

OCTOBER 26, 2023

---

# Getting the most out of AVEVA's Industrial Platform

Platform Developer Deep Dive

Collin Bardini

Konstantin Chudnovskiy

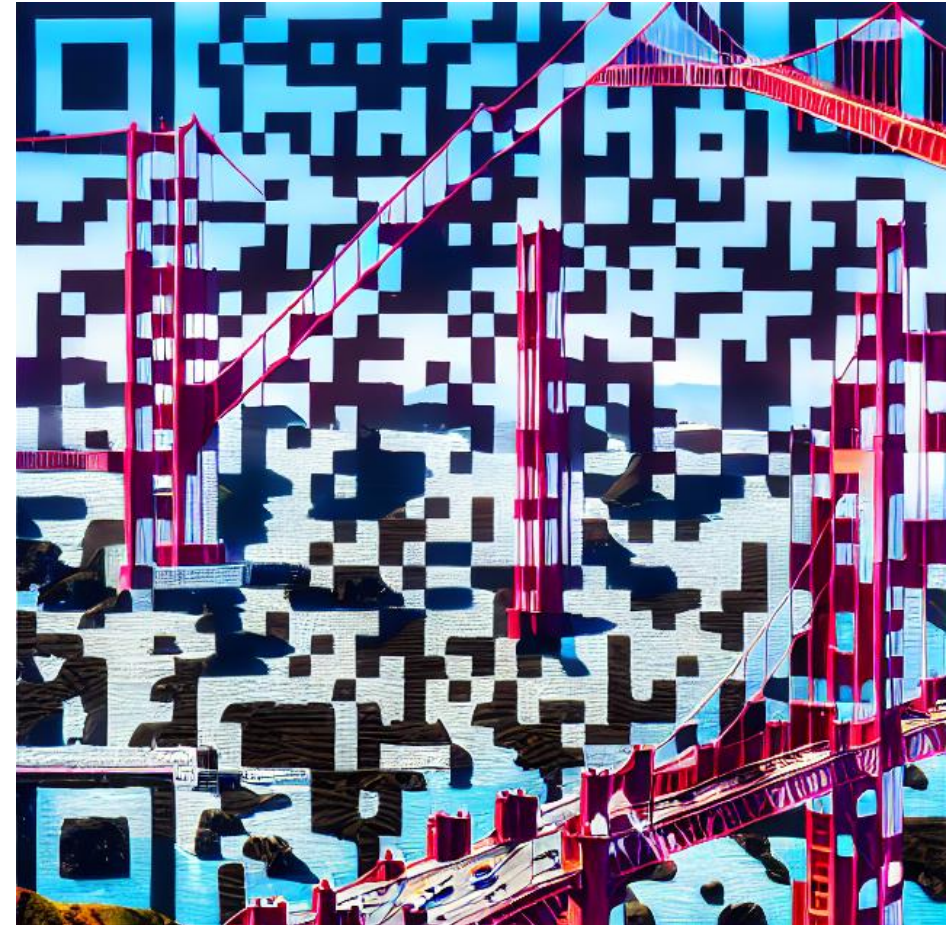
**AVEVA**

---

# EV Car Chargers



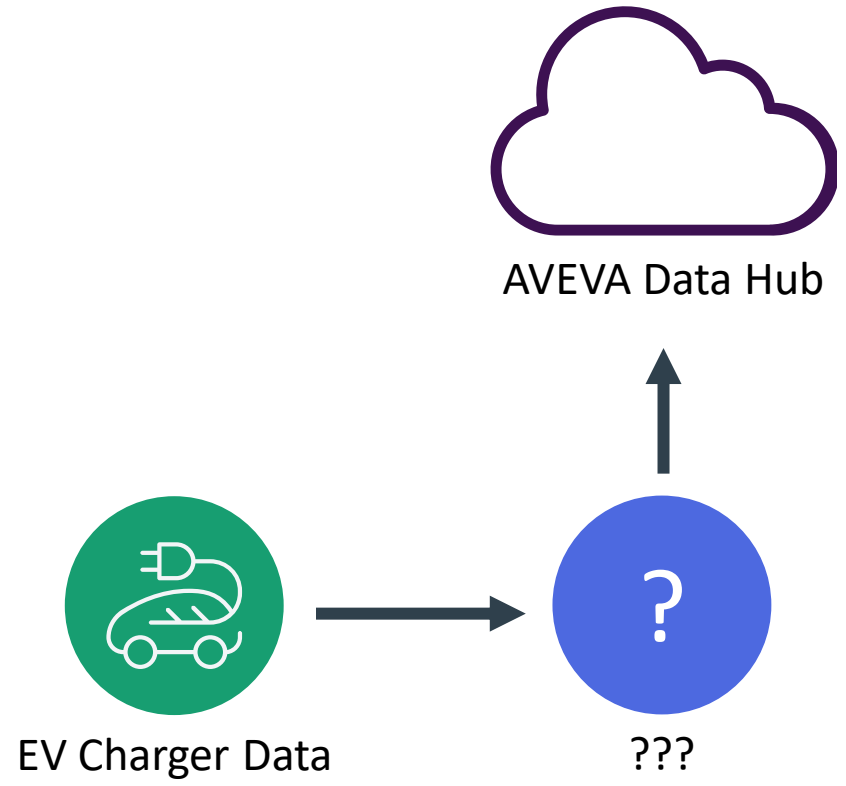
- 
- Code demos available on GitHub
  - [https://github.com/osisoft/demos-aveva\\_world\\_2023](https://github.com/osisoft/demos-aveva_world_2023)



---

# Setting Up Data Infrastructure





# Can I use an Adapter or PI to Data Hub?

## Adapters

- Standard products for sending data to AVEVA Data Hub
- Connect to the following data sources:
  - MQTT (Sparkplug B or generic)
  - OPC UA
  - Azure Event Hubs
  - BACnet
  - DNP3
  - Modbus TCP
  - RDBMS
  - Structured Data Files (csv, json, or XML)

## PI To Data Hub

- Forms a native connection to an on-premises PI Server
- Can replicate AF Elements as Assets
- Standard product

# OMF or SDS writes?

## Data collection: OMF vs. SDS data writes

Last Updated Apr 18, 2022 [AVEVA Data Hub](#) [Industrial Information](#)

[Improve this Doc](#)

