Bloor

White Paper

White Paper by Bloor
Authors **Philip Howard**
Publish date **May 2018**

IBM Informix: the white swan

A black swan is an event or occurrence that *"deviates beyond what is normally expected of a situation and is extremely difficult to predict"*; the term was popularised by Nassim Nicholas Taleb. We have titled this paper *"IBM Informix: the white swan"* not just because the use of the Informix database should help users to sleep well at night (SWAN) but also because the entire history of Informix makes that outcome entirely predictable.

Introduction

Bloor Research has previously published reports looking at the suitability of IBM Informix for deployment in the cloud and in hybrid cloud environments, and we have also considered its potential role within Internet of Things (IoT) scenarios (see ***IBM Informix and the internet of things*** and ***IBM Informix on cloud***). In this paper we discuss a variety of other aspects of Informix. These are mainly, but not only, concentrated around things that help you to sleep well at night (SWAN) and take stress out of your day: high availability, resilience, disaster recovery, ease of administration, and so on. But this paper is not limited to such considerations. For example, knowing that you can scale your database when you need to, is another stress reliever. A further potential source of worry for users is whether their database will be able to cope with new and innovative technologies that may emerge in the future. In practice, Informix has a history of adapting to support new applications and use cases and, having already discussed Informix's role in the cloud and IoT, in this paper we will discuss how new features have been introduced into Informix to support Blockchain and other new use cases.

We will discuss these various issues in three sections: ease of administration; flexible grid, which enables both continuous availability and scalability; and smart triggers, which is the new feature that enables capabilities such as Blockchain.

“In this paper we will discuss how new features have been introduced into Informix to support Blockchain and other new use cases.”

Ease of administration

> **It is frequently the case that users of the applications in question have no idea that they are running on top of Informix, and nor should they.**

IBM Informix has a long history of being deployed as the underpinning for a variety of applications, typically developed by third party software houses. It is frequently the case that users of the applications in question have no idea that they are running on top of Informix, and nor should they. The point is that the database is invisible to the user. However, this is only possible if the database has relevant features, typically summarised by the expression *"fire and forget"*. Informix's longevity in supporting such environments provides strong evidence that this is exactly how the database should be viewed: that it is easy to implement and deploy, that it performs appropriately, that it is robust and does not fall over, that it is flexible enough to support different deployment models, and that it is easily updated (without appreciable downtime) when requirements change. In so far as IBM Informix is concerned, most of these can be taken as given. However, there are a couple of points worth discussing.

Firstly, it is essential that databases that underpin applications – that is, databases embedded within the application – are invisible to the user at all times. This is true regardless of whether you are simply collecting data and passing it on (in IoT implementations, for example) or in transactional environments, or if you are performing some analytics on the data. Particularly with respect to analytics, to get good performance, you need, at least in most environments, to create indexes, materialised views and other such database constructs to achieve the necessary performance. While this is feasible, it is not flexible enough when new or additional requirements need to be supported. Every time you add functionality you need to change the supported indexes. Worse, different workloads may mean that different indexes, materialised tables and so forth will be differently suitable for different customers. Moreover, these workloads may change over time. What this will mean is that the database will need to be tuned on an ongoing basis to maintain performance, which is impractical in many environments, and is often undesirable (for example, in cloud deployments). For these reasons a traditional relational database will not be suitable for embedding at any level within your infrastructure, precisely because these all require exactly this sort of tuning. Fortunately, this is not the case with IBM Informix because the product only requires minimal tuning.

Flexible Grid

IBM Informix is frequently deployed in environments that are mission critical. To support such environments, IBM Informix provides several options, including High Availability Data Replication (HADR), Remote Secondary Standby Database Servers and Shared Disk Secondary Servers. As stand-alone capabilities, each of these should be self-explanatory. However, an option that combines the last two of these is known as Flexible Grid, which is worth discussing in more detail.

The Flexible Grid supports the definition of a multi-node heterogeneous cluster that makes it possible to run an application on any node within the grid. This means that you can have a geographically dispersed environment with different commodity hardware (and operating systems) implemented in different locations depending on need, and yet have the whole environment centrally controlled. An example of this is illustrated in ***Figure 1*** where clusters of processing nodes are distributed across several continents. Note that each individual cluster is using remote secondary standby (RSS), high availability disaster recovery (HADR) and shared disk secondary (SDS) servers. Replicated data may be stored either on premises or in the cloud. It is also worth mentioning the support for shared disk mirroring. The connections load balancer optimises performance between the clusters.

