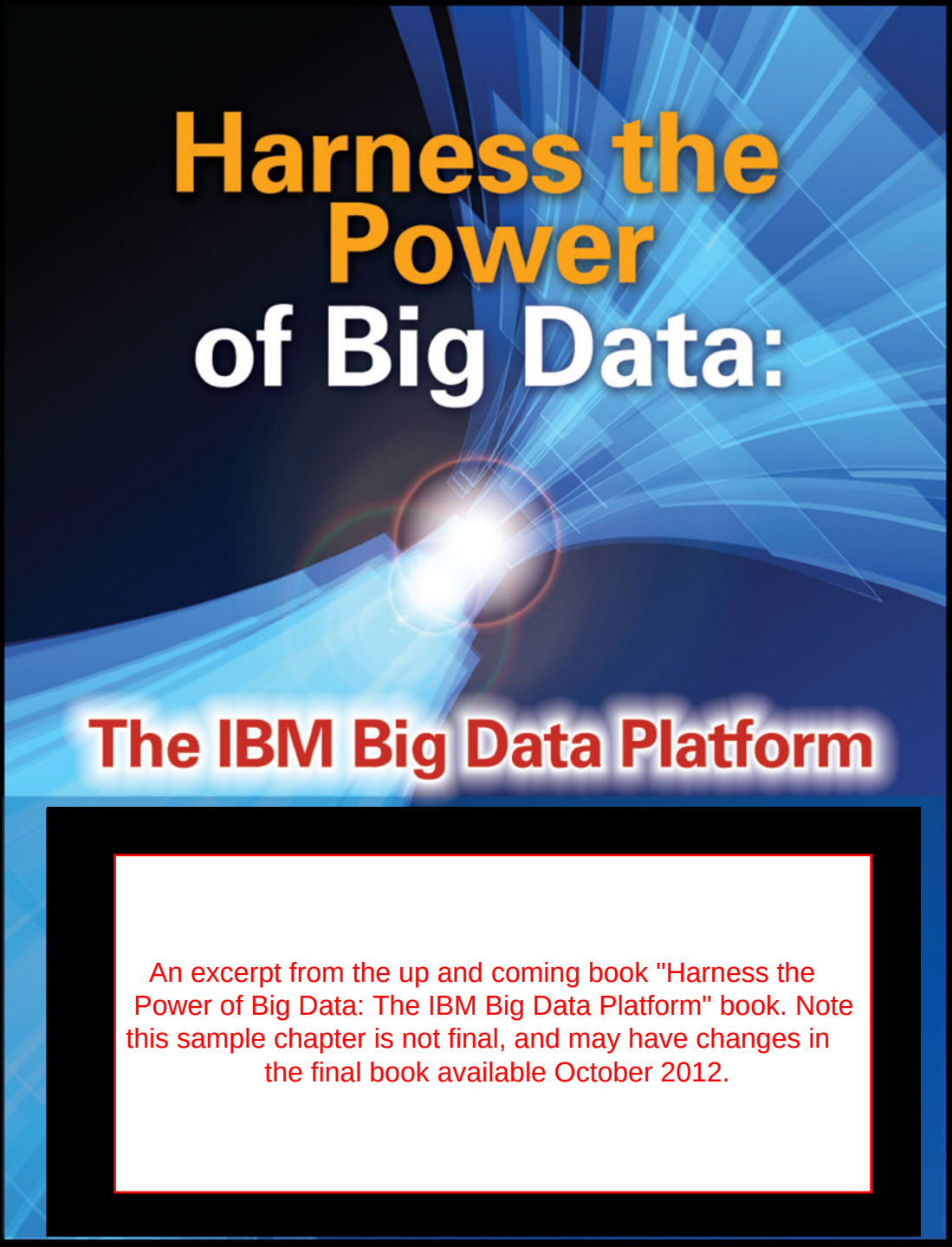

An excerpt from the up and coming book "Harness the Power of Big Data: The IBM Big Data Platform" book. Note this sample chapter is not final, and may have changes in the final book available October 2012.

