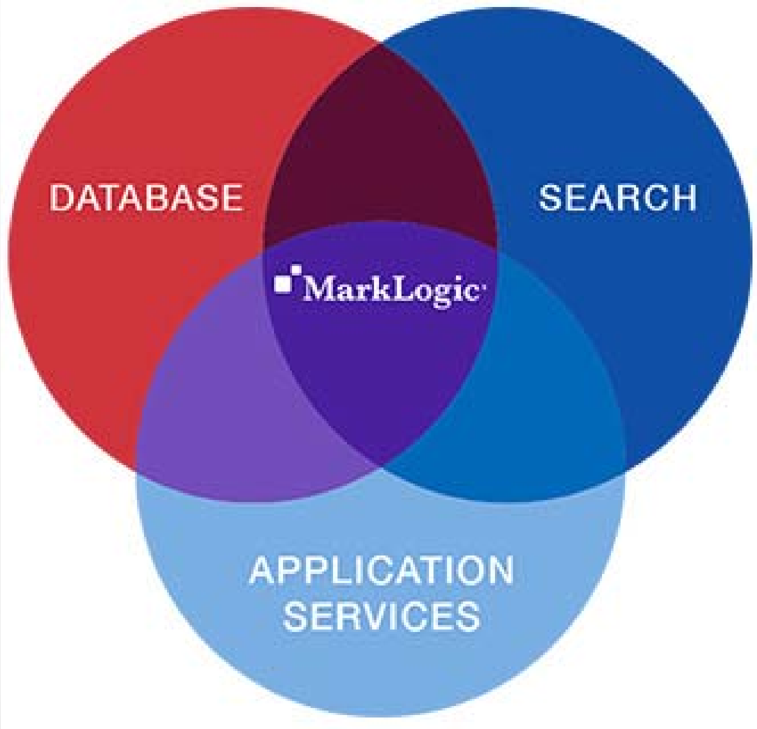
MarkLogic

BUILD. ITERATE. INNOVATE. FASTER

First look at MarkLogic

- EASY TO GET DATA IN
- EASY TO GET DATA OUT
- ENTERPRISE READY
- FLEXIBLE DEPLOYMENT

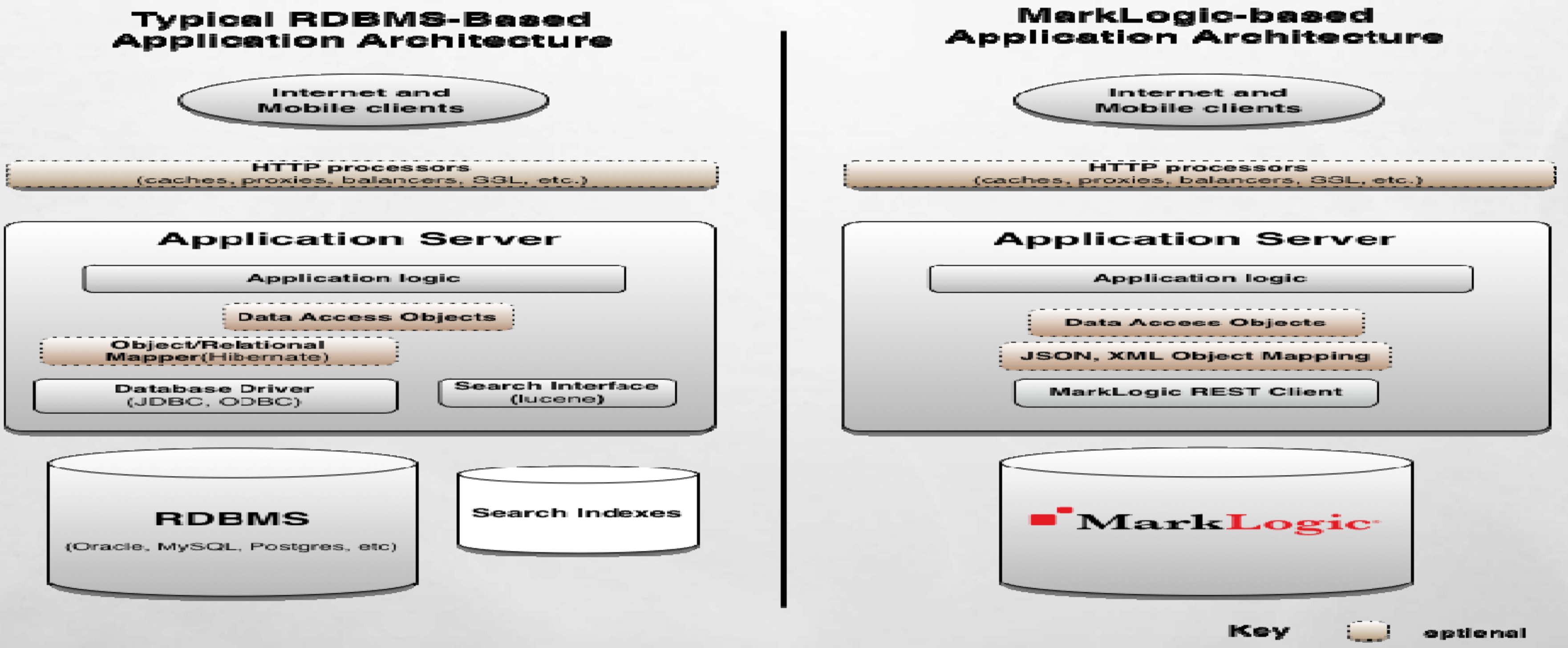


Brief history

- 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™



Typical RDBMS-Based Application Architecture

Internet and Mobile clients

HTTP processors (caches, proxies, balancers, SSL, etc.)

Application Server

Application logic

Data Access Objects

Object/Relational Mapper(Hibernate)

Database Driver (JDBC, ODBC)

Search Interface (lucene)

RDBMS (Oracle, MySQL, Postgres, etc)

Search Indexes

MarkLogic-based Application Architecture

Internet and Mobile clients

HTTP processors (caches, proxies, balancers, SSL, etc.)

Application Server

Application logic

Data Access Objects

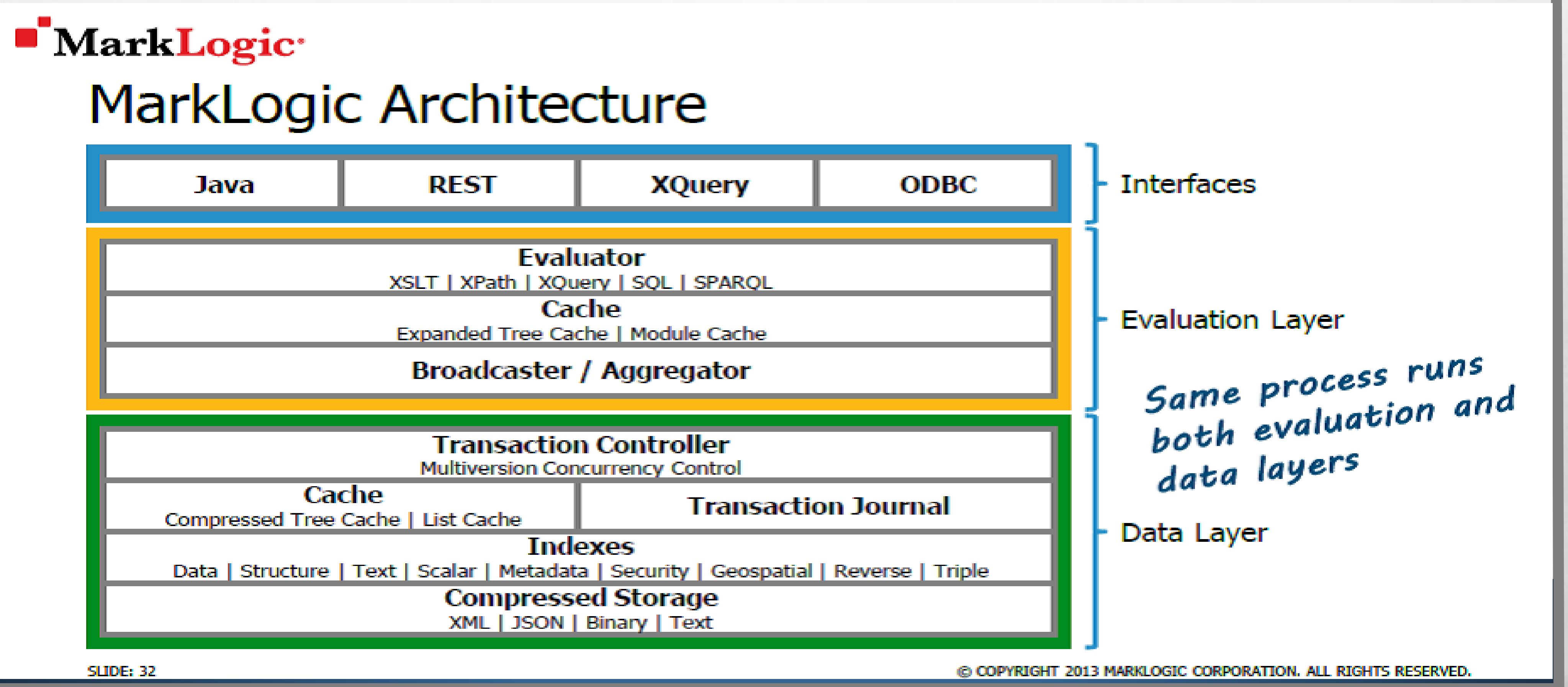
JSON, XML Object Mapping

MarkLogic REST Client

MarkLogic

Key ▨ optional

System architecture



MarkLogic Architecture

Java | REST | XQuery | ODBC

Interfaces

Evaluator
XSLT | XPath | XQuery | SQL | SPARQL

Cache
Expanded Tree Cache | Module Cache

Broadcaster / Aggregator

Evaluation Layer

Transaction Controller
Multiversion Concurrency Control

Cache
Compressed Tree Cache | List Cache

Transaction Journal

Indexes
Data | Structure | Text | Scalar | Metadata | Security | Geospatial | Reverse | Triple

