Distributed Aircraft Maintenance Environment
DAME

Pattern Matching against Distributed Datasets within DAME

Andy Pasley
University of York

Overview

- Distributed Aircraft Maintenance Environment (DAME) project
- Vibration data and search problem
- AURA strategy
- Architecture and storage
- Demonstration using signal data explorer
- Future challenges

Project Partners

EPSRC Funded, £3.2 Million, 3 years, commenced Jan 2002.

UK pilot project for e-Science

4 Universities:

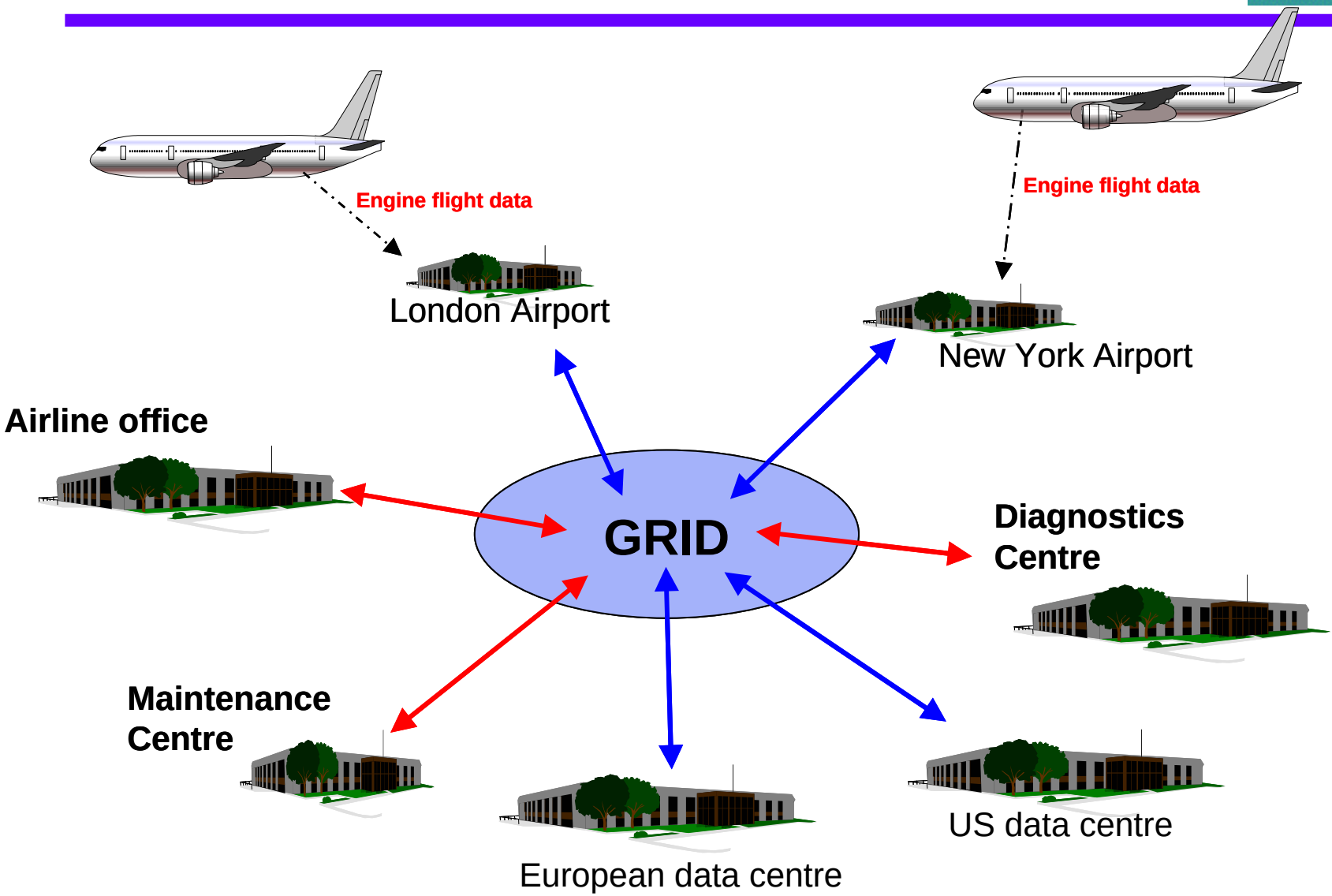
- University of York, Dept of Computer Science
- University of Sheffield, Dept of Automatic Control and Systems Engineering
- University of Oxford, Dept of Engineering Science
- University of Leeds, School of Computing and School of Mechanical Engineering

Industrial Partners:

- Rolls-Royce
- Data Systems and Solutions
- Cybula Ltd

Distributed Aircraft Maintenance Environment
DAME

Operational Scenario



DAME Grid Challenges

Building a demonstration system as proof of concept for Grid technology in the aerospace diagnostic domain

Two primary Grid challenges:

- Management of large, distributed and heterogeneous data repositories
- Rapid data mining and analysis of fault data

Other key (commercial) issues:

- Remote, secure access to flight data and other operational data and resources
- Management of distributed users and resources
- Quality of Service issues (and Service Level Agreements)

