

Building Search Based Applications

Introduction



Sushil Prabhu

Chief Executive Officer,
OpenCrowd



Stephen Coursen

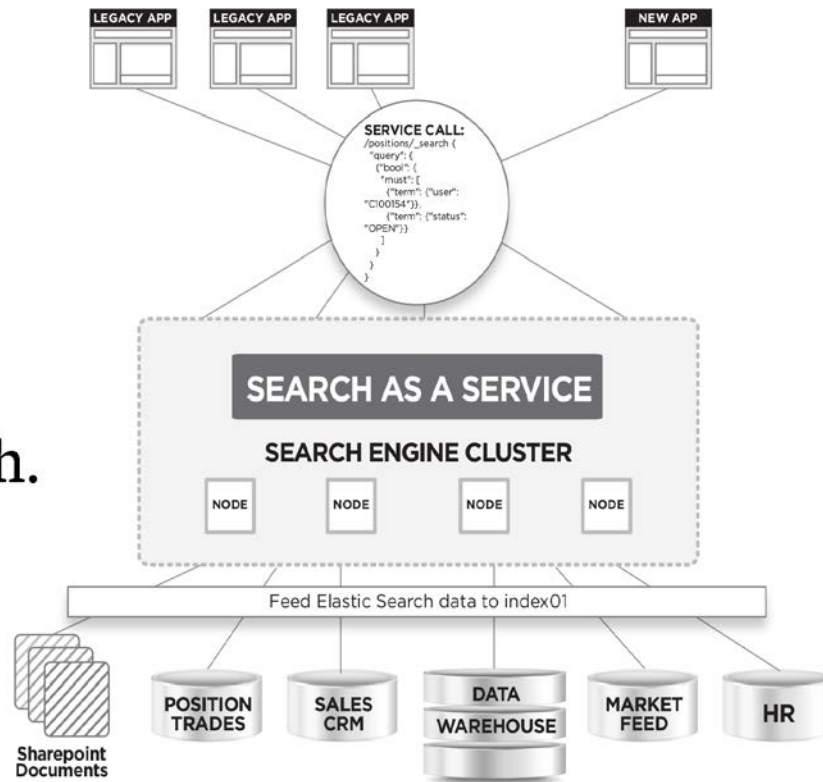
Director of Engineering,
OpenCrowd

Agenda

- What are Search Based Applications?
- How can you benefit?
- How do you build one?
- Demo

What are Search Based Applications?

Search Based Applications are a new breed of systems that leverage search as a backbone for data retrieval



A Search Based Application

Client Profile

Issue Tickets

Real-Time

Auto-Suggest

The screenshot shows a client portfolio dashboard for Acme, Inc. (ACE). The dashboard includes a search bar at the top right with the text "Hello Marlene | Logout" and a search icon. Below the search bar are tabs for "MY DASHBOARD", "MY ACCOUNTS", and "PRODUCTS & SERVICES". The main content area is divided into several sections:

- Client Profile:** Displays "Acme, Inc. (ACE)" with details: "Life Insurance", "MARKET CAP: \$57.31B", and "TOTAL ASSETS: \$885.25B".
- Service Requests:** A table with columns: "INITIATE TRANSACTION" (3), "CHANGE" (8), "INFORMATION" (28), "PROBLEM" (2), and "CONSULTATIVE" (0).
- Performance:** A line chart showing performance over time, with a date range of "Data as of 26-MAY-2014 02:53 PM EDT".
- Documents:** A table with columns: "TYPE", "TITLE", "STATE", "VERSION", "DESCRIPTION", "LAST UPDATE", "UPLOAD NEW", and "SIGNATURE". The table lists several documents, including "Singapore Lockbox Terms & Conditions", "Singapore Clearing Terms & Conditions", "Global Account Terms", "Banking Resolution", "Certificate for Incumbency", and "W-9".
- Contacts:** A list of contacts including "Nadine Barrett" (Treasury Services Sales), "Anne Glower" (Sales), and "Shawn Tyler" (Sales).

