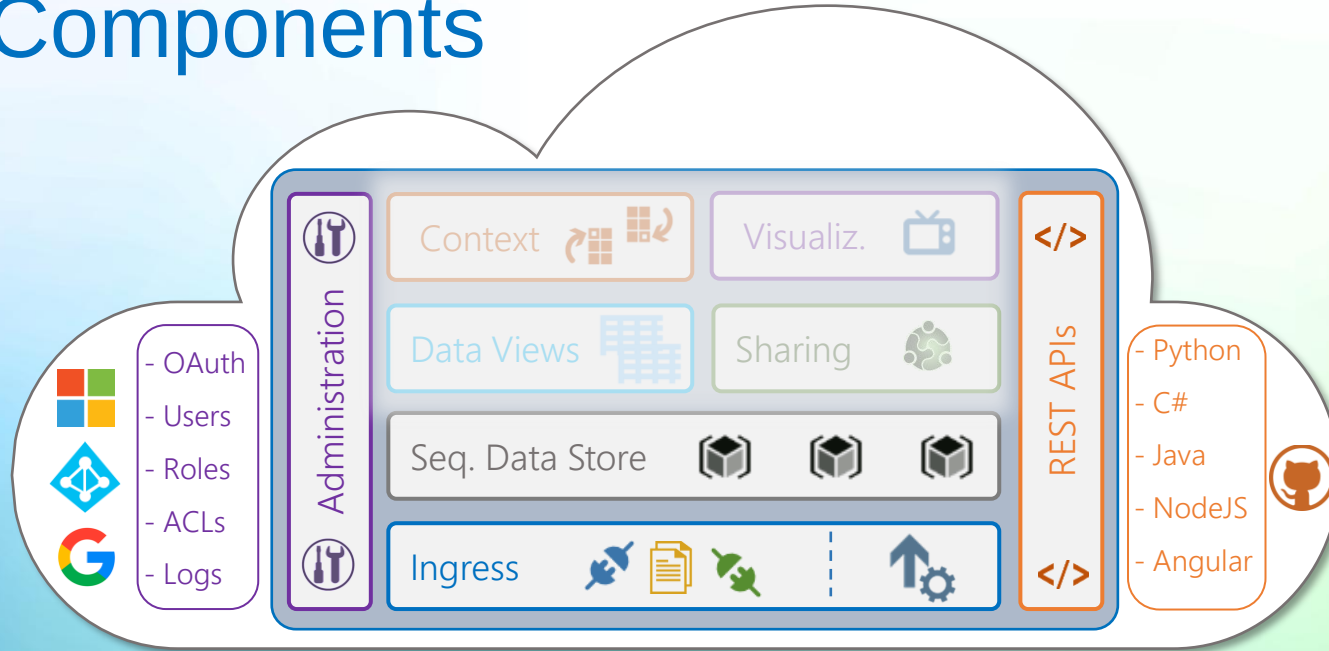
OSIsoft Cloud Services Features

- Managed, secure, multi-tenant data platform
- Operated & maintained by  **OSI**soft.
- High speed, scalable, elastic data ingress
- Flexible, resilient, data storage
- Modern, secure REST APIs
- Built and deployed on  **Microsoft Azure**



