Designing an Effective Enterprise Search Solution

Executive Summary

There are many diverse requirements for search capabilities that emerge within an enterprise. This white paper addresses the top five most desired enterprise search requirements. The solution discussed herein is based on various implementation experiences we have gained over years of using multiple search tools. The top five search requirements include:

- **Diverse Content:** Ability to crawl, index and search diverse content repository.
  - The Web, Microsoft SQL database and SharePoint content management systems.
- **Secured Search:** Ability to crawl secured content and make it accessible to only authorized people and/or groups.
  - Single sign-on, forms-based authentication.
- **User Interface:** Ability to provide various user interface (UI) components to serve end users with precise results.
  - Guided navigation, related search terms, related articles and best bets.
  - AutoSuggest with terms combined from real-time search and custom (user configurable) terms in data stores.
- **Desktop Search:** Ability to integrate with content stored in the desktop.
- **Social Search:** Ability to find other people, ratings and expertise within the organization.

Defining the Enterprise Search Engine

There are quite a few proprietary and open source enterprise search tools available in the market. The Google Search Appliance is chosen here for its ease of use and its ability to handle most of the aforementioned requirements. (Refer to the Appendix for the architecture diagram that provides an alternate approach using Apache Solr 3.1 and Nutch 1.3.)

The Google Search Appliance provides quite a few traditionally requested enterprise search features out-of-the-box (OOTB). Even though the appliance fits the hardware plug-and-play model, it provides a flexible framework for integrating with external systems such as content management systems, document management systems, security systems, federated search and both Google and non-Google services on the cloud. The following lists the features required to implement the top five requirements. For the sake of gauging the complexity of the implementation, our list differentiates custom components from Google out-of-the-box (GOOTB) capabilities. A sample approach is also provided for developing the required custom components to architect an effective enterprise search solution using the Google Appliance.

- Google Web crawler for crawling and indexing Web content (GOOTB).
- Google DB connector for crawling and indexing Microsoft SQL database (GOOTB).

Cognizant

- Google SharePoint connector for crawling and indexing SharePoint content (GOOTB).

1. Google forms authentication for index time authorization and serve time authentication (GOOTB).
2. Google front-end configuration for:
   - Faceted search, aka guided navigation (limited OOTB).
   - Related search terms (GOOTB).
   - Related articles (GOOTB).
   - Best bets (GOOTB).
   - AutoSuggest (GOOTB and custom application).
3. Google desktop search component integration (external Google component).
4. Google results integration with internal rating system.
   - Integration with Google people search (GOOTB).
   - Integration with expertise system (custom).
   - Integration with custom rating system (custom).

Component Description

The following components are critical to Google-powered enterprise search architecture (see Figure 1 for a schematic view).

- **Google Search Appliance:** Google search appliance (GSA) is at the center of this proposed search architecture. GSA, an enterprise search appliance, is a packaged hardware search engine. The plug-and-play model provides many necessary enterprise search features out of the box. The tool also provides a flexible (connector) framework for which developers/implementers can integrate the appliance with other content sources.

  Just like Google.com, the Google search appliance is extremely adept in crawling, indexing and serving Web pages. Additionally, GSA provides connectors to index and search any relational databases, content management systems (i.e., EMC Documentum, Microsoft SharePoint, Open Text Livelink, IBM FileNet) and local/network file systems or file shares. Connectors to other non-supported CMS (content management system) tools can either be developed from scratch using Google's connector framework or can be purchased as a product from Google's partner Websites.
- **Google Web Crawler:** Intranet Web can be indexed using Google's Web crawler. GSA allows implementers to configure the "start URL," "follow URL pattern" and "do not crawl pattern" details, with additional options that allow us to force a recrawl of specific patterns as and when required. The crawl status alone can be tracked using the "crawler diagnostics" which details if a URL was successfully

Enterprise Search Component Diagram (using GSA)

