Request for Proposal for CTPP Application Program Interface and Web-Based Interface

The American Association of State Highway and Transportation Officials invites proposers to respond to this Request for Proposals for Census Transportation Planning Products (CTPP) Web Application Programming Interface (API) and Graphical User Interface Development. The contract will be awarded for a period of 1 (one) year.

Proposals meeting the criteria detailed below will be accepted until **11:59 PM EST November 12, 2021** by email to pweinberger@aashto.org

Proposers may submit questions until October 15, 2021 by email to pweinberger@aashto.org with a copy to colby@manhangroup.com

All questions and answers will be posted on October 22, 2021.

Background on AASHTO and CTPP

CTPP data products are created from American Community Survey (ACS) data collected by the US Census Bureau (CB). These data products are a set of several hundred tables compiled for various geographic levels, such as census tracts, transportation analysis zones (TAZs), transportation analysis districts (TADs), counties, and Census places. The geographical coverage of CTPP products is the entire United States and Puerto Rico. CTPP tables are available for 1990, 2000, 2006–2010, and 2012–2016. The next data product release will be based on the 2017–2021 ACS. CTPP tables are specified by the CTPP Oversight Board.

Through an agreement with the Census Bureau and the CTPP Board, the Census Bureau creates the tables in the specified geographies covering the nation. The Census Bureau transfers these special tabulations to the CTPP Program in the form of ASCII text files for each state. Upon this transfer, CTPP data, though Census-derived, are no longer formally considered Census data, and as such are ineligible for inclusion in Census data access tools such as DataFerret nor official Census APIs. Access to the CTPP data is currently provided via three methods:

1. A disk containing all CTPP datasets for the entire United States can be shipped upon request, primarily for data users who have analysis needs extending beyond a single state or region;
2. An FTP site (ftp://data5.ctpp.transportation.org/) provides direct access to download state-specific extracts of CTPP data, which is useful for analysts at State DOTs as well as single-state MPOs; and
3. A web-based application allows analysts to select specific tables and geographies as well as preparing simple queries and summaries or visualizations via a graphical user interface (see http://data5.ctpp.transportation.org/).

The CTPP board and Data Access Software Subcommittee are seeking a vendor to help improve CTPP table access tools by updating the design of underlying data structures, establishing a new web services API for the data, and providing an improved web-based UI. To support the development of this RFP and gather background information for prospective bidders, AASHTO

consultants have engaged the CTPP user community and sought feedback on current and proposed new data access tools through a variety of efforts, including:

1. A Special Interest Group (SIG) workshop held at AASHTO's GIS-T conference;
2. A current user survey, specifically focused on identifying key user types / customer segments and use cases or workflows, mapping these to specific data access tools and features;
3. A "focus group" discussion with interested respondents from the survey; and
4. Beta Test pool recruitment from the user community.

The findings of these user outreach and customer research efforts can be found in the consultant report attached as Appendix A, along with consultant-led evaluation of possible alternative API and database setup options (using a reference implementation based upon widely accepted industry standards). Prospective bidders are encouraged to review this report prior to submitting a proposal. A previous report, entitled " Assessing the Utility of the 2006-2010 CTPP Five-Year Data", prepared for AASHTO by Cambridge Systematics, Inc. in 2016, as well as additional information on users' experiences with CTPP data can be found at the CTPP Research Repository, located at: https://ctpp.transportation.org/research_repository/

Purpose/Project Overview

The aim of the desired project will be to design, build and host a data access solution for the Census Transportation Planning Products Program (CTPP) tables, reflecting current technology and user needs. This new solution is expected to have robust performance and deliver a high-quality user experience in a cost-efficient manner. The system to be developed should include three components: database backend, API, and front-end data portal (i.e. a graphical web user interface). An open source software development strategy is preferred, though not required; if proposers wish to incorporate some proprietary components into their recommended system architecture, they must justify this in their proposal based upon performance advantages or other compelling motivations.

Though teams including subcontractors who specialize in specific components of the overall system are allowed, a prime contractor should be designated to oversee all vendor team efforts and serve as the primary Point Of Contact (POC) with AASHTO's consultant project managers and staff, as well as volunteers from the Data Access Subcommittee (DAS) of the CTPP board, in addition to taking full responsibility for quality and timeliness of deliverables. Bids from interested vendors are expected to cover all the required components; partial bids addressing only selected tasks within the overall scope of work may be deemed non-responsive and excluded from consideration.

List of Stakeholders

Starting from the first Journey-to-work tabulation in 1990, the CTPP data and related products have always been the result of collaboration between multiple agencies and multiple entities. The following lists the major stakeholders of this project.

1. The CTPP User Community

The CTPP user community is the most important stakeholder of this project. The spectrum

of CTPP users includes, but is not limited to, planners, modelers, cartographers, researchers, developers and government officials. The consultant is expected to seek the feedback of the CTPP user community throughout the life of the contract.

2. The CTPP Oversight Board

The CTPP Oversight Board is responsible for overseeing the CTPP program. More details regarding the board can be found here: https://ctpp.transportation.org/ctpp-oversight-board/.

- *Data Access Software Subcommittee*

  The Data Access Software Subcommittee is a subcommittee under the supervision of the CTPP Oversight Board. The task of the Data Access Software Subcommittee is to oversee the development of software to access the CTPP product based on ACS data, from developing the technical requirements and RFP to selecting the contractor, overseeing the work, and approving the product. The subcommittee is the primary decision maker of this project.

- *Chair of Data Access Software Subcommittee*

  The Data Access Software Subcommittee is led by the chair of the subcommittee: Benjamin Gruswitz, Delaware Valley Regional Planning Commission Socioeconomic and Land Use Analytics manger.

- *Point of Contact (POC)*

  Penelope Weinberger, AASHTO transportation data program manager, is the point of contact for this project.

3. Management Service Provider

Manhan Group, LLC is a consulting firm which, under contract to AASHTO and at the direction of the CTPP Data Access Software Subcommittee, has engaged in CTPP customer research and technical evaluation efforts supporting the development of this RFP. Manhan Group staff will continue to provide technical oversight and project management services for this project under the leadership of the subcommittee.

Feedback and Review Process

As AASHTO's designated consultant in charge of project management for the CTPP API/GUI update, Manhan Group will review and evaluate interim deliverables submitted by the vendor no less frequently than once a month, reporting progress findings to the AASHTO program manager and Data Access Subcommittee Chair. However, the selected vendor will also be expected to have their own, much more frequent internal testing process and is strongly encouraged to submit a QA/QC plan with their proposal detailing the steps and processes to be taken to ensure a high-quality, smoothly functioning end product.

As the most important stakeholders in the CTPP GUI update process, the user community will be

involved to the maximum extent possible, including feedback on preliminary interface design as well as hands-on testing of pre-release product versions. A core group of Beta testing volunteers have been identified by Manhan Group through its initial customer research process. Manhan Group will coordinate between the vendor and the user community to facilitate up to six Beta test cycles throughout the year-long development process, providing pre-release product access to selected users, as well as compiling feedback for AASHTO program manager and DAS chair review, and then synthesizing recommendations to the vendor for product improvement.

User involvement in the CTPP API is equally critical; however, a different approach towards community outreach and testing will be applied. We expect the selected vendor to be able to provide early access to an API endpoint before the GUI is finalized. Manhan Group will coordinate a "hackathon", inviting members of the user community with a more technical orientation as well as university researchers and civic hacking groups to explore the API and use it to develop impactful, meaningful applications or analyses addressing issues or questions of importance to their work. This process will be used to gather insights into potential API use cases as well first-hand feedback on usability of the API, which will be synthesized and passed along to the vendor, with recommended changes to be incorporated into the final product.

Project Scope

Tasks

The contract between AASHTO and the successful offeror will be for activities affiliated with the following Four (4) task areas (e.g.):

Task 1: Design, build, deploy, host and administer the backend database server

The vendor will design, build, host and administer a secure database with high performance to store content including but not limited to the following: 1) the existing CTPP data sets and metadata, 2) geospatial data corresponding to the CTPP summary levels, and 3) equivalency tables which identify variables which are compatible across different CTPP data sets. In addition, the design of the database should account for the upcoming CTPP dataset releases, as well as data usage information collected over time after the API and web portal have been launched.

The vendor is expected to:

- Configure the database server;
- Design and build the database for fast query performance;
- Develop pipelines for transforming and loading data into the database;
- Perform performance tuning based on periodic performance/usage analysis and users' feedback;
- Perform regular database management and administrative tasks, such as user management, system monitoring, regular maintenance, backup/recovery, security checks, software update, etc.;
- Respond to users and API developers' database-related support requests.

Task 1 Deliverable

- Documentation on architecture of the database server, including instructions on installation and configuration

- Documentation on performance improvement strategies
- Technical documentation on the data model, including external model (views, functions, and procedures to the external users), conceptual model (tables, relationships, constraints, triggers, rules, index), internal model and physical model if applies (internal database storage structure, file organization, indexing techniques)
- Extract, transform, load (ETL) script for batch loading of data
- Build, test, deploy, and launch final database server
- Prepare final technical documentation and end user guide.

As an alternative, the selected vendor can propose a cloud-based serverless solution. In this case, the contractor is responsible for the design and setup of the cloud architecture as well as any of the above responsibilities which are still relevant.

Task 2: Design, build, deploy and host Web Services API endpoint
The vendor will design, build and host an API/Web Services server. It is acceptable to customize and deploy existing web mapping server technologies (e.g. Open Source GeoServer, Map Server, QGIS server or ARCGIS server) and/or a REST/GraphQL API server. The API shall allow a user to query and filter the data and retrieve associated metadata based on geography summary level, table ID, field selections, or value expressions. Additionally, the API will also allow users to load feature layers for any CTPP geographical summary level and any CTPP tabulation in a common GIS application.

The selected contractor is expected to:

- Set up and Configure the API/Web Services server;
- Generate OGC compliant service layers for any combination of CTPP geographical summary levels and tabulation; for multiple tabulations or tabulations involving longitudinal comparison across CTPP data sets, enable the generation of OGC compliant service layers on the fly;
- Design and develop the API that has the benefits of both REST API and GraphQL API;
- Implement API authentication and keep the API secure;
- Support the development of the web application;
- Provide usage examples and a web-based API explorer;
- Develop a Python package or R library, as an example scripting interface for the API;
- Host the API and provide periodic improvements based on periodic performance/usage analysis and users' feedback.

Task 2 Deliverable

- Documentation on the framework and architecture, including instructions on installation and configuration
- For a REST API, provide OpenAPI description of the API, including list of resources and corresponding URIs to realize core functions of data retrieval
- For a GraphQL API, provide design of schema, resolver functions and usage samples
- Navigation model to navigate resources
- APIs for data exploration should be self-explanatory through the above documentation

- Documentation on how metadata will be integrated with data in the API

Task 3: Design, develop and host a front-end data portal web application
The vendor will design, develop and host front-end data portal/web application allowing less technical users to explore, view, query and filter data.

The vendor is expected to:
- Develop a web application tailored closely to users' expectations regarding functionality and performance, while reflecting current best practices in user experience (UX) design
- Implement UX design change management process tracking usability testing of functionality requirements. This includes but is not limited to tasks such as:
  - Creating low-fidelity wireframe and high-fidelity wireframes
  - Creating user flow mockup prototypes
  - Drafting usability documentation, with Beta testing group, per change management process plan
- Log data usage information, to support ongoing analysis of CTPP table utilization
- Additionally, implement features and functionalities which must be updated or added to the website, including but not limited to the following:
  - Allow users to manipulate (add, delete, change) columns shown in the data view using regular expressions. For example, allow users to extract ST_FIPS, CO_FIPS from the geoid field to create two derived fields and allow users to download data in the data view as it is
  - Allow download of data in most common geospatial formats, such as: ESRI Shapefile, MapInfo Tab file, CSV, DBF, GML, KML, Interlis, SQLite, GeoPackage,SpatiaLite, ESRI GeoDatabase (MDB format), ESRI GDB database, GeoJson
  - Allow download of thematic layers in PNG, PDF, JPEG
  - Allow the download of data for CTPP "large geographies" across the entire US, and allow the download of any table and of any summary level for any state, including large states such as California or Texas
  - Allow filtering of data by search terms, expressions, and regular expressions
  - Allow aggregation of data from smaller to larger geographies as well as from different variables (such as commuters by rail and trolley modes), including the recalculation of MOEs for the aggregated estimates
  - Allow longitudinal comparisons between data from different CTPP product dataset vintages including CTPP: 1990, 2000, 2006-2010, & 2012 -2016
- Develop security and backup features to safeguard the system from unauthorized access, intrusion, and other cyber vulnerabilities.
- Fix bugs promptly
- Host the web UX/UI application and implement enhancements based on user feedback and technology updates

Task 3 Deliverable
- Document the proposed solutions per change management process plan

- Build, test, deploy, and launch the web application
- Provide documentation on test cases, results, and resolution; automation scripts for testing
- Provide documentation for the implemented UX/UI solution, including but not limited to a user manual with data dictionary , and user interface training examples.

Task 4: Post Implementation Hosting, Maintenance, Update and Training
The vendor will host and maintain the product for 5 years after a finished product has been developed; keep technologies used for the product updated; and implement improvements based on user feedback gathered by AASHTO and its consultants throughout the life of the contract. The vendor will also provide tutorial content and organize end-user training sessions.

Task 4 Deliverable

- Training Tutorial
- Periodic report on user feedback incorporated and any other updates or improvements
- Periodic report on usage of the product

Budget:
The combined budget for initial design, development and deployment should not exceed $410,000.

Proposals including initial budgets that go above this cap without explanation may be disqualified. The vendor is required to provide a cost breakdown for initial implementation and also provide a separate quote for annual hosting, maintenance and update. For the initial implementation, the vendor is required to provide a cost breakdown by tasks and milestones, including job descriptions, person hours, rate; if non-labor cost is needed, provide a breakdown by categories as well. For the annual hosting, maintenance and update, the vendor is required to provide a separate cost breakdown as well as the expected site performance. If there are two proposing vendors with equal or similarly good qualifications and project understanding, a selection decision may be made based upon cost.

Required Minimum Qualifications

The following qualifications are the minimum required qualifications that the successful bidder must have in order for the contract to be awarded:

1. The Proposer must have experience with a similar role in the lifecycle of a comparable software development process.
2. The Proposer must have live web API or web UI applications to present.
3. The Proposer’s team programming staff must demonstrate a very strong computer science background based upon either education or experience
4. The proposer is strongly recommended to demonstrate knowledge and skills related to the tasks that the proposal intends to bid, for example:
    - Knowledge and skills in design, development and administration of geospatial database, API server, or web UI applications with mapping component.
    - Knowledge in high performance, scalability and high availability architecture and solutions
    - Knowledge in security monitoring and protection solutions for database, API and Web

applications
- Skill in data analysis and reporting; including an understanding of public data access requirements
- Skills in creating business intelligence products and strategies

Desired Qualifications
- Application development experience with public agency clients, public agency data sources, and public agency privacy/security concerns.
- Experience building applications based on U.S. Census and/or transportation data sources

If the Proposer represents a team including sub-contractors, it is acceptable for certain minimum or desired qualifications to be met by a subcontractor rather than the prime contractor; however, the Proposer must clearly explain which qualifications are supplied by which firms on the team.

Vendor Technical Proposal Content
The vendor's technical proposal shall include the following information:

1. Introduction- Firm's name and contact information. Description of the firm's interest and commitment to provide (service requested).
2. Personnel- Listing and one paragraph biographies of personnel with a focus on their (requested service) knowledge and experience. Table delineating the roles and responsibilities of personnel.
3. Approach- Description of the vendor's approach to performing each task, including but not limited to the following items:
    - *A project management plan that identifies key project staff and defines a communication protocol as well as a project schedule with milestones for each task.* The management plan is intended to establish mutual understanding between AASHTO staff, consultant project managers and DAS members about the timing and deliverables of the products throughout the project.
    - *A review of the existing solution to identify critical issues, including proposed strategies for implementing improvements.*
    - *Draft architecture and initial design for each task.*
      The project management team, including DAS, prefers open source solutions. The vendor must provide performance benchmarks or conduct their own benchmarks and trade-off analysis to justify their choice of major components in the architecture, such as backend database management system, map server, web server, front-end toolkit or any other hardware, software, hosting platform, or system configuration choices which might have a major impact on performance. Please describe methodology, assumptions, and performance results of benchmarks and trade-off analysis.
    - *The time interval at which user feedback may be incorporated.* The contract is expected to adopt an agile workflow, with vendors/developers working closely with DAS and the project management team to incorporate user feedback through the early stage of the design and development (as well as continuously through the life of the contract). The "early stage" of development, defined as the period before the finished product has been developed and deployed, should not last more than one

year. During this early stage, the contractor is expected to iteratively improve their design and products based on Alpha and Beta test processes as well as feedback from monthly meetings with the Board and the project management team. After a finished product has been developed and deployed, the contractor should continue to incorporate user feedback from the CTPP user groups and the board on quarterly basis. Links to live API/UI applications built by the vendor

4. <u>References-</u> Contact information for two appropriate references familiar with your work. If possible, please include links to live web applications and web service APIs developed by the bidding vendor or consulting team.

CTPP API AND WEB UI DEVELOPMENT

Customer Research and Technical Assessment Report

Contents

Executive Summary

This report documents the findings of Manhan Group's investigation into the state of Census Transportation Planning Package (CTPP) data access practices, undertaken to support AASHTO's overall goal of issuing an RFP for CTPP Application Programming Interface (API) and web user interface (UI) development. More specifically, this report presents the results of our current CTPP user outreach efforts, as well as our technical assessment of both existing and possible near-future CTPP data access systems.

