FP Data Solutions

Analiza Big Data z Azure Data Lake

Tomasz Krawczyk

FP Data Solutions

Agenda

- Big Data
- Azure Data Lake
- Azure Data Lake Store and Analytics
  - U-SQL (Extentions)
  - Azure Data Lake Analytics Runtime
  - Costs and Optimization
  - Azure Data Lake and Azure Data Factory
- Demo and Q&A

FP Data Solutions

Big Data - introduction

www.future-processing.com

FP Data Solutions

163 Zetta bytes by 2025

163 Zetta bytes =

163 000 000 000 000 000 000 000 bytes

163 $*2^{70}$ bytes 163 * 1024^{7} bytes

If 1 bytes = One grain of rice

then Zetta byte = Fills the Pacific Ocean

Watching the Netflix catalog

489 million times

FP Data Solutions

3Vs of Big Data

- Data Volume
  - **Byte** One grain of rice
  - **Kilobyte** Cup of rice
  - **Megabyte** 8 bags of rice
  - **Gigabyte** 3 semi trucks
  - **Terabyte** 2 container ships
  - **Petabyte** Blankets Manhattan
  - **Exabyte** Blankets west coast states
  - **Zettabyte** Fills the Pacific Ocean
  - **Yottabyte** **As earth-sized rice ball**
- Data Variety
  - Structured
  - Unstructured
  - Semi-structured
  - All the above
- Data Velocity
  - Near to Real Time
  - Batch

FP Data Solutions

Schema-on-Read vs Schema-on-Write

SCHEMA-ON-READ (HADOOP OR ADLS):

- Copy data in its native format
- Create schema + parser
- Query Data in its native format (does ETL on the fly)

SCHEMA-ON-WRITE (RDBMS):

- Create static DB schema
- Transform data into RDBMS
- Query data in RDBMS format

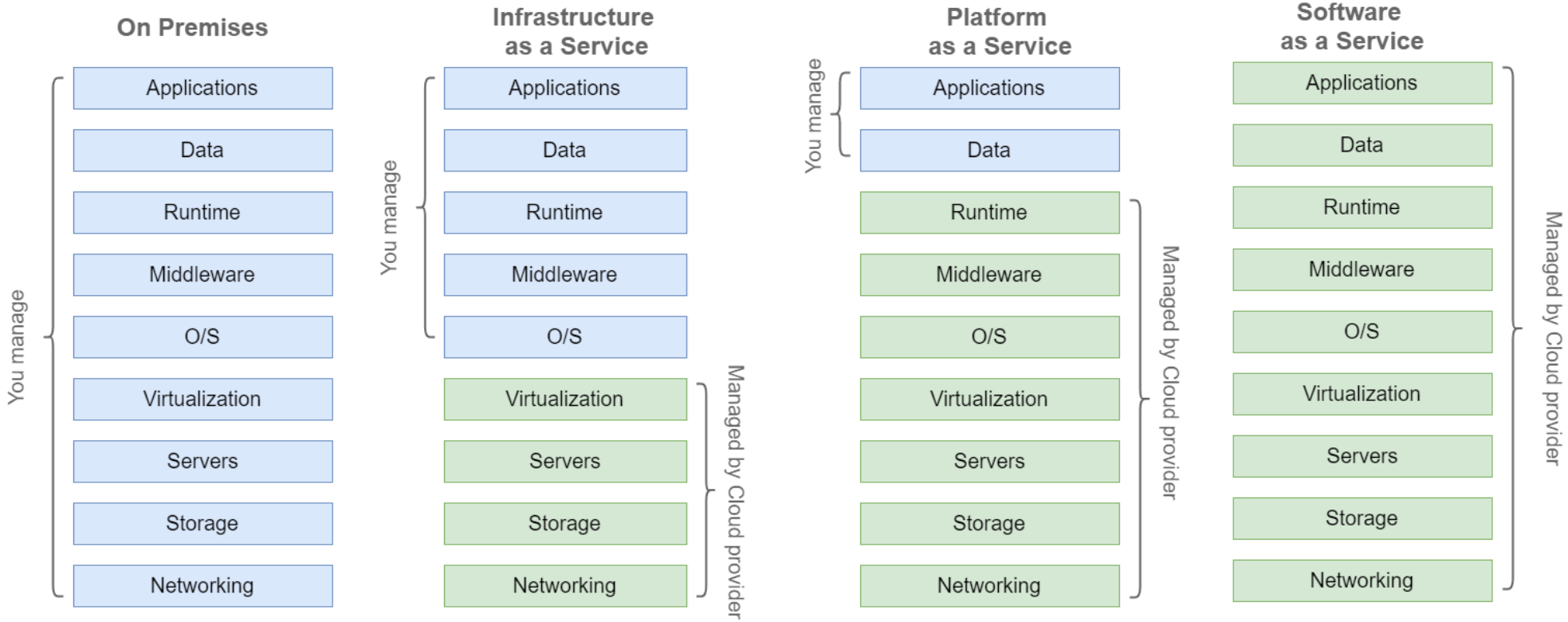
FP Data Solutions

How do you process all the data ?
Why Cloud?

www.future-processing.com

FP Data Solutions

Cloud Models



FP Data Solutions

Azure – Services

FP Data Solutions

Data Lake Approach

What is Data Lake ?

"If you think of a **datamart** (a subset of a data warehouse) as a store of bottled water – cleansed and packaged and structured for easy consumption – the **data lake** is a large body of water in a more **natural state**„

Pentaho CTO James Dixon

Source: https://premiumwaters.com

Source :https://snowbrains.com

I(ngest) S(tore) A(nalyse) S(urface) A(ct)

Make Me More Money

FP Data Solutions

Data Lake ?

Ingest all data
regardless of requirements

Store all data
in native format without schema definition

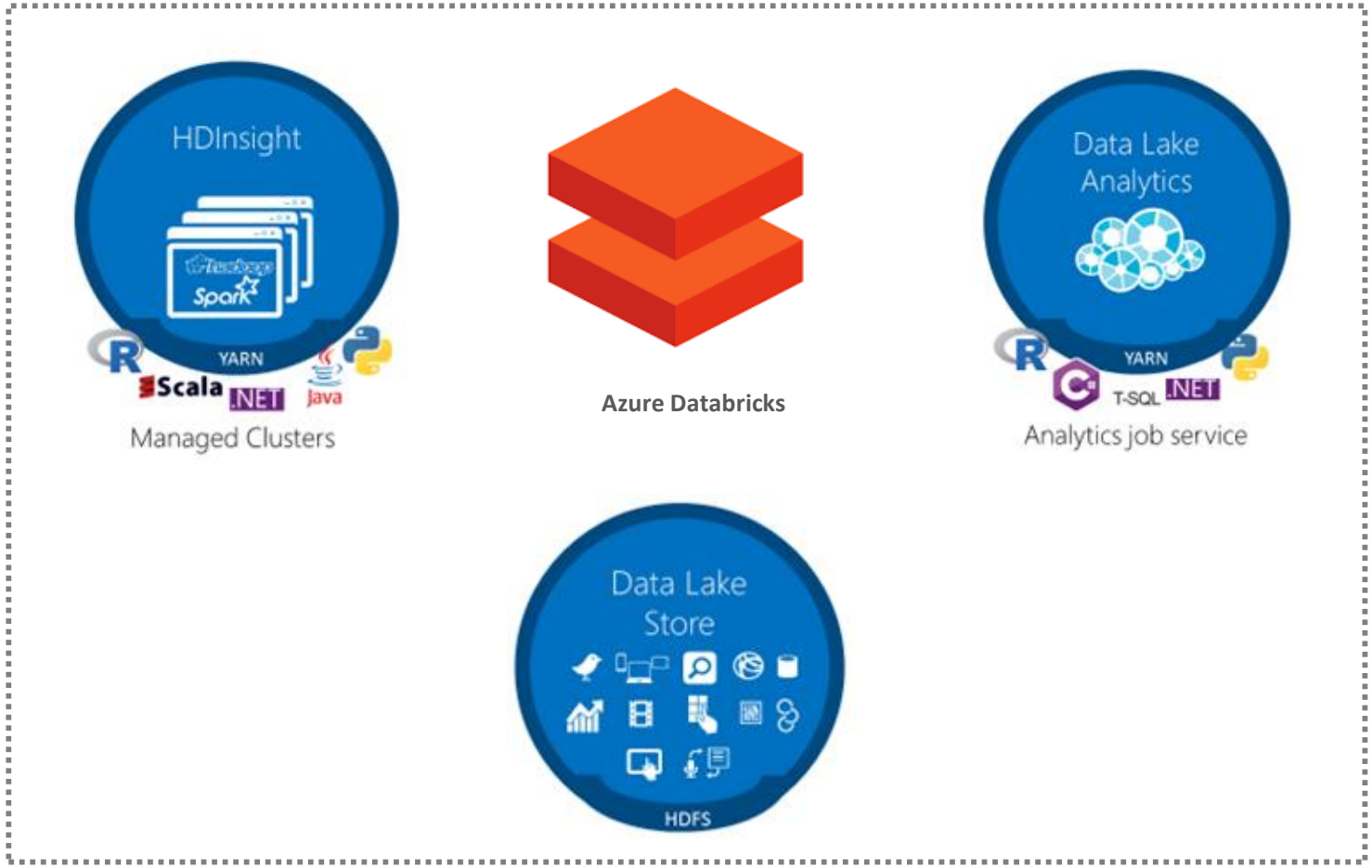
Do analysis
Using analytic engines like Hadoop

FP Data Solutions

Big Data on Azure

Analytics

Storage



FP Data Solutions

Azure Data Lake Store (Gen1 & Gen2)

- Built for Hadoop
- WebHDFS-compatible REST interface
- Unlimited storage, petabyte files
- Highly-available and secure
- Supports files and folders objects
- Files are split apart into Extents (250 MB)
- For availability and reliability, extents are replicated (3 copies).
- Enables: Parallel read and Parallel write

FP Data Solutions

Azure Data Lake Store vs Azure Blob Storage

	Azure Data Lake Store	Azure Blob Storage
Purpose	**Optimized storage for big data analytics workloads**	General purpose object store for a wide variety of storage scenarios
Use Cases	Batch, interactive, streaming analytics and machine learning data such as log files, IoT data, click streams, large datasets	Any type of text or binary data, such as application back end, backup data, media storage for streaming and general purpose data
Key Concepts	Data Lake Store account contains folders, which in turn contains data stored as files	Storage account has containers, which in turn has data in the form of blobs
Structure	**Hierarchical file system**	Object store with flat namespace
API	REST API over HTTPS	REST API over HTTP/HTTPS
Hadoop File System Client	Yes	Yes
Data Operations - Authentication	Based on Azure Active Directory Identities	Based on shared secrets - Account Access Keys and Shared Access Signature Keys.
Data Operations - Authorization	**POSIX Access Control Lists (ACLs). ACLs based on Azure Active Directory Identities can be set file and folder level.**	For account-level authorization – Use Account Access Keys For account, container, or blob authorization - Use Shared Access Signature Keys

FP Data Solutions

Azure – Big Data Storage

Azure Blob Storage

- General purpose object store
- Object store with flat namespace
- Hot/cold/archive tiers
- Data replication and redundancy options

Azure Data Lake Storage (Gen1)

- Unlimited storage, petabyte files
- **WebHDFS**-compatible REST interface
- Hadoop and big data optimizations
- Supports files and folders objects

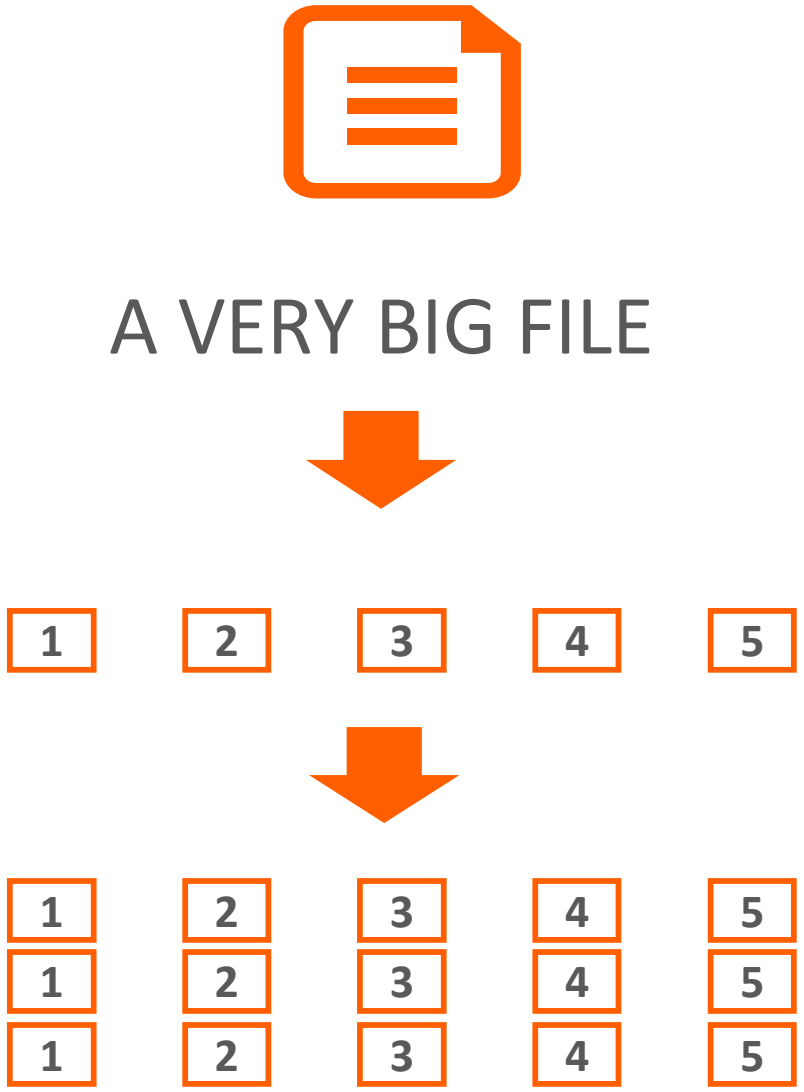
Azure Data Lake Storage (Gen2)