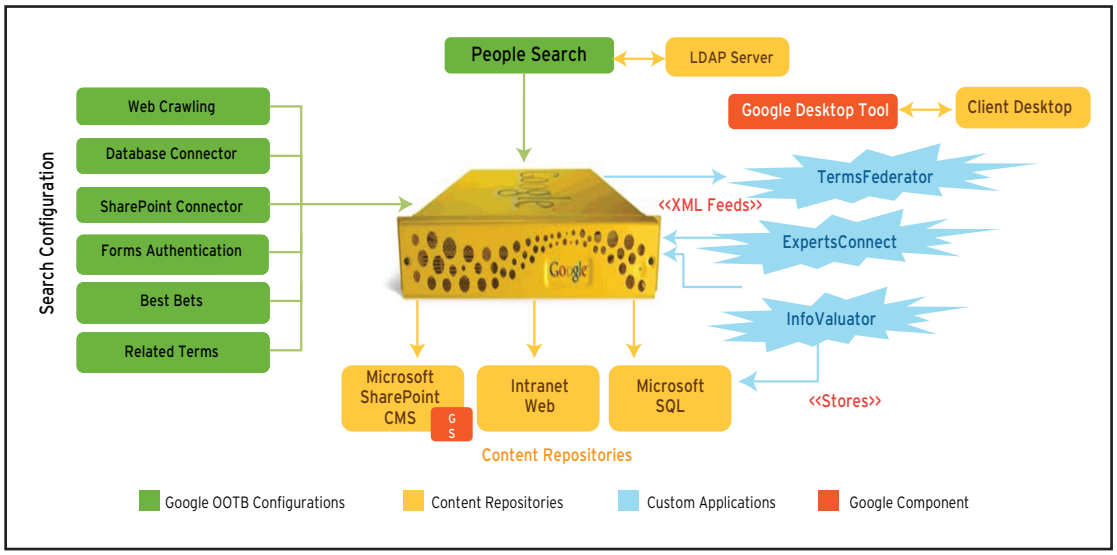

Figure 1

indexed or not. While crawling, if the Googlebot encounters an error or if the pages are not crawler friendly, appropriate error messages are displayed in the diagnostics that aid in troubleshooting.

- **Disadvantage:** As efficient and good as it sounds, one disadvantage of Web crawler is Google's inability to reveal the exact page that is currently being processed.
- **Alternative:** The OS console monitor and/or tracking log files are some ways that could help track URL crawl status. At any point of time, a developer should be able to view the current URL being crawled and issues faced (if any) with security. Almost all tools provide this feature – such as Solr, FAST, Endeca and Autonomy.

- **Database Connector:** DB crawling is again made trivial in Google by its ability to provide the implementer just the bare minimal details to fill in before triggering the database indexing. The key configurations needed are username, password, JDBC URL, database name, results URL hostname, JDBC driver class name, SQL query and primary key field. Stylesheets can be used to allow users to preview a document in a specific format in the search results page.
  - **Disadvantage:** A few key disadvantages faced during the implementation are:
    - Google's inability to allow end implementers to schedule DB crawl.
    - Google's way of removing content from index is quite primitive and time-consuming. The only way is to include the URL in the do not crawl (DNC) list and wait for the appliance to realize it. It is even more complicated for connector/XML-fed content.
    - Poor diagnostics for connector/XML-fed content.
  - **Alternative:** Compared to GSA, we found Apache Solr is a better option for indexing the database via data import handler.
    - 2000 documents in Solr would take about 5-10 seconds.
    - The Solr console provides a very good overview of documents crawled and status of crawl. Log files can be configured to capture the crawler status as well.
    - Delta query in Solr can be achieved using the regular "deltaquery" with the TIMESTAMP column. Additionally, XML import (using /update handler) can be used to import new or modified documents in the form of an XML file.
    - Solr provides an effective way to remove content from the index, either via the admin console or via XML import (/update with delete option).

- **SharePoint Connector:** With its introduction of SP2010, Microsoft provides an easy Web-based medium for uploading, maintaining and sharing documents within the enterprise. The portal-collaboration-CMS-DMS (document management system) apparatus provides tight integration with Integrated Windows Authentication (IWA) for securing sites and/or pages and/or documents.

  Google provides connectors to very few CMS systems out of the box. But one such system is Microsoft's SharePoint – which makes sense since SharePoint is gaining momentum within the enterprise due to its ease of use and cost benefits. For seamless integration with SharePoint, GSA uses the content feed which handles authorization along with the connector configuration. For a smooth handshake between the Google connector and SharePoint sites, an additional Google services component must be installed on the SharePoint server. The connector supports late security binding with bulk authentication at query time.

