



AVEVA WORLD



AVEVA™ PI System™ and Graph Databases

Tyler Sontag
IT/OT Supervisor

Kevin Brashears
Senior Software Engineer





NORTH AMERICA

LOCATIONS

- Office
- Refinery
- Crude, Products or Asphalt Terminal
- Product Pipeline
- Crude Oil Pipeline
- Chemicals Pipeline

FHR Background

2020 - OSIsoft PI World San Francisco - Oil and Gas

Doctor PI: The democratization of the PI System



- PI World 2020 presentation

- Graph databases are a class of NoSQL databases that leverage flexible data-modeling with an emphasis on the relationship between entities.
- Examples use cases:
 - Asset Management information
 - Work orders
 - Purchase orders
 - Anomaly investigation systems
 - Financial systems



FHR AVEVA™ Statistics



Table Text Code	labelName	labelCount
1	"PIHistorian"	11
2	"PIPoint"	8835846
3	"PIAssetFrameworkServer"	13
4	"PIAssetFrameworkDatabase"	123
5	"PIAssetFrameworkElement"	361926
6	"PIAssetFrameworkElementTemplate"	8503
7	"PIAssetFrameworkAttribute"	18387268
8	"PIAssetFrameworkAnalysis"	934380
9	"PIAssetFrameworkEventFrame"	3364722
10	"PIAssetFrameworkEventFrameAttribute"	900

PISCES (PI System Change Events Service)

- .NET Windows application
 - 13 AF Servers
 - 11 PI Servers
- Use the “eventing” provided by the AF SDK
 - Real-time processing
- Process the metadata/contextual data changes (AF SDK to JSON) and send to FHR’s Event Bus
 - Note: not sending time-series data to Graph, only metadata
 - Template change will reprocess it’s derived
- Ingest (custom code) into FHR’s Graph database

AFElementTemplate

```
return new Dictionary<string, object>()
{
    { "allowExtend", elementTemplate.AllowElementToExtend },
    { "analysisTemplateIds", analysisTemplateIds },
    { "attributeTemplateIds", attributeTemplateIds },
    { "baseTemplate", IdNameDictionary(elementTemplate.BaseTemplate) },
    { "categories", IdNameDictionary(elementTemplate.Categories) },
    { "database", IdNameDictionary(elementTemplate.Database) },
    { "description",
string.IsNullOrEmpty(elementTemplate.Description) ? null :
elementTemplate.Description },
    { "id", elementTemplate.ID.ToString() },
    { "name", elementTemplate.Name },
    { "namingPattern",
string.IsNullOrWhiteSpace(elementTemplate.NamingPattern) ? null :
elementTemplate.NamingPattern },
    { "notificationRuleTemplateIds", notificationRuleTemplateIds },
    { "ports", AFPortsDictionary(elementTemplate.Ports) },
    { "security", elementTemplate.Security.GetSecurityString() },
    { "server", IdNameDictionary(elementTemplate.PISystem) },
    { "type", elementTemplate.Type.ToString() },
};
```

AFAttribute

```
return new Dictionary<string, object>()
{
    { "categories", IdNameDictionary(attribute.Categories) },
    { "configString", string.IsNullOrEmpty(attribute.ConfigString) ? null : attribute.ConfigString },
    { "database", IdNameDictionary(attribute.Database) },
    { "dataReference", attribute.DataReferencePlugIn?.Name ?? "None" },
    { "description", string.IsNullOrEmpty(attribute.Description) ? null : attribute.Description },
    { "displayDigits", attribute.DisplayDigits },
    { "element", IdNameDictionary(attribute.Element) },
    { "enumerationSet", enumerationSet },
    { "id", attribute.ID.ToString() },
    { "isConfigurationItem", attribute.IsConfigurationItem },
    { "isExcluded", attribute.IsExcluded },
    { "isHidden", attribute.IsHidden },
    { "name", attribute.Name },
    { "parent", attribute.Parent == null ? null :
        new Dictionary<string, string>()
        {
            { "id", attribute.Parent.ID.ToString() },
            { "name", attribute.Parent.Name },
        }
    },
    { "piPoint", pointResolved ? new Dictionary<string, object>()
        {
            { "id", pointID },
            { "name", pointName },
            { "serverId", pointServerId },
            { "server", pointServerName },
        } : null
    },
    { "server", IdNameDictionary(attribute.PISystem) },
    { "template", attribute.Template == null ? null :
        new Dictionary<string, string>()
        {
            { "id", attribute.Template.ID.ToString() },
            { "name", attribute.Template.Name },
        }
    },
    { "type", attribute.Type.Name },
    { "uom", attribute.DisplayUOM?.Abbreviation ?? null },
    { "value", value },
};
```