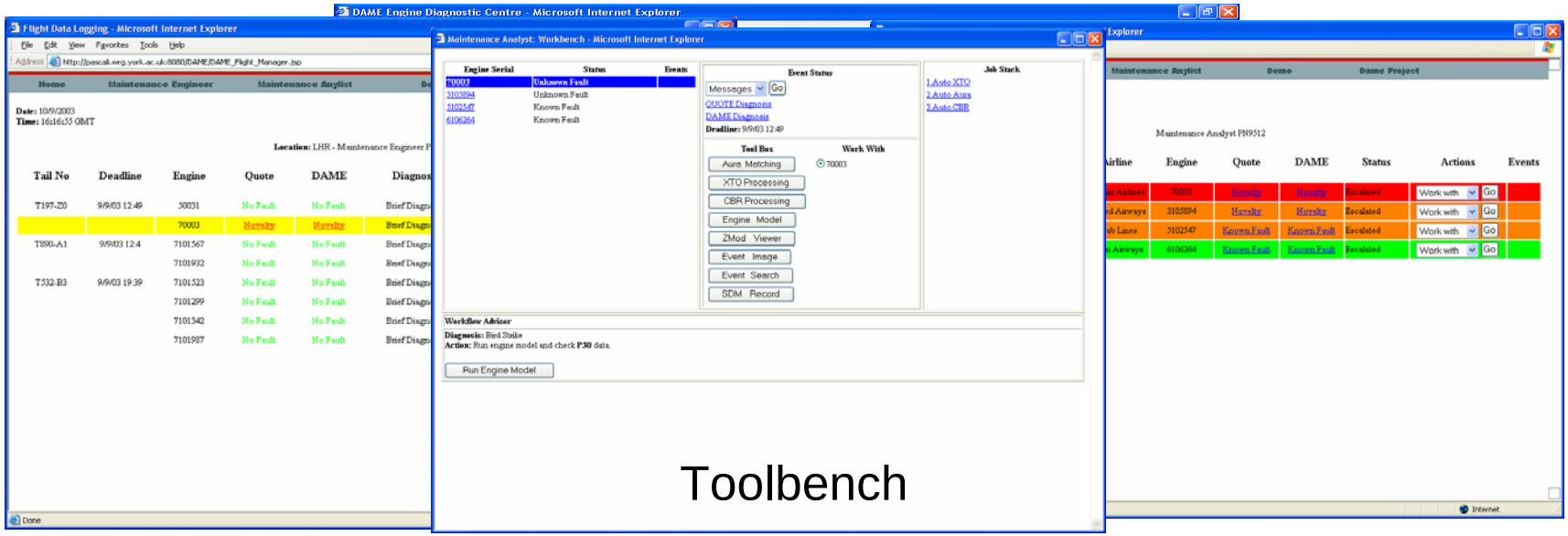
Demonstrator

Fully operational system on the WRG

- Demonstrated the basic system architecture and main services

Maintenance Analyst

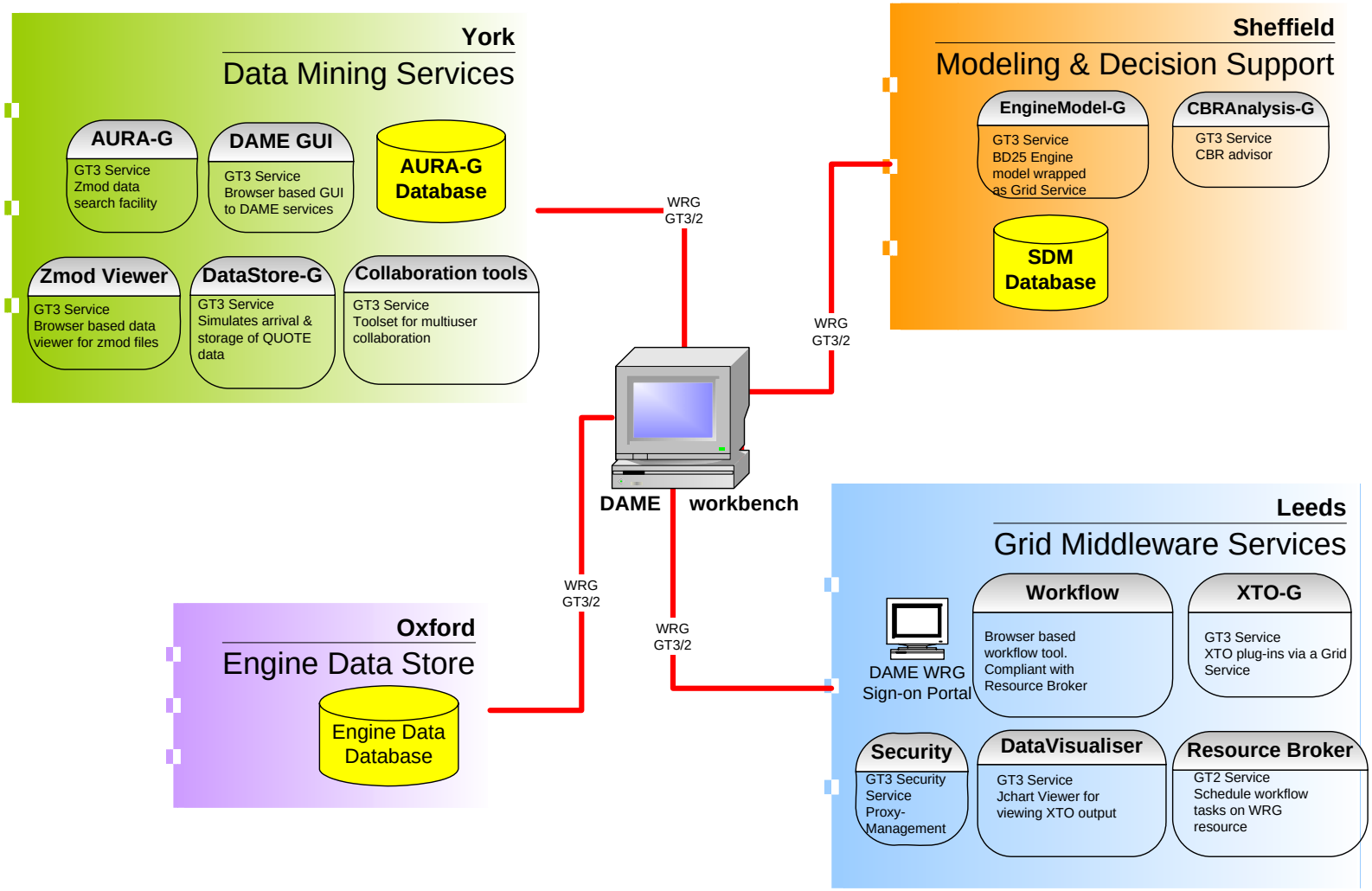
Maintenance Engineer



Toolbench

