PRACTICAL DATA* VISUALIZATION WITH JAVASCRIPT

WHEN, *WHAT, AND HOW

Ceilyn Boyd

ceilyn_boyd@harvard.edu

Program Management @ the Harvard Library

ORIGINAL PRÉCIS

This talk introduces a methodology intended to help you decide:

- *When* to use JavaScript to visualize your data (and when to choose alternate means)
- *What* data, or aspects of your data to render visually, and
- *How* best to use the tools at your disposal to visualize your data

WHY

Background and Motivation

SURVEYING THE LANDSCAPE

Image source.

Demand for, and popularity of data analysis, data mining, data-driven visualization has led to ...

...THIS

Image source.

SO MANY OPTIONS

- Proliferation of technologies, languages, platforms, applications, and tools with overlapping functionality
  - Excel (PowerPivot, Pivot Charts) - Google Sheets/Charts - Jupyter - MATLAB
    - Octave - OpenRefine - Plot.ly - R - SAS - Stata - Tableau
  - JavaScript - Python - Ruby
  - WebGL - SVG
- Proliferation of graphics and visualization libraries & APIs
  - Bokeh - Chart.js - D3.js - Dygraphs.js - Google Charts API - jqPlot.js - Processing.org, Processing.js - VisPy - Vis.js

SO MUCH DATA

- Library-related data:
  - ARL data - Google Analytics - Harvard Library operational data - Harvard Library Open Metadata - Program/project assessment data & results
- Library-related APIs:
  - Dataverse API - DPLA API - HathiTrust - WorldCat API Library Cloud
- Research data & APIs:
  - Census.gov - Factiva API - New York Times APIs - SEC.gov - Twitter API

MANY QUESTIONS & CONSIDERATIONS

- When does visualization work best for libraries and library data?
- Data
  - How to model, structure, and manipulate data for visualization
  - How to work with data from different sources
  - How to decide when to use which visualization based upon data
- What makes a successful visualization?
- Concerns about knowledge, skills, and access to technology and training

WE HAVE DATA. WE WANT TO ANALYZE & VISUALIZE IT...WHAT NEXT?

Image source.

PRÉCIS REDUX & REWRITTEN

This talk will focus on a methodology intended to help you:

- *Choose* amongst potential tools (including JavaScript) to visualize your data

It will use three Library-related examples to:

- *Demonstrate* the methodology in action, and
- *Demonstrate* how to use JavaScript for visualization workflows

AUDIENCE

People who:

- Want to add interactive visualizations to their Web documents (to create Data-Driven Web Documents)
- Are new or relatively new to programming but who are familiar with HTML, CSS, and some JavaScript
- For whom programming is not a frequent part of their daily work
- Have experience with programming but who do not usually work with data, graphics, or visualization
- Need to decide how best to create sustainable visualizations

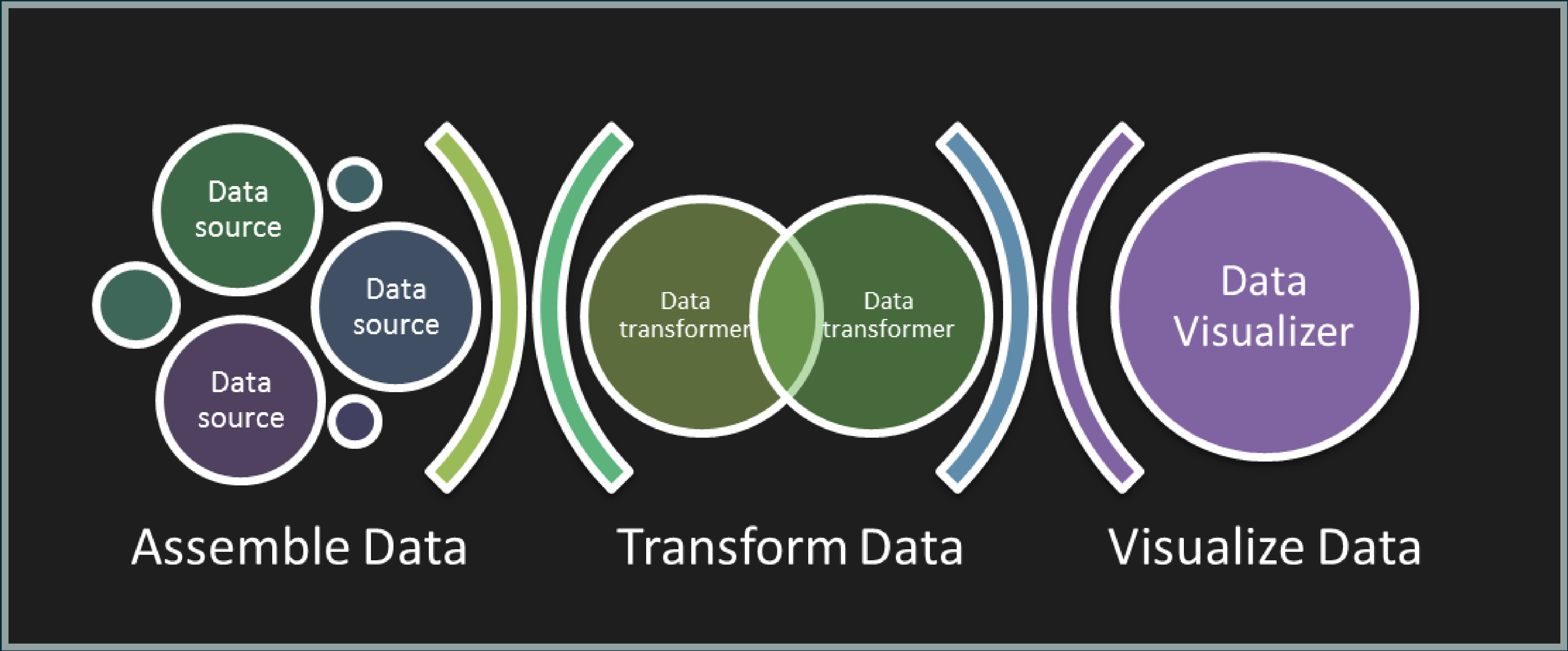
DATA VISUALIZATION WORKFLOW

METHODOLOGY & FRAMEWORK

Image source.