Client Managers

Client Contracts



Index



Confluence

SIEBEL

Logfiles...

Index

Current challenge in the enterprise

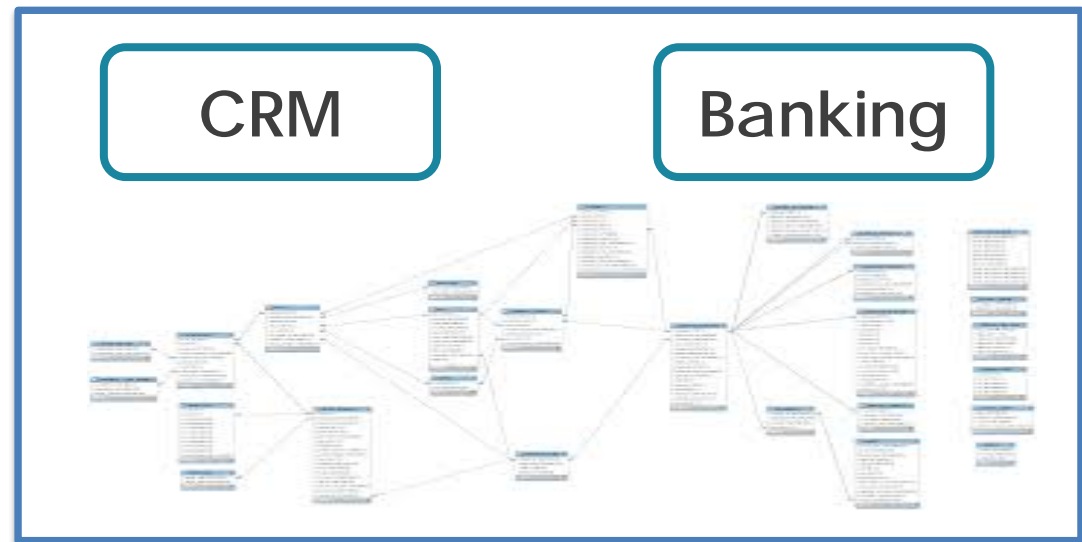
There is a constant need to enrich applications with new sets of data and in many cases these data sets are huge

The issue is that in most applications, data relationships are predefined

Bringing new and a large set of data in an application is a major undertaking

New Data

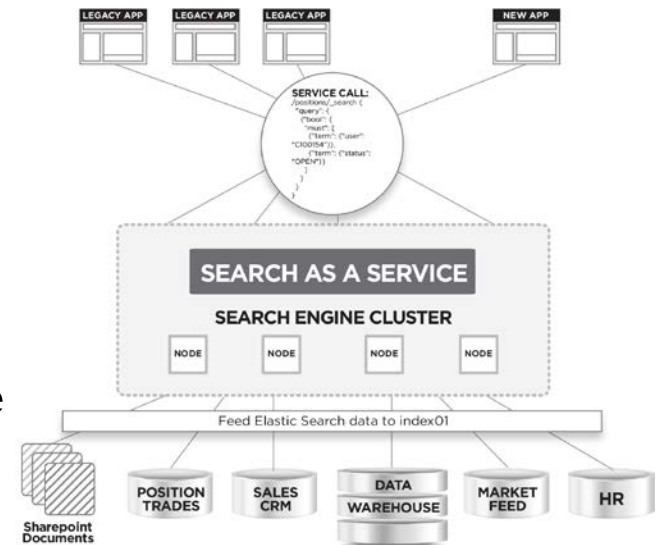
Boxed in



The real need is for systems that can adapt to new data

Search Based Architecture offers four key benefits making this pattern a promising solution for data intensive applications:

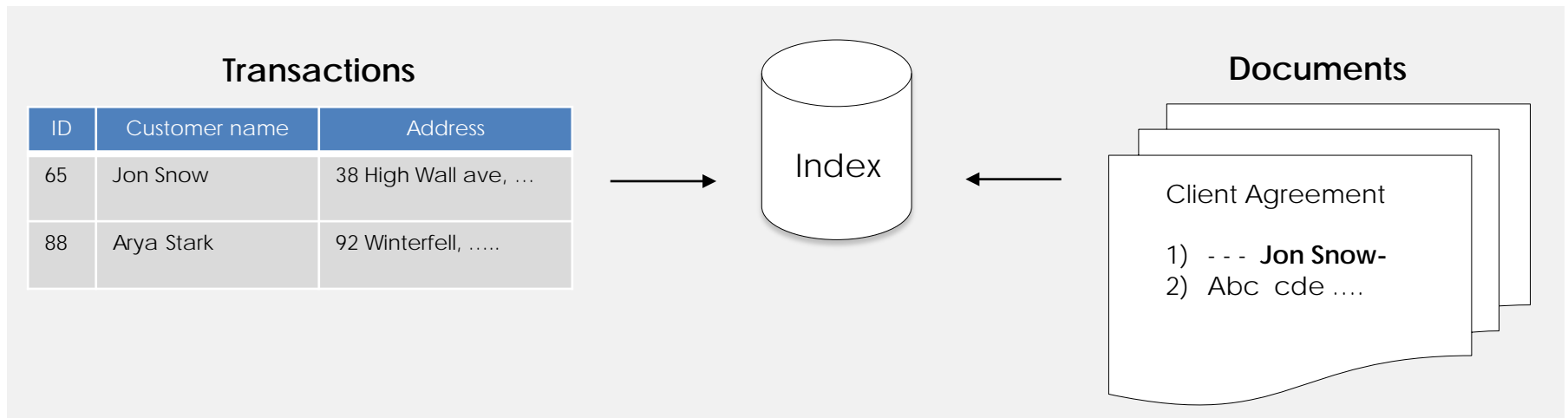
- Adaptive to new data
- Split second response time
- Very Intuitive end user experience
- Access to data can be offered as a service



Adaptive to new data – Dynamic structure


The key difference in Search Based Applications is that information is primarily stored and retrieved from an INDEX

- Schema can grow dynamically thus enabling the relationship to evolve and making it easier to bring new data types
- Data types are mapped dynamically



Split second performance to retrieve data

- Search engines are built for extremely fast retrieval of information
- Information is stored in flat de-normalized structure making retrieval a lot less CPU intensive operation




InfoQ
En | 中文 | 日本 | Fr | Br
1 100 271 Max unique visitors

<http://www.infoq.com/articles/kiln-elasticsearch>

How Fog Creek Software Made Kiln's Search 1000x Faster with Elasticsearch

Posted by [Kevin Gessner](#) on Sep 11, 2013 | [4](#) Discuss



Split second performance to retrieve data

- Most engines perform a wide range of calculations from a simple count to complex aggregations on the fly
- Queries previously deemed complex and resource intensive are a lot easier:

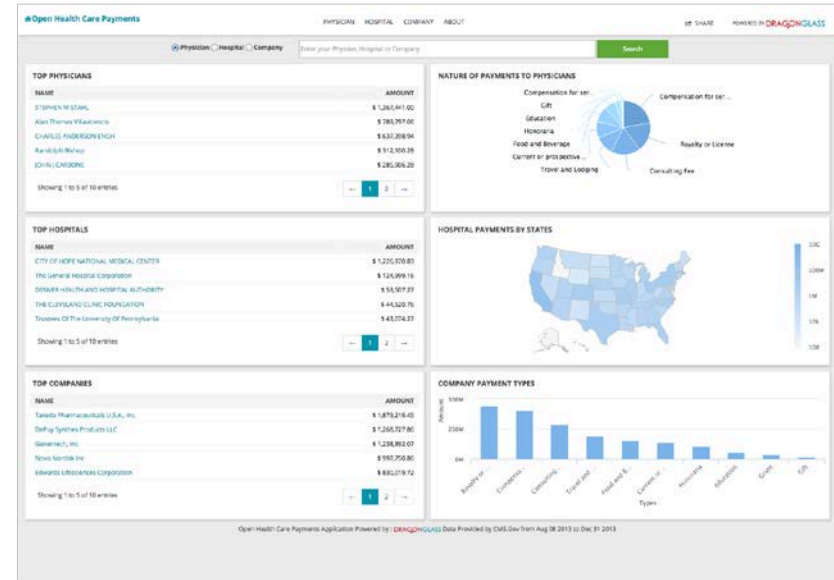
“Give me all client assets grouped by asset classes that have exposure to Greece and were purchased last year with total asset value greater than 100 million and exclude assets that belong in the sin category from this year”

