

MTR 06B0000255

MITRE TECHNICAL REPORT

# Sensor Data Analysis Framework Research Roadmap

## Integrating Stream Processing and Persistent Data Retrieval

February 2006

Don Landing, D580

Mark Munson, D560

Stephan Nadeau, D580

Seth Landsman, D580

Brigit Schroeder, D560

Eddy Cheung, W909

Virgil Zetterlind, D540

Dan Potter, D580

**Sponsor:** The MITRE Corp.  
**Dept. No.:** D580  
**Derived By:** -

**Contract No.:** -  
**Project No.:** 03MSR002

The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

©2006 The MITRE Corporation. All Rights Reserved.

**MITRE**  
Corporate Headquarters  
McLean, Virginia

## **Abstract**

*Many sensors produce large volumes of data regarding objects that change with respect to the location and time. Evaluating this stream of information in a timely fashion requires the integration of current and historical data. The Sensor Data Acquisition Framework (SDAF) research project seeks to understand and investigate various approaches to integrating streamed and historical sensor data with respect to spatio-temporal queries. In addition, techniques for addressing scalability, efficiency and data quality for integrated high volume sensor streams will be assessed.*



# Table of Contents

<b>1</b>	<b>SDAF Data Analysis Framework (SDAF) Roadmap</b>	<b>1</b>
1.1	Overview	1
1.2	Purpose	1
1.3	Schedule	1
<b>2</b>	<b>Research</b>	<b>2</b>
2.1	Research Question	2
2.2	Research Objective	2
2.3	Academia Collaboration	2
2.4	MITRE Corp. Collaboration	3
2.4.1	Trustworthy Data	3
2.4.2	ISR Forensics	4
2.4.3	Kaleidoscope	6
<b>3</b>	<b>Architecture and Design</b>	<b>7</b>
3.1	System Components	7
3.2	Information Flow & Analysis	10
3.2.1	Historical Query	10
3.2.2	Pre-Event Query	11
3.2.3	Combined Query	12
3.3	Deployment	13
3.4	Analysis Objectives	14
3.5	Analysis Framework Server	15
3.5.1	Phase 1 Analysis Framework	15
3.5.2	Phase 2 Analysis Framework Services	17
3.6	Analysis Framework Client	19
3.6.1	GUI components	19
3.6.2	Object Model	20
3.7	Stream Processing and Analysis	21

3.8 Data Persistence and Retrieval	23
<b>4 SDAF Development</b>	<b>25</b>
4.1 Personnel	25
4.2 Equipment	26

## List of Figures

Figure 1 – ISR Forensics MOIE.....	5
Figure 2 - PSP-CDM Architecture Model .....	6
Figure 3 - SDAF Phase 1 Architecture.....	8
Figure 4 - SDAF Phase 2 Architecture.....	9
Figure 5 - Historical Queries.....	10
Figure 6 - Pre-Event Queries .....	11
Figure 7 - Combined Query Operations.....	12
Figure 8 - Deployment Diagram.....	13
Figure 9 - Phase 1 Sensor Data Analysis Framework Layering .....	16
Figure 10 - Phase 2 Sensor Data Analysis Framework Services .....	18
Figure 11 - Test Client GUI.....	19
Figure 12 - GUI Object Model .....	21
Figure 13 - Borealis Data Flow Diagram.....	22
Figure 14 - GMTI Data Warehouse Baseline .....	24

# 1 SDAF Data Analysis Framework (SDAF) Roadmap

## 1.1 Overview

Many sensors produce data regarding objects that are continuously changing with respect to location and time. Effectively evaluating this data requires the integration of streamed and historical information. The Sensor Data Acquisition Framework (SDAF) research project seeks to understand and investigate various approaches to integrating streamed and historical sensor data with respect to spatio-temporal queries.

Analysis Objectives for the FY '06-'07 research will address multiple issues relating to the integration of streamed and historical sensor data. Initial analysis will focus on methods and techniques for data integration and query fusion. Later analysis will deal with issues of data quality and pedigree. Additional analysis will address special issues relating to advanced streaming media operations.

## 1.2 Purpose

This document has two major purposes:

- To describe the fundamental components and architectural roadmap for establishing an analysis framework featuring streaming and persistent data from various moving-object sensors.
- Clearly define the research objectives and approaches needed to satisfy the FY '06 Research objectives with respect to SMTI and media streaming data and to describe outyear efforts.

## 1.3 Schedule

### Project Calendar

---

Symposium Summary Brief due	April, 3, 2006
Tech Symposium – McLean, VA	April 27, 2006
Tech Symposium – Bedford, MA	May 4, 2006

## **2 Research**

### **2.1 Research Question**

How are sensor data streams from multiple sensors and archived data optimally structured to support exploitation and semi-automated analysis while still ensuring that pedigree and other metadata are preserved?

### **2.2 Research Objective**

For FY '06, the SDAF MSR will seek to develop an integrated query approach that simultaneously accesses streaming and archive data sets from multiple sensor types. Research with academia will attempt explore and understand spatio-temporal queries of streaming and persistent moving objects and moving sensor data. The SDAF MSR will work with internal MITRE research programs to understand their relationships/contributions to the SDAF MSR. A framework will be developed to assist in the analysis of spatial-temporal queries based on differing index techniques and data schemas.

Continuing in FY '07, methods will be examined to extend sensor types to include high volume sensor processing in areas such as streaming media technologies. In addition, the development and maintenance of data quality and pedigree will be explored.

### **2.3 Academia Collaboration**

The SDAF MSR is collaborating with on-going research in distributed stream processing technology by collaborating with researchers who are involved with the development of the Borealis stream engine. Borealis is a second-generation stream processing engine developed by faculty and graduate students at:

- MIT, Cambridge, MA
- Brandeis University, Waltham MA
- Boston University, MA
- Brown University, Providence, RI



SDAF has worked closely with the academic community at MIT, Brown and Brandeis through ongoing bi-weekly development meetings and engineering exchanges to implement and extend features and capabilities within Borealis. This collaboration has been instrumental in moving Borealis towards a deployable release with respect to sensor data processing.

Initial collaboration has been directed towards exploring issues surrounding the processing of sensor data from multiple concurrent sources. Issues such as the handling of data quality issues, event prioritization and data pedigree are being addressed. With data arriving at multiple times, the ability to reprocess and update earlier results has formed an additional line of investigations.

Emerging areas of research are delving into data integration issues with combining real-time sensor data and historical or persisted data. Since the persisted data may take a significant time to query, specialized techniques may be required to reduce the processing latency for stream based analysis. Research into areas of speculative query processing and pre-fetch optimization are being formulated.

## **2.4 MITRE Corp. Collaboration**

Due to the broad nature of stream and archival integration efforts, the SDAF MSR has attempted to leverage and collaborate with current and previous MITRE research efforts whenever possible. For the purposes of integrating ground/surface based moving target indicator (G/SMTI) data, SDAF is leveraging efforts from the following Mission Oriented Investigation & Experimentation efforts:

- Trustworthy Data
- ISR Forensics
- Kaleidoscope.

### **2.4.1 Trustworthy Data**

The objective of the Trustworthy Data MOIE, led by Dave Becker, is to define, capture, and use the semantics of enterprise data quality to improve enterprise decision support and information processing. The will impact the use of tools, techniques and information by humans to improve their judgment.

