CSCU9YQ - NoSQL Databases
Lecture 8: MongoDB Use Cases

Gabriela Ochoa

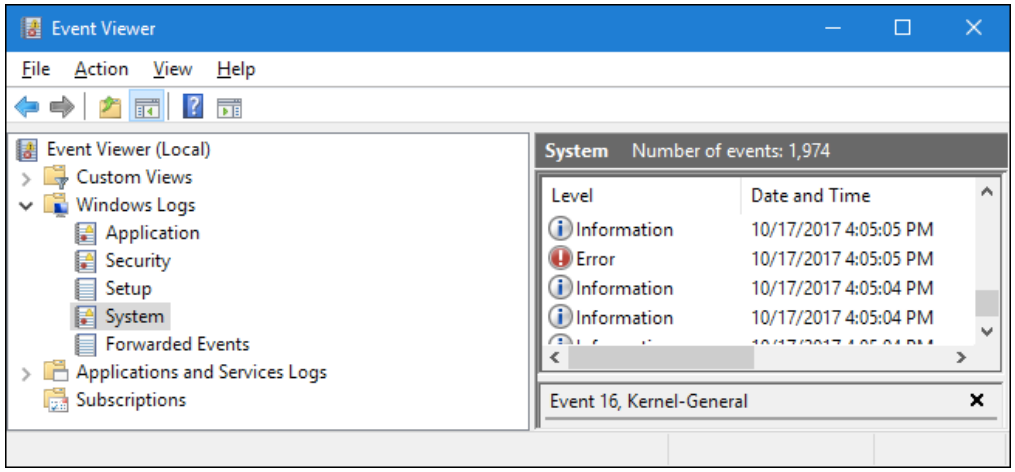
NoSQL Use Cases

- Relational DB
  - will still be around for a long time
  - right kind of DB for handling centralised applications that require sophisticated transaction handling (e.g., general accounting).
- NoSQL DB
  - will continue growing in importance
  - better suited to support widely distributed cloud applications and their specific use cases

Document DB Suitable Use Cases

- Event Logging
- Content Management Systems, Blogging Platforms
- Web Analytics or Real-Time Analytics
- E-commerce Applications

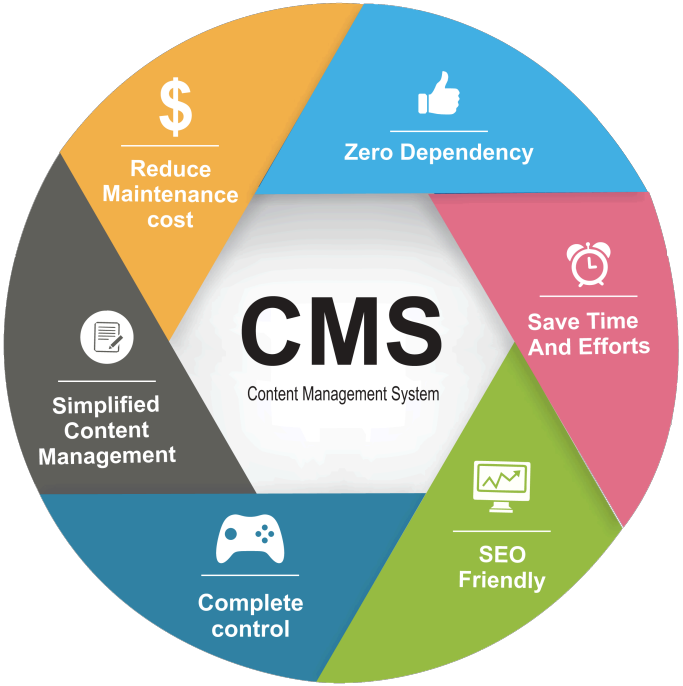
Event Logging



- Different applications want to log events and errors each with their own format
- With a Document DB
  - Documents can store all these different types of events
  - Specially true when the data captured by the events keep changing (flexible schema)
  - Events can by sharded by the application that originated it, or by type of event (loggin, rt

Content Management Systems

- Manage the creation and modification of digital content.
- Typically support multiple users in a collaborative environment



- With a Documents DB
  - JSON formant and flexible schema are useful
  - Easy to store user comments, registrations, profiles and other documents

Web/Real Time Analytics

- Measurement, collection, analysis and reporting of web data for purposes of understanding and optimising web usage.
- With a Documents DB
  - Easy to store page views or unique visitors, as part of the document can be updated
  - New metrics can be easily added without schema changes.