Compressed Storage
XML | JSON | Binary | Text

Data Layer

Same process runs both evaluation and data layers

SLIDE: 32

MarkLogic™

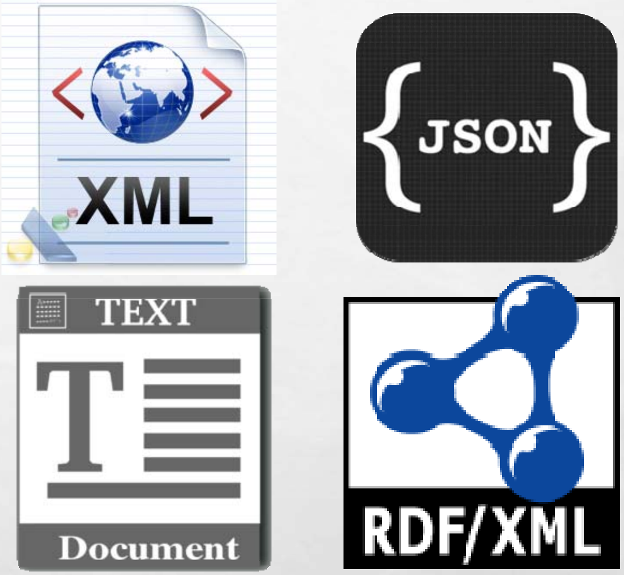
Key features

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

MarkLogic™

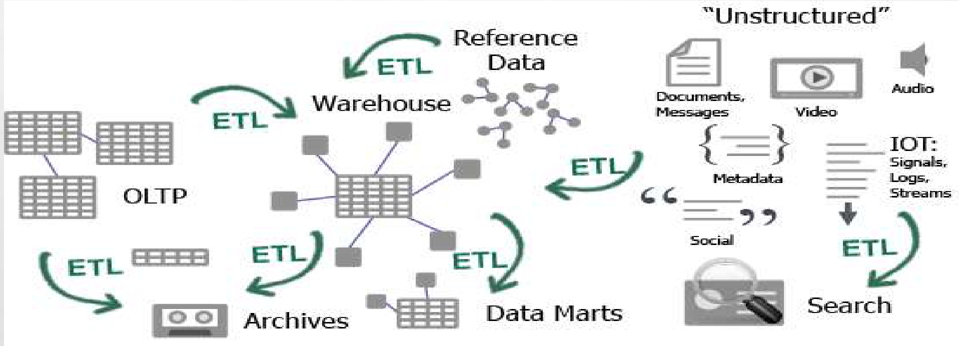
Document centric

- **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



Search oriented

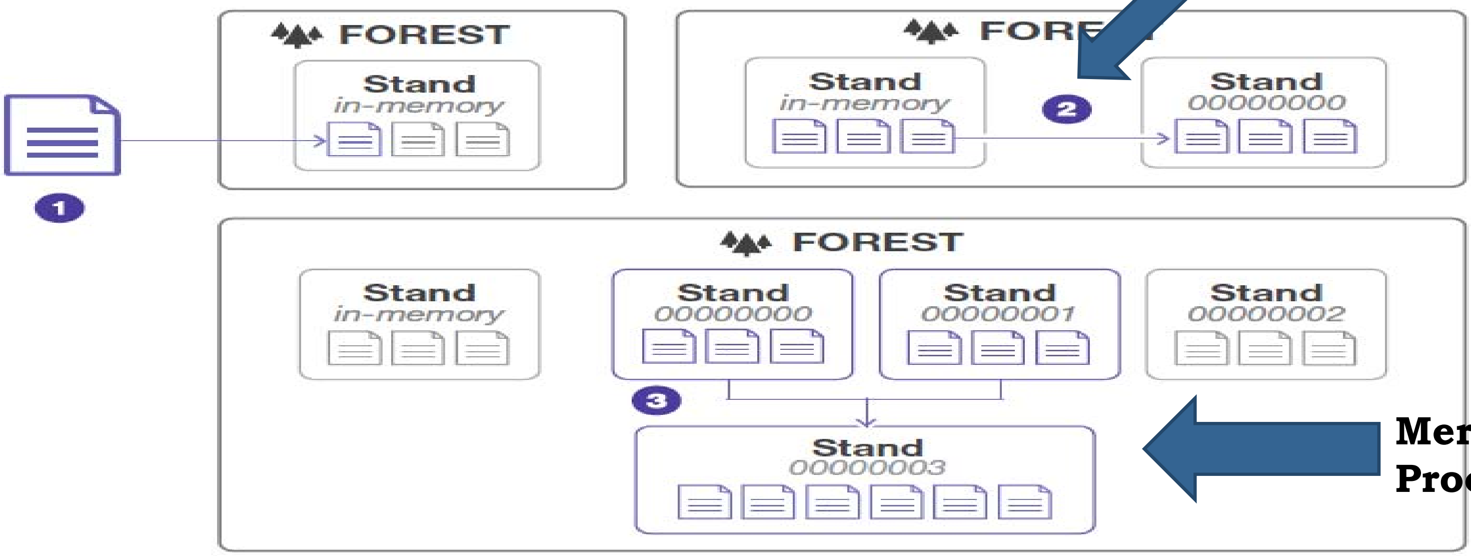
- SIMPLE QUERIES (URI/KEY-VALUE LOOK UP)

```
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
```

