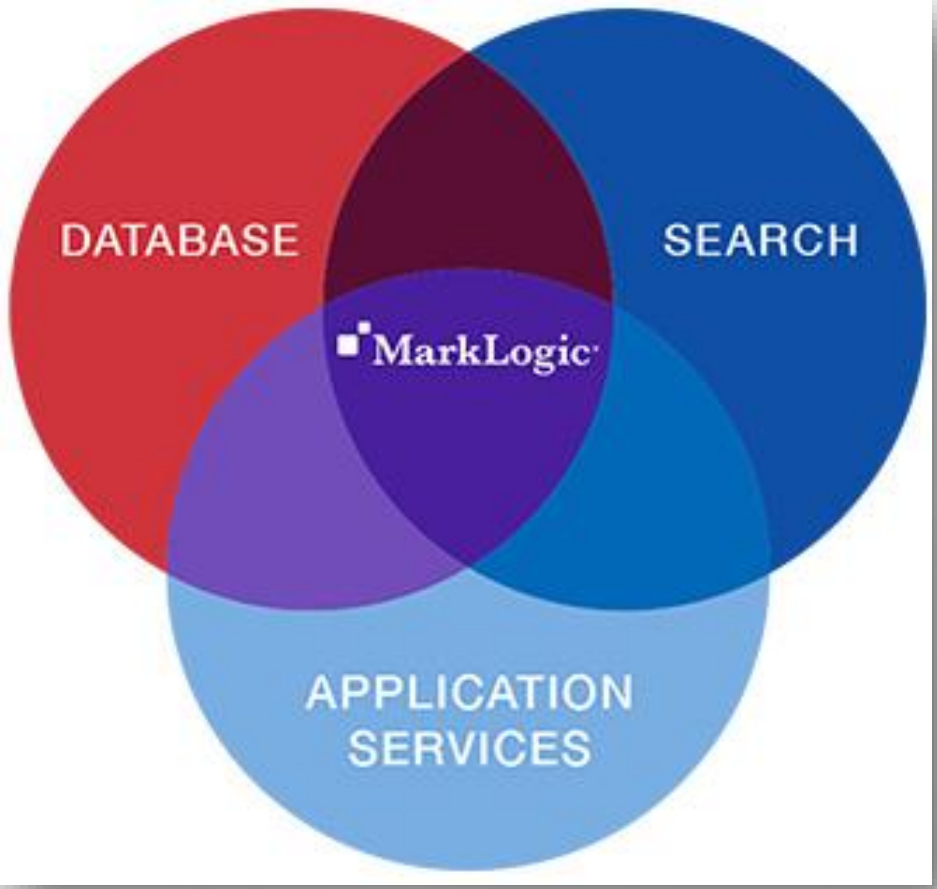
MarkLogic

BUILD. ITERATE. INNOVATE. FASTER

First Look at MarkLogic

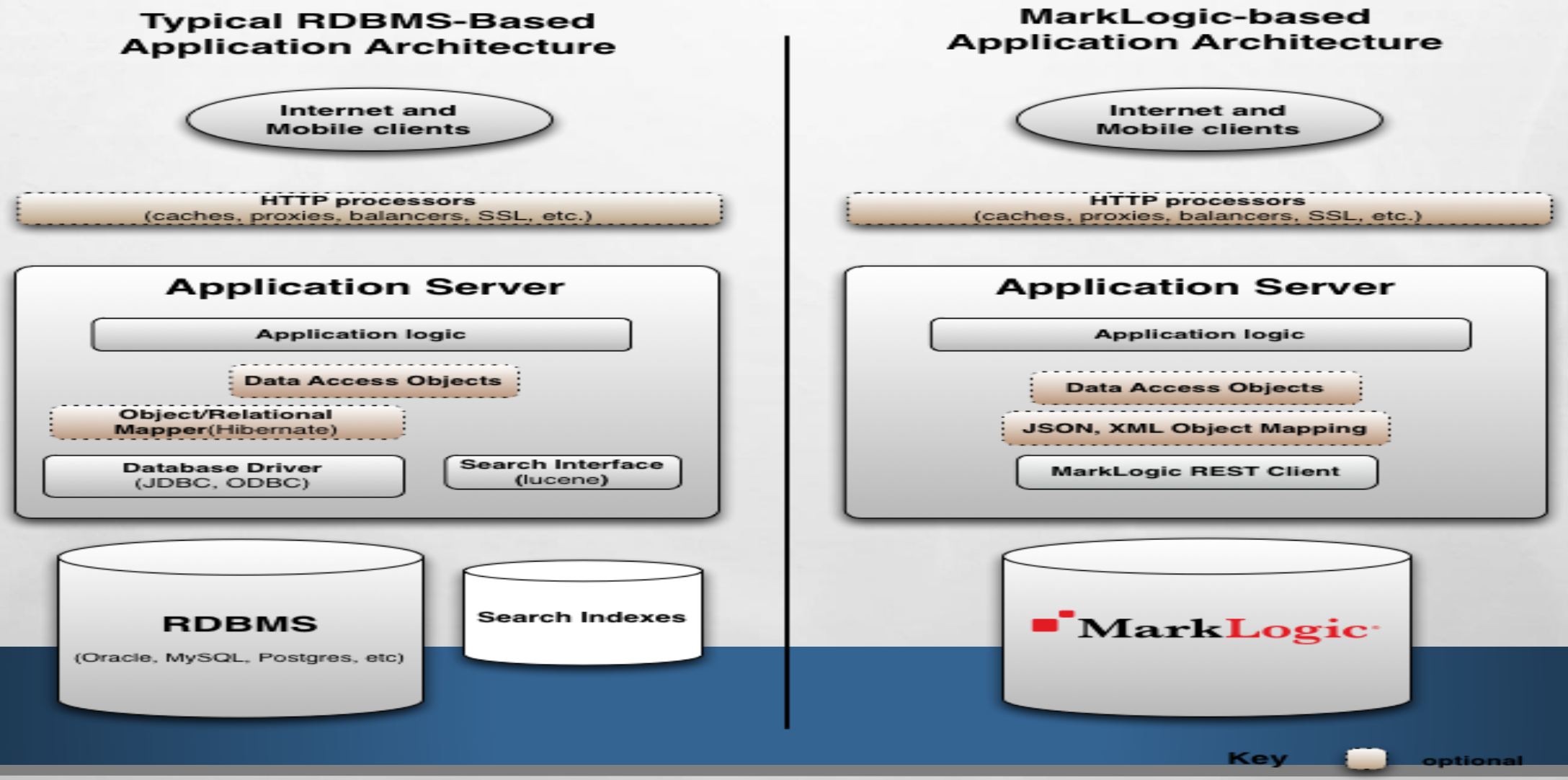


BRIEF HISTORY

- Founded In The Year 2001.
- Founders : Christopher Lindblad, Paul Pedersen and Frank R. Caufield
- Initially baptized As Cerisent.
- Initially focused to address shortcomings with existing search and data products by using XML document markup.
- Used XQuery as the query standard for accessing collections of documents.