PISCES Functionality



Service

- Continuously monitors for changes, processes and sends to Event Bus
 - `PIServer.FindChangedPIPoints()`
 - `PISystem.FindChangedItems()`
- Persists provided cookie support, to not miss updates
- Periodically sends static Identities (SecurityMapping, etc.)
- Periodically sends PI point current values

Console

- Allows for one-time extracts
 - Priming
 - Fix missing/incorrect data
- Can reprocess any AFObject that the Service monitors

```
Windows PowerShell
PS C:\Repos\Pisces\src\Pisces.Console\bin\Debug> .\Pisces.Console.exe --help
Pisces.Console 1.0.0.0
Copyright c Flint Hills Resources Inc. 2025

AFAnalysisBackfill      Backfill AF Analysis metadata
AFAnalysisStatusBackfill Backfill AF Analysis status details
AFAnalysisTemplateBackfill Backfill AF Analysis Template metadata
AFCaseBackfill          Backfill AF Case metadata
AFCategoryBackfill      Backfill AF Category metadata
AFDeleteBackfill        Backfill deleted metadata
AFDatabaseBackfill      Backfill AF Database metadata
AFElementBackfill      Backfill AF Element metadata (includes AF Attributes associated with the Element)
AFElementTemplateBackfill Backfill AF Element Template metadata (includes AF Attribute Templates associated with the Element)
AFEnumerationSetBackfill Backfill AF Enumeration Set metadata
AFEventFrameBackfill    Backfill AF Event Frame metadata
AFEventFrameTemplateBackfill Backfill AF Event Frame Template metadata
AFModelBackfill         Backfill AF Model metadata
AFModelTemplateBackfill Backfill AF Model Template metadata
AFSecurityBackfill       Backfill AF Security Identities and Mappings metadata
AFServerBackfill        Backfill AF Server metadata
AFTransferBackfill       Backfill AF Transfer metadata
AFTransferTemplateBackfill Backfill AF Transfer Template metadata
PICollectiveBackfill     Backfill PI Collective metadata
PIDeleteBackfill        Backfill deleted metadata
PIPointBackfill         Backfill PI Point metadata
PISecurityBackfill       Backfill PI Security Identities and Mappings metadata
FullAFBackfill          Backfill all metadata for AF server
FullPIBackfill          Backfill all metadata for PI server
Service                 Continuous monitoring of metadata changes
help                   Display more information on a specific command.
version                Display version information.
```

Processed Identities



AF

- AFIdentity.Analysis
- AFIdentity.AnalysisTemplate
- AFIdentity.Case
- AFIdentity.CategoryAnalysis
- AFIdentity.CategoryAttribute
- AFIdentity.CategoryElement
- AFIdentity.CategoryNotificationRule
- AFIdentity.CategoryReferenceType
- AFIdentity.CategoryTable
- **AFIdentity.Database**
- AFIdentity.Element
- AFIdentity.ElementTemplate
- AFIdentity.EnumerationSet
- AFIdentity.EventFrame
- AFIdentity.Model
- AFIdentity.NotificationRule
- AFIdentity.NotificationRuleTemplate
- **AFIdentity.SecurityIdentity**
- **AFIdentity.SecurityMapping**
- **AFIdentity.System**
- AFIdentity.Transfer