- Multi-modal combining features from both of the above
- Not a separate service: Azure Storage with new features

FP Data Solutions

Azure Data Lake Storage

- Files are split apart into Extents (250 MB)
- For availability and reliability, extents are replicated (3 copies).
- Enables:
  - Parallel read
  - Parallel write



FP Data Solutions

Security in Azure Data Lake Storage

- Authentication (Azure Active Directory)
- Authorization (RoleBaseAccessControl)
- Network isolation (Firewall and virtual networks)
- Data protection (Encryption +Azure Key Vault)
- Auditing
- Set file expiry

FP Data Solutions

Working with Azure Data Lake Store

- Developer
  - Azure Portal
  - Azure PowerShell
  - Azure Cross-platform CLI
  - Using Data Lake Tools for Visual Studio
  - Azure Storage Explorer
- Production
  - Azure Portal
  - Azure PowerShell
  - Azure Cross-platform CLI
  - Azure Data Factory
  - Apache Sqoop
  - **WebHDFS-compatible REST interface**

FP Data Solutions

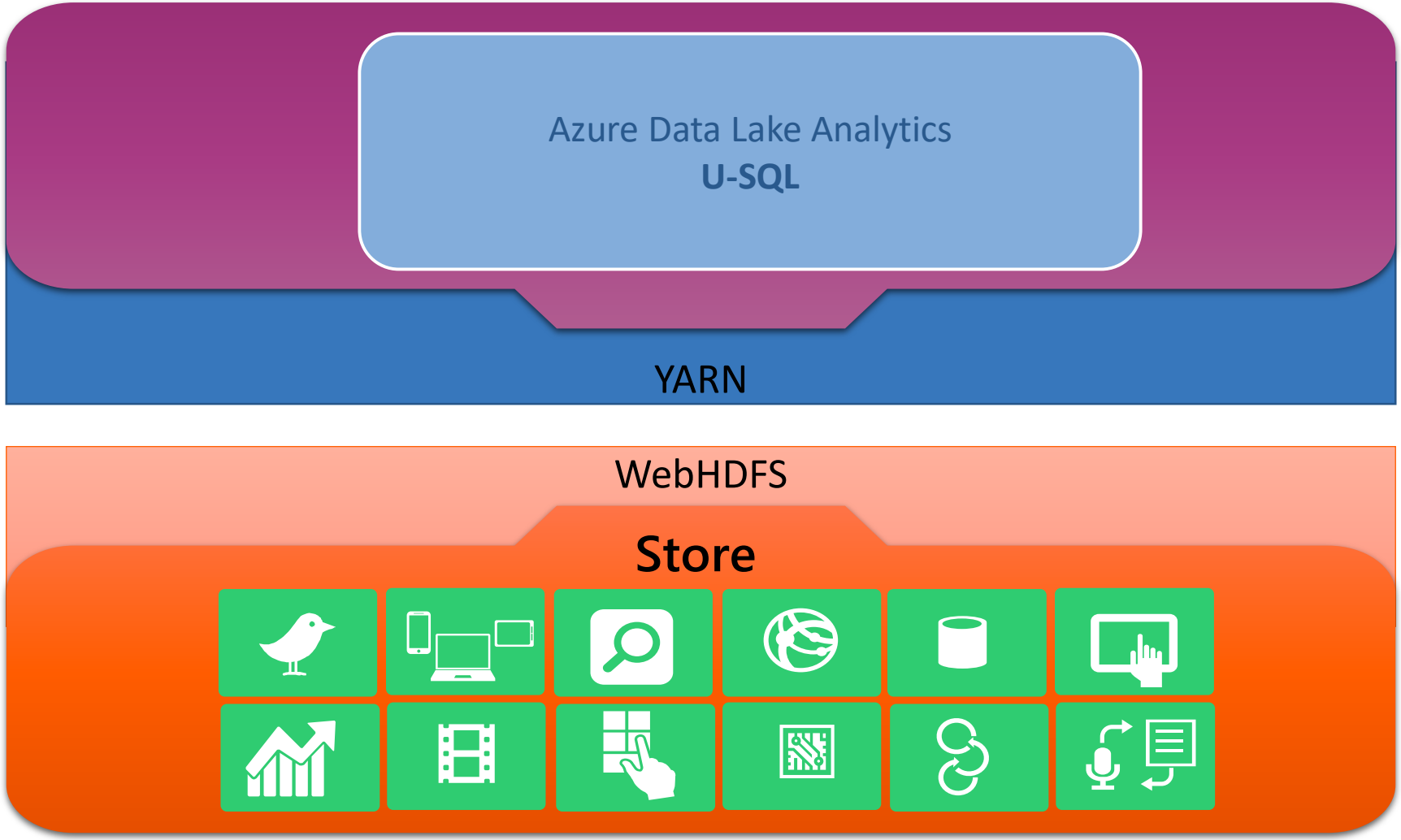
Azure Data Lake Analytics

A distributed analytics service built on Apache YARN that dynamically scales to your needs

- Pay **PER QUERY** & Scale **PER QUERY**
- **FEDERATED QUERY** across Azure data sources
- Includes **U-SQL**, a language that unifies the benefits of SQL with the expressive power of C#
- No limits to **SCALE**
- Optimized to work with **ADL STORE**

