



Actian Zen

Modern Edge Data Management

A Technical Overview

White Paper

Contents

| | |
|---|----|
| Introduction..... | 1 |
| The Age of Edge Intelligence..... | 1 |
| Action Zen and the Requirements of Modern Edge Data Management..... | 2 |
| Core Design Principles..... | 3 |
| The Simplicity and Strength of Key Value Storage..... | 4 |
| SQL or NoSQL? Data Management at the Edge..... | 5 |
| The Demands of Data Management at the Edge..... | 6 |
| Zeroing Out the Need for In-Field Support..... | 10 |
| A Deeper Dive into the Actian Zen Architecture..... | 11 |
| Micro Kernel Database Engine (MKDE)..... | 11 |
| NoSQL and Btrieve..... | 13 |
| SQL and the Relational Database Engine..... | 14 |
| Zen Control Center (ZenCC)..... | 14 |
| Zen Data Management..... | 15 |
| Zen Data Types Supported..... | 15 |
| Zen Data File..... | 16 |
| Zen Data Transfer and Sharing..... | 17 |
| Zen Data Backup and Restoration..... | 17 |
| Zen Security..... | 18 |
| Zen Security Models..... | 18 |
| Zen Encryption..... | 19 |
| Zen Security Auditing Features..... | 19 |
| Zen Deployment Models..... | 19 |
| Dedicated and Embedded in an Application..... | 20 |
| Stand-alone in Support of One or More Applications..... | 20 |
| Client/Server Mode and Multiple Deployment Instances..... | 21 |
| Zen Data Integration..... | 22 |
| Zero-ETL Zen Environments..... | 24 |
| Application and Non-Zen Data Integration..... | 24 |
| Conclusion..... | 26 |
| About Actian-Activate your Data™..... | 27 |

Introduction

A decade ago, only 20% of the people in the US owned a cell phone. Today the number stands at 96%, with some people owning more than one mobile device. Globally, the number of mobile devices is forecasted to grow to 16.8 billion by 2023, but that number doesn't even take into consideration the proliferation of devices on the Internet of Things (IoT). Today, the average large manufacturer, agribusiness, or defense organization has a mass IoT sensor grid averaging 50,000 IoT devices, and that's on top of an infrastructure rife with high-speed devices supporting everything from medical instrumentation and intelligent transport systems to machine vision and surveillance systems. Factor in all those IoT devices, and projections of 75 billion digital devices by 2025 seem not only reasonable but perhaps even conservative. After all, while it may have been reasonable to assume that 8 billion people might use only one or two digital devices apiece, there's no underlying constraint that might suggest an upper limit to the number of IoT devices that will be out there generating, capturing, processing, storing, and communicating data. The 75 billion number may simply be the sign on the side of the digital highway saying "Welcome to the future!"

Even more important than the numbers, though, are the capabilities inherent in these rapidly evolving devices. With fast processors, vast caches of memory, and high-speed communication capabilities, the IoT presents an enormous set of data management challenges. Some of those challenges represent opportunities; others represent threats.

Consider the cellphone. Four out of five phones in the US are smart phones, and audio calls represent less than 5% of the data sent and received on those devices. In fact, more data moves between the service provider and the handset—in the form of software updates—than moves between provider and handset for voice calls every month. Today's Gen Z consumers in the US move an average of 10GB of data per month; by 2025, with the expansion of 5G, that number is projected to be 45GB per month.

And that's just a sampling of first world consumers. By 2025, what wasn't already connected will be connected. What was connected on wired or proprietary low-bandwidth connections will move to some form of 4G or WLAN. It is projected that at least 28% of these devices will leapfrog over interim technologies and move directly to 5G.

All that data. All those applications. Opportunities and threats alike flow past us like grains of sand through our fingers, which means that today's developers need to find better ways to manage those challenges so that neither the opportunities nor the threats slip by unnoticed.

The Age of Edge Intelligence

The place to seize the opportunities and hold off the threats lies in that thin area known as "the Edge." That's where local devices, as well as Edge gateway devices (think environments ranging from a transoceanic cargo ship to your local Home Depot branch), will perform the majority of activities related to data aggregation, cleansing, metadata collection and analysis, even advanced analytics. There are good reasons for this:

- The latency inherent in accessing centralized data stores may be too great, especially when local AI is making decisions based on local data.
- The cost of moving large quantities of data for centralized analysis can be mitigated by using local resources for processing and analysis.
- Security and data privacy concerns increase demand for local data retention and analysis.
- Grid installations of similar devices may need to aggregate and analyze data against parallel sets of devices (for example, PVT) for local response, in real time, to exceptions.

While it may seem like sci-fi today, it will be commonplace for machine learning to run inference algorithms locally on devices such as cameras performing machine vision for part defects or facial recognition. Imagine fusing Alexa with multiple intelligent sensory devices; the underlying hardware technology and economics are reaching critical mass to support this.

Given these realities, the emerging Age of Edge Intelligence raises questions about how best to manage the information flowing through and across this critical interface. It's not simply a question of developing the algorithms to provide intelligence at the Edge. It's a question of creatively rethinking the underlying data management strategies that can both foster and govern the performance of those algorithms. It's a question of how we embed robust, secure, high-speed data management capabilities *at* the Edge to facilitate Edge Intelligence.

This paper offers Actian's answers to some of these questions. It outlines the requirements of a Modern Edge Data Management strategy and shows how data managers can use Actian Zen to support the work of developers, product designers, and service managers who deal with Edge-based machine-to-machine and human-to-machine connections across their entire range of projects. Further product details, performance benchmarks, and other supporting use cases can be found at www.actian.com/data-management/zen-embedded-database/ and for developers at zendocs.actian.com.

Action Zen and the Requirements of Modern Edge Data Management

At its core, data management can be defined as any operation that needs to store data. Databases, data warehouses, and data lakes, like Hadoop, that we find in traditional and cloud data centers have evolved to handle large volumes and varieties of data, often being created, read, updated, and deleted by multiple programs or applications simultaneously.

But data management at the Edge is nothing like data management in a traditional data center or even in the cloud. Traditional and cloud data centers are relatively consistent and well-controlled environments. In contrast, Edge environments exhibit extreme heterogeneity. Edge applications developed and deployed across devices, gateways, and intelligent equipment rely on a wide range of programming and scripting languages, operating systems, and hardware platforms. For development teams, this presents difficulties in the area of data transformation, metadata management, security vulnerabilities, and vendor and asset management.

These difficulties notwithstanding, data processing, analytics, and queries of data to local and cloud databases will increase at the Edge. There is simply too much power, memory, and bandwidth available *at* the Edge for these activities to be confined to the data center.

For these reasons, seven key considerations must inform the database products supporting Modern Edge Data Management. Developers need the following:

1. The ability to manage large sets of persistent data, particularly in nomadic situations with sporadic periods of dis-connectivity.
2. The ability to manage stateful and stateless sharing of data in both peer-to-peer and Edge-to-cloud configurations across IoT *and* traditional intranet and internet environments.
3. The ability to run on multiple operating systems as well as in standard, embedded, and virtual machine (VM) configurations.
4. The support of embedded access via various interfaces, including CLI, programming and scripting languages.
5. The ability to collect, aggregate, and fuse high-speed, multi-channel data collection and aggregation with disparate large grids of slower-speed data collection.
6. The ability to work with a wide variety of data types (not only integer and string data, but also more complex types like JSON, BLOB, traditionally structured, etc.).
7. Plug-and-play compatibility with advanced analytics, reporting, and visualization tools and platforms (both commercial-off-the-shelf and open source).

Action Zen consists of a single, secure, scalable portfolio of database products designed to meet the needs of on-premises, cloud, mobile, and IoT application developers grappling with the challenges of Modern Edge Data Management. Action Zen provides persistent local and distributed data across intelligent applications deployed in enterprise, branch, and remote field environments, including mobile devices and the IoT. You can use Action Zen to develop and deploy database applications for Intel- or ARM-based architectures running Microsoft Windows 10, Windows 2016 and 2019 Server, Linux, macOS and iOS, and Android, as well as Raspbian Linux, Windows IoT Core, and Windows Nano Servers. This combination of power, versatility, and flexibility provides developers with the capabilities they need to support the latest application requirements for local data and embedded analytics.

Core Design Principles

Given the complex challenges inherent in the Modern Edge Data Management list above, four principles have driven the design of Action Zen: simplicity, modularity, portability, and open standards. Guided by these principles, the Zen designers have focused on high performance, support for a wide range of development and deployment environments, easy adoption by engineers and developers, ease of use, minimal need for in-field support, and ongoing compatibility with previous generations of Action Zen releases. In this section we review how these principles manifest themselves as specific architectural features.

The Simplicity and Strength of Key Value Storage

The Zen database management engine is constructed on a key value store (KVS) architecture rather than a relational database management system (RDBMS) architecture but provides full ACID compliance like Enterprise-grade RDBMS systems. This is important because Edge Intelligence is focused on distributed environments and inline analytics which demand a highly responsive database management system (DBMS) architecture. Both the KVS and RDBMS architectures were conceived decades ago but diverged through evolution based on customer demands and requirements in distinctly different market spaces, with RDBMS systems focusing on large consolidated systems and KVS systems focusing on small or distributed packaged applications. The majority of KVS-based databases are NoSQL. They are not full-fledged RDBMS systems, but they are simple and fast because they do not pre-define the data structures with a set of tables and fields.

From the perspective of Modern Edge Data Management, the KVS architecture provides compelling performance advantages that can be leveraged to design databases that can operate effectively across a wide range of platforms and environments and that can scale from very small to very large.