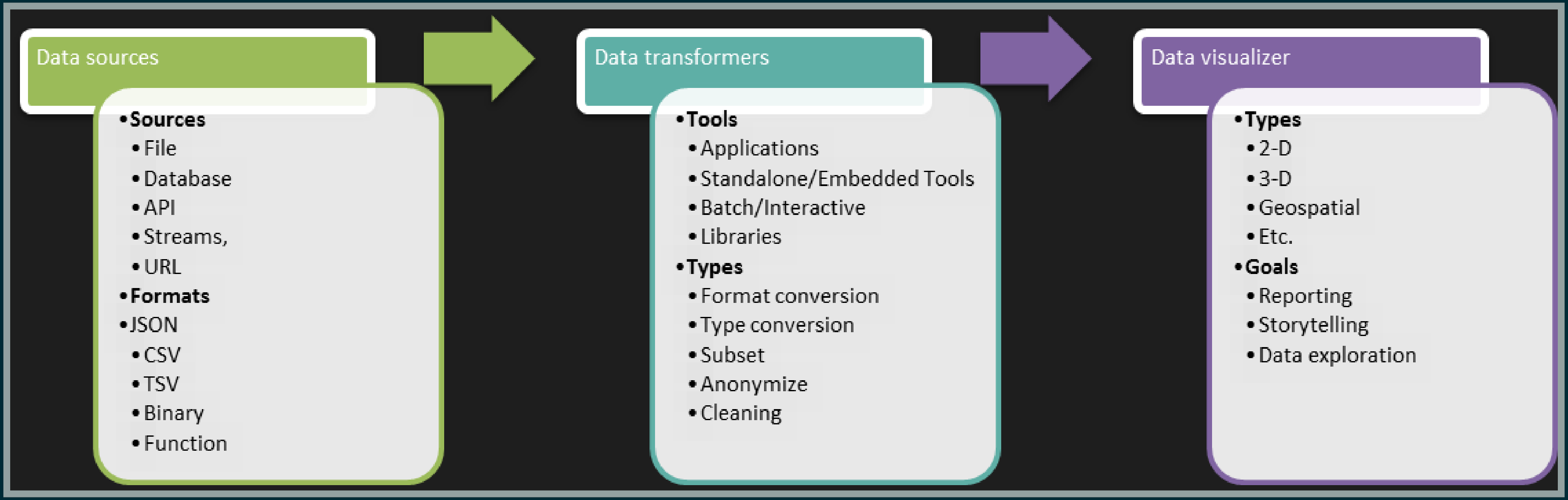
DATA VISUALIZATION WORKFLOW

Goal: Reporting, Storytelling



DATA VISUALIZATION WORKFLOW

COMPONENTS



Data sources

- **Sources**
  - File
  - Database
  - API
  - Streams,
  - URL
- **Formats**
- JSON
  - CSV
  - TSV
  - Binary
  - Function

Data transformers

- **Tools**
  - Applications
  - Standalone/Embedded Tools
  - Batch/Interactive
  - Libraries
- **Types**
  - Format conversion
  - Type conversion
  - Subset
  - Anonymize
  - Cleaning

Data visualizer

- **Types**
  - 2-D
  - 3-D
  - Geospatial
  - Etc.
- **Goals**
  - Reporting
  - Storytelling
  - Data exploration

WORKFLOW COMPONENTS

DATA SOURCE

A component that provides data to the workflow. It could be a repository for any kind of data, such as a database, a file, an API, a program; or it could be a function. Multiple data sources might be used in a visualization workflow.

WORKFLOW COMPONENTS

DATA TRANSFORMER

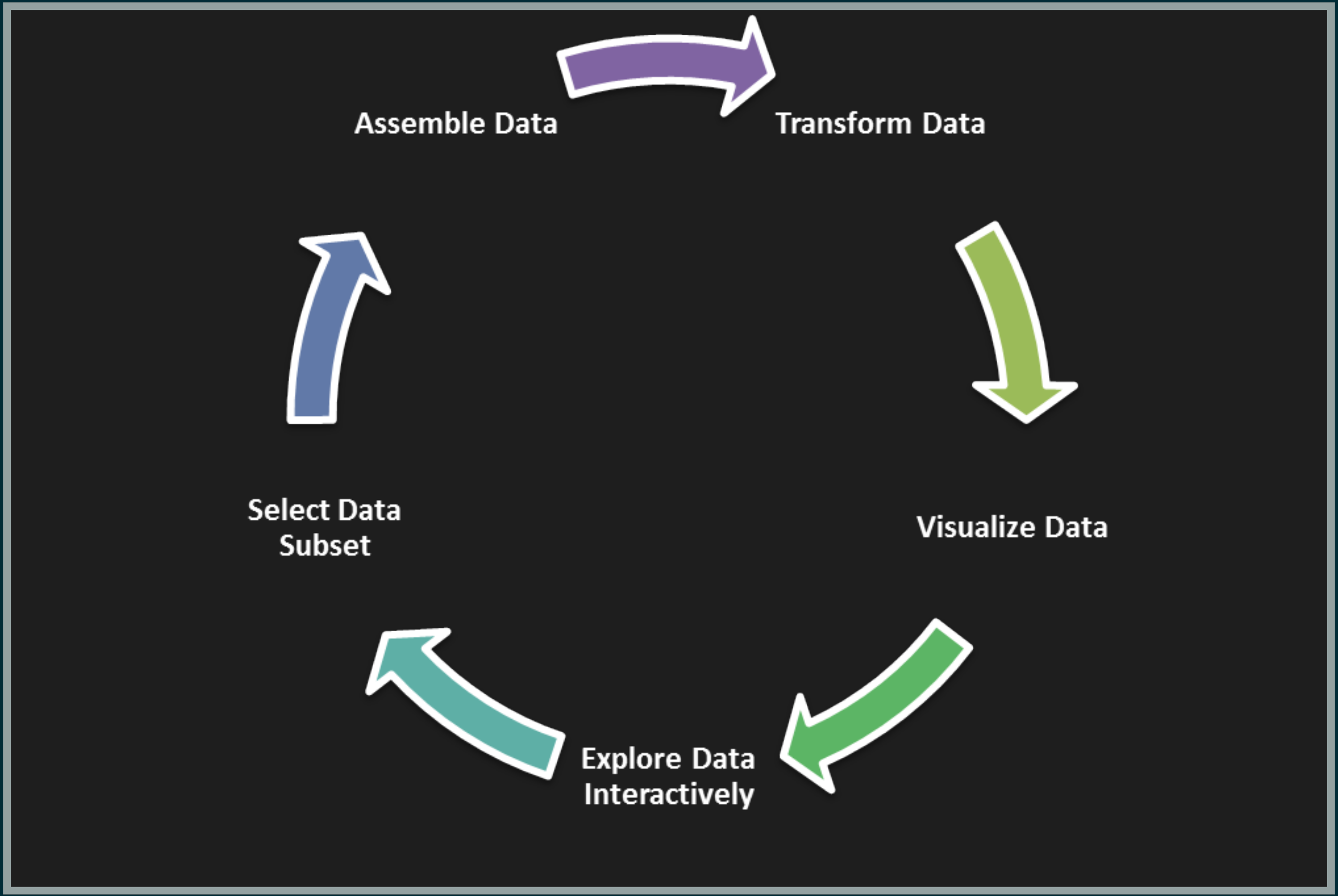
A component that performs some kind of data processing, such as a filter or a file format converter. Multiple data transformers may be needed in a workflow to prepare data for visualizers.