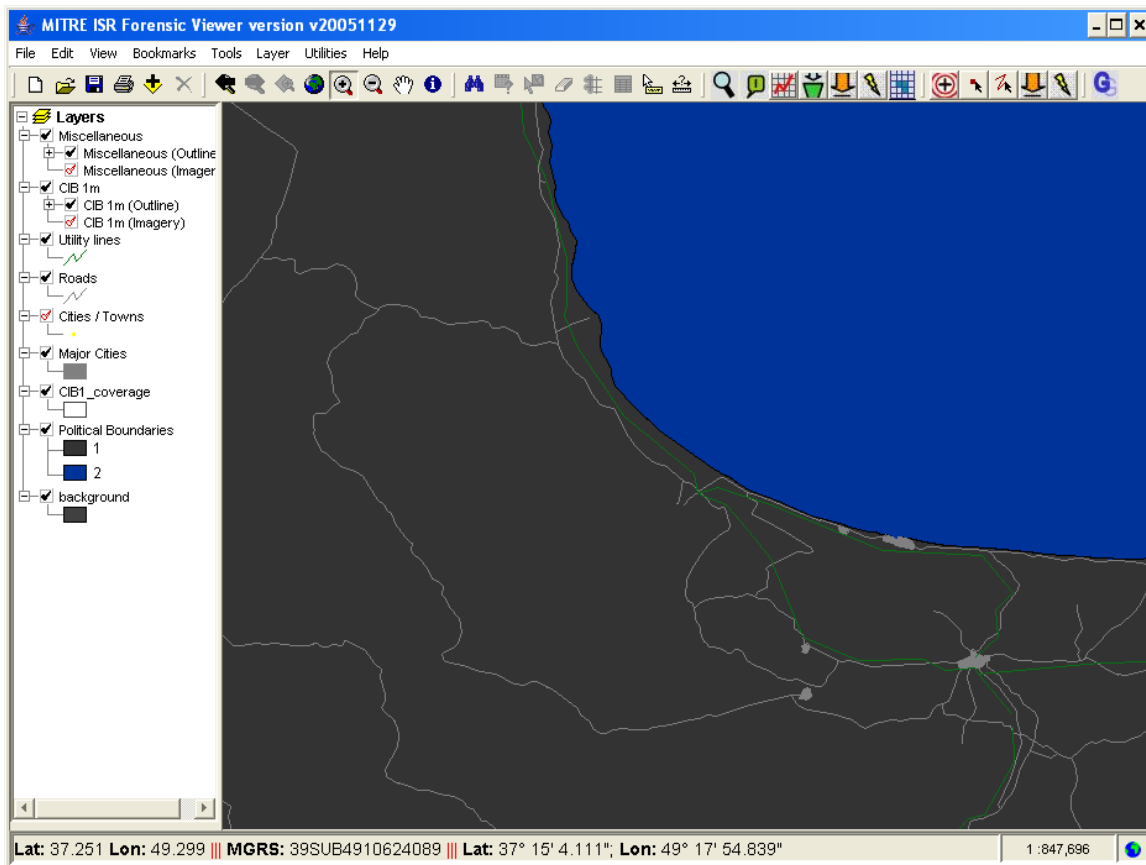
Data quality is impacted by a number of facets of which include:

- Accuracy: degree to which the reported information value is in conformance with the true or accepted value
- Precision: exactness of value
- Consistency/Validity: degree of satisfaction of constraints
- Completeness: degree to which values are present in the attributes that require them
- Pedigree: history of data origin
- Timeliness: degree to which specified data values are up to date

The SDAF MSR plans to provide streaming data examples that will then be tested through the Trustworthy Data MOIE. As data streams are filtered and re-combined, the data can lose pedigree, precision, and accuracy.

#### **2.4.2 ISR Forensics**

The ISR Forensics MOIE, led by Curtis Brown, has a task underway to develop a true Ground Moving Target Indicator (GMTI) data warehouse. This data warehouse, as developed by Peter Sylvester, will begin to break apart the mission files and store them in a manner that can be queried for investigative purposes.



**Figure 1 – ISR Forensics MOIE**

**GUI tools allow for complex selection and discovery through the analysis of persisted sensor data.**

This version of GMTI Data Warehouse addresses the needs for fast ingest and retrieval of processed GMTI mission data. It does not, at this time, address the need for data pedigree to be preserved as that raw data (I&Q) is processed. There is a belief by other agencies that access to the original data is required in order to derive additional data attributes.

### 2.4.3 Kaleidoscope

The Kaleidoscope MOIE, led by Steve Matechik, employs innovative visualization techniques to render multi-sensor fusion products consistent with the attributes of network centric warfare and effects-based operations. Specifically, JSTARS GMTI tracks will be automatically correlated with Video MTI (VMTI) tracks derived from Predator UAV electro-optic sensors to meet the Air Force Chief of Staff's challenge to provide "the best MTI available."

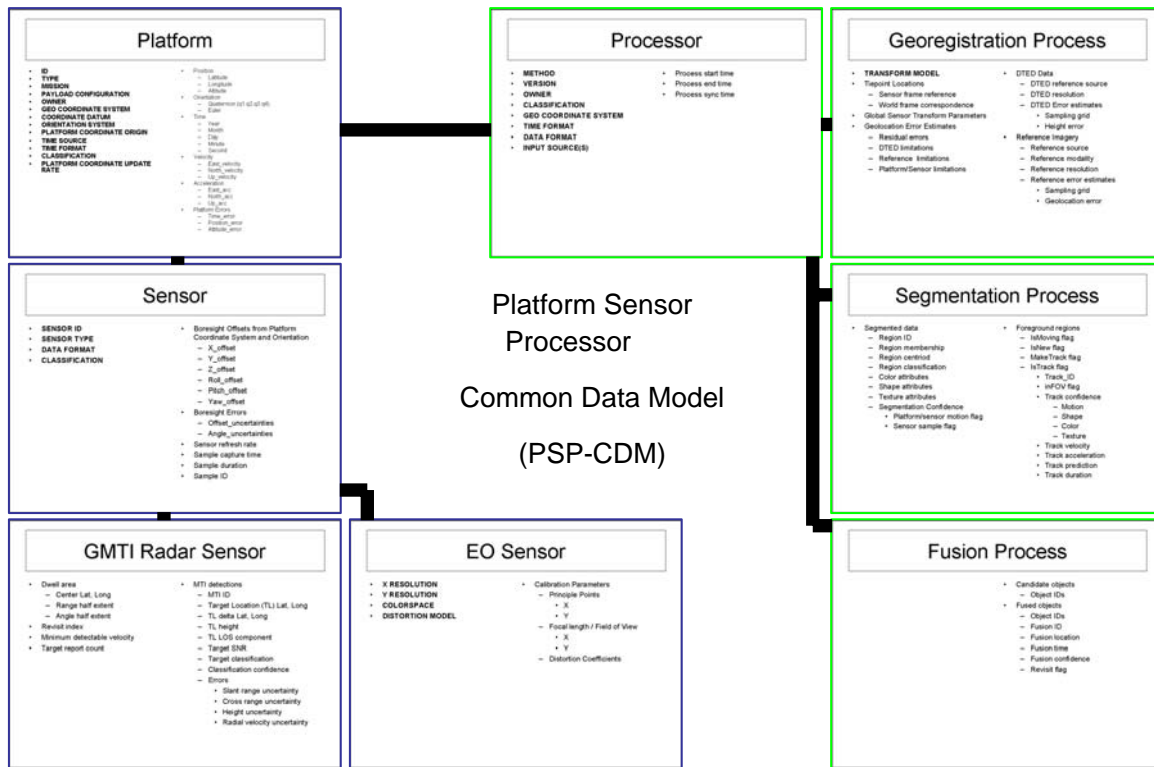


Figure 2 - PSP-CDM Architecture Model

The Common Data Model begins to develop mappings between the different sensor types

Part of their efforts includes the development of a model called Platform Sensor Processor Common Data Model (PSP-CDM) which is meant to begin developing a logical mapping of the sensor outputs to metadata attributes. The PSP goes beyond typical sensor models by providing specific fields for storing derived features such as geo-location, target segmentation, moving target tracks in a common store.

