



# MedTator: A Lightweight Interactive Multi-Document Annotation Tool

S36: Systems Demonstrations - Improving Workflow:  
Informatics Tools No Researcher Should Be Without

**Huan He, PhD\***

Sunyang Fu, PhD, Liwei Wang, MD, PhD, Andrew Wen, MS, Sijia Liu, PhD, and Hongfang Liu, PhD †

Department of AI and Informatics, Mayo Clinic, Rochester, MN, USA

\* He.Huan@mayo.edu

† Liu.Hongfang@mayo.edu

# Disclosure

- I have no relevant relationships with commercial interests to disclose.

# Learning Objectives

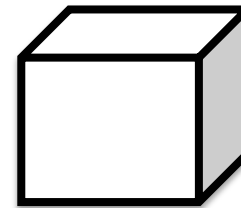
- Learn the serverless architecture for tool development
- Learn how to use MedTator for corpus annotation

# Background

“Hi, I have wonderful data, could you \_\_\_\_\_ with AI?”



“Data”

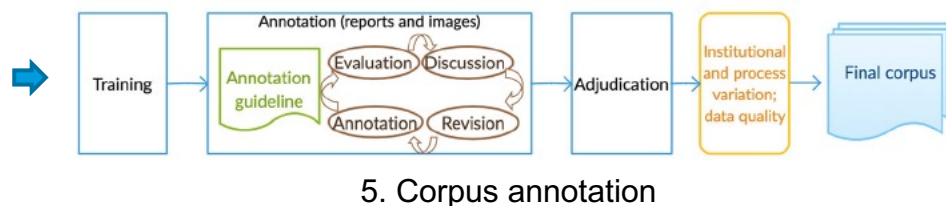
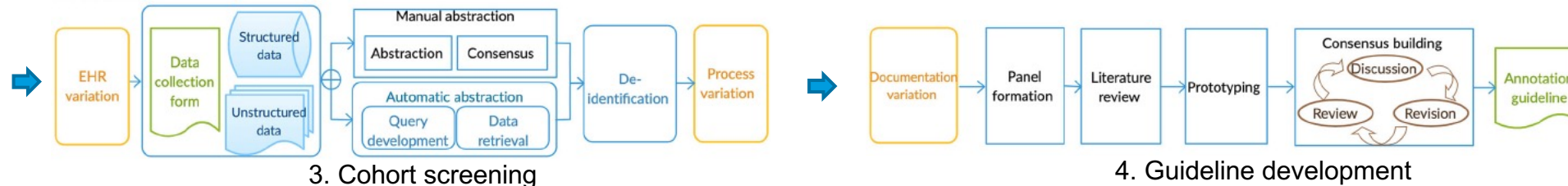


A fancy system

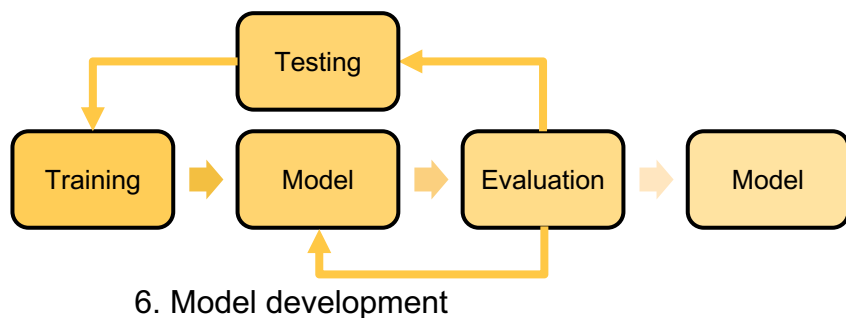
# Adventure starts here



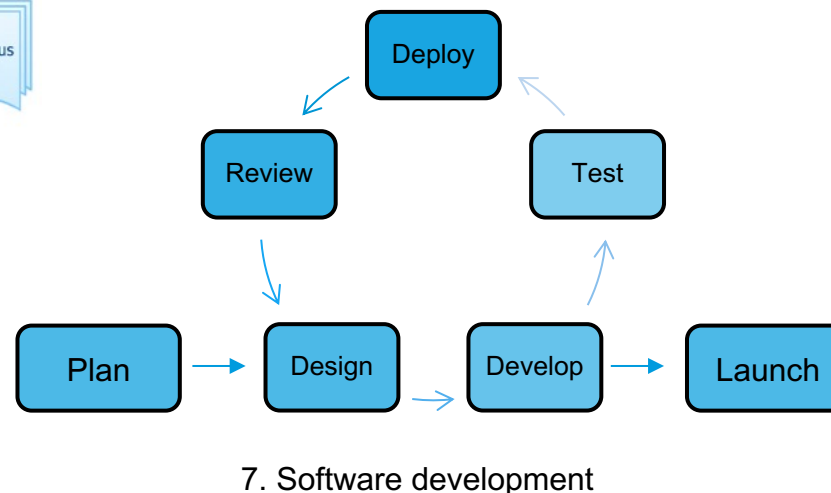
“Data”



