CENTERS FOR MEDICARE & MEDICAID SERVICES

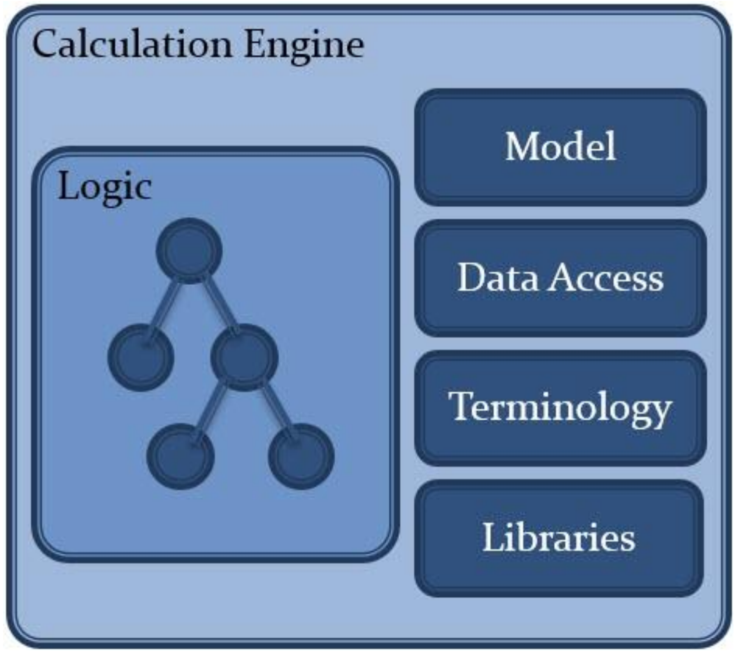
Electronic Clinical Quality Measure (eCQM) Calculation Architecture

An Overview of the Use of Clinical Quality Language (CQL) and Expression Logical Model (ELM) in an eCQM calculation environment

www.esacinc.com

General electronic Clinical Quality Measure (eCQM) Calculation Architecture

Calculation of eCQMs involves the following conceptual components:



Calculation Engine is what performs the measure calculations

Logic is the description of how the measure calculates against the clinical information, for example, patient records

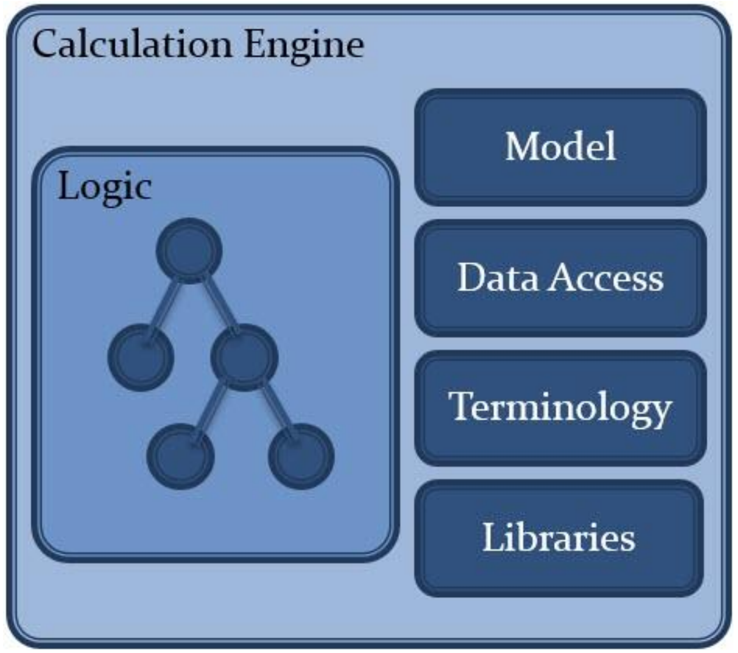
Model is the structured representation of clinical information that is used to calculate the measure

Data Access is how records of clinical information are retrieve from the underlying system, for example, an Electronic Health Record (EHR)