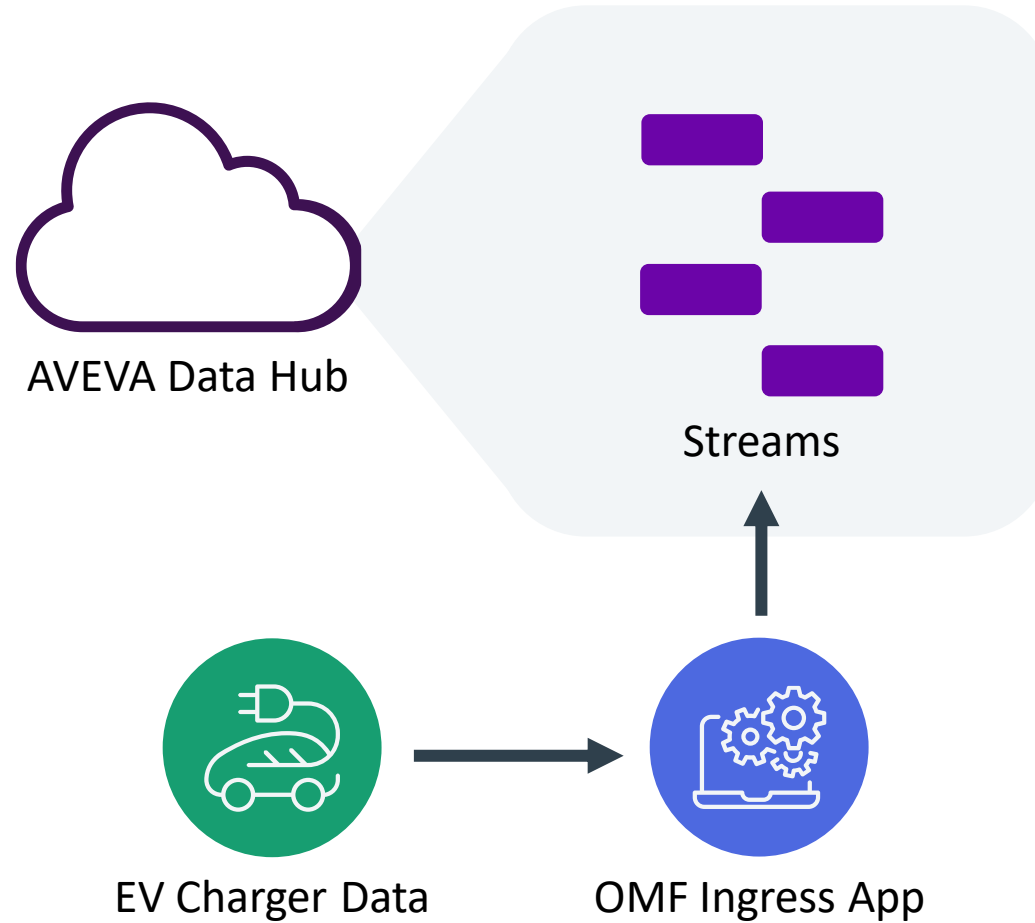
When collecting data programmatically, you have the option of using either the Open Message Format (OMF) or Sequential Data Store (SDS) data writes. OMF is typically the best option for simple and efficient data collection. However, each option has its pros and cons, and this topic highlights the key differences.

### Data collection option comparison

The following table lists the factors that you should consider when choosing between OMF and SDS, as well as which option is the better for each factor. Some factors (**Security** and **Performance**) have no inherently superior option and require additional consideration. For more information on these factors, see [Situation dependent factors](#).

Factor	OMF	SDS
Cross-platform	✓	
Ease-of-use	✓	
Flexibility		✓
Security	-	-
Performance	-	-

<https://docs.aveva.com/bundle/data-hub/page/developer-guide/omf-dev/omf-sds-vs-omf.html>





---

# Performance best practices for OMF

## Critical

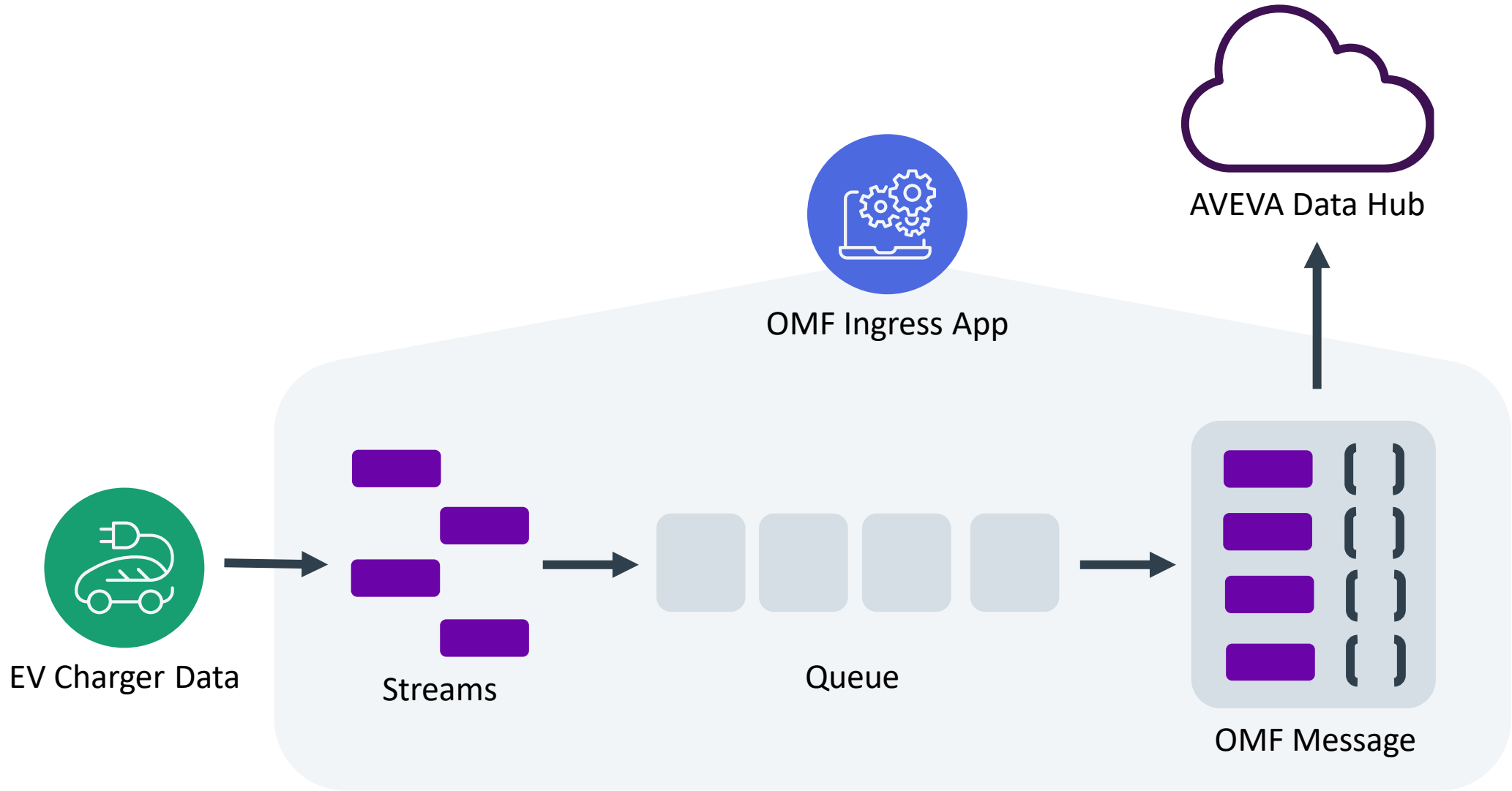
- Avoid out of order (ooo) data


## Very Important

- Maximize content per stream in a message
- Maximize Content per OMF message (192KB)
- Use gzip compression
- You can send data in parallel across steams (still need to account for ooo data)

## Nice to Have

- Exclude default values
- Avoid duplicate entries for streams/containers






```
msg_body = gzip.compress(bytes(json.dumps(omf_message_json), 'utf-8'))
headers = self.getHeaders(message_type, action)


return self.request(
    'POST',
    self.OMFEndpoint,
    headers=headers,
    data=msg_body,
    verify=self.VerifySSL,
    timeout=600
)
```

```
if (response.status_code == 504 or response.status_code == 503):
    if (failures >= 0 and failures >= max_retries):
        logging.error('Server error. No more retries available.')
        return response
    else:
        timeout = 3600 if failures >= 12 else 2 ** failures
        logging.warning('Server error. Retrying...')
        time.sleep(timeout)
        failures += 1
else:
    success = True
```