Current User Survey

Manhan group conducted a survey of current and potential CTPP users, designed to gather information on how the CTPP user community utilizes the current UI, what improvements they would like to see developed, and their interest in utilizing a web services API for data access. The survey was launched on Monday May 3rd, 2021, and concluded on May 21st, 2021. In order to reduce the burden on respondents and avoid asking them irrelevant questions, the survey was structured with multiple pathways depending on how respondents identified their level and mode of access. For example, while someone who does not directly access the data themselves would experience a relatively short survey, someone who accesses CTPP information using either FTP download or the web-based tools would be asked a longer sequence of questions.

One of the main goals of our survey research was to understand the different types of people and organizations accessing and using CTPP data today, with specific attention to how these customer segments align or match up with specific modes of access to the CTPP data using the options available. The majority of respondents, 61%, represented government agencies, with the remainder including 19% consultants, 10% local authorities, and 6% academia. Regional planning agency respondents represented areas with a fairly even distribution of population sizes. In order to understand the geographic distribution (and representativeness) of the survey sample, we asked respondents to identify the state in which they were located; since this method of geographic identification would not apply to multi-state MPOs, we also asked respondents representing regional planning agencies whether their jurisdiction crossed any state boundaries. The responses indicated that 25% of agencies crossed state borders, which has relevance to the technical design of future CTPP data access methods, since the current FTP system offers separate data tables by state for download, which may be less convenient for multi-state MPO users, while a web services API could potentially provide a better solution. A related concern is whether web UI performance could be improved by assuming that users are going to primarily download data for their own region.

Opinions of the Current Web UI

Not a single person who responded to the survey found the web application excellent in terms of ease of use/intuitiveness. In terms of the placement of menu bars/toolbars/menu items as well as the grouping of items in menu bars/toolbars, only 1% find the existing application excellent. Conversely, 27% find features like query/rendering time, navigation flow of the application, and ease of use/intuitiveness poor or bad.

More detailed survey questions about specific user interface features and functions, along with our own testing, allowed us to identify major issues with the existing web application, such as:

- Lack of Automated Programming Interface (API) capability;
- Absence of any web service to render features that can be used in the GIS environments that many rely upon for work, e.g. QGIS, ArcGIS Online or ArcGIS Pro;
- Slow query/rendering performance;
- Specific ease-of-use/intuitiveness issues with software;
- Inconsistency of placement and grouping of menu items with users' expectations;
- Need for improved geographic aggregation, particularly map-based capabilities;
- Need for more output formats;
- Need for additional software improvements over time; and
- Inability to filter out unwanted columns.

Focus Group

The user data gathering process included stakeholder outreach to understand requirements for a data access API and/or a more improved CTPP Web user interface. Towards this end goal, we recruited survey respondents to participate in a follow-up focus group discussion about the current system's pitfalls, limitations and weaknesses as well as strengths, including use cases and common tasks that are currently well-served.

The CTPP API/UI focus group met on May 20, 2021, with 11 participants representing DOT staff, consultants, a mobility technology provider, planning and transit agencies, and an academic researcher. The participants were primarily recruited from the pool of early respondents to our web-based survey. A majority of the selected participants interacted with travel demand models, but not all were model developers, and some of the participants supported other planning activities and issues relating to a diverse set of topics such as environmental justice, housing, and transit ridership.

Regarding users' workflows when the utilizing CTPP web application, the focus group discussion diverged based on applications and technical expertise, reflecting divergent pain points. A consulting travel demand modeler mentioned utilizing the CTPP to understand average travel time for people who go to work every day in the county, as well as pulling home-based-work trip length distributions for travel model calibration. Their typical process included both travel time table and map functions because they could see the county and also the census block in

the study area they were looking at. They noted that they would like to look at these data in more detail (i.e. more stratified cross-tabulations), but found the interface difficult to navigate. Another consulting engineer described themselves as an applier of travel demand models, not a developer coding or calibrating models. Their workflow involved importing CTPP data into GIS and developing spatial analyses to develop origin-destination routings from flows. They mentioned that the current web interface is cumbersome to use, but were wary of utilizing a code-driven interface (e.g. an API), not having sufficient programming experience to do so.

One of the primary goals of the focus group was to explore questions related to the future web services API for CTPP data which could not be answered via the survey. Since this API does not exist yet, "DIY" script-based interfaces to the data previously published to Github by users were examined as possible sources for lessons learned and/or good future practices. The university researcher in the group, who had developed their own script-based interface to CTPP data, described their experience with Census APIs, noting that, prior to the emergence of the TidyCensus package for R as the dominant, preferred method of Census data access for many in the research community, there were a variety of other packages developed by individuals as well as the Census Bureau. The co-evolution of Census APIs and the TidyCensus package is an example worth studying further, as a potential best practice CTPP custodians could emulate.

This university researcher qualified that since APIs are designed for computer programming languages which are usually not easy for people who are more accustomed to graphical user interfaces, a CTPP API would not serve many existing users. Indeed, aside from one other power user, the rest of the group noted that most of them lacked the technical knowledge to exploit such features. Furthermore, the power user noted that when training other departments, a simple web interface would remain critical for simple, practical, queries. It should be noted that the majority of the group stated that they would prefer that both an effective, intuitive UI and a well-documented API be developed, although the UI could be stripped-down in terms of features, relative to the existing interface. All members of the focus group expressed enthusiasm about staying involved with the web UI update effort, including providing input and feedback to the selected vendor regarding new and improved features.

Technical Evaluation: Back-End Reference Implementation

A "reference implementation" is a prototype system set up to aid in testing and development of specifications for a production-ready system to be built later. Manhan Group used this approach to understand the database requirements associated with CTPP API/UI development.

A web API/UI system is typically built with a 3-tier architecture, including: 1) a database backend, 2) an API/Services Server (as the middleware), and 3) the front-end web application. Our reference implementation used open source PostgreSQL/PostGIS as the database backend, along with two open source mapping servers as the OGC Web Services/REST API servers and Postgraphile as the GraphQL API server.

The open source PostgreSQL+PostGIS was chosen as the database engine for our reference implementation because PostgreSQL is fast, reliable and ACID compliant. Also, PostGIS is renowned for its large collection of spatial functionalities as well as fast performance. Our assessment of this database backend focused on three specific database performance improvement techniques: table partitioning, indexing, and materialized views. Since other proprietary database engines such as ORACLE and Microsoft SQL server have similar techniques as well, the knowledge gained through this assessment might still be useful in evaluating potential contractors' proposed solutions for performance improvement even if a different database engine is chosen.

We found that partitioning doesn't always result in better query performance. When the unpartitioned table is small enough or the partition resulted in too many tables, the overhead caused by the need to traverse the partition structure can override the benefits gained by smaller index size. Hence, we recommend partitioning by state only for CTPP tables part 3 and small geographies such as census tracts. We also found that, while indexing has the potential to speed up data access, the design of index techniques needs to account for the way users might query the database. Furthermore, our tests indicated that materialized views is a technique which is worth exploring for the improvement of query performance in case of queries involving time consuming calculations like cross tabulation.

Technical Evaluation: API Design Considerations

The current user survey indicated enthusiastic support among respondents for the development of an API, with 63% saying they would use one if developed. We examined several popular web services API technologies, such as REST (Representational State Transfer) as well as technologies targeted towards geospatial data. These include legacy OGC web service standards as well as the more recent "OGC API Features", also known as WFS3, which is considered RESTful, unlike the decades-old OGC web service standards (e.g. WMS, WFS, WCS, and WPS). Most OGC standards are supported by popular mapping servers such as MapServer, GeoServer, QGIS Mapserver, and ArcGIS Server, which can work with different spatial database engines.

If only a PostGIS database needs to be supported, the lightweight RESTful geospatial feature server pg_featureserv is a good option, with standard support for the OGC API - Features REST API. GeoServer provides REST API endpoints which allow programmers to interact with the server via scripting languages to add/delete layers or conduct other server management tasks. This platform can be easily deployed and has a web user interface, whereas pg_featureserv needs some minimal configuration. Both server software packages were tested in this study.

Despite all the benefits brought by the REST API, each resource endpoint exposed by REST API has predefined fixed structure, which leads to over-fetching and under-fetching problems. For example, a REST API wouldn't allow a user to pick a subset of fields from among the collection of fields shown in the above figure. GraphQL is a new API technology designed to solve this

problem. There are many off-the-shelf GraphQL server software packages readily available. In this study, Postgraphile, a node.js module, was used to instantly spin up a GraphQL server by pointing it towards our testing PostgreSQL database.

Our successful evaluation of GeoServer and pg_featureserv as web API/web service server in this study indicates that building on top of an existing solution for geospatial data might be a more cost-effective option than building a custom REST API from scratch. In addition, the quick evaluation of Postgraphile server software indicates that GraphQL might be a useful component to include in this web API project for users interested in getting a more specific subset of data or fields than a Rest API usually offers.

Conclusions

CTPP data is used in a wide variety of studies, and CTPP users are a diverse group of analysts, planners, engineers, and managers in many different kinds of organizations around the country. While most of these users currently rely upon the web-based user interface provided by AASHTO for data access, some users complain about the web interface's usability and many say that they will use an API if developed in the future--even though their current workflow does not resemble that of a typical web services API user. Even technical "power users" acknowledge that an API, by itself, cannot satisfy the needs of all (nor even most) current users. Furthermore, an intuitive, user-friendly, and effective web user interface is recognized as key to expanding CTPP data utilization outside of its traditional market.

1 Introduction

1.1 Background

This report documents the findings of Manhan Group's investigation into the state of Census Transportation Planning Products (CTPP) data access practices, undertaken to support AASHTO's overall goal of issuing an RFP for CTPP Application Programming Interface (API) and web user interface (UI) development. More specifically, this report presents the results of our current CTPP user outreach efforts, as well as our technical assessment of both existing and possible near-future CTPP data access systems.

1.2 User Outreach

Manhan Group conducted a survey of current CTPP users, designed to gather information on how the CTPP user community utilizes the current CTPP UI, what improvements they would like to see developed, and their interest in utilizing an API (which does not exist today) for data access. It should be noted that we were aware that the 2016 Utility Report also included a survey of current users; however, we also noted that the user interface had changed since then, and thus warranted a more up-to-date analysis. In addition, new questions were asked regarding the future option of a web services API for CTPP data.

Manhan Group also facilitated an open Special Interest Group (SIG) discussion at the 2021 GIS-T conference regarding CTPP data access as well as spatial data and web mapping applications for transportation generally. This meeting occurred prior to release of the survey, and aided in design of the survey questionnaire, as well as planning for later outreach events. However, it did not reach a significant number of current CTPP users.

Participants in the SIG meeting as well as respondents to the survey were given an opportunity to join a smaller focus group discussion hosted by Manhan Group via web meeting. The focus group provided an opportunity to delve deeper into specific survey questions warranting more discussion, as well as exploring wider-ranging questions concerning the option of a future web services API for accessing CTPP data. A secondary motive behind both the current user survey and focus group was to engage and recruit users for future customer research and Beta testing of new software features.

1.3 Technical Evaluation

Manhan Group also conducted an independent technical assessment of both the existing user interface for web-based CTPP data access, as well as the potential specifications and/or requirements associated with building an API-based system. This included development of a pilot reference implementation, which we tested in order to understand performance tradeoffs related to key database design decisions.

1.4 Report Organization

Subsequent sections of this report cover the following topics:

- **Identification of current and potential future CTPP users** - i.e., using survey responses to understand who accesses the CTPP, for which purposes, and via which interfaces or tools. Because survey respondents were recruited from a pool of existing users, other methods (including discussions with focus group participants) were employed to identify use cases and potential customer segments who do not currently access CTPP data for a variety of reasons.
- **Evaluating current CTPP data access workflows** - i.e., using survey data and focus group discussions to understand which methods of CTPP data access are employed by which customer segments, for which ultimate use cases, in terms of specific steps and processes (whether covered by the existing web user interface or FTP data download combined with outside software tools). We also present findings of our own critical analysis of the existing user interface based upon interactive testing.
- **Potential future data access methods and workflows** - i.e., including testing conducted by Manhan Group using a reference implementation in order to understand database design decisions that could affect the performance of any web services API and/or web user interface built upon that back-end. This section also summarizes discussions held with focus group participants concerning the value of a potential future API for various types of CTPP users and use cases.

For those interested in even more detail regarding our user outreach findings, both a transcript of the focus group discussion as well as a table of survey responses can be made available electronically upon request. Any and all information that might personally identify specific users, such as e-mails or names of specific agencies, will be stripped from this version of the data, as well as geographic information that would undermine respondent confidentiality and anonymity. Such topics are addressed at a summary level in the following section, instead.

2 Identification of Current and Potential Future CTPP Users

As described previously, one of the main goals of our survey research was to understand the different types of people and organizations accessing and using CTPP data today, with specific attention to how these customer segments align or match up with specific modes of access to the CTPP data using the options available today. In the following sub-sections, we provide an overview of the survey design and implementation, and then present high-level information on who responded to the survey. It is not necessarily assumed that the survey sample is entirely representative of the CTPP user community--in fact it is reasonable to assume that respondents would represent more engaged CTPP users--so this information is presented as much to caveat subsequent response summaries as to characterize existing customer segments.

2.1 Survey Content

In order to reduce the burden on respondents and avoid asking them irrelevant questions, the survey was structured with multiple pathways depending on how respondents identified their level and mode of access to CTPP data. For example, while someone who does not directly access the data themselves would experience a relatively short survey, someone who accesses CTPP data using either FTP download or the web-based tools will be asked a longer sequence of questions. This allowed us to get more detailed information from certain users without being too demanding of those who might have less hands-on experience. The survey examined the following general topics:

- Respondent's organization and areas of study responsibility
- Respondent's degree of expertise with CTPP data, including future recruitment representation
- Use of specific CTPP datasets
- Types of analyses performed using CTPP data
- Respondent organization's workflow process
- Respondent individual's workflow process
- Respondent's use of web-based CTPP data access system
- CTPP data usage after download
- User Characterization of the functional components of web based CTPP features:
  - Table Features
  - Map Features
- User perceptions of CTPP software usability
- Respondent's desired features and functionality
  - Map
  - Tables
  - Layers
- User interest in using an API for CTPP data access
- Respondent's interest in participation as Beta testing new software

With increased awareness and understanding of previous efforts such as the Utility Assessment, the goal of user *recruitment* for ongoing engagement was designed into the survey structure. As such, the purpose of the survey is the following:

1. Identify key "user types" and estimate what share of CTPP users fall into each category;
2. Identify the core needs of each "user type" in order to better define what vendors must provide, at minimum, as well as what extra features will actually be appreciated (versus inefficient allocation of resources to underutilized features); and
3. Identify users who are sufficiently motivated to join the subsequent "focus group" and then, potentially, be recruited into a panel of users which will engage directly with the selected vendor for additional customer research and Beta testing of new software.

2.2 Survey Sample

The web-based survey was launched on Monday May 3rd, 2021, ran for three weeks, and concluded on May 21st, 2021. The survey was announced on forums frequented by CTPP users, and emailed to 1,420 account holders using the 2012-2016 vintage dataset with the current software. The web survey was opened 441 times with 166 responses.

2.3 Organization Type and Location

Figure 1 shows that the majority of respondents, 61%, represented government agencies, with the remainder including 19% consultants, 10% local authorities, and 6% academia. The regional planning agency respondents represented areas with a fairly even distribution of population sizes, as shown in Table 1 (note: rows do not add to 100% due to a 3% non-response rate).

FIGURE 1: ORGANIZATION REPRESENTATION

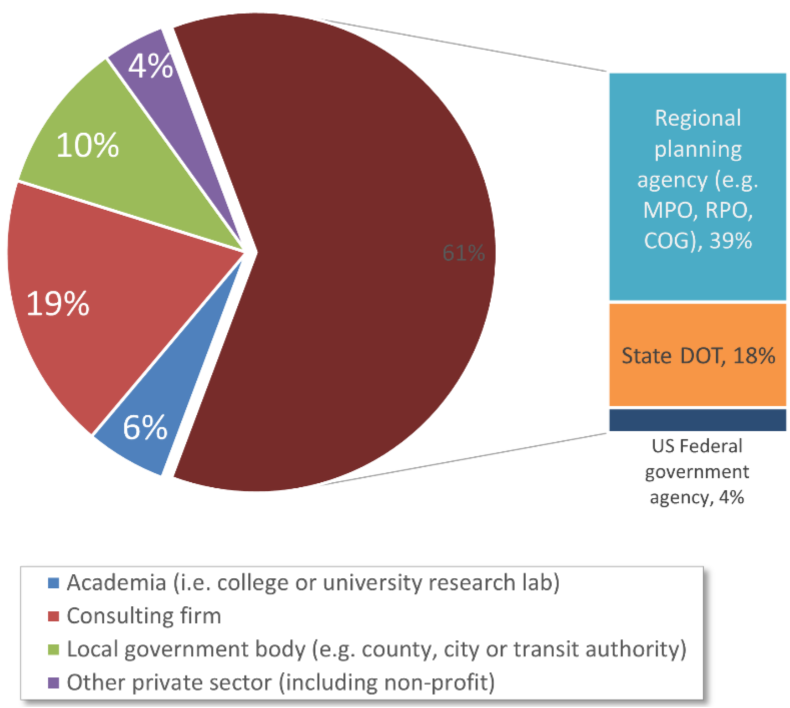


TABLE 1: AGENCY POPULATION REPRESENTATION

Regional planning agency (e.g. MPO, RPO, COG) Population	Representation
0 to 140,000	9%
1,000,001 to 2,500,000	15%
140,001 to 250,000	14%
250,001 to 600,000	23%
600,001 to 1,000,000	11%
More than 2,500,000	25%

In order to understand the geographic distribution (and representativeness) of the survey sample, we asked respondents to identify the state in which they were located. Table 2 summarizes responses to this question, grouped by Census Region and Division.

