ATAM for NoSQL Database Selection

Using Architecture (Not Products) to Guide Database Selection

Dan McCreary
Kelly-McCreary & Associates

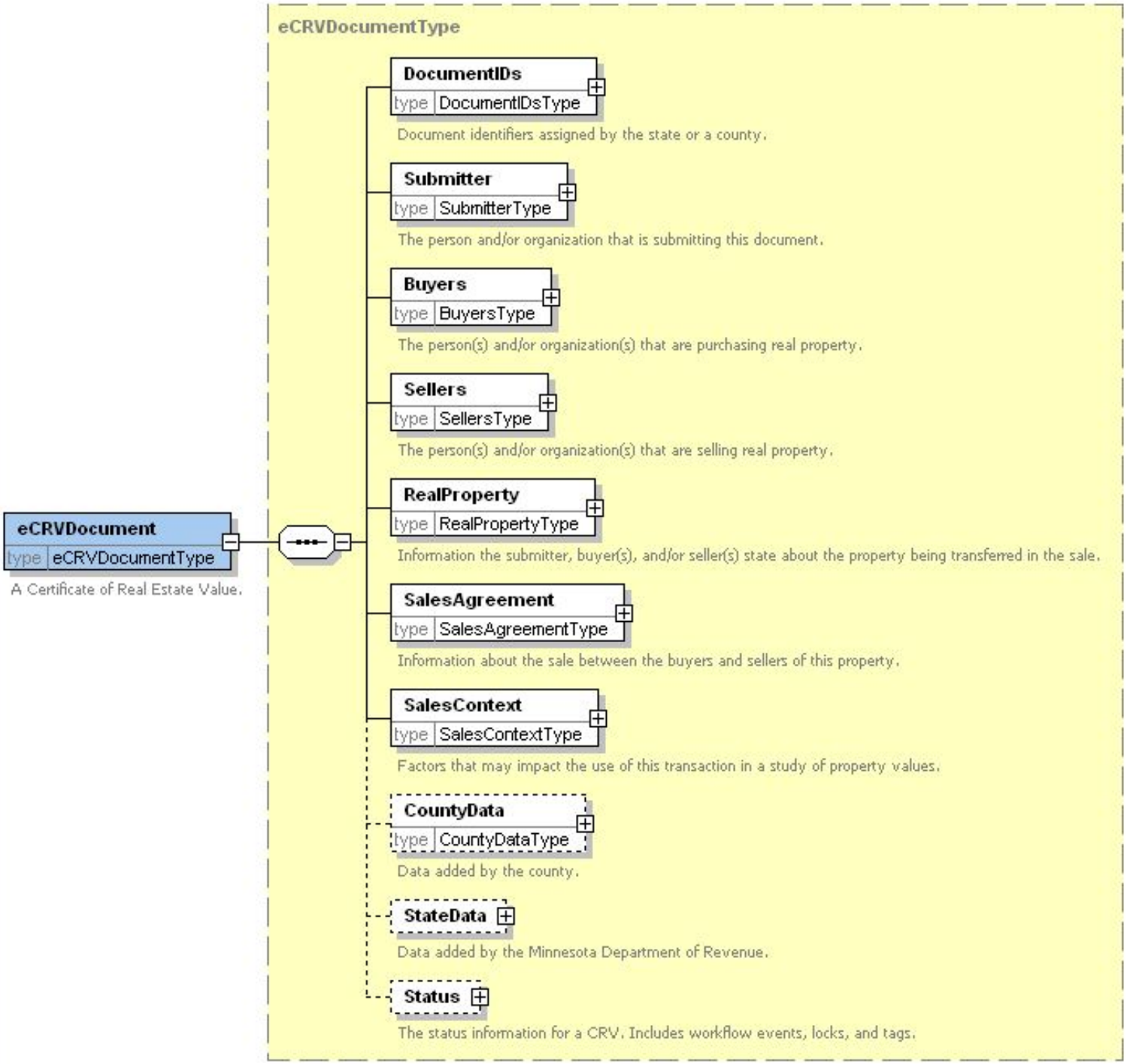
Session Description

New NoSQL databases offer more options to the database architect. Selecting the right NoSQL database for your project has become a nontrivial task. Yet selecting the right database can result in huge cost savings and increased agility. This presentation will show how the Architecture Tradeoff Analysis Method (ATAM) can be applied to objectively select the best database architecture for a project.

We review the core NoSQL database architecture patterns (key-value stores, column-family stores, graph databases, and document databases) and then present examples of using quality trees to score business problems with alternative architectures. We'll address creative ways to use combinations of NoSQL architectures, cloud database services, and frameworks such as Hadoop, HDFS, and MapReduce to build back-end solutions that combine low operational costs and horizontal scalability. The presentation includes real-world case studies of this process.

My Story

- How I got into using ATAM for database selection
- Background as of 2006
  - Worked for Steve Jobs at NeXT
  - Object-oriented development Objective-C, Java
  - Strong focus on object-relational mapping
    - DBKit, WebObjects, Hibernate, Oracle, Sybase, MS-SQL Server
  - 7 years doing XML (CriMNet), Metadata Registries (ISO/IEC 11179) and Semantics (RDF, OWL etc.)


eCRVDocumentType
DocumentIDs
type DocumentIDsType
Document identifiers assigned by the state or a county.
Submitter
type SubmitterType
The person and/or organization that is submitting this document.
Buyers
type BuyersType
The person(s) and/or organization(s) that are purchasing real property.
Sellers
type SellersType
The person(s) and/or organization(s) that are selling real property.
RealProperty
type RealPropertyType
Information the submitter, buyer(s), and/or seller(s) state about the property being transferred in the sale.
SalesAgreement
type SalesAgreementType
Information about the sale between the buyers and sellers of this property.
SalesContext
type SalesContextType
Factors that may impact the use of this transaction in a study of property values.
CountyData
type CountyDataType
Data added by the county.
StateData
Data added by the Minnesota Department of Revenue.
Status
The status information for a CRV. Includes workflow events, locks, and tags.
eCRVDocument
type eCRVDocumentType
A Certificate of Real Estate Value.

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

MINNESOTA · REVENUE

Property Taxation Metadata Registry

Metadata Registry Home | About | Feedback | Links | Bibliography

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

Metadata Registry Home | Training | FAQ

Minnesota Department of Revenue. Page generated on 2006-05-25-05:00

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 Rocks!

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

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:

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

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):

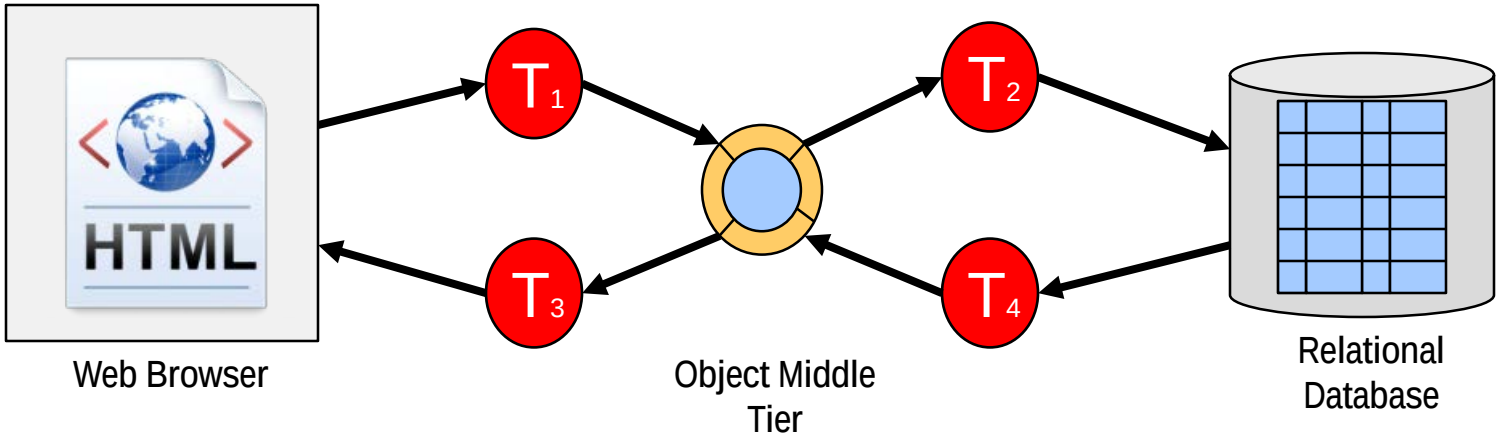
Comment:

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

Use a Native XML Database!