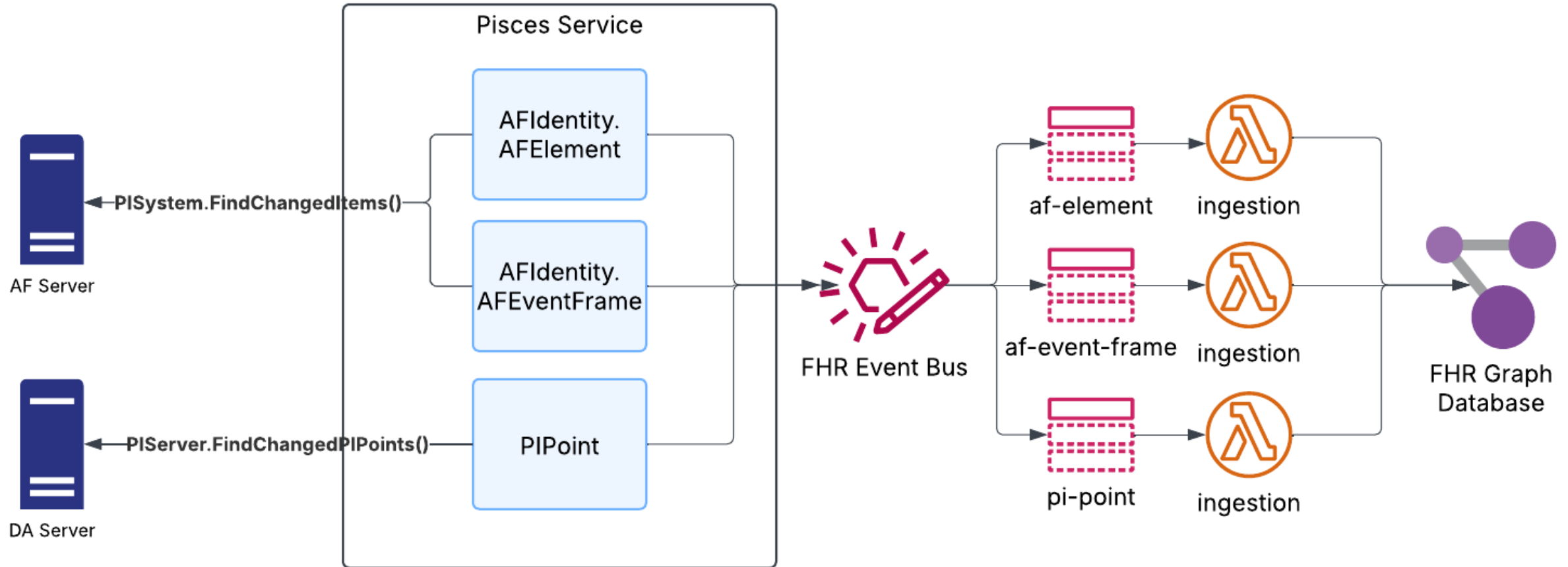
PI

- **PIIdentity**
- **PIMapping**
- PIPointChange
- **PIServer**
- **PICurrentValue**