WORKFLOW COMPONENTS

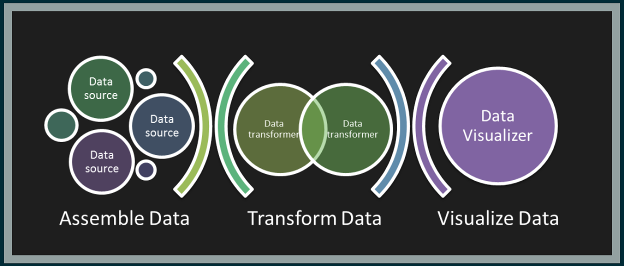
DATA VISUALIZER

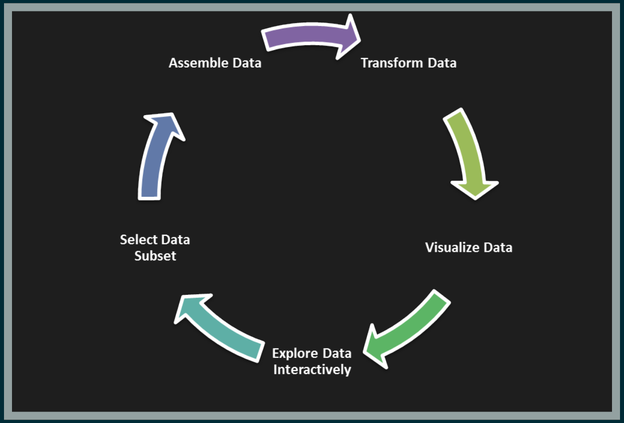
A component that presents a graphical representation of information or data that is suitable for display on a device (e.g. via a Web browser or app), in a document, or in print.

