GRACE

powered by

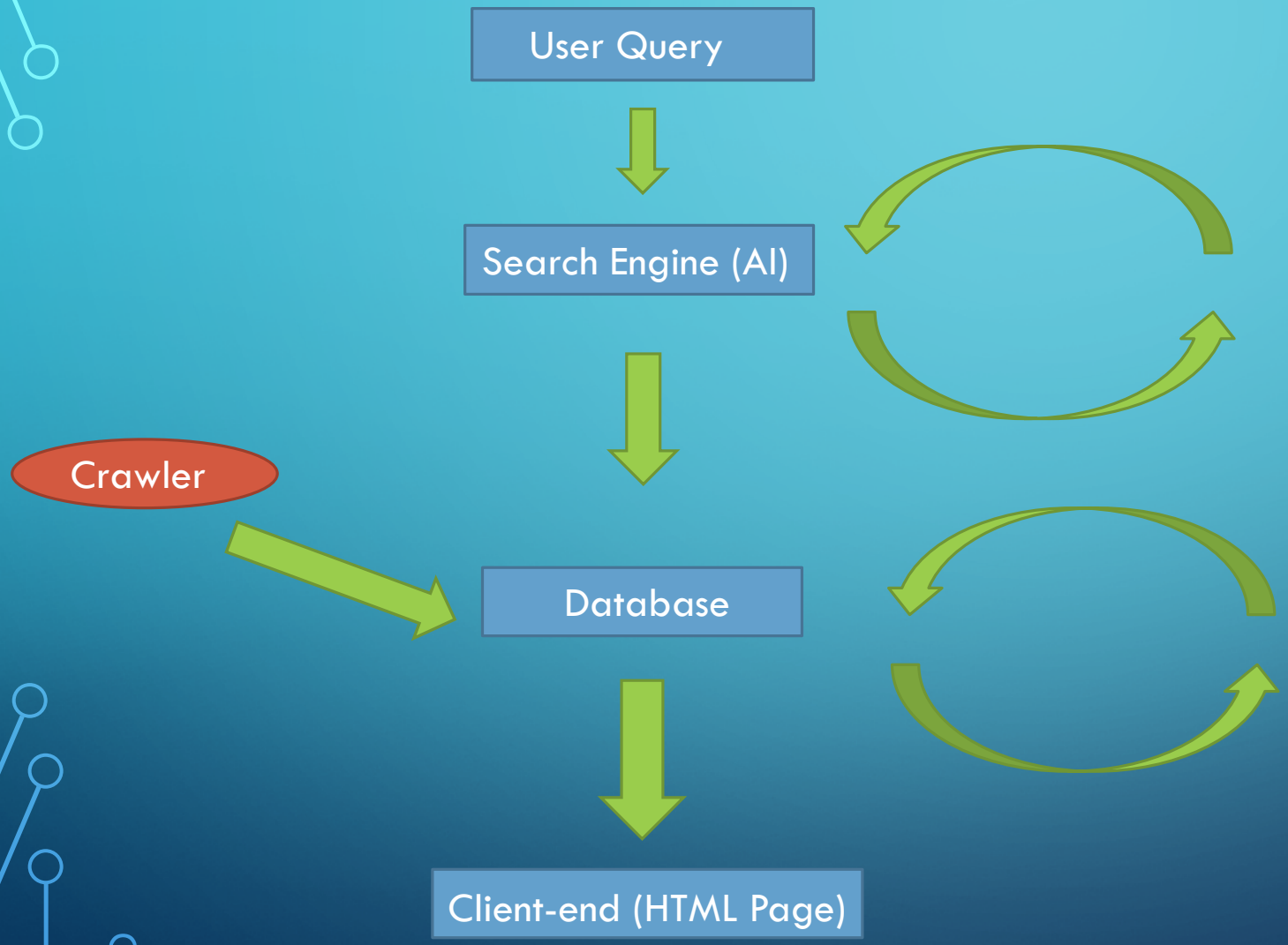
IBM Bluemix™

Sadik Erisen, Tamar Tokman, Ross Kernez, Anis Shili, Tristan Draper, Arnav Lohe

CORE PROJECT

- How it all began?
- What is it?
- What does it do?
- How does it work?