Not only are DML (data manipulation language) operations replicated, but so are DDL (data definition language) operations. This means that when a table is created on one node, it is automatically created on all nodes within the Flexible Grid. In addition, it is possible to automatically start replication of the table as soon as it is created, which greatly simplifies the deployment of a replication solution. A major feature of the Flexible Grid is that it supports continuous availability. That is, operations can continue regardless of whether downtime is planned or unplanned.

For example, Game Show Network uses the Informix Flexible Grid and – at the point that we spoke with them – had had no unplanned (or planned) downtime for two years. According to Susan Marciano, Vice President of Technical Operations: *"the Flexible Grid feature of Informix enables us to perform rolling upgrades without any outages, so players can go on playing with no interruption and no impact on our revenue. That's worth its weight in gold."* Other use cases occur wherever 24x7 operations are critical and, especially, where costs are a major factor, since the Flexible Grid runs on commodity hardware and is therefore well suited to deployment either on premises or in the cloud. Minimal administration is a further major benefit along the same theme.

"the Flexible Grid feature of Informix enables us to perform rolling upgrades without any outages, so players can go on playing with no interruption and no impact on our revenue. That's worth its weight in gold."

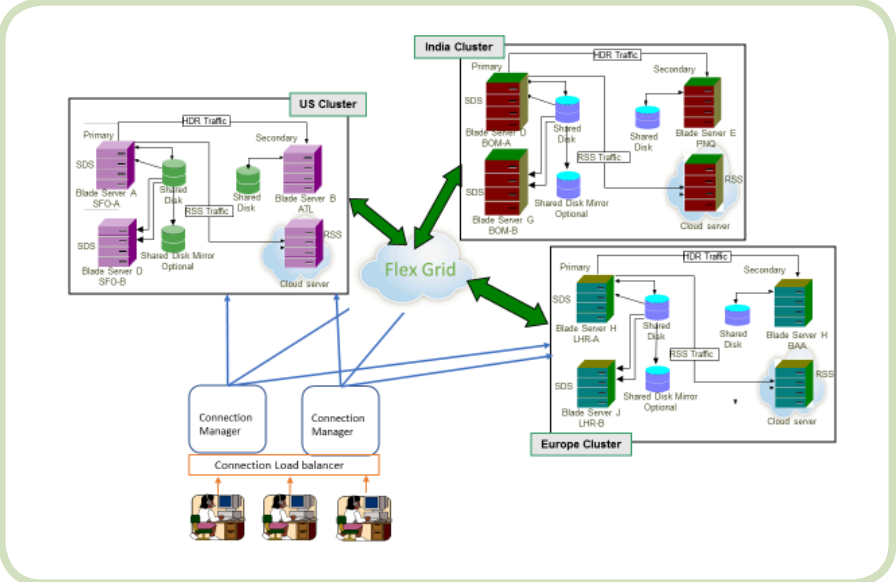

Figure 1: Informix Replication Reference Architecture

Smart triggers

> **Smart triggers extend this concept by creating notifications that are pushed outside the database whenever a relevant event takes place.**

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. Usually, these are used to maintain the integrity of the database. Smart triggers extend this concept by creating notifications that are pushed outside the database whenever a relevant event takes place. To take a simple example, suppose you are a bank and you want to be alerted whenever an unusually large deposit is made into a client's bank account: perhaps because you are worried about money laundering, or because you want the opportunity to offer financial advice. Using smart triggers, you can define a trigger within your application – using SQL select statements and predicate clauses, implemented within the Informix database – which will notify you, via what are known as *"push notifications"*, whenever such a deposit is made. In other words, smart triggers are a way of implementing event-driven processing.

Smart Trigger	Regular Trigger (Insert/Update/|Delete)
Post Commit	Pre Commit
Register Trigger on a specific Dataset/Event	Trigger gets fired for all changes
Asynchronous and Linear Scalability	Synchronous
Data is in JSON format	SQL format
Triggered action gets executed in the client	Triggered action gets executed in the server
Natural fit for event driven programming model	–
No schema changes required to define new smart trigger	Require schema changes and exclusive lock on the table to modify trigger definition

Table 1: Smart triggers

Note that, unlike conventional database triggers, push notifications and smart triggers work specifically with committed data. This is important because some data that is presented to the database server is rejected, rolled back or not accepted for some reason. Push notifications are defined in the Informix server (but are created in the client application) where the definition is a simple SQL statement, and where the SELECT clause determines which bits of selected data to push and the WHERE clause determines the specific database rows to push. Smart triggers provide a Java API to specify a call-back function that gets called when the push criteria are satisfied.

There are other differences between smart triggers and traditional database triggers. For example, the former do not require any of the schema changes that are associated with conventional triggers, and this makes them much more flexible and user-friendly than conventional triggers. A summary of the differences between smart and regular triggers is provided in ***Table 1***.