Kurt Cagle

Web Form

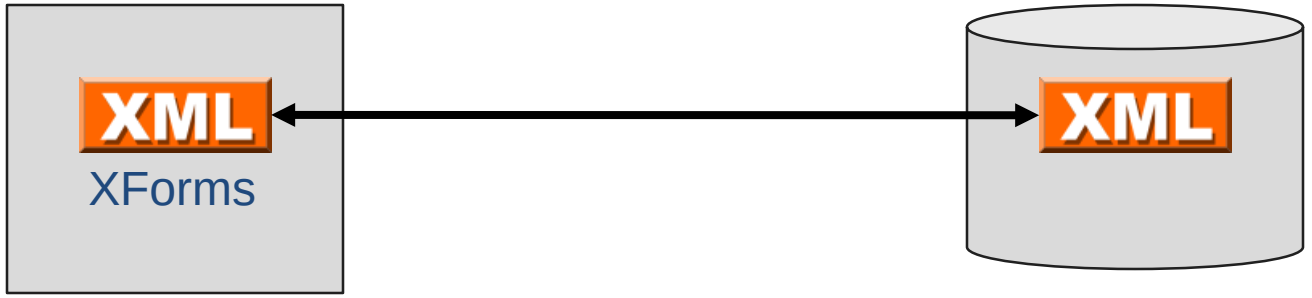
Save

Web Browser

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

eXist-db

Zero Translation



- XML lives in the web browser (XForms)
- REST interfaces
- XML in the database (Native XML, XQuery)
- XRX Web Application Architecture
- No translation!

Database Tradeoff Analysis

My Way

- 10,000 lines of code to break CRV XML document into Java component and use Hibernate to store each element into columns of tables
- 45 SQL inserts store
- 20 joins to extract
- Six months
- Four developers (Java, SQL, DBA, Project Manager)

Kurt's Way

- One line of codes to store CRB
- One day to write
- One week to test
- 100x increase in agility

The Paradigm Shift

RDBMSs are the **only** way to store enterprise data

There are **many** ways to store enterprise data and if you pick the right one your **agility** can go up x1,000

Reality Sets In

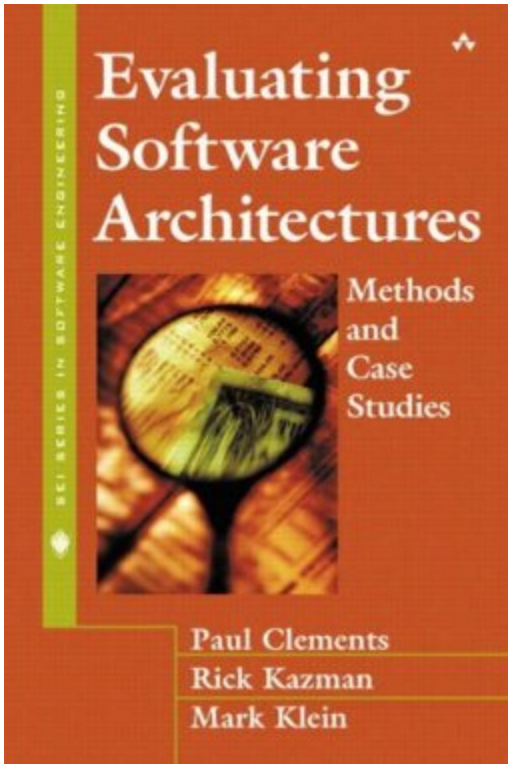
What I wanted

- **Objective** architecture analysis
- **Fairly** weigh the pros and cons of each alternative
- Be surrounded by people that know the strengths and weakness of **many** alternatives

What I got

- Architecture decisions driven by an RDBMS **license**
- Architecture decisions made by **one person** with limited exposure to alternatives
- Architecture decisions made by **lack of knowledge**
- Architecture by **fear** of the unknown

Key Book for ATAM



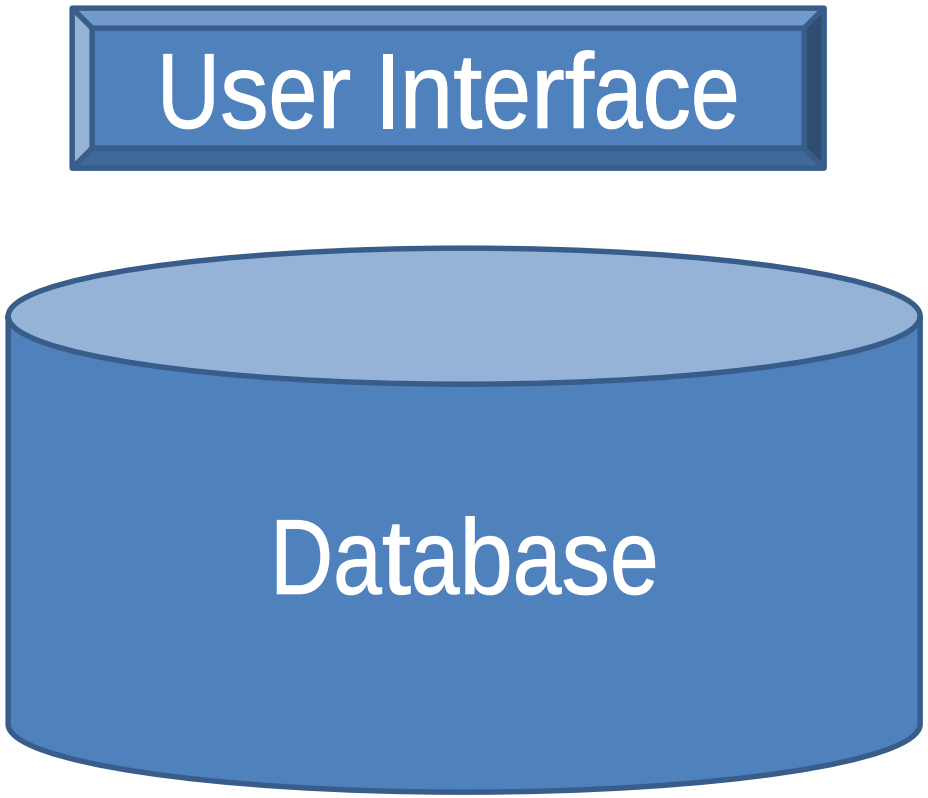
Evaluating Software Architectures: Methods and Case Studies

by Paul Clements, Rick Kazman, and Mark Klein

- Addison-Wesley, 2001

We need "Evaluating Database Architectures"!

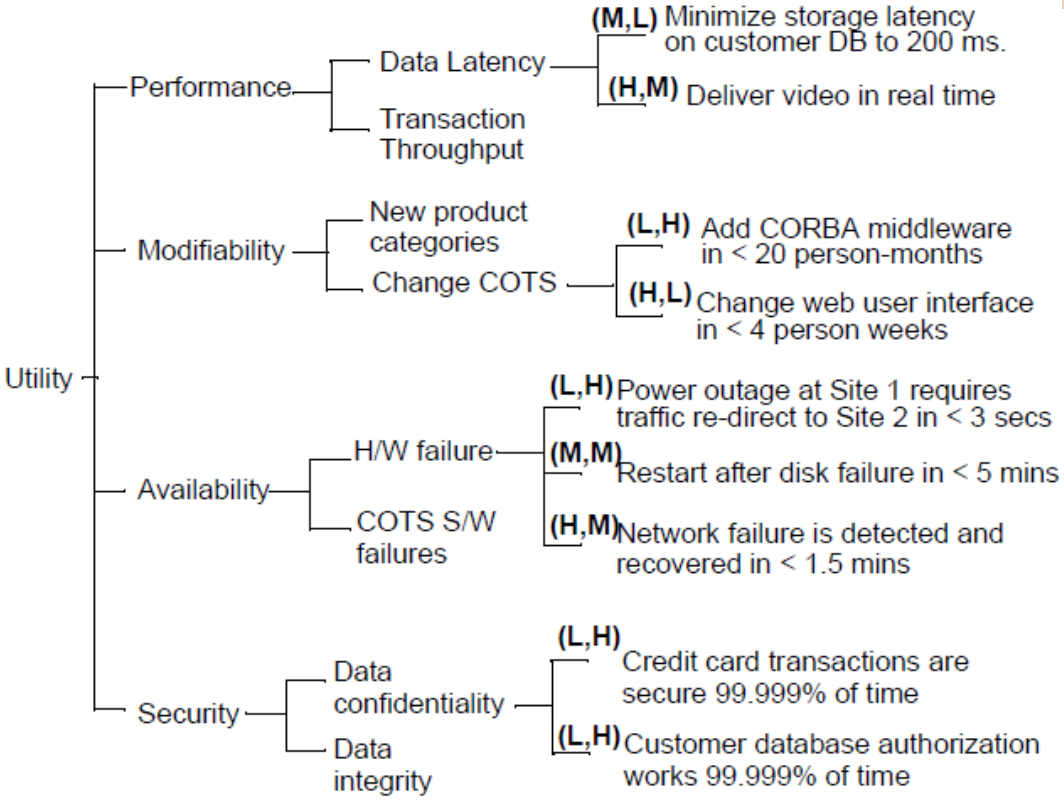
How Important is the Database?