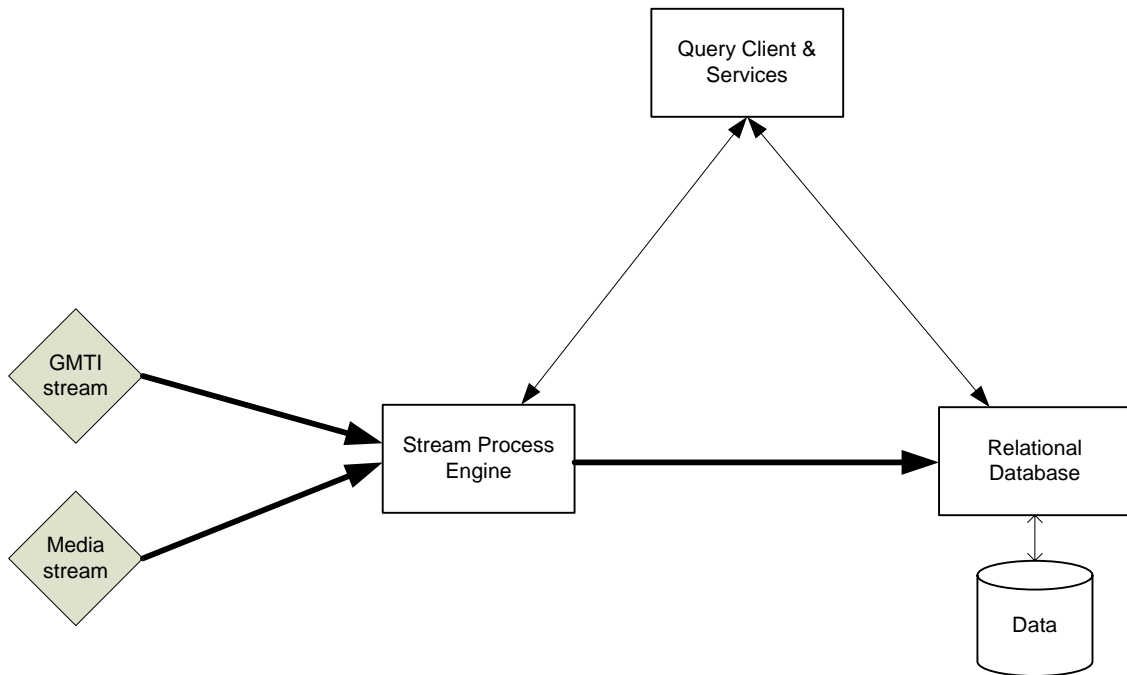
By building a common data store for UAV metadata and derived information, Kaleidoscope's archive system will allow for contextual queries into sensor data expanding the possible search beyond time and space. Supported extended queries will include finding moving target occurrences, filtering moving targets based on motion, size and color, and finding coincident targets in mutli-INT collections.

The data models and UAV persistent store will be critical for the SDAF MSR in order to build its framework. Kaleidoscope's data store will be exposed via an enriched Cursor-on-Target (eCOT) XML schema and will also support the streaming of real-time data feeds.

## **3 Architecture and Design**

### **3.1 System Components**

The system required to perform initial research in Phase 1 is composed of three main processes as shown in Figure 3. The first process is a Stream Process Engine that is used to query streaming data that is received from supplied sensor streams. The second process is a relational database that will archive process information that is derived from the raw stream data. Lastly, an evaluation client process will be used to measure and test various query and retrieval mechanisms that are under investigation.

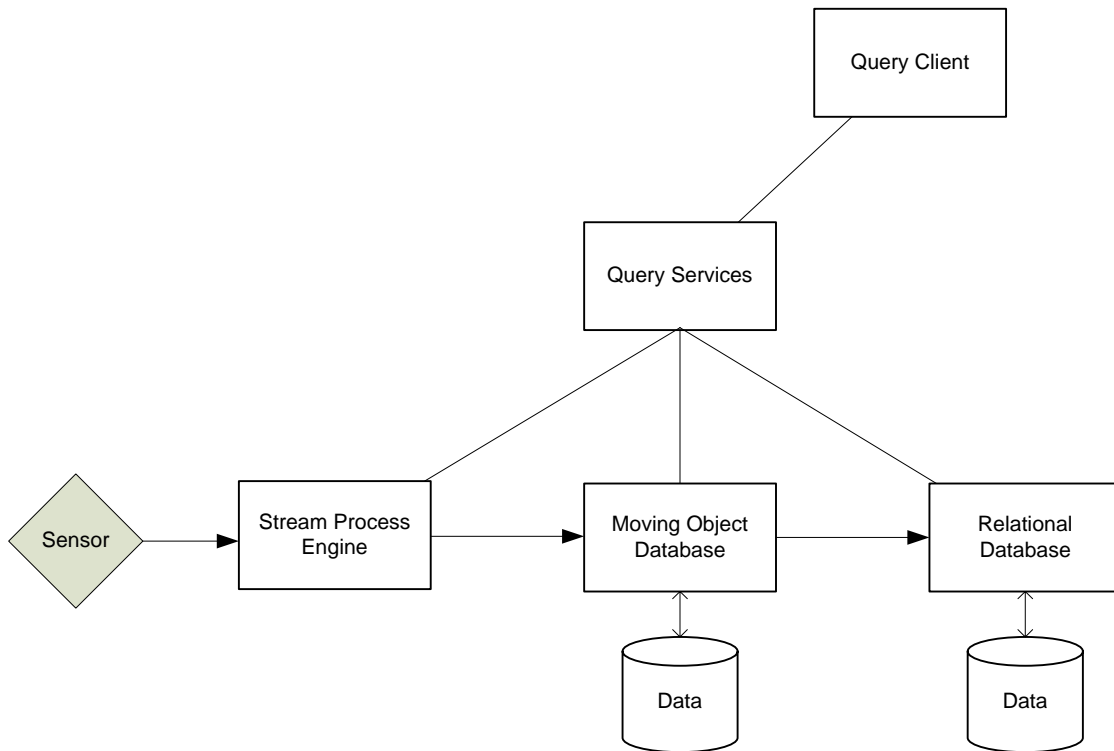


**Figure 3 - SDAF Phase 1 Architecture**

**Phase 1 Architecture combines all query analysis and sensor fusion services within a single test client.**

The first three processes will constitute the first architecture phase. Initial research will focus on the ability to process and query streams from two forms. The first will be composed of Ground Moving Target Indicator (GMTI) data that is derived from NATOEX data files. The second stream will mimic higher bandwidth video data similar to that provided by moving video sensors. The second stream will present processes meta-data that is suitable for combination with other MTI data sources.

Phase 2 is composed of additional processes and services which will expand on the initial research to test techniques for integration of data pedigree with more sophisticated queries, streams and archival data. Figure 4 shows the initial process used which will abstract the query operations from the business logic used to process and organize sensor fusion and query state.



**Figure 4 - SDAF Phase 2 Architecture**

**Phase 2 Architecture shows the inclusion of optional query services and a framework for exercising Moving Object indexing and pedigree analysis.**

Initial evaluation of the SDAF framework has suggested that a time gap exists between the short term data persistence that is available within Borealis stream process engine (SPE), and the completion time for data ingest required for index processing in a geo-spatial relational database. This gap is filled by the inclusion of a Moving Object Database process between the SPE and the persistent relational database.

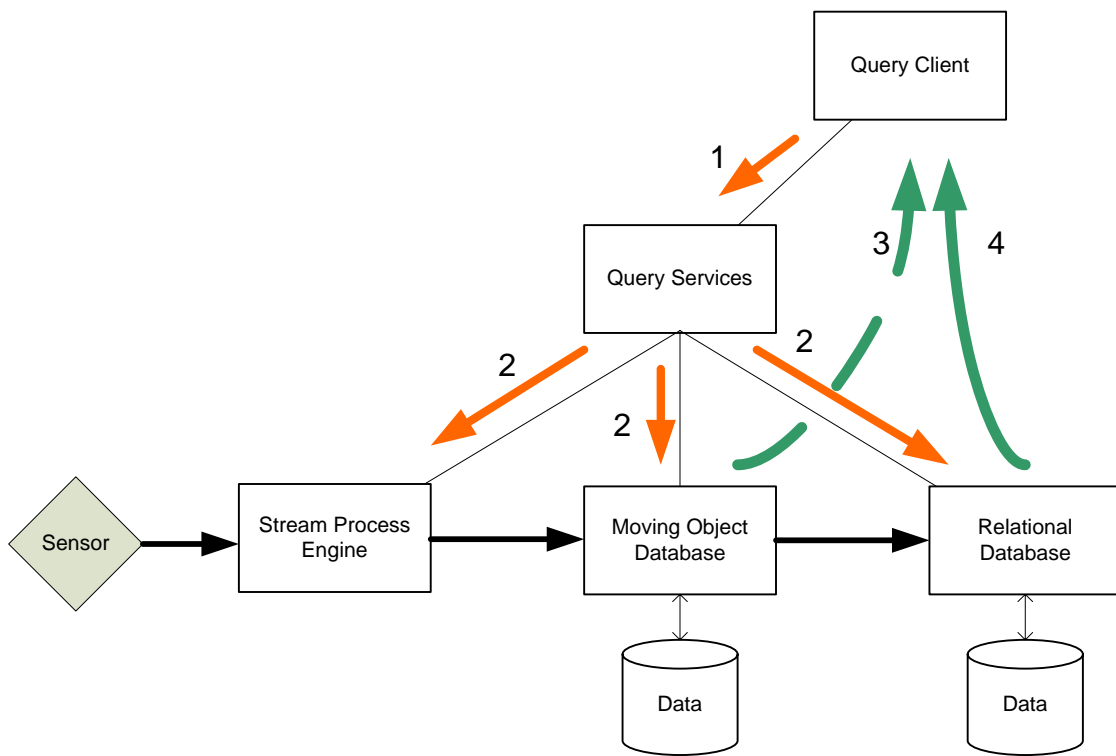
The Moving Object Database process will add short term persistence mechanisms between the SPE and the historical data storage. This process will leverage knowledge gained from the initial investigation and will allow for advanced analysis of data precision and the impact of stream processing operations.

