MongoDB, Mongoose and Cloud Storage

Frank Walsh, Diarmuid O’Connor

Agenda

- Cloud Databases
- MongoDB
- Mongoose
- Mongo in the cloud

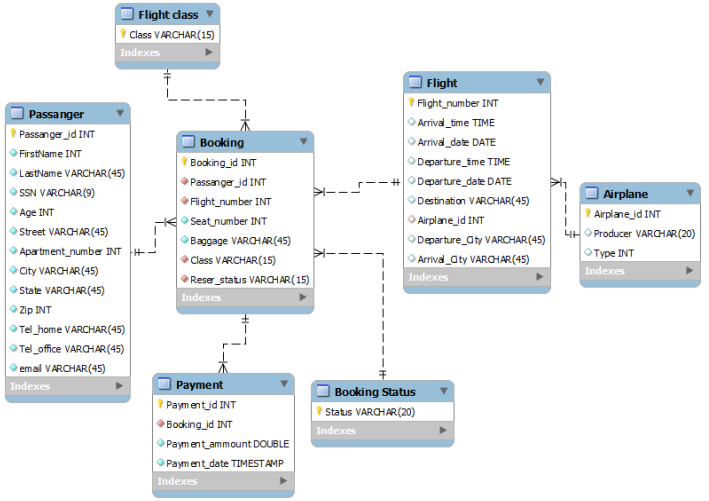
Databases in Enterprise Apps

- Most data driven enterprise applications need a database
  - Persistence: storage of data
  - Concurrency: many applications sharing the data at once.
  - Integration: multiple systems using the same DB
- Enterprise Application DBs require backups, fail over, maintenance, capacity provisioning.
  - Traditionally handled by a Database Administrator (the DBA).

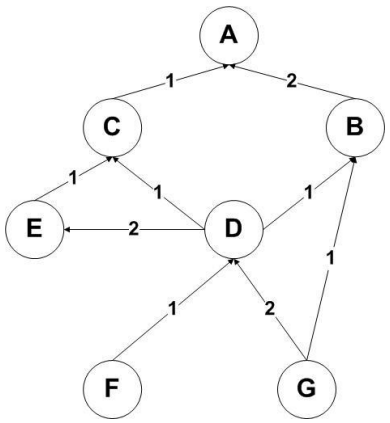
Structured & Unstructured Data

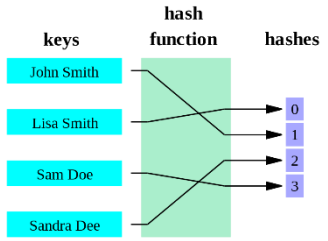
- Relational Databases:
  - Organise data into structured tables and rows
  - Relations have to be simple, they cannot contain any structure such as a nested record or a list
- In memory data structures
  - Much more varied structure
  - Lists, Queues, Stacks, Graphs, Hashing