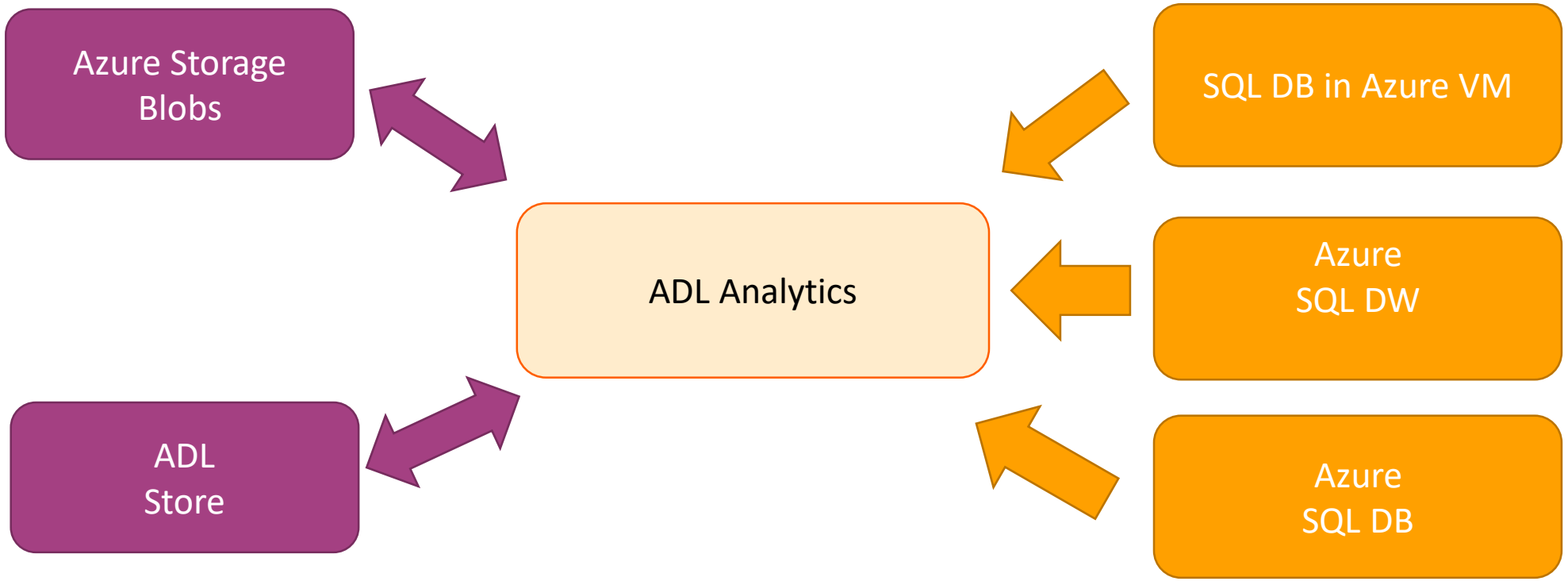
FP Data Solutions

Azure Data Lake Analytics



FP Data Solutions

ADLA Sources and Sinks



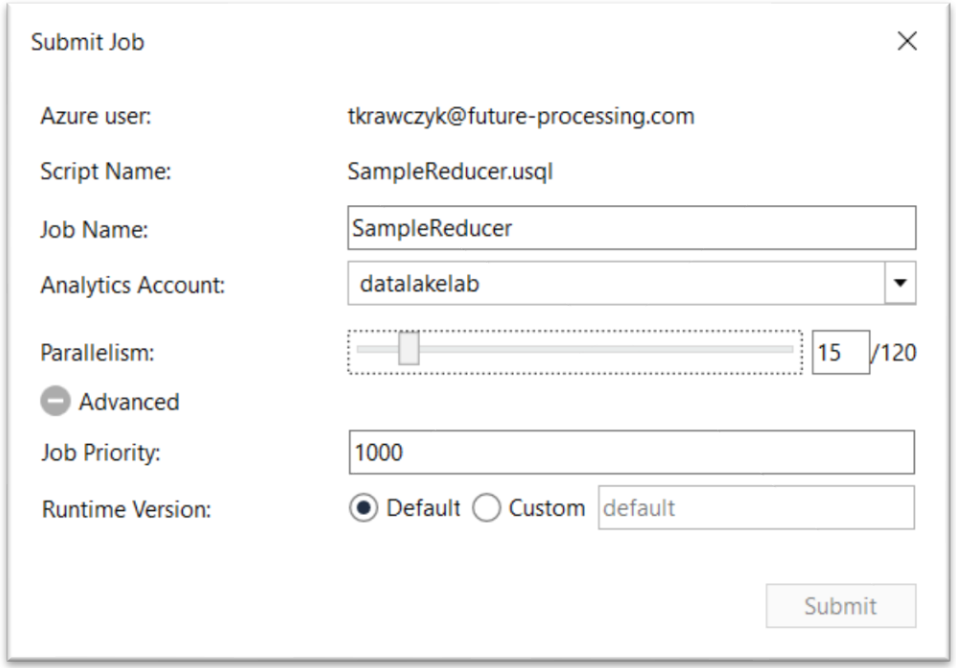
FP Data Solutions

Data Lake Analytics Runtime

1 ADLAU ~= A VM with 2 cores and 6 GB of memory

- Limited Network Access *

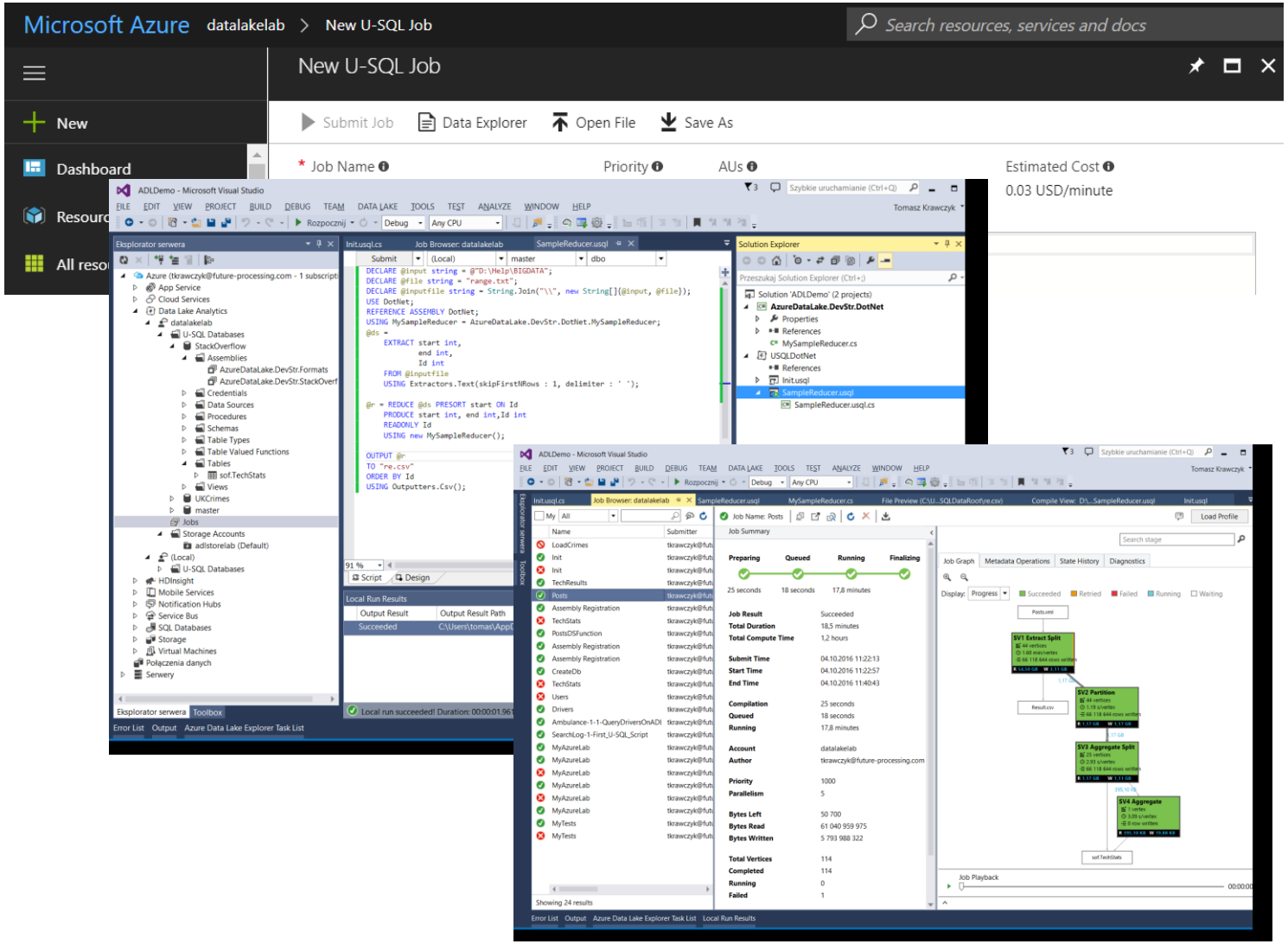
Parallelism N = N ADLAUs



FP Data Solutions

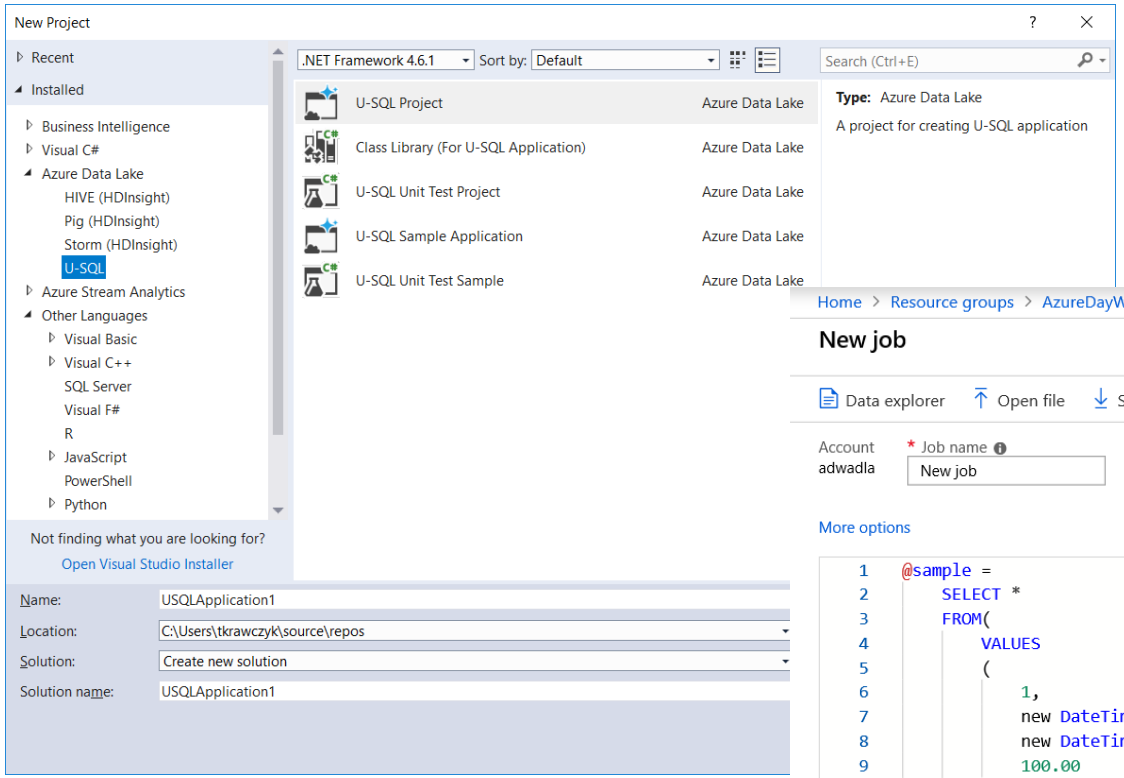
Data Lake Analytics Developer Tools

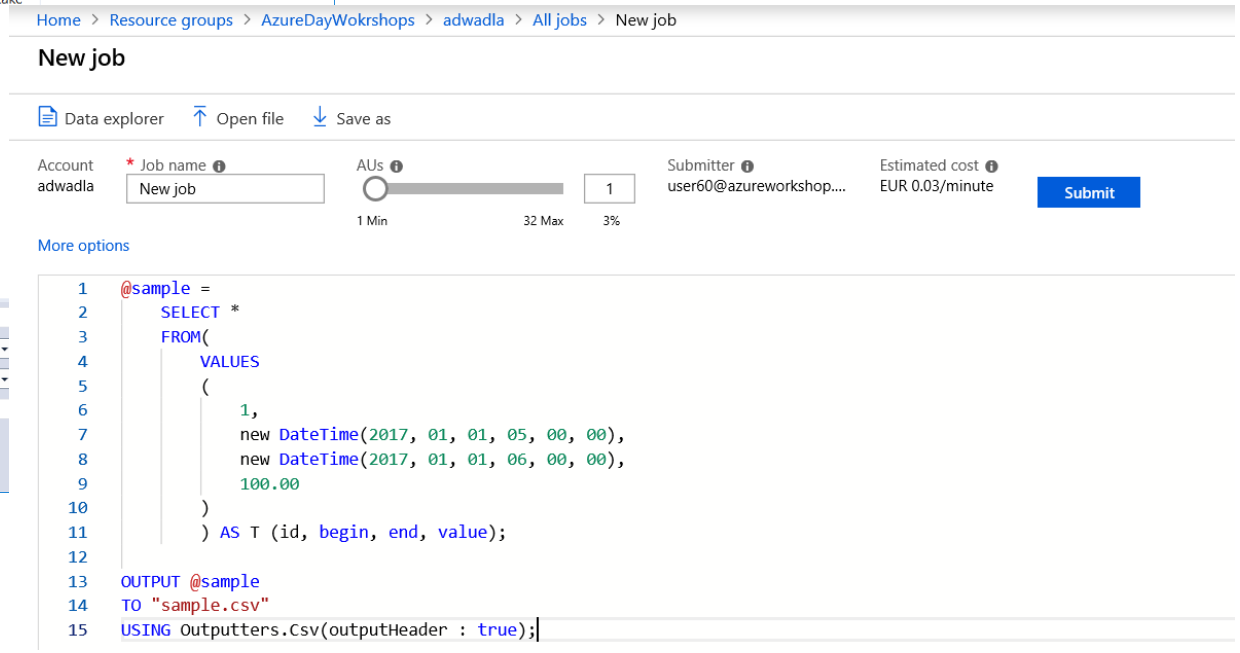
- Azure Portal
- Visual Studio
  - ADLA Tools
- Visual Studio Code
- Power Shell



FP Data Solutions

• Data Lake Analytics Developer Tools





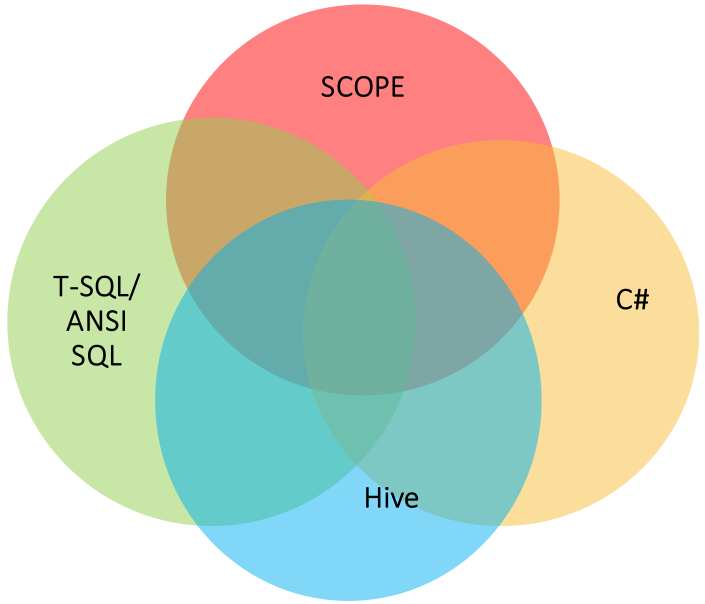
DEMO

- Create ADLA
- Run Sample Job
- Visual Studio Data Lake Analytics Tools
- Task1, Task2, Task3

FP Data Solutions

U–SQL - new language for Big Data

- Familiar syntax to millions of SQL & .NET developers
- Unifies declarative nature of SQL with the imperative power of C#
- Unifies structured, semi-structured and unstructured data
- Distributed query support over all data



SQL DECLARATIVITY + C# EXTENSIBILITY = U-SQL

Azure Data Lake

U–SQL Language Overview

FP Data Solutions

```
DECLARE @projectsInput string = @"Projects\{file}.csv";
DECLARE @eventDate DateTime = System.DateTime.Parse("2018/10/15");
DECLARE @numbers int = 2;
REFERENCE ASSEMBLY USQLCSharpDemo;
USING ImageColorsProcessor = USQLCSharpDemo.ImageColorProducer;
@projects =
    EXTRACT project string,
            startDate DateTime,
            endDate DateTime,
            file string
    FROM @projectsInput
    USING Extractors.Csv(skipFirstNRows : 1, quoting : true);
@rs =
    EXTRACT content byte[],
            fileName string
    FROM @imgFiles
    USING new BinaryExtractor();
@assignments =
    SELECT user.ToUpper() AS user,
           new SqlArray<string>(projects.Split(new char[]{'|'},
StringSplitOptions.RemoveEmptyEntries)) AS projs
    FROM @usersprojects;
@agg =
    SELECT project,
           COUNT( * ) AS units
    FROM @details WHERE project.StartsWith("My")
    GROUP BY project;
@myprojects =
    SELECT us.project,
           p.endDate
    FROM @details AS us
         JOIN
             @projects AS p
         ON p.project == us.project
    WHERE user.StartsWith("Me")
ORDER BY p.endDate DESC
FETCH 10 ROWS;
OUTPUT @myprojects
TO "myprojects.csv"
USING Outputters.Csv();
```

- C# Types
- .NET Assemblies
- Apply Schema on Read
- Extractor
- Rowset(s)
- External Extractor
- SQL Dialect (SELECT FROM)
- SQL Aggregation(s)
- .NET Methods
- SQL Dialect (JOIN, WHERE, ORDER BY ...)
- Output

FP Data Solutions

U-SQL Data Types

Numeric
- sbyte
- int
- long
- float
- double
- decimal
- short
- byte
- uint
- ulong
- ushort

Text
- char
- String (128 kB)

› Complex
- MAP<k,v>
- ARRAY<v>

› Miscellaneous
- bool
- Guid
- DateTime
- byte[]

› RowSets

SQL.ARRAY<T> == IList<T>
SQL.MAP<T,U> == IDictionary<T,U>

```
@m = SELECT new SqlArray<string>
        (tweet.Split( new char[]{' '}).Where(x =>
        x.StartsWith("@"))) AS mentions
FROM @t;

@m = SELECT m.Substring(1) AS m,"mention" AS category
        FROM @m
        CROSS APPLY EXPLODE(mentions)
AS t(m);
```

