Making Sense of NoSQL

Dan McCreary

Wednesday, Nov. 13th 2014

midwest
architecture community
collaboration

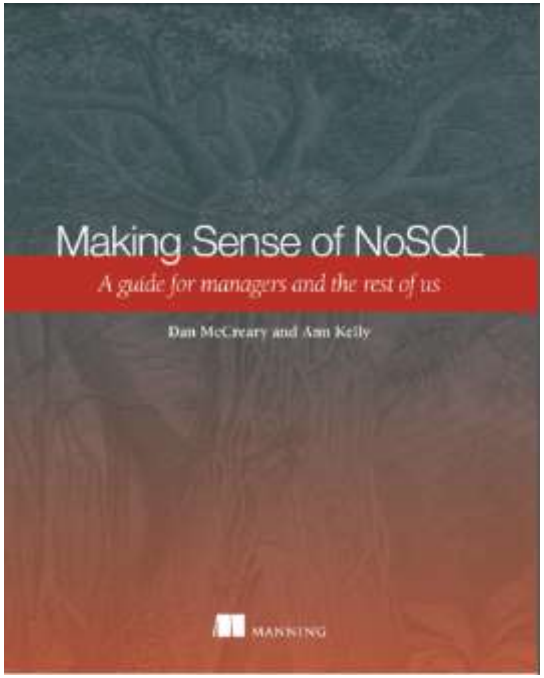
Agenda

- Why NoSQL?
- What are the key NoSQL architectures?
- How are they different from traditional RDBMS Systems?
- What types of problems do they solve?
- How to learn more

Background for Dan McCreary

- Co-founder of the **NoSQL Now!** conference
- Background
  - Bell Labs
  - NeXT Computer (Steve Jobs)
  - Owner of 75-person software consulting firm
  - US Federal data integration (National Information Exchange Model NIEM.gov)
  - Native XML/XQuery for metadata management since 2006
  - Advocate of web standards, NoSQL and XRX systems
  - As of Monday – Principal Engineer with MarkLogic

Making Sense of NoSQL



- Coauthor (with Ann Kelly) of "Making Sense of NoSQL"
- Guide for managers and architects
- Focus on NoSQL architectural tradeoff analysis
- Basis for **40 hour course** on database architectures
- http://manning.com/mccreary

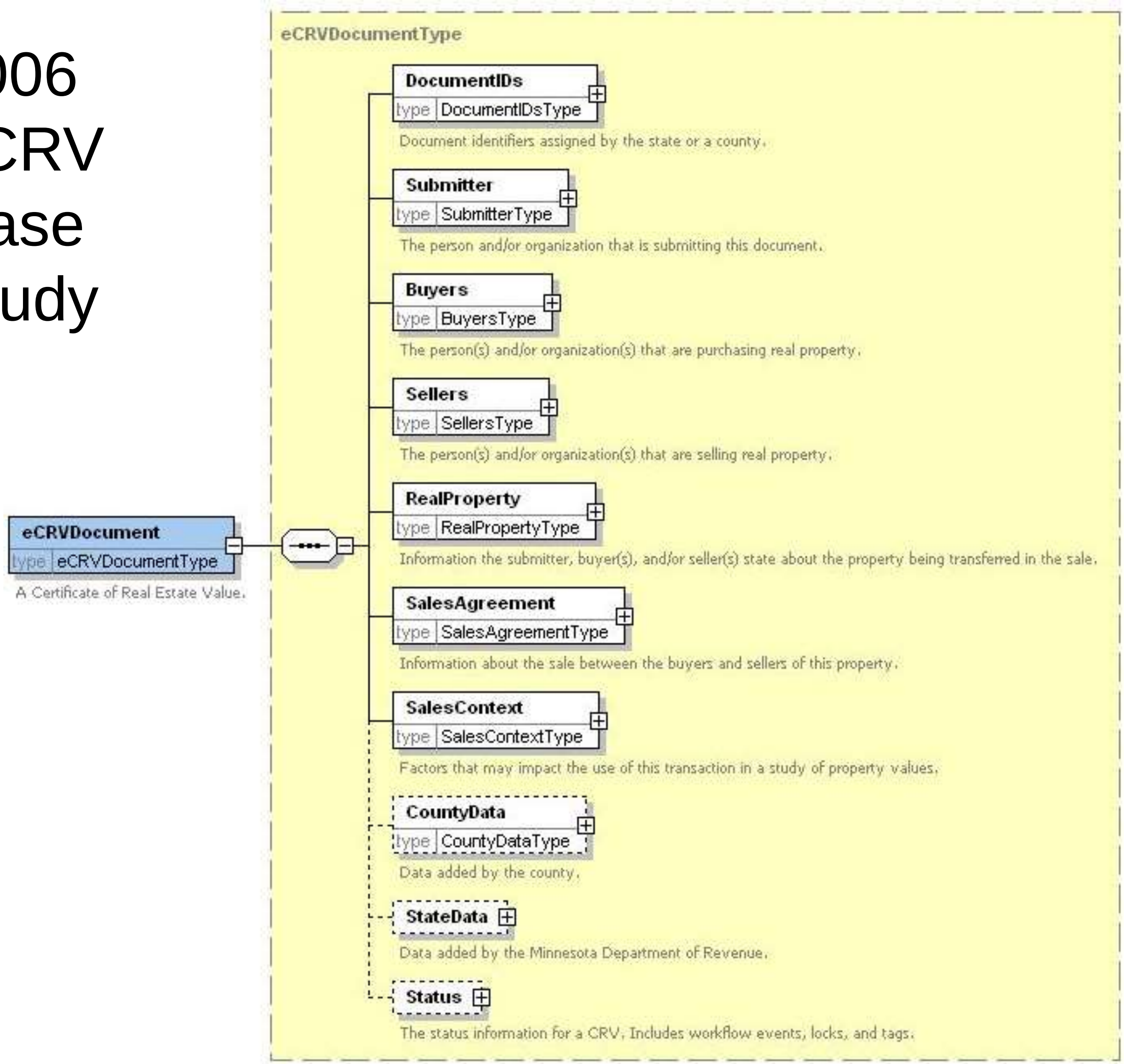
The Story of Property Tax Forms

Alex Bleasdale

Arun Batchu

- How did I get into NoSQL?
- In 2006 a state agency in Minnesota wanted to standardize property tax records across 87 counties
- A story about standards
  - XML, XML Schema, XForms, XQuery, NEIM
- A story about agility
- A story about new technology adoption

2006 eCRV Case Study



MINNESOTA·REVENUE

Property Taxation Metadata Registry

Metadata Registry Home | About | Feedback | Links | Bibliography

Metadata Registry

Browse Metadata Registry by Data Element Name

- Glossaries
- Browse Elements
- XML Schemas
- Links
- Metrics

#	Data Element Name	Status	Primary Owner Team
1	Activity	assigned-to-review-team	NIEM-universal
2	ActivityDate	assigned-to-review-team	NIEM-universal
3	ActivityEndDate	initial-draft	DataStandards
4	ActivityFederalFiscalYear	initial-draft	DataStandards
5	ActivityFiscalYear	initial-draft	DataStandards
6	ActivityStartDate	initial-draft	DataStandards
7	ActivityStateFiscalYearNumber	initial-draft	DataStandards
8	Address	assigned-to-review-team	NIEM-universal
9	AddressCityName	initial-draft	DataStandards
10	AddressLine1Text	initial-draft	DataStandards
11	AddressLine2Text	initial-draft	DataStandards
12	AddressPostalCodeID	initial-draft	DataStandards
13	AddressStateCode	initial-draft	DataStandards
14	AngularMinute	initial-draft	DataStandards
15	AngularSecond	initial-draft	DataStandards
16	Contact	initial-draft	DataStandards
17	ContactEmailID	initial-draft	DataStandards
18	ContactFAXText	initial-draft	DataStandards
19	ContactPhoneText	initial-draft	DataStandards
20	CRV	initial-draft	CRV
21	CRVAdjustmentCode	initial-draft	CRV
22	CRVCountyAuditorID	initial-draft	CRV

