Jason Hunter
Principal Technologist

MarkLogic at a Glance

MarkLogic Server is a purpose-built database for managing unstructured information

- 4th fastest growing software company in Silicon Valley
- 500+ live implementations
- Headquarters in San Carlos, California
- Offices in Silicon Valley, DC, New York, London, & Frankfurt

200+ Customers

Media Customers

LexisNexis®
SIMON & SCHUSTER A CBS COMPANY
McGraw Hill Education
OXFORD UNIVERSITY PRESS
DATAMONITOR
ELSEVIER
THOMSON REUTERS
nature.com
PEARSON
INGRAM INGRAM BOOK GROUP
Springer science+business media

Government Customers

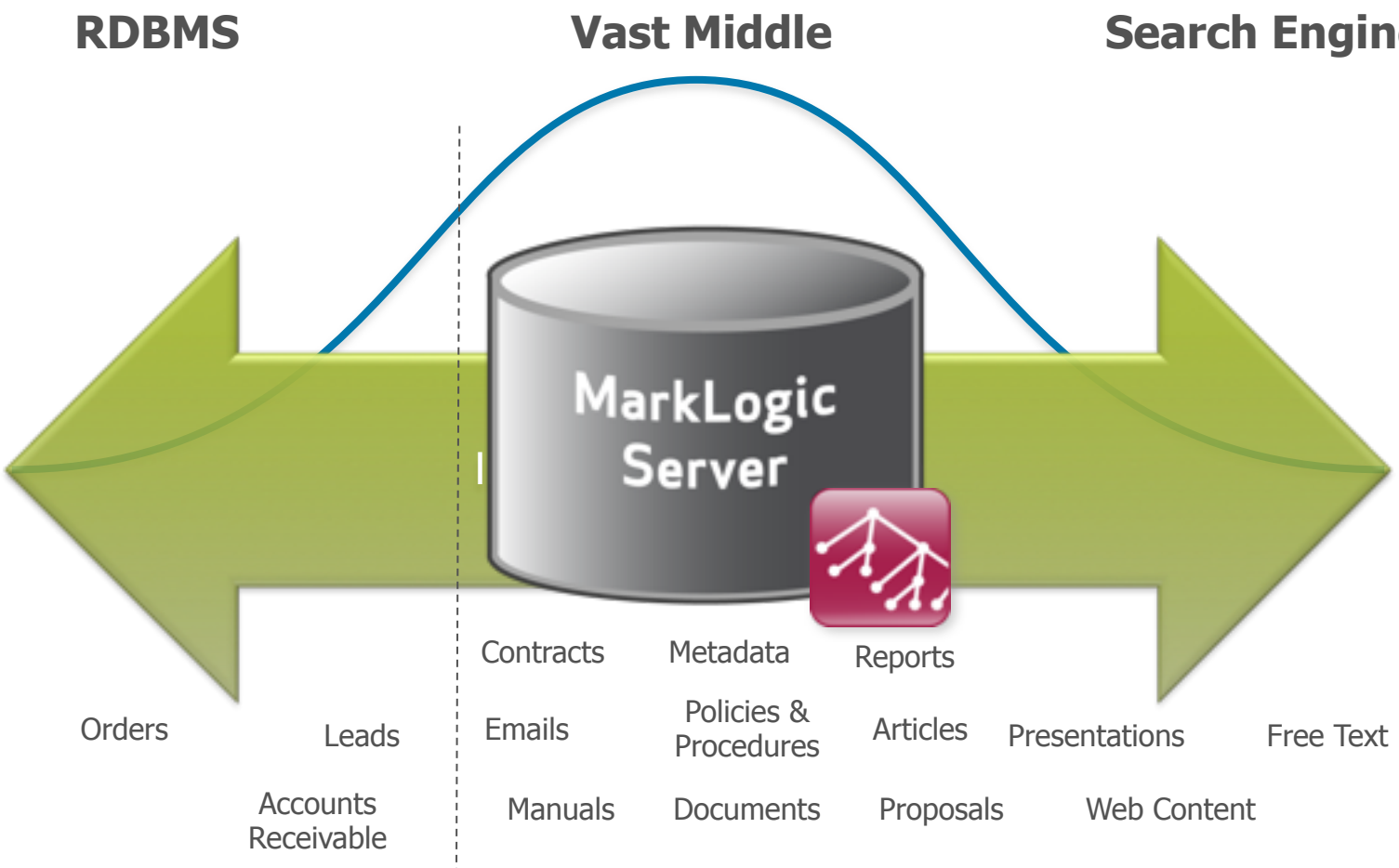
CQ Congressional Quarterly Inc.
DISA
U.S. AIR FORCE
LIBRARY OF CONGRESS
U.S. ARMY

Financial Services and Other Customers

JPMorgan
The Economist
jetBlue AIRWAYS
YELL.COM
audible.com
STANDARD &POOR'S
VISA
UNITED
THE CHURCH OF JESUS CHRIST OF LATTER-DAY SAINTS
OpenConnect

MarkLogic™

The Information Continuum



MarkLogic

MarkLogic Server in Ten Adjectives

- Document-centric
- Transactional
- Search-centric
- Structure-aware
- Schema-free
- XQuery- and XSLT-driven
- Extremely fast
- Clustered
- Analytical
- Database server

MarkLogic

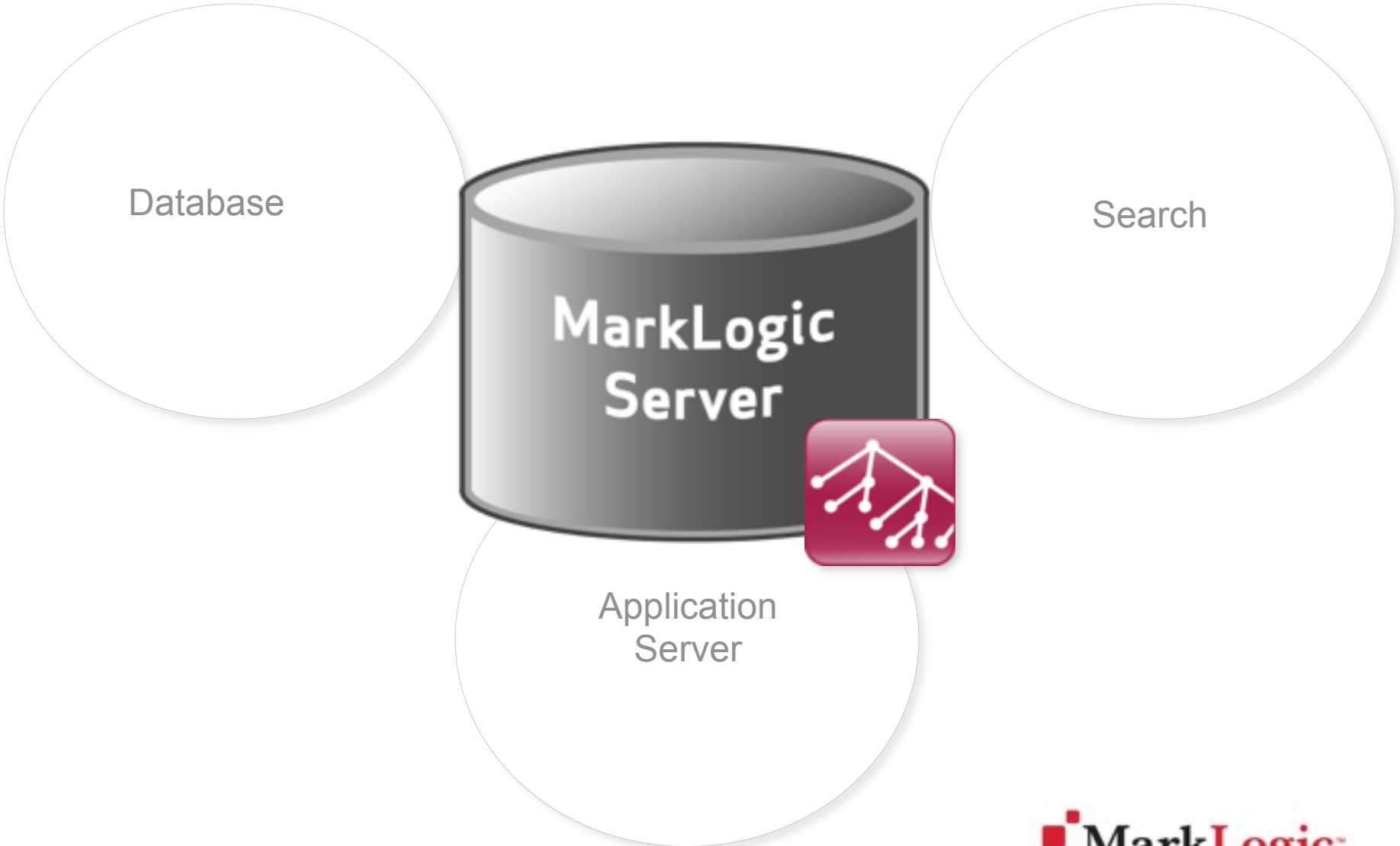
Information Applications

Categories include:

Common Repository	Metadata Catalog	Digital Content Delivery	Information Intelligence	Social Applications Platform
Consolidate information from variety of sources for better access and maintenance	Maintain repository of metadata to facilitate information sharing and discoverability	Repurpose existing information and distribute across devices and channels	Exploit heterogeneous information leveraging content analytics to discover trends and patterns	Share information to improve processes and support better decision-making
• Elsevier • JPMorgan Chase • Congressional Quarterly • Intel Community	• Library of Congress • National Archives • Intel Community	• Oxford University • JPMorgan Chase • Wiley • jetBlue	• State Department • Open Connect • Intel Community • Docgenix	• Warrior Gateway • BusinessWeek • US Army