Process functionality for efficiently manipulating and querying historical data will be refactored within a set of services for use by the streaming processes and other direct client queries. These methods for rapid data correlation and query will continue to evolve during Phase 2 research.

## 3.2 Information Flow & Analysis

### 3.2.1 Historical Query

In a standard historical query view, raw data is combined with information from multiple sources to create a result set on the client display. To achieve this, the client submits a query request that is sent to multiple data and stream functions. Matching result data is returned to the calling client. Figure 5 shows the typical sequence of operation.



**Figure 5 - Historical Queries**

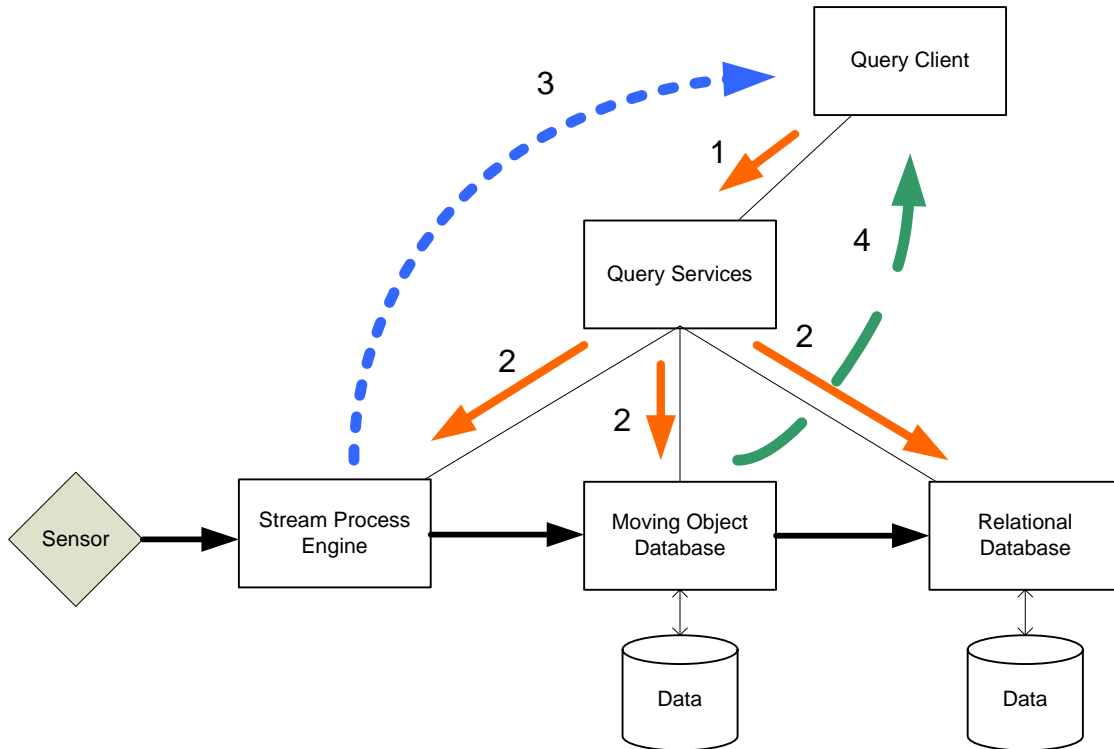
**Historical queries focus on data that is likely to return from existing data repositories.**

For a time bound query, the query services may elect to not query data sources that cannot result in return data. For the example shown in Figure 5, the Query Service would likely not forward a request to the Stream Process Engine. This would result in a static result set, and would reduce the overall processing that is required to satisfy the request.



### 3.2.2 Pre-Event Query

Pre-Event queries can capture both current and future results. In this scenario, the query represents a live combination of occurring and anticipated events. Figure 6 depicts a typical transaction flow. The returns for current events are expected to return multiple results over time as the dataset stream continually evolves.

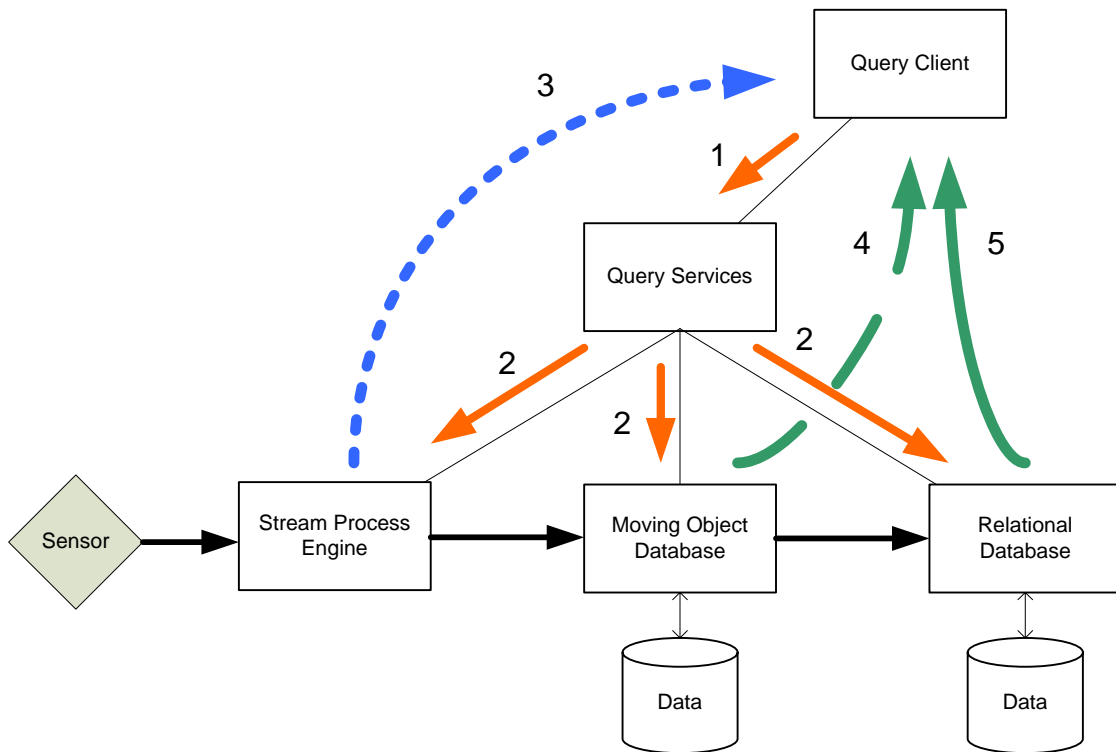


**Figure 6 - Pre-Event Queries**

**Pre-Event queries result in returns from raw data which are summarized and converted to information within the early stages of stream processing.**

### 3.2.3 Combined Query

A combined query results in an open ended temporal result that lives as a live query. Historical data is combined with current and future events as they occur. Figure 7 shows the typical transaction scenario for a combined query on a particular sensor.



