UpStage Team

Investigation of  
Data Storages

UpStage 2015 S2

# Table of Contents

[Table of Contents 1](#_Toc431944404)

[Introduction 2](#_Toc431944405)

[Data Storage Requirements 4](#_Toc431944406)

[XML 5](#_Toc431944407)

[Relational Database 7](#_Toc431944408)

[JSON 8](#_Toc431944409)

[Rationale for choosing JSON 9](#_Toc431944410)

[References 10](#_Toc431944411)

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Changes |
| 1 | September 21, 2015 | Alyssa Byun, and William Stokes |  |
| 2 | October 1, 2015 | Alyssa Byun, and Siatua Uili | * Introduction added * More criteria added * References added * JSON added * Plain text file removed * Rationale changed |

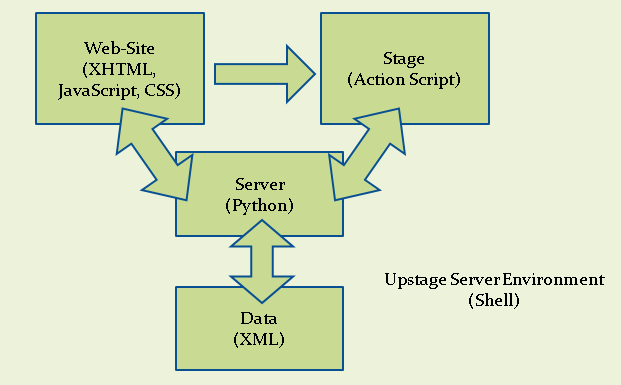
# Introduction

It is important to take note of the persistent data in a website to analyse which data storage system is the best suitable candidate for a project.

**UpStage’s persistent data:**

* Media types:
  + Audio – eg. Avatar voice
  + Image – eg. Avatar
* Object information:
  + Audio
  + Avatar
  + Backdrop
  + User
  + Prop
  + Stage
  + Performance

**Current System Architecture** (from the 2013 UpStage induction PowerPoint)



Data storage in the current implementation of UpStage uses XML files to list all the different objects then each object has a folder with its information for instance player:name- name, password- \*\*\*\*, email- [name@mail.com](mailto:name@mail.com). These objects have been listed under persistent data as object information.

**New Storage requirements:**

These are the requirements for data storage we will be looking into:

*Stores text data:* The persistent data in UpStage is mainly information on the objects and then there are location paths of the media. The UpStage server holds folders with the persistent media files and this is used as the source when coding hence why storage will need to be able store text data (media location path).

*Performance:* There are different ways to store data, but is it suitable for web applications like UpStage?

*Security:* At the moment UpStage faces problems with users being able to delete others avatars and this is why security is a criteria for a new system so it is able to restrict direct access to the data (currently a problem with the existing UpStage).

**Candidates:**

For future reference we mention here what we have chosen to study and why others have not been considered.

We have decided to investigate XML since it is the current data storage so its existing structure could possibly be reused if it suits the research requirements. Relational database has been identified as a useful tool for organising data that is connected through relationships which could be useful in linking together the different objects in UpStage. (“Relational database (RDB),” n.d.). JSON is a considerable candidate due to the fact that many have used it in place of XML (“JSON”, 2015).

Other options we have failed to include in this research have not met some of the criteria such as Object Oriented Databases for its inability to store strings (text data) as it can only store data in forms of objects with attributes and methods (“Object oriented databases,” n.d.). Plain text was another candidate but was eliminated from further study due to its lack of security (“Plain text passwords,” 2015).

# Data Storage Requirements

* Performance
  + How fast it is
  + How it suits web application
  + How much it requires the system resources
* Security
  + Restrict direct access to data
* Suitable for storing text data

# XML

Extensible Markup Language (XML) is a markup language (much like HTML) that defines a set of rules for encoding documents in a format which is both human and machine readable (“XML”, 2015). It was designed to store and transport data (“W3schools.com”, n.d.).

**General**

Pros:

* Self-describing:  
  It can be used to define the common traits. XML contains not only the data, but also the necessary metadata.
* The data is simply written in such a way to allow machines to read it and make it accessible to a database.
* Human readable:  
  - effectively working with unstructured data  
  - allows team to quickly understand the data
* Hierarchical:  
  It contains the information about the relationship of data items to each other in the form of the hierarchy. It does not require any primary key or foreign key relationships.

**Performance**

Cons:

* It is not suitable for massive data as XML files are just text files which needs to be read, parsed and written to. It would be slow with a large data.
* Relatively wordy compared to JSON (results in more data for the same amount of information).
* All new browsers have built-in XML parser but it could be a bit tricky when it comes to cross-browser XML parsing (Haq, Khan & Hussain, 2015).

**Security**

Cons:

* Low security:  
  Bhuvaneswari and Sujatha (2011) stated that “With a few lines of JavaScript code, you can read an external XML file and update the data content of your web page.” This means that is not suitable for a system that requires a high security.

# Relational Database

Database is an organised collection of schemes, tables, queries, etc. Well-known database management systems include MySQL, Microsoft SQL Server, Oracle and so on.

**General & Performance**

Pros:

* Reduced data redundancy through normalisation.
* Greater data integrity and independence from applications programs.
* Improved data sharing:  
  Database Management System (DBMS) provides an environment in which end users have better access to more and better-managed data.
* Improved data access:  
  DBMS makes it possible to produce quick answers to ad hoc queries. From a database perspective, a query is a specific request issued to the   
  DBMS for data manipulation.
* Handles big sized data quickly compared to XML (“ResearchGate”, 2015).
* Easy generating relations.

