Near Real-Time IoT Analytics of Pumping Stations in PowerBI

Mehmet Bakkaloglu
Business Insights & Analytics, Hitachi Consulting

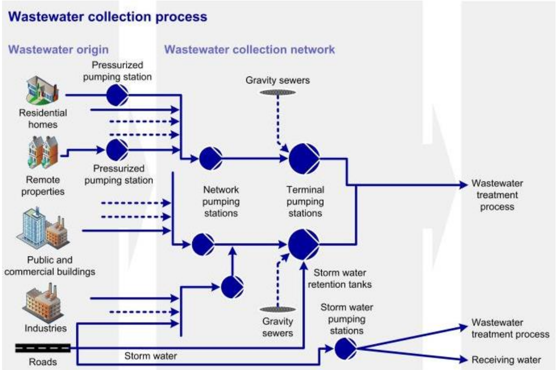
Team: Siddhartha Mohapatra, Vaughan Rees, John Shiangoli, Sidhartha Mahapatro

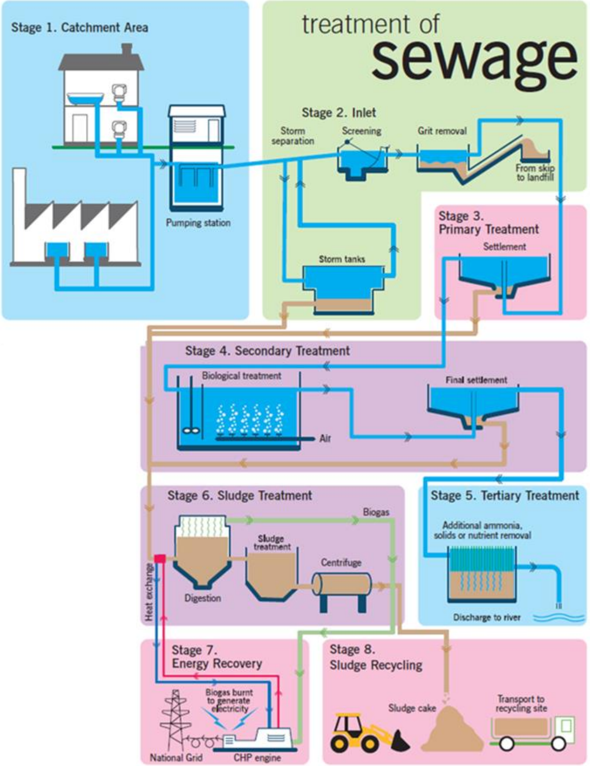
SQLBits Conference
7 April 2017

Contents

- Background
- Challenge
- Dashboards
- Solution
- Future improvements
- Comparison to other options

Waste Water Network





PS: Pumping Station
CSO: Combined Sewer Overflow
WWTW: Waste Water Treatment Works

Objective: Prevent Waste Water Spills

- Spills are damaging to the environment and may incur a fine from the Environment Agency

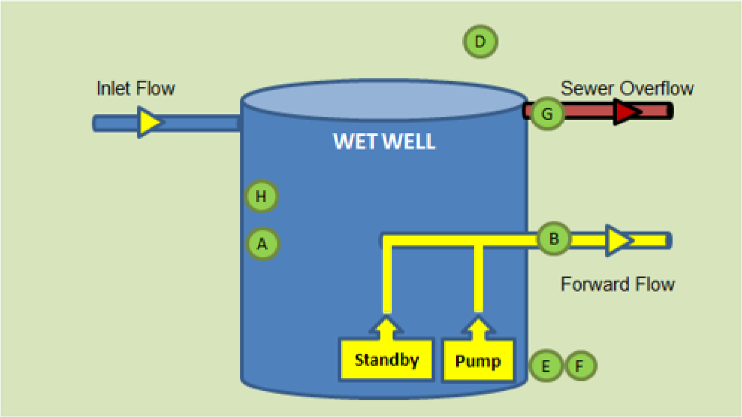
The Data

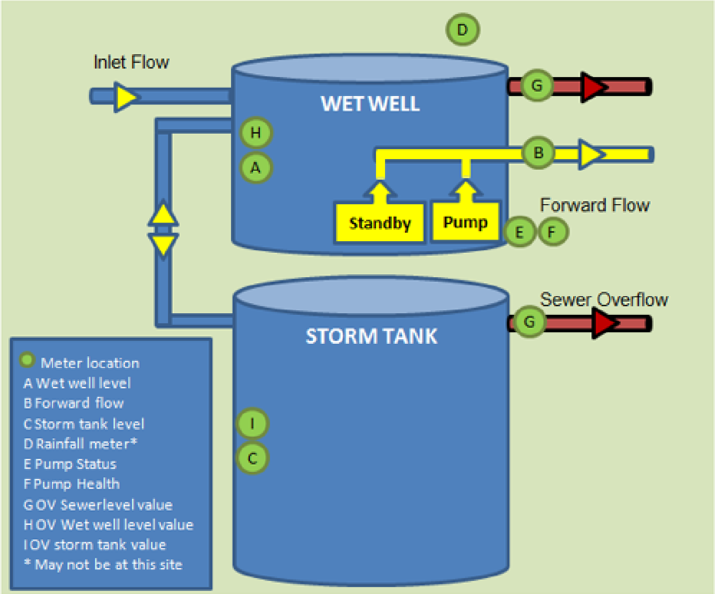
- ~600 Waste Water Treatment Plants, ~900 Pumping Stations, ~250 CSOs
- Sites have different architectures, different sets of IoT sensors (SCADA signals), different naming conventions for sensors, and some of them 100s of sensors
- Analogue signals up to every 15 min
- Digital signals only when there is a change from 1 to 0 and 0 to 1 (therefore there can be a long gap between receiving a signal)
- This particular solution currently has 45 million rows (100 sites, Aug 16 – Mar 17). With new sites added, it will be around 400 million rows (600 sites, 1 years' data).

For the purposes of this session Site Names, Catchment Names, Beach Names have been masked, and postcodes mapped to random Scottish postcodes

Pumping Stations

- Pumping Station sites can have multiple wet wells and/or storm water tanks
- Simplified diagrams showing the IoT sensors:





Challenge

- How to produce a generic set of dashboards for pumping stations that will:
    - Show us the likelihood of a spill at a pumping station
    - Help us investigate the cause of a spill, **including going back in history**
    - Show the asset status

✓ With access to only flat files of IoT data (SCADA)

✓ Using Azure & PowerBI

✓ In near real-time (~15 minutes latency)

✓ With compelling visualisations

PowerBI Dashboards

Current Status

HITACHI
Inspire the Next

Dashboard showing the likelihood of a spill at a site

- Used by site engineers and operators
- Shows the logic used to derive likelihood of spill
- Can also filter by beach to determine whether spills are affecting a particular beach
- Can be used on mobile devices as well

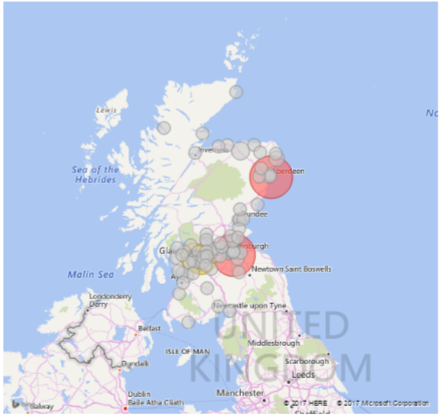
Pollution Insights : Current Status

Area
- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5
- ☐ 6

Catchment
- ☐ Catchment1
- ☐ Catchment10
- ☐ Catchment11
- ☐ Catchment12
- ☐ Catchment13
- ☐ Catchment14
- ☐ Catchment15
- ☐ Catchment16
- ☐ Catchment17
- ☐ Catchment18
- ☐ Catchment19
- ☐ Catchment2
- ☐ Catchment20

Beach
- ☐ (Blank)
- ☐ Beach10
- ☐ Beach11
- ☐ Beach12
- ☐ Beach13
- ☐ Beach14
- ☐ Beach15
- ☐ Beach16
- ☐ Beach17
- ☐ Beach18
- ☐ Beach19
- ☐ Beach2
- ☐ Beach20

