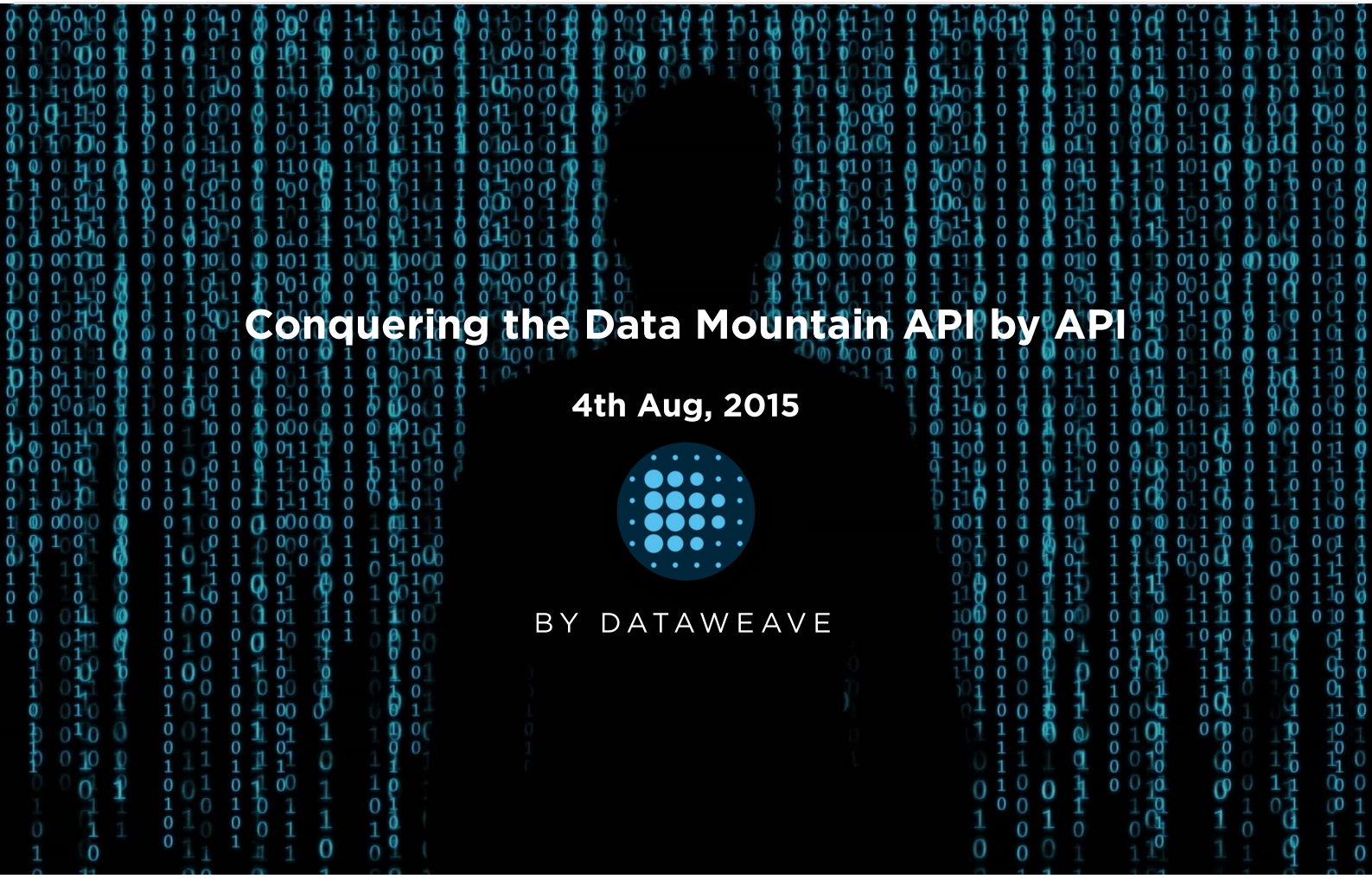

Let's revisit our raison d'être: DataWeave is a platform on which we do large-scale data aggregation and serve this data in forms that are easily consumable. The nature of the data that we deal with is that: (1) it is publicly available on the web, (2) it is factual (to the extent possible), and (3) it has high utility (decided by a number of factors that we discuss below).