  Google Services (GS) is a component that must be installed at the SharePoint server end to ensure site/sub-site indexing and bulk authorization at index and query time. For more information, the step-by-step approach is easily available on the Google open source site.
  - **Disadvantage:** Even if Google is executing a bulk late binding, performance issues at query time are inevitable when the document volume is high.
  - **Alternative:** One alternate is to consider the site/page/document level security as an additional metadata, develop an application that would post-filter the results based on end-user security attributes. This is again a primitive method and has its own disadvantages in terms of query time latency.

» There are quite a few consumer off the shelf (COTS) products like Vivisimo, Microsoft FS4SP (Fast Search for SharePoint), FAST ESP – now fast search for Internet search (FSIS) – that are designed to better handle SharePoint SP2010 for a higher cost. We have found FAST ESP to be an effective tool that handles diverse CMS including SharePoint and offers early binding security model for better performance. Now, the tightly integrated SP2010 search – FS4SP – seems to be preferred by many enterprise intranets. SP2010 has its own cost benefits if the enterprise's primary technology stack includes Windows and other Microsoft product suites.

- **Forms Authentication:** Google provides a simple way to define forms authentication at both index time and query time. Forms authentication at index time allows the GoogleBot to register the security cookie based on the domain of the secured website that requires indexing. This cookie is carried with the GoogleBot, allowing the bot to crawl and index secured content.

  At query time, Google uses the query time configuration to make an HEAD request that would allow the logged-in user (within a specific domain) to view only the content that he is authorized to view. This late binding security model has its disadvantages in terms of query time latency and performance issues at the hosting server end when too many HEAD requests are made to the web server. Performance degradation is inevitable with higher QPS and/or higher results count.

  - **Alternative:** There are tools that support an early binding security model that allows the search engine to cache the user security groups along with the content. The disadvantage of early binding is that real-time security changes are not immediately reflected in the system.
  - **Note:** One disadvantage with Apache Solr is that it does not handle secured content. The only way to serve secured content is to store the security tags/groups as one of the metadata and implement a field (or metadata) constrained search.

- **AutoSuggest:** GSA provides an open source component called "search-as-you-type" which allows end implementers to fetch real-time results from the appliance (see Figure 2). In order to integrate the results from Google with that of custom (suggestive) terms, developers need to build a special component.

  TermFederator is a custom component that fetches user configurable (suggestive) terms stored in the database and/or any other CMS. (Note: TermFederator is only a consumer of the terms stored in the database and any governance for these terms is outside the scope of this architecture.) Adequate caching can be used at this component-end to improve retrieval performance of the database/CMS query. The TermFederator is configured in GSA as a "Onebox" module. At the time of end-user query, real-time results from Google and results from the TermFederator are merged into a single unified suggestion terms list. The key advantage is that GSA handles on-the-fly federation between real-time results and custom terms with minimal overhead at query time.

How AutoSuggest Works

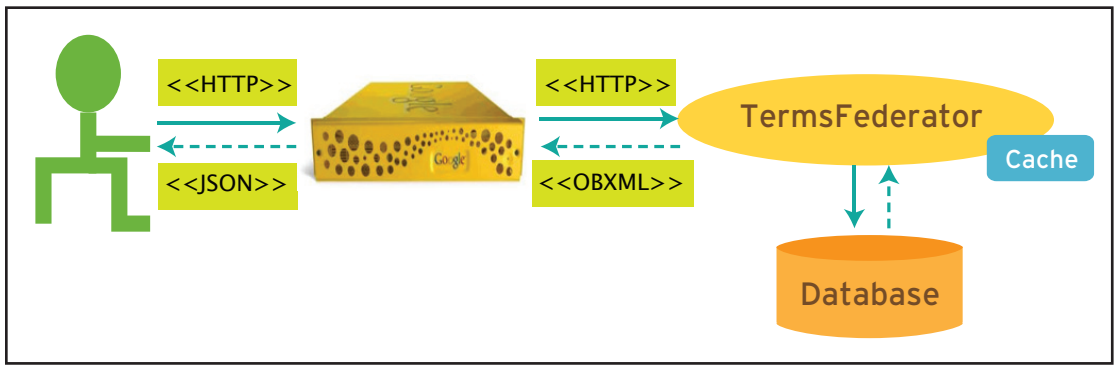

Figure 2

- **Disadvantage:** Onebox modules are designed to respond within one second. This could result in no results from TermFederator if there is any delay at the database.
- **Alternative:** "TermComponent" in Apache Solr is an effective autosuggest tool. Terms stored in any local text file can be made available to Solr at startup. A separate component designed to merge alphabetically (or sequentially) the top N terms from Solr and from the local text file would address this requirement as well.

