

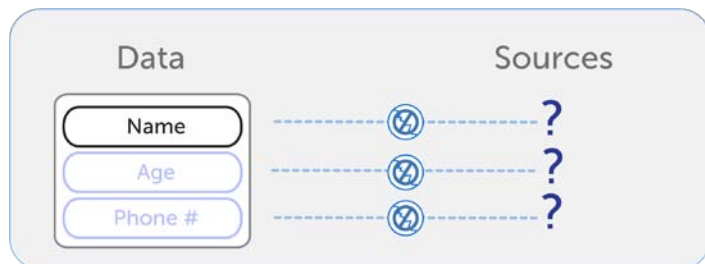
## Hard Technical Problems in Civil Liberties Protection

Protecting civil liberties has become an increasingly important problem for intelligence and law enforcement. However, protecting civil liberties requires solving a number of hard technical problems that invalidate most existing approaches to storing and using data in the intelligence community. Specifically, the fundamental issue in protecting civil liberties is that different data has different requirements in terms of privacy and access: law enforcement agencies face this challenge in distinguishing intelligence data from data that can be used as evidence; intelligence agencies face this challenge because of the many additional protections that must be applied to data on U.S. citizens as opposed to data on foreign citizens. This poses challenges in terms of representing the data, viewing and searching this data, and auditing the products of analysis to determine that all data was used lawfully.

Palantir has developed new technologies and a rigorous framework to: protect privacy and civil liberties; empower policymakers and administrators to enforce legal, regulatory, and policy requirements; and, equally important, ensure that the implementation of all requirements is audited. Some of our breakthrough technologies which protect privacy and civil liberties include Palantir's **Access Control Model**, **Revisioning Database** and **Immutable Audit Logs**:

### **Access Control Model and Revisioning Database**

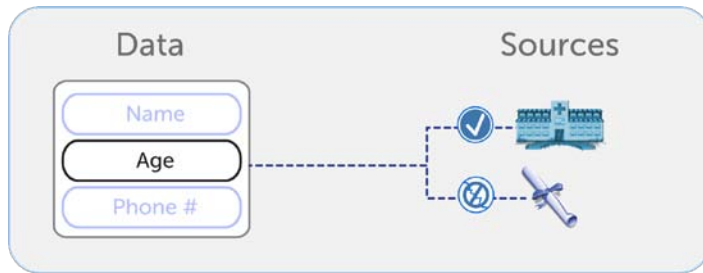
The first and most fundamental problem when integrating data from sources at different levels of protection is simply how one represents the data along with the level of protection to apply. Most existing approaches simply copy the data into a new format, while losing the sourcing information. For instance, consider a typical schema: a table where each row represents a person, with columns for name, address, and phone number. In this schema, there is simply no place to store the source of each of these pieces of information. Without that source, there is no way to determine at what level to protect the data.



*A standard database schema will not capture the sourcing of data at all.*

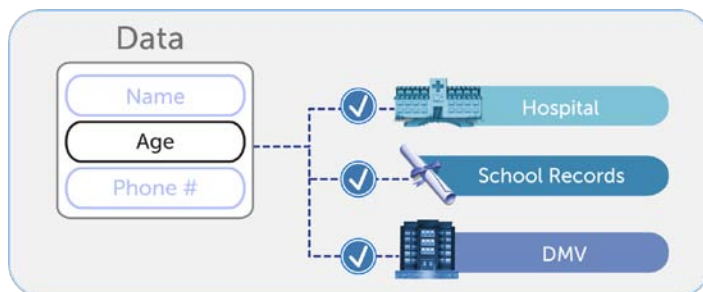
This is not just an issue with products that perform data integration between several databases. In the intelligence community, one of the most important aspects of an analyst's job is to read cable traffic and structure it into a form that can be presented to others. With current software products, the typical approach used is to simply copy and paste the information from a source document into the software analysis environment. There is thus no connection back to the source document. One common work-around is to simply attach every document from which information was gained to each object of analysis; while this allows some connection between the underlying document and the structured information, there is no way to tell which attributes of the object came from which document, or from which part of the document - which means that granular information protection is impossible using this approach.

In order to store sourcing, software must be designed from the ground up to do so. However, naïve approaches to maintaining sourcing may stumble upon a second issue: each attribute may come from multiple sources. A less naïve product that attempts to maintain sourcing might maintain two database tables: one for the person, and another for each of his properties. The property table can then include a column that specifies what source the property came from. However, a phone number will frequently be attached to a person from multiple sources, some of which may need to be protected and some of which may not.



*Simple attempts to capture sourcing fail when a particular property comes from multiple sources that require different levels of protection.*

Palantir solves this problem using its **Revisioning Database** by recording rows that connect each attribute of each object back to each of the data sources that created it. This involves three tables: a table for the *Object* (a person), a table for the *Property* (a phone number), and a table for the *Data Source Record* (what row in a database or what text offset in a document was used to generate this phone number). From the set of all Data Source Records, Palantir’s **Access Control Model** can determine the appropriate access controls to be applied to each property of an object.