DATA EXPLORATION WORKFLOW

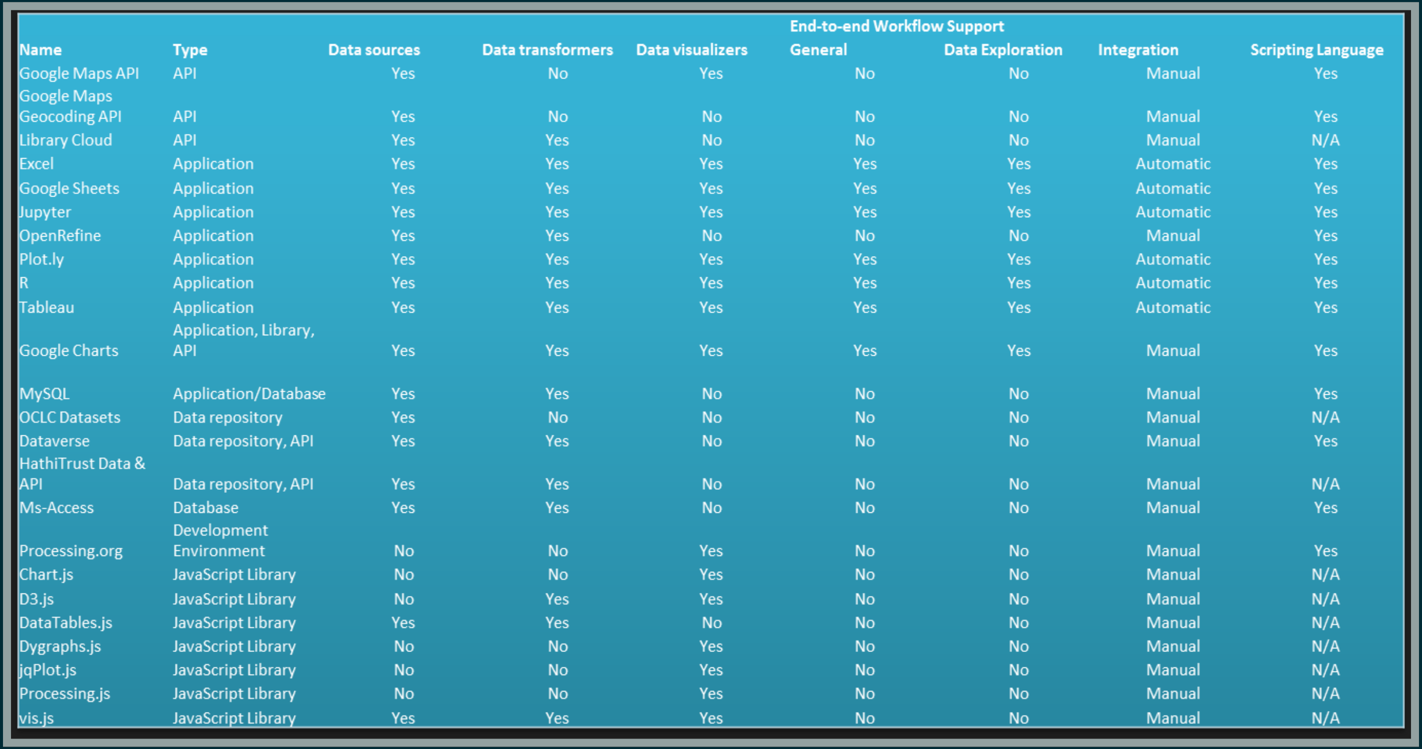


APPLICABLE TO USE-CASES, SYSTEMS, TOOLS, & LIBRARIES





COMPARE DATA ANALYSIS & VISUALIZATION OPTIONS



					End-to-end Workflow Support			
Name	Type	Data sources	Data transformers	Data visualizers	General	Data Exploration	Integration	Scripting Language
Google Maps API	API	Yes	No	Yes	No	No	Manual	Yes
Google Maps Geocoding API	API	Yes	No	No	No	No	Manual	Yes
Library Cloud	API	Yes	Yes	No	No	No	Manual	N/A
Excel	Application	Yes	Yes	Yes	Yes	Yes	Automatic	Yes
Google Sheets	Application	Yes	Yes	Yes	Yes	Yes	Automatic	Yes
Jupyter	Application	Yes	Yes	Yes	Yes	Yes	Automatic	Yes
OpenRefine	Application	Yes	Yes	No	No	No	Manual	Yes
Plot.ly	Application	Yes	Yes	Yes	Yes	Yes	Automatic	Yes
R	Application	Yes	Yes	Yes	Yes	Yes	Automatic	Yes
Tableau	Application	Yes	Yes	Yes	Yes	Yes	Automatic	Yes
Google Charts	Application, Library, API	Yes	Yes	Yes	Yes	Yes	Manual	Yes
MySQL	Application/Database	Yes	Yes	No	No	No	Manual	Yes
OCLC Datasets	Data repository	Yes	No	No	No	No	Manual	N/A
Dataverse	Data repository, API	Yes	Yes	No	No	No	Manual	Yes
HathiTrust Data & API	Data repository, API	Yes	Yes	No	No	No	Manual	N/A
Ms-Access	Database	Yes	Yes	No	No	No	Manual	Yes
Processing.org	Development Environment	No	No	Yes	No	No	Manual	Yes
Chart.js	JavaScript Library	No	No	Yes	No	No	Manual	N/A
D3.js	JavaScript Library	No	Yes	Yes	No	No	Manual	N/A
DataTables.js	JavaScript Library	Yes	Yes	No	No	No	Manual	N/A
Dygraphs.js	JavaScript Library	No	No	Yes	No	No	Manual	N/A
jqPlot.js	JavaScript Library	No	No	Yes	No	No	Manual	N/A
Processing.js	JavaScript Library	No	No	Yes	No	No	Manual	N/A
vis.js	JavaScript Library	Yes	Yes	Yes	No	No	Manual	N/A

(See handout)

WORKFLOW CONSIDERATIONS

WHAT TO CONSIDER WHEN BUILDING A VISUALIZATION

IMPLEMENTATION CONSIDERATIONS ARE...

- Related to an individual workflow component
- Related to implementation of a workflow segment
- Related to the entire workflow

IMPLEMENTATION CONSIDERATION CATEGORIES INCLUDE...

- Legal & Ethical
- Technical
- Usability
- Sustainability
- Aesthetic

COMPONENT CONSIDERATIONS

Component considerations involve:

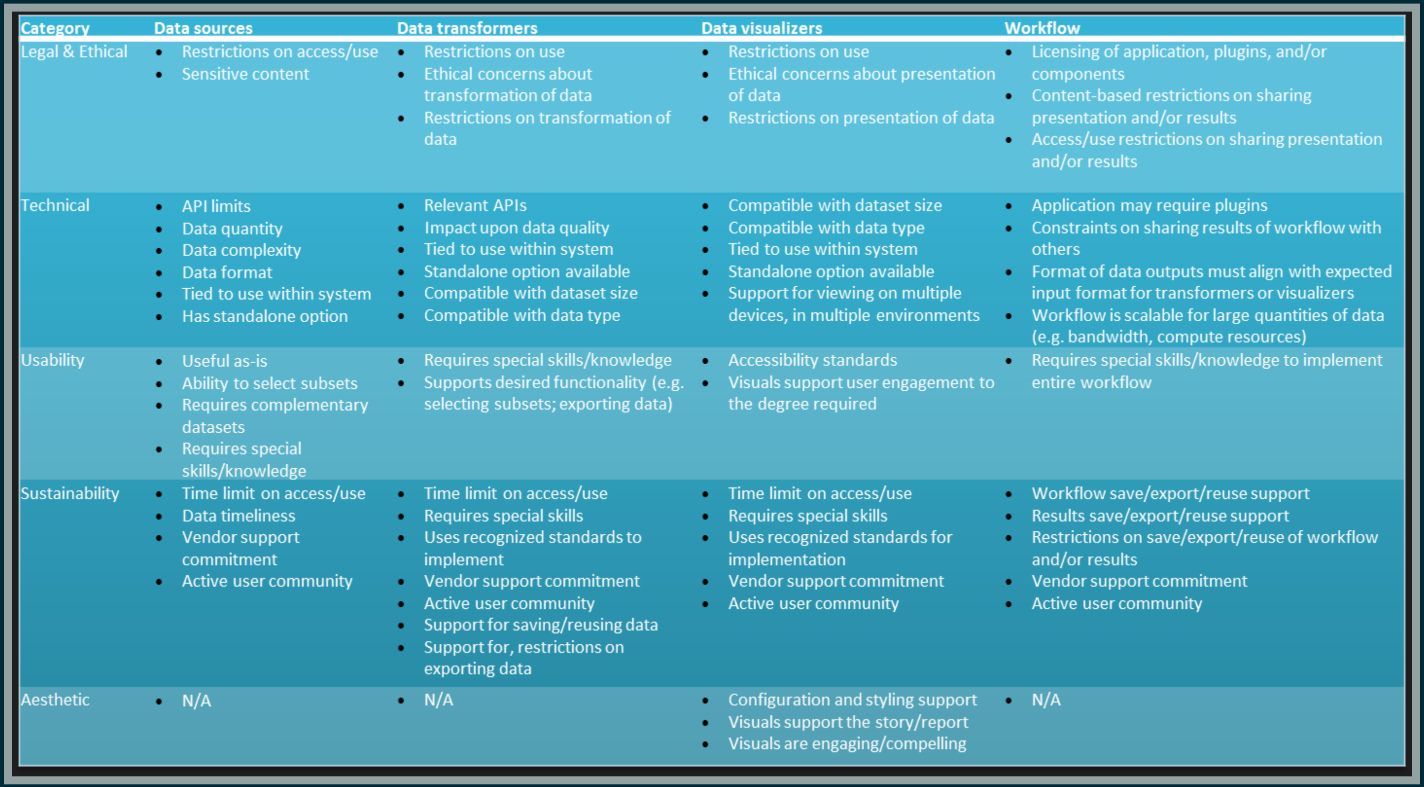
- Data
- Component inputs & outputs
- Representation of the data within the component
- Presentation of the data