Metadata Registry

- Glossaries
- Browse Elements
- XML Schemas
- Links
- Metrics

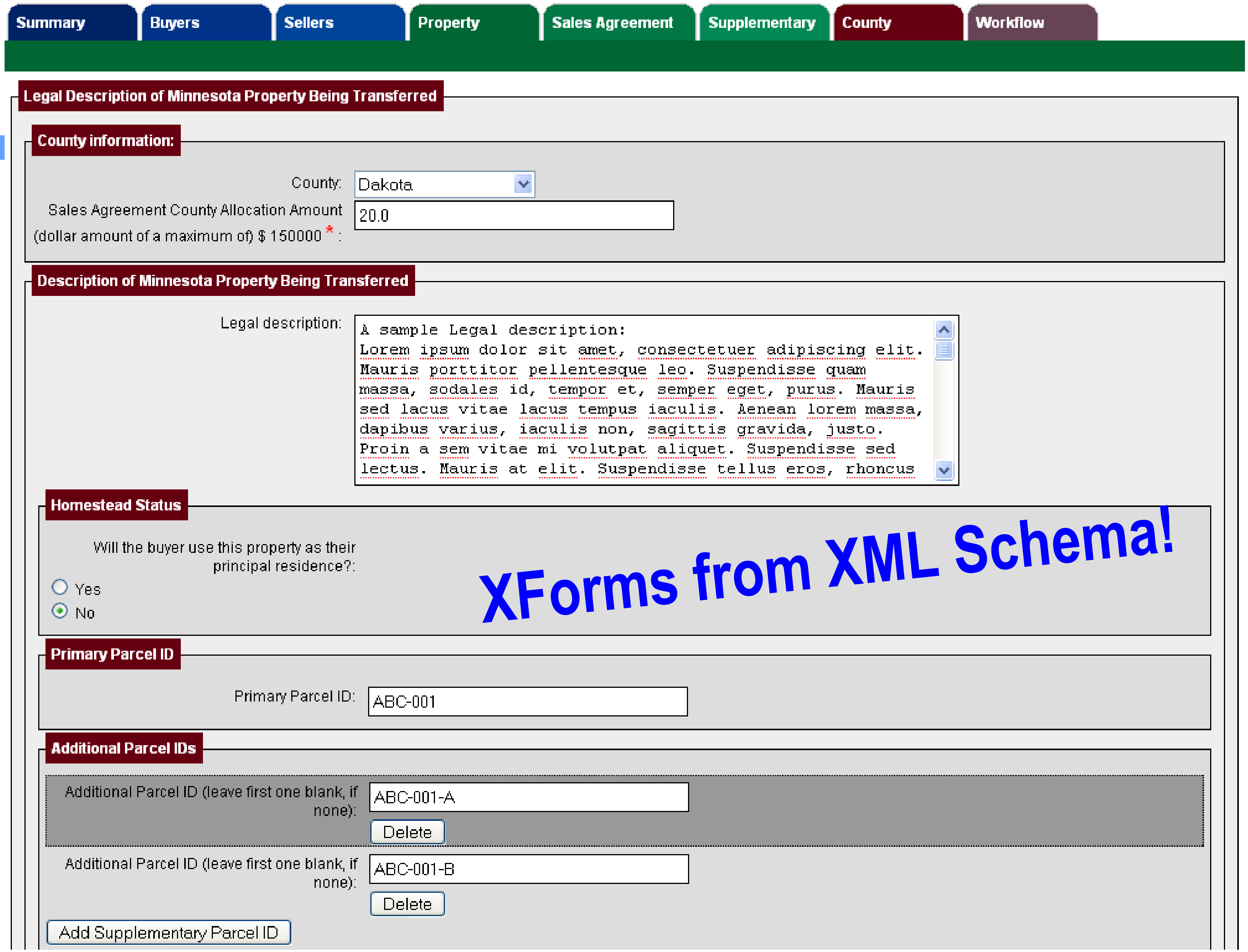
CRV

Approval Status: initial-draft
Primary Owner Team: CRV

ISO Name Components

Object: CRV

Definition:	A certificate of real estate value document that must be filed with the county auditor whenever real property valued over $1,000 is sold or conveyed in Minnesota.
Complex:	true
Subclass Of:	DocumentForm
Abbreviation:	CRV
Screen Label:	CRV
Metadata Source:	Back of form PE20, Minnesota Department of Revenue glossary
Usage:	The Minnesota Department of Revenue uses information on the CRV to determine if assessors through Minnesota are valuing property according to the same standards, and to determine how much state aid will go to all school districts and cities in the next year. The value of the real property in ache school district and city affects the amount of financial aid the state will provide.
Referenced in:	DataStandards
General Note:	Information reported on the CRV includes the sales price, the value of any personal property, if any, included in the sale, and the financial terms of the sale. The CRV is eventually filed with the Property Tax Division of the Department of Revenue. The deed types must be warranty deed, contract for deed, quit claim deed, trustee deed, executor deed or probate deed. If the value of the property is less than $1,000 the deed must have the following written on that back: The sale price or other consideration given for this property was $1,000 or less.
Enumerated:	false


Summary
Buyers
Sellers
Property
Sales Agreement
Supplementary
County
Workflow
Legal Description of Minnesota Property Being Transferred
County information:
County: Dakota
Sales Agreement County Allocation Amount (dollar amount of a maximum of) $ 150000 * : 20.0
Description of Minnesota Property Being Transferred
Legal description:
A sample Legal description:
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Mauris porttitor pellentesque leo. Suspendisse quam massa, sodales id, tempor et, semper eget, purus. Mauris sed lacus vitae lacus tempus iaculis. Aenean lorem massa, dapibus varius, iaculis non, sagittis gravida, justo. Proin a sem vitae mi volutpat aliquet. Suspendisse sed lectus. Mauris at elit. Suspendisse tellus eros, rhoncus
Homestead Status
Will the buyer use this property as their principal residence?:
Yes
No
XForms from XML Schema!
Primary Parcel ID
Primary Parcel ID: ABC-001
Additional Parcel IDs
Additional Parcel ID (leave first one blank, if none): ABC-001-A
Delete
Additional Parcel ID (leave first one blank, if none): ABC-001-B
Delete
Add Supplementary Parcel ID

MINNESOTA E ■ CRV Project

You are currently logged in as **dakota** of **dakota** and you have the role[s] of: **auditor assessor supervisor auditor-dakota**

Minnesota Certificate of Real Estate Valuation Form - editing CRV# 19-08-49

Auditor ID# DAK-09821 Save CRV

County Edit Form Version 1.0 created on December 17th, 2007

Back to Dashboard (abandon edit) | Save form information (not available for demo) | Send feedback to the team

CRV Status and lock preview

County Accepted:	☑
Auditor Complete:	☐
Assessor Complete:	☐
County Final (and Lock):	☐
State Review Complete:	☐
State Study Lock:	☐
County Edit Lock:	☐
State Edit Lock:	☐

Summary | Buyers | Sellers | Property | Sales Agreement | Supplementary | County | Workflow

Current Workflow for this CRV Document

ID	Date/Time	User	Org	Code	Activity	Comment
1	2008-01-14, 08:42	anonymous	anonymous	original		
2	2008-01-14, 08:44	dakota	dakota	county-accepted		County Accepted CRV
3	2008-01-14, 08:45	dakota	dakota	activity		CRV edited and saved
4	2008-01-14, 08:47	dakota-as	dakota	assessor-assigned		assigning to assessor
5	2008-01-15, 10:39	dakota	dakota	activity		CRV edited and saved

Add new Workflow event

CRV Workflow Code:

County Code Text (only relevant on county code selected):

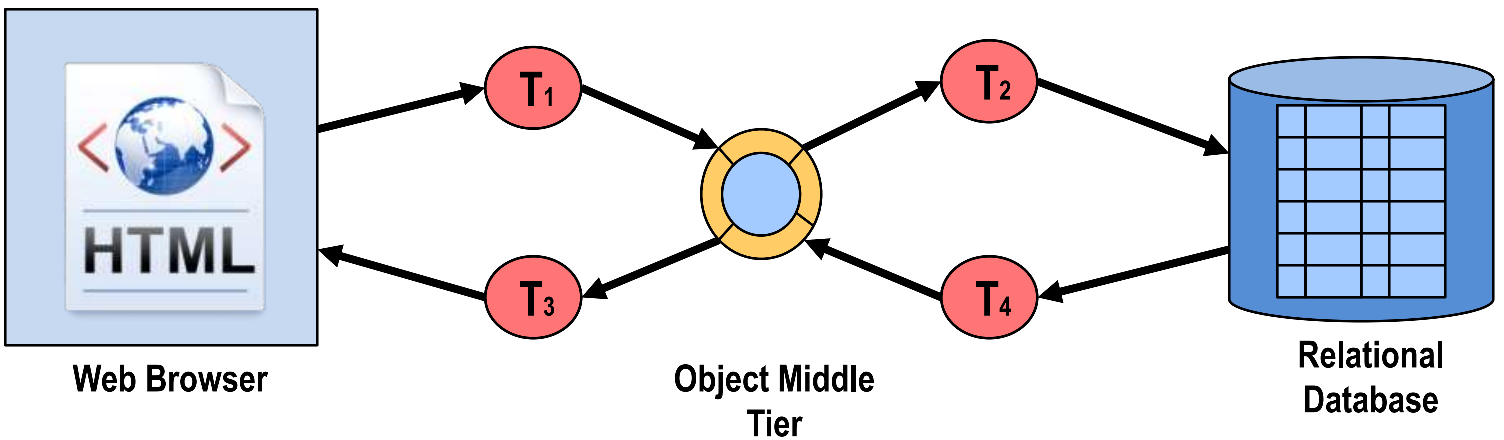
Comment:

Add Workflow Code

Tags

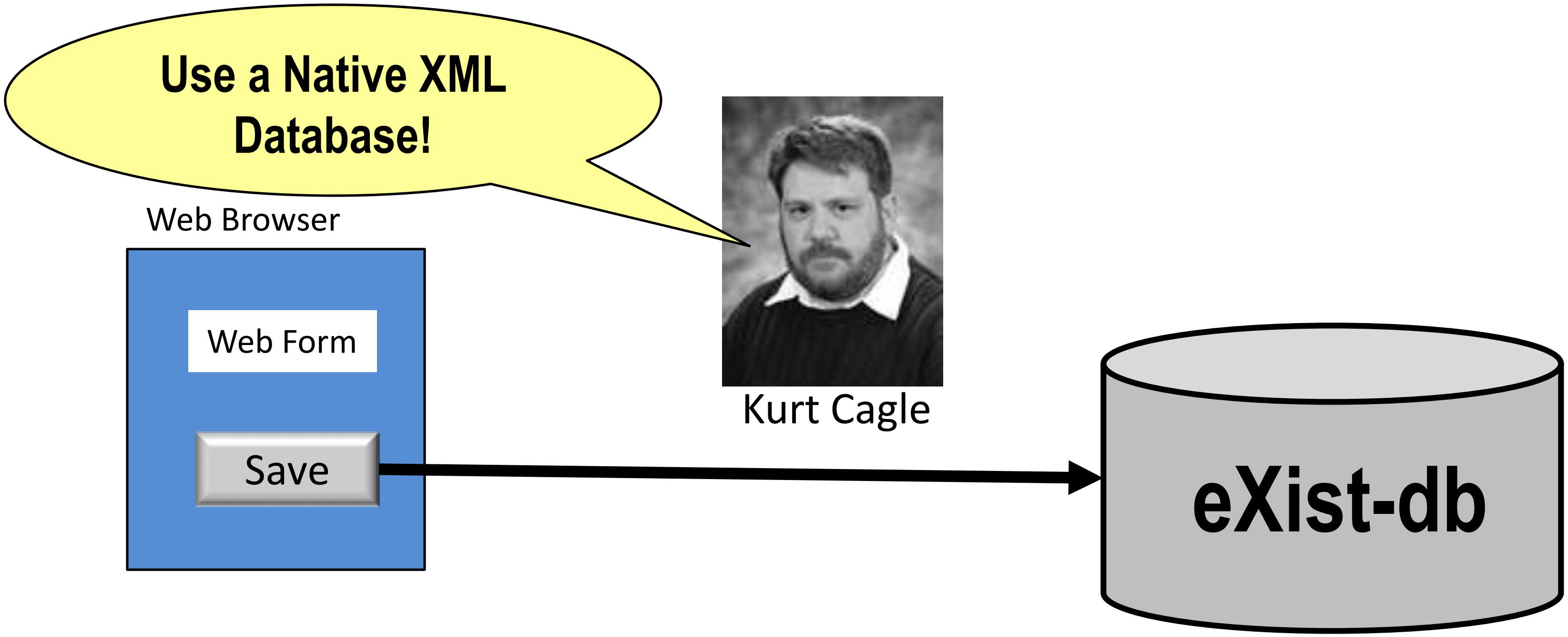
ID	Org	Date/Time	Keyword	Comment	Work
dakota	dakota	2008-01-14, 08:44	county-accepted	CRV has been accepted and automatically tagged by the accept process	2

