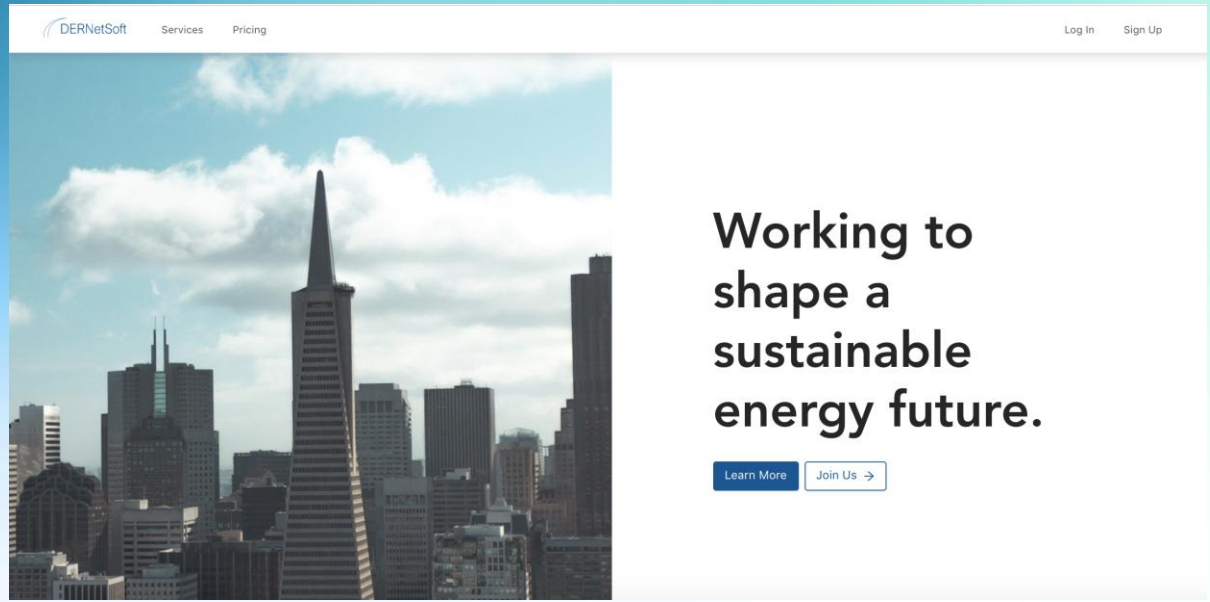
Data Analytics to enhance Advanced Energy Communities planning and operation

Presented by:

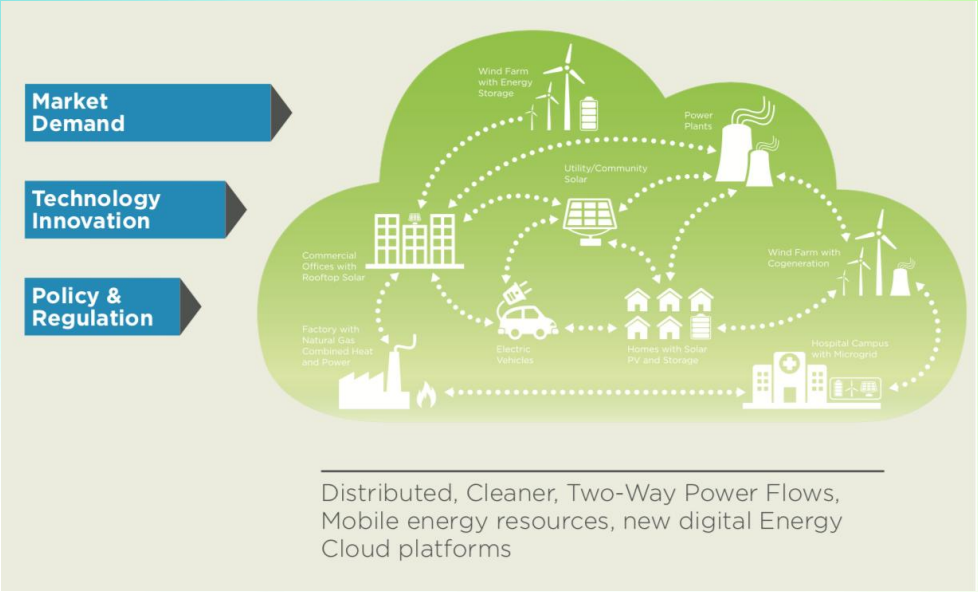
DERNetSoft



- DERNetSoft is tackling climate change.
- A scalable platform solution to be easily deployed in cities and local communities.

DER Market Trends

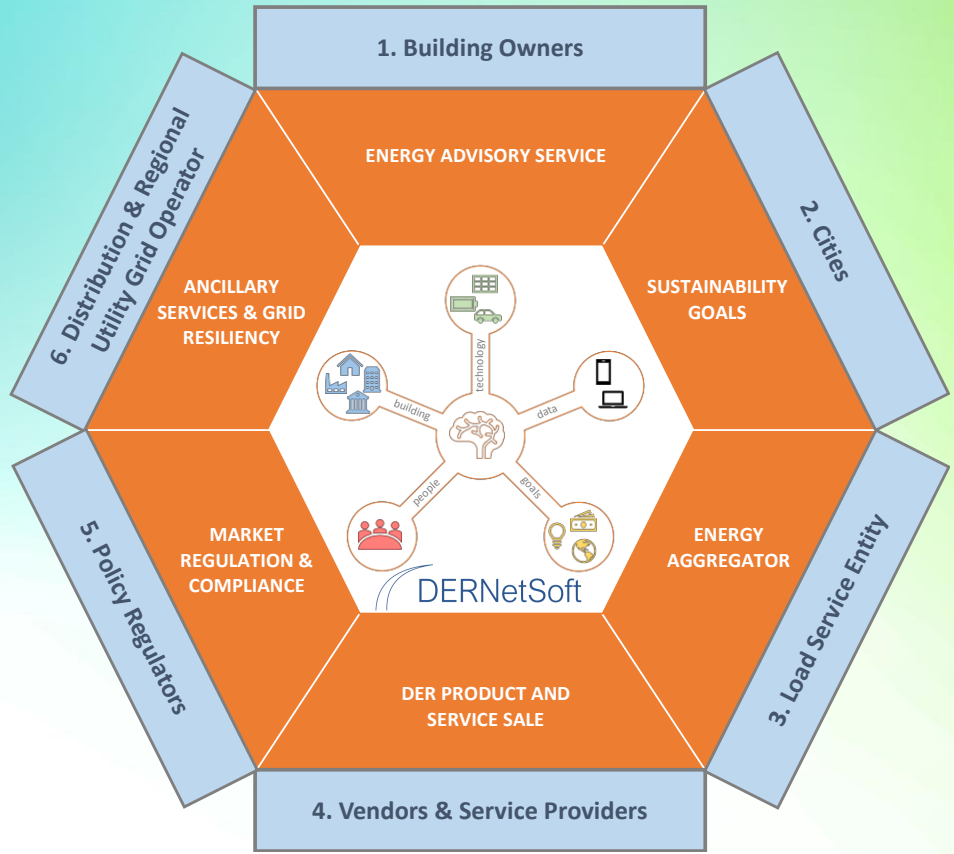
- DER capacity is growing
- DERs include “negative generation” (efficiency and demand response)
- DERs could be a business opportunity for utilities
- How we understand the economic value of DERs is evolving



