

# Matching Engine

The Spanish Point Matching Engine identifies music that has been streamed on digital and traditional services so artists get paid.

Built on modern cloud technologies and including - innovative algorithms it matches streaming music log files at a fraction of the cost and at multiple times the performance of legacy systems. It gives you the confidence to tackle streaming data with higher accuracy and operational costs that you control.



## Digital Streaming Data - the challenge

To pay their members collecting societies must manage extreme increases in data from streaming services. Missing, inaccurate meta data or conflicting information only adds to the problem. Societies are struggling to put in place the information systems to tackle this and with payments per stream at fractions of cents they are looking for alternatives. The Spanish Point Matching Engine solves this by offering increased accuracy, very high performance and scalable usage-based costs. With over 60 configuration parameters and out of the box integration tools it can easily be included into societies existing systems.

## Modern Cloud Architecture

The Matching Engine is built on a modern cloud architecture using Microsoft Azure. Storage, application components and integration services are all built on Azure PaaS services supporting auto-scaling and resilience. With over 140 data centres worldwide the Matching Engine can easily be deployed to a data centre near you.

## Autoscaling

The Matching Engine offers powerful autoscaling options for all streaming data sources. As your data volumes change the Matching Engine can scale to meet demand and scale down once processing is complete. Autoscaling takes advantage of the elasticity of the cloud while easing management overhead. It reduces the need for an operator to continually monitor the performance of the matching process and make decisions about adding or removing resources.

It includes a "look ahead" scaling feature; monitoring data volumes and data types in the Ingestion Pipeline and scaling to meet the demand when the data is ingested into the Matching Engine. Scaling and the cost of cloud resources are placed under your control. Scaling rules can be configured by data source and be automatically invoked when the data type is detected in the Ingestion Pipeline.

## High Performance

Horizontal scaling is supported as standard. Leveraging Azure App Services and with a design that is cloud native the Matching Engine is optimised for

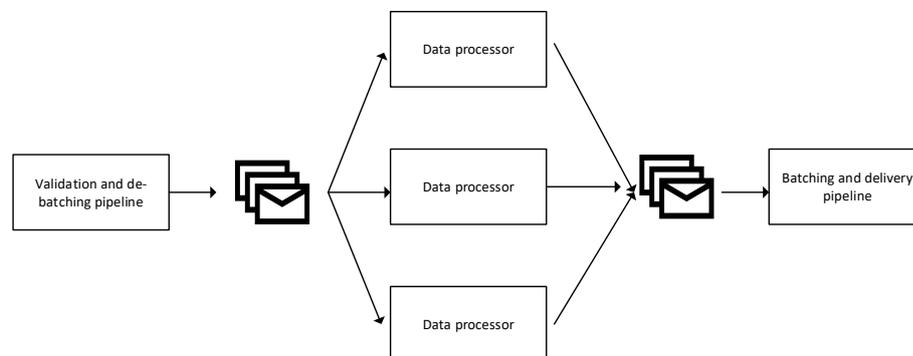
Azure Service scaling. One customer's production environment autoscales to handle more than one billion licensee specific recording identifiers linking to a repertoire of more than 15 million unique musical works processing more than 30,000 work matching requests per minute.

Testing and Microsoft's autoscaling guidance shows the Matching Engine is capable of multiple times this performance.

## Data Ingestion

To provide data flow to the Matching Engine you can use the Matching Engine's associated sophisticated Bulk Ingestion Pipeline.

This highly scalable ingestion and extraction engine manages the processing of batch-based messages and can be configured to provide batch responses. This ingestion and extraction engine is based on a batching / de-batching pattern as per the following diagram (for batch data ingestion):



The Bulk Ingestion Pipeline provides validation, de-batching and the batching delivery to the Matching Engine. It is built on Azure Data Factory a high performance scalable ETL tool.

The data processor layer is independently scalable and supports auto scaling so that when large files are being processed the solution will scale based on defined parameters to meet the load and will then scale back down to a configured minimum when there are no files being processed.

## Configuration Options

All Matching Engine configuration parameters exist in a three-level hierarchy and can be configured differently at each level. This hierarchy is then used to identify the correct configuration setting for a given source or characteristic of the metadata submitted that is to be matched. E.G. A general work title similarity % could be defined and an alternate work title similarity % that applies to all publisher sourced matching could also be

defined.

