

# Database for Semantic Style Sheets



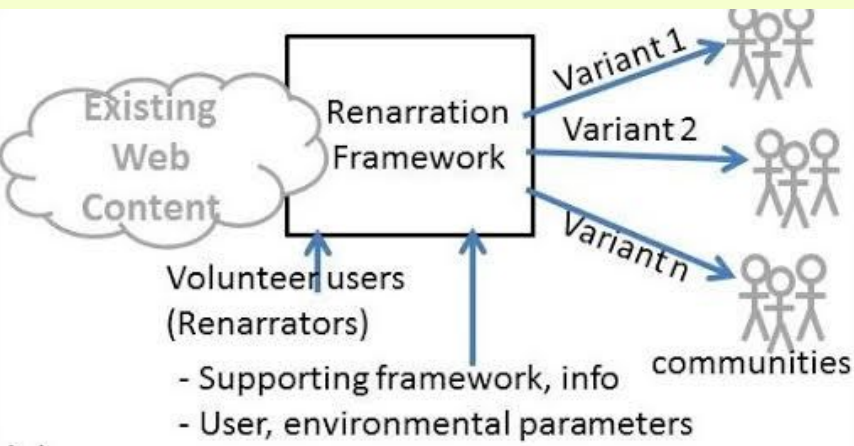
## INTRODUCTION

### Web Accessibility, a problem?

- 54.2% websites on the World Wide Web are in English.<sup>1</sup>
- Nearly 50% internet users are from Asia, a region where English is a foreign language.<sup>2</sup>
- These stats highlight a potential language gap between the presented web content and the browsing user.
- We can also call it a semantic gap.

### The concept of Renarration of the Web<sup>[1] [2] [3]</sup>

- Renarration is the act of transforming the original source content into a new target representation.
- Transforming a source into a new renarrated target could involve simplifying some text, translating it, elaborating on something, adding images, adding new links etc

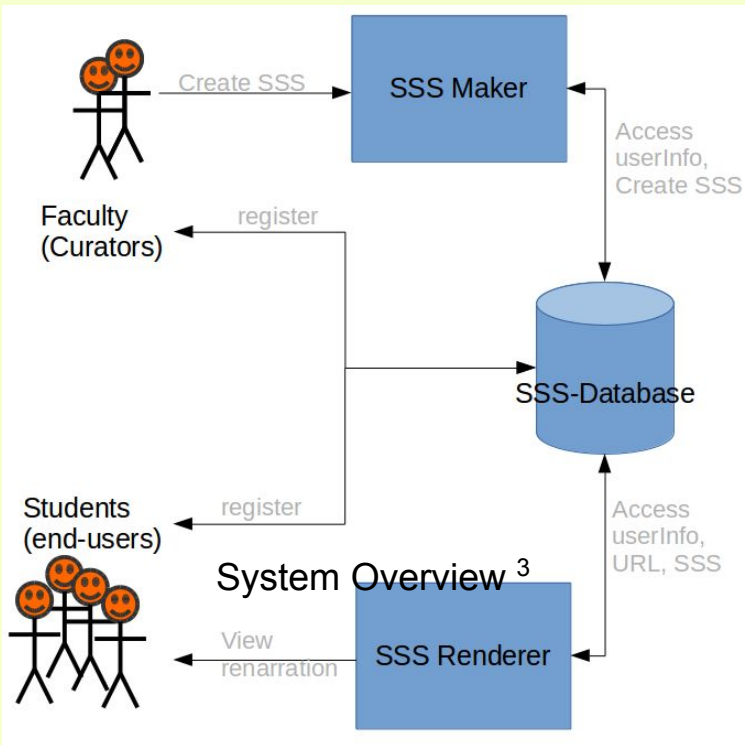


### Semantic Style Sheet (SSS)

It is a notion proposed to address renarration.

### SSS-Framework:

- SSS-Renderer** allows varied users to view the alternative or variant renderings of the renarrated original source.
- SSS-Maker** allows curators to create semantic style sheets to be used for creating variants of an original source.