Figure 1 below provides a simple representation of how a KVS architecture works.

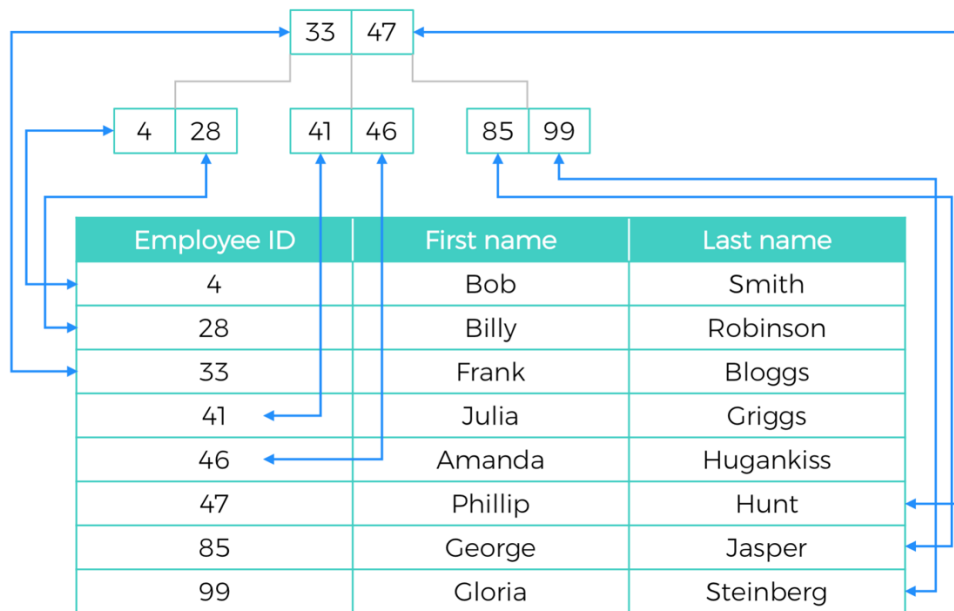


Figure 1: Key Value Store structure and how it works in conjunction with a B-tree indexing scheme

Each row of data in a KVS table has a key and one or more values paired with that key. In Figure 1, the Employee ID number is the key, and the associated values reside in the first name and last name columns. These values could be any data type, including large binary objects. In response to a query, the code logic in the database engine sorts on the key values as B-tree indexes, enabling rapid navigation through the series of pairs that constitute the data table or database. Here, the blue lines show how a query using a greater-less-than-equal-to algorithm would arrive at the desired associated value.

SQL or NoSQL? Data Management at the Edge

Edge Data Management requires speed and balance with respect to reads and writes, support for multiple and flexible data types, and the ability to run in multiple modes (time-series data collection, ad hoc queries of existing data sets, and the ability to handle transactional reads and writes in complex client-server modes of operations, such as publisher/subscriber). This is one reason Actian has opted for KVS as the underlying data management engine for Zen.

[Sidebar] Not all NoSQL Databases are alike

Not all NoSQL databases are KVS, and not all NoSQL databases are the same. Figure 2 provides a brief description of the four most popular types of NoSQL database. At Actian we have products based on each of these options but have opted for KVS as the underlying data management engine in Actian Zen because it satisfies the Edge Data Management need for speed and balance.

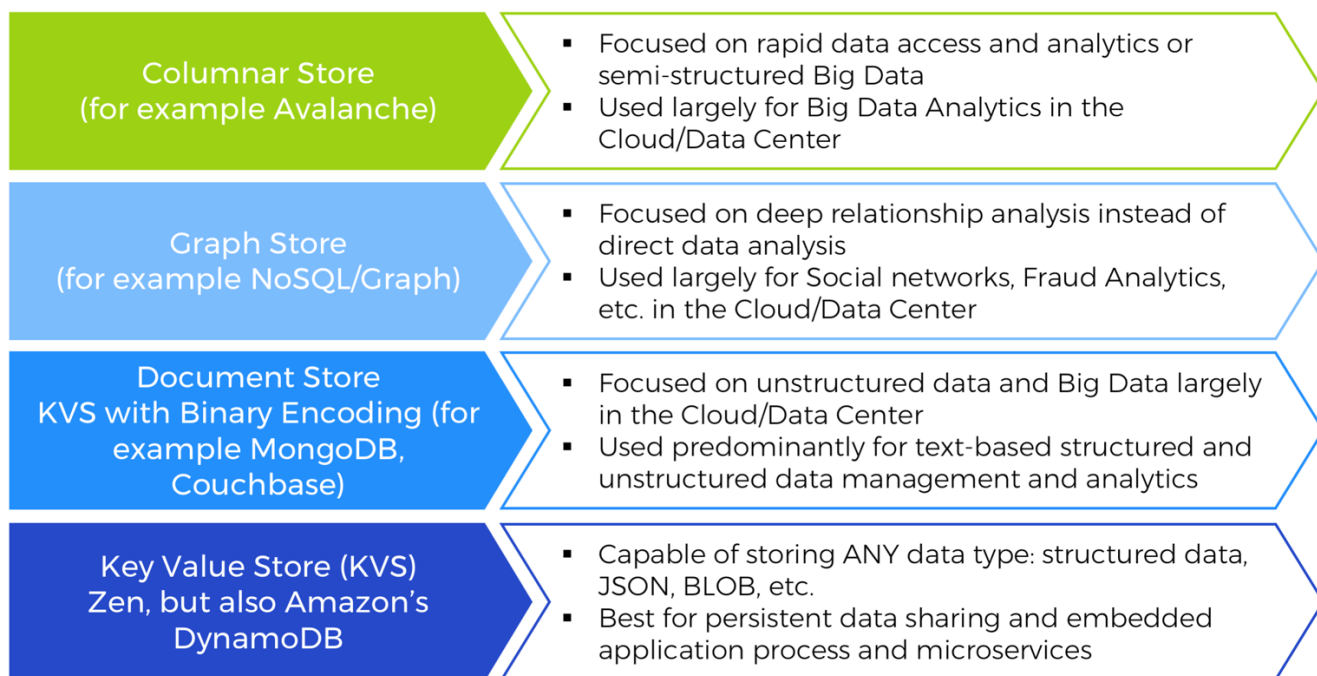


Figure 2: Four of the most popular types of NoSQL databases

But there are other reasons as well. While most KVS databases are NoSQL, KVS can underpin SQL as well. Actian Zen uses the KVS data management engine to support both SQL *and* NoSQL access. This is important because NoSQL and SQL offer different advantages in different Edge Data Management scenarios. If you are streaming data over time and need to write data to memory in a serial manner, NoSQL provides certain advantages. In Zen you can use a NoSQL option for automatic timestamping. Alternately, if you are managing ad hoc queries from multiple requestors, queries that may change the value of the stored data, SQL can provide the critical advantage of ACID compliance (which most NoSQL databases cannot). Figure 3 below provides a comparison of the key advantages and disadvantages of using NoSQL versus SQL as a means of data access and management.

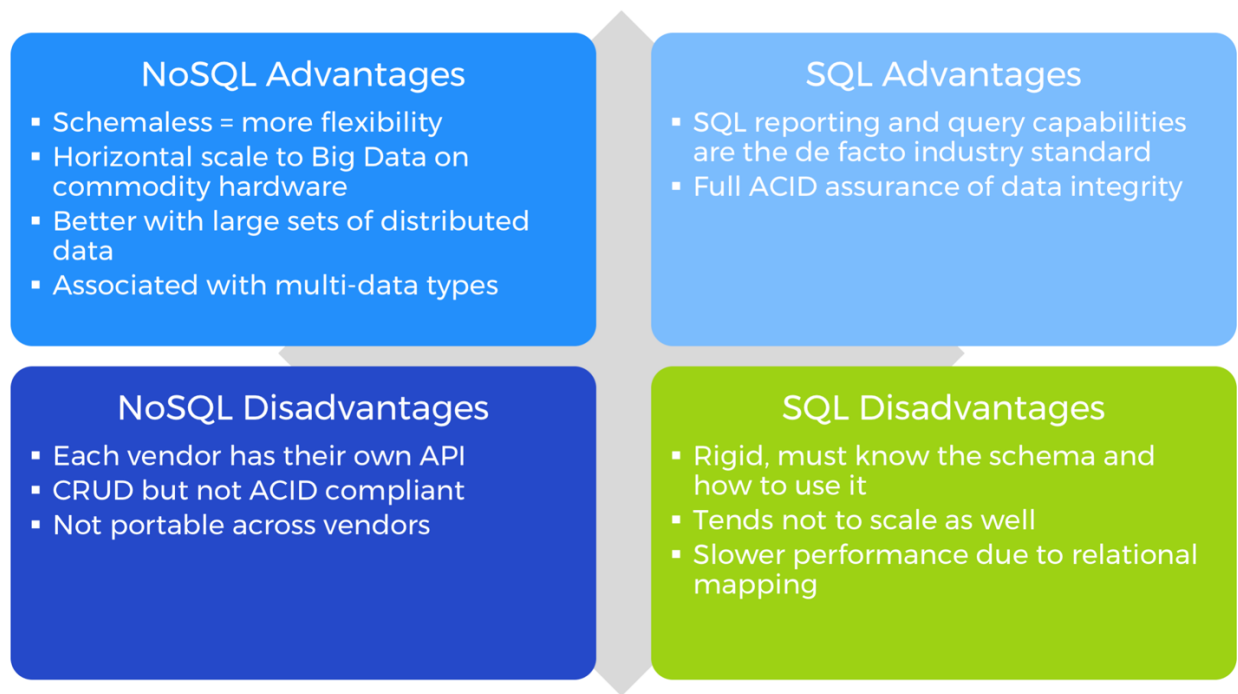


Figure 3: The relative advantages and disadvantages of NoSQL and SQL

The Demands of Data Management at the Edge

At the Edge, though, the use cases for temporarily storing data are far more varied. They range from time-series data pushed by a low data-rate IoT sensor to high-volume transactional data pushed by a point-of-sale device and form-based data pushed from the web UI of a mobile phone. This variety has resulted in the creation of data management systems ranging from specialized file systems to stripped-down databases for caching data such as IndexedDB or SQLite.

