Making PI Data Ingress c**OMF**ortable with PI Web API

Presented by: Max Drexel

©2020 OSIsoft, LLC

Brought to you by

- **Max Drexel**
- mdrexel@osisoft.com
- Software Developer
- PI Web API & OMF

©2020 OSIsoft, LLC

Set the stage

- **Who:** A PI System pro that isn't afraid to get their hands dirty
- **What:** A data source that isn't connected to your on-prem PI System
- **Where:** At the edge, with limited compute resources
- **When:** No better OSI-provided solution exists (like a connector)
- **Why:** We want that data, and we're willing to write the code ourselves

©2020 OSIsoft, LLC

The data source

©2020 OSIsoft, LLC

The options?

- Interface/Connector/Adapter
  - We're assuming one doesn't exist; if it does, we would use it
- AFSDK
  - Depends on Windows, which may not be available at the edge
- PI Web API
  - Designed more for reading than writing; performance concerns
- OSIsoft Message Format (OMF)
  - Tailor made for this sort of scenario

©2020 OSIsoft, LLC

Why OMF?

- Designed for high performance data writes
- Low client overhead
  - If you can send HTTPS requests, you're already OMF-compatible
- Future-proofs your application – with no changes, you can write to:
  - OSIsoft Cloud Services
  - Edge Data Store
  - Your on-prem PI System

©2020 OSIsoft, LLC

A brief OMF refresher

- Stands for OSIsoft Message Format
- A specification that abstracts out the backend
  - We don't communicate in terms of Data Archive PI Points, or OSIsoft Cloud Service Streams
  - As long as the backend accepts our messages and responds correctly, we can assume everything is working correctly
- Designed purely for ingress (writing) to OSI products
  - Simple to use
  - Edge-friendly
  - Performant

©2020 OSIsoft, LLC

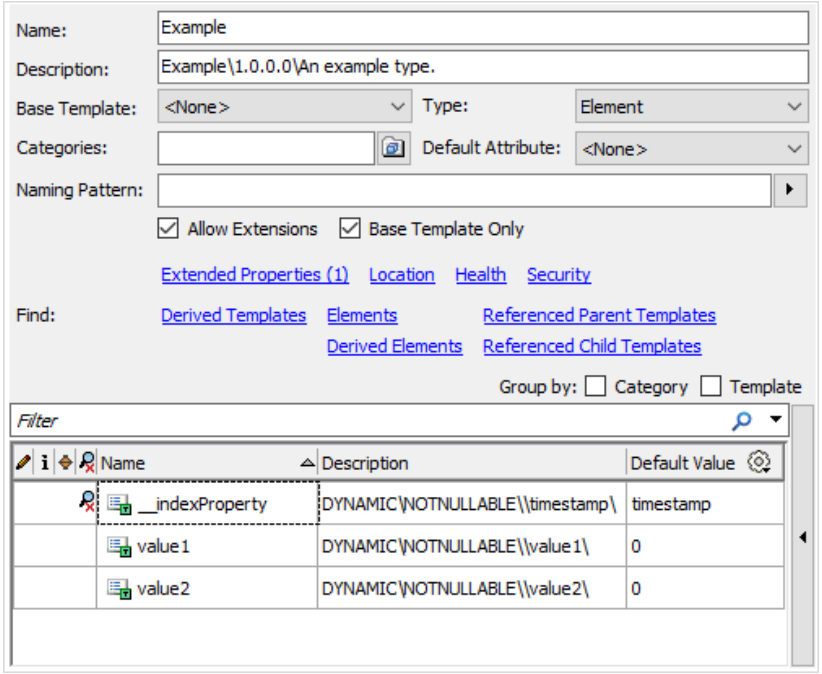
OMF and the PI System

- OMF clients send requests containing “messages”
- The type of the message controls what kind of PI System resource we’ll interact with:
  - TYPE messages: AF Element Templates
  - CONTAINER messages: Groups of PI Points
  - DATA messages
    - STATIC DATA: AF Elements
    - DYNAMIC DATA: Time-series data
    - LINK DATA: AF hierarchy/PI Point references

©2020 OSIsoft, LLC

TYPEs

id	classification	description	properties
ExampleType	dynamic	An example type.	timestamp: type = string, format = date-time, isindex = true value1: type = number, format = float64 value2: type = integer, format = int32



©2020 OSIsoft, LLC

CONTAINERs

id	ExampleContainer
typeid	ExampleType
name	Example container.
description	An example container.

Name	Stored Values	Point Source	Point Type	Point Class	Descriptor
ExampleContainer.value1	Real-time data	PIWebAPI_OMF	Float64	classic	An example container.
ExampleContainer.value2	Real-time data	PIWebAPI_OMF	Int32	classic	An example container.

General | Archive | Classic | Security | System

Name: ExampleContainer.value1 Rename Server: RESTUNIT

Descriptor: An example container.