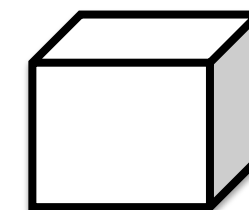
Annotated Corpus



6. Model development



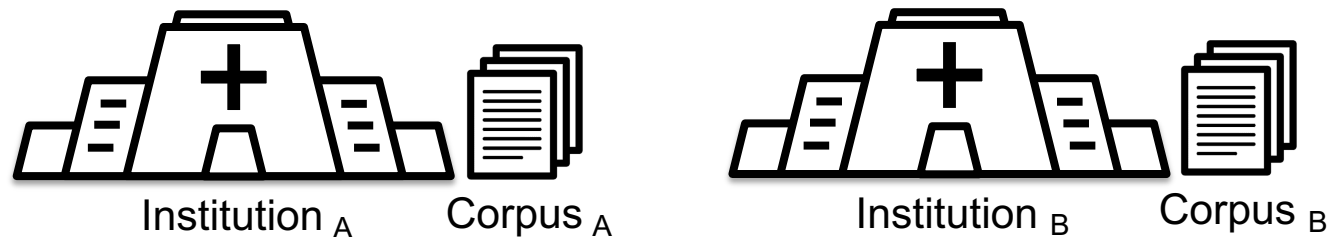
7. Software development



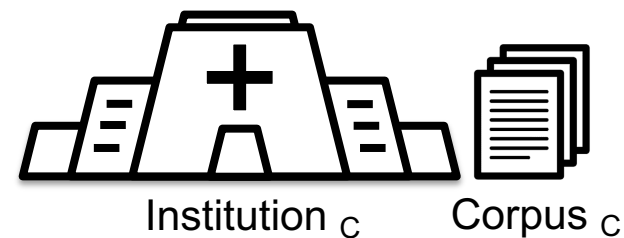
A fancy system

# To conduct multi-site annotation

- Data security
- Corpus management
- Multi-site collaboration



A text annotation tool is needed!



# Text Annotation Tools

<https://github.com/mariananeves/annotation-tools>

MAE

YEDDA

Anafora

brat

INCEPTION

Prodigy

PubTator

TeamTat

FLAT

LabelStudio

and many!

# The Installation of Existing Tools

1. Prepare a server and get permissions
2. Download the package of an annotation tool
3. Read through README, instruction of installation
4. Install the prerequisites
  - Oh, something wrong, version conflict, not found, exception, error ...
5. Install the tool and configure it
  - Oh, something wrong but different error
6. Finally, tool is installed but how to make it work?
7. Read through manual and try again
  - I did what they said in the manual, but it still didn't work, why?



# The Architecture of Existing Tools

- Standalone tools

- MAE
- AGTK
- Analec
- @Note
- BioAnnotate
- Callisto
- Ellogon
- Glozz
- MMAX2
- NOMAD
- RAD

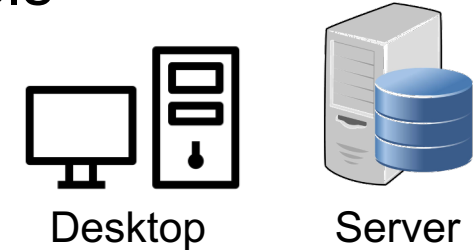
...



- Web-based tools

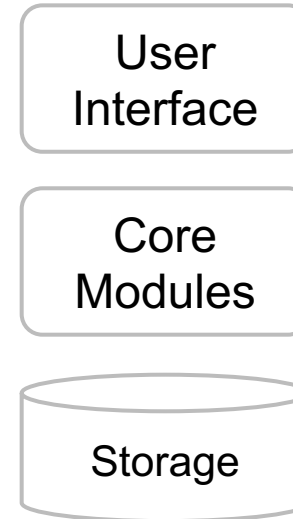
- Anafora
- eHost
- YEDDA
- BioQRator
- brat
- Catma
- FLAT
- WebAnno
- INCEpTION
- MAT
- TextAE

...