TABLE 2: PARTICIPATION BY CENSUS REGIONS AND DIVISIONS

REGIONS		DIVISIONS	
WEST	25%	PACIFIC	19%
		MOUNTAIN	6%
MIDWEST	19%	WEST NORTH CENTRAL	4%
		EAST NORTH CENTRAL	15%
SOUTH	39%	WEST SOUTH CENTRAL	14%
		EAST SOUTH CENTRAL	8%
		SOUTH ATLANTIC	17%
NORTHEAST	17%	MIDDLE ATLANTIC	13%
		NEW ENGLAND	4%
TOTAL	100%	SUB TOTAL	100%

Since this method of geographic identification would not apply to multi-state MPOs, we also asked respondents representing regional planning agencies whether their jurisdiction crossed any state boundaries. Figure 2 shows that the majority of responses did not fall into this category, though 25% agencies crossed state borders.

FIGURE 2: REGION CROSSES STATE BOUNDARIES

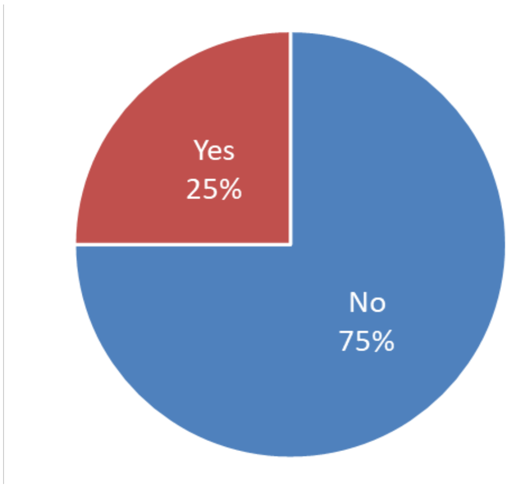
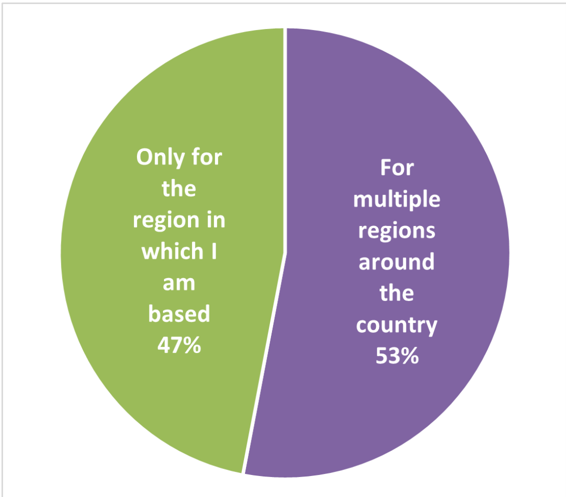


FIGURE 3: CTPP ACCESS FOR MULTIPLE REGIONS



This issue has relevance to the technical design of future CTPP data access methods as well; the current FTP system offers separate data tables by state for download, which may be less convenient for multi-state MPO users, while a web services API could potentially provide a better solution for this specific data access edge case.

A related concern is whether web UI performance could be improved by assuming that users are going to primarily download data for their own region, streamlining the workflow for subsequent selections as well as possibly improving back-end database efficiency. However, this assumption would not apply to users who work with multiple regions across the country, such as consultants with many different MPO clients, academic researchers studying many different metropolitan areas, or even analysts at federal and large state agencies which encompass many different metro regions. Thus, we asked whether the respondents pulled CTPP data for only their own region, or more than one. Figure 3 shows a surprisingly even split in responses to this question, with the thin majority (53%) gathering CTPP data for multiple regions.

2.4 Level of CTPP Expertise

While organization type and location represent one possible way to segment CTPP users for the purposes of understanding factors driving choice of data access mode, user types could also be differentiated by level of expertise and/or rank within their organization--the expectation being that senior staff are unlikely to directly access the data themselves, and novice users are likelier to use a web user interface than FTP download.

As shown in Figure 4, only 12% of survey respondents relied entirely on someone else to access CTPP data. Figure 5 provides more detail regarding who accessed the CTPP data for this subset of users, showing that responses were evenly split between partner agencies, staff, and others (e.g. consultants).

FIGURE 4 WHO ACCESSES CTPP DATA?

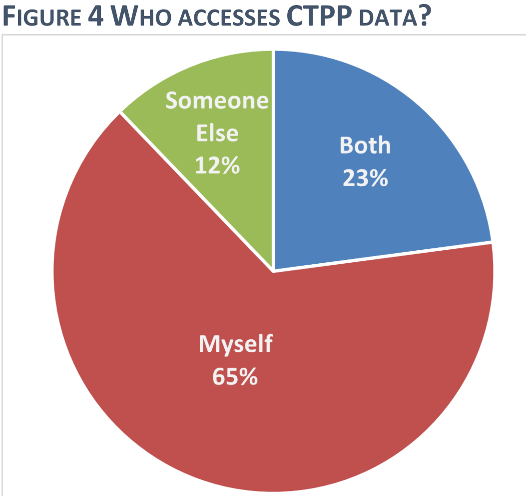
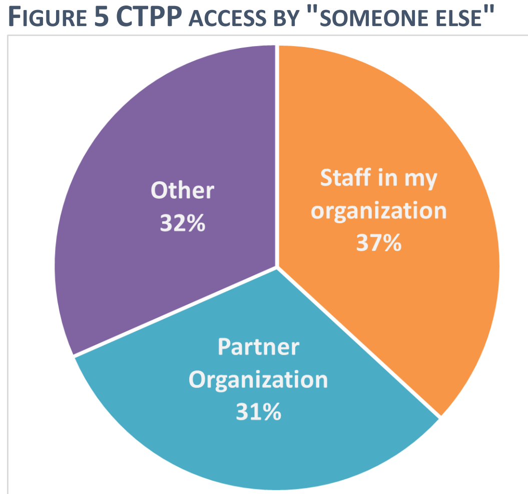


FIGURE 5 CTPP ACCESS BY "SOMEONE ELSE"



About 51% of respondents considered themselves either experts or experienced in using CTPP, as shown in Figure 6. A more objective measure of knowledge and expertise, however, might be completion of training provided by AASHTO (or others). As shown in Figure **7**, 31% of survey participants reported having received training in using the CTPP data web application. Similarly, 33% of participants trained others on utilizing the CTPP data set.

FIGURE 6: LEVEL OF EXPERTISE WITH CTPP DATA

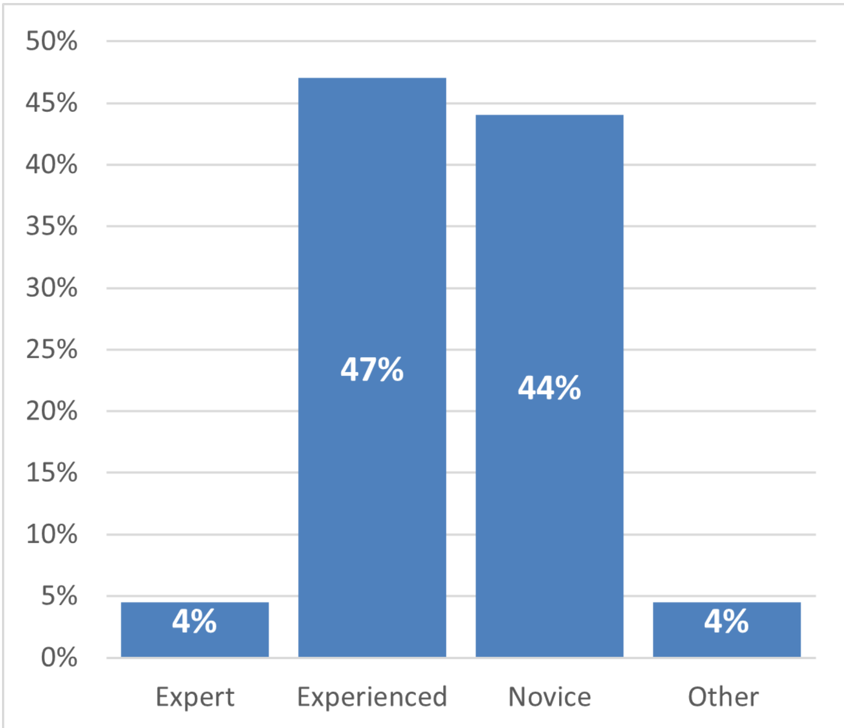


FIGURE 7: RECEIVED TRAINING IN USING CTPP DATA

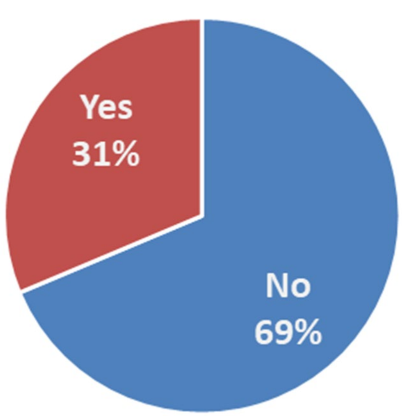
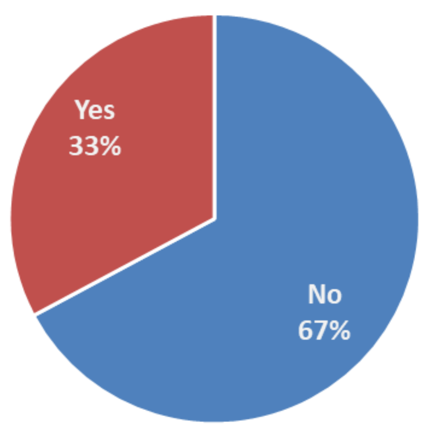


FIGURE 8: TRAINED OTHERS HOW TO USE & ACCESS CTPP DATA



2.5 Purpose of data access

Yet another way to cluster CTPP users might be their intended purpose for accessing the data. This could reasonably be expected to influence their chosen mode of data access; for example, not all analyses require origin-destination commuter flow data, though its availability is one of the core strengths of the CTPP.

Table 3 summarizes the types of analyses performed using CTPP data, as reported by survey respondents (note that percentages do not add to 100 because this was a "select all that apply" type of question; the numbers reported are given as a percentage of the sample size, N=168). The majority of respondents use the CTPP to analyze origin-destination commuter flow data and almost half use it to get information about current travel behavior. The next four most common responses all related to travel demand modeling, whereas responses relating to economic analysis models and other travel forecasting tools (e.g. FTA's STOPS) were selected the least often, of all the options provided.

TABLE 3 TYPES OF ANALYSES PERFORMED USING CTPP DATA

Share	"I use CTPP..." response
57%	To analyze origin-destination commuter flows.
49%	To get information about current travel behavior
36%	To get socio-economic data for the same geography I use in my travel demand model
25%	To calibrate my travel demand model
20%	To help me develop parameters and assumptions for my travel demand model
18%	For some type of travel forecasting analysis other than my travel demand model
13%	As input to an economic analysis model
8%	Other (please explain)

Though a small percentage of overall responses, the reported "other" uses of CTPP data that respondents described may provide a glimpse of insight into potential use cases for CTPP that could be encouraged by making the data more accessible. Respondents falling into this reported using the CTPP:

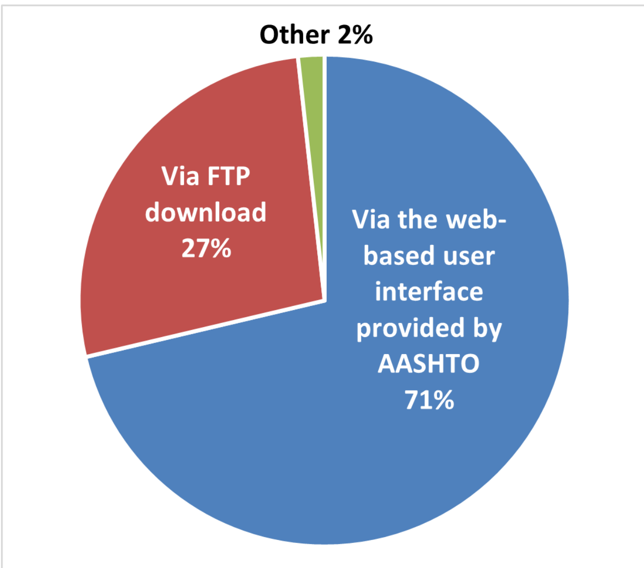
- to characterize equity and diversity;
- for labor shed analysis, worker characteristics and trends;
- to understand the potential spread of COVID;
- for Transit Market Analysis; and
- to understand how access to jobs (as forecast by us) aligns with actual behavior.

Focus group discussions (discussed at greater length in a subsequent section) indicated that preservation and improvement of a user-friendly web UI for data access was especially vital to expanding the utilization of CTPP for some of the "other" (perhaps "emerging") use cases mentioned above.

2.6 Mode and methods of data access

For the purposes of this study, possibly the most important dimension by which CTPP users could be differentiated is the way they access the data. A majority of the respondents, or 71%, access CTPP data via the web access interface, as shown in Figure 9, while 27% access data via the FTP download. Our preliminary findings support the assumption that novice users and emerging use cases are especially dependent upon the web user interface, whereas many of those who select the download option could be characterized as expert or "pro" users of CTPP data (though not necessarily experts in the CTPP web UI).

FIGURE 9: CTPP DATA ACCESS INTERFACE



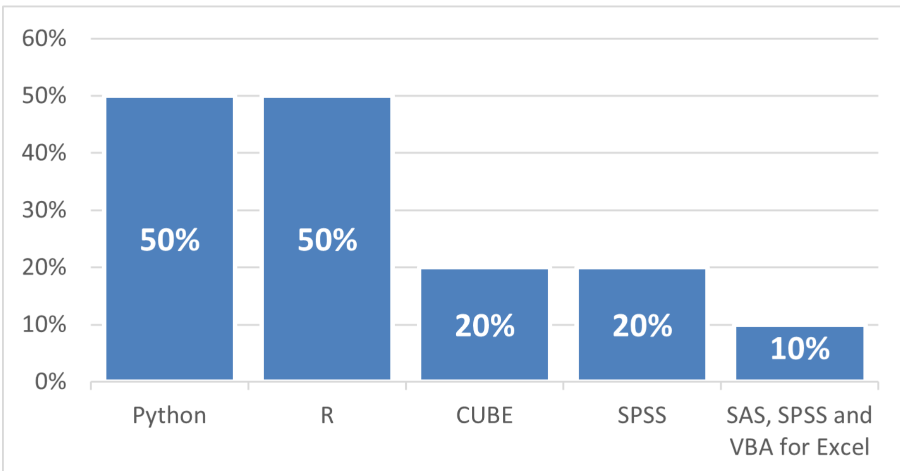
For those who simply downloaded CTPP data via FTP external tools are required in order to do any summarization, visualization, or analysis, and this would also be the case for future API users. Therefore we asked respondents who indicated that they used FTP download to identify their analysis tool of choice, in order to better understand what kinds of third-party integration should potentially be supported by the API to be developed. Their responses are summarized in Table 4. A majority used spreadsheet or GIS software to analyze the data.

TABLE 4 TOOLS USED BY THOSE WHO CHOOSE FTP DOWNLOAD (N=31, "SELECT ALL THAT APPLY")

Share	Response to "How do you typically work with the CTPP data after downloading it?"
58%	Spreadsheet software (e.g. Microsoft Office or LibreOffice)
55%	GIS or travel demand modeling package (e.g. ArcGIS or CUBE)
35%	Statistical data analysis software (e.g. SAS, SPSS or RStudio)
29%	Code scripts to produce summaries and analyses visualizations using a scripting language
26%	Database (e.g. MS SQL or PostgreSQL) for queries and summaries.
10%	Other

The users whose current workflow involves downloading CTPP data via FTP and then coding analysis and summary scripts are closest to the archetypal API use case (where a user would code a script that calls the API in order to download a subset of the data, and then work with these data to perform some additional processing). Given the desire to build an API that meshes well with tools that current users already use, we asked the ten survey respondents who fell into this category to list their preferred scripting languages they preferred. The most common responses were Python and R, each of which were listed by 5 respondents (i.e. 50%). Non-open-source scripting languages such as CUBE, SPSS, SAS, and VBA were listed, but with lower frequency than the two open-source languages cited above (which are also widely known as the dominant tools of choice for most data science work across a variety of domains).

FIGURE 10: SCRIPTING OR PROGRAMMING LANGUAGES UTILIZED IN CTPP DATA ANALYSIS (N=10)



The survey requested additional details about the workflow of web-based CTPP interface users as well. Specifically, we asked the 79 respondents who indicated that they used the GUI about their preferred navigation path, giving three options: a) pick a table from CTPP tables list, then view data, then select geography; b) pick a table from CTPP tables list, then select geography, then view data; or c) select geography, then pick a table from CTPP tables list, then view data. Option c) was most popular, with a 59.5% share of responses indicating a "geography first" preference, followed by the table → geography → data option.

However, when asked how they viewed the data, more respondents said that they simply viewed the tables, rather than using the mapping or charting features, as shown in Table 6. Thus, some GIS-like features, such as geographic selection, seem more important than others, like mapping.

TABLE 5 WHICH PARTS OF THE EXISTING WEB-BASED CTPP DATA ACCESS SYSTEM DO YOU USE?

