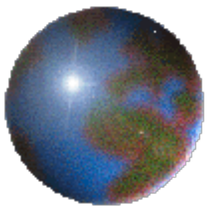


A Moving Objects Database Infrastructure for Hurricane Research: Data Integration and Complex Object Management

Markus Schneider, Shen-Shyang Ho, Malvika Agrawal, Tao Chen, Hechen Liu, Ganesh Viswanathan

UF UNIVERSITY of FLORIDA
The Foundation for The Gator Nation

UNIVERSITY OF MARYLAND 18 56

NASA

Problems

- Weather event related data like hurricane best track data and satellite data available in diverse file formats
  - Incompatibility of data formats leads to non-uniform data representations
  - Retrieval, combination, and analysis of data within the same source and especially across different sources are rather complex and inefficient
- Web-based weather event information portals, data archives, and forecast services provide “only” excellent visualizations
  - Restricted to certain data sources
  - Provide only limited, simple, and hard-coded query, retrieval, and analysis functions
  - No ad-hoc querying possible (instead: write application program for it)
  - In particular: no support for representing spatiotemporal objects (e.g., tropical cyclones, wild fires, weather systems) and spatiotemporal operations on them (e.g., determine all tropical cyclones of category 3 (in space and time) that crossed Florida between 1995 and 1998)

Our Solution: Moving Objects Database Technology

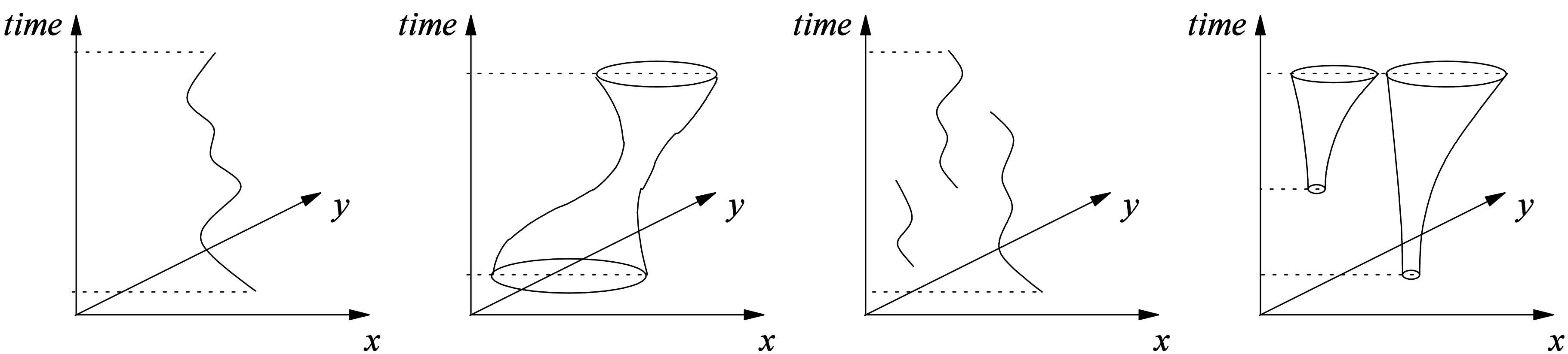
- Goal of our AIST project
  - The main goal of our project is to provide the NASA workforce with previously unavailable database management, analysis, and query capabilities that will advance the research and understanding of dynamic weather events and be based on weather data derived from the NASA mission sensor measurements.
- Database technology
  - Consistent and uniform data representation (tables, records, fields), efficient data retrieval, ad-hoc query (language) support, data indexing
  - Multi-user access, recovery, backup, concurrency control, transactions
- Moving objects database technology (MODT)
  - A moving objects database system is a full-fledged database system with additional capabilities to store, retrieve, query, and manipulate geometric data (called spatial objects) and spatiotemporal data (called moving objects) in its data model and query language.

What are Moving Objects? (I)