SEGMENT & FULL WORKFLOW CONSIDERATIONS

Segment/Workflow considerations involve:

- Resources: e.g. Skills & knowledge
- Licenses & agreemenets affecting use, sharing, presentation of results
- Technical: e.g. compute capacity, bandwidth, security, scalability
- Sustainability, reuse, automation, preservation/reproduction of results

SUMMARY OF COMPONENT & WORKFLOW CONSIDERATIONS

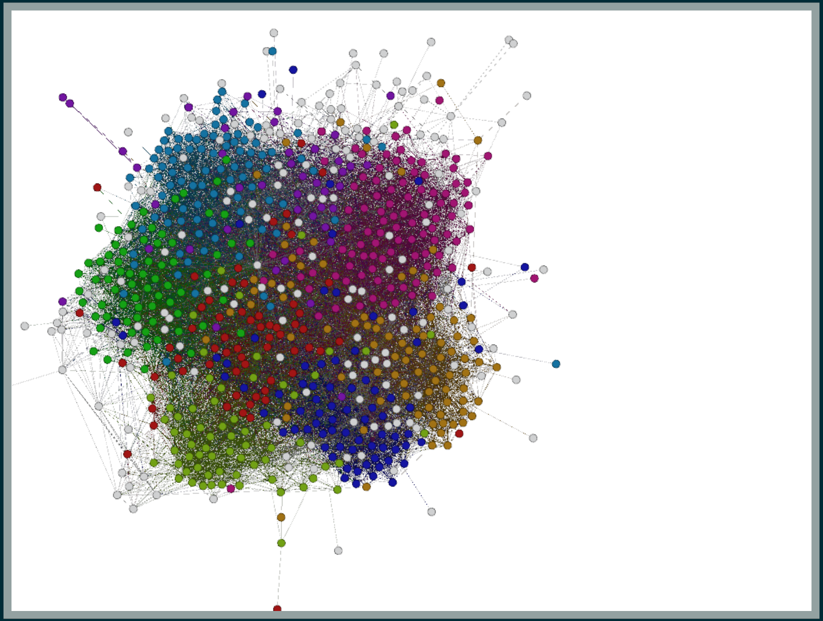


Category	Data sources	Data transformers	Data visualizers	Workflow
Legal & Ethical	• Restrictions on access/use • Sensitive content	• Restrictions on use • Ethical concerns about transformation of data • Restrictions on transformation of data	• Restrictions on use • Ethical concerns about presentation of data • Restrictions on presentation of data	• Licensing of application, plugins, and/or components • Content-based restrictions on sharing presentation and/or results • Access/use restrictions on sharing presentation and/or results
Technical	• API limits • Data quantity • Data complexity • Data format • Tied to use within system • Has standalone option	• Relevant APIs • Impact upon data quality • Tied to use within system • Standalone option available • Compatible with dataset size • Compatible with data type	• Compatible with dataset size • Compatible with data type • Tied to use within system • Standalone option available • Support for viewing on multiple devices, in multiple environments	• Application may require plugins • Constraints on sharing results of workflow with others • Format of data outputs must align with expected input format for transformers or visualizers • Workflow is scalable for large quantities of data (e.g. bandwidth, compute resources)
Usability	• Useful as-is • Ability to select subsets • Requires complementary datasets • Requires special skills/knowledge	• Requires special skills/knowledge • Supports desired functionality (e.g. selecting subsets; exporting data)	• Accessibility standards • Visuals support user engagement to the degree required	• Requires special skills/knowledge to implement entire workflow
Sustainability	• Time limit on access/use • Data timeliness • Vendor support commitment • Active user community	• Time limit on access/use • Requires special skills • Uses recognized standards to implement • Vendor support commitment • Active user community • Support for saving/reusing data • Support for, restrictions on exporting data	• Time limit on access/use • Requires special skills • Uses recognized standards for implementation • Vendor support commitment • Active user community	• Workflow save/export/reuse support • Results save/export/reuse support • Restrictions on save/export/reuse of workflow and/or results • Vendor support commitment • Active user community
Aesthetic	• N/A	• N/A	• Configuration and styling support • Visuals support the story/report • Visuals are engaging/compelling	• N/A

(See handout)

EVALUATING VISUALIZATIONS

TOWARDS A STRATEGY



Network Graph 'Hair Ball', Source.

VISUALIZATION EVALUATION CRITERIA

1. Data overview
   - The capacity of a visualizer to provide a global view of all significant data elements simultaneously
2. Data navigation
   - The ease with which a user can use the visualizer to locate/engage with a specific data element
3. Expressive power
   - The capacity of a visualizer to communicate the full range of relationships within the dataset
4. Aesthetics
   - The visual utility and intuitive value associated with the visualizer

Source: Visualizing Back-of-Book Indexes, Boyd & Wade, 2012.

