GeoLens

Jared Koontz

2/24/2015

Outline

1. Introduction
2. Background
3. Research Issues in Visual Analytics
4. Aggregation Components in GeoLens
    1. Autonomous Histogram Creation
    2. Geohash Based Self-Adjustable Data Tiles
5. Data Flow
6. Demo

Introduction

- This will be a presentation of the research I completed during my graduate semesters.
  - Some of you will be very familiar with the contents of this presentation
  - Others will be hearing this information for the first time

- My defense is scheduled in the next month
  - So this is a bit of practice for me
  - As well as bring the rest of you up to date

Introduction

- The topic of my research is visual data analytics
  - Geospatial time-series data with additional features
    - Data with
      - Timestamp
      - Latitude longitude Pair
      - As well as features about the earth:
        - Wind Speed, Temperature, etc.
  - How can we visualize this data?
    - Where is 39° N, 105° W?

Background

- I will be presenting the visual analytics engine for Galileo entitled GeoLens.
  - An understanding of various aspects of Galileo is crucial for understanding GeoLens.
- There are aspects of Galileo that GeoLens relies on, that I do not have time to discuss
- However, a knowledge of what a geohash is essential to GeoLens

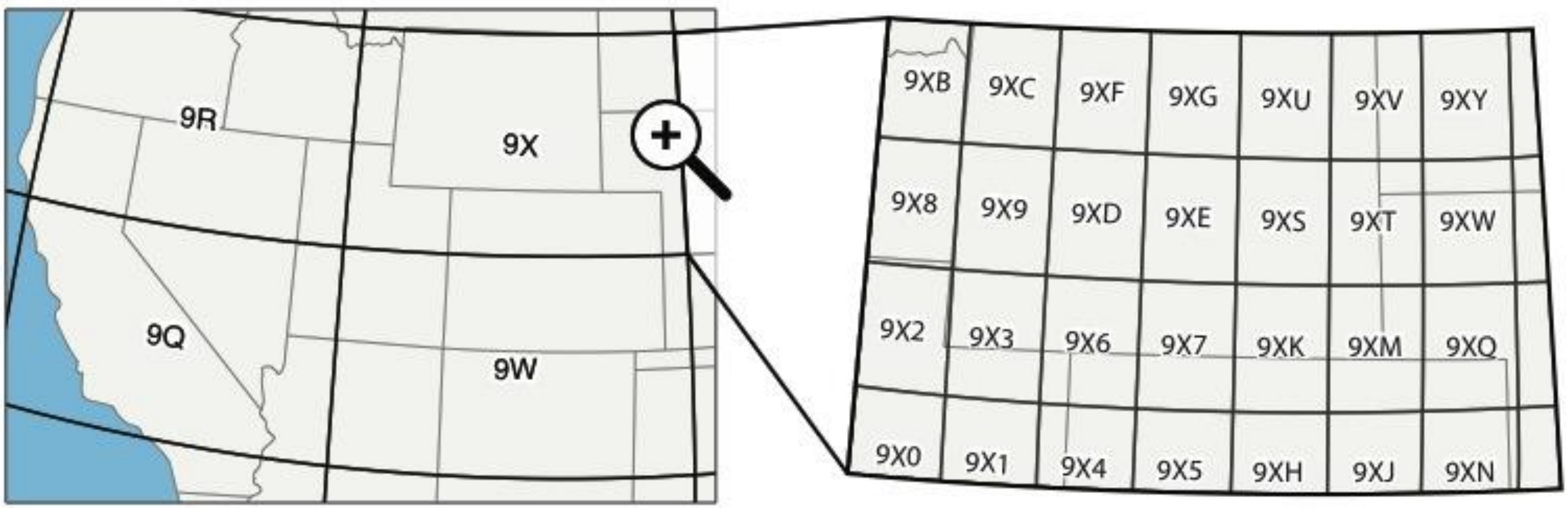
Background

- Galileo is a storage system that uses a geohash for
  - Preserving geospatial information in the data
  - Achieving data dispersion
- What is a geohash?
  - The geohash algorithm divides the Earth into a hierarchy of bounding boxes
  - These boxes are represented by strings:

    9XJQBFF45TN9 → The Oval at CSU

Background

- Coarser-grained groups can be achieved by decreasing the resolution of the hashes
  - 9XJQ = 20x30 km rectangle
  - 9X = 600x1000 km rectangle



The Main Issues In Visualization Research

- Visualizations are inherently multi-scale
- Perceptual scalability
- Interactive scalability
- Read-only back-end visualization structures

The Main Issues In Visualization Research

- Visualizations are inherently multi-scale
- The same data structure should support
  - both large and small queries
  - coarse and fine grain queries

- For example this data structure needs the ability to:
  - Allow the user quick indexing over multiple years of a time index
  - As well as the ability the drill down to an individual hour or day.
  - Should be able to handle a query spanning many continents,
  - As well as a query only covering a few blocks.