Very Intuitive End User Experience

You can bring Google or Amazon style easy-to-use experience to enterprise applications such as:

- Auto suggest
- Auto complete
- Facets
- Alerts
- Fuzzy search
- Analytics on search results

Improve findability by guiding the user based on behavioral intelligence and trends.



Ex: Open Healthcare Payments Application
openpayments.dragonglass.me

Before and After Search UX

Older form based search:

The screenshot shows a search form with three tabs: "Physician" (selected), "Teaching Hospitals", and "Company Making Payments". The form contains six input fields: "First Name", "Last Name", "City", "State" (a dropdown menu), "ZIP", and "Specialty". At the bottom, there is a message: "Please be patient, search results take a few moments to load." and a "Search" button.

Older search requires prior knowledge and precise input which can be un-forgiving

Newer search with auto-suggest:

The screenshot shows a search interface with three radio buttons: "Physician" (selected), "Hospital", and "Company". A search input field contains the text "John". Below the input field is a dropdown menu with the following suggestions:

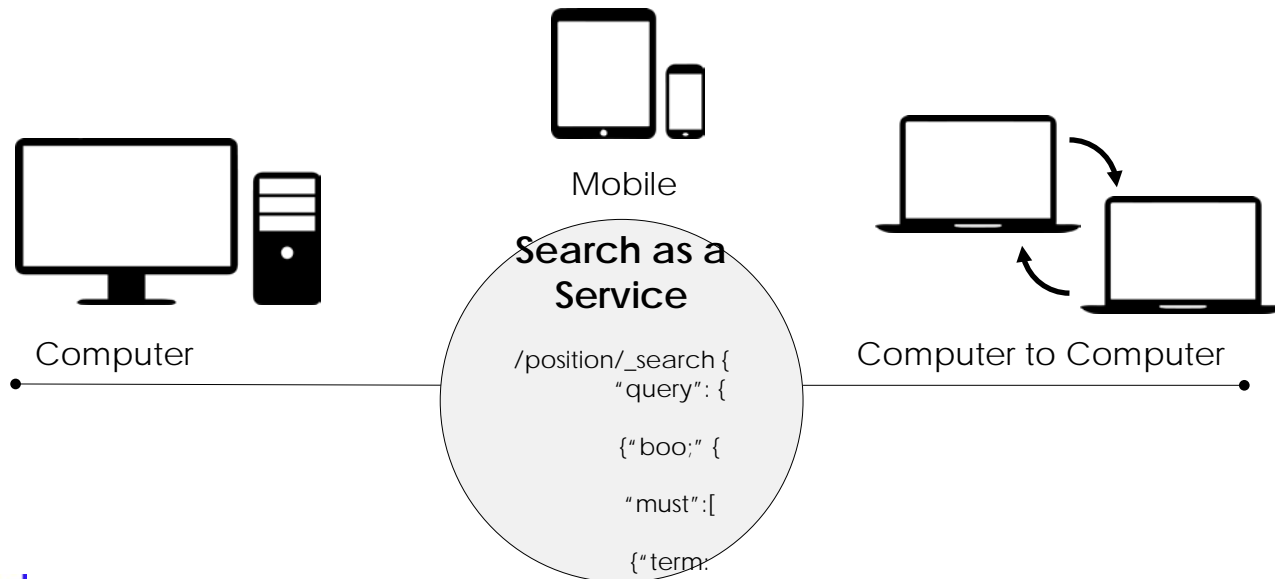
- A. SILVIA ROSS 3101 JOHN HUMPHRIES WYND RALEIGH NC
- AAMIR RASHEED 52 HARRISON ST JOHNSON CITY NY
- AARON DAVID JOHNSON 726 S 2ND ST BISMARCK ND
- AARON DONALD JOHN KULWICKI 6001 E BROAD ST COLUMBUS OH
- AARON DWIGHT JOHNSON 1485 CHESTER BLVD RICHMOND IN
- AARON ELLIOTT JOHNSON 515 SOUTH MOORE ST. BLUE EARTH MN
- AARON ERIC JOHNS 3300 W 15TH AVENUE GARY IN
- AARON GRANT ORME 35 JOHNSON AVE DILLON MT
- AARON JOHN GUYER 3334 CAPITAL MEDICAL BLVD TALLAHASSEE FL
- AARON JOHN MOEGGENBORG 2020 E PRESTON ST MT PLEASANT MI