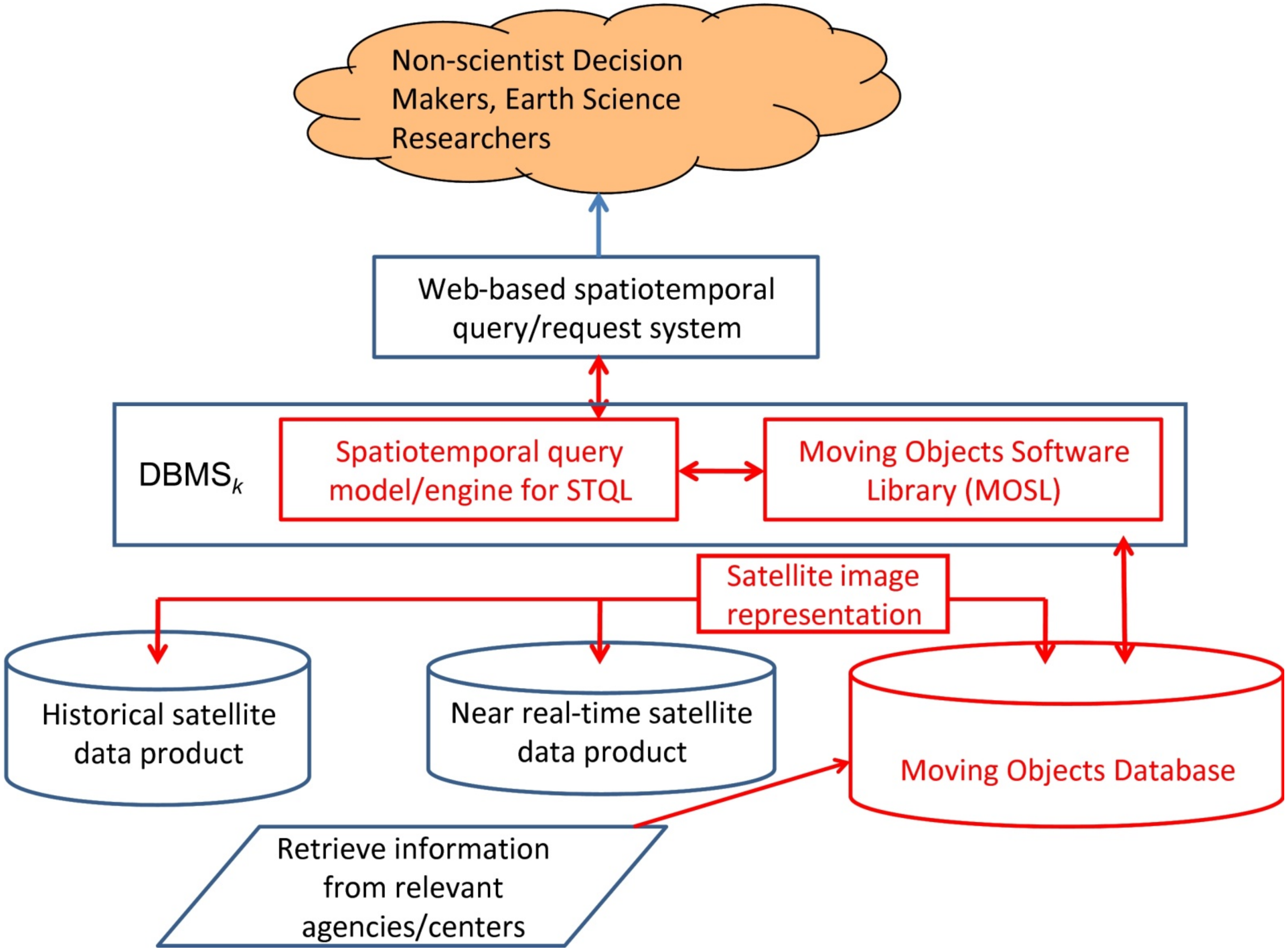
- A moving object represents the *continuous* evolution of a spatial object over time.
- Moving point
  - Only time-dependent location is of interest
  - Examples: eyes of tropical cyclones, cell phone users, terrorists, whales
- Moving region
  - Also the time-dependent shape and/or areal extent is of interest
  - Examples: hurricanes, forest fires, oil spills, diseases, glaciers
- Moving line
  - The time-dependent shape and/or linear extent is of interest
  - Examples: traffic jams, front of an army; boundary of any moving region

What are Moving Objects? (II)

- Formally: Let $\alpha \in \{point, line, region\}$. Then a moving object *m* is a function of a spatiotemporal data type $\tau(\alpha) = time \rightarrow \alpha$.
  - $mpoint = \tau(point) = time \rightarrow point$
  - $mline = \tau(line) = time \rightarrow line$
  - $mregion = \tau(region) = time \rightarrow region$



System Architecture



System Architecture Components (I)

- Moving Objects Database (MOD)
  - keeps tropical cyclone and hurricane data provided by public sources and web sites in a centralized repository
  - is a full-fledged database with additional support for spatial and spatiotemporal data in its data model and query language
  - stores the moving objects created by MOSL as attribute values of a relational, object-relational, object-oriented, or other database systems