10%-20%

80%-90%

Reference Utility Tree from Book

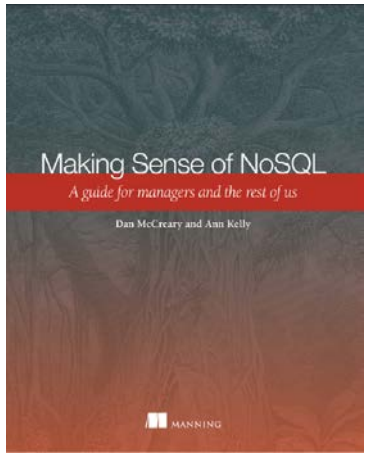


Note: Mostly Database Issues

Anger, Wiki, Conference, Book

2011, 2012

2013 NOSQL NOW! AUGUST 20-22 | SAN JOSE

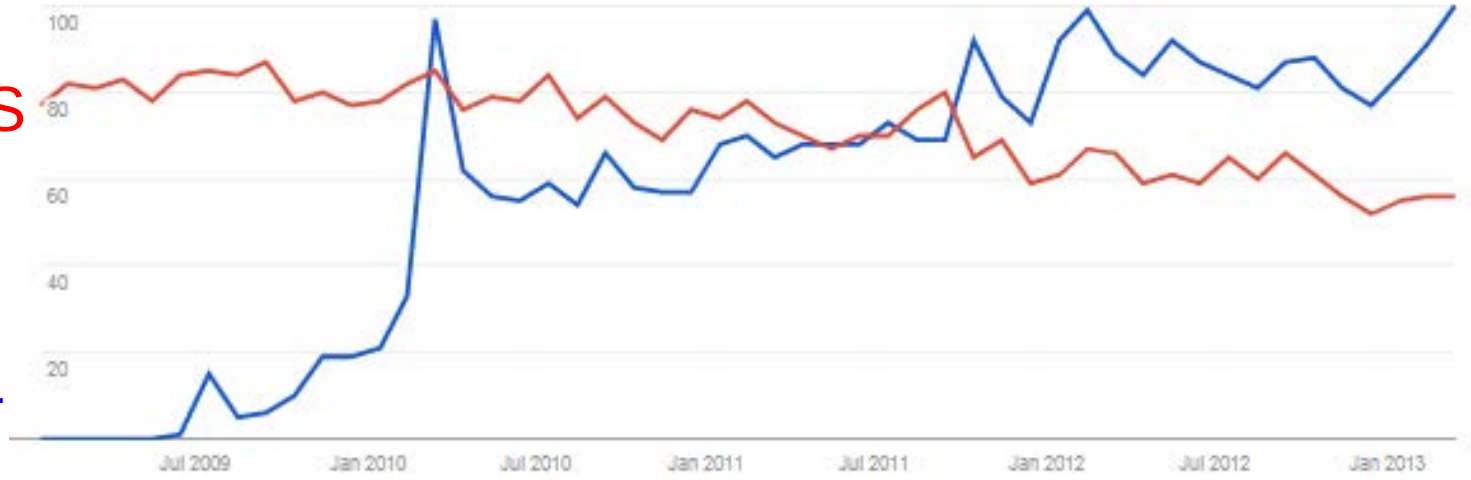


RDBMS vs. NoSQL

- NoSQL is real and it's here to stay

RDBMS

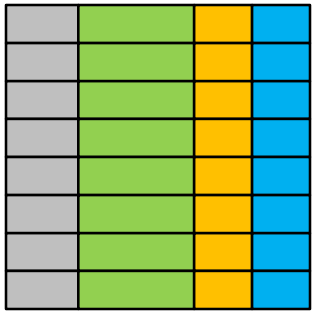
NoSQL



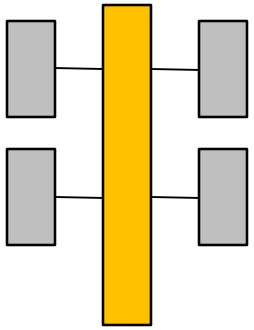
http://www.google.com/trends/explore#q=nosql%2C%20rdbms&date=1%2F2009%2051m&cmpt=q

Before NoSQL

Relational

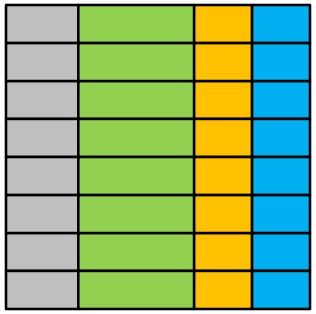


Analytical (OLAP)

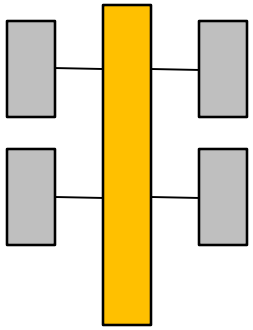


After NoSQL

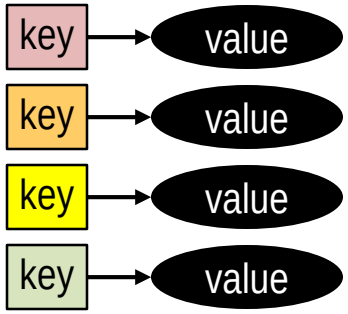
Relational



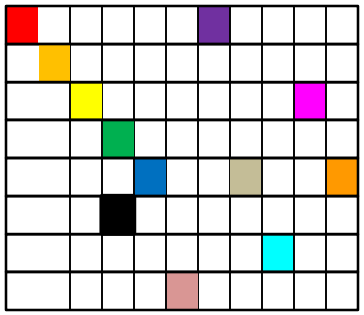
Analytical (OLAP)



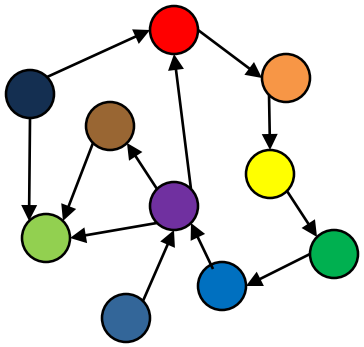
Key-Value



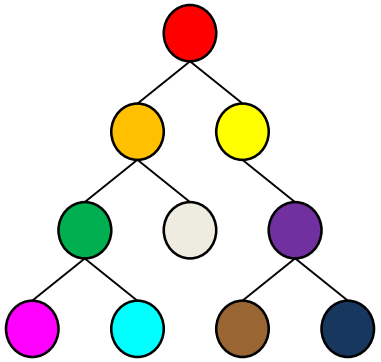
Column-Family



Graph

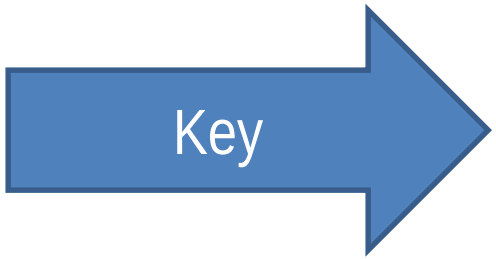


Document



21

Suggested Process



1. Define high level Requirements
2. Consider all six major architectures (SQL and NoSQL)
3. Select database architecture **first**
4. Select products that support the architecture **second**

Finding the right tool for the job

The Problem:

Many possible Solutions:

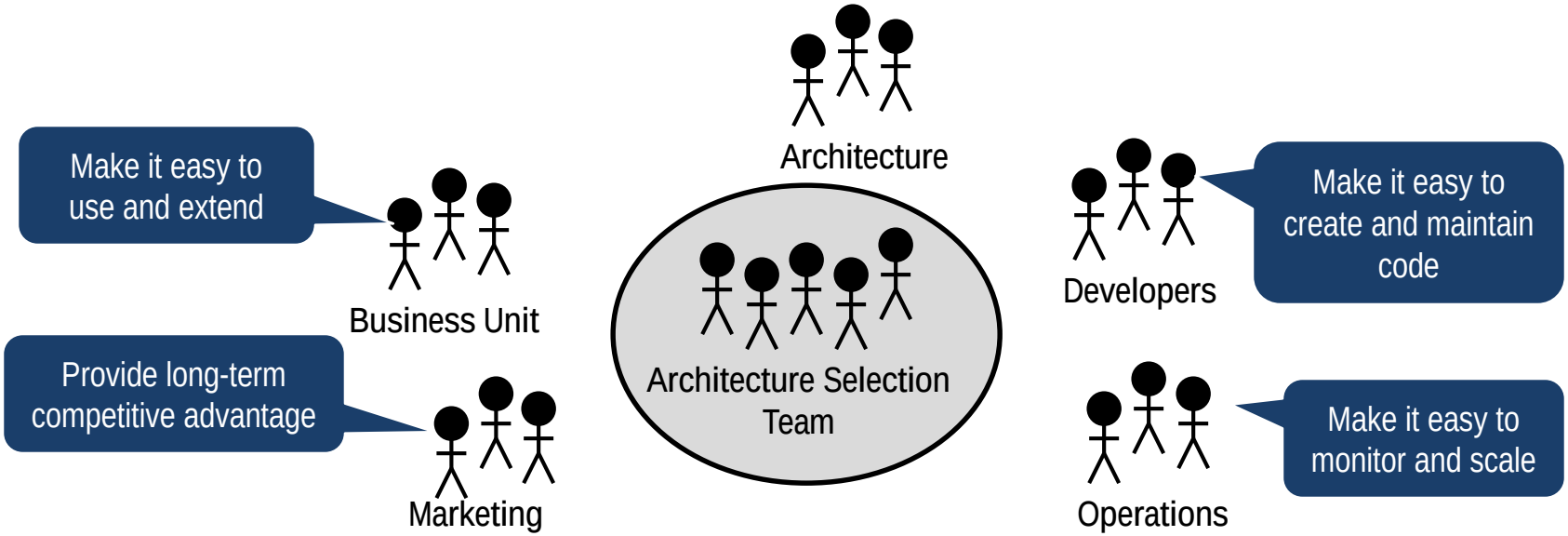






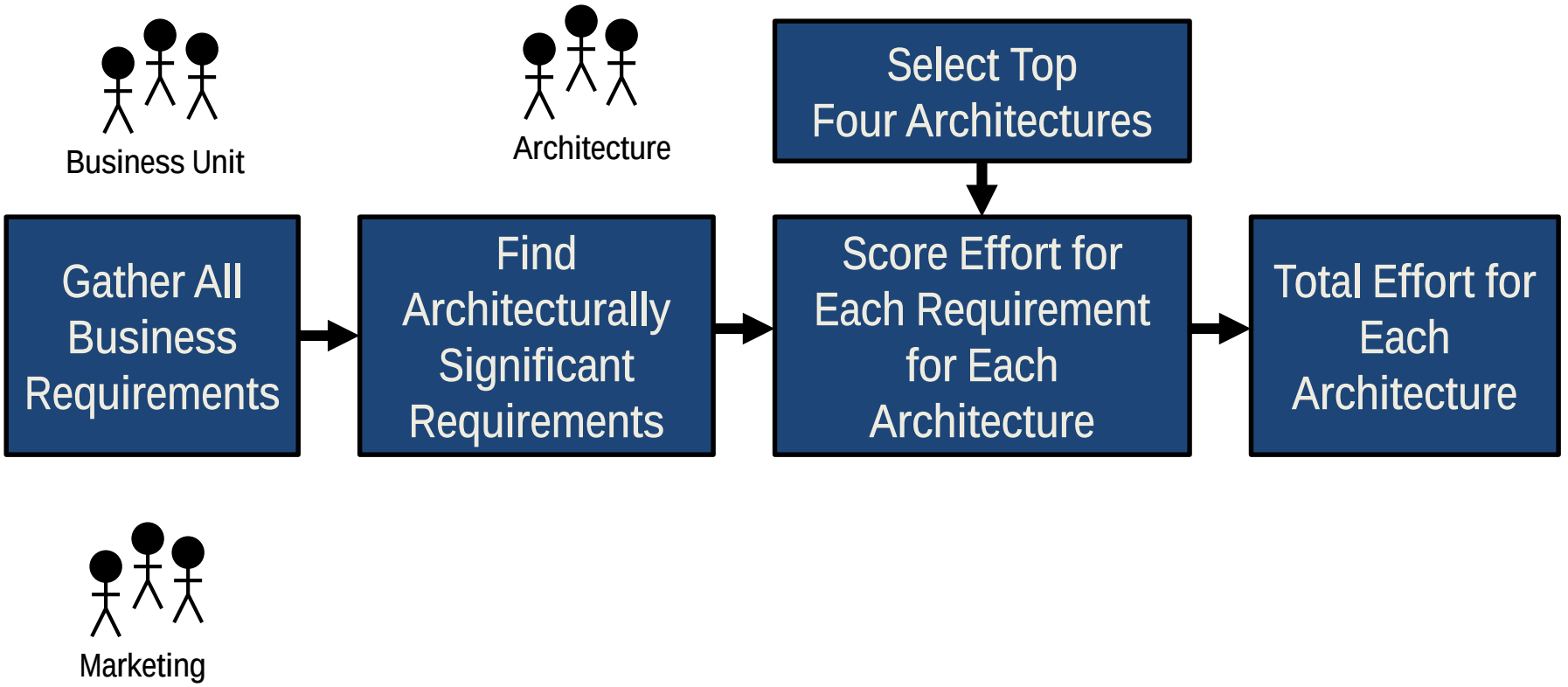

What tool will have the best fit? Multiple tools? One item vs. many?

Architecture Selection



- Don't underestimate the role of **marketing** to promote good architecture!