(Source: Navigant Research)

Digital Marketplace

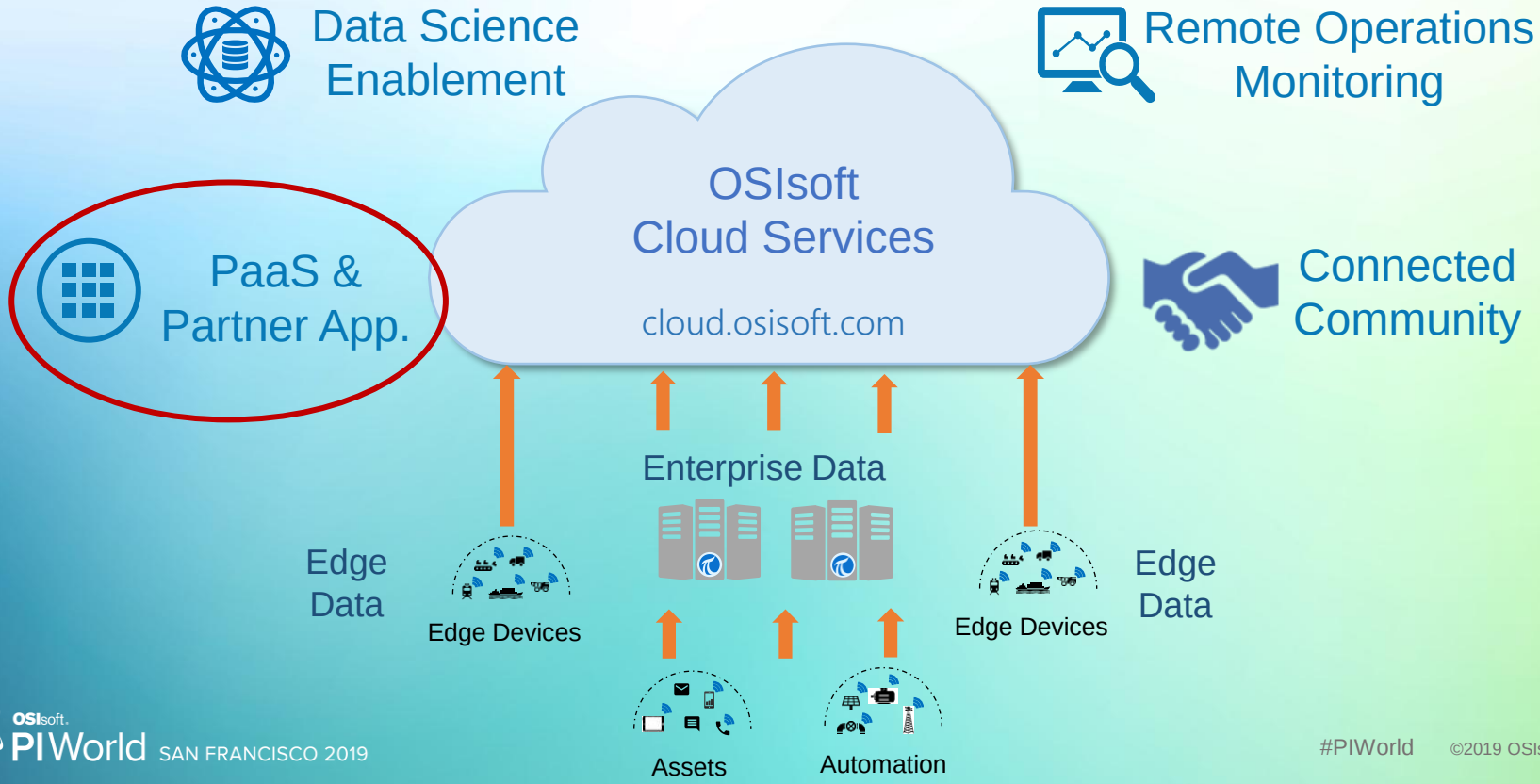
- Energy awareness
- Compliance reporting
- Peak demand forecasting
- Load Profile sharing
- Building benchmarking
- DER leads generation
- Community data aggregation
- …. more

