

COPPEREYE RETENTION SERVER



DATA CHALLENGES FOR IT IN 2010

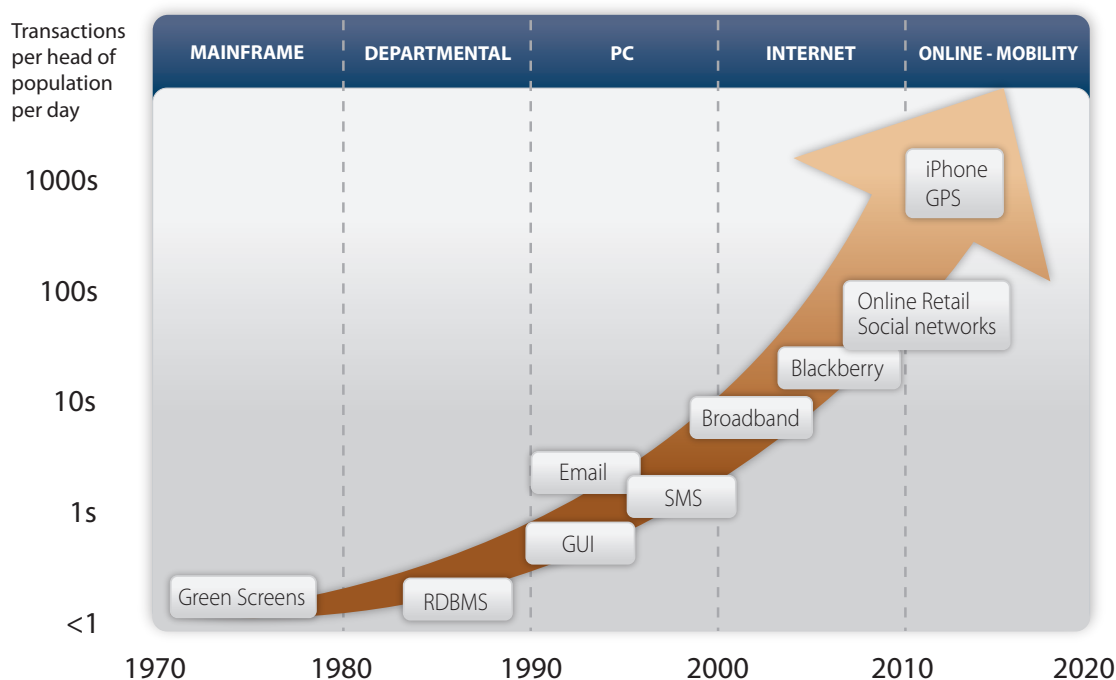
It is a universally accepted fact that data continues to grow with no bounds. The ability of IT systems to store data has evolved to the stage where their users do not delete data and regulation has left people with a fear of deleting anything anyway.