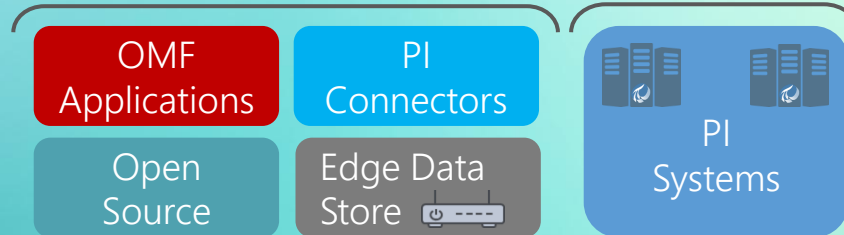
cloud.osisoft.com

OCS Components



OSIssoft Messaging Format

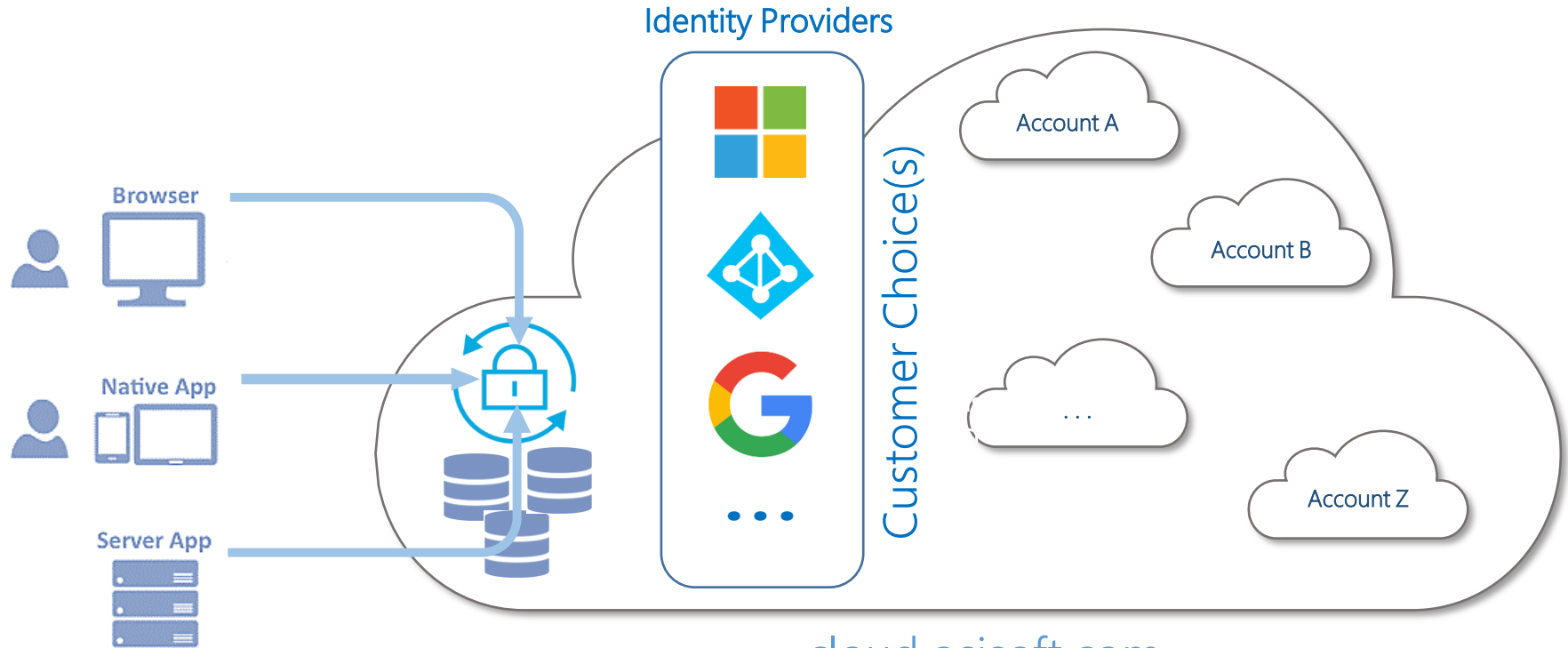
Native



Sign In & Authentication

- Identity Providers
