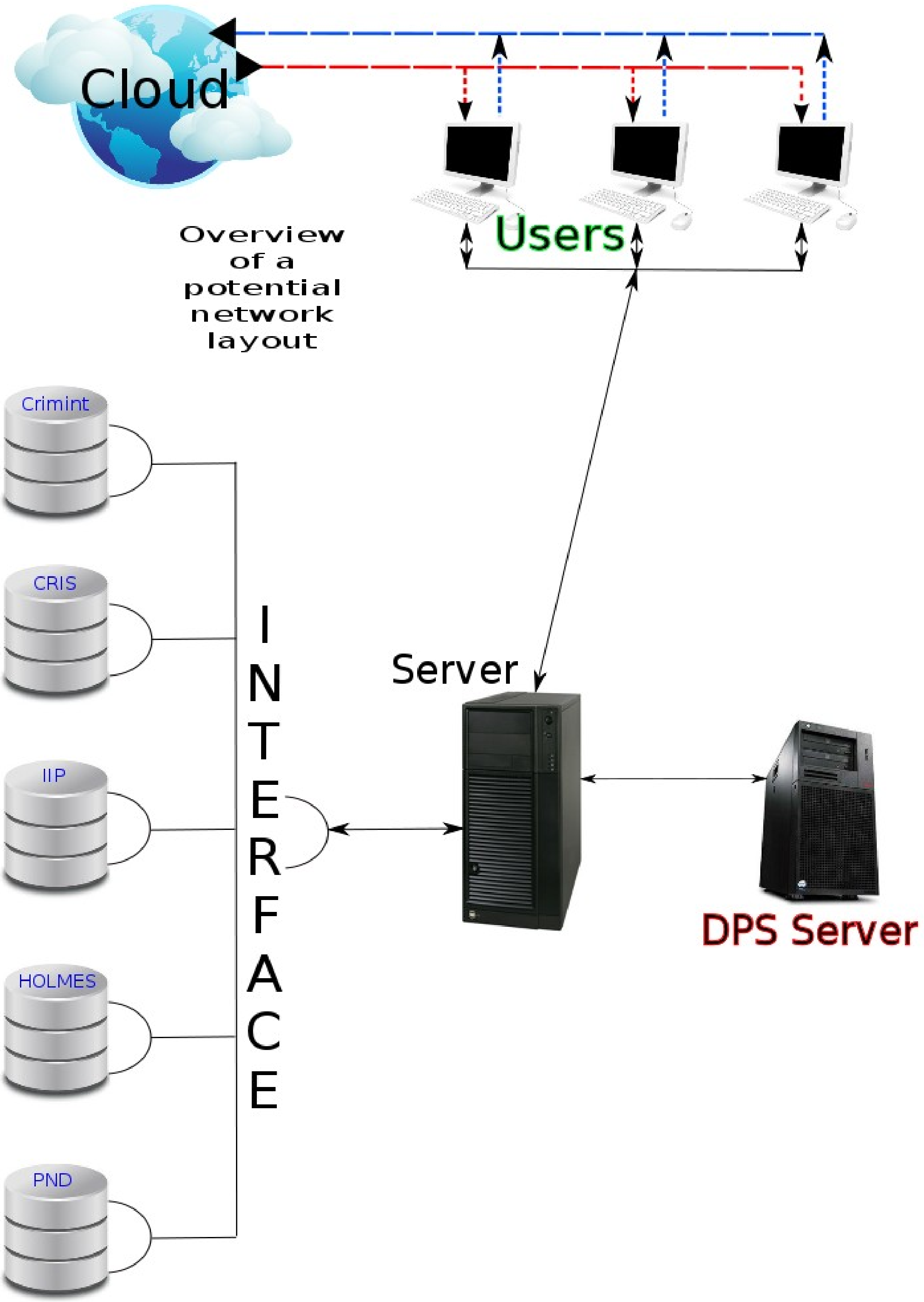
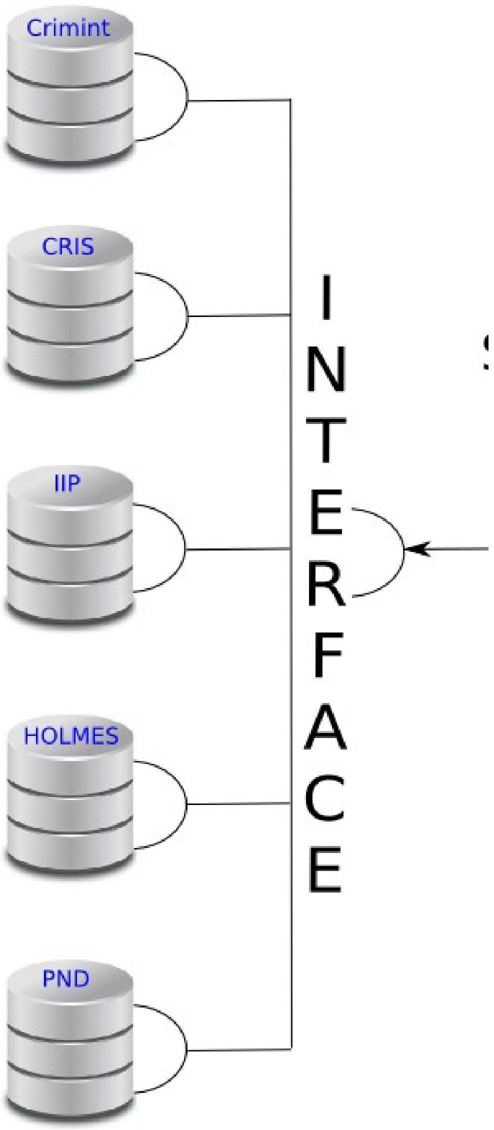
Big Data Searching