IMPLEMENTING VISUALIZATION WORKFLOWS WITH JAVASCRIPT

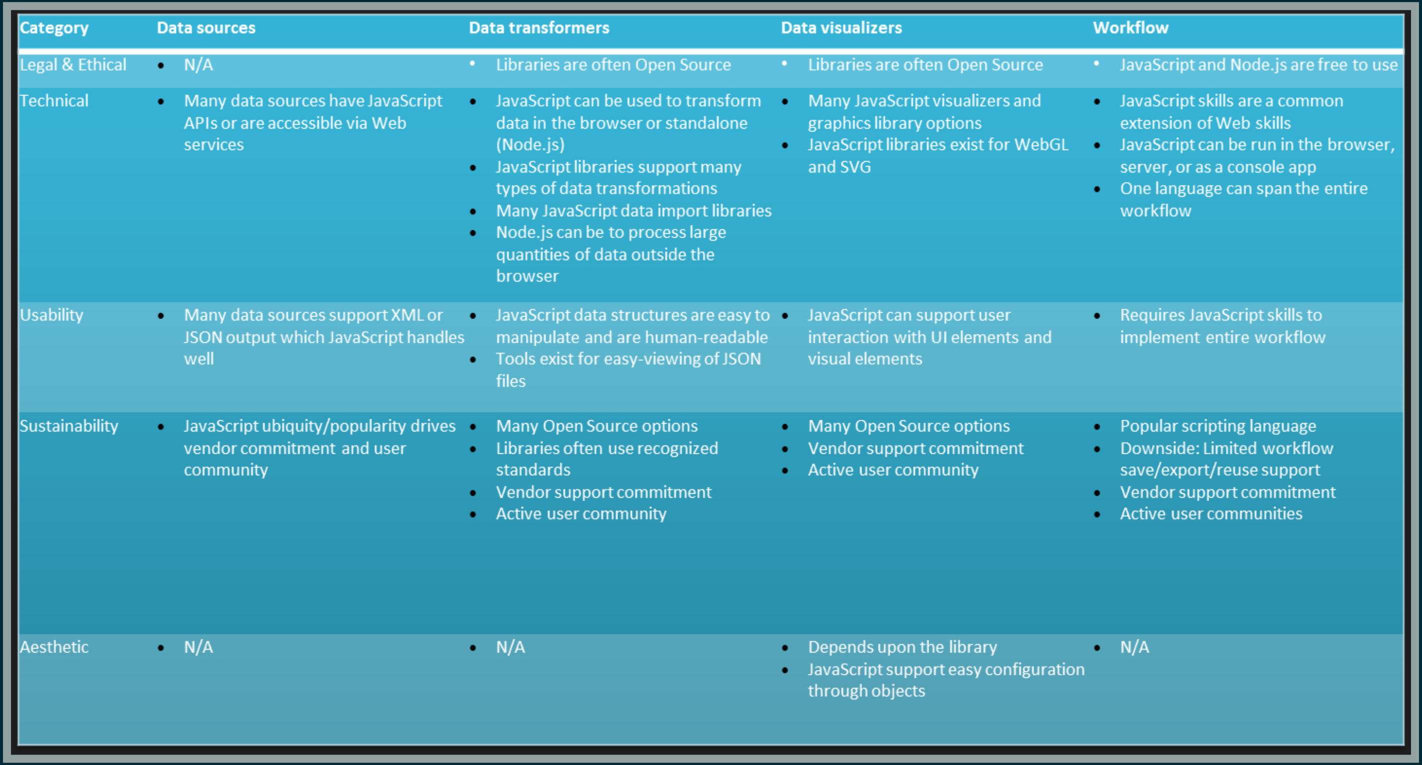
BENEFITS OF JAVASCRIPT

- Established Web technology
  - Ubiquitous browser support
  - Easy integration with UI elements for building applications (e.g. jQuery)
  - Variety of frameworks for building applications
  - Many visualization and data manipulation libraries
- Standalone console & server-side support via Node.js
  - Many data manipulation libraries
  - Many APIs/libraries for connecting to data sources (e.g. MySQL, MS-Access, Library Cloud, Dataverse)

BENEFITS OF JAVASCRIPT (CON'T)

- One language for data acquisition, analysis, visualization, application development, and document delivery
- End-to-end visualization workflow support
- Natural fit for Data-Driven Documents
- Useful for prototyping and production
- No licensing fees

SUMMARY OF JAVASCRIPT CONSIDERATIONS



Category	Data sources	Data transformers	Data visualizers	Workflow
Legal & Ethical	- N/A	- Libraries are often Open Source	- Libraries are often Open Source	- JavaScript and Node.js are free to use
Technical	- Many data sources have JavaScript APIs or are accessible via Web services	- JavaScript can be used to transform data in the browser or standalone (Node.js) - JavaScript libraries support many types of data transformations - Many JavaScript data import libraries - Node.js can be to process large quantities of data outside the browser	- Many JavaScript visualizers and graphics library options - JavaScript libraries exist for WebGL and SVG	- JavaScript skills are a common extension of Web skills - JavaScript can be run in the browser, server, or as a console app - One language can span the entire workflow
Usability	- Many data sources support XML or JSON output which JavaScript handles well	- JavaScript data structures are easy to manipulate and are human-readable - Tools exist for easy-viewing of JSON files	- JavaScript can support user interaction with UI elements and visual elements	- Requires JavaScript skills to implement entire workflow
Sustainability	- JavaScript ubiquity/popularity drives vendor commitment and user community	- Many Open Source options - Libraries often use recognized standards - Vendor support commitment - Active user community	- Many Open Source options - Vendor support commitment - Active user community	- Popular scripting language - Downside: Limited workflow save/export/reuse support - Vendor support commitment - Active user communities
Aesthetic	- N/A	- N/A	- Depends upon the library - JavaScript support easy configuration through objects	- N/A

(See handout)