RDBMS v MarkLogic

MarkLogic™

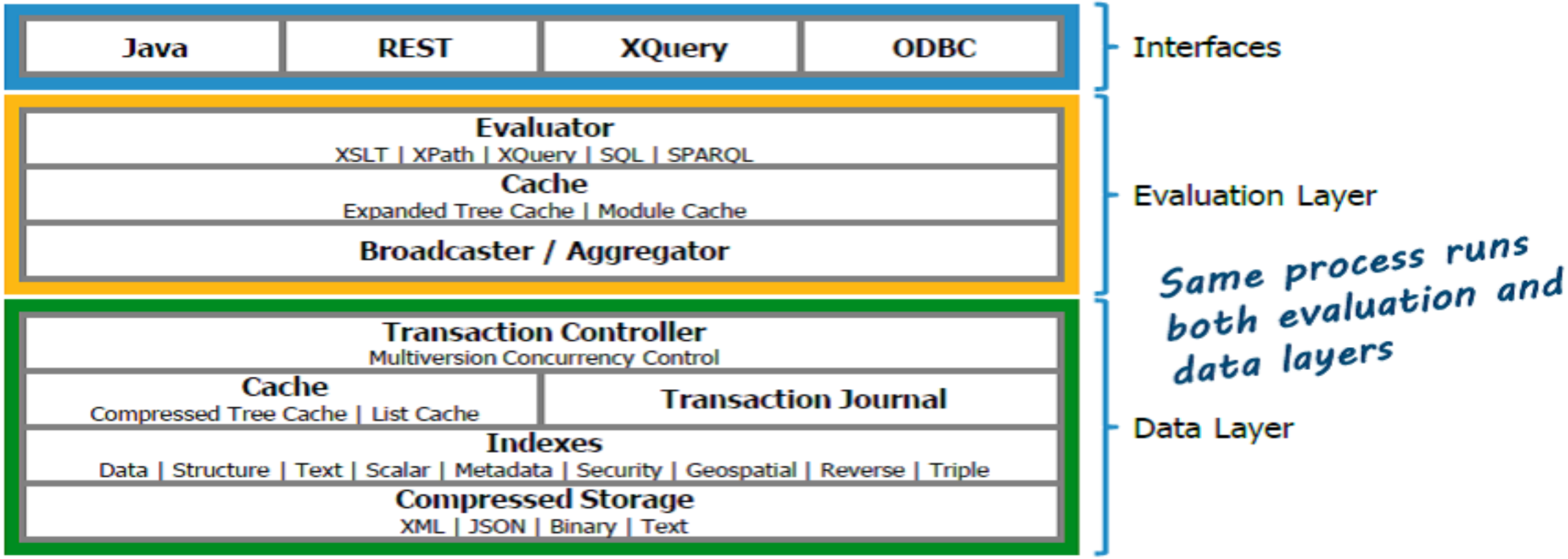


System Architecture

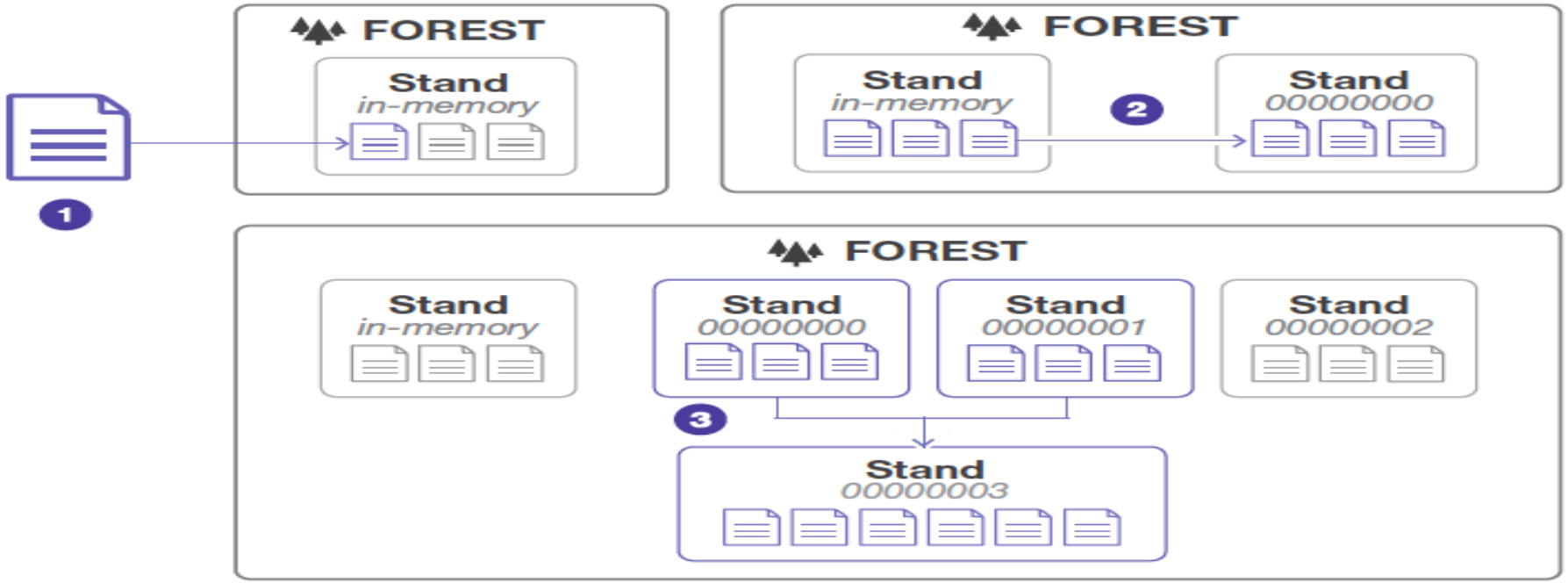
MarkLogic™

MarkLogic

MarkLogic Architecture



Data Management



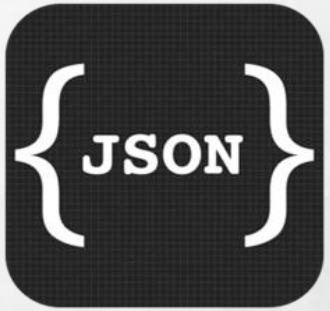
MarkLogic™

Key Features

- DOCUMENT CENTRIC
- MULTI MODEL
- TRANSACTIONAL (ACID)
- SEARCH ORIENTED
- STRUCTURE AWARE
- SCHEMA AGNOSTIC
- HIGH PERFORMANCE AND SCALABILITY
- HIGH AVAILABILITY

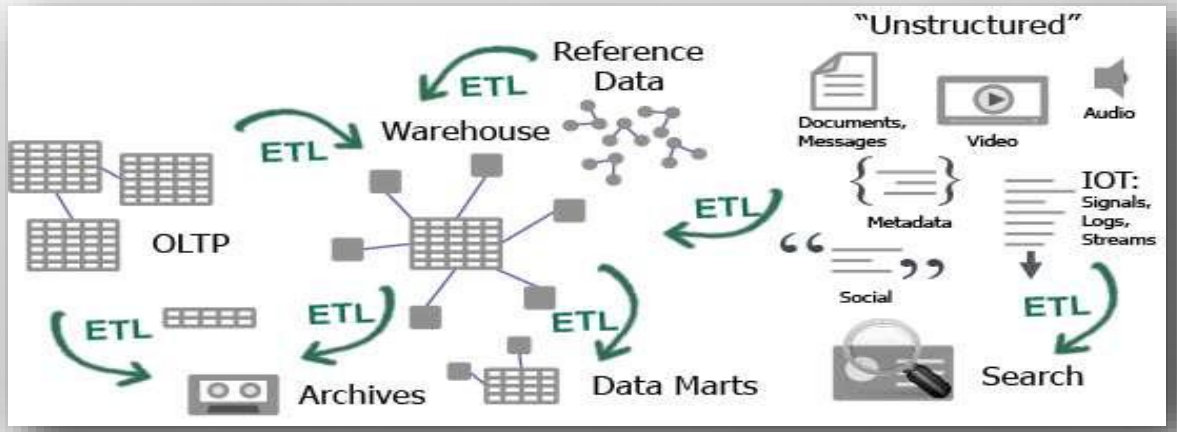
Document Centric

- **SUPPORTED DOCUMENT TYPES :-**
  - XML
  - JSON
  - TEXT DOCUMENTS
  - RDF TRIPLES
  - BINARY DOCUMENTS

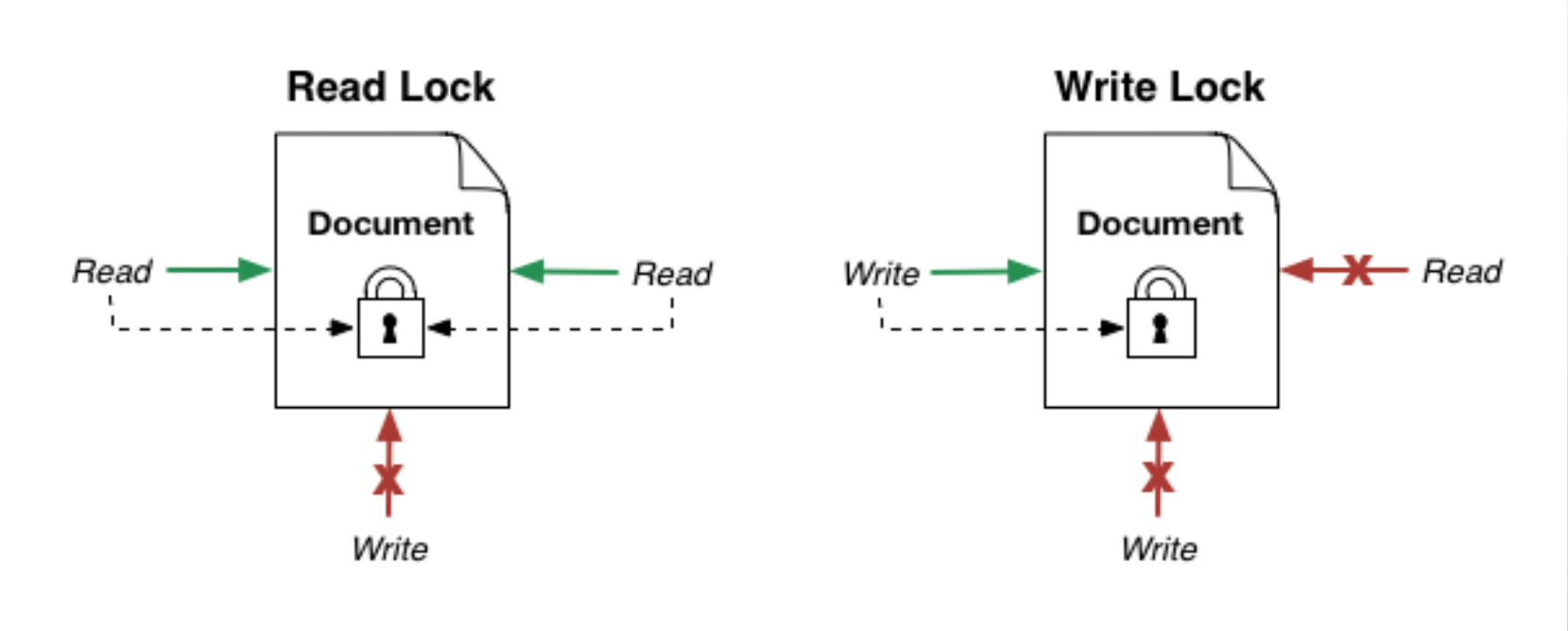


Multi-Model

- **TYPES OF DATA MODEL:-**
  - Document Store
  - Native XML
  - Resource Description Framework(RDF)
  - Search Engine



Transactional



Search Oriented

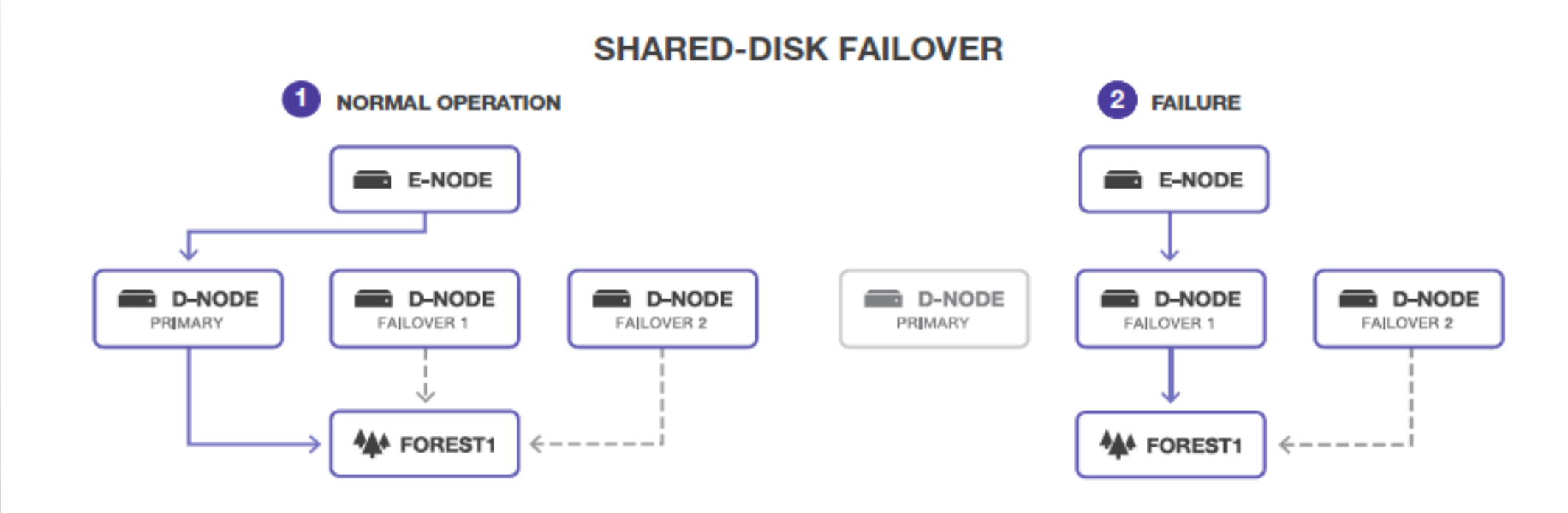
- SIMPLE QUERIES (URI/KEY-VALUE LOOK UP)

```
curl -X GET --anyauth --user username:password \
'http://myhost:port/v1/documents?uri=/my-document'
```