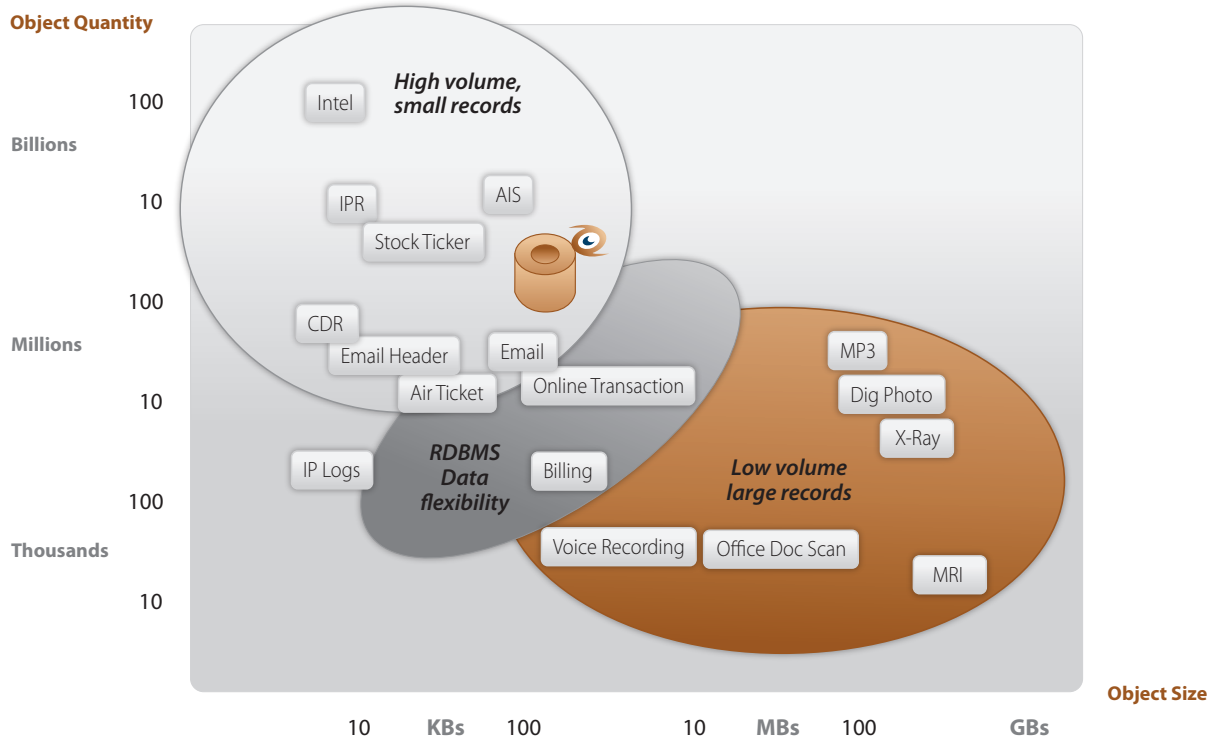
The fastest growth of data objects is now driven by the provision of ever more online devices and their proliferation from data centers, through departmental systems, to the personal computers. The mass acceptance and usage of Smartphones is causing the latest inflection point in the exponential rise of data record creation by individual citizens.

Very simply, the more users of online systems, the more records that are created and need to be retained and available, though rarely, if ever, accessed. Every online action creates data and the biggest volume is of very small unchanging records that grow into the billions.

Storing billions of data objects is no longer an issue. The cost of large storage systems have become almost irrelevant in real terms compared to their real cost 10 or even 5 years ago. The real IT challenge with this data is how to load, manage, order and retrieve a few items amongst 100s of billions of records.

CopperEye's core technology was designed to enable a quantum shift in the number of records that can be managed in this emerging hyper-volume world.





The figure above demonstrates the range of data structures that now proliferate in the online world and maps their average size to the number of records that are created.

RDBMS first became prominently used in the late 1980s. They were designed to enable the connection of data through its inherent relationships and the fast amendment of data design and development of applications. This led to the introduction of departmental computing and the total acceptance of RDBMS technologies as the platform of choice for application developers – from the departmental team to SAP. The RDBMS has proliferated and is omnipresent. It has been a wonderful success.

As technology solutions have developed over the last 20 years the underlying RDBMS technology has not fundamentally evolved. It basically managed data volumes the same way it always has. Hardware technology has followed Moore's Law, it has enabled significant RDBMS performance gains and supported the widespread acceptance of RDBMS based applications.

All data types came to be stored in RDBMS, regardless of whether they met the requirements of the relational model. A good example of this would be Binary Large Objects (BLOBs). They have no relational structure, use up very large quantities of RDBMS workspace and licensing for little application value. Over time this has been recognised as a shortcoming and technology (typically very large storage subsystems) have been evolved to offload BLOB technology – the best example of this is Healthcare Imaging – X-Rays, MRI Scans etc etc.

The corollary of this is the very high volume small record. These objects have little relational connectivity and place enormous loads on the RDBMS through the indexing demands on this data. Typically, in an enormous data store of small records, only a tiny subset of the data is accessed by any report – this is a dichotomy that the RDBMS technology is unsuited to address and is a perfect match for CopperEye. As data volumes grow (which they will) and the number of small records grows (which is implicit), CopperEye potential market grows exponentially.

