

Faceted Search with Data Classification for Efficient Remote Data Access

Today's world demands ever-more efficient remote data access. Developer-implemented faceted search streamlines the ability to pinpoint relevant information across a large data repository.

To understand faceted search, start with full-text search or the ability to find words, phrases or numbers anywhere these may appear. A search engine like dtSearch can instantly search terabytes of mixed "Office" files, PDFs, emails along with nested attachments, online data and databases. For remote data access, the products can run "on premises" or on cloud platforms with no built-in limit on the number of concurrent search threads.

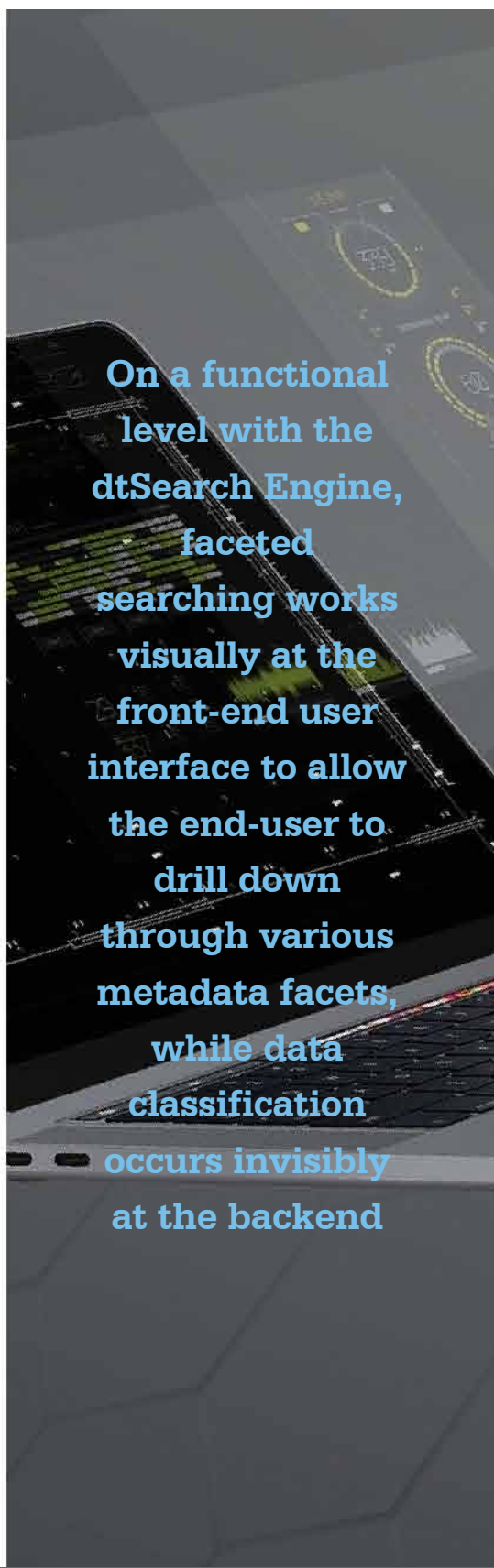
With over 25 dtSearch options for search, an end-user could enter a basic full-text structured query like gummy bears and jelly beans or a more expansive full-text query like *gummy bears or jelly beans and not (chocolate bars w/ 35 of candy wrap*)*. Or the end-user might enter an unstructured natural language search like *get me gummy bears jelly beans*. After a search, the end-user can browse through the full text of retrieved content with highlighted hits.

If *gummy bears and jelly beans* leads to just a handful of hits, browsing through them all with highlighted-hit displays would be quick. But now suppose that the dataset comes from a candy company. In that case, *gummy bears and jelly beans* might result in millions of hits, and combing through them all to get to the relevant information might take weeks.

Enter faceted search. Faceted search leverages metadata to hone in on a relevant data slice. For the candy company that might be a specific supplier name, factory location, and ship from date or date range. Faceted search can utilize such metadata – or facets – to refine a full-text search.

With faceted search, the end-user searching candy company data could click on the year 2020, then drill down to the month of March, then further drill down to a specific date range in the middle of the month. At the same time, the end-user could select another facet corresponding to a specific supplier and another facet relating to a single region of the country, a subfacet for a particular state, and eventually drill down to one factory location. Selecting those facets along with a full-text search for *gummy bears and jelly beans* can refine search results from millions of hits to just the right few.