MarkLogic

MarkLogic Server



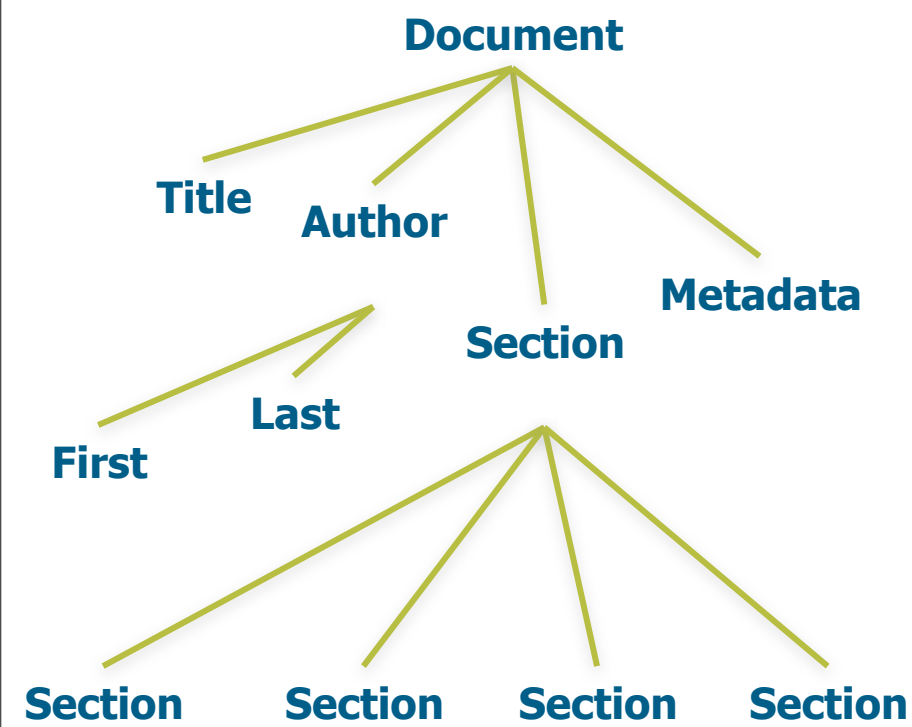
MarkLogic

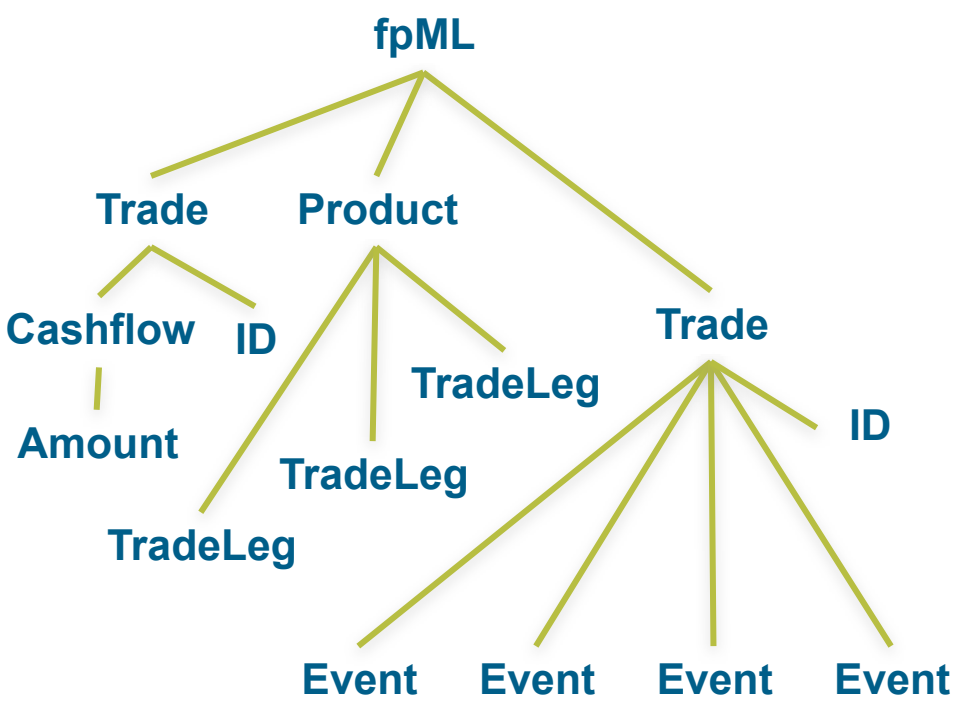
Universal Index

MarkLogic™

Data Model

- A database for unstructured (and semi-structured) information
- XML Data Model





MarkLogic

Example Document

```
<article>
   <title>A Relational Model of Data for Large Shared Data Banks</title>
   <author><first-name>Edgar</first-name><last-name>Codd</last-name></author>
   <abstract>
        Future users of data banks must be protected from having to know how the data is organized
        in the machine (the internal representation). . . . Changes in data representation will often be
        needed . . .
   </abstract>
   <body>
          <section>
             <section> … has values which uniquely identify each element ...  </section>
          </section>
          <section> … version of <product>IMS</product> provides the user . . .  </section>
   </body>
   <metadata><vol>13</vol><number>6</number><year>1970</year></metadata>
</article>
```


MarkLogic

1) Text

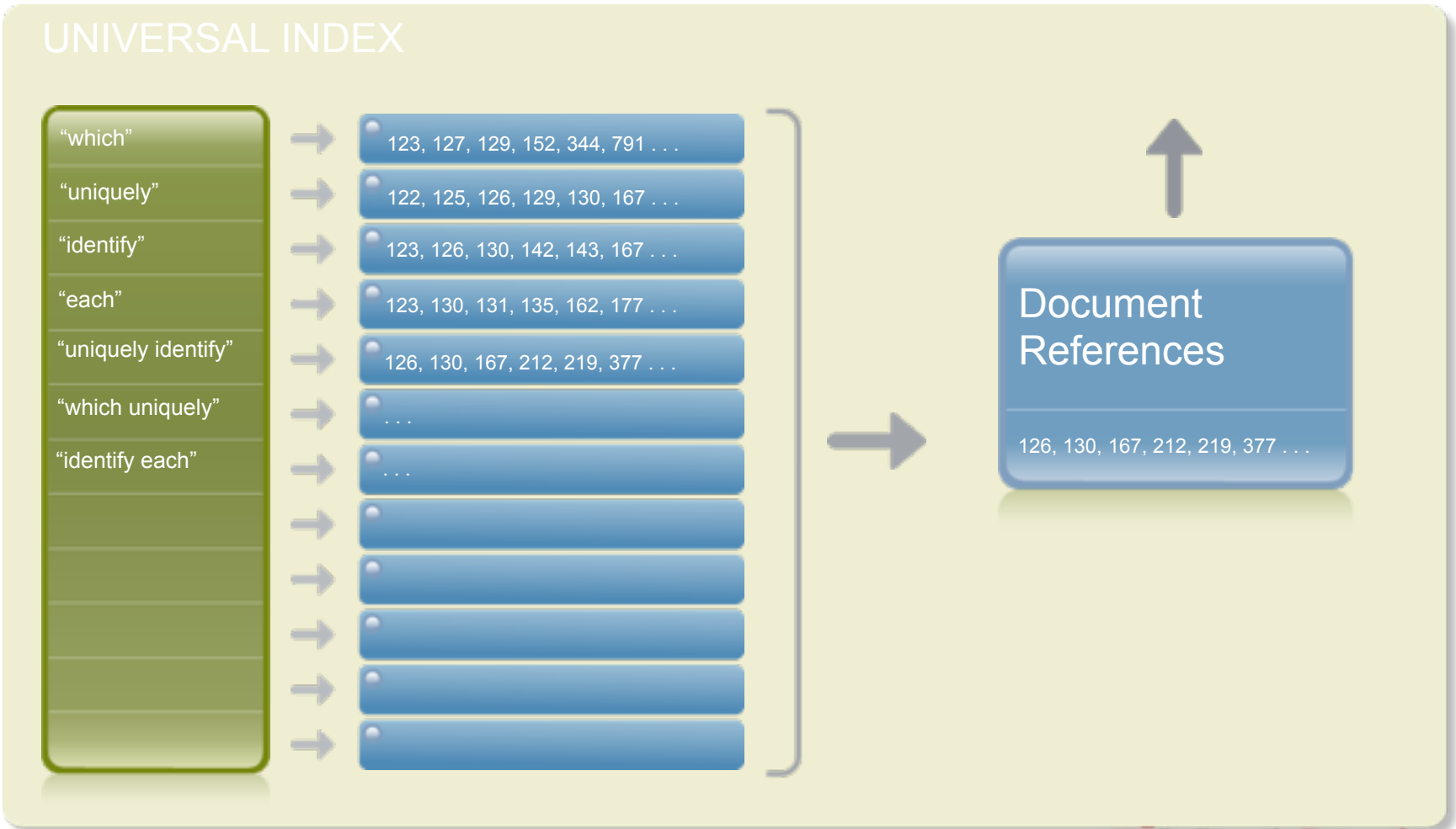
Find all documents that contain the phrase “uniquely identify”

```
<article>
   <title>A Relational Model of Data for Large Shared Data Banks</title>
   <author><first-name>Edgar</first-name><last-name>Codd</last-name></author>
   <abstract>
        Future users of data banks must be protected from having to know how the data is organized
        in the machine (the internal representation). . . . Changes in data representation will often be
        needed . . .
   </abstract>
   <body>
          <section>
             <section> … has values which uniquely identify each element ... </section>
          </section>
          <section> … version of <product>IMS</product> provides the user . . .  </section>
   </body>
   <metadata><vol>13</vol><number>6</number><year>1970</year></metadata>
</article>
```


MarkLogic

1) Text

Find all documents that contain the phrase “uniquely identify”



MarkLogic™

2) Structure

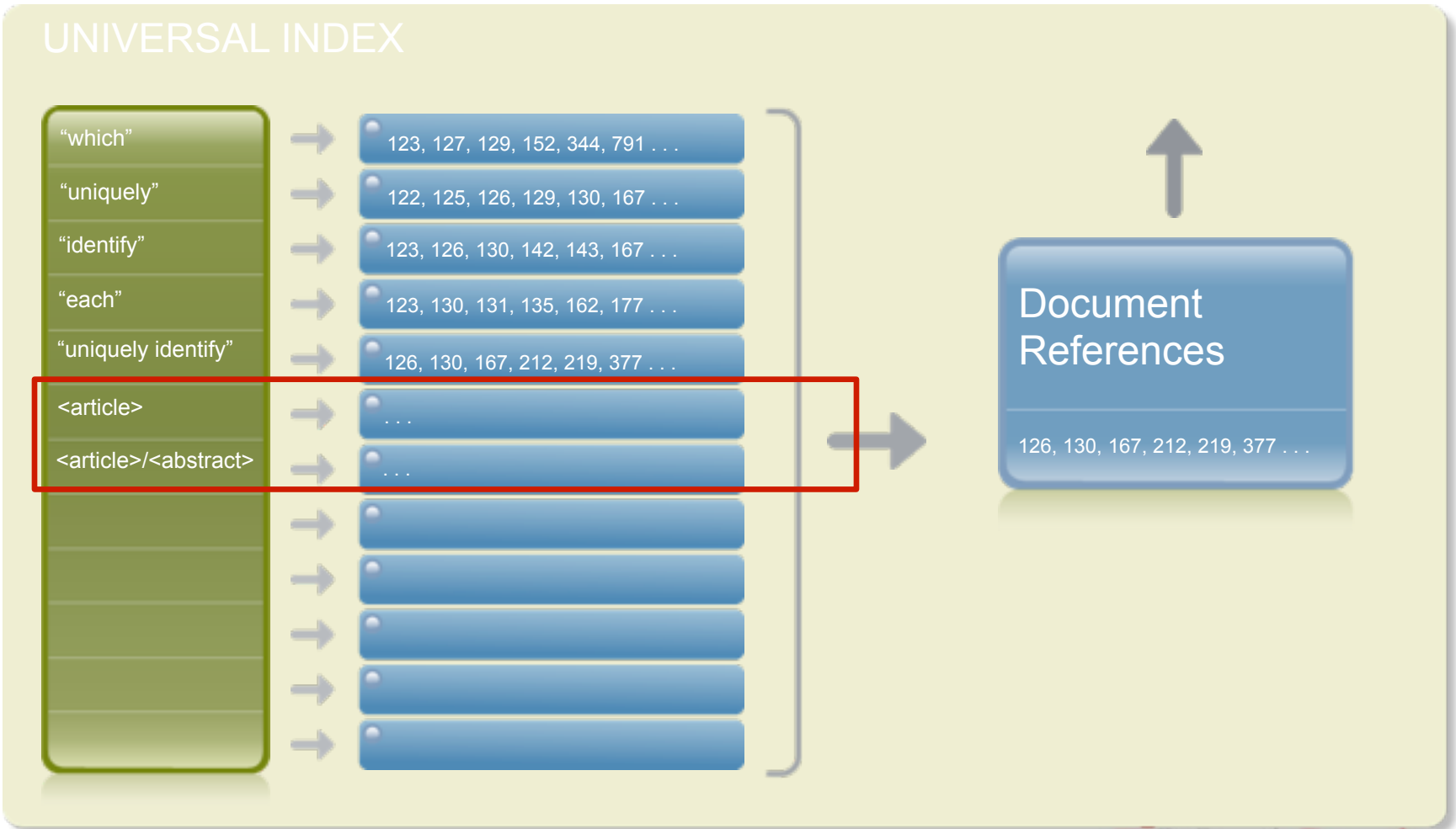
Find all articles that have an abstract

```
<article>
    <title>A Relational Model of Data for Large Shared Data Banks</title>
    <author><first-name>Edgar</first-name><last-name>Codd</last-name></author>
    <abstract>
        Future users of data banks must be protected from having to know how the data is organized
        in the machine (the internal representation). . . . Changes in data representation will often be
        needed . . .
    </abstract>
    <body>
        <section>
            <section> … has values which uniquely identify each element ... </section>
        </section>
        <section> … version of <product>IMS</product> provides the user . . . </section>
    </body>
    <metadata><vol>13</vol><number>6</number><year>1970</year></metadata>
</article>
```