Share (N=79)	Response (select all that apply)
74.7%	Just Tables
55.7%	Map View
36.7%	Chart View

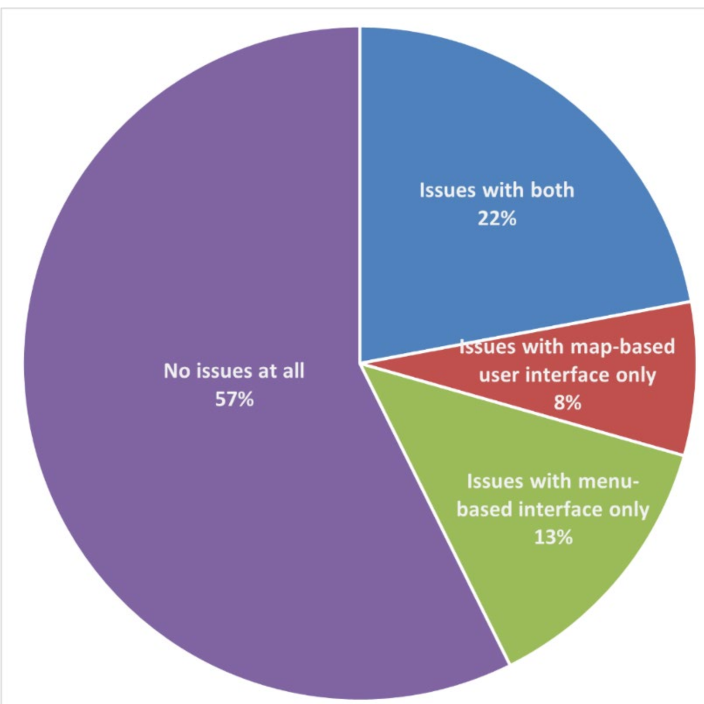
3 The Existing CTPP Web UI Application

In 2016[1], Cambridge Systematics conducted a user survey on attitudes toward the existing access software. According to this survey, less than 40% strongly or somewhat agree that the existing software is easy to use. In order to better understand the factors which may be affecting ease of use, Manhan Group performed an in-depth evaluation of the existing application through interactive testing with sample user cases. We used the information gathered through that evaluation to further refine our user survey questionnaires and collect users' opinions and rankings on specific aspects of the application. The results of these investigations are presented below.

3.1 User's Observations on overall CTPP Data Interface

The survey asked users if the fact that the current web application has two interfaces (for working with map / geography data and table data, respectively) created issues with the user's workflow, shown in Figure 11, below. The majority of respondents stated there were no issues; however 43% of users did encounter problems.

FIGURE 11: ISSUES ENCOUNTERED UTILIZING TWO DIFFERENT UI - MAP & TABLE



[1]Assessing the Utility of the 2006-2010 CTPP Five-Year Data (Information Gathering, User Survey and Peer Exchange) [Report][Appendix]

Figure 12, below, summarizes the survey response to the question "How satisfied you are with various aspects of the existing CTPP web application". Strikingly, it shows that not a single person who responded to the survey finds the web application excellent in terms of ease of use/intuitiveness. In terms of the placement of menu bars/toolbars/menu items as well as the grouping of items in menu bars/toolbars, only 1% find the existing application excellent. 27% find features like query/rendering time, navigation flow of the application, and ease of use/intuitiveness poor or bad. Additionally, Table 6 provides additional observations and comments from survey participants' experience with the current web application.

FIGURE 12: USER'S SATISFACTION WITH CURRENT CTPP WEB APPLICATION

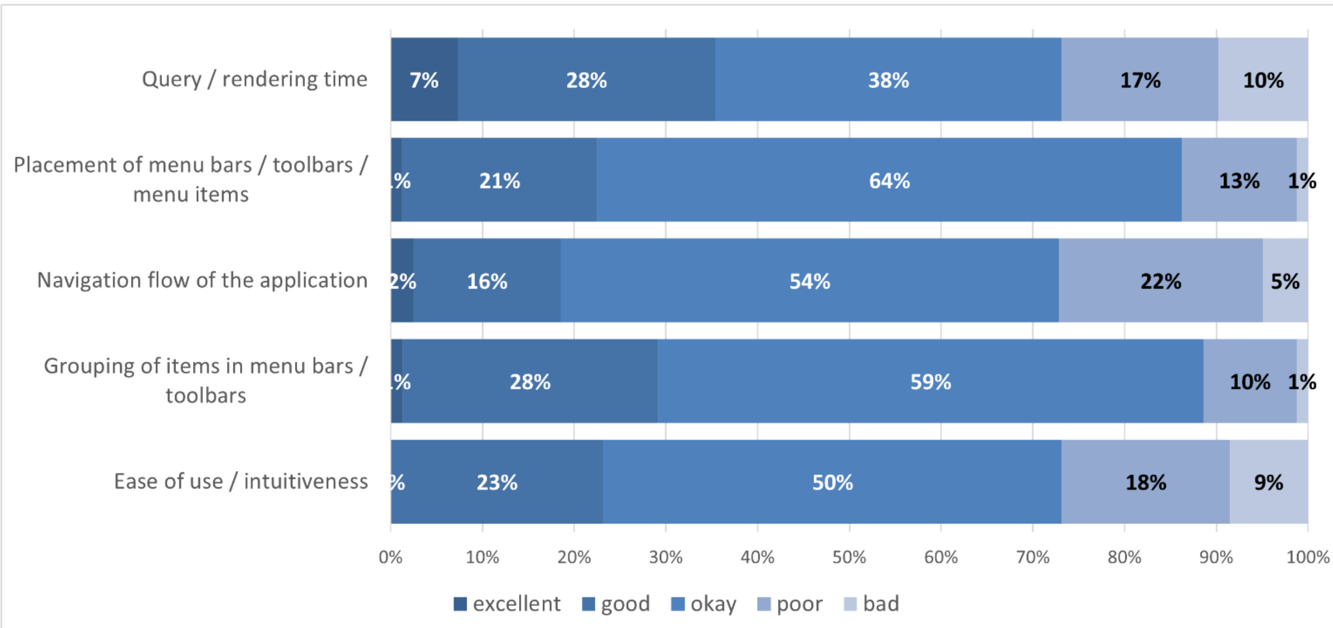


TABLE 6: USER'S OPINIONS ON CTPP DATA INTERFACE

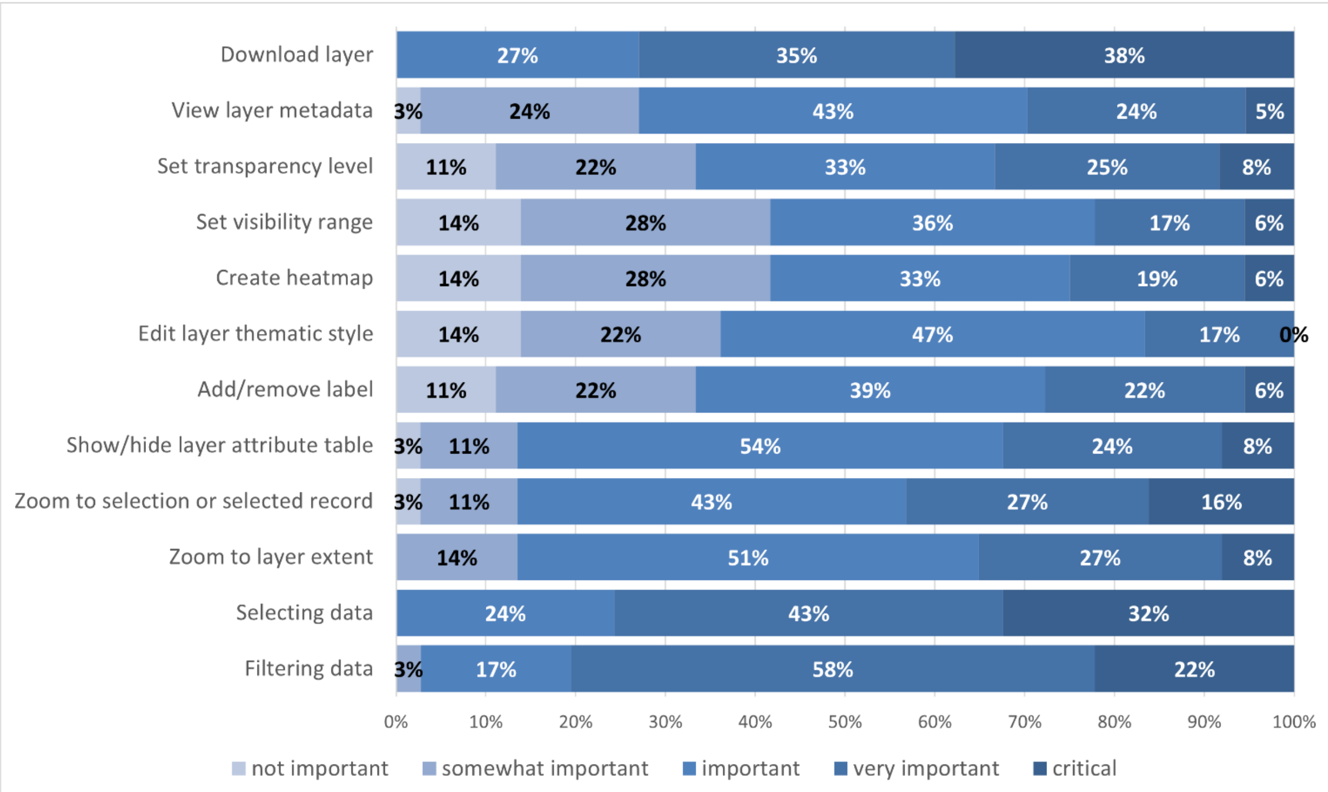
User's overall experienced using CTPP data interface
- somewhat difficult to batch select smaller geographies within a bigger one (i.e. tracts within a county) - map can be slow
A map is just a report in graphical format. Interchangeability is important, going back and forth. I'd like to be able to constrict/save a script of my query that can recreate the table or map. This will save me mch time having to relearn CTPP, since I am not a frequent user.
Both are not intuitive, selecting geographies is clunky, map interface usually doesn't work at all.
difficulties in selecting the correct geographies
Difficulty selecting desired geographies in table view
I find it incredibly frustrating to understand how to ask for the data that I want. It is also not clear what is available and what is not. It requires extensive training to use at a basic level.

User's overall experienced using CTPP data interface
I have issues with the map extent always zooming out to the whole country after I've selected a few tracts and then do anything else. I always end up scrolling in to zoom repeatedly.
It is just clunky, and not clear how to find the information I wanted. I usually end up watching those tutorials on Youtube each time I log in and want to find the data I am looking for. Which obviously speaks to a usability/learnability issue. Its also tough b/c the overall look/feel "seems" like the ACS, which has its serious limiations for data searching, but I generally can get to the data tables I need by clicking around.... its not the case with the CTPP. Also, I dont think I ever got the larger data downloads I submitted a req for. I remember writing an email days after my request, and found out someone had submitted so many queries it would take 13 weeks for my request to get to the top of the queue. I ended up getting some of the data via FTP.
It's often difficult to select track to track data flows for large counties. The application keeps wanting to default to State to State which is unhelpful for those working at the county level. Also it would be nice to be able to select data at the tract level for multiple counties for each tract. Right now, I haven't able to do this.
It's really not intuitive, that's the biggest issue. A well-designed interface shouldn't require the user to have any training. A help function or informational pointers should be enough.
Just in general too clunky, and difficulty navigating in the data layer to any small geography not located in Alabama
locating tables that fit the definition
Need to have easier ability to filter, select and deselect multiple geographies. And to see what has been selected in table view.
Past record exports limit
Search for a geography (e.g., county) brings up all the geographies with similar name around the nation. If i select my state at the beginning, subsequent geography searches should bring matching records only in my state.
Sluggish
sometimes there can be slow updating of selections in map view
The process of selecting custom geographies is laborious and non-intuitive
The table setup is not totally intuitive. Results can be very slow to load.
The table sorting is not intuitive, but I also have not taken the training.
They were only minor user issue, that once the quarks were figured out it was fine to use.
Unable to remove from currently selected, based on search terms. Can only clear all selected or unselect items individually.
Very counter intuitive to learn and difficult to download larger datasets e.g. all the flow data of a region.

3.2 User’s Impressions of Web Based CTPP Layer features

Figure 13 summarizes users’ ranking of specific layer functions. It shows that filtering data, selecting data, and downloading data layers are ranked as the most critical layer-related functions, while being able to edit layer thematic style is considered the least critical.

FIGURE 13: IMPORTANCE RATINGS OF LAYER RELATED FEATURES – USER INTERFACE

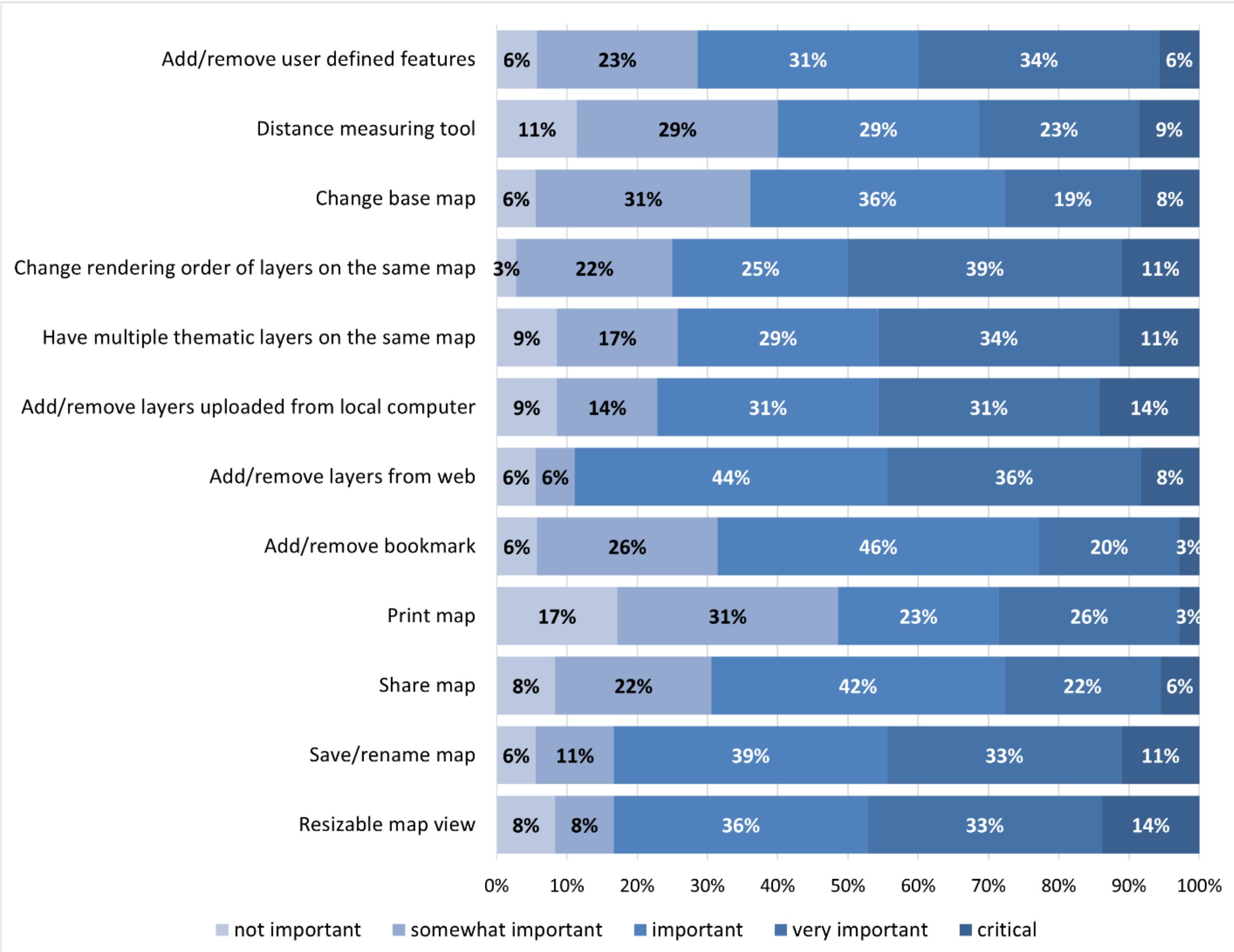


Three survey respondents (9% out of 33 who answered the question) felt that important layer-related capabilities were missing. Specifically, they cited access to the layer as a web feature service, the ability to customize a heat map, and the ability to download data as layer.

3.3 User’s Impressions of Web Based CTPP Map features

Figure 14 summarizes users’ ranking of specific map features. It shows that resizable map view and the capability to add/remove layer from users’ local computers are ranked as the most critical. Meanwhile, the capability to share maps or add/remove bookmarks are not considered very critical.