Four Translations



- T_1 – HTML into Java Objects
- T_2 – Java Objects into SQL Tables
- T_3 – Tables into Objects
- T_4 – Objects into HTML

Kurt's Suggestion




```
store($collection, $file-name, $data)
```

Equivalent of 45 SQL inserts in 1 line of code!

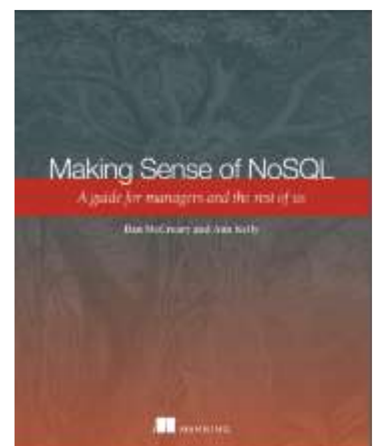
Zero Translation

- XML lives in the web browser (**X**Forms)
- **R**EST interfaces
- XML in the database (Native XML, **X**Query)
- **XRX** Web Application Architecture
- No "impedance mismatch", No translation!
- Department tried it and then went back to HTML, Java and SQL
- *…but I was forever changed…I began to question everything I had been taught about databases*

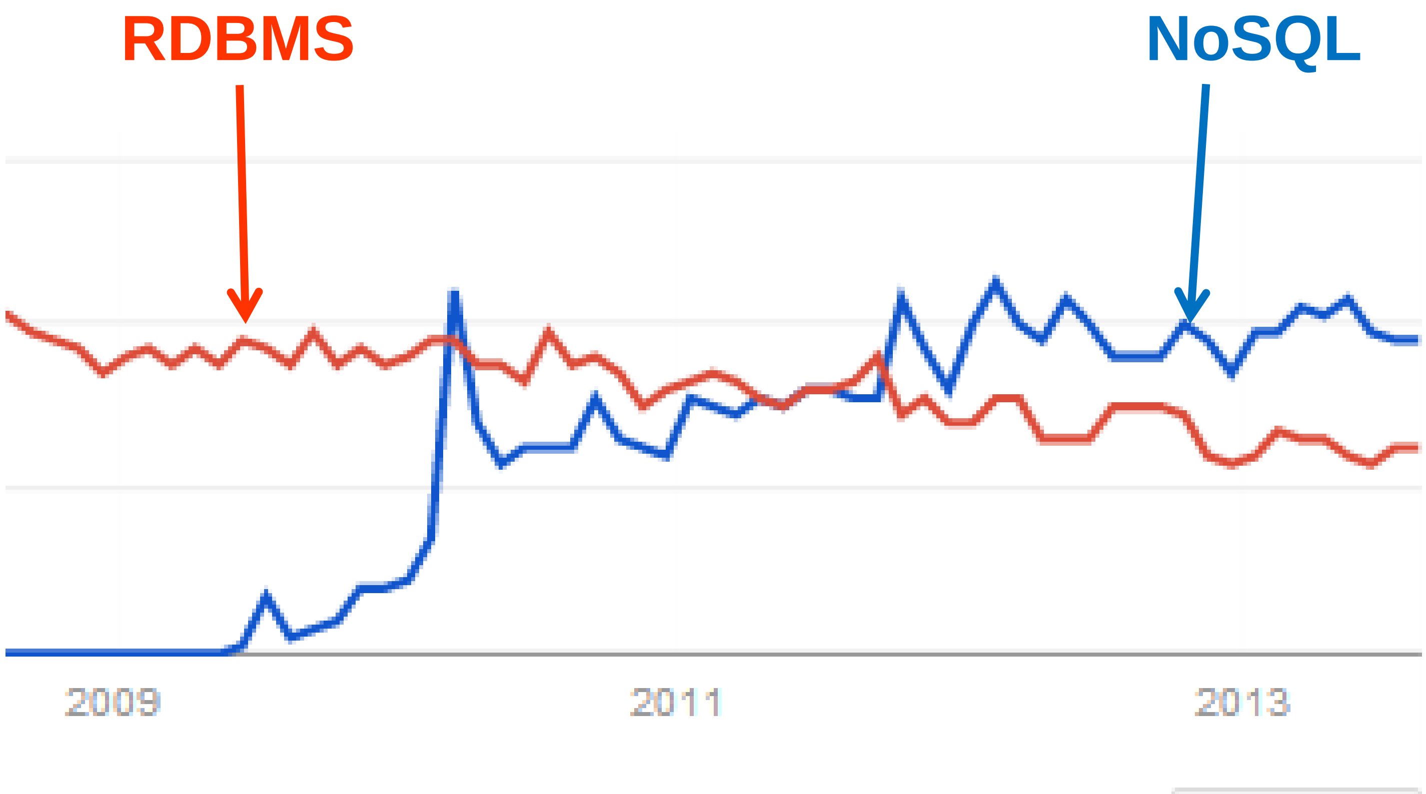
Anger, Wiki, Conference, Book

2011, 2012, 2013, 2014

2014 NOSQL NOW! AUGUST 19-21 | SAN JOSE



NoSQL on Google Trends



http://www.google.com/trends/explore?q=NoSQL%2C+RDBMS#q=NoSQL%2C%20RDBMS&cmpt=q

The NO-SQL Universe

Key-Value Stores

riak

FOUNDATIONDB

redis

AEROSPIKE

Document Stores

Couchbase

mongoDB

CouchDB relax

RAVENDB

Column-Family Stores

MAPR

HBASE

Cassandra

Graph/Triple Stores

neotechnology

InfiniteGraph

AllegroGraph® RDFStore

bigdata®

existdb

MarkLogic

BaseX

XML

Sample of NoSQL Jargon

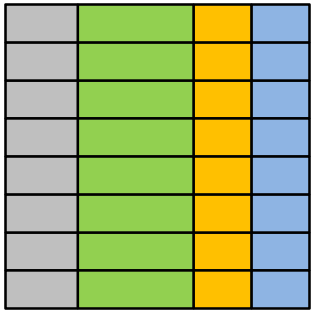
Document orientation
Schema free
MapReduce
Horizontal scaling
Sharding and auto-sharding
Brewer's CAP Theorem
Consistency
Reliability
Partition tolerance
Single-point-of-failure
Object-Relational mapping
Key-value stores
Column stores
Document-stores
Memcached

Indexing
B-Tree
Configurable durability
Documents for archives
Functional programming
Document Transformation
Document Indexing and Search
Alternate Query Languages
Aggregates
OLAP
XQuery
MDX
RDF
SPARQL
Architecture Tradeoff Modeling
ATAM
Erlang

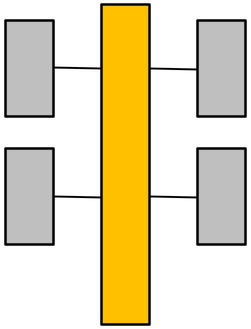
Note that within the context of NoSQL many of these terms have different meanings!