- Moving Objects Software Library (MOSL)
  - provides a system of spatiotemporal data types together with a large number of spatiotemporal operations (e.g., *Intersection*, *Union*, *Difference*) and spatiotemporal predicates (*Inside*, *Meet*, *Disjoint*, *Overlaps*; *Enters*, *Leaves*, *Crosses*, *Bypasses*)
  - provides historical spatiotemporal data types like *hmpoint*, *hmline*, and *hmregion*
  - is database-independent and application-neutral

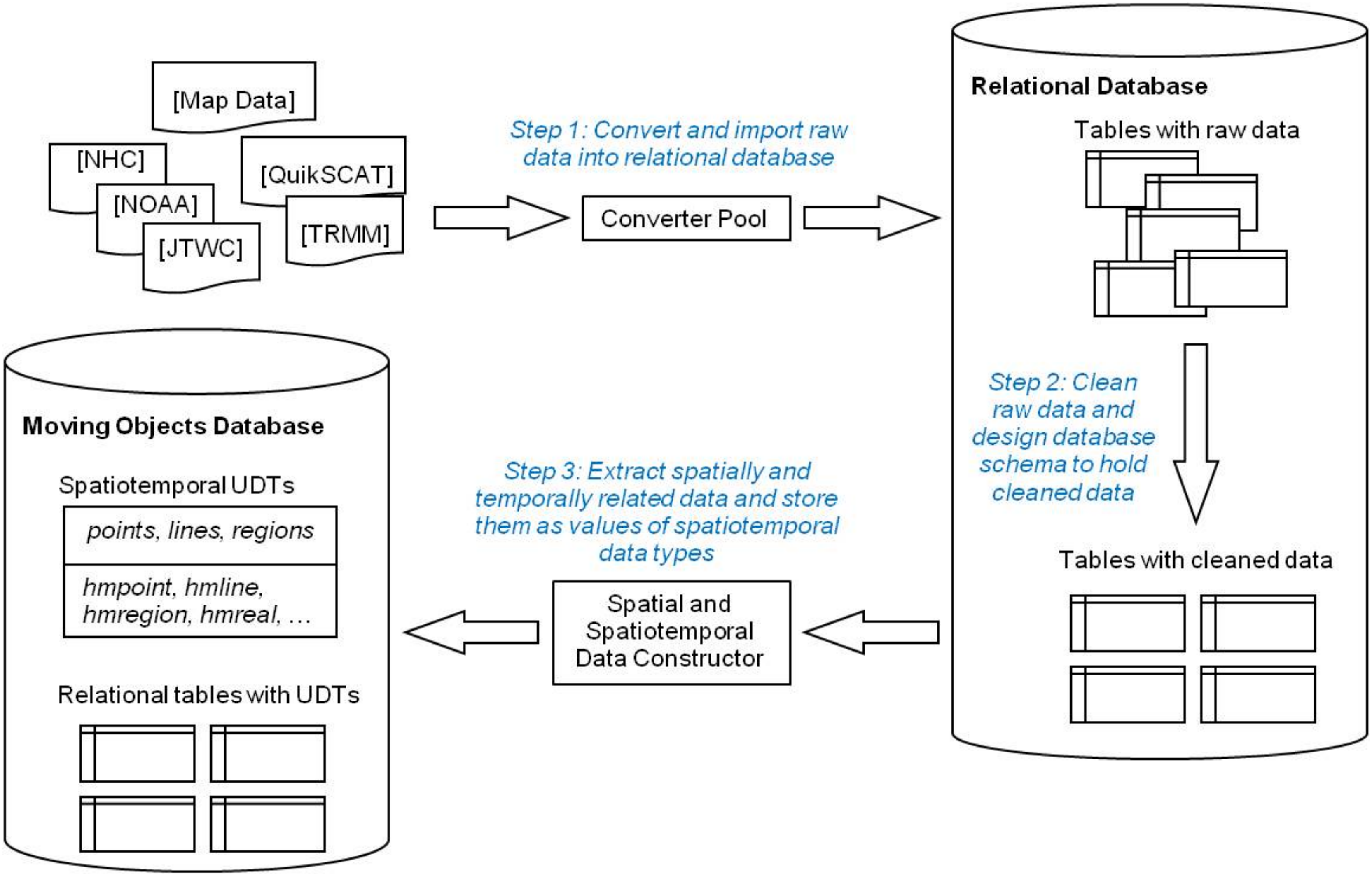
System Architecture Components (II)

- SpatioTemporal Query Language (STQL)
  - provides the communication interface between the moving objects database for tropical cyclone data and the user/scientist
  - enables users to pose ad hoc spatiotemporal queries on moving objects in general and tropical cyclone data in particular
  - allows users to obtain immediate response
  - enables the retrieval of satellite data based on user queries (with SDR)
- Satellite Data Retrieval Component (SDR)
  - takes output from an STQL query as input for satellite data retrieval
  - enable users to manipulate satellite data for future data analysis