Distributed Aircraft Maintenance Environment
DAME

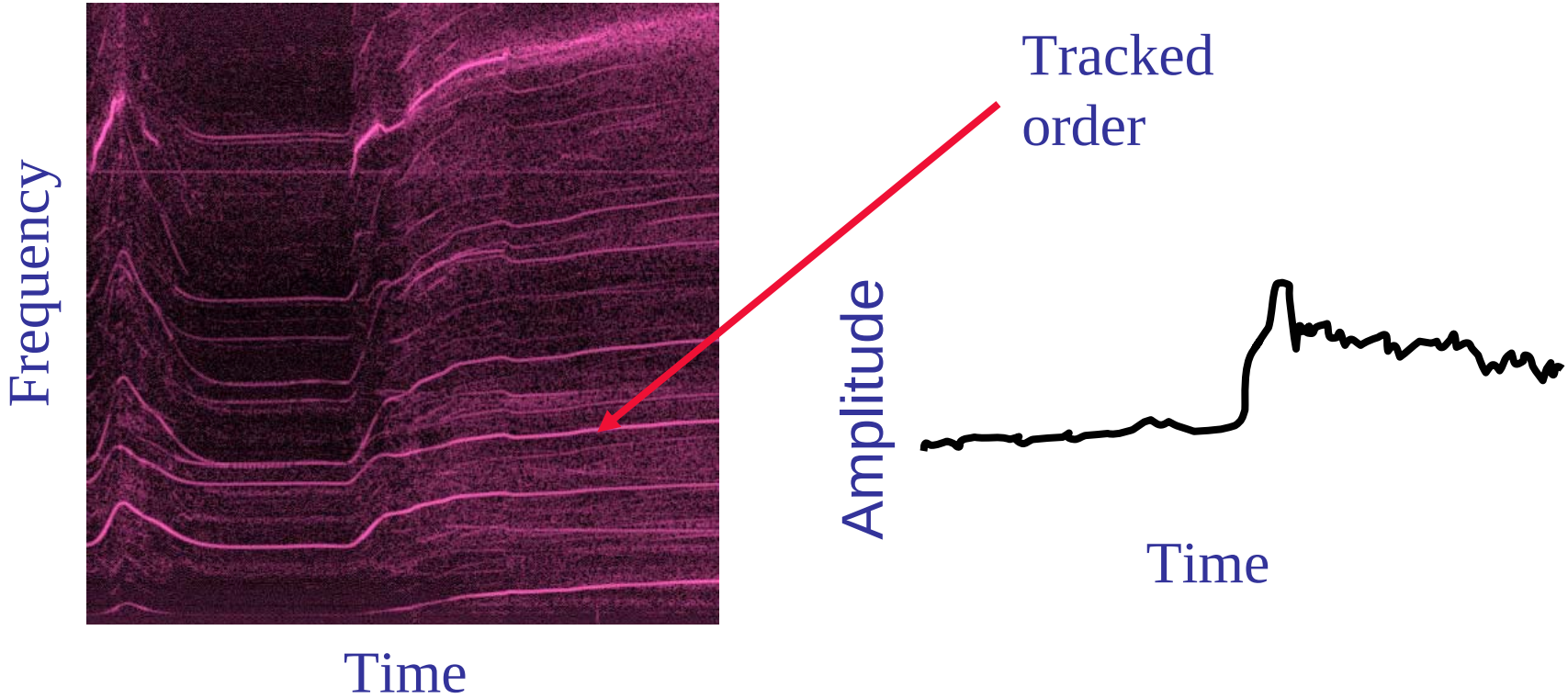
White Rose Grid Distribution



Vibration data and search problem

Z-mod Data

- Vibration data from sensors forms Z-mod data.
- Tracked orders extracted from Z-mod data