Stored Values: Real-time data Point Source: PIWebAPI_O Point Class: classic

Point Type: Float64 Digital Set:

Eng Units: Display Digits: -5

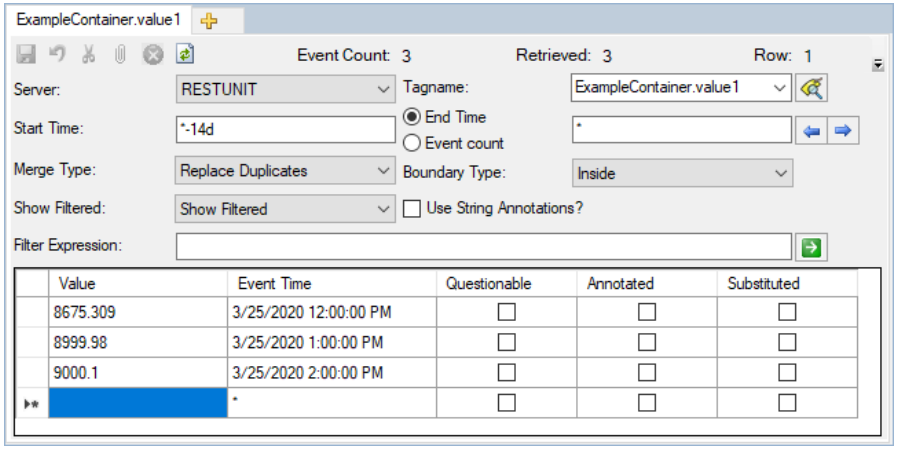
Exdesc: OmfPropertyIndexer=value1, OmfPropertyType=Number, OmfPropertyNullable=False, OmfPropertyF

Source Tag:

©2020 OSIsoft, LLC

DYNAMIC DATA

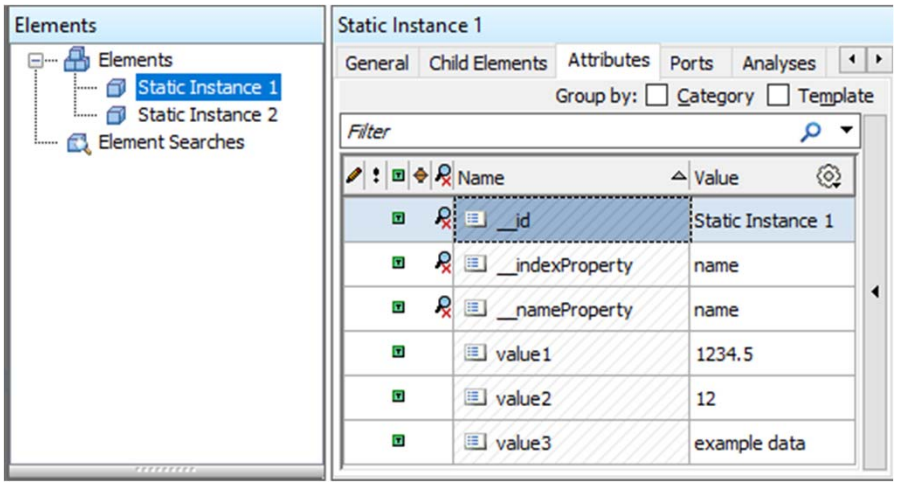
containerid	values		
	timestamp	value1	value2
ExampleContainer	2020-03-25T12:00:00Z	8675.309	40
	2020-03-25T13:00:00Z	8999.98	41
	2020-03-25T14:00:00Z	9000.1	42



©2020 OSIsoft, LLC

STATIC DATA

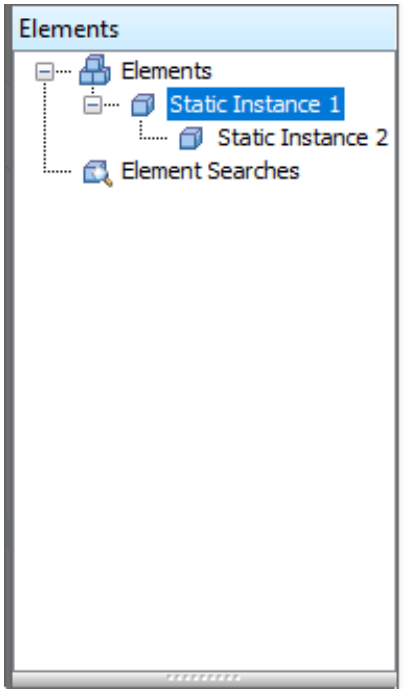
typeid	values			
DifferentType	name	value1	value2	value3
	Static Instance 1	1234.5	12	example data
	Static Instance 2	678.9	34	more example data



©2020 OSIsoft, LLC

LINK DATA (Static)

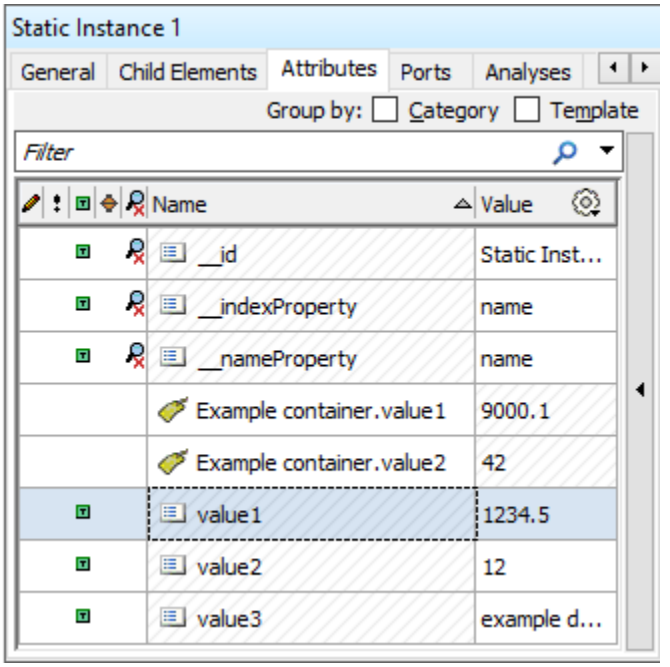
typeid	values	
	source	target
__link	typeid: DifferentType index: Static Instance 1	typeid: DifferentType index: Static Instance 2