A green "Search" button is located to the right of the input field.

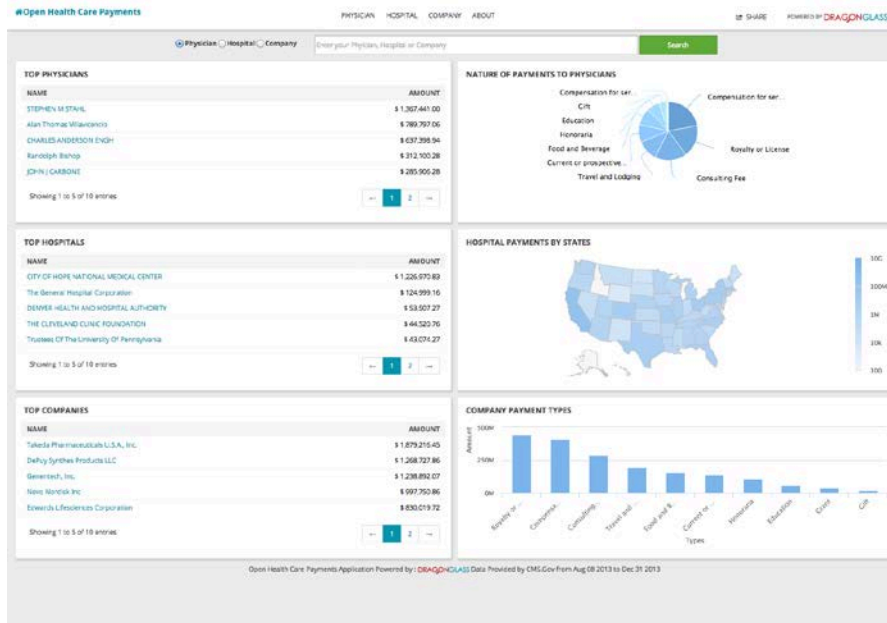
Newer search is self correcting with a focus on guiding the user to discover what they are looking for

Access to data can be offered as a service

- Most search engines offer simple web services access to the Index, specifically REST
- REST based API makes it very easy to offer “Search as a Service”
- This micro service can be consumed by applications or other systems



A Sample Search Based application

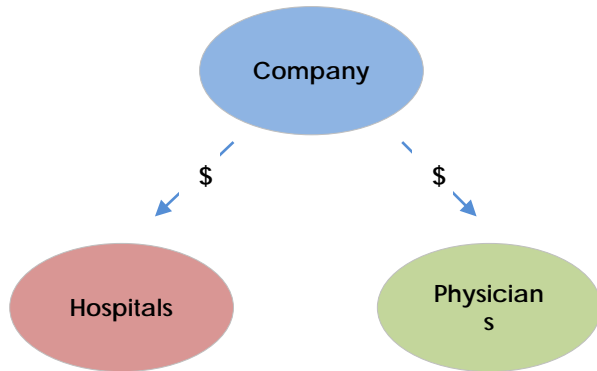


The Open Health Care Payments application uses DragonGlass to provide insight into payments to physicians and hospitals from medical companies. The application makes it easy to find physicians and hospitals as well as provides analysis of the payments they have received. Raw data for the application is publicly available at: openpaymentsdata.cms.gov site.

