ICE InsureTech Core Architecture

a look at how we have adapted “The New Architecture” to suit our insurance solutions

ice
ACTURIS GROUP

Al Robertson
Chief Technology Officer

ICE InsureTech Core Architecture

a look at how we have adapted “The New Architecture” to suit our insurance solutions

CONTENTS

Introduction

Following on from an excellent recent report from Celent called “The New Architecture for Core Systems“, we thought it would be appropriate to introspectively review how ICE’s architecture compared with this trend.

The Celent report highlights three key architectural trends in modern application design: Cloud technologies, Public / Private APIs and Microservices.

We have taken the comparison further to review our journey from Cloud Ready to Cloud Native, and why we believe that ICE has delivered a leading platform architecture for Insurance Core Systems.

Cloud Native vs Cloud Ready

ICE InsureTech have always supported various deployment scenarios; on-premise, third-party datacentre, and more recently public cloud. The use of the term "The Cloud" has become ubiquitous in modern IT speak, but essentially "The Cloud" really is just "*someone else's computer*".

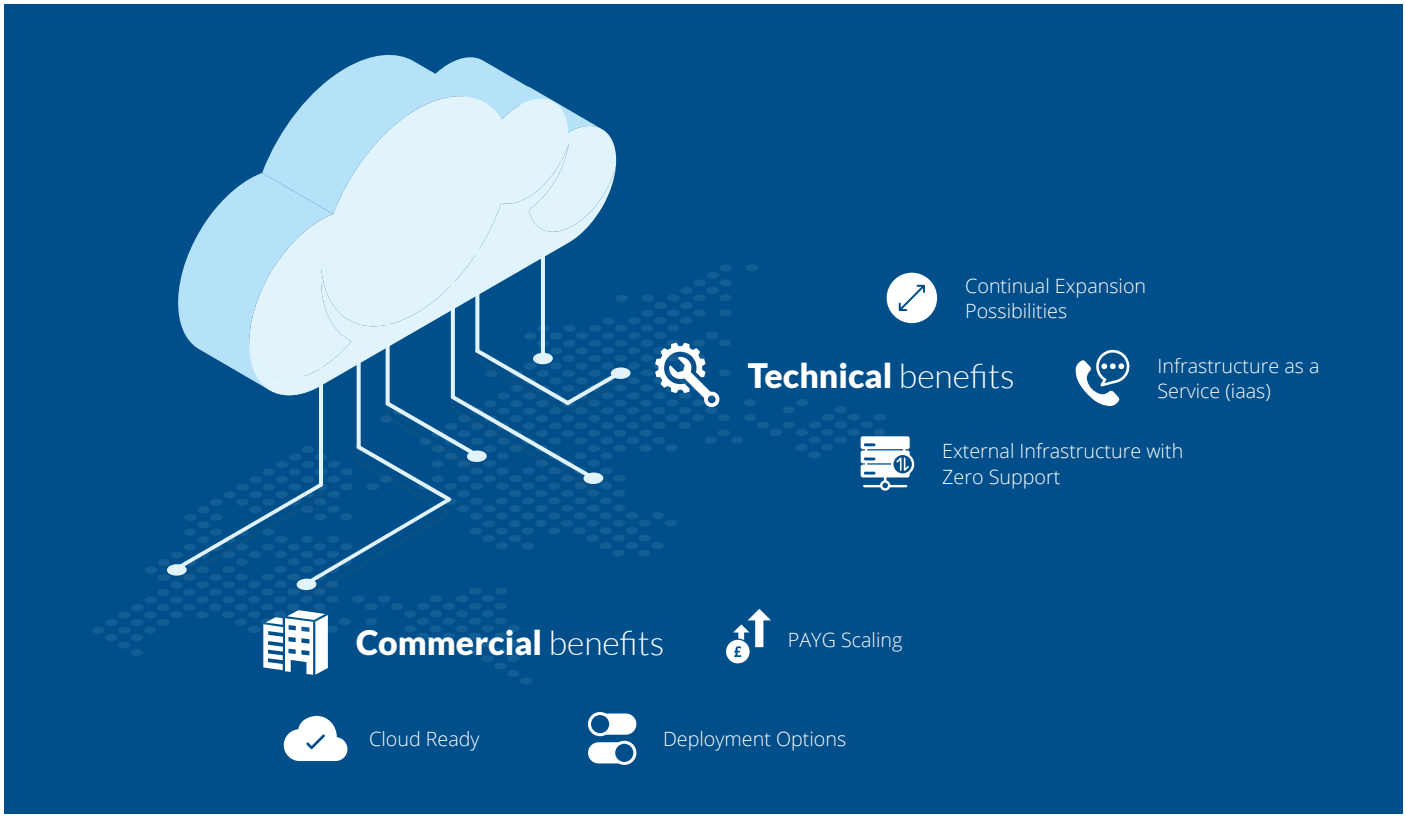
The ability to deploy our applications to the public cloud makes them Cloud Ready. Being Cloud Ready only really offers a small set of advantages over any traditional hosted application – essentially just using The Cloud as an offsite compute and data storage facility, and lowering the wait times that can occur with infrastructure procurement and on-premise hosting.