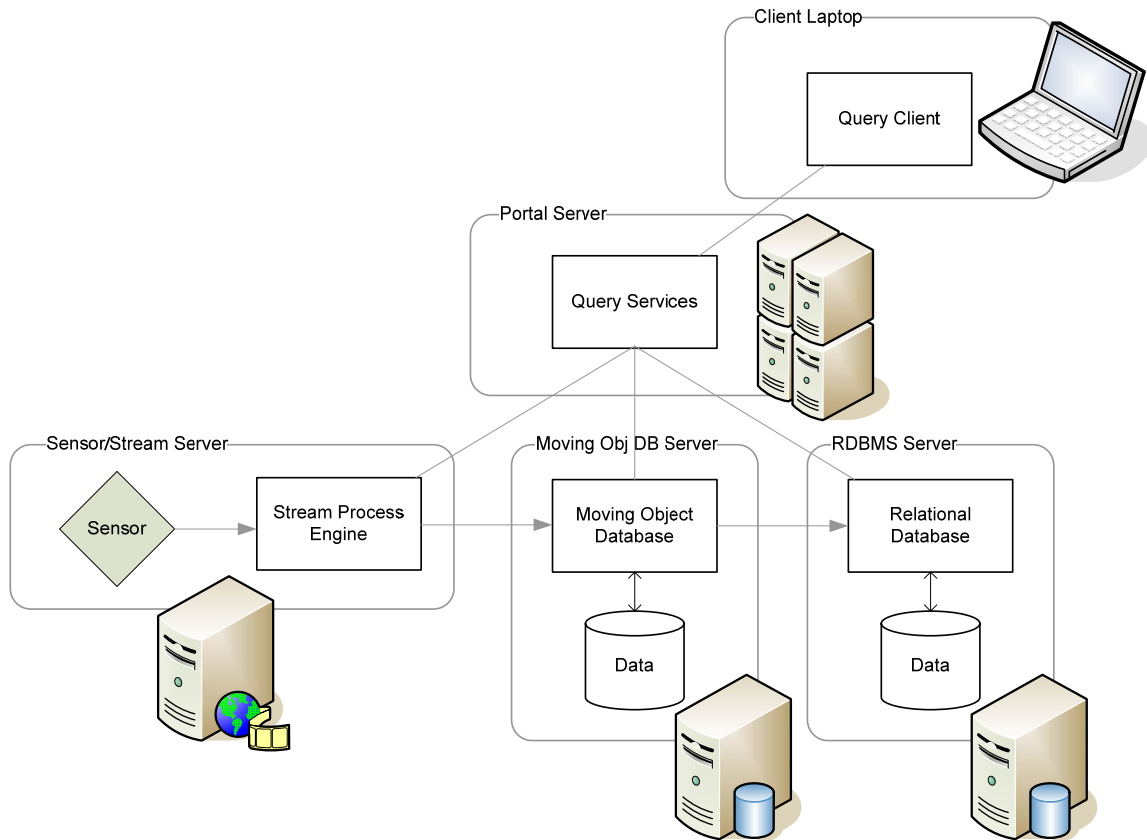
**Figure 7 - Combined Query Operations**

**Combined Query operations result in data returns from both current and historical information sources.**

For combined queries, the order of return is not certain. Current events may occur either before or after the results of a historical query are returned. If a moving object database does not exist, multiple queries to a historical database may be required to ensure that non-indexed data is fully realized in the result set.

### 3.3 Deployment

Major deployment for the described processes will likely require distributed processing across multiple platforms in a dispersed wide-area-network. Figure 8 shows the likely distribution of processes onto specific platforms. Smaller test environments may elect to combine processes as needed, at the possible expense of processor performance.



**Figure 8 - Deployment Diagram**

**Typical process deployments for a basic stream query operation are illustrated by the placement of phase 2 processes on deployed compute platforms.**

### 3.4 Analysis Objectives

Analysis Objectives for the FY '06-'08 research will address multiple issues relating to integration of streamed and historical sensor data. Initial analysis will focus on methods and techniques for data integration and query fusion techniques. Later analysis will deal with issues of data quality and pedigree. Additional analysis will address special issues relating to advanced streaming media operations. Selectable functions within the framework will allow for analysis in the following areas:

#### Phase 1 (FY'06 – FY'07)

- **Stream Processing Efficiency** – Selection and comparison of multiple evaluation techniques for assessing stream based performance. Techniques such as the Linear Road technique will be employed assess process capability.
- **Persistence Analysis** - Alternative methods for organizing sensor data will include Partition strategies and alternative Index Method comparison.
- **Spatio-Temporal Query Analysis** – Techniques for processing and distributing query operations across multiple sensor platforms for streamed and archival data will be studied
- **Query fusion for Streaming and Persistent Data** – Methods for combining and querying disparate sensor types will be addressed.

#### Phase 2 (FY'07 – FY'08)

- **Data Pedigree** – Stream processes can modify and recombine data into new data sources. Defining strategies for maintaining data pedigree will be essential to evaluating stream processing operations.
- **Data Quality** – As sensors platforms proliferate, multiple sensors will be available to detect an event. Mechanism for retaining data quality within a combined sensor query and prioritizing results will be investigated.
- **Integration Analysis for Advanced Media Queries** – Integration of higher bandwidth sensors such as video and moving imagery will be investigated with respect to query analysis and result fusion.

## **3.5 Analysis Framework Server**

The Analysis Framework Server components are meant to construct an evolvable platform for testing multiple approaches for sensor data integration and querying for merged stream and historical data. Basic components for the framework are generated in Phase 1, and will be integrated for test purposes using a standard layering approach into an integrated server/client library.

In Phase 2 lessons learned from the initial framework development will be applied to extensible services architecture. Functional components will be packaged into a container-based architecture with the exposure of standard services. The intent will be to assess issues of data quality, pedigree, and spatio-temporal query efficiency from within a net-centric compatible environment.

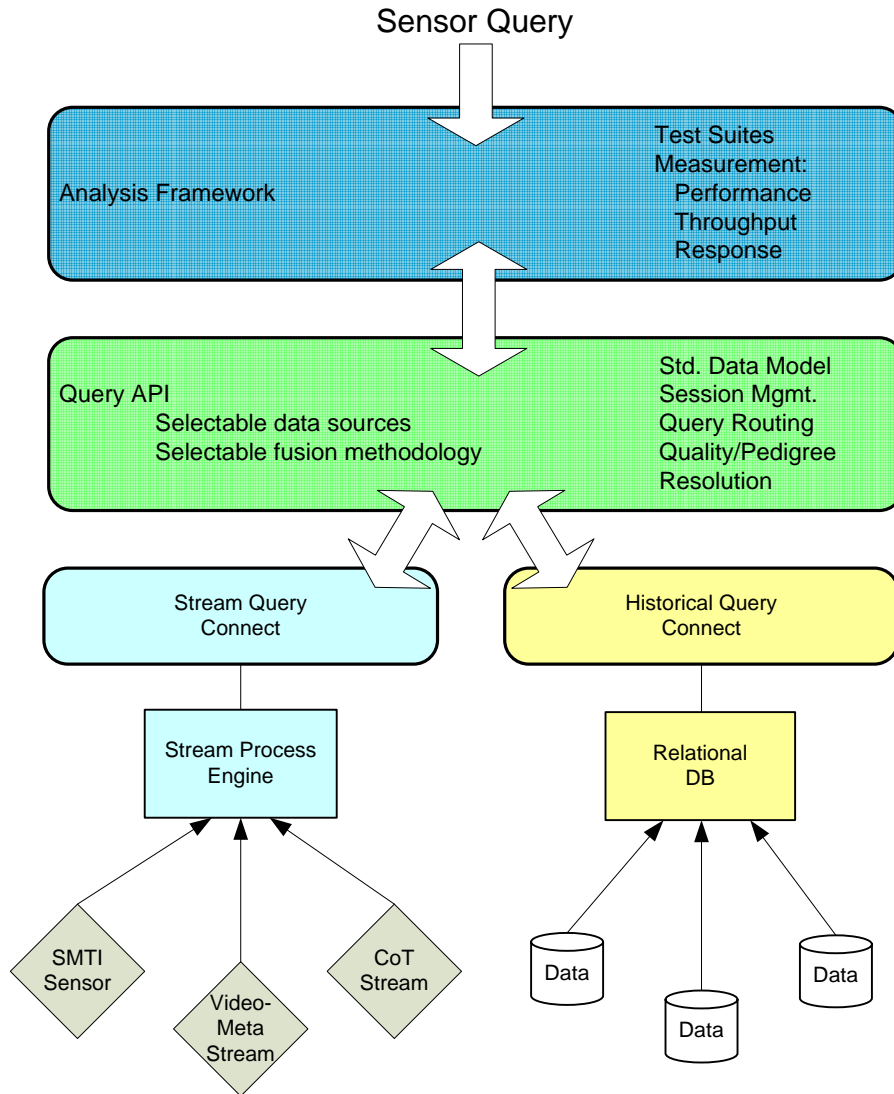
### **3.5.1 Phase 1 Analysis Framework**

Basic components for the Phase 1 Analysis Framework are outlined in Figure 9. The components are grouped into three main areas. These are the Analysis layer, the Query layer, and the Connect layer.