The Main Issues In Visualization Research

- Another research issue in visualization research is perceptual scalability
- plotting every data point in a large data set can overwhelm a user's perceptual capabilities
  - there is so much noise on the screen
- Larger, higher resolution displays can be used to increase the scalability of information visualizations.
  - However, not everyone has access to these expensive displays
- We need to control how much data is displayed on the screen.

The Main Issues In Visualization Research

- Another research issue in visualization research is interactive scalability
- moving large data sets and preparing them can lead to high latency.
- This is due to
  - I/O
  - Efficient query evaluation
  - Effective data transfer
- These latencies hinder how interactive the program is
  - interactivity is essential with effective visualization.

The Main Issues In Visualization Research

- Galileo can store data streams
  - Many visualization systems do not need to do deal with frequently updated datasets.
  - The first step in many visualization systems is to transform the raw data into a queryable data-structure
    - This step is not cheap
- For example, a visualization system entitled Nanocubes creates a nanocube for querying the data system
  - For a dataset with 1 billion points, this process takes over three hours
    - A new nanocube needs to be created each time an update occurs
- We can not afford this costly pre-processing.

The Main Issues In Visualization Research

- A crucial aspect to effectiveness of a visualization is brushing and linking
- The idea of brushing and linking is to *combine* different visualization methods.
  - Overcome the shortcomings of single techniques.
- Interactive changes made in one visualization are automatically reflected in the other visualizations.
- Brushing – The act of selecting some subset of the data
- Linking – Showing those selected points in all visualizations.

Brushing And Linking

- Example:
  - Two-part display
    - a histogram
    - a list of document titles
  - The histogram could show how many documents were published each month
  - An example of brushing and linking would be allowing the user to assign a color to one bar of the histogram
  - All the titles in the list display that were published during the chosen month will also be highlighted in that color.

Aggregation Components

- To deal with both perceptual and interactive scalability, we created two data structures for aggregation
  - Geohash Based Self-Adjustable Data Tiles
  - Autonomous Histogram Creation
- We need some sort of data reduction
  - We can not visualize all the data, there may not be enough pixels on the screen
  - The cost of moving the entire data set is too large
- We choose aggregation as our data reduction technique
  - There is not enough time to delve into the reasons of why we choose aggregation

Geohash Based Self-Adjustable Data Tiles

- To support geospatial aggregation we created the geohash based self-adjustable data tiles
  - utilize the geohash information held by nodes

- These self-adjust their resolution based on the geographic size of the query
  - Large area → small geohash string length
  - Small area → large geohash string length

Geohash Based Self-Adjustable Data Tiles

- When a query is submitted into a node in Galileo, the geospatial area in the form of a polygon
  - If this polygon covers more than one geohash area rectangle at the current resolution
    - the polygon is split into different polygons for evaluation at different nodes

- Visualize all of the geohashes that are
  - inside of this polygon
  - are two characters (configurable value) longer than the current global resolution

Geohash Based Self-Adjustable Data Tiles

- Examples:
  - Query area – United States
    - "no enclosing geohash"
    - Use two character geohash boxes
  - Query area – Northern Colorado
    - Enclosing hash = two characters
    - Use four character geohash boxes

Geohash Based Self-Adjustable Data Tiles

- In this way we have a bounded size on the geospatial aspect of our geospatial visualization
  - It is bounded in the max amount of geohash tiles we create
  - guaranteed to give us enough geohash boxes

- Without overloading the user's ability to perceive it.

Geohash Based Self-Adjustable Data Tiles

- GeoLens creates a dictionary with geohash values
  - populates it with values from the data that are inside of this box.

- This is done by averaging the values in this box
  - reporting this average as the value for this geohash area.

Autonomous Histogram Creation

- The autonomous histogram generation aggregates the frequency of data occurrence
  - provides a quick sketch of the values per feature

- This is a snapshot of the entire data-set, where the geohash boxes is a snapshot of the entire data-sets.
  - Allows for brushing and linking between the two data structures.

Autonomous Histogram Creation

- Histogram creation is a trivial task.
  - We just need to specify a set of uniform width bins
    - from the minimum value to the maximum value
  - Essentially, all we is a bin-width

- There are a variety of ways we could obtain a bin width with which to aggregate data
  - A simple way would be to prompt the user

Autonomous Histogram Creation

- This would not be ideal because different features will have different widths
  - the user might not know a good width for all of these features

- The same features, but in different areas, could require different bin sizes
  - For example:
  - an area that regularly experiences cold climates
    - might not need the same guidelines as an area that experiences warm weather