<sup>1</sup> <http://w3techs.com/technologies/overview/contentlanguage/all>

<sup>2</sup> <http://www.internetworldstats.com/stats.html>

<sup>3</sup> Information from literate documents in the project repository at [github.com/renarration](https://github.com/renarration)

## OBJECTIVES

- Design and develop an API to integrate SSS-Maker and SSS-Renderer with the Database.
- Convert JSON format of SSS-Framework data model schema into JSON-LD.

## METHODS/ SOLUTIONS

### Development of Data store called SSS-Database

#### System Design

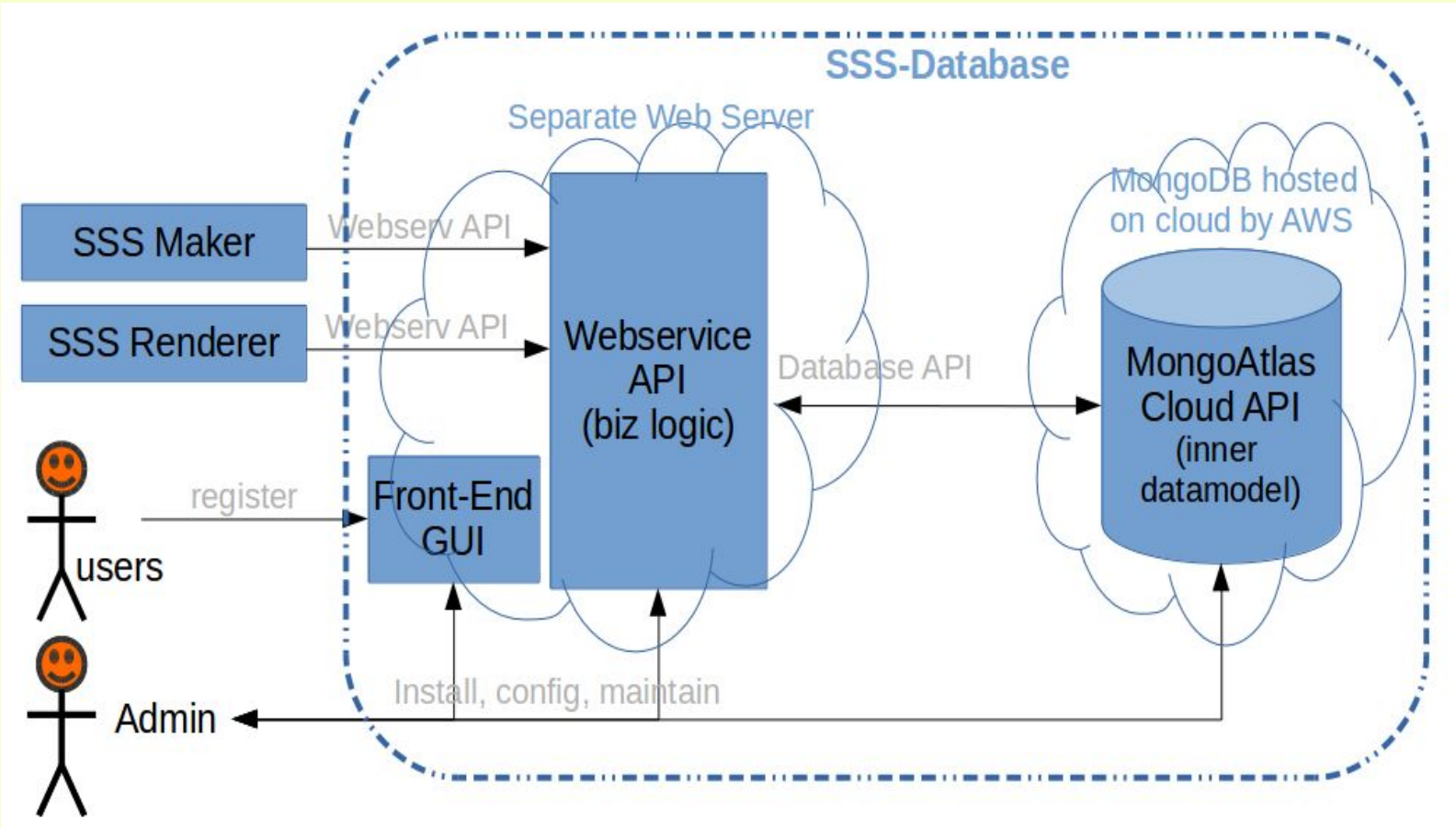
The role of SSS-Database is to receive and store data from SSS-Maker and provide that data to SSS-Renderer.