- Sign in the OCS Customer Portal
- Users management

OCS Authentication



cloud.osisoft.com

PI to OCS

- Overview
- Deployment

OCS Account

Sequential Data Store

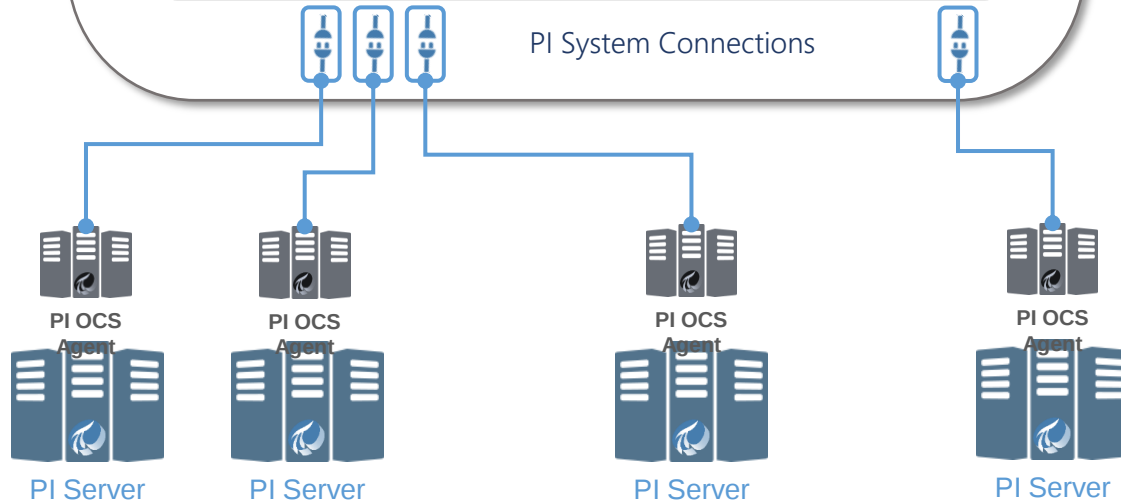
Namespaces



...