Cons:

* Increased costs:  
  Database systems require sophisticated hardware and software and highly skilled personnel. The cost of maintaining the hardware, software and personnel required to operate and manage a database system can be substantial.
* Management complexity:  
  Database systems interface with many different technologies and have a significant impact on resources. The changes introduced by the adoption of a database system must be properly managed.

**Security**

Pros:

* DBMS procides a framework for better enforcement of data privacy and security policies (“My Reading Room”, n.d.).

Cons:

* Damage to database affects virtually all applications programs

# JSON

JavaScript Object Notation (JSON) is an open standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs. It is the primary data format used for asynchronous browser/server communication, largely replacing XML (“JSON”, 2015). Although originally derived from the JavaScript scripting language, JSON is a language-independent data format. Code for parsing and generating JSON data is readily available in many programming languages.

Pros:

* Lightweight data exchange language, as it can represent the same data in XML in fewer bytes
* Faster network transmissions and read/write
* Easy for humans to read and write:  
  - effectively working with unstructured data  
  - allows team to quickly understand the data
* Easy for machines to parse and generate
* Simple syntax, which results in less “markup” overhead compared to XML
* Easy to use with JavaScript as the markup is a subset if JavaScript object literal notation and has the same basic types a JavaScript
* It has distinction between string, number and Boolean (e.g. {“count”:”1”} vs {“count”:1}) while XML does not. The distinction saves a bit of programmer’s work to convert to appropriate type manually. You can define data type of values in XML with XML Schema, but it is complicated and not always available.
* Reasonably succinct compared with XML for instance
* Key-value pairs (KVPs):  
  - perfectly suited to the unstructured data.
* Most of JavaScript libraries and AJAX toolkits have good support of JSON (Haq, Khan & Hussain, 2015).

Cons:

* It does not have a data type
* Encapsulated. You cannot readily stream or append data but have to break it up into individual objects (same as XML).
* Not structured and validated the way XML is

# Rationale for choosing JSON

Any comparison of data storages depends heavily on the data that needs to be stored. The data storage of UpStage will not have to handle large volume of data within the system as mentioned in the introduction. This considered, storing data into a relational database is not a good way to store the persistent data of our system as our research shows that the relational database is more suitable for a large volume of data comparing to XML or JSON.

We narrowed the candidates for the data storage of UpStage to two types; XML and JSON. Russo (2014) illustrates that the usage of JSON will increase more than the usage of XML in near future as people is moving away from a legacy technology like XML into a new domain like JSON. That is because there are few major factors from both the human and machine side; the first is ease of use. Both XML and JOSN are excellent examples of human readable languages, but KVPs are another major improvement found in JSON compared to XML.

Although the UpStage data storage will not have to handle a large sized data, it still only requires a lightweight data exchange language. Russo (2014) also provides an evidence that the JSON code contains less characters than the XML code (30% increase in data sent in the XML – meaning additional costs). Also, JSON allows data to be transferred and would be faster than XML solutions because the compact encoding allows for much smaller data to be transferred (Haq, Khan & Hussain, 2015).

In terms of security, JSON is easily serialised language while XML is not, JSON has stronger security. Therefore, JSON became a good support as an alternative to XML these days (Haq, Khan & Hussain, 2015).

# References

Bhuvaneswari, N. S., & Sujatha, S. (2011). *Integrating SOA and web services.* Aalborg, Denmark: River Publishers.

Haq, Z, U., Khan, G, F., & Hussain, T. (2015). A comprehensive analysis of XML and JSON web technologies. *New Development in Circuits, Systems, Signal Processing, Communications and Computers,* 102-109. Retrieved from <http://www.inase.org/library/2015/vienna/bypaper/CSSCC/CSSCC-14.pdf>

JSON. (n.d.). In *Wikipedia.* Retrieved September 29, 2015, from <https://en.wikipedia.org/wiki/JSON>

MyArch. (2007). *JSON pros and cons.* Retrieved September 29, 2015, from <https://myarch.com/json-pros-and-cons/>

MyReadingRoom. (n.d.). *Advantages and disadvantages of DBMS.* Retrieved October 3, 2015, from <http://www.myreadingroom.co.in/notes-and-studymaterial/226-advantages-and-disadvantages-of-dbms.html>

Object oriented databases. (n.d.). Retrieved from <http://www.comptechdoc.org/independent/database/basicdb/dataobject.html>

Plain text passwords. (2015). Retrieved October 7, 2015, from Pidgin: <https://developer.pidgin.im/wiki/PlainTextPasswords>

Quora. (2011). *What are the pros and cons of JSON?* Retrieved September 29, 2015, from <https://www.quora.com/What-are-the-pros-and-cons-of-JSON>

Relational database. (n.d.). Retrieved October 7, 2015, from Techopedia: <https://www.techopedia.com/definition/1234/relational-database-rdb>

ResearchGate. (2015). *What should one choose XML storage or a relational model?* Retrieved October 5, 2015, from <http://www.researchgate.net/post/When_should_one_choose_XML_storage_or_a_relational_model>

Russo, J. (2014). *What is JSON? An introductory guide.* Retrieved October 5, 2015, from <https://blog.logentries.com/2014/11/what-is-json-an-introductory-guide/>

W3schools.com. (n.d.). *Introduction to XML.* Retrieved October 2, 2015, from <http://www.w3schools.com/xml/xml_whatis.asp>