FP Data Solutions

U-SQL Declare Variables

```
DECLARE @text1 string = “Big Data";
DECLARE @text2 string = @“Azure as a Big Data Platform";
DECLARE @text3 char = 'a';
DECLARE @text4 string = "BEGIN" + @text1 + "END";
DECLARE @text5 string = string.Format("BEGIN{0}END", @text1);
DECLARE @text6 string = string.Join(" ", new String[]{@text1,
“2017"});

DECLARE @numeric1 sbyte = 0;
DECLARE @numeric2 short = 1;
DECLARE @numeric3 int = 2;
DECLARE @numeric4 long = 3L;
DECLARE @numeric5 float = 4.0f;
DECLARE @numeric6 double = 5.0;

DECLARE @d1 DateTime = System.DateTime.Parse("1979/03/31");
DECLARE @d2 DateTime = DateTime.Now;

DECLARE @misc1 bool = true;
DECLARE @misc2 Guid = System.Guid.Parse("BEF7A4E8-F583-4804-9711-
7E608215EBA6");
DECLARE @misc4 byte [] = new byte[] { 0, 1, 2, 3, 4};
```

FP Data Solutions

U-SQL EXTRACT

```csv
Project,StartDate,EndDate
"BigBang",2015-01,2016-12
"BigData",2016-08,2016-10
"Project1",2016-01,2016-09
"Project2",2016-11,2016-12
"Project3",2014-01,2018-12
"Project4",2016-01,2017-08
"Project5",2016-01,2017-12
"Project6",2015-01,2017-02
"Project7",2015-01,2016-12
"Project8",2015-08,2016-08
"Project9",2014-01,2016-12
"NewProject",2017-12,2018-12
```

```
DECLARE @projectsInput string =
@"Projects\projects.csv";
@projects =
    EXTRACT project string,
            startDate DateTime,
            endDate DateTime
    FROM @projectsInput
    USING Extractors.Csv(
        skipFirstNRows : 1,
        quoting : true);
```

Apply Schema on Read

Extractor
with additional options

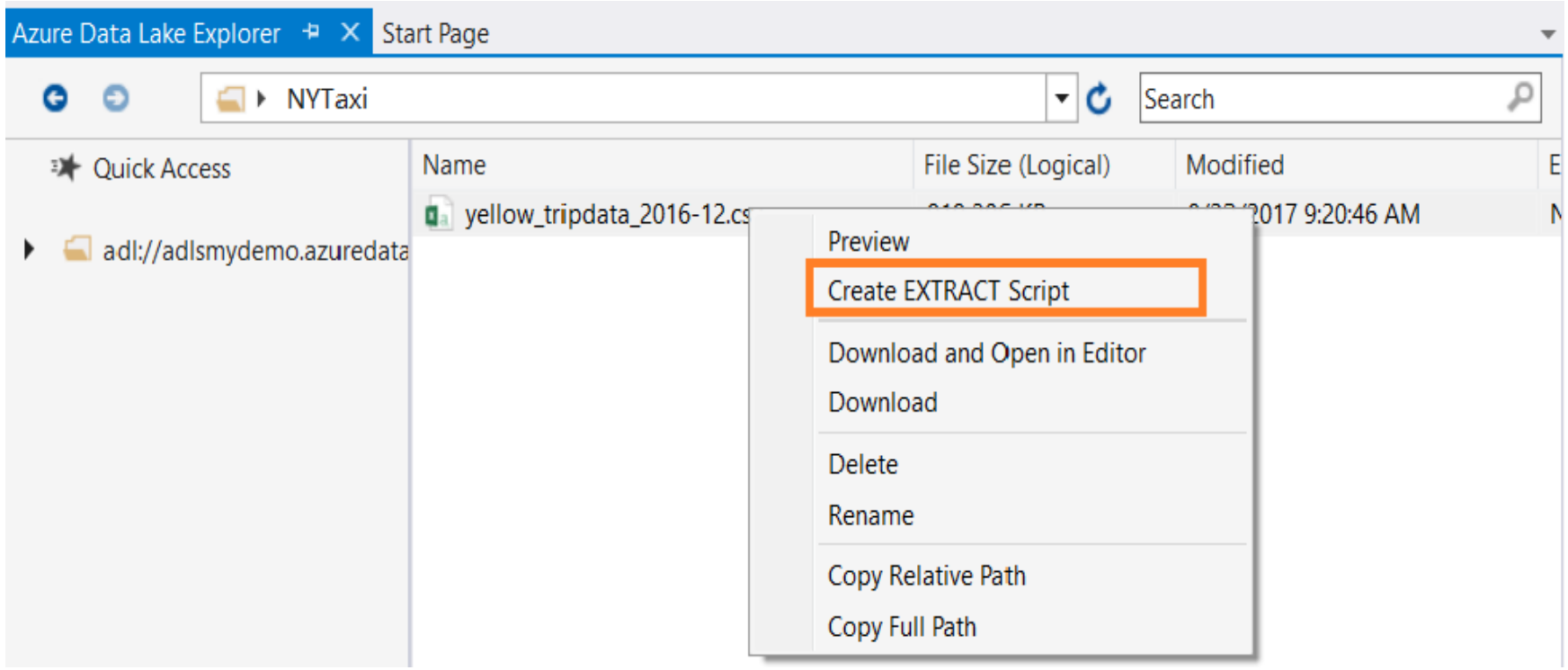
FP Data Solutions

U-SQL EXTRACTORS

- List of EXTRACTORS AND OUTPUTERS
  - CSV
  - TEXT
  - TSV
  - GZIP (*)
  - PARQUET (*)
- API
  - IExtractor
  - IOutputter

FP Data Solutions

U-SQL CREATE EXTRACT SCRIPT



FP Data Solutions

U-SQL ROWSETS

```
@postCodes =
    EXTRACT id string,
            postcode string,
            latitude string,
            longitude string
    FROM @inputPostCodes
    USING Extractors.Csv(skipFirstNRows:1);

@topCities =
    EXTRACT id int,
            name string,
            population string,
            postcode string
    FROM @input10topCities
    USING Extractors.Text(delimiter : ';');

@topCitiesWithGPS =
    SELECT tc.name,tc.population,
pc.latitude,pc.longitude
    FROM @topCities AS tc
         JOIN
             @postCodes AS pc
         ON pc.postcode == tc.postcode;
```

Rowset 1 (@postCodes)

Rowset 2 (@topCities)

Rowset 3 (@topCitiesWithGPS)

FP Data Solutions

U-SQL FILESETS

```
DECLARE @inputCrimes = @"mySamples/UKCrimes/
{Date:yyyy}-{Date:MM}/{Input}-street.csv";
@crimes =
    EXTRACT CrimeID string,
            Month string,
            ReportedBy string,
            FallsWithin string,
            Longitude string,
            Latitude string,
            Location string,
            LSOACode string,
            LSOAName string,
            CrimeType string,
            LastOutcomeCategory string,
            Context string,
            Date DateTime,
            Input string
    FROM @inputCrimes
    USING Extractors.Csv(silent : false,skipFirstNRows:1);
```

FileSets ({Date},{Input})

Virtual Columns

FP Data Solutions

U-SQL

FILTERING AND SORTING

- WHERE
  - AND & OR
  - ==, >=, != (C# OPERATOR(s))
  - CONTAINS (C# string)
- ORDER BY
  - ROWSETS
    - requires a FETCH
  - OUTPUTS

JOINS

- INNER JOIN
- FULL OUTER JOIN
- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- CROSS JOIN
- LEFT SEMIJOIN
- RIGHT SEMIJOIN
- LEFT ANTISEMIJOIN
- RIGHT ANTISEMIJOIN

FP Data Solutions

U-SQL

FILTERING AND SORTING

- WHERE
  - AND & OR
  - ==, >=, != (C# OPERATOR(s))
  - CONTAINS (C# string)
- ORDER BY
  - ROWSETS
    - requires a FETCH
  - OUTPUTS

JOINS

- INNER JOIN
- FULL OUTER JOIN
- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- CROSS JOIN
- LEFT SEMIJOIN
- RIGHT SEMIJOIN
- LEFT ANTISEMIJOIN
- RIGHT ANTISEMIJOIN

FP Data Solutions

U-SQL

AGGREGATIONS

- GROUP BY
- HAVING
- MAX ,MIN ,SUM,COUNT, ARRAY_AGG

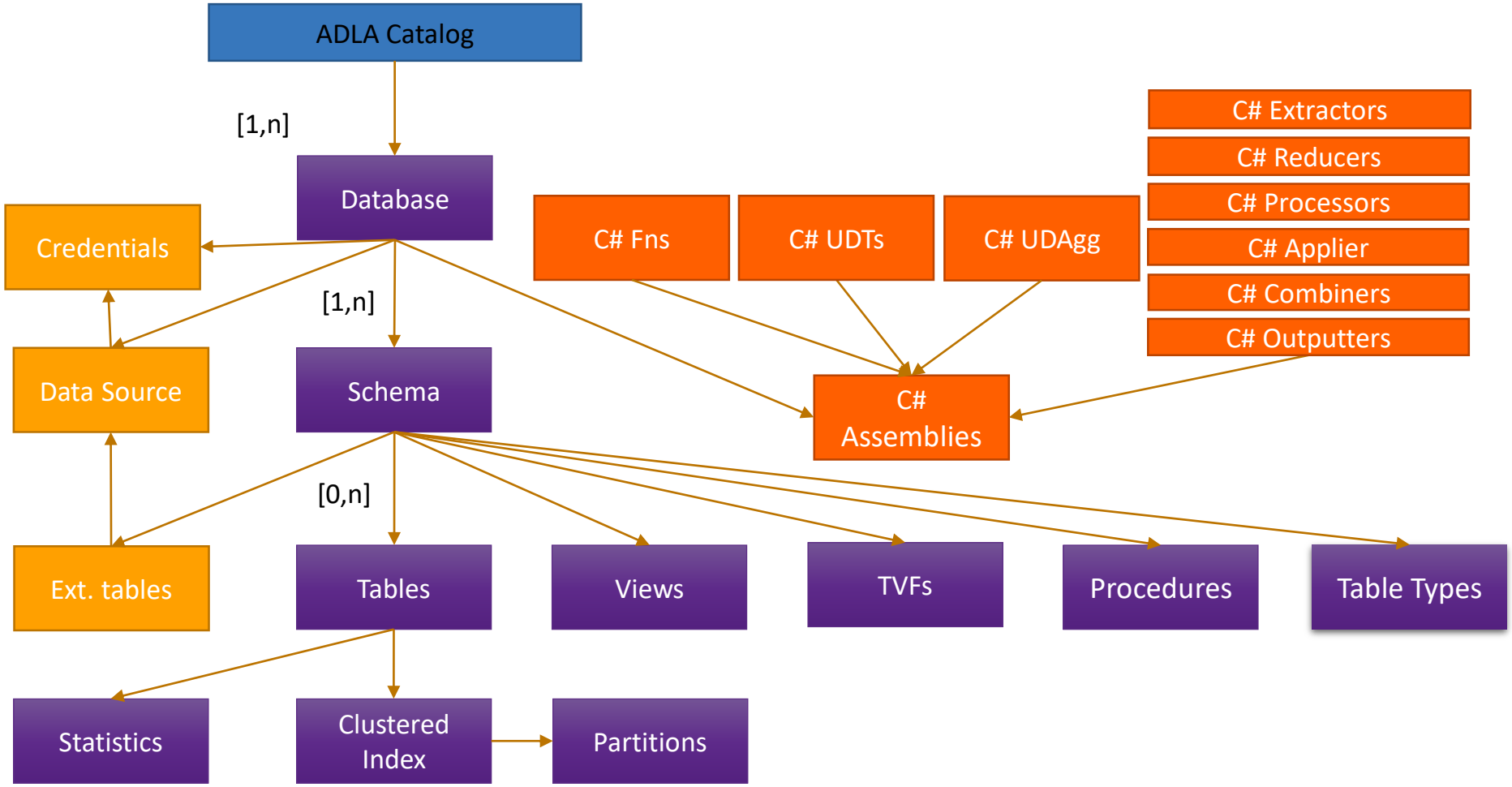
RANKING FUNCTIONS

- RANK
- DENSE_RANK
- NTILE
- ROW_NUMBER

ANALIYTIC WINDOW FUNCTIONS

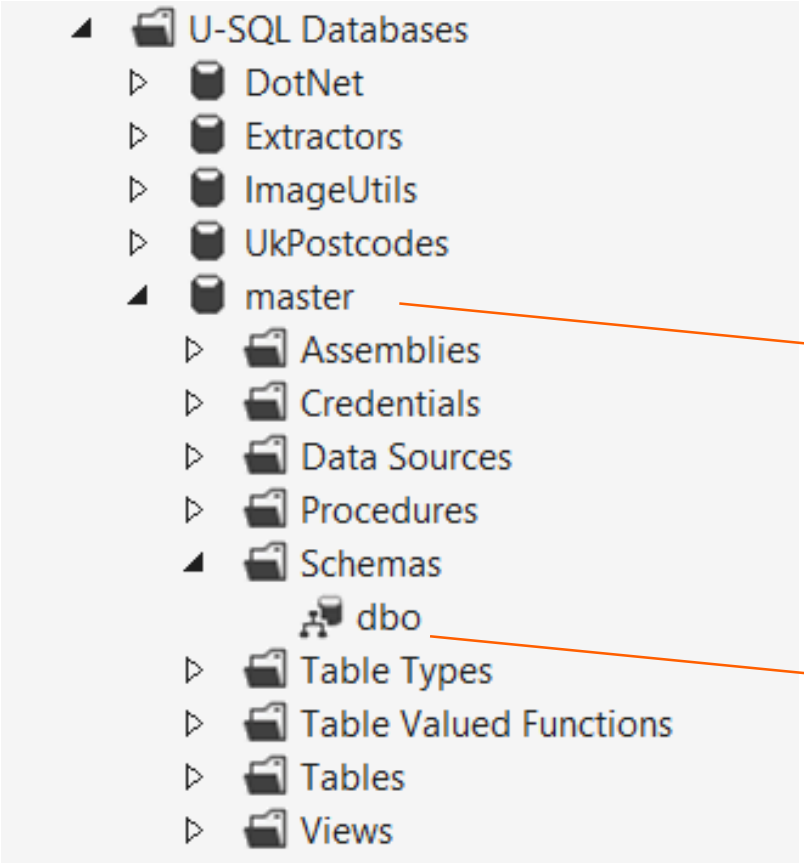
- CUME_DIST
- PERCENT_RANK
- PERCENTILE_CONT
- PERCENTILE_DISC
- CUME_DIST

FP Data Solutions



FP Data Solutions

U-SQL DATABASES




```
CREATE DATABASE IF NOT EXISTS UKCrimes;
USE DATABASE UKCrimes;
CREATE SCHEMA IF NOT EXISTS cr;
```