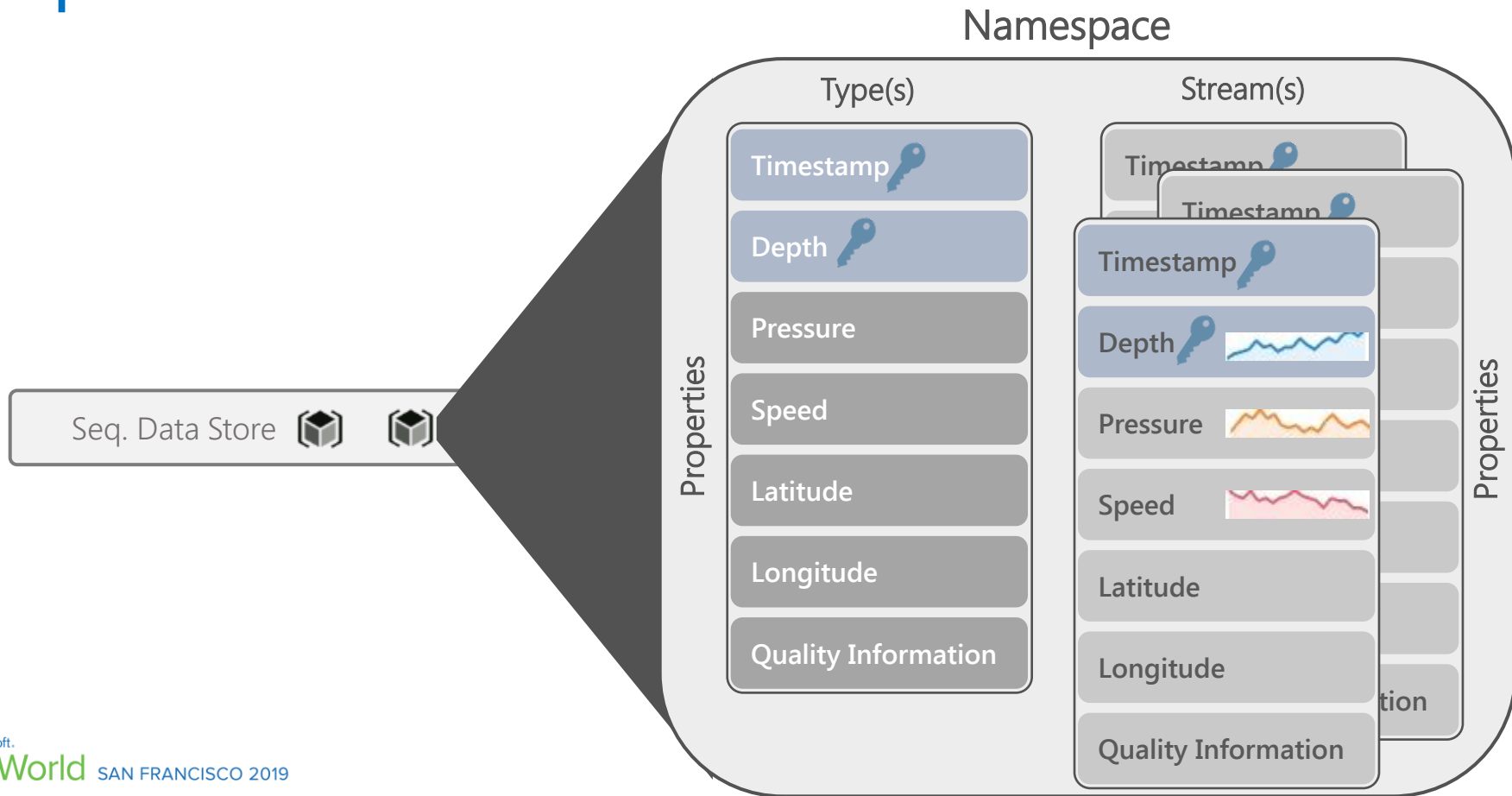
PI System Connections



SDS Types & Streams

- Simple Types
- Complex types

Sequential Data Store - SDS



Complex and Non-Time Index Data Type Example

Simple Data Type Indexed by Time

```
{"Latitude": "29.7817",  
"Timestamp": "2017-12-01"}
```

```
{"Longitude": "-95.6112",  
"Timestamp": "2017-12-01"}
```

```
{"Heading": "42",  
"Timestamp": "2017-12-01"}
```

```
{"Heart-rate": "143",  
"timestamp": "2017-12-01"}
```

```
{"Temperature": "98.6",  
"Timestamp": "2017-12-01"}
```

```
{"StepCount": "10,000",  
"Timestamp": "2017-12-01"}
```



Complex Data Type Indexed by Time & StepCount

```
{"Latitude": "29.7817",  
"Longitude": "-95.6112",  
"Heading": "42",  
"Heart-rate": "143",  
"Temperature": "98.6",  
"Timestamp": "2017-12-01",  
"StepCount": "10,000"}
```

Simple Type

Timestamp 

Depth

Quality bit

Timestamp 

Pressure

Quality bit

Timestamp 

Speed

Quality bit

Complex Type

Platform Pressure

Timestamp 

Depth 

Pressure

Speed

Quality Information

...

Batch Measurement

Lot # 

Ph

Density

Date

Color

...

⋮

Timestamp 

Voltage

Amperage

Power

Status

...