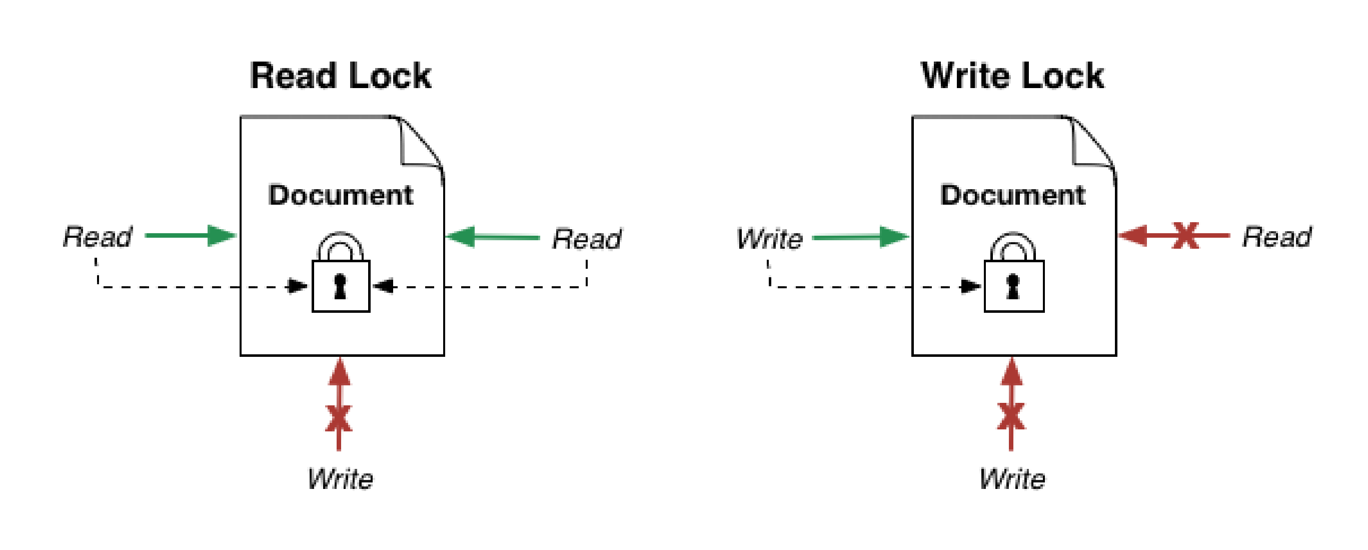
Data management



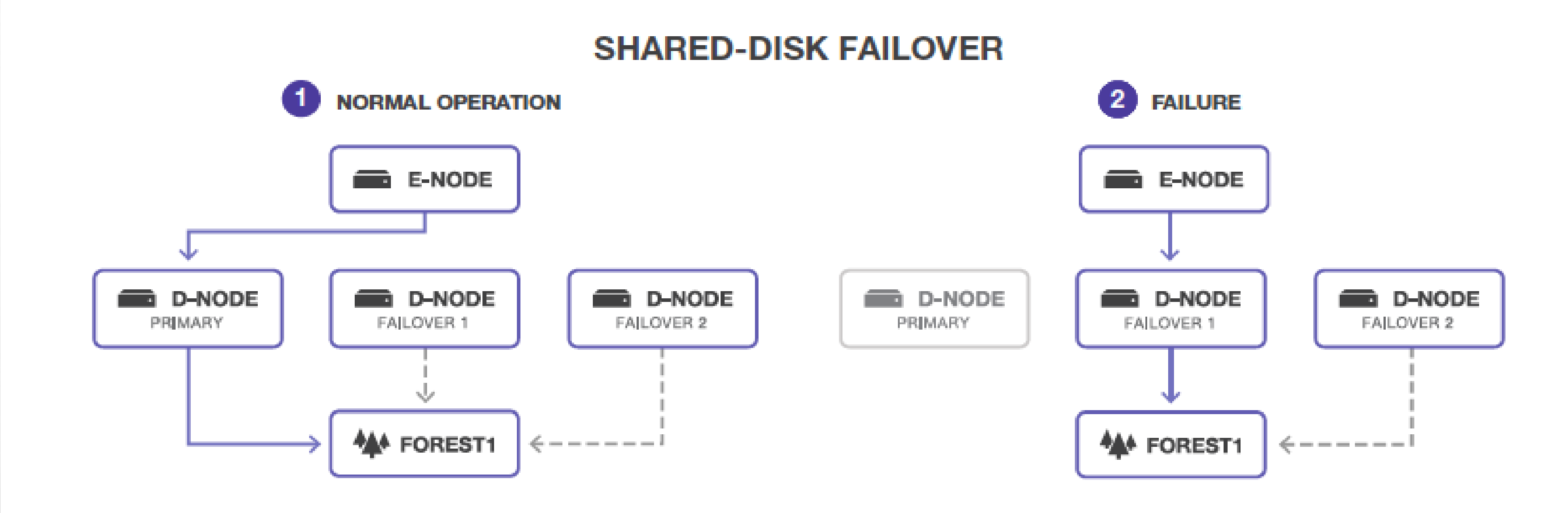
In Memory to
On-Disk Stand

Merger
Process

Transactional (ACID)

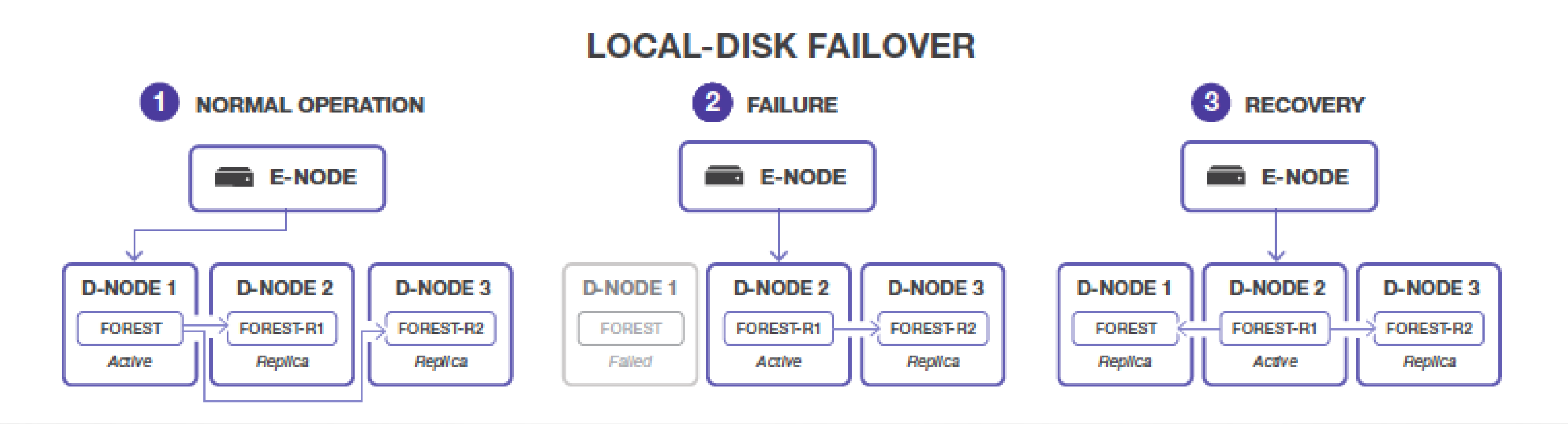


High availability



MarkLogic™

High availability (cont.)



LOCAL-DISK FAILOVER

1 NORMAL OPERATION

E-NODE

D-NODE 1: FOREST, Active
D-NODE 2: FOREST-R1, Replica
D-NODE 3: FOREST-R2, Replica

2 FAILURE

E-NODE

D-NODE 1: FOREST, Failed
D-NODE 2: FOREST-R1, Active
D-NODE 3: FOREST-R2, Replica

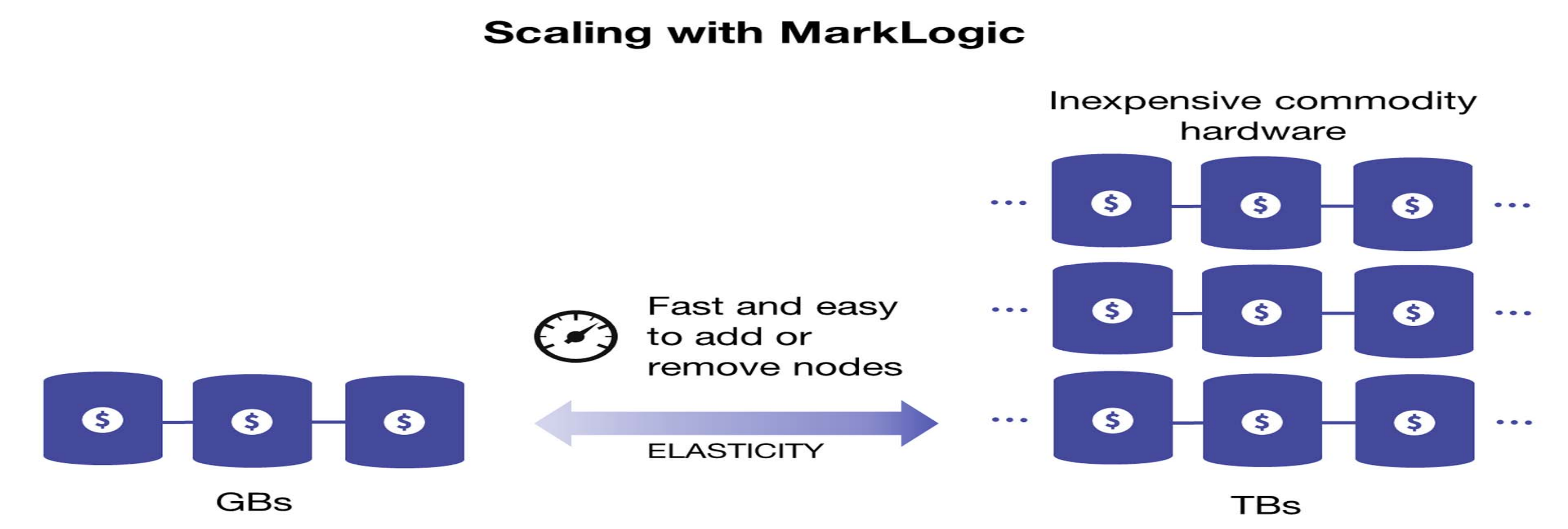
3 RECOVERY

E-NODE

D-NODE 1: FOREST, Replica
D-NODE 2: FOREST-R1, Active
D-NODE 3: FOREST-R2, Replica

MarkLogic™

Scalability



Scaling with MarkLogic

Inexpensive commodity hardware

Fast and easy to add or remove nodes

ELASTICITY

GBs

TBs

MarkLogic™

Pricing & licensing $

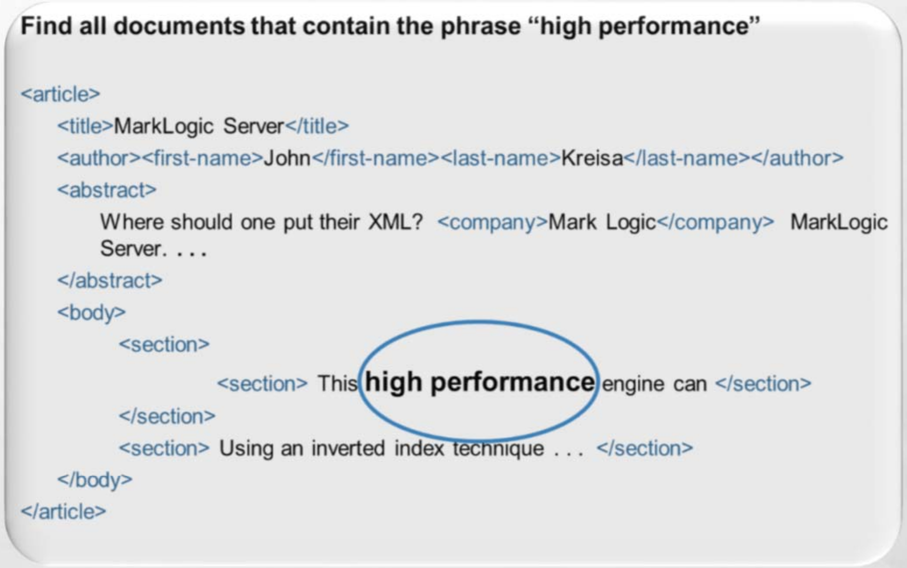
- FREE DEVELOPERS LICENSE.
- ESSENTIAL ENTERPRISE AT $18K/YEAR.
- ESSENTIAL ENTERPRISE ON AMAZON WEB SERVICES AT $0.99/HR.