Default data base

Default schema

FP Data Solutions

U-SQL DATABASES

- MANAGED TABLES and EXTERNAL TABLES
- ONLY INSERT
- CONSISTS OF FOUR THINGS:
  - A NAME
  - COLUMNS
  - A CLUSTERED INDEX
  - DISTRIBUTION (PARTITIONING) SCHEME

```
CREATE TABLE T
(
    id int,
    date DateTime,
    INDEX IDX
    CLUSTERED(id)
    PARTITIONED BY (date)
    DISTRIBUTED BY
    HASH(id)
    INTO 4
);
```

Azure Data Lake

FP Data Solutions

U-SQL VIEWS and FUNCTIONS

VIEWS

```
CREATE VIEW IF NOT EXISTS vCrimes
    AS
EXTRACT CrimeID string,
        Month string,
        Date DateTime,
        Input string
FROM @"\UKCrimesCities\{Date:yyyy}-
{Date:MM}\{Input}-street.csv"
USING Extractors.Csv(silent : false,
 skipFirstNRows : 1);
```

FUNCTIONS

```
CREATE FUNCTION tvf_Crimes(@input string)
RETURNS @result TABLE(CrimeID string,
 Month string)
AS
BEGIN
    @crimes =
    EXTRACT CrimeID string,
            Month string
    FROM @input
    USING Extractors.Csv(silent : false,
skipFirstNRows:1);

    @result = SELECT CrimeID,
            Month
            Input FROM @crimes;
    END;
```

Azure Data Lake

FP Data Solutions

U-SQL STORED PROCEDURES

```
DROP PROCEDURE IF EXISTS DemoDb.dm.SampleSP;
CREATE PROCEDURE DemoDb.dm.SampleSP(@startDate DateTime,
@endDate DateTime, @outputName string)
BEGIN

    @sample =
        SELECT *
        FROM(
            VALUES
            (
                1,
                new DateTime(2017, 01, 01, 05, 00, 00),
                new DateTime(2017, 01, 01, 06, 00, 00),
                100.00
            )
 )
AS T (id, begin, end, value);


    @rs =
        SELECT id,
               begin,
               end,
               value
        FROM @sample
        WHERE begin >= @startDate AND end <= @endDate;

    OUTPUT @rs
    TO @outputName
    USING Outputters.Csv();
   END;
```

```
DECLARE @startDate DateTime =
            new DateTime(2017, 01, 01, 05, 00, 00);
DECLARE @endDate DateTime =
            new DateTime(2017, 01, 01, 07, 00, 00);
DECLARE @outputName string = "sp_result.csv";
DemoDb.dm.SampleSP
(
    @startDate,
    @endDate,
    @outputName
);
```

FP Data Solutions

U-SQL TOP 5'S SUPRICES FOR SQL DEVS

- U-SQL is case sensitive
- AS is not as
  - C# keywords and SQL keywords overlap
- = != ==
  - C# expression language
- null IS NOT NULL
  - C# nulls are two-valued (HasValue)
- No UPDATE, DELETE, nor MERGE

DEMO

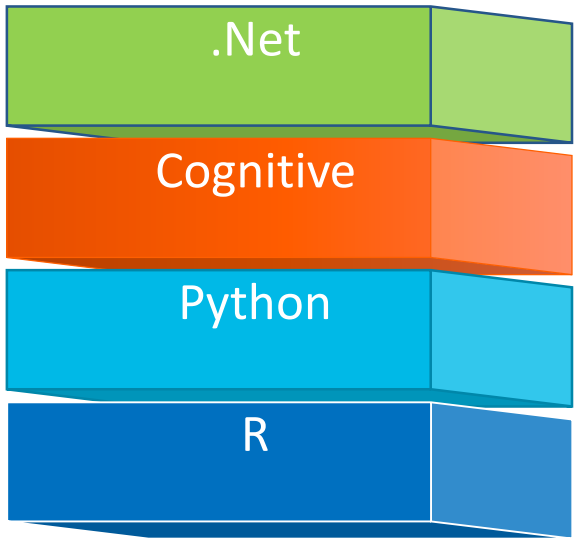
- Samples\Array
- Samples\Map
- Demo000IISLogs
- Demo001UKCrimes
- Task4

Azure Data Lake

FP Data Solutions

U-SQL EXTENTIONS

- .NET Extensions
- Built-in Cognitive capabilities
- Extensions for Massively Parallel processing
  - R Language
  - Python



FP Data Solutions

U-SQL .NET EXTENTIONS

- C# Functions/Methods
  - C# UDTs
  - C# UDAggs
- Extractors USQL: EXTRACT
  - Converts files into rowset
- Reducers USQL: REDUCE
  - Take n rows and produce m rows (normally m<n)
- Processors USQL: PROCESS
  - Take one row and produce one row
  - Pass-through versus transforming
- Appliers USQL: CROSS APPLY
  - Take one row and produce 0 to n rows
  - Used with OUTER/CROSS APPLY
- Combiners USQL: COMBINE
  - Combines rowsets (like a user-defined join)
- Outputters USQL: OUTPUT
  - Converts rowset into file

FP Data Solutions

U-SQL .Net Framework and System Assemblies

U-SQL uses the .NET Framework version 4.5
(code in a 64-bit (x64) format)

- Preloaded System Assemblies
  - mscorlib.dll
  - System.dll
  - System.Core.dll
  - System.Data.dll
  - Microsoft.Analytics.Interfaces.dll
  - Microsoft.Analytics.Types.dll

Example:

```
REFERENCE SYSTEM ASSEMBLY [System.XML];
```

FP Data Solutions

U-SQL Creating Extentions

- Methods
  - Class + static method

```
public class IpConverter
{
        public static string ToIp4Format(string ip)
        {
                return IPAddress.Parse(ip).MapToIPv4().ToString();
        }
}
```

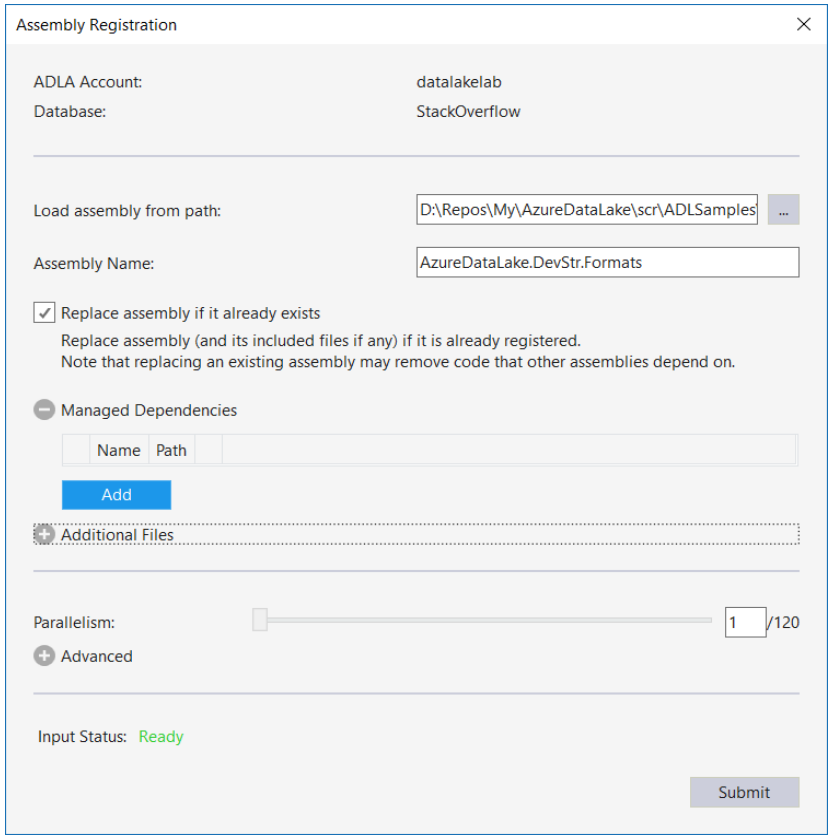
- Extractors, Reducers, Processors, Appliers, Combiners, Outputters
  - New project type: Class Library (For U-SQL Applications)
  - Abstract classes
    - IExtractors IReducer, IProcessor, IApplier, ICombiners, IOutputters

```
public class NameReverseProcessor : IProcessor
{
 public override IRow Process(IRow input, IUpdatableRow output)
 {
        var s = input.Get<string>("name");
        output.Set<string>("reversed", Reverse(s));
        return output.AsReadOnly();
 }
 private static string Reverse(string s)
 {
        char[] charArray = s.ToCharArray();
        Array.Reverse(charArray);
        return new string(charArray);
 }
}
```

FP Data Solutions

U-SQL Registering .Net Assemblies

```
DECLARE @AssemblyPath string = @"Assemblies/";
DECLARE @AssemblyExt string =
@AssemblyPath+"ADLAExt.dll";
USE 4Developers;
DROP ASSEMBLY IF EXISTS ADLAExt;
CREATE ASSEMBLY ADLAExt FROM @AssemblyExt;
```



FP Data Solutions

U-SQL Using .Net Assemblies

```
USE DATABASE [4Developers];
REFERENCE ASSEMBLY [ADLAExt];

USING IpConverter = ADLAExt.Utils.IpConverter;

@ds =
    SELECT IpConverter.ToIp4Format(c_ip) AS Ip,
           date.Date AS Date
    FROM @iisLogs;
```

Use data base (optional)

Add reference to assembly

Create alias

Use method from assembly

FP Data Solutions

U-SQL .Net Extractor

```
[SqlUserDefinedExtractor(AtomicFileProcessing = true)]
   public class BinaryContentExtractor : IExtractor
   {
       public override IEnumerable<IRow> Extract(IUnstructuredReader
input, IUpdatableRow output)
       {
           using (var ms = new MemoryStream())
           {
               input.BaseStream.CopyTo(ms);
               var content = ms.ToArray();
               output.Set(0, content);
               yield return output.AsReadOnly();
           }
       }
   }
```