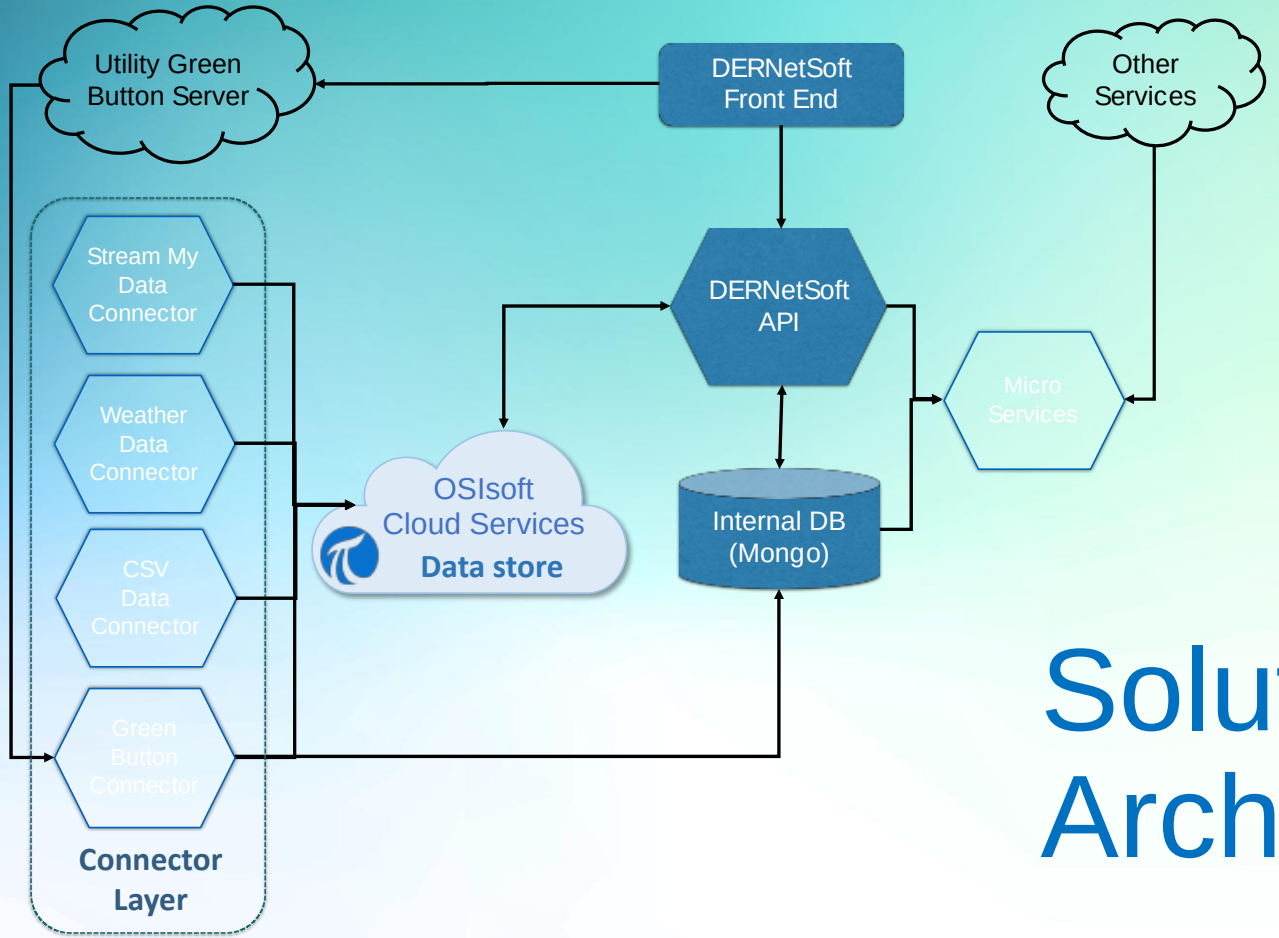


Business Challenge

- Manage multiple time series data stream (weather, usage, market, static)
- Manage different data granularity
- Use Forecasting model
- Use Advanced Analytics
- Provide Simple and Aggregated view
- Service scalability and reliability