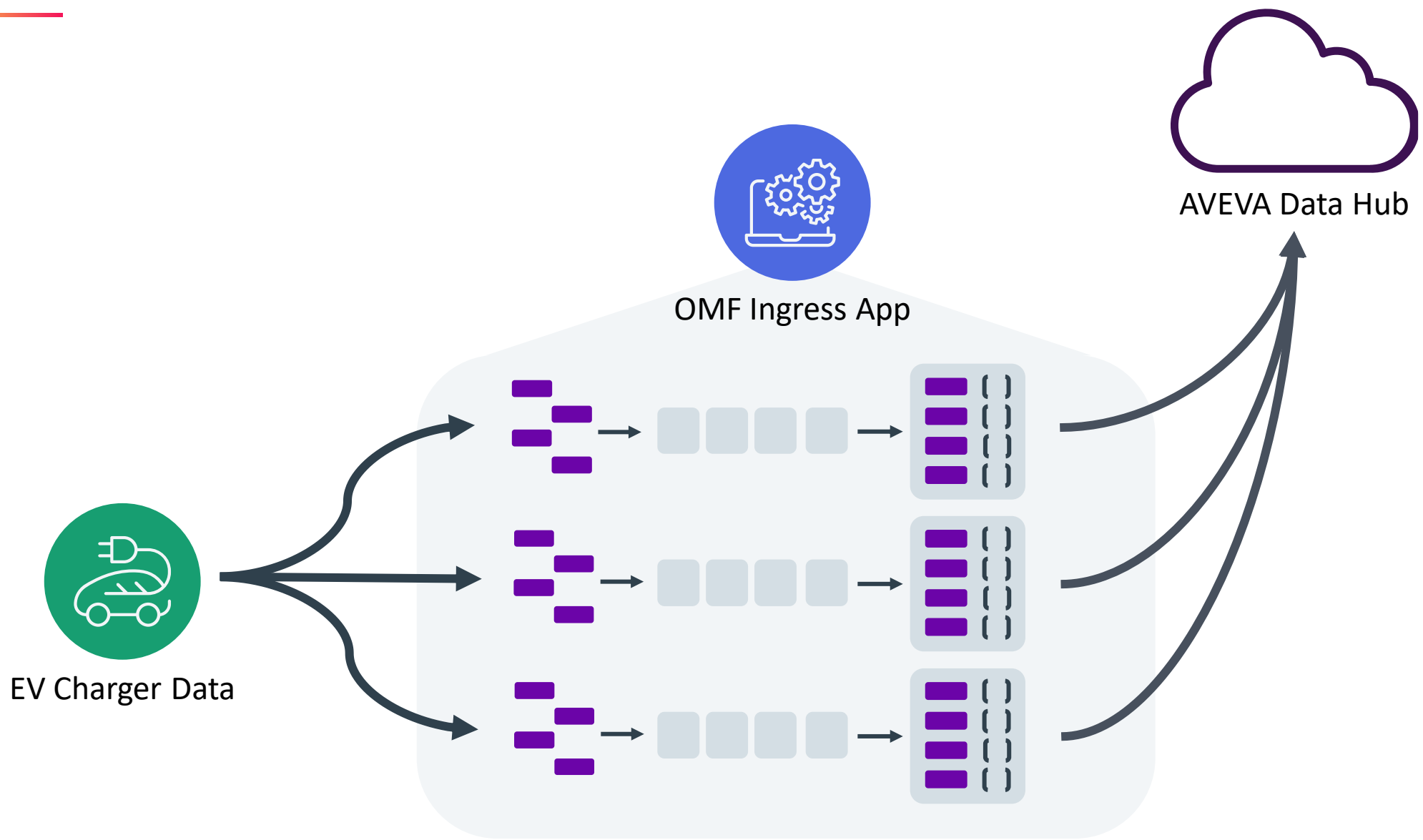


```
# Get or create types in data hub
response = omf_client.omfRequest(
    OMFMessageType.Type, OMFMessageAction.Create,
charger_data_client.getTypes())
omf_client.verifySuccessfulResponse(response, 'Error creating types')

# Get or create streams from data source in data hub
streams = charger_data_client.getStreams()
response = omf_client.omfRequest(
    OMFMessageType.Container, OMFMessageAction.Create, streams)
omf_client.verifySuccessfulResponse(response, 'Error creating
containers')
```



```
event: ChargerData = queue.pop()
if event.Id not in payload:
    payload[event.Id] = [
        {'Timestamp': event.Timestamp, 'Value': event.Value}]
else:
    payload[event.Id].append(
        {'Timestamp': event.Timestamp, 'Value': event.Value})
```