©2020 OSIsoft, LLC

LINK DATA (Dynamic)

typeid	values	
	source	target
__link	typeid: DifferentType index: Static Instance 1	containerid: ExampleContainer



©2020 OSIsoft, LLC

Our data source's OMF representation

- We need to define our data source in terms of OMF: TYPE, CONTAINER, and DATA messages
- Our containers are created using types
- Our time-series data is stored in containers
- It's easiest to start from the TYPE, and work our way down

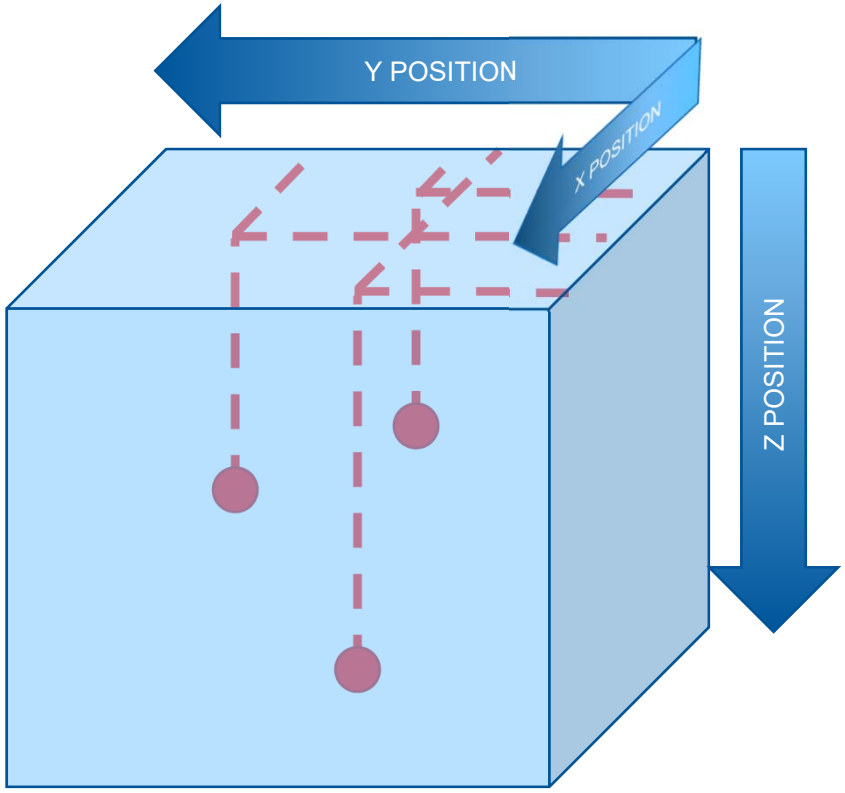
©2020 OSIsoft, LLC

Things to identify

- The usual business:
  - What data do we want to collect?
  - Which pieces of data are static (asset info) vs. dynamic (time-series)?
  - What resulting asset hierarchy do we want?
  - What data types are associated with the data (float32, int32, etc.)?
- New OMF-specific question: what pieces of data are collected together?

©2020 OSIsoft, LLC

What does “collected together” mean?



- Each measurement includes:
  - A timestamp
  - The X position
  - The Y position
  - The Z position
  - Whatever measurements we’re making at that coordinate (temperature, pressure, etc.)
- These measurements are sampled together, and give context to each other

 ©2020 OSIsoft, LLC

Why does “collected together” matter?

- An OMF container is a group of measurements that are sampled together, like the example we just saw
- Some OMF backends, like OCS, have much richer support for these situations – for example, you could query for data by the depth
- It still matters for an on-prem PI System:
  - Performance considerations
  - Data quality concerns

©2020 OSIsoft, LLC

Examples

Data Sources & Containers

©2020 OSIsoft, LLC

Example 1: Accelerometer

- Measures instantaneous acceleration in 3 dimensions
- We want to sample the measured acceleration over time
- We also want to store the manufacturer's name, serial number, and the model number of the sensor

©2020 OSIsoft, LLC

Example 1: OMF Representation

- Static Type
  - **Serial**
  - ManufacturerName
  - Model
- Dynamic Type
  - **Timestamp**
  - Accel_X
  - Accel_Y
  - Accel_Z

AccelerometerStatic	
String	**(index)**
String	
String	

AccelerometerDynamic	
Date-Time	**(index)**
Float32	$\frac{m}{s^2}$
Float32	⋮
Float32	⋮

©2020 OSIsoft, LLC

Example 2: Thermal camera

- Measures absolute temperature in an eight-pixel by eight-pixel grid
- We want to sample the temperature measurements over time
- We also want to store the current focal length, and the last time the camera was calibrated

©2020 OSIsoft, LLC

Example 2: OMF Representation

- Dynamic Type
  - **Timestamp**
  - Temperature1_1
  - Temperature1_2
  - Temperature1_N...
  - Temperature2_1
  - Temperature2_N...
  - TemperatureN_1…
  - TemperatureN_N

ThermalCamera_Temperature

Date-Time	**(index)**
Float32	Celsius
Float32	Celsius
Float32	Celsius
Float32	Celsius
Float32	Celsius
Float32	Celsius
Float32	Celsius