SO WHAT IS CLOUD NATIVE?

Despite its name, a cloud-native approach is not focused on where applications are deployed, but instead on how they are built, deployed and managed.

Cloud-native application development is an approach to building and running an application that can take full advantage of the cloud computing model based on four key tenets:

1. **Service Based Architecture**
   - Microservice or SOA
2. **APIs For Communication**
   - Public / Open Business / Private APIs
3. **Use of Containers**
   - Linux Containers / Container Images - Packaged Applications
   - Container Orchestration
4. **DevOps**
   - Continuous Integration / Continuous Delivery
   - Auto scale, Infrastructure as a Service (IaaS)



AND WHAT ARE **THE BENEFITS?**

According to the Cloud Native Computing Foundation (CNCF) a true Cloud Native strategy is about agility, lowering risk, scalability and resilience:

Agility

- Speed of delivery through to production
- Automation of build, test and deployment (CI/CD)
- Ability to upgrade components independently

Lower Risk

- Containers are packaged / portable applications
  - Helps to avoid "works in dev, fails in production" scenarios
- Ability to upgrade smaller components individually and more frequently
- Security benefits of container approach
- Simplified patching strategy

Scalability

- Microservice architecture for independently scalable components
  - Aggregator Gateway
  - B2C + B2B Portals
  - Quote & Buy API
  - Policy Servicing API
- Container orchestration platform – ability to scale on demand
- Availability of near limitless infrastructure in Public Cloud
- Hybrid Cloud capability to extend from On-Premise to Public Cloud

Resilience

- Declarative approach to policies for High Availability, Quotas etc.
- Platform Automation: Health Checks / Failover / Recovery / DR

ICE's journey to **Cloud Native**

In 2004 all ICE deployments were on "bare-metal" infrastructure, hand crafted to achieve maximum efficiency from the computing resources available.

Around ten years ago we progressed to deployment on Virtual Machines (VMs), with every architectural component deployed inside its own VM. This enabled each component to be templated and run in isolation, with the trade-off of additional resources needed to host a full operating system, maintenance overhead of the OS, middleware, application and configuration on every VM.

LINUX **CONTAINERS**

2 years ago ICE InsureTech started using Linux container technology, a significant progression in our cloud native strategy.

Linux containers are lightweight and efficient when compared to more traditional virtualisation solutions. Many containers can run on the same host sharing the OS kernel with other containers, each running as isolated processes in user space. Containers use less resource than traditional VMs to make more efficient use of infrastructure and they start almost instantly. They are also extremely portable across Clouds and lend themselves perfectly to distributed architectures.

Docker (**https://www.docker.com**) introduced the concept of a container image. A container image is a portable, packaged application. It provides several features which enhance security, governance, automation, support and certification over the entire application lifecycle without fear of architecture and infrastructure lock-in.

Containerisation could be the subject of a white paper on its own – the following resource is a great guide to the essentials: **https://opensource.com/article/18/8/sysadmins-guide-containers**

RUNNING **CONTAINERS**

Any organisation wanting to run multiple containers in a production environment is going to need to manage them effectively.

Container orchestration (dynamic container management) is the principle of using a software based controller to manage container life-cycles and networking in order for them to run as intended.

ICE InsureTech selected OKD (**https://www.okd.io**) and OpenShift for our enterprise-grade container orchestration platforms. OKD is a packaged version of Kubernetes (**https://kubernetes.io**), an open-source solution for automating deployment, scaling and management of containerised applications. It was originally designed by Google and now maintained by the CNCF. OKD aims to provide a "platform for automating deployment, scaling, and operations of application containers across clusters of hosts". It works with a range of container technologies, including Docker. OKD adds developer and operations-centric tools on top of Kubernetes to enable rapid application development, easy deployment and scaling, and

long-term lifecycle maintenance for small and large teams.