MarkLogic

2) Structure

Find all articles that have an abstract



MarkLogic

3) Values

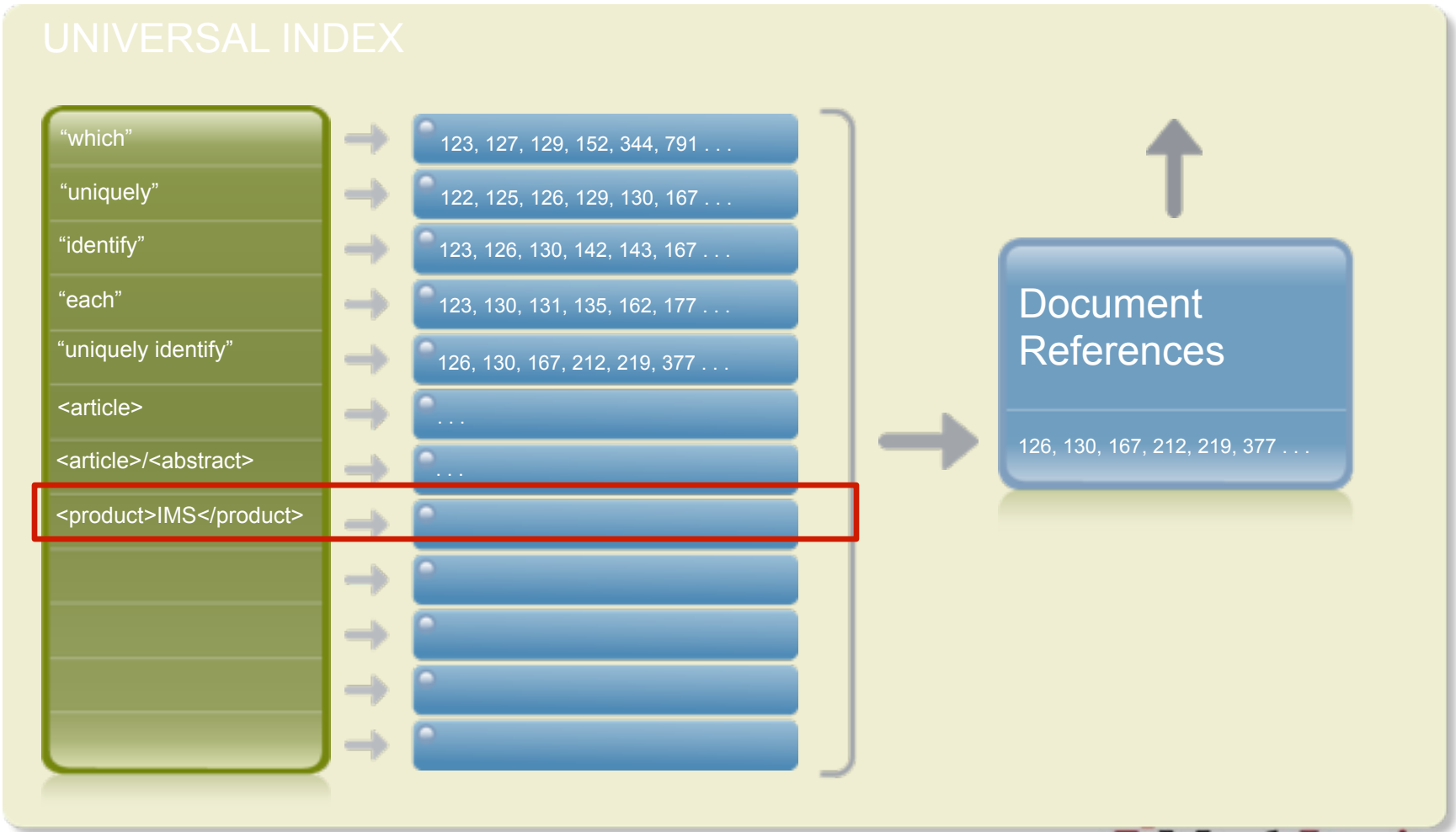
Find all documents that mention the product “IMS”

```
<article>
    <title>A Relational Model of Data for Large Shared Data Banks</title>
    <author><first-name>Edgar</first-name><last-name>Codd</last-name></author>
    <abstract>
        Future users of data banks must be protected from having to know how the data is organized
        in the machine (the internal representation). . . . Changes in data representation will often be
        needed . . .
    </abstract>
    <body>
        <section>
            <section> … has values which uniquely identify each element ... </section>
        </section>
        <section> … version of <product>IMS</product> provides the user . . . </section>
    </body>
    <metadata><vol>13</vol><number>6</number><year>1970</year></metadata>
</article>
```


MarkLogic

3) Values

Find all documents that mention the product “IMS”



MarkLogic

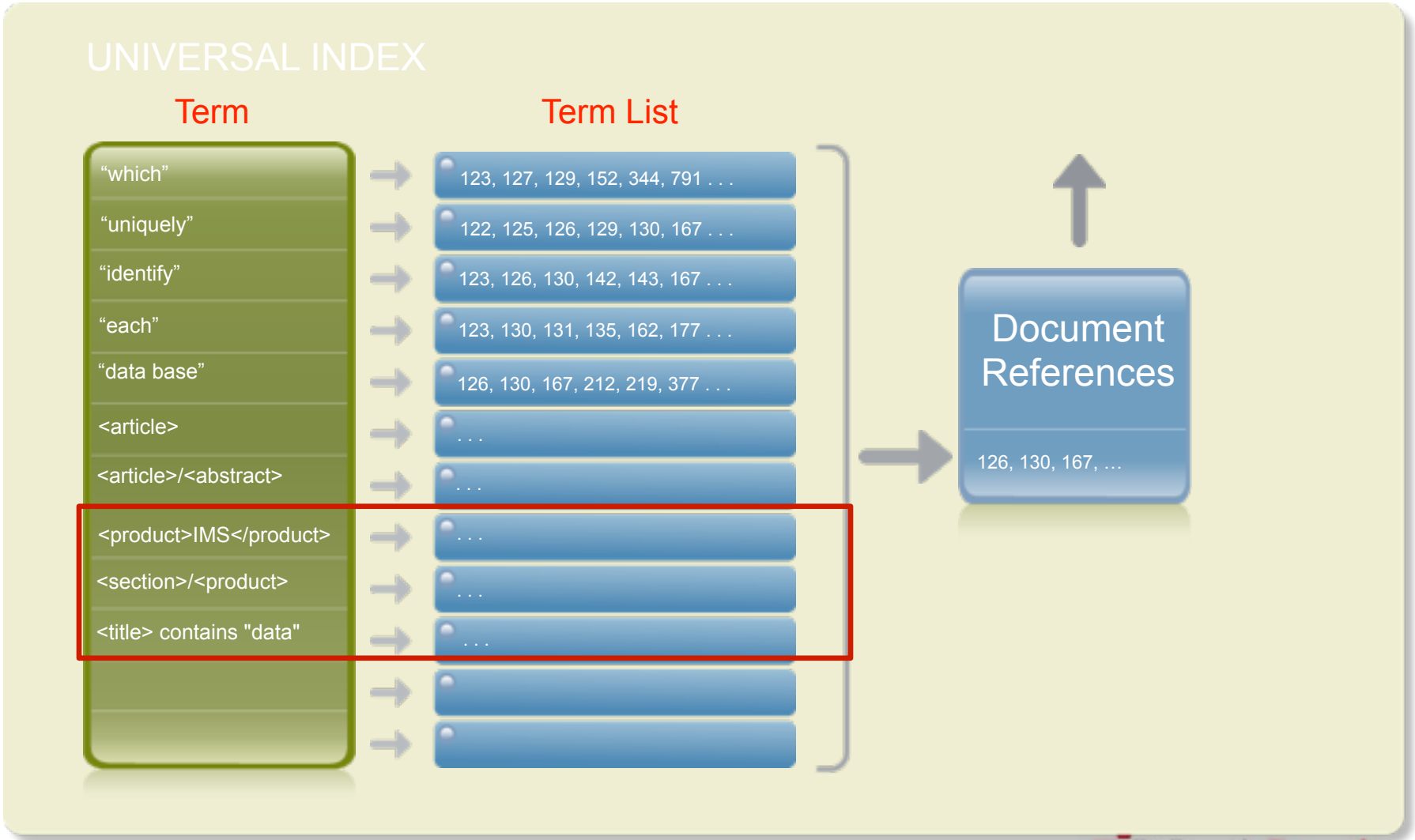
4) Structure, Values, and Text

Find articles that contain “data” in the title and mention the product IMS in a section

```
<article>
    <title>A Relational Model of Data for Large Shared Data Banks</title>
    <author><first-name>Edgar</first-name><last-name>Codd</last-name></author>
    <abstract>
        Future users of data banks must be protected from having to know how the data is organized
        in the machine (the internal representation). . . . Changes in data representation will often be
        needed . . .
    </abstract>
    <body>
        <section>
            <section> … has values which uniquely identify each element ... </section>
        </section>
        <section> . . version of <product>IMS</product> provides the user . . . </section>
    </body>
    <metadata><vol>13</vol><number>6</number><year>1970</year></metadata>
</article>
```


MarkLogic

4) Structure, Values, and Text



MarkLogic

5) Scalars

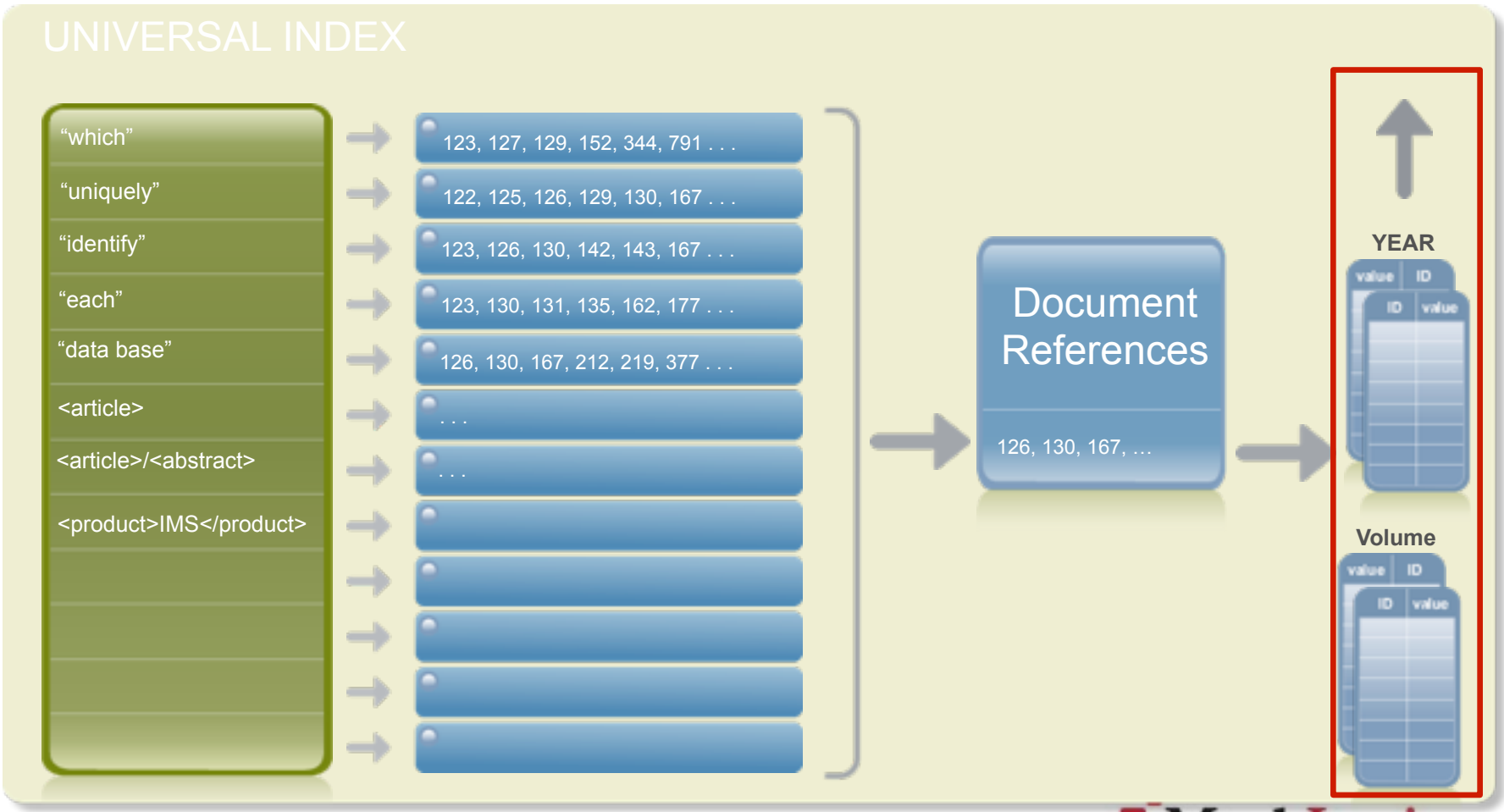
How many of the articles that contain "data base" were written in each of the last 5 decades?

```
<article>
    <title>A Relational Model of Data for Large Shared Data Banks</title>
    <author><first-name>Edgar</first-name><last-name>Codd</last-name></author>
    <abstract>
        Future users of data banks must be protected from having to know how the data
        is organized in the machine (the internal representation). . . . Changes in data
        representation will often be needed . . .
    </abstract>
    <body>
        <section>
            <section> … has values which uniquely identify each element ... </section>
        </section>
        <section> … version of <product>IMS</product> provides the user . . . </
    section>
    </body>
    <metadata><vol>13</vol><number>6</number><year>1970</year></metadata>
</article>
```