PAUL ZIKOPOULOS
DIRK DEROOS
KRISHNAN PARASURAMAN
THOMAS DEUTSCH
DAVID CORRIGAN
JAMES GILES

Part IV

Unlocking Big Data

7

If Data Is the New Oil—You Need Data Exploration and Discovery

Vivisimo! Say it with us—use an Italian inflection while waving your hands for emphasis—as though you just ate the best-tasting piece of authentic pizza in your life! In April, 2012, we were saying this with gusto when IBM announced the acquisition of Vivisimo, a software company specializing in the indexing, searching, and navigation of data from multiple data sources.

One of the biggest challenges in business analytics today is that organizations store their data in distinct silos. We do this because it makes sense: for example, we keep our transactional data in online transaction processing (OLTP) databases, our email in Lotus Domino or Microsoft Exchange servers, and our call center engagement logs in customer relationship management (CRM) repositories such as SugarCRM. Each repository has specific availability requirements, security settings, service-level agreements (SLAs), and associated applications. But when it comes to building a complete view of all the relevant data for a particular customer from every data source in your organization, you're out of luck. The infrastructure that makes your silos effective tools for their designed purposes also makes them difficult to integrate. After all, you don't look at a picture by zooming in to 250 percent and examining one spot—unless, of course, you are editing it. There's huge

value in zooming out and seeing the full picture, which you can only do by pulling in data from many sources.

IBM InfoSphere Data Explorer (formerly known as Vivisimo Velocity Platform – for the remainder of this chapter, we'll call this Data Explorer for short) represents a critical component in the IBM Big Data platform. Data Explorer technology enables users to access all of the data that they need in a single *integrated* view, regardless of its format, how it's managed, or where it's stored. Being able to retrieve data from all available repositories in an organization is a key part of doing analysis involving Big Data, especially for exploratory analysis. (We talked about this in Chapter 3 – search and discovery is listed as one of IBM's five strategic ways to get started with Big Data.) Data Explorer includes a framework to easily develop business applications, called Application Builder. The customizable web-based dashboards you can build with Application Builder provide user and context-specific interfaces into the many different data sources that Data Explorer can crawl and index.

Data Explorer makes searching across your Big Data assets more *accurate*. The underlying indexes are smaller (compressed), don't need to be maintained as often as other solutions, and you can request more granular index updates instead of having to update everything. The efficient index size, coupled with the ability to dynamically expand the number of index servers makes this a highly *scalable* index and search solution. Data Explorer also includes a powerful *security* framework that enables users to only view documents that they are authorized to view based on their security profiles in the data's originating content management systems.

Data Explorer is a productivity boost for your organization at a time when it needs it the most: the dawn of the Big Data era. This technology has helped a large number of our clients unlock the value of Big Data by providing a number of techniques to locate, secure, and personalize the retrieval of business data.

Consider today's jumbo jet airplanes – typically each plane has support staff that are dedicated to it for years – like it's one of their children. In the same manner you get calls from the principal's office when your kid is sick (or in trouble), if something goes wrong with a specific airplane, a call goes out to its 'parents'. Now think about the last time you sat in a plane sitting at the gate for an extended period of time because there was a mechanical problem (something the authors of this book can relate to all too often). The airline calls that specific plane's support team. The worst thing an airline can do is keep that

plane at the gate – it amounts to thousands of dollars lost per minute. In this scenario, a call comes in from the bridge to the customer support team who has to scramble to resolve whatever the problem could be. Of course, that plane is like a customer – it has a profile, a past, and so on. Time is money – in this case, airport fees, waiting fees, customer satisfaction, and other costs mount while the clock is ticking, and it's all adding up. One large airplane manufacturer IBM worked with had information locked away in separate systems, making it nearly impossible for support teams to access all of their knowledge repositories, each with a different security schema (data was in SAP, FileNet, Content Manager, Siebel, file shares, and more). Using Data Explorer, this large airplane manufacturer was able to build a single point of access to all of their repositories with seamless and more granular security controls on the data. A common back-end infrastructure was used to service multiple front-end applications for data retrieval. The support teams got such an injection of productivity from being able to "zoom in" and "zoom out" on the scope of the problem at hand, that the number of personnel that were needed to support an aircraft decreased. This allowed the client to realize better revenue yields as new planes could be staffed with existing teams, as opposed to hiring new staff for new plane deliveries. In the end, they were able to reduce help-desk resolution latencies by 70 percent, which ultimately resulted in multimillion dollars in savings and downstream customer satisfaction, all with Data Explorer.