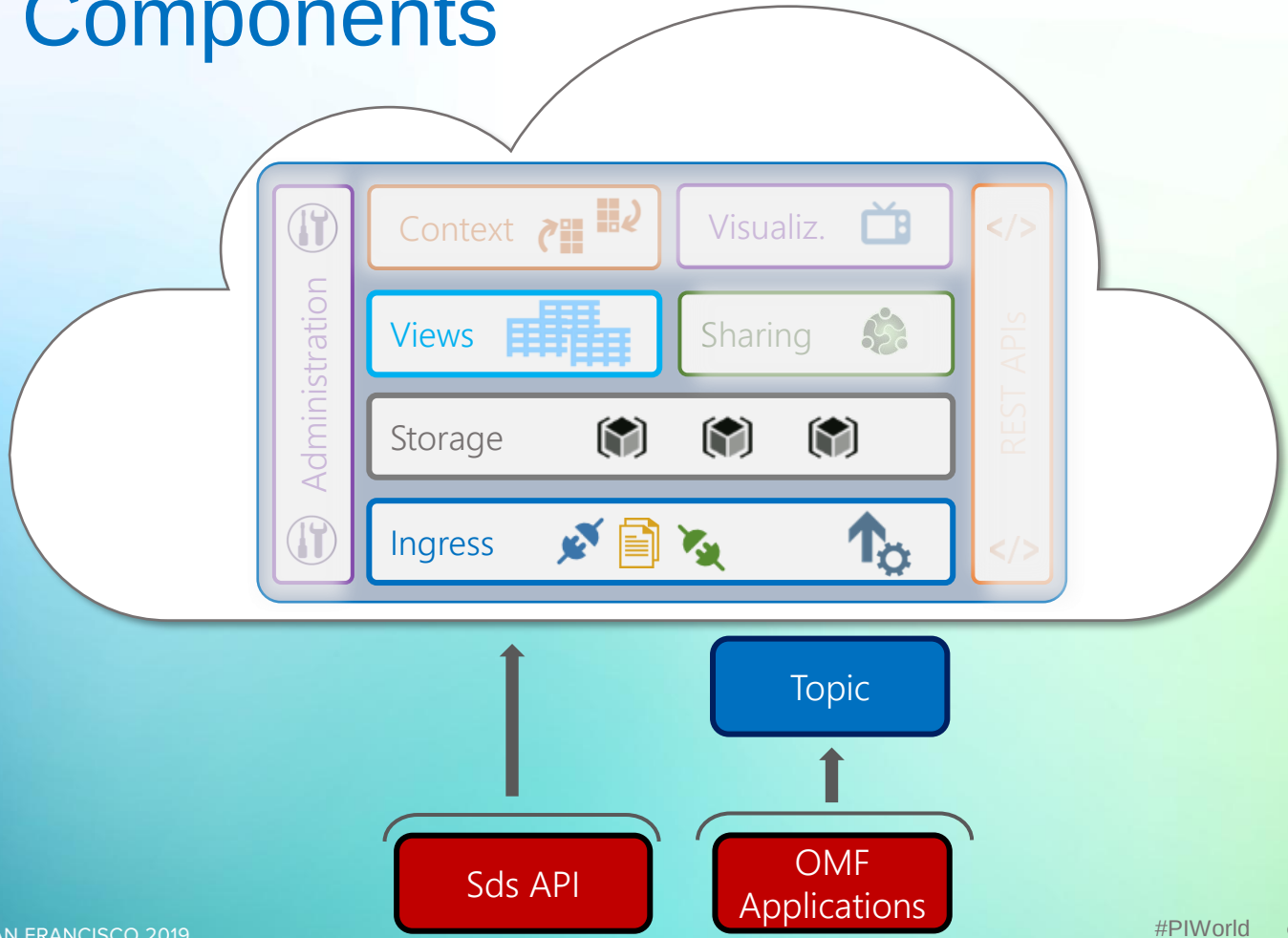
OCS – DERNetSoft Scenarios





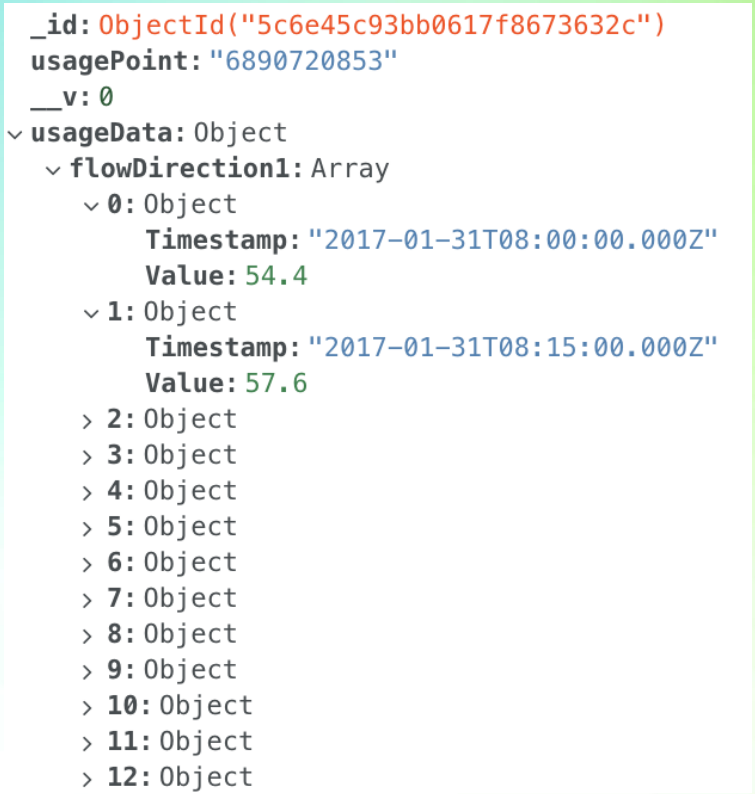
Solution Architecture

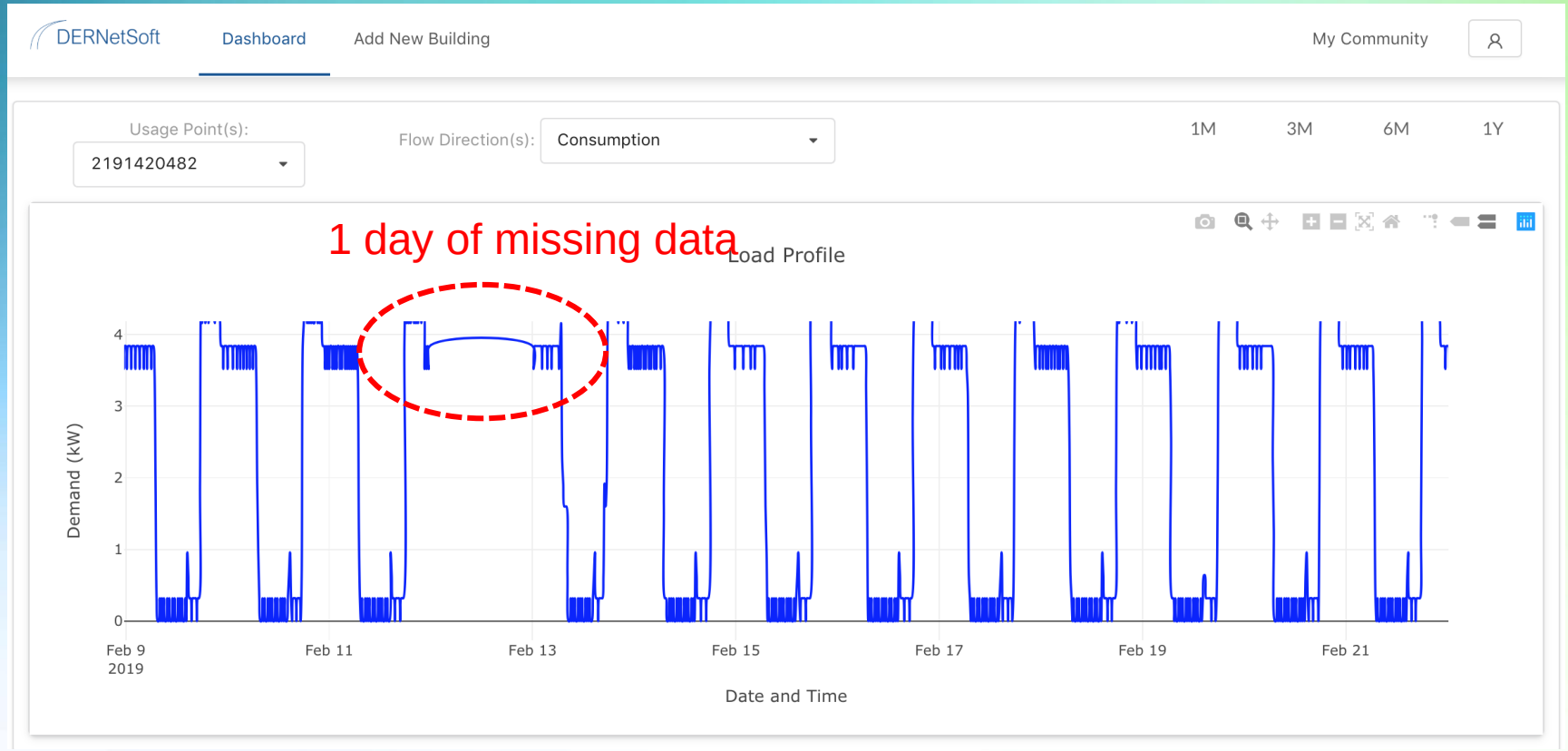
OCS Components



Data Store – Before

- MongoDB for both asset management and time series data
- Deeply embedded objects for energy usage data required significant effort for a simple update command
- No rigidity left us with unexpected results





Data Store – After

Sequential Data Store

Easy to get started

Easy to organize time series data

Simple and secure APIs

Sds Client vs OMF

Sds Client

- Lowest level to the hardware
- API access to individual Sds Objects

Example

.../Namespaces/dernetsoft-aec/Streams/GreenButtonStream...

