InterSystems IRIS Data Platform

An Architecture Guide

InterSystems data technology is known for unmatched performance, scalability, interoperability, and reliability. What makes it so fast and flexible? It starts with a unique architecture.

This paper describes the InterSystems core data platform architecture, what the "special sauce" is, and why the architectural approach delivers the highest performance and resiliency with the lowest TCO.

InterSystems®
Creative data technology

The InterSystems IRIS data platform underlies all InterSystems applications, as well as thousands of customer and partner applications across Healthcare, Financial Services, Supply Chain, and other ecosystems. It is a converged platform, providing transactional-analytical data management, integrated interoperability, and data integration, as well as integrated analytics and AI. It supports the InterSystems Smart Data Fabric approach to managing diverse and distributed data. A high-level summary of the strengths of this data platform is shown in Figure 1.

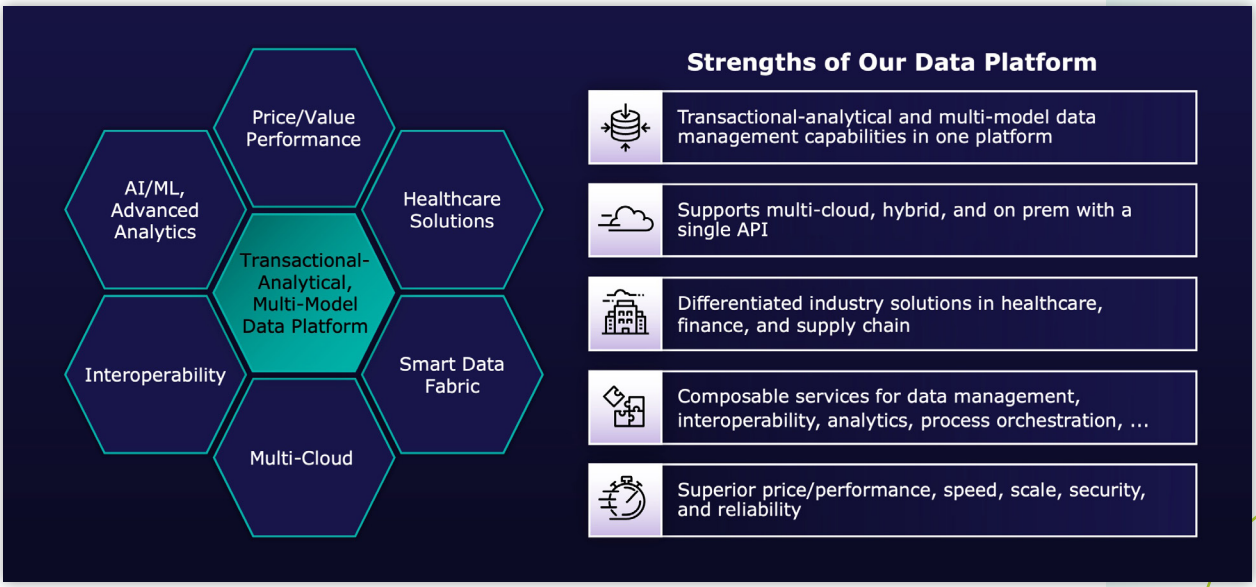


Figure 1 – High-level Overview of InterSystems Data Technologies

The Problems We Solve

To get a flavor of the kind of problems solved by InterSystems data technologies, consider some examples of scale:

- Process **2B+** real-time equity trades daily.
- Manage **>1B** patient records worldwide.
- Real-time tracking **20M+** shipping containers globally

InterSystems IRIS data platform is often used for demanding, high-performance data intensive applications. Examples include:

- Ingesting and analyzing **2.5M** real-time events per second for a financial services data provider in a core commodity-trading application
- Processing **500M+** database operations per second for a large Healthcare software company, managing the largest collection of healthcare data in the world
- Supporting **40K** simultaneous power users for a large Healthcare Provider
- Replacing a variety of data management services with a single, converged implementation of our InterSystems IRIS data platform

Many applications are served by InterSystems in highly regulated contexts, where security, reliability, and compliance are essential. Many are often complex, high scale, and high performance. The key to serving this variety of applications at high scale and high reliability is found in the underlying architecture.

InterSystems IRIS Architecture

Our architecture is built in five layers, as shown in Figure 2.

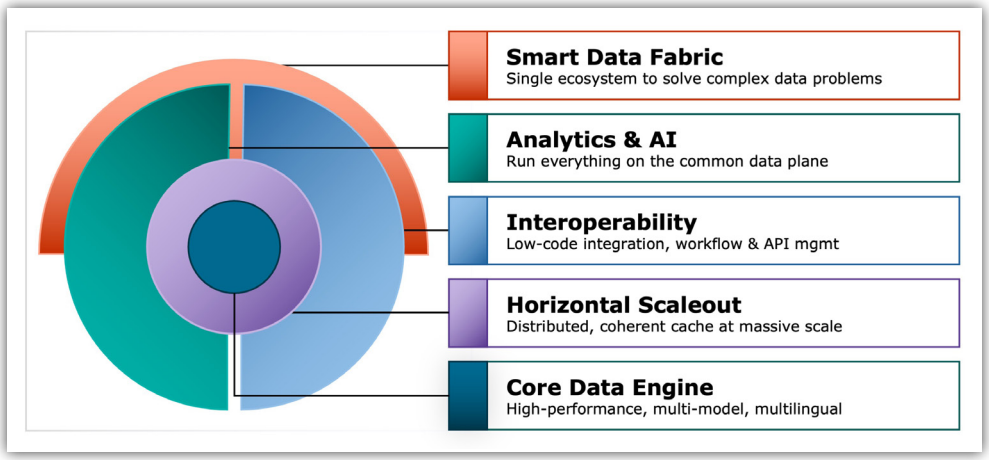


Figure 2 – Underlying Architecture of InterSystems IRIS

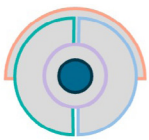
At the core of our architecture are facilities for high-performance, multi-model, multi-lingual data processing in our core data engine, also known as the **Common Data Plane**. Around that lives a remarkable facility for scaling out extremely high volumes of data and high transaction rates that can reach over a billion database operations per second.