MarkLogic

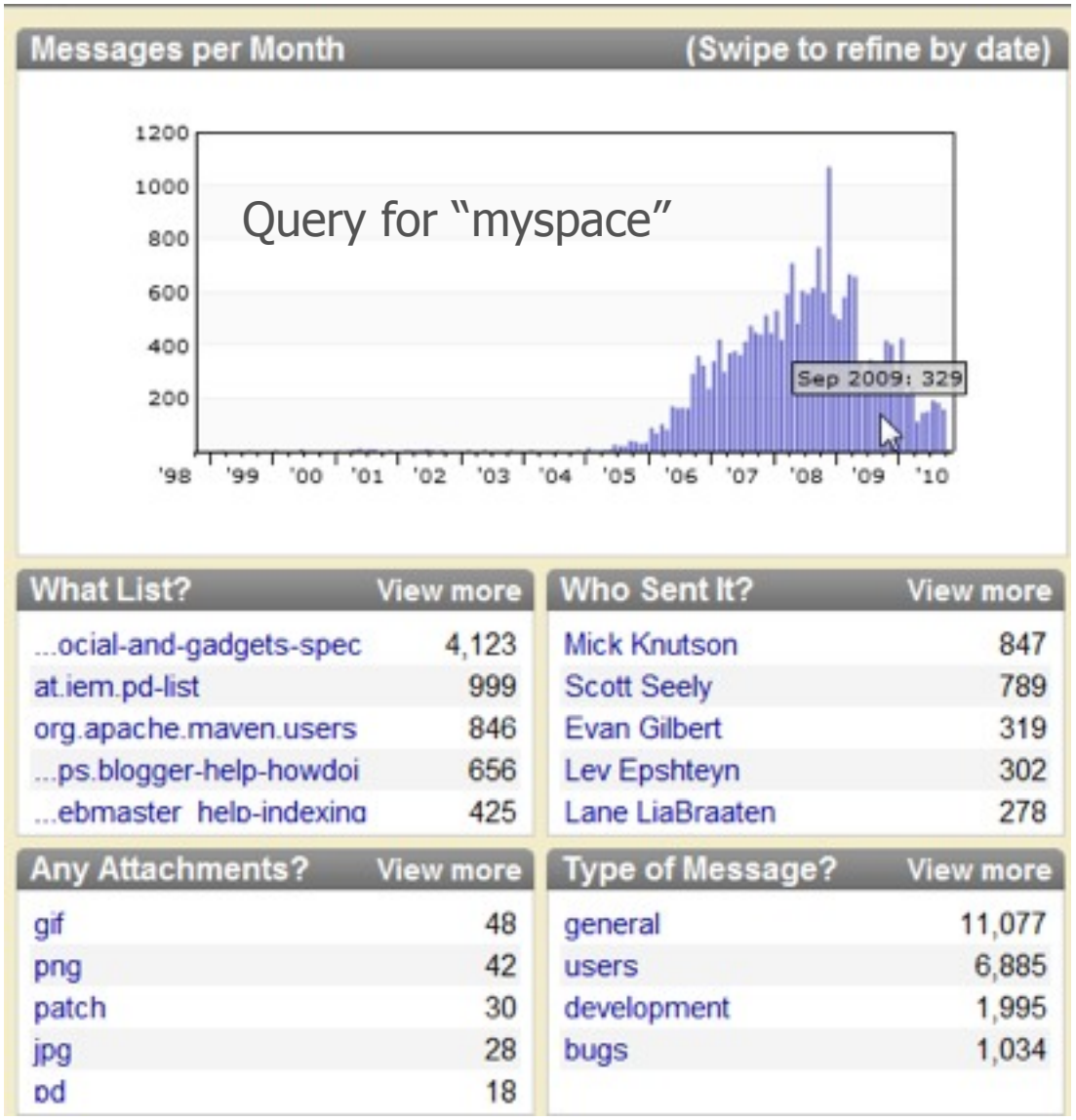
5) Range Indexes: Scalar Queries and Aggregation

How many of the articles that contain "data base" were written in each of the last 5 decades?



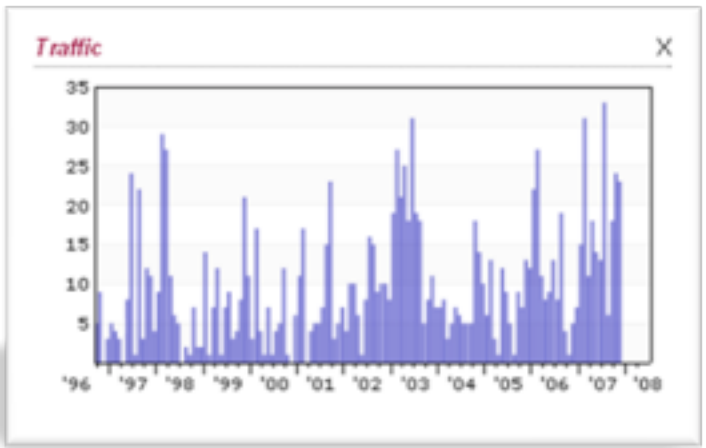
MarkLogic

5) Range Indexes: Scalar Queries and Aggregation



MarkLogic

5) Range Indexes: Scalar Queries and Aggregation



MarkLogic

6) All Of The Above

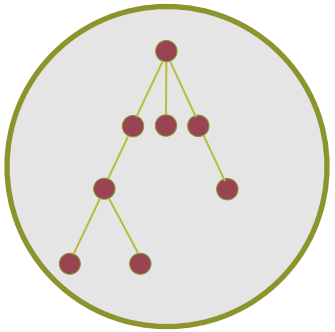
Find all articles that contain “data” in the title and mention the product IMS in a section, grouping by year.

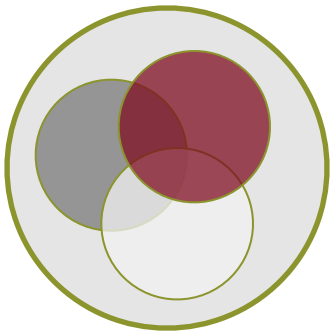
```
<article>
    <title>A Relational Model of Data for Large Shared Data Banks</title>
    <author><first-name>Edgar</first-name><last-name>Codd</last-name></author>
    <abstract>
        Future users of data banks must be protected from having to know how the data is organized
        in the machine (the internal representation). . . . Changes in data representation will often be
        needed . . .
    </abstract>
    <body>
        <section>
            <section> … has values which uniquely identify each element ... </section>
        </section>
        <section> . . version of <product>IMS</product> provides the user . . . </section>
    </body>
    <metadata><vol>13</vol><number>6</number><year>1970</year></metadata>
</article>
```

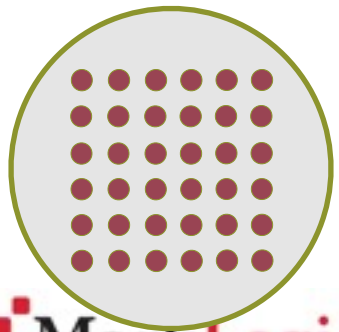

MarkLogic

7) Collections and Security

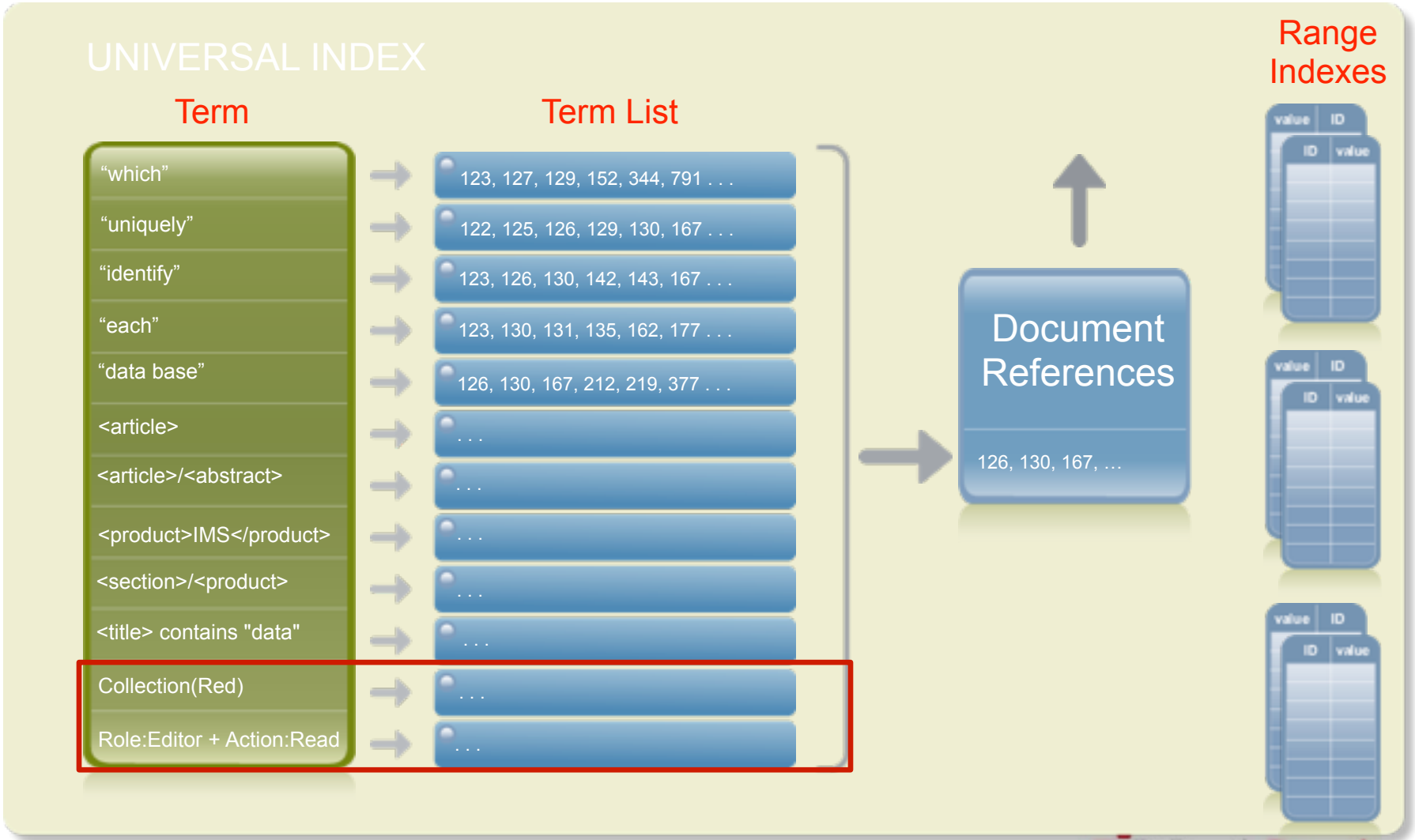
- Directories
  - Exclusive, hierarchical, analogous to file system, based on URI
- Collections
  - Set-based, N:N relationship
- Security
  - Invisible to your app