Site Type	Area	SiteName	Status	RiskLevel	Rainfall Level	PFF (l/s)	Last Signal Time
4	6	Site400	Red	Spilling	Light Rain	23.80	21/03/2017 19:00:11
4	5	Site720	Red	Spilling	Light Rain		21/03/2017 19:01:00
4	1	Site1638	Amber	High	Light Rain	38.80	21/03/2017 19:01:28
0	6	Site2033	Amber	High	Light Rain	0.00	21/03/2017 19:00:11
0	2	Site1876		High	Steady Rain	4.64	21/03/2017 19:00:00
4	1	Site1795		Medium	Heavy Rain		21/03/2017 19:00:00
4	5	Site1003		Low	Light Rain		21/03/2017 19:01:00
4	4	Site1011		Low	Light Rain		21/03/2017 19:00:00
0	6	Site1040		Low	Light Rain	7.13	21/03/2017 19:00:00
4	2	Site1054		Low	Light Rain		21/03/2017 19:00:00
0	6	Site1063		Low	Light Rain		21/03/2017 19:01:30
4	5	Site1067		Low	Light Rain	229.20	21/03/2017 19:01:00
4	3	Site1074		Low	Light Rain		21/03/2017 19:00:00
4	5	Site1084		Low	Light Rain		21/03/2017 19:01:00
0	1	Site1102		Low	Light Rain	0.02	21/03/2017 19:01:28
3	4	Site1103		Low	Light Rain	26.34	21/03/2017 19:00:00
4	5	Site1106		Low	Light Rain		21/03/2017 19:01:00
0	4	Site1127		Low	Light Rain	46.71	21/03/2017 19:00:00
4	1	Site1170		Low	Light Rain	0.00	21/03/2017 19:00:00
4	6	Site1181		Low	Light Rain		21/03/2017 19:01:30
4	2	Site1190		Low	Light Rain		21/03/2017 19:01:30
4	3	Site1226		Low	Light Rain	0.00	21/03/2017 19:00:00
4	4	Site1248		Low	Light Rain		21/03/2017 19:00:00
2	3	Site1269		Low	Light Rain	0.01	21/03/2017 19:00:00



Site Type

Site Type	Site Type Description	Link
0	Regular	
1	With Tide Meter	
2	Inhibited by other site	
3	Storm Tank instead of Wet well	
4	No Pump status signal	
5	Pumped Overflow signal for Spills	

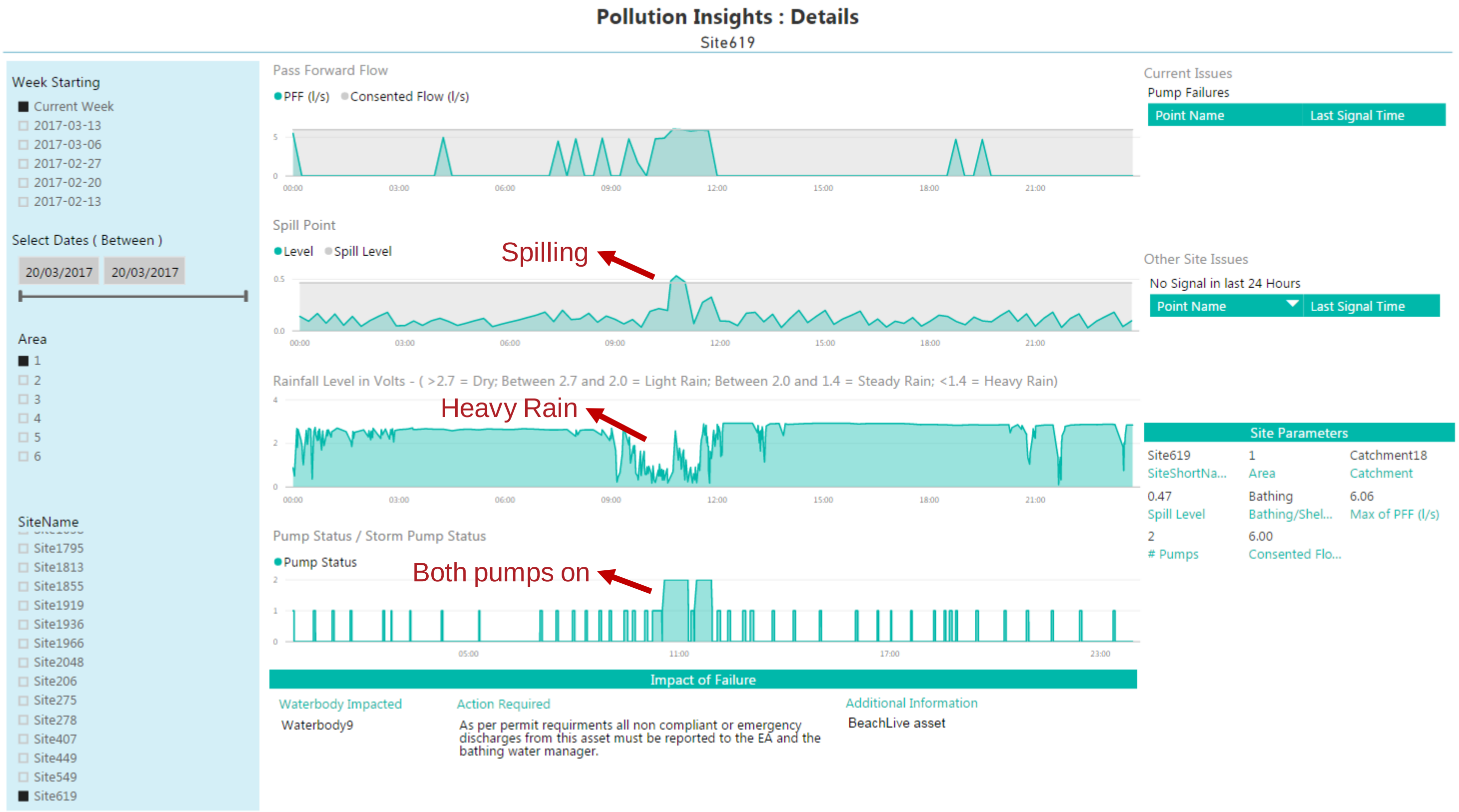
Status	SiteName	Status Reason
Red	Site400	Light Rain; OV Lorna Irwin Screen Level Value (E14374)(0.761m) > Spill Level (0.605m); PFF(23.8 l/s) < Consented Flow (45 l/s);
Red	Site720	Light Rain; OV Sewer Overflow Level Value (E1594)(3.304m) > Spill Level (1.068m);
Amber	Site1638	Light Rain; OV SSO Chamber Level Value (E5401)(0.363m) <= 95% of Spill Level (0.37525m) AND > 80% of Spill Level (0.316m); PFF(38.8 l/s) < Consented Flow (78 l/s);
Amber	Site2033	Light Rain; OV Storm Tank Level Value (E8247)(2.084m) <= 95% of Spill Level (2.2724m) AND > 80% of Spill Level (1.9136m); PFF(. l/s) < Consented Flow (57 l/s); No Pumps ...