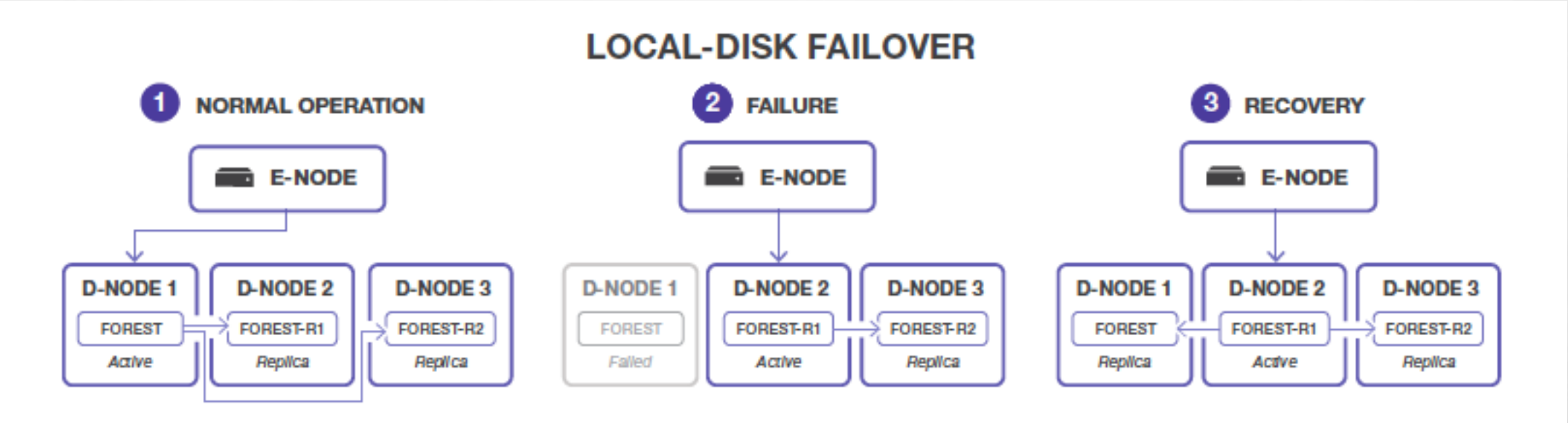
- COMPLEX QUERIES (BASED ON WORDS/PHRASES/DOCUMENT STRUCTURE)

```
for $result in cts:search(
/article[@year = 2010],
cts:and-query((
cts:element-word-query(
xs:QName("description"),
cts:word-query("pet grooming")
), cts:near-query(
(cts:word-query("cat"), cts:word-query("puppy dog")), 10
), cts:not-query(
cts:element-word-query(
xs:QName("keyword"), cts:word-query("fish")
)
)
)))[1 to 10]
return
```

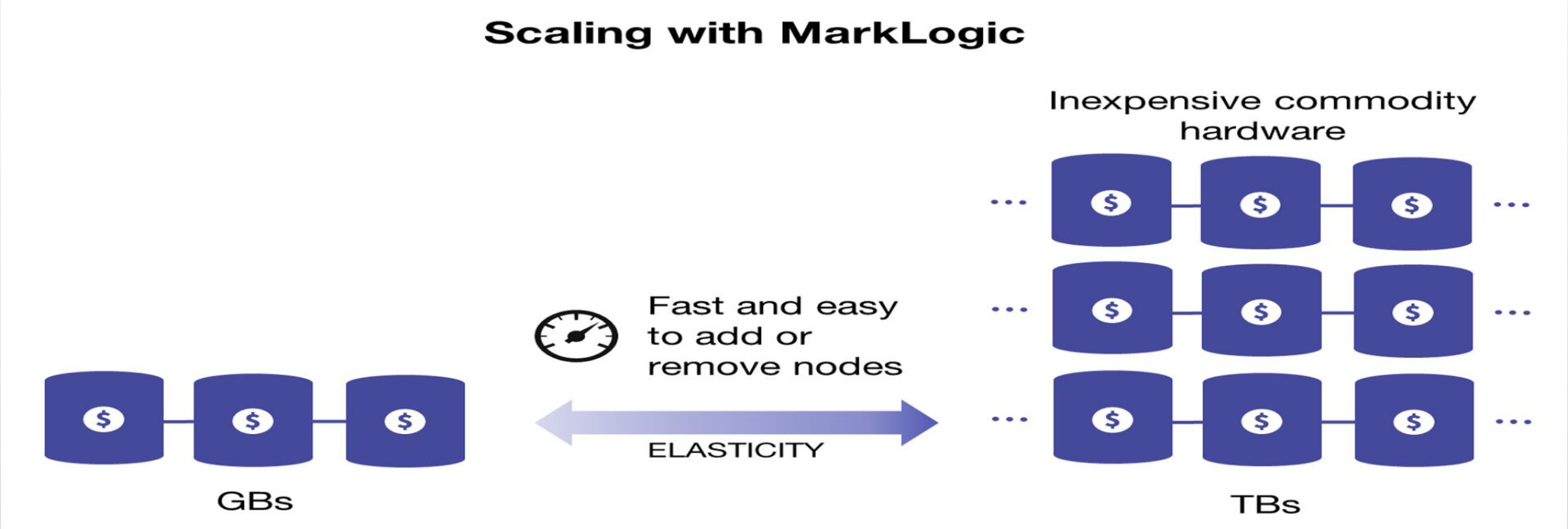
High Availability



High Availability (Contd..)



Scalability



Cost Effective

- FREE DEVELOPERS LICENSE.
- ESSENTIAL ENTERPRISE AT $18K/YEAR.
- ESSENTIAL ENTERPRISE ON AMAZON WEB SERVICES AT 0.99/HR.

DEEP IN FUNCTIONALITY

Basics

- QUERY
  - Standard text search
  - Element-level XML search
  - Native XQuery interface
- MANIPULATE
  - Navigate within content
  - Modify content programmatically
  - Combine content from multiple sources
- RENDER
  - Transform XML schema or DTDs
  - Output to various formats

Find all documents that contain the phrase "high performance"

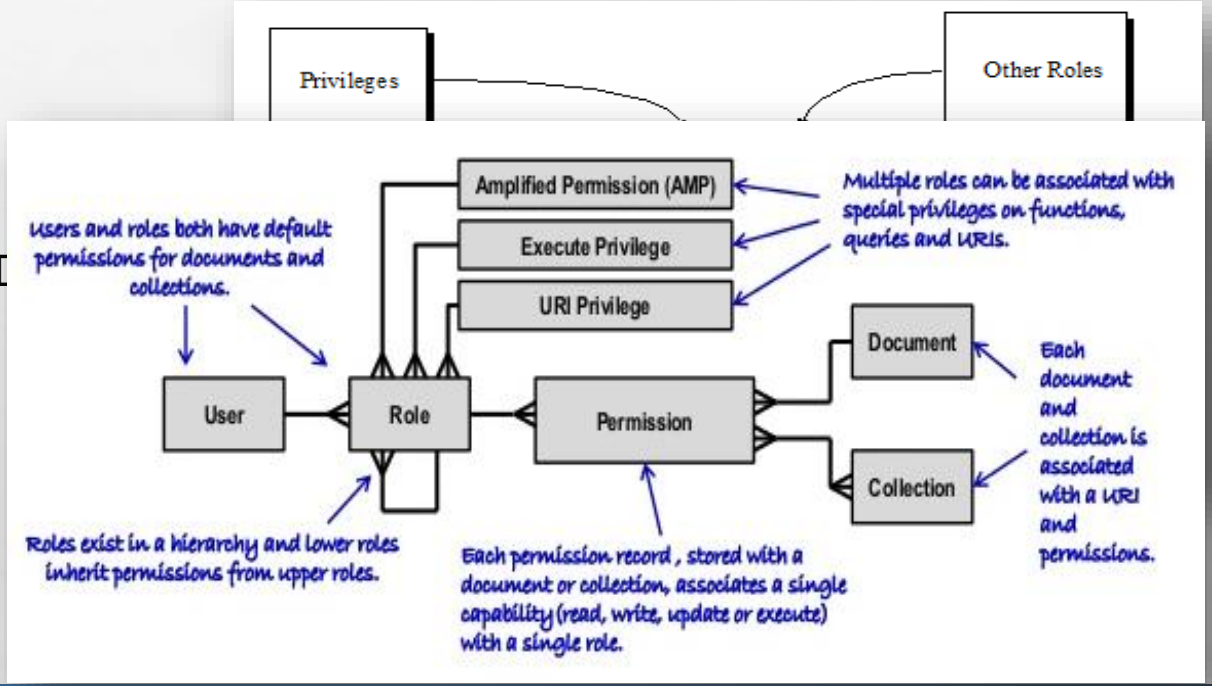
```
<article>
    <title>MarkLogic Server</title>
    <author><first-name>John</first-name><last-name>Kreisa</last-name></author>
    <abstract>
        Where should one put their XML? <company>Mark Logic</company> MarkLogic
        Server. . . .
    </abstract>
    <body>
        <section>
            <section> This high performance engine can </section>
        </section>
        <section> Using an inverted index technique . . . </section>
    </body>
</article>
```