- Dynamic Type
  - **Timestamp**
  - FocalLength

ThermalCamera_FocalLength

Date-Time	**(index)**
Float32	millimeters

- Dynamic Type
  - **MaintenanceStartedAt**
  - MaintenanceDuration

ThermalCamera_Maintenance

Date-Time	**(index)**
UInt32	seconds

©2020 OSIsoft, LLC

Best Practices

PI Web API OMF Ingress Services

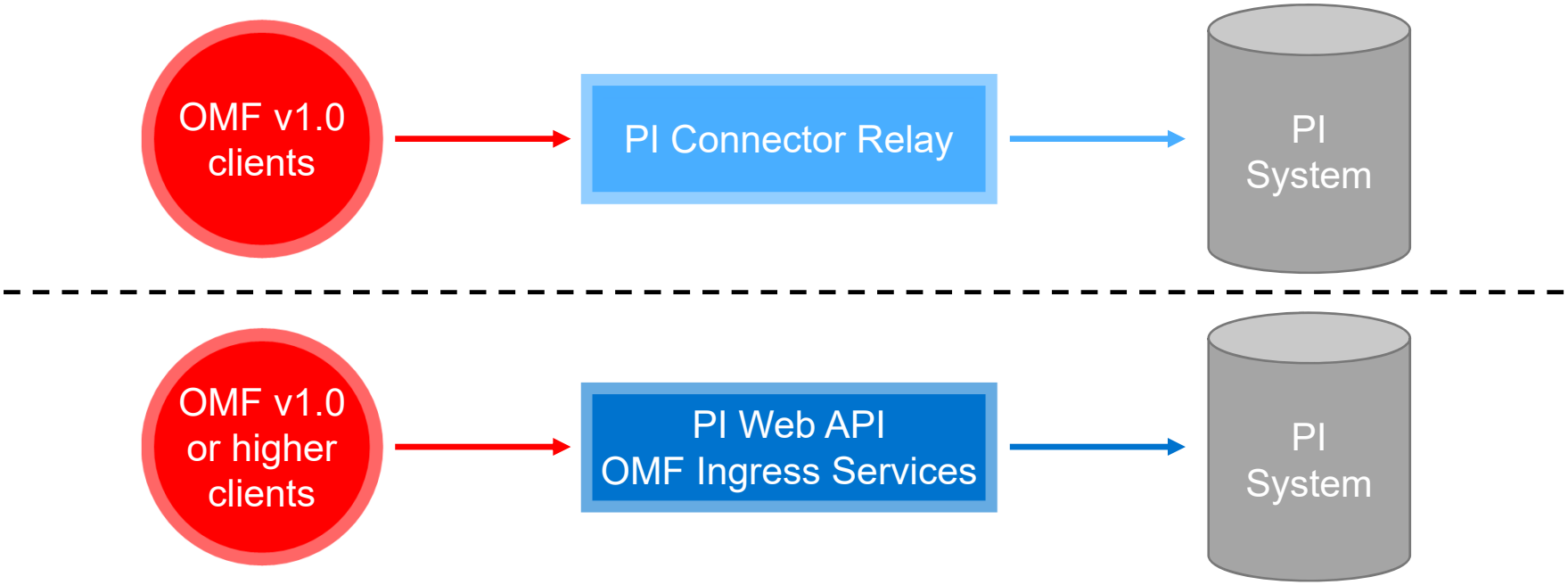
©2020 OSIsoft, LLC

Using PI Web API OMF Ingress Services

- OMF is backend agnostic, but we're working with an on-prem PI System
- The PI System supports OMF via the PI Web API
- Working directly with the PI Web API gets us some developer niceties:
  - PI Web API-specific response bodies
  - Error messages
  - Exception information
  - User-parameters
  - Warnings
- Our application **should not depend on these features!** They're specific to the PI Web API, and not all OMF services will support them!

©2020 OSIsoft, LLC

PI Server OMF Migration



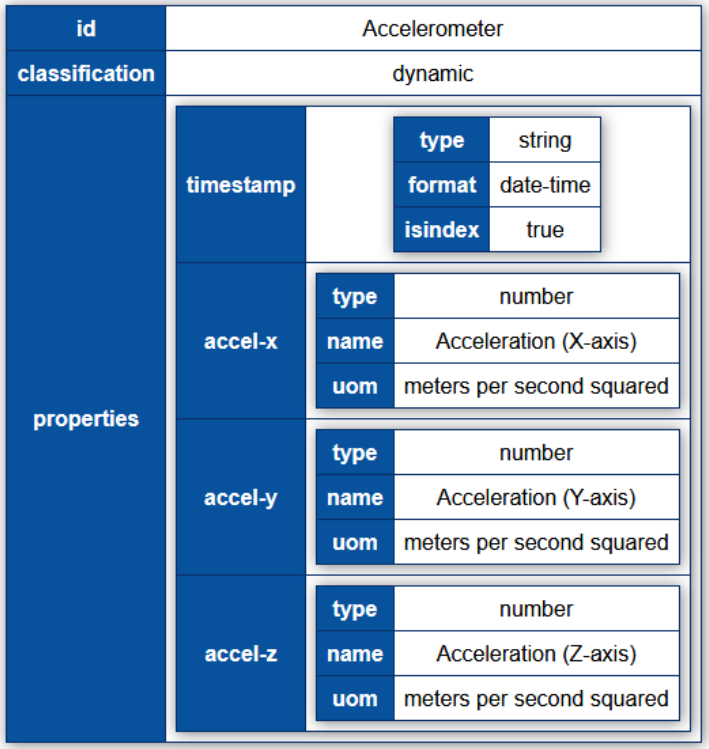
©2020 OSIsoft, LLC

Migrating to PI Web API

- PI Web API 2019 SP1 introduces built-in migrations
- When installing or upgrading to 2019 SP1, some resources created by earlier OMF gateways will be migrated
  - When upgrading from 2019, all existing resources can be used/migrated
  - When upgrading from the PI Connector Relay, compatible PI Points will be migrated
- **Migrations may cause unexpected side effects!**
  - For more information, check the PI Web API OMF Companion Guide