OMF

- A level “above” Sds
- An application publishes OMF to a Topic, from which a Subscription writes to OCS

Example

.../Namespaces/dernetsoft-aec/omf

Use Cases

Use Case List

1. Building Energy metric comparison
2. PV Solar, Energy Storage sizing, Peak Forecasting
3. Advanced Energy Community view

Use Case 1

Building Energy metric comparison

SERVICE

Building energy metrics locational comparison

(Energy Usage Intensity)

REQUIREMENTS

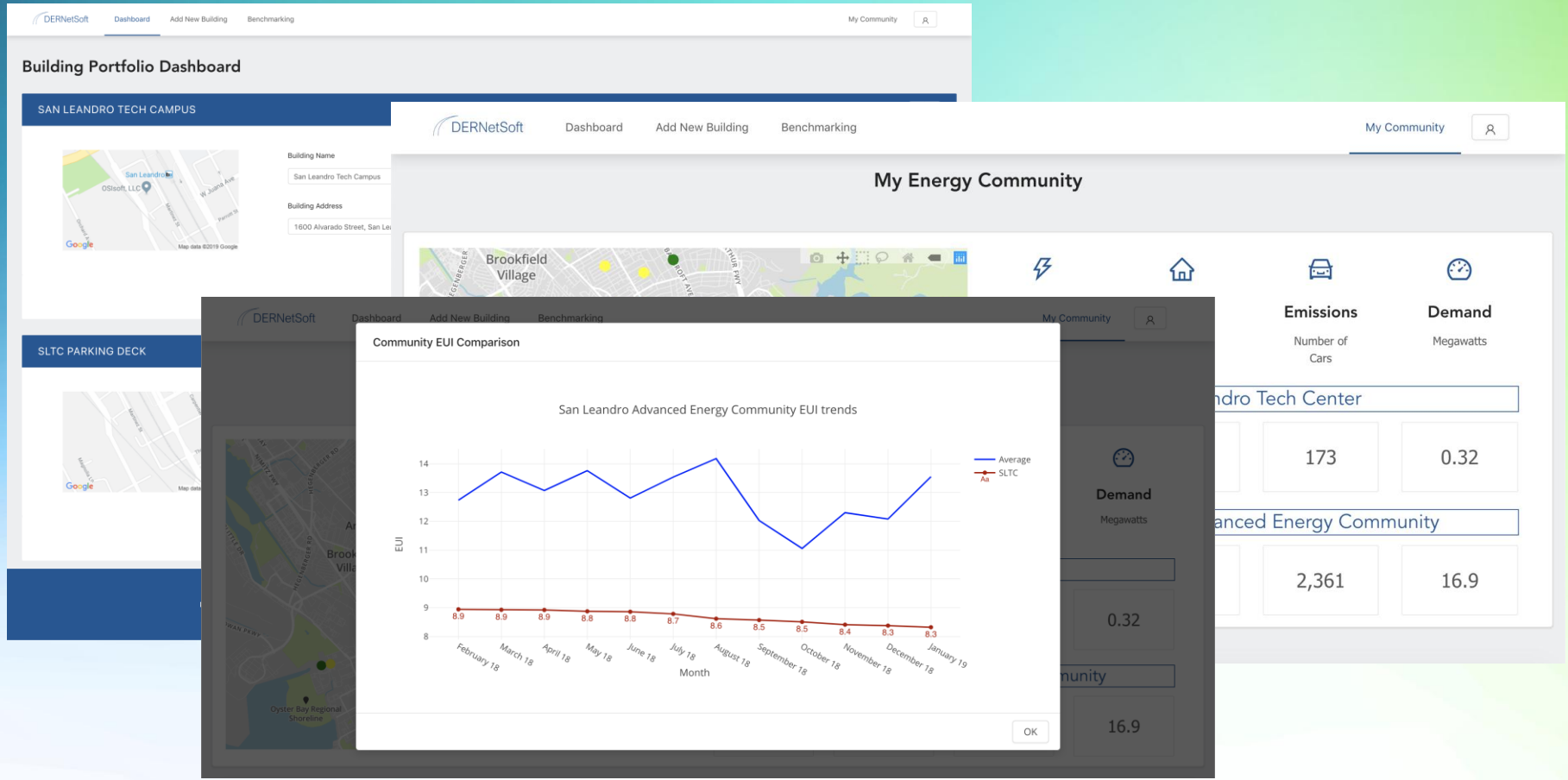
- Time Series data management
- Energy metrics calculation and community aggregation

PRODUCT

- OCS data store
- Stream Metadata

BENEFITS

- Reliability and scalability
- Faster data access

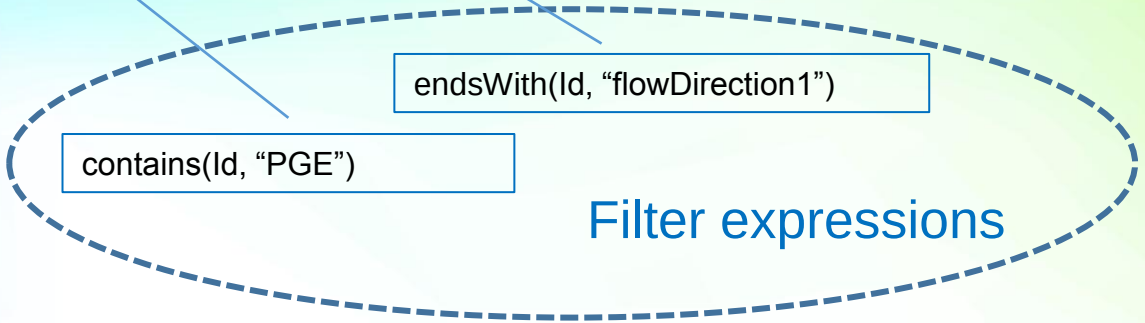


SdsStream definition

```
{
    "TypeId": "GreenButton_Type",
    "Id": "GreenButtonStream_PGE_6890720853-flowDirection1",
    "Name": "Green Button Stream GreenButtonStream_PGE_6890720853-flowDirection1",
    "Description": "A stream to hold Green Button timestamp/value pairs usage point: 6890720853-flowDirection1",
}
```

Metadata definition

```
{
    "squareFootage": "132750",
    "zipCode": "94577"
}
```