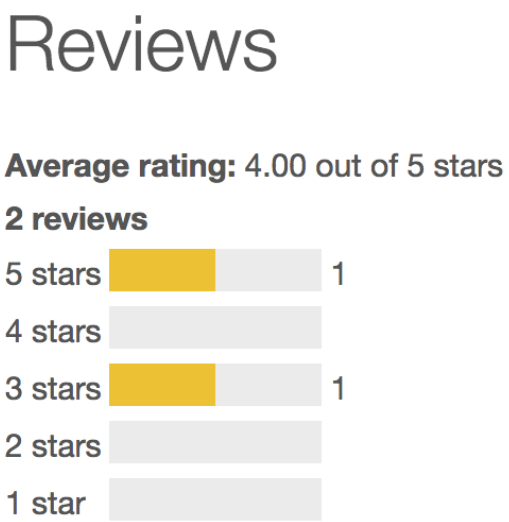
E-Commerce Applications

- Manage the creation and modification of digital content.
- Typically support multiple users in a collaborative environment

- With a Documents DB
  - JSON formant and flexible schema are useful
  - Easy to store user comments, registrations, profiles and other documents

Specific Example: Product Review Site

- Undergraduate project supervised by Dr. K. Swingler
- Web front end and Python, MongoDB at server
- Allows users to review products
- Instead of writing sentences, they are invited to enter key, value pairs that describe features of the product



Product Review Site

- Example of data provided:
  - Product: JP Excite Ride Surfboard
  - Value: A little expensive, but worth it, 3/5
  - Control: Very easy to control, 5/5
  - Speed: Quite fast for a freestyle board, 4/5
- The identity of the products and the features that they contain cannot be known at design time, so schemaless design is needed

JSON Object

```
{Product: “JP Excite Ride Surfboard”,
Category:”Windsurf boards”,
ReviewerID:2354242,
{Value: {Description:”A little expensive,
but worth it”, Score:3/5}},
{Control:{Description: “Very easy to
control”, Value:5/5}},
{Speed: {Description:”Quite fast for a
freestyle board”, 4/5}}}
```

Use Cases

- Defining use cases is essential in NoSQL design as it informs decisions of:
  - What the aggregate model should be
  - What fields are indexed
  - How data should be sharded

Use Case 1

- *Description*: A user wants all the reviews of a single known product
- *Prevalence*: A common search
- *Suitable Aggregation*: Obvious document structure is one document per product with an array of review documents within it

Use Case 2

- *Description*: A user knows the category of the product, but needs help choosing
- *Prevalence*: A common search
- *Suitable Aggregation*: Keeping all products in a category in one document is not practical
- *Suitable Sharding*: Keeping all products in a category on the same shard is sensible

Use Case 3

- *Description*: A user wants to see all the reviews they (or another chosen user) have made
- *Prevalence*: A rare search
- *Suitable Aggregation*: To optimise this search, aggregate by user with all a user's reviews in the same document

Decision

- Aggregate by product seems most sensible:

```
{Product: “JP Excite Ride Surfboard”,
Category:”Windsurf boards”,
Reviews:[. . .]}
```

Embed or Link?

- Should the reviews array contain full reviews or links?
- To get all reviews in one read, best to embed
- No need for normalised form, as reviews rarely get updated and are stored only once
- Link to reviewer data, don’t embed

Solution

```
{Product: “JP Excite Ride Surfboard”,
Category:”Windsurf boards”,
OverallScore:4/5,
Reviews:
[{ReviewerID:2352,
Value:{Description:”Good”,Score:4/5},
Speed:{Description:”Slow”,Score:2/5}},
{ReviewerID:2352,
Value:{Description:”Great”,Score:4/5},
Speed:{Description:”Fast”,Score:4/5}}
]}
```

Challenges: Knowing the keys

- If each review can have different field names, how can I know what to search for?
- Could maintain an array of keys in the product document:

```
{Product: “JP Excite Ride Surfboard”,
Category:”Windsurf boards”,
Keys:[“Value”,”Speed”,”Control”], …}
```

- Similarly, when new reviews are entered, possible available keys can be shown to try and avoid synonyms

Examples from MongoDB.org

Single View	Internet of Things	Mobile	Real-Time Analytics
MetLife	ENERNOC	FOURSQUARE	McAfee An Intel Company
stripe, CERN	BOSCH, Silver Spring Networks	O2, ADP	salesforce
Telefonica	Tech Mahindra	Parse	CRITTERCISM

Catalog	Personalization	Content Management
CARFAX Vehicle History Reports	ticketmaster	VIACOM
Under Armour, orange	intuit, Expedia	Forbes, ebay
OTTO	eHarmony	The National Archives