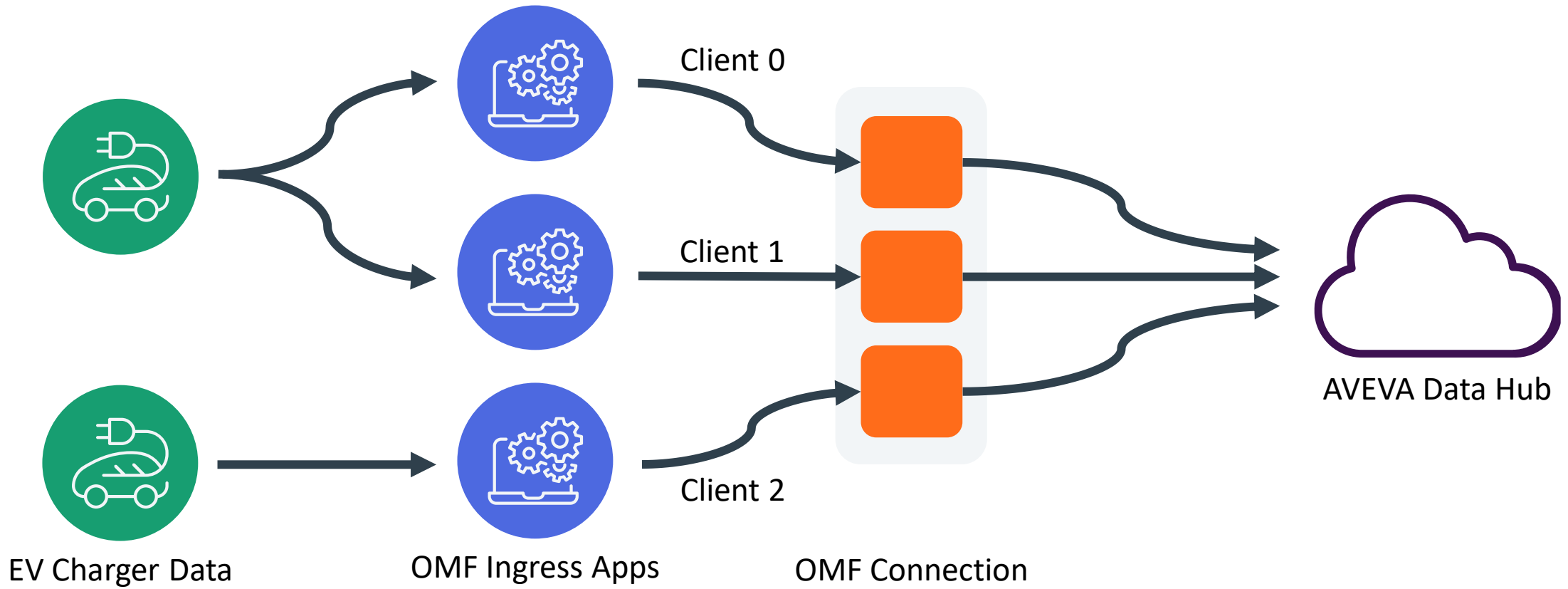


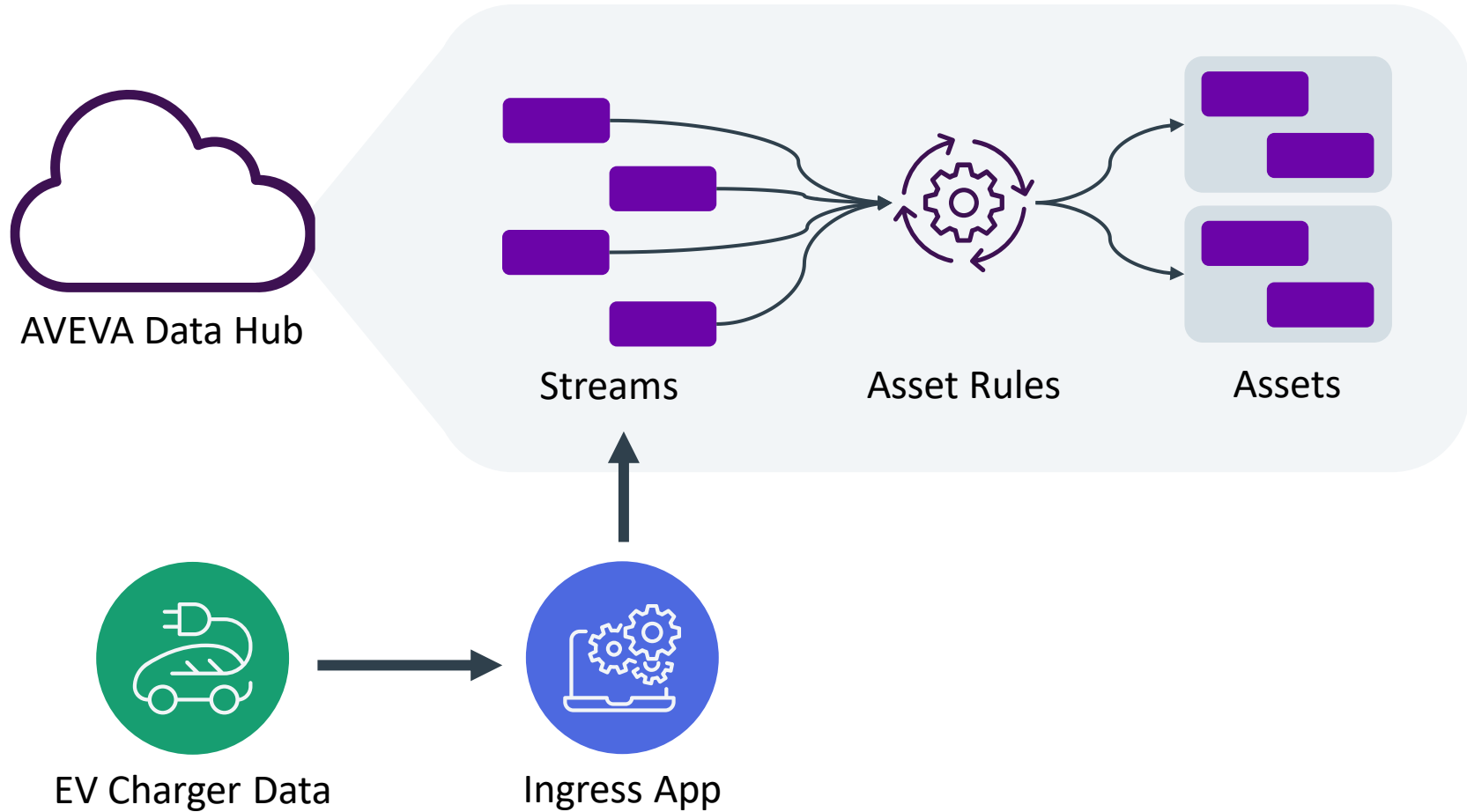
---

```
process_partitions = []
for i in range(0, len(streams), int(len(streams)/max_processes)):
    process_partitions.append(streams[i:i + int(len(streams)/max_processes)])

processes = []
for process_partition in process_partitions:
    process = Process(target=processManager, args=(appsettings, process_partition))
    processes.append(process)
    process.start()
```







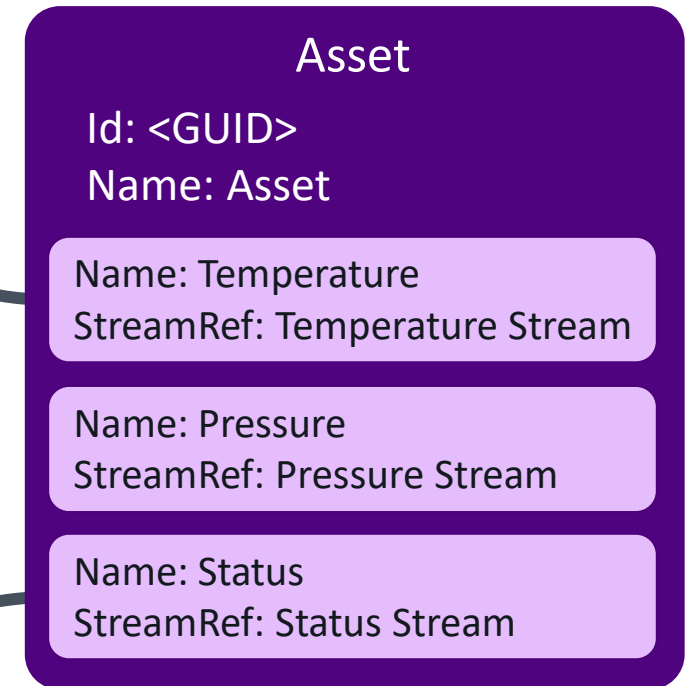
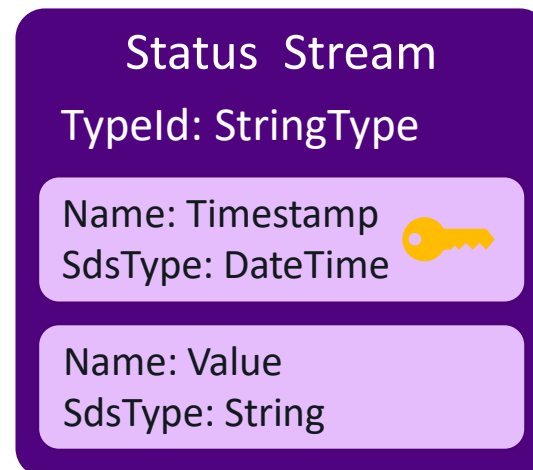
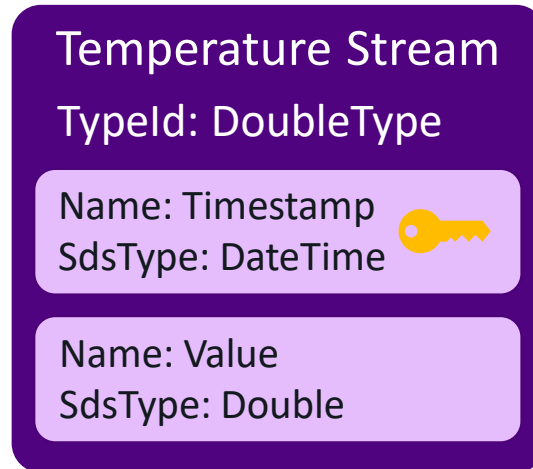
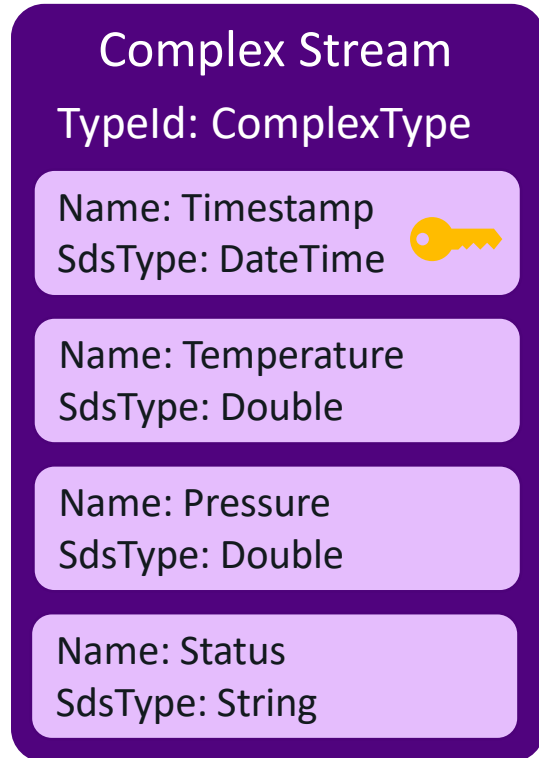
# AVEVA™ Data Hub ▸ Sequential Data Store