THE DATA FLOW



AI

- Language Processing
- Image Processing *
- Voice Recognition

* TBD

Database Processing

- Ranking
- Indexing
- Clustering
- Traversing

CORE PROJECT – CLIENT-SIDE

- Applying Rest API and retrieving data from our back-end.
- Consuming live data using web services.
- Using Angular 2 technologies

Grace Search Engine

CLIENT-SIDE SCREENSHOT

```
62
63    callWeatherService()
64    {
65      this._sharedService.findWeather(this.id_city, this.id_state)
66        .subscribe(
67        lstresult =>
68        {
69          this.op_city = lstresult["query"]["results"]["channel"]["location"]["city"];
70          this.op_region = lstresult["query"]["results"]["channel"]["location"]["region"];
71          this.op_country = lstresult["query"]["results"]["channel"]["location"]["country"];
72          this.op_text = lstresult["query"]["results"]["channel"]["item"]["condition"]["text"];
73          this.op_temp = lstresult["query"]["results"]["channel"]["item"]["condition"]["temp"];
74          this.location = lstresult["query"]["results"]["channel"]["item"]["condition"]["location"];
75
76          return lstresult;
77
78        },
79
80        error => {
81          console.log("Error. The findWeather result JSON value is as follows:");
82          console.log(error);
83        }
84        );
85    }
```

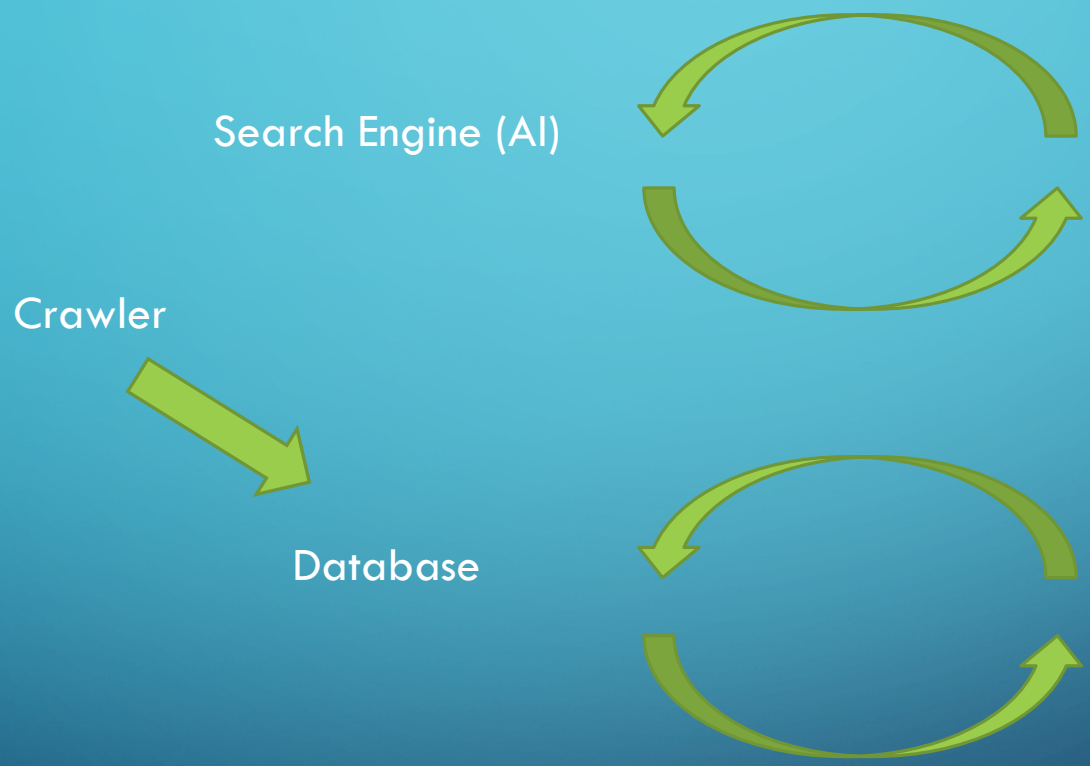
An example of our Weather Component

CORE PROJECT – SERVER SIDE

- Search Engine
- Web Crawler
- Retrieving semi-structured / structured data from database
- IBM Bluemix Auto-Scale / Better Performance
- IBM Bluemix Storage