©2020 OSIsoft, LLC

PI Web API OMF response bodies



OperationId	08a4e14f-b05e-434c-9972-fcf1a398d5f2
Messages	

©2020 OSIsoft, LLC

PI Web API OMF response bodies

OperationId	fd03915c-8ae0-4095-9bbe-2cc07424da08

Messages

MessageIndex	Events	Status
0	See below	See below

Events

EventInfo	ExceptionInfo	Severity	InnerEvents
See below	null	Warning	

EventInfo

Message	Attempted to create a duplicate type.
Reason	null
Suggestions	
EventCode	4019

Parameters

Name	Value
TypeId	Accelerometer
TypeVersion	1.0.0.0

Status

Code	201
HighestSeverity	Warning

©2020 OSIsoft, LLC

PI Web API OMF response bodies

MessageIndex	Events				Status
	EventInfo	ExceptionInfo	Severity	InnerEvents	
0	Message: A type with the specified ID and version already exists. Reason: Types must have a unique ID and version. Suggestions: Delete the conflicting type. / Specify a type ID and type version combination not already in use. EventCode: 4011 Parameters: Name / Value — TypeId: Accelerometer; TypeVersion: 1.0.0.0	Type: OSIsoft.OMF.Loggers.OmfLoggableException Message: A type with the specified ID and version already exists.	Error		Code: 409 HighestSeverity: Error

©2020 OSIsoft, LLC

PI Web API OMF Event Codes

- Event codes uniquely identify the type of event that occurred
- Event codes are permanent
  - New ones may be added, or old ones removed, but an event codes' meaning will never be changed
- Useful for logging, or if you're posting a question to PI Square ☺

Event Code	Name	Severity Level	HTTP Status Co
1001	FeatureNotImplemented	Error	501
1002	FeatureNotSupported	Error	501
1003	StorageLayerTransactionsFailed	Error	500
1004	UnhandledException	Error	500
2001	MissingRequiredOmfHeader	Error	400
2002	DuplicateOmfHeaderSpecified	Error	400
2003	ActionNotValid	Error	400
2004	CompressionNotValid	Error	400
2005	MessageTypeNotValid	Error	400
2006	MessageFormatNotValid	Error	400
2007	OmfVersionNotValid	Error	400
3001	MessageDecompressionFailure	Error	400
3002	ParserException	Error	400
3003	ValueUnableToBeParsed	Error	400
3004	ValueOutOfRange	Error	400
3005	OmfFieldUntrimmedWhitespace	Error	400
3006	IllegalTypeFormatCombination	Error	400
3007	PropertyNamesMustBeUnique	Error	501
3008	FieldRequired	Error	400
3009	ReservedPrefixNotAllowed	Error	400
3010	TagCannotBeNull	Error	400

©2020 OSIsoft, LLC

Be aware of your state

- CREATE – Add or Assert
  - If a resource exists with the ID you were trying to use, the request will fail if the resource doesn't match what you were trying to create
- UPDATE – Insert or Replace
  - If a resource exists with the ID you were trying to use, it will be replaced
- DELETE – Assert and Remove
  - If a resource exists with the ID you were trying to use, *and* the resource matches what you were trying to delete, it will be removed