Static identities are extracted at periodic frequency to ensure accuracy

Note: ElementTemplate handles Element/EventFrame/Model/Transfer

Architecture



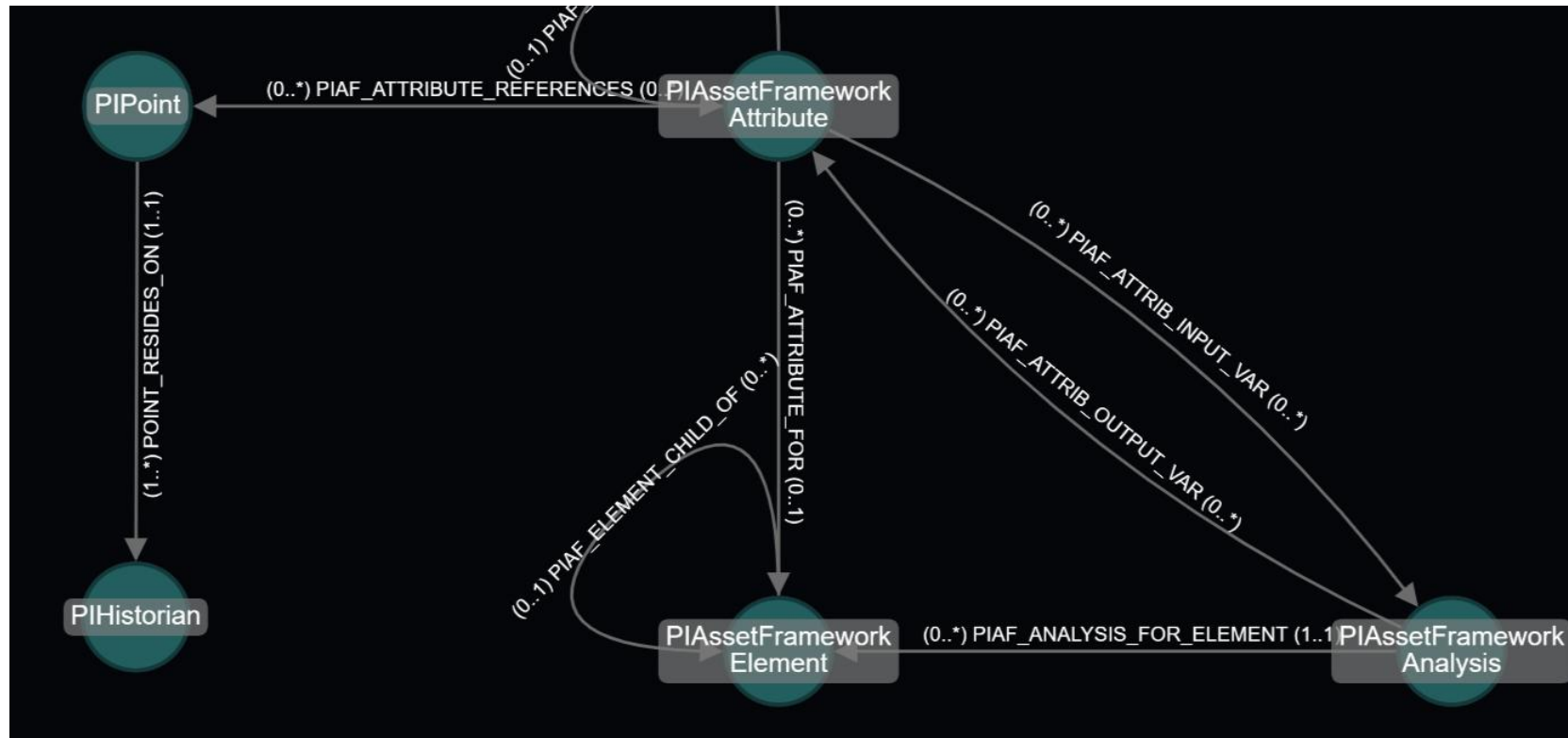


Use Cases

Practical applications of using this data in Graph

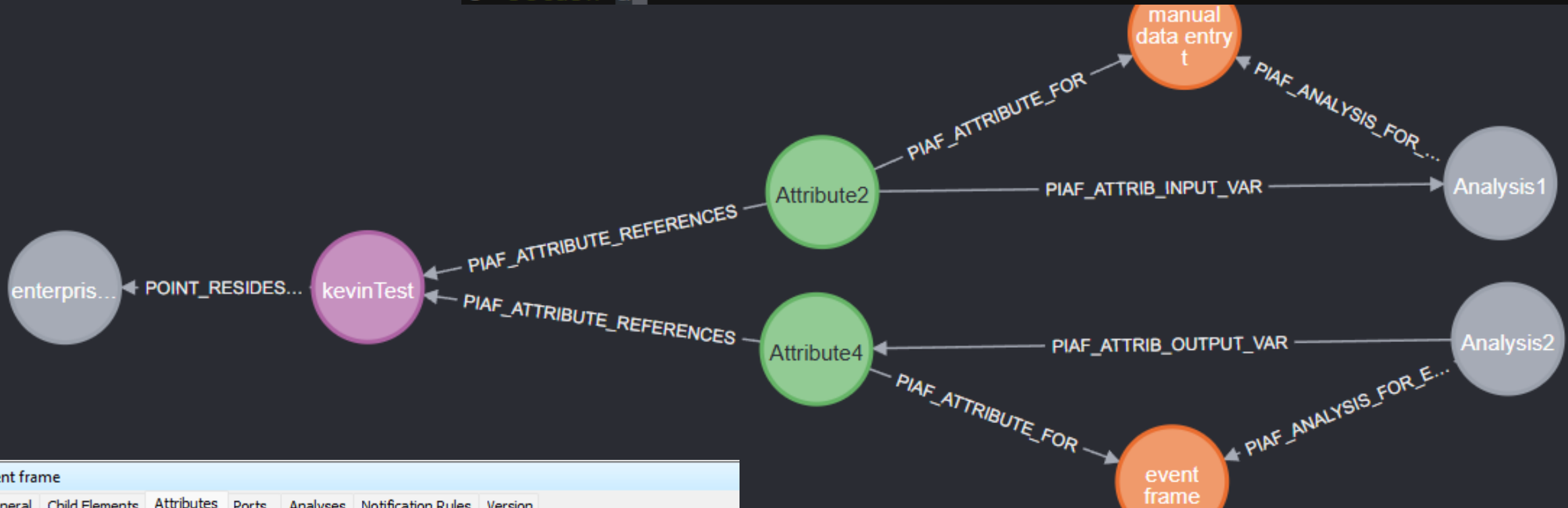
PI Point Analysis Outputs

- Problem Statement:
 - I have a PI point that has data that looks weird.
 - I think two different analyses are writing to the same point.
 - How do I find what additional analyses might be writing to it?



Success!

```
1 match (h:PIHistorian{name:'enterprise_dev'})
2 ←[:POINT_RESIDES_ON]-(p:PIPoint{name:'kevinTest'})
3 ←[:PIAF_ATTRIBUTE_REFERENCES]-(a:PIAssetFrameworkAttribute)
4 -[:PIAF_ATTRIBUTE_FOR]→(e:PIAssetFrameworkElement)
5 where exists { (a)←[:PIAF_ATTRIB_OUTPUT_VAR]-(PIAssetFrameworkAnalysis) }
6 return a
```