FOUNDATIONS OF THE SEARCH ENGINE

- Indexing
- Query
- Ranking



- Language Processing
- Image Processing *
- Voice Recognition

* TBD

Database Processing

- Ranking
- Indexing
- Clustering
- Traversing

INDEXING

- Building the Inverted Index
- Query types
- Parsing The Collection

```
def cons(self, i):
    """cons(i) is TRUE <=> b[i] is a consonant."""
    if self.b[i] == 'a' or self.b[i] == 'e' or self.b[i] == 'i' or self.b[i] == 'o' or self.b[i] == 'u':
        return 0
    if self.b[i] == 'y':
        if i == self.k0:
            return 1
        else:
            return (not self.cons(i - 1))
    return 1

def m(self):
    """m() measures the number of consonant sequences between k0 and j.
    if c is a consonant sequence and v a vowel sequence, and <..>
    indicates arbitrary presence,

       <c><v>       gives 0
       <c>vc<v>     gives 1
       <c>vcvc<v>   gives 2
       <c>vcvcvc<v> gives 3
       ....
    """
    n = 0
    i = self.k0
    while 1:
        if i > self.j:
            return n
        if not self.cons(i):
            break
        i = i + 1
    i = i + 1
```

QUERY TYPES

- One-word
- Free-Text
- Phrase

- Parsing the Collection
  - Concatenation
  - Lowercasing words
  - Tokenization
  - Filtering
  - Stemming

RANKING

- Machine Learning Techniques
  - Naive Bayes Classifier / Neural networks
  - Linear Regression / Neural networks
  - Decision Trees

➢ Inverse document frequency (IDF)

➢ Term frequency (TF)

```
QUEUE_FILE = PROJECT_NAME + '/queue.txt'
CRAWLED_FILE = PROJECT_NAME + '/crawled.txt'
NUMBER_OF_THREADS = 256
queue = Queue()
Spider(PROJECT_NAME, HOMEPAGE, DOMAIN_NAME)


# Create worker threads (will die when main exits)
def create_workers():
    for _ in range(NUMBER_OF_THREADS):
        t = threading.Thread(target=work)
        t.daemon = True
        t.start()


# Do the next job in the queue
def work():
    while True:
        url = queue.get()
        Spider.crawl_page(threading.current_thread().name, url)
        queue.task_done()


# Each queued link is a new job
def create_jobs():
    for link in file_to_set(QUEUE_FILE):
        queue.put(link)
    queue.join()
    crawl()
```

CRAWLERS

Our version of Internet Crawler

What does it do?