It has two components:

- Database** to store data.
- Webservice API** which facilitates SSS-Maker and SSS-Renderer to interact with database.
- It contains two parts, these are
  - Functions for interaction with MongoDB API
  - API Endpoints for interaction with SSS-Maker and SSS-Renderer.

#### Stack Choice

- JSON objects are used in SSS-Framework data model.
- To store JSON objects, MongoDB database is used which is a NOSQL database. It can store JSON objects in form of BSON(Binary Script Notation) objects and provides flexibility in schema.
- REST API is developed as a Python-Flask app.
- In Python, PyMongo package is used to interact with MongoDB API. PyMongo gives a lot of flexibility and control over database operations.



#### Result

Endpoints	Description	Method	Payload type	Payload content	Return content	Return Content-type
/api/update/sss/	Add new sss-info document to collection or modify existing one.	POST	application/json	refer here	Raw mongoDB response	application/json
/api/get/sss/username/	Get list of renarrations done by a user.	POST	application/json	{'username': ' '}	List of user sss-info records	application/json
/api/get/sss/sss-name/	Get list of renarrations for given sss-name.	POST	application/json	{'sss-name': ' '}	List of user sss-info records	application/json
/api/get/sss/ren-url/	Get list of renarrations for a given ren-url.	POST	application/json	{'ren-url': ' '}	List of user sss-info records	application/json
/api/get/id=< >	Get user SSS-info with given unique ID.	GET			Document for given id	application/json

### Conversion of SSS-Framework data model schema from JSON to JSON-LD

- To convert JSON to JSON-LD, types and attributes were identified.
- Type is analogous to an entity and attribute is the attribute of the entity.
- To make this model standard, these types were matched with the available vocabulary on schema.org .

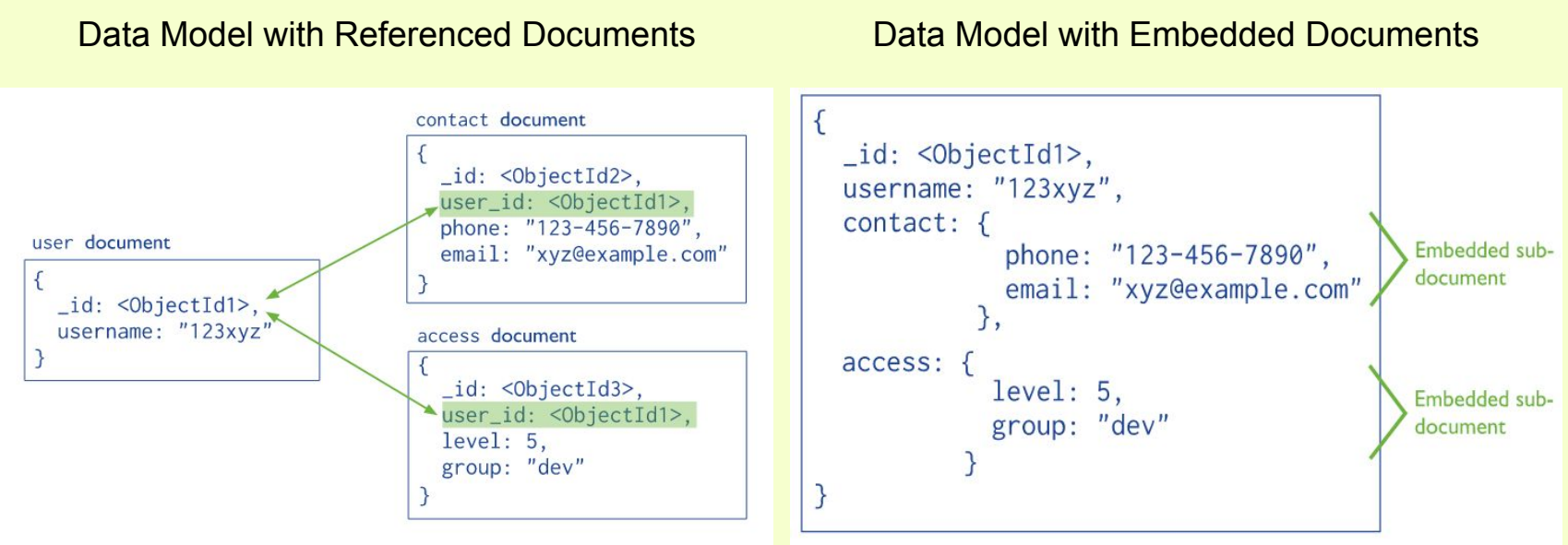
Entity-attribute, type-property Table			
Entity	Attribute	Type	Property
user-info	user-name	Person	name (string)
	degree-program	CollegeOrUniversity	department (string)
	year	??	??
SSSinfo	user-college	CollegeOrUniversity	name (string)
	title	Person	jobTitle (string)
	description	Thing	description (string)
event	name	Thing	name (string)
	url	Thing	url (url string)
	media type	entryPoint	contentType (string)
env-info	start	action	startTime (time)
	stop	action	endTime = time
	browser	??	string
	time of day	??	string