FIGURE 14: IMPORTANCE RATINGS OF MAP RELATED FEATURES – USER INTERFACE



Five survey respondents (or 15 out of 33 answering the question) felt that important map capabilities are missing, such as the ability to download data as a layer file, improved and intuitive interaction with geographic features to select origin-destination configurations of flow data, and layer labels. One user wrote:

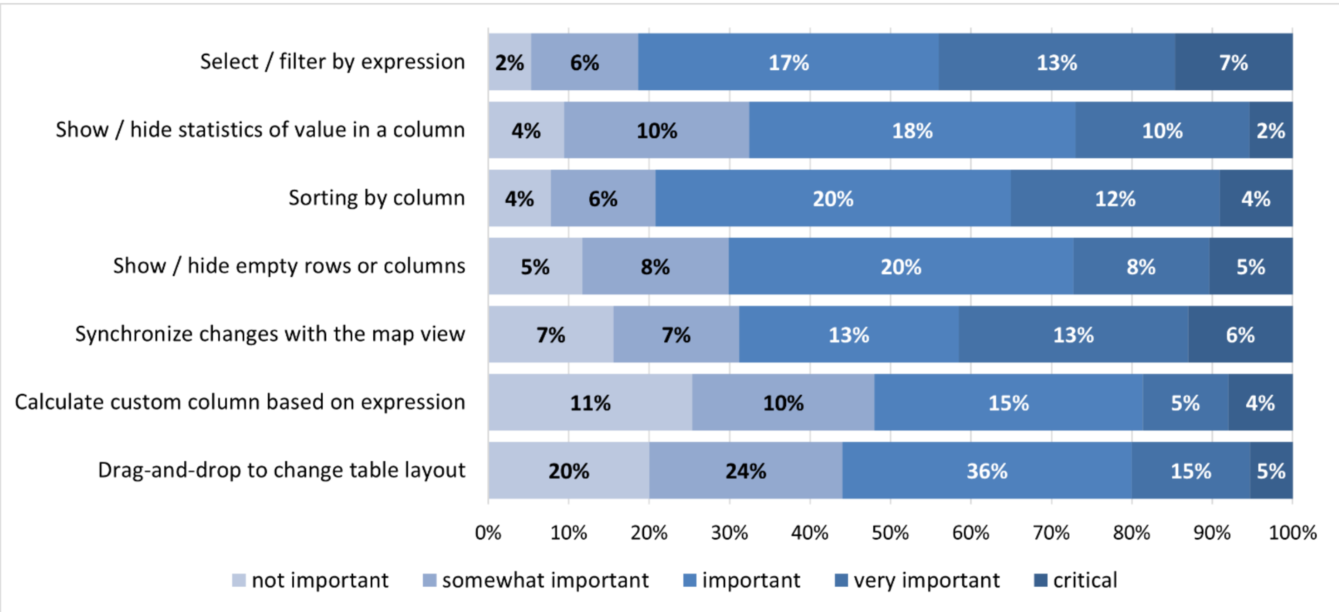
> "Provide me with a URL to the map service that renders the features and I will be able to use it in the environment I do my work, ArcGIS Online or ArcGIS Pro. This will save me many steps. You can save reinventing web mapping and let people use the FREE ArcGIS Online."

This is a sentiment that was echoed by participants in the CTPP GIS-T SIG organized by Manhan Group, as well as the smaller focus group of interested respondents to the survey. However, this approach would not meet the needs of users who do not have access to or are not familiar with GIS--including perhaps those previously identified as having "emerging use cases".

3.4 User's Impressions of Web Based Table features

Figure 15 summarizes users' rankings of specific table features. It shows that the ability to select and filter data by expression and the synchronized map view are ranked as the two most critical table features. Meanwhile, the ability to drag/drop to change table layout and to compute columns by expression are considered the least critical.

FIGURE 15: IMPORTANCE RATINGS OF TABLE RELATED FEATURES – USER INTERFACE



More people answered yes to the question "Is there a table-related capability which is currently missing but important to you?" than similar questions on layer or map-related capabilities. About 22% of survey respondents (N=68 for this question) felt that some important table capabilities are missing, such as:

- A GEOID that is directly compatible with Census GIS shapefiles.
- Alert to users when the margin of error is too large to reliably use the data for their planning assumptions.
- Export summary tables
- Export to GIS layer
- Extended scrolling without clicking through pages.
- There should be an easy way to select which column to display
- To remove or at least separate the MOE columns from the estimate of value columns
- Upload geography area such as ONTHEMAP LEHD data

Some users contributed longer responses to this open-ended question, such as:

> "In a multi-state analysis, it was difficult to get unique, ST_FIPS + Co_FIPS in a form I could easily use in GIS. I also had problems selecting counties in a state and counties that surround it, within a specified distance (e.g., two counties deep or w/in 60 miles, etc.)"
>
> "It's often hard to select tract level data for all the census tracts in a large county. The current application keeps wanting to default to state to state flows which are completely unhelpful if you need county tract to tract level data."
>
> "So far I've downloaded the info I need table-by-table, though the option to download batches of tables at once would be good for some users. This feature might already exist."

In summary, the survey, informed by our own testing, has allowed us to identify the relative importance of various layer, map and table view features, which, together with the major issues identified and summarized below, could help a potential contractor to develop a new web UI application.

- Lack of Automated Programming Interface (API) capability.
- Lack of web service that renders the features that a user can use in the environment they do work, ArcGIS Online or ArcGIS Pro, QGIS
- Slow query/rendering performance
- Specific ease-of-use/intuitiveness issues with software.
- The placement and grouping of menu items and tools need to match users’ expectations better
- Needs improved geographic aggregation, particularly map-based capabilities.
- Needs More output formats
- Needs software improvements over time
- Don’t allow the filtering out of unwanted columns.
- Needs capability to extract portion of `geoid` which matches `geoid` in census features better.

4 CTPP Users Focus Group

4.1 Summary

The user data gathering process included stakeholder outreach to understand requirements for a data access API and/or a more improved CTPP Web user interface. Towards this end goal, we recruited survey respondents to participate in a follow-up focus group discussion about the current system's pitfalls, limitations, weaknesses as well as strengths, including use cases and common tasks that are currently well-served. Specifically, the group discussed the following topics:

- End Goal of using CTPP data
- User's Work Flow
- Pain Points with CTPP Web application
- API applications and solutions
- Beta Testing Recruitment

4.2 Focus Group

The CTPP API/UI focus group met on May 20, 2021, with 11 participants representing DOT staff, consultants, a mobility technology provider, planning and transit agencies, and an academic researcher. The participants were primarily recruited from the pool of early respondents to the web-based survey, along with one attendee of the GIS-T SIG meeting on data access tools. A majority of the selected participants interacted with travel demand models, but not all were model developers, and some of the participants supported other planning activities and issues relating to a diverse set of topics such as environmental justice, housing, and transit ridership.

Since we could not directly ask survey respondents to evaluate a CTPP data access API which does not yet exist, we provided with information about examples of existing script-based and API-like interfaces to CTPP data created by various expert users, to solicit reactions and draw out responses on whether and how an API could meet users' needs. This included an example from one of the focus group participants, Dr. Greg Macfarlane, who gave a live demonstration of his interface and provided insights into its development, particularly commenting on how he would have proceeded differently had an API been available.

4.2.1 End Goal – What are we using the data for?

As mentioned before, a majority of respondents to both the survey and group discussion deal with travel demand models at various levels. CTPP data is also used for a wide variety of studies relating to commuter flows, including transit planning, housing and issues associated with COVID. For example, an agency transit planner in our focus group utilizes the data sets for travel forecasting and transit market analysis, while a consulting engineer in the group primarily utilizes Journey To Work and Reverse Journey work flows for development impact studies.

The experience of one particular CTPP "power user" in our focus group may be of particular note. A travel demand modeler at a small state DOT, this individual also responds to a wide variety of other data-related requests, including supporting other departments with any kind of data question--not just modeling, but environmental justice, transit, anything transportation related. A unique example mentioned utilized CTPP for analysis related to COVID, noting that while the data could not be used to directly measure travel behavior during the pandemic, it could be used to provide officials with a picture of the prior travel pattern serving as a starting point for behavioral adaptations. This user trained others in his office as well as several departments within the state (i.e. outside DOT) in using CTPP for their various studies, such as a recent project where the housing office needed to know how many people in a certain town came there to work versus how many people went to other places to work for a variety of reasons. If all the jobs available in the town were lower income drawing workers from elsewhere, then that is an argument for building more affordable housing. This user reported that this use case was well-served by the current CTPP web-based user interface, and represents an example of the kind of task that could be completed by someone who is not a transportation planner, doesn't have a travel demand model at their fingertips and is a technical person who knows how to code.

4.2.2 Work Flow – Process and Procedures?

Regarding workflow utilizing CTPP web application, the focus group discussion diverged based on applications and technical expertise, reflect divergent pain points, which are discussed in the following section.

A consulting travel demand modeler mentioned utilizing the CTPP for understanding average travel time for people who go to work every day in the county as well as pulling home-based-work trip length distributions for travel model calibration. Their typical process included both travel time table and map functions because, they could see county and also the census block in the study area they are looking at. They noted that they would like to look at these data in more detail (i.e. more stratified cross-tabulations), but found the interface difficult to navigate.

Another consulting engineer described themselves as an applier of travel demand models, not a developer coding or calibrating models. Their workflow involved importing CTPP data into GIS and developing spatial analyses to develop origin-destination routings from flows. They mentioned that the current web interface is cumbersome to use, but was wary of utilizing a code-driven interface (e.g. an API), not having sufficient programming experience to do so.

The public agency transit planner mentioned previously stated that they would select the table that they wanted first, then selected the geography, to be later incorporated into Excel for further tabulations. This user stated that they felt like everything in the web user interface was geared to small state to state flows. They sometimes might need the flow between five counties, but the software kept assuming they wanted state to state flows, noting that the relevance of state to state flows is particularly low the western United States. This user found

that they had to do multiple iterations of the same exercises just to get the software to behave correctly, echoing the sentiment mentioned by others that the software is not intuitive to use.

A city DOT user qualified that they did not have as much experience as the rest of the focus group. They did a study last year on COVID-related topics and found it really cumbersome and difficult to pull out the data they needed. They were doing a simple analysis that involved multiple states, needing FIPS codes in character format that had state and county in the same field. Ideally they would like to interact with a simple query requesting specific data items and putting them in a table without having to go through and review many extraneous details that needed to be filtered to get to the desired data.

4.2.3 Pain Points & Solutions – Missing features and/or potential improvements

In general, the group all shared similar observations that the current web interfaces is cumbersome to use. A few participants noted the limitations associated with download data size limitations, especially when dealing with large areas, noting that multiple iterations of downloads can take days to accomplish. Effective documentation was also a concern; however, during the group discussion it became clear that not all of the participants were aware of the training resources available.

A mobility data provider representative mirrored previous comments regarding the UI/UX, specifically noting that answering a question about to where everyone who lived in a city commuted did not seem very intuitive, versus other sources where people get their data or information from. This focus group participant stated that they had skimmed through the CTPP training videos to figure out what they needed to do, but noted that if users didn't use the CTPP tool very frequently you then they might forget what they had learned from the videos.

A university researcher who participated in the focus group stated that they appreciated the availability in CTPP data of basic three way tables that don't exist in the ACS tabulations, but are helpful in setting controls for synthetic population. They pointed out "Once I create a synthetic population that's controlled on more dimensions, then I can use that synthetic population to do detailed market analysis. And yet it's the output of a model, but so is everything in the CTPP... at the tract level, I'm much more likely to get margins of error that are smaller than the estimate I'm trying to control at. And so... I create the synthetic population controlled to those three way tables at the tract level."

The researcher continued: "it annoys me greatly when I download a table from the Web interface, and it's littered with extra data. Like there's a header that makes it easy to look at when you open it up in Excel. But... I've just downloaded a twenty thousand line table; I've no desire to look at this thing in Excel. Thus, having the tables be machine readable from the beginning would be really, really helpful. Let's spread these columns out, or instead of putting every detail of the geographic element in the same cell, separated by commas and spaces, kind of irregularly. Put them in different columns so that you can use your data tools ahead of time or sort of out of the box rather than go through seven hours of data cleaning just to get a

table... it's clear that the design of this was intending that people would download 10 rows of data at a time and have a nice little spreadsheet that they would create. But that's not how I've ever used it. If all I wanted was ten rows, I would just leave it on the web and take a screenshot."

4.2.4 Future Directions and Potential Benefits of an API

One of the primary goals of the focus group was to explore questions related to the future web services API for CTPP data which could not be answered via the survey. Since this API does not exist yet, "DIY" script-based interfaces to the data previously published to Github by users were examined as possible sources for lessons learned and/or good future practices.

The aforementioned university researcher who participated in the focus group presented their own script-based interface, giving a real time example of its functionality. They also described the learning process involved developing this interface, mentioning that they gained some very important insights which might suggest possible future directions. They qualified that since APIs are designed for computer programming languages which are usually not easy for people who are used to graphical user interfaces, a CTPP would not serve many existing users.

As a model for one possible path that could be followed by the CTPP, the researcher described their experience with Census APIs, noting that there were a variety of packages that individuals had developed for R, including one developed by the Census Bureau. "Then a developer came along with one [TidyCensus] that was... really clean and easy to use. As a result, the whole community just moved over to that one." The researcher commented: "I use TidyCensus in every one of my papers. It is a fabulous tool."

The state DOT travel demand modeler previously described also identified themselves as an avid user of TidyCensus. They were especially pleased that the package developers added functions to map flows automatically, noting that such functions and features like that, which save users some time from having to write a whole new script to do it themselves, would be helpful to incorporate in to future improved CTPP interfaces.

The TidyCensus example illustrates that the packages written for scripting languages to access the data through an API add as much or more value as the API itself, and thus it would be helpful if the API developer could also provide a package for access using R or Python as well as basic usage examples in R or Python (or some user samples). However, the university researcher emphasized that the developer should primarily focus their efforts on a well-documented API and then the community, noting that "it's easier to create the API and let the community create the packages rather than try to start with the packages in mind." They felt that the census API and the TidyCensus R package should serve as a model for future CTPP API development, noting that it would give people the ability to develop their own front ends, serving different use cases and/or user types.

4.2.5 Solution suggestions

The rest of the group noted that while they are very interested in the points raised by the university researcher and power user, not all of them had the technical knowledge to exploit all of those features. Furthermore, the power user noted that when training other departments, a simple web interface would remain critical for simple practical queries. It should be noted that the majority of the group stated that they would prefer that both an effective, intuitive UI and a well-documented API be developed, although the UI could be stripped-down in terms of features, relative to the existing interface. Ideally, the API would be developed with significant open-source components allowing for the user community to develop custom tailored applications and data access packages (similar to what occurred with TidyCensus).

4.2.6 Recruitment for Beta Testers

The focus group hosts described the second phase of this project, including the participation of beta testers to work with the chosen vendor in guiding the design of future improvements. We emphasized that the project will need user community feedback going into greater depth through an iterative process: come up with a solution, test that solution, establish whether it addresses the problem effectively, or creates more problems. A quick survey of the group indicated that all of the participants of the focus group would be interested in participating in a future Beta testing cohort group.

5 Technical analysis of database design options

To better understand the technical requirements and challenges associated with the type of desired solution identified by focus group participants, Manhan Group developed and tested a database for CTPP serving as a reference implementation for testing to help guide the design of development project tasks/deliverables as well as the estimate of development hours/cost for the web user interface and API.

An API/Web UI project is typically built with a 3-tier architecture, including: 1) database backend, 2) API/Services Server (as the middleware), and 3) the front-end web application. Our technical review provides assessment on of the first two of these three aspects by testing a reference implementation using open source PostgreSQL/PostGIS as the database backend , two open source mapping servers as the OGC Web Services/REST API servers and Postgraphile as the GraphQL API server. For the front-end assessment, the project management team evaluated the existing solution at http://data5.ctpp.transportation.org/ in addition to the user survey analysis discussed previously.

5.1 PostgreSQL/PostGIS database as CTPP Database backend

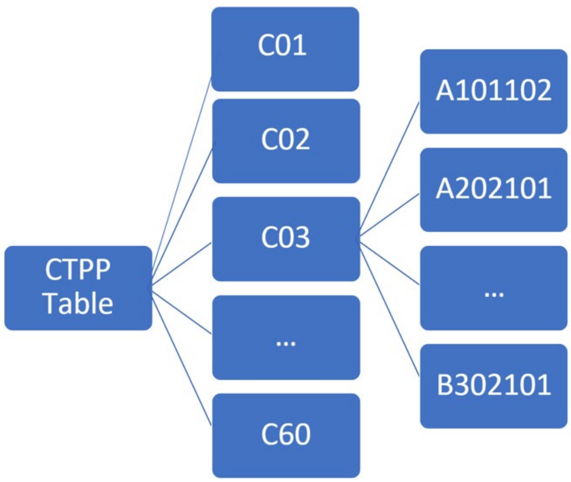
The open source PostgreSQL+PostGIS was chosen as the database engine because PostgreSQL is fast, reliable, ACID compliant and PostGIS is renowned for its large collection of spatial functionalities and fast performance. (PostgreSQL/PostGIS is also available as a serverless cloud solution called AMAZON Aurora, which offers advantages such as automatic scalability and backup, global distribution of data, and freedom from infrastructure provisioning and management. The assessment of the database backend focuses on aspects which are relevant to both on-premises and cloud serverless solutions. Specifically three specific database performance improvement techniques were evaluated: table partitioning, index, and materialized views. Since other proprietary database engines such as ORACLE and Microsoft SQL server have similar techniques as well, the knowledge gained through this assessment might still be useful in evaluating potential contractors' proposed solutions for performance improvement even if a different database engine is chosen.

5.2 Table Partitioning and Index

The 2012-2016 CTPP data from the FTP site at ftp://data5.ctpp.transportation.org/ was loaded into a test PostgreSQL database for testing purpose. The data from the FTP site is in long form, with fields `geoid`, `tblid`, `lineno`, `est`, `moe`, and `source`. Fields `geoid`, `tblid` and `lineno` uniquely identify each row. The field `geoid` identifies a geographical entity such as a census tract. The first three characters of `geoid`, in the form of `C01`, `C02`, …, `C60` indicate different geographical summary levels; The field `tblid`, containing values with letter A or B

followed by 6 digits (`A101102`, `A202101`, `B302101`, ...) identifies a CTPP tabulation. The field `lineno` indicates different fields belonging to the same table tabulation. We keep the long form data structure in the database because this structure is simple and flexible enough to accommodate any expected changes in the future CTPP tabulation. Besides, this structure is also naturally suited for one the specific technique we would like to evaluate: horizontal table partitioning. For this study, the table partitioning by both geographical summary level and CTPP tabulation is implemented. The partition structure is illustrated in Figure 16:

FIGURE 16: PARTITION STRUCTURE ILLUSTRATION



The above structure results in about 2000 partitions, with more than 300 having more than 1 million records and over a dozen having even more than 40 million records. The tables with millions of records, for example tables of summary level C54 (census tract), could be further partitioned by state. To test the impact of partitioning versus index on query performance, queries against tables using different indices as well as before and after partitioning were analyzed and compared.