Before NoSQL

Relational

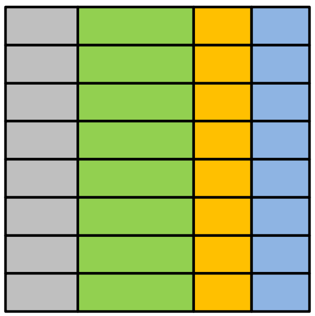


Analytical (OLAP)

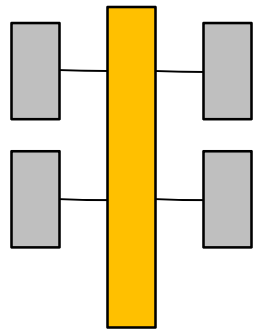


After NoSQL

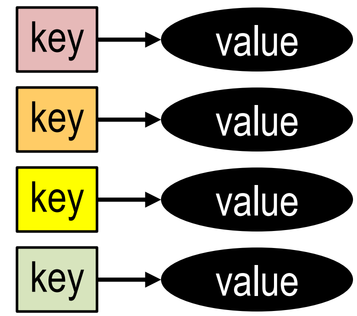
Relational



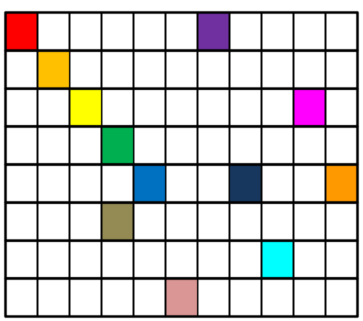
Analytical (OLAP)



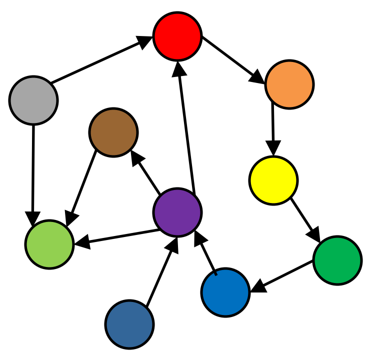
Key-Value



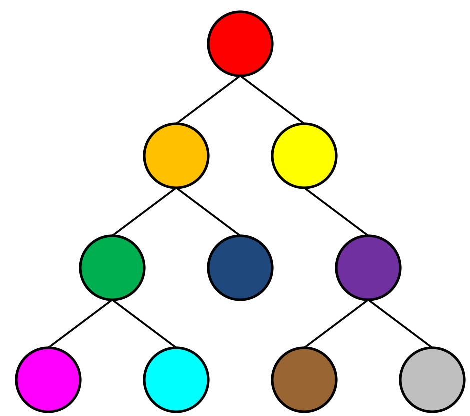
Column-Family



Graph



Document



Food for thought…

- What percentage of database transactions run on **RDBMSs** in the following organizations?

- What percentage of all transactions in **Minnesota** run on RDBMSs?
- Why is this number different?
- Is our data fundamentally different?

NoSQL – The Big Tent

http://www.flickr.com/photos/morgennebel/2933723145/

- **"NoSQL"** – a label for a "meme" that now encompasses a large body of innovative ideas on data management
- "Not Only SQL"
- Focus on non-relational databases and hybrids
- A community where new ideas are quickly recombined to create innovative new business solutions

Historical Context

Mainframe Era

- 1 CPU
- COBOL and FORTRAN
- Punchcards and flat files
- $10,000 per CPU hour

Commodity Processors

- 10,000 CPUs
- Functional programming
- MapReduce "farms"
- Pennies per CPU hour

Two Approaches to Computation

1930s and 40s

John von Neumann

Alonzo Church

*Manage **state** with a program counter.*

*Make computations act like math **functions**.*

Which is simpler? Which is cheaper? Which will scale to 10,000 CPUs?

Standard vs. MapReduce Prices

John's Way Alonzo's Way

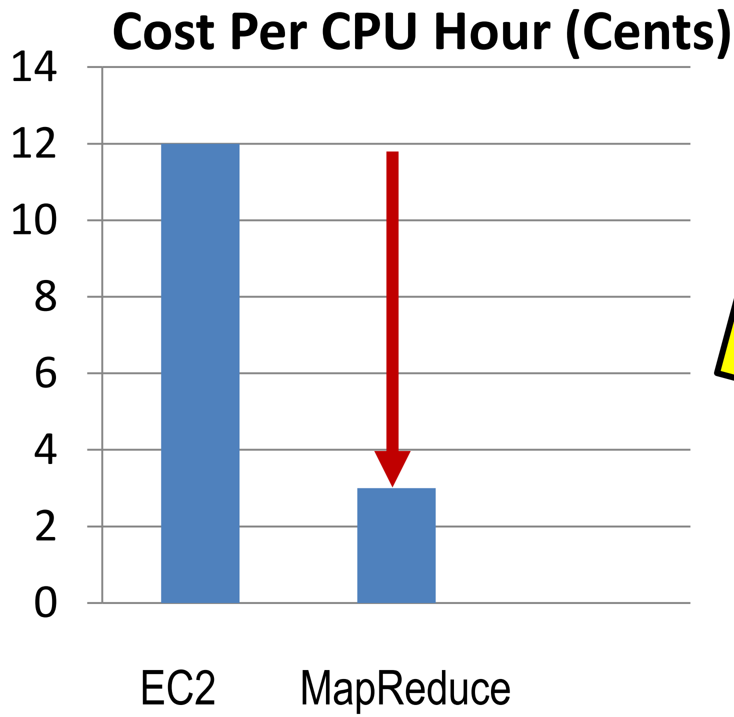
Pricing for Amazon EC2 (On-Demand) and Amazon Elastic MapReduce

Region: US East (N. Virginia)

	Amazon EC2 Price	Amazon Elastic MapReduce Price
Standard On-Demand Instances		
Small (Default)	$0.06 per hour	$0.015 per hour
Medium	$0.12 per hour	$0.03 per hour
Large	$0.24 per hour	$0.06 per hour
Extra Large	$0.48 per hour	$0.12 per hour

http://aws.amazon.com/elasticmapreduce/#pricing

MapReduce CPUs Cost Less!



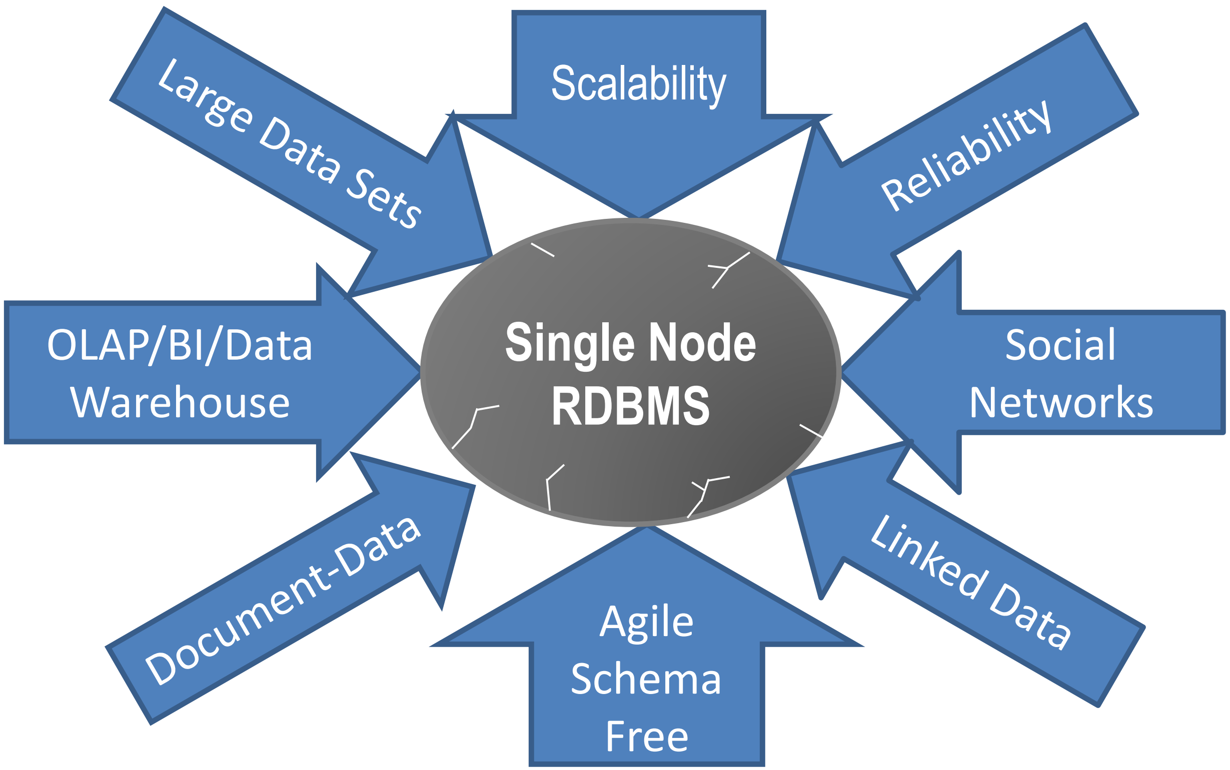
75% Cost Reduction!

Cut cost from 12 to 3 cents per CPU hour!
Perhaps Alonzo was right!