Indexing Data from Multiple Sources with InfoSphere Data Explorer

Data Explorer is a search platform that can index data from multiple data sources, and that provides a single search interface, giving users the ability to see all of the relevant data in their organization and beyond. Although Data Explorer handles the indexing and searching, the data itself remains in the original data sources. (This *ship function to data* paradigm is one of the principles behind Hadoop as well.) Figure 7-1 shows the architectural layout of the main components of Data Explorer.

Connector Framework

When developing your search strategy, you first determine which data sources you need to access. Data Explorer makes this easy by including a Connector Framework that supports over 30 commonly used data sources, including

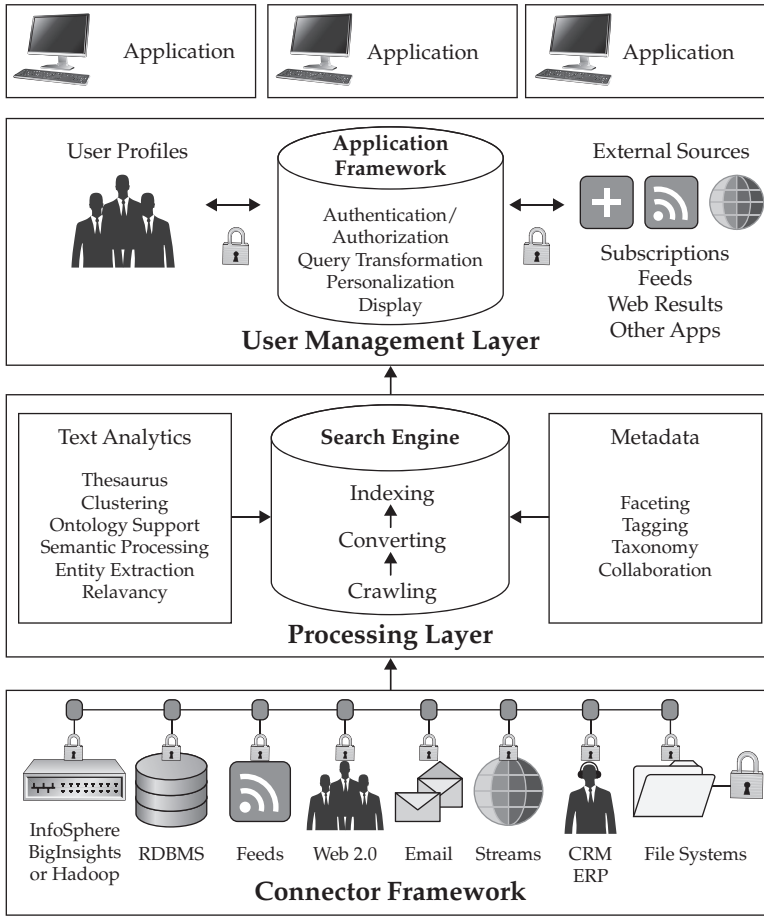

Figure 7-1 *Data Explorer architecture*

content management repositories, CRM systems, wikis, email archives, supply chain management stores, and more. There are also connectors to InfoSphere Streams and InfoSphere BigInsights, showing the deep integration of its components in the IBM Big Data platform. If you have a data source for which a connector doesn't exist, don't worry - Data Explorer also includes a mature framework for building additional connectors to proprietary data sources.

The Connector Framework taps into supported data sources to process data for indexing. We want to clearly state that Data Explorer doesn't *manage* information in the data sources; it just maintains an index of the available content for searching, navigation, and visualization.

There are many instances where people depend on data stores, such as web repositories, outside of their organization. You can use Data Explorer to add these remote sources to the unified search environment you've built for your internal sources as well. Data Explorer doesn't index these remote sites, but interfaces with the remote site's search engine to pass queries to them. It then receives result sets, interprets them, and presents them to end users alongside data from local sources.