Use Case 2

PV Solar, Energy Storage sizing, Peak Forecasting

SERVICE

C&I Building

Peak Demand forecasting and DER sizing

REQUIREMENTS

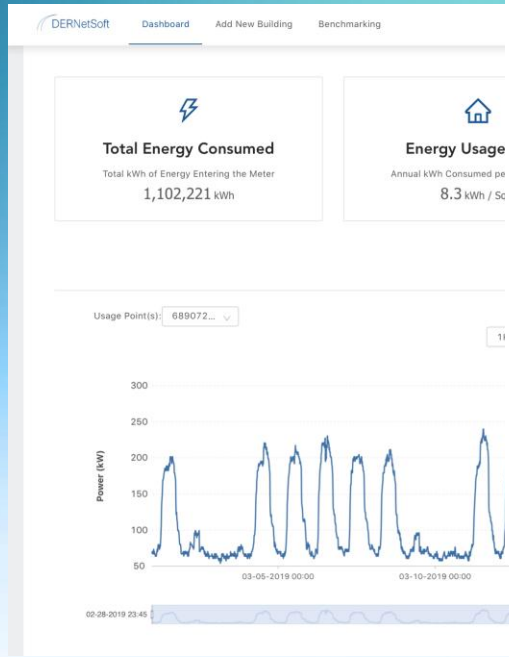
- Integration of different time series data
- Advanced analytics
- Peak demand forecasting

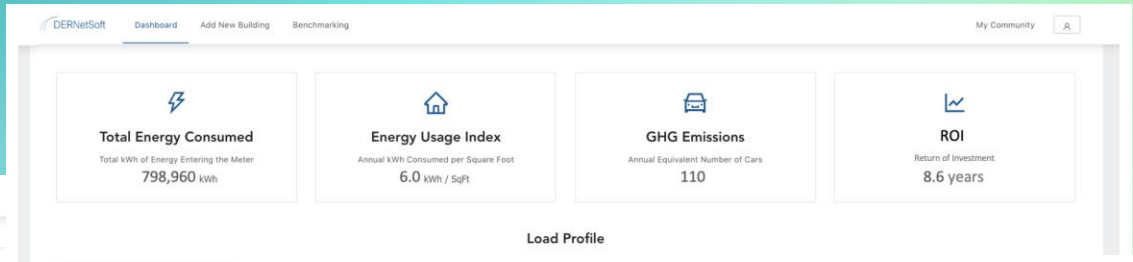
PRODUCT

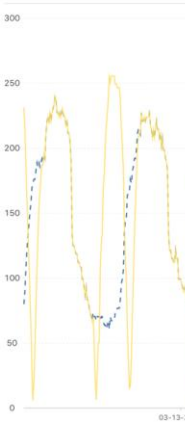
- OCS data store
- Data Views

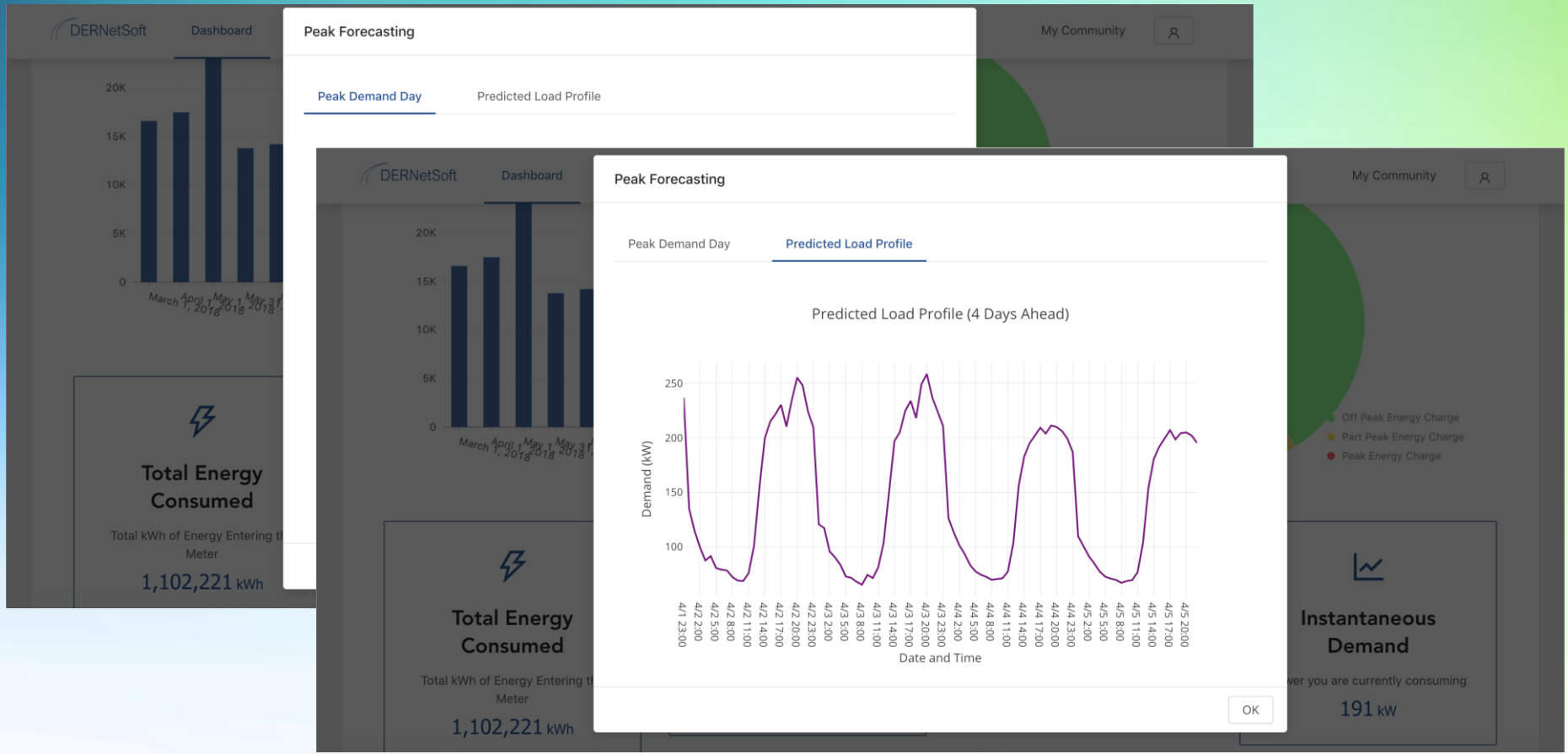
BENEFITS

- Easy data stream integration
- Data access improvement for machine learning engine