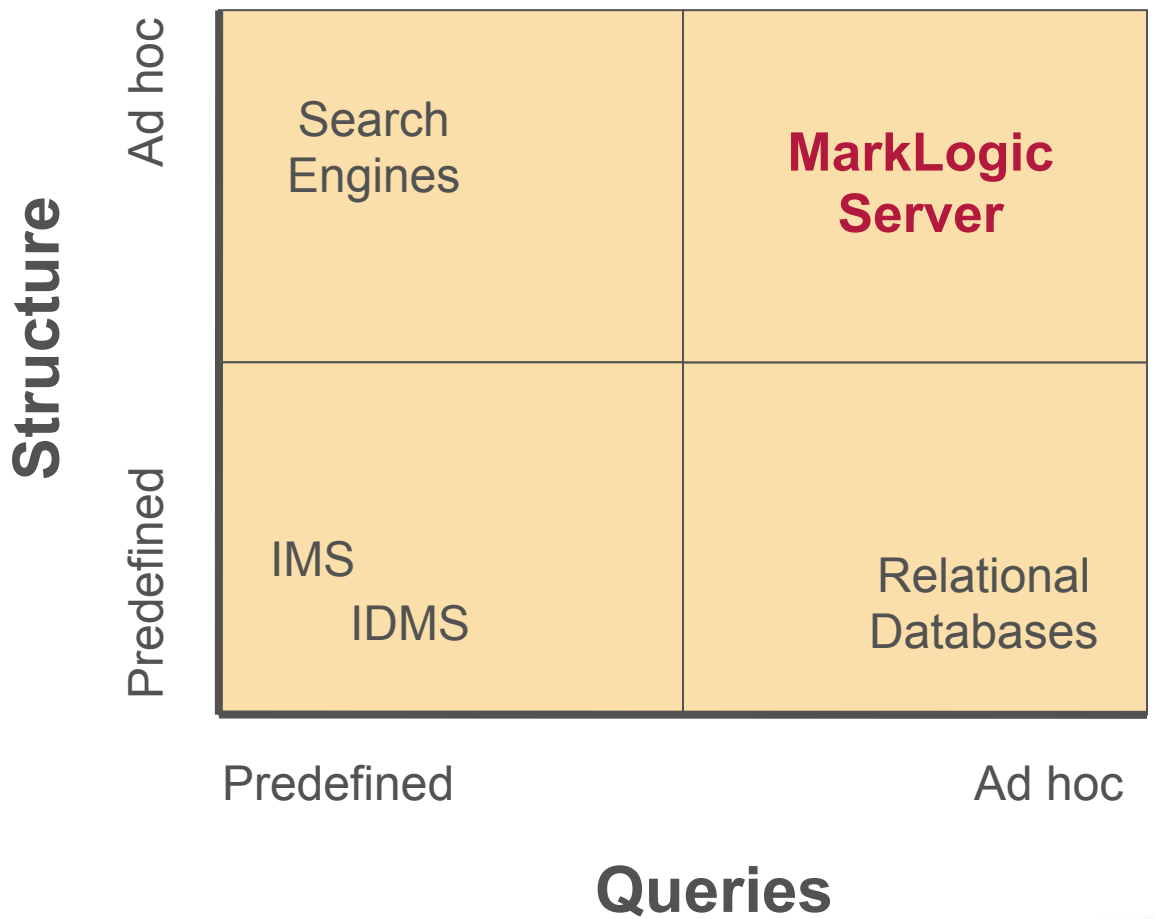


7) Collections and Security



MarkLogic

Degrees Of Flexibility



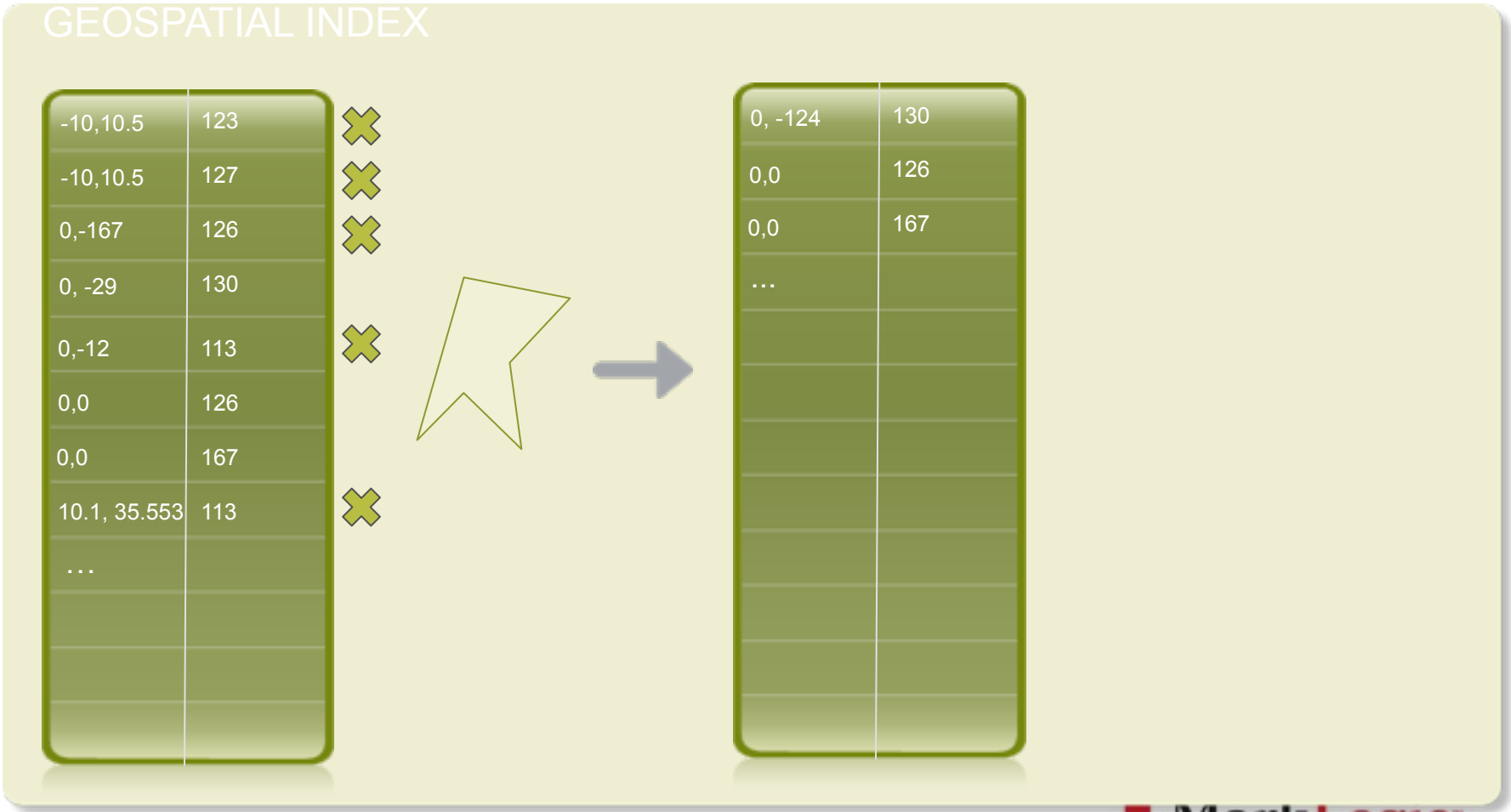
MarkLogic

Other Index Features

MarkLogic™

Spatial Indexing

Points ordered in latitude major order; special scan operators apply geospatial query constraints



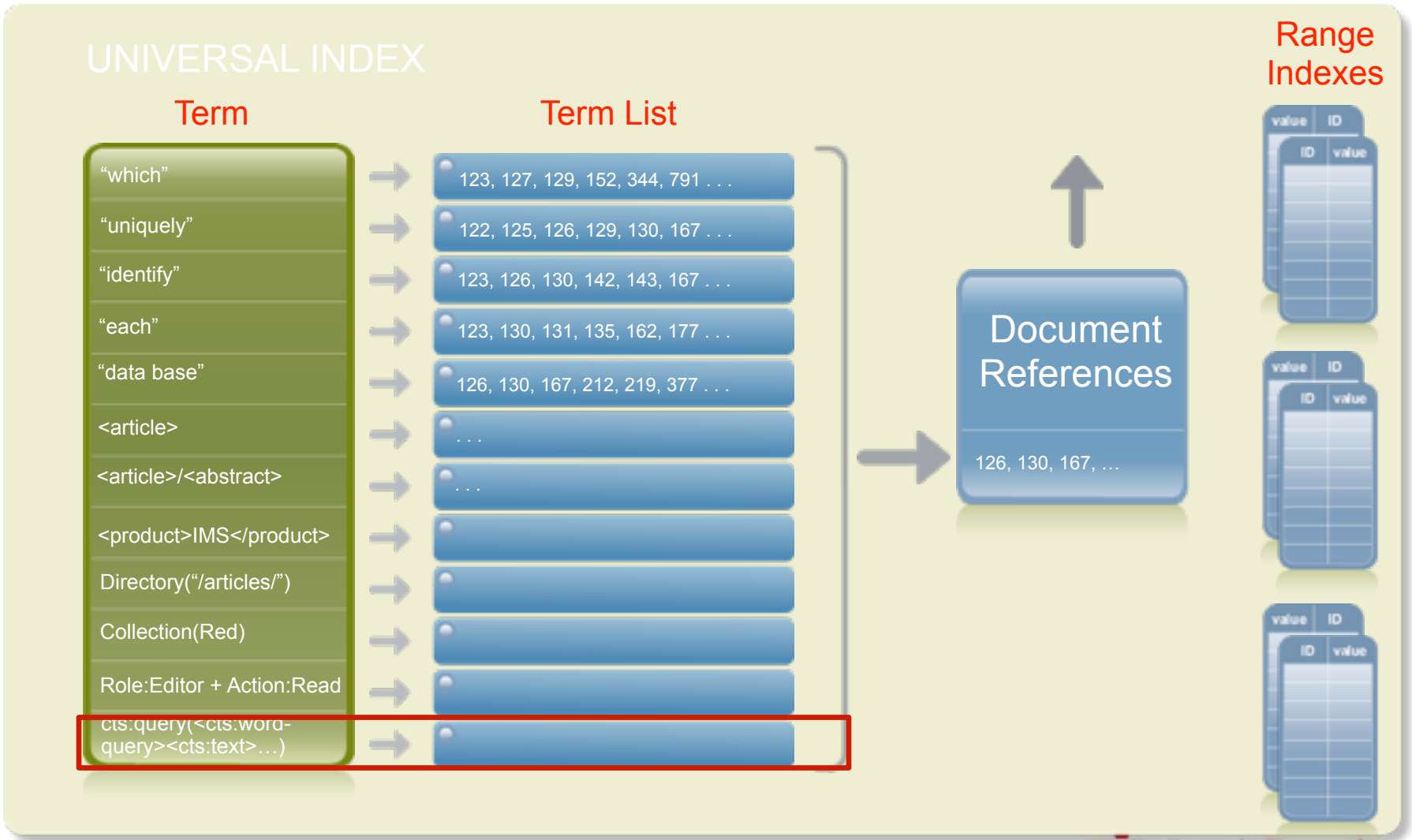
MarkLogic

Spatial Query

- Data examples
  - Latitude / Longitude
  - Any other pair (e.g. volume / price)
- Query types
  - Point (exact value)
  - Point-Radius (circle)
  - Lat/Lon bound (Mercator "rectangle")
  - Polygon (10K+ vertices)
- Composition with…
  - Full Text
  - XML structure
  - XML semantics
  - Other range indexes (e.g. temporal)

MarkLogic™

Registered Query



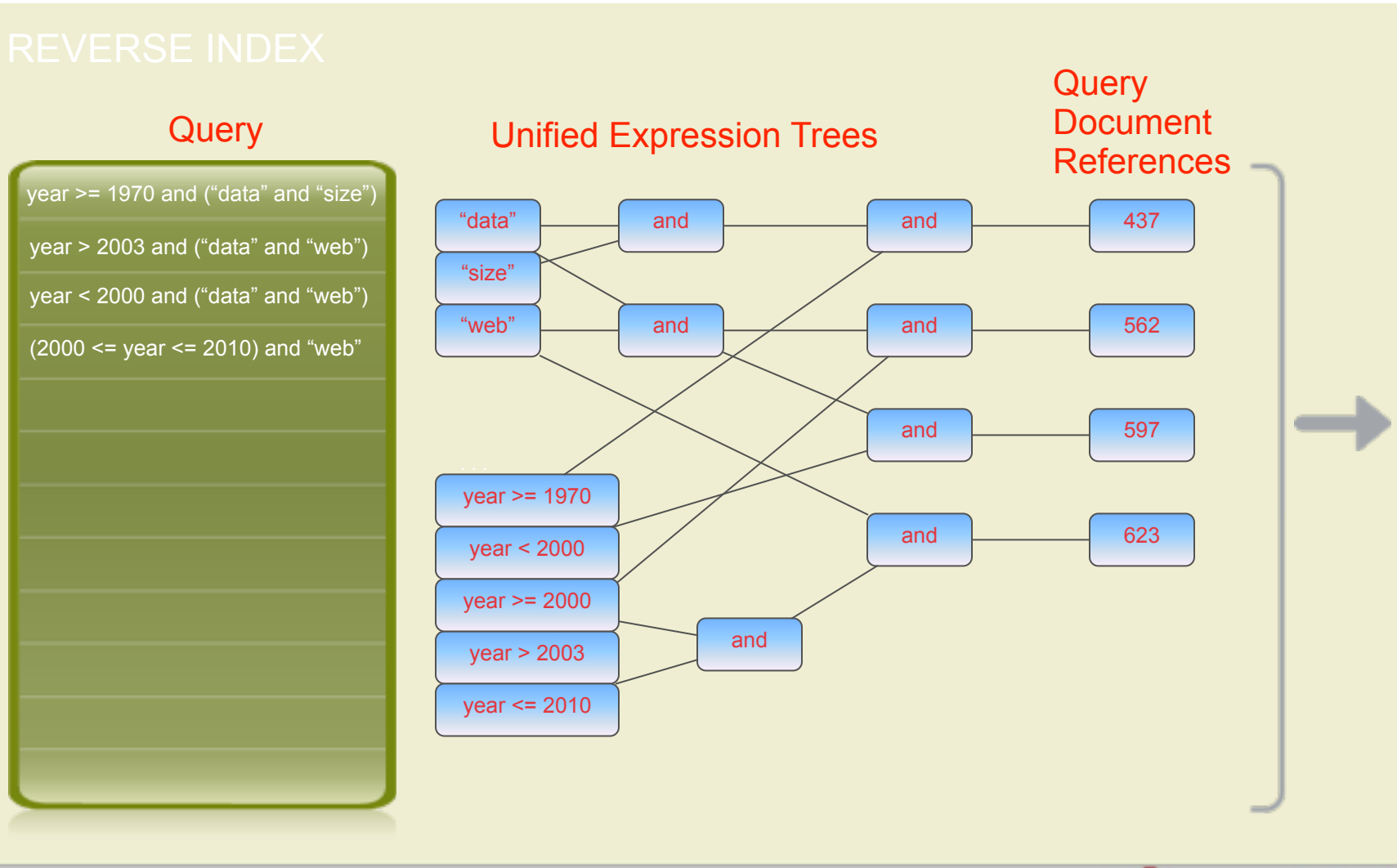
MarkLogic

Reverse Query -- "Alerting"

- Instead of matching documents, you match queries
- Real-time search, selectors, tippers, standing queries, filters, "triggers*", content-based routing, stream DBMS, etc.