The opposite end of the spectrum, of course, is a data management system that can handle large volumes and varieties of data being created, read, updated, and deleted by multiple programs or applications, oftentimes with simultaneous operations. Databases, data warehouses, and data lakes like Hadoop have evolved to specialize in this end of the spectrum in both the Cloud and traditional Data Centers.

When you consider the combination of these Edge, Cloud, and traditional Data Center environments, the databases and other data management systems number in the hundreds. Any one of them—or a class of them—may satisfy a particular need for data management, whether it be on a small system, a large distributed environment, structured data with a schema, or JSON data with documents. But the key point is that there is no one-size-fits-all solution applicable across all these different data management use cases.

Increasingly though, Modern Edge Data Management demands that we address data processing and analytics requirements as data moves from the Edge to the Cloud, from the Cloud to the Data Center, and then back again. Not surprisingly, different devices along the Edge present different requirements for data management. See the sidebar for more information on the role of data processing and analytics in key Edge devices.

[sidebar] Data Processing and Analytics Requirements at the Edge



Figure 4: Where Edge Data Management is Required and Changing Demands on how and where it's needed.

From left to right: Grids & Sentinels are at network Edge, front-ending a range of more sophisticated and well-resourced devices which, in turn, all feed in parallel to Edge gateways which then feed into more centralized On-Premises Edge systems. The On-Premises Edge systems represent the new data center use cases and opportunities for shifting some of what is in the growing Cloud and stagnated Data Center environments to a lower latency, underutilized set of Edge resources.

The bars below the device descriptions represent some of the key trends driving the requirements for more robust data management. Key drivers here include the shift to advanced analytics, such as sensor fusion, and decision-making at the point of action on Edge devices or at Edge gateways.

The data management demands presented by this spectrum of devices and use cases are manifold. In the realm of Edge analytics applications there exists a wide range of use cases and device types. On the high end, on devices with 64-bit CPUs, GPU arrays, and high-definition video and audio resources, Machine Learning (ML) systems are supporting machine-vision use cases. ML for real-time applications not only requires low latency but also often mandates pre-processing, at least for Signal-to-Noise Reduction. At the low end, on 32-

bit microcontrollers sporting just tens of kilobytes of memory, we see ML systems leveraging unsupervised routines for low-power operations. Here a gesture, a change in a PVT sensor, or some other distinct event will drive a sentinel function running ML, which in turn will trigger a set of actions, possibly including the power-up of more intelligent and resource-intensive system modules that can execute some defined task.

Many intelligent Edge devices already familiar to users, such as in-home Virtual Assistants and IoT Gateways, represent platforms that combine multiple use cases. They may aggregate data and run distributed applications on local Edge servers, which could be small enough to run in a closet or the fuselage of an airplane, or large enough to occupy several server racks (on premises or in a private cloud).

Mobile workforce applications present other Edge Data Management challenges, particularly those where there are network disconnects (whether planned or unplanned, such as those experienced in a delivery truck on a rural route). The mobile devices must be able to share and distribute data when connected, but they must also be able to work only with local data when disconnected.

Mobile scenarios work best with a single data management platform spanning all involved platforms with distributed application support. Another set of scenarios benefiting from a single data platform are those in which a product line relies on many identical core components or functional services but presents features and functionality in distinct ways depending on the tier of service or device a customer has selected. A line of medical imaging equipment might run the gamut between portable CT Scanners on one end and fixed, high-end laboratory scanners on the other. To use a single data management platform across all these products would require the flexibility to run the data management system on a stand-alone 64-bit MCU running Android or embedded Linux on the low end while at the high end it could run on an embedded multi-core Intel 64-bit CPU server and Windows Server 2019 with multiple TBs of memory for local image storage. These same scalability considerations can be seen in everything from intelligent farm equipment (from an APV to an industrial harvester) to intelligent in-seat entertainment centers (which offer different options in first class, business, economy, and so forth).

Many traditional software vendors and systems integrators have created products and services to address these needs, and generally they approach the challenge by integrating distinct products and services that have been optimized for each of the platforms. The platforms come first; the integration comes second. Much work goes into outfitting the products and services with a uniform set of APIs, file systems, and function calls to facilitate data management across the platforms.

Action Zen enables developers to take a different approach. Using Action Zen, developers can rely on a simple but powerful microkernel core as the lowest common denominator, then build upon that core to add the features and functionality intended to address the entire range of Edge requirements. Figure 5 maps the Action Zen product portfolio against the functional areas and use cases covered in the [sidebar] Data Processing and Analytics Requirements at the Edge sidebar.

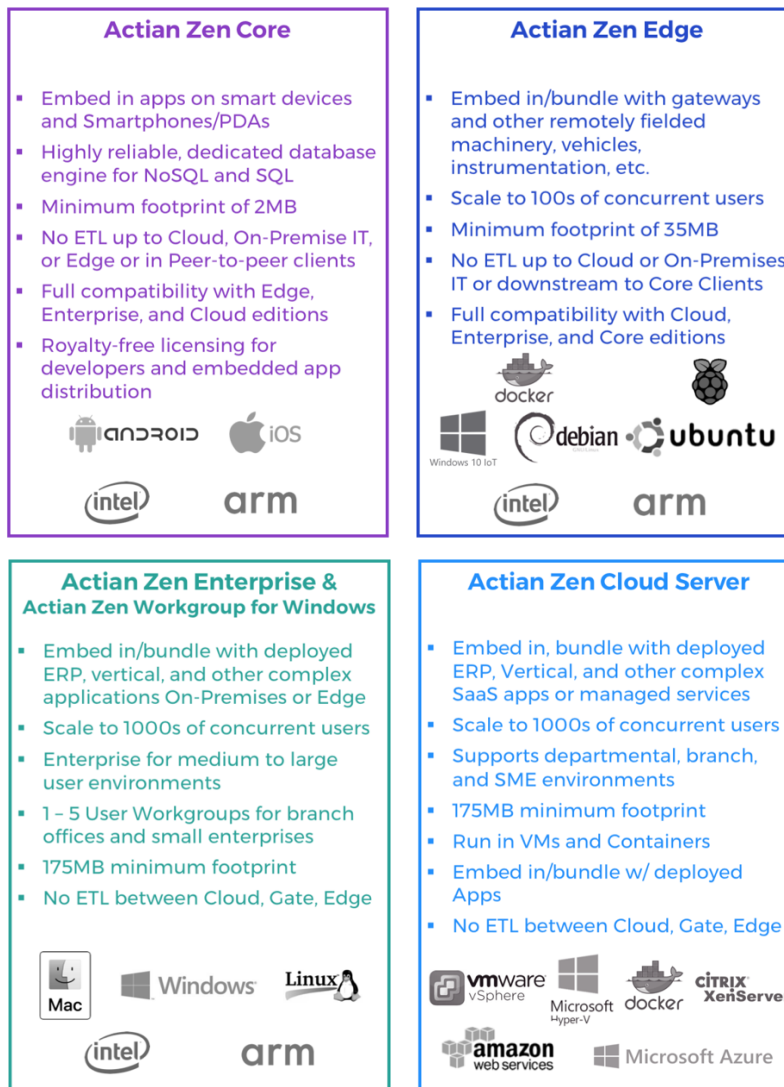


Figure 5: Actian Zen Product Family

Actian Zen Core edition empowers devices and intelligent machines where a local application needs data processing and analytics that may be persistent, stateful, and operate in a stand-alone or intermittently-connected communications mode. Examples of such devices include smartphones, tablets, and a broad range of IoT devices. Developers can embed Zen Core directly in an application for programmatic or scripted access.

Zen Edge edition extends the functionality of the Zen Core edition with support for more complex distributed applications and a range of standard interfaces (ADO.NET, ODBC, JDBC, etc.). It can manage multiple data streams concurrently in a publish-and-subscribe model on Linux and other IoT environments. Developers can install and run Zen Edge separately from the client application, which can run locally or remotely, on a different device, as the use case demands. Zen Edge supports concurrent user access for single or multiple applications, on independent as well as shared data services.

Zen Enterprise Server edition extends the functionality of Zen Edge to scale up to support the needs of thousands of end-users or applications. It supports more complex operating environments (including on-premises and virtual). Zen Enterprise Server edition adds in

database management that can operate locally on the Zen Enterprise instance while also serving as a management platform for downstream Zen Core and Edge instances.

Zen Cloud Server edition provides all the features and functionality of Zen Enterprise Server edition but runs on virtual machines and in containers, either in Public or Private cloud environments.

Action Zen products can easily be deployed on most major server, desktop, mobile, and virtual environments based on Intel and ARM 64-bit architectures. All editions rely on a common architecture built on NoSQL API and SQL queries and the same database structures and management and control services. They all write to a common file format and can be deployed to run remotely and without ongoing supervision, even when operating in an embedded and/or occasionally-connected manner. No other generally available, off-the-shelf database products run on mobile ARM-based platforms with embedded Linux, on AWS and Microsoft Azure Public Clouds, on private clouds built on VMware or Microsoft Hyper-V, and on local servers and workstations. This deployment flexibility enables developers to port applications from one platform to another and to securely share data between themselves, back to on-premises servers, or up into the Cloud, all with a common interface to the requisite data management functionality.