Details (1)

Dashboard allowing users to identify the cause of a spill

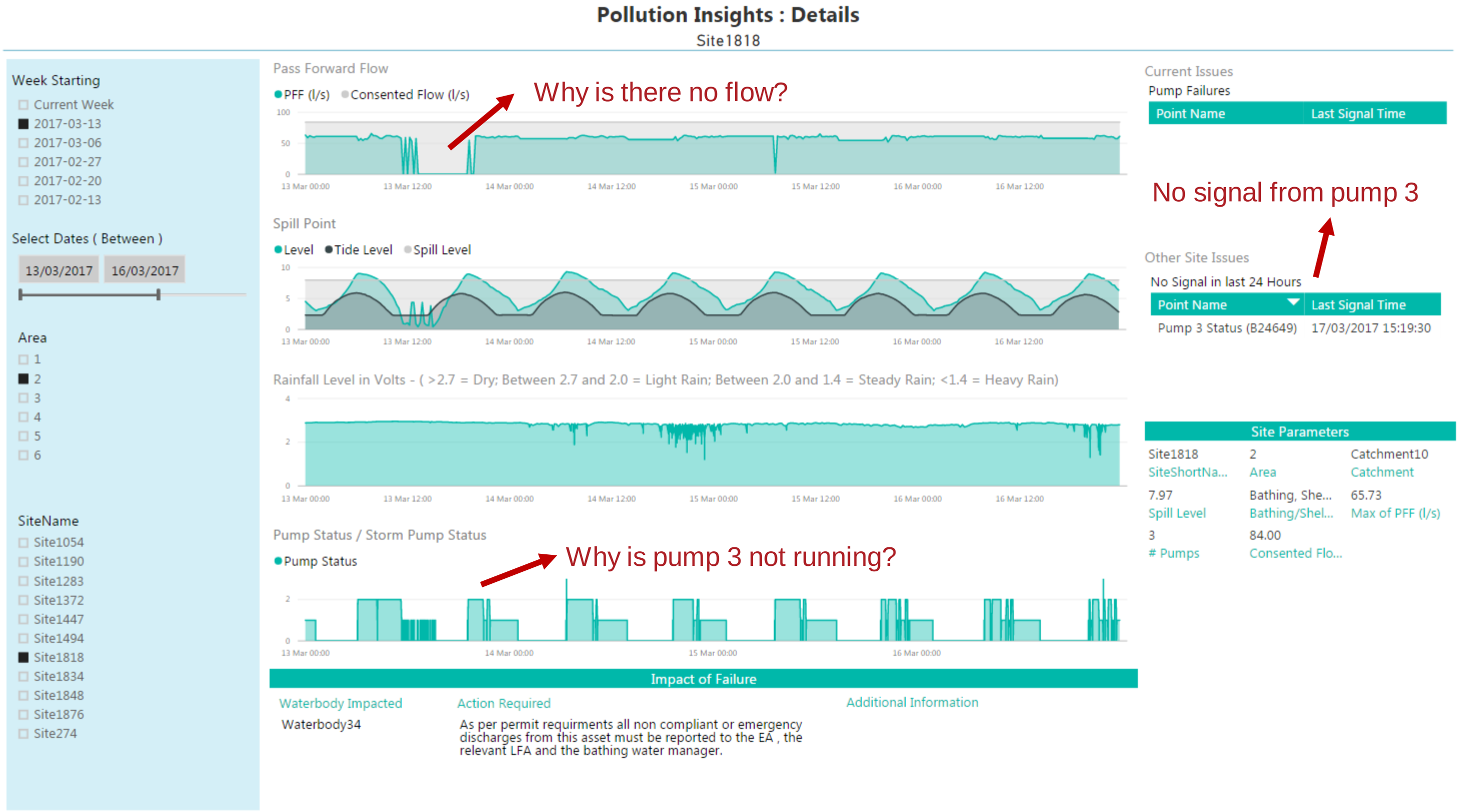
Example:

- It might be that pump was not running when it should have been, which would indicate an issue with the pump
- It might be that when there is more than one pump running the flow did not increase, which might indicate a blockage e.g. dead animal or garbage
- etc.



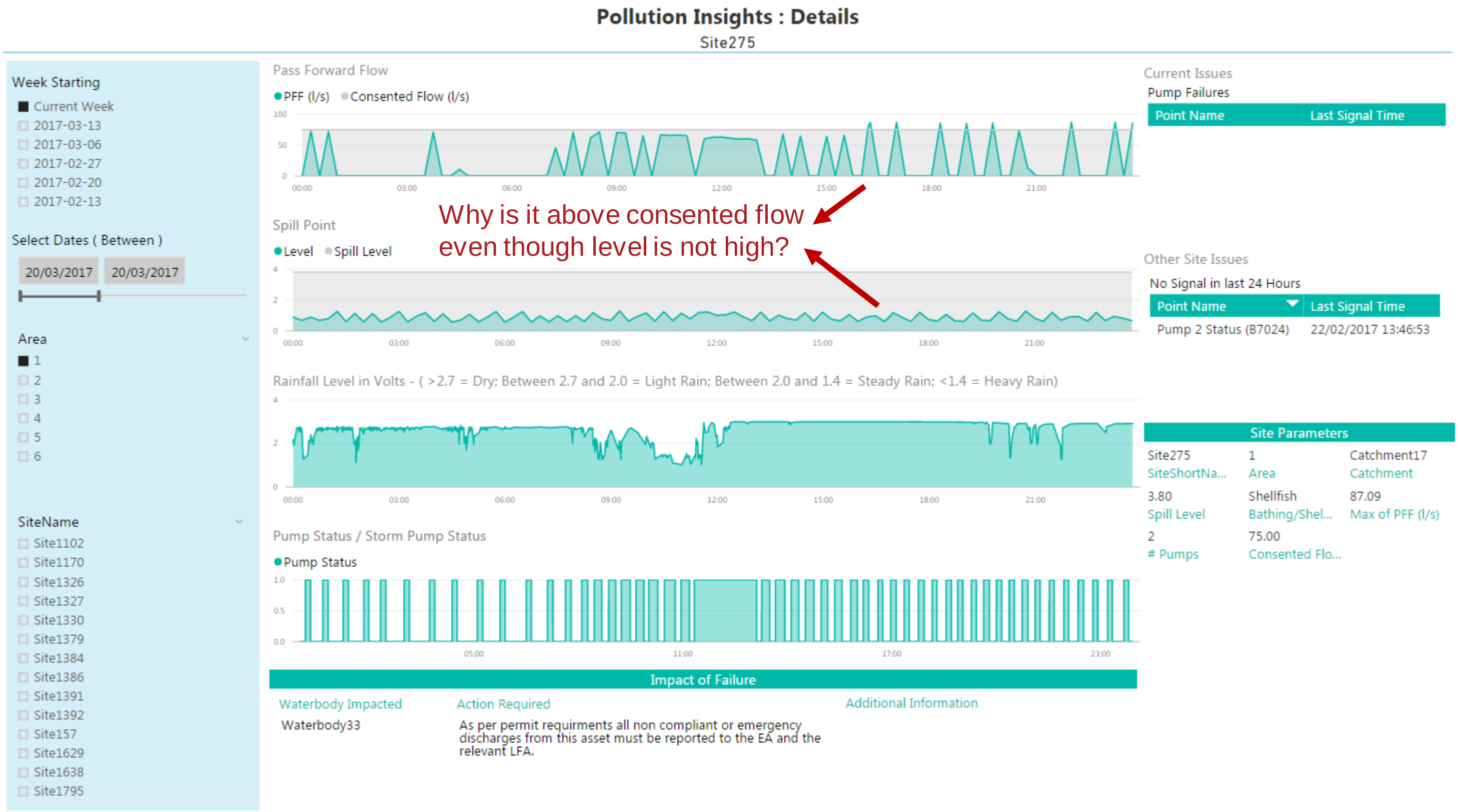
Details (2)

HITACHI
Inspire the Next



Details (3)

HITACHI
Inspire the Next



Signal Status

Hitachi
Inspire the Next

- Shows whether there was a signal in the last day
- If there was not a signal in the last day this could mean there is need for maintenance
- Last signal value vs signal average

