The Data Innovation solution consists of three separate toolkits for use by Vivli Data Contributors, Data Requestors, and Vivli scientific staff. Use of these tools within the Vivli workflow allows Vivli Data Contributors to confidently share small data sets using the Vivli data sharing process.

Synthesizer

The Vivli Data Contributor (DC) runs the Synthesizer component on raw data stored in a relational database (Raw Data Set or RDS) to create a Synthetic Data Set (SDS). The Synthesizer tool allows the DC to classify each field (column) from a privacy point of view as:

1) OK, no changes required
2) Trivial masking required (the data is altered using common anonymization techniques like substitution, shuffling, statistical relations, rearrangement of numbers or dates, etc.)
3) Complex masking required (oversampling techniques are used to regenerate the data to match the statistical characteristics of the RDS)
4) Remove column, not required

After categorization, the Synthesizer produces an SDS with Category 2 columns that are anonymized, Category 3 columns that are oversampled to share the original data's statistical properties, and Column 4 columns removed. The SDS is then stored in the Vivli Secure Research Environment, accessible to the Vivli Data Requester (DR).

The Synthesizer is a custom package built with the ability to accept the categories and also to run some commonly available static data anonymization tools, such as that provided by the Dynatrace Anonymization API.

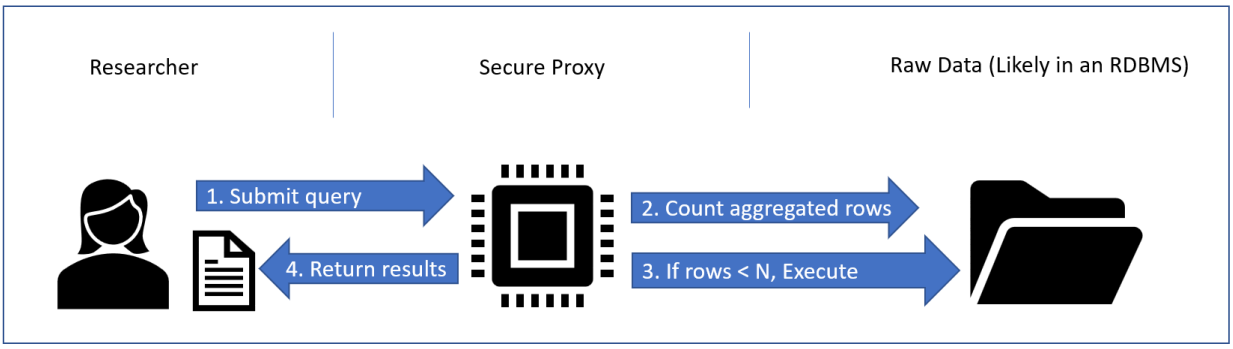
Secure Proxy for Differential Privacy and Code-To-Data Security

The secure proxy component is designed to provide user access to a raw (unmasked, but anonymized) dataset in a controlled manner, by providing two mechanisms for querying the dataset which will restrict egress of PII data, described below.

This description assumes that the raw data is loaded into a relational database supporting the SQL protocol, but these techniques can be applied to other platforms.

Differential Privacy Mechanism

The first mechanism allows real-time querying of the raw dataset via a differential privacy query interface. This interface will allow a user to query aggregates (sums, averages, slopes and intercepts) as long as they are aggregated over a minimum number of individual rows of data. This will allow researchers to perform basic research against the raw dataset while developing a more sophisticated script to be executed against the raw data directly through the Code-To-Data mechanism below.



There are several mechanisms to accomplish this, a couple examples of which we will briefly describe:

1. Provide direct access to the SQL DB hosting the data, but limit read access to stored procedures which only execute aggregate operations over minimum rows:

   Example: exec spSelect(EXPRESSION, FROMWHERE)

   Exec spSelect 'AVG(MEASUREMENT)', 'MEASUREMENTS WHERE TYPE = 'BP''

   The stored procedure could then execute: Select count(*) from MEASUREMENTS WHERE TYPE='BP' to determine if the aggregate would be over more than N rows, before executing the final SELECT statement and returning the result.

2. Provide a basic query mechanism via a custom proxy service that executes incoming queries against the raw data but reviews the query execution plan output by the database to ensure that results are aggregated over a minimum number of rows.