Zeroing Out the Need for In-Field Support

All Zen database products include features easily taken for granted by Data Center and Web Developers, including features that are not necessarily available and often reinvented from scratch by developers in Edge environments. In other cases, required features are implemented in ways that Data Center and Cloud databases are unconcerned with. For example, a core requirement at the Edge is to avoid any database administration—a need so widespread and critical in traditional environments that there is an entire profession and college curriculum to support the administration of complex organizational databases. The needs of Modern Edge Data Management are different. Databases at the Edge must be able to operate with minimal database administrator (DBA) intervention—certainly without intervention that requires a heavy touch to start, stop, configure, catalog, or monitor deployments.

Action Zen enables a Zero-DBA database deployment through use of a Distributed Tuning Interface (DTI), an API that can be used for remote configuration, monitoring, and diagnostics. This Zero-DBA approach eliminates the need for technical support personnel to interact physically with an Action Zen database in the field.

A Deeper Dive into the Actian Zen Architecture

As previously noted, all Actian Zen editions build on a single scalable, secure, and modular architecture, as illustrated at a high level in Figure 6.

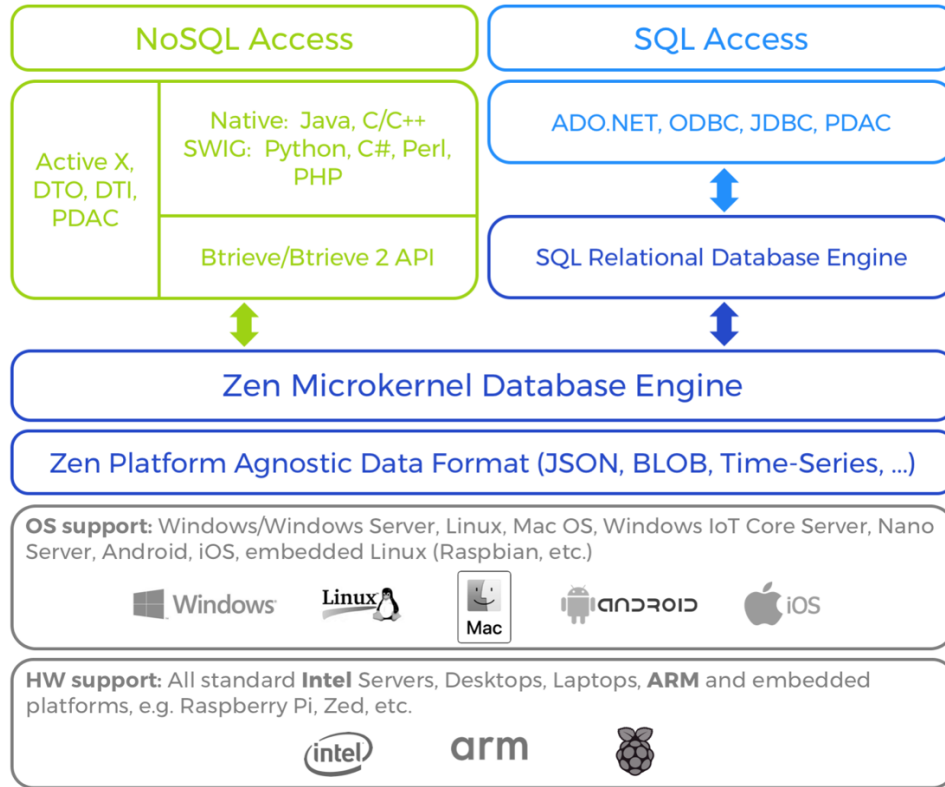


Figure 6: Actian Zen Architecture

The exact implementations differ from level to level to make the most of the underlying physical or virtual resources, but the databases themselves present a uniform architecture to all applications and a common API and SQL conformity to all developers and end users. The subsections below examine the key components of the Actian Zen architecture in greater detail.

Micro Kernel Database Engine (MKDE)

The Zen Micro Kernel Database Engine (MKDE) drives the core database operations, including all data access methods. The MKDE’s core operations enable your applications to retrieve, insert, update, or delete records by key value or by sequential or random-access methods.

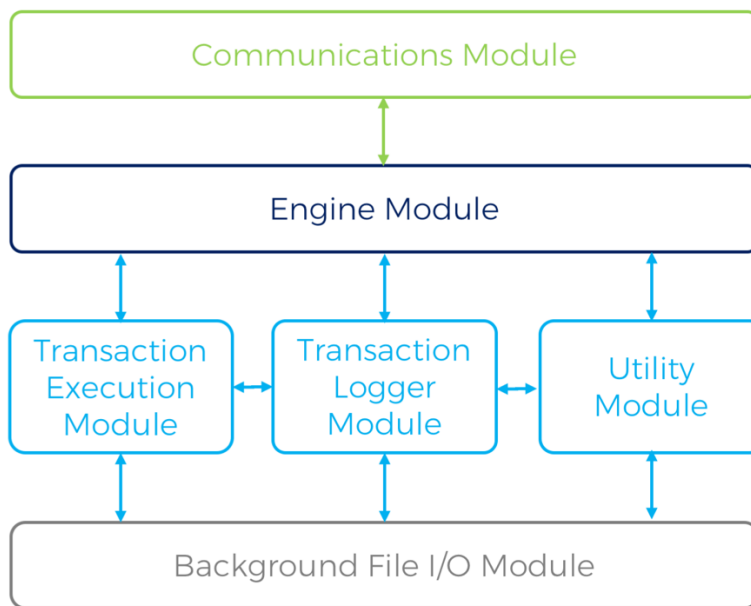


Figure 7: Micro Kernel Database Engine Architecture

As mentioned previously, the databases housed in the MKDE are stored in a common file format that can currently hold as much as 64TB of data. These Micro Kernel files (Zen files) contain the records and B-Tree style indexes which allow for lightning-fast CRUD operations. They can be accessed easily in schema-less environments using direct API calls or with higher-level JSON or BLOB files. Schemas, if used, reside in separate Data Dictionary Files (DDFs). With a 64TB limit for the current file format, a single table can have billions of records.

The MKDE can easily work with multiple Zen data files, including those stored locally as well as on remote systems. MKDE functionality can be accessed either from ANSI SQL commands (embedded in code or via CLI) or via the NoSQL API, which can be accessed using almost any programming or scripting language. Figure 8 illustrates how NoSQL and SQL calls are executed within Actian Zen.

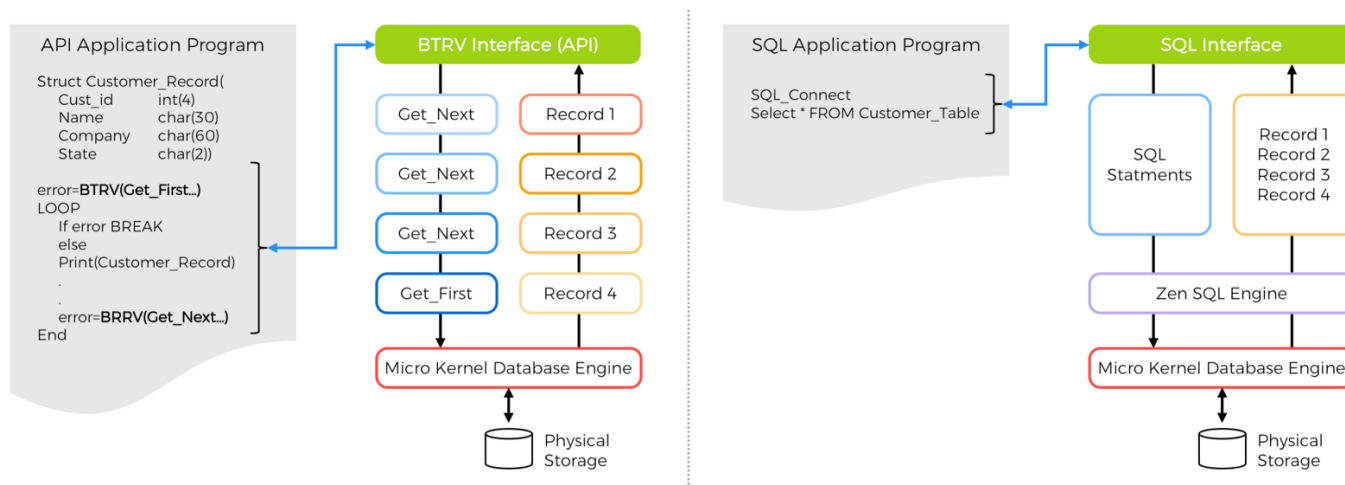


Figure 8: Actian Zen NoSQL and SQL Access Method Comparison

For maximum flexibility, simplicity, and performance, Actian recommends that the Btrieve option be used where transactional operations are directly applied to a database. However,

this does not prevent using the data as a relational database or with any SQL operation that can leverage an available schema stored as a DDF.

[Sidebar] Actian Zen Software Development Kits (SDKs)

The Actian Zen Software Development environment is comprised of several software development kits (SDKs) that enable developers to create applications using APIs for transactional access methods (NoSQL or Btrieve), relational access methods (SQL), or a combination of the two. Either method, SQL or NoSQL (Btrieve), can be used from almost any programming language, for example C/C++/C# or Java, as well as through most popular scripting languages, including Python, Perl, PHP, and JavaScript.