# System Architecture Perspective

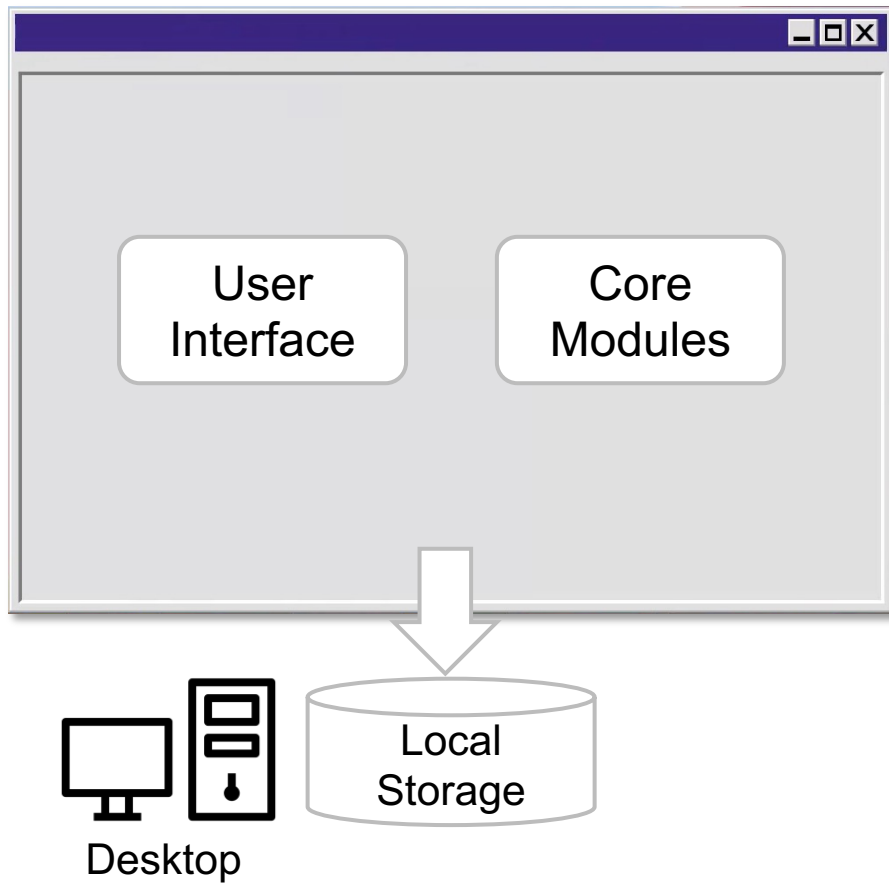
- Input
  - How the data enter system?
- Process
  - What to do with the entered data?
- Output
  - Where to save the processed data?