## CHALLENGES FACED

#### Challenges faced in development of SSS-Database

- In the narratives of SSS-Framework data model relational SQL Database terminology was used but generated data was in JSON, and all interactions between components were in JSON.
- To convert the Data Model into NoSQL , a lot of changes were done in narratives.
- Designing the structure of JSON document which is to be stored in collection was challenging because there was 'many to many' relationship between entities and in NoSQL, relations are represented either in reference document or in embedded document form.
- The design decision is taken after analyzing the type of queries to be received by the system.



SOURCE: [http:// docs.mongodb.com/manual/core/data-modeling-introduction/](http://docs.mongodb.com/manual/core/data-modeling-introduction/)

#### Challenges faced in converting JSON to JSON-LD

- Unable to find appropriate types for some of the entities and attributes in schema.org vocabulary.
- Using non-standard semantic tags is not a good practice because they are rarely recognized by the web.

## RESULTS

- Literate documents are completed for the SSS-Database design and implementation.
- Successfully integrated the SSS-Maker and SSS-Renderer.
- Successfully deployed on local lab server.
- Converted data model JSON schema into JSON-LD with nearest appropriate types available on schema.org .
- Verified JSON-LD structure with Google Structured Data Testing Tool.

## CONCLUSION

- Semantic Style Sheet is a powerful concept and there is scope of writing a DSL for renarrating a webpage.<sup>[1]</sup>
- Making the renarrations addressable will give us the ability of adding same renarration to different webpages and assemble multiple renarrations together.
- New types of semantic tags need to be added to schema.org, available tags are not enough.

## LEARNING OUTCOMES

- Design, development and documentation of a RESTful API.
- Integration and deployment of different components of a software together.
- Design and development of MongoDB NoSQL database.
- Hosting MongoDB on a server machine.
- Analyzing the data model design decisions by operation atomicity, and query patterns and trends.

## FUTURE WORK

- In current implementation of SSS-Database, it is assumed that all users already exist in database. The concept of sessions is missing. It will be added in upcoming versions.
- Need to define custom semantic tags to fully convert JSON to semantic structure.
- Develop a domain specific language to renarrate a web page.

Upcoming Version	Description
V2	Web transformation oriented SSS-Framework
V2.1	Enhance DOM to support hierarchy (i.e. nested DOM elements)
V2.2	Support for visualizing DOM as non-linear graph..

SOURCE: <https://github.com/renarration/sss-framework/blob/master/sss-format.org>

## CITATIONS

[1] Gollapudi Vri Sai Prasad, S. Chimalakonda and V. Choppella, *Towards a Domain-Specific Language for the Renarration of Web Pages*. APSEC 2017 (yet to be published)

[2] Prasad, Gollapudi Vri. "Renarrating Web Content to Increase Web Accessibility." *Proceedings of the 10th International Conference on Theory and Practice of Electronic Governance*. ACM, 2017

[3] Prasad, Gollapudi Vri Sai, T. B. Dinesh, and Venkatesh Choppella. "Overcoming the new accessibility challenges using the sweet framework." *Proceedings of the 11th Web for All Conference*. ACM, 2014



Intern: Raghav Mittal (raghavmittal101@gmail.com),  
Mentor: Sai Gollapudi (saigollapudi1@gmail.com)