The following is prospective overview of a passive data searching system. By *Passive Data Searching*, I mean a system which can search for data but cannot input or change existing data. I will attempt to explain each area in juxtaposition to other methods.

Overview

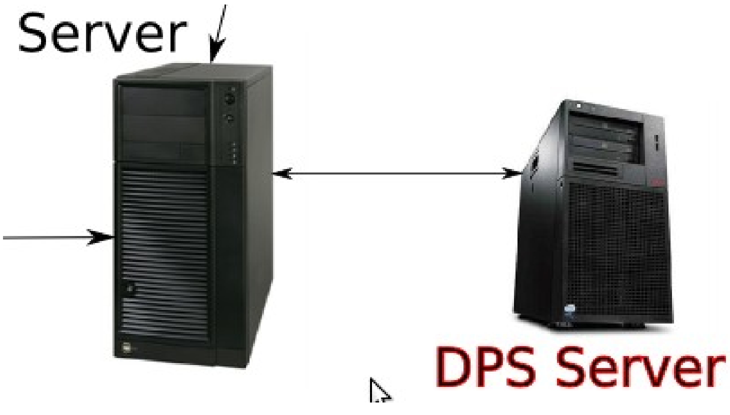


The previous diagram can be divided into four areas. The first area expresses that the current MPS database servers could be connected to via an interface:



This has the advantage of leaving the data in-situ without the expensive overheads of trying to copy all the data into a new database with associated time and hardware costs. This method would also allow for databases to be added gradually thus making the system more modular. NB. To input new data (or change existing data) the user would have to logon to the individual database.

The second area describes the processing and audit system:



The Server takes the search request from the User and queries in parallel the connected databases (known as *Federated searching*). The results are then returned to the Requester detailing the name of the database and the results e.g. CRIS

Result

Holmes

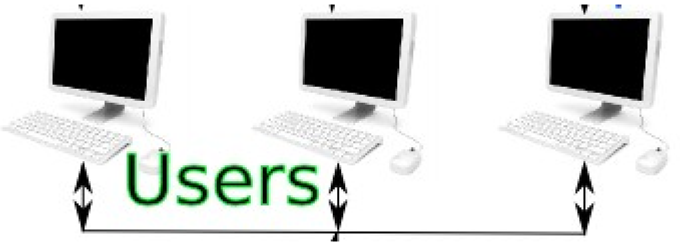
Result …..............etc

The DPS Server takes an audit of the searches. For example, *User, Date/Time, Search terms*. The DPS Server would not need to record the results of the search as this could be replicated any time.