Next are two major subsystems: one that focuses on analytics and artificial intelligence (AI) and another that focuses on interoperability and data integration. These subsystems follow our fundamental philosophy of running everything close to the data to provide high performance with a minimal footprint.

Finally, around the subsystems, we have built a smart data fabric that enables customers to solve complex problems in a single stack.

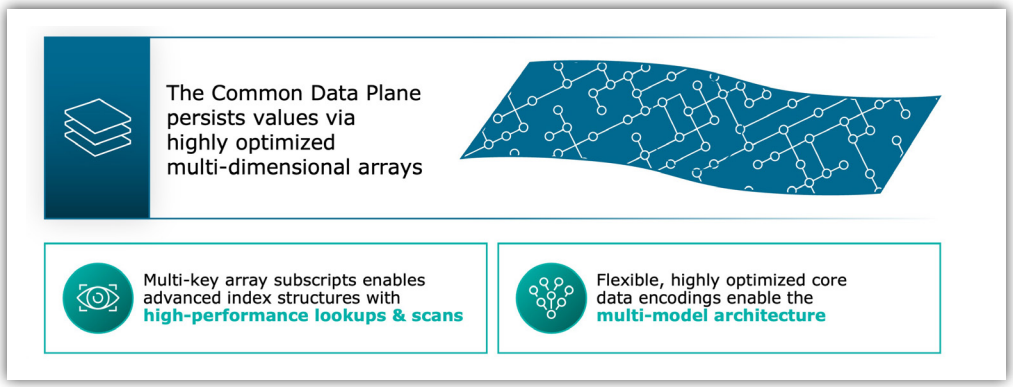
The following sections explore these layers and how they interact to give a better sense of what makes InterSystems IRIS technology so special.

The Common Data Plane: Natively Multi-Model with a Single Copy of the Data



Famous for its performance, the core of InterSystems technology is a highly efficient mechanism for data storage, indexing, and access. Unlike other database providers, we do not provide a natively relational or document database. We use an underlying storage format called **globals**. They are modeled in a highly optimized, multi-dimensional array-style format that is built as a B+ tree that is automatically indexed with every operation.

Built at a layer below data models—such as relational, object, or document—a single storage format is projected into different data formats and models. This is referred to as a **Common Data Plane**.



The underlying **global format** is highly efficient and translatable to many different data models, as shown in Figure 3. Globals (denoted with an up-caret **"^"** prefix) can have many subscripts, each of which can be numeric, alphanumeric, or symbolic. Globals are powerful and represent data in a general way that simultaneously supports many data paradigms with a single copy of the data. Cases like associative and sparse arrays are easy to process in this approach.

We also encode in the storage format itself, using encodings (denoted with a dollar sign **"$"** prefix) that provide a small footprint and low latency because of the disk and I/O optimizations.

The format of these encodings is the same in memory, on disk, or on the wire. This minimizes the transformations involved in ingesting data and achieves amazing speeds expected from an in-memory database, but with persistence typical of a disk-based database.

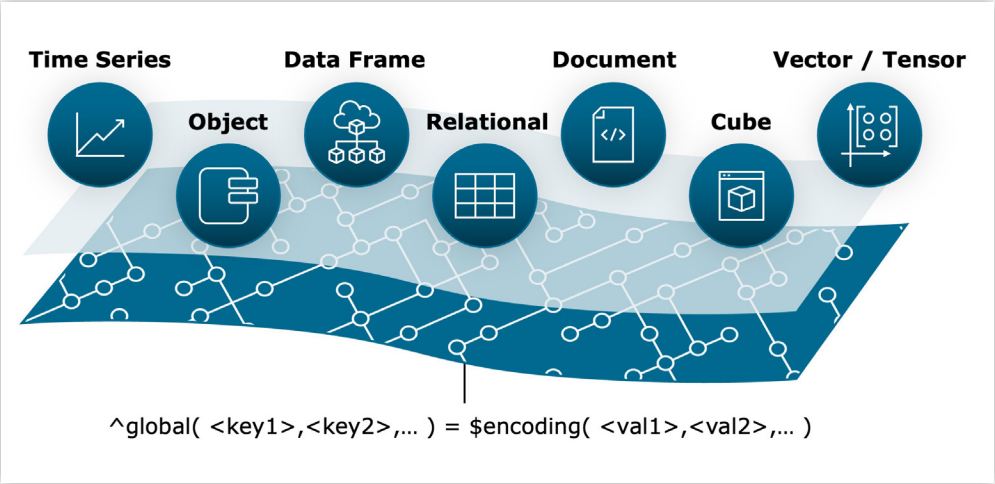


Figure 3 – Globals and Encodings

An example of how a single global to support multiple data models is illustrated by a case where you are using SQL or BI tools and want to access the data in a relational format, in tables with rows and columns. If you are doing object-oriented development, however, we automatically project those objects into globals and subsequently project that data into relational format. Similarly, we can project JSON or other document formats into a relational form.

This capability means that—rather than having multiple data stores, one relational, another object, and another document, and stitching them together—we have one copy projected to all these different forms, without duplication, moving, or mapping. From this also comes a convenient combination of schema-on-write and schema-on-read. As with a data lakehouse, you can depend on a level of structure like a data link, after inserting the data and figuring out the best schema for that data based on its current use. This global structure works well for structured data, as well as for documents and semi-structured or unstructured data.

A few encodings, engineered very tightly, are used to store data and indices efficiently, as shown in Figure 4. While lists are the default storage encoding, InterSystems IRIS may represent data and indices in one or more of these encodings based on the data characteristics and/or the developers' specifications. Vectors store a large number of the same datatype efficiently and are used for columnar storage in analytics, for vector search, for time series, and for more specialized cases. Packed-value arrays (known as **$pva**) are ideal for document-oriented storage. Bitmaps are used for Boolean data and for highly efficient bitmap indices.