User Interface:

- Best Bets – aka Keymatches, aka AdWords.
- Related search terms same as synonyms.
- Query expansion, same as dictionary-based search, allows users to do bidirectional search based on terms maintained as a part of this configuration.
- Faceted search, aka Guided Navigation: GSA does not support faceted search. But this feature can be achieved via metadata constrained search at query time, similar to how it is implemented in Solr. The structure of content within the appliance is flat and there is no hierarchy and/or taxonomy maintained.
  - **Disadvantage:** Facet count in GSA is not available OOTB.
  - **Alternative:** As a key requirement in most ES implementations, faceted search is currently GSA's primary drawback. But again, if we think of alternates, here are a few options:
    - **Continue using GSA:** We could develop an application that can return count based on certain fields (metadata) that are available. If not carefully planned, this could be another query time overhead. Additionally, GSA partners commercially sell components supporting "parametric search" that can be evaluated for our requirement.
    - **Considering other COTS/Open Source:** (Oracle) Endeca and (HP) Autonomy maintain content hierarchy for guided navigation. Indexing hierarchical content is unsupported with future Microsoft FS4SP. But faceted search will continue to be supported. Faceted search is one of Apache Solr's strongest features and is implemented within many e-commerce Websites.
- **Related Results:** Collective results from different and/or the same domain that refer to the same content is again OOTB with Google.
- **External Data Federation:** OneBox module in GSA allows us to federate search results from external data source/store. GSA internally handles uncluttered merging of the Onebox results with organic results in sequence. (Note: no relevancy or algorithm is applied for merging.) This is one powerful feature that Google provides in comparison to other tools on the market.

Among social search, expertise search with expert rating are common requirements. Enterprises are keen to find out those who specialize or have expertise in specific fields.

- **Assessing Social Search:** Social search – rather, personalized social search – has emerged as a key requirement within today's enterprises. Among social search, expertise search with expert rating are common requirements. Enterprises are keen to find out those who specialize or have expertise in specific fields. This information is used to assist internal research, consulting, resource allocation (across departments) and/or simple networking. Papers published by experts are most sought after within an enterprise. Research papers rated by even a contact or an expert are valued high as compared to documents that are not rated at all.

  People search is an OOTB GSA feature. People search in GSA becomes easy with provision for integration with lightweight directory access protocol (LDAP) servers. But for social search, we would need to develop additional components that would allow us to tag people as experts and allow end users to rate content. The data that is captured and stored via any of these custom components are imported as external metadata into the GSA appliance. GSA links any external metadata imported to the content already available in the index based on the content universal resource identifier (URI). We can thus accomplish linking between people information that's already indexed with that of expert and rating meta information.

  ExpertsConnect is a custom component that keeps track of people and contacts within GSA. These contacts provide a link to "prospective" experts. This information is fed to GSA as external metadata and hence is linked to the people data that are indexed from a directory residing on an LDAP server.