Pollution Insights : Signal Status

Area
1
2
3
4
5
6

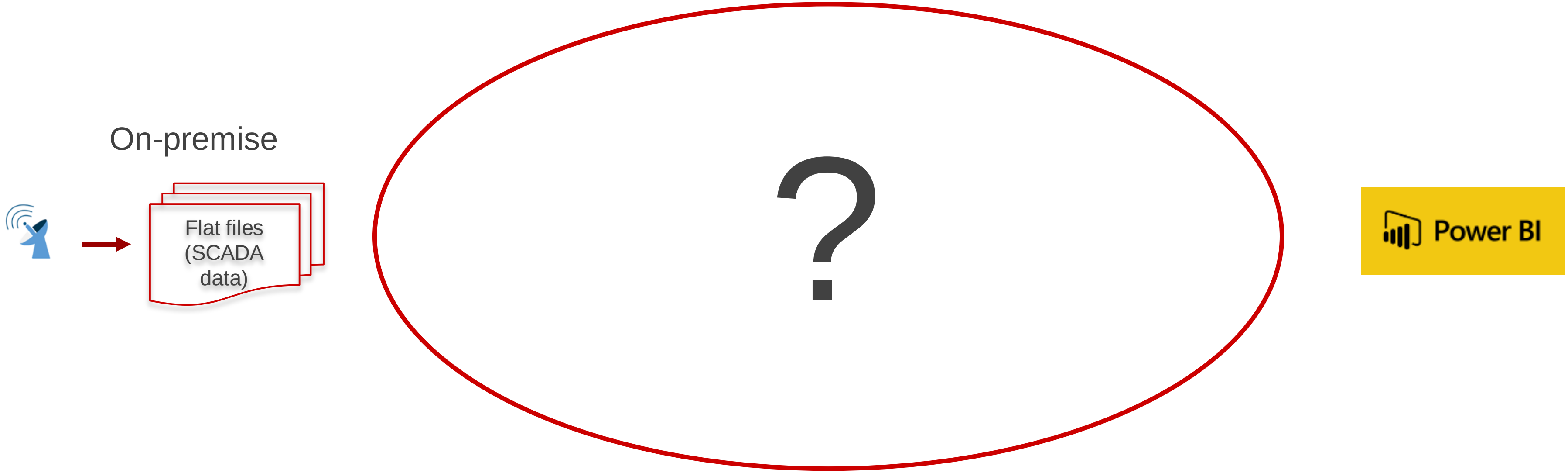
Catchment
Catchment1
Catchment10
Catchment11
Catchment12
Catchment13
Catchment14
Catchment15
Catchment16
Catchment17
Catchment18
Catchment19
Catchment2

SiteName
Site1327
Site1384
Site1966
Site549

Area	SiteName	Point Name	Measurement Unit	Last Signal Time	Signal Average	Last Signal Value	Signal in Last Day
1	Site1327	Pump 2 Status (B21518)	on/off			Pump has stopped	No
1	Site549	Pump 1 Status (B16888)	on/off	11/03/2017 17:23:41		Pump has stopped	No
1	Site549	Pump 2 Status (B16891)	on/off	16/03/2017 12:30:33		Pump has stopped	No
1	Site1327	Pump 1 Status (B21516)	on/off	21/03/2017 02:43:22		Pump has stopped	Yes
1	Site1327	Pump 3 Status (B21520)	on/off	21/03/2017 02:43:22		Pump has stopped	Yes
1	Site1327	Pump 4 Status (B21522)	on/off	21/03/2017 02:43:22		Pump has stopped	Yes
1	Site1327	OV Storm Tank Level Value (E6567)	m	21/03/2017 03:00:00	0.51	0.674 m	Yes
1	Site1327	Rainfall Meter Value (E11122)	Volts	21/03/2017 19:00:00	2.33	0.000 Volts	Yes
1	Site1384	OV Storm Sump Level Value (E10886)	m	21/03/2017 19:00:00	0.88	0.972 m	Yes
1	Site1384	Rainfall Meter Value (E9228)	Volts	21/03/2017 19:00:00	2.37	2.627 Volts	Yes
1	Site1384	Discharge Flow Meter Value (E10883)	l/s	21/03/2017 19:00:00	45.63	3.000 l/s	Yes
1	Site1966	Sewer EDM Level Value (E2354)	m	21/03/2017 15:30:00	0.10	0.059 m	Yes
1	Site1966	Rainfall Meter Value (E11122)	Volts	21/03/2017 19:00:00	2.33	0.000 Volts	Yes
1	Site549	Pump 3 Status (B16894)	on/off	21/03/2017 02:23:43		Pump has stopped	Yes
1	Site549	OV Sewer Level Value (E5584)	m	21/03/2017 02:30:00	0.00	0.000 m	Yes
1	Site549	Rainfall Meter Value (E10924)	Volts	21/03/2017 19:00:00	2.33	2.823 Volts	Yes
1	Site1327	Pump 3 Health (B21519)	on/off			Pump is healthy	
1	Site1327	Pump 1 Health (B21515)	on/off	10/08/2016 07:40:13		Pump is healthy	
1	Site1327	Pump 2 Health (B21517)	on/off	27/10/2016 14:09:19		Pump has failed	
1	Site1327	Pump 4 Health (B21521)	on/off	13/01/2017 13:55:25		Pump is healthy	
1	Site549	Pump 1 Health (B16887)	on/off	23/11/2016 15:07:51		Pump is healthy	
1	Site549	Pump 2 Health (B16890)	on/off	23/11/2016 15:07:55		Pump is healthy	
1	Site549	Pump 3 Health (B16893)	on/off	07/03/2017 15:36:47		Pump is healthy	

Architecture

- How do we fill in the gap to meet the requirements?



First set of requirements to be addressed

HITACHI
Inspire the Next

- Names of signals from different sites need to be standardised
- In some cases, due to the architecture of sites, signals from multiple sites need to be reported as one site
- Not all sites have rainfall data, therefore we need to map it from the nearest site
- For rainfall level use average of last one hour of rainfall voltage
- Likelihood of a spill at a pumping station requires complex logic
- Visualise Pump Status properly

AssetPrefix	AssetName	AssetSuffix	PointShortName	AssetType	Unit	MetricName
Output	Flow Meter	NULL	Value	Flow Meter	m3/hr	Forward Flow
Rising Main	Flow Meter	NULL	Value	Flow Meter	l/s	Forward Flow
OV Pumped	Flow Meter	NULL	Value	Flow Meter	l/s	Forward Flow
FP	Flow Meter	NULL	Current Discharge Value	Flow Meter	l/s	Forward Flow
Outlet	Flow Meter	NULL	Value	Flow Meter	l/s	Forward Flow
Pumped	Flow Meter	NULL	Value	Flow Meter	l/s	Forward Flow
Pump 2	Flow Meter	NULL	Value	Flow Meter	l/s	Forward Flow
Pump 1	Flow Meter	NULL	Value	Flow Meter	l/s	Forward Flow
Outlet	Flow Meter	NULL	Value	Flow Meter	l/s	Forward Flow
Discharge	Flow Meter	NULL	Value	Flow Meter	l/s	Forward Flow

MappingType	EntityTypeFrom	EntityTypeTo
Telemetry Point	Signal	MetricName
Signal Calculations	Signal	Calculation
Override Mapping	Signal	SiteName
Site Mapping	Site Name	Site Short Name

Likelihood of a spill

Working with site engineers, analysing historical data, and based on available sensors:

Around 100 sites classified into 6 types:

Site Type	Site Type Description
0	Regular
1	With Tide Meter
2	Inhibited by other site
3	Storm Tank instead of Wet well
4	No Pump status signal
5	Pumped Overflow signal for Spills

Around 300 rules defined using various signals and thresholds to determine RAG Status and Risk Level

- Is there a Flow signal?
- Flow vs Consented Flow?
- Rainfall vs Threshold?
- Wet Well Level vs Spill Level vs Thresholds?
- Pump(s) running?
- Is there a Tide Meter?
- Tide increasing/decreasing?
- Storm pump(s) running?
- Is there an inhibitor?
- Is there a Pumped Overflow signal?

Spill RAG#	Risk Level	RAG Status
12	Spilling	Red
11	Very High	Red
10	High	Red
9	Medium	Red
8	Spilling	Amber
7	High	Amber
6	Medium	Amber
5	Low	Amber
4	Spilling	Green
3	High	White
2	Medium	White
1	Low	White

Some spills are allowed e.g. if it is raining heavily, but other spills are not allowed. RAG Status identifies this.