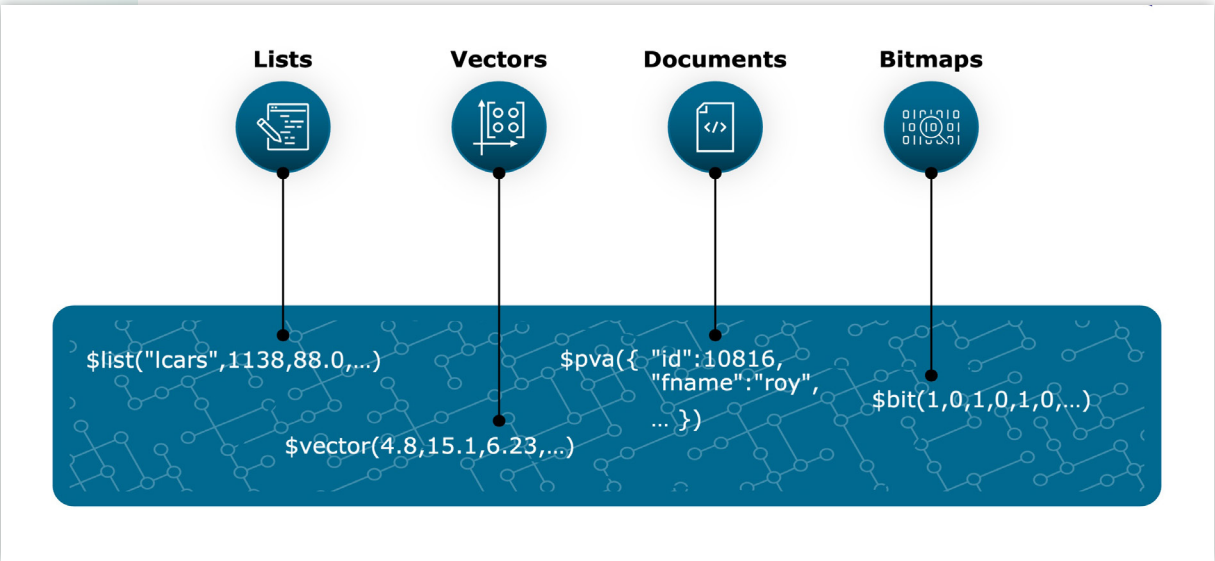


Figure 4 – Key Encodings

All these data structures are automatically indexed in a highly optimized update upon every operation. Built-in indexing has been used by many successful customers to carry out low-latency, full-transactional steps, like the "billion database operations per second" mentioned earlier. Such consistent indexing, performed almost instantly, gives us consistent, low-latency access to all data in any format.

Multi-model facilities made possible by the underlying global format are virtually instantaneous because there is only one copy of the data to change, and thus no time or space needed for data replication. This also grants major advantages in ingestion speed, reliability, and scale-out. *Where most multi-model databases actually replicate their data internally, InterSystems IRIS does not.*

Encodings can be combined by the system. For example, Figure 5 shows medical device data that is transmitted in a document format (such as the commonly used HL7 v2 formats). This could include a set of metadata about the device (kept in a list encoding), an initial timestamp, and a long series of values (kept in a vector encoding). This data can be projected into different data models, for example, a relational model. These steps happen under the hood, so to speak. The user simply sees instant relational access to the data. An update to the table from SQL is reflected instantly in the document projection, and vice versa.

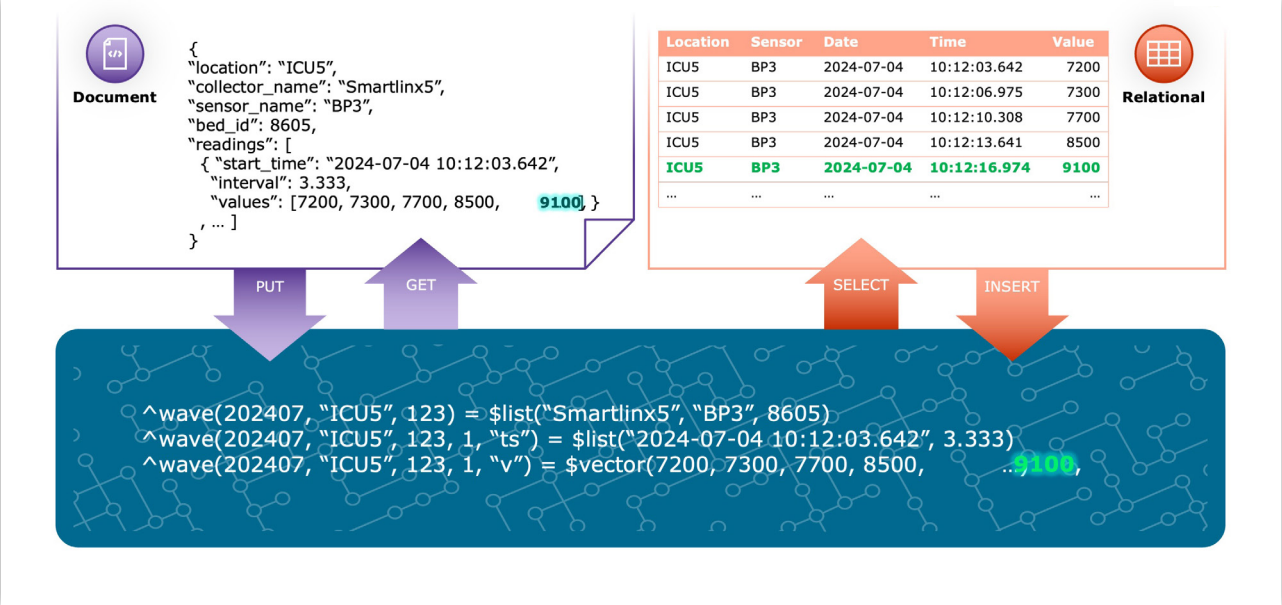


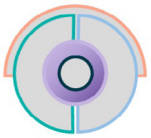
Figure 5 – Example Using a Compound Encoding

The multi-lingual capability that globals provide means that you can work in the programming language of your choice, with effortless access to all needed formats. Clearly true with relational access through standards like JDBC and ODBC, it is also true of automatic matching of objects in .NET or in Java to an underlying format. From a development perspective, you do not need to worry about the object relational mapping; you just work with an object, and we take care of the storage format.