- Finds all the clickable elements
- Creates a queue of those elements

Our version of Scrapers

- Find and parse text into database

  using JSON format

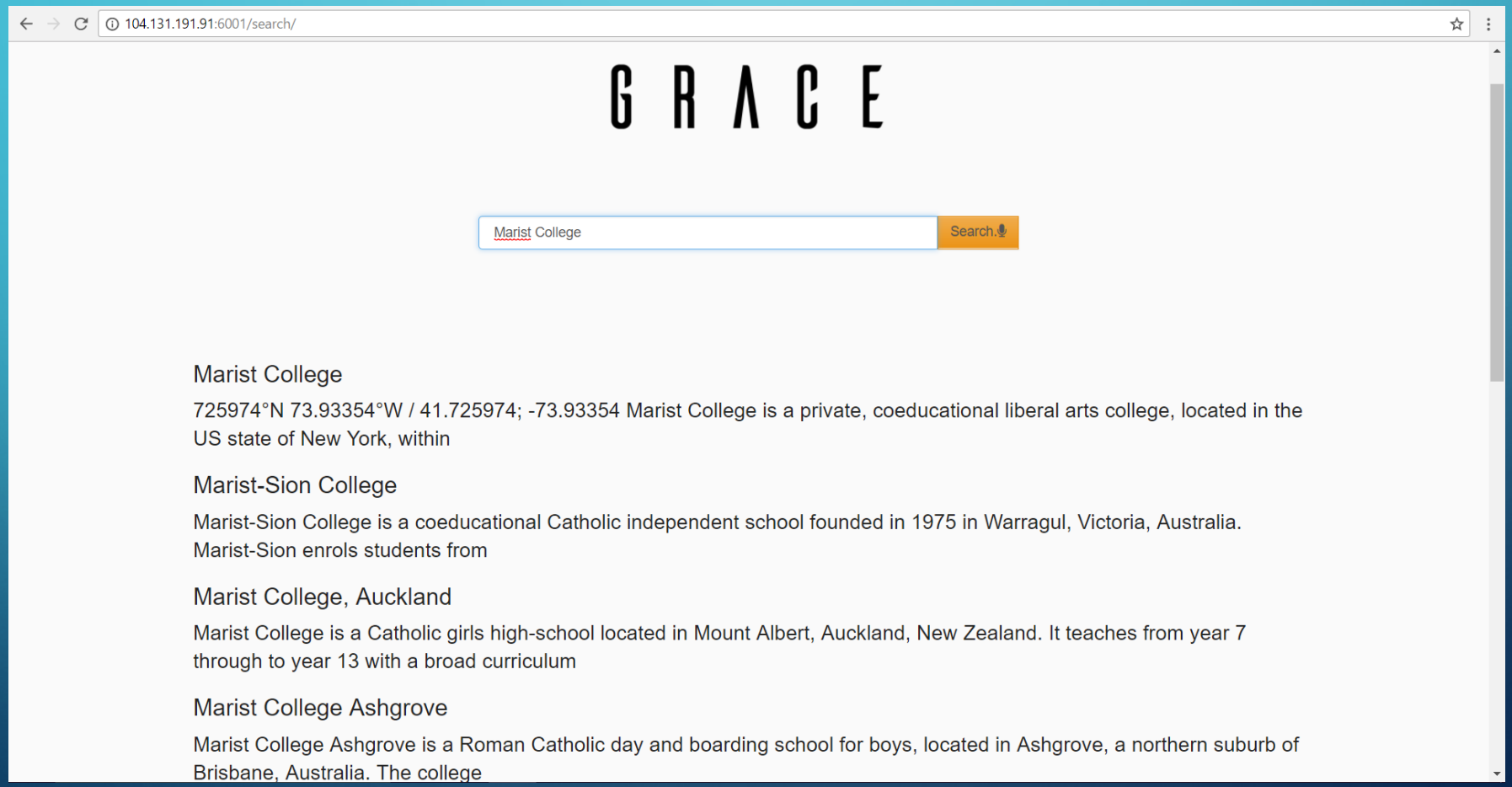
An Example From Web Services
Pulling data every 15 min from News.org

```
Date: 3/May/2017
Author : Sadik F. Erisen

Credited to Newsapi.org :
Here are API services for newsapi.org thats where are getting our
semi-sturctured data.
https://newsapi.org/v1/articles
https://newsapi.org/v1/

'''

import requests
import json
import urllib.request, urllib.error, urllib.parse

api_key = '96f666317b07432e8a120f8a21d119af'

class NewsServices():

    def parse_articles(articles):
    '''
    This function takes in a response to the NYT api and parses
    the articles into a list of dictionaries
    '''
    news = []
    for i in articles['response']['docs']:
        dic = {}
        dic['id'] = i['_id']
        if i['abstract'] is not None:
            dic['abstract'] = i['abstract'].encode("utf8")
        dic['headline'] = i['headline']['main'].encode("utf8")
        dic['desk'] = i['news_desk']
        dic['date'] = i['pub_date'][0:10] # cutting time of day.
        dic['section'] = i['section_name']
        if i['snippet'] is not None:
            dic['snippet'] = i['snippet'].encode("utf8")
        dic['source'] = i['source']
        dic['type'] = i['type_of_material']
        dic['url'] = i['web_url']
        dic['word_count'] = i['word_count']
        # locations
        locations = []
        for x in range(0,len(i['keywords'])):
            if 'glocations' in i['keywords'][x]['name']:
                locations.append(i['keywords'][x]['value'])
```

IBM Bluemix™

- Using IBM Bluemix
- Bluemix VM for the Core Application
- Bluemix Autoscaling for the Web Crawlers

RESULTS OF THE CLIENT-END



CONCLUSION

G R A C E

Languages Used:
JavaScript, TypeScript, Python

Frameworks Used:
Angular 2, Django, Scikit Learn, NLTK (language processing), SciPy

Q & A

- Developers :
  - Sadik Erisen : serisen@me.bergen.edu
  - Tamar Tokman : tamartokman@gmail.com
  - Ross Kernez : rosskernez@yahoo.com
- Mentor:
  - Prof. Emily Vandalovsky : evandalovsky@bergen.edu

Thank You

Bergen Community College

BCC STEM

STEMatics

Program Director : Luis De Abreu
ldeabreu@bergen.edu