MarkLogic

The Reverse Index



MarkLogic

Carpool Matchmaking with Composed Queries

- Driver
  - A non-smoking woman driving from San Ramon to San Calros, leaving at 8am, listens to rock, pop, hip-hop, wants $10 for gas
  - Requires a female passenger within 5 miles of start and end
- Passenger
  - Woman will pay up to $20
  - From: 3001 Summit View Dr, San Ramon, CA 94582
  - To: 400 Concourse Driver, Belmont, CA 94002
  - Requires a non-smoking car
  - Won't listen to country music

MarkLogic

```
let $from := cts:point(37.751658,-121.898387) (: San Ramon :)
let $to := cts:point(37.507363, -122.247119)  (: San Carlos :)
return xdmp:document-insert(
  "/driver.xml",
  <driver>
    <from>{ $from }</from>
    <to>{ $to }</to>
    <when>2010-01-20T08:00:00-08:00</when>
    <gender>female</gender>
    <smoke>no</smoke>
    <music>rock, pop, hip-hop</music>
    <cost>10</cost>
    <preferences>{
      cts:and-query((
        cts:element-value-query(xs:QName("gender"), "female"),
        cts:element-geospatial-query(xs:QName("from"),
          cts:circle(5, $from)),
        cts:element-geospatial-query(xs:QName("to"), cts:circle(5, $to))
      ))
    }</preferences>
  </driver>)
```

```
xdmp:document-insert(
  "/passenger.xml",
  <passenger>
    <from>37.739976,-121.915821</from>
    <to>37.53244,-122.270969</to>
    <gender>female</gender>
    <preferences>{
      cts:and-query((
        cts:not-query(cts:element-word-query(xs:QName("music"), "country")),
        cts:element-range-query(xs:QName("cost"), "<=", 20),
        cts:element-value-query(xs:QName("smoke"), "no"),
        cts:element-value-query(xs:QName("gender"), "female")
      ))
    }</preferences>
  </passenger>)
```

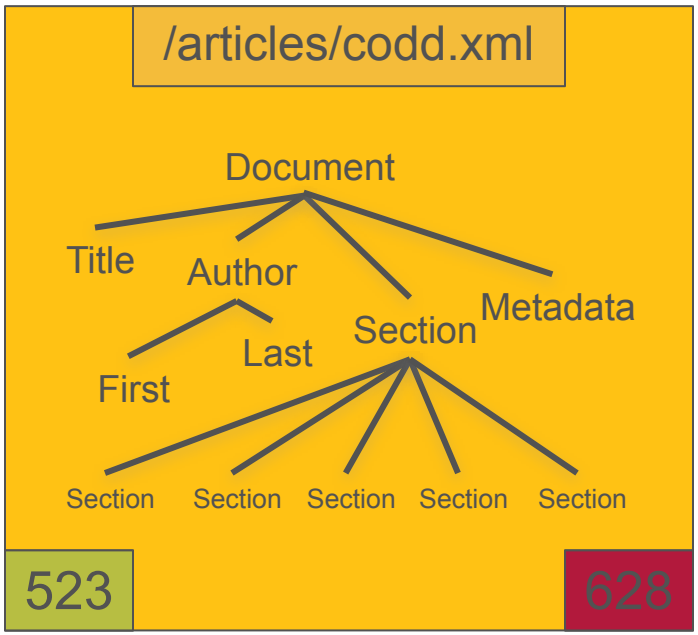
```
(: I'm the driver, find me passengers :)

let $me := doc("/driver.xml")/driver
for $match in cts:search(/passenger,
  cts:and-query((
    cts:query($me/preferences/element()),
    cts:reverse-query($me))
  ))[1 to 3]
return base-uri($match)
```

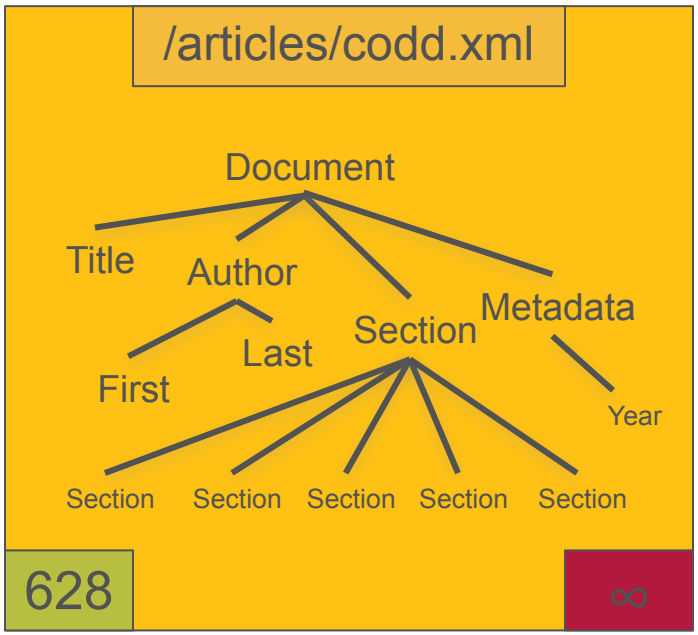
```
(: I'm a passenger, find me a driver :)

let $me := doc("/passenger.xml")/passenger
for $match in cts:search(/driver,
  cts:and-query((
    cts:query($me/preferences/element()),
    cts:reverse-query($me))
  ))[1]
return base-uri($match)
```

Transaction and Storage System

MarkLogic™

Multi-Version Concurrency Control





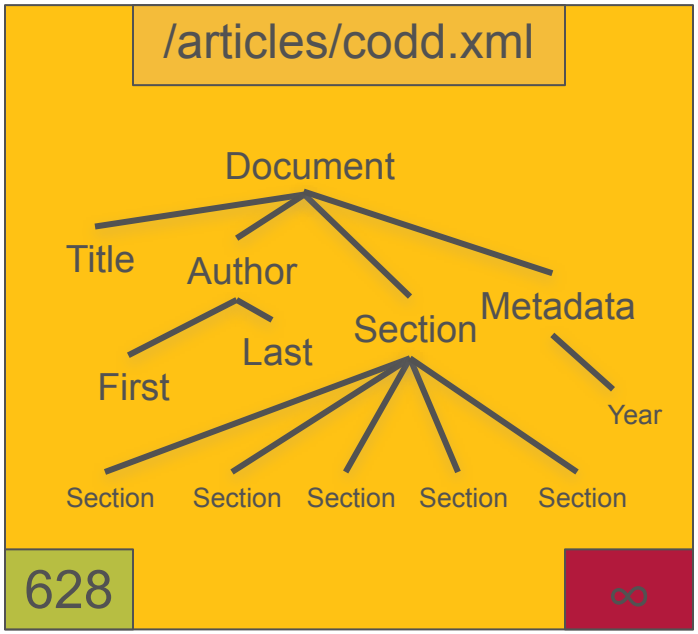
c Creation Timestamp

d Deleted Timestamp

MarkLogic

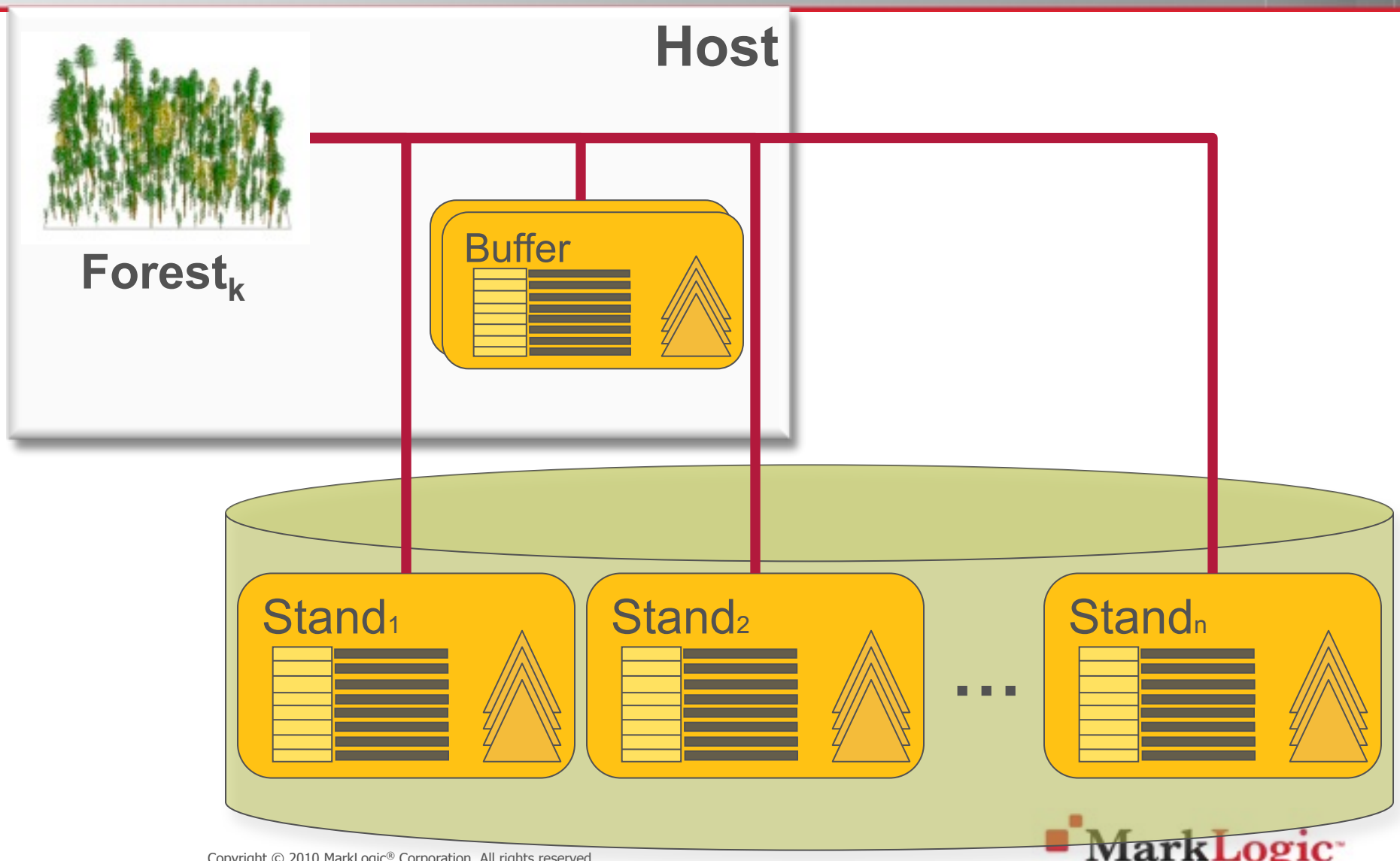
Multi-Version Concurrency Benefits

- High Throughput
  - Queries don't require locks
  - Queries and Updates do not conflict
- ACID
  - Cluster consistency: 2-phase commit
- Zero-latency ingestion and Indexing
  - Append Only
- Ingest/update rates of ~400GB per partition per day