Selection Process

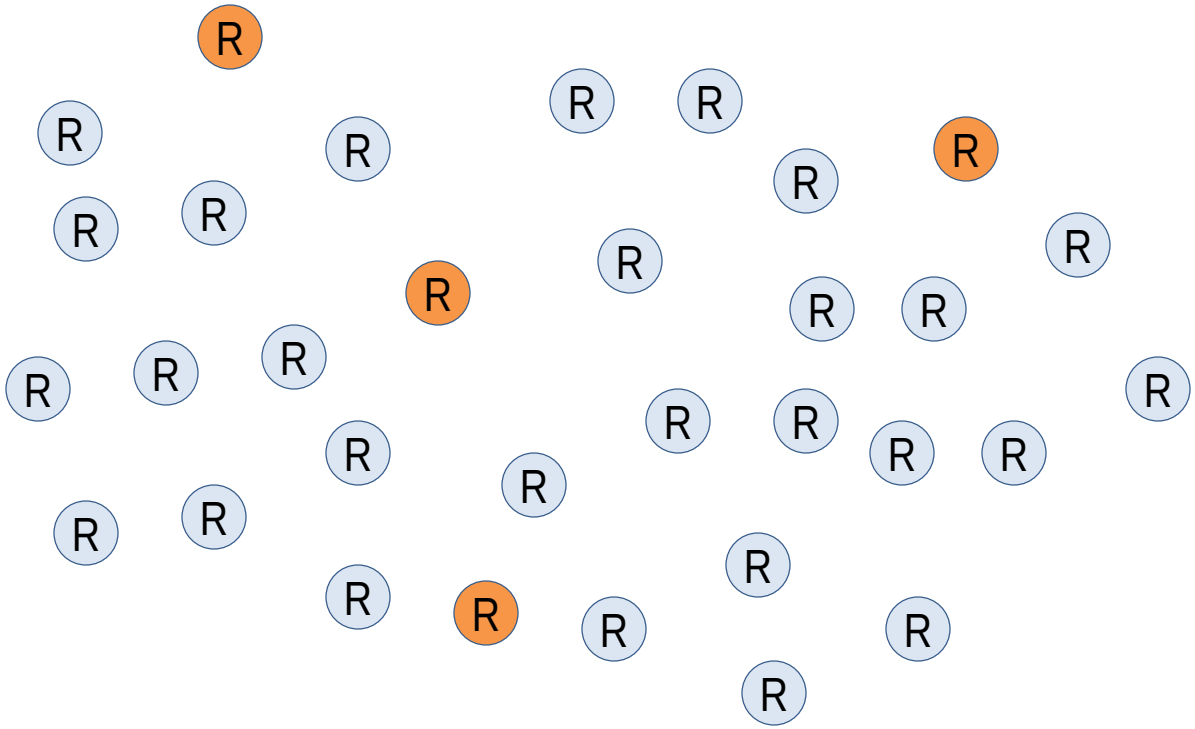


Use Case Driven Difficulty Analysis

Architectural Trade-off Analyis for Project ABC							
		Database Architecture					
Category	Use Case	RDBMS	OLAP Cube	Key-Value Store	Col-Family	Graph	Document
Injest	Load data						
	Load code tables						
	Add record						
Validate	Structure						
	Required fields						
	Optional fields						
Update	Batch						
	Record-by-record						
Search	Fulltext						
	Change sort order						
Export	Reports in HTML						
	Export as XML						
	Export as JSON						
	Totals						

- architecture-score-card, not a product score card!

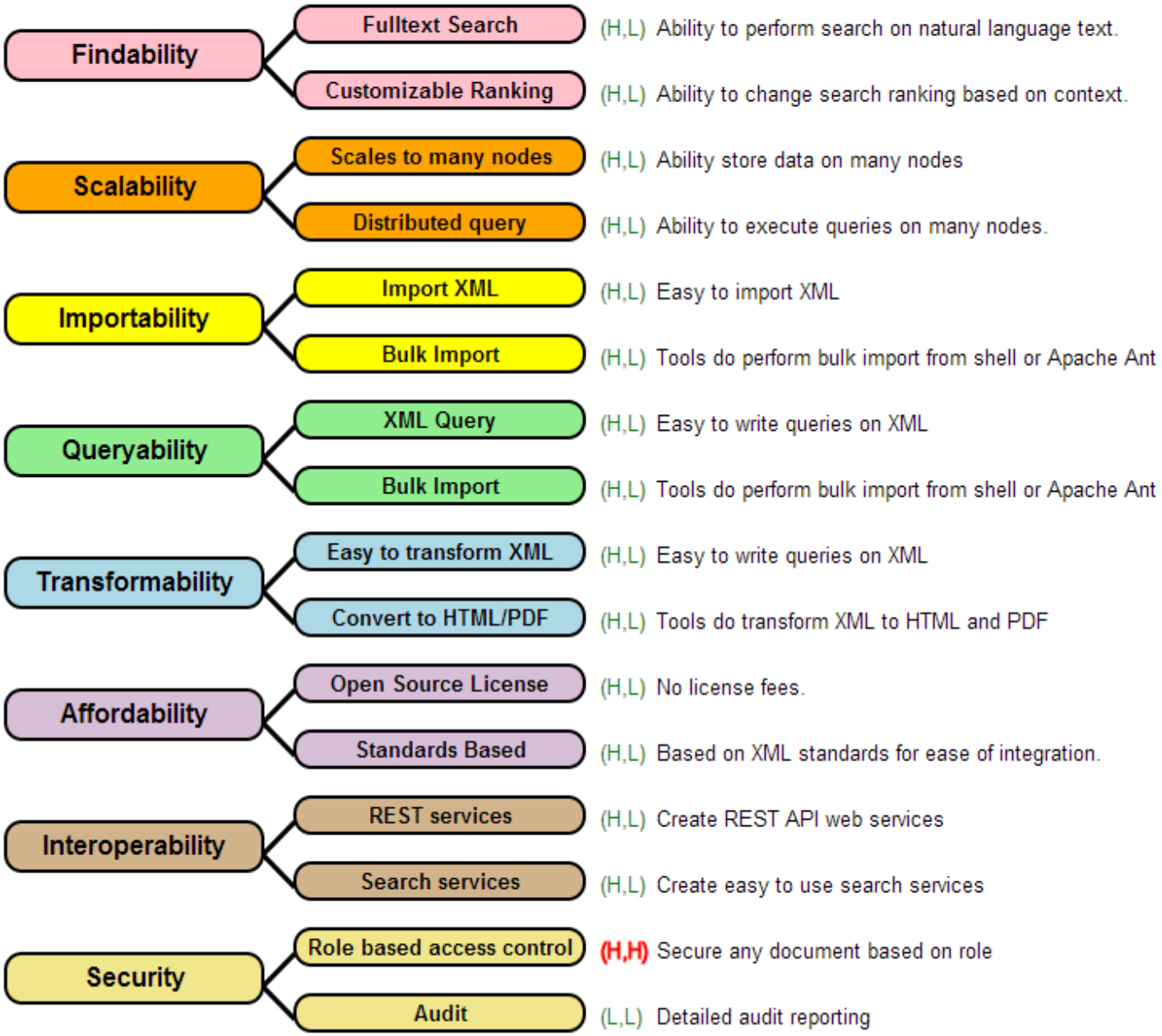
Architecturally Significant Features



- A requirement that drives overall architecture
- Requires experts to know when a requirement is significant

Quality Attribute Utility Tree
Project: Legislative Statute Archive for Library of Congress

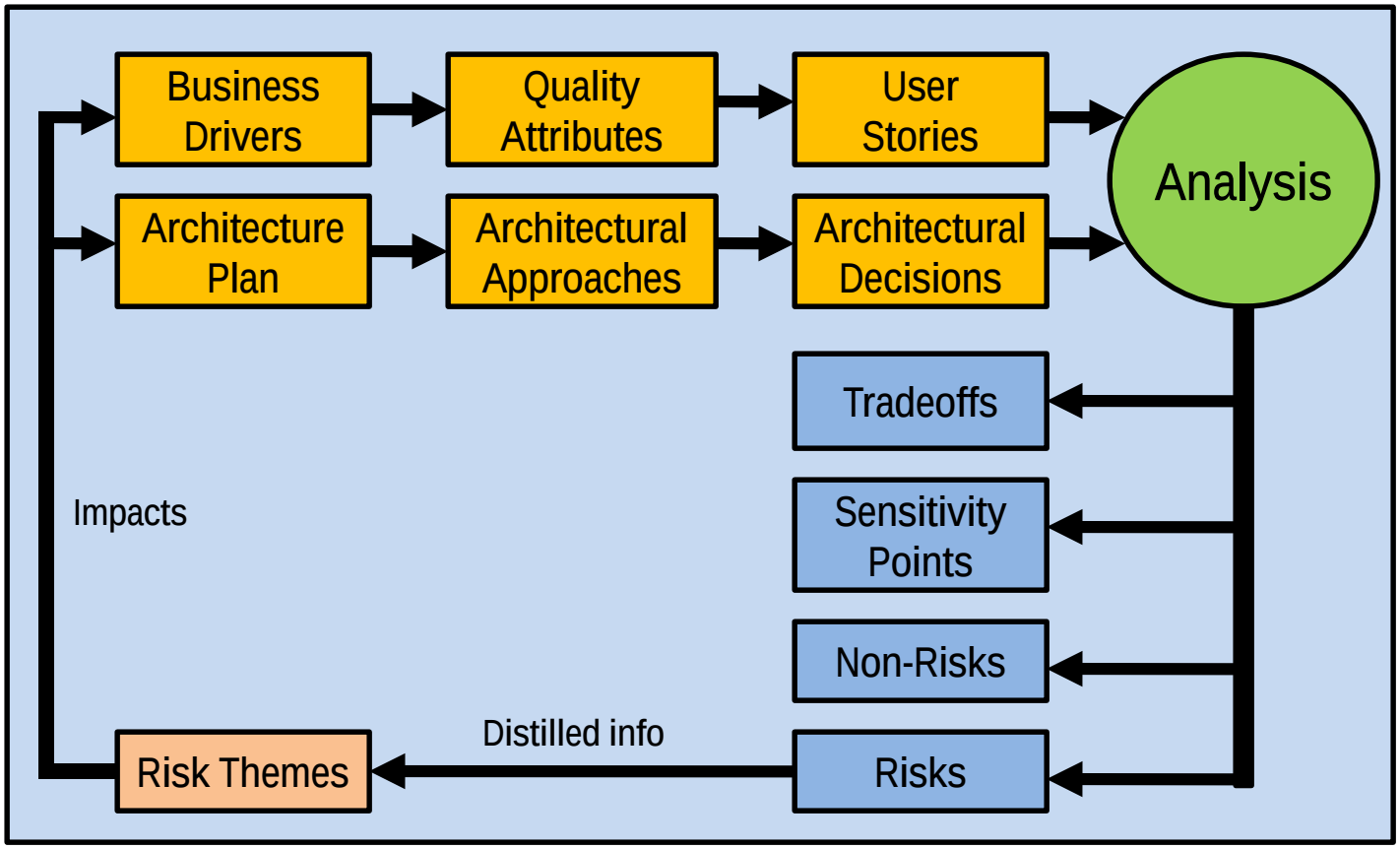
Author: Dan McCreary
Description: Find state statues stored in XML formats.