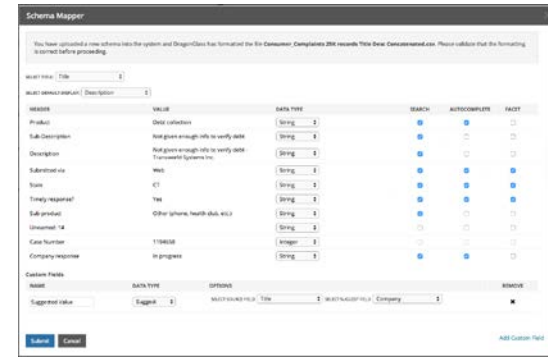
openpayments.dragonglass.me

Open Healthcare Payment Application

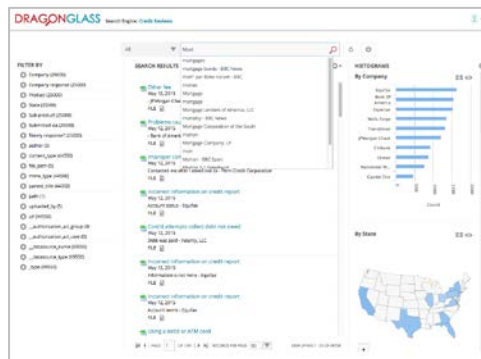
Building a Search Based Application



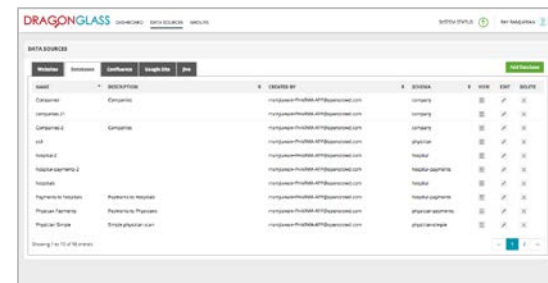
1. Model the Information



2. Create Index & Upload Data



3. Query the Information

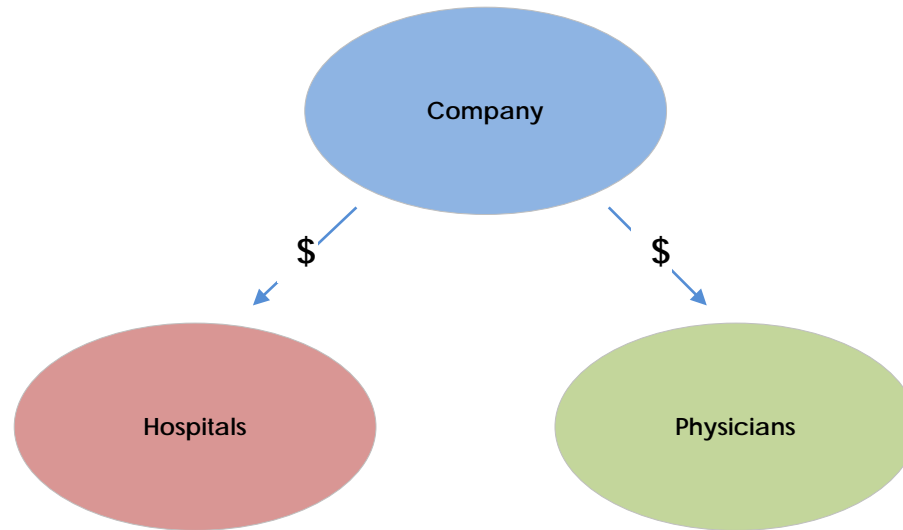


4. Enhancing the System with New Data

Note: In this exercise we will use Elastic Search Engine

Model your base entities

Just like with any other app, the data you use will impact the answers you can get. Since we wanted physician, hospital, and company data, we broke the documents up into those 3 entities:



Physician {"first name": "STEPHEN", "last name": "BURKHART", "state": "TX", "zip code": "78258", "specialty": "Allopathic & Osteopathic Physicians", "total_payment_amount": 7356275.69,

Create an index & upload the data

- You can either create an index with a schema or you can decide to use dynamic schema which is created as one uploads the data
- The data from cms.gov site is available as a giant three million record csv. To make it easier we loaded the data in a MySQL database.
- We wrote SQL queries to model the base entities and we set these queries as data sources in DragonGlass (basically REST API calls to Elastic Search)
- As the files from MySQL were uploaded/indexed in Elastic Search the data was ready to be searched

You can now query the information in multiple ways

- As a simple REST call from your browser
`http://openpayments.dragonglass.me:8080/`
- From a programmable widget
- A instant data discovery search result page with facets, auto suggest, etc.
- A Micro service to be consumed by other systems



Search
Discovery
App

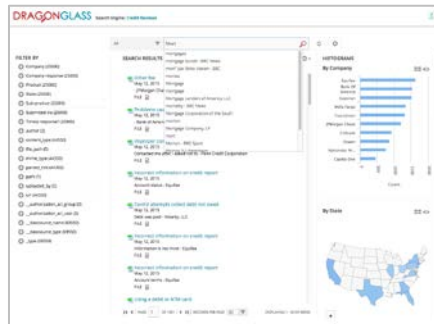


Chart widget



How do I bring new data in my current app?

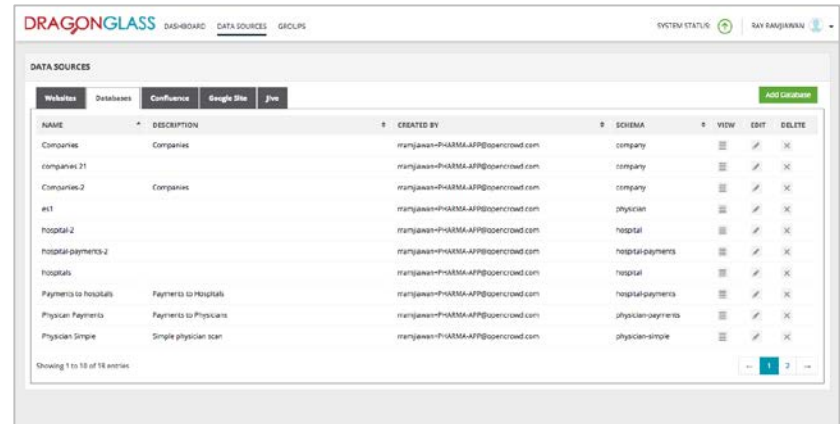
- For example: Associating state wide Medicare & Medicaid payment information with the OpenCMS data
- As all the base entities have a "state" field the initial association will happen automatically
- Once the new data is loaded all existing search queries, aggregations...will automatically include the new data in the result set

```
Physician {"first name": "STEPHEN", "last name": "BURKHART", "state": "TX", "zip code":  
"78258", .....
```

```
Medicare { "hospital_id": 5323, "physician_id": 223436, "state": "TX", "city": "Houston",  
"payment_amount": 324.73.....
```