Likelihood of a spill

HITACHI
Inspire the Next

RuleSK	SiteType	PassForwardFlowFlag	RainfallFlag	WetWellLevelFlag	PumpStatusFlag	TideLevelFlag	StormPumpStatusFlag	InhibitFlag	PumpedOverflowFlag	RiskLevel	RAGStatus
11	0	No PFF Signal	Rainfall >= Threshold	Wet Well Level <= Threshold2 AND > Threshold1	No Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Low	Amber
12	0	No PFF Signal	Rainfall >= Threshold	Wet Well Level <= Threshold2 AND > Threshold1	Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Low	White
13	0	No PFF Signal	Rainfall >= Threshold	Wet Well Level <= Spill Level AND > Threshold2	No Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Medium	Red
14	0	No PFF Signal	Rainfall >= Threshold	Wet Well Level <= Spill Level AND > Threshold2	Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Medium	White
15	0	No PFF Signal	Rainfall >= Threshold	Wet Well Level > Spill Level	No Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	High	Red
16	0	No PFF Signal	Rainfall >= Threshold	Wet Well Level > Spill Level	Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Spilling	Green
17	0	PFF < Consented Flow	Rainfall >= Threshold	Wet Well Level > Spill Level	Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Low	Amber
18	0	PFF < Consented Flow	Rainfall >= Threshold	Wet Well Level > Spill Level	No Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	High	Red
19	0	PFF < Consented Flow	Rainfall >= Threshold	Wet Well Level <= Spill Level AND > Threshold2	Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	High	Amber
20	0	PFF < Consented Flow	Rainfall >= Threshold	Wet Well Level <= Spill Level AND > Threshold2	No Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Very High	Red
21	0	PFF < Consented Flow	Rainfall >= Threshold	Wet Well Level <= Threshold2 AND > Threshold1	Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	High	Amber
22	0	PFF < Consented Flow	Rainfall >= Threshold	Wet Well Level <= Threshold2 AND > Threshold1	No Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	High	Amber
33	0	PFF >= Consented Flow	Rainfall < Threshold	Wet Well Level <= Threshold1	Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Low	White
34	0	PFF >= Consented Flow	Rainfall < Threshold	Wet Well Level <= Threshold1	No Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Low	White
35	0	PFF >= Consented Flow	Rainfall < Threshold	Wet Well Level <= Threshold2 AND > Threshold1	Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Low	Amber
36	0	PFF >= Consented Flow	Rainfall < Threshold	Wet Well Level <= Threshold2 AND > Threshold1	No Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Low	White
37	0	PFF >= Consented Flow	Rainfall < Threshold	Wet Well Level <= Spill Level AND > Threshold2	Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Very High	Red
38	0	PFF >= Consented Flow	Rainfall < Threshold	Wet Well Level <= Spill Level AND > Threshold2	No Pumps Running	No Tide Meter	No Storm Pumps Running	N	NA	Very High	Red

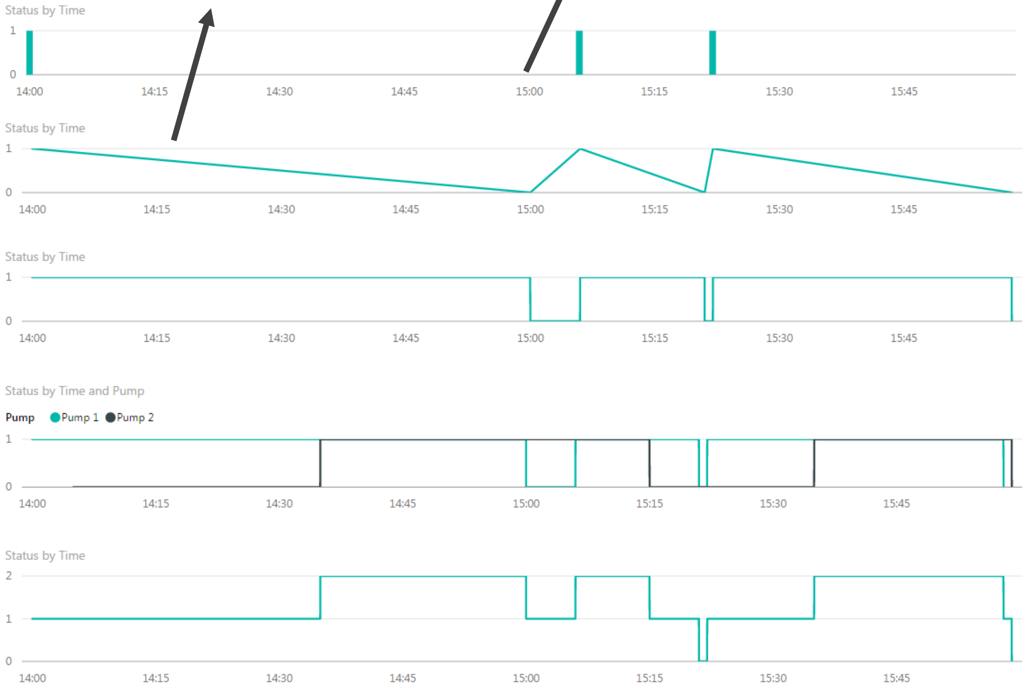
We need a platform that enables us to implement such a rules engine in near real-time

Visualise Pump Status

- Pump Status is a digital signal that comes only when there is a change from 1 to 0 and 0 to 1
- It does not come at a regular interval
- In order to visualise it and to be able to filter properly, generate data points at 5 minute intervals and at end points
- While this is not difficult to do, we need **a platform that enables us to generate data points in near real-time**

Filtering on > 14:15 would show no pump status until 15:00

Zeros are missing



Original data points

Original data points visualised differently

With generated data points

2 pumps (option 1)

2 pumps (option 2)

Where to implement these requirements

a) Data Warehouse
b) Stream Analytics
c) Storm on HDInsight
d) Spark on HDInsight
e) Somewhere else?

- Easiest choice: Do it in a Data Warehouse – but what about latency?

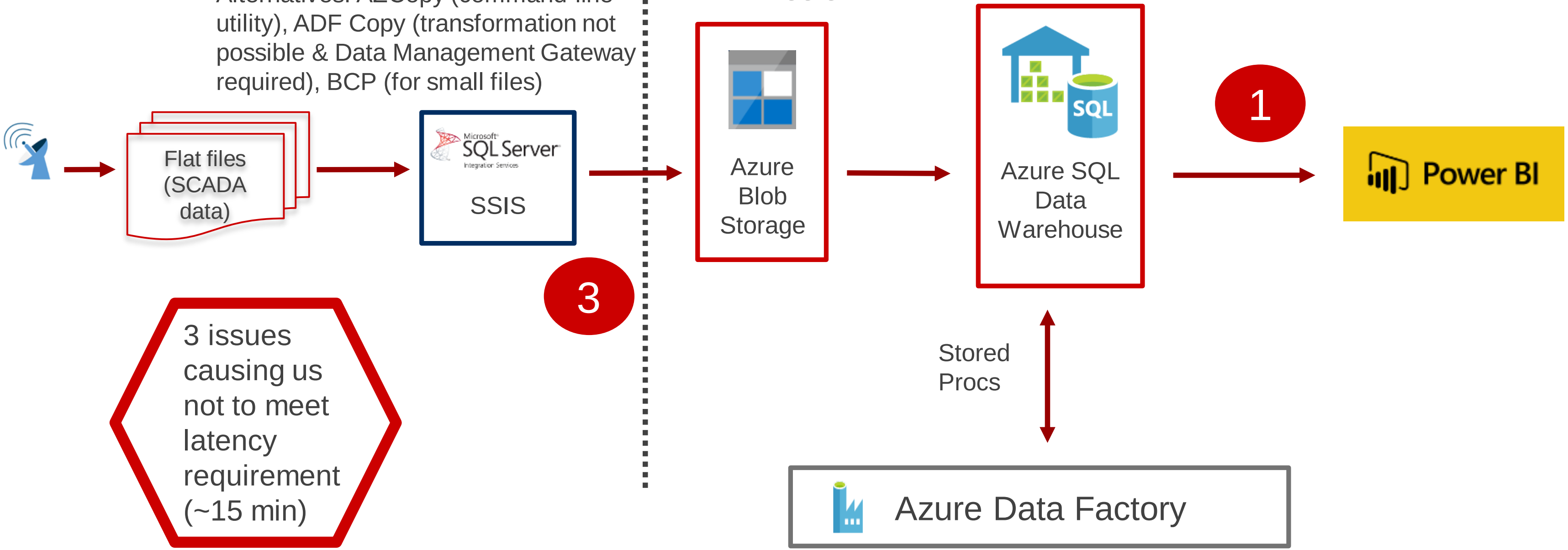
Basic Architecture

On-premise

Cloud

Use **SSIS Azure Blob Upload Task**
- Can do transformations, conditional tasks, and connect to various sources
- Alternatives: AZCopy (command-line utility), ADF Copy (transformation not possible & Data Management Gateway required), BCP (for small files)