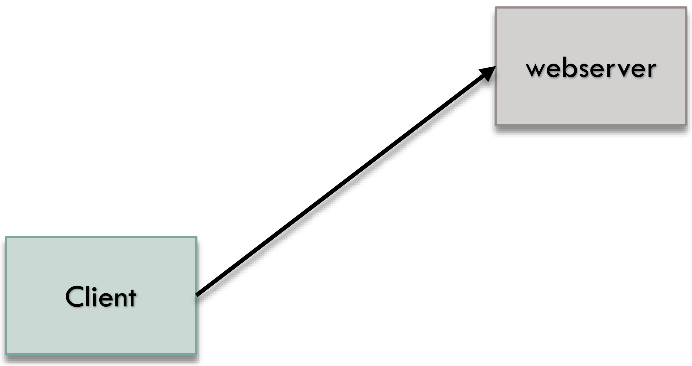
Autonomous Histogram Creation

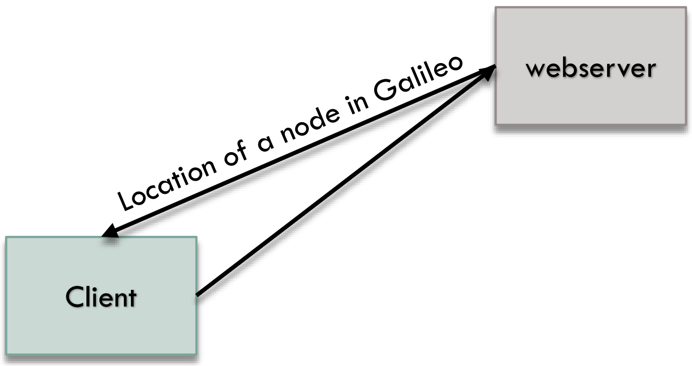
- In addition the same area might have different optimum bin widths, depending on the time of year.
- We do not leave it up to the user to supply our system with a value
- We derive a bin width based on the data.
  - “Autonomous Creation”
- We research which autonomous creation algorithm would be suit our use case
  - But once again, there is not enough time to go into details.

Data Flow

- The two components that make up the VisGraphs are the histograms and image tiles.
  - Which ones should the system create?
  - For what features?

- This will be shown in the data flow for the whole system.


webserver
Client


webserver
Location of a node in Galileo
Client

Receiving Query Input From the User

- We are faced with the need for geocoding
  - the process of translating a human readable name location on the earth
- There are many geolocation services available such as geonames
  - However, this is an online database
    - we do not want to rely on someone's network we have no control over

Receiving Query Input From the User

- There are some offline solutions such as the NGA earth-info
  - large files and require processing
  - Additionally, the output of these processes is a latitude and longitude pairs, denoting the center point of this area

- We would need additional software to then find the series of latitude and longitude coordinates that bound this area.

Receiving Query Input From the User

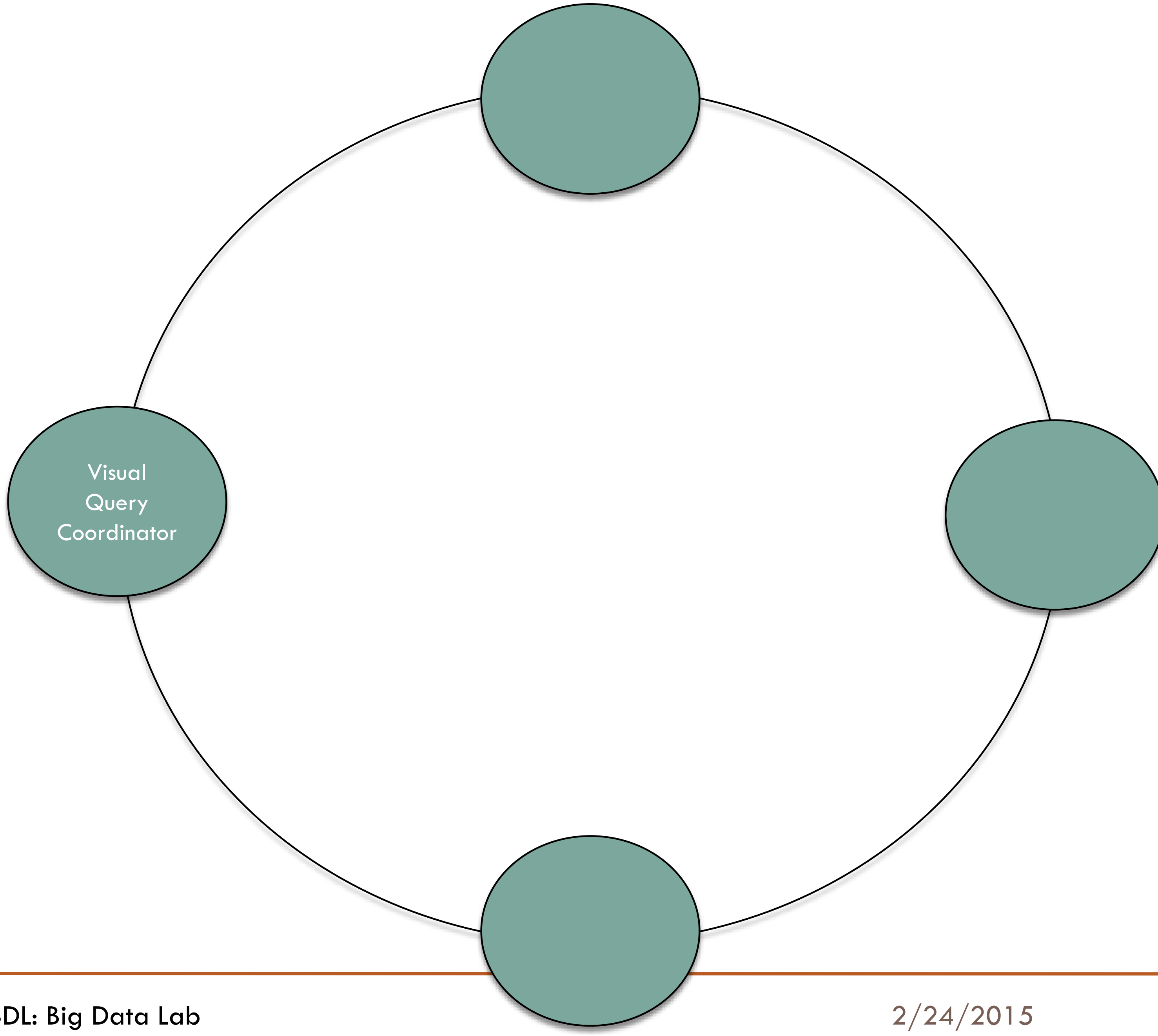
- To overcome this problem, and to receive a query area, we allow the user to draw their area they want directly on the map.