5.2.1 Table Partitioning Test

The query shown in Figure 17 is against a table named `ctpp_v1.a302103_c54`, which is not partitioned by state. It contains about 8.9 million records for the whole country and for all attributes (indicated by `lineno`) in tabulation `A302103`. The query aims to extract information for the Delaware Valley Regional Planning Commission (DVRPC) metropolitan area.

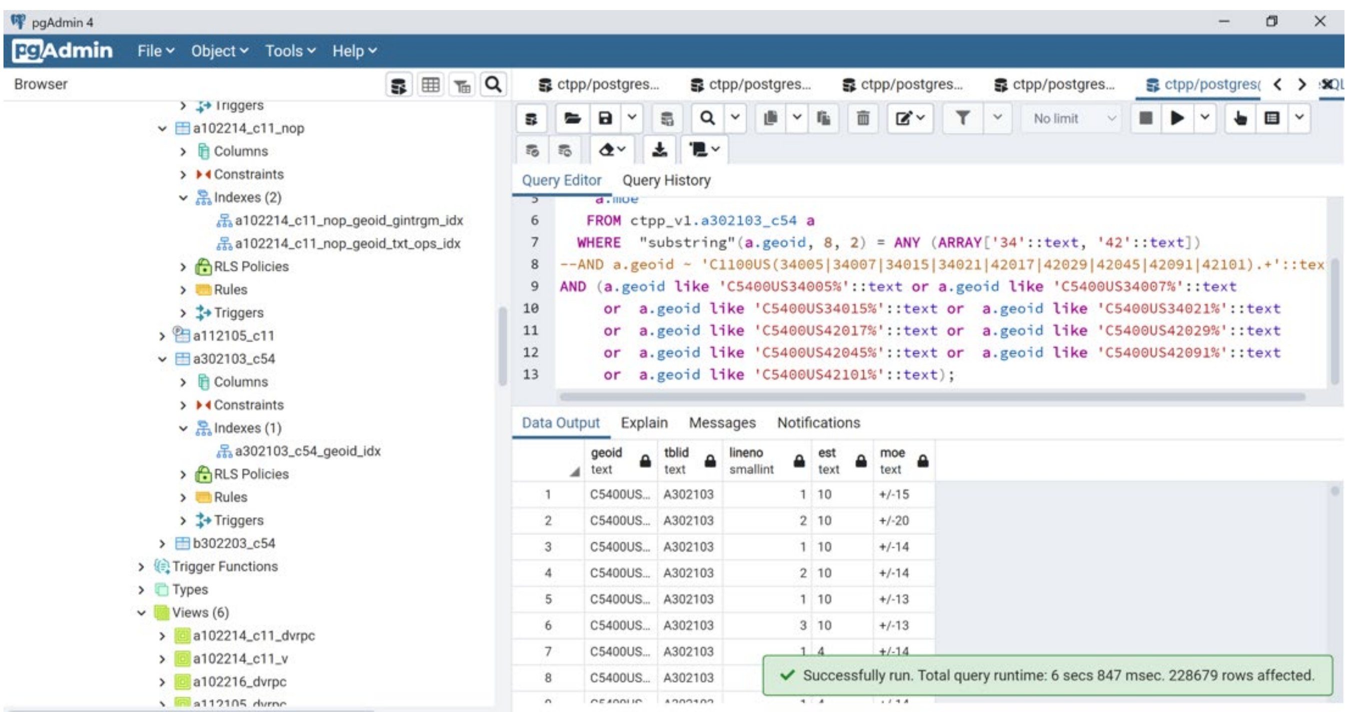
FIGURE 17: QUERY ON TABLE A302103_C54 TO SELECT DATA FOR DVRPC AREA

Query Editor   Query History

```
SELECT a.geoid,a.tblid, a.lineno, a.est, a.moe
FROM ctpp_v1.a302103_c54 a
WHERE  "substring"(a.geoid, 8, 2) = ANY (ARRAY['34'::text, '42'::text])
AND (a.geoid like 'C5400US34005%'::text or a.geoid like 'C5400US34007%'::text
or  a.geoid like 'C5400US34015%'::text or  a.geoid like 'C5400US34021%'::text
or  a.geoid like 'C5400US42017%'::text or  a.geoid like 'C5400US42029%'::text
or  a.geoid like 'C5400US42045%'::text or  a.geoid like 'C5400US42091%'::text
or  a.geoid like 'C5400US42101%'::text);
```

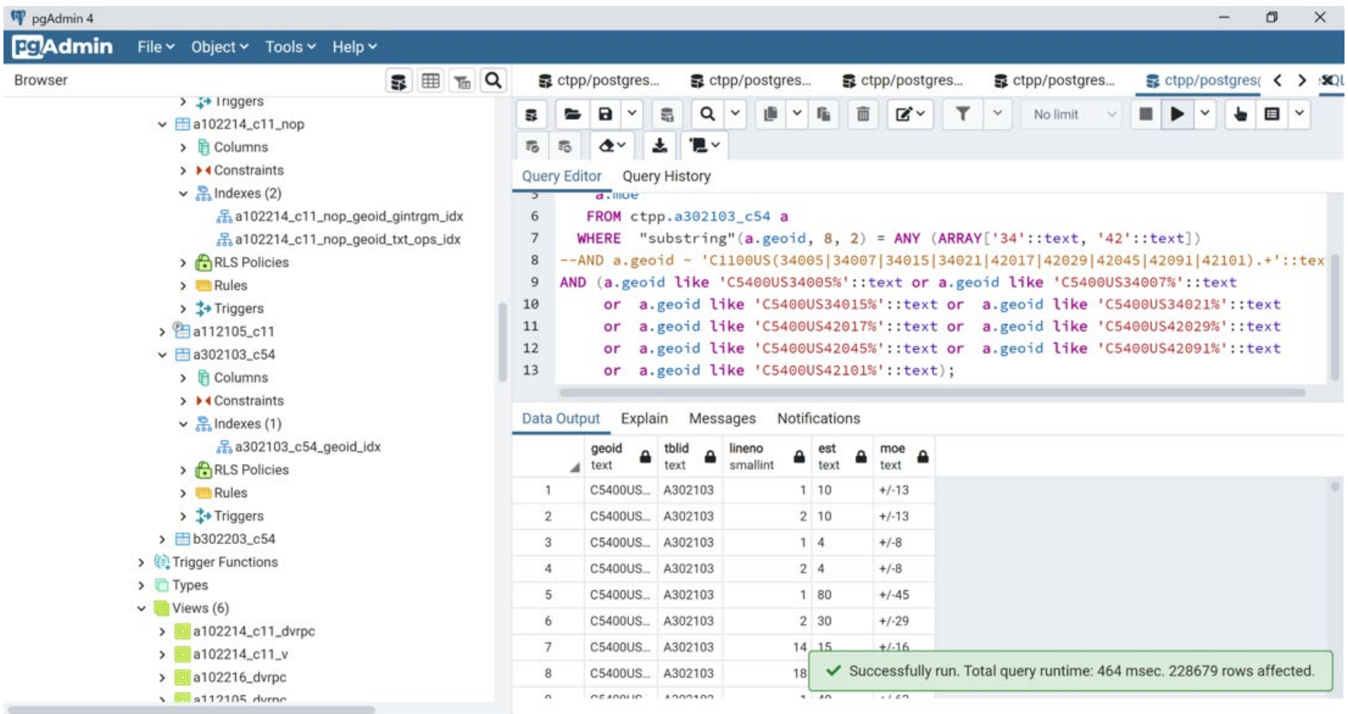
When the query was run for the first time (the index used not in cached memory yet), it took 6.847 seconds to finish as shown in Figure 18.

FIGURE 18: QUERY RUN TIME ON TABLE A302103_C54 WHEN UNPARTITONED



The same query, run against a partitioned table, also for the first time, took 0.464 seconds to finished as shown in Figure 19.

FIGURE 19: QUERY RUN TIME ON TABLE A302103_C54 WHEN PARTITONED



The subsequent query runs against both tables would take less time with the query against the partition table seems to be always faster (approximately 2 seconds vs 1 second).

Indices were used to filter rows in the execution of both queries. Further investigation indicated that differences in the query performance were due to the difference in the index size as shown in Figure 20.

FIGURE 20: INDEX SIZE OF PARTITIONED VS UNPARTITIONED TABLE

Query Editor Query History

```sql
SELECT n.nspname,c.relname,
pg_size_pretty(pg_relation_size(i.indexrelid::regclass)) AS index_size,
pg_size_pretty(pg_relation_size(i.indrelid::regclass)) AS table_size
FROM pg_index i
     LEFT JOIN pg_class c ON c.oid = i.indexrelid
     LEFT JOIN pg_namespace n ON n.oid = c.relnamespace
WHERE n.nspname in ('ctpp', 'ctpp_v1')
AND c.relname in ('a302103_c54_34_geoid_idx',
                  'a302103_c54_42_geoid_idx',
                 'a302103_c54_geoid_idx')
AND pg_relation_size(i.indrelid::regclass) > 0;
```

Data Output Explain Messages Notifications

	nspname name	relname name	index_size text	table_size text
1	ctpp	a302103_c54_34_geoid_idx	11 MB	27 MB
2	ctpp	a302103_c54_42_geoid_idx	13 MB	35 MB
3	ctpp_v1	a302103_c54_geoid_idx	261 MB	704 MB

The query in Figure 20 returns the size of the index used in each of two queries of this section. In the case of unpartitioned table, the size of the index is 261MB. While in the case of partitioned tables, since the data for the two states of interests (state fip: 34 and 42) are actually in two much smaller table partitions, the index the query would need to look through to find the data is also much smaller (11MB for state 34 table and 13MB for state 42 table); hence the query is faster.

However, partitioning doesn't always result in better query performance. When the unpartitioned table is small enough or the partition resulted in too many tables, the overhead caused by the need to traverse the partition structure can override the benefits gained by smaller index size. Hence, we recommend the partitioning by state only for tables of CTPP part 3 and of small geographies such as census tracts.

5.2.2 Index for Pattern Matching

The `geoid` field in CTPP tables combines a geographical summary level code with the Census FIPS codes for state, county and smaller geography all together; hence there is a need to query the data using a substring of the field. Indexing for pattern matching can greatly improve the speed of this kind of query. PostgreSQL offers two types of operator classes which support indices for pattern matching: trigram operators classes `gin_trgm_ops` or `gist_trgm_ops` through `pg_trgm` extension and the built-in operator classes `text_pattern_ops`,

varchar_pattern_ops, and bpchar_pattern_ops. The query performance of index defined by the operator class gin_trgm_ops was compared with that defined by the operator class text_pattern_ops below.

The following two queries are both against a table named ctpp_v1.a102214_c11_nop. It contained about 7 million records for the whole country and for all attributes (indicated by lineno) in tabulation A102214. As before, the query aims to extract information for the DVRPC metropolitan area.

The first query, shown in Figure 21, uses the regular expression a.geoid ~ 'C1100US(34005|34007|34015|34021|42017|42029|42045|42091|42101).+'::text in its "where" clause to extract data using FIPS codes of counties in this metropolitan area.

FIGURE 21: QUERY USING REGULAR EXPRESSION TO SELECT DATA FOR DVRPC AREA

Query Editor   Query History

```
SELECT a.geoid, a.tblid, a.lineno,  a.est, a.moe
FROM ctpp_v1.a102214_c11_nop a
WHERE "substring"(a.geoid, 8, 2) = ANY (ARRAY['34'::text, '42'::text])
AND a.geoid ~ 'C1100US(34005|34007|34015|34021|42017|42029|42045|42091|42101).+'::text;
```

Figure 22 shows that the above query took about 1.451 seconds to finish and an index named "a102214_c11_nop_geoid_gintrgm_idx" is used to speed up the query.

FIGURE 22: RUN TIME AND QUERY PLAN OF THE QUERY USING REGULAR EXPRESSION

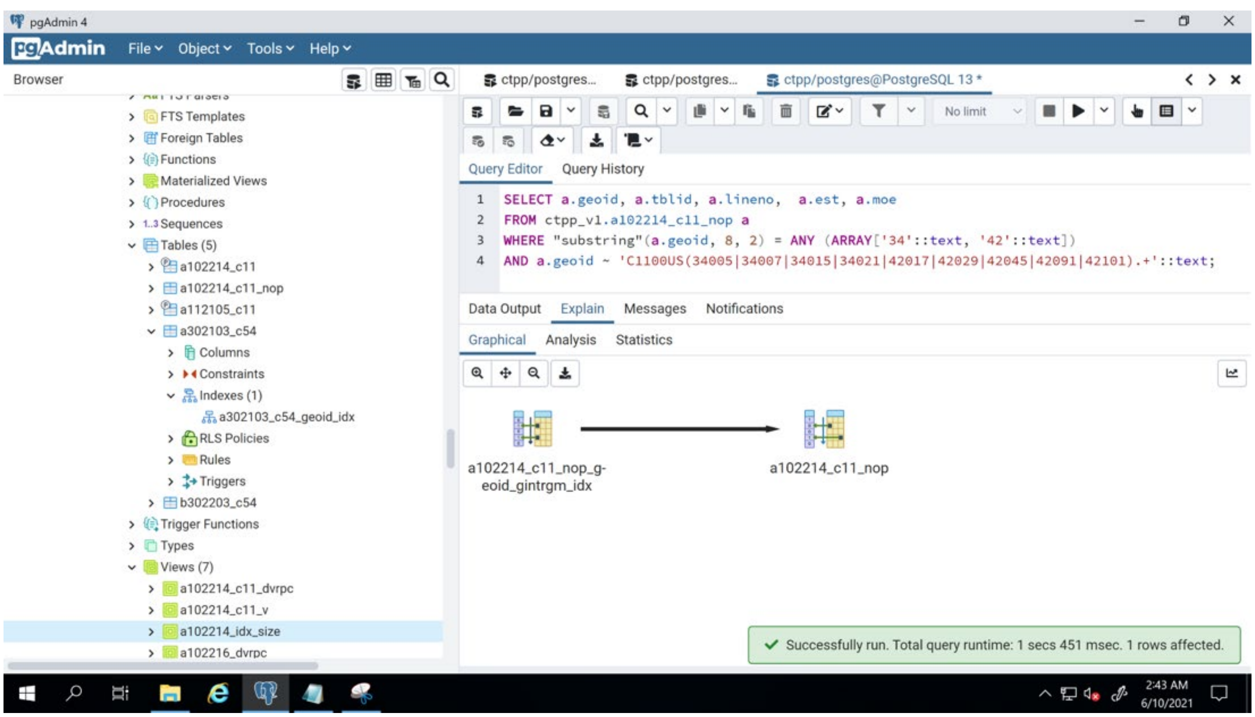


Figure 23 shows a query which uses the “like” expression in its “Where” clause instead. Unlike the first query, the second query needs to spell out the condition for each county individually.

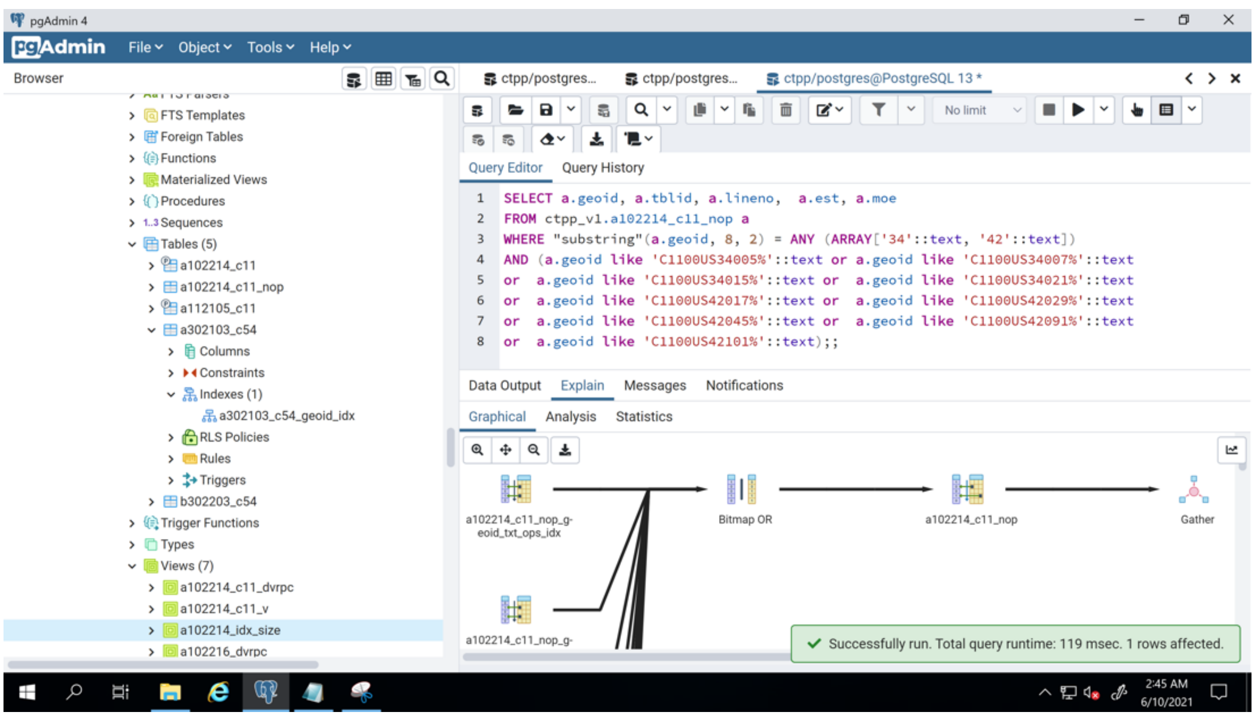
FIGURE 23: QUERY USING “LIKE” EXPRESSION

Query Editor Query History

```
SELECT a.geoid, a.tblid, a.lineno,  a.est, a.moe
FROM ctpp_v1.a102214_c11_nop a
WHERE "substring"(a.geoid, 8, 2) = ANY (ARRAY['34'::text, '42'::text])
AND (a.geoid like 'C1100US34005%'::text or a.geoid like 'C1100US34007%'::text
or  a.geoid like 'C1100US34015%'::text or  a.geoid like 'C1100US34021%'::text
or  a.geoid like 'C1100US42017%'::text or  a.geoid like 'C1100US42029%'::text
or  a.geoid like 'C1100US42045%'::text or  a.geoid like 'C1100US42091%'::text
or  a.geoid like 'C1100US42101%'::text);
```

Figure 24 shows that the second query only took 0.119 seconds to return the same information as before and an index name “`a102214_c11_nop_geoid_txt_ops_idx`” is used to speed up the query.