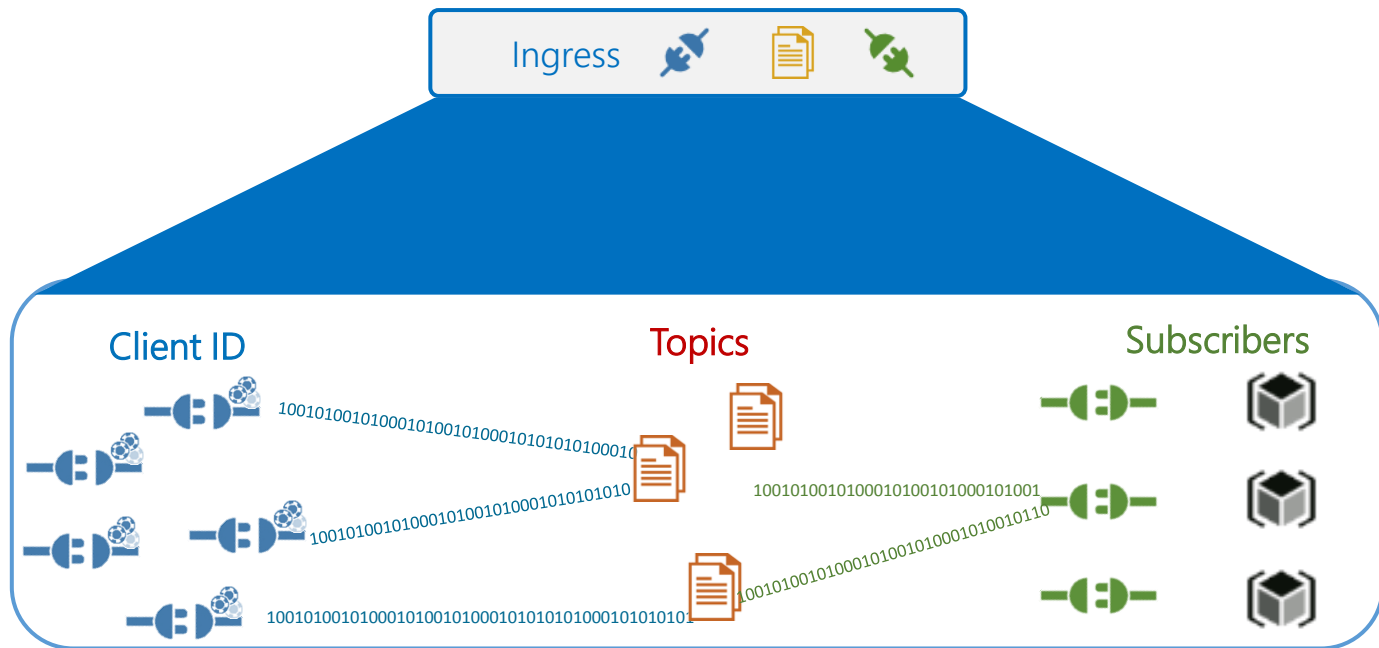
SDS Storage Explorer

- Type designer
- Stream creation
- Access Control
- Data retrieval

OMF Connections

- Create a new connection
- Get Client ID & Client Secret
- OMF Message validation
- OMF Explorer

OMF Ingress



API Console – Basics

- Creating a stream
- Writing data to the stream
- Querying stream data

API Console – Advanced

- Secondary indexes
- UoM conversion
- TableH Output Format

Using Data via Jupyter

First, we need to set up the Jupyter Notebook and Python environment so we can talk to Cloud Services

That means, we're going to import some libraries and make sure we can access Cloud Services

In [1]:

Slide Type

Skip



```
#Slide show: jupyter nbconvert "WITSML Sample Queries.ipynb" --to slides --post serve
import pandas as pd
import pprint
import sys
import os
from os import walk
from os.path import isfile, join
import json
import io
```

Out[5]:

	PropertyId	LastValue	PropertyName	Description
0	DEPTH	2100.68	MeasuredDepth_(m)	Measured Depth
1	CreationDate	2016-03-05T23:49:37Z	CreationDateIndex	Creation DateTime for use as compound index in...
2	startIndex	1850.42	BeginningIndex	Starting Depth of data in this drilling run
3	endIndex	2100.68	EndingIndex	Ending Depth of data in this drilling run

Selecting Data via Jupyter

Slide Type Slide ▼

So let's select some data smartly

Here we'll see that using some basic Pandas options, we can just select the data we want, from the CreationDates we have in our drilling runs.

We'll bring back just a few properties, and the first 5 rows of data.

In [10]:

Slide Type Sub-Slide ▼

```
pp.pprint(gwv_sorted[['A16H', 'CreationDate', 'DEPTH', 'ROP5']][gwv_sorted['CreationDate']=='2015-11-17T14:21:14Z'][:5])  
pp.pprint(gwv_sorted[['A16H', 'CreationDate', 'DEPTH', 'ROP5']][gwv_sorted['CreationDate']=='2015-12-03T03:27:12Z'][:5])
```

	A16H	CreationDate	DEPTH	ROP5
0	NaN	2015-11-17T14:21:14Z	4217.879883	NaN
2	NaN	2015-11-17T14:21:14Z	4217.975098	30.886269
5	NaN	2015-11-17T14:21:14Z	4217.999023	NaN
7	NaN	2015-11-17T14:21:14Z	4218.126953	31.069880
10	NaN	2015-11-17T14:21:14Z	4218.279785	31.114563

	A16H	CreationDate	DEPTH	ROP5
4	0.99826	2015-12-03T03:27:12Z	4217.975098	30.886271
9	1.00593	2015-12-03T03:27:12Z	4218.126953	31.069880
12	0.99987	2015-12-03T03:27:12Z	4218.279785	31.114559
15	0.99561	2015-12-03T03:27:12Z	4218.432129	30.829010
18	0.99763	2015-12-03T03:27:12Z	4218.583984	31.015459

Using Pandas, OCS, and then plotting...