Summary of the features and benefits of the common data plane

Feature	Benefits
Support for a wide range of data types and access methods	• Eliminates the need for multiple engines • Avoids data duplication, moving, mapping • Delivers automatic schema-on-read and schema-on-write
Consistently indexing all data automatically	• Enables consistent access to all data • Provides low latency and full ACID transactions
High ingestion and transaction speeds	• Captures incoming data at high speed for high throughput, real-time use cases • Performs 100s of millions of transactions per second sustained
Highly efficient use of storage	• Optimizes resource efficiency (disk, I/O) • Eliminates the need for data replication
Multi-lingual and multi-model	• Simplifies architecture and operations

Horizontal Scale-Out: Guaranteed Consistency with a Built-In Distributed Cache



Around the core data engine, is layered a distributed cache coming with built-in consistency guarantees. This cache uses our **Enterprise Cache Protocol**, or ECP, and satisfies textbook guarantees for consistency under distributed data and failure. As shown in Figure 6, ECP builds in these consistency rules to maintain data integrity across a distributed system even in the presence of failures, encapsulating them directly.

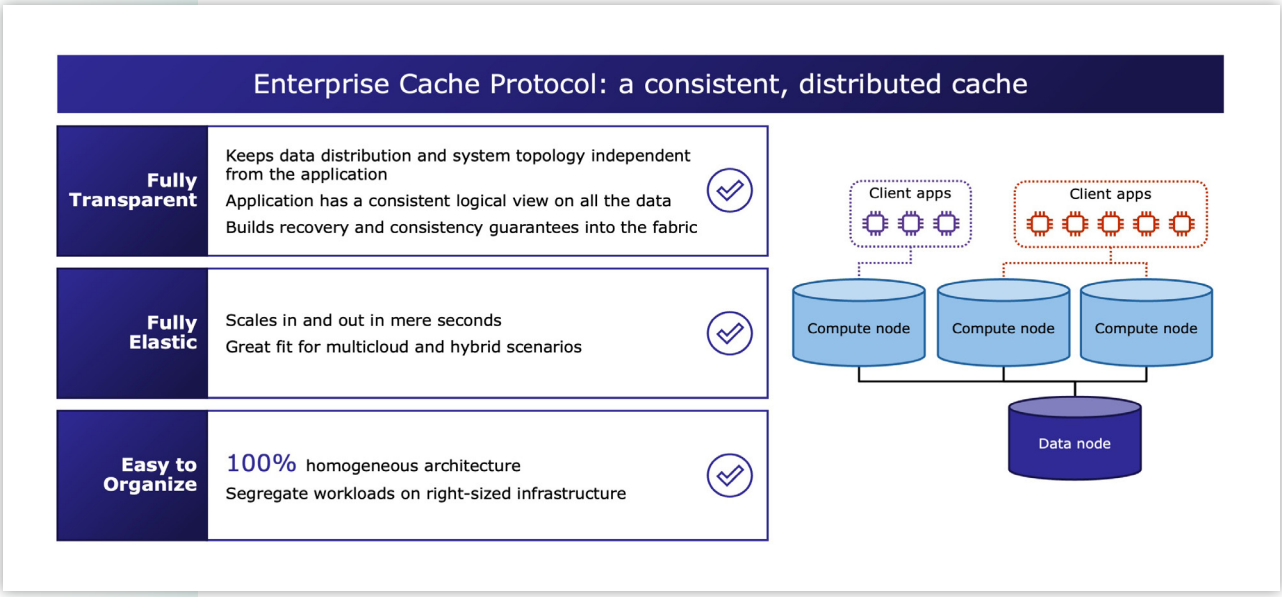


Figure 6 – Enterprise Cache Protocol (ECP)

In other words, the performance of the distributed data stays high, even at scale. You can spread these ECP nodes for horizontal scaling, managing higher throughput. You also can spread them for distribution of data, meaning that you can have in-memory performance without having to live within the memory that is available for any node.

One example of this scale-out is shown in Figure 7. This shows the benchmarked performance of a longstanding InterSystems customer who migrated from **InterSystems Caché** (our second-generation platform) to **InterSystems IRIS** (our third-generation platform) and applied ECP. This switch, along with the continual performance optimizations done by InterSystems, resulted in a throughput of over a billion global references within the database per second in the lab, allowing significant headroom over the current largest live deployment.