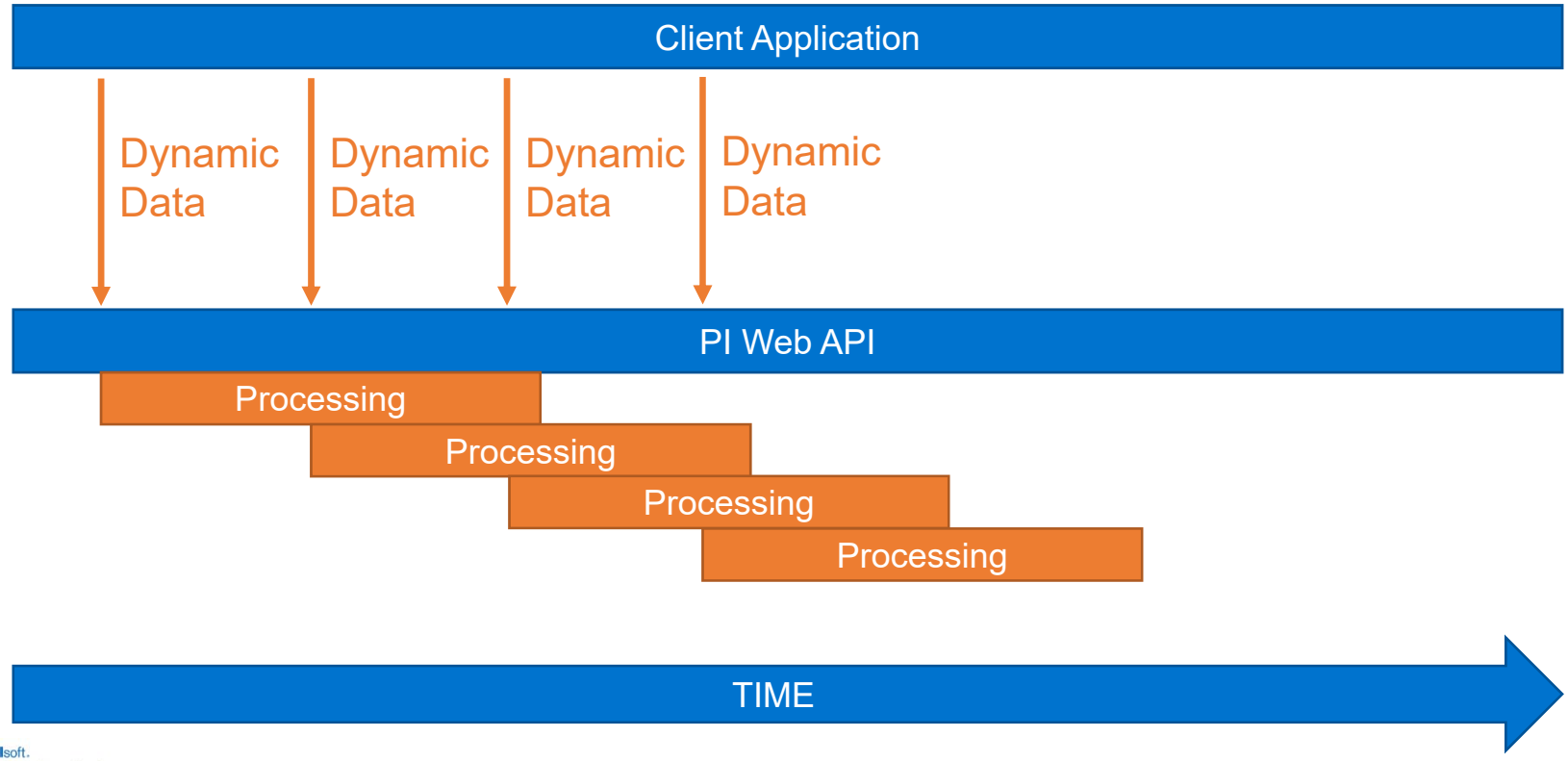
©2020 OSIsoft, LLC

Performance best practices

- *Avoid* mixing static and dynamic data
- Clients should use separate Windows identities
  - If you don't do this, increasing the number of clients can result in poor performance – we'll see why soon
- Batch your data messages
- Buffer your values