Streams ▾ Search for Streams 0061\*

- Filter Communities...
- Communities
  - AVEVA
  - AVEVA-Canvass
  - AVEVA-Canvass2
  - AVEVA-DSA
  - AVEVA-EnergyAI
  - AVEVA-ESS
  - AVEVA-IOTA
  - AVEVA-Rovisys
  - AVEVA-Taco
  - AVEVA-Tendeka
  - AVEVA-ZG
  - Big Buffalo Site
  - Demo Test Todd
  - EV Sustainability Calculations
  - FabioTestCommunity
  - HENN Data Sharing
  - HVAC
  - Internal Community
  - Lityx
  - My New Community

<input type="checkbox"/>	Name ↓	Id	Description	Type
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-TotalUniqueUsers	0061-02-03-13-C33-Tesla-TotalUniqueUsers		TimeIndexed.Int64
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-TotalSiteCost	0061-02-03-13-C33-Tesla-TotalSiteCost		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-TotalReturningUsers	0061-02-03-13-C33-Tesla-TotalReturningUsers		TimeIndexed.Int64
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-TotalMicrosessions	0061-02-03-13-C33-Tesla-TotalMicrosessions		TimeIndexed.Int64
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-TotalKwhDelivered	0061-02-03-13-C33-Tesla-TotalKwhDelivered		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-TotalCostActual	0061-02-03-13-C33-Tesla-TotalCostActual		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-Status	0061-02-03-13-C33-Tesla-Status		TimeIndexed.String
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-StartedSessionCount	0061-02-03-13-C33-Tesla-StartedSessionCount		TimeIndexed.Int64
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-PilotSignalAmps	0061-02-03-13-C33-Tesla-PilotSignalAmps		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-MaxKwhDelivered	0061-02-03-13-C33-Tesla-MaxKwhDelivered		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-GhgAvoidedLbs	0061-02-03-13-C33-Tesla-GhgAvoidedLbs		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-GasolineAvoidedLbs	0061-02-03-13-C33-Tesla-GasolineAvoidedLbs		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-CompletedSessionCount	0061-02-03-13-C33-Tesla-CompletedSessionCount		TimeIndexed.Int64
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-CommutedMilesAvoided	0061-02-03-13-C33-Tesla-CommutedMilesAvoided		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-AvgSessionIdleMin	0061-02-03-13-C33-Tesla-AvgSessionIdleMin		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-AvgSessionDurationMin	0061-02-03-13-C33-Tesla-AvgSessionDurationMin		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-AvgKwhDelivered	0061-02-03-13-C33-Tesla-AvgKwhDelivered		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-AvgChargeDurationMin	0061-02-03-13-C33-Tesla-AvgChargeDurationMin		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-12-C32-Webasto DX-TotalUniqueUsers	0061-02-03-12-C32-Webasto DX-TotalUniqueUsers		TimeIndexed.Int64

# Complex Types vs. Simple Types with Assets



# Naming Patterns

Stream Name: 0061-01-02-04-T1-Tesla-PilotSignalAmps

Asset Identifier  
Metadata  
Measurement

Stream Name: 0061-01-02-04-T1-Tesla-AvgKwhDelivered

Stream Name: 0061-01-02-04-T1-Tesla-Status

## Car Charger Asset

Id: ChargingStation\_0061\_01\_02\_04\_T1

Name: Charging Station T1

Metadata:

EVSE Type: Tesla

Site: 01

Site Account: 0061

Properties:

Name: Pilot Signal Amps

StreamId: 0061-01-02-04-T1-Tesla-PilotSignalAmps

Name: Average Kwh Delivered

StreamId: 0061-01-02-04-T1-Tesla-AvgKwhDelivered

Name: Status

StreamId: 0061-01-02-04-T1-Tesla-Status

Streams ▾ Search for Streams 0061\* X 🔍 + Add Stream ⋮

- Filter Communities...
- Communities
  - AVEVA
  - AVEVA-Canvass
  - AVEVA-Canvass2
  - AVEVA-DSA
  - AVEVA-EnergyAI
  - AVEVA-ESS
  - AVEVA-IOTA
  - AVEVA-Rovisys
  - AVEVA-Taco
  - AVEVA-Tendeka
  - AVEVA-ZG
  - Big Buffalo Site
  - Demo Test Todd
  - EV Sustainability Calculations
  - FabioTestCommunity
  - HENN Data Sharing
  - HVAC
  - Internal Community
  - Lityx
  - My New Community

<input type="checkbox"/>	Name ↓	Id	Description	Type
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-TotalUniqueUsers	0061-02-03-13-C33-Tesla-TotalUniqueUsers		TimeIndexed.Int64
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-TotalSiteCost	0061-02-03-13-C33-Tesla-TotalSiteCost		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-TotalReturningUsers	0061-02-03-13-C33-Tesla-TotalReturningUsers		TimeIndexed.Int64
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-TotalMicrosessions	0061-02-03-13-C33-Tesla-TotalMicrosessions		TimeIndexed.Int64
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-TotalKwhDelivered	0061-02-03-13-C33-Tesla-TotalKwhDelivered		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-TotalCostActual	0061-02-03-13-C33-Tesla-TotalCostActual		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-Status	0061-02-03-13-C33-Tesla-Status		TimeIndexed.String
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-StartedSessionCount	0061-02-03-13-C33-Tesla-StartedSessionCount		TimeIndexed.Int64
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-PilotSignalAmps	0061-02-03-13-C33-Tesla-PilotSignalAmps		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-MaxKwhDelivered	0061-02-03-13-C33-Tesla-MaxKwhDelivered		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-GhgAvoidedLbs	0061-02-03-13-C33-Tesla-GhgAvoidedLbs		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-GasolineAvoidedLbs	0061-02-03-13-C33-Tesla-GasolineAvoidedLbs		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-CompletedSessionCount	0061-02-03-13-C33-Tesla-CompletedSessionCount		TimeIndexed.Int64
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-CommutedMilesAvoided	0061-02-03-13-C33-Tesla-CommutedMilesAvoided		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-AvgSessionIdleMin	0061-02-03-13-C33-Tesla-AvgSessionIdleMin		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-AvgSessionDurationMin	0061-02-03-13-C33-Tesla-AvgSessionDurationMin		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-AvgKwhDelivered	0061-02-03-13-C33-Tesla-AvgKwhDelivered		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-13-C33-Tesla-AvgChargeDurationMin	0061-02-03-13-C33-Tesla-AvgChargeDurationMin		TimeIndexed.Double
<input type="checkbox"/>	0061-02-03-12-C32-Webasto DX-TotalUniqueUsers	0061-02-03-12-C32-Webasto DX-TotalUniqueUsers		TimeIndexed.Int64



← Back to Overview

Name: SLTC-EV-Chargers-2 | Asset Type: Charging Station

✎ Edit Rule Details

1 Extract Tokens

2 Map Tokens to Values

3 Asset Configuration

4 Asset Preview

Preview Information

Details Errors Settings

Statistics

Assets Processed	72
Streams Processed	44417
Streams Matched	1296
Execution Errors	0
Metadata	216
Stream References	1296