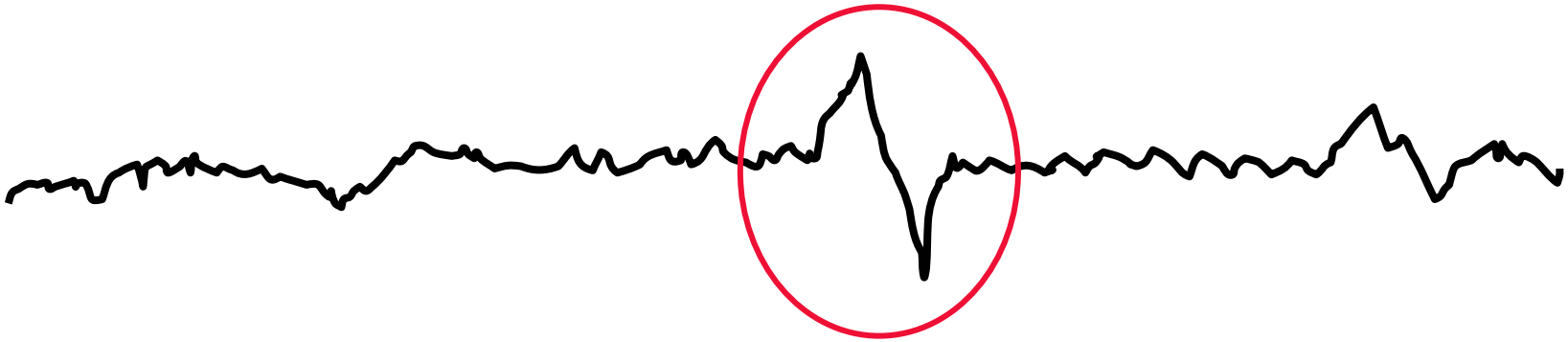


Pattern Matching Problem

- Collected vibration data from all engines in flight
- *Detect unusual events on recent flights*
  - QUICK on wing statistical classifier system
- ➢ *Search for similar events on other engines*
  - Uses AURA pattern matching methods to search large vibration data sets
- *Reason using historical data and search results*
  - CBR tools which access service records

Novelty

- Novelty or anomaly identified in tracked order data by QUICK



Forms ‘query’ sub-sequence

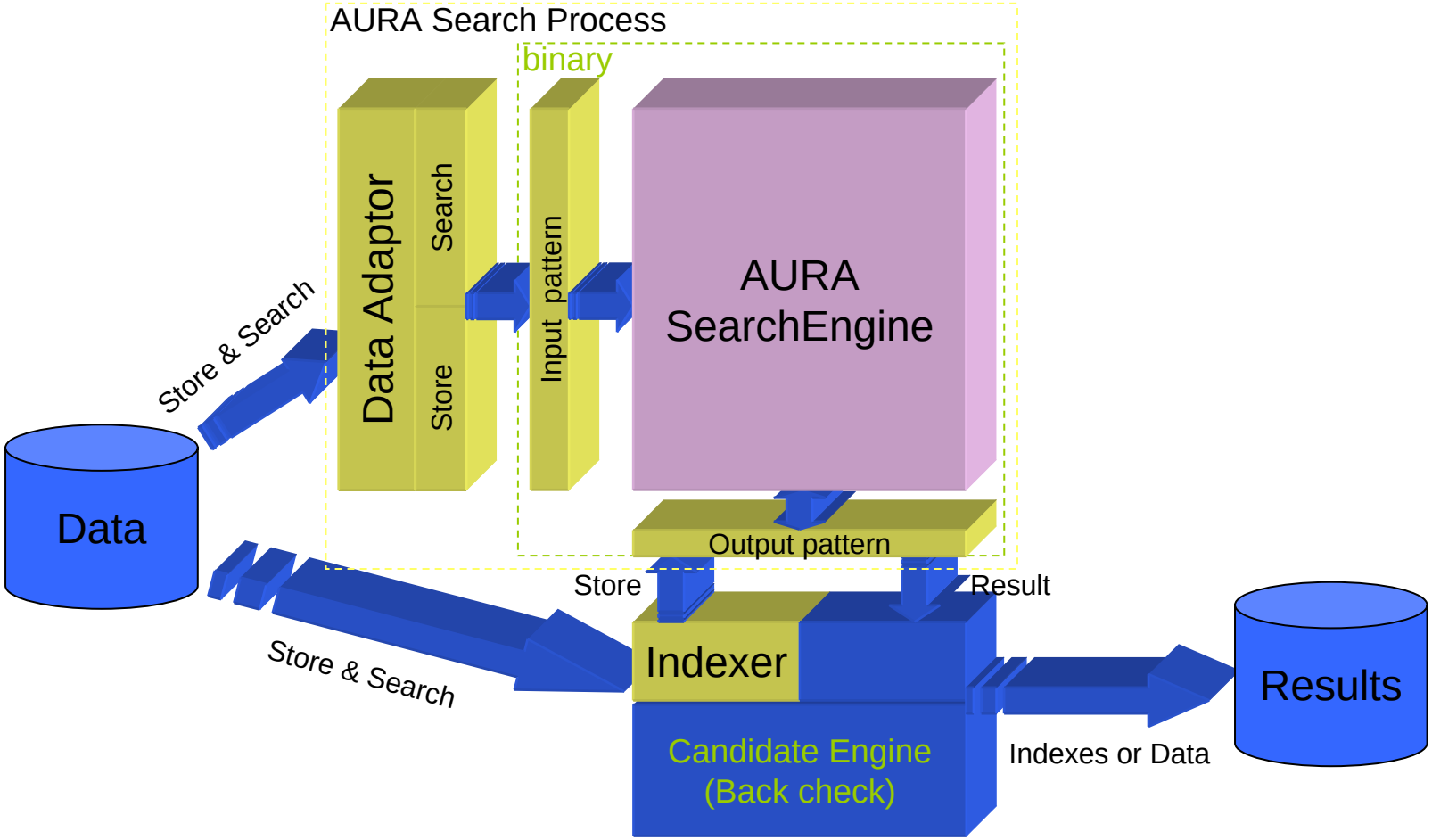
Search Problem

- Search for sub-sequences similar to the query in a large volume of tracked order data.
  - Need to investigate all possible alignments
  - Benchmark method is sequential scan
  - Noisy data: imprecise matching required
  - Various possible similarity measures
    - Euclidian distance
    - Correlation