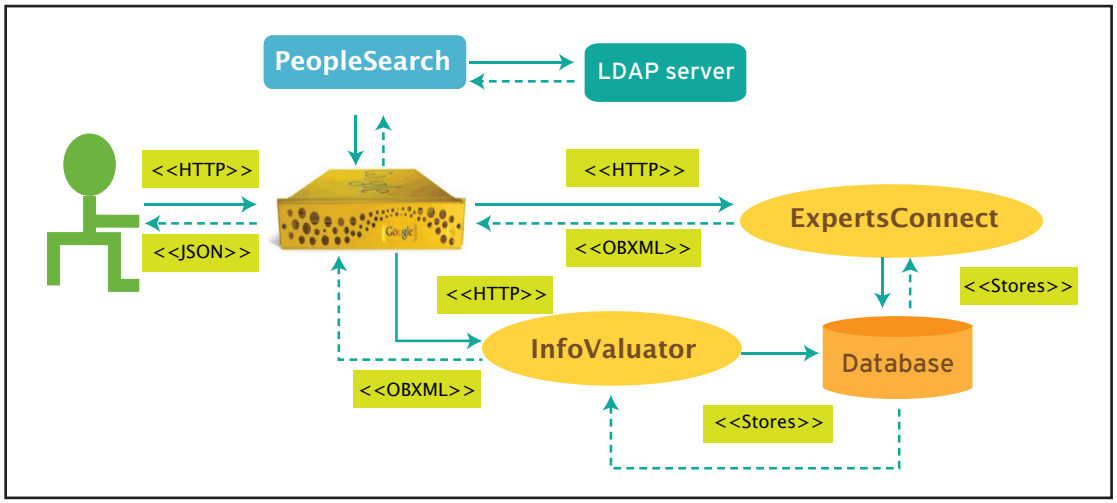

Figure 3

At query time, any people information associated with organic results are tagged. Additional details, pertaining to designation, department, etc., are fetched as a separate Onebox result based on the information that is indexed from LDAP. These details are linked with the contact details that are externally fed and this augmented set of "prospective" experts is displayed as a single Onebox result along with the organic search engine results page (SERP).

The organic results, when integrated with a rating system would allow people to rate and/or tag content. This can be achieved using the "InfoValuator" component.

- **InfoValuator:** InfoValuator component captures end-user rating and saves a combination of user identity, content URI and value rating in the backend data store. On a scheduled basis, this data is fed into the appliance as external metadata and is associated with the content and/or people information. The rating is fetched along with Google's organic search results. The UI can be designed to allow (any logged-in) user to rate the content on the fly, which the InfoValuator component would capture and store.
- **Accessing Desktop Search:** Google component for desktop search was decommissioned as of September 2011. But it was one solid platform-independent tool that allowed implementers to index and search desktop content. The tool came with a provision to configure an internal Google appliance as one of the search sources. Any search made from the Google toolbar for desktops returned results from both the desktop and from the appliance. Note: The Google desktop search tool maintained an index within the user's own file system.

  - **Alternates include:**
    - **COTS vendor like Autonomy and Microsoft FAST ESP** provide options to implement specific independent components for desktop search. But these components are not cost-effective for a simple desktop search.
    - **Apache Lucene and/or Solr is a good option for desktop search.** The tool which is easy to install and configure comes with a default Jetty server that can be used to index local file systems. Again, Lucene/Solr are limited to searching files within a desktop and cannot handle indexing e-mail servers out of the box.

Conclusions

There is no one search engine that fulfills all enterprise search requirements. HP Autonomy claims this lofty perch but it comes with a huge cost overhead, with the base cost crossing half a million dollars. Open source search engines are widely used by many but large enterprises still fear open source products for their lack of professional product support. Google Search Appliance has been the preferred tool for many medium to large enterprises for its ease of use, ready-to-go model and all-inclusive support package that comes with the purchase of the appliance. But