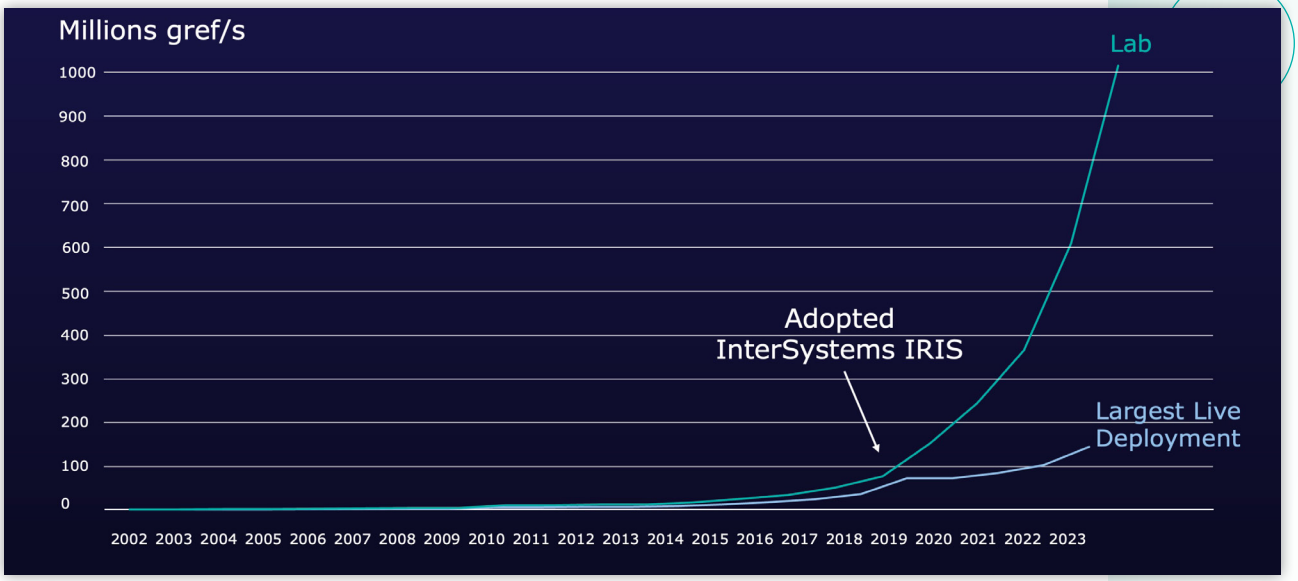


Figure 7 – Example: Customer Scale-Out with ECP

ECP works especially well in the cloud because of its scale-out. We've built that into our **InterSystems Kubernetes Operator** (IKO) to provide auto scaling, and we can add and remove nodes using ECP transparently to the application. Scaling out like this is essentially linear, and you can independently scale out the ingestion versus the data processing versus the data storage and optimize for your workload.

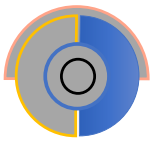
Because ECP is robust to changes in topology, a node can die without affecting transaction processing. You can add in nodes on the fly, and they can pick up the load. That provides seamless elasticity, which means that you can size things dynamically and thereby enjoy a net lower cost.

ECP is transparent to the application; no changes are needed to scale out any application. Customers also have the flexibility to associate specific workloads with specific sets of notes in an InterSystems IRIS cluster. For example, reporting or analytics workloads might be assigned to one pod, and transaction-heavy workloads to another.

Summary of the features and benefits of horizontal scale-out

Feature	Benefits
Built-in high-performance distributed data architecture	• Guarantees consistent, real-time access to any data • Provides the performance of in-memory only solutions with built-in durability, lower total cost of ownership (TCO), and no restart delays
Single, consistent representation of data on disk, in memory, and on the wire	• Provides a performant, efficient, and self-managing solution • Simplifies development and maintenance • Replaces 2- or 3-tier infrastructure with a single technology
Easily scalable solution, linearly, vertically, and horizontally	• Maintains superior performance at scale • Reduces resource requirements and total cost of ownership (TCO)
Ability to independently scale data along with ingestion and analytic workloads	• Optimizes resource efficiency (disk, I/O) • Eliminates the need for data replication
Ability to add and remove query and ingest capacity without service interruptions	• Provides seamless elasticity, greater flexibility, and lower cost • Runs anywhere

Bring It All Together with InterSystems IRIS Interoperability



The next layer of InterSystems IRIS architecture is a built-in interoperability subsystem. It integrates data across messages, devices, and different APIs. It also integrates bulk data, in either the ETL or the ELT (extract-transform-load or extract-load-transform) patterns.

InterSystems IRIS Interoperability uses the common data plane as a built-in repository for all elements of message handling and data integration. This benefits from the performance and reliability of the first two layers, as well as the multi-model capabilities. For example, bulk structure data tends to be relationally oriented, and many messaging protocols tend to be document oriented.

By default, interoperability is persistent – meaning that data messages and transformations are stored within the system for auditing, replay, and analytics. Unlike in many other interoperability middleware offerings, delivery can be guaranteed, traced, and audited across the board. You can confirm that a message was delivered or see who sent what to whom, the type of information important for both analytics and forensics.

The general paradigm for InterSystems IRIS Interoperability is object-oriented. This aids in creation and maintenance of adapters: object inheritance minimizes the effort required to build any needed custom adapters, including testing. It also helps with the creation and maintenance of data transformations. As shown in Figure 8, use of a common object can dramatically reduce the number of transformations needed between different data formats or protocols. Rather than building and maintaining a data transformation for each pair, a single transformation for each data format into a common object provides a simpler approach that is easier to test and maintain.

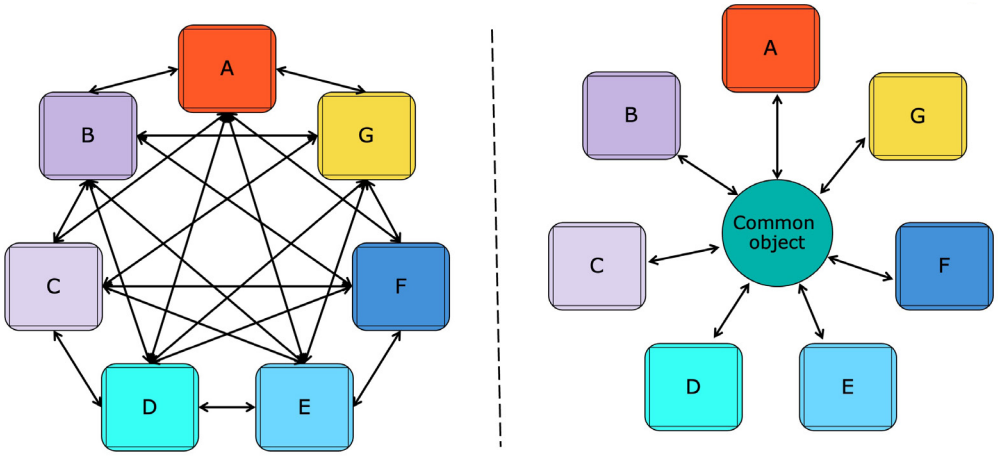


Figure 8 – Simplifying Data Transformations Using a Common Object

Within the InterSystems IRIS Interoperability subsystem live many powerful features, as shown in Figure 9. These cover a wide range of integration scenarios, across messages, devices, and APIs.