Progress

Status	Complete
Percent Done	100%

ID	Type	Name	Description	Site Account	Site	EVSE Type	Started Ses	Total Site C	Total Cost A	Total Return	Total Micro	Avg Session	Avg
ChargingSt...	Charging Station	Charging Station C06		Westlake	520 S El Ca...	Tesla	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C07		Westlake	520 S El Ca...	Webasto DX	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C08		Westlake	520 S El Ca...	Tesla	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C09		Westlake	520 S El Ca...	Webasto DX	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C10		Westlake	520 S El Ca...	Webasto DX	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C11		Westlake	520 S El Ca...	Tesla	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C12		Westlake	520 S El Ca...	Tesla	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C13		Westlake	520 S El Ca...	Webasto DX	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C14		Westlake	520 S El Ca...	Webasto DX	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C15		Westlake	520 S El Ca...	Tesla	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C16		Westlake	520 S El Ca...	Tesla	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C17		Westlake	520 S El Ca...	Webasto DX	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C18		Westlake	520 S El Ca...	Webasto DX	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C19		Westlake	520 S El Ca...	Tesla	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C20		Westlake	520 S El Ca...	Webasto DX	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-
ChargingSt...	Charging Station	Charging Station C21		Westlake	520 S El Ca...	Webasto DX	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-02-02...	0061-

Stream Id: 0061-02-02-12-C17-Webasto DX-AvgSessionDurationMin

Items per page: 50 1 - 50 of 72

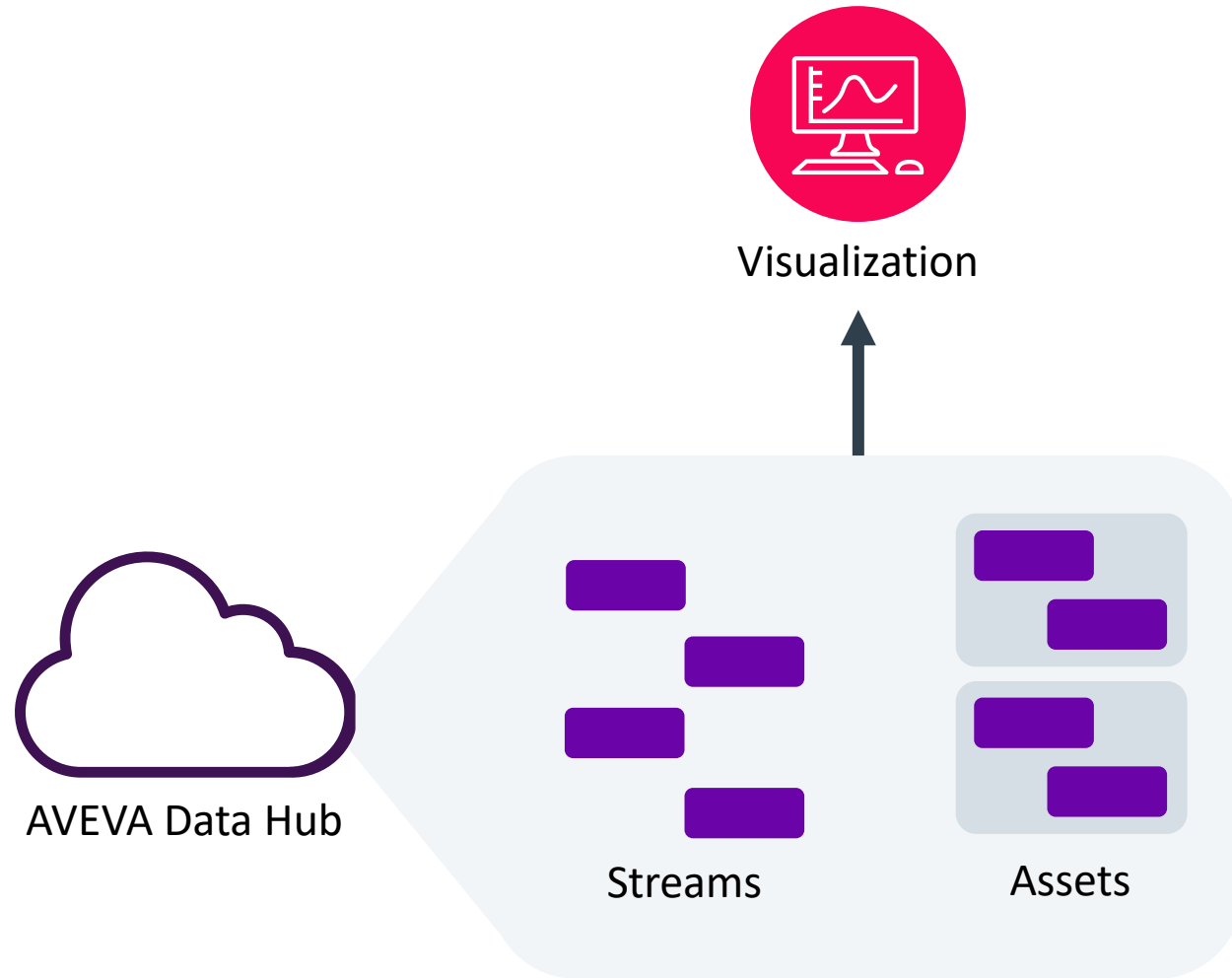
Cancel Back Save as Draft Save & Execute

---

# Creating a Custom Visualization







# Streamlit

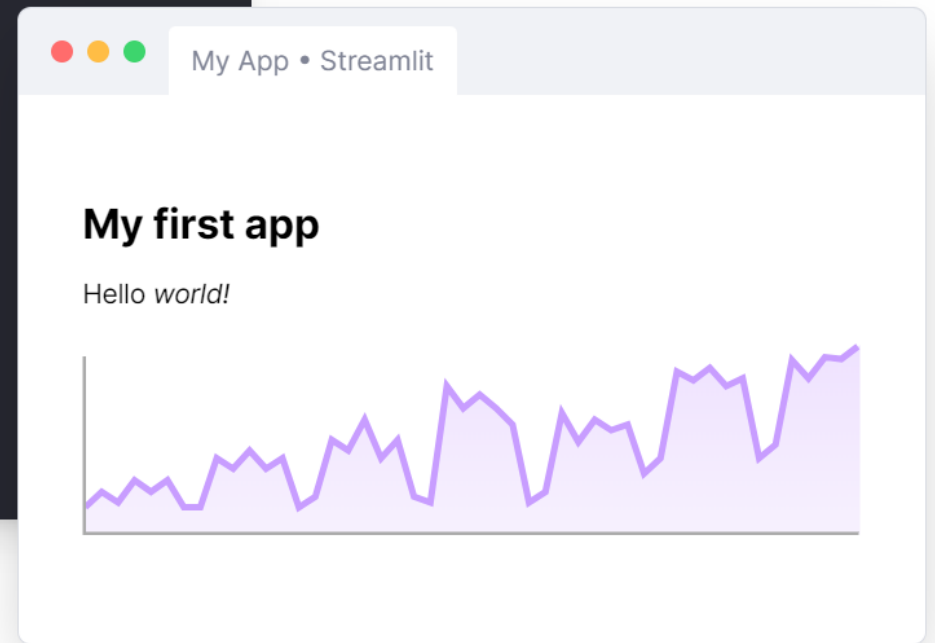
- A fast way to build/deploy web applications using Python
- Comes with built in components
- The community creates and shares custom components
- Useful for creating data visualizations/dashboards

```
MyApp.py

import streamlit as st
import pandas as pd

st.write("""
# My first app
Hello *world!*
""")

df = pd.read_csv("my_data.csv")
st.line_chart(df)
```