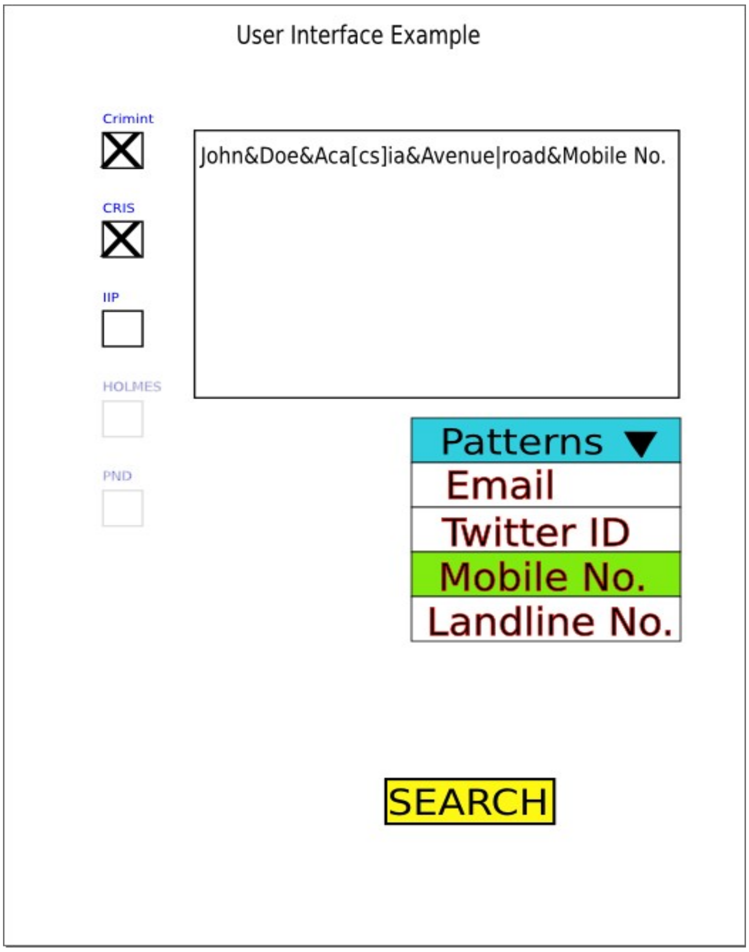
There are many configurations for the Server in terms of searching. One method is to hold common searches temporarily in memory to reduce access times. For example, the majority of searches may be against data recorded within the last 6 months.

Another method to consider is to use Apache Hadoop. This is the de facto method for handling large data requests over multifarious databases. However Hadoop requires that all the data from each database is loaded into the Hadoop database (i.e. Hadoop does not allow for Federated Searching). Once the data is in the Hadoop database, the search is very fast and allows for numerous search methods and the ability to express the results in various formats. Hadoop can handle both structured and unstructured data regardless of its native format i.e. email content, photographs audio files etc. Problems occur when deciding the time intervals that Hadoop updates itself from the original database sources. For example, a Holmes record could be updated after Hadoop had updated itself with the potential for a search result to be missed. However the risk could be reduced through reduced synchronisation times.

The third section relates to the UI (User Interface) and the User experience:



The UI should be simple and allow the user to query multiple databases. The system should be intuitive to allow the User to conduct searches with little to no training. The UI should also have the ability to connect to public sources of information held in the internet/cloud. All in all, the user requires a 'one stop shop' for all their Intel research.


User Interface Example

Crimint ☒

CRIS ☒

IIP □

HOLMES □

PND □

John&Doe&Aca[cs]ia&Avenue|road&Mobile No.