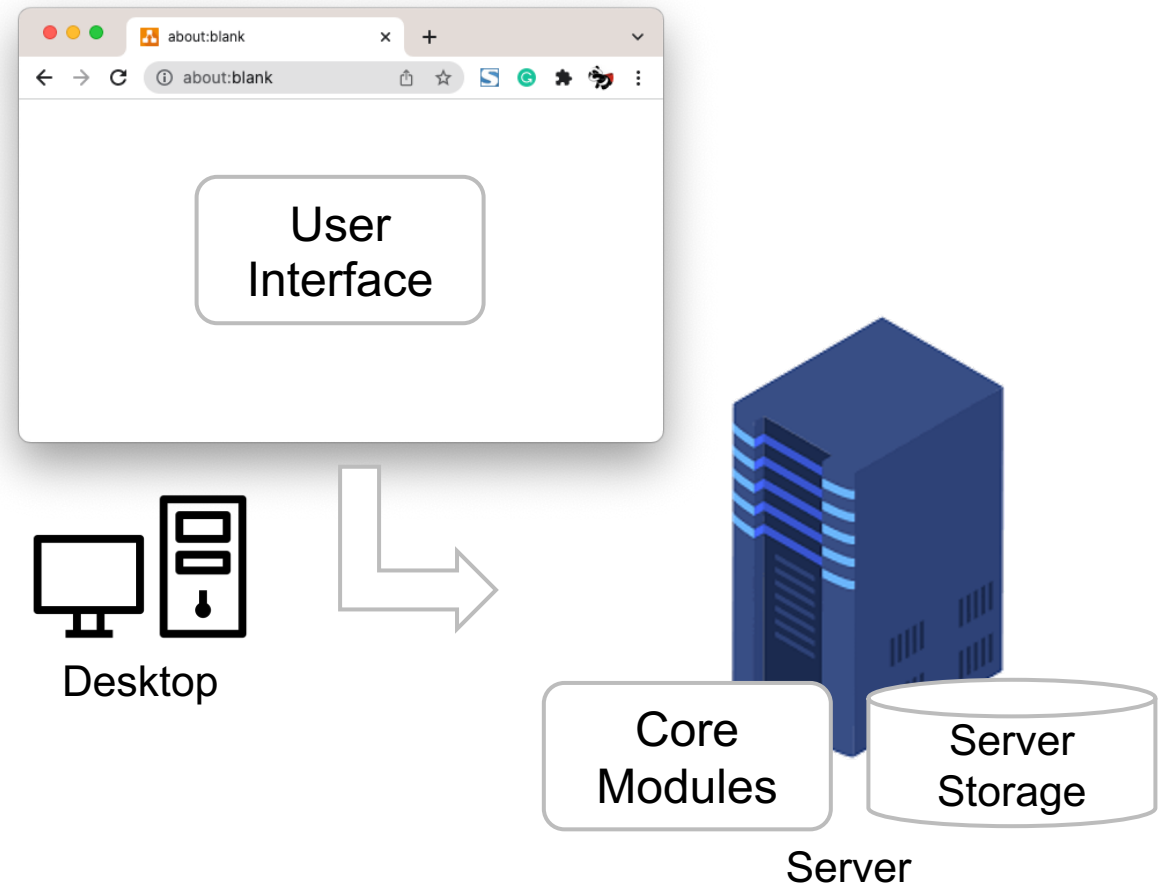
Text annotation Tool

# The Architecture of Existing Tools

- Standalone tools

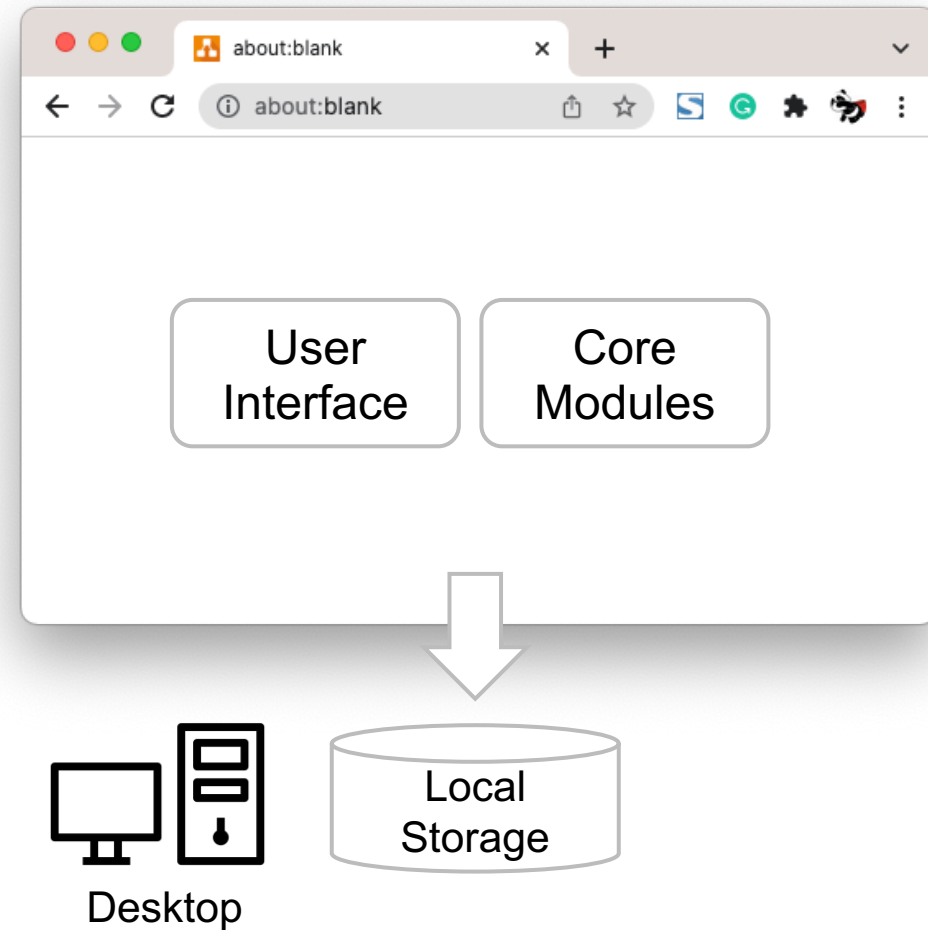


- Web-based tools



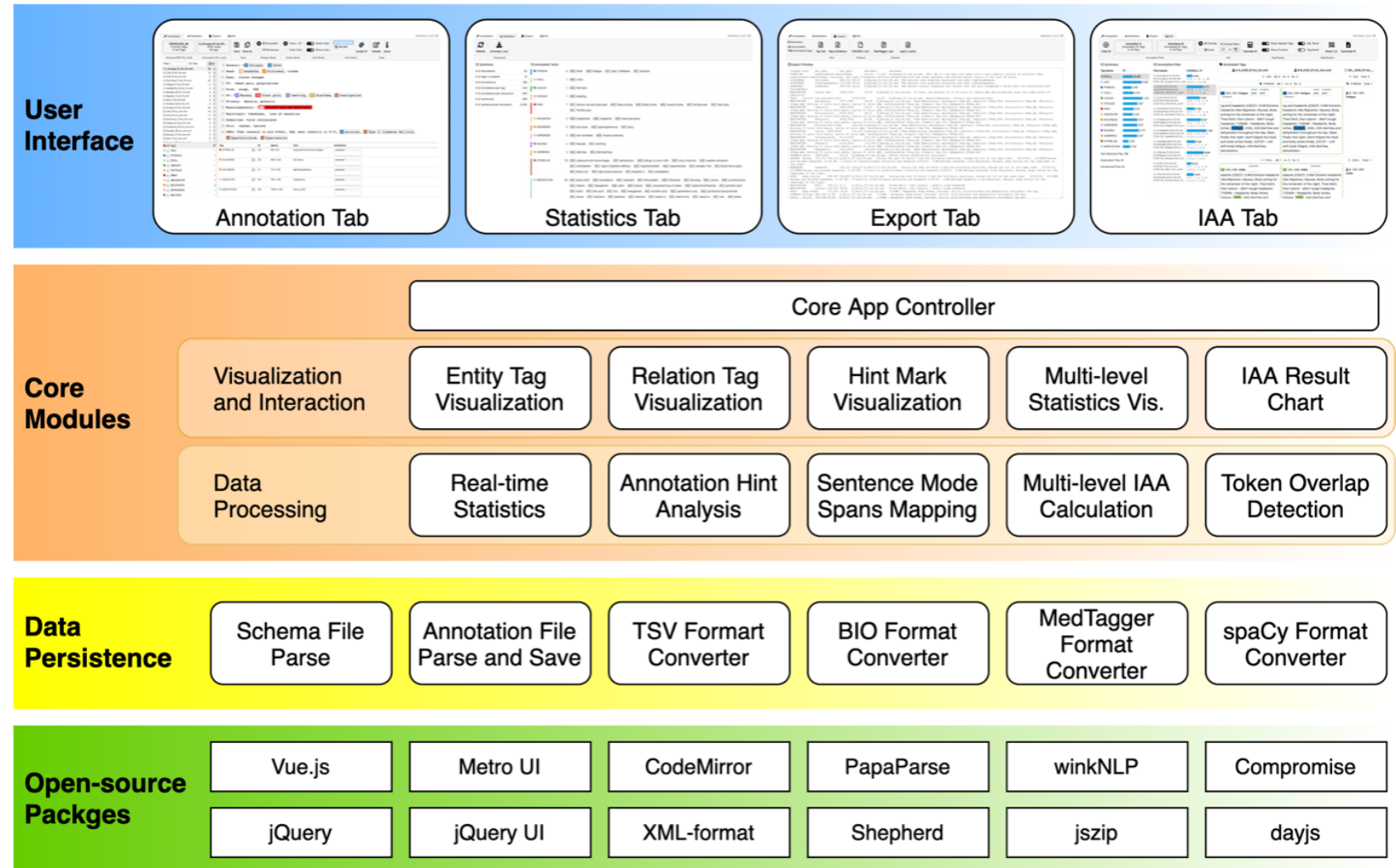
# MedTator - Serverless Architecture

- Similar to standalone tools:
  - Data is saved in local storage
  - UI and core run together
- Similar to web-based tools:
  - UI runs in web browser
- Advantages
  - No need to install