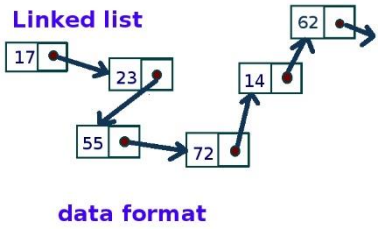
Relational Database



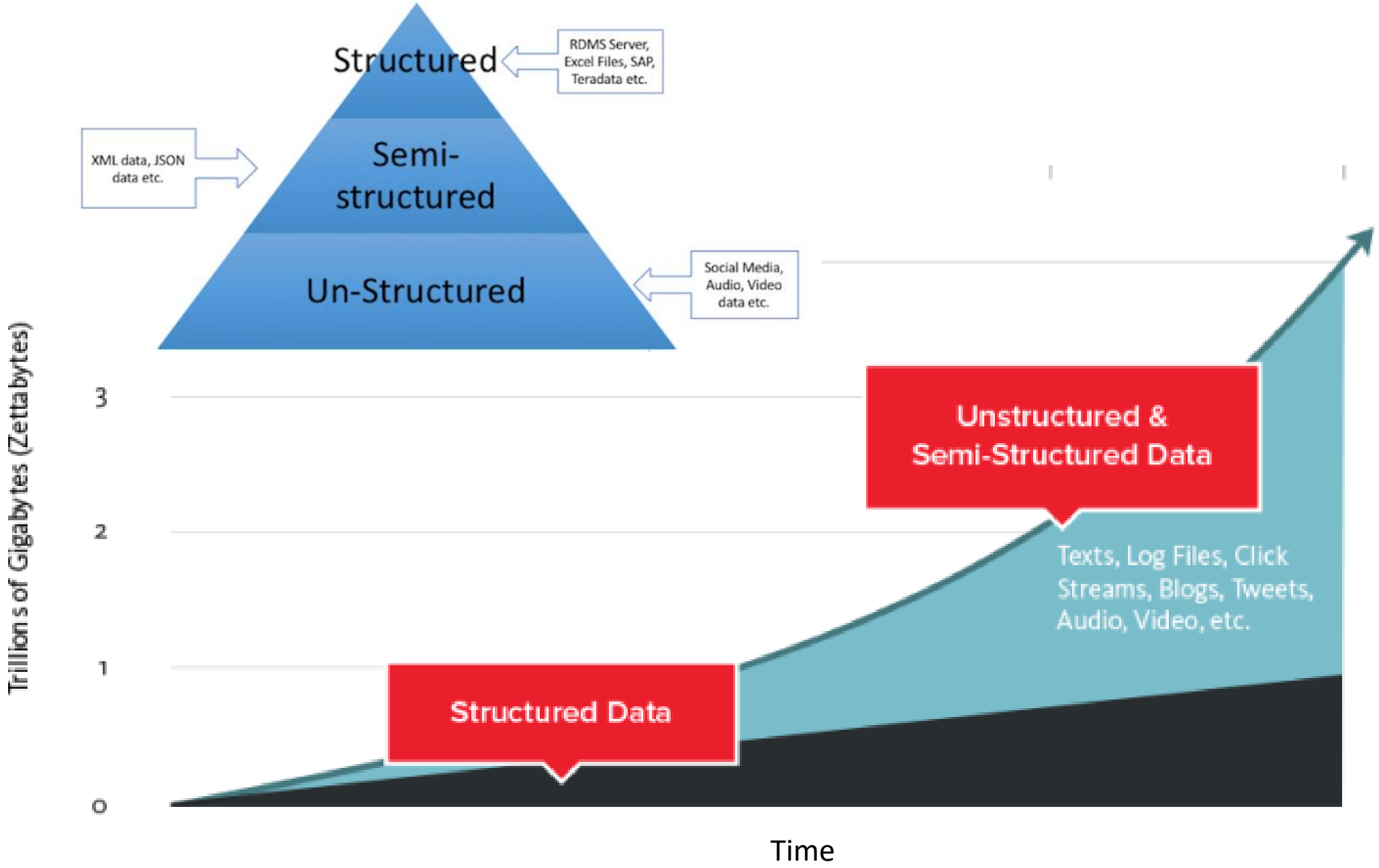
In Memory Data Structures







data format


Structured
RDMS Server, Excel Files, SAP, Teradata etc.
Semi-structured
XML data, JSON data etc.
Un-Structured
Social Media, Audio, Video data etc.
Trillions of Gigabytes (Zetta bytes)
3
2
1
0
Unstructured & Semi-Structured Data
Texts, Log Files, Click Streams, Blogs, Tweets, Audio, Video, etc.
Structured Data
Time

Databases in the Cloud

- For some apps, a traditional relational database may not be the best fit
  - Organisations are capturing more data and processing it quicker – can be expensive/difficult on traditional DB
  - Traditionally, relational database is designed to run on a single machine in predicable environment
  - May be economic to run large data and computing loads on clusters.
  - Hard to estimate scaling requirements, particularly if it's a web app?
  - Are you going to do Data mining?
- One approach is to use the **Cloud** for you DB
  - Designed for scale
  - Can be outsourced so you don't have to deal with infrastructure requirements.

Cloud DB Advantages

- Removes Management costs
- Inherently scalable
- No need to define schemas(if NoSQL) etc.
- Lots of Cloud DB offerings out there
  - SQL based
  - NoSQL based
- If organisation policy/standards do not allow outsourcing:
  - Can host yourself, most NoSQL DBs are free.

Cloud Database Practices

- Drop Consistency
  - this makes distributed systems much easier to build
- Drop SQL and the relational model
  - simpler structures are easier to distribute:
    - key/value pairs
    - **structured documents**
    - **pseudo-tables**
    - tend to be schema-free, accepting data as-is
- Offer HTTP interfaces using XML or **JSON**
  - Web APIs!!!

Designing Distributed Data

- App data is not homogeneous
  - some kinds of data will be much larger
- consider using different databases for different requirements.
- user details,billing - needs consistency
  - require traditional database
- user data,content - needs partition tolerance
  - replicate to keep safe
- analytics,sessions - needs availability
  - "eventually consistent" is good enough