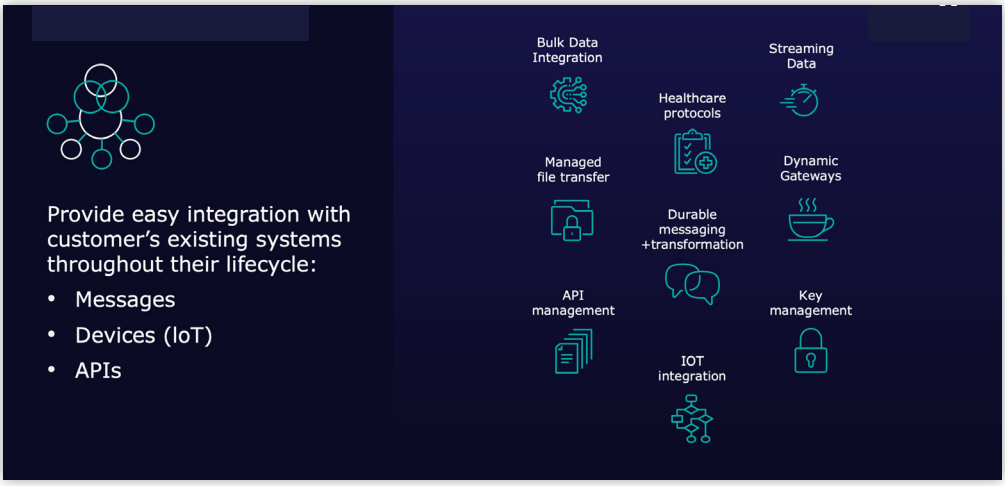


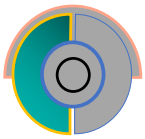
Figure 9 – Interoperability Overview

This interoperability includes built-in full lifecycle API management, streaming facilities, IoT integration, compatibility with cloud services, and more. We also provide dynamic gateways in multiple languages, enabling integration of existing applications into these data flows in the language of your choice, with high performance.

Summary of the features and benefits of interoperability

Feature	Benefits
Built-in interoperability	• Integrates easily with data and applications inside and outside the firewall • Simplifies software requirements
Single, consistent architecture across database and interoperability	• Simplifies building and maintenance • Eliminates duplication of data and effort
Highly scalable message-based integration	• Handles high-volume message flows with low and predictable latency
Persistent and fully traceable messages by default	• Enables transparency, forensics, maintainability, and analytics
Full lifecycle API management	• Supports API-first building, execution, deployment, monetization, and monitoring • Supports APIs and microservices-based applications
Dynamic gateways to and from Java, .NET, Python, node.js, and Go	• Integrates existing applications and components using the language of your choice

Make Sense of Your Data with Built-In Analytics and AI



InterSystems IRIS Interoperability sits alongside a set of built-in analytics and AI facilities, as shown in Figure 10. Each of these capabilities runs "close to the data," meaning that in general we bring processing to the data rather than, at considerable cost and delay, move data to the processing.

Figure 10 – Analytics and AI in InterSystems IRIS

Analytics Close to the Data

There are several analytics facilities built into InterSystems IRIS.

One is **InterSystems IRIS BI**, which is a MOLAP-type, cube-based architecture for business intelligence (BI), optimized for latency. Because this set of subsystems is built into InterSystems IRIS, we can trigger on SQL and events in the cube with only 10-20 milliseconds from data to dashboard. Having a single copy of data across transactions and analytics helps keep this latency low. Because ECP allows one set of nodes to operate on analytics in isolation from the transactional workload, analytics poses no risk to transactional responsiveness, while there is never a need for more than a single copy of the data.

Another facility is **Adaptive Analytics**, which, unlike InterSystems IRIS BI, does not use prebuilt cubes. It dynamically optimizes and builds virtual cubes as it goes, making these available for both BI and Adaptive Analytics is a ROLAP-type headless analytics facility that includes seamless integration with all leading BI tools, such as Tableau, PowerBI, Qlik, Excel, and others.

AI and ML Close to the Data

Alongside the analytics facilities are several ML and AI facilities.

Integrated ML allows you to write automatic machine learning (ML)-style models using SQL. You simply write an SQL command, then create, train, validate, and predict with the model. The results can be directly used in SQL. Thus, developers familiar with SQL can use ML predictions in their applications.

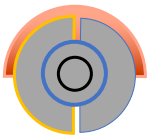
Python sits directly within the kernel of the data platform, so it runs with maximum performance directly against the data. You do not need to port from a development or lab environment where you build models into a production environment where you run those models. You can build and run in the same cluster and therefore have assurance that what you have built and what you run use the same data in the same format and are therefore consistent. Data science projects are simple and fast.

InterSystems IRIS embedded **vector search** capabilities let you search unstructured and semi-structured data. Data is converted to vectors (or embeddings), then stored and indexed in InterSystems IRIS for semantic search, retrieval-augmented generation (RAG), text analysis, recommendation engines, and other use cases.

Summary of the features and benefits of analytics and AI

Feature	Benefits
Built-in analytics and AI	• Easily integrates analytics, AI, data integration, and data storage • Simplifies software requirements
SQL and cube-event triggers	• Provides low latency from data to insight
Streaming support	• Supports immediate action with real time analytics
Integrated ML (machine learning)	• Supports adding predictions easily using your existing SQL skill set
Adaptive Analytics	• Simplifies complex data for use by business analysts with standard BI tools at extreme scale
Embedded Python	• Includes data science and data engineering directly in the database • Simplifies and accelerates projects
Vector Search	• Adds semantic search using your existing SQL skill set • Supports GenAI applications using RAG (retrieval augmented generation) patterns

Enabling a New Approach: The Smart Data Fabric



These layers—the core data engine, the ECP layer to scale out Interoperability, and our analytics facilities—are part of our unique ability to power a **Smart Data Fabric** architecture.