DEEP IN FUNCTIONALITY

MarkLogic™

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>
```

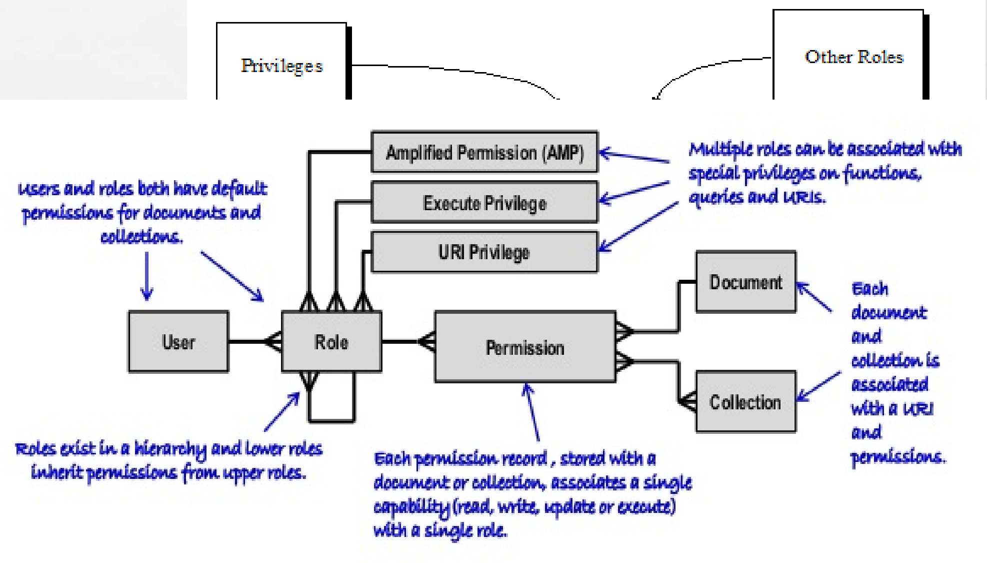
Advanced

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

MarkLogic™

Security

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



Users and roles both have default permissions for documents and collections.

Multiple roles can be associated with special privileges on functions, queries and URIs.

Each document and collection is associated with a URI and permissions.

Roles exist in a hierarchy and lower roles inherit permissions from upper roles.

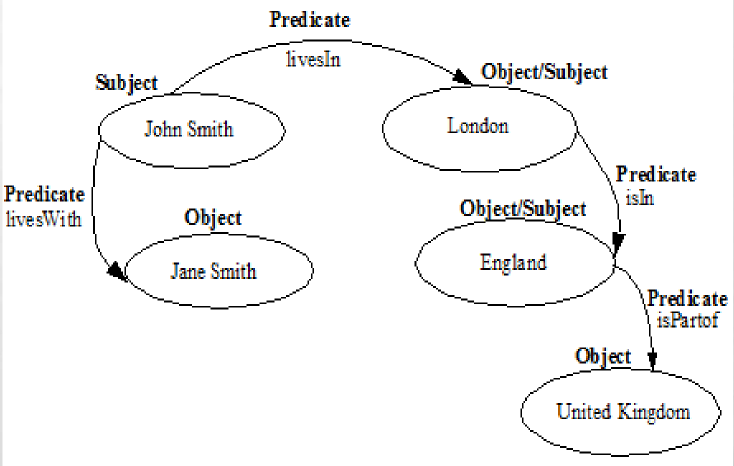
Each permission record, stored with a document or collection, associates a single capability (read, write, update or execute) with a single role.

MarkLogic™

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

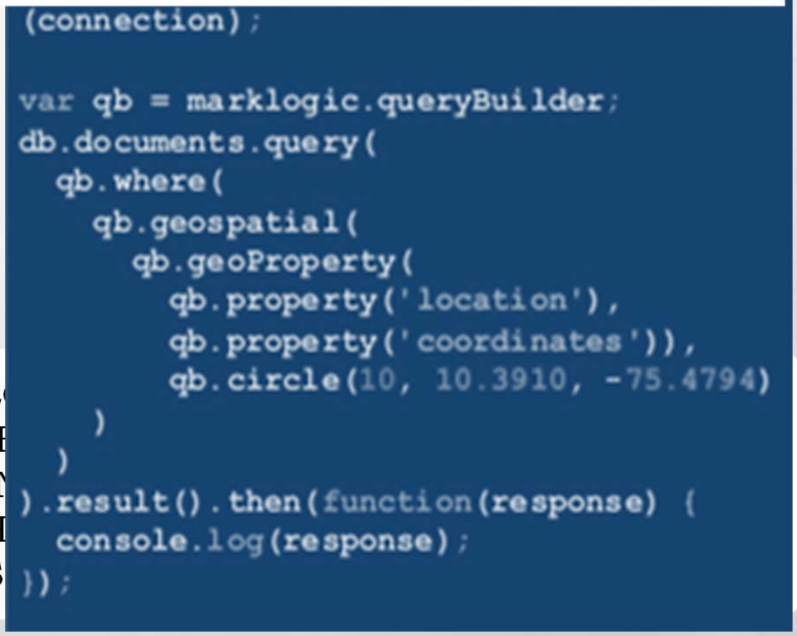


MarkLogic™

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”



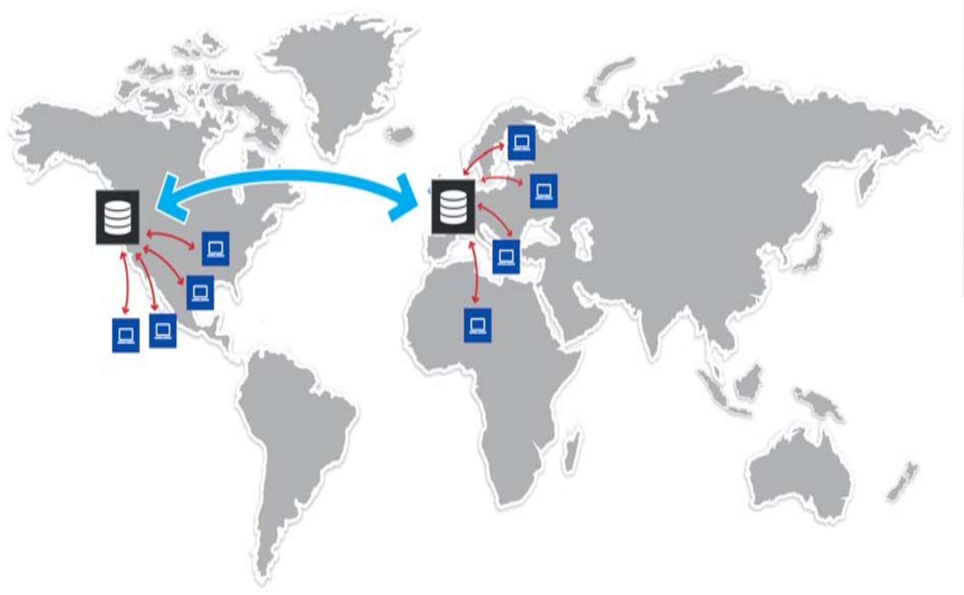
```
(connection);