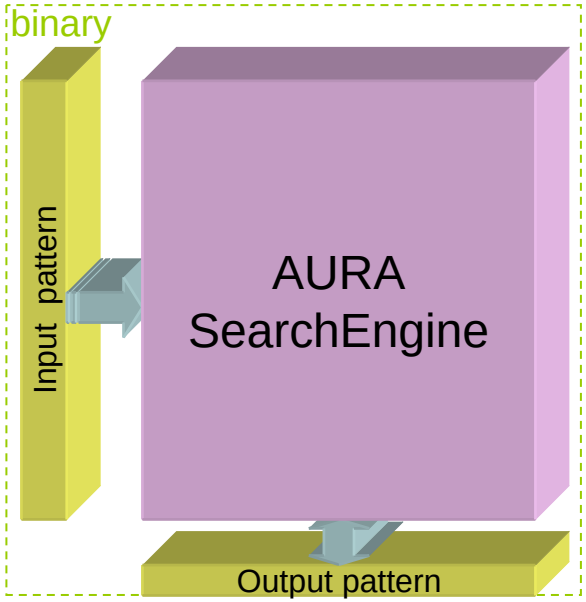
AURA

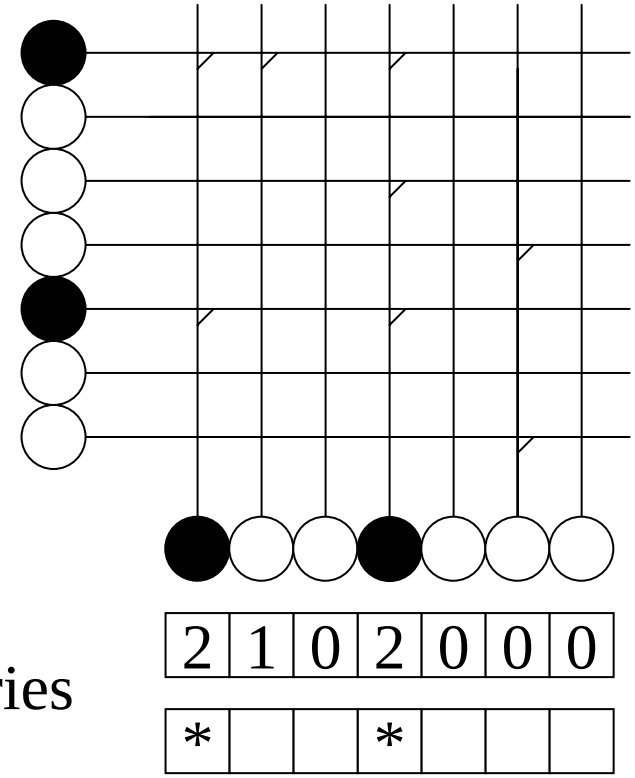
- Family of generic techniques for pattern matching using Correlation Matrix Memories (CMMs)
- Proven technology for searching large data sets
  - Sclable high performance
  - Find exact and near-matches
  - Wide range of data types
  - Can be parallelised
- Operation
  - Takes vectors and compares them to stored examples
  - Uses bit level comparison methods and binary matrix operations.

AURA Technology



Data Storage & Recall





Correlation Matrix Memories

AURA Encoding

- Application specific encoding required for efficient searching
  - Similarity metric
  - Integer ‘bins’
  - Reduction in dimensionality
  - Can integrate traditional methods

Performance

- Fast method of discarding poor matches
- AURA search roughly 30x faster than sequential scan
- Candidate matches typically <1% of total
- Back check stage very efficient due to reduction in volume of data
  - Typically 1% or less of processing time for full sequential scan.

Architecture and Storage

- Terabytes per year of raw zmod data
  - Access is required by many DAME services
- 1Tb per year of tracked orders that need to be searched against
  - Access required by Signal Data Explorer
- Observed in a distributed manner
  - Delivery to a central repository makes high bandwidth requirements

Objectives

- Distributed search
  - Transparent
    - Distribution of search and collation of results
  - Efficient
    - Use of processing and communications resources
  - Extensible
    - Permit addition / removal of resource
  - Concurrent
    - Support multiple simultaneous searches