Load into DW using **PolyBase**
- Fastest way of loading
- Rate increases when DWU is increased
- This is not the case for other options such as BCP, ADF, SSIS



Improvement 1 – Azure Tabular Model

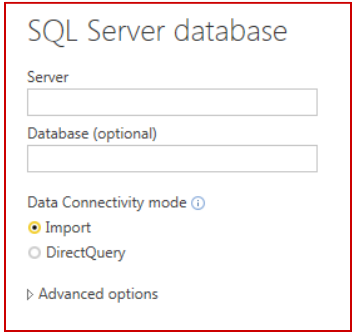
- PowerBI **DirectQuery** is unacceptably slow, has functional limitations, creates load on DW
- PowerBI **Import Mode** even with Pro licence can only be refreshed **up to 8 times a day,** has a size limit, and no partitioning

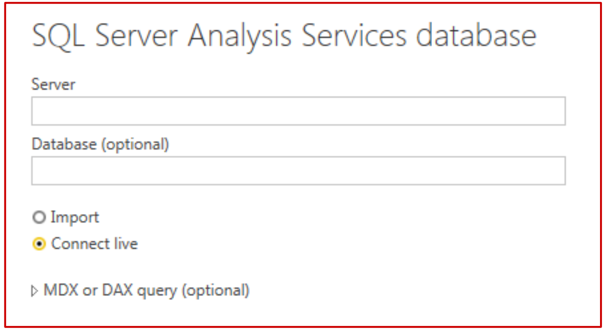
➢ How can we make this near real-time?

- Use **Azure Analysis Services Tabular Model** – we can process it as frequently as we wish
- Improve processing performance by **partitioning Tabular Model into current and historical**
- **Live Connection** from PowerBI to Tabular Model works super fast

(Azure Analysis Services came out in October 2016 & is still in preview)

Power BI Connection Options





How to process partitions of Azure Tabular Model & streamline with ETL process

HITACHI
Inspire the Next

- No out-of-the-box way of automating the processing of Azure Tabular Model yet
- Do it in C# using Tabular Object Model libraries (TOM) and run from Azure Data Factory using Azure Batch

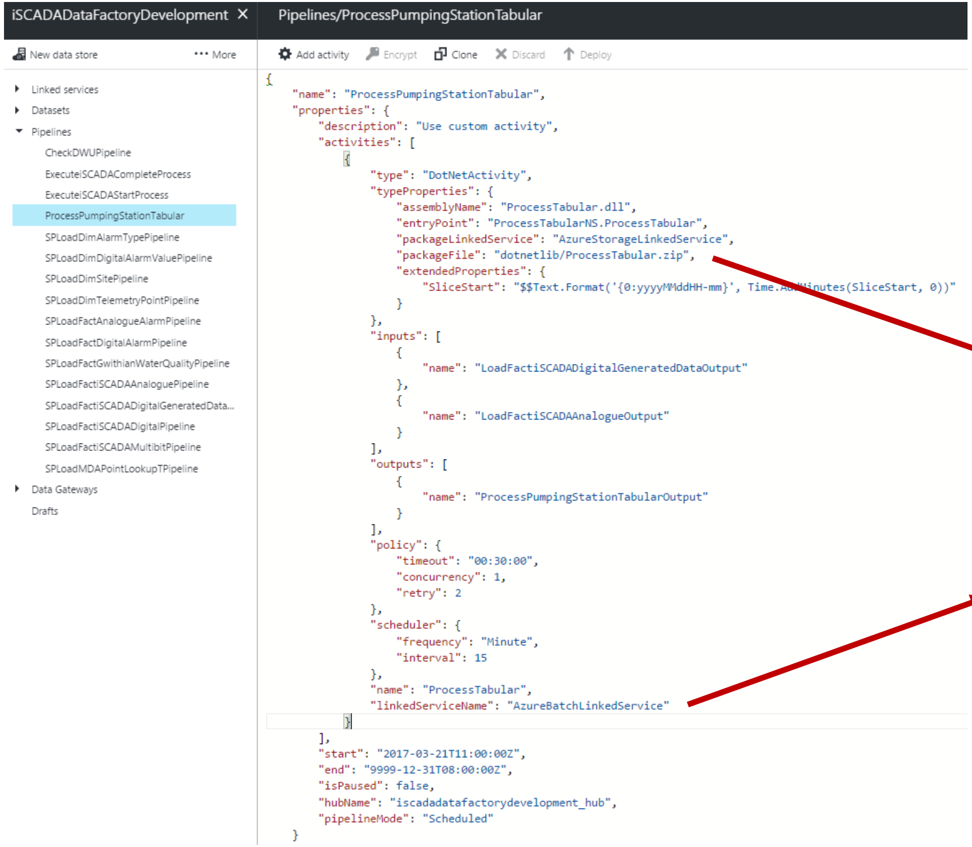
```
using Microsoft.AnalysisServices.Tabular;
using static Microsoft.AnalysisServices.Tabular.Database;

...
public IDictionary<string, string> Execute(
            IEnumerable<LinkedService> linkedServices,
            IEnumerable<Dataset> datasets,
            Activity activity,
            IActivityLogger logger)
{
        ...
        if (currentDateTime.Date == lastProcessDate.Date)
        {
            // If during the day, process only current partition
            model.Tables[tableName[i]].Partitions[0].RequestRefresh(RefreshType.Full);
            ...
        }
        else
        {
            // If new day, process full
            model.RequestRefresh(RefreshType.Full);
            ...
        }
        ...
}
```

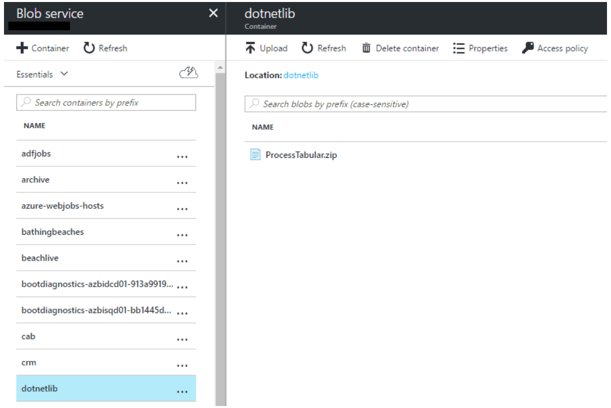
How to run C# from ADF using Azure Batch

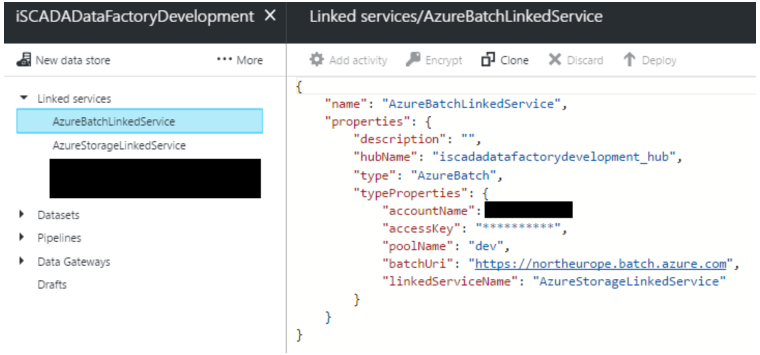
1. Create a **.NET Class Library** project with just the Execute method of the IDotNetActivity interface
2. Build it, create a zip file of the binaries, and upload to Azure Blob Storage
3. Create an Azure Batch account and pool
4. Add a pipeline to Azure Data Factory solution to run the C# code using Azure Batch

https://docs.microsoft.com/en-us/azure/data-factory/data-factory-use-custom-activities