Advanced

- SECURITY
- SEMANTIC INFERENCE OF FACTS
  - USING RULE SETS, AND SPARQL
- GEOSPATIAL
- DATABASE REPLICATION
- CLOUD TEMPLATES
- TIERED STORAGE
- BITEMPORAL

SECURITY

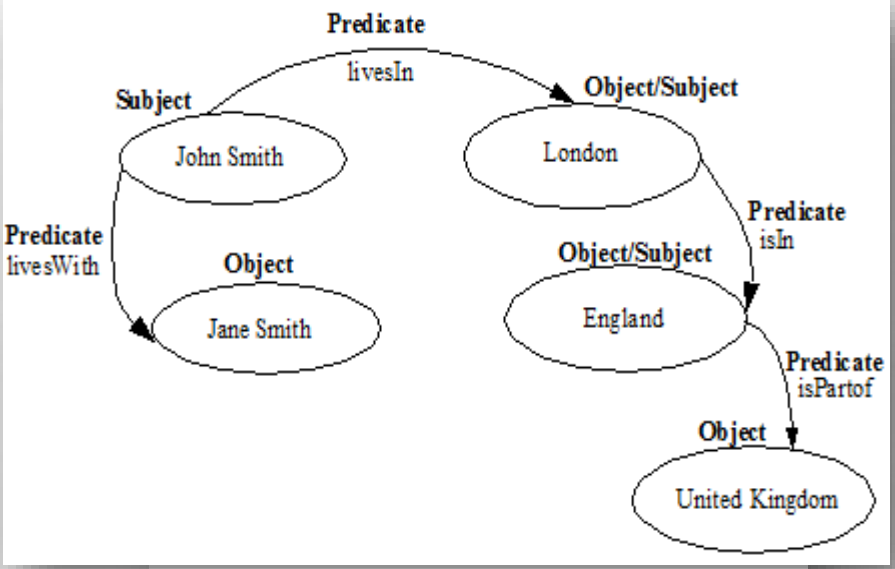
- ROLE-BASED ACCESS CONTROL
  - SECURITY DATABASE, ADMINISTRATION
- AUTHENTICATION
  - INTERNAL OR EXTERNAL USING LDAP AND
- CONFIGURATION MANAGEMENT
- ATOMIC FORESTS



SEMANTICS

- DATA IS STORED AS TRIPLES
  - SUBJECT, PREDICATE, OBJECT
- TRIPLE INDEX USED FOR EFFICIENT QUERY
- GENERATE NEW FACTS AND META DATA
- WORK AS A GRAPH MODEL
- COMBINATION QUERY

e.g. John livesin London
London isin England



GEOSPATIAL

- POINTS AND REGIONS OF INTEREST, INTERSECTING PATHS.
- GEOSPATIAL QUERIES, INDEXES AND SHAPES
  - POINTS, (COMPLEX) POLYGONS, CIRCLES, BOXES
- TEXT (WKT) AND WELL-KNOWN BINARY (WKB)
  - POINT, LINESTRING, TRIANGLE, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRYCOLLECTION
- INTEGRATION WITH LEADING GEOSPATIAL VENDORS
  - ROBUST VISUALIZATION

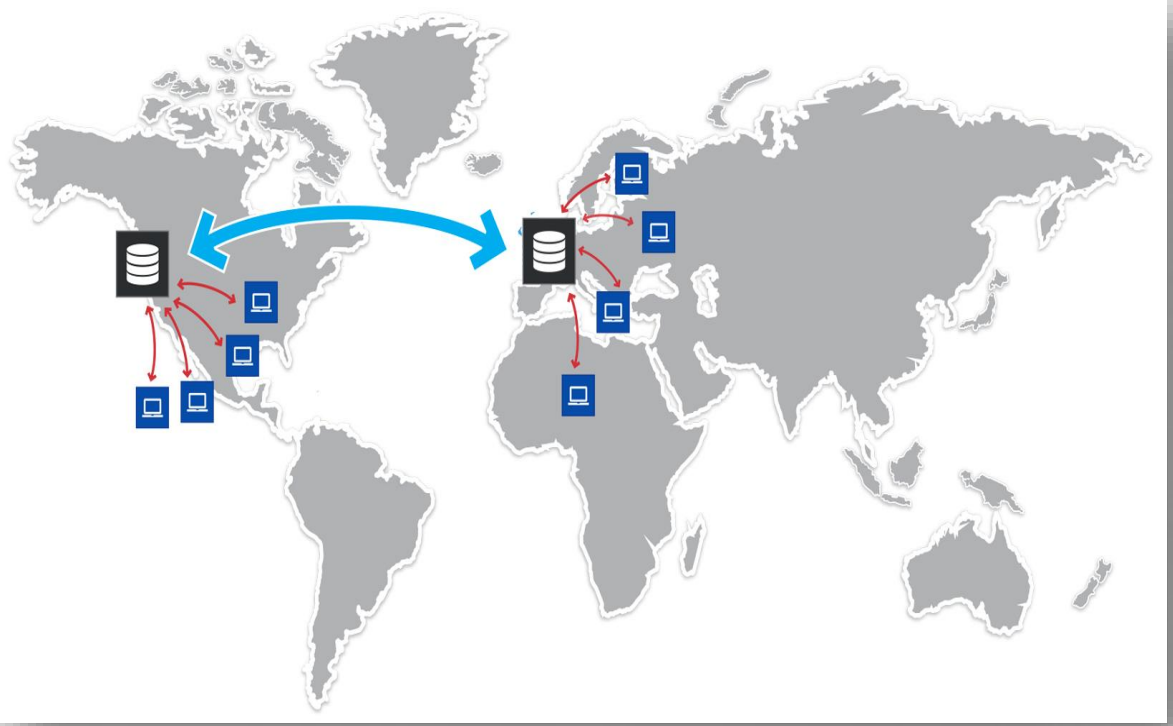
"SHOW ME A LIST OF HOSPITALS THAT FALL WITHIN THE BOUNDARIES OF THIS CERTAIN SET OF COORDINATES"

```
(connection);

var qb = marklogic.queryBuilder;
db.documents.query(
  qb.where(
    qb.geospatial(
      qb.geoProperty(
        qb.property('location'),
        qb.property('coordinates')),
        qb.circle(10, 10.3910, -75.4794)
    )
  )
).result().then(function(response) {
  console.log(response);
});
```

DATABASE REPLICATION

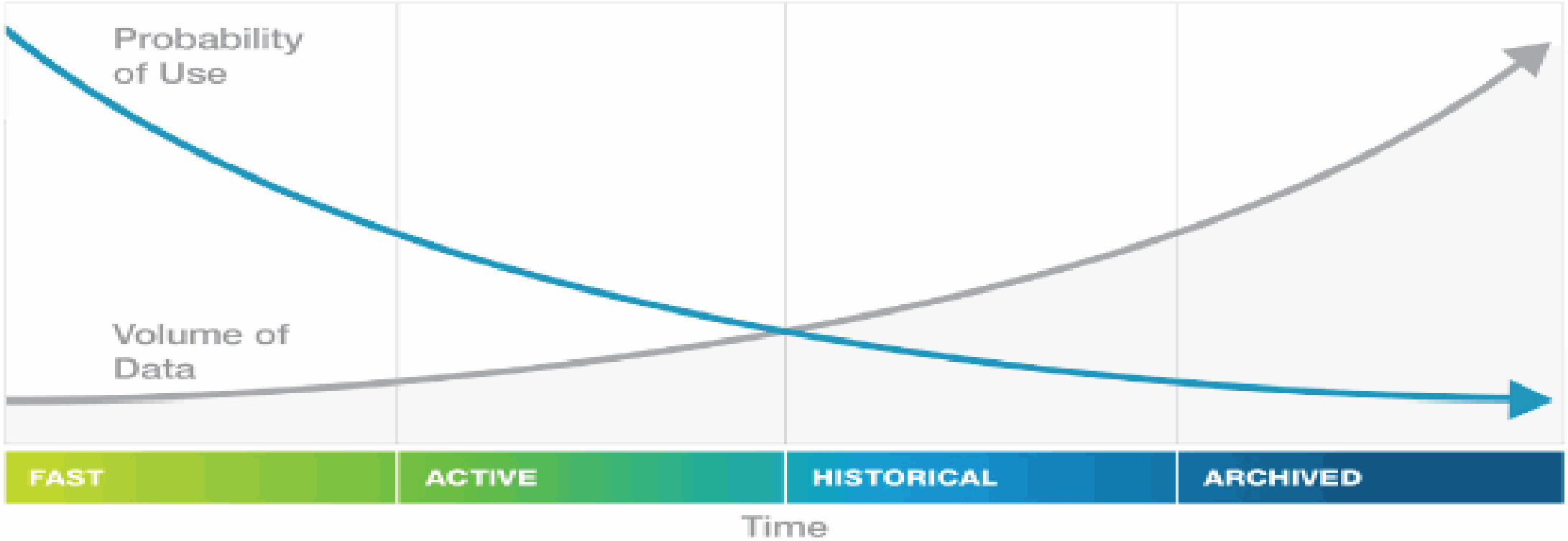
- FLEXIBLE REPLICATION
  - FILTERED AND MANIPULATED BEFORE REPLICATION
  - QUERY-BASED: UPDATES OF QUERY DYNAMICALLY UPDATE REPLICATED DATA.
- GEOGRAPHICALLY DISPERSED CLUSTERS AND MOBILE USERS
- MASTER-SLAVE ARCHITECTURE
- TRANSITIVE REPLICATION
- SAFE UPDATES