DragonGlass Admin UI

– data upload



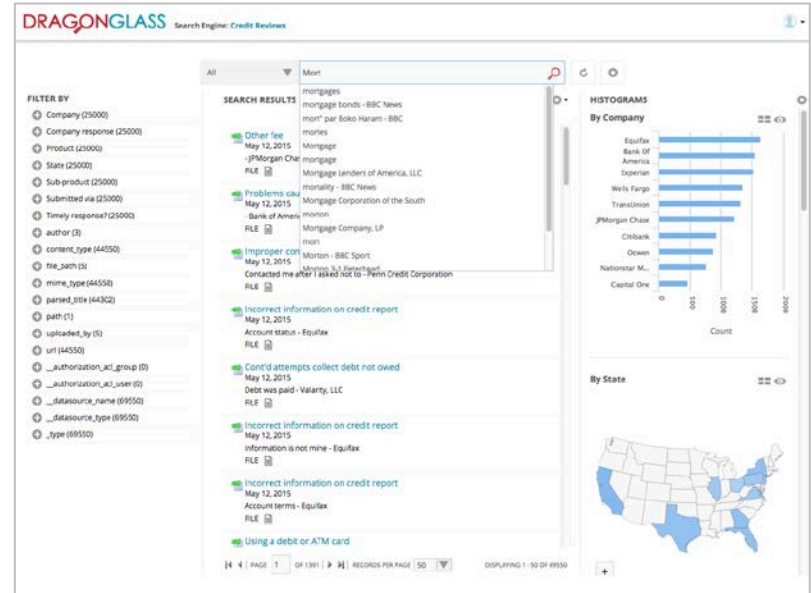
The screenshot shows the DragonGlass Admin UI interface. At the top, there is a navigation bar with 'DRAGONGLASS' logo, 'DASHBOARD', 'DATA SOURCES', and 'GROUPS'. On the right, there are 'SYSTEM STATUS' and 'RAY BANGKUN' indicators. Below the navigation bar, there is a 'DATA SOURCES' section with a '+ ADD Database' button. A table lists various data sources with columns for NAME, DESCRIPTION, CREATED BY, SCHEMA, VIEW, EDIT, and DELETE. The table contains 18 entries, with the first few being 'Companies', 'companies 21', 'companies 2', 'et1', 'hospital 2', 'hospital payments 2', 'hospitals', 'Payments to hospitals', 'Physician Payments', and 'Physician Simple'.

NAME	DESCRIPTION	CREATED BY	SCHEMA	VIEW	EDIT	DELETE
Companies	Companies	manjawan-pharma-afp@opencrowd.com	company			
companies 21		manjawan-pharma-afp@opencrowd.com	company			
companies 2	Companies	manjawan-pharma-afp@opencrowd.com	company			
et1		manjawan-pharma-afp@opencrowd.com	physician			
hospital 2		manjawan-pharma-afp@opencrowd.com	hospital			
hospital payments 2		manjawan-pharma-afp@opencrowd.com	hospital payments			
hospitals		manjawan-pharma-afp@opencrowd.com	hospital			
Payments to hospitals	Payments to Hospitals	manjawan-pharma-afp@opencrowd.com	hospital payments			
Physician Payments	Payments to Physicians	manjawan-pharma-afp@opencrowd.com	physician payments			
Physician Simple	Simple physician scan	manjawan-pharma-afp@opencrowd.com	physician-simple			

Search as a service REST API

http://openpayments.dragonglass.me:8080/_search/default/physician-simple/_search?aggs=payment_state_sort_payment_amount

Data Discovery Application



Programmable Widget

SEARCH CHART PORTLET

Portlet Title:

Entries:

Search Path:

Search Query:

Chart type:

Aggregation Name:

Chart definitions:

Drilldown definitions:

DragonGlass, Enabling Search Powered Applications



- Automates data extraction
- Creates a Index store
- Auto generates a data discovery application
- Provides dashboard widgets and API
- Admin services to manage and monitor the platform

Thank You

Visit us at:
DragonGlass.me

41 East 11th St, 11th Floor
New York, NY 10003
www.opencrowd.com

Phone: 917-770-9229
Fax: 212-768-1792
E-mail: info@opencrowd.com

Sushil Prabhu
Phone: 212-768-1730
sprabhu@opencrowd.com