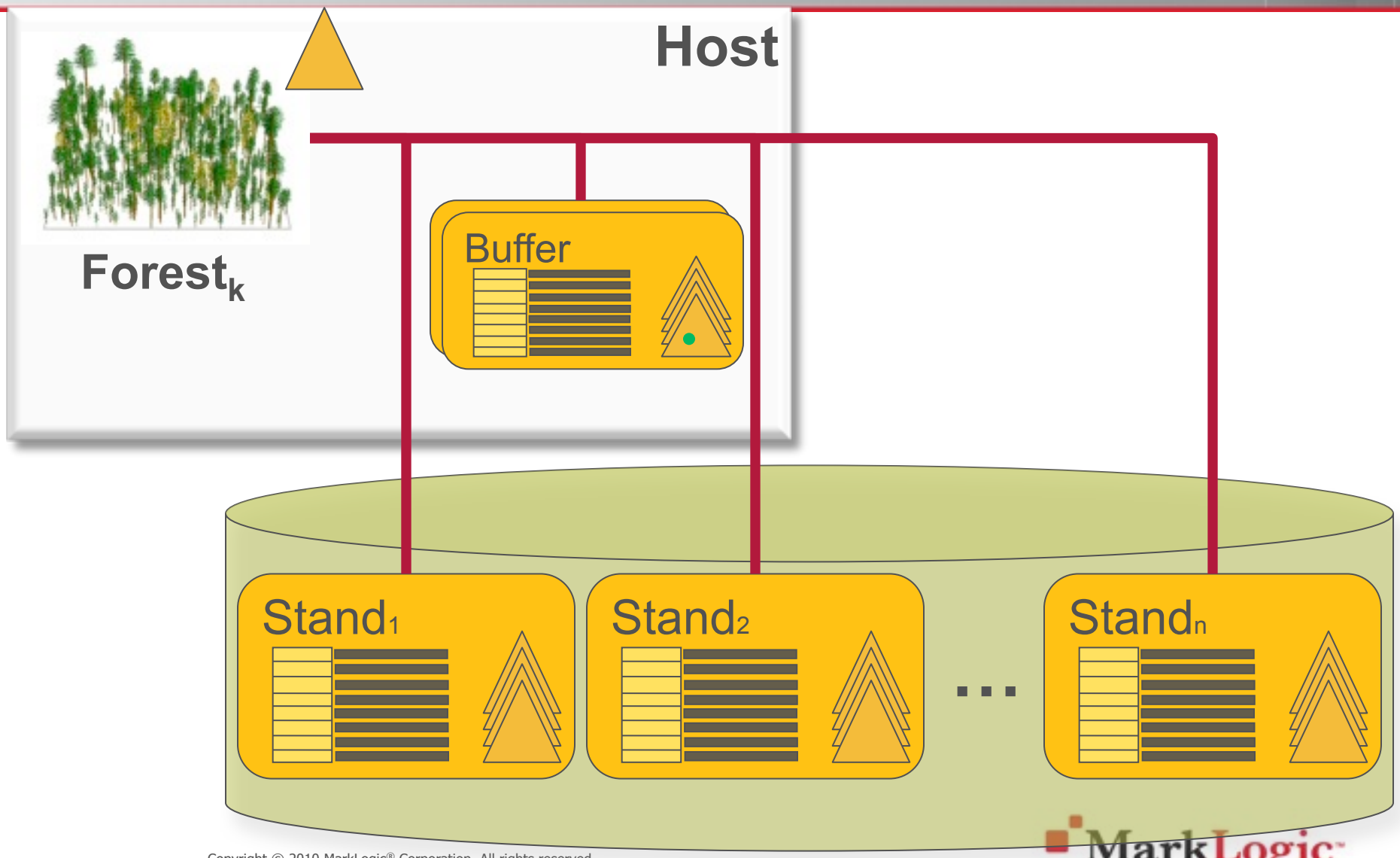
MarkLogic

Forests contain Stands



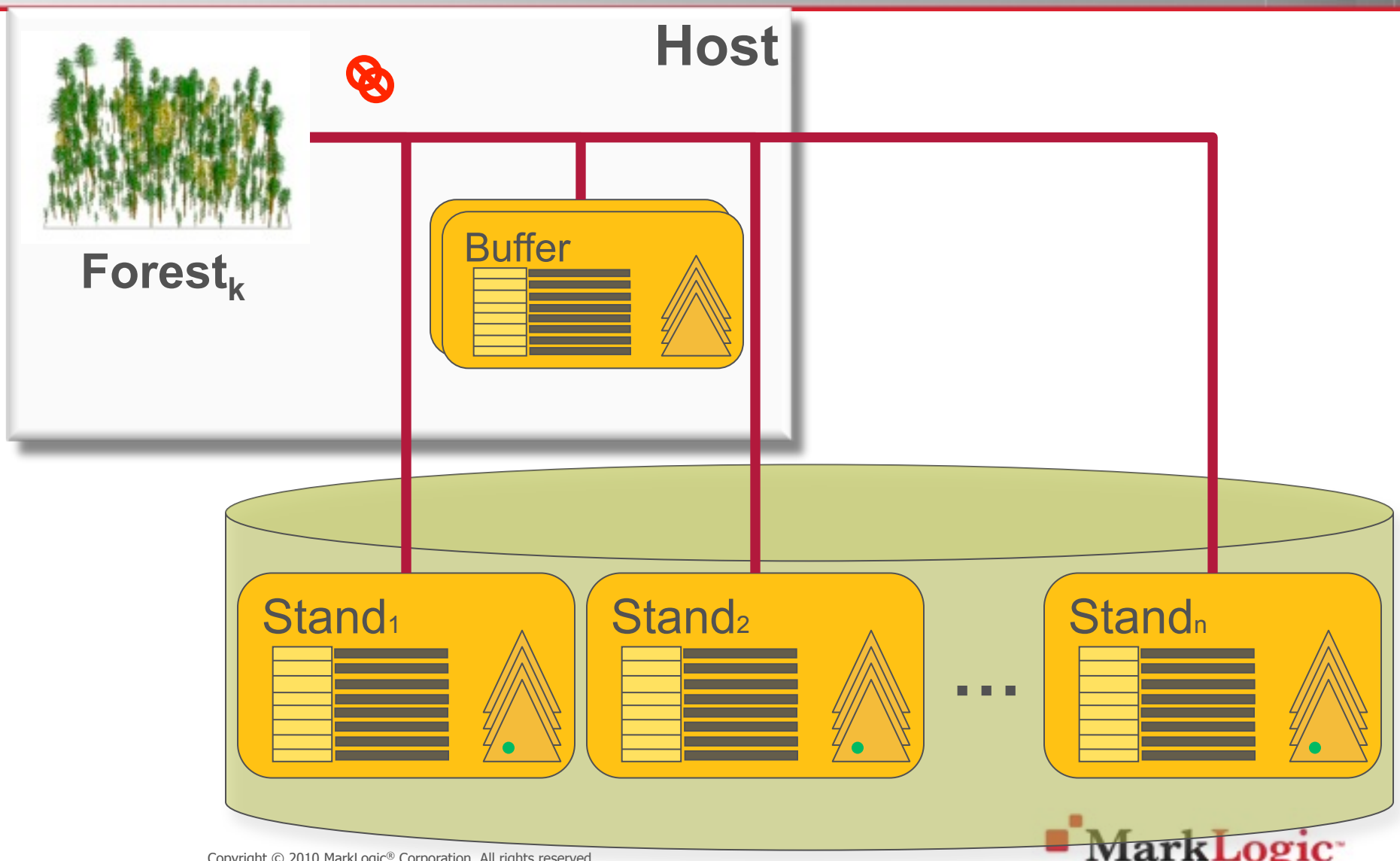
MarkLogic™

1. Create A New Tree

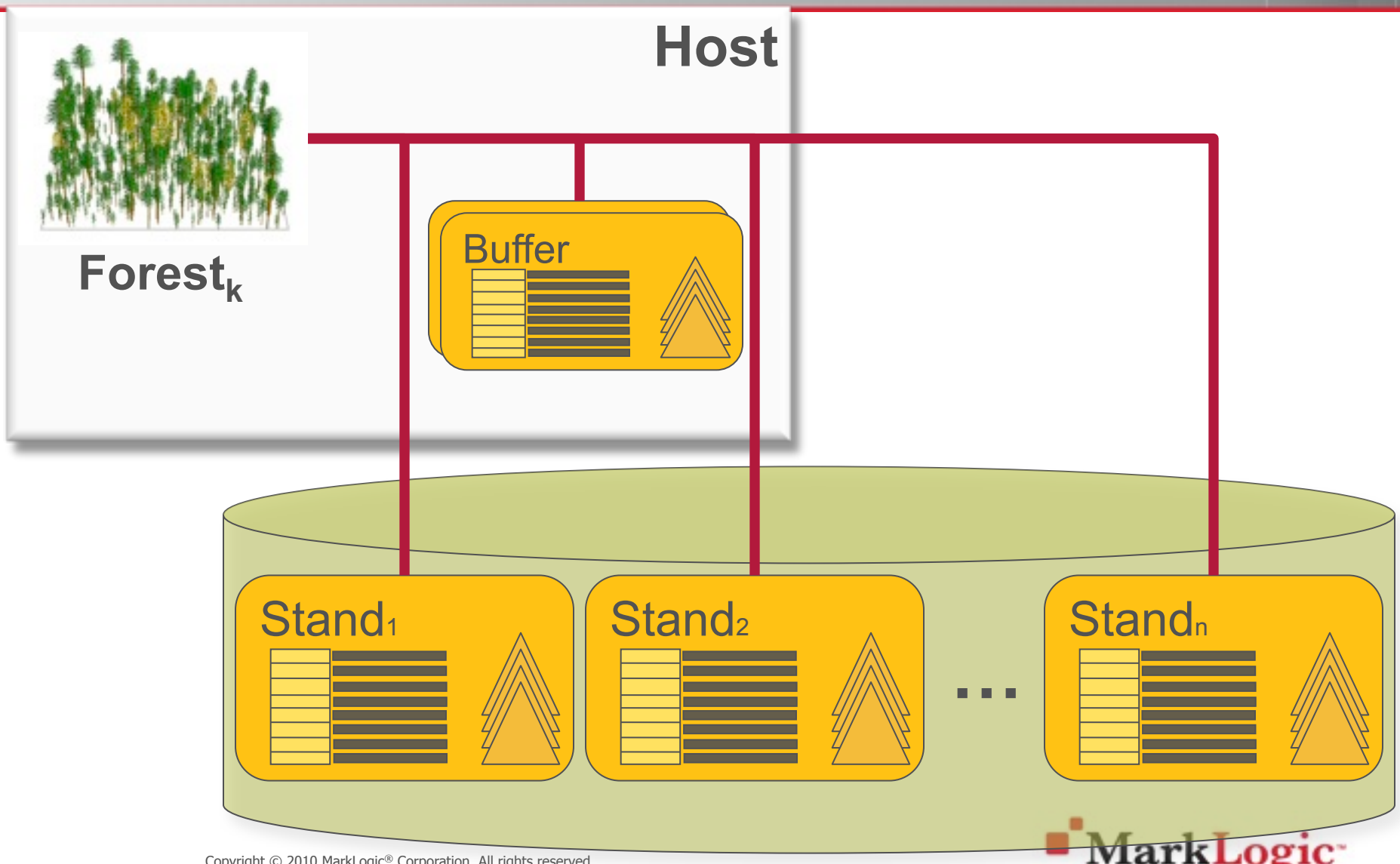


MarkLogic™

2. Expire Trees

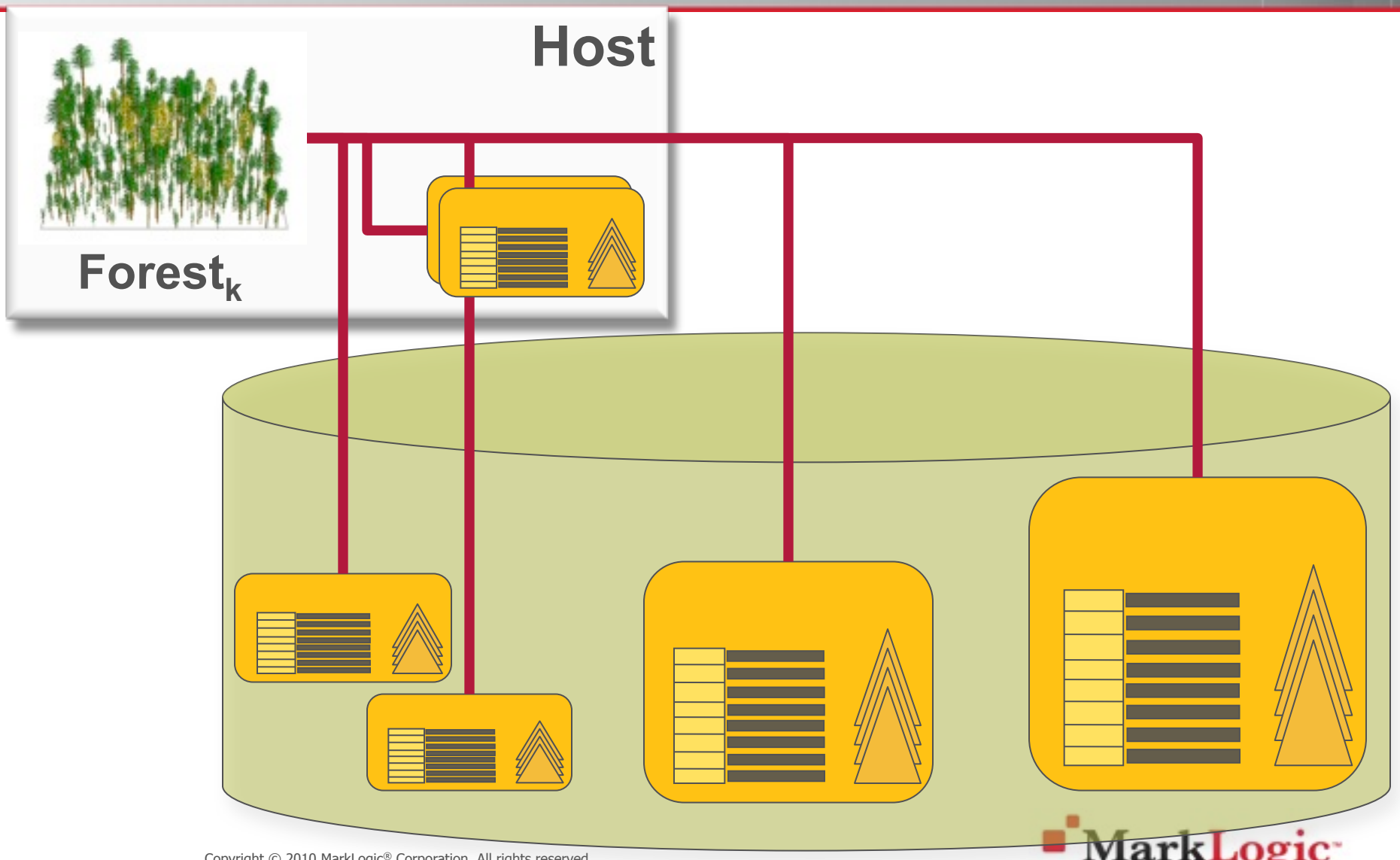


3. Save A Buffer To Disk



MarkLogic™

4. Optimization: Merge Stands



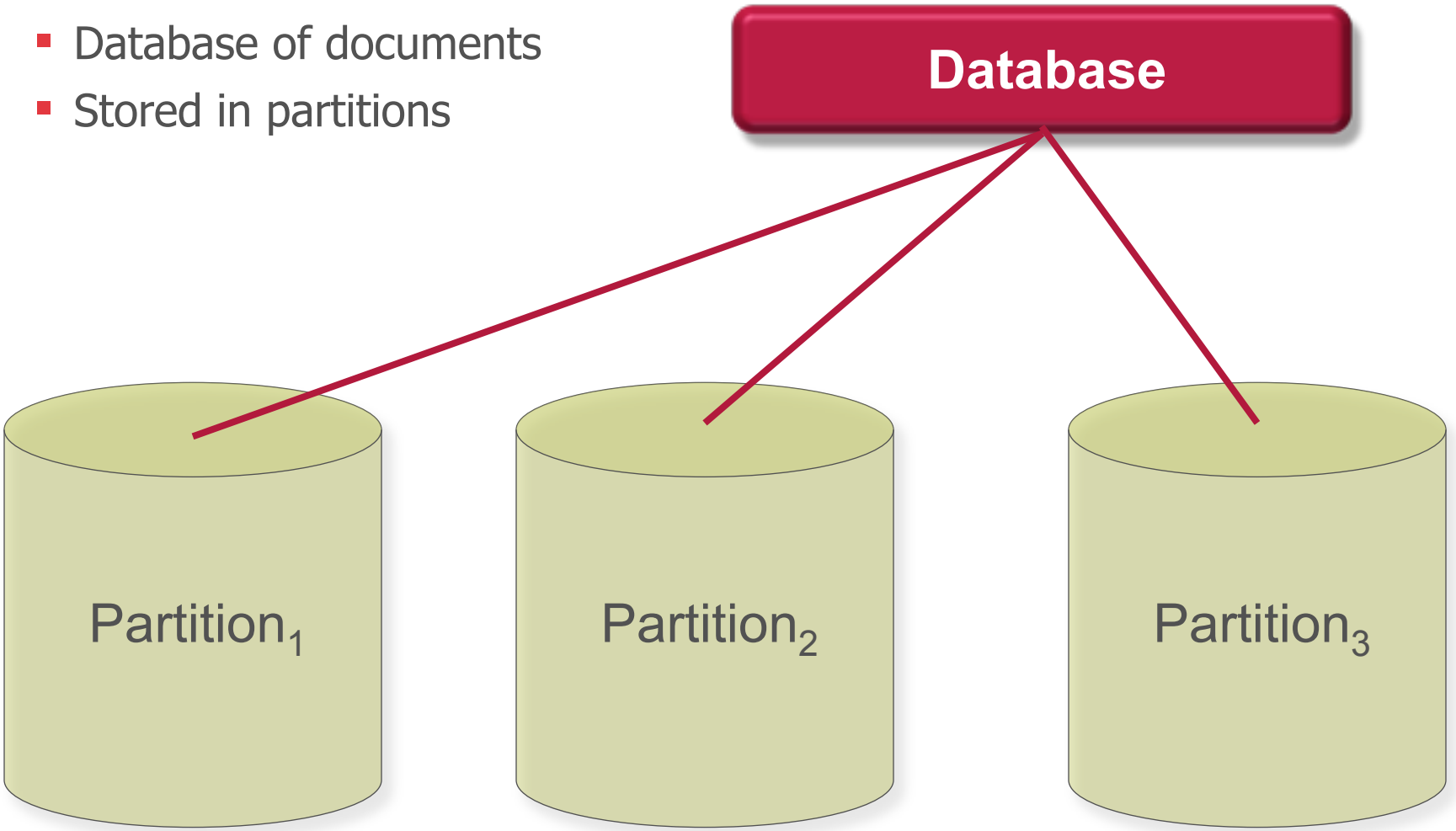
MarkLogic™

Cluster Architecture

MarkLogic™

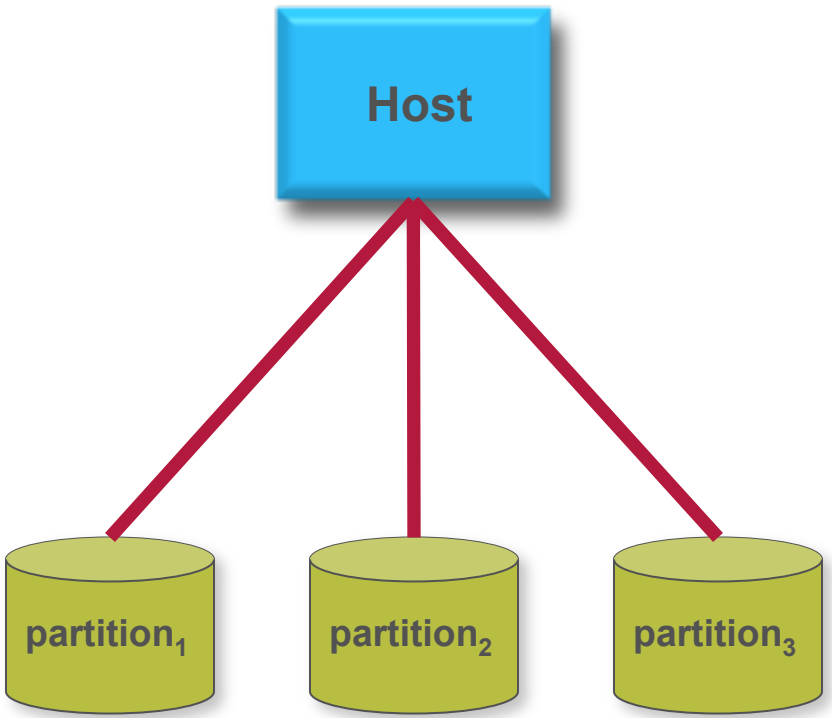
Databases

- Database of documents
- Stored in partitions



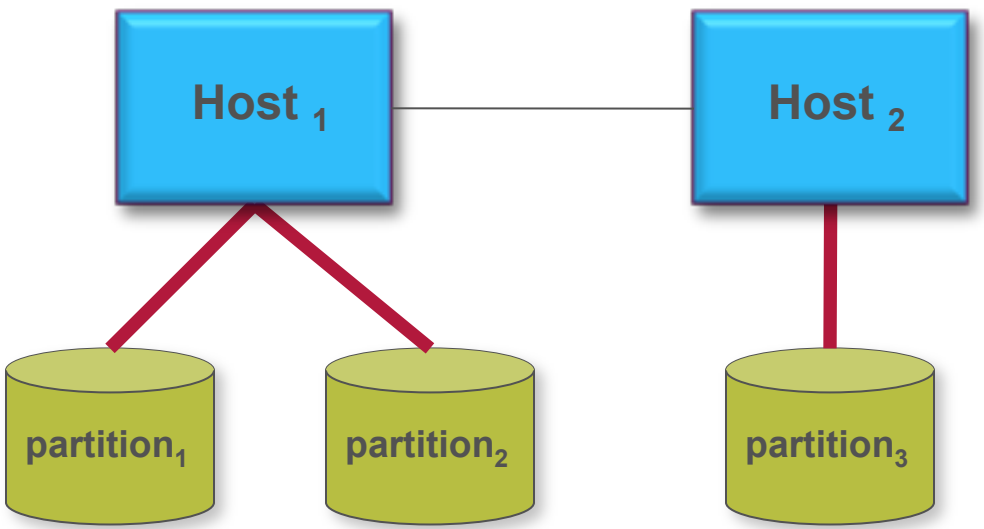
MarkLogic

Simple Architecture



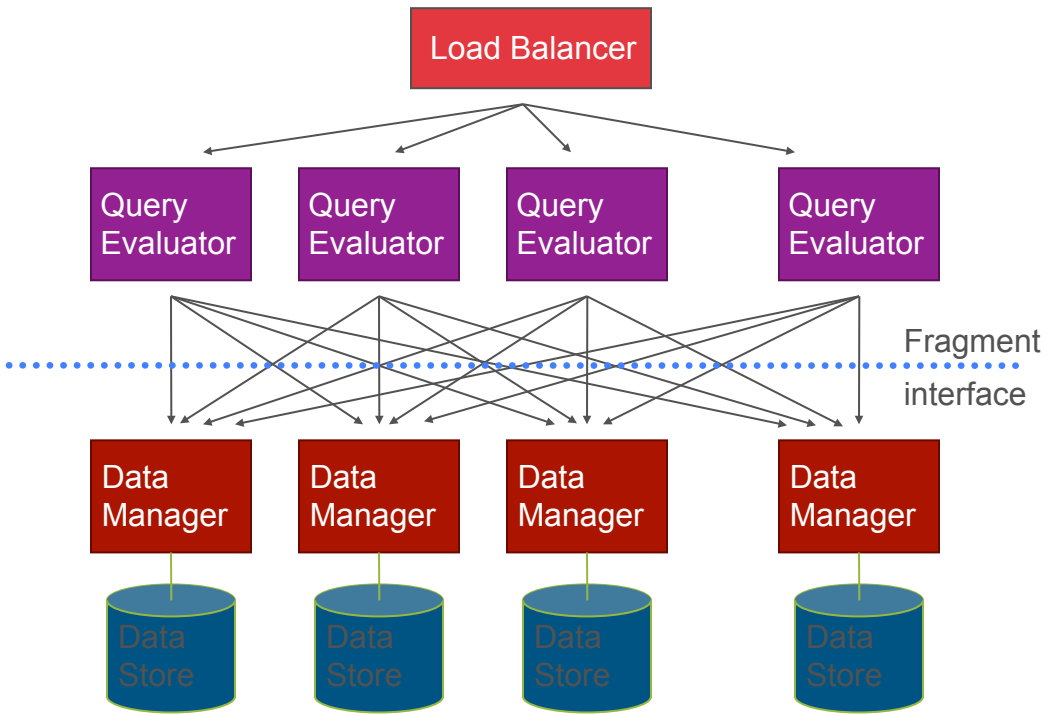
MarkLogic

Shared Nothing Architecture



MarkLogic

Core Technology: Scalability



Increase number of evaluators to scale query processing power

Increase number of data managers to scale data set size

Replicate data managers to scale peak effective I/O rate

MarkLogic™

MarkLogic Server Features

DBMS Features

- Extreme Scalability
- Real-time Transactional Updates
- High-Capacity CRUD
- Geospatial indexing
- Triggers
- Transactional backup
- Replication
- Ease of Administration
- High Availability
- Analytics

Search Features

- Integrated XML and text search
- Faceted Navigation
- Fielded search
- Alerting ("profiling")
- Relevance tuning
- Language processing
- Entity extraction / enrichment
- Foreign language support
- Thesaurus, taxonomy support
- Automatic classification

MarkLogic

Questions?

Jason Hunter
jhunter@marklogic.com

MarkLogic™