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);
});
```

L
“F
IN
H
S

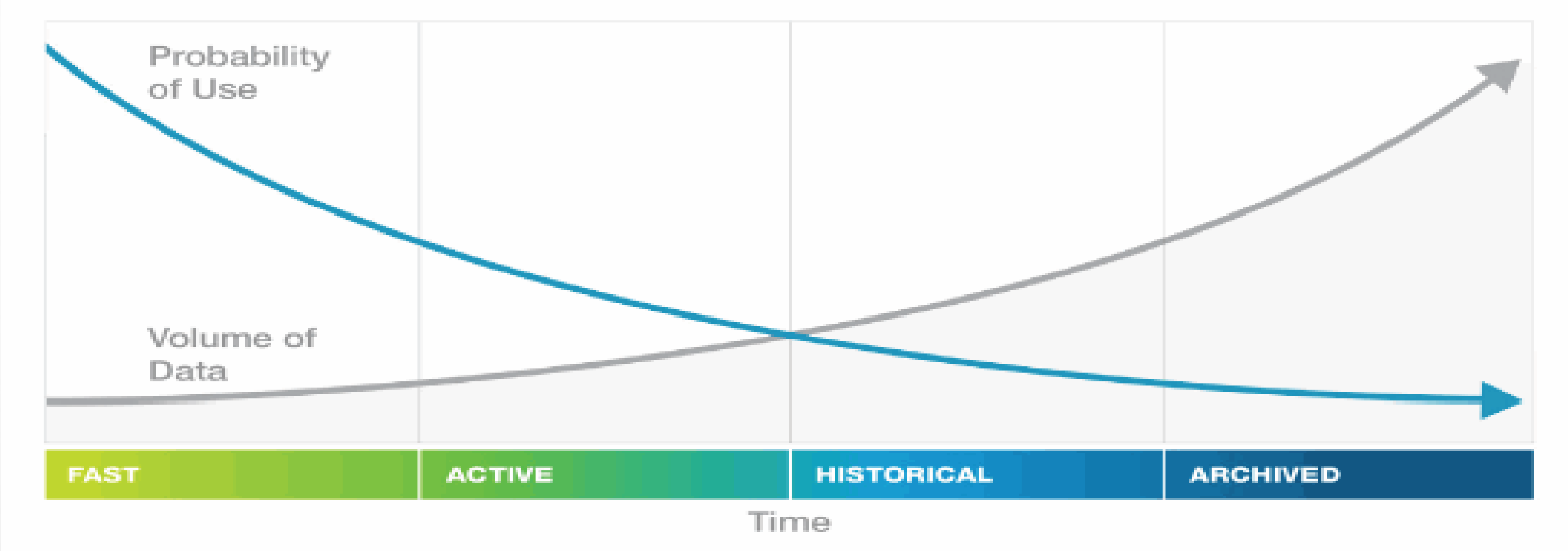
Database replication

- 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



Tiered storage

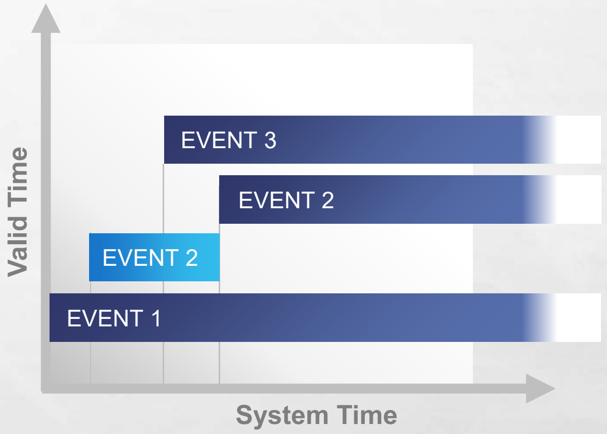
MarkLogic™



Update

MarkLogic™

- 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

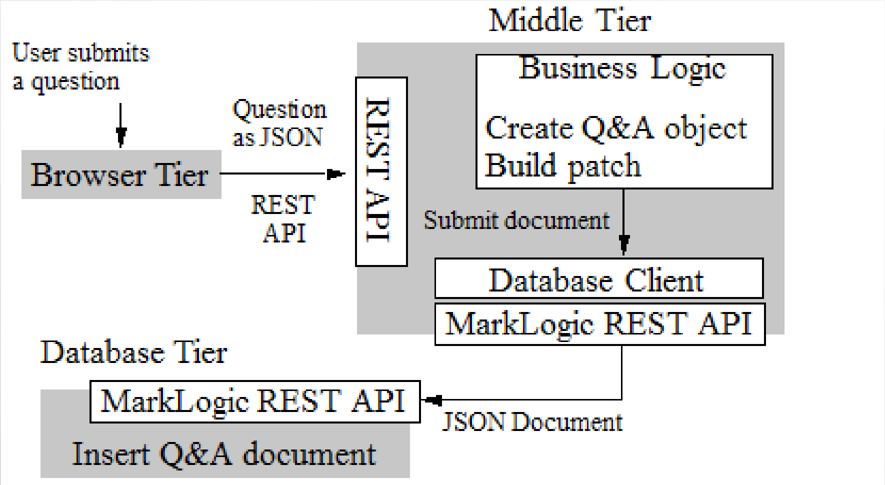


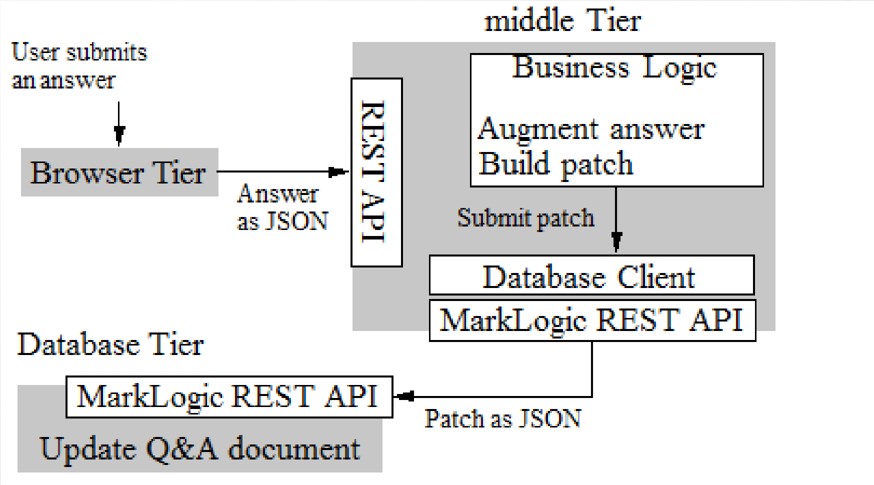
Valid Time – Real-world time, information “as it actually was”

System Time – Time it was recorded to the database

MarkLogic™

Query/ answer processing





DEVELOPMENT

Developer tools

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 o