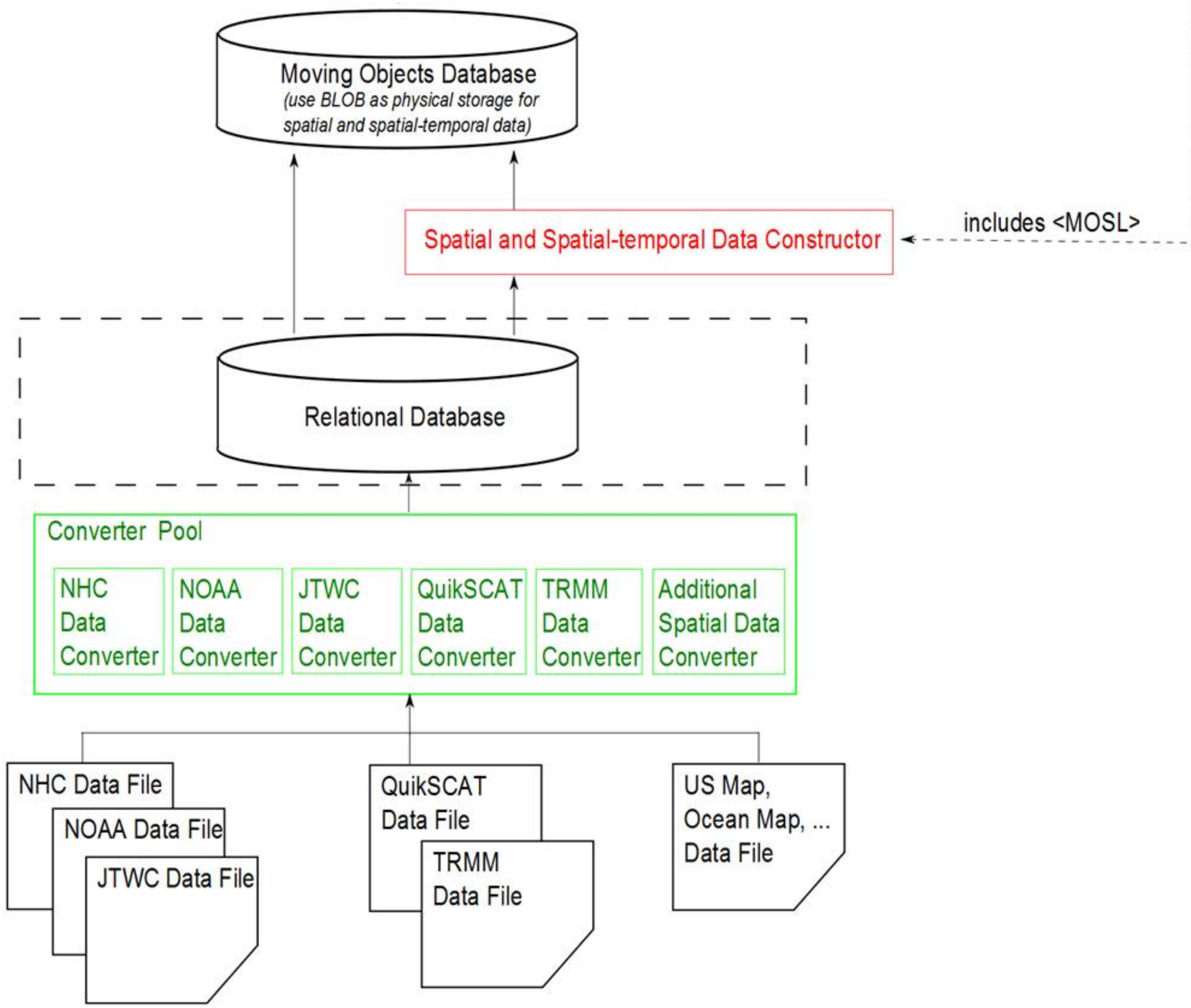
Data Migration and Integration Problem

- Storing hurricane data from varies sources in plain text or complex file formats such as HDF is not effective for users to perform queries on the movement of hurricanes
- Example
  - QuickSCAT data is stored in tens of thousands of HDF files differentiated by time range (each HDF file stores the scanning result of a satellite for a period of two hours)
  - It is only possible to perform queries within one HDF file each time by a particular application program.
  - A query like “Find all time intervals between time t_1 and time t_2 when the wind speed exceeds 50 m/s at the location (x, y)” cannot be answered efficiently.
  - Method: find all HDF files between t_1 and t_2, extract the data from all the files, filter the data by the condition of the query in each file, and union all results in the end.