WHOLE FILE

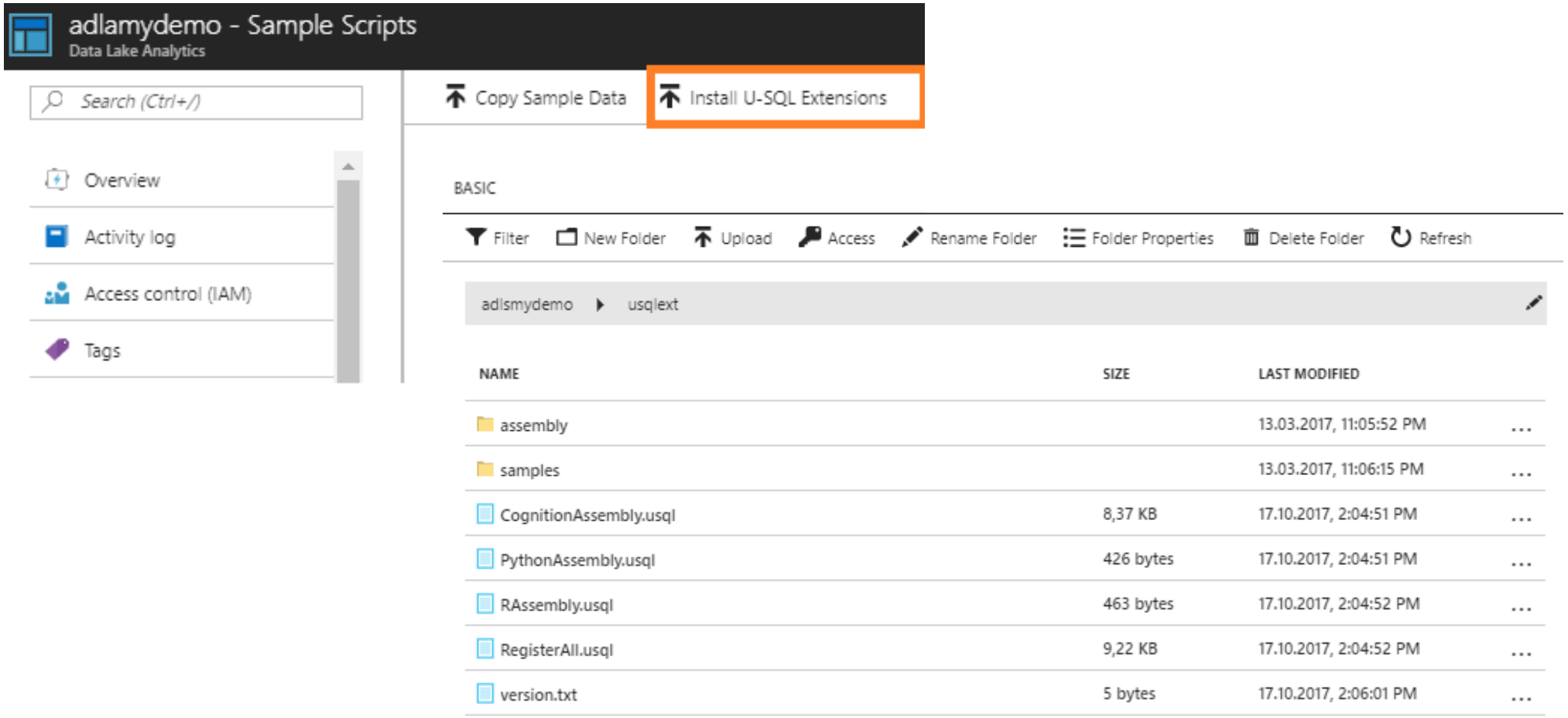
RETURN CONTENT IN FIRST COLUMN

```
@rs =
    EXTRACT content byte[],
            fileName string
    FROM @imgFiles
    USING new BinaryExtractor();
```

Azure Data Lake

FP Data Solutions

U-SQL Python, R Language, Cognitive



FP Data Solutions

U-SQL Python, R Language, Cognitive

- Register Assemblies
  - CognitionAssembly.usql
  - PythonAssembly.usql
  - RAssembly.usql
  - RegisterAll.usql

```
CREATE DATABASE IF NOT EXISTS master;
USE DATABASE master;

DROP ASSEMBLY IF EXISTS [ExtPython];
CREATE ASSEMBLY IF NOT EXISTS [ExtPython]
FROM @"/usqlext/assembly/python/ExtPy.dll"
WITH ADDITIONAL_FILES =
(
    @"/usqlext/assembly/python/ExtPy.pdb",
    @"/usqlext/assembly/python/UsqlPythonInvokePackage.zip",
    @"/usqlext/assembly/python/UsqlPythonDeployPackage.zip",
    @"/usqlext/assembly/python/version.python"
);
```

```
CREATE DATABASE IF NOT EXISTS master;
USE DATABASE master;

DROP ASSEMBLY IF EXISTS ExtR;
CREATE ASSEMBLY IF NOT EXISTS ExtR
FROM @"/usqlext/assembly/R/ExtR.dll"
WITH ADDITIONAL_FILES = (
    @"/usqlext/assembly/R/ExtR.pdb",
    @"/usqlext/assembly/R/DynamicInterop.dll",
    @"/usqlext/assembly/R/RDotNet.dll",
```

FP Data Solutions

U-SQL Cognitive Extentions

- Built-in Cognitive capabilities
  - Imaging: Detect faces
  - Imaging: Detect emotion
  - Imaging: Detect objects (tagging)
  - Imaging: OCR
  - Text: Key Phrase Extraction
  - Text: Sentiment Analysis

FP Data Solutions

U-SQL R and Python Extentions

- BASED ON REDUCER(s)

```
@PyOutput =
    REDUCE @Extended
    ON Par
    PRODUCE Par int,
            SepalLength double,
            SepalWidth double,
            PetalLength double,
            PetalWidth double,
            Species string,
            SepalRatio double,
            PetalRatio double
    USING new Extension.Python.Reducer(pyScript : @myPyScript);

@ROutput =
    REDUCE @PyOutput
    ON Par
    PRODUCE Par,
            RowId int,
            ROutput string
    READONLY Par
    USING new Extension.R.Reducer(command : @myRScript,
                                  rReturnType:"charactermatrix",
                                     stringsAsFactors:true);
```

Python Reducer

Python Reducer

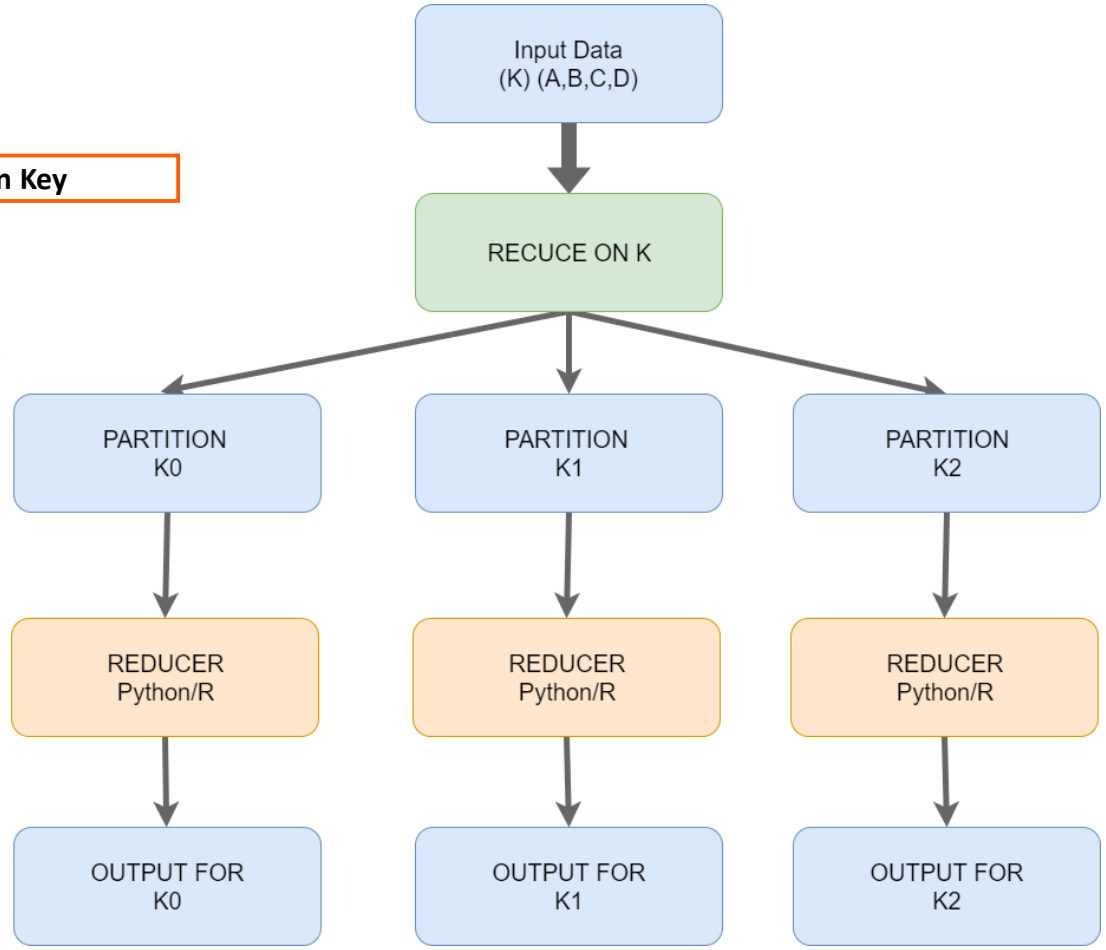
Azure Data Lake

FP Data Solutions

U-SQL REDUCER

```
@ROutput =
    REDUCE @PyOutput
    ON Par
    PRODUCE Par,
            RowId int,
            ROutput string
    READONLY Par
    USING new Extension.R.Reducer(command : @myRScript);
```

Partition Key



DEMO

- Samples\Array
- Samples\Map
- Demo003Stackoverflow
- Demo004Cognitive
- Demo005PythonDeploy
- Demo006GetADLUInfo
- Task5
- Task6
- Task7