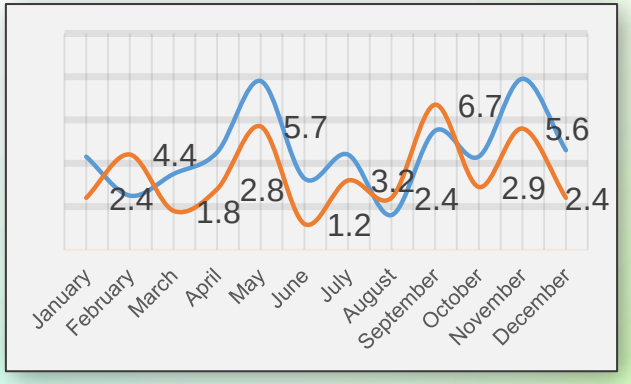






Data Views

REST APIs </>



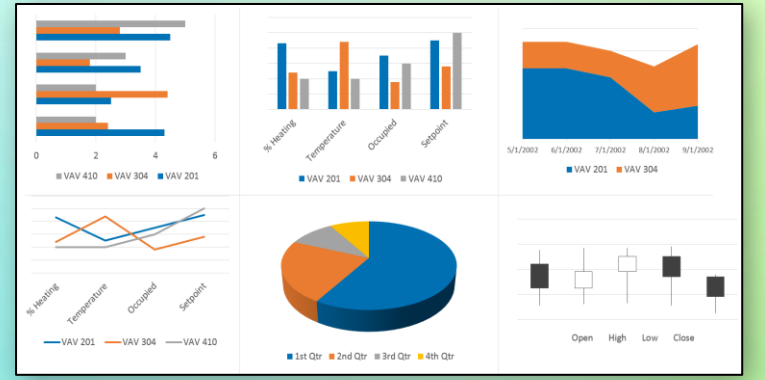
Data Views

OSIsoft
Cloud Services

Enterprise Data

Edge Data

REST APIs </>



Data Views definition

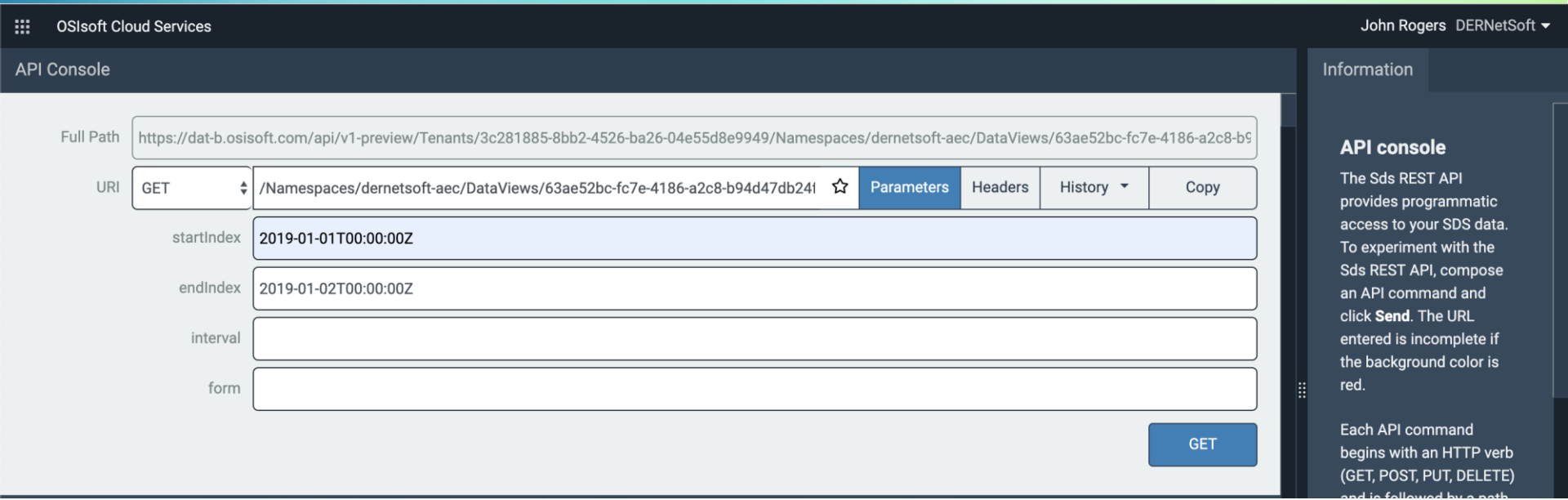
```
"Id": null,
"Name": "PM_Ingress_6890720853",
"Description": "Data View for PM ingress for usage point 6890720853",
"Queries": [
    {
        "Id": "ConsumptionStream",
        "Query": {
            "Resource": "Streams",
            "Field": "Id",
            "Value": "6890720853",
            "Function": "Contains"
        }
    },
    {
        "Id": "WeatherStream",
        "Query": {
            "Resource": "Streams",
            "Field": "Id",
            "Value": "_94577",
            "Function": "EndsWith"
        }
    }
],
```