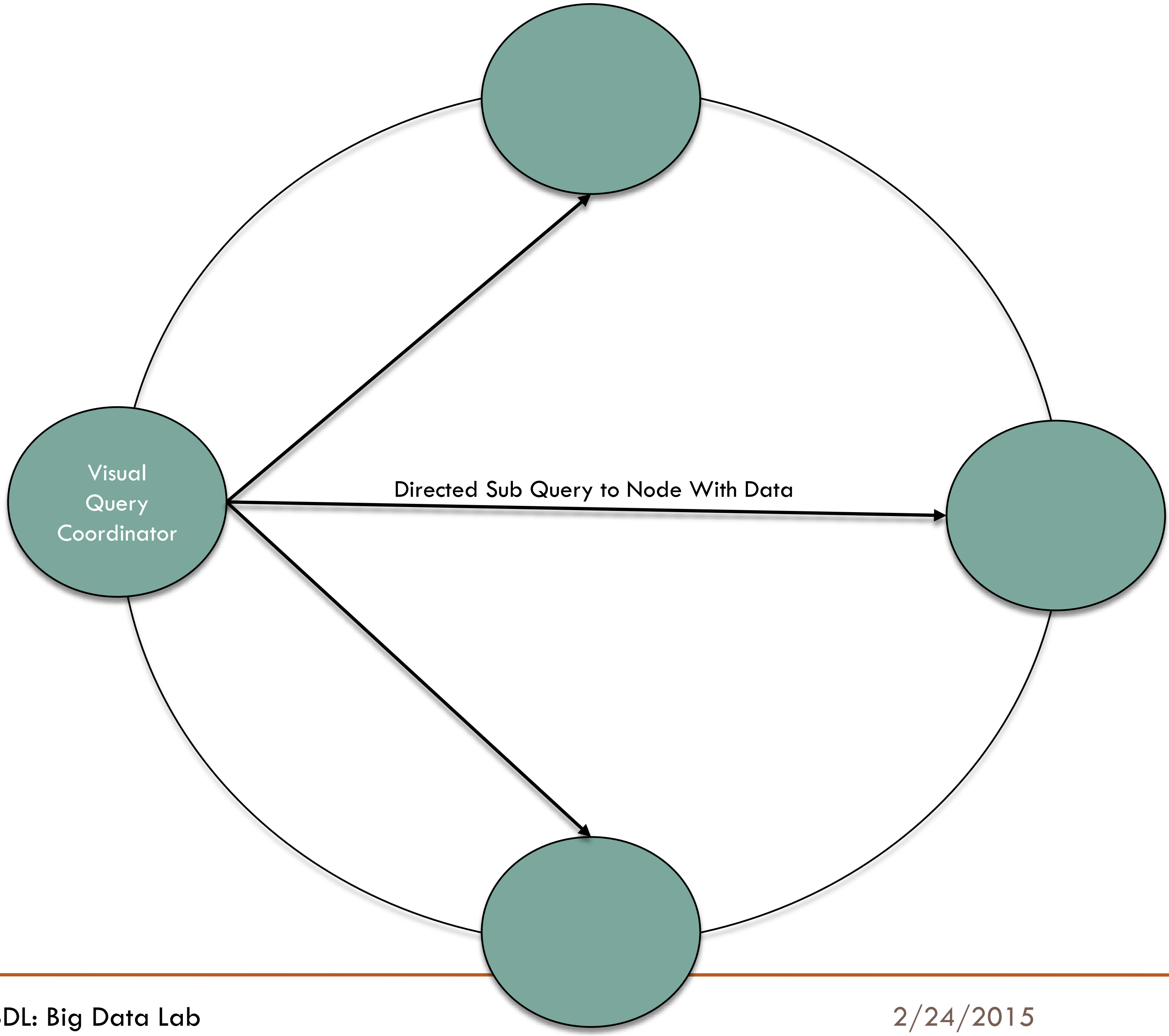
Receiving Query Input From the User

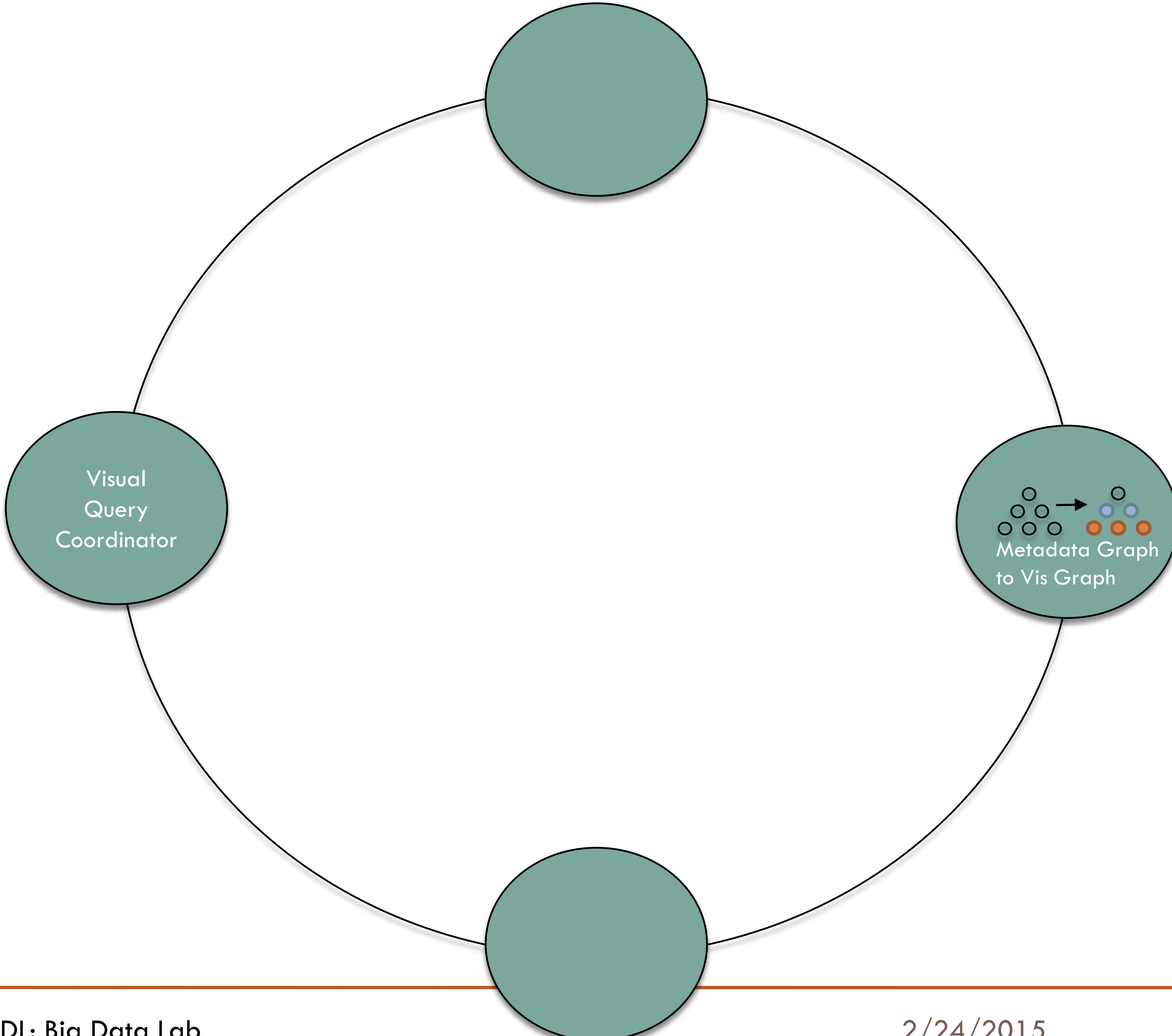
- GeoLens can also save polygons users have submitted so they do not need to be drawn again.
- The features a user is interested should be included in the visual query
- Galileo provides all of the features it is currently indexing
- the user can select all of the features they would like to examine