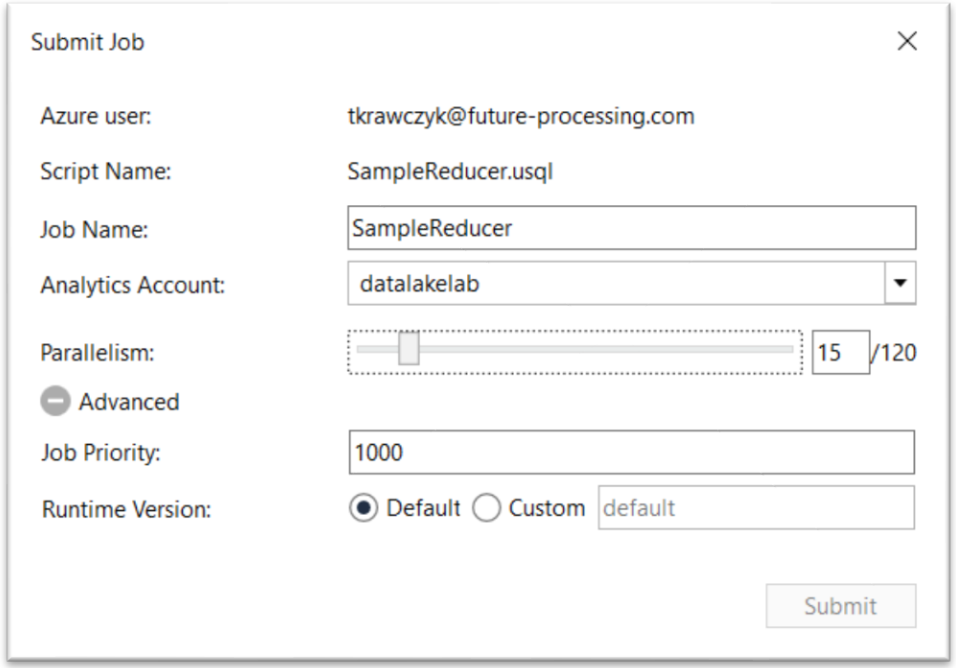
FP Data Solutions

Data Lake Analytics Runtime

1 ADLAU ~= A VM with 2 cores and 6 GB of memory

- Limited Network Access *

Parallelism N = N ADLAUs



FP Data Solutions

Data Lake Analytics Runtime

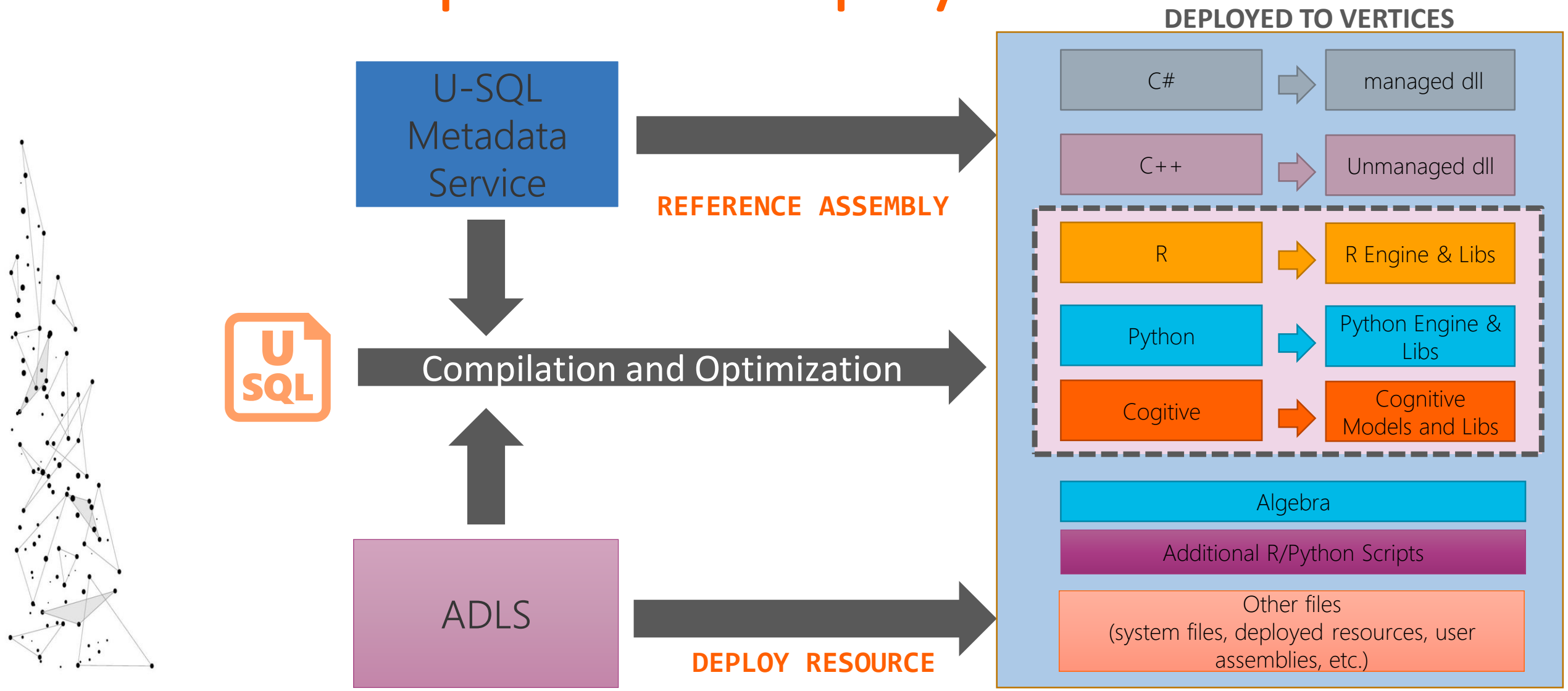
ADLUA (VERTEX) = Virtual Machine

```
public static string GetVMInfo()
        {
            var sb = new StringBuilder();
            var myManagementClass = new
                ManagementClass("Win32_PerfRawData_Counters_HyperVDynamicMemoryIntegrationService");
            var myManagementCollection =
                myManagementClass.GetInstances();
            var myProperties =
                myManagementClass.Properties;
            var myPropertyResults =
                new Dictionary<string, object>();

            foreach (var obj in myManagementCollection)
            {
                foreach (var myProperty in myProperties)
                {
                    myPropertyResults.Add(myProperty.Name,
                        obj.Properties[myProperty.Name].Value);
                }
            }

            foreach (var myPropertyResult in myPropertyResults)
            {
                var item = $"{myPropertyResult.Key}:{myPropertyResult.Value}";
                sb.AppendLine(item);
            }
            return sb.ToString();
        }
```

```
Processor:Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz;
Microsoft Windows Server 2012 R2 Datacenter
OSArchitecture:64-bit
SystemDevice:\Device\HarddiskVolume1
SystemDirectory:C:\Windows\system32
SystemDrive:C:
TotalSwapSpaceSize:
TotalVirtualMemorySize:8314420
TotalVisibleMemorySize:2096692
```

FP Data Solutions

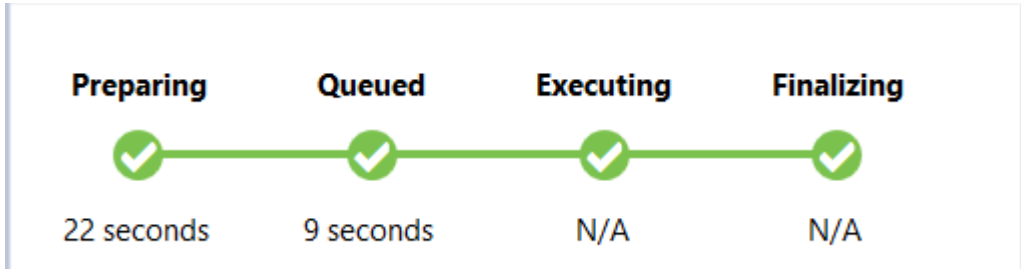
ADLA Compilation & Deployment



FP Data Solutions

ADLA Job Lifecycle States

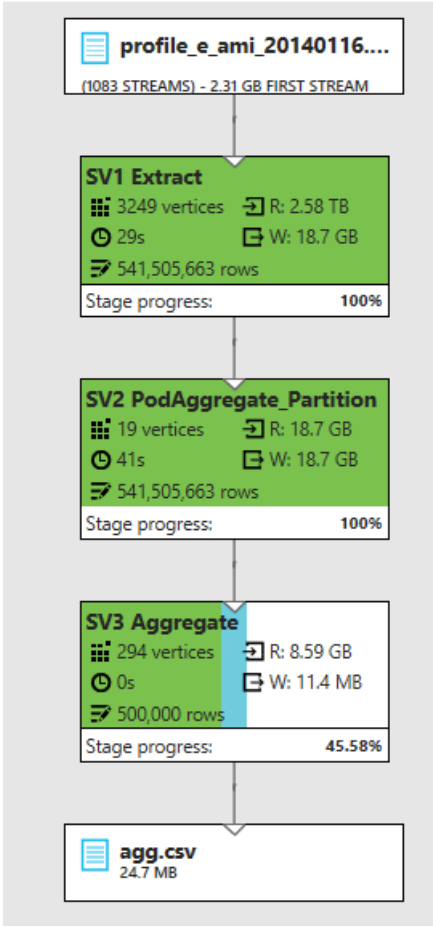
```
@output =
    SELECT
        MAX(Duration) AS DurationMax,
        MIN(Duration) AS DurationMin,
        AVG(Duration) AS DurationAvg,
        SUM(Duration) AS DurationSum,
        VAR(Duration) AS
DurationVarianve,
        STDEV(Duration) AS DurationStDev,
    FROM @searchlog
    GROUP BY Region
    HAVING DurationMin > 1;
```

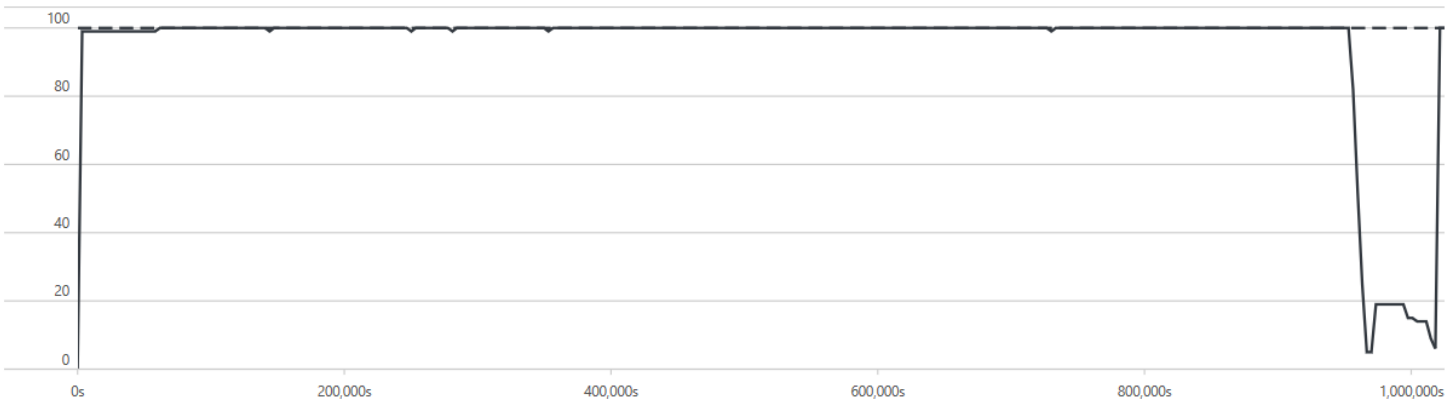


Result = Succeeded | Failed | Cancelled

FP Data Solutions

Azure Data Lake Analytics Optimization





	Actual	Balanced	Fast	Custom
	Used / Allocated	Used / Allocated	Used / Allocated	Used / Allocated
AUs allocated	100	1105	1381	1
AU-hours	28.57	45.24	50.4	27.14
Run time	17min 8s	2min 27s	2min 11s	1d 3hr
Estimated cost	USD 42.85	USD 67.86	USD 75.61	USD 40.71
Efficiency	N/A	60%	54%	100%
		Select	Select	Select