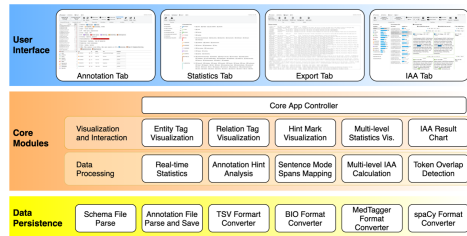
# Functionalities

- Text annotation
- Statistics
- Adjudication
- Export



# Implementation

## Single HTML file



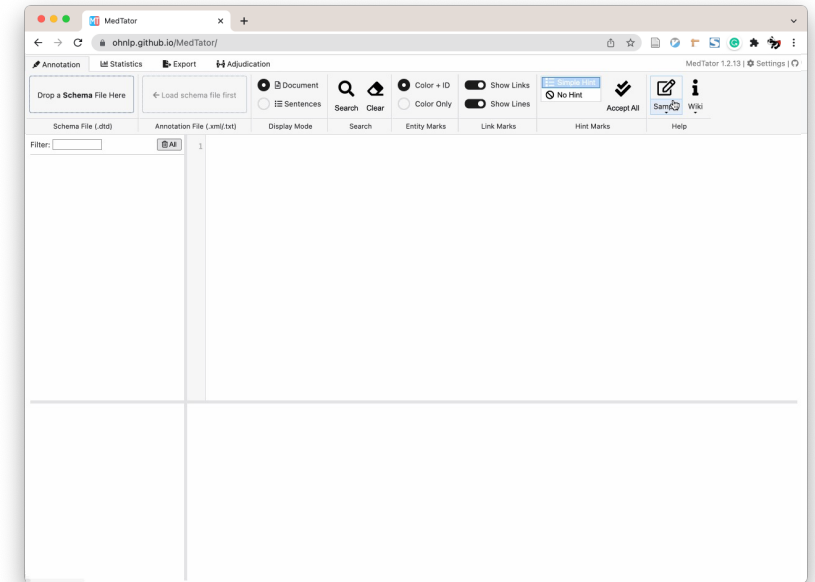
index.html



## JS libraries



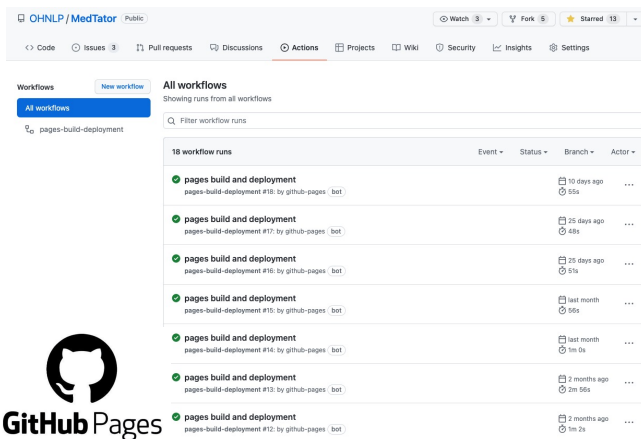
## Web Browser



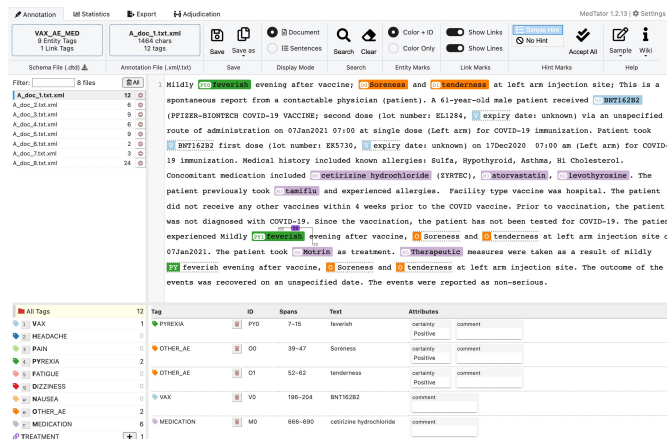
Chromium-based browser

# Advantages

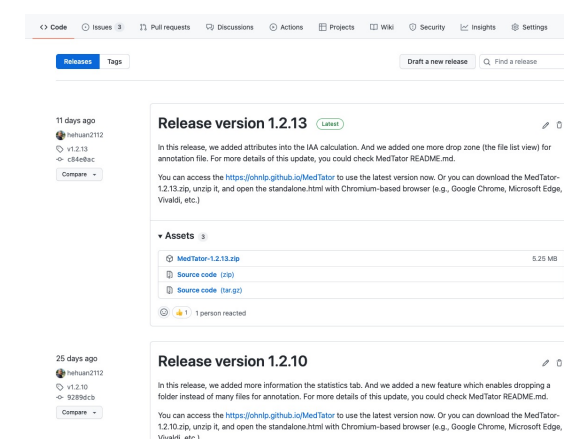
- Researchers are not concerned with capacity planning, management, maintenance, or scaling of virtual machines, or physical servers.
- **Free** public services: GitHub Pages, GitLab Pages, BitBucket Pages...
- Easy to deploy, “install”, and update