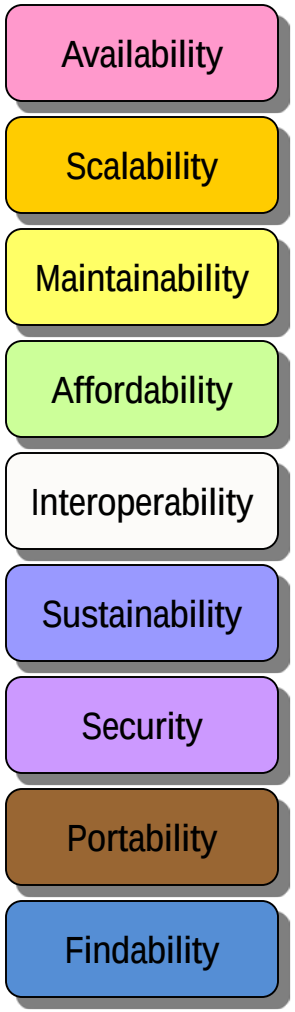
Variations Used

- Change focus from "Performance" to "Scalability"
- Change "Modifiability" to "Agility"
- Increased emphasis on:
  - Big Data
  - Searchability
  - Monitorability
  - Supportability
  - Affordability

ATAM Process Flow



Sample Utility Tree

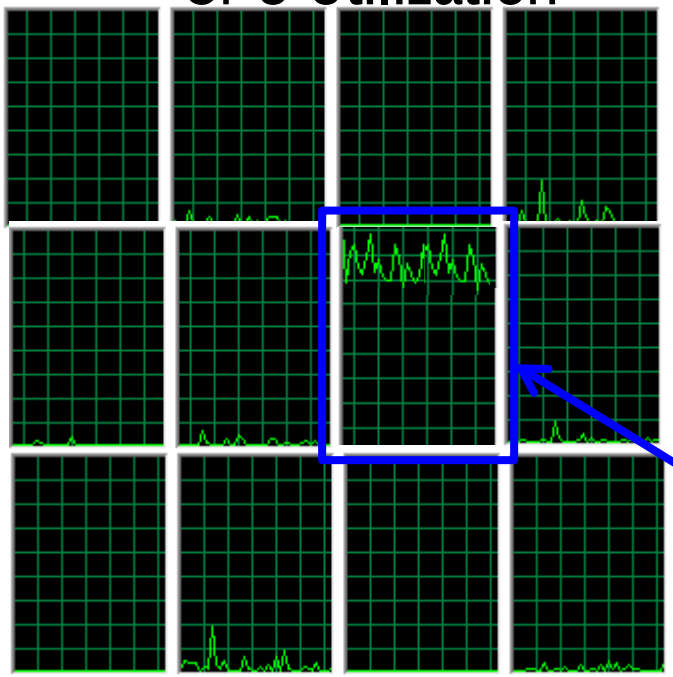


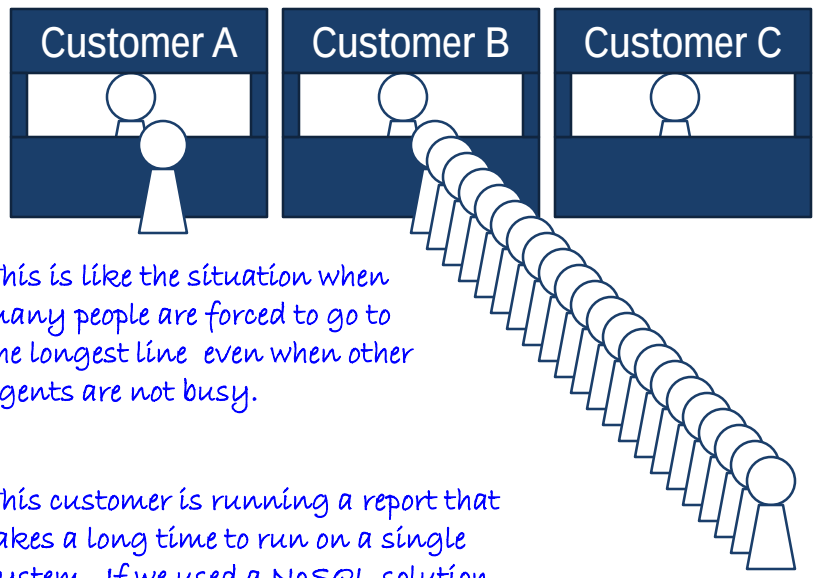
- Each topic (Quality Attribute) helps focus the discussion of a selection team
- The topics vary from project to project
- Big Data projects focus on "Scalability" and "Findability" etc.
- Objective ranking of requirements before you begin talking about architecture alternatives

Hand in Glove

- The **quality** of the fit is driven by the quality of each finger's fit
- If one finger doesn’t fit the entire glove has a poor fit
- "Quality trees" help us evaluate the overall fitness of a problem and solution
- Removes focus on a single dimension

Finding the right metaphor



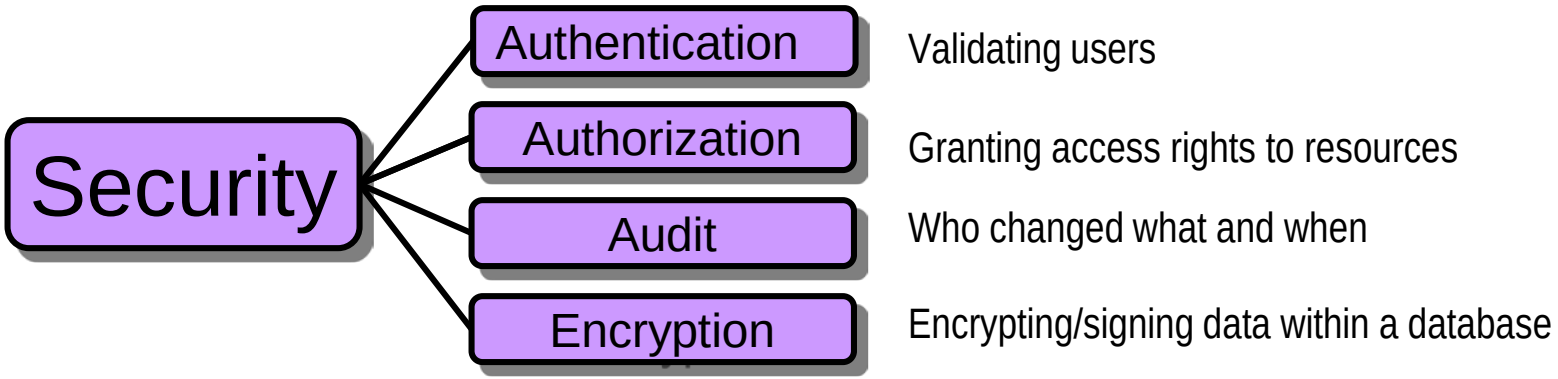


This is like the situation when many people are forced to go to the longest line even when other agents are not busy.

