The manuscript by Endsley and Billmire describe a technical prototype of the Carbon Data Explorer (CDE), which is both an API and application for sharing and visualizing geophysical datasets on the web. The CDE is intended to be installed at modeling or data centers, where the primary data are converted to plain text and stored in a database (MongoDB). Web applications then access these data with the web-compatible language JavaScript via JSON, where the user's web browser renders and analyzes the data. The manuscript demonstrates a novel way in which primary model data can be brought to the user's web browser for visualization and analysis, as opposed to the current (server-side) paradigm that produces a picture or summary of the data. However, the paper struggles to make the case for why having the data directly in the user's browser is desirable. I believe there are some major issues of scale and performance that are not addressed in the text. The paper also does not put the CDE's ability to "manage, aggregate, visualize, and share datasets" in the context of the THREDDS Data Server, which is a ubiquitous tool for sharing model data online and the backbone of the Earth System Grid Federation used in CMIP5. I believe my major comments below need to be addressed before the paper it suitable for publication. Although the web application in its current form is rough around the edges, it shows promise. I hope the authors can address my comments by conducting benchmark testing to better discuss scalability and performance.

Major Comments

1) Conceptually, I can see how a web application having direct access to data would be powerful and the author's goal of pursuing this is a worthy endeavor. However, as spatial resolution increases, the size of the data can become immense. The text claims the CDE and this approach results in high performance (page 4, line 22), but no performance metrics are given. A plot benchmarking the number of grid cells in a slice vs time (DB access + transfer + render time) would be very instructive. I also cannot understand how a text representation of a number used in the CDE does not massively increase the size of the original data. Typically a floating point number takes 4 bytes of storage, whereas a string representation of the number would be 1 byte per character (using ASCII). For example, 1234.567 would be 4 bytes as float but 8 as string, doubling the file size that must be transferred to the user. It would be interesting to see some metrics of the ratio of the original data size for a slice to the CDE text size (including any JSON markup), which likely includes some compression. Secondly, as spatial resolution increases, the amount of data transferred to the client would dramatically go up. The current CDE shows 6 characters on the maps, so I will use 6 bytes in my examples. A global 1° x 1° grid would result in 380 Kb (360*180*6) of data per slice, which is reasonable. However, as resolution increases to half and a quarter degree (which is not unreasonable to expect from newer models) the data size becomes 1.5 Mb and 6.2 Mb respectively. These larger package sizes would slow down both the transfer time of the data slice and the browser's ability to parse and render the data. I work with a 30 arcsecond grid over the contiguous US, which would be ~130 Mb per map (7025*3105*6), which is not practical. These resolution-scaling and general technical limitations

need to be discussed in the text. Daily time series for a point can also be data intensive for long records.

2) The introduction states the goal of the CDE is to share, visualize and analyze geophysical data and includes references to FTP, WMS mapping and OpenDAP, but does not mention THREDDS, which is primarily how these tasks are currently being addressed in the modeling community. Although THREDDS does not directly have many analysis features, Ferret-THREDDS provides powerful tools. The authors need to be clear what core goals the CDE addresses that these existing tools do not capture. In the context of data sharing, what does CDE do that OpenDAP does not? Is the distinction OpenDAP generally requires desktop applications, whereas the goal here is to provide web browser access? Again, the case needs to be made for why client-side data provides superior capabilities to the current server-side approach.

Minor Comments

I find the web-based data access demonstrated by the CDE very interesting. Since the software is dubbed a prototype, I'll ask the obvious question: could the concept of JSON data slice access be integrated into the THREDDS software stack as opposed to using MongoDB? I would expect the technology would get more traction in the modeling community if it were built into THREDDS directly, which is widely deployed. When datasets can be rather large, I don't practically see modeling centers hosting two copies of their datasets, one for THREDDS (which they will do for CMIPs) and a text version for CDE.

It is not clear from the introduction that the CDE is software that is installed at a modeling data center as opposed to an online service or API. Likewise, since the software is installed locally, should the name Carbon Data Explorer be generalized?

Page 3 line 2-3 : "We present the tool as a prototype system that addresses the challenges of increasing scientific data volumes, the need for online analysis..." See my previous comments on floating point numbers versus strings. The paper does not demonstrate that the data is smaller once loaded into CDE than the original files. At this point in the text, the paper has not demonstrated the "*need* for online analysis" (ie client-side) as opposed to server-side or offline analysis. Can large-scale meaningful analysis (beyond mapping) practically be done in real-time on the client browser with current technologies?

Page 4 lines 3-9 : THREDDS is the glue that binds these things together.

Page 4 line 20 : Says the CDE 'solves' slow online data access and real time analysis. Maybe not solves, but addresses. What are the "rich analytical capabilities" mentioned? Are they JavaScript math libraries? Is the text referring to temporal and spatial averages or more complex forms of analysis?

Page 5 lines 1-4 : This section needs revision. Is CMIP6 the first time “distributed analyses” will be used in a CMIP? I don’t think the Meehl et al., 2014 reference support this, but rather the organization of CMIP6 will be distributed (in a non-computing sense). Secondly, sharing “web-compatible scientific datasets” was accomplished with the ESGF used in CMIP5.

Page 5 lines 11-12 : How does the CDE “lower or eliminate barriers to bringing scientific results online”? Files and metadata still need to be organized prior to loading into CDE. How is the process of making data available via CDE easier than other software such as THREDDS?

Page 11 line 12: “Data can be quickly aggregated in time or space from within the web application”. No metrics for performance are given. The application seems to handle one polygon/ROI, can data “quickly” be aggregated over many polygons such as counties, states or counties?

Projections and grids : Many atmospheric models and ocean models have irregularly spaced grids. Can CDE read / map those grids appropriately?

Web app feedback (all in Safari):
Temporal aggregation works, but is grayed out. (same for difference)

I managed to produce a “Request Entity Too Large” error. See comments on scaling above.

Coordinated View didn’t work for me at all at the time of testing. (works in Firefox)