A sophisticated security model enables Data Explorer to map the access permissions of each indexed data element according to the permissions maintained in the repository where it's managed, and to enforce these permissions when users access the data. This security model extends to the field level of individual documents, so that passages or fields within a document can be protected with their own permissions and updated without having to re-index the full document. As such, users only see data that would be visible to them if they were directly signed in to the target repository. For example, if a content management system's field-level security governs access to an Estimated Earnings report, it might grant a specific user access to the Executive Summary section, but not to the financial details such as pre-tax income (PTI), and so on. Quite simply, *if you can't see the data without Data Explorer, you won't be able to see the data with Data Explorer*.

Data Explorer connectors detect when data in the target data source is added or changed. Through these connectors, the Connector Framework ensures that the indexes reflect an up-to-date view of information in target systems.

The Data Explorer Processing Layer

The Data Explorer Processing Layer serves two purposes, each reflecting a distinct stage: indexing content as it becomes available and processing search queries from users and applications. At the beginning of this workflow, the Connector Framework makes data from each repository available to be crawled. As the data is parsed, it is transformed and processed using a number of different analytic tools, including entity extraction, tagging, and extraction of metadata for faceted navigation. Throughout this data-crunching stage, the processing layer maintains an index for the content from connected data sources. If your enterprise has existing information that describes your data sets, such as taxonomies, ontologies, and other knowledge representation standards, this information can also be factored into the index that Data Explorer builds.

Security information that is received from target data sources is ingested by the Processing Layer and also included in the indexes that Data Explorer builds for each target data source. This enables the granular role-based security capabilities that we described earlier, ensuring that users receive only the information that they are authorized to view, based on their security permissions with each target data source.

Like the other main components of the IBM Big Data platform, Data Explorer is designed to handle extremely high volumes of data by scaling out its footprint to large numbers of servers. It's been used in production settings to index trillions of records and petabytes of data.

From a high-availability perspective, the Data Explorer servers feature master-master replication, and failover capability. Whenever a server is taken offline, all search and ingestion traffic is redirected to the remaining live server. When the original server is put back online, each of its collections automatically synchronizes with a peer. If a collection has been corrupted, it is automatically restored. For planned outages, Data Explorer servers can be upgraded, replaced, or taken out of the configuration without any interruption of service (indexing or searching).

The Secret Sauce: Positional Indexes

An index is at the core of any search system and is a leading factor in query performance. In Big Data implementations, differences in index structure, size, management, and other characteristics are magnified because of the higher scale and increased data complexity. Data Explorer has a distinct advantage because it features a unique *positional index* structure that is more compact and versatile than other search solutions on the market today.

To truly appreciate why a positional index makes Data Explorer a superior enterprise search platform, you need to understand the limitations of conventional indexes, known as *vector space indexes* (see Figure 7-2).

When text is indexed using the vector space approach, all of the extracted terms are weighted according to their frequency within the document (weight is positively correlated with frequency). At query time, this weighting is also influenced by the uniqueness of the search term in relation to the full set of documents (weight is negatively correlated with the number of occurrences across the full set of documents - quite simply, if a word doesn't occur often, it's "special".) This balance between frequency and uniqueness

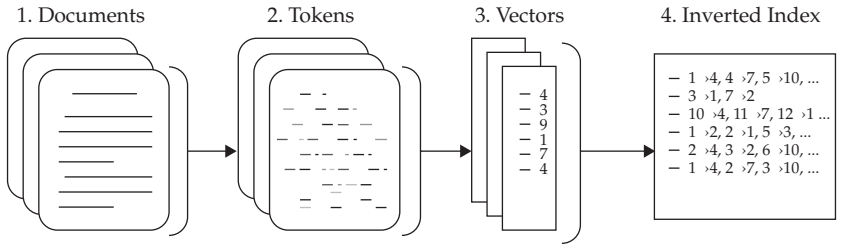

Figure 7-2 *A vector space index*

is important, so that frequently used terms like "the", which can occur often in individual documents—but also occur often in a whole set of documents—don't skew search results. For each document, all of the term weights are processed to form a vector calculation. When queries are issued against a search engine using a vector space index, a similar vector is calculated using just the terms in the search query. The documents whose vectors most closely match the search term's vectors are included in the top-ranked search results.