Home

Car Charger Dashboard

# Chargers Overview

Charging Station C01	Charging Station C02	Charging Station C03	Charging Station C04	Charging Station C05	Charging Station C06	Charging Station C07	Charging Station C08	Charging Station C09
Charging Station C10	Charging Station C11	Charging Station C12	Charging Station C13	Charging Station C14	Charging Station C15	Charging Station C16	Charging Station C17	Charging Station C18
Charging Station C19	Charging Station C20	Charging Station C21	Charging Station C22	Charging Station C23	Charging Station C24	Charging Station C26	Charging Station C27	Charging Station C28
Charging Station C29	Charging Station C29	Charging Station C30	Charging Station C31	Charging Station C32	Charging Station C33	Charging Station H1	Charging Station H2	Charging Station H3
Charging Station T1	Charging Station T10	Charging Station T11	Charging Station T12	Charging Station T2	Charging Station T3	Charging Station T4	Charging Station T5	Charging Station T6
Charging Station T7	Charging Station T8	Charging Station T9	Charging Station W1	Charging Station W10	Charging Station W11	Charging Station W12	Charging Station W13	Charging Station W14
Charging Station W15	Charging Station W16	Charging Station W17	Charging Station W18	Charging Station W19	Charging Station W2	Charging Station W20	Charging Station W21	Charging Station W22
Charging Station W23	Charging Station W24	Charging Station W3	Charging Station W4	Charging Station W5	Charging Station W6	Charging Station W7	Charging Station W8	Charging Station W9

UNPLUGGED  ADAPTIVE  IDLE  NOT CHARGING  DISABLED\_CHARGER

---

```
resource = st.secrets['resource']
api_version = st.secrets['api_version']
tenant_id = st.secrets['tenant_id']
namespace_id = st.secrets['namespace_id']

# Read client from session state or create one if it does not already exist
if 'client' in st.session_state:
    client = st.session_state['client']
else:
    client_id = st.secrets['client_id']
    client = ADHClient(api_version, tenant_id, resource,
                      client_id, accept_verbosity=True)
    st.session_state['client'] = client
```

---

```
# Set metrics data
```

```
charger_data = client.Assets.getAssetLastData(  
    namespace_id,  
    selected_charger.Id  
).Results
```

```
...
```

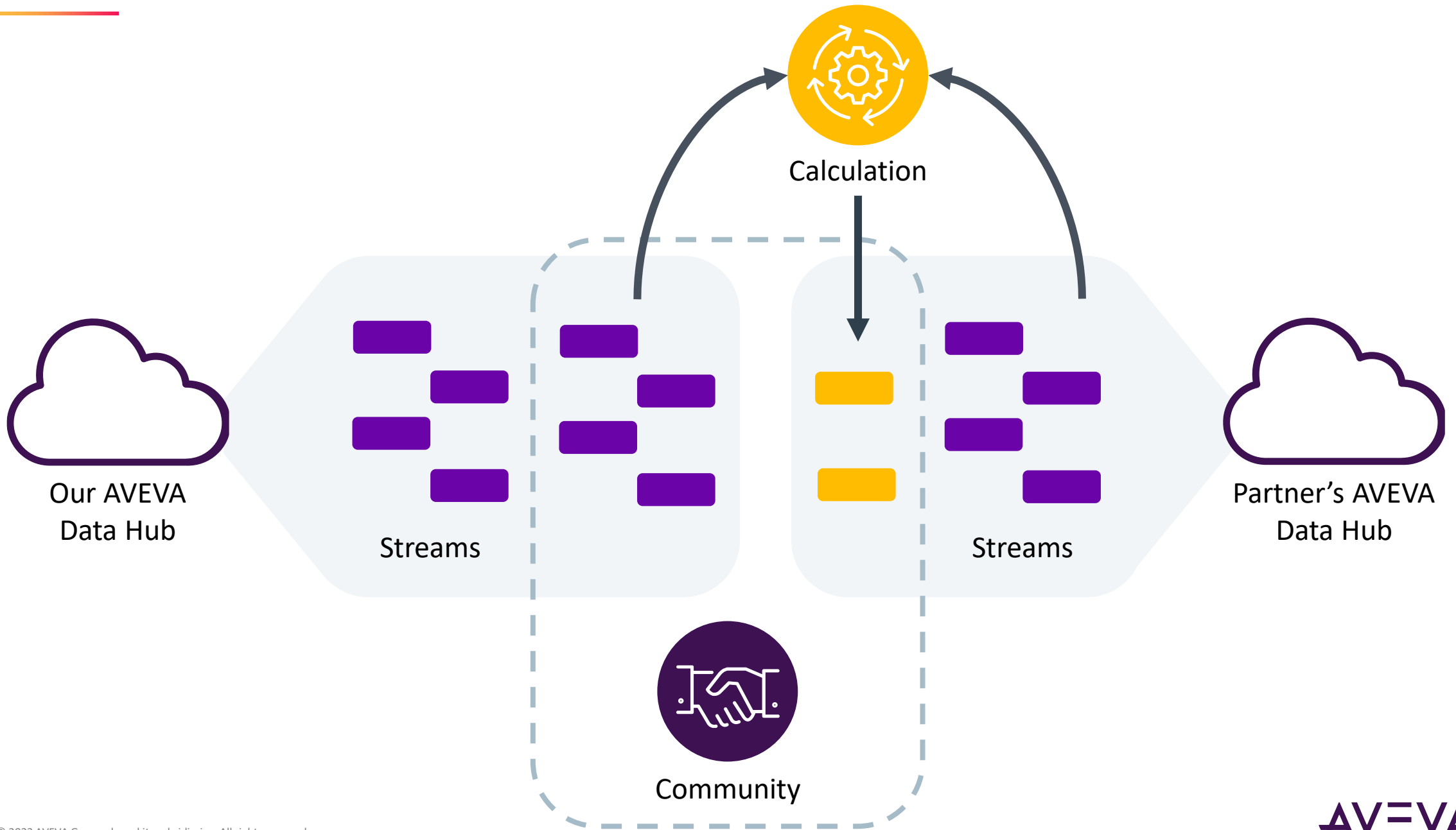
```
# Set trend data
```

```
raw_data = client.Assets.getAssetSampledData(namespace_id, selected_charger.Id,  
start_index=(  
    datetime.utcnow() - timedelta(hours=2)).isoformat(),  
end_index=datetime.utcnow().isoformat(), intervals=350).Results  
amps_data_frame = pd.DataFrame.from_dict(raw_data["PilotSignalAmps"])  
amps_data_frame['Timestamp'] = pd.to_datetime(amps_data_frame['Timestamp'])
```

---

# Working with Partner to Perform Calculation

**AVEVA**



- Home
- Data Management** >
- Data Collection >
- Visualization >
- Analytics >
- Security >
- Developer Tools >
- Support >

← Back to Communities ⋮

### EV Sustainability Calculations

74d13d01-f98e-4c77-9c0e-ae092bd68a2

✔ All Tenants Active  
Created Sep 14, 2023

How do I share data with this community?