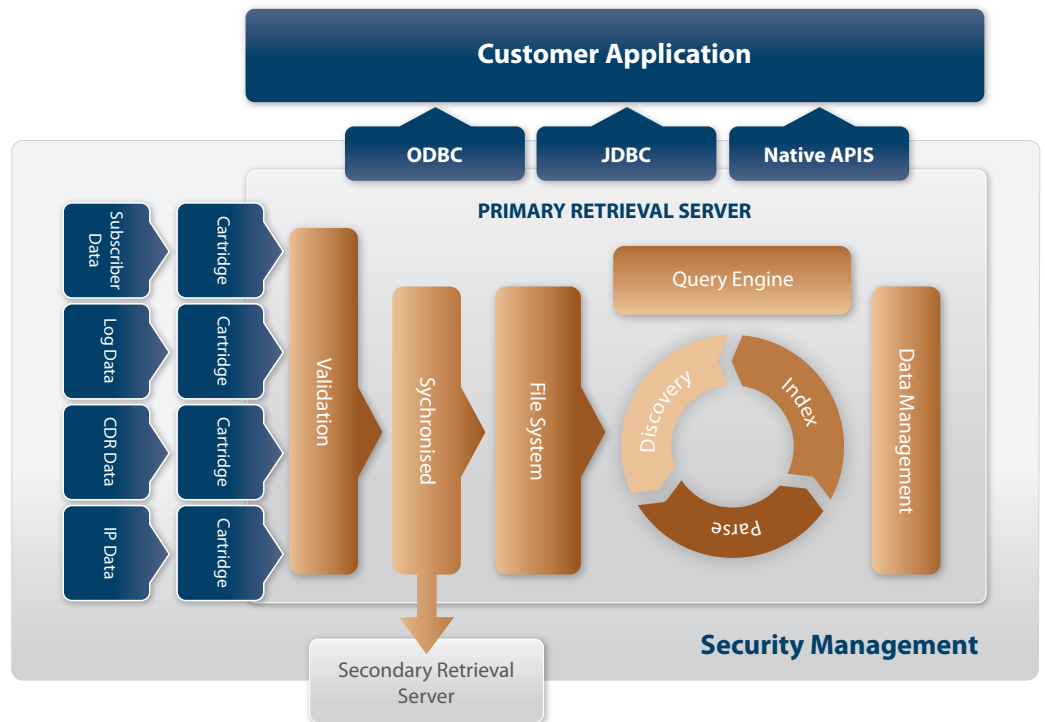
COPPEREYE RETENTION SERVER

CopperEye Retention Server (RS) is a full function data management solution that provides:

- Automatic data loading and indexing
- Full lifecycle data management
- Full admin and audit functions
- SQL Reporting Interface to external applications
- Industry Standard Data Interfaces to external world

CopperEye RS provides the solution developer with the technology to build a new generation data store for mass volume small, static records with limited relational connectivity – in reality the fastest growing area of data management and the one least suited to the technology offered by the current generation of RDBMS

CopperEye RS can load, index and maintain any type of data record and integrate to any application or alternative data store. In reality CopperEye RS typically lives beneath applications previously built on RDBMS, or beneath the RDBMS itself offering more appropriate data management than is possible with the RDBMS technology.



WHAT DOES IT DO?

CopperEye RS is a high-performance, multi-threaded data indexing and query engine that was developed to give the user timely and rapid access to flat data files.

DISCOVERING

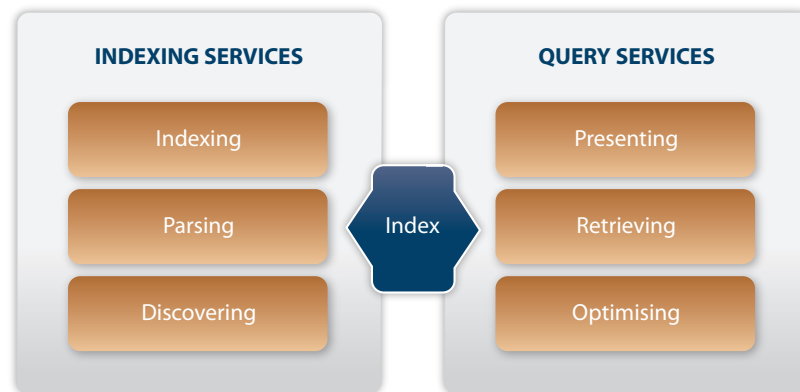
CopperEye RS discovers new data files as they appear in target directories and automatically updates its data management repository. Discovered files are then parsed and indexed.