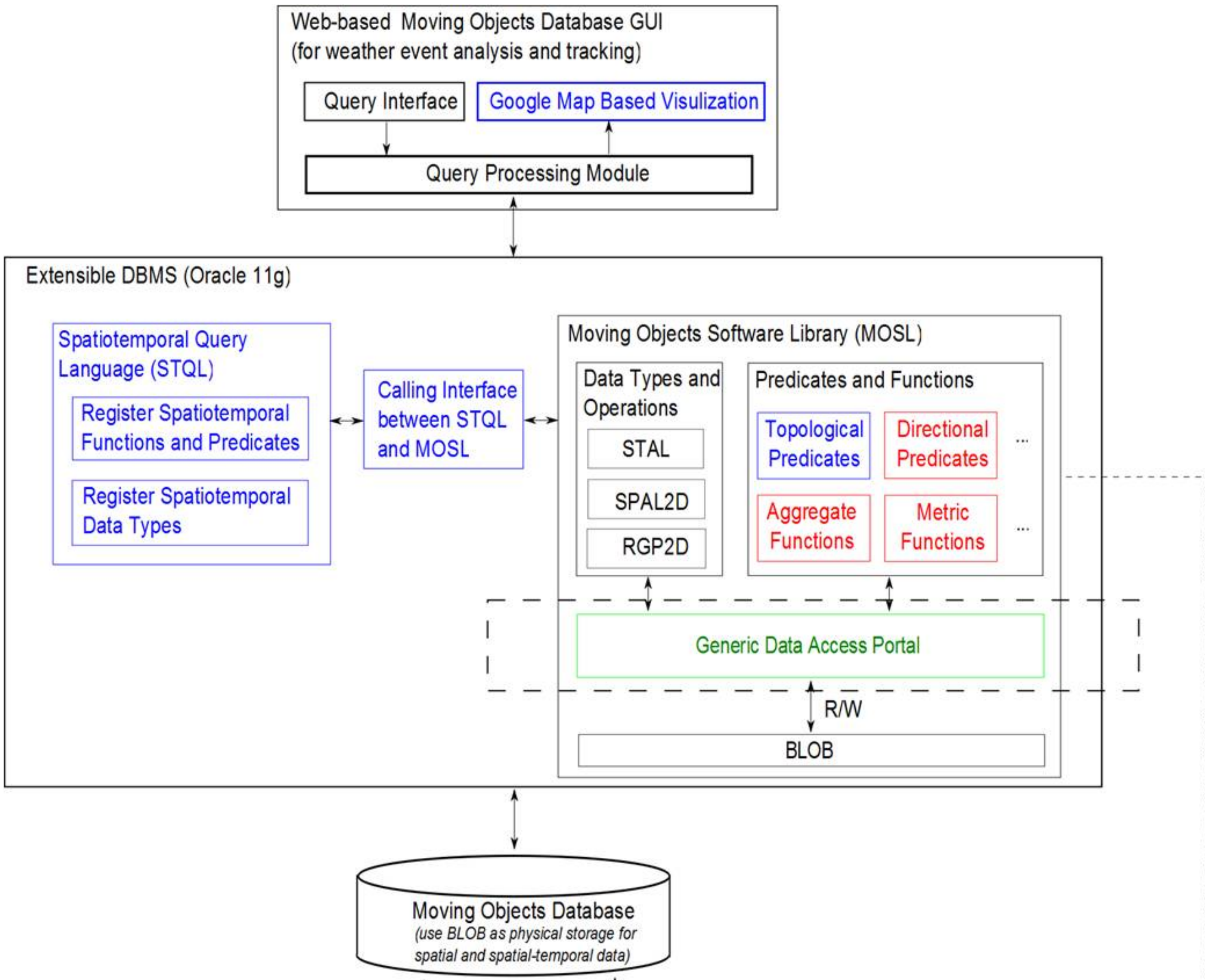
Data Migration: Data Processing Steps



Data Migration: System Architecture



MOSL Architecture



Conclusions and Future Work

- Conclusions
  - Data migration and integration and the system architecture of a moving objects database system for ad-hoc querying and retrieval of tropical cyclone data and their associated satellite measurements
  - Complex object management (not discussed here, see paper)
  - Use database systems for your data management!
- Future Work
  - Provide all system components with additional functionalities
  - Continue the development of MOSL
  - Continue the design of STQL
  - Optimization of the storage and indexing for satellite data
  - Exploration of additional scientific analysis operations
- Current prototype at http://phoenix.cise.ufl.edu. Try it out.

Thank You!