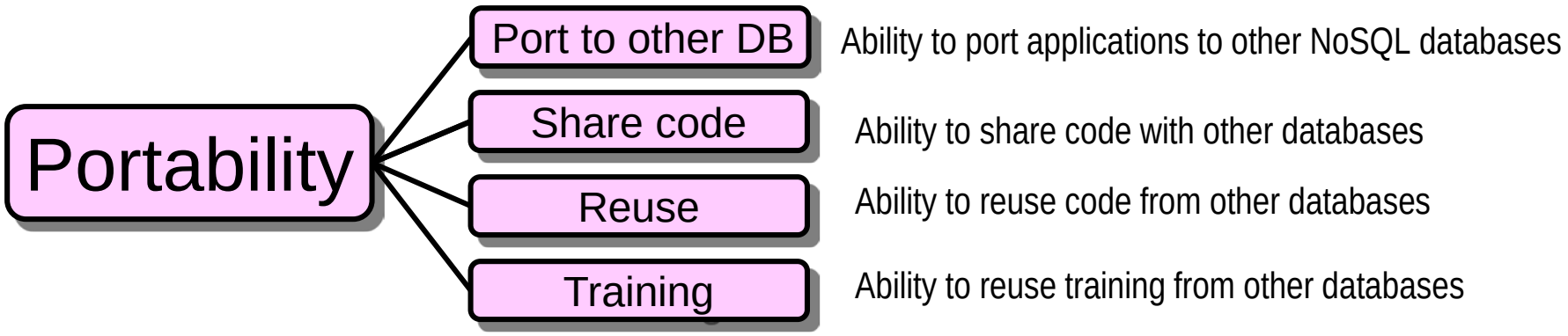
This customer is running a report that takes a long time to run on a single system. If we used a NoSQL solution the report would be evenly distributed on all servers and runs in 1/12 the time.

Change in focus of Security

- Requirements by threat type (DOS, Injection, Internal, Social Engineering)



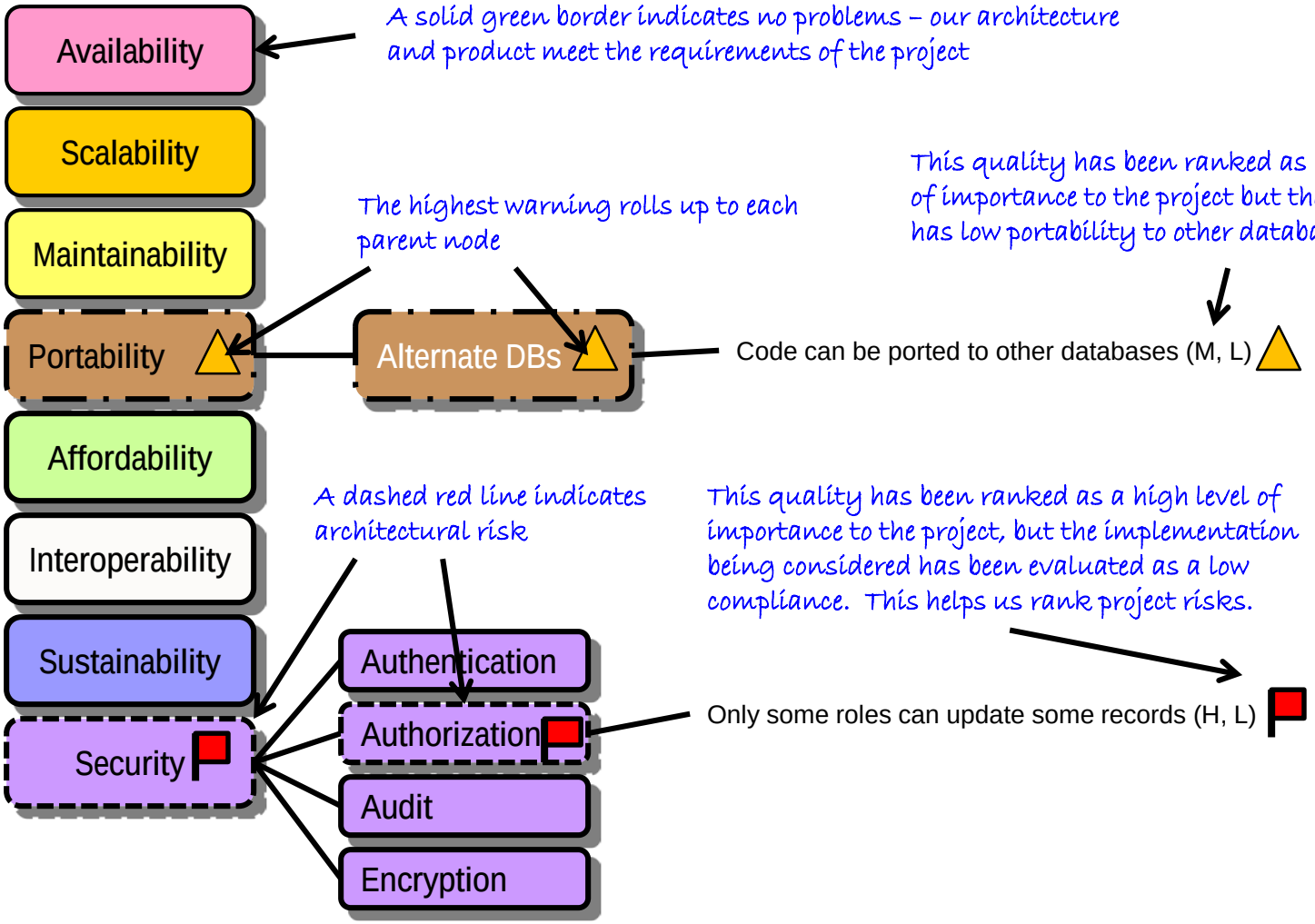
The Big Problem: Standards



- The ability to share code between other NoSQL databases (even within the same type) is **very** limited


Project
1 Scalability
Replication
Autosharding
Rebalancing
Consistent hashing
1 Availability
Replication
Rack awareness
Writes to multiple data centers
Partition tolerance
2 Findability
Fulltext search
Ranked search
KWIC libraries
3 Maintainability
Unit testing tools
Regression tests
3 Affordability
License costs
Annual maintenance costs
Staff training
Operations
2 Interoperability
2 Sustainability
1 Security
Authentication
Authorization
Role-based access control
Dynamic security controls
Audit
Encription
3 Portability
Common query language
Simple API
Standardized interfaces

Using Quality Trees to Communicate Risk



Maintainability

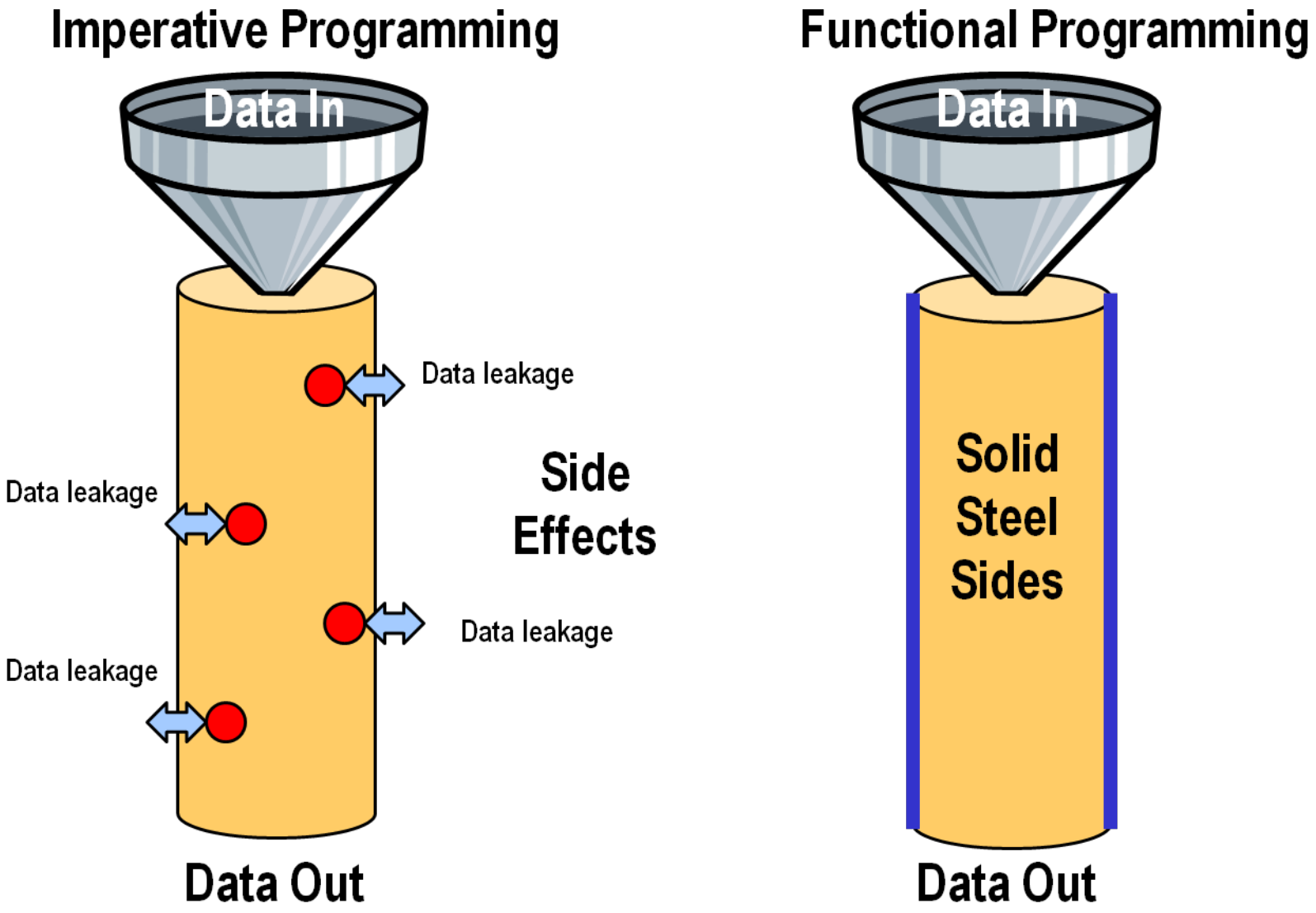
- "…is the ability of the system to undergo changes with a degree of ease. These changes could impact components, services, features, and interfaces when adding or changing the functionality, fixing errors, and meeting new business requirements."