FIGURE 24:RUN TIME AND QUERY PLAN FOR THE QUERY USING "LIKE" EXPRESSION



As with the evaluation in the previous section, we checked the index size. In this case, the difference in the query performance is again seems to be caused by the difference in the index size, as shown below.

FIGURE 25: SIZE OF INDEX USING TRGM_OPS VS TEXT_PATTERN_OPS

Query Editor   Query History

```
SELECT n.nspname, c.relname,
    pg_size_pretty(pg_relation_size(i.indexrelid::regclass)) AS index_size,
    pg_size_pretty(pg_relation_size(i.indrelid::regclass)) AS table_size
   FROM pg_index i
     LEFT JOIN pg_class c ON c.oid = i.indexrelid
     LEFT JOIN pg_namespace n ON n.oid = c.relnamespace
  WHERE n.nspname = 'ctpp_v1'
  AND c.relname in ('a102214_c11_nop_geoid_txt_ops_idx',
                    'a102214_c11_nop_geoid_gintrgm_idx')
  AND pg_relation_size(i.indrelid::regclass) > 0;
```

Data Output   Explain   Messages   Notifications

	nspname name	relname name	index_size text	table_size text
1	ctpp_v1	a102214_c11_nop_geoid_gintrgm_idx	143 MB	460 MB
2	ctpp_v1	a102214_c11_nop_geoid_txt_ops_idx	49 MB	460 MB

The index named "`a102214_c11_nop_geoid_gintrgm_idx`" is defined using the `gin_trgm_ops` operator class, whereas the index named "`a102214_c11_nop_geoid_txt_ops_idx`" is defined using the `text_pattern_ops` operator class. Figure 25 shows that the latter has much smaller size which results in much faster query. However, it only supports left anchored search pattern, when the wildcard is at the end of a search term; hence its use is more limited than trgm_ops index which supports pattern match in general, left anchored or not.

Other factors besides the size of the index, such as btree index versus `gin` or `gist` index, can impact the query performance that are not addressed here. The findings of this assessment only indicate that the design of index techniques need to account for the way users might query the database.

5.3 Materialized View

Like views, Postgresql materialized views can be used to represent the data in a more natural or intuitive way without changing the underlying table. Unlike views, data in Postgresql materialized views are stored physically and can have an index, thus having potentially much faster query performance. For this study, we used materialized views to create wide form presentation of data for CTPP tabulations. The number of columns in these views can vary greatly from one tabulation to another, which in some sense compensates for the oversimplification of long form table structure. More importantly, in the wide form view, each record identifies a unique geographical entity which makes it easier to join with spatial table and then be used in GIS applications or Mapping servers. Since the conversion from long form data to wide form data requires a time-consuming function (cross tabulation) a materialized view is chosen to store the result of the conversion query. The following section demonstrates the improvement of query performance through the use of materialized views.

5.3.1 Sample Query of a Materialized View

The materialized view defined in Figure 26 converts the long form data of summary level `c11`, `tblid a101101` for California (statefp 06) into wide form using cross tabulation function.

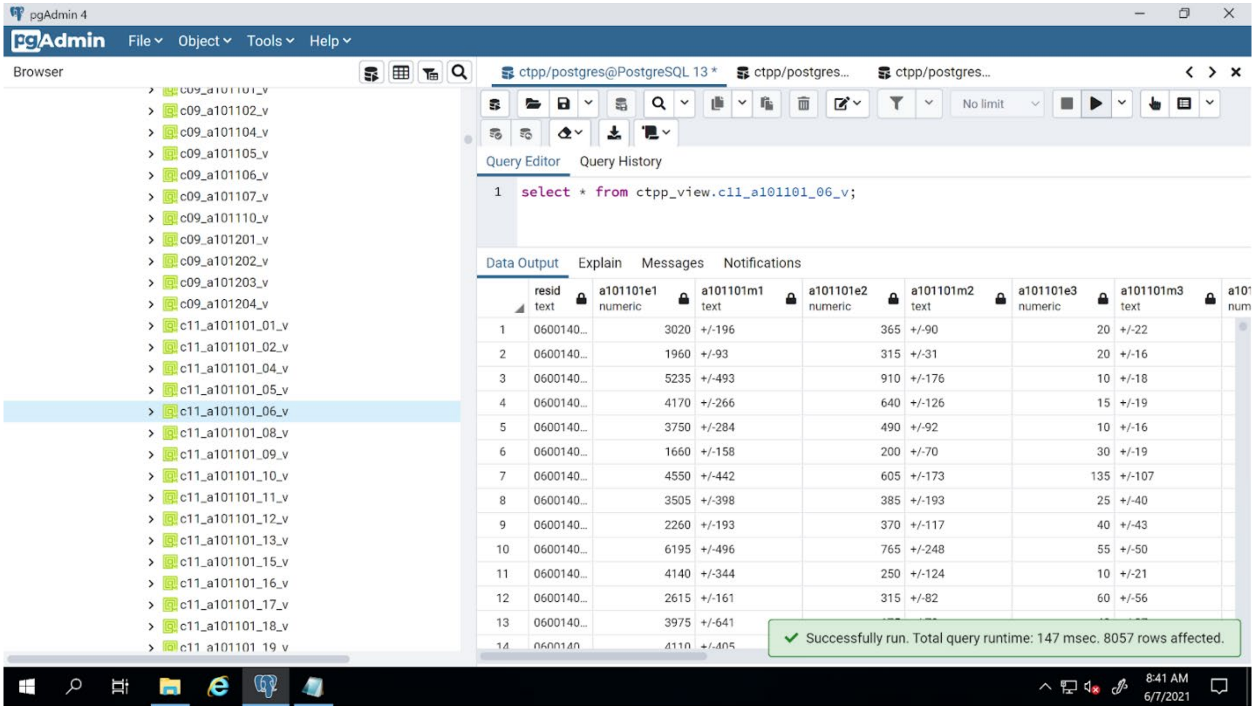
FIGURE 26: QUERY TO CREATE A MATERIALIZED VIEW USING CROSS TABULATION FUNCTION

Query Editor   Query History

```
CREATE MATERIALIZED VIEW ctpp_view.c11_a101101_06_v AS
SELECT * FROM crosstab('select substring(geoid,8) as resid, tblid || mychar || lineno,
case when mychar=''e'' then txt2numeric(est)::text else moe end as val
from ctpp.d2012thru2016 cross join unnest(array[''e'',''m'']) as mychar
where
left(geoid, 3)=''C11'' and tblid=''A101101''  and substring(geoid,8,2)=''06''
order by geoid,tblid,lineno,mychar'::text, 'select tblid || mychar || lineno
from ctpp_dict.acs_ctpp_2012thru2016_table_shell
cross join unnest(array[''e'',''m'']) as mychar
where tblid=''A101101'' order by tblid,lineno,mychar'::text)
ct(resid text, a101101e1 numeric, a101101m1 text, a101101e2 numeric,
a101101m2 text, a101101e3 numeric, a101101m3 text, a101101e4 numeric,
a101101m4 text, a101101e5 numeric, a101101m5 text, a101101e6 numeric,
a101101m6 text, a101101e7 numeric, a101101m7 text, a101101e8 numeric,
a101101m8 text, a101101e9 numeric, a101101m9 text, a101101e10 numeric,
a101101m10 text, a101101e11 numeric, a101101m11 text)
```

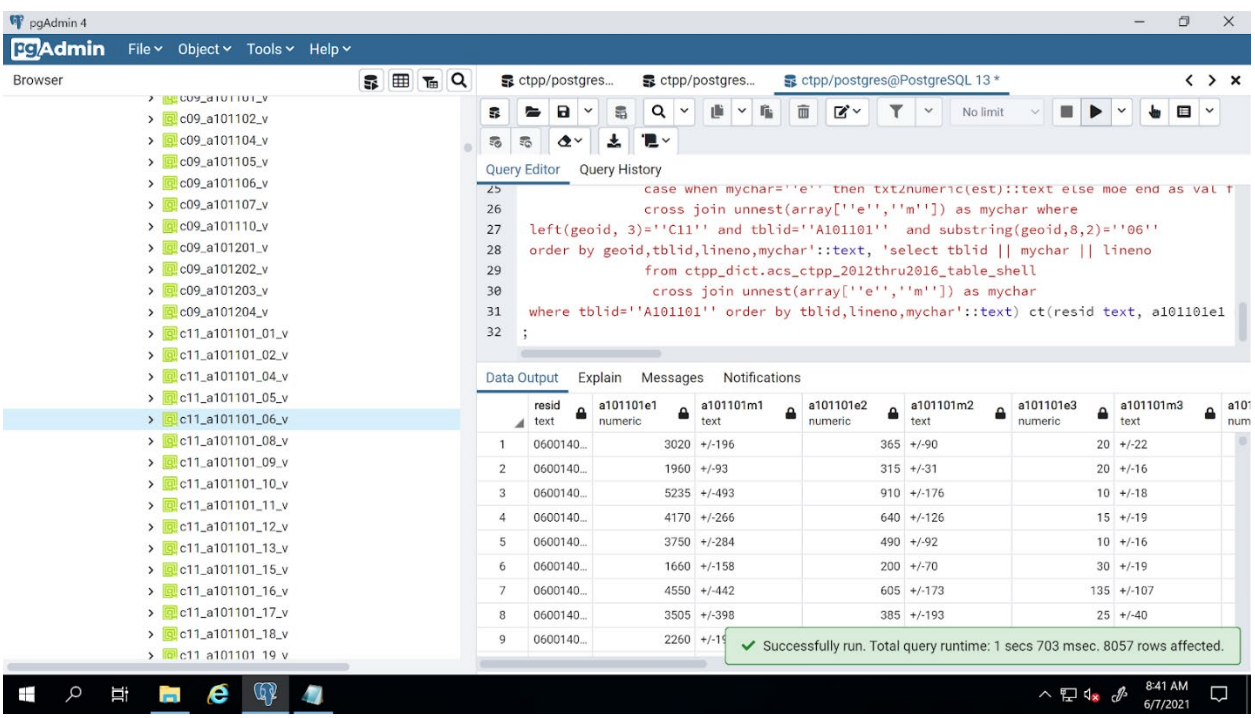
Figure 27 shows that it takes 0.147 seconds to return all records from the materialized view created above.

FIGURE 27: RUN TIME OF THE QUERY ON THE MATERIALIZED VIEW



The query shown in Figure 28 extracts the same data by querying against the table used to generate the materialized view directly and takes 1.703 second to finish.

FIGURE 28: QUERY AGAINST THE LONG FORM TABLE USING CROSS TABULATION FUNCTION

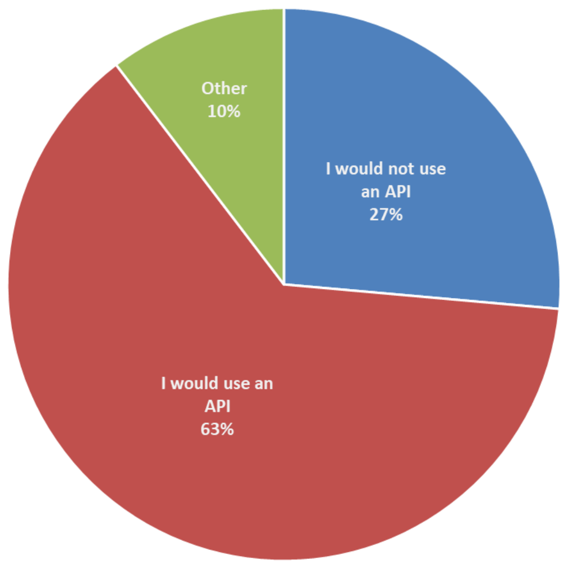


Between the two queries in Figure 27 and 28, the query against the materialized view is significantly faster. This comparison demonstrates that use of materialized views is a technique which is worth exploring for the improvement of query performance in case of queries involving time consuming calculations like cross tabulation.

6 API Design Considerations

The survey respondents' shows that there is enthusiastic support behind the development and utilization of an API, with 63% saying they would use one if developed, as shown in Figure 29.

Figure 29: Interest in using API



This is interesting and surprising, given that most users (71%) said that they relied upon the web-based data access interface provided by AASHTO, and only 10 out of 168 respondents (or 6%) said that they downloaded CTPP data via FTP and then performed analysis and summary tasks using a scripting language such as Python or R (i.e. the closest workflow to that required in order to use a web services API). Several explanations for these responses are possible:

- Respondents may not have understood what an API was or what skills would be required in order to use it;
- Respondents may be willing to learn the skills (i.e. scripting languages) necessary to use the API, motivated by its promise of a more straightforward workflow than the current web user interface offers; or
- Respondents may understand what an API is as well as the potential it opens up for third parties (e.g. more technical staff or consultants) to develop applications and tools facilitating data access for non-technical users, and anticipate using an API in this way.

In any event, given the high hopes placed on this planned new CTPP data access tool, it is especially important that an API be designed in a careful and thoughtful manner to best serve the users who may benefit from it. The following sections describe our analysis of various API design techniques and their potential application to CTPP data access.

6.1 Possible Web API/Web Service Server design

REST or RESTful (Representational State Transfer), a popular web API design technology, converts HTTP requests (post, get, put, delete) into CRUD (create, read, update, delete) operations against data or database in the backend. REST or RESTful refers to a set of specific architectural principles identified by Roy Fielding in his doctoral thesis, "Architectural Styles and the Design of Network-based Software Architectures.", which are summarized in the book "RESTful Java with JAX-RS" by Bill Burke[2] and quoted below:

- ***Addressable resources***
  The key abstraction of information and data in REST is a resource, and each resource must be addressable via a URI (Uniform Resource Identifier).
- ***A uniform, constrained interface***
  Use a small set of well-defined methods to manipulate your resources.
- ***Representation-oriented***
  You interact with services using representations of that service. A resource referenced by one URI can have different formats. Different platforms need different formats. For example, browsers need HTML, JavaScript needs JSON (JavaScript Object Notation), and a Java application may need XML.
- ***Communicate statelessly***
  Stateless applications are easier to scale.
- ***Hypermedia As The Engine Of Application State (HATEOAS)***
  Let your data formats drive state transitions in your applications.

Because of the above principles, REST API has many benefits, including scalability, fast performance through cache, capability to interact with different clients in a consistent way, linkable resources, and more.

Unlike nonspatial data, requests for spatial data could have infinite variations. For example, for the spatial query of restaurants by a user standing on a street, the query differs depending on where the user is standing as well as the maximum distance the user is willing to travel to get to the restaurant. The combination of these parameters is an infinite set because for the data domain for both coordinates and distance are any real number or any positive real number. Mapping servers are designed to solve this problem by reading data from the database and

[2] RESTful Java with JAX-RS, Bill Burke, https://www.oreilly.com/library/view/restful-java-with/9780596809300/ch01.html

rendering images and spatial data for delivery through the web on the fly based upon user requests. For spatial enabled web API design, the Open Geospatial Consortium (OGC) is the source of "[royalty free, publicly available, open geospatial standards"](), including the decades old OGC web service standards (WMS, WFS, WCS, WPS, etc.) and the recent OGC API Features also known as WFS3. Unlike the legacy OGC web service standards, OGC API Features is considered RESTFUL. Most OGC standards are supported by popular mapping servers like MapServer, GeoServer, QGIS Mapserver, and ArcGIS Server, which can work with different spatial database engines. If only PostGIS database needs to be supported, a lightweight RESTful geospatial feature server `pg_featureserv` is a good alternative. `pg_featureserv` supports OGC API - Features REST API standard.

GeoServer can be easily deployed and has a web user interface, whereas `pg_featureserv` needs minimum configuration. Both server software packages were tested in this study. The GeoServer software was installed on a different computer from the PostgreSQL/PostGIS database to test the performance of a remote connection. A `pg_featureserv` service was set up and run on the database server.

GeoServer provides REST API endpoints which allow programmers to interact with the server via scripting languages to add/delete layers or conduct other server management tasks. Due to the limited budget of this project, our evaluation focused on the use of the GeoServer web interface to create web service layers.

Sample GeoServer service layers were created using SQL view by joining TIGER geographical tables in the PostgreSQL database with the CTPP data table or views. Both WMS services and WFS services were tested successfully by loading the service layers thus defined in the QGIS desktop applications.