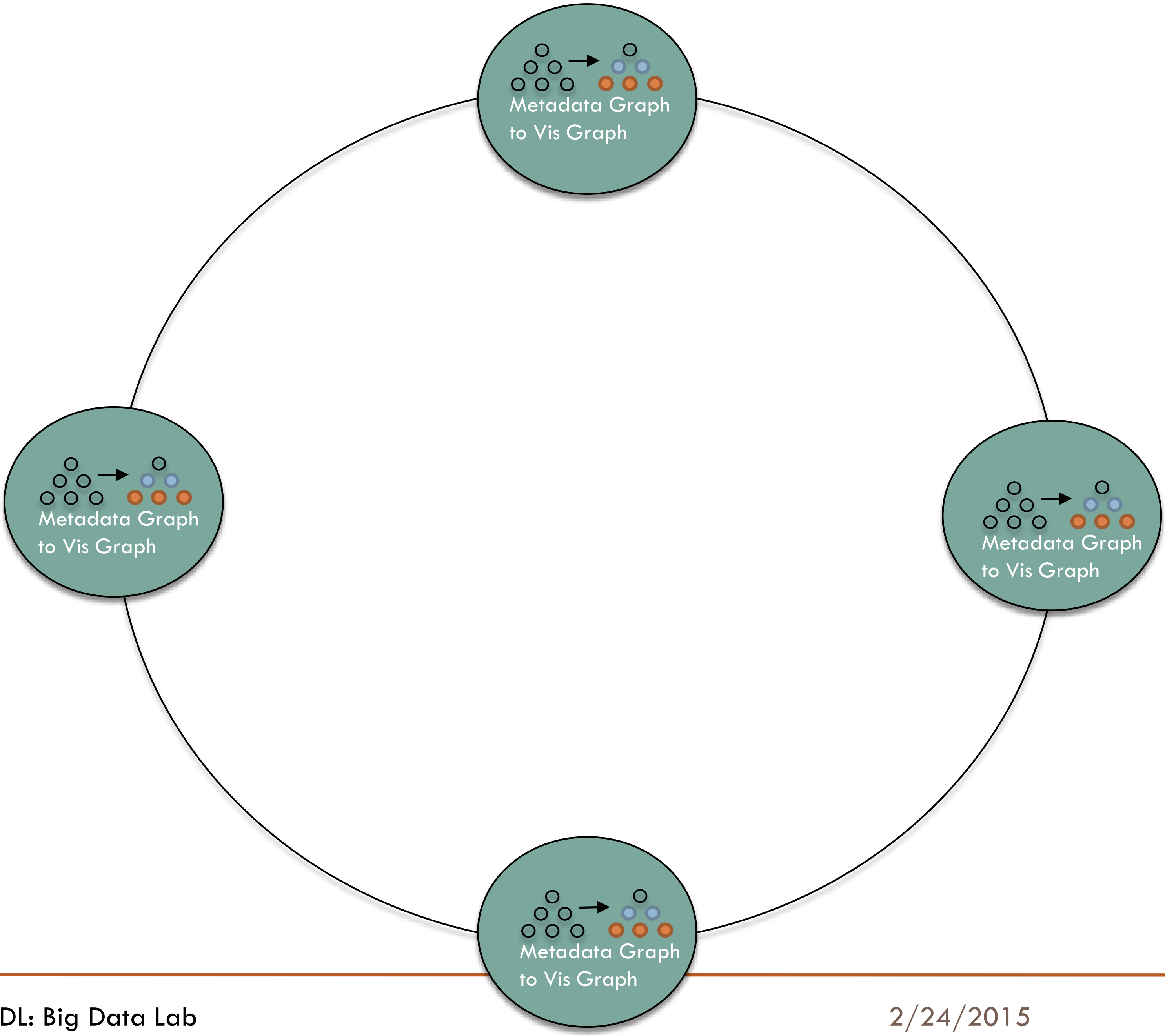

Client
Visual Query Result: Wanted Features, Time, And Location
Galileo

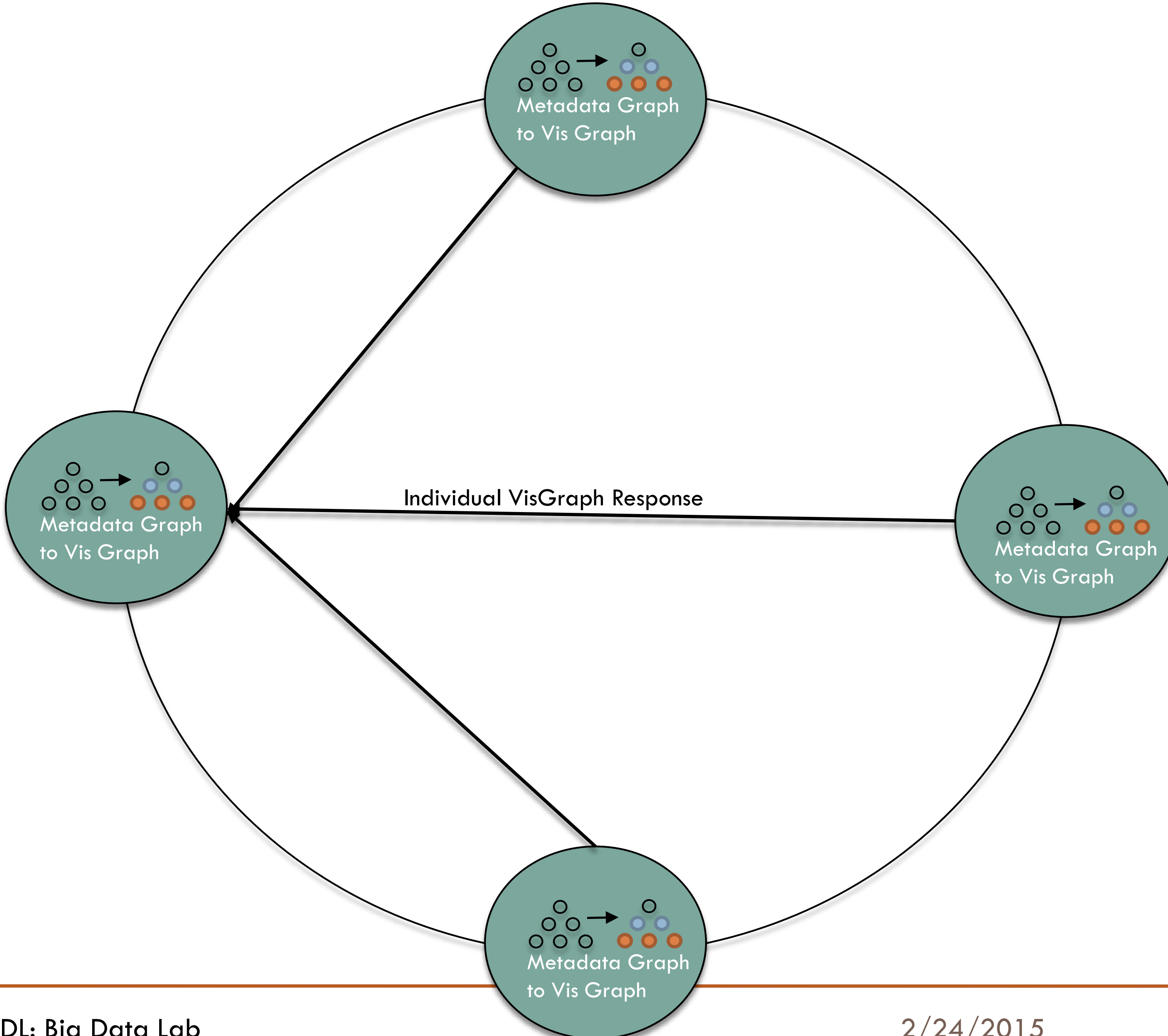
Within Galileo Group

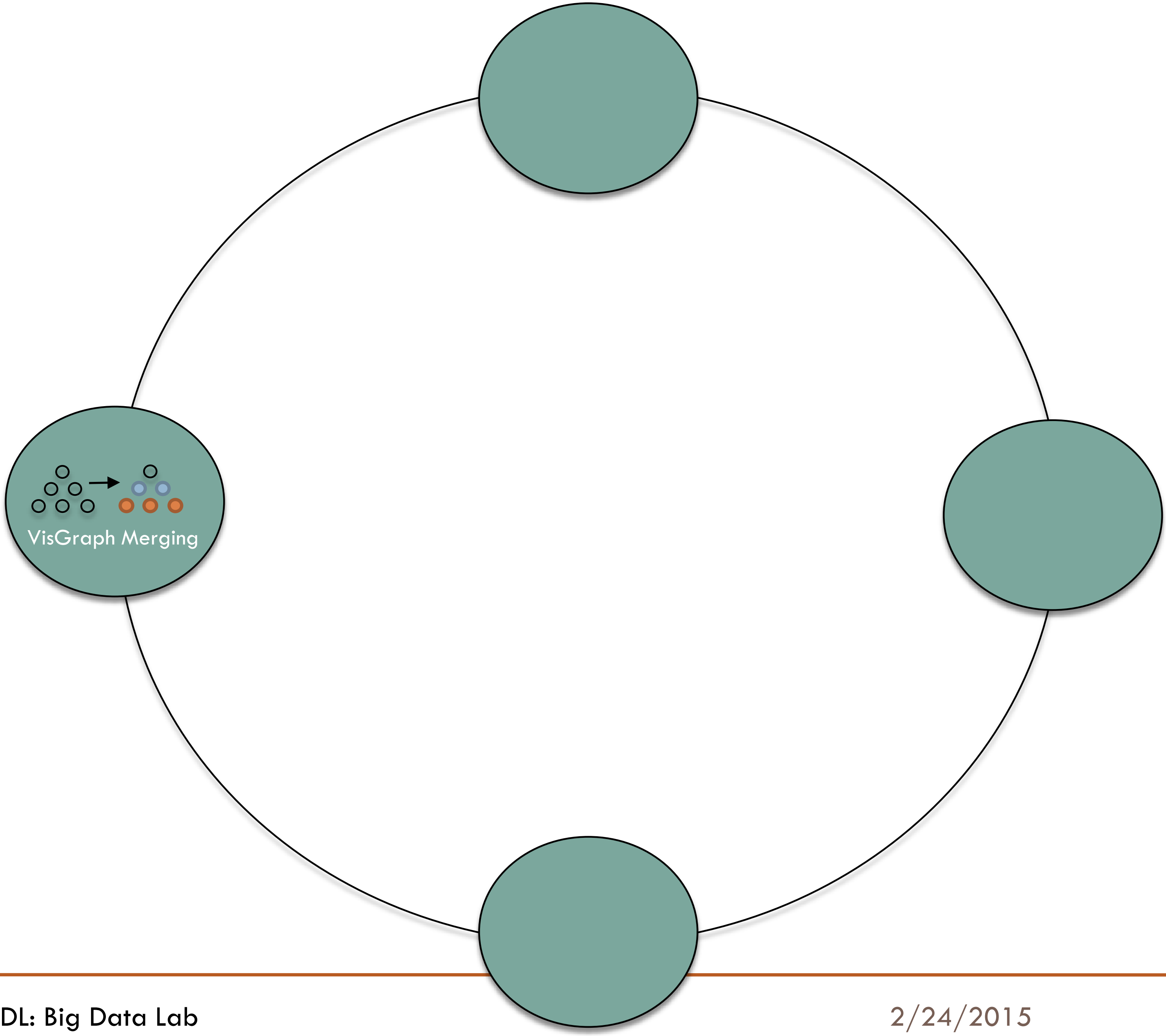

Visual
Query
Coordinator

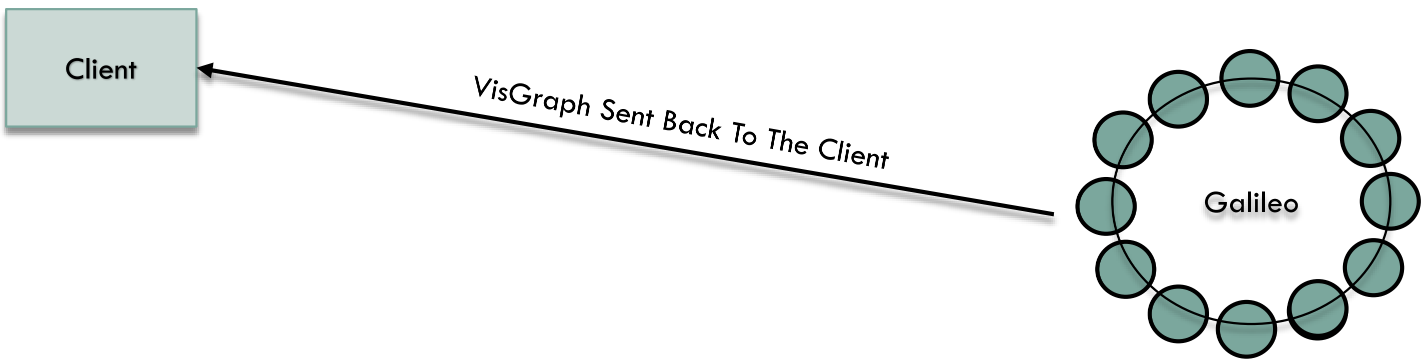

Visual
Query
Coordinator
Directed Sub Query to Node With Data

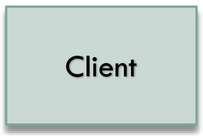

Visual
Query
Coordinator
Metadata Graph
to Vis Graph


Metadata Graph to Vis Graph
Metadata Graph to Vis Graph
Metadata Graph to Vis Graph
Metadata Graph to Vis Graph

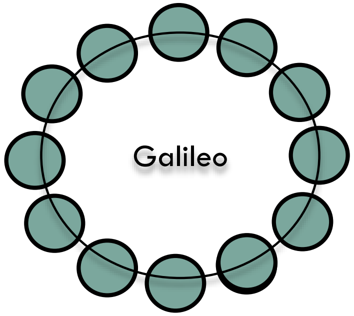

Metadata Graph to Vis Graph
Metadata Graph to Vis Graph
Individual VisGraph Response
Metadata Graph to Vis Graph
Metadata Graph to Vis Graph


VisGraph Merging


Client
VisGraph Sent Back To The Client
Galileo



VisGraph Traversed To
Display Image Tiles
And Histograms



Data Flow

- Every new query starts the process all over again.

Demo

- Here is the *almost* completed re-do of the client side visualization.
- Missing two components:
  - Geohash colors for the highest stage.
  - Interactive brushing and linking
- www.cs.colostate.edu/~koontz/geolens

Questions? Suggestions?