Terminology is concerned with determining whether clinical information is related to the measure logic through looking at coded values
Libraries enable the reuse of measure logic across measures and decision support artifacts

Database Management System (DBMS)-based eCQM Calculation Architecture

These components are present in current calculation systems, though they may be implemented differently in different environments. For example, an implementation primarily based around a DBMS such as MSSQL Server may have:



Calculation Engine in this case is the overall DBMS such as Oracle or Microsoft SQL Server

Logic defined as stored procedures in the database, typically hand-translated from the human-readable

Model is defined as tables or views in the database, typically mapped from the source EHR to HL7 V3-style structures

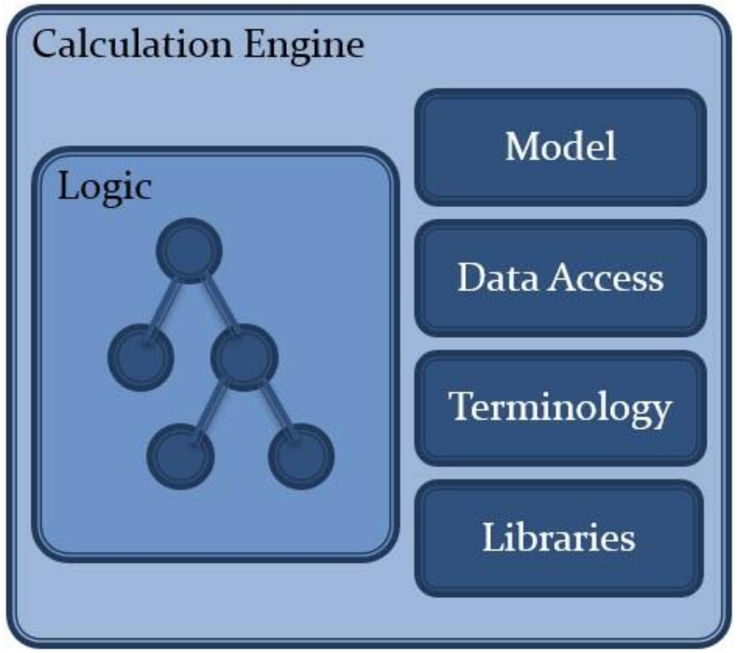
Data Access is performed by the database system via index access

Terminology is typically "cached" as tables in the database and related using filters and joins in the logic

Libraries of commonly used patterns in the measure definitions may be abstracted as additional stored procedures

Service-based eCQM Calculation Architecture

As another example, the measure calculation may be performed in a service layer in a platform such as .NET. In this case:



Calculation Engine is the middleware service layer that actually performs the calculations