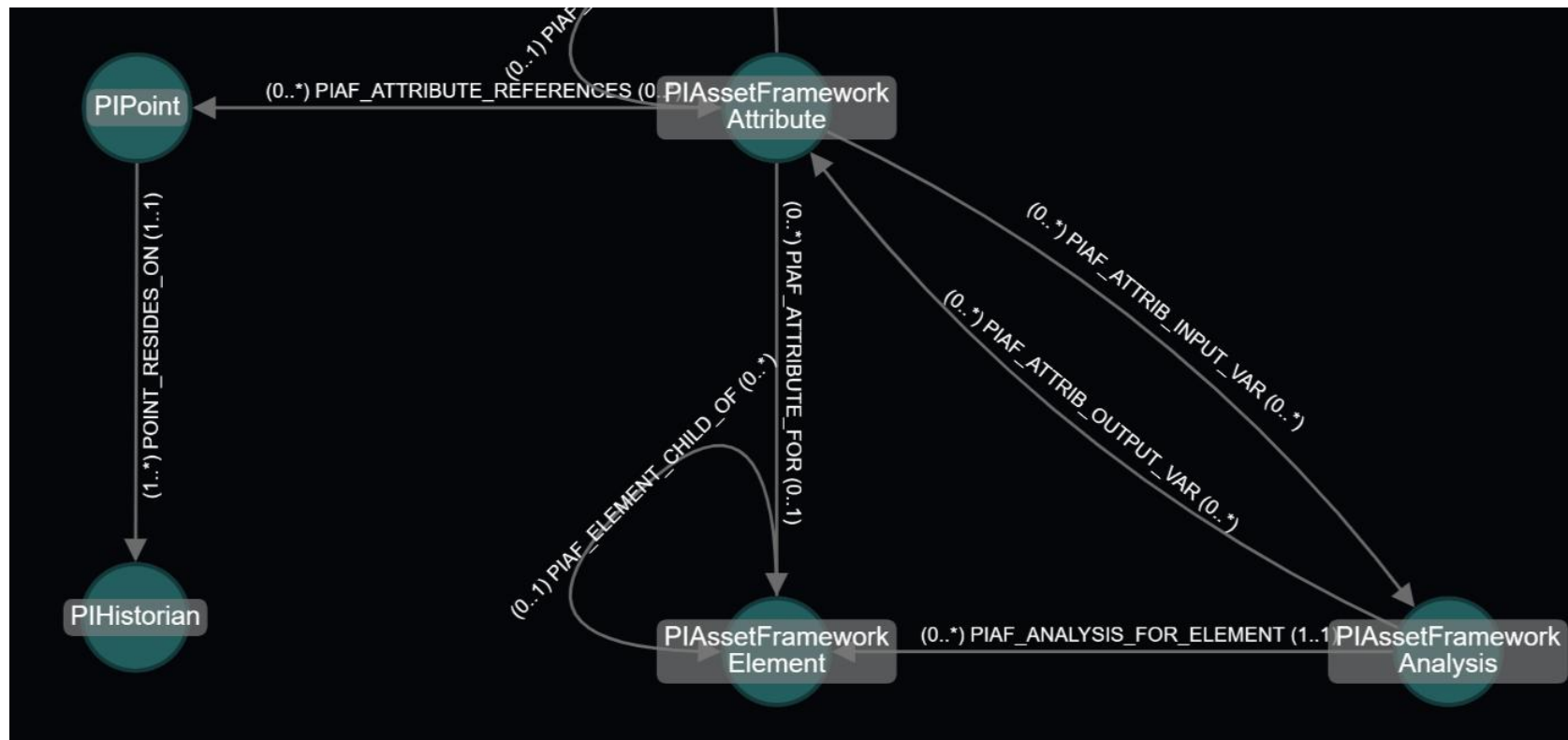


event frame			
General Child Elements Attributes Ports Analyses Notification Rules Version			
Filter			
	Name	Value	Settings...
	Attribute1	hopefully this works	
	Attribute2		
	Attribute3	12.226	\\enterprise_dev\SINUSOID
	Attribute4	6.1976	\\enterprise_dev\kevinTest

manual data entry test2			
General Child Elements Attributes Ports Analyses Notification Rules Version			
Filter			
	Name	Value	Settings...
	Analysis1_Variable1	14.758	\\enterprise_dev\SINUSOID
	Attribute1	my new value	
	Attribute2	6.7733	\\enterprise_dev\kevinTest

PI Point Analysis Inputs

- Problem Statement:
 - I need to make an EngUnits change to a PI point, it was setup in barrels per day (BPD) but should be thousand barrels per day (MBPD).
 - What calculations will need to be updated after making the units change?