The Zen SDKs provide access to the Distributed Tuning Interface (DTI) functions that enable developers and data engineers to handle the features and functionality that, in traditional data center settings, they would normally defer to DBAs. DTI and its object-oriented partner Distributed Tuning Objects (DTO) enable developers to develop a range of useful, customized server administration tools and interfaces quickly and easily. For example, in Windows environments, developers can use any OLE Automation controller or create a COM client using programming languages such as C/C++, Visual Basic, or Delphi.

Developers interested in using Zen SDKs should leverage the documentation set available at zendocs.actian.com to get started. More detailed documentation about the APIs is available at docs.actian.com/zen/v14/.

NoSQL and Btrieve

Btrieve is a NoSQL API that interfaces directly with the MKDE. Two Btrieve APIs are available:

- The original Btrieve API, which enables database access through a single, parameterized function call.
- The Btrieve 2 API, which provides object-oriented database access using functions named to reduce the learning curve for new developers.

The Btrieve APIs provide compatibility with:

- C/C++/C#, Delphi, COBOL, Visual Basic, and Java programming languages.
- JavaScript, Python, Perl, and PHP scripting languages.
- Several popular development environments and SDKs, including the Java Development Kit (JDK) 1.4 and higher, Microsoft Visual environments, Micro Focus, Android, Apple, and Embarcadero.

The Zen SDK for Btrieve includes all of the needed header files to use each of the interfaces, documentation for the new object-oriented classes and methods, and new 32- and 64-bit libraries for C and C++. The SDK also comes with SWIG (Simplified Wrapper and Interface Generator) files for Perl, PHP, C#, and Python, enabling developers familiar with those languages to use them to access Btrieve via the C and C++ libraries.

The Btrieve SDK also includes a set of Java Class Libraries (JCL) that are analogous to JDBC. These libraries enable object-oriented applications to leverage all the features of the Btrieve API set and to take advantage of the performance gains derived from interacting with the MKDE directly (compared to using SQL access methods through the Relational Database Engine to access the MKDE).

SQL and the Relational Database Engine

Action Zen supports a broad range of SQL statements and related tasks for data definition, manipulation, and control. The statements and tasks are fully compliant with both ANSI and IBM Systems Application Architecture (SAA) standards.

- **Data definition** statements enable you to create, drop, or modify tables as well as views, indexes, triggers, stored procedures, and user-defined functions. These SQL statements cause the Zen Relational Engine to update the database metadata in the DDFs and can create and manipulate the underlying Micro Kernel file structures as well.
- **Data manipulation** statements enable you access and modify the contents of your database through standard relational dataset theory. Zen can execute standard SQL data manipulation statements to retrieve and modify table data, define transactions, create views and temporary files, and execute stored procedures and triggers.
- **Data control** statements enable you define security parameters for your database. When you create a dictionary, no security is defined for it until you explicitly enable security for that dictionary. Zen allows you to enable and disable security, create and delete users and groups, add and drop users from groups, change user passwords, and grant and revoke rights to tables and columns.

The Zen SQL Relational Database Engine (SRDE) stores all the database metadata in a set of DDFs and controls the connections to each database defined in your environment. At the same time, the flexibility of Action Zen ensures that you can access these databases as though they were NoSQL stores using Btrieve too.

Zen Control Center (ZenCC)

The Action Zen Control Center (ZenCC) is a utility integrated into Zen Enterprise Server and Zen Cloud Server that enables users (or developers during test and QA for deployments) to connect to Zen Core, Zen Edge, Zen Enterprise and Zen Cloud engines, set up and modify databases and tables, query and update data, tune engine performance, and access the Zen documentation library.

ZenCC is a graphical tool that uses a file explorer motif, aptly named *Zen Explorer*. It enables comprehensive configuration and management of Zen servers and clients from any Zen Enterprise or Zen Cloud instance running on Windows, Linux, or MacOS. Figure 9 depicts the range of functionality provided by ZenCC.

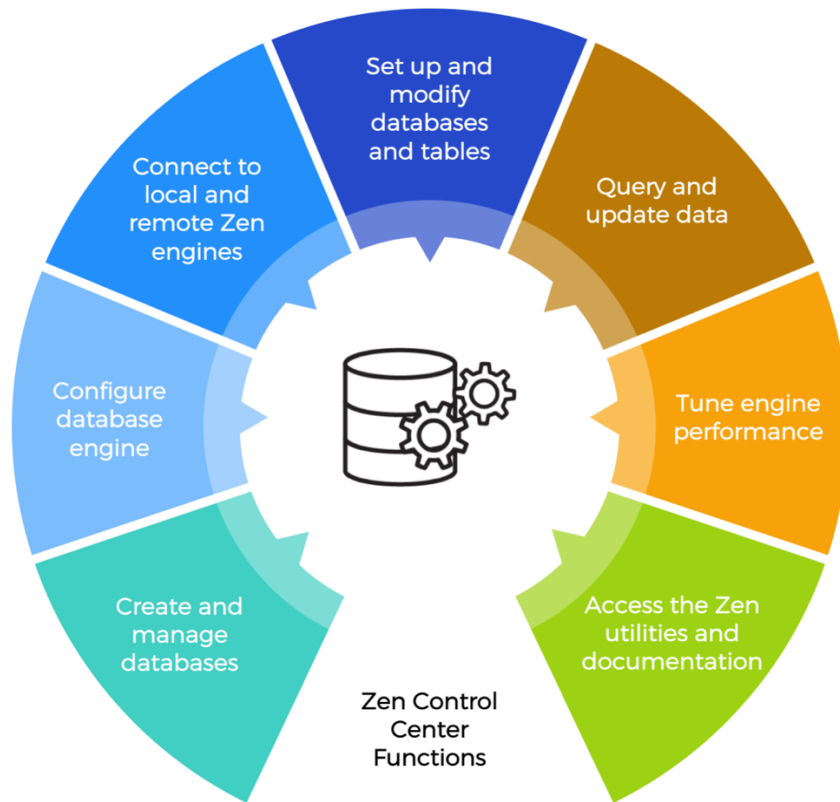


Figure 9: Zen Control Center (ZenCC) Functions

ZenCC enables you to configure all the properties of a local client or any registered engines, local or remote, including those running in environments lacking the resources to support ZenCC locally (such as Windows Nano Server, Windows IoT Core, and Raspbian).

ZenCC also has a built-in SQL Editor that allows you to edit and execute multiple SQL statements. It includes a graphically-oriented Table Editor in which you can add, delete, or change the characteristics of columns, indexes, and referential integrity (RI) constraints in a table.

Additionally, ZenCC UI provides a text view of the results of running SQL statements, an outline view of SQL statements, and a grid view—similar to a spreadsheet—that presents the results for each field as a column of cells containing data that can be edited directly.

Zen Data Management

Zen Data Types Supported

Action Zen natively supports all the data types most commonly found in Modern Edge Data Management scenarios:

- Signed and unsigned integers from 1 to 8 bytes

- Auto-incrementing identity types from 2 to 8 bytes
- 4-byte and 8-byte floating point values
- Single- and double-byte character strings (null-terminated or space-padded)
- Multiple date and time formats
- Seven different COBOL numeric data types
- Specialty data types like BLOB/CLOB, GUID, and LOGICAL

Further, Zen handles most any user-defined, semi-structured, or unstructured data including JSON and BLOB data. In the case of JSON files, Zen can parse various objects and structures within a JSON file using its JSON filter and query operator to provide high-speed access and data-aware search capabilities.

For IoT environments requiring time-series support, such as statistical process control monitoring, scientific research, and other high-speed analytics cases, Zen can apply septasecond or nanosecond time stamps to data entered using SQL or the Btrieve API. These high-resolution time stamps can reference either UTC or local time.

Finally, Zen supports an autoincrement data type, which autoincrements the addition of data records to the file and can be used as a unique value/ID per record within each Zen data file. This can be used as a criterion in SQL queries or Btrieve requests for post-processing or as a trigger to invoke data aggregation, purging, or other cleanup operations to further reduce the need for manual database administration.

Zen Data File

As noted above, all Zen family members use the same Zen data file format. By supporting a single file format—regardless of platform or engine type—you can migrate database files from one Zen platform to another, eliminating the need for complicated translations or extract, transform, and load (ETL) processing.

Each Zen file contains a single database table and can be managed independently. In Actian Zen version 13, the maximum file (and therefore table) size is 64 TB. To save critical storage space in smaller environments, Zen offers two types of data compression: record and page. These can be used separately or together. Compression reduces the size of the data files, but it can also improve performance (depending on the type of data) and offset some of the performance penalty when moving data between Zen instances on different platforms. Compression factors can be as high as a factor of 8, but overall file compression is heavily dependent upon the type of data subject to compression.

All Zen data files and data dictionaries can be viewed from the operating system, which simplifies both deployment and backup. The Actian Zen engine also provides read/write compatibility stretching from the current (v13) file format all the way back to the v6 file format, which was first released in 1992! Ongoing backward compatibility is a hallmark of Actian Zen, so you can build your applications today and expect the data to be accessible for many years into the future.

Zen Data Transfer and Sharing

All editions of Actian Zen—from Zen Core through Zen Cloud Server—support data transfer and sharing across networks. To access and manage network files from a machine where a Zen application is running, you use an appropriate client requester on that machine. Zen calls made by your application go through the Zen client, which sends them to the correct database engine (which may be local or remote across a network). The engine sends a reply back to the client, which then passes the reply to your application. The connection-oriented TCP/IP protocol (either v4 or v6) is supported across all operating environments, with the ability to automatically reconnect in certain environments if a disruption occurs. Wire encryption is available for all remote communications, ensuring the security required to protect critical or personal data that may cross public network segments.