The Geoserver parameterized SQL view was also tested successfully. For example, the SQL statement in Figure 30 allows a user to define one service layer for the query of data for any state by using parameter "%statefp%" rather than a static state FIP code:

FIGURE 30: QUERY TO CREATE GEOSERVER PARAMETERIZED SQL VIEW

```
SELECT a.gid, a.name, a.namelsad, a.mtfcc, a.funcstat, a.aland, a.awater,
a.intptlat, a.intptlon, b.*, a.the_geom::geometry(multipolygon,4269)
FROM tiger.tract a JOIN ctpp_view.c11_a101101_%statefp%_v b
ON b.resid = a.tract_id::text
where a.statefp='%statefp%';
```

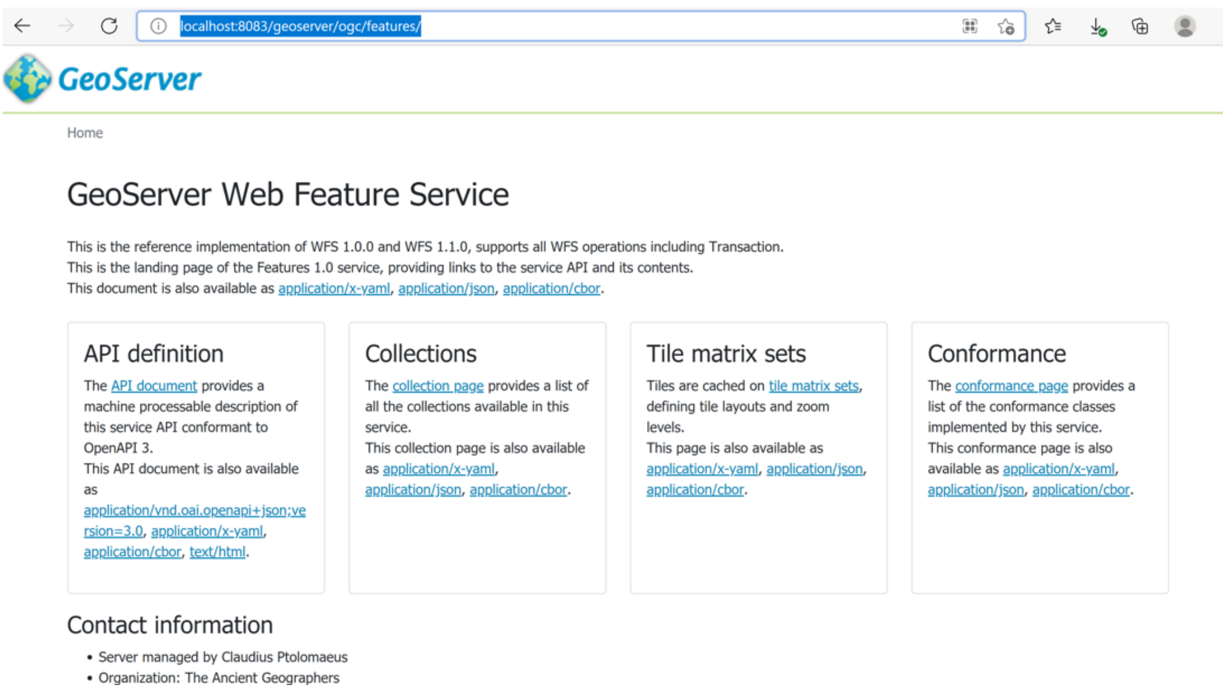
While requesting the web services, the user can add a "`viewparams`" query string to the URL to get the data for the state of interest. A sample url returning json representation of the WFS features is shown below, where "`viewparams=statefp:06`" is the part of query string used to get features for the state of California (state FIP code: 06).

```
localhost:8083/geoserver/manhan/ows?service=WFS&version=2.0.0&request=GetFeature&typeName=manhan:c11_a101101_v&maxFeatures=50&outputFormat=application/json&viewparams=statefp:06
```

By changing the query string “`outputFormat=application/json`” in the above url to another format available in this service for example: “`outputFormat=text/html`”, it will return the same resources in another representation.

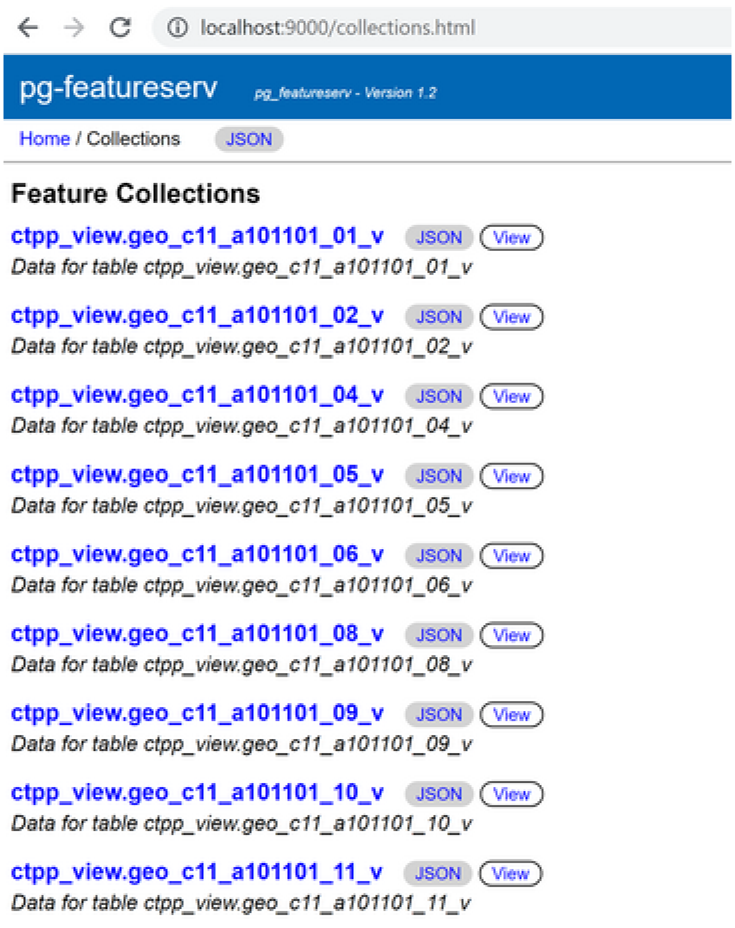
Unlike the WFS (version 2.0.0) service shown in the above URL, the OGC API Features service of GeoServer is still in heavy development and will undergo changes, though currently available as a plugin. This feature was installed and tested successfully. For our development server, the description of this service is exposed through the URL: `localhost:8083/geoserver/ogc/features`.

FIGURE 31: LANDING PAGE OF GEOSERVER OGC API FEATURES SERVICE



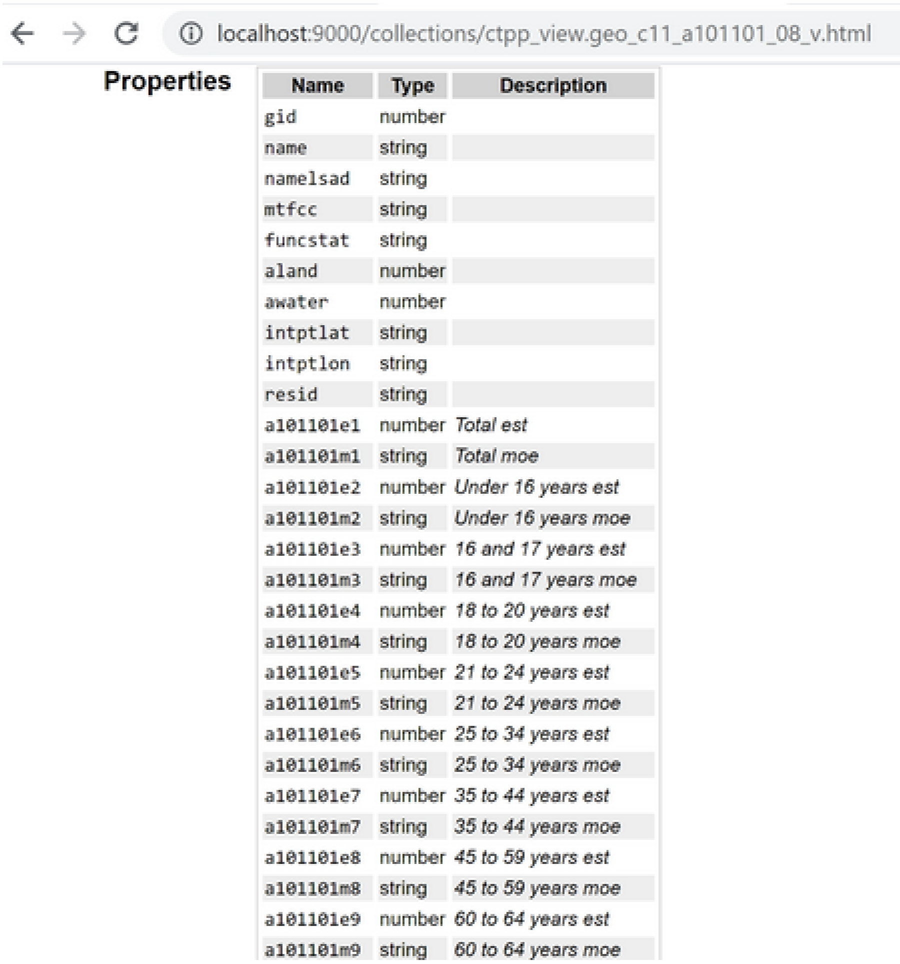
Unlike GeoServer, `pg-featureserv` automatically discovers and exposes all tables and functions it has access to. Most work needs to be done in the backend database, including the definition of layers and access privileges. In this test, the user configured for this mapping server has read-only access to the database schema `ctpp_view`. Once the service got started, the feature layers in the schema are automatically exposed in the URI endpoint shown in the Figure 32 below:

FIGURE 32: SAMPLE COLLECTIONS OF RESOURCES EXPOSED BY PG_FEATURESERV



`pg_featureserv` also automatically converts comments on tables, views, and columns generated using commands like "`COMMENT ON COLUMN ctpp_view.geo_c11_a101101_08_v.a101101e7;`" into description of web resources exposed through API as shown in Figure 33.

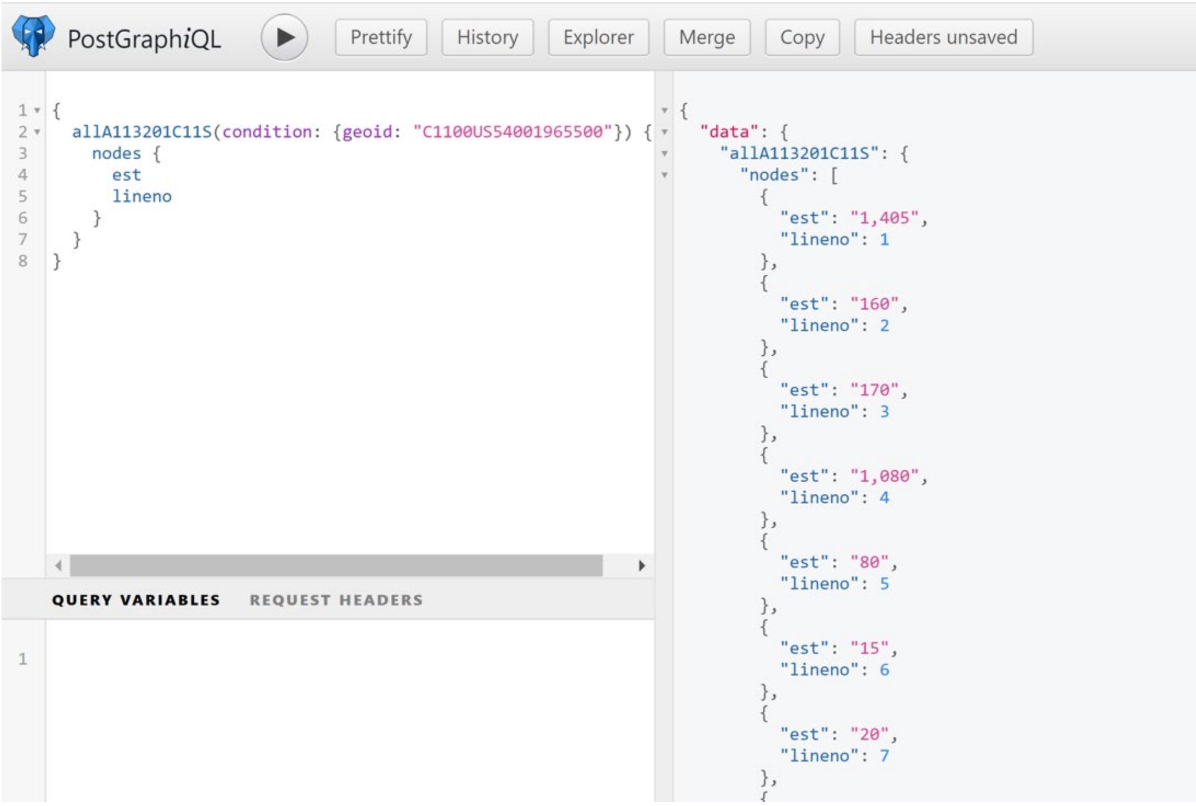
FIGURE 33: SAMPLE AUTOMATICALLY POPULATED META DATA IN PG_FEATURESERV



Name	Type	Description
gid	number	
name	string	
namelsad	string	
mtfcc	string	
funcstat	string	
aland	number	
awater	number	
intptlat	string	
intptlon	string	
resid	string	
a101101e1	number	*Total est*
a101101m1	string	*Total moe*
a101101e2	number	*Under 16 years est*
a101101m2	string	*Under 16 years moe*
a101101e3	number	*16 and 17 years est*
a101101m3	string	*16 and 17 years moe*
a101101e4	number	*18 to 20 years est*
a101101m4	string	*18 to 20 years moe*
a101101e5	number	*21 to 24 years est*
a101101m5	string	*21 to 24 years moe*
a101101e6	number	*25 to 34 years est*
a101101m6	string	*25 to 34 years moe*
a101101e7	number	*35 to 44 years est*
a101101m7	string	*35 to 44 years moe*
a101101e8	number	*45 to 59 years est*
a101101m8	string	*45 to 59 years moe*
a101101e9	number	*60 to 64 years est*
a101101m9	string	*60 to 64 years moe*

Despite all the benefits brought by the REST API, each resource endpoint exposed by REST API has predefined fixed structure, which leads to over-fetching and under-fetching problems. For example, a REST API wouldn't allow a user to pick a subset of fields from among the collection of fields shown in the above figure. GraphQL is a new API technology designed to solve this problem. There are many off-the-shelf GraphQL server software packages readily available. In this study, Postgraphile, a node.js module, is used to instantly spin up a GraphQL server by pointing it towards our testing PostgreSQL database. Due to the limited budget, GraphQL API was not extensively evaluated in this project. We will only use the simple query in Figure 34 to demonstrate that GraphQL API can be used to select and filter information a user is most interested in. Based on the query, the user in this case is only interested in getting the values for the fields: "est" and "lineno" from tabulation `A113201_C11` for records meeting the condition geoid="C1100US54001965500".

FIGURE 34: SAMPLE GRAPHQL QUERY TO SELECT FIELDS



The successful evaluation of GeoServer and `pg_featureserv` as web API/web service server in this study indicates that building on top of an existing solution might be a more cost-effective option than building a custom API from scratch. In addition, the quick evaluation of Postgraphile server software indicates that GraphQL might be useful component to include in this web API project for users interested in getting a more specific subset of data or fields than what a Rest API usually offers.

7 Conclusions

CTPP data is used in a wide variety of studies, and CTPP users are a diverse group of analysts, planners, engineers, and managers in many different kinds of organizations around the country. While most of these users currently rely upon the web-based user interface provided by AASHTO for data access (i.e. rather than downloading the data via the FTP site which is currently the only alternative option), some users complain about the web interface's usability and many say that they will use an API if developed in the future--even though their current workflow does not resemble that of a typical web services API user. Even technical "power users" acknowledge that an API, by itself, cannot satisfy the needs of all (nor even most)

current users. Furthermore, an intuitive, user-friendly, and effective web user interface is recognized as key to expanding CTPP data utilization outside of its traditional market.

A question not explicitly addressed by this outreach effort, but critical to understanding the potential future evolution of the CTPP as a platform, is whether third parties (i.e. other than AASHTO or non-technical CTPP end-users) would develop web user interfaces for specific applications and sub-markets, e.g. for a local geographic area, or for some specific type of task (e.g. travel model calibration, or housing studies, or environmental justice analysis). We did see some precedent for this in the comments of participants in our GIS-T SIG as well as the smaller focus group; for example, a SIG participant described a web-based application they had developed for bicycle/pedestrian planning that used a local extract of CTPP data in its back-end. It is furthermore encouraging to note that several "DIY" script-based interfaces to CTPP data have already been developed by third parties and published to Github, even without an API. One of these, the CTPPr package developed for R by Anthony Fucci at Westat, closely resembles an API access tool in the way it allows users to specify specific datasets and tables for specific geographies (see https://github.com/Westat-Transportation/CTPPr for more information), even if, in the background, the package relies upon the current FTP download pages, rather than any REST endpoint or similar web services API.

However, code libraries that facilitate data access using a scripting language such as R are only part of what third parties in the user community need to create in order to foster an effective platform that serves all potential users well. Ideally, packages need to be available for Python or some other general-purpose programming language which is used to develop dynamic web based applications, with clear enough documentation and examples that technical community members can begin to develop their own front-end interfaces for less technical data end-users. The back-end and API code need not necessarily be "open source" (because the whole point of a web services API is that it allows client applications to be agnostic about server-side platform choices), but ideally an enhanced and updated web user interface that is built upon a new API would be open source, with code that is commented liberally so as to serve as a template and source of examples for aspiring third-party front-end creators.

If the API is implemented in a manner that readily supports open geospatial standards, then the learning curve imposed upon end-users could be greatly reduced, if it enables a workflow that allows them to simply add the desired data as a layer in their preferred GIS. Several users have commented that, rather than trying to build a fully-featured web-based mapping application on par with ArcGIS Online, efforts to improve upon the current web-based user interface should focus on enhancing the ease and efficiency with which the desired data can be identified and accessed for further use within other toolsets. The danger in this approach is that if the web mapping capabilities of the user interface are de-emphasized too much, users who do not have other GIS or analysis capabilities will lose the ability to quickly make simple maps and summary tables. In this event, future work should include coordination with potential developers of specialized front-ends for such users.