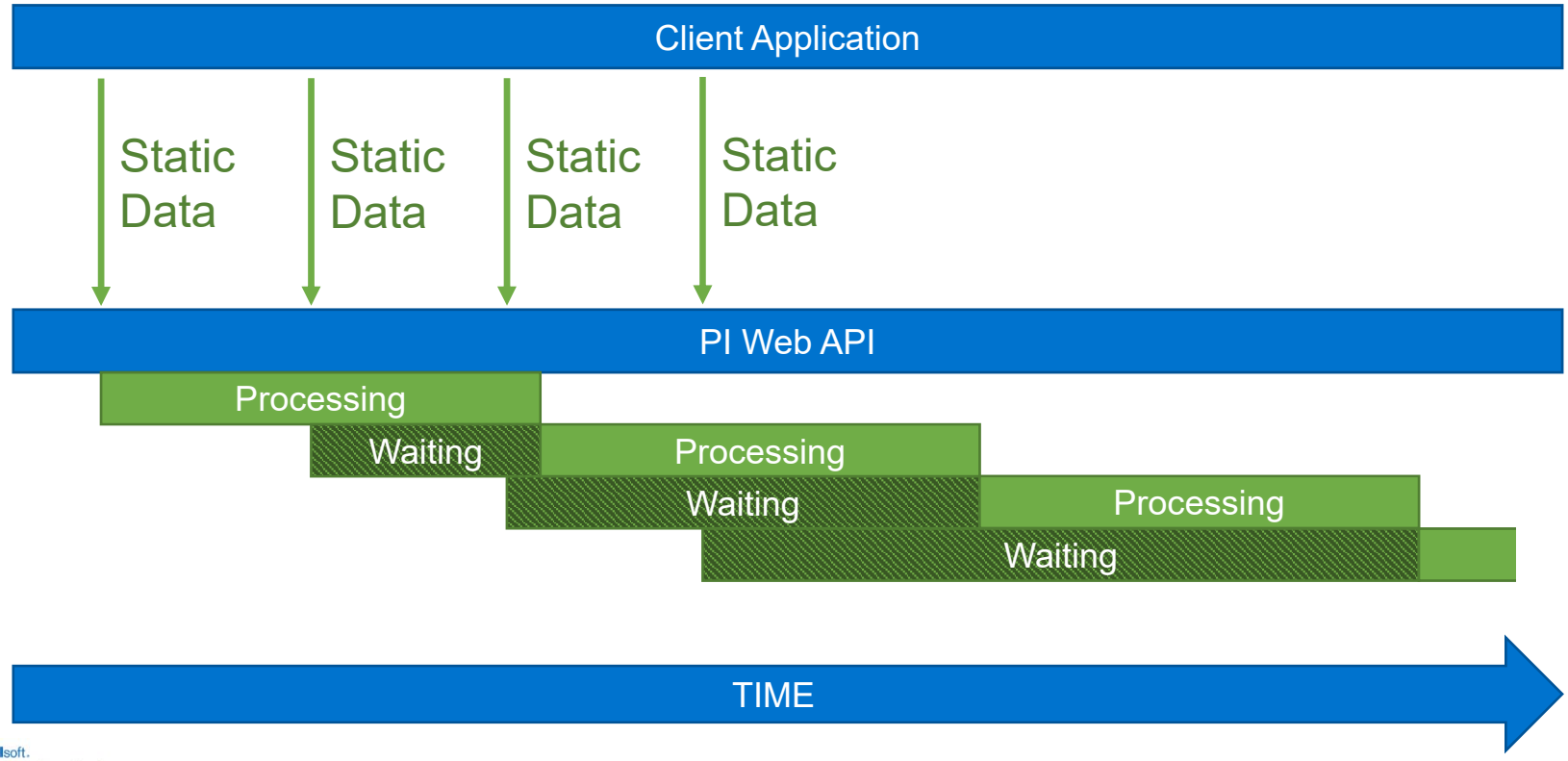
©2020 OSIsoft, LLC

Don’t mix Static and Dynamic Data



©2020 OSIsoft, LLC

Don’t mix Static and Dynamic Data



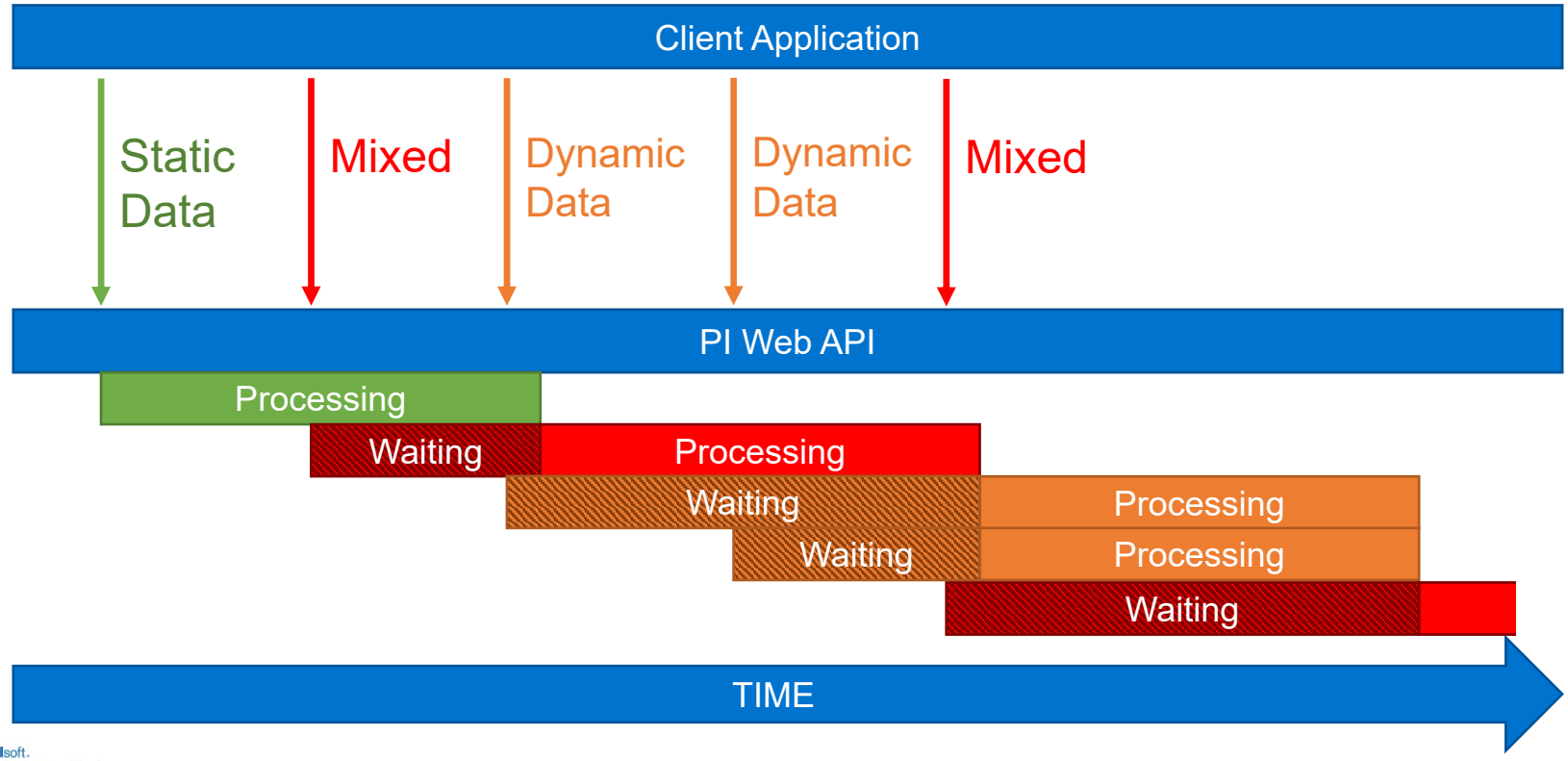
©2020 OSIsoft, LLC

Don’t mix Static and Dynamic Data

- Static data runs in Exclusive mode
- Dynamic data runs in Concurrent mode
- **A request that contains both static and dynamic data runs in Exclusive mode**
  - Don’t send static and dynamic data in the same request, because your dynamic data will be stuck waiting until it’s safe for the static data to start processing
- Try sending all your static data when your client starts up, and avoid sending it after that

©2020 OSIsoft, LLC

Don’t mix Static and Dynamic Data



©2020 OSIsoft, LLC

Using separate Windows identities

- All that scheduling we just saw is per-identity
- If we spread our requests across multiple Windows identities, we'll have less contention
- If you have independent data sources, using a separate Windows identity for each data source will give you the best throughput