Zen Data Backup and Restoration

Data loss can occur even at the Edge, so an important part of Modern Edge Data Management is a backup strategy. As noted above, Actian Zen files are all visible to the OS, which makes it easy to incorporate Zen files into your existing backup strategy. Standard operating system utilities, third-party software, or even simple scripting solutions (for when users are not accessing the data) can ensure full or incremental backup and recovery of your Zen files.

If you need to backup database files while they are in use, whether by users or automated processes, three solutions are available:

- **Continuous Operations Mode** is a special file handling mechanism inside the MKDE that tells the engine to close data files (if they were open) and store all changes in temporary data files (called delta files) without any interruption to users/applications. The backup can then access the normal data files, which are essentially frozen in time. When the backup is complete, you turn off Continuous Operations Mode, which asks the engine to apply all of the changes stored in the delta file back to the original data files. Because this is handled internally by the MKDE, this option is available on all platforms and engine types and can be initiated by the developer through an API or via a script.
- The **Zen Backup Agent** can be invoked by a pre- and post-processing command-line call from your backup software. When turned on, Backup Agent automatically invokes Continuous Operations Mode on all currently open files, then monitors for new files being opened during the duration of the Backup Agent session. When the backup is complete, your backup software sends another call to Backup Agent to return all files to normal mode. Backup Agent is included Zen Enterprise Server and Zen Cloud Server.¹
- The **Microsoft Volume Shadow Copy Service (VSS)** can be used in place of Zen Backup Agent for Zen Enterprise or Zen Cloud Server implementations running on Windows Server. The Zen VSS Writer component is installed and enabled automatically, and it causes the database to quiesce (freeze) until VSS has obtained a full volume snapshot. The backup software can then mount the VSS snapshot and copy data from there, knowing

¹ Backup Agent is offered as a separate option for Zen Workgroup for Windows.

that it has a transaction-consistent snapshot.

Regardless of the backup solution you choose, you are always backing up your Actian Zen data at the Zen file level. If you need to restore your data, you can restore either the entire database or just one file, depending on your specific needs.

Archival Logging

Archival logging is a mechanism that you can use to supplement regular operating system backups. Archival logging enables you to keep a log of all database operations on a per-file basis since your last backup. In case of a system failure, you can restore the data files from a backup then roll forward the changes from the log files to return the database to the state it was in prior to the system failure.

Zen Security

At the application level, access to Actian Zen and Actian Zen files can be secured by authenticating users through the operating system, through an Active Directory account, or as a named Zen database user. At the operational level, Actian Zen provides security features that can ensure the integrity of both the relational table and column levels for SQL-based applications and at the file level for Btrieve applications that call the MKDE. Actian AuditMaster, an add-on to Zen, provides detailed application-independent audit trails at the underlying MKDE level, enabling you to monitor and identify the original source of each database event.

Zen Security Models

Actian Zen secures the relational database and MKDE engines in slightly different ways.

With the Relational Database Engine, Zen offers SQL authentication and authorization configurations for each local database or, if on a Windows platform with Zen Enterprise Server or Zen Cloud Server, integration with Windows domains and AD. In both cases, password-protected permissions can be set for a master user, groups, and individual users within a group. Permissions can be granular, extending to tables, columns within tables, views, and stored procedures. You can manage users and groups for each database using the ZenCC GUI or SQL scripting.

The MKDE security model also leverages the user authorizations and authentication mechanisms provided by the underlying operating system environment. Additionally, advanced security options are available within Actian Zen that decouple database rights from the OS user access rights, relying instead on separate, database-level user credentials and rights.

Zen Encryption

All Zen data files support owner-name encryption to provide protection for data at rest. While previous Zen file format versions supported only AES-192-bit and AES-128-bit encryption, the most recent Zen file format (v13) supports AES-256-bit encryption for data on-disk.

Action Zen also supports the encryption of network traffic between the Zen client and the database engine via Blowfish—a well-known and time-tested public domain algorithm—implemented using 40-, 56-, and 128-bit keys. This type of encryption is often called wire encryption because it protects the data when it is traveling on the network “wire,” though it works across any network infrastructure, including wireless. While wire encryption is not required, it provides additional deterrence against unauthorized access to the data transmitted by your application.

Zen wire encryption is not tied to any particular security model. All Zen security configurations can be used with or without turning on wire encryption.

As with all security using encryption, the greater the deterrence, the slower the resulting performance, since some amount of processor time is needed to execute encryption and decryption algorithms. File encryption can be enabled on a file-by-file basis, while wire encryption can be disabled, set to always-on (for Wi-Fi, WAN, or public networks that you do not fully trust), or set on a client-by-client or server-by-server basis.

Zen Security Auditing Features

In addition to Zen’s built-in database user access management, Action AuditMaster, an add-on product for Zen, logs database events such as successful and failed login attempts and database operations, storing user ID, network station ID, time and date, application name, database name, table type, and table activity. It logs before-and-after images for all updated or deleted records and can even support a limited “undo” capability. Action AuditMaster works at the underlying MKDE level, so it is application-independent and can monitor and identify the original source of each database event in the audit trail.

Zen Deployment Models

Action Zen can be deployed in the office, in the data center, on the Edge, and in the cloud. It runs on servers, laptops, mobile phones, or complex equipment ranging from MRIs and vehicles to IoT gateways, cameras, sensors, and more. You name it: as long as there are sufficient resources to support the desired Zen offering—in terms of the operating environments and underlying hardware platforms (see Figure 6 on page 11)—you can run Action Zen.

Deploying and configuring Zen will depend on how you are embedding and dedicating Zen to an application (whether you are using it as a local data management engine, for example, or in a client/server mode). This section examines your principal deployment options.

Dedicated and Embedded in an Application

As seen in Figure 10, Zen Core, the smallest edition of the Zen database, has a ~4MB footprint that holds three embeddable library modules that deliver the core Zen database engine functionality. These modules consist of the MKDE, the SRDE, and a Zen client module that enables direct access to the Zen Edge, Zen Enterprise, and Zen Cloud Server editions. A developer/engineer embeds these modules directly into an application to enable Btrieve API calls or SQL queries from an application—either to the Zen Core engine embedded in the application or, via the embedded client module, to a remote Zen server. The data tables and any views of it are then spun up as a process within the application. They consume additional memory as needed, constrained only by the memory limitations of the device on which it is running. Each application can have its own separate data tables and supporting processes providing for data isolation.

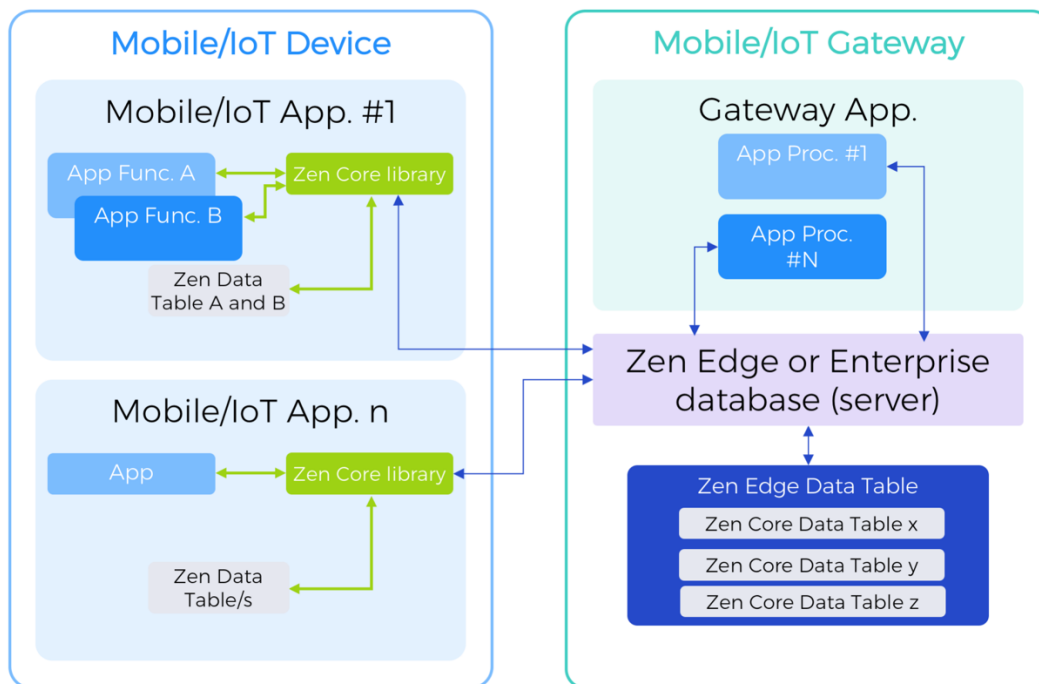


Figure 10: Zen Core Embedded in Stand-Alone Apps Connected to Zen Edge or Enterprise Server in a Client/Server Configuration

Because a developer can embed Zen Core in an application, the end user has no awareness of it. An application with Zen Core “feels” no different than any other file that runs in the target operating environment.

Stand-alone in Support of One or More Applications

In the application-embedded deployment model (above), a device’s local application logic would contain the embedded data management functionality and leverage a local Zen Core engine. A similar deployment model can be leveraged using the Zen Edge edition of Actian Zen—but here the application logic may not always tie to a local Zen database engine.