CLOUD TEMPLATES

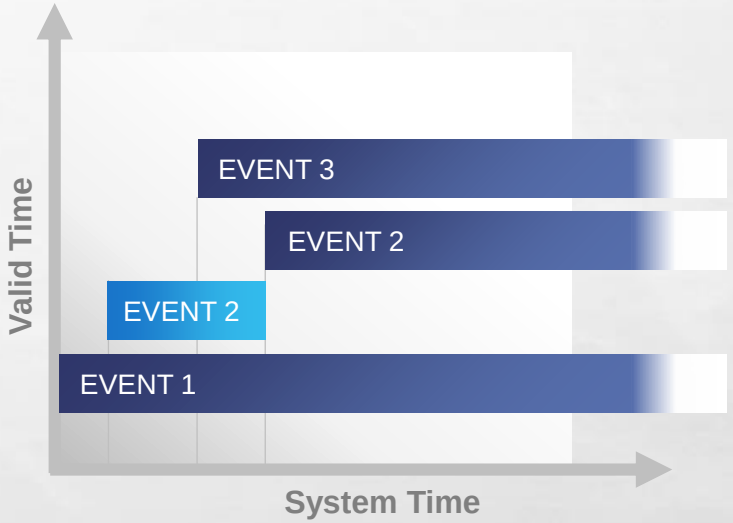
- PRE-PACKAGED CLOUD FORMATION TEMPLATES, AMIs FOR CREATING MANAGED CLUSTERS ON AMAZON EC

TIERED STORAGE



UPDATE

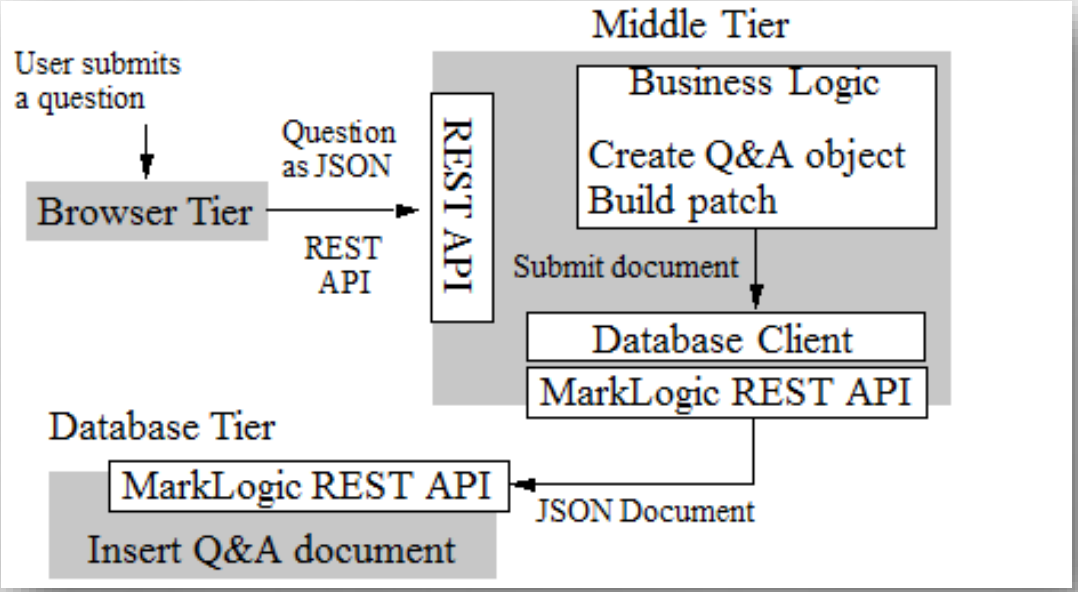
- USING TEMPORAL DATABASE
  - No update! No delete!
  - Only insert and read-at-a-time
  - Every document has two timestamps
    - “created”, “expired”
- HIGH THROUGHPUT
- BITEMPORAL
  - Rewind the information
  - Capture evolving data and business through time

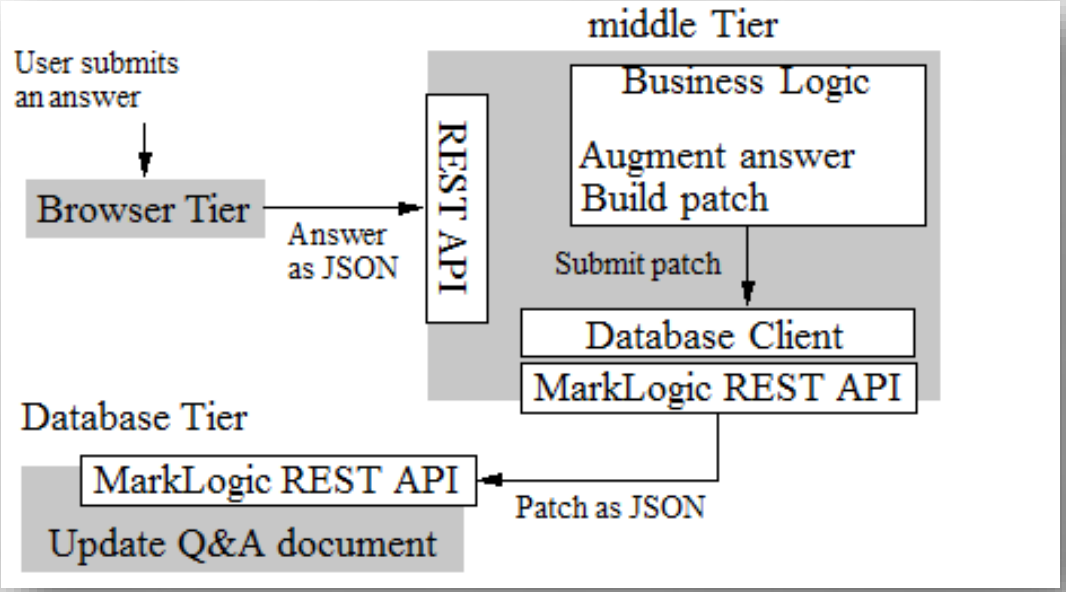


Valid Time – Real-world time, information “as it actually was”

System Time – Time it was recorded to the database

QUERY/ ANSWER PROCESSING





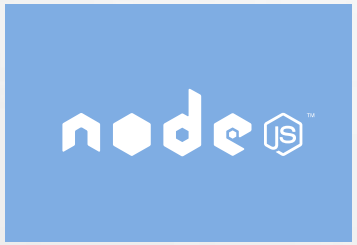
DEVELOPMENT

DEVELOPER TOOLS

MarkLogic™

JSON

Unified indexing and query for today's web and SOA data



Node.js Client API

Enterprise NoSQL database for Node.js

Java Client API

NoSQL agility in a pure Java interface

Server-Side JavaScript

JavaScript runtime *inside* MarkLogic using

Xquery API

Query XML documents using XPath expressions

e.g. Construct a JSON c

```
object-node { "p1" : "v1", "p2"
, "p3" : fn:true(), "p4" : null-no
"v1", "p2" :  [1, 2, 3] , "p3" : tru
```

e.g. Iterate through the results (the raw documents)

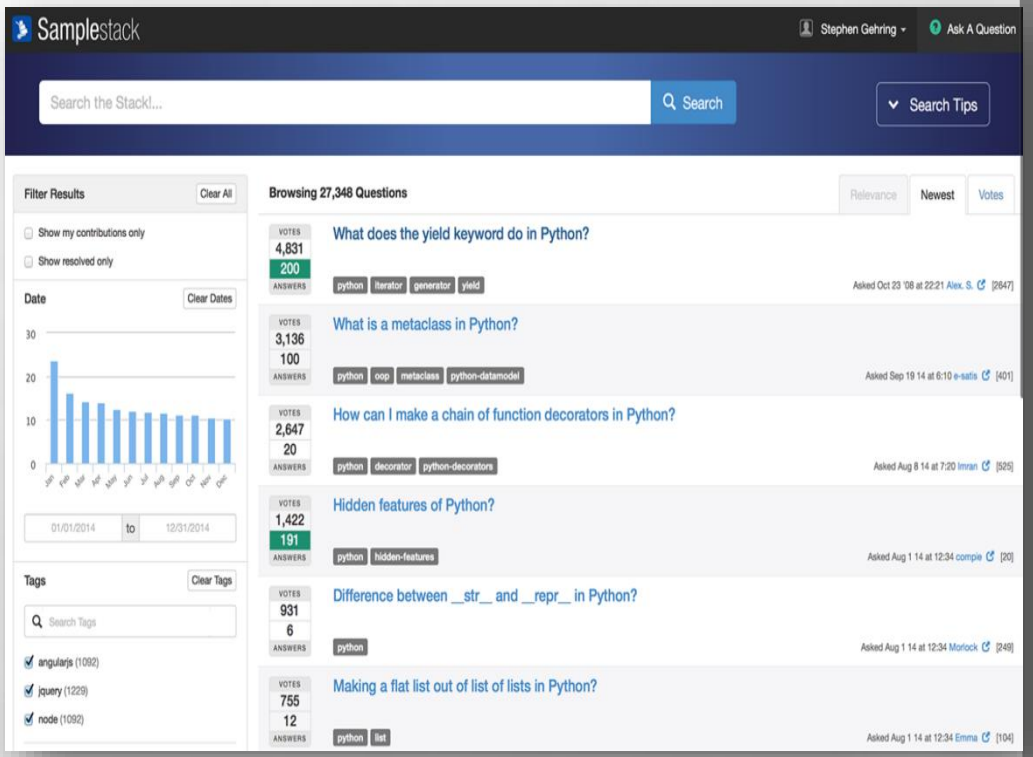
```
DocumentPage page
=client.newDocumentManager().search(query,1);
for (DocumentRecord doc : page) {
   System.out.println(doc.getContent(new
JacksonParserHandle())); }
```

the database every
ollection

```
elete("collection-uri")
```

SAMPLESTACK

- END-TO-END THREE-TIERED APPLICATION IN JAVA AND NODE.JS
  - QUESTION AND ANSWER SITE
- ENCAPSULATES BEST PRACTICES AND INTRODUCES KEY MARKLOGIC CONCEPTS
- USE SAMPLE CODE AS A MODEL FOR BUILDING APPLICATIONS
  - UI , FULL TEXT SEARCH, SEARCH RESULT FILTERING, USERS AND ROLES, FACETS
  - DOCUMENT MODEL, DOCUMENT INSERTION AND UPDATE
  - TRANSACTIONS AND DATA INTEGRITY
- MODERN TECHNOLOGY STACK SHOWS WHERE MARKLOGIC FITS IN YOUR ENVIRONMENT