Success!

```
1 match (h:PIHistorian {name: 'enterprise_dev'})←[:POINT_RESIDES_ON]-(p:PIPoint {name: 'SINUSOID'})←[:PIAF_ATTRIBUTE_REFERENCES]-(a:PIAssetFrameworkAttribute)-[:PIAF_ATTRIBUTE_FOR]→(e:PIAssetFrameworkElement)
2 where exists (a)-[:PIAF_ATTRIB_INPUT_VAR]→(:PIAssetFrameworkAnalysis)
3 return e.path as path, a.name
```

	path	a.name
1	"\\Enterprise Dev\\HTS\\event frame"	"Attribute3"
2	"\\Enterprise Dev\\HTS\\continuous write"	"Attribute2"
3	"\\Enterprise Dev\\HTS\\Backfill test"	"inputAttribute"

continuous write

General Child Elements Attributes Ports Analyses Notification Rules Version

Filter

	Name	Settings...
+	Attribute1	\\enterprise_dev\\SINUSOID_CALC
	Attribute2	\\enterprise_dev\\SINUSOID

continuous write	
General Child Elements Attributes Ports Analyses Notification	
✓	Analysis1 ✓
Add a new variable	
Name	Expression
Variable1	'Attribute2' * Sqr(7) * Pi()

Data Reference Daisy Chaining



- Graph excels at recursive lookups
- Example:
 - AF attribute references another AF attribute on the element
 - That AF attribute references a parent AF element's AF attribute
 - That AF attribute is a table lookup
 - That table lookup has a formula to determine which result to use
 - Eventually we get to the AF attribute that references a PI point
- Daisy chaining data references negatively affects PI Analysis Service
- Can be difficult to trace through PI System Explorer
- Output By...
 - Levels of recursion
 - Attributes with >3 chains
 - Graph Bubble Format
 - Show graphically the amount of recursion
 - What the actual PI point is
 - Traverse the recursion and show what the actual PI point is

Data Quality Checks



Data Quality: AF attributes with PI point data reference and no PI point relationship

```
1 match (n:PIAssetFrameworkAttribute)-[:PIAF_ATTRIBUTE_REFERENCE_IS]->(:PIAssetFrameworkAttributeDataReference { name: 'PI Point' })
2 return
3     n,
4     (not exists { (n)-[:PIAF_ATTRIBUTE_REFERENCES]->(point:PIPoint) }) as currentlyAnIssue
```

Data Quality: AF attribute precision loss for AF analyses

```
1 match (n:PIAssetFrameworkAttribute)-[:PIAF_ATTRIBUTE_REFERENCE_IS]->(:PIAssetFrameworkAttributeDataReference { name: 'PI Point'})
2 match (n)-[:PIAF_DATATYPE_IS]->(attributeDataType:PIAssetFrameworkAttributeDatatype)
3 optional match (n)-[:PIAF_ATTRIBUTE_REFERENCES]->(point:PIPoint)-[:POINT_DATATYPE_IS]->(pointDataType:PIPointDatatype)
4 return
5     n,
6     case
7         when attributeDataType.datatype = 'DateTime' and pointDataType.datatype = 'Timestamp' then true
8         when attributeDataType.datatype = 'Single' and pointDataType.datatype <> 'Float32' then true
9         when attributeDataType.datatype = 'Double' and pointDataType.datatype <> 'Float64' then true
10        when attributeDataType.datatype = 'Int16' and pointDataType.datatype <> 'Int16' then true
11        when attributeDataType.datatype = 'Int32' and pointDataType.datatype <> 'Int32' then true
12        when attributeDataType.datatype = 'String' and pointDataType.datatype <> 'String' then true
13        when attributeDataType.datatype = 'AFEnumerationValue' and pointDataType.datatype <> 'Digital' then true
14        else false
15    end as currentlyAnIssue
```