Tenants      My Members      My Groups      My Clients      Usage

Filter tenants... [View Streams](#)

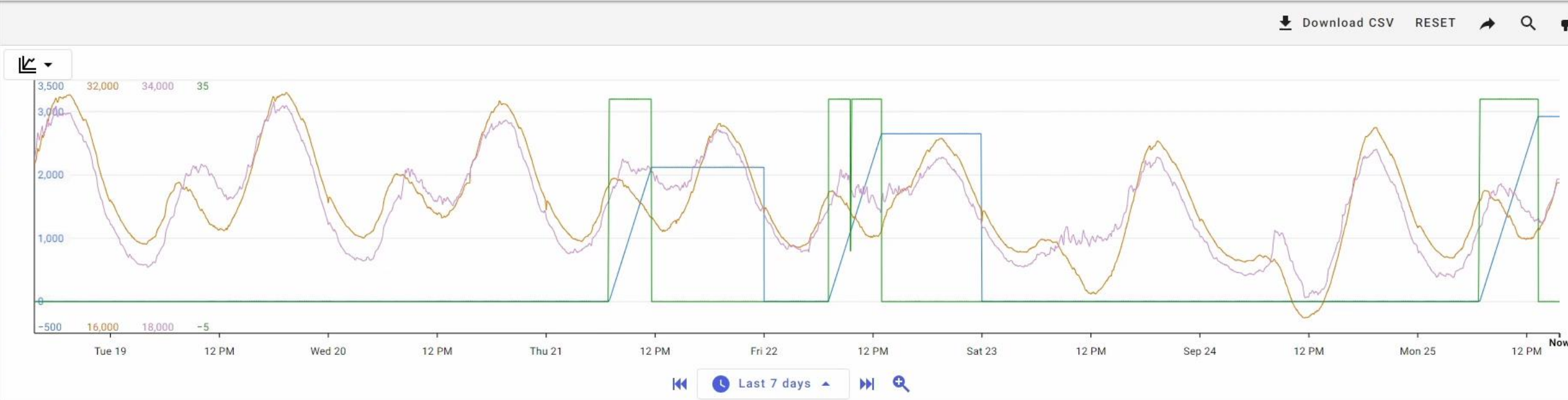
**Administrative Tenant**

<input type="checkbox"/>	Name ↑	Status	Users (43)	Clients (1)	Streams (1369)	
<input type="checkbox"/>	Americas-ADH-LH1	✔ Active	34	0	1297	
<input type="checkbox"/>	DataHubProductionSandbox1	✔ Active	9	1	72	


COLLAPSE <



AVEVA™ Data Hub | Trend



Name	Timestamp	Value	UOM	Min	Max	Avg	Total
0061-02-02-06-C11-Tesla-TotalizedDailySavedEmissions	Sep 25, 2023, 3:29:00 PM	2,925.261		0	2,925.261	504.182	5,060,209...
CAISO-TotalDemand	Sep 25, 2023, 3:20:00 PM	25,514		16,971	31,214	24,255.543	48,848,716...
CAISO-TotalSupply	Sep 25, 2023, 3:20:00 PM	27,744		20,244	32,624	26,110.701	52,584,258...
0061-02-02-06-C11-Tesla-PilotSignalAmps	Sep 25, 2023, 3:37:00 PM	0		0	32	3.226	32,520



```
public class SdsStream
{
    public string Id { get; set; }
    public string Name { get; set; }
    public string Description { get; set; }
    public string TypeId { get; set; }
    public IList<SdsStreamIndex> Indexes { get; set; }
    public SdsInterpolationMode? InterpolationMode { get; set; }
    public SdsExtrapolationMode? ExtrapolationMode { get; set; }
    public IList<SdsStreamPropertyOverride> PropertyOverrides { get; set; }
}
```



```
private static readonly string TENANT_ID
    = Environment.GetEnvironmentVariable("ADH_PROD_TENANT_ID");
private static readonly string NAMESPACE_ID
    = Environment.GetEnvironmentVariable("ADH_PROD_NAMESPACE_ID");
private static readonly string COMMUNITY_ID
    = Environment.GetEnvironmentVariable("ADH_PROD_COMMUNITY_ID");
private static readonly string CLIENT_ID
    = Environment.GetEnvironmentVariable("ADH_PROD_CLIENT_ID");
private static readonly string CLIENT_SECRET
    = Environment.GetEnvironmentVariable("ADH_PROD_CLIENT_SECRET");
```

```
double energy = (chargerAmps.ElementAt(i + 1).Value +
chargerAmps.ElementAt(i).Value) / 2 *
(chargerAmps.ElementAt(i + 1).Timestamp -
chargerAmps.ElementAt(i).Timestamp).TotalSeconds *
CHARGE_VOLTAGE * WATTSECONDS_TO_MEGAWATTHOURS; // MWh
double emissions = energy * totalEmissions.ElementAt(i).Value
/ demand.ElementAt(i).Value * MEGATONS_TO_KG; // Kg
double savedEmissions = energy * ENERGY_TO_SAVED_EMISSIONS;
// Kg

double netSavedEmissions = savedEmissions - emissions;
totalizedSavedEmissions += netSavedEmissions;
```




```
// Retrieve charger amperage data
using (HttpRequestMessage request = new (HttpMethod.Get, new
Uri($"{stream.Self}/Data/Interpolated?startIndex={sliceStartIndex.ToString(T_FORM
AT)}&endIndex={sliceEndIndex.ToString(T_FORMAT)}&count={count}"))))
{
    request.Headers.Add("Community-Id", COMMUNITY_ID);
    response = await httpClient.SendAsync(request).ConfigureAwait(false);
}
```

---

```
// Retrieve electricity supply data
response = await httpClient.GetAsync(
    new Uri($"api/v1/Tenants/{TENANT_ID}/Namespaces/{NAMESPACE_ID}/Streams/CAISO-
TotalDemand/Data/Interpolated?startIndex={sliceStartIndex.ToString(T_FORMAT)}&endIndex={
sliceEndIndex.ToString(T_FORMAT)}&count={count}", UriKind.Relative));
CheckIfResponseWasSuccessful(response);

IEnumerable<TimeIndexedDouble> demand =
    JsonConvert.DeserializeObject<IEnumerable<TimeIndexedDouble>>(await
response.Content.ReadAsStringAsync());
```



```
response = await httpClient.PutAsync(  
    new  
    Uri($"api/v1/Tenants/{TENANT_ID}/Namespaces/{NAMESPACE_ID}/Streams/{outputStreamId}/Data  
    ", UriKind.Relative),  
    content1)  
    .ConfigureAwait(false);  
CheckIfResponseWasSuccessful(response);
```



---

## Collin Bardini

Senior Software Developer - Product Readiness

- AVEVA
- [collin.bardini@aveva.com](mailto:collin.bardini@aveva.com)



## Konstantin Chudnovskiy

Software Architect

- AVEVA
- [konstantin.chudnovskiy@aveva.com](mailto:konstantin.chudnovskiy@aveva.com)



This presentation may include predictions, estimates, intentions, beliefs and other statements that are or may be construed as being forward-looking. While these forward-looking statements represent our current judgment on what the future holds, they are subject to risks and uncertainties that could result in actual outcomes differing materially from those projected in these statements. No statement contained herein constitutes a commitment by AVEVA to perform any particular action or to deliver any particular product or product features. Readers are cautioned not to place undue reliance on these forward-looking statements, which reflect our opinions only as of the date of this presentation.

The Company shall not be obliged to disclose any revision to these forward-looking statements to reflect events or circumstances occurring after the date on which they are made or to reflect the occurrence of future events.

 [linkedin.com/company/aveva](https://www.linkedin.com/company/aveva)

 [@avevagroup](https://twitter.com/avevagroup)

#### ABOUT AVEVA

AVEVA is a world leader in industrial software, providing engineering and operational solutions across multiple industries, including oil and gas, chemical, pharmaceutical, power and utilities, marine, renewables, and food and beverage. Our agnostic and open architecture helps organizations design, build, operate, maintain and optimize the complete lifecycle of complex industrial assets, from production plants and offshore platforms to manufactured consumer goods.

Over 20,000 enterprises in over 100 countries rely on AVEVA to help them deliver life's essentials: safe and reliable energy, food, medicines, infrastructure and more. By connecting people with trusted information and AI-enriched insights, AVEVA enables teams to engineer efficiently and optimize operations, driving growth and sustainability.

Named as one of the world's most innovative companies, AVEVA supports customers with open solutions and the expertise of more than 6,400 employees, 5,000 partners and 5,700 certified developers. The company is headquartered in Cambridge, UK.

Learn more at [www.aveva.com](https://www.aveva.com)