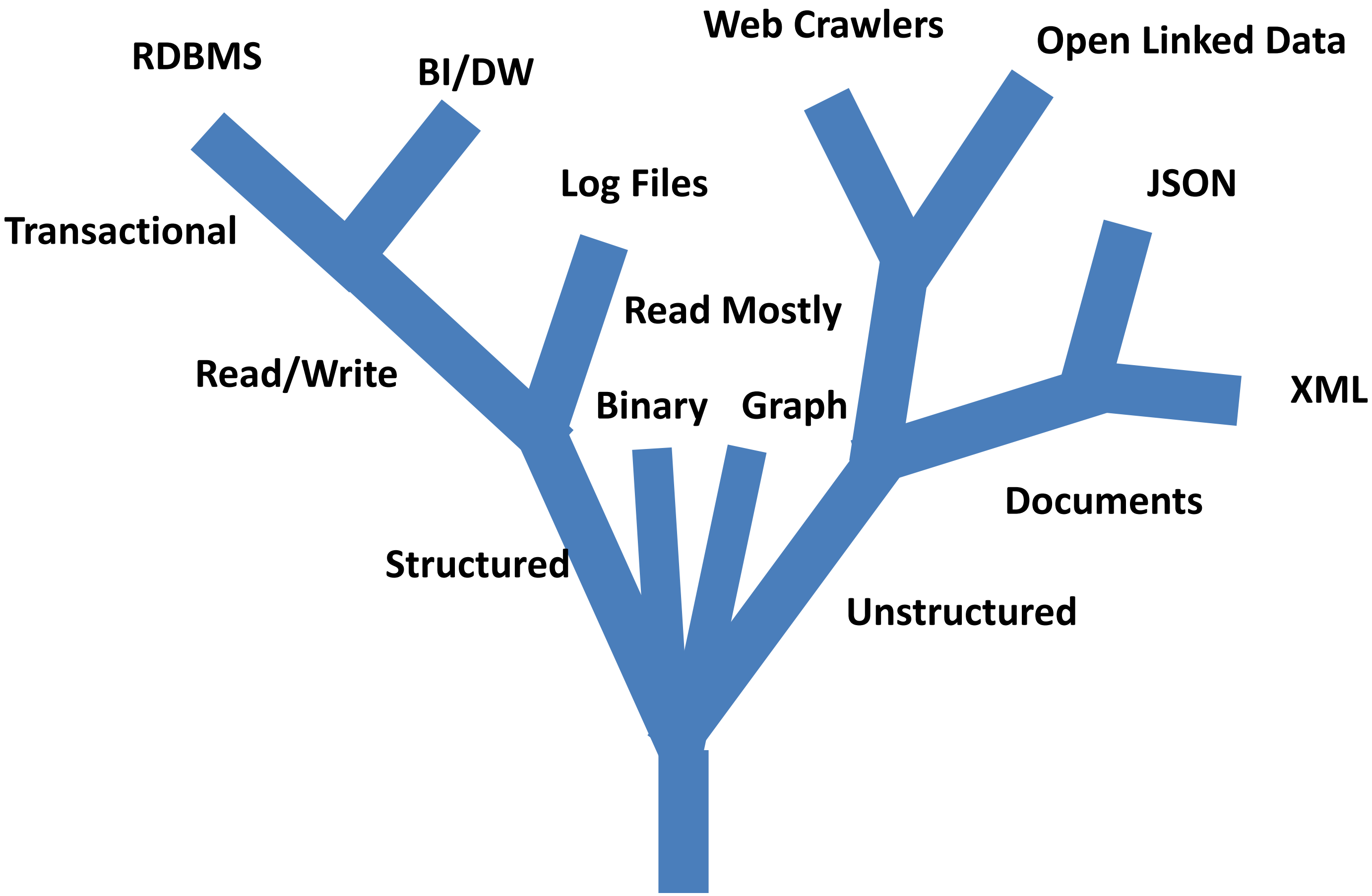
Why? (hint: how "shareable" is this process)

http://aws.amazon.com/elasticmapreduce/#pricing

Pressures on Single Node RDBMS Architectures



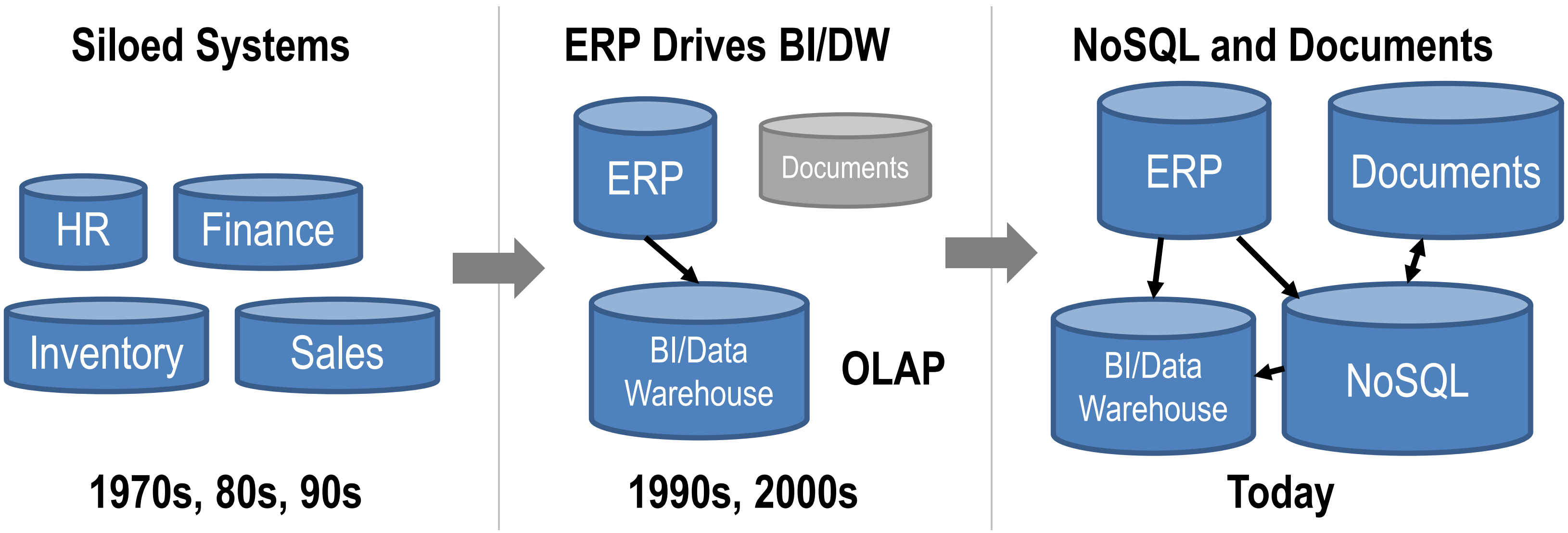
An evolving tree of data types



Many Uses of Data

- Transactions (OLTP)
- Analysis (OLAP)
- Search and Findability
- Enterprise Agility
- Discovery and Insight
- Speed and Reliability
- Consistency and Availability

Three Eras of Enterprise Data



- NoSQL will not replace ERP or BI/DW systems – but they will complement them and also facilitate the integration of unstructured document data

Simplicity is a Virtue

Photo from flickr by PSNZ Images

- Many modern systems derive their strength by dramatically limiting the features in their system and focus on a specific task
- Simplicity allows database designer to focus on the primary business drivers
- Simplicity promotes "separation of concerns"

Google MapReduce

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

- 2004 paper that had huge impact of functional programming on the entire community
- Copied by many organizations, including Yahoo

Google Bigtable Paper

Bigtable: A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber
{fay,jeff,sanjay,wilsonh,kerr,m3b,tushar,fikes,gruber}@google.com
Google, Inc.

- 2006 paper that gave focus to scaleable databases
- designed to reliably scale to petabytes of data and thousands of machines

Amazon's Dynamo Paper

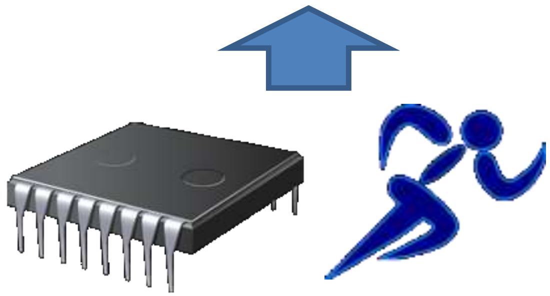
All Things Distributed

Werner Vogels' weblog on building scalable and robust distributed systems.

- Werner Vogels
- CTO - Amazon.com
- October 2, 2007
- Used to power Amazon's Web Sites
- One of the most influential papers in the NoSQL movement

Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swami Sivasubramanian, Peter Vosshall and Werner Vogels, "Dynamo: Amazon's Highly Available Key-Value Store", in the *Proceedings of the 21st ACM Symposium on Operating Systems Principles*, Stevenson, WA, October 2007.

Scale Up vs. Scale Out



Scale Up

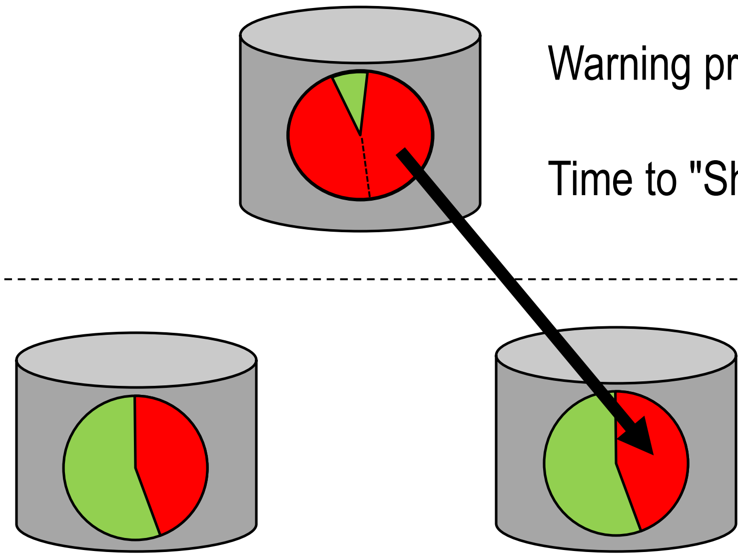
- Make a single CPU as fast as possible
- Increase clock speed
- Add RAM
- Make disk I/O go faster

Scale Out

- Make Many CPUs work together
- Learn how to divide your problems into independent threads

Automatic Sharding

Before Shard



Warning processor at 90% capacity!