```
object-node { "p1" : "v1", "p2"
, "p3" : fn:true(), "p4" : null-noc
"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())); }
```

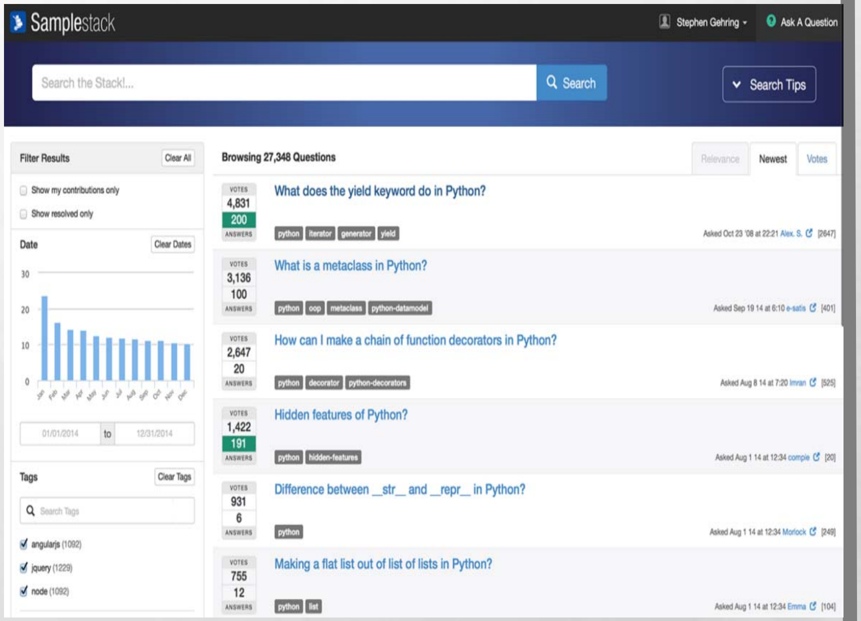
the database every
ollection

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

MarkLogic™

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



Browsing 27,348 Questions

VOTES 4,831 200 ANSWERS — What does the yield keyword do in Python? — python iterator generator yield — Asked Oct 23 '08 at 22:21 Alex. S. [2647]

VOTES 3,136 100 ANSWERS — What is a metaclass in Python? — python oop metaclass python-datamodel — Asked Sep 19 14 at 6:10 e-satis [401]

VOTES 2,647 20 ANSWERS — How can I make a chain of function decorators in Python? — python decorator python-decorators — Asked Aug 8 14 at 7:20 Imran [525]

VOTES 1,422 191 ANSWERS — Hidden features of Python? — python hidden-features — Asked Aug 1 14 at 12:34 compie [20]

VOTES 931 6 ANSWERS — Difference between __str__ and __repr__ in Python? — python — Asked Aug 1 14 at 12:34 Morlock [249]

VOTES 755 12 ANSWERS — Making a flat list out of list of lists in Python? — python list — Asked Aug 1 14 at 12:34 Emma [104]

IMPLEMENTATION CONCEPTS

MarkLogic™

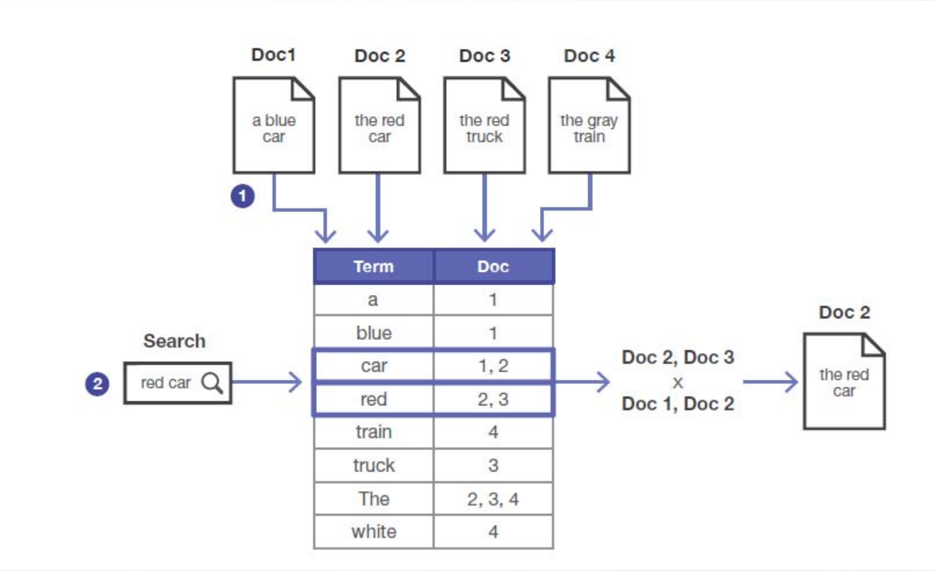
Word indexing

INVERTED INDEX

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

HOW DOES IT SEARCH TWO DIFFERENT WORDS ??

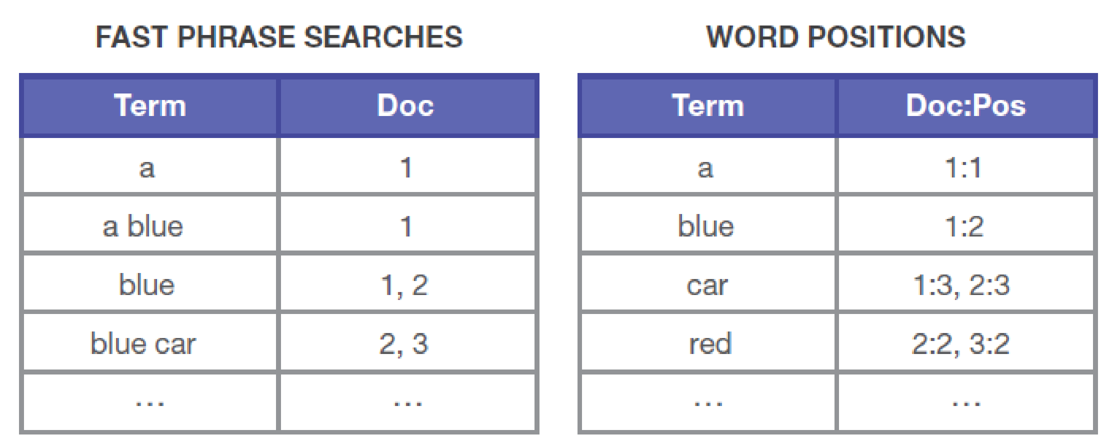
- USE THE SAME DATA STRUCTURE AND GET THE INTERSECTING DOCUMENTS



Term	Doc
a	1
blue	1
car	1, 2
red	2, 3
train	4
truck	3
The	2, 3, 4
white	4

Phrase indexing

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



FAST PHRASE SEARCHES

Term	Doc
a	1
a blue	1
blue	1, 2
blue car	2, 3
…	…

WORD POSITIONS

Term	Doc:Pos
a	1:1
blue	1:2
car	1:3, 2:3
red	2:2, 3:2
…	…

Which indexing is used in MarkLogic??...

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

Indexing structure

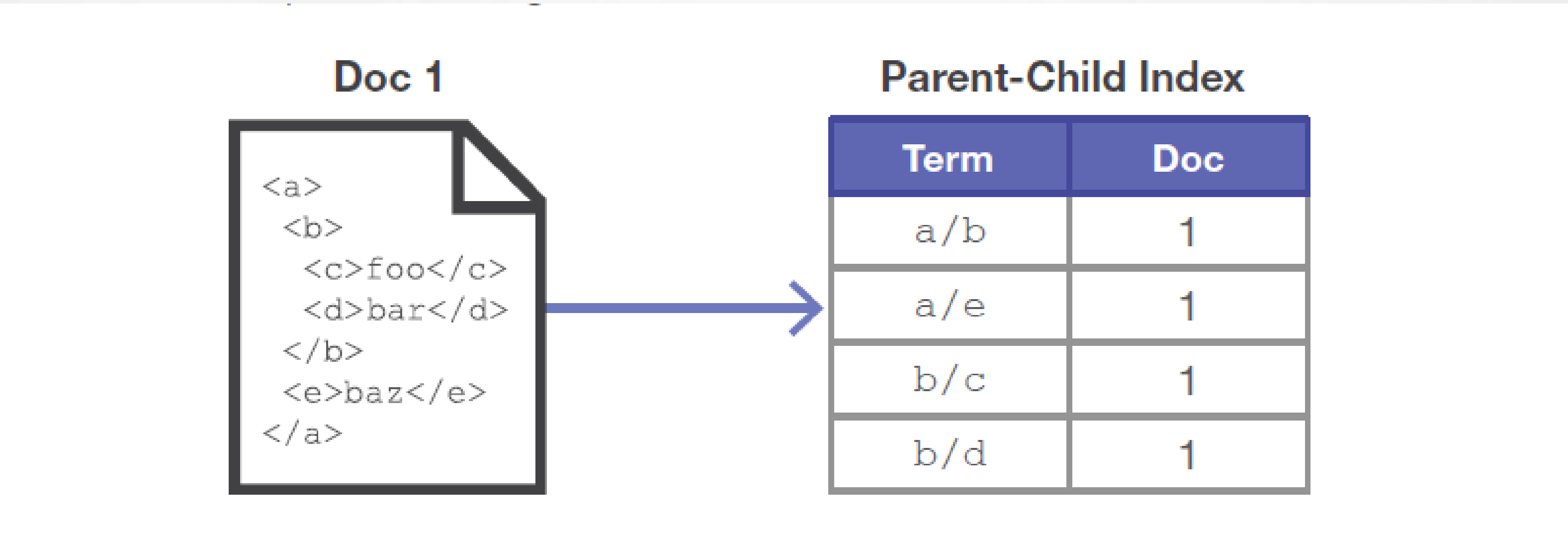
- PARENT-CHILD INDEX FOR MAINTAINING HIERARCHICAL STRUCTURE OF XML AND JSON DOCUMENTS
- IT'S SIMILAR TO FAST PHRASE SEARCH BUT USES CONSECUTIVE TAGS
- SEARCHING AN ADVANCE DATABASE BOOK TITLED "INSIDE MARKLOGIC SERVER" USES THE FOLLOWING PARENT-CHILD HIERARCHY

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

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

MarkLogic™

Indexing structure (cont.)



Doc 1

```
<a>
  <b>
    <c>foo</c>
    <d>bar</d>
  </b>
  <e>baz</e>
</a>
```

Parent-Child Index

Term	Doc
a/b	1
a/e	1
b/c	1
b/d	1

Range index

- SUPPORT ***FAST RANGE*** QUERIES, - DOCUMENTS WITHIN PARTICULAR SET OF DATES
- **DATA TYPE AWARE EQUALITY** QUERIES – COMPARE DATES BASED ON SEMANTIC VALUE RATHER THAN ITS LEXICALLY CORRECT INITIALIZED VALUE
- GET **ORDER BY** RESULTS – SEARCH RESULTS SORTED BY ITEM PRICE
- **CROSS DOCUMENT** JOINS – MERGING TWO DOCUMENTS, ONE CONTAINING THE NAME OF THE PEOPLE AND THE OTHER CONTAINING THE DATE OF BIRTH OF THE PEOPLE

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)

Geospatial 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

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)

```
xdmp:eval("doc('/json/sample_doc.json')",
<options xmlns="xdmp:eval">
<timestamp>96825</timestamp>
</options>)
```

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*

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=/xml_d
ocs/sample_lock.xml

HTTP/1.1 200 Document Retrieved
Content-type: application/xml
 ETag: “254768939037681240"
Connection: close
```

```
$ curl --anyauth --user user:password -i -X
PUT -d"<modified-data/>"
-H "Content-type: application/xml"
-H "If-Match: 254768939037681240"
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.xml'  -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=sample_document_uri***

METADATA

*http://host:port/version/documents?uri=sample_document_uri&**category=category_of_metadata***

CONTENT AND METADATA

*http://host:port/version/documents?**uri=doc_uri&category=metadat_content_desc***

REST API Searching

SEARCHING

```
curl --anyauth --user user:password -X
GET  -H "Accept: application/json"
http://localhost:8000/LATEST/search?q=
hamlet
```

```
...
"matches":
[ { "path":
"fn:doc("/shakespeare/plays/
hamlet.json")/PLAY/TITLE",
"match-text": [
"The Tragedy of ",
{ "highlight": "Hamlet" },
", Prince of Denmark"
] }
,]
...
```

REST API Streaming

STREAMING

NO NEED TO LOAD THE ENTIRE CONTENT INTO MEMORY

```
curl --anyauth --user user:password -i -o stream_sample.jpg -x get  -h "accept: application/jpg" -r "0-178564"  http://localhost:8000/latest/documents?uri=/stream/stream_test.jpg
```

REST API Patch UPDATE

```
curl --anyauth --user user:password -x post -d
@./patch_example.xml -i -h "content-type:
application/xml“  -h "x-http-method-override: patch"
http://localhost:8000/latest/documents?uri=/patch
/patch_example.xml
```

PATCH TEMPLATE

```
<rapi:patch
xmlns:rapi="http://marklogic.com/rest-api">
  <rapi:insert />
  <rapi:replace-insert />
  <rapi:replace/>
  <rapi:delete />
</rapi:patch>
```

REST API Patch UPDATE (cont.)