`git push` to GitHub Pages to deploy on free domain name  
<https://ohnlp.github.io/MedTator/>



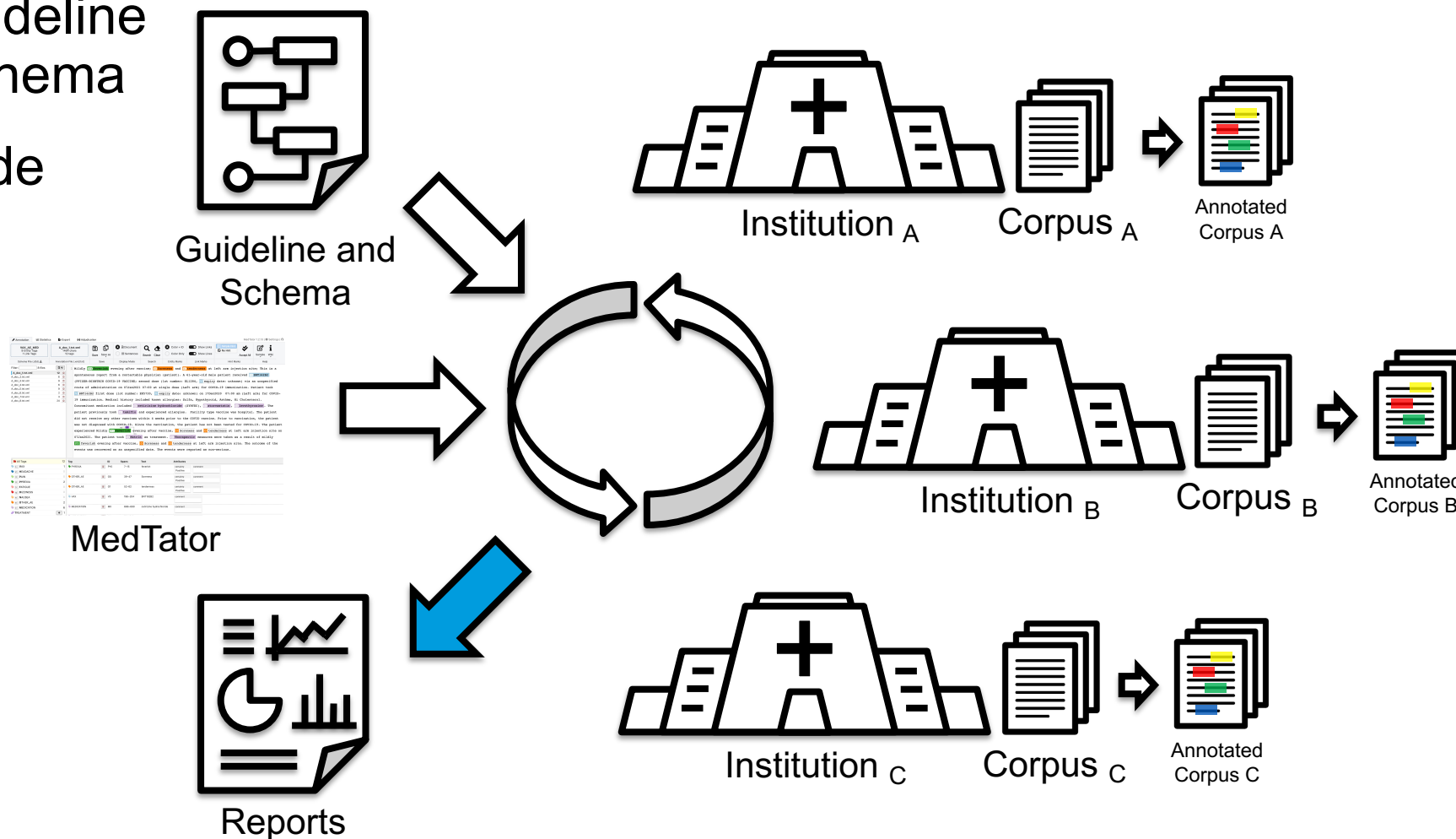
Global access with fast public CDN  
 No need to install  
 (Or run as a local static webpage)