EXAMPLES

JAVASCRIPT FOR DATA-DRIVEN DOCUMENTS

THREE HARVARD LIBRARY-RELATED EXAMPLES

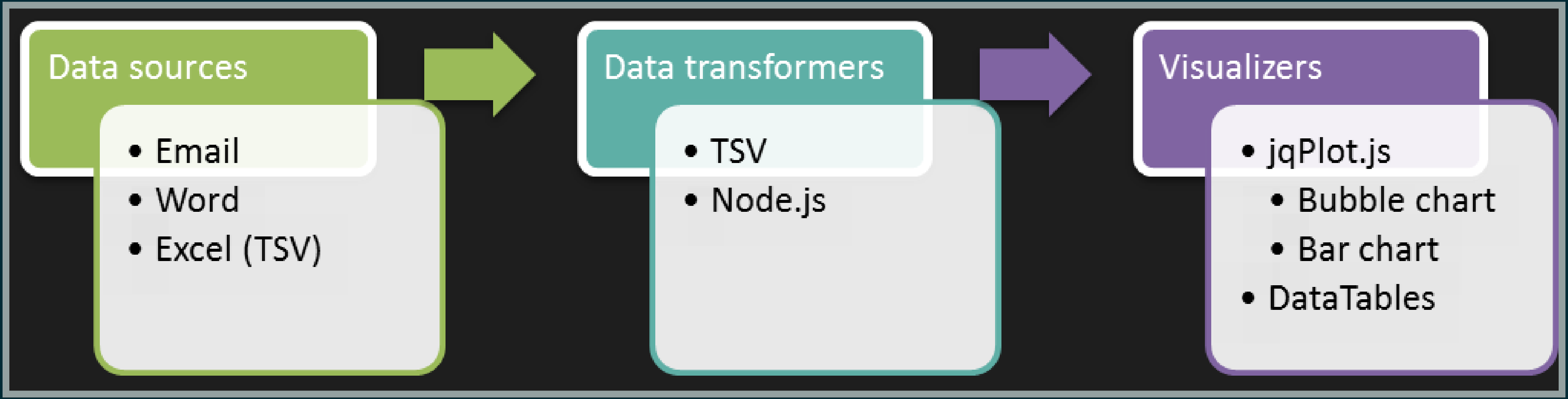
1. Selected Datasets for the Towards a Collections & Content Development Strategic Plan Project (TCCDS) (HTML & CSS, with JavaScript & Node.js)
2. Selected Datasets for the Colonial North American Project (HTML & CSS, with JavaScript & Node.js)
3. OASIS Timeline Viewer Application (Data exploration: Browser only)

TOWARDS A COLLECTIONS & CONTENT DEVELOPMENT STRATEGIC PLAN DATASETS

DEMO

TCCDS WORKFLOW & CONSIDERATIONS

Workflow



Data sources

- Email
- Word
- Excel (TSV)

Data transformers

- TSV
- Node.js

Visualizers

- jqPlot.js
  - Bubble chart
  - Bar chart
- DataTables

Considerations

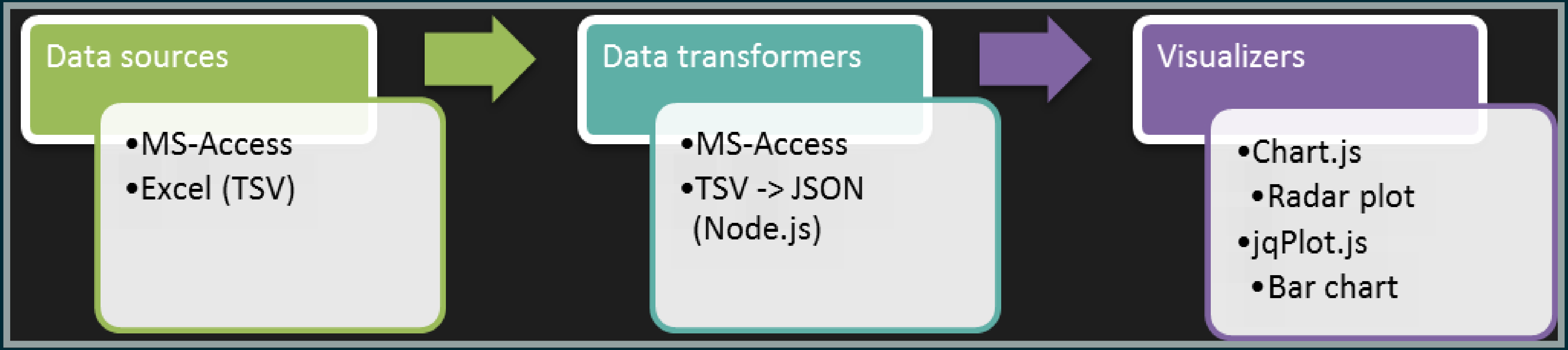
Data sources	Data transformers	Visualizers	Workflow
• Interviews contain sensitive data • Tabular data is easy to work with	• Data is primarily textual, not numeric • Requires custom transformers	• Many options • Size of datasets has impact on visual design	• Manual component integration • Management of multiple visualizers

COLONIAL NORTH AMERICAN DATASETS

DEMO

CNA WORKFLOW & CONSIDERATIONS

Workflow



Data sources
- MS-Access
- Excel (TSV)

Data transformers
- MS-Access
- TSV -> JSON (Node.js)

Visualizers
- Chart.js
  - Radar plot
- jqPlot.js
  - Bar chart

Considerations

Data sources

- Multiple data sources
- Medium-sized dataset (5,000+ data points)
- Tabular data is easy to work with

Data transformers

- Mixed text, numerical, and categorical data
- Data is primarily textual, not numeric
- Requires custom transformers
- JavaScript + MS-Access