The use of OKD also allows the ICE solution to automatically scale components to suit variable demand and make use of Infrastructure as a Service (IaaS), if available.

Auto-scaling using IaaS is especially important where there are significant variations between peak and trough demand. It lowers the requirements for capital expenditure, not having to provision infrastructure for peak demand periods and allows us to offer the flexibility of a Pay-as-you-Go (PAYG) model. We believe this is an important feature of the ICE solutions.

Our use of OKD also provides high availability in compute and storage services. OKD is continuously monitoring the health of PODs (microservices or architectural components) and if it detects an issue with a component, it will remove it from operation, spin up a replacement POD, and when readiness checks have successfully completely, it will add the new POD to the operational cluster. This feature is fully automated with zero manual IT intervention.

Crucially, Red Hat also provide a commercially supported offering of OKD called OpenShift (**https://www.openshift.com**) which affords our customers significant piece of mind when they have ICE InsureTech as a first line of fault resolution but also Red Hat to support OpenShift on Red Hat Enterprise Linux (RHEL).

Another aspect of cloud native applications is service based architecture. There are big debates on the move from SOA to microservices...

Main Benefits of a Container based Architecture

- **Agile application creation and deployment:** Increased ease and efficiency of container image creation compared to VM image use.
- **Continuous development, integration, and deployment:** Provides for reliable and frequent container image build and deployment with quick and easy rollbacks (due to image immutability).
- **Dev and Ops separation of concerns:** Create application container images at build/release time rather than deployment time, thereby decoupling applications from infrastructure.
- **Observability:** Not only surfaces OS-level information and metrics, but also application health and other signals.
- **Environmental consistency across development, testing, and production:** Runs the same on a laptop as it does in the Cloud.
- **Application-centric management:** Raises the level of abstraction from running an OS on virtual hardware to run an application on an OS using logical resources.
- **Loosely coupled, distributed, elastic, liberated micro-services:** Applications are broken into smaller, independent pieces and can be deployed and managed dynamically – not a fat monolithic stack running on one big single-purpose machine.
- **Resource isolation:** Predictable application performance.
- **Resource utilization:** High efficiency and density.