Time to "Shard" – copy ½ data to a new processor

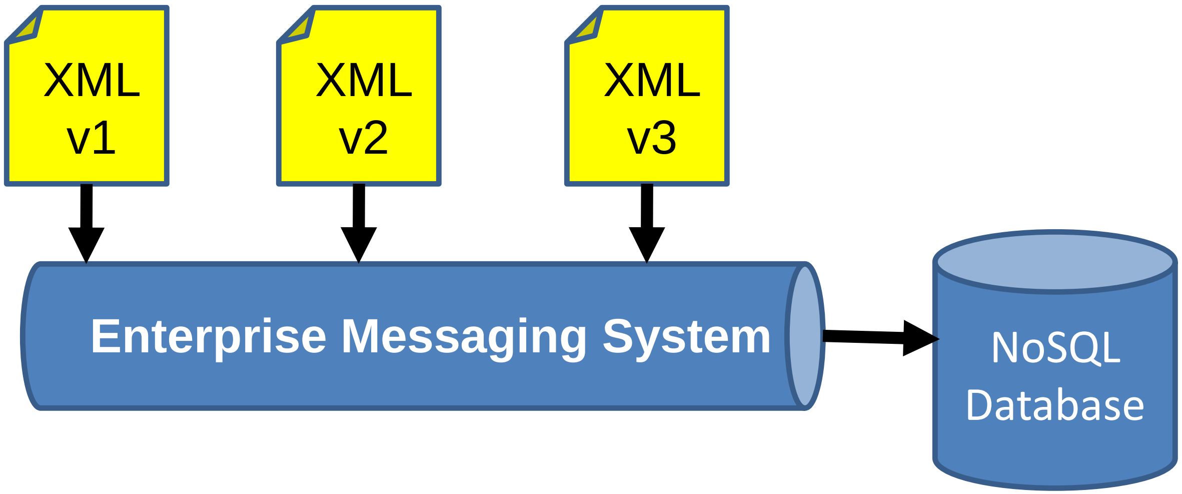
After Shard

Each processor gets ½ the load

Original processor **New processor**

- When one node in a cluster has too much of a load the system should be able to automatically rebalance the data distribution
- Note: Auto-sharding is not the same as replication!

Schema-Free Integration



"We can easily store the data that we **actually** get, not the data we **thought** we would get."

Horizontal Scalability

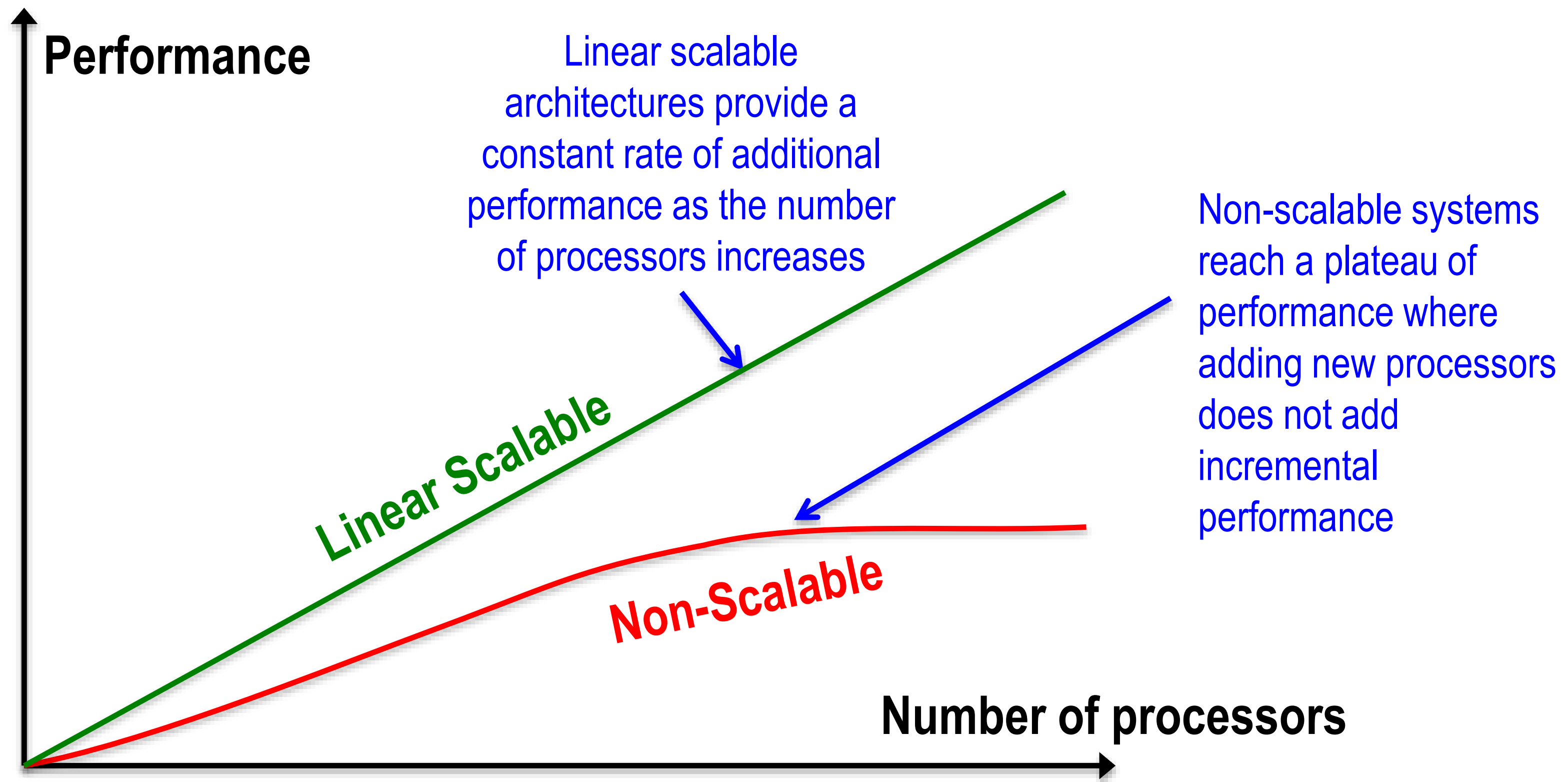
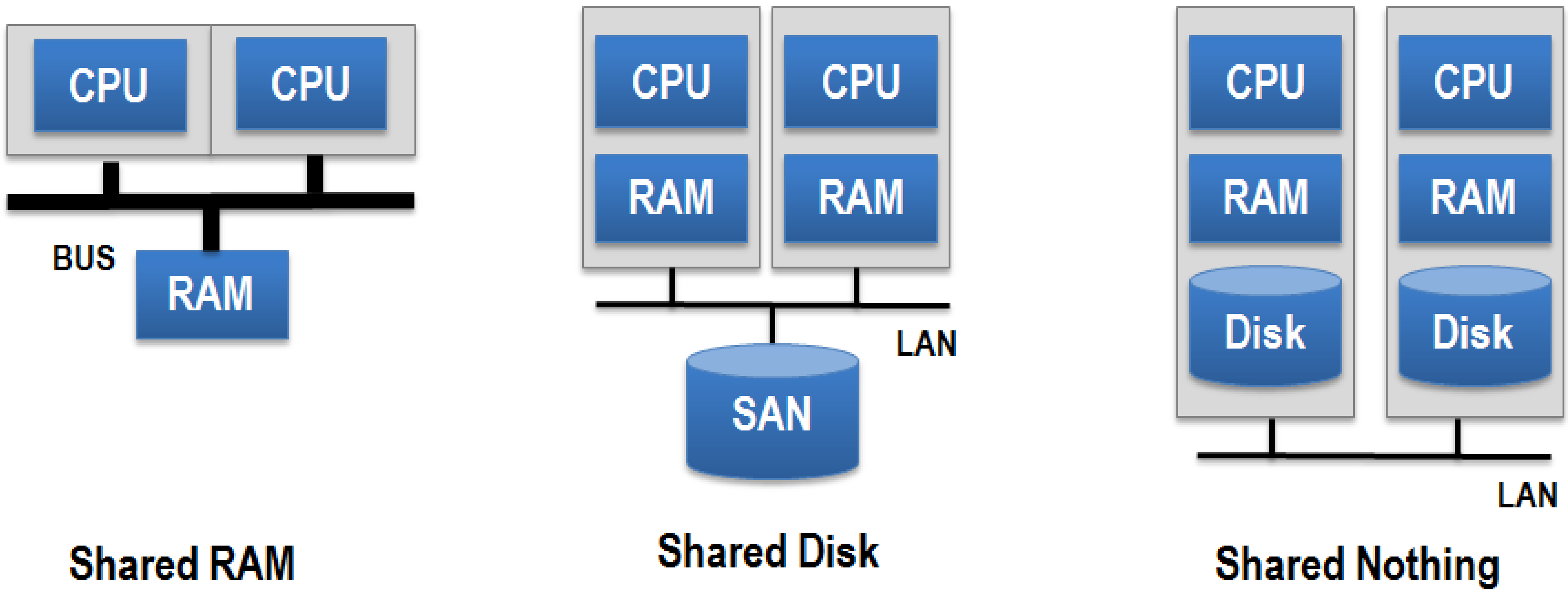