```
<rapi:patch
xmlns:rapi="http://marklogic.com/rest-api">

  <rapi:insert context="/header/p[1]">

    <rapi:attribute-list attr1="val1" />

  </rapi:insert>

</rapi:patch>
```

Before Update	After Update

<header>
  <p>one</p>
  <p>two</p>
  <p>three</p>
</header>
	<header>
	<p attr1=“val1">
	    one
	  </p>
	  <p>two</p>
	  <p>three</p>
	</header>

REST API DELETE Request

BLANK DIRECTORY OR COLLECTION NAME DELETES THE ENTIRE DATABASE

SINGLE DOCUMENT

*http://host:port/version/documents?**uri=path_of_document_uri***

MULTIPLE DOCUMENTS

*http://host:port/version/search?**collection=name_of_the_collection***

APPLICATIONS

When MarkLogic?

- SPARSE, DIVERSE DATA
- QUERIES DATA ACCORDING TO POWER LAW
- RENDER RESULT IN SPECIFIC FORMAT DIRECTLY
- TERABYTES OF DATA IN DIFFERENT GEOGRAPHICAL LOCATIONS.
- NEED FASTER RESULTS.
- ELASTIC SECURITY, REPLICATION
- …….

MarkLogic™

Use cases

HealthCare.gov

BBC

Mitchell1

MarkLogic™

Project - HealthCare.gov

- 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

Open Enrollment starts November 1. Are you ready?

First time applying here?

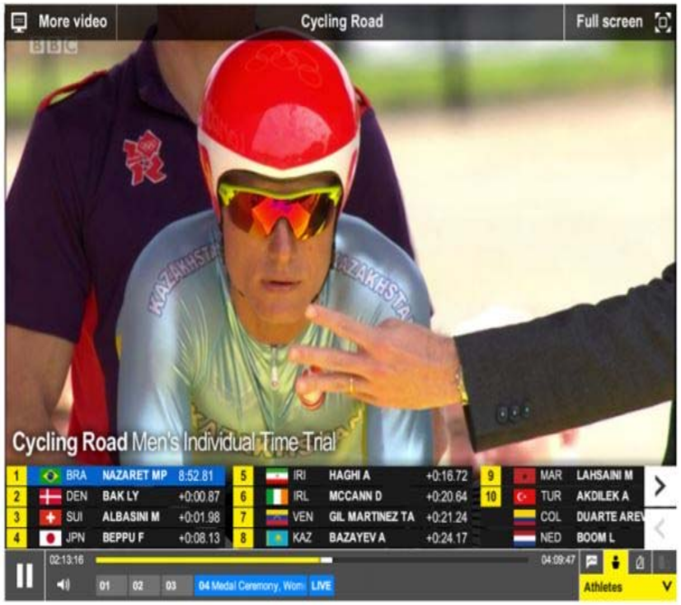
Have a 2016 Marketplace plan?

Have a 2016 plan in Kentucky? Learn about using HealthCare.gov for 2017.

MarkLogic™

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



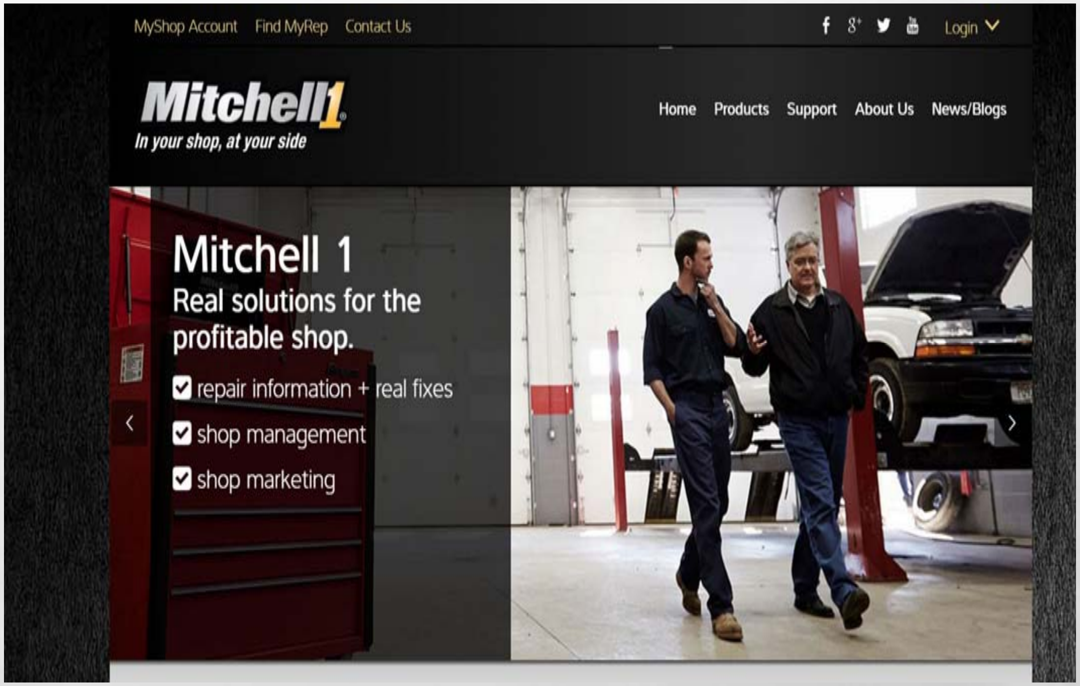
1	BRA	NAZARET MP	8:52.81	5	IRI	HAGHI A	+0:16.72	9	MAR	LAHSAINI M	
2	DEN	BAK LY	+0:00.87	6	IRL	MCCANN D	+0:20.64	10	TUR	AKDILEK A	
3	SUI	ALBASINI M	+0:01.98	7	VEN	GIL MARTINEZ TA	+0:21.24		COL	DUARTE AREV	
4	JPN	BEPPU F	+0:08.13	8	KAZ	BAZAYEV A	+0:24.17		NED	BOOM L	