*The Palantir Revisioning Database tracks all information back to every source from which it was derived, allowing the Access Control Model to provide complete protection of private data.*

Defining the representation is only part of the problem, however – we still need to define how that representation is used. In this case, the two most important uses are viewing and searching for objects. Viewing an object requires looking at each data source (whether a database row or a document) in order to determine what part of the object is visible to an analyst subject to particular privacy protections. Naïvely accessing this through the database is impractical due to its extreme demands on system resources, as each view of an object could involve traversing hundreds of rows joined across three different tables. Palantir solves this problem through a combination of strategic denormalization and aggressive memory caching, providing a scalable way to support quick and frequent object viewing by hundreds or thousands of analysts.

Searching against the objects in the system also poses difficult technical challenges because the search engine must be privacy-aware. A search engine returns a list of results sorted by the relevance to the original query. Computing this relevance score requires iterating over all the fields of the object. In order to ensure private information is not leaked, the search engine itself must be able to determine which fields are accessible to the requesting analyst and compute relevance only based on those fields. No off-the-shelf search servers allow different fields for objects to be at different access control levels. Palantir’s search server, on the other hand, has been built to be security-aware in order to make civil liberties protection possible.

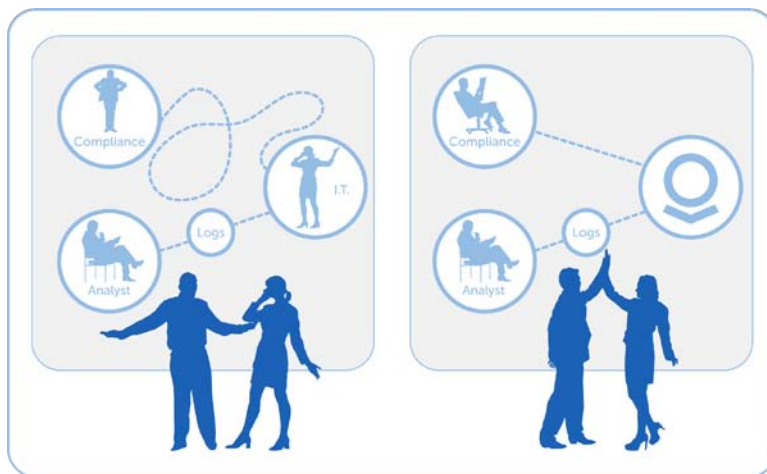
### **Immutable Audit Logs**

A system that protects civil liberties must provide a simple auditing capability to verify that information has not been used improperly in the course of creating an analysis product. Moreover, this audit must be accessible to non-technical users who may not be part of the organization using the data, such as an organization-wide compliance officer. This auditing information must include sourcing information, as described above. However, any system that allows analysts to combine data and create products from it must allow analysts to edit the data in order to correct errors, make formatting changes, or address differences in terminology between different data sets. For example, consider the task of reconciling phone numbers or addresses from

multiple data sets. In order to audit this data, therefore, we must be able not only to track it back to its original source but also to see any transformations that have been applied to the data.

The standard approach to providing this functionality is to write all changes by analysts to an audit log, which is stored separately from the data and accessible to IT staff. Unfortunately, this approach means that auditing can only be performed by administrators who have technical training and a strong knowledge of the system in question. In practice, this means that audits can only occur after the fact, as part of an investigation, and they can only be performed by the technical staff of the organization under investigation.

Taking the opposite approach, Palantir's Revisioning Database stores data in a way that tracks all edits to the data in an **Immutable Audit Log**, and makes them accessible with appropriate protections to a non-technical audience as part of its normal user interface. This means that analysts can pro-actively verify that their products comply with civil liberties protections as a vetting process before information is released. It also means that this compliance can be articulated and demonstrated to non-technical third parties. Thus verification can occur in an ongoing manner, and, should an investigation occur, anyone with the access to see the data can independently verify compliance with civil liberties protections.



*Most systems store logs in a format accessible only to technical insiders. A compliance officer must make a request to the IT staff of the organization being monitored in order to verify that information is being used properly. Palantir instead makes its Immutable Audit Logs available as part of its easy-to-use interface, which is designed for non-technical analysts. A compliance officer could use this interface directly to verify that civil liberties are being protected.*

In fact, Palantir is the *only* product that has been designed and built from the ground up to enforce civil liberties protections on integrated data sets. It is also a product that has been deployed across the intelligence community in challenging situations ranging from small forward analysis teams to enterprise-wide data integration, usually at a fraction of the cost of a custom-built solution. Palantir's civil liberties protection is not just a requirements document or a services project – it is built into the technical foundation of every deployment.