

# Mo.net

## Introducing Mo.net Quotations Service

November 2020  
Revision 14

### Background

The lifecycle of a traditional life insurance product includes a variety of related but slightly different calculations. These range from pre-sale illustrations & new business quotations, through to internal & regulatory reporting functions, and back-office claims, surrenders and & transfers. Each stage of the lifecycle is generally supported by its own distinct set of benefit / value calculations.