Visualizers

- Many options
- Size of datasets has impact on visual design

Workflow

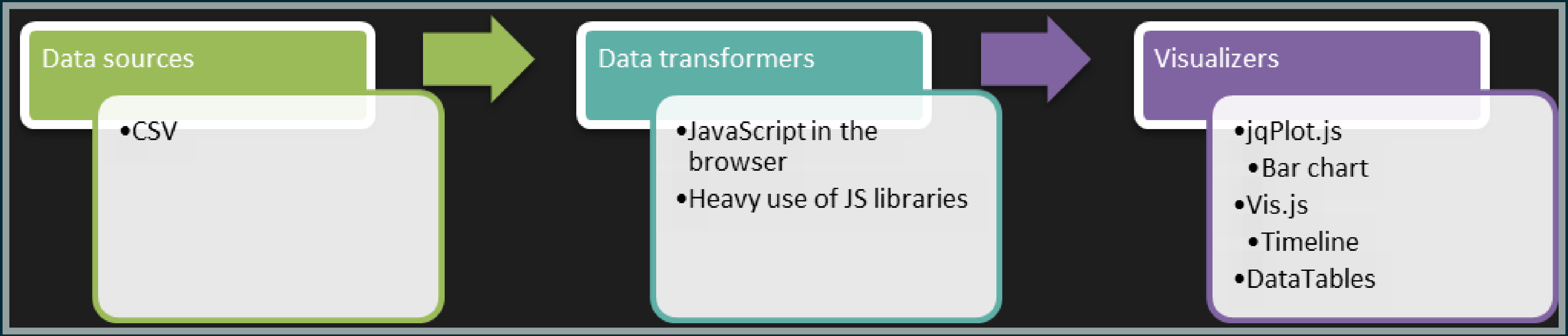
- Manual component integration
- Management of multiple visualizers
- Project is ongoing so datasets will be updated frequently

OASIS TIMELINE VIEWER

DEMO

OASIS VIEWER WORKFLOW & CONSIDERATIONS

Workflow



Data sources

- CSV

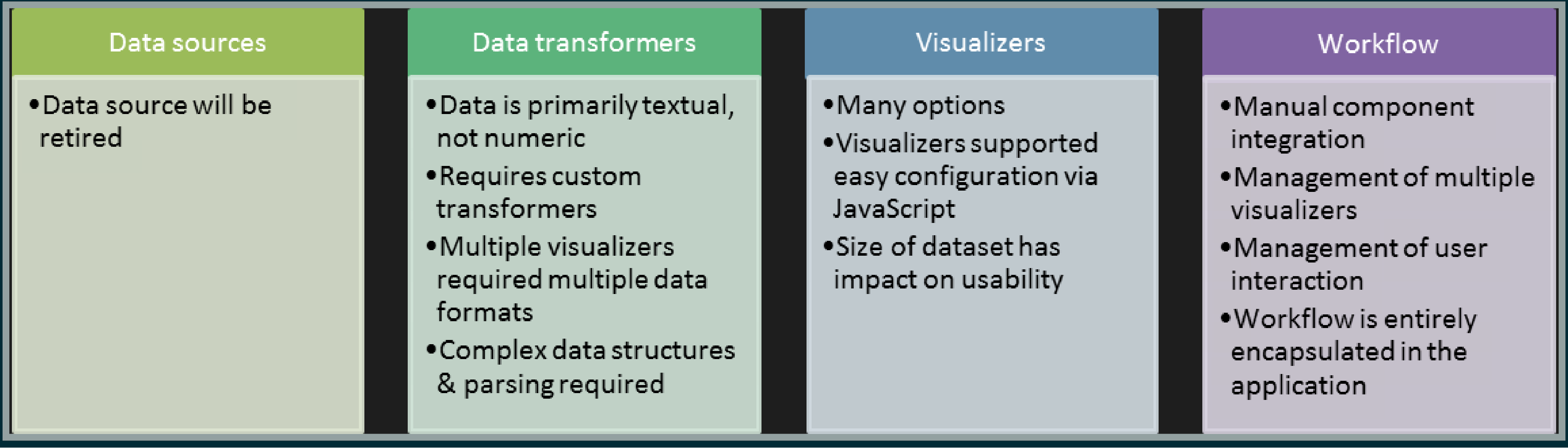
Data transformers

- JavaScript in the browser
- Heavy use of JS libraries

Visualizers

- jqPlot.js
  - Bar chart
- Vis.js
  - Timeline
- DataTables

Considerations



Data sources

- Data source will be retired

Data transformers

- Data is primarily textual, not numeric
- Requires custom transformers
- Multiple visualizers required multiple data formats
- Complex data structures & parsing required

Visualizers

- Many options
- Visualizers supported easy configuration via JavaScript
- Size of dataset has impact on usability

Workflow

- Manual component integration
- Management of multiple visualizers
- Management of user interaction
- Workflow is entirely encapsulated in the application

DATA-DRIVEN DOCUMENT TEMPLATE

```
/**
 * Create the drivers bubble chart of number of mentions
 *
 * @param {String} String containing JSON/TSV data
 * @return {Void}
 */
function createVisualizationFunction(dataString) {
        // parse the string and create JSON objects
        var data = JSON.parse(dataString);

        //
        // process the data to get it into the proper
        // format for the target visualizer
        //
        var processedData = processData(data);

        // create the visualizer, initialized with processedData, adding
        var visualizer = createVisualizerUsingLibraryFunction('visualizat
```

QUESTIONS?

THE END

Thanks to Bobbi, Reinhart, and Gloria for the invitation, and...

Thanks to you all for participating!

RESOURCES

WEB AND JAVASCRIPT RESOURCES

- General
  - Mozilla Web technology for developers. Includes online references for HTML, CSS, JavaScript (client-side), Web APIs, Graphics (SVG, WebGL), MathML
- JavaScript
  - Mozilla JavaScript Language Reference
  - JavaScript Design Patterns
  - Secrets of the JavaScript Ninja. John Resig, Bear Bibeault
  - JavaScript: the Definitive Guide. David Flanagan
  - jQuery in Action. Bibeault, Katz

VISUALIZATION RESOURCES

- Visualize This: the FlowingData Guide to Design, Visualization, and Statistics. Nathan Yau
- Duke Library’s Introduction to Visualization LibGuide
- Sunlight Foundation’s Visualization Style Guide
- Data Visualization Catalogue

NODE.JS RESOURCES

- About Node.js
  - Open-source, cross-platform JavaScript runtime environment for developing server-side Web applications
  - Uses Chrome V8 JavaScript engine, implemented in C++
  - Runs on: Windows XP and later, Mac OS X 10.5+, Linux
  - Server-side, Event-driven, Asynchronous, Many modules available
- Node.js home page and docs
- Felix's Node Guide
- Node.js @ Lynda.com
- Node Beginner Book
- Node.js the Right Way. Jim Wilson
- Node.js in Action. Cantelon, Harter, Holowaychuk, Rajlich