Developers using Zen Edge can separate out the Zen client application logic and run it remotely on other devices—with or without the Zen database engine running on the same device. Zen Edge can handle multiple database connections from any Zen client on any

supported platform requesting CRUD (NoSQL) or SQL operations. This deployment flexibility can be used where an application is distributed across multiple platforms at the Edge, for example, and needs to share data in a peer-to-peer fashion. Or, it can be used in a scenario involving aggregation upstream or dissemination downstream of critical data and metadata management and analysis to devices running the Zen client application logic on each downstream device.

Alternately, Zen Edge can support data sharing and fusion among multiple applications of different types. This scenario can be stateful (based on the application logic) or stateless (innately part of the Zen database capabilities).

Zen Workgroup, Zen Enterprise Server, and Zen Cloud Server can support these scenarios in the same way that Zen Edge does. These larger engines can support multiple local applications accessing data through any supported API. They can also service requests from an even greater number of Zen clients, enabling access to Zen Core and Zen Edge as downstream clients (acting as publishers of data going upstream or as subscribers to data going downstream). The end result is that in Actian Zen you gain a platform that can distribute, share, aggregate, and disseminate data across applications running on vastly different platforms, in different programming languages, yet all running within the constraints imposed by local resources.

Client/Server Mode and Multiple Deployment Instances

The Zen Enterprise and Zen Cloud Server engines can support thousands of clients (Zen Edge and/or Zen Core or third-party data services/requestors) that need upstream support from an edition of Zen server that can simultaneously handle concurrent data reads and writes between the clients and the servers. The users and processes accessing the data can be located directly on the server or they can be running on any supported device in the environment, including a desktop PC, Cloud server, IoT device on the Edge, or mobile device. As noted, all Zen engines support every API at the same time, enabling simultaneous high-speed, heads-down data entry from some clients, relational queries and reporting from other clients, and system management calls from still other clients.

This combination of performance and flexibility becomes extremely useful when Zen servers are deployed on the Edge as gateways (as depicted in Figure 11). It is easy to imagine how a complex environment may present several different downstream IoT grids. One might be for simpler, sub-Kbps data-rate sensors measuring pressures, volumes, or temperature reads. A separate grid might be engaging with sophisticated camera systems performing machine vision and generating enormous data sets at high speeds. A mixture of Zen servers—from Zen Cloud to Zen Enterprise to Zen Edge to Zen Core—can aggregate data at different levels for further analysis by servers or processes at other levels.

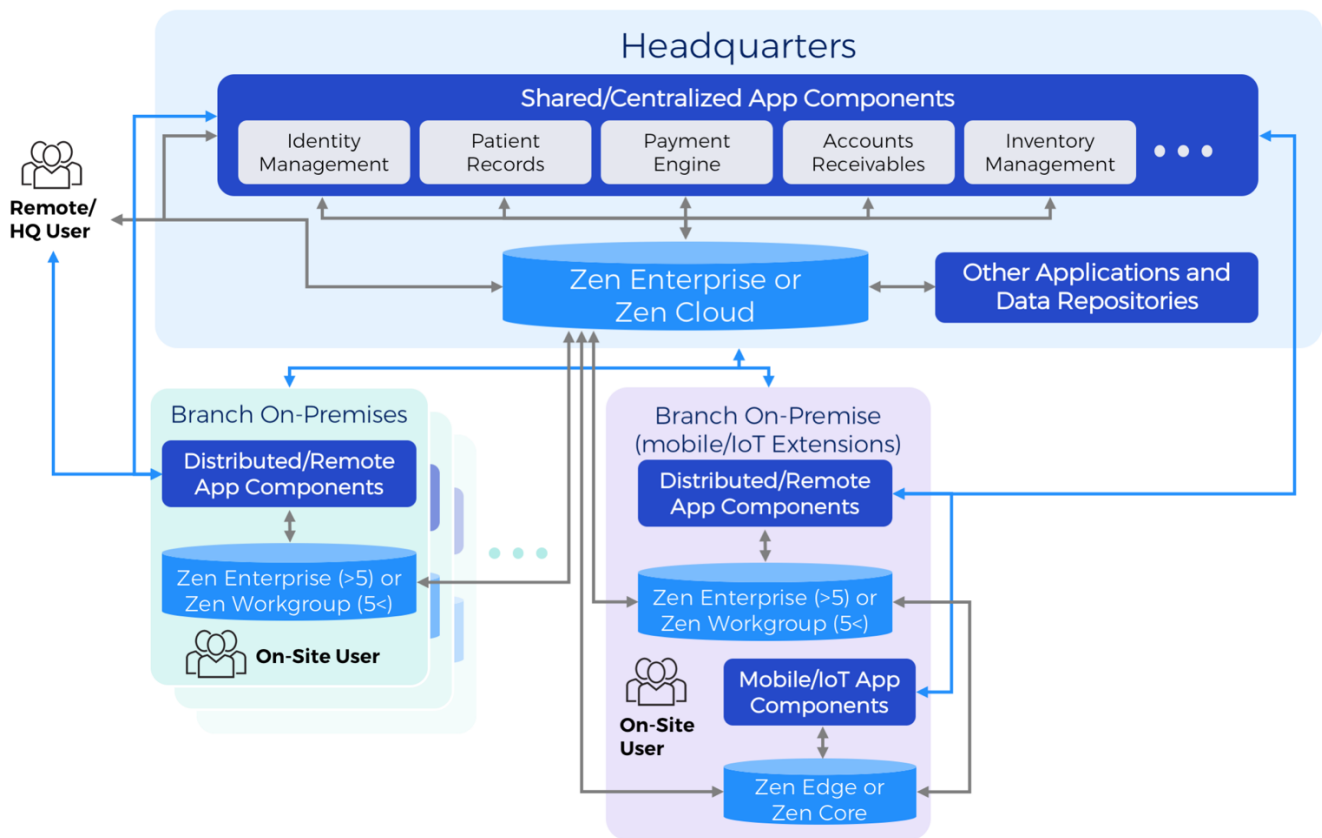


Figure 11: Custom/Vertical Multiuser Application Data Management

Zen Data Integration

Given the wide variety of systems, platforms, programming languages, and data formats currently in use in the world of Modern Edge Data Management, you will inevitably need to address the challenge of data integration. Sooner or later, you will need to read and write non-Zen data formats or pass data to non-Zen applications. Frequently this is accomplished by leveraging automation tools or writing custom code to perform ETL functions, either deep within your own application or through some other service.

ETL activities account for a significant portion of the time and cost associated with the development and integration of IoT systems and mobile applications. There is no way to avoid it. Data invariably needs to be moved to and from devices, and that requires some degree of extract and load. When, where, how, and why that data moves—from an event-driven perspective—are questions whose answers are often driven by business-specific use cases, where demands can greatly complicate application logic and the challenges of data management.

Action Zen can significantly reduce the time and effort associated with extract and load operations through the following key functionality:

- Standard querying and supporting functionality to speed up determination of what should be extracted or loaded.
- Automated timestamping, replication, and backup.

- Automatic reconnect if connection drops.

While Zen can accelerate the extract and load aspects of ETL, it's the *transform* part of ETL that presents the most time-consuming integration tasks. Moreover, the performance degradation and potential security vulnerabilities associated with transformation grow greater as data moves from system A to system B, from system B to system C, and so on.

When an all-Zen environment can be utilized, the transform part of ETL can in many cases be all but eliminated. Figure 12 below depicts a typical manufacturing environment in which different data management systems are used at different points—including flat files from resource constrained devices, SQLite for mobile devices, and other databases supporting various functions.