PARSING

CopperEye RS parses newly discovered files and processes them according to their associated format definitions and schedule. File definitions may be added and updated dynamically in response to changing file formats and visibility requirements.

CopperEye RS supports multiple file parsing conventions, including:

- ASCII and binary file formats
- Fixed and variable format records
- Fixed, variable and delimited fields
- A rich and extensible portfolio of field data types



INDEXING

As CopperEye RS parses a data file, it indexes fields to allow optimal retrieval of records at query time. Each index records the field value together with the file identity and record offset allowing the contents to be retrieved directly in a single disk I/O. The original data is maintained and can be stored where the user desires.

CopperEye indexing technology supports extremely efficient insertion and query operations without the need for high-performance disk subsystems. In fact, the technology has been benchmarked at 100 times faster than conventional B-tree indexing for index insertion operations – allowing many more indexes to be supported without undue impact on performance. More indexes mean more query flexibility and efficiency. It makes a new generation of application possible.

Indexes are stored as binary files in a nominated location on the file system. A single index file can routinely scale up to many hundreds of millions and billions of record entries without impacting performance. Automated

index file partitioning guarantees complete scalability regardless of retained volumes and enables enormous retention ranges.

The algorithms within CopperEye indexing technology require much less disk I/O than are demanded by conventional index structures, such as B-tree indexes. Therefore, insertion operations do not suffer from the traditional disk I/O bottlenecks often associated with conventional indexing.

The speed and performance of CopperEye indexing means that both data and index files can be hosted on commodity based, inexpensive disk storage systems.

With rapid index insertion rates, there is minimal latency between data files appearing on the file system and the availability of query access. It's important to note that CopperEye RS delivers immediate access to the data because CopperEye does not deny query access while new files are loaded.

OPTIMISING

When a query is performed, CopperEye RS chooses the most appropriate index or indexes to use according to the path of lowest cost based on the disk I/O operations. Each index dynamically maintains statistics internally – providing a true and up-to-date cost of each possible query without the headache typically encountered with traditional RDBMS systems, which require statistics to be separately updated as a regular maintenance task. CopperEye RS can combine search results from multiple indexes to maximize the selectivity of the search and minimize the number of disk I/Os to return the final result.

RETRIEVING

While servicing a query, CopperEye RS retrieves records from the underlying files using the file identities and record offsets returned by each index. Therefore, each record is directly and efficiently retrieved in a single disk I/O using the record offset supplied by the index. CopperEye RS routinely sorts the file identities and record offsets prior to retrieval to ensure the most efficient exploitation of available file system buffering.

CopperEye RS is designed to service selective queries (returning a small percentage of records) from a massive data population. These queries will typically retrieve granular detail data rather than aggregate data used for

analytics. Because of this, CopperEye RS can be deployed to complement aggregation and analytic systems in a drill-down/drill-through role to avoid the impracticality of storing minute detail in the analytical database.

PRESENTING

CopperEye RS returns records and fields from a query in rows and columns in a relational view over ODBC/JDBC. This mapping is part of the overall solution configuration and can be changed in response to file format changes and new file visibility requirements. A record can appear in multiple views if requested, and each view can be defined to display all fields or a subset of the fields in each record. Multiple record types can also be normalized into a single view.

When submitted to CopperEye RS, queries return results as though the data were stored in relational tables. The structure of these relational views can be queried through the data dictionary – just as in conventional relational database – allowing users to actively discover the data available to them and ensuring all access to the underlying CopperEye RS is transparent to the user and application.

UK Corporate Headquarters

CopperEye Labs Ltd, 1st Floor, 1 St Andrew's Hill, London, EC4V 5BY, UK
t +44 (0) 1454 203 338 f +44 (0) 1454 203 330

e contact@coppereye.com
w www.coppereye.com