There are a number of limitations with the vector space approach, most of which stem from the fact that after a document has been reduced to a vector, it's impossible to reconstruct the full document flow. For example, it's impossible to consider portions of a document as separate units, and the only clues provided about such documents are the frequency and uniqueness of their indexed terms.

Big Data and modern search applications require more than just information about term frequency and uniqueness. Positioning information is required to efficiently perform phrase or proximity searches, to use proximity as a ranking factor, or to generate dynamic summaries. So, when keeping track of document positions (for example, the proximity of multiple search terms within a document), it's necessary for conventional index solutions to create a structure in addition to their vector space index—usually a document-specific positional space index. Of course, as with most things in life, nothing is free: this additional index comes at a cost as it takes more time to index documents, and the resulting index requires a significantly larger volume footprint.

As we mentioned earlier, Data Explorer also uses a positional space index—but the difference here is that there is no underlying vector space index. The positional space index is more compact than the traditional vector

space indexes, because Data Explorer uses just one efficient structure rather than two less efficient document-based structures. In a positional space index (see Figure 7-3), a document is represented as a set of tokens, each of which has a start and end position. A token can be a single word or a content range (for example, a title, or an author's name). When a user submits a query, the search terms match a passage of tokens, instead of a whole document. Data Explorer doesn't compute a vector representation, but instead keeps all positioning information directly in its index. This representation enables a complete rebuilding of the source documents, as well as the manipulation of any subparts.

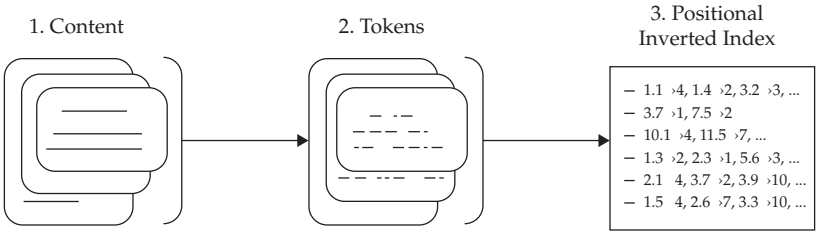

Figure 7-3 *A positional space index*

In Big Data deployments, index size can be a major concern because of the volume of the data being indexed. Many search platforms, especially those with vector space indexing schemes, produce indexes that can be 1.5 times the original data size. Data Explorer's efficient positional index structure produces a compact index, which is compressed, resulting in index sizes that are among the smallest in the industry. In addition, unlike vector space indexes, the positional space indexes don't grow when data changes; they only increase in size when new data is added.

Another benefit of positional space indexes is field-level updating, in which modifications to a single field or record in a document cause only the modified text to be re-indexed. With vector space indexes, the entire document needs to be re-indexed. This removes excessive indexing loads in systems with frequent updates, and makes small, but often important, changes available to users and applications in near-real time.

The concept of field-level security, which is related to field-level updates, is particularly useful for intelligence applications, because it enables a single classified document to contain different levels of classification. Data Explorer

can apply security to segments of text within a document, including fields. A given document, although indexed only once, can appear differently to different groups of users based on their security settings (think back to the business plan example earlier in this chapter). Such security settings are not possible with vector space indexing, because users either have access to the whole document or to none at all, limiting an organization's flexibility in sharing information across the enterprise.

Index Auditing

For Big Data applications that require detailed auditing and accounting, the ingestion of data into indexes can be fully audited through Data Explorer's audit-log function. Data Explorer generates an audit-log entry for each piece of content being sent for indexing. The log entry is guaranteed to contain all of the errors and warnings that were encountered during crawling and indexing. Using the audit log, Data Explorer deployments can ensure that 100 percent of the content is always accounted for: each item is either indexed or has triggered an error that was reported to the administrator. Such content-completeness audits are a key requirement for many legal and compliance discovery applications.