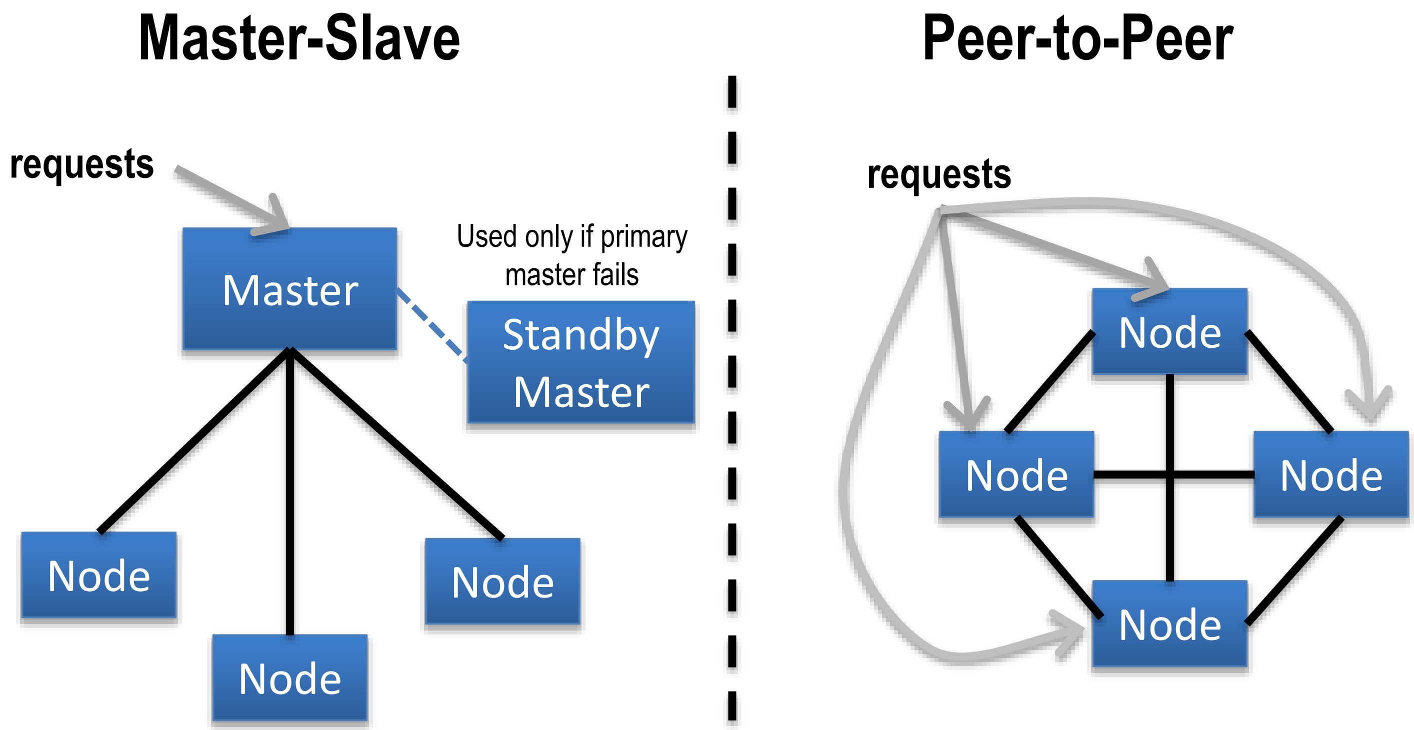
Figure 6.2

Shared Nothing Architecture



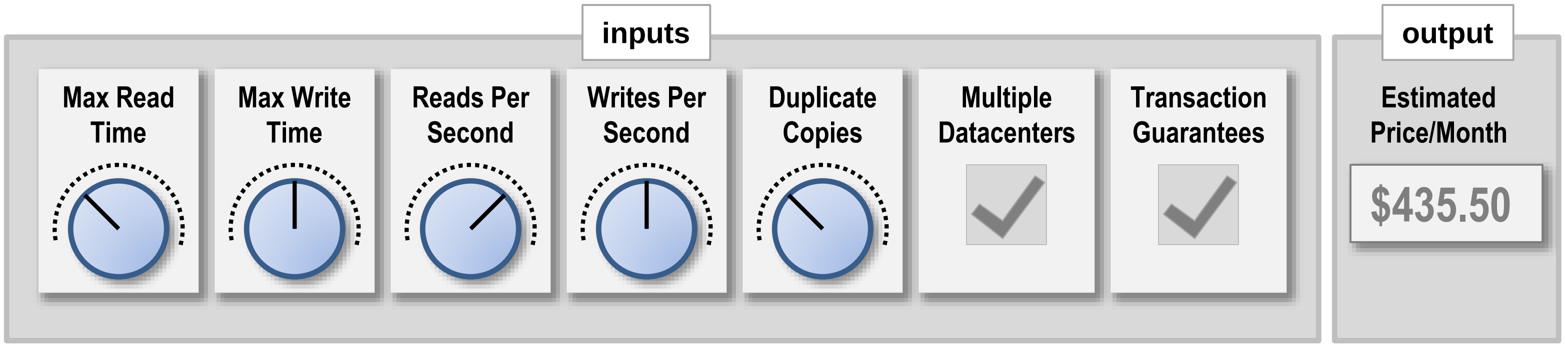
- Every node in the cluster has its own CPU, RAM and disk

Master-Slave vs. Peer to Peer



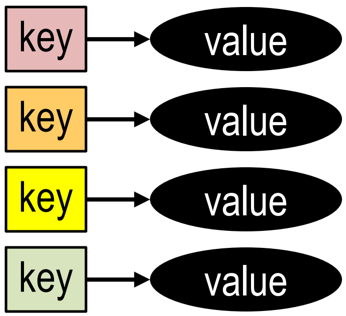
- The Master node may become a bottleneck in large clusters
- Many newer NoSQL architectures are moving toward a true peer-to-peer system

The Tunable SLA



- NoSQL systems that use many commodity processors can be precisely tuned to meet an organizations service level agreements

Key-Value Stores



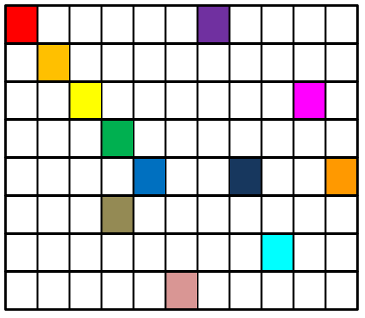
- Keys used to access opaque blobs of data
- Values can contain any type of data (images, video)

Pros: scalable, simple API (put, get, delete)

Cons: no way to query based on the content of the value

Examples:
Berkley DB,
Memcache,
DynamoDB, S3,
Redis, Riak

Column-Family



Examples:
Cassandra, HBase, Hypertable, Apache Accumulo, Bigtable

- Key includes a row, column family and column name
- Store versioned blobs in one large table
- Queries can be done on rows, column families and column names
- Pros: Good scale out, versioning
- Cons: Cannot query blob content, row and column designs are critical

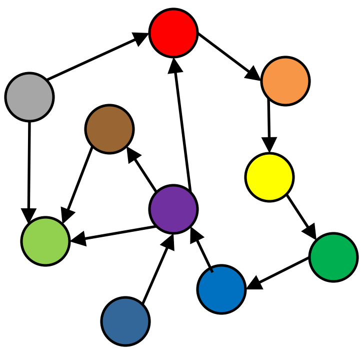
Column store key-value

Key

Row-ID	Column Family	Column Name	Timestamp	Value

- The key is composed of:
  - row id (string)
  - Column family (grouping of columns)
  - Column name (string)
  - Timestamp (64-bit value)
- Value
  - any blob (byte stream)

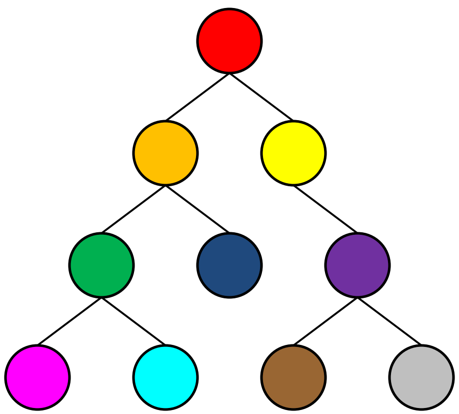
Graph Store



Examples:
Neo4j, AllegroGraph, Bigdata triple store, InfiniteGraph, StarDog

- Data is stored in a series of nodes, relationships and properties
- Queries are really graph traversals
- Ideal when relationships between data is key:
  - e.g. social networks
- **Pros:** fast network search, works with public linked data sets
- **Cons:** Poor scalability when graphs don't fit into RAM, specialized query languages (RDF uses SPARQL)

Document Store

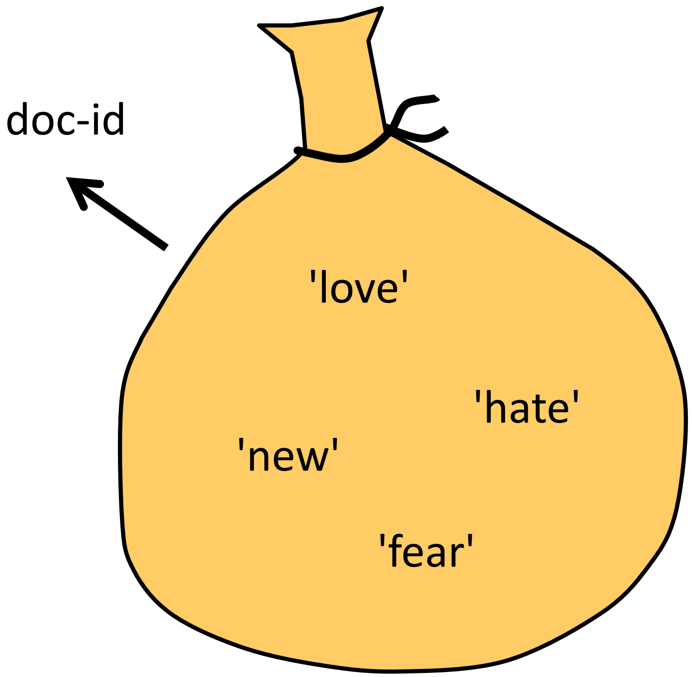


Examples:
MarkLogic, MongoDB, Couchbase, CouchDB, RavenDB, eXist-db

- Data stored in nested hierarchies
- Logical data remains stored together as a unit
- Any item in the document can be queried
- **Pros:** No object-relational mapping layer, ideal for search
- **Cons:** Complex to implement, incompatible with SQL

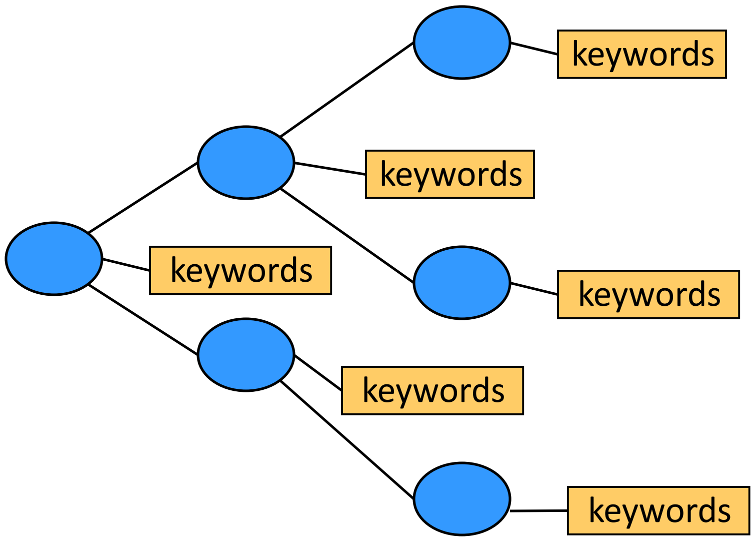
Two Models

"Bag of Words"