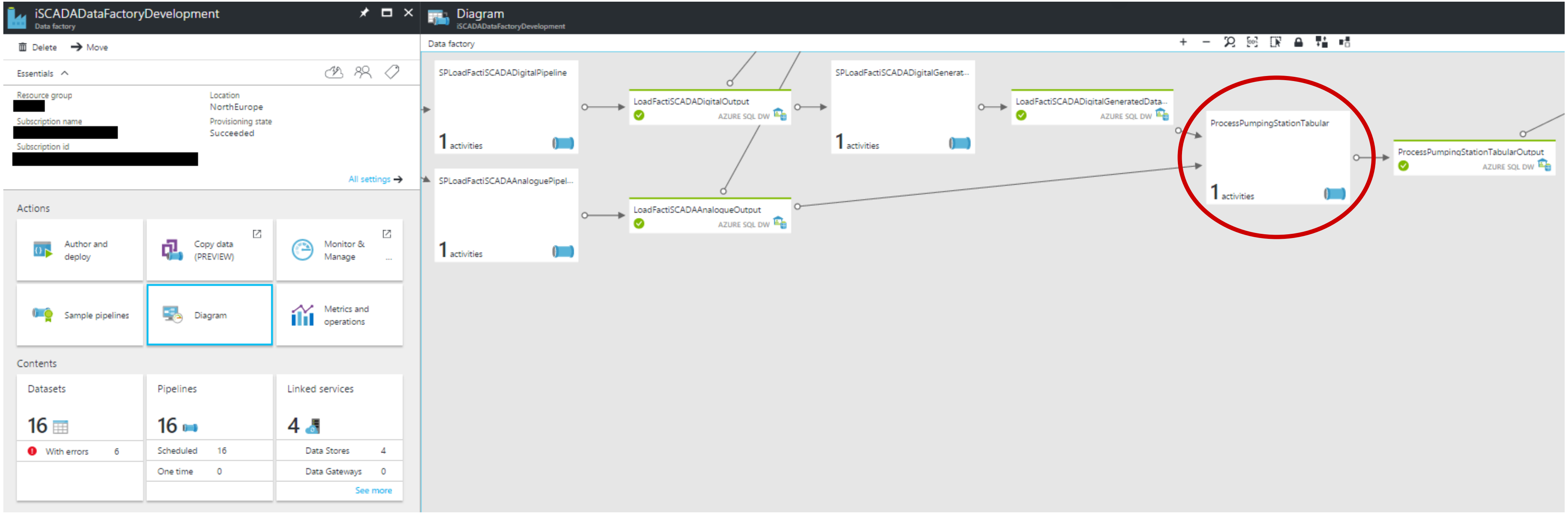
Process Tabular Model from Azure Data Factory





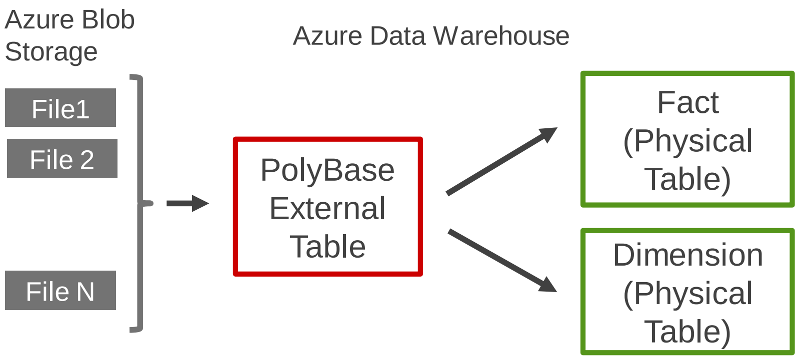


Add pipeline to ADF to process Tabular Model



Improvement 2 – Archive Blob Files

HITACHI
Inspire the Next




```
CREATE EXTERNAL TABLE [stg].[iSCADADigital]
(
[dv_id] [bigint] NULL,
[db_addr] [int] NULL,
[time] [datetime2](7) NULL,
[value] [bit] NULL,
...
...
)
WITH
( DATA_SOURCE = [iSCADAAzureStorage],
  LOCATION = N'digital/',
  FILE_FORMAT = [PipeDelimitedText],
  REJECT_TYPE = VALUE,
  REJECT_VALUE = 0)
```

With data growing rapidly, even using PolyBase, loading data into Azure Data Warehouse physical table slows down

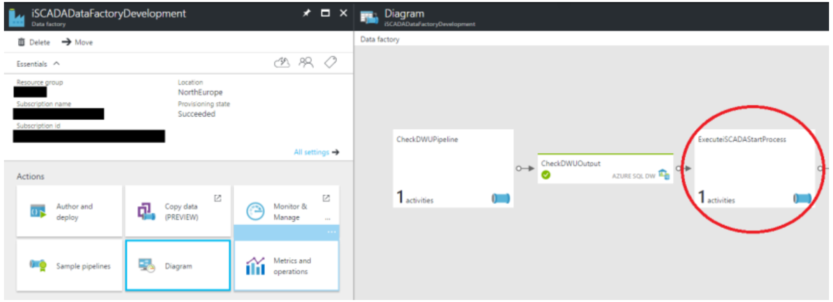
- Archive Blob Files that have been loaded into Data Warehouse physical table, so the external table points to only the new files
- Archiving is done from SSIS retrospectively

(Alternative would be to define PolyBase on a file (rather than a folder) but it would have to be on the fly)

Improvement 3 – Streamline SSIS & ADF

- Minimise latency between SSIS completion and ADF start
- Create a Stored Proc that will be the first task in ADF that will check for the completion of SSIS upload of files into Azure Blob

```sql
CREATE PROC [ctl].[iSCADAStartProcess] AS

BEGIN
       ...
       -- Check Upload Status
        WHILE ISNULL(@ProcessStatus, '') = ''
        BEGIN
           SET @ProcessStatus = (SELECT TOP 1 ProcessStatus
                                 FROM   ctl.ProcessAudit
                                 WHERE  ProcessName = ‘SSISUpload'
                                        AND ProcessStatus = 'END');
        END
       -- Now log the start of Azure Data Factory Phase
       -- And load data into Azure Data Warehouse
       ...
END
```