Business Agility

The "big win" for many NoSQL converts

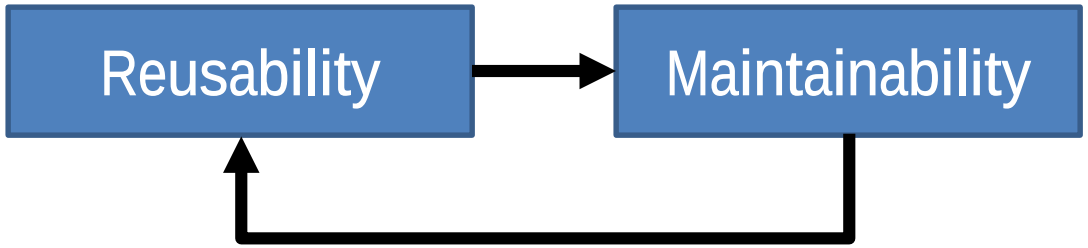
"We came for the scalability, we stayed for the agility"

SAAM has a stronger focus on change impact analysis


Imperative Programming
Data In
Data leakage
Data leakage
Side
Effects
Data leakage
Data leakage
Data Out
Functional Programming
Data In
Solid
Steel
Sides
Data Out

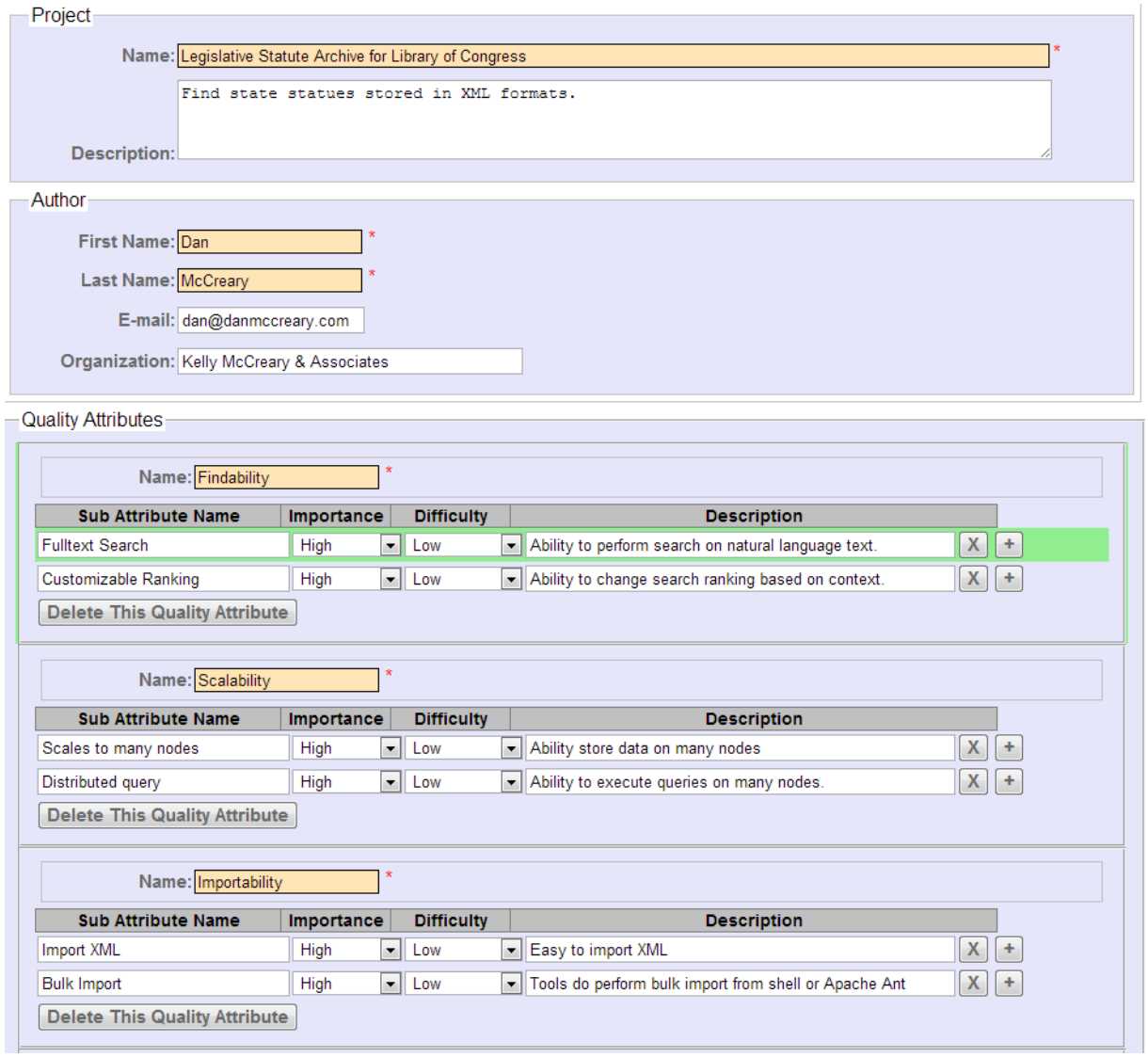
Reusability

- defines the capability for components and subsystems to be suitable for use in other applications and in other scenarios. Reusability minimizes the duplication of components and also the implementation time.



Take Home: reusing transforms (MapReduce etc.) is the key to agility in Big Data

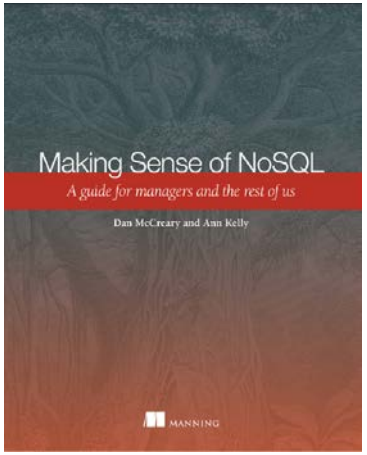
XForms Quality Attribute Utility Tree



Summary

- We need solution architects that are trained in the pros and cons of multiple database architectures
- Select a database **architecture** first, then select a product
- "One size fits all" will not keep organizations competitive
- ATAM (and SAAM) are great processes to help understand the alternatives and objectively weigh the consequences of architectural decisions

Reference



- Making Sense of NoSQL
- Manning Publications
- Available in PDF now via MEAP
  - http://manning.com/mccreary
- In print in July, 2013

Ann Kelly

Dan McCreary