Objectives

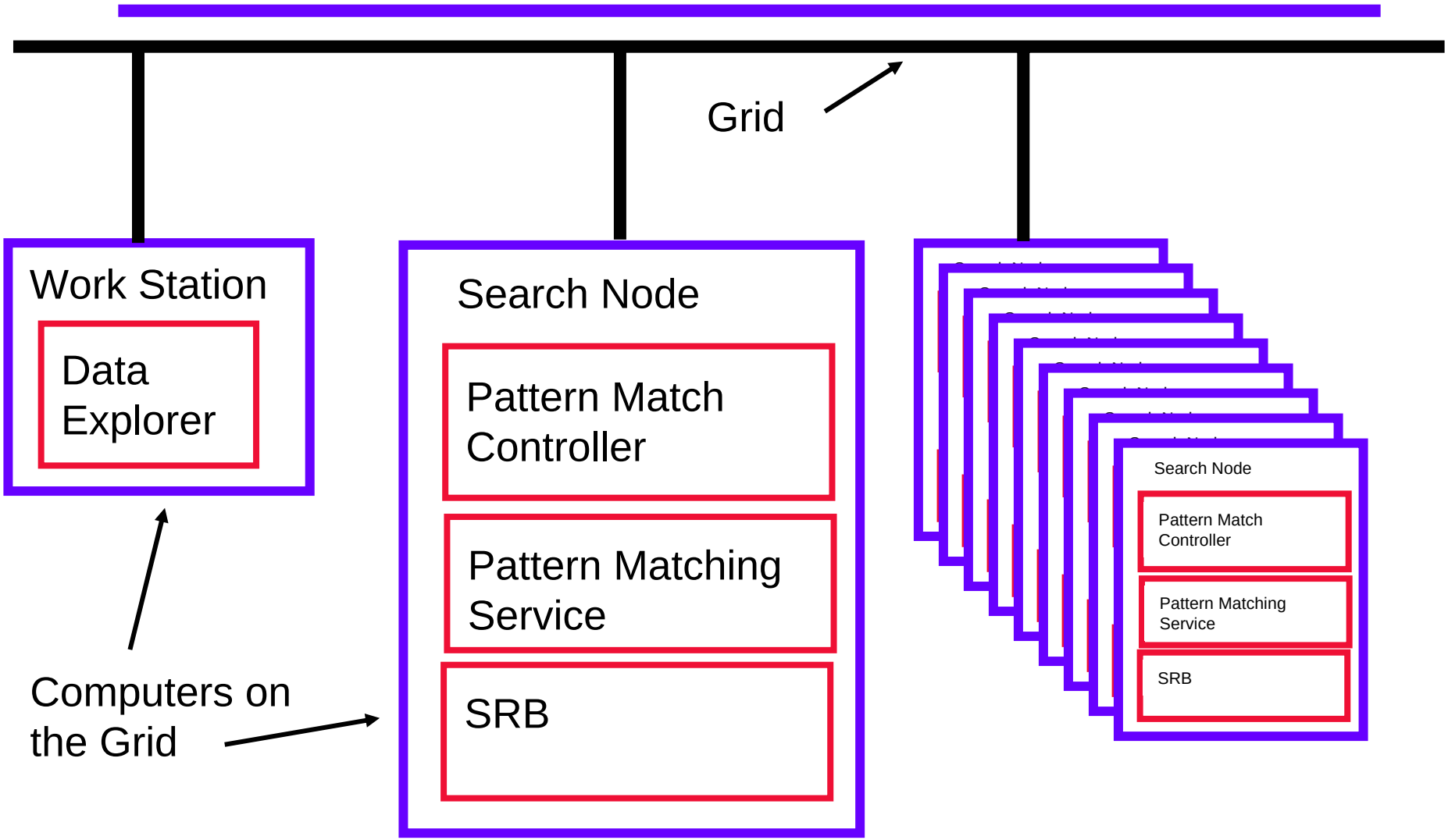
- Generic mechanisms
  - Suitable for different types of time series data and a variety of search methods
- Robust architecture
  - Graceful degradation when some components unavailable
  - Provision of intermediate results before all searching completed

Design

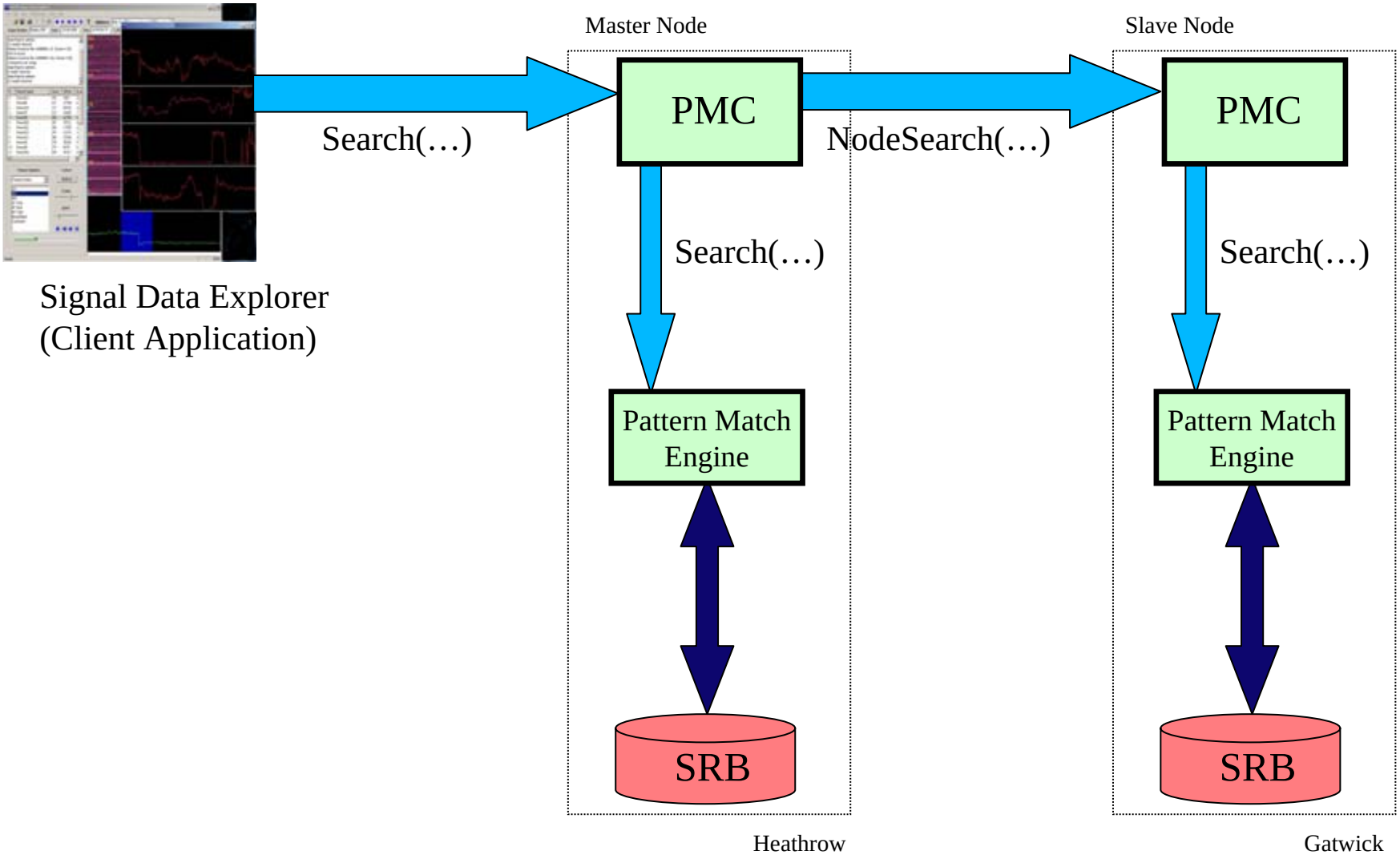
- Pattern match controller (PMC) service
  - Controls distribution and collation of the search
  - Generic service
  - Simple interface
  - Minimal communications overheads
- Pattern matching service
  - Performs the search
  - Can be implemented in a variety of ways
  - Conforms to a simple API
- Storage resource broker (SRB)
  - Used to store and retrieve data and metadata
  - Provides a single logical view onto all stored data