#### Analysis Framework Layer

All queries through the SDAF framework can utilize analysis services to select and measure techniques for accessing and fusing multi-sensor query operations. The analysis layer coordinates pass through calls to expose the Query API, while allowing for collection of data by analysis routines.

Imbedded test suites within the Analysis Framework perform consistent sets of operations to measure and compare data access and fusion efficiency for a range of query operations across both space and time.



**Figure 9 - Phase 1 Sensor Data Analysis Framework Layering**

**Major software components for performing combined queries are shown in a layered organization**

Query Layer

The Query layer is responsible for executing combined query operations and assembling result sets for continuous delivery. Query sessions result in multiple calls to query connections based on their likelihood to result in matching data. Selection criteria could be based on location, time, and type of data being requested.

Session management is maintained by the query layer, and resources are allocated and released as needed to maintain an active session. When session control is released, the Query layer will eliminate session constructs, and release needed queries and resources.

For phase 1, queries will assume that sensor state management may be obtained directly from the sensor connections. This means that the Query Layer will only maintain the state of the session, but will not provide sensor state operations, and will act as a stateless service in this regard.

### Connect Layer

The Connect layer provides direct services for obtaining and querying data from a sensor provider via a standard query API and simplified data model. The connect layer is responsible to executing sensor queries, and translating results into a common model format.

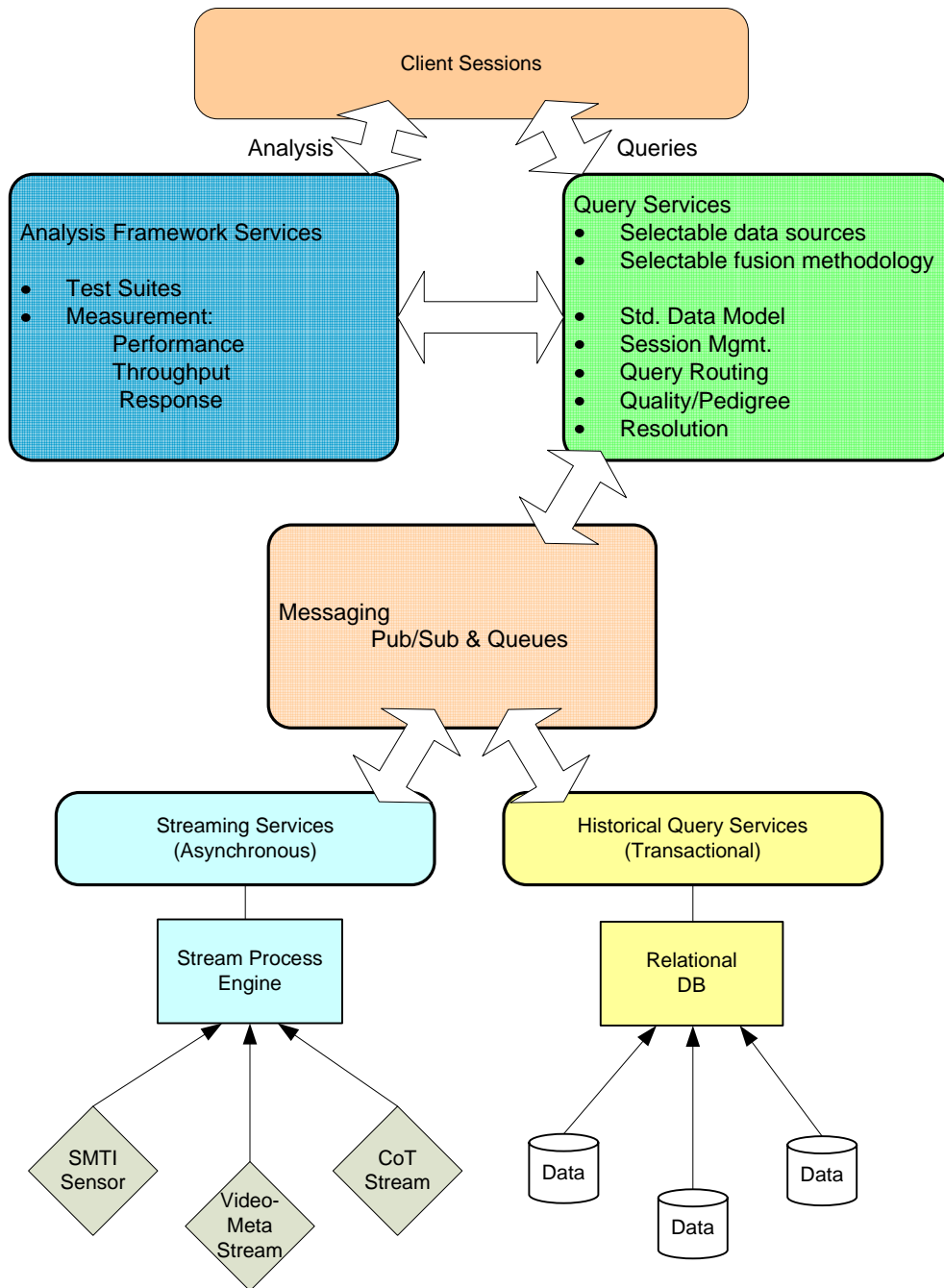
Common model information structure leverages knowledge gained from several research projects on sensor data modeling and platform description meta-data. The common model allows for rapid fusion of query results by focusing on common areas of interest that are likely to be searched.

The connect layer can handle queries from both streaming and historical relational DB sources. In the case of relational calls, a result set is obtained that causes a single transactional return. For stream processing, an active session is established to allow for asynchronous data exchange with the query layer.

### **3.5.2 Phase 2 Analysis Framework Services**

In Phase 2, components from the Phase 1 Analysis Framework are organized as service components within a container based architecture as shown in Figure 10. Within this architecture, both Analysis Framework and Query operations are recasted as available services that can interact with established sessions. Session management is no longer an integral part of the query operation, but is instead handled by the containment architecture.

Query Services are established to address issues of sensor data fusion and update coordination. Additional constructs to handle data quality and pedigree will be introduced into the query operations. For advanced imagery and streaming media, the need for alternative structures will be investigated that preserve the integration and update of asynchronous query operations.



**Figure 10 - Phase 2 Sensor Data Analysis Framework Services**



### 3.6 Analysis Framework Client

Presentation of live query operations is accomplished with the assistance of a GUI based client. Complex results involving geographical and temporal operations need to be supported, along with the ability to process historical and live data feeds.

#### 3.6.1 GUI components

The visible components of the GUI are the result of existing query operations as shown in Figure 11.