FP Data Solutions

Azure Data Lake vs Spark

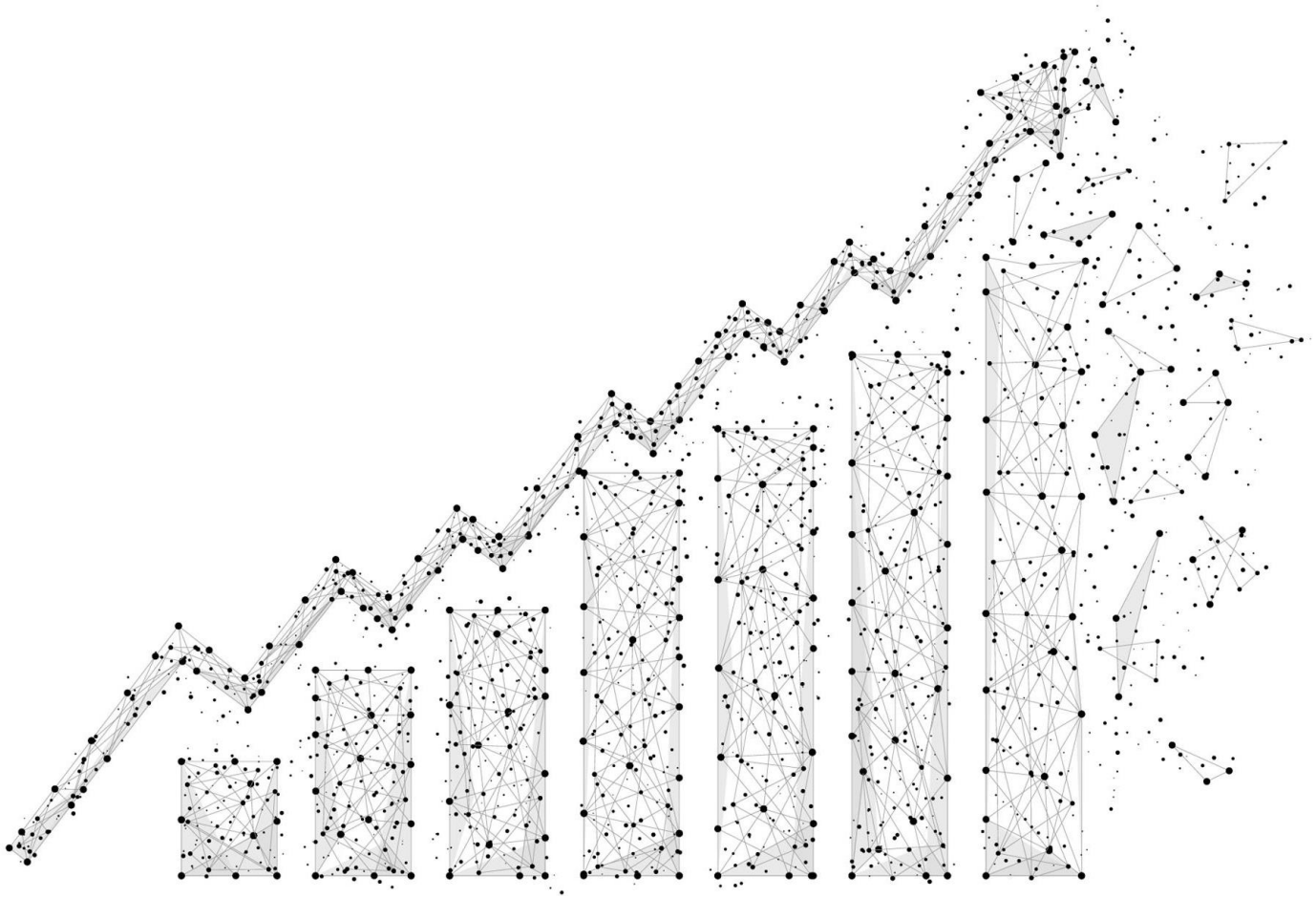
Why

Python is a king of data science

Data sources and sinks

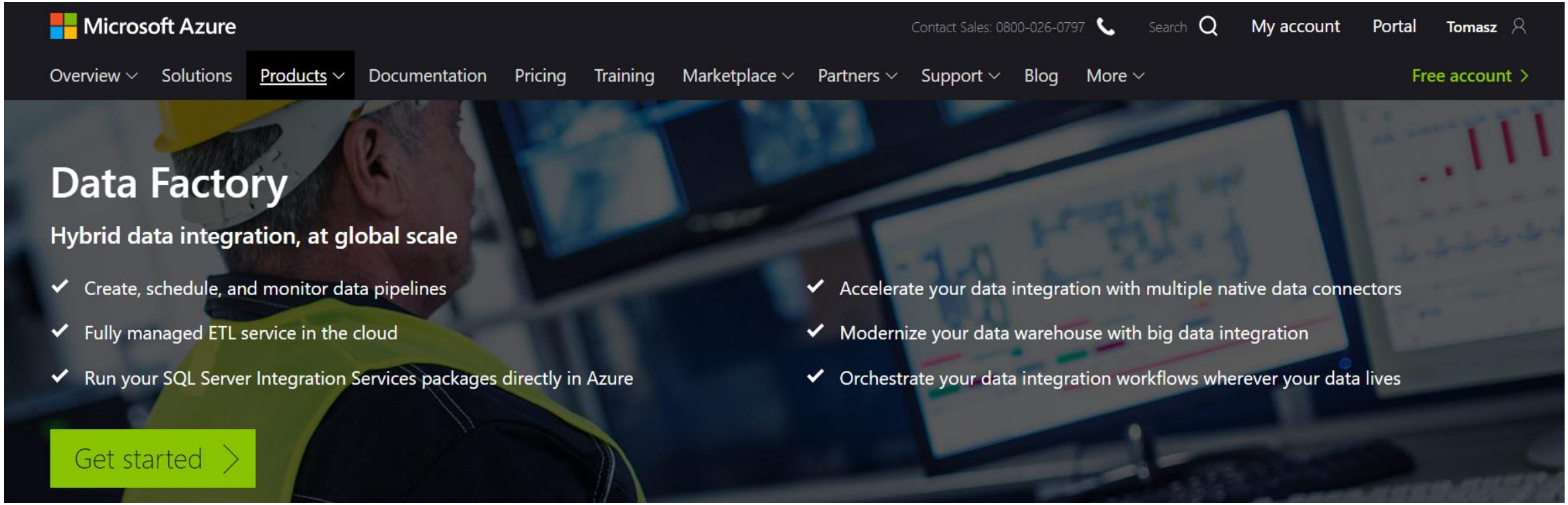
DEMO

- Demo003Stackoverflow ADLU



FP Data Solutions

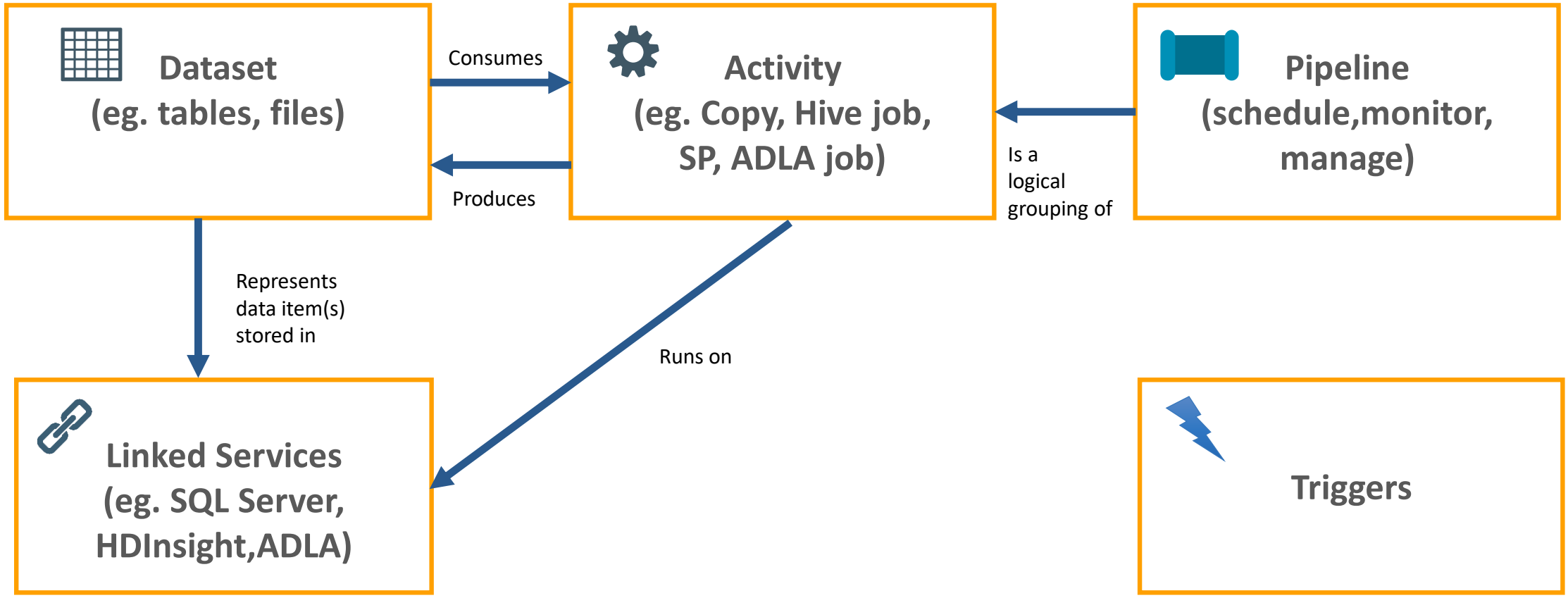
Azure Data Factory (V2)



https://azure.microsoft.com/en-us/services/data-factory/

FP Data Solutions

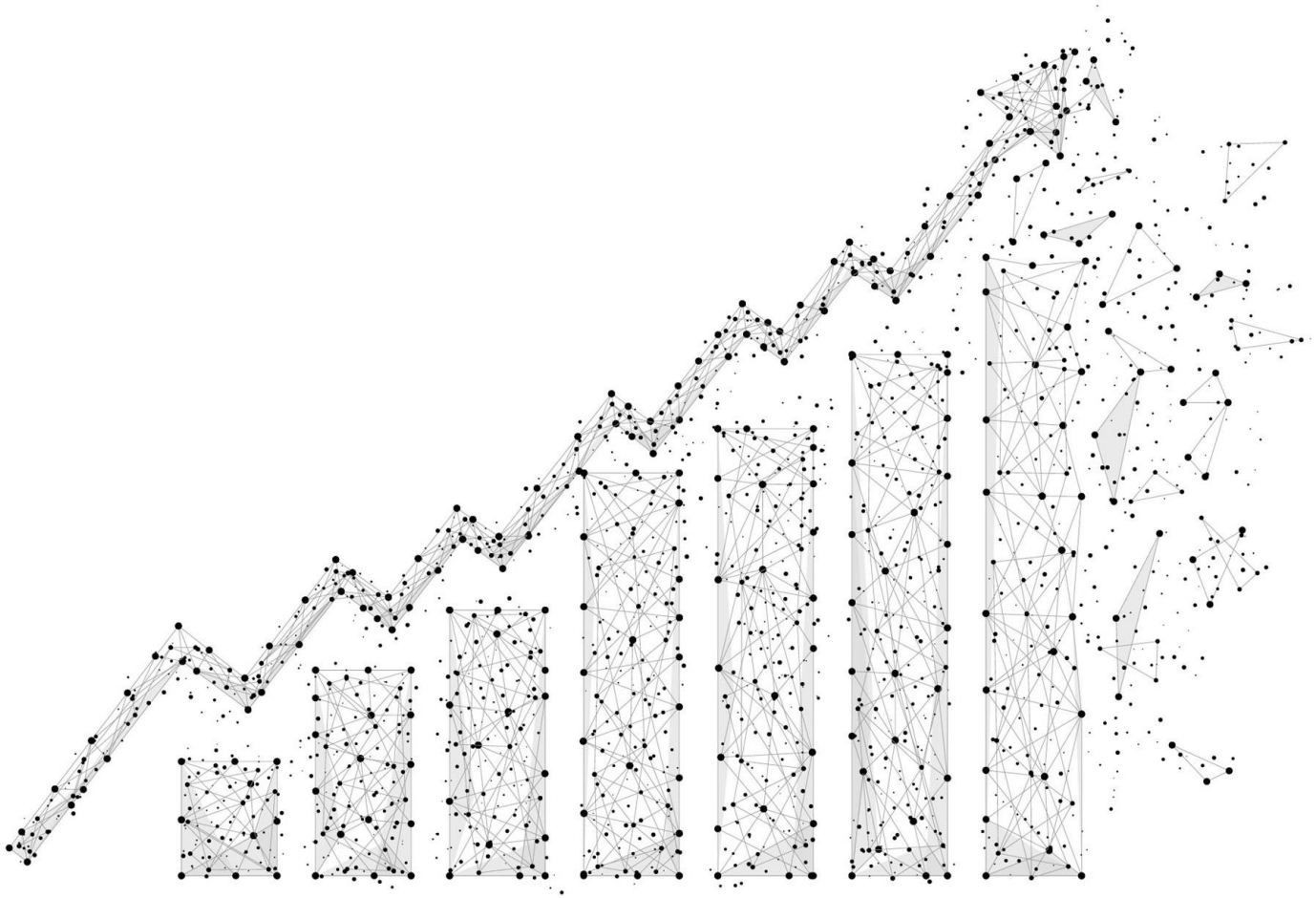
Azure Data Factory Concept



FP Data Solutions

DEMO

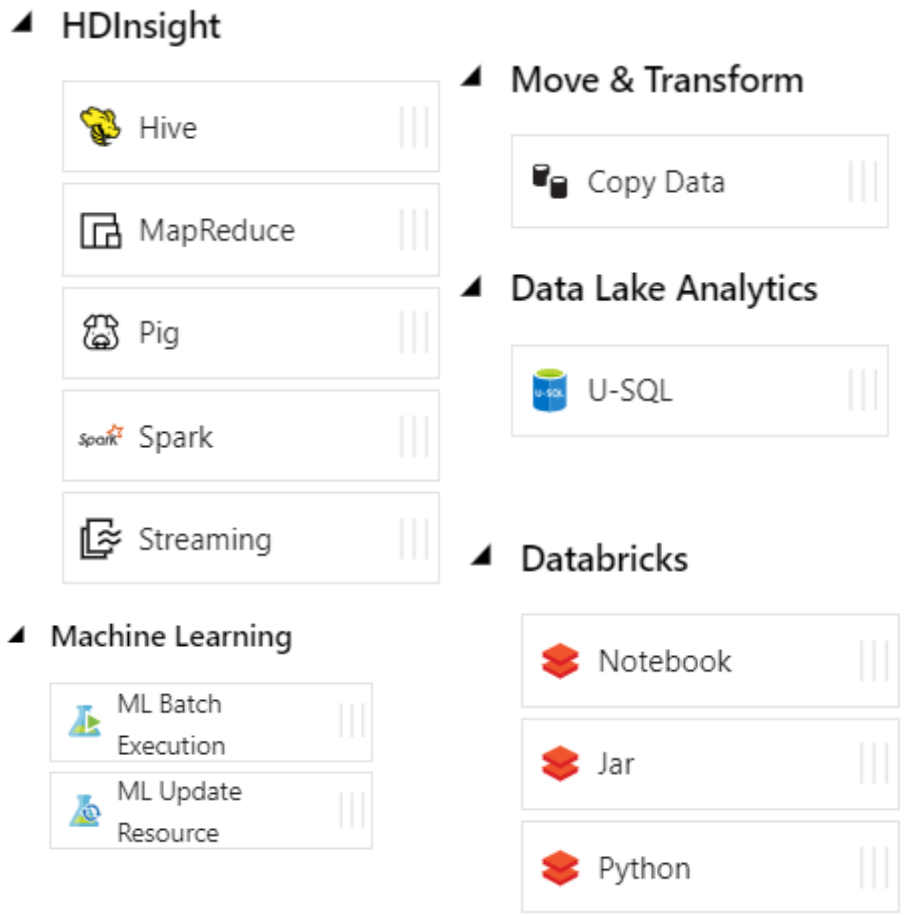
- Demo ADF
- Task8



FP Data Solutions

Azure Data Factory - Data activities

- Data movement activities : Copy Activity
- Data transformation activities :
  - HDInsight (Hive, Pig, MapReduce ,Hadoop Streaming ,Spark)
  - Azure Databricks Notebooks, Jars, Pythons
  - Machine Learning activities: Batch Execution and Update Resource (*)
  - Stored Procedure
  - Data Lake Analytics U-SQL
  - DotNet (Custom)



FP Data Solutions

Resources

My examples (and demos)

- https://github.com/FP-DataSolutions/AzureDataLake
- https://github.com/cloud4yourdata/CommunityEvents/tree/master/DataCommunity201812_AzureDataLake
- https://github.com/cloud4yourdata/CommunityEvents/tree/master/4DevKatowice2018/USQL
- https://github.com/cloud4yourdata/usql/

Blogs and pages

- http://aka.ms/AzureDataLake
- http://usql.io
- http://blogs.msdn.microsoft.com/mrys/
- http://blogs.msdn.microsoft.com/azuredatalake/

Documentation, articles ...

- http://aka.ms/usql_reference
- https://azure.microsoft.com/enus/documentation/services/data-lake-analytics/
- https://msdn.microsoft.com/en-us/magazine/mt614251
- https://channel9.msdn.com/Search?term=U-SQL#ch9Search
- https://www.youtube.com/results?search_query=U-SQL

FP Data Solutions

THANK YOU!

tkrawczyk@future-processing.com