IMPLEMENTATION CONCEPTS

INDEXING

WORD INDEXING

- INVERTED INDEX
  - WORD -> DOCUMENT RELATION
  - EVERY ENTRY IS CALLED A TERM LIST

- HOW DOES IT SEARCH TWO DIFFERENT WORDS ??
  - USE THE SAME DATA STRUCTURE AND GET THE INTERSECTING DOCUMENTS
- (USE FIGURE 1 AS AN EXAMPLE)

INDEXING PHRASES

- USE THE SAME WORD-INDEXING DATA STRUCTURE
- USE WORD POSITIONING INFORMATION
- ENHANCE THE INVERTED INDEX WITH ADDITIONAL INFORMATION SUCH AS MULTIPLE WORDS

SO WHICH ONE IS USED IN MARKLOGIC??……

- ANYONE OF THESE SETTINGS IS USED AT RUNTIME
- EACH APPROACH HAS ITS OWN ADVANTAGE AND DISADVANTAGE

INDEXING STRUCTURE

- PARENT-CHILD INDEX FOR MAINTAINING HIERARCHICAL STRUCTURE OF XML AND JSON DOCUMENTS
- IT'S SIMILAR TO FAST PHRASE SEARCH BUT USES CONSECUTIVE TAGS

- SEARCHING AN ADVANCE DATABASE BOOK TITLED "INSIDE MARKLOGIC SERVER" USES THE FOLLOWING PARENT-CHILD HIERARCHY

  <BOOK><METADATA>ADVANCE DATABASE</METADATA>

  <TITLE>INSIDE MARKLOGIC SERVER</TITLE>……<BOOK>

METADATA INDEXING AND RELEVANCE

- PARENT-CHILD INDEX FOR MAINTAINING HIERARCHICAL STRUCTURE OF XML AND JSON DOCUMENTS
- SHORT DOCUMENTS WITH EQUAL NUMBER OF HITS OR DOCUMENTS CONTAINING RARE HIT WORDS ARE PRIORITIZED
- TERM LISTS ARE USED TO INDEX DIRECTORIES, COLLECTIONS AND SECURITY RULES -> UNIVERSAL INDEX

RELEVANCE = LOG(TERM FREQUENCY) * (INVERSE DOCUMENT FREQUENCY)

POINT IN TIME QUERY

- IN DATABASE EACH QUERY IS REGISTERED WITH A TIME STAMP WHEN THE QUERY STARTS
- AT PRESENT TIME, WE CAN QUERY THE DATABASE AS IT WAS AT AN ARBITRARY TIME IN THE PAST
- USEFUL FOR LOCALLY TESTING A FEATURE (DATABASE ROLL BACK)

CLUSTERING

INCREMENTALLY ADD NEW SERVERS AS PER REQUIREMENT AND REDUCE FAIL OVERS, OPTIMIZE THE CACHE AND USE IT FOR DIFFERENT FUNCTIONALITIES

MARKLOGIC IN CLUSTER CAN OPERATE IN ANY OF THE TWO ROLES – **1) E –NODE, 2) D-NODE**

- **E-NODE** -> HANDLE THE REQUESTS
- **D-NODE** > HANDLE THE DATA INDEXING

ADVANCE TEXT HANDLING

- TEXT SENSITIVITY – SUCH AS CASE-SENSITIVE, E.G.- ‘POLISH’ AND ‘POLISH’
- STEMMED INDEXED SEARCH -> SEARCH FOR ‘RUN’, MARKLOGIC RETURNS RESULTS WITH KEYWORD ‘RUNNING’, ‘RUN’, ‘RUNS’, ‘RAN’
- FROM MARKLOGIC 8.0 STEMMED INDEXING IS BY DEFAULT ENABLED
- WILDCARDED SEARCH QUERIES, SUCH AS MARK*, MAR*LOG*

GEO SPATIAL INDEX

- QUERY TERMS BASED ON GEOSPATIAL INDEXES PRESENT IN THE DOCUMENT
- MATCH BY EXACT LATITUDE LONGITUDE OR AGAINST AN AD HOC POLYGON OF VERTICES, WHICH CAN BE USED TO DRAW CITY BOUNDARIES
- SUPPORTS POLAR REGION CO-ORDINATES, AND ANTI-MERIDIAN LONGITUDE BOUNDARY NEAR THE INTERNATIONAL DATE LINE AND CONSIDERS THE ELLIPSOID SHAPE OF EARTH
- POINT QUERIES ARE RESOLVED BY RANGE INDEXES AND POLYGON QUERIES ARE RESOLVED BY USING HIGH SPEED COMPARATORS TO DETERMINE POINT POSITION
- SPECIAL TRIGONOMETRY OPERATIONS TO RESOLVE SEARCHES RELATED TO POLAR CO-ORDINATES

SEMANTICS

MARK LIVES IN GAINESVILLE

- SUBJECT -> MARK, PREDICATE -> LIVES IN, OBJECT -> GAINESVILLE
- USES TRIPLE INDEX FOR FASTER RETRIEVAL AND ALL THREE PERMUTATIONS ARE STORED IN SORTED ORDER
- FOR SMALLER SPACE UTILIZATION, THE TRIPLE VALUES ARE ASSOCIATED WITH AN INTEGER ID AND EACH QUERY RESULT IS MAPPED FROM THE ID TO TRIPLE VALUE
- USES THE TRIPLE TYPE INDEX TO STORE THE DATA TYPE OF THE TRIPLES

- USE FIGURE 18

OPTIMISTIC LOCK

- DOES NOT HOLD LOCK ON THE DOCUMENT IN BETWEEN READ AND UPDATE OPERATION
- CONDITIONAL UPDATE USING VERSION ID
- IT'S CONTENT VERSIONING NOT DOCUMENT VERSIONING

$ curl --anyauth --user user:password -i -X HEAD -H "Accept: application/xml" http://localhost:8000/LATEST/documents?uri=/docs/sample_lock.xml

HTTP/1.1 200 Document Retrieved
Content-type: application/xml
ETag: "2547689390376812 40"
Server: MarkLogic
Connection: close

*$ curl --anyauth --user user:password -i -X PUT -d"**<modified-data/>**" -H "Content-type: application/xml" -H "**If-Match: 2547689390376812 40**" http://localhost:8000/LATEST/documents?uri=/docs/sample_lock.xml*

PROGRAMMING
WITH REST API

REST API INSERT (PUT / POST) REQUEST

SAMPLE_XMLFILE.XML

<ROOT>HELLO WORLD </ROOT>

SAMPLE_JSONFILE.JSON

<TITLE> HELLO JSON </TITLE>

*$ CURL --ANYAUTH --USER USER:PASSWORD -X POST -D@'./**SAMPLE_XMLFILE.XM**L' -H "CONTENT-TYPE: APPLICATION/XML" 'HTTP://LOCALHOST:8000/LATEST/DOCUMENTS?URI=**/XML/FIRST_FILE.XML**'*

*$ CURL --ANYAUTH --USER USER:PASSWORD -X POST -D@'./**SAMPLE_JSONFILE.JSON**' -H "CONTENT-TYPE: APPLICATION/JSON" 'HTTP://LOCALHOST:8000/LATEST/DOCUMENTS?URI=**/JSON/FIRST_FILE.JSON**'*

REST API INSERT/UPDATE CONTENT AND METADATA

*CURL -X PUT -T ./MARKLOGIC_ARCHITECTURE.JPG --ANYAUTH --USER USER:PASSWORD -H "CONTENT-TYPE: IMAGE/JPEG" 'HTTP://LOCALHOST:8000/LATEST/DOCUMENTS?**URI=/IMAGES/MARKLOGIC_ARCHITECTURE.JPG&COLLECTION=NOSQL_DB_ARCHITECTURE&PROP:SPECIES=“MARKLOGIC**'"*

REST API DATA RETRIEVAL (GET REQUEST)

DOCUMENT

*HTTP://HOST:PORT/VERSION/DOCUMENTS?**URI=DOC_URI***

METADATA

*HTTP://HOST:PORT/VERSION/DOCUMENTS?URI=DOC_URI&**CATEGORY=METADATA_CATEGORY***

CONTENT AND METADATA

*HTTP://HOST:PORT/VERSION/DOCUMENTS?**URI=DOC_URI&CATEGORY=CONTENT&CATEGORY=METADATA_CATEGORY***

REST API SEARCHING AND STREAMING

SEARCHING

*$ CURL --ANYAUTH --USER USER:PASSWORD 'HTTP://LOCALHOST:8000/LATEST/SEARCH?**Q=HELLO**'*

STREAMING

NO NEED TO LOAD THE ENTIRE CONTENT INTO MEMORY

*CURL --ANYAUTH --USER USER:PASSWORD -I -O SAMPLE.JPG -X GET -H "ACCEPT: APPLICATION/JPG" **-R "0-511999"** HTTP://LOCALHOST:8000/LATEST/DOCUMENTS?URI=/PICTURES/TEST.JPG*