User Management Layer

The User Management Layer includes all of the resources that users and applications need to interact with data that's indexed by Data Explorer. Most importantly, this includes an interface to the Data Explorer search engine, which handles all of the data requests from connected sources. Before querying any data, users must authenticate. The user's identity and group affiliations for each repository that has been indexed are stored in user profiles or accessed at login time from a directory service. (We'll note that if you are using LDAP or Active Directory services, this information is automatically retrieved.) After they are logged in, users can have a personalized interface that reflects their profile.

This layer also contains Data Explorer's ability to federate queries to external sources that are not natively indexed by Data Explorer, such as premium subscription-based information services on the Internet. The results can be merged with native Data Explorer results to create an enriched and expanded view of relevant information.

Data Explorer gives end users the ability to comment, tag, and rate content, as well as to create shared folders for content that they want to share with other users. All of this user feedback and social content is then fed back into Data Explorer's relevance analytics to ensure that the most valuable content is presented to users. Users can also comment on their search results. The comments are field-security protected, and can be created or viewed only if the user has the proper permissions. In addition, users with the appropriate permissions can save results into folders, and those folders can be personal, shared at the group level, or shared across the enterprise. This makes for a powerful collaboration environment, where users' profiles can be returned in search results, based on their activities. Suppose a user named Anna adds comments and tags including the word "Hadoop" to various documents. Any search queries including the term "Hadoop" will then return Anna's user profile – even if there is no mention of Hadoop in her profile data and job description.

Beefing Up InfoSphere BigInsights

Because Data Explorer is now an integrated component of the IBM Big Data platform, its enterprise-scale indexing and search capabilities also apply to InfoSphere BigInsights (BigInsights). While the Hadoop technologies upon which BigInsights is built have tremendous power to run complex workloads against large volumes of structured and unstructured data, there are use cases for which Hadoop is not a practical solution. The Hadoop Distributed File System (HDFS) is designed for large-scale batch operations that run against most or all of the information in a data set. However, queries involving small subsets of data in HDFS perform poorly. By indexing content stored in HDFS, Data Explorer offers a way to address the need for rapid response times without compromising the strengths of BigInsights.

Moreover, the ability to extract, recognize, and leverage metadata can greatly enhance search precision, usability, and relevance. In fact, search can add structure to unstructured content by recognizing entities and other important terms in natural language text. Data Explorer adds semantic capabilities, such as categorization, clustering, and faceted navigation, enabling you to browse search results by topic, or to navigate to a single result without ever typing a query.

An App with a View: Creating Information Dashboards with InfoSphere Data Explorer Application Builder

In any medium-to-large-sized organization, there are multiple repositories for business data, and employees invariably spend vast amounts of time searching for information. This is especially the case for many customer-facing roles, such as sales or product support, in which people depend heavily on many systems that contain critical business information, such as CRM data, product knowledge, and market data.

Data Explorer includes a compelling tool: the Application Builder application framework. Application Builder enables you to build a multifunction dashboard for the many relevant data sources that your employees need to access. While Data Explorer provides the raw power to index many disparate data sources and efficiently handle queries, Application Builder is a front-end interface for presenting all this information. Figure 7-4 shows the architecture of Application Builder and Data Explorer working together. All of the powerful features of Data Explorer, such as its high-performing search and the preservation of user-specific access to data, are harnessed in Application Builder.

The options available for Application Builder applications are boundless. For example, you can federate Internet-based data sources to pull in news feeds, or stream financial data from securities markets or social media data sources like Twitter. The social capabilities from Data Explorer also work here: Users can collaborate through collective data tagging efforts, and share knowledge through comments and recommendations that are applied to data stored in sources aggregated by Application Builder.

Application Builder integrates with BigInsights and InfoSphere Streams (Streams). BigInsights and Streams can both feed data to Data Explorer, which can then syndicate it to Application Builder users. Alternatively, Application Builder can consume data directly from BigInsights or Streams. For example, a Streams feed can be surfaced as a live data source in Application Builder. In addition to hosting content that is customized for its users, Application Builder provides Data Explorer search interfaces. For example, the faceted search, clustered search, and recommendation engines can all be surfaced in Application Builder.