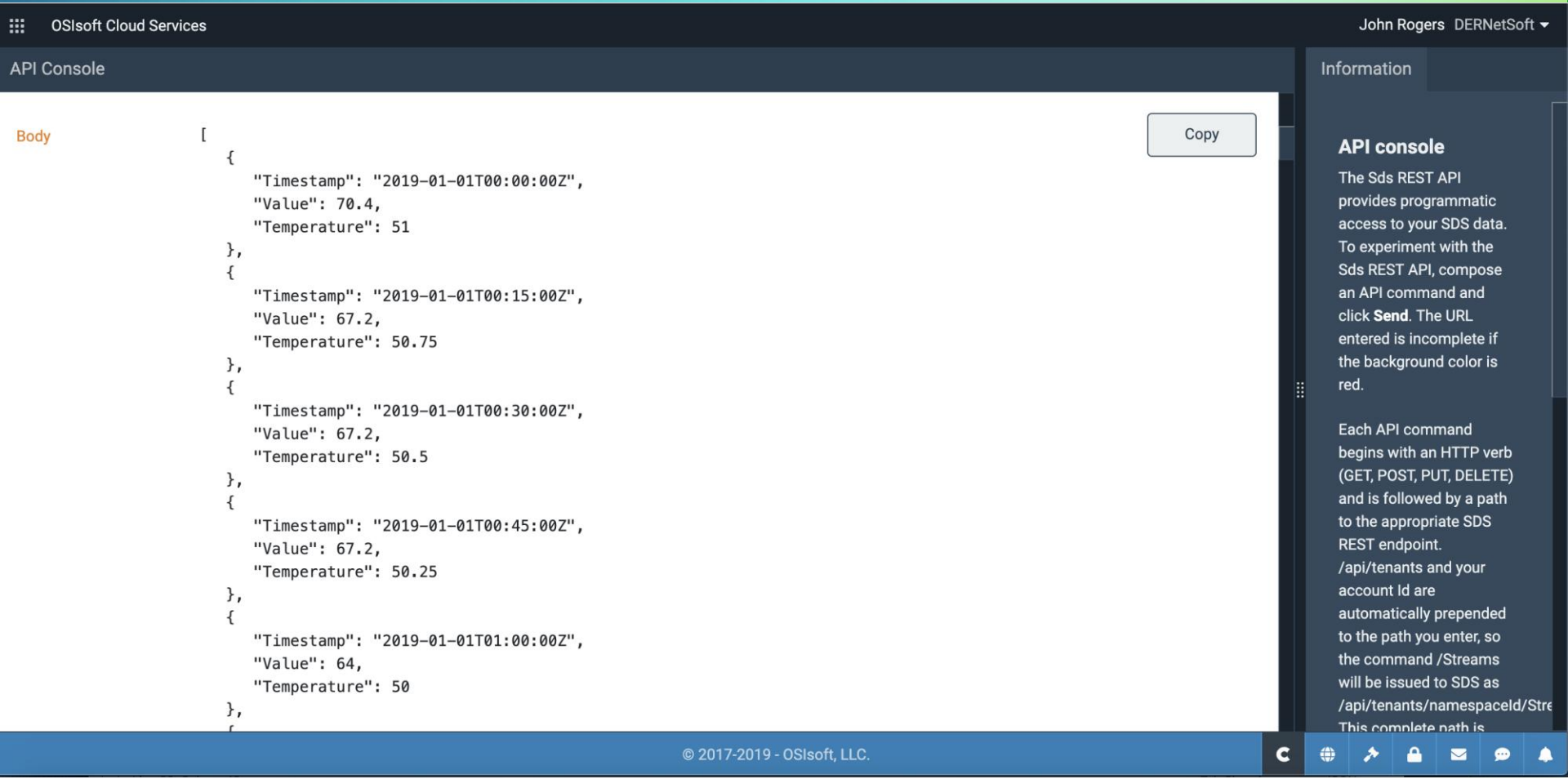
Rules definition

```
"GroupRules": [],
"IndexDataType": "DateTime",
"IndexConfig": {
    "StartIndex": "2018-01-01T00:00:00Z",
    "EndIndex": "2018-01-02T00:00:00Z",
    "Mode": "Interpolated",
    "Interval": "00:15:00"
}
```

DEMO

OCS Console




OSIsoft Cloud Services
John Rogers DERNetSoft
API Console
Information
Body
Copy
[
{
"Timestamp": "2019-01-01T00:00:00Z",
"Value": 70.4,
"Temperature": 51
},
{
"Timestamp": "2019-01-01T00:15:00Z",
"Value": 67.2,
"Temperature": 50.75
},
{
"Timestamp": "2019-01-01T00:30:00Z",
"Value": 67.2,
"Temperature": 50.5
},
{
"Timestamp": "2019-01-01T00:45:00Z",
"Value": 67.2,
"Temperature": 50.25
},
{
"Timestamp": "2019-01-01T01:00:00Z",
"Value": 64,
"Temperature": 50
},
API console
The Sds REST API provides programmatic access to your SDS data. To experiment with the Sds REST API, compose an API command and click Send. The URL entered is incomplete if the background color is red.
Each API command begins with an HTTP verb (GET, POST, PUT, DELETE) and is followed by a path to the appropriate SDS REST endpoint. /api/tenants and your account Id are automatically prepended to the path you enter, so the command /Streams will be issued to SDS as /api/tenants/namespaceId/Stre
This complete path is
© 2017-2019 - OSIsoft, LLC.

Use Case 3

Advanced Energy Community view

SERVICE

Advanced Energy Community view

REQUIREMENTS

Data streams aggregation across multiple parameters to share with energy providers and grid operators

PRODUCT

- OCS data store
- Data Views
- Metadata

BENEFITS

- Secure and scalable data sharing
- Value Added Service opportunity Increase

DERNetSoft Dashboard Add New Building Benchmarking My Community

My Energy Community

Consumption	EUI	Emissions	Demand
MWh Consumed	kWh / Sq Ft	Number of Cars	Megawatts

San Leandro Tech Center

2,460	8.3	173	0.32

San Leandro Advanced Energy Community

26,715	12.9	2,361	16.9

Power of Data Views

metadata key/value pairs

Unique stream IDs

Data Views

Data View definition

Metadata definition

```
{
    “zipCode”: “94577”,
    “DERTech”: “Fuel Cell”,
    “Customer”: 5c0595044403e05b566dc403
}
```

```
{
    "Id": "099a0671-9f6c-443d-a959-8835fd863245",
    "Name": "DataViewDefinition_Name",
    "Description": "DataViewDefinition_Description",
    "Queries": [
        {
            "Id": "GB_Stream_PGE",
            "Query": {
                "Resource": "Streams",
                "Field": "Id",
                "Value": "PGE",
                "Operator": "Contains"
            }
        },
        {
            "Id": "GB_Stream_FlowDirection1",
            "Query": {
                "Resource": "Streams",
                "Field": "Id",
                "Value": "flowDirection1",
                "Operator": "EndsWith"
            }
        }
    ],
    "GroupRules": [
        {
            "Id": "ZipCode",
            "Type": "StreamMetadata",
            "TokenRules": {
                "Tokens": [
                    "zipCode"
                ]
            }
        }
    ],
    "IndexDataType": "DateTime"
}
```

Lesson learned

1. OCS fits nicely in a microservice architecture
2. OCS improves time series data management
3. OCS enables advanced analytics and machine learning oriented applications
4. OCS simplifies large data aggregation view

DERNetSoft Contact

Alberto Colombo

- President
- DERNetSoft Inc.
- alberto@dernetsoft.com

John Rogers

- Software Engineer
- DERNetSoft Inc.
- john@dernetsoft.com

Questions?

Please wait for the **microphone**

State your **name & company**

Please remember