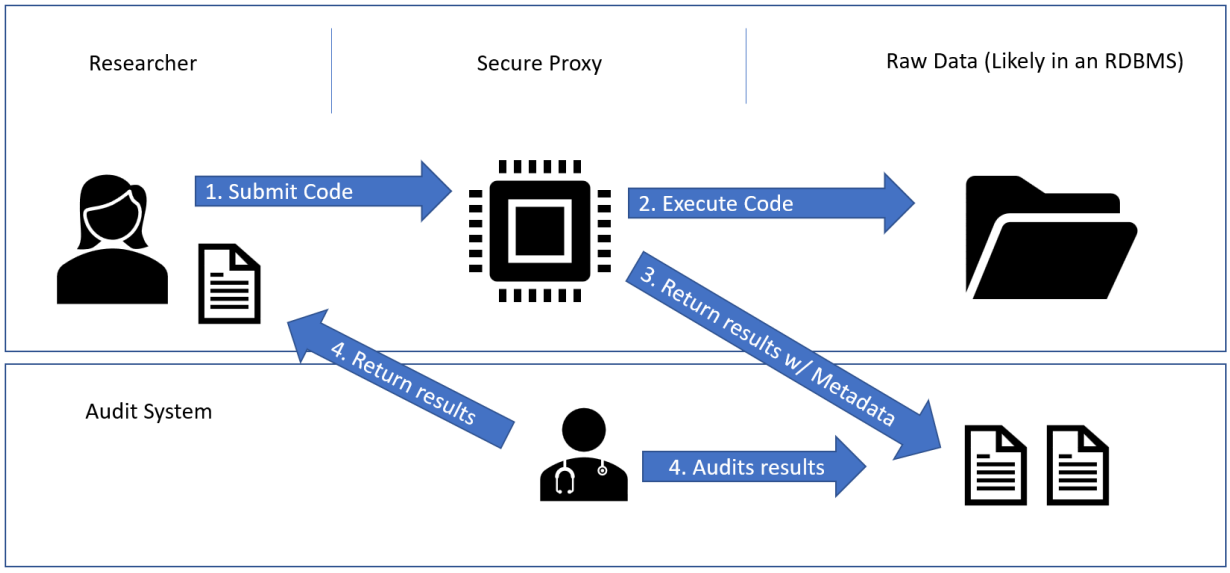
3. Provide a basic query mechanism via a custom proxy service that transforms incoming queries against the raw data to return the counts that would be used in generating aggregates before executing the original query and returning the results.

Some basic research and proofs of concept are required before a particular approach can be responsibly selected.

Code-To-Data Security

The second mechanism allows for submission of code to be executed against the raw dataset in using a code-to-data paradigm. A researcher may create code to run against the raw data by developing the code (e.g. a SQL Script) against a synthetic dataset of the same format, and then submit that code for execution against the real, raw dataset. The Secure Proxy would execute that code and return the results (and associated execution metadata such as query execution plans) in secure fashion to the Auditor system for audit. Once the Auditor system has determined that the results do not violate privacy, it will release them to the researcher, either directly via email or download, or by delivering to a secure computing environment.

By delivering the code to the data, the researcher never has direct access to the data, but can still execute an analysis against it. By providing a synthesized dataset, we allow the researcher to code against representative data, making the code-to-data approach far more practical. Without the synthesized data, it would be impossible to debug any analytic code before submission.



Auditor

This toolkit has two components

1. Database Query Auditor
    a. Query monitor: Logs queries to the real database
    b. Call stack authenticator: Reviews queries to sensitive information
2. Results Auditor

. Automated tools: Detect schemes to leak sensitive information in results

A Vivli staff member with expert scientific background evaluates the automated results, and does a final review, paying extra attention to flagged items.

Database Query Auditor

The Database Query Auditor authenticates queries to the real data stored in the secure proxy. It ensures that queries to individual rows or to PII are made by an authenticated source. The rationale is that some queries are performed to make valid aggregate calculations or modeling, while some are performed by attackers seeking to obtain individual records or PII. The former should be permitted, whereas the latter should be detected and prohibited. To perform this authentication, we use a Query Monitor and a Call Stack Authenticator.

The Query Monitor observes and logs queries to the real database. This log tracks the data accessed, whether this data is sensitive, and the source of the query. This log is then reviewed by the Call Stack Authenticator.

The Call Stack Authenticator reviews the source of the queries to sensitive data and determines if these are coming from authenticated sources. An authenticated source is a piece of code that will output or return only aggregate information (averages if applicable, models). An authenticated source might be a machine learning algorithm that needs to access individual rows. Any code outside of authenticated libraries is not authenticated, and any sensitive queries are flagged for review by the Vivli staff expert reviewer.

Results Auditor

The Results Auditor reviews the outputs of the submitted code. It consists of automated tools and a human auditor. The results, if approved by the auditor, are sent back to the researcher.

The results auditor uses a suite of automated tools to detect sensitive information leakage. Each tool is formulated to prevent against specific attacks. Some examples of automated tools are as follows:

- Raw value matching detection. Since the auditor has access to the real dataset, it is easy to detect matches between data in the output and data in the real dataset. Any such matches will be flagged for review.
- Steganography detection. It is possible to embed information in an unassuming format. Existing techniques for steganography detection use compression to detect if an image, for example, contains more information or entropy than expected.
- Frequency analysis. Attackers might use a bijection to mask sensitive data and evade detection.

Expert Review

Finally, the results are reviewed by a scientific expert from Vivli. The Vivli reviewer takes a final look at the code submitted and the results before approving them to be sent to the researcher. Anything flagged for review by the automated tools or query auditor is brought to the attention of the human auditor. The human auditor sends approved results back to the researcher.

Further development:

- Decision Support AI: Results and approvals can be monitored by an artificial intelligence. By observing examples of results and their approval status, it may be possible to develop a model that can predict potentially problematic results