The parameters are contained in a simple JSON format file that can be edited easily by an experienced business user.

The core matching engine contains over sixty different configuration parameters. Some examples of configuration parameters that are available include:

- **WorkTitleSimilarity**

Threshold % of similarity between the search string and the returned potential work title. If the % figure returned  $\geq$  this threshold value then the title (and therefore its corresponding work) will be considered in the bucket of works to be further examined to see if they are a match. Typical value 90%.

- **TitleIgnoreWords**

The Matching engine will discard the words listed in the IgnoreWords configuration parameter for similarity text matching of titles. Word list delimited by semi-colon ";". Typical value "Remix;"

- **ContributorIgnoreWords**

The matching engine will discard the words listed in the IgnoreWords configuration parameter for similarity text matching of IP Names. Word list delimited by semi-colon ";". Typical value "Unknown Writer;Unknown"

- **ContributorSimilarity**

Threshold % of similarity between the search string and the returned potential IP Name. If the % figure returned  $\geq$  this threshold value, then it will be considered to have been matched successfully. Typical value 70%.

- **EnsureAllContributorsMatch**

Determine if all contributors must match or if all contributors with specific roles must match. "All" means all contributors regardless of role. Can also be a list of role codes with semi-colon as delimiters. Used in combination with MinMatchingContributors. May be omitted.

- **MinMatchingContributors**

The minimum number of contributors on the input work that must match with a work in the index in order for the work to be determined a match. Note: works in combination with the EnsureAllContributorsMatch parameter setting above.

- o [WorkMatchSequence](#)

Semi-colon separated list of number types and title types that are used for work level matching. E.G. "ISWC;SOC;Title" mean try to match by ISWC first, the Society work number and then all types of work title.

- o [ContributorMatchSequence](#)

Semi-colon separated list of number types and name types that are used for contributor level matching

## Multilingual / Multi-Character Set

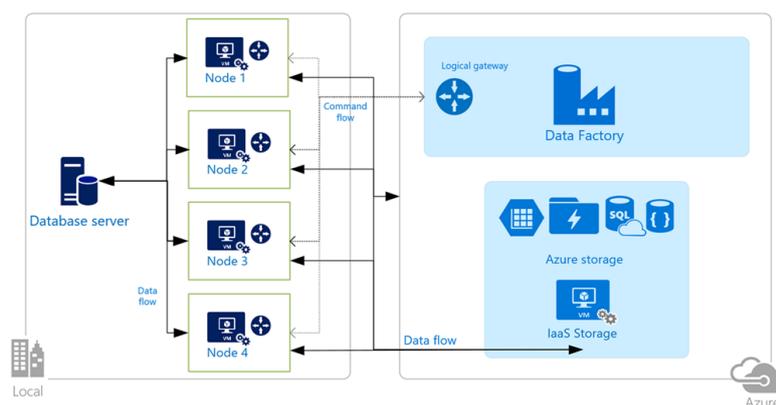
Multilingual repertoire works data bases are supported as Is multi-character sets. Configuration rules and parameters can be adjusted for your language to optimise matching for common strings.

## Input File Formats

All the common log file formats are supported e.g. DSPs such as Sony, Universal, Spotify, Apple and YouTube. Additional formats can be easily configured.

## Connection to Repertoire

Connection to on-premise and cloud repertoire and works databases is supported. Azure Data Factory is configured to manage the data extraction and loading to Azure storage for use by the Matching Engine. Azure Data Gateway Is Installed locally to your existing repertoire database, either on-premise or your cloud environment and configured to manage the extraction.



## Setup and Configuration

You have a choice of setup and configuration options. We can provide documentation and training to your IT services team or you can avail of our package service offering to quickly have the Matching Engine setup and configured to work with your data and log files.

## Fully Managed

The Matching Engine can be provided as a fully managed service with daily monitoring, regular health checks and full system administration. Alternatively, you can decide to have the Matching Engine managed by your existing IT services team. We provide a full set of configuration, system administration and run book documentation.

## Cost-Effective

The Matching Engine is designed to be cost-effective. You pay for the Matching Engine by the month and only for the cloud resources you consume.