While smart triggers have potential applicability in many environments and for multiple uses cases, one of the more innovative environments in which they will be important is in supporting Blockchain.

Blockchain

There are plenty of definitions of blockchain available on the Internet (for example, see Wikipedia). To summarise, blockchain provides a secure, immutable, distributed mechanism to store information of interest for multiple relevant parties. The secure nature of blockchain means that there is growing enthusiasm for this technology. A practical example of its use might be in a large retailer's supply chain, where its suppliers, logistics companies and others, are all providing information about the manufacture and transportation of goods that are coming to the retailer. The blockchain records the chain-of-custody for each of these shipments of goods as it passes from the manufacturer through

various middlemen until reaching the retailer. The blockchain records each of these changes in custody, along with information about the shipment (condition, temperature, weight and so forth) at each stage. Information in the blockchain can trigger payment to the manufacturer as the block is added which shows that the transport company picked up the shipment in good condition.

Information that gets added to a blockchain is generally the same data that is stored in an organisation's enterprise database (system of record) using existing business processes. Informix smart triggers can be created to automatically push specific data to the blockchain smart contract once the data is committed to the database using these normal business applications. For instance, in the retailer example discussed above, a supplier that generates widgets for the retailer may create monthly shipments to the retailer. The supplier can configure a smart trigger to push the appropriate data to the retailer's blockchain as soon as the shipment data is committed to the supplier's Informix database. This mechanism allows the supplier to participate in the retailer's blockchain with almost no impact to their normal business processes.

Figure 2 illustrates the architecture, in more abstract terms based upon an IoT environment. In fact, this is a security application developed by SmartAxiom, an IBM partner that has developed this solution. What is happening in ***Figure 2*** below is that some sort of information (packet) is received by an edge device.

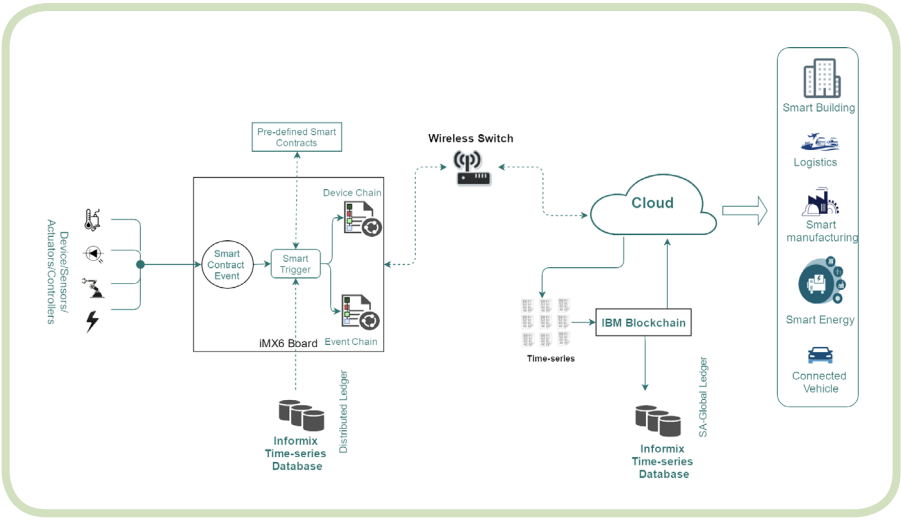

Figure 2: Blockchain architecture

This has IBM Informix embedded within it and a push notification is generated automatically when the data is received, supporting the creation of a *"block"* that passes through the Blockchain process. Once received in the global general ledger a further Informix smart trigger might be activated to initiate, say, a payment or another action. ***Figure 3*** shows the overall multiple chain architecture with different types of edge devices embedding Informix database.