REST API PATCH UPDATE

MAY BE ADDED

REST API DELETE

BLANK DIRECTORY OR COLLECTION NAME DELETES THE ENTIRE DATABASE

SINGLE DOCUMENT

*HTTP://HOST:PORT/VERSION/DOCUMENTS?**URI=DOCUMENT_URI***

MULTIPLE DOCUMENTS

*HTTP://HOST:PORT/VERSION/SEARCH?**COLLECTION=COLLECTION_NAME***

APPLICATION

Supported Languages

MarkLogic™

C/C++

python™

Java™

C#

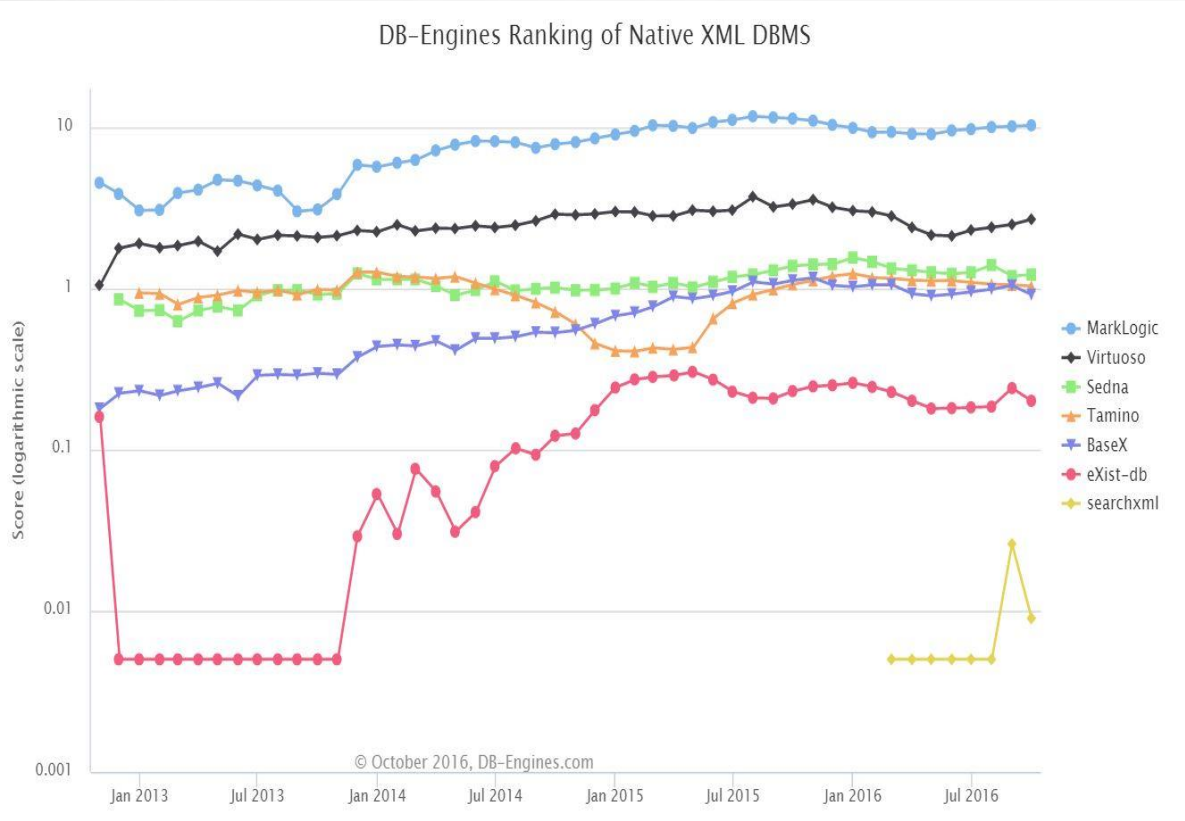
JS JavaScript

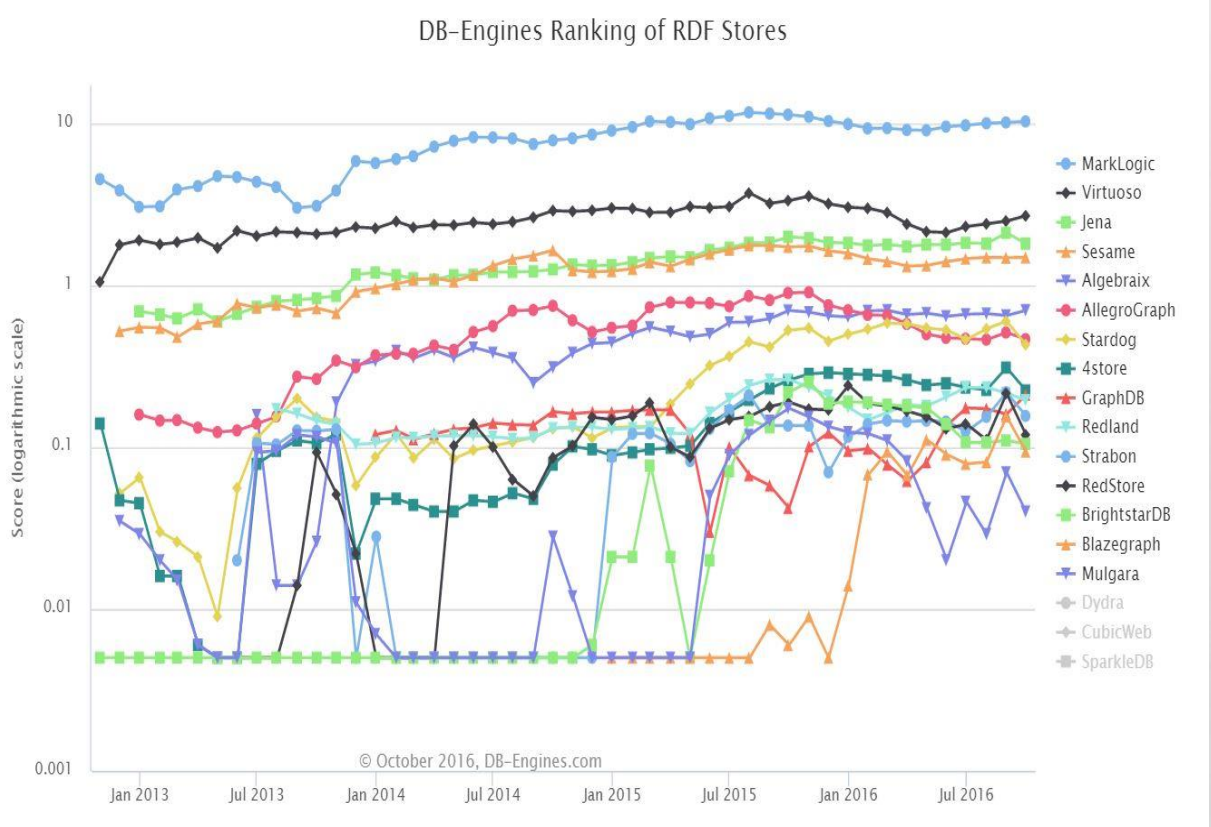
RUBY Programming

php

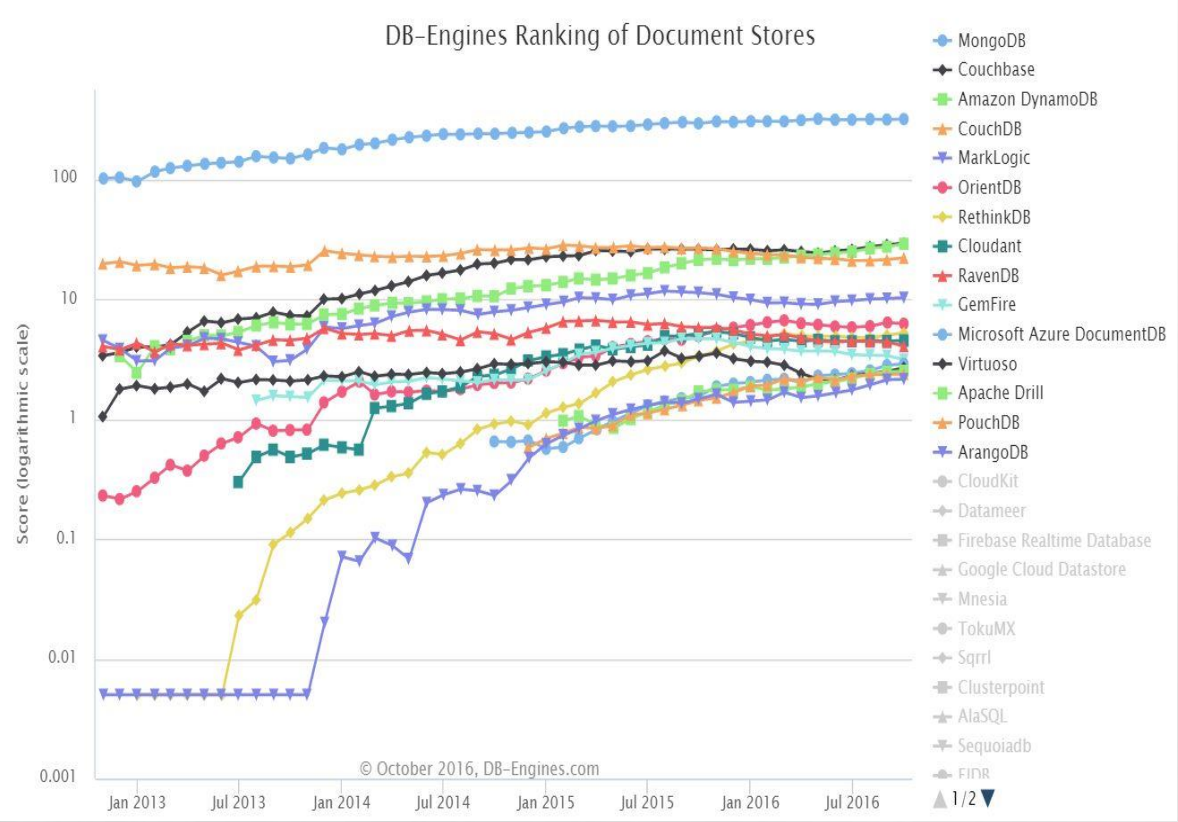
Perl

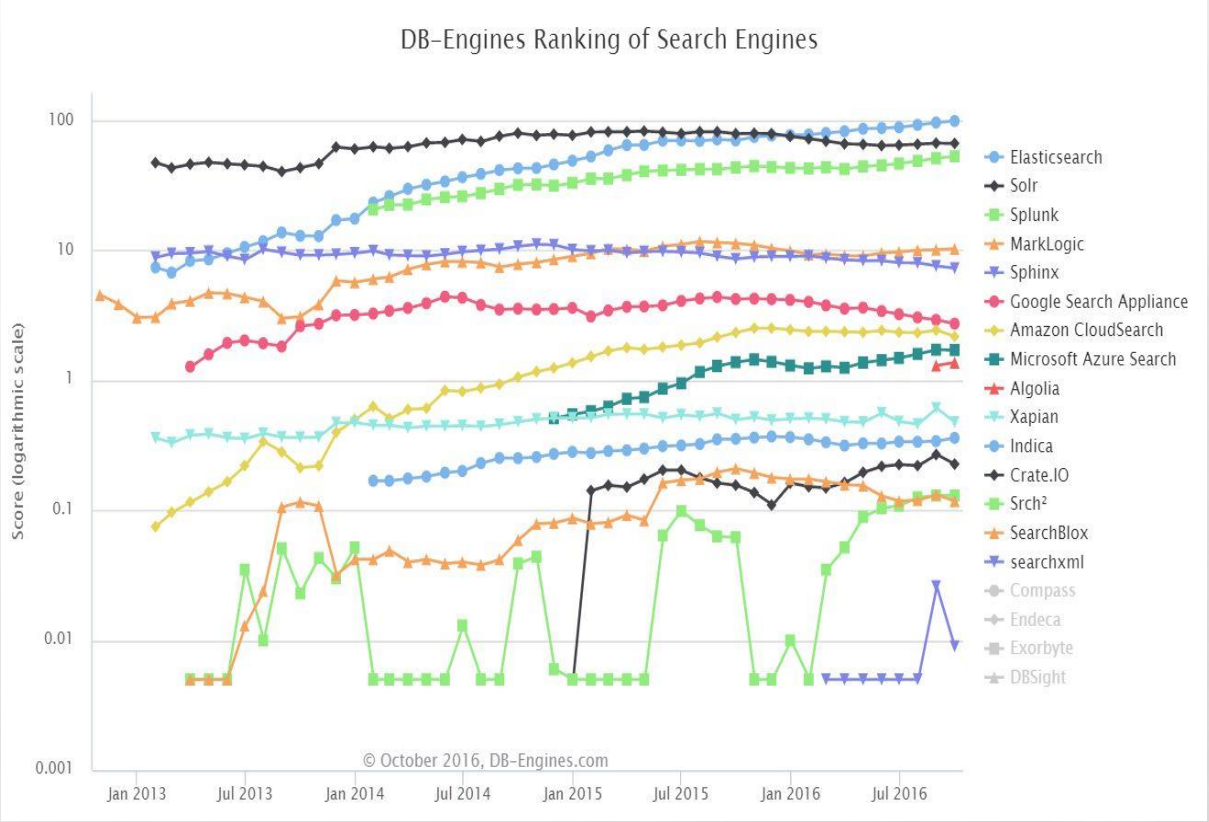
Trend Charts





Trend Charts (Contd..)





Project - HealthCare.gov

MarkLogic™

- FASTER TIME TO PRODUCTION: 18 MONTHS, WITHIN NEXT 6 MONTHS – 5500+ TRANSACTIONS PER SECOND
- SCALABILITY:
  160,000 CONCURRENT USERS,
  99.9% AVAILABILITY,
  QUERY RESPONSE TIME <0.1 SECOND
- SCHEMA-AGNOSTIC DATA MODEL: SEAMLESS ONLINE SHOPPING FOR USERS