Logic is represented as methods in a development language such as Java or .NET

Model is defined as .NET classes, typically derived from HL7 V3 models

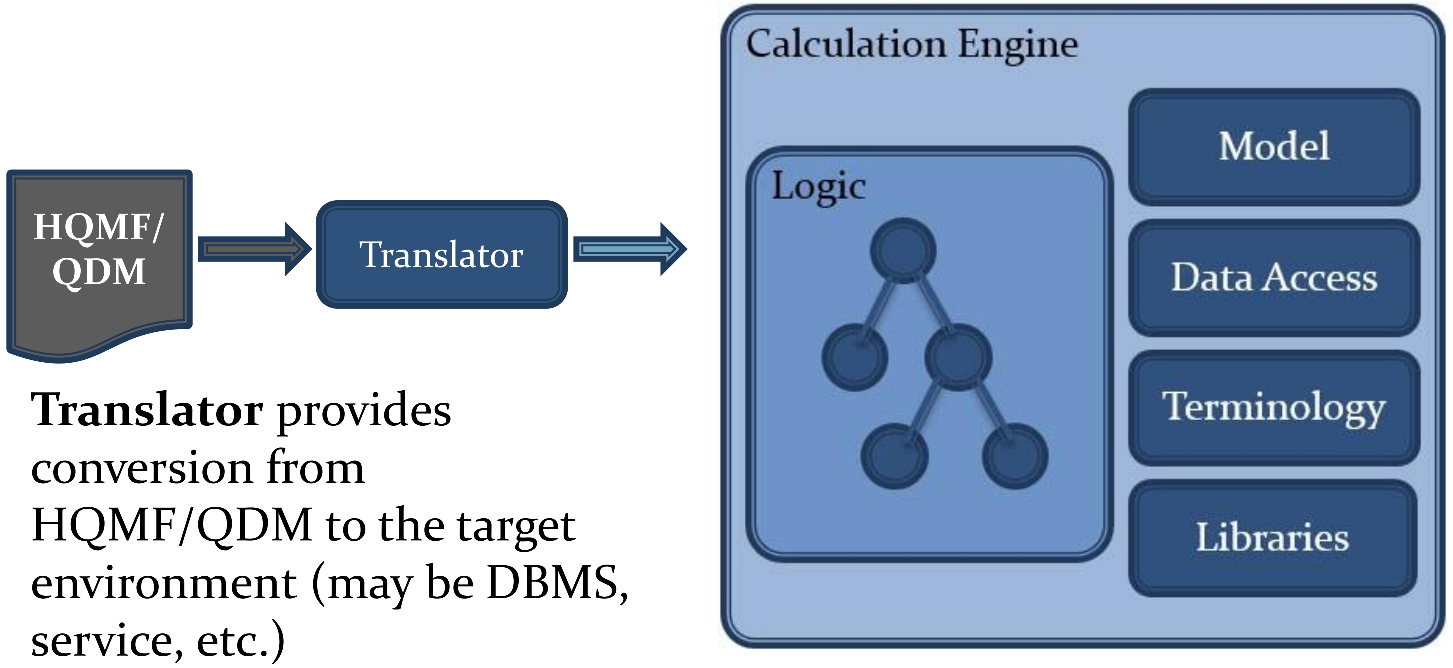
Data Access is performed by a service layer, accessing a database or HL7 V3 documents

Terminology may be provided by a full terminology service, or by caching relevant terminologies

Libraries in this case are just .NET assemblies containing commonly used calculation methods

Current Health Quality Measures Format (HQMF) eCQM Calculation Architecture

Building on this example for an HQMF/Quality Data Model (QDM) calculation environment specifically:

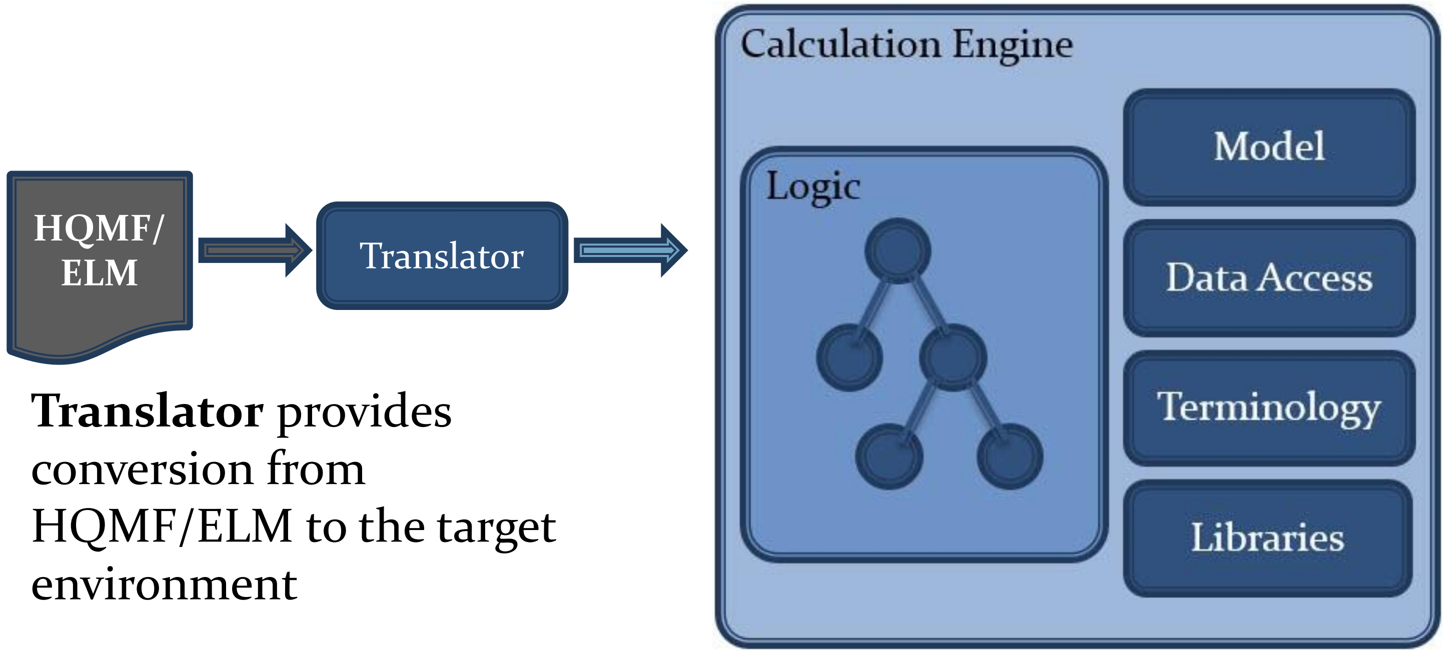


Translator provides conversion from HQMF/QDM to the target environment (may be DBMS, service, etc.)

- Translation process may be **manual** or **automatic**
- For most implementations, this is a manual, time-consuming, and error-prone process
- Implementing an **automatic** translation process is possible with HQMF/QDM, but extremely difficult

Near Term HQMF/Clinical Quality Language (CQL) eCQM Architecture

If the environment already has a translation component, the transition to CQL involves changing the translator to use Expression Logical Model (ELM), rather than HQMF/QDM as the source for the measure definitions. All other components could potentially remain the same in this environment:



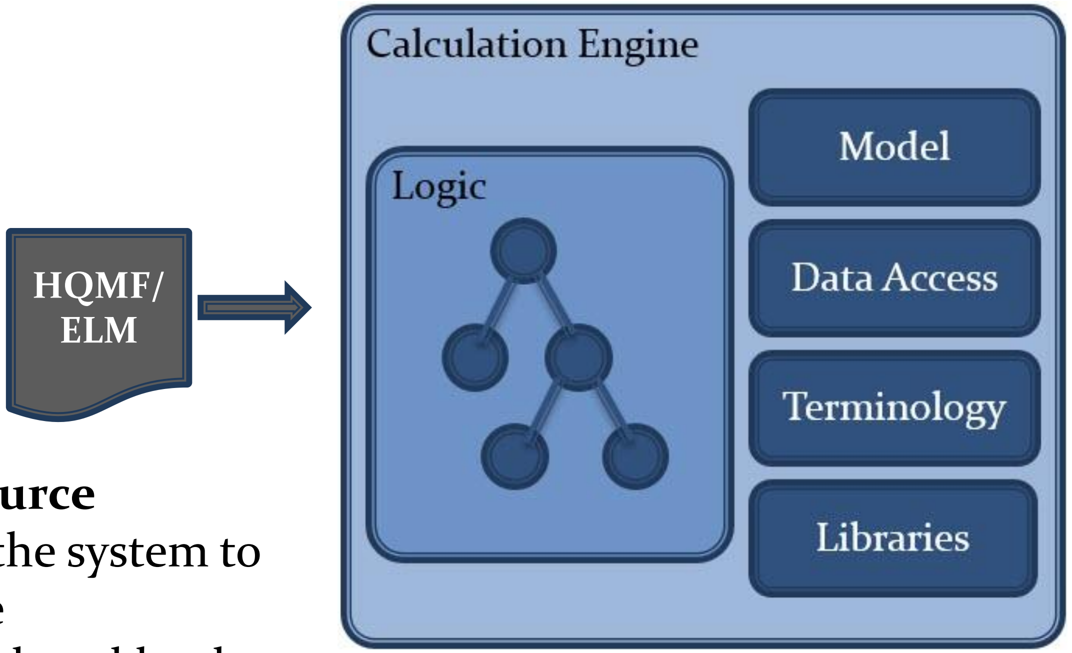
Translator provides conversion from HQMF/ELM to the target environment