Figure 1: MongoDB Use Cases

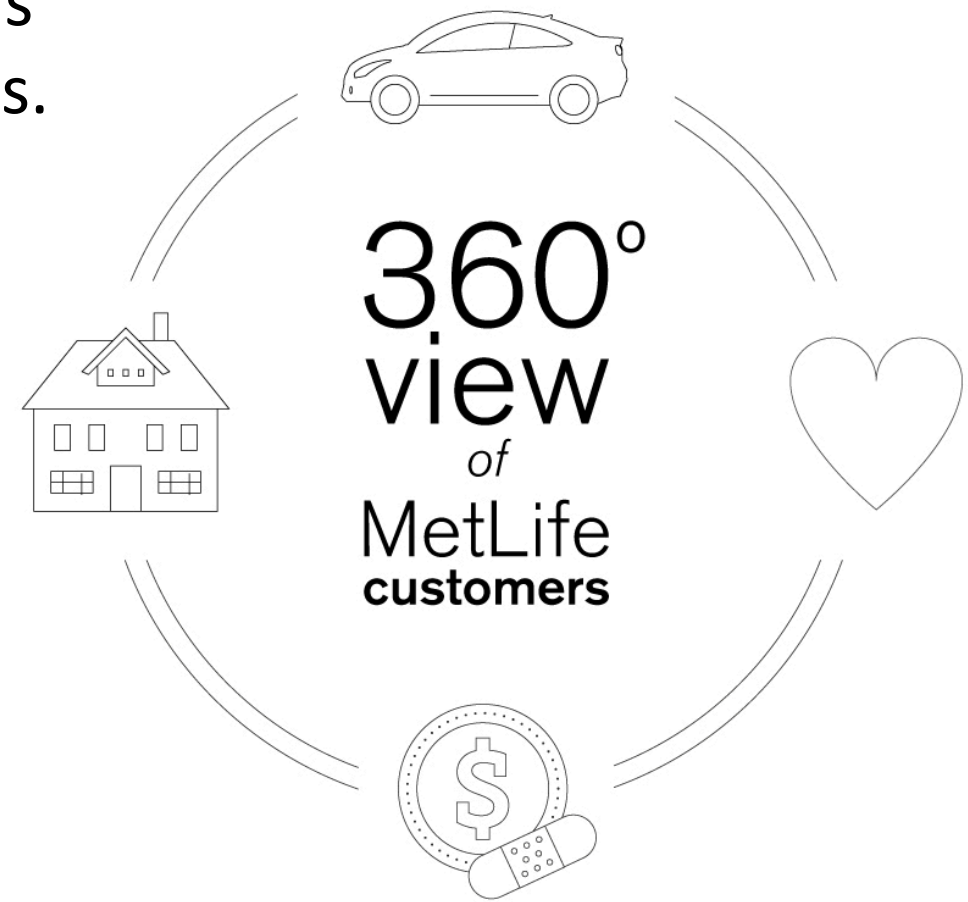
Examples from MongoDB.org

- MongoDB White Papers
  - https://www.mongodb.com/white-papers
- Fortune 500 (world's 500 largest companies) companies and startups are using MongoDB
  - to create new types of applications
  - improve customer experience
  - accelerate time to market and reduce costs.