Data fabric is an architectural pattern that provides common governance over a wide variety of data and data sources. A common pattern for a data fabric is to bring in data from multiple sources; normalize, deduplicate, cross-correlate and improve the data; and then make it available for a variety of different applications; see Figure 11.

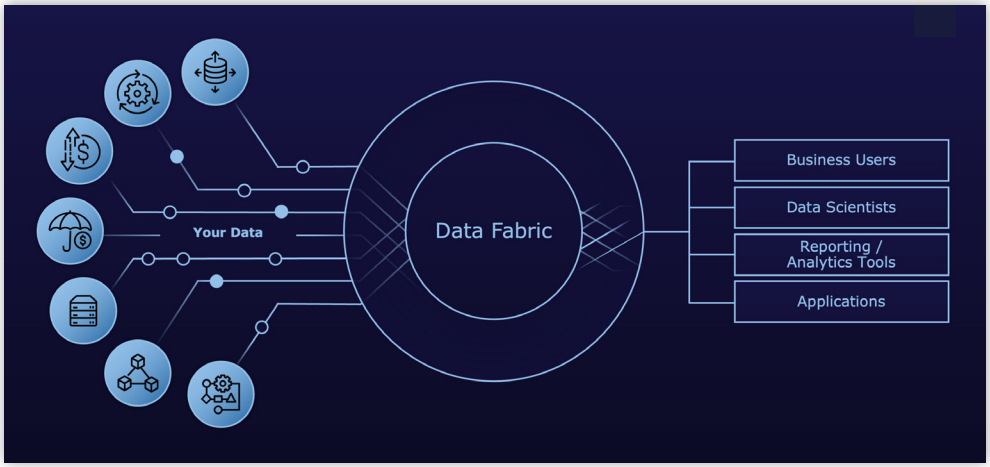


Figure 11 – Generic Data Fabric Architecture

Within most data fabrics, there are multiple capabilities including ingestion, pipelining, metadata, and more. What makes the InterSystems approach smart is the inclusion of analytics and AI within the data fabric, as shown in Figure 12.
One of the key tenets of InterSystems technology is "connect or collect." Some facilities within InterSystems IRIS, like foreign tables or federated tables, let you work or "connect" with data where it lies. Or you can choose to collect that data. There's common data governance, in addition to the facilities discussed previously.

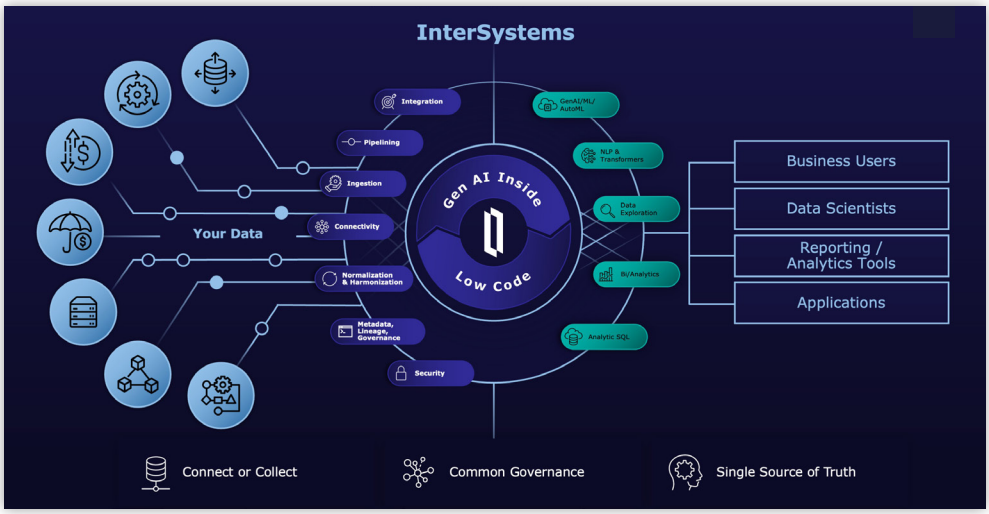


Figure 12 – InterSystems Smart Data Fabric Approach

Living Close to the Data

Because our mission-critical data applications live close to the data, we are uniquely positioned to support:

• High-speed, highly scalable, highly reliable, transactional computation

• Advanced analytics with embedded business intelligence (BI), machine learning (ML), and Python

We can operate at such high speed, high scale, and high reliability because our approach of living close to the data is embodied in our underlying architecture.

As a company, InterSystems is privately held. We focus on the long term, and we religiously maintain the simplicity and purity of our architecture. As a result, the layers of our underlying software architecture, which have been built out over time, are well crafted and highly optimized.

Deploy Wherever You Want To

InterSystems IRIS is agnostic with respect to cloud provider and runs on premises, in the cloud of your choice, in heterogeneous and hybrid scenarios, or in multi-cloud environments. The fastest growing part of our business is our cloud services, which are available across multiple clouds. The flexibility to run wherever you want to deploy is key. That distinguishes InterSystems IRIS from, for example, the facilities provided by the cloud vendors themselves or many of the current options for data warehouses. You can run InterSystems IRIS and applications built with it wherever you want.

Of course, InterSystems IRIS itself is available as a cloud managed service.

Using InterSystems IRIS in the Real World

We have reviewed the different layers of the InterSystems IRIS architecture, the core data engine, horizontal scale-out through ECP, built-in interoperability, built-in analytics and AI, and the smart data fabric.

This architecture has been evolved deliberately and proven over many years. We keep the architecture clean while adding new capabilities. For example, we added Python to the heart of the database and made it available in all layers above that. It scales out seamlessly and can be used for interoperability and analytics, and as a facility within our Smart Data Fabric. We have also added ML and genAI on the analytics layer, which means it is available through interoperability or in Smart Data Fabric scenarios.