In [14]:

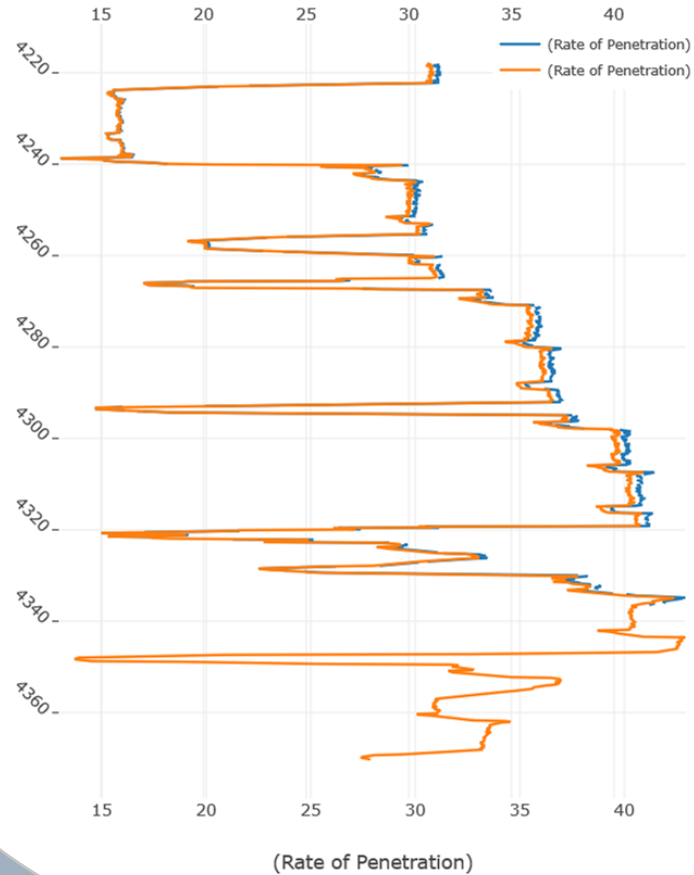
Slide Type Sub-Slide ▾

```
# First, get some data - 1000 rows, from a couple drilling runs
drillRun1 = gvw_sorted[['A16H', 'CreationDate', 'DEPTH', 'ROP5']][gvw_sorted['CreationDate']=='2015-11-17T14:21:14Z'][:1000]

drillRun2 = gvw_sorted[['A16H', 'CreationDate', 'DEPTH', 'ROP5']][gvw_sorted['CreationDate']=='2015-12-03T03:27:12Z'][:1000]

#Now, let's plot these to see what we've done...
verti_plot2(gvw_x1=drillRun1['DEPTH'],
            gvw_y1=drillRun1['ROP5'], yname1='(Rate of Penetration)',
            gvw_x2=drillRun2['DEPTH'],
            gvw_y2=drillRun2['ROP5'], yname2='(Rate of Penetration)',
            plot_title= 'Vertical Plot of WITSML Depth Frame from Cloud Services')
```

Vertical Plot of WITSML Depth Frame from Cloud Services (Rate of Penetration)



Selecting via multiple indices

```
gww_results = client.getWindowValues(namespace_id=sampleNamespaceId,  
                                     start = '4217.822|2015-11-17T14:21:59.000Z',  
                                     end   = '4884.95|2015-11-17T14:21:59.000Z',  
                                     stream_id= streamId)
```

```
stIdx = 0  
enIdx = 755
```

```
gww_dataFrame = gww_sorted.interpolate() #pd.DataFrame(gww_results).interpolate()#.fillna(0)  
gww_dataFrame_drop = gww_sorted.dropna(axis = 0, subset=tinySubHeadingDepth, how='any')
```

```
pretty_plot(gww_dataFrame['DEPTH'][:enIdx],  
            gww_dataFrame['ROP5'][:enIdx], 'Rate of Penetration (in/min)',  
            gww_dataFrame['RPM'][:enIdx], 'Rev/Min of Drill',  
            'Plot of WITSML ROP and RPM vs. Depth from OCS - interpolated')
```