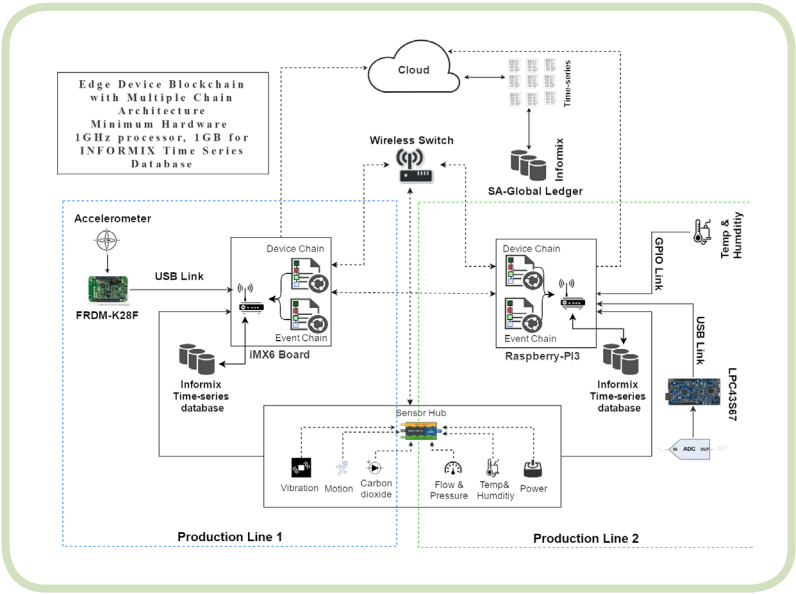

Figure 3: Multiple chain architecture

Conclusion

This paper has omitted discussion of many of the significant features of IBM Informix. We have not discussed, for example, the Informix Warehouse Accelerator, its support for unstructured data (though the support for JSON is implied by the product's implementation of smart triggers), or its geo-spatial and time series capabilities, which are of particular importance in IoT environments. All of these have been discussed in our previous papers. With the exception of smart triggers, what we have been discussing here are what are sometimes known as the *"ities"* or *"ilities"* such as usability, scalability, reliability and so on, those fundamental capabilities that allow database administrators to sleep well at night (SWAN). They provide the sort of enterprise class capabilities that are required in mission critical environments. We could even include the introduction of smart triggers in this category as an example of Informix's adaptability in supporting new environments and requirements such as blockchain.

A black swan is an event or occurrence that *"deviates beyond what is normally expected of a situation and is extremely difficult to predict"*; the term was popularised by Nassim Nicholas Taleb. We have titled this paper ***"IBM Informix: the white swan"*** not just because the use of the Informix database should help users to sleep well at night but also because the entire history of Informix makes that outcome entirely predictable.

What we have been discussing here are what are sometimes known as the *"ities"* or *"ilities"* such as usability, scalability, reliability and so on, those fundamental capabilities that allow database administrators to sleep well at night (SWAN).

FURTHER INFORMATION
Further information about this subject is available from
www.bloorresearch.com/update/2363

About the author

PHILIP HOWARD
Research Director / Information Management

Philip started in the computer industry way back in 1973 and has variously worked as a systems analyst, programmer and salesperson, as well as in marketing and product management, for a variety of companies including GEC Marconi, GPT, Philips Data Systems, Raytheon and NCR.

After a quarter of a century of not being his own boss Philip set up his own company in 1992 and his first client was Bloor Research (then ButlerBloor), with Philip working for the company as an associate analyst. His relationship with Bloor Research has continued since that time and he is now Research Director, focused on Information Management.

Information management includes anything that refers to the management, movement, governance and storage of data, as well as access to and analysis of that data. It involves diverse technologies that include (but are not limited to) databases and data warehousing, data integration, data quality, master data management, data governance, data migration, metadata management, and data preparation and analytics.

In addition to the numerous reports Philip has written on behalf of Bloor Research, Philip also contributes regularly to ***IT-Director.com*** and ***IT-Analysis.com*** and was previously editor of both ***Application Development News*** and ***Operating System News*** on behalf of Cambridge Market Intelligence (CMI). He has also contributed to various magazines and written a number of reports published by companies such as CMI and The Financial Times. Philip speaks regularly at conferences and other events throughout Europe and North America.

Away from work, Philip's primary leisure activities are canal boats, skiing, playing Bridge (at which he is a Life Master), and dining out.

Bloor overview

Technology is enabling rapid business evolution. The opportunities are immense but if you do not adapt then you will not survive. So in the age of Mutable business Evolution is Essential to your success.

We'll show you the future and help you deliver it.

Bloor brings fresh technological thinking to help you navigate complex business situations, converting challenges into new opportunities for real growth, profitability and impact.

We provide actionable strategic insight through our innovative independent technology research, advisory and consulting services. We assist companies throughout their transformation journeys to stay relevant, bringing fresh thinking to complex business situations and turning challenges into new opportunities for real growth and profitability.

For over 25 years, Bloor has assisted companies to intelligently evolve: by embracing technology to adjust their strategies and achieve the best possible outcomes. At Bloor, we will help you challenge assumptions to consistently improve and succeed.

Copyright and disclaimer

Bloor Research International Ltd
20–22 Wenlock Road
LONDON N1 7GU
United Kingdom

Tel: **+44 (0)20 7043 9750**
Web: **www.bloorresearch.com**
Email: **info@Bloor.eu**