©2020 OSIsoft, LLC

Data message batching

containerid	values			
Accelerometer1	timestamp	accel_x	accel_y	accel_z
	2020-03-25T12:00:00Z	0.232	0.142	0.034

containerid	values			
Accelerometer2	timestamp	accel_x	accel_y	accel_z
	2020-03-25T12:00:00Z	0.023	-0.047	0.266

containerid	values			
Accelerometer3	timestamp	accel_x	accel_y	accel_z
	2020-03-25T12:00:00Z	-1.842	1.145	0.785

containerid	values			
Accelerometer4	timestamp	accel_x	accel_y	accel_z
	2020-03-25T12:00:00Z	0.005	1.049	0.441

containerid	values			
Accelerometer1	timestamp	accel_x	accel_y	accel_z
	2020-03-25T12:00:00Z	0.232	0.142	0.034
Accelerometer2	timestamp	accel_x	accel_y	accel_z
	2020-03-25T12:00:00Z	0.023	-0.047	0.266
Accelerometer3	timestamp	accel_x	accel_y	accel_z
	2020-03-25T12:00:00Z	-1.842	1.145	0.785
Accelerometer4	timestamp	accel_x	accel_y	accel_z
	2020-03-25T12:00:00Z	0.005	1.049	0.441

©2020 OSIsoft, LLC

Buffering values

- We want a low overhead-to-data ratio
- Try to buffer data on the client
  - This may not be applicable to some clients due to resource constraints or data integrity concerns
- Overhead per OMF message is consistent:
  - Need to retrieve the container's PI Points
  - Need to retrieve the container's OMF Type
- If you buffer on the client and batch your writes, you'll see much better total throughput rates

©2020 OSIsoft, LLC

Fewer messages

containerid	values
Accelerometer1	timestamp: 2020-03-25T12:00:00Z, accel_x: 0.232, accel_y: 0.142, accel_z: 0.034
Accelerometer1	timestamp: 2020-03-25T12:00:10Z, accel_x: 0.432, accel_y: 0.513, accel_z: 0.122
Accelerometer1	timestamp: 2020-03-25T12:00:20Z, accel_x: 1.324, accel_y: 1.426, accel_z: -1.843
Accelerometer1	timestamp: 2020-03-25T12:00:30Z, accel_x: 2.534, accel_y: 1.985, accel_z: -2.953
Accelerometer1	timestamp: 2020-03-25T12:00:40Z, accel_x: -1.242, accel_y: -3.345, accel_z: 2.765

containerid	values
Accelerometer1	timestamp: 2020-03-25T12:00:00Z, accel_x: 0.232, accel_y: 0.142, accel_z: 0.034 timestamp: 2020-03-25T12:00:10Z, accel_x: 0.432, accel_y: 0.513, accel_z: 0.122 timestamp: 2020-03-25T12:00:20Z, accel_x: 1.324, accel_y: 1.426, accel_z: -1.843 timestamp: 2020-03-25T12:00:30Z, accel_x: 2.534, accel_y: 1.985, accel_z: -2.953 timestamp: 2020-03-25T12:00:40Z, accel_x: -1.242, accel_y: -3.345, accel_z: 2.765

©2020 OSIsoft, LLC

Optimal performance scenarios

- Performance factors to consider:
  - **Containers** - The number of **Containers** sent per data request
  - **Clients** - The number of **Clients** sending the **Containers**
  - **Properties** - The number of **Properties** (PI Points) per **Container**
  - **Values** - The number of **Values** sent for each **Property**

Containers	Client(s)	Properties	Values	Performance(events/s)
2000	10	5	1	>100k
1000	10	10	10	>200k
1000	15	10	10	>250k
100	20	10	10	>300k

©2020 OSIsoft, LLC

PI Web API OMF security best practices

- PI Web API needs to be configured for Basic authentication
- PI Web API CORS needs to be configured
- Assume CSRF defense is enabled
  - Clients should include the *X-Requested-With* header
- Check the PI Web API manual for information on how to configure these settings

©2020 OSIsoft, LLC

Miscellaneous tricks

- Set an appropriate logging verbosity by adding the *SeverityLevel* configuration attribute to your PI Web API configuration
- Turn on Debug mode when you're doing development (and make sure to turn it off once you're done)
- If possible, enable buffering on the PI Web API server machine
- Make sure your AF and DA servers are up to date

©2020 OSIsoft, LLC

Where can I get started?

- OMF Sample Code
  - https://github.com/osisoft/OSI-Samples-OMF
- OMF Specification
  - https://omf-docs.osisoft.com/en/v1.1/
- PI Web API OMF Companion Guide
  - https://livelibrary.osisoft.com/LiveLibrary/web/ui.xql?action=html&resource=publist_home.html&pub_category=OSIsoft-Message-Format-(OMF)
- OMF Performance Blog Post
  - https://pisquare.osisoft.com/people/jwu/blog/2019/09/18/guidance-on-writing-efficient-omf-applications-for-pi-web-api
- PI Web API Manual
  - https://livelibrary.osisoft.com/LiveLibrary/web/ui.xql?action=html&resource=publist_home.html&pub_category=PI-Web-API

©2020 OSIsoft, LLC

Questions?

Please wait for
the **microphone**

State your
name & company

Save the Date...

AMSTERDAM
October 26-29, 2020

OSIsoft. PI World

©2020 OSIsoft, LLC

©2020 OSIsoft, LLC