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



MarkLogic™

And many more...

UNIVERSITY OF TORONTO

American Society of Agronomy

Bank of America Merrill Lynch

KUNGL. BIBLIOTEKET
National Library of Sweden

American Psychological Association

WARNER HOME VIDEO

citi

Morgan Stanley

U.S. AIR FORCE

BOEING

DHL PARCEL

J.P.Morgan

McGRAW HILL FINANCIAL

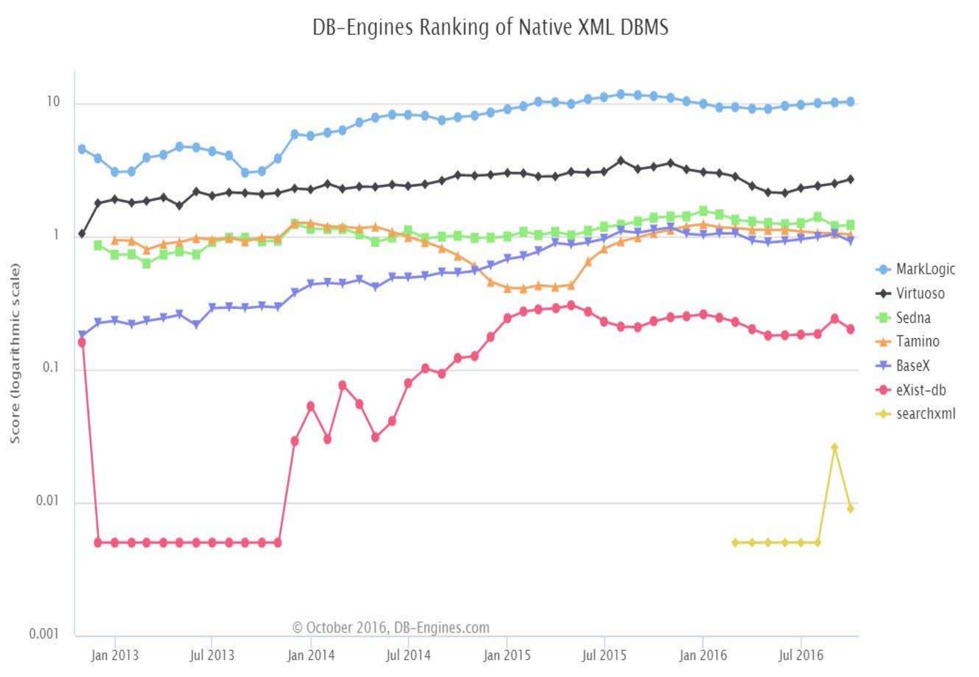
LOCKHEED MARTIN

KPMG

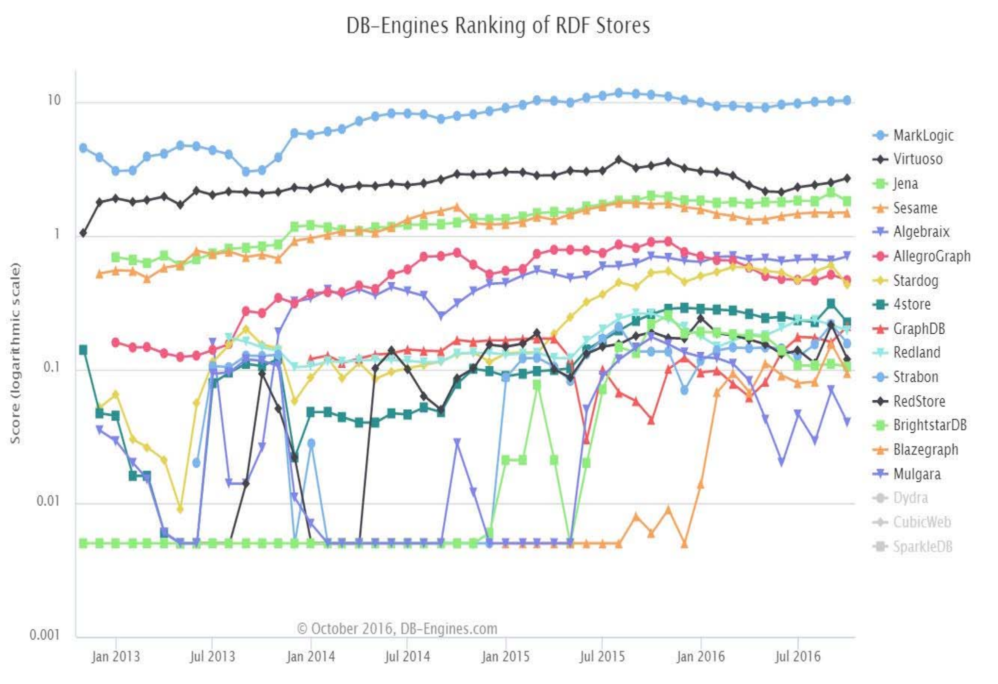
NBC

Trend charts

MarkLogic™

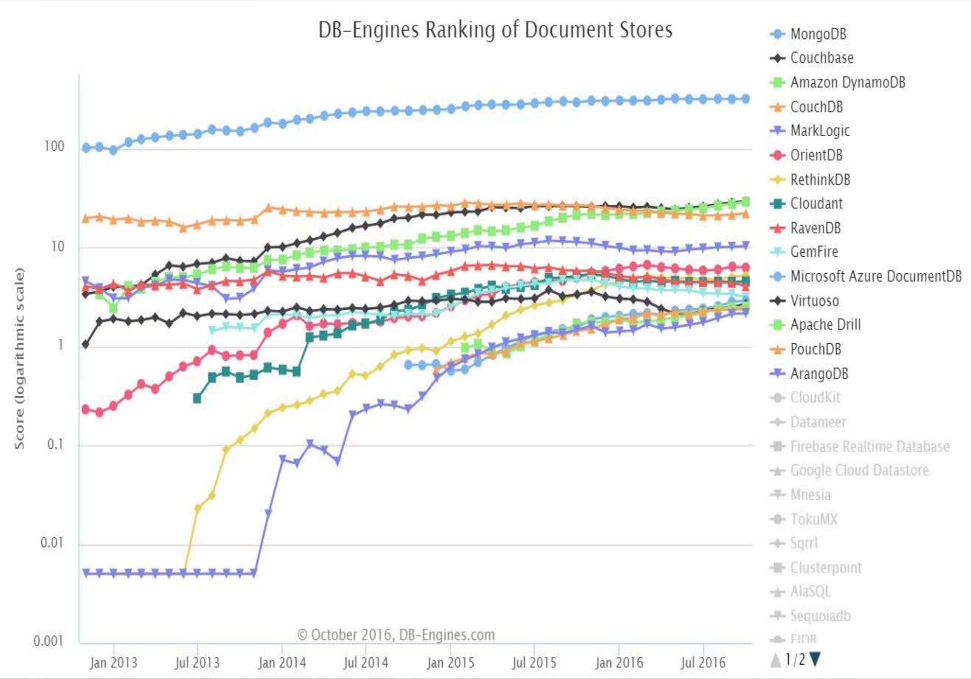

DB-Engines Ranking of Native XML DBMS

© October 2016, DB-Engines.com

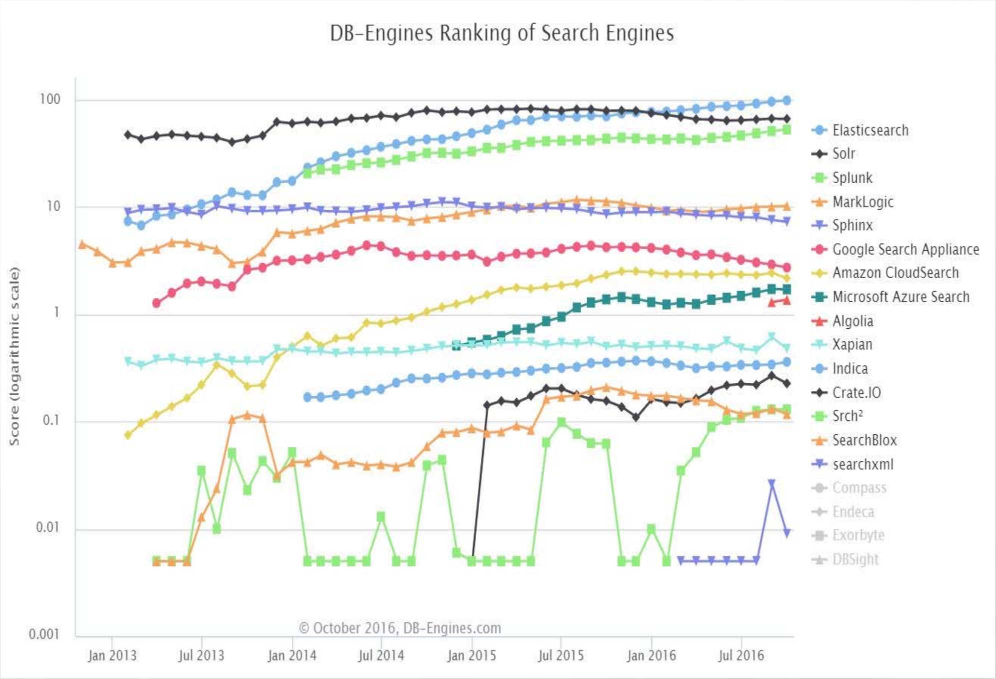

DB-Engines Ranking of RDF Stores

© October 2016, DB-Engines.com

Trend charts (cont.)

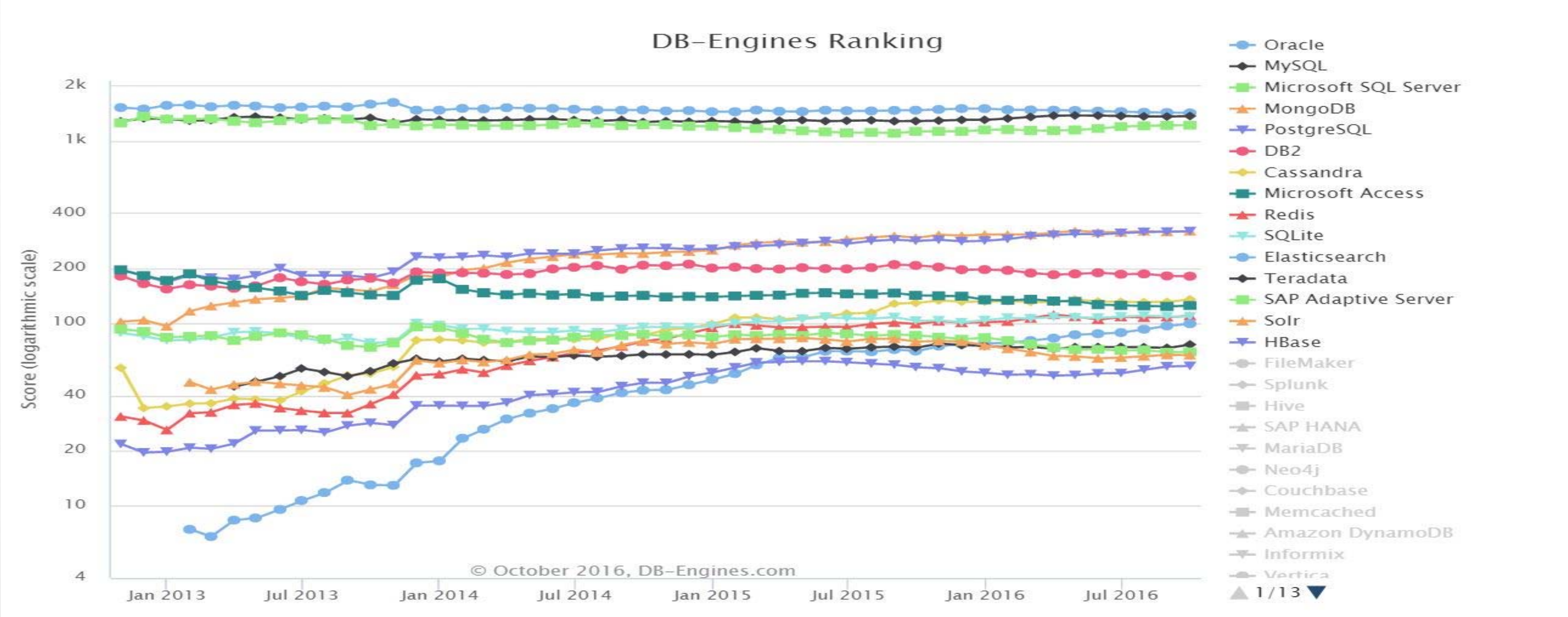

DB-Engines Ranking of Document Stores

© October 2016, DB-Engines.com


DB-Engines Ranking of Search Engines

© October 2016, DB-Engines.com

Why not MarkLogic?



References

- M. CORPORATION, POWERED, AND M. S. 7, "REST APPLICATION DEVELOPER'S GUIDE — MARKLOGIC 8 PRODUCT DOCUMENTATION," 2016. [ONLINE]. AVAILABLE: HTTPS://DOCS.MARKLOGIC.COM/GUIDE/REST-DEV.
- J HUNTER. INSIDE MARKLOGIC SERVER, 2011.
- DB-ENGINES RANKING. KNOWLEDGE BASE OF RELATIONAL AND NOSQL DATABASE MANAGEMENT SYSTEMS, 2015.
- MARKLOGIC. HTTP://WWW.MARKLOGIC.COM/, 2001.
- MARKLOGIC SERVER, CONCEPTS GUIDE. HTTPS://DOCS.MARKLOGIC.COM/GUIDE/CONCEPTS.PDF
- MARKLOGIC. HTTPS://EN.WIKIPEDIA.ORG/WIKI/MARKLOGIC, 2016.

MarkLogic™

thank you

GROUP 11

AVIRUP CHAKRABORTY

RASHA ELHESHA

SAPTARSHI CHAKRABORTY

DEBARSHI MITRA