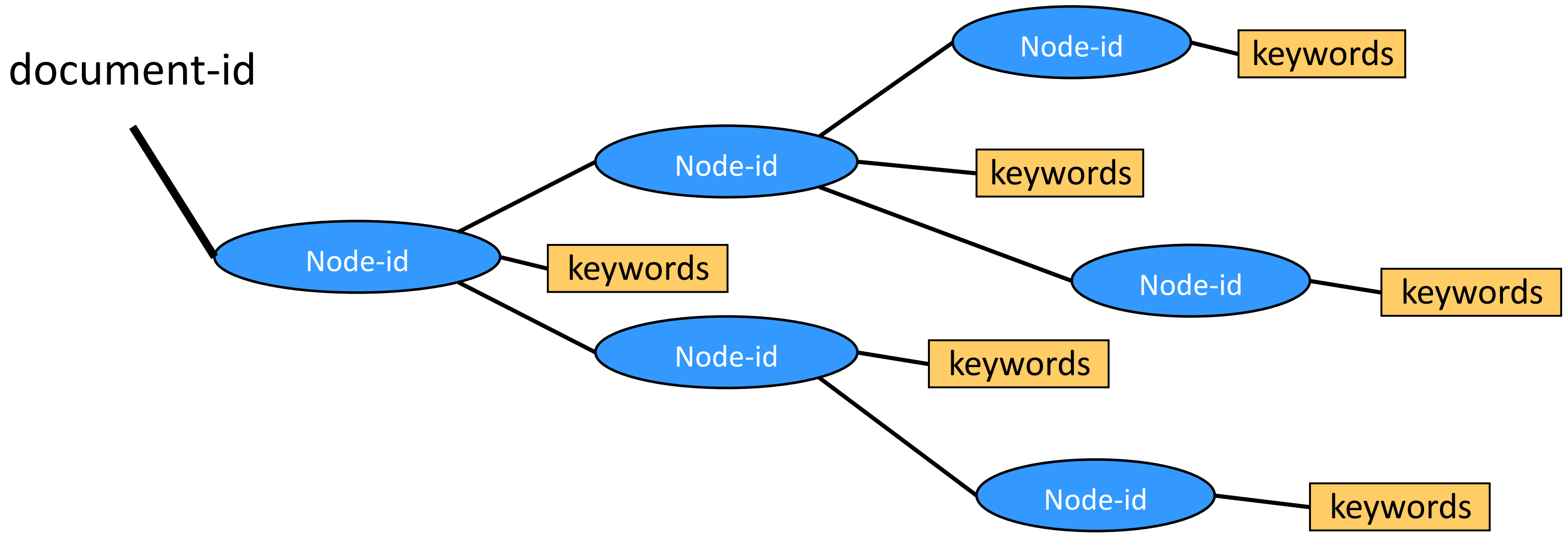
- All keywords in a single container
- Only count frequencies are stored with each word

"Retained Structure"



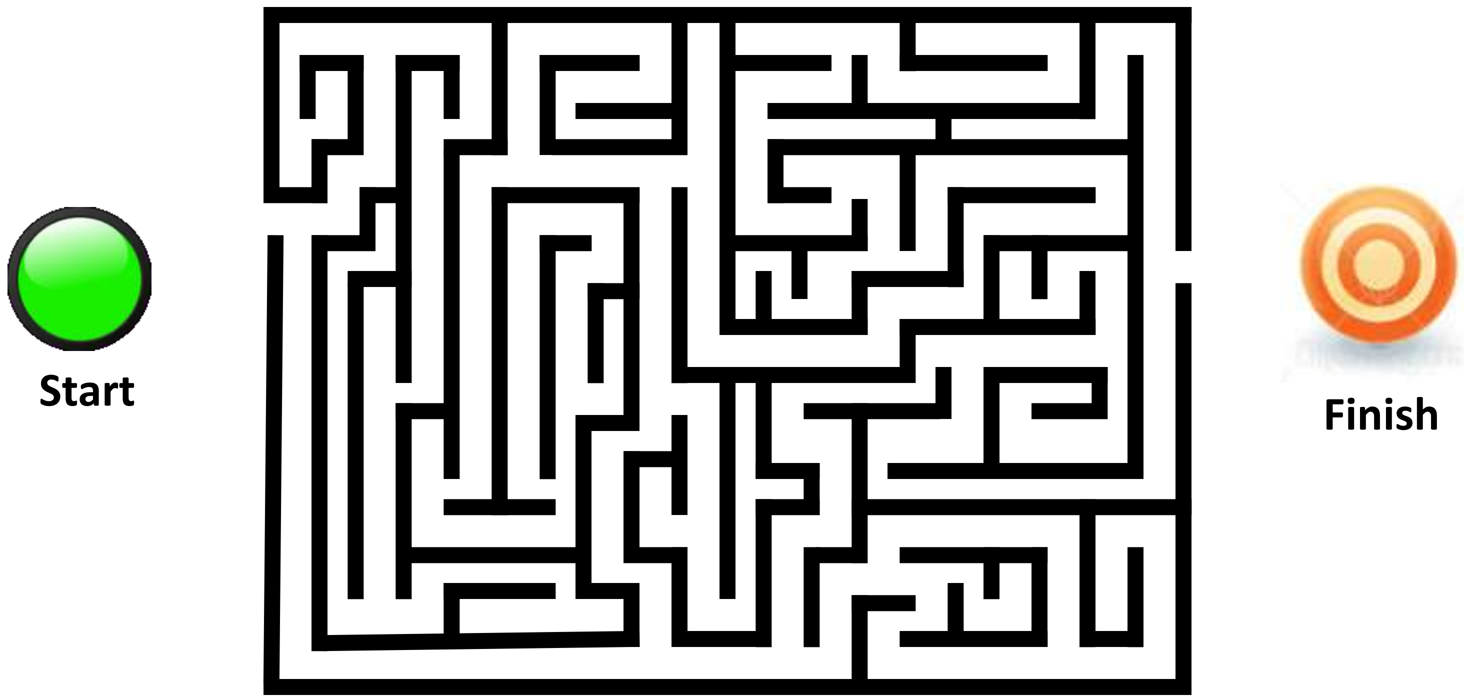
- Keywords associated with each sub-document component

Keywords and Node IDs



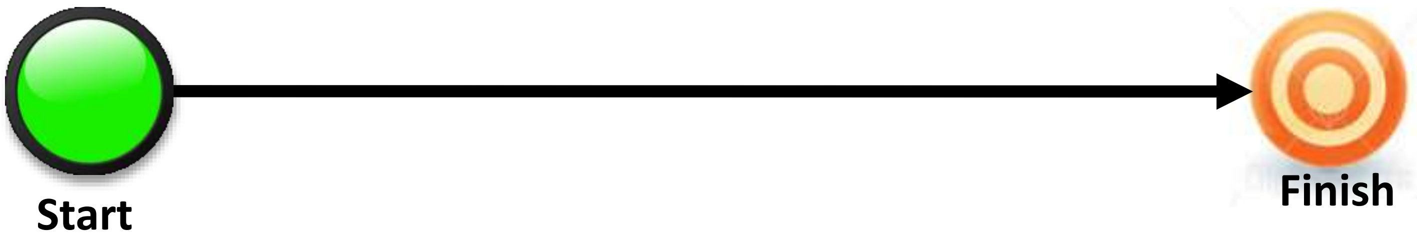
- Keywords in the reverse index are now associated with the **node-id** in every document

Using the Wrong Architecture



Credit: Isaac Homelund – MN Office of the Revisor

Using the Right Architecture



Find ways to remove barriers and empower the **non programmers** on your team.

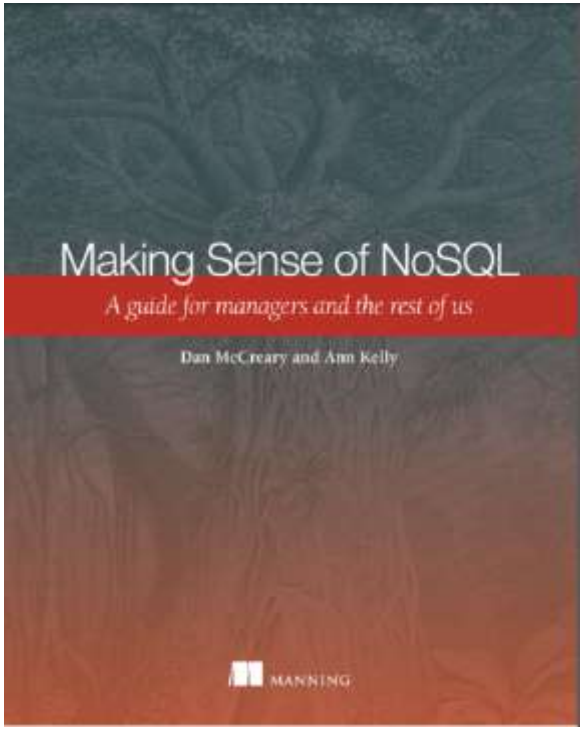
Further Reading and Questions

Dan McCreary

dan@danmccreary.com

twitter @dmccreary

Thank You!



http://manning.com/mccreary