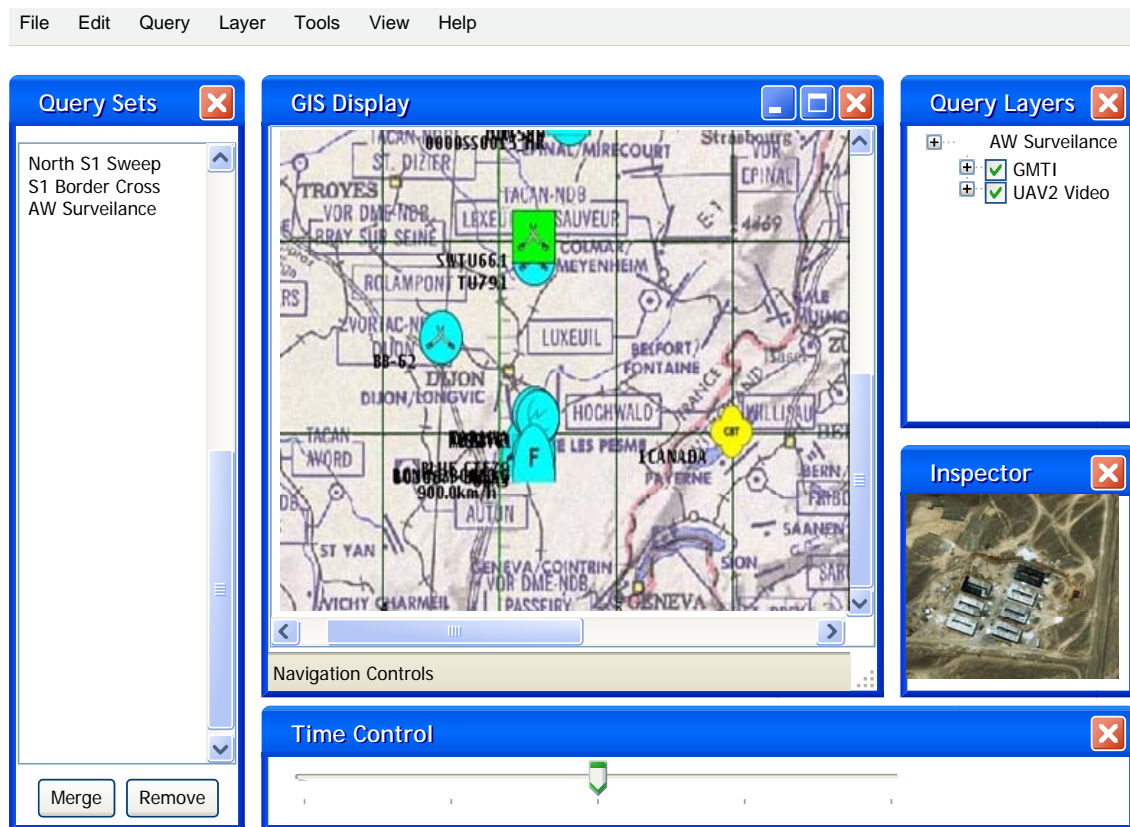


Figure 11 - Test Client GUI

Major display components for a typical Query Set are shown in this GUI mockup.

Query sets represent a live query in operation. As updates/changes are received via stream input, the query results will change accordingly. For queries with an unbounded temporal range, the query may continue indefinitely until terminated by the requestor.

Spatial representation of the result set is displayed in the GIS Display, along with temporal control from the Time Control window. Additional selection and operational non-modal dialogs exist on the right to assist the operator in analyzing and merging existing media data.

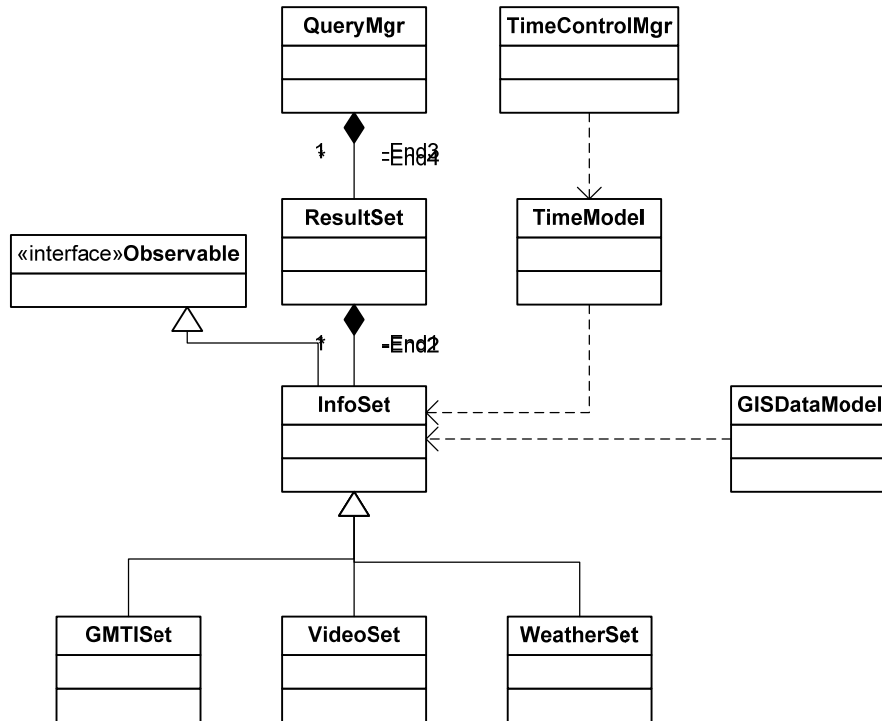
Temporal control may wrap a region of time, or may continue to expand in a streaming fashion. A selection device will be employed to allow multiple time viewpoints to be combined within a single GIS display window.

Some media stream layers may not be available for display within the GIS display. In this case other applications may be employed to assist if necessary.

### **3.6.2 Object Model**

As active queries are generated against streaming data, data models that represent a continuously evolving answer will be held at a “State” server. During Phase 1, the client will perform this function exclusively by building an internal data model of receive results. Figure 12 shows the basic framework for holding Query results.

Each query causes the generation of a query ‘Result Set’. Each ResultSet contains a series of sensor/information datasets as depicted by the ‘InfoSet’ object. As updates stream in, information revisions are added or applied to related InfoSet objects. Application of the observable pattern allows interested presentation components to update their displays accordingly.



**Figure 12 - GUI Object Model**

**Query operations result in a Result Set that comprises multiple information layers (Sets).**

Many of the GUI controls will interact with the code data elements through a Model-View-Controller (MVC) paradigm, which abstracts the underlying data structure from the generic operations used to perform control and view operations.

Due to the range of time elements that are possible in a result set, a special ‘Time Model’ class will be employed to control the specify range or unique time value that will be utilized to view resulting data.

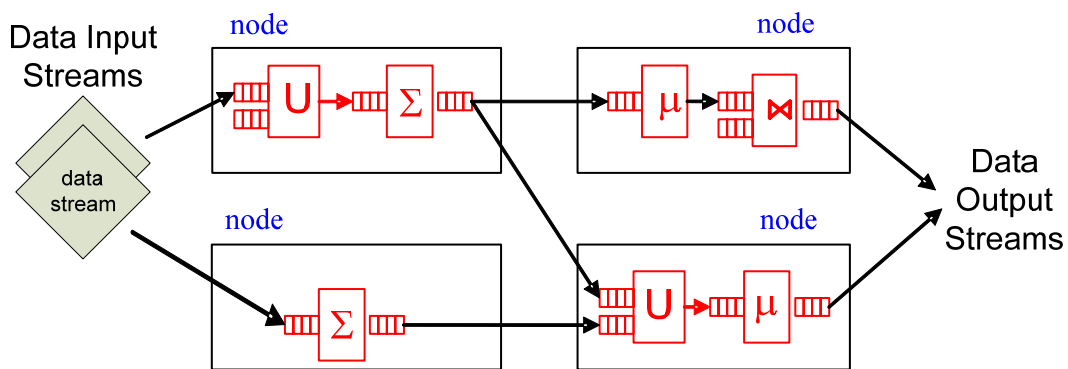
### 3.7 Stream Processing and Analysis

The Borealis stream processing engine is capable of querying and modifying data streams on a continuous run-time basis. Stream processing is highly dynamic in that query operators on the data stream may change, and processed data can revised or be corrected, at run time.

This functionality can be extended over a distributed network via inter-node communication, creating a network of Borealis nodes.

Borealis has been developed for the Linux x86 platform, primarily under Red Hat Linux. The C++ code base is in a beta development phase and is undergoing constant revision to enhance its capabilities.

As shown in Figure 13, a series of operators, or “boxes,” are used within Borealis to define the operational path of streaming data. Input Streams consisting of data tuples are continuously pushed through a network of grouped boxes which collectively form a query. Each box processes groups of data tuples atomically. Queries may be distributed across a network of Borealis nodes or deployed on a single node. The query deployment is defined by an XML schema; specifying both box inputs and outputs, and required parameters. Multiple sources may publish data to an input stream and likewise multiple client applications may subscribe to data output streams.