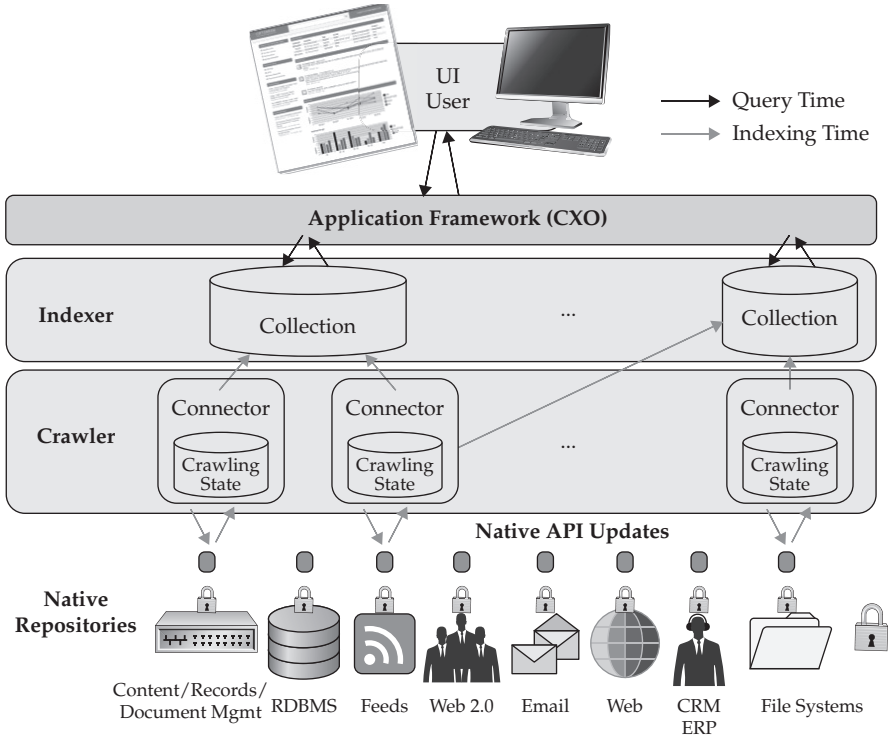

Figure 7-4 *The Application Builder architecture*

The dashboards you create with Application Builder aren't just about pulling in data from different sources. The real power comes from the definition of entity-relationship linkages between your users and the available data sets in Data Explorer. With this, your dashboards can provide information from all these data sources that are relevant to your users, without needing them to search for it. For example, your support personnel can have dashboards tailored to show data relevant to their own customers' accounts, such as their purchase history, open support tickets, and order status. This can also include data from external sources, such as news feeds relevant to the customer, or their stock price.

In short, Application Builder enables you to create custom mashups of content your employees need, where the search tools provide a mashup of content from multiple sources. The beauty of this architecture is that not only do users now have unified access to information across their business—in

many cases they don't even need to search for it, as this technology 'connects the dots' and brings relevant information to your users automatically.

Wrapping It Up: Data Explorer Unlocks Big Data

Data Explorer helps organizations unlock and optimize the true business value of all their information, regardless of application or source. It provides industry-leading Big Data indexing and search techniques and includes a rich interface that enables you to easily build and expose personalized dashboards to end-user communities. Data Explorer was designed from the inside out for the Big Data era, with its positional indexing technology, granular security lockdown, and more. And with Data Explorer's connectivity framework, data can stay in the silos where it's managed, while data scientists, researchers, and business users can focus on asking the key questions.

In the Big Data world, it's even more important than ever to have powerful, accurate, agile, and flexible search across all of your data assets, because that data is in new shapes and sizes, in larger amounts than ever before, and arriving on your doorstep faster than ever. In our experiences, many large organizations are guilty of not knowing what they could already know – they have mountains of data assets, but they aren't holistically linked to users – leaving them to scramble and 'get lucky' when searching. Data Explorer is an inflection point – it's great news for any organization needing to unlock information trapped in silos. You'll want to be saying it too! Vivisimo!