Article contributed
by [dtSearch®](#)



On a functional level with the dtSearch Engine, faceted searching works visually at the front-end user interface to allow the end-user to drill down through various metadata facets, while data classification occurs invisibly at the backend

In this way, faceted search can vastly streamline data access. But where does metadata need to reside to enable developer integration of full-text and faceted search? Continuing with the dtSearch Engine example, the metadata can be in any of the following:

- Files like MS Office files, PDFs and emails as well as online content might contain metadata in the form of existing fields.
- A developer can add metadata “on the fly” while indexing.
- The metadata could reside in a backend database like SQL, NoSQL or SharePoint.

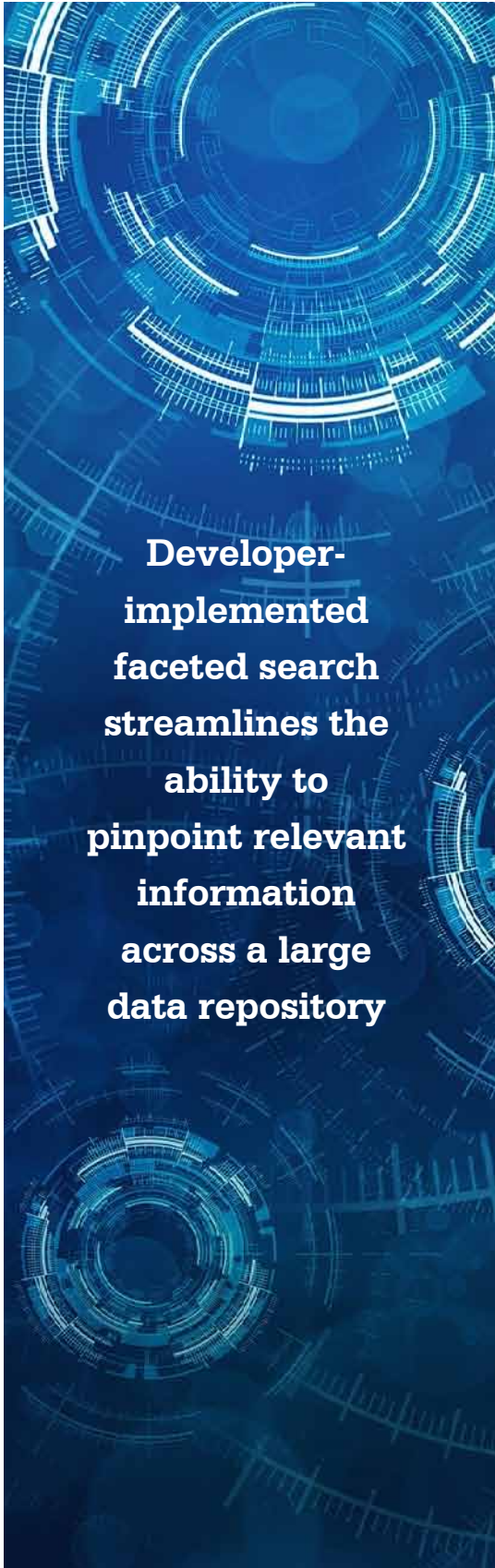
With the last database option, the dtSearch Engine could search the database and synthesize those search results with the full-text content of files stored as so-called BLOB data in the database or external files that the database references. And it doesn't have to be any single one of the above three options. The dtSearch Engine can use metadata inside a structured database, metadata added “on the fly” while indexing and metadata inside individual documents and emails, leveraging all of these for faceted search.

One other search process relevant to remote data access can use the same types of metadata as faceted search. Data classification filters search results based on security settings and other internal classification requirements. The classification can cover broad categories of end-users, or can be more granular so that each end-user only sees data relating to that specific individual's authorizations.

On a functional level with the dtSearch Engine, faceted searching works visually at the front-end user interface to allow the end-user to drill down through various metadata facets, while data classification occurs invisibly at the backend. With data classification, each end-user can still select metadata facets, enter a full-text search like *gummy bears and jelly beans*, and see highlighted hits as before. But behind the scenes, the dtSearch Engine would filter the retrieved data to correspond to the end-user's predefined scope of access.

Because dtSearch can instantly search terabytes, many dtSearch enterprise and developer customers are Fortune 100 companies and federal, state and international government agencies. But anyone with data to search can go to dtSearch.com and download a fully-functional 30-day evaluation version of dtSearch's desktop application.

Article contributed
by dtSearch®



**Developer-
implemented
faceted search
streamlines the
ability to
pinpoint relevant
information
across a large
data repository**