**Figure 13 - Borealis Data Flow Diagram**

**Data streams flow through operator “boxes” that collectively form distributed query operations.**

Stream processing engines are ideal for the evaluation and assessment of real-time sensor data. For example, route planning on roadways can be improved by correlating the detection of events (accidents, construction, etc.) with real-time sensor and historical data. Similar techniques could assist the warfighter by, observing and processing interesting activity in a battlefield environment.

Borealis may simultaneously process and analyze multiple streams of netted sensor data to create new information streams. For example, rapid emergency response to a disaster or

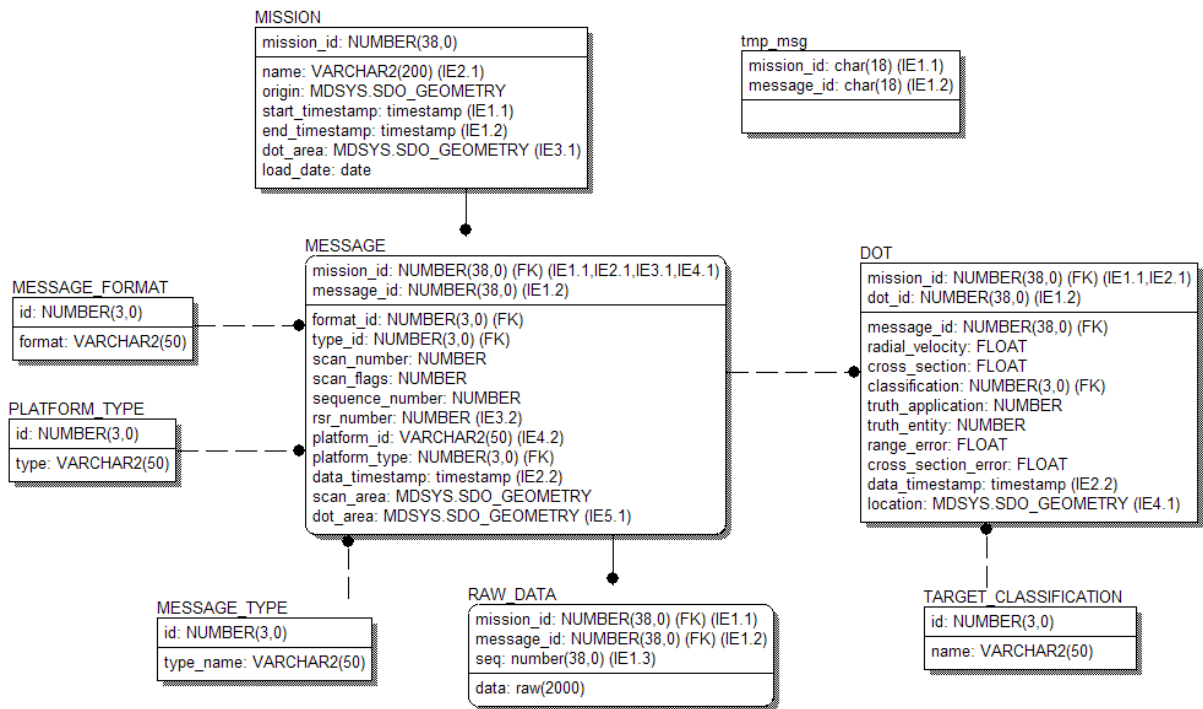
emergency situation could be aided by the integration of GPS data on responder vehicle locations and the analysis of sentinel video.

### **3.8 Data Persistence and Retrieval**

For persistence data storage and retrieval, the first phase 1 base case will be Ground - Moving Target Indicator (MTI) data in the NATOEX format. The data consists of variable length packets. Each packet has a 128 byte Header segment plus zero or more MTI segments. The maximum packet size cannot be more than 1472 bytes. The Header segment contains the type of message being sent, the number of subsequent MTI segments in the packet and other information. The MTI segment specifies target information and is repeated once for each target. No more than 42 target reports can be contained in a single packet.

Of all the information in a NATOEX packet, the location of the target and the time when the target was detected will probably be of most interest. This spatial-temporal nature of the data requires a database management system that supports and facilitates the storage, retrieval, update, and query of spatial features in a database. For its ability to handle large amount of data and its matured spatial features, Oracle 10g Release 2 has been selected as the persistence data store.

The format of the NATOEX data lends itself very nicely to a database schema with three tables: MISSION table contains the mission ID, start time, end time, elapsed time, etc. The HEADER table contains packet header information. A depiction of this schema relationship is shown in Figure 14. The MTI table contains the target information. The location information of a target (latitude, longitude) will be stored as a single column of object type SDO\_GEOMETRY in the MTI table. Once data has been loaded into the tables, a spatial index must be created on the tables for efficient access to the data. This is because Oracle spatial uses a two-tier query model to resolve spatial queries. The primary filter permits fast selection of candidate rows to pass along to the secondary filter. The secondary filter applies exact selection criteria to produce the result set. Oracle uses the spatial index to implement the primary filter. In addition, Oracle does not require the use of both primary and secondary filters. In some cases, just using the primary filter is sufficient to produce the desired results.



Note:  
 Tables message, dot, and raw\_data are partitioned on mission\_id.  
 Some relationships shown are not in physical model for performance and ease of maintenance.  
 The tmp\_msg entity is a "global" temporary table used during query processing.

**Figure 14 - GMTI Data Warehouse Baseline**

**The GMTI database schema that is utilized by the ISR Forensics MOIE forms the baseline for integrated query operations.**

To further enhance the performance and scalability of the database, logical tables can be partitioned into one or more physical tables, each with its own index. This is done by specifying a partition key. In the case of NATOEX data, the time field would be a good choice to use as the key. Oracle uses the partition key to decide which partition to put each row of data in. The benefits of partitioning are many. Some of the more important ones are:

- Store data in different tablespaces on a partition-by-partition basis
- Store indexes in different tablespaces on a partition-by-partition basis

This allows the spread of I/O load associated with table or index accesses across multiple disk drives and/or controllers.

- Search multiple table or index partitions in parallel
- Eliminate partitions from consideration based on a partition key

The last bullet is one of the most important ways partitioning can enhance performance. Partition elimination is the automatic exclusion of partitions that will not be participating in a query. If a query includes a partition key as a predicate in the WHERE clause, Oracle will automatically route the query to the partition or partitions that are associated with the query, eliminating (and not searching) those partitions that will not have data included in the result set. Partition elimination significantly reduces the amount of data and index information searched to return results.

Another technique to enhance performance is spatial partition pruning. In this case, data is grouped into partitions based on location values such as latitude and longitude. Spatial partition pruning is similar to partition elimination, but it is based on location and does not require a partition key to be specified on the input query. At query time, an area-of-interest of the query is compared to the Minimum Bounding Rectangle (MBR) of each partition. If they do not overlap, spatial partition pruning will occur and the data associated with that partition will not be searched. Both techniques will be explored fully to provide efficient query of spatial data in the persistence data store.

## 4 SDAF Development

### 4.1 Personnel

Primary development contributors to the SDAF MSR are:

Don Landing – Co-PI  
Stephan Nadeau – Co-PI – Research dev.  
Mark Munson – Proj. Lead - Analysis  
Seth Landsman – Research Analysis  
Brigit Schroeder– Stream Process Engine dev.  
Eddy Cheung - Database dev.  
Virgil Zetterlind – Exploitation dev.  
Dan Potter – GUI and software dev

## 4.2 Equipment

Development and testing of research components will be performed using the following equipment system list:

- Borealis Stream Processor
  - Gretel.mitre.org [129.83.174.77]
  - OS: Red Hat Enterprise 4 (Beta)
  - Loaded with Borealis Streaming Processing Engine
- DataBase Warehouse Server
  - Bueller.mitre.org [129.83.169.245]
  - OS: Fedora Core 4
  - Oracle DBMS for MTI persistence testing
- Stream Server / Client development
  - Sumatra.mitre.org [129.83.169.92]
  - OS: Fedora Core 4
  - SimServer / Borealis (secondary)
  - Client development
- Client Demonstration laptop/server – [TBD]