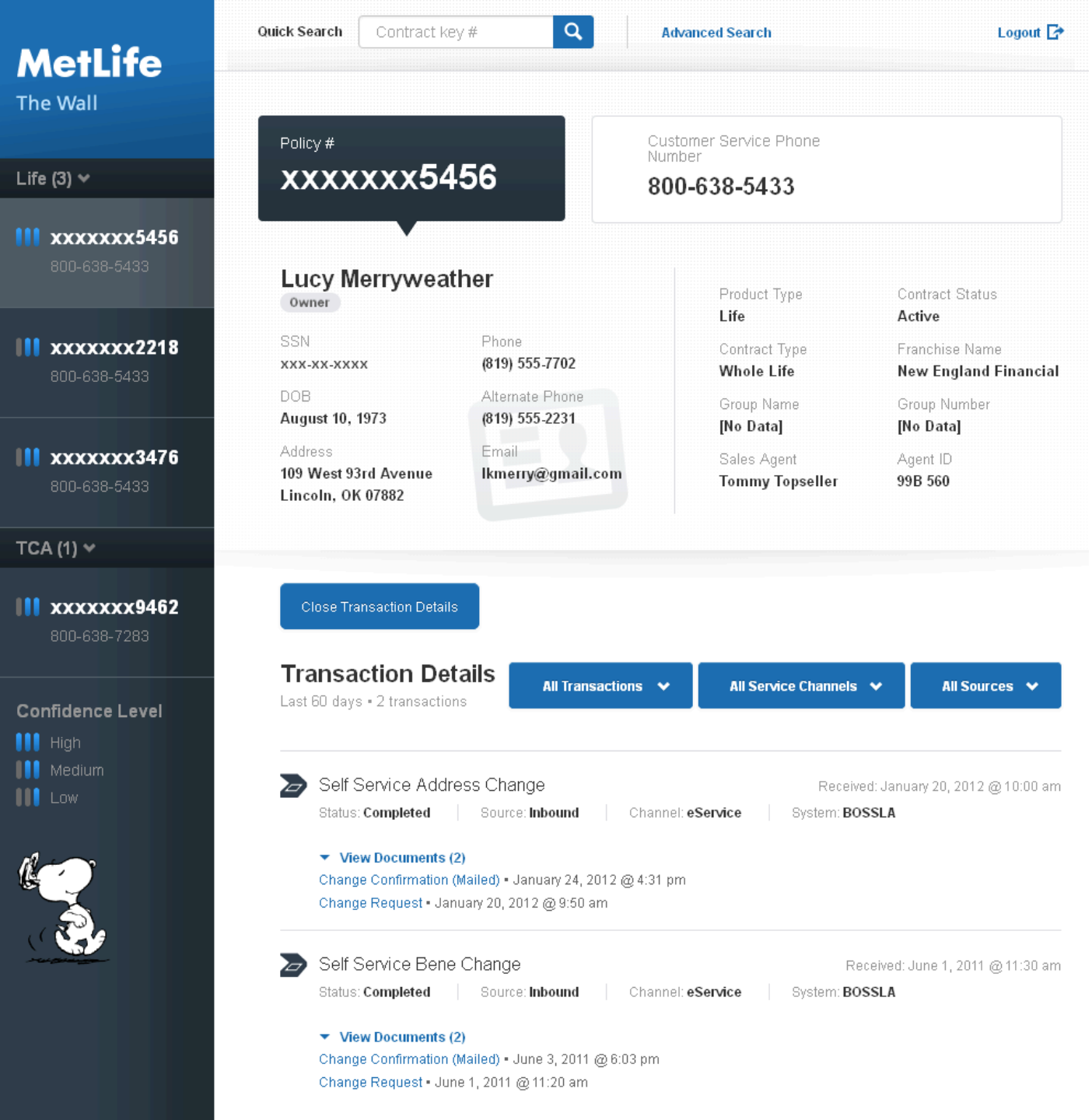
MongoDB usage statistics

- Has been downloaded over 15 million times
- 2,000+ customers
- Significant customers
  - ADP, Bosch, Cisco, eBay, Expedia, MetLife, Telefonica, Ticketmaster and Verizon.
- Among the Fortune 500
  - Financial Services (40), Retailers (15), Telecomm (15), Technology (15), Healthcare (15), Electronics (10), Media Entertainment (10)

- MetLife is one of the world's largest insurance companies.
- Customer data was ‘siloed’ (isolated), difficult for call centre representatives to resolve customer issues efficiently
- 360 View developed with MongoDB in 3 months
- Mongo DB allowed the aggregation of data in a single data hib



The 360-degree customer view: creating a complete view by aggregating data from the various touch points that a customer may use to contact a company to purchase products and receive service and support.



The Wall App aggregates over 100 million customers, 100 products and over 70 source systems into a single data hub.

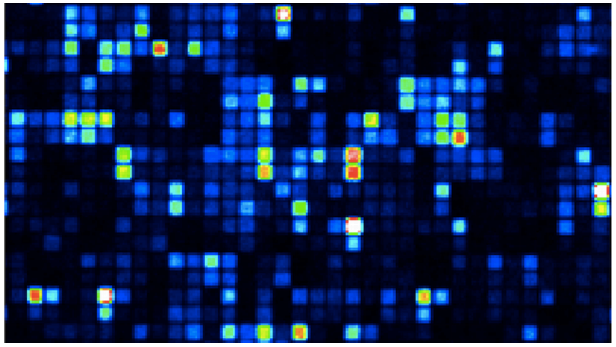
It presents the data in an intuitive, Facebook-like interface for customer service representatives

Viewed by MetLife and the industry as an overwhelming success

Genentech

- Develops drugs to treat significant medical conditions
  - mainly cancer, but also Alzheimer, Parkinson, flu, hepatitis, arthritis
- Uses genetic information to produce drugs
  - Need to provide researchers with new genetic strains so as to understand the cause of diseases and to test new drugs.

Genentech



- Using a relational DB, the Genentech team needed to change the schema every time they introduced a new experiment
- MongoDB is able to capture the variety of data generated by genetic tests and integrate it with the existing Oracle DB
- This reduced development time from months to weeks or even days.

Conclusion

- Making the right choice of database can deliver quantified business results.
  - Shorter development time
  - Improved customer experience
  - Enabling new types of applications
  - Achieving higher revenues or reducing costs