- Note, however, that since the QDM logic and CQL are very different approaches, changes to the translator to use ELM may require changes to the calculation engine
- In other words, this is a potential approach, but it is a non-trivial lift

Updated 1/15/2018

Alternative HQMF/CQL eCQM Architecture

An alternative enabled by using CQL is to use a native CQL/ELM engine. In this alternative, the vendor focus would be on development of the Data Access layer component, and using an open source engine implementation:

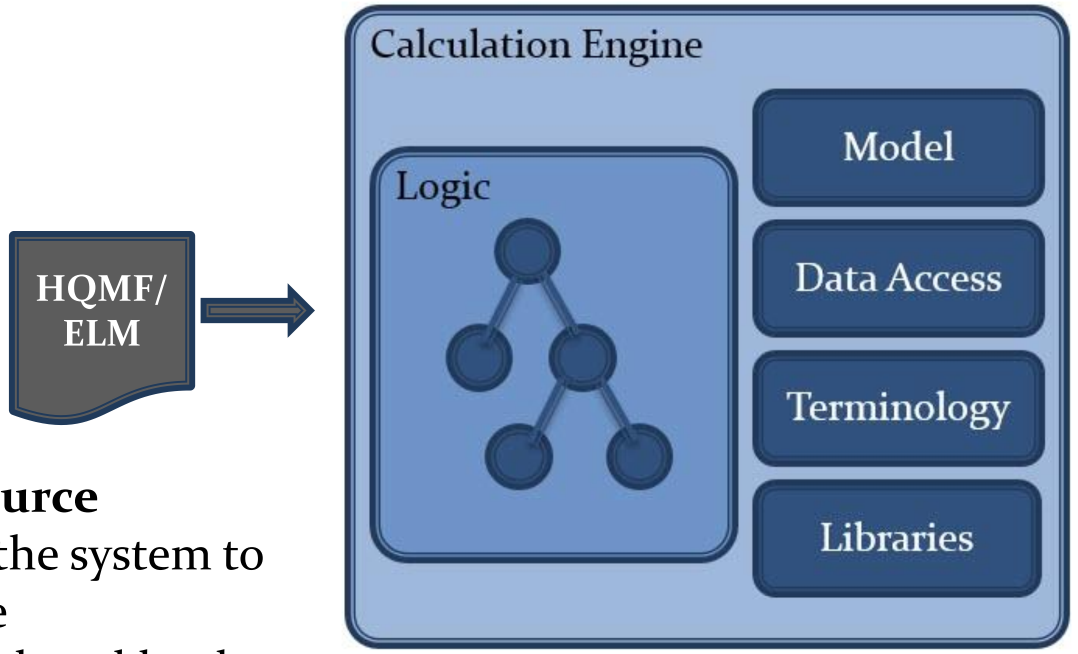


Native Open Source Engine enables the system to directly consume HQMF/ELM produced by the MAT

- The availability of open source tooling changes the focus of implementation from the engine itself to the mapping of the data source, a simpler process because the data model is still QDM

Long Term HQMF/CQL eCQM Architecture

Longer term, the goal is to enable harmonized data model standards to be used for both clinical quality measurement and decision support:



Native Open Source Engine enables the system to directly consume HQMF/ELM produced by the Measure Authoring Tool (MAT)

- Ideally, these same data model standards will be part of the interoperability capabilities of EHRs, further reducing the burden on implementers for consuming measure and decision support artifacts