SDS Streams – Metadata

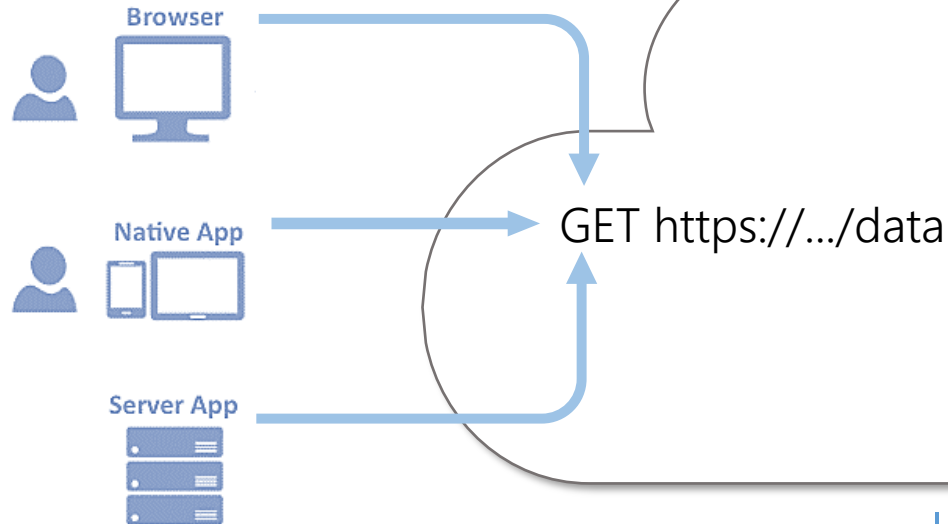
- PI to OCS generated metadata
- Create metadata
- Find streams by metadata
 - area:sltc1
 - tags:environmental
 - query=pointid:1

Authorization – Access Control

- Identities
- Roles
- ACLs
 - Namespaces
 - SDS Types
 - SDS Streams

OCS Authorization

Roles:
Engineer,
Display-Creator



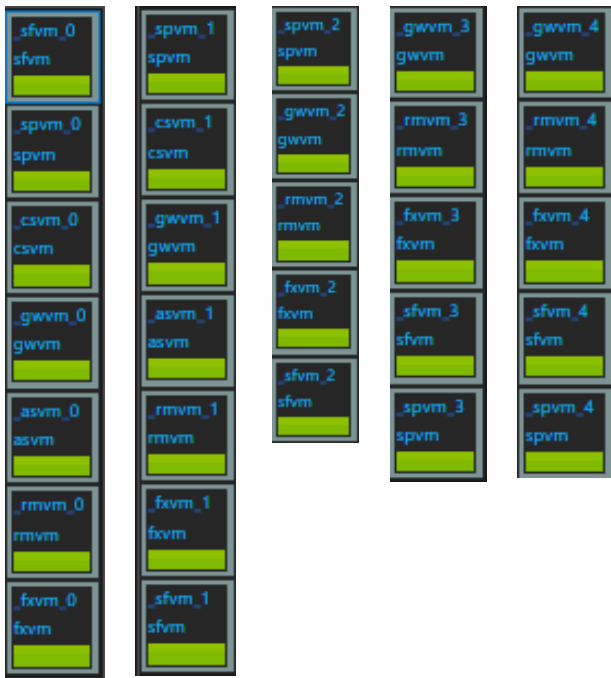
cloud.osisoft.com

Code Samples

- Client Libraries
- Nuget.org
- Console App

Hyperscale OCS Cluster

Smart Small...



OCS Scale Out*



* We're pretty sure this is how it works...

Get Bigger...



Monitoring the Cluster...



Some Numbers...

Data

- ~2 Billion equivalent PI Tags
 - ~100M AMI Meters
 - 20 properties each
- ~64M events/sec ingress
- Trillions of events stored

OCS Cluster

- 477 nodes
- 3132 Cores
- 11TB of RAM

Questions?

Please wait for
the **microphone**

State your
name & company



Please remember

TO DOWNLOAD
APP, SEARCH
OSISOFT