The primary access channel for our data are the Data API. Other access channels such as visualizations, reports, dashboards, and alerting systems are built on top of our data APIs. Data Products such as PriceWeave, are built by combining multiple APIs and packaging them with reporting and analytics modules.

Even as the platform is capable of aggregating any kind of data on the web, we need to prioritize the data that we aggregate, and the data products that we build. There are a lot of factors that help us in deciding what kinds of data we must aggregate and the APIs we must provide on DataWeave. Some of these factors are:

- **Business Case:** A strong business use-case for the API. There has to be an inherent pain point the data set must solve. Be it the Telecom Tariffs AP or Price Intelligence API—there are a bunch of pain points they solve for distinct customer segments.
- **Scale of Impact:** There has to exist a large enough volume of potential consumers that are going through the pain points, that this data API

would solve. Consider the volume of the target consumers for the Commodity Prices API, for instance.

- **Sustained Data Need:** Data that a consumer needs frequently and/or on a long term basis has greater utility than data that is needed infrequently. We look at weather and prices all the time. Census figures, not so much.
- **Assured Data Quality:** Our consumers need to be able to trust the data we serve: data has to be complete as well as correct. Therefore, we need to ensure that there exist reliable public sources on the Web that contain the data points required to create the API.

Once these factors are accounted for, the process of creating the APIs begins. One question that we are often asked is the following: how easy/difficult is it to create data APIs? That again depends on many factors. There are many dimensions to the data we are dealing with that helps us in deciding the level of difficulty. Below we briefly touch upon some of those:

1. Structure: Textual data on the Web can be structured/semi-structured/unstructured. Extracting relevant data points from semi-structured and unstructured data without the existence of a data model can be extremely tricky. The process of recognizing the underlying pattern, automating the data extraction process, and monitoring accuracy of extracted data becomes difficult when dealing with unstructured data at scale.

2. Temporality: Data can be static or temporal in nature. Aggregating static datas sets is an one time effort. Scenarios where data changes frequently or new data points are being generated pose challenges related to scalability and data consistency. For e.g., The India Local Mandi Prices AP gets updated on a day-to-day basis with new data being added. When aggregating data that is temporal, monitoring changes to data sources and data accuracy becomes a challenge. One needs to have systems in place that ensure data is aggregated frequently and also monitored for accuracy.

3. Completeness: At one end of the spectrum we have existing data sets that are publicly downloadable. On the other end, we have data points spread across sources. In order to create data sets over these data points, these data points need to be aggregated and curated in order for them to be used. These data sources publish data in their own format, update them at different intervals. As always, “the whole is larger than the sum of its parts”; these individual data points when aggregated and presented together have many more use cases than those for the individual data points themselves.

4. Representations: Data on the Web exists in various formats including (if we are particularly unlucky!) non-standard/proprietary ones. Data exists in HTML, XML, XLS, PDFs, docs, and many more. Extracting data from these different formats and presenting them through standard representations comes with its own challenges.

5. Complexity: The data sets wherein data points are independent of each other are fairly simple to reason about. On the other hand, consider network

data sets such as social data, maps, and transportation networks. The complexity arises due to the relationships that can exist between data points within and across data sets. The extent of pre-processing required to analyse these relationships makes these data sets is huge even on a small scale.

6 .(Pre/Post) Processing: There is a lot of pre-processing involved to make raw crawled data presentable and accessible through a data API. This involves, cleaning, normalization, and representing data in standard forms. Once we have the data API, there can be a number of way that this data can be processed to create new and interesting APIs.

So, that at a high level, is the way we work at **DataWeave**. Our vision is that of curating and providing access to all of the world's public data. We are progressing towards this vision one API at a time.

Originally published at blog.dataweave.in.

- DataWeave Marketing

4th Aug, 2015

DATA ENGINEERING