- ENTERPRISE GRADE DATABASE PLATFORM: HIGH AVAILABILITY AND SECURITY

Project – BBC (London Olympics)

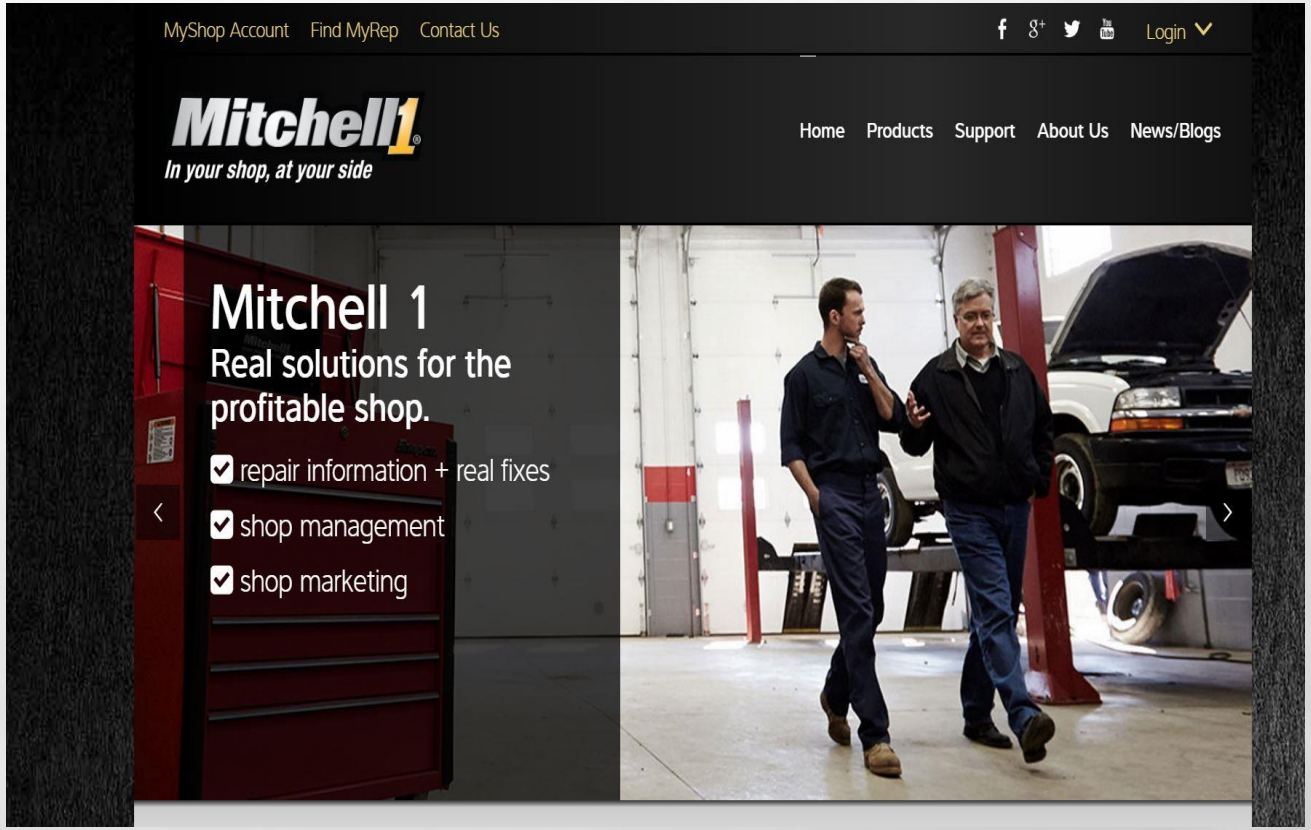
- DYNAMIC UPDATE ON EACH OF 10,000 ATHLETE PAGES
- OLYMPIC VIDEO CONTENT REQUESTS: 106 MILLIONS
- 2.8 PETABYTES OF DATA ON BUSIEST DAY
- EASY LOADING OF DATA: VIDEOS, ARTICLES, TWEETS, IMAGES, STATISTICS

Dynamic Content Delivery

During live-streaming users could choose different views to appear at the bottom of the application, called iPlayer. Here, athlete information populates the screen.

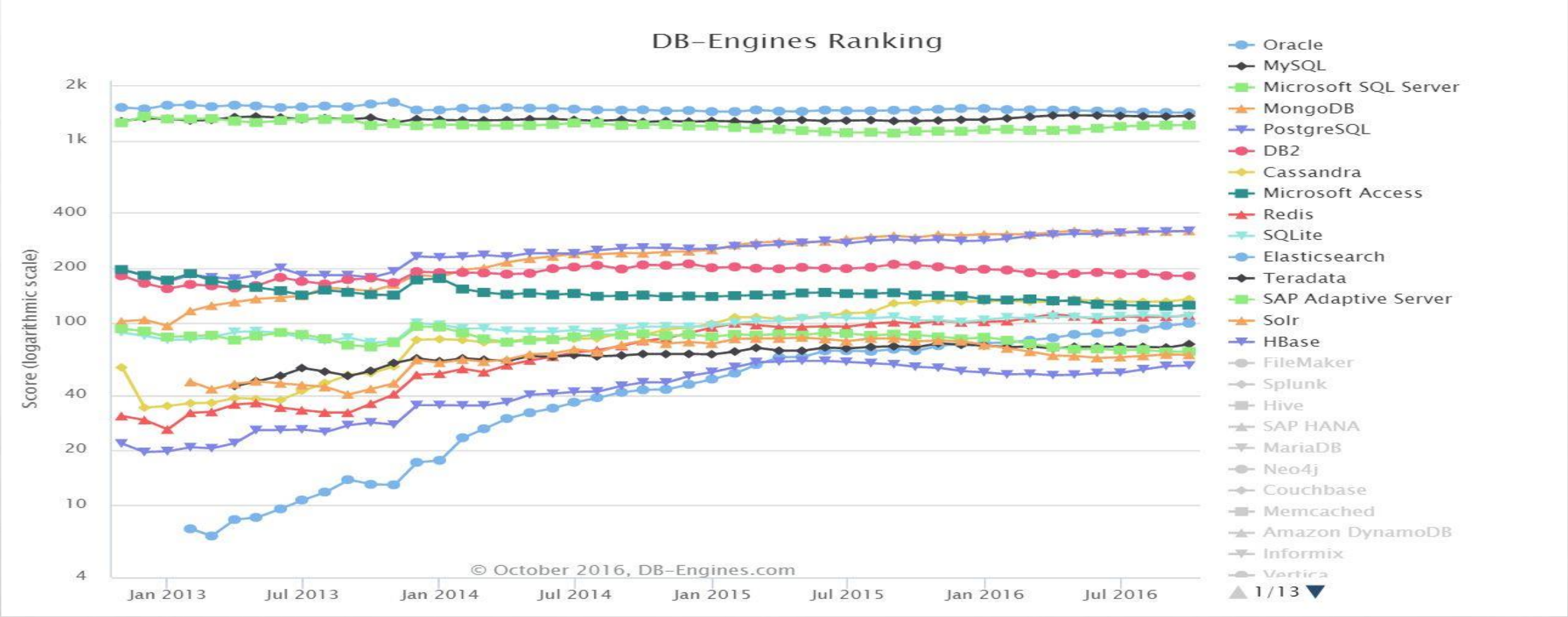
Project – Mitchell1

- COMPLEX DATA MANAGEMENT AND INTEGRATION
- ENHANCEMENTS EVERY 2 WEEKS COMPARED TO ONCE OR TWICE PER YEAR
- INCREASE IN REVENUE WITH BETTER CUSTOMER EXPERIENCE
- COST REDUCTION WITH LESS MANUAL DATA TRANSFER



And Many More..

Why Not MarkLogic?



THANK YOU

GROUP 11

AVIRUP CHAKRABORTY

RASHA ELHESHA

SAPTARSHI CHAKRABORTY

DEBARSHI MITRA