Business Logic Checks



Business Logic: AF attributes with no data reference should be configuration items

```
1 match (s:PIAssetFrameworkServer {name: 'Enterprise Dev'})
2   ←[:PIAF_DB_HOSTED_ON]-(d:PIAssetFrameworkDatabase {name: 'Configuration'})
3   ←[:PIAF_ELEMENT_RESIDES_ON]-(e:PIAssetFrameworkElement)
4   ←[:PIAF_ATTRIBUTE_FOR]-(a:PIAssetFrameworkAttribute)-[:PIAF_ATTRIBUTE_REFERENCE_IS]
5   →(:PIAssetFrameworkAttributeDataReference { name: 'None' })
6   where a.isConfigurationItem = false
7   return
8     'x' as Selectedx,
9     replace(e.path, '\\\\Enterprise Dev\\\\Configuration\\\\', '') as Parent,
10    a.name as Name,
11    'Attribute' as ObjectType,
12    true as AttributeIsConfigurationItem
```

Table

Text

Code

	Selectedx	Parent	Name	ObjectType	AttributeIsConfigurationItem
1	"x"	"OS\soft\PIANO\AnalysisService\PersistedFileData"	"RecalculationRequestedHashCode"	"Attribute"	true
2	"x"	"OS\soft\PIANO\AnalysisService\PersistedFileData"	"PersistedCalculationsHashCode"	"Attribute"	true

	A	B	C	D	E	
1	Selectedx	Parent	Name	ObjectType	AttributeIsConfigurationItem	
2	x	OS\soft\PIANO\AnalysisService\PersistedFileData	RecalculationRequestedHashCode	Attribute	TRUE	
3	x	OS\soft\PIANO\AnalysisService\PersistedFileData	PersistedCalculationsHashCode	Attribute	TRUE	
4						

Future - LLM Use Case

- Natural language chat bots
- “Show me the PI points for compressor XYZ in unit A”
- Embedded AVEVA™ PI Vision™ ad hoc display with results

Assistant

 Working on a request for: what is the ambient temperature at the fort worth terminal - ...provide our Team a moment...

 
13:38

Agent Team!

 A

Name	Description	Value	Units	Trend	Minimum	Maximum
	Fort Worth 30 day forecast	No Data	°F		No Data	No Data
	Fort Worth 16 day forecast	No Data	°F		No Data	No Data
	Ft Worth Ambient Temperat	77.05	°F		62.49	77.05
	Fort Worth 4 day forecast A	No Data	°F		No Data	No Data

Future – AVEVA PI Vision Displays



- Use the AVEVA PI Vision API to extract displays in JSON format
- Create ingestion into Graph
- Create relationships to AF elements/AF attributes and PI points
- Use Case:
 - I have issue with a pump in Unit X
 - I know PI point XYZ is a pump pressure of Unit X
 - Look up PI point/AF element XYZ in Graph to find all the PI Vision displays that are built already for this unit
 - If it's AF contextualized already then give me PI Vision links with the &path= populated to the correct context

PISCES Monitoring

- PISCES Service publishes custom Windows Performance Monitor Counters
- Use PI Interface for Performance Monitor to collect
- Monitor service performance
- Monitor changes throughput by type
- AF analyses alert us when we stop processing changes
- AVEVA PI Vision display to visualize



Flint Hills Resources uses AVEVA PI System with enterprise graph database for advanced analytics and governance

Challenge

- Governing operations data grew more difficult as data infrastructure and usage scaled
- Answering complex data quality questions required full analysis across systems
- Investigating problems with existing approaches distracted from other business initiatives

Solution

- Integrated AVEVA™ PI System™ data (PI points, AF elements, AF attributes, etc.) with graph database to support growth.

Results

- **Identified 60,000 misconfigured data points across the enterprise for immediate savings**
- **Caught new misconfigurations with daily reporting, limiting their impact**
- **Accelerated use of emerging technologies to address difficult business opportunities**