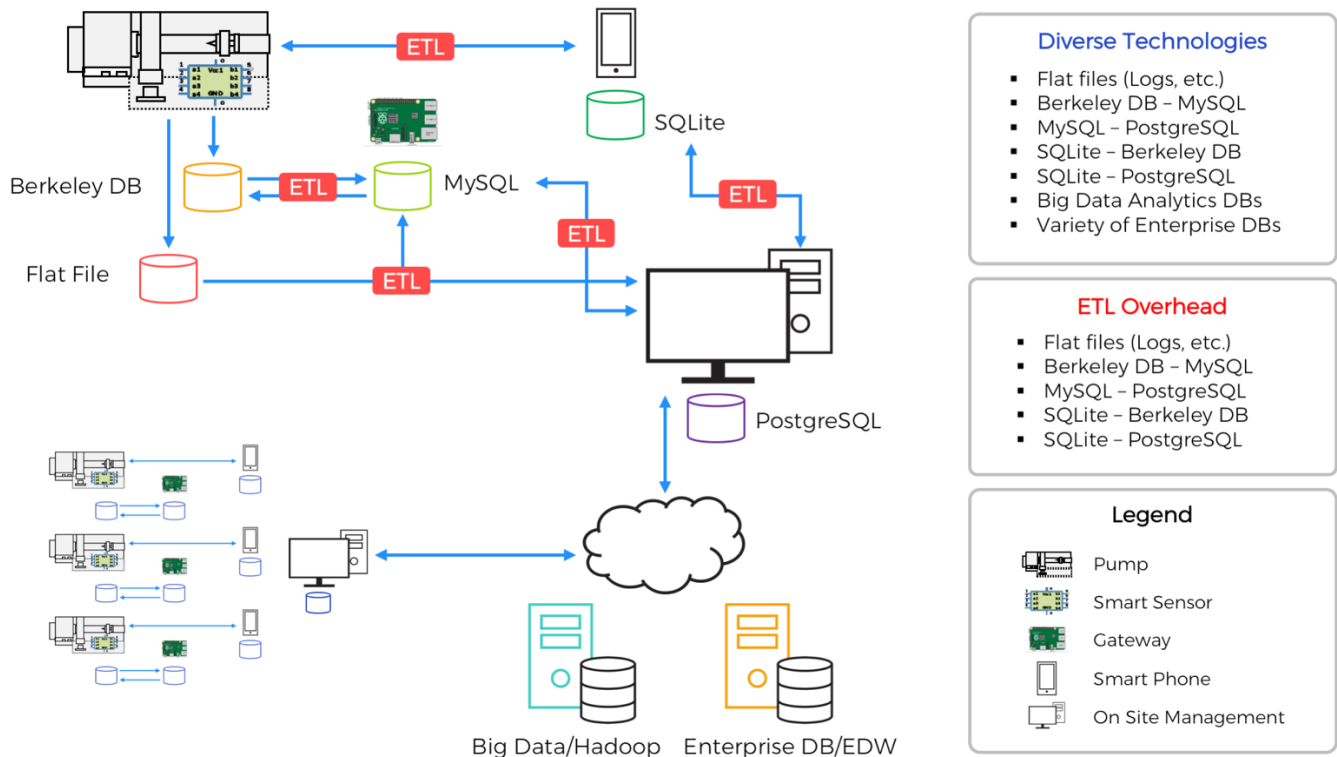


Figure 12: Heterogenous Environment with ETL

Compare the complexity of that scenario to the one presented in Figure 13, which substitutes an edition of Zen for each of these flat file and database systems. Zen's common data format across all supported platforms enables a simple file copy operation to replace the transform piece of ETL. Furthermore, the interoperability of client and server functionality across all Zen editions ensures that an application on any client can directly access data at any Zen server. This enables direct, real time, ad hoc access to data where it is being created.

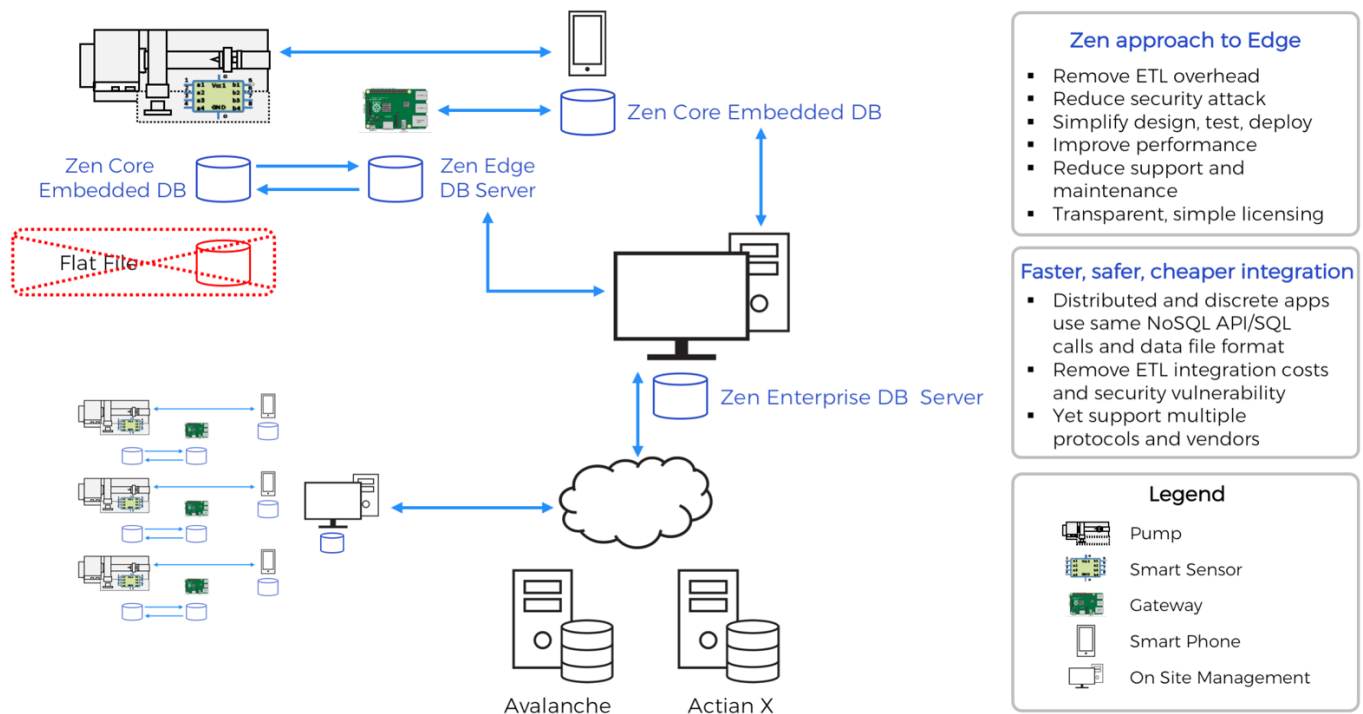


Figure 13: Homogenous Environment with Zero-ETL

An important point to keep in mind is this: With ETL there is inevitably a time lag during which you do not have access to the latest data. Using Zen throughout eliminates that lag time and enables you to drill down and poll the latest data on any device. With Zen installed, all your devices could access the latest instructions/algorithms from your control center directly, without waiting for a slower batch ETL process to migrate/sync the data across.

Zero-ETL Zen Environments

For simplicity, and because the bulk of the performance penalty and integration costs largely are incurred here, we use the term “Zero-ETL” to describe an all-Zen environment. By employing Zen in a homogeneous manner—particularly in your Edge configurations—you gain numerous advantages:

- Avoid security vulnerabilities introduced by code that decrypts and reorganizes and/or reformats data to meet the requirements of differing flat file and database standards.
- Avoids performance bottlenecks introduced by executing application logic on data loading and extraction around decompression, decrypting, and encrypting.
- Avoids the integration costs associated with designing and developing solutions across multiple architectures.

Application and Non-Zen Data Integration

When an all-Zen environment is not possible, Zen can still improve standard ETL integration. Actian Zen supports the typical interfaces used by common ETL solutions, including:

- **ADO.NET:** Zen provides an ADO.net SDK to support data access and command execution from ASP.NET, Windows, and Console applications. The Zen ADO.net SDK is certified up through Windows 2019 and for the ADO.NET Data Provider 4.3 or higher. Zen provides ADO.NET SDK Standard to support .NET applications that target .NET Core 2.0 or later and Windows Universal Projects.
- **JDBC:** For interactions between Zen and Java Applications, Zen supports the JDBC 4.1 driver (with backward compatibility to version 2) based on JRE 8 from OpenJDK for Windows, Linux, and macOS environments.
- **ODBC:** Zen provides an ODBC connector (ODBC v3.51 for Core levels 1 and 2) that works with any standard DBMS and provides SQL access independent of application API language or underlying operating environment.

These mechanisms enable you transfer data to and from nearly any application or database, JSON, XML files, and more.

Conclusion

Modern Edge Data Management presents a range of challenges that previous generations of data management systems are ill-equipped to meet—from working with a broad range of data types to working with a wide range of systems and sensors interacting in a complex combination of connected and occasionally-connected configurations. There are enormous volumes of data and enormous numbers of devices, all of which hold enormous numbers of opportunities for those who strive to capture them. Doing that successfully, though, requires a new kind of data management solution.

Action Zen is uniquely positioned to meet the needs of Modern Edge Data Management. It offers a portfolio of database editions that can cover the needs of most developers, data engineers, systems architects, and their line of business partners. Action Zen delivers:

- A single modular architecture whose database editions run the gamut from small-scale embeddable versions to full-blown cloud servers and whose capabilities range from supporting persistent data sets in embedded stand-alone applications to supporting concurrent, highly dynamic client/server interactions between thousands of upstream and downstream publisher/subscribers.
- SQL or NoSQL data access to any data type (JSON, BLOB, documents, traditional structured, and more) using a Micro Kernel Database Engine (MKDE) that employs a single consistent file format across all editions and that supports both encryption and compression as required.
- Access the MKDE and underlying Zen data files using the programming and scripting languages most relevant to Modern Edge Data Management, including C/C++/C#, Java, JavaScript, Python, Perl, Visual Basic, COBOL, and many more.
- Database access that is hardware and operating environment agnostic yet proven on 32-bit and 64-bit Intel and ARM platforms.
- Data files that are OS and platform agnostic and that can be shared across local and remote systems without decryption, decompression, or time-consuming data transformations.
- Simple yet comprehensive built-in support for local and remote management, backup, sharing, and auditing.
- Zero-DBA approach to field and remote deployment, supported by a built-in API for automated database tuning and configuration as well as built-in utilities to facilitate remote configuration and management.

Action Zen can deliver the capabilities that your technical staff is seeking and, in turn, can use to drive improved business outcomes for your entire organization.

About Actian—Activate your Data™

Actian, the hybrid data management, analytics, and integration company, delivers data as a competitive advantage to thousands of customers worldwide. Through the deployment of innovative hybrid data technologies and solutions, Actian ensures that business-critical systems can transact and integrate at their very best—on premises, in the cloud, or both. Thousands of forward-thinking organizations around the globe trust Actian to help them solve the toughest data challenges to transform how they run their businesses, today and in the future.

For more information or to see our products in action, visit <https://www.actian.com>.



2300 Geng Rd, Suite 150, Palo Alto, CA 94303
+1 888 446 4737 [Toll Free] | +1 650 587 5500 [Tel]



© 2020 Actian Corporation. Actian is a trademark of Actian Corporation and its subsidiaries. All other trademarks, trade names, service marks, and logos referenced herein belong to their respective companies. (WP05-0420)