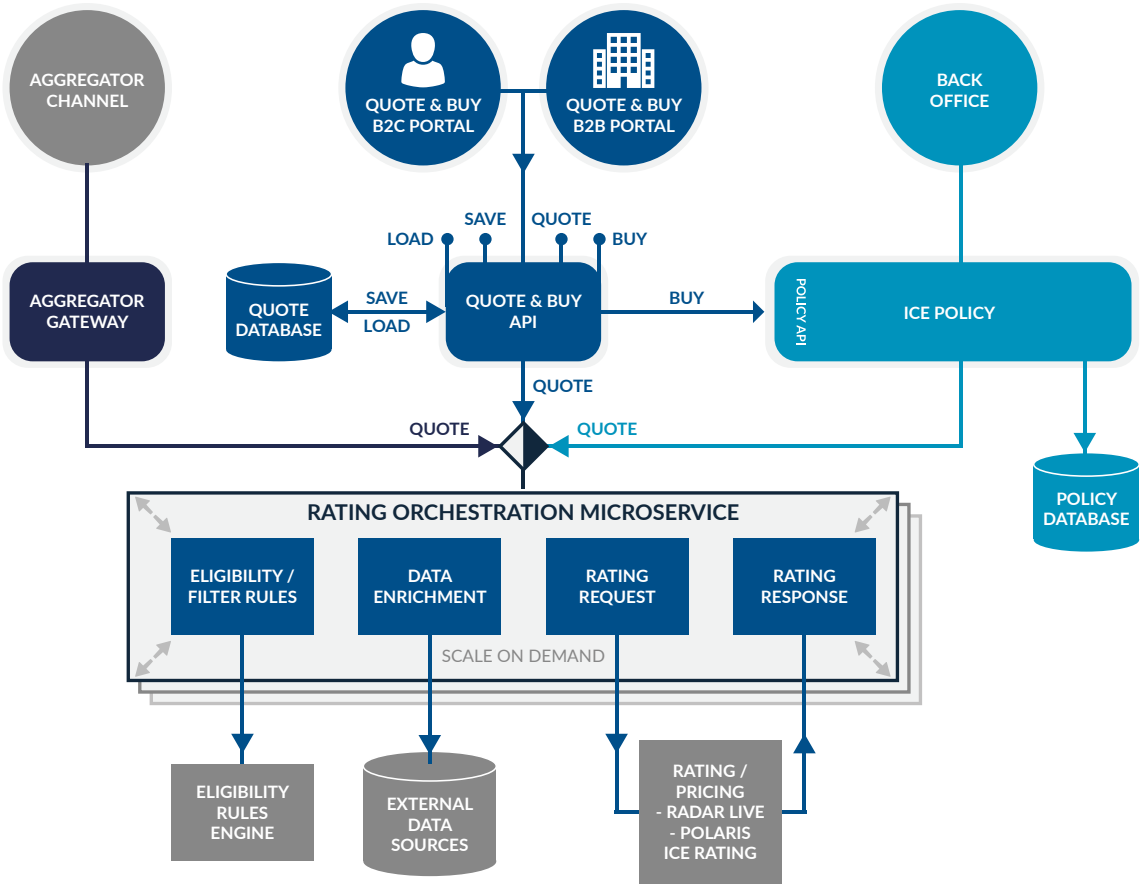
Microservices vs SOA

Microservice based architectures are currently in vogue. They "deliver small, discrete, and individually deployable services". One of the key benefits of microservices is the ability to independently deploy functional components, theoretically enabling improved agility in release cycles of individual microservices vs a releasing a full stack.

ICE has been architected from its inception to be assembled from thousands of services using a Service Oriented Architecture (SOA) approach on top of a single domain model.

We believe it is prudent not to rush to a microservice architecture. There are potential negative performance and operational complexities of a microservice architecture that may not be suitable for all aspects of an insurance solution. Our current microservice strategy is to target functional components where independent scaling is a high priority, e.g. rating orchestration and enrichment, aggregator gateway, quote and buy API etc.

This will allow us to deploy a topology best suited for the demands placed on an environment.



Application Programming Interfaces (API'S)

An API allows components to interact through clearly defined methods. In an insurance platform, it can be crucial to expose key services to third party systems and components that require access to core platform functionality. Examples of these services are Get Policy Details, Create Quote, Create FNOL, etc. To ensure we have the capability to do this securely and swiftly, the ICE suite has a combination of Public, Open Business and Private APIs.

Private APIs are for "internal use" only – they are not documented and liable to change without any notice.

Public APIs, on the other hand, are documented and versioned. Both private and public APIs are secured with our security framework.

Open Business APIs are used for integration with trusted partner systems, e.g. ICE / Third Party Portals, Client ESB's etc. These are secured at the protocol level and are never available to the public web.

We have always maintained that clients should be able to choose the "best of breed" components for their application suite, and connect them to perform in the most optimal way for their business.

Every ICE installation is accompanied by a wide array of integration components to provide APIs, access key industry data sources and third party systems; e.g. rating engines, document composition / storage, payment gateways and General Ledger systems. We achieve this using the ICE Enterprise Service Bus (ESB). The ESB hosts all the integration components for the client installation and is a key architectural component in the ICE suite.

Conclusion

At ICE InsureTech, we believe in continuous improvement, and our core platform architecture is no exception. Technology is evolving, at a greater pace now than ever before. When do you jump on the train? If you jump at every next big thing, you will be forever refactoring. If you wait for the perfect architecture / framework / pattern to come along, you will never innovate or improve.

This paper presents an insight into the ICE architecture and how we have evolved to a next-generation platform as defined by Celent, and the benefits, pitfalls and the applicability of a cloud native strategy in a multi-channel insurance platform. We have heavily invested in our platform to take advantage of these technological trends and our clients are reaping the benefits.

This is all possible because of the investment of industry giants such as Google, Red Hat and Docker etc. It is widely accepted that Kubernetes has won the container orchestration platform race and it was the most discussed open source project on GitHub in 2017. There is a tidal wave of noise around the adoption of containers, Docker, Kubernetes and OpenShift.

Each of our eight new clients in the last 12 months has selected the ICE cloud native option using OpenShift or OKD. Virtual Machine based deployments appear to be a thing of the past.

We have a great team combining thought leadership in insurance and technology, and an enviable track record for timely delivery, speedy implementation and customer satisfaction.

#ThereIsAnAlternative


ice
ACTURIS GROUP