Architecture Alternate Using Apache Solr 3.1 & Nutch 1.3

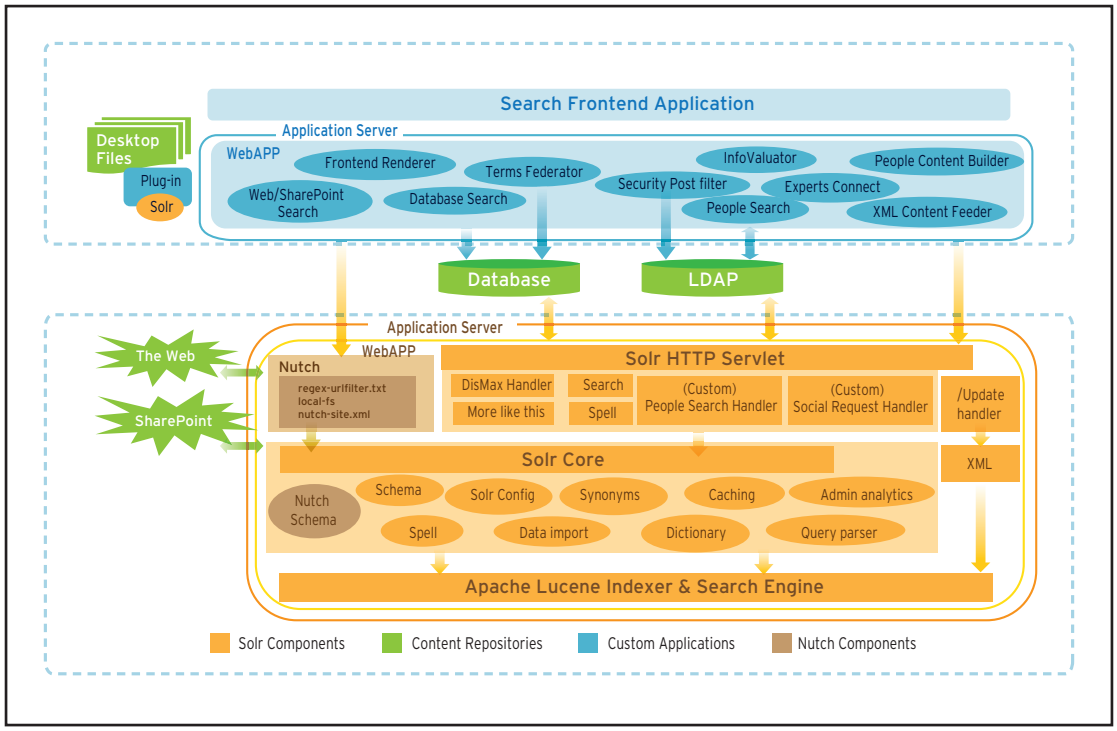

Figure 4

again, even Google is not the right fit for many requirements that we have seen so far. Custom search application development is inevitable and if well planned, we can basically use any tool in the market to implement enterprise search as a full-fledged application. Identifying the TCO (total cost of ownership) and ROI (return on investment) ahead of time would help enterprises strike a right balance between choosing the right search tool and investing time and money on extending the tool for required customizations.

References

Google Search Appliance Document Reference

Google Search Appliance SharePoint Implementation

Apache Solr 3.1

About the Author

Aruna Vaidyanathan is an Enterprise Search Architect within Cognizant's Portal Content Collaboration (PCC) Practice. With 10 years of IT experience, Aruna has spent seven-plus years integrating enterprise search projects within various industries such as manufacturing and logistics, consumer goods and life sciences. As a part of an 800-member practice with over 80 enterprise search professionals, Aruna's primary role is to evaluate top enterprise search products in the market and provide domain-specific consultation and search product implementation recommendations to Cognizant clients. Aruna holds a master's degree in computer applications and can be reached at Aruna.Vaidyanathan@cognizant.com.

About the Practice

Cognizant's Portal, Content and Collaboration (PCC) practice is an 800-member focus group that consolidates expertise and service delivery offerings in the PCC space. Enterprise Search is a primary practice with proven capabilities to provide end-to-end search solutions for Cognizant customers in the PCC domain. With over 80 professionals ranging from technical specialists to senior architects, the practice focuses on enterprise search consulting, systems integration, solution migration and post-implementation support. The practice holds partnership agreements with various leading enterprise search product vendors and can be reached at search@cognizant.com.

About Cognizant

Cognizant (NASDAQ: CTSH) is a leading provider of information technology, consulting, and business process outsourcing services, dedicated to helping the world's leading companies build stronger businesses. Headquartered in Teaneck, New Jersey (U.S.), Cognizant combines a passion for client satisfaction, technology innovation, deep industry and business process expertise, and a global, collaborative workforce that embodies the future of work. With over 50 delivery centers worldwide and approximately 137,700 employees as of December 31, 2011, Cognizant is a member of the NASDAQ-100, the S&P 500, the Forbes Global 2000, and the Fortune 500 and is ranked among the top performing and fastest growing companies in the world. Visit us online at www.cognizant.com or follow us on Twitter: Cognizant.

World Headquarters
500 Frank W. Burr Blvd.
Teaneck, NJ 07666 USA
Phone: +1 201 801 0233
Fax: +1 201 801 0243
Toll Free: +1 888 937 3277
Email: inquiry@cognizant.com

European Headquarters
1 Kingdom Street
Paddington Central
London W2 6BD
Phone: +44 (0) 20 7297 7600
Fax: +44 (0) 20 7121 0102
Email: infouk@cognizant.com

India Operations Headquarters
#5/535, Old Mahabalipuram Road
Okkiyam Pettai, Thoraipakkam
Chennai, 600 096 India
Phone: +91 (0) 44 4209 6000
Fax: +91 (0) 44 4209 6060
Email: inquiryindia@cognizant.com