Distributed Aircraft Maintenance Environment
DAME

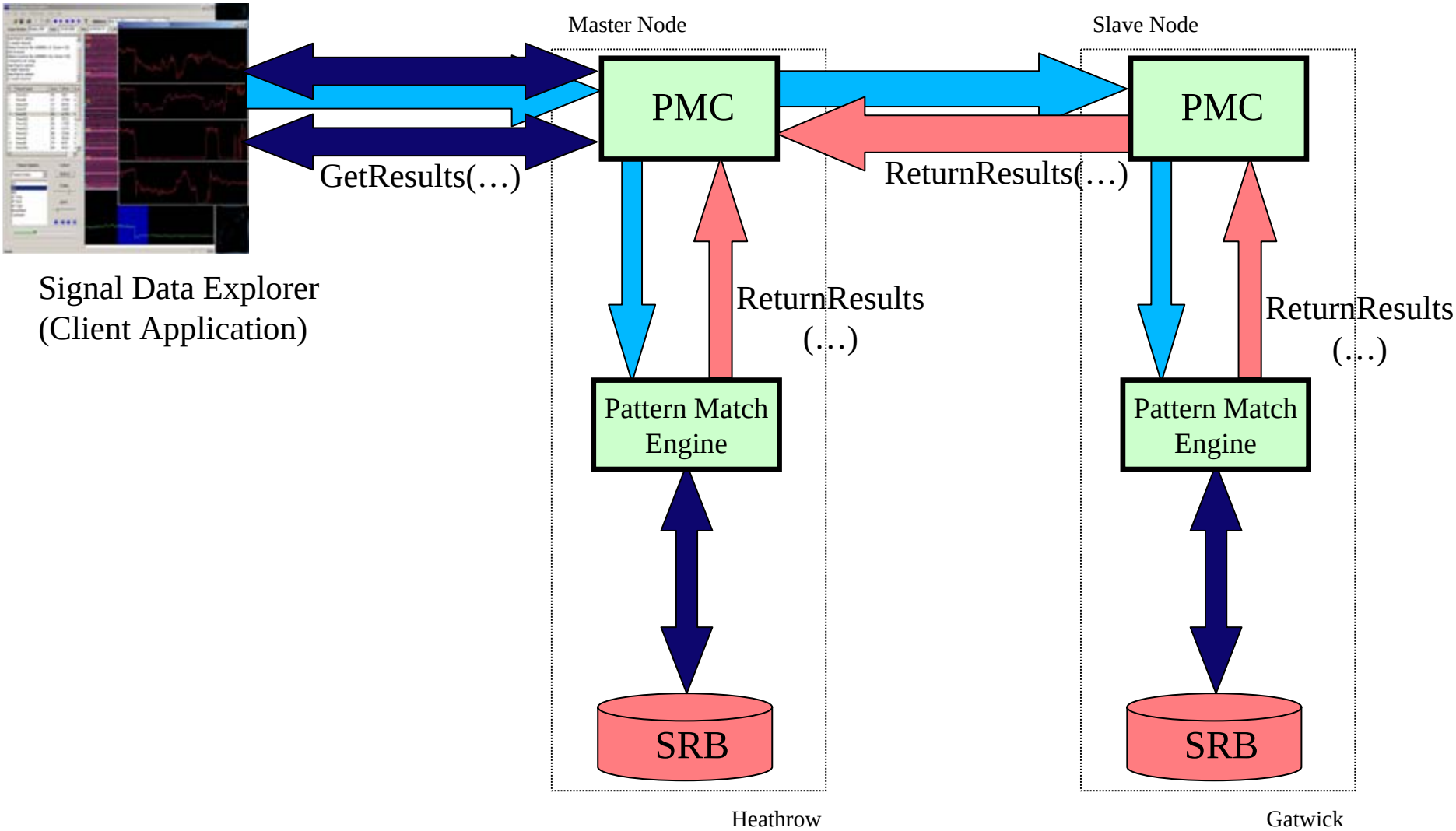
Physical Architecture



Search Process



Search Process

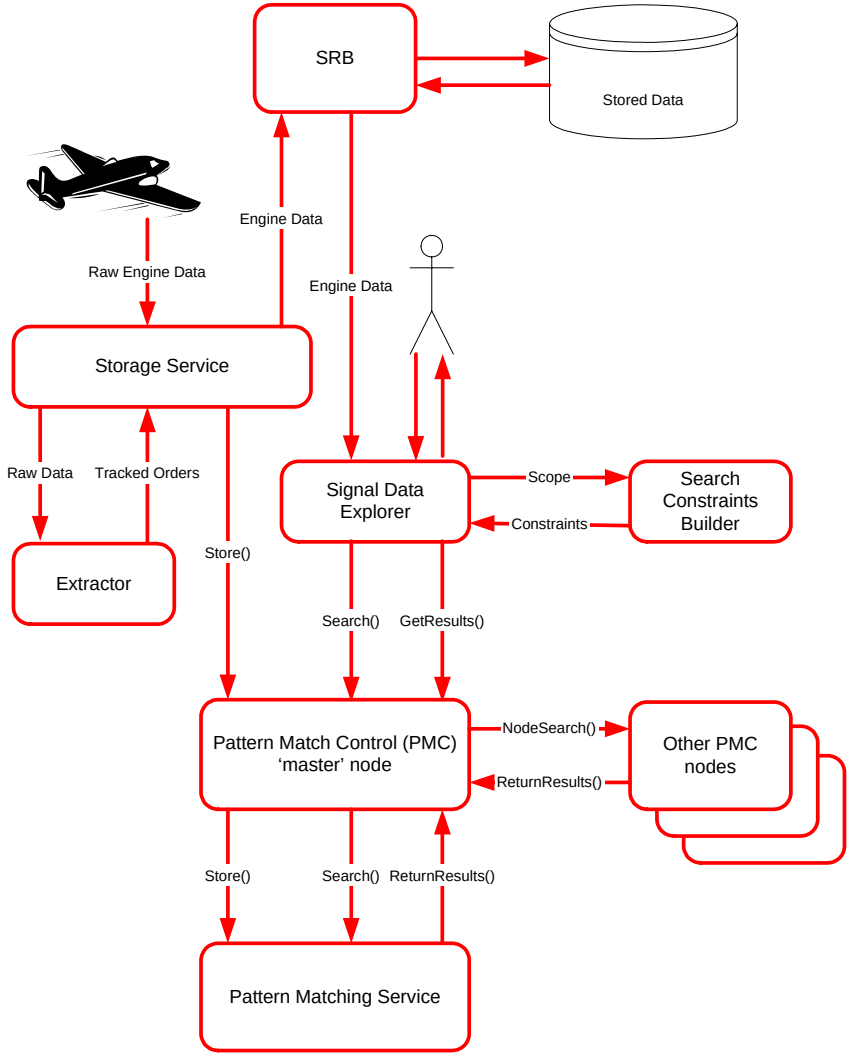


Implementation

- PMC
  - Java GT3 Grid service
  - Hosted within a Tomcat 4.1.24 installation
- Pattern matching service
  - Communicates with PMC using proprietary encoding
  - Uses SRB client library to access data
- Storage resource broker (SRB)
  - SRB server running at all WRG sites
  - Single metadata catalogue (MCAT) hosted at York

APIs

- Client developers need only use simple store(), search() and getResults() API calls.
- Pattern matching service developers need only implement a simple interface of search() and store(), and use the returnResults() API call.



Summary

- Transparent ✓
  - PMC distributes search to multiple pattern matching services.
  - Results collated and returned to Data Explorer
- Efficient ✓
  - PMC has minimal overheads
  - SRB handles used to identify results – minimal communications bandwidth required for search

Summary

- Extensible ✓
  - PMC uses a distributed catalogue of other PMC locations
    Permits simple addition/removal of search nodes
- Concurrent ✓
  - PMC uses unique search ids based on ‘master’ PMC id
  - Results kept for a time to allow access from other workstations

Summary

- Generic mechanisms ✓
  - PMC interface independent of type of time series data searched or algorithms used
  - Generic SRB handles used to identify data to search and results
- Robust architecture ✓
  - High availability as clients may use any PMC node as ‘master’ for a search.
  - Temporary results built up and may be accessed before entire search complete
  - Partial results in event of unavailable nodes
  - Automatic clean-up after timeout

Signal Data Explorer

- Tool to allow investigation of data outside of an automatic workflow by a domain expert
- Accesses local data stores or remote (distributed) data sets and searching services.
- For this demo, searches against data held on the White Rose Grid at York, Leeds and Sheffield

Next Steps

- Scaling trials on engine data
  - Realistic number of concurrent users supported
  - Investigate performance as the number of nodes and/or volume of data is increased
- Compare overhead in search time / network requirements to a centralised architecture
- Federated MCATs
  - Create several SRB ‘zones’ each with a metadata catalogue

Complex Search Requests

- Search Requests
  - May contain several query patterns to be matched against. The results for one pattern may constrain the search space for another query pattern
  - If treated as several individual queries may not be processed efficiently

- A ‘result’ may consist of data stored at more than one node
  - Slave nodes may be required to issue ‘sub-searches’ to other nodes in the system