Replacing Multiple Systems with InterSystems IRIS

InterSystems IRIS can replace multiple software packages. One example is replacing two distinct databases with InterSystems IRIS, as shown in Figure 13. Instead of having a SQL layer and then an in-memory cache to optimize performance above that SQL level, InterSystems IRIS simply includes both as built-in capabilities.

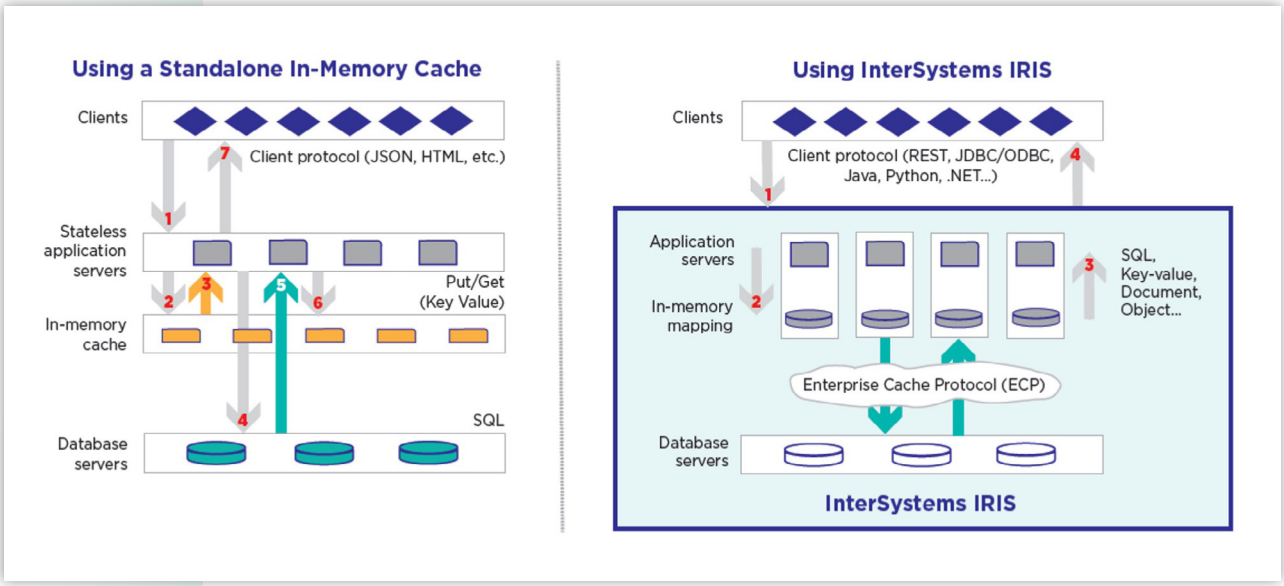
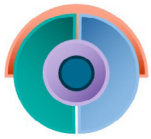


Figure 13 – Simplifying a Database Architecture by Replacing Two Products with One

The beauty of this approach is that it is fully integrated: it is a two-layer database where InterSystems IRIS handles all consistency internally. This eliminates the need to establish affinity between two separate, external databases, manage scaling and disaster recovery for both, and operate both of them. The result is a much simpler architecture and configuration, where everything works together.

Conclusion



The basic philosophy of running close to the data has led us to develop **InterSystems IRIS**, a converged data platform that has one multi-model, multi-workload database management system instead of multiple data stores; a single, multi-model data representation across memory, disk, and the application on the wire; and a single data structure underneath all data models, cross-relational objects, documents, key values, and so on. This delivers all the application-level benefits of a multi-model schema without data duplication; faster, easier development and maintenance using relational, object, document, and key-value as needed.

A single data representation across memory, disk, application, and wire eliminates time-consuming copying and reformatting, accelerating ingestion, transactions, and queries. This consistency in turn supports a more reliable and secure platform. A unified data fabric prevents most failures of data consistency and synchronization. Fewer components mean simpler scaling, easier maintenance, and higher reliability.

A single structure underlies all data models—relational, object, document, graph, vector, and others, as needed—in your applications. This unity of structure, the common data plane, in turn supports easier and faster development and maintenance. This simplicity brings compute closer to the data, with better and smoother performance as you scale, without added complexity or larger footprint. You can even use microservices patterns without having to move enormous data sets between services.

This philosophy of running compute, analytics, AI, and interoperability close to the data has paid off handsomely, delivering very high performance at scale. The all-in-one simplicity of InterSystems IRIS results in greater speed, reliability, and safety. The simpler architecture results in both a faster system and a safer system that is more robust, easier to operate, easier to troubleshoot, and more.

These major steps forward—simplification, unification, consolidation, and acceleration—are reachable today by partnering with InterSystems.

InterSystems IRIS: The InterSystems Difference

Unique Architectural Approach

Our integrated, interoperable, multi-model, multi-lingual engine provides the highest performance and resiliency with the lowest TCO.

We Bring the Processing to the Data

Our approach means less moving of data. That means less potential for data errors, faster processing, greater security, and lower costs.

Highly Flexible

InterSystems IRIS includes the tools to solve unfamiliar problems and adapt as business needs change. Every aspect from data transformations to workflow can be tailored, and low-code tools let you put some customization in the hands of business users.

Fast Time to Value

A wide range of capabilities are pre-integrated and designed to work seamlessly together, which simplifies development and deployment to accelerate business outcomes.

Industry-Leading Support

We are focused on making our customers successful and ready for any challenge, which is demonstrated by some of the highest customer satisfaction ratings in the category.

Powering the World's Most Important Applications

Our software powers mission-critical applications in almost every industry — from healthcare and financial services to supply chain and space exploration.

For more information about InterSystems IRIS Data Platform, see our website and downloadable materials at **InterSystems.com**. Ready for a free trial? Try it **here!**

InterSystems®
Creative data technology