Release new version online

# Advantages - Federated annotation

- Only share the guideline and annotation schema
- No centralized node
- No server cost
- No PHI data transfer
- Same annotation user experience





# Limitations - File management in HTML5

- File system access API  
- Not supported by all browsers

## Browser compatibility

[Report problems with this compatibility data on GitHub](#)

mdn web docs | References | Guides

References > Web APIs > File System Access API

RELATED TOPICS

File System Access API

- Interfaces
  - FileSystemHandle
  - FileSystemFileHandle
  - FileSystemDirectoryHandle
  - FileSystemWritableFileStream
- Methods
  - window.showOpenFilePicker()
  - window.showSaveFilePicker()
  - window.showDirectoryPicker()
  - DataTransferItem.getAsFileSystem...

### File System Access API

Secure context: This feature is available only in [secure contexts](#) (HTTPS), in some or all [supporting browsers](#).

The File System Access API allows read, write and file management capabilities.

### Concepts and Usage

This API allows interaction with files on a user's local device, or on a user-accessible network file system. Core functionality of this API includes reading files, writing or saving files, and access to directory structure.

Most of the interaction with files and directories is accomplished through handles. A parent [FileSystemHandle](#) class helps define two child classes: [FileSystemFileHandle](#) and [FileSystemDirectoryHandle](#), for files and directories respectively.

These handles represent the file or directory on the user's system. You must first gain access to them by showing the user a file or directory picker. The methods which allow this are [window.showOpenFilePicker](#) and [window.showDirectoryPicker](#). Once these are called, the file picker presents itself and the user selects either a file or directory. Once this happens successfully, a

	🖥️						📱					
	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	WebView Android	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet
<a href="#">FileSystemHandle</a> ⚠️	86	86	No	No	72	15.2	No	86	No	No	15.2	14.0
<a href="#">isSameEntry</a> ⚠️	86	86	No	No	72	15.2	No	86	No	No	15.2	14.0
<a href="#">kind</a> ⚠️	86	86	No	No	72	15.2	No	86	No	No	15.2	14.0
<a href="#">name</a> ⚠️	86	86	No	No	72	15.2	No	86	No	No	15.2	14.0
<a href="#">queryPermission</a> ⚠️ ⚠️	86	86	No	No	72	No	No	86	No	No	No	14.0
<a href="#">requestPermission</a> ⚠️ ⚠️	86	86	No	No	72	No	No	86	No	No	No	14.0

✓ Full support    ✗ No support    ⚠️ Experimental. Expect behavior to change in the future.

⚠️ Non-standard. Check cross-browser support before using.

# Limitations - Functionality (for science)

- Limited support in scientific computing libraries for advanced features

	<b>JavaScript</b>	<b>Python</b>
Computation	NumJS Danfo.js std.js	NumPy Pandas SciPy, statsmodels
Machine Learning / NLP	tensorflow.js compromise, winkNLP	scikit-learn, tensorflow spaCy, flair
Data Visualization	D3.js, p5.js ECharts Plotly	Matplotlib, seaborn

# Demonstration

- Live demo with sample datasets:  
<https://ohnlp.github.io/MedTator/>

The screenshot displays the MedTator 1.2.0 interface. The top navigation bar includes 'Annotation', 'Statistics', 'Export', and 'Adjudication'. The main toolbar contains various icons for document management, search, and display options. The central area shows a document with text and highlighted entities. A table at the bottom lists the detected tags.

Tag	ID	Spans	Text	Attributes
OTHER_AE	O0	86-93	anxiety	comment
OTHER_AE	O1	94-105	nervousness	comment
DIZZINESS	D0	199-204	dizzy	comment

# Thank you!



Email: Hongfang Liu (Liu.Hongfang@mayo.edu)  
Huan He (He.Huan@mayo.edu)

MedTator: <https://ohnlp.github.io/MedTator/>

The screenshot displays the MedTator 1.2.0 interface. The top navigation bar includes 'Annotation', 'Statistics', 'Export', and 'Adjudication'. The main toolbar contains various icons for document management, search, and display options. The central text area shows a document snippet with several entities highlighted in colored boxes: 'anxiety', 'nervousness', 'dizzy', 'lightheaded', 'not feeling well', 'faint', and 'ice pack'. Below the text, a table lists the detected tags with their IDs, spans, and text.

Tag	ID	Spans	Text	Attributes
OTHER_AE	O0	86~93	anxiety	comment
OTHER_AE	O1	94~105	nervousness	comment
DIZZINESS	D0	199~204	dizzy	comment