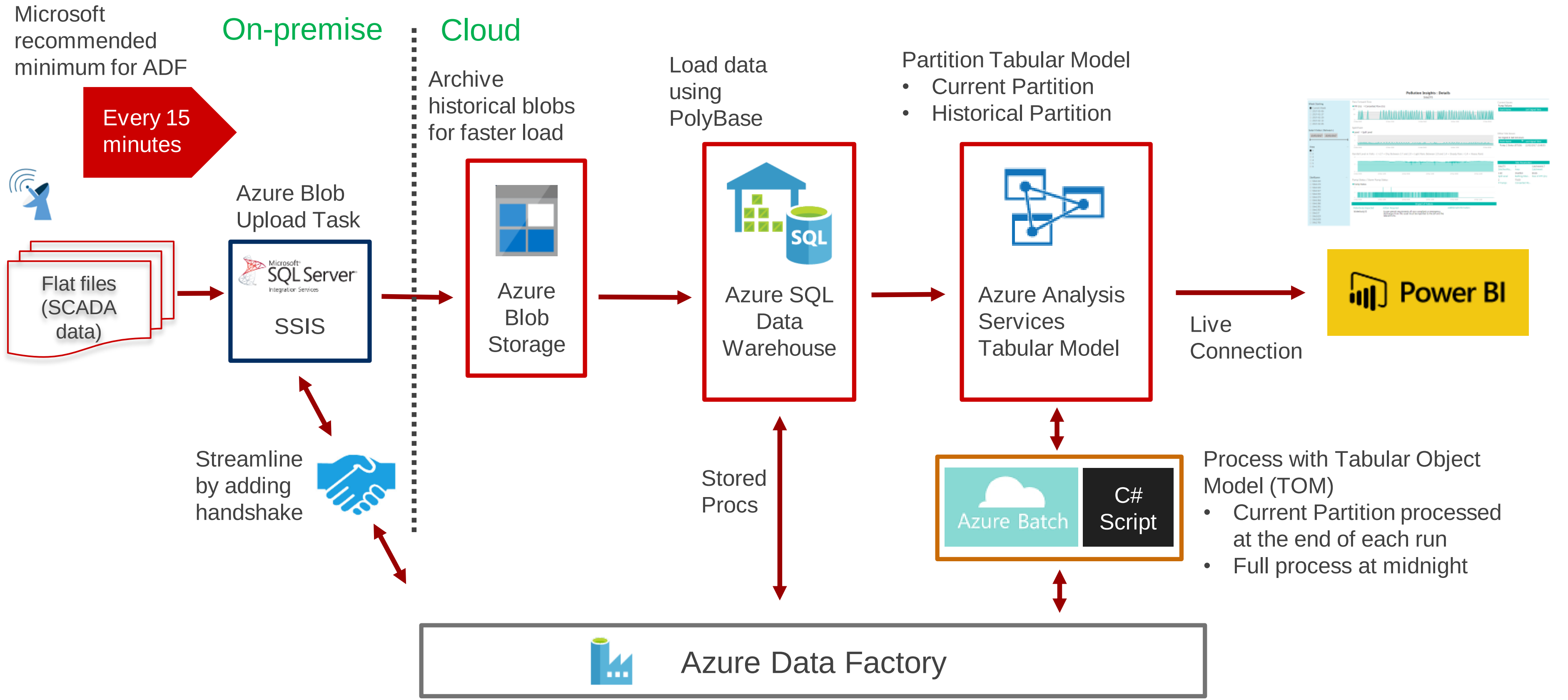


Pipelines/ExecuteiSCADAStartProcess

Add activity Encrypt Clone Discard Deploy

```json
{
    "name": "ExecuteiSCADAStartProcess",
    "properties": {
        "activities": [
            {
                "type": "SqlServerStoredProcedure",
                "typeProperties": {
                    "storedProcedureName": "ctl.iSCADAStartProcess",
                    "storedProcedureParameters": {}
                },
```

Architecture



How to improve this solution

- As data grows, may need a more sophisticated partitioning scheme for Tabular Model. Currently 45 million rows (Aug 16 – Mar 17). With new sites added, it will be around 400 million rows (600 sites, 1 years' data).
- Automatically scale up Azure Analysis Services when doing full processing overnight, and then scale down
- Use machine learning to find the correlation between signals – this could help improve the logic to predict the likelihood a spill (Utilising Azure Stream Analytics & Azure Machine Learning)
- Store historical RAG Status and Risk Level, and use Machine Learning to predict future RAG Status and Risk Level
- Other ideas:
    - Detect pump blockages from Flow & Pump Status
    - Pump energy use vs Flow rate – determine whether pumps are efficient or need to be serviced

Comparison to alternative solutions

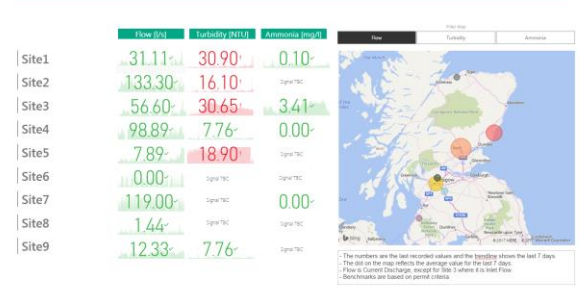
Solution	Handle complex rules	Handle out of order events	Latency	Visualisations	Cope with large amounts of data	Historical data analysis alongside near real-time data	Scale up & down	Cost
Azure Data Warehouse & Analysis Services	Yes	Yes	~ 15 min (Microsoft recommended minimum for ADF)	PowerBI	Yes	Yes	Yes	Pay-as-you-go (PaaS)
Azure Stream Analytics	May struggle (SQL & reference data from Azure Blob but no UDFs, no extensible code)	Yes	Low	PowerBI	Yes	Results need to be stored in DW	Yes	Pay-as-you-go (PaaS)
Storm on HDInsight	Code in Java or C#	Has to be implemented	Low	PowerBI	Yes (Very Large)	Results need to be stored	Yes	Pay-as-you-go (PaaS)
Spark on HDInsight (Spark Streaming)	Scala or Java	Yes (by batching data)	Batching adds some latency	PowerBI	Yes (Very Large)	Results need to be stored	Yes	Pay-as-you-go (PaaS)
On-premises SQL Server & Analysis Services	Yes	Yes	Low (Although loading data may be slow)	SSRS or with on-premises gateway in PowerBI but may need ExpressRoute for performance	Not as powerful as Azure DW	Yes	No	High initial setup cost, and later upgrade cost

A lot more can be done with Waste Water IoT data

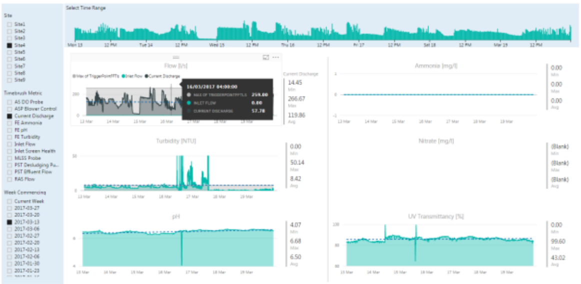
HITACHI
Inspire the Next

- Another similar but larger project we did was on **Waste Water Treatment Works**
- Common dashboards for all sites to analyse performance, identify issues, prevent failures
- Dashboard for each phase of treatment: Final Effluent → Inlet & PST → Filters & HST → ASP → FST etc.

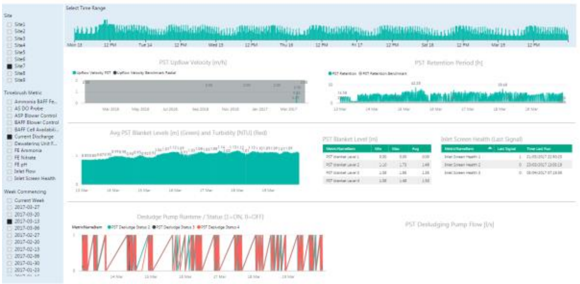
WWTW Special Measures : Performance Overview



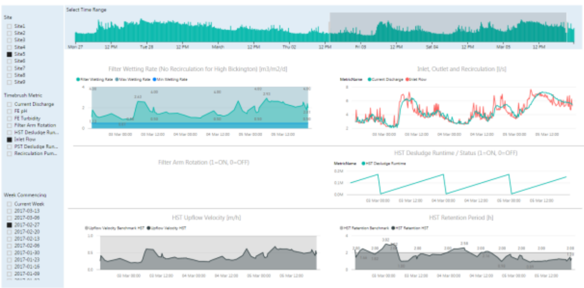
WWTW Special Measures : Final Effluent



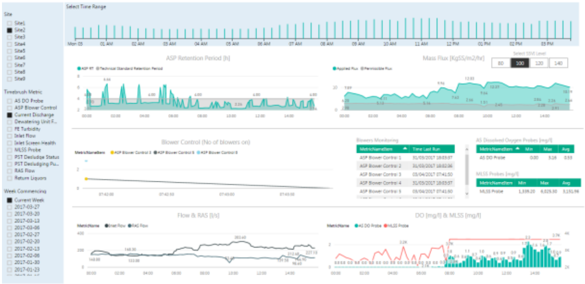
WWTW Special Measures : Inlet & PST



WWTW Special Measures : Filters and HST



WWTW Special Measures : ASP



WWTW Special Measures : FST

Contact

LinkedIn: https://www.linkedin.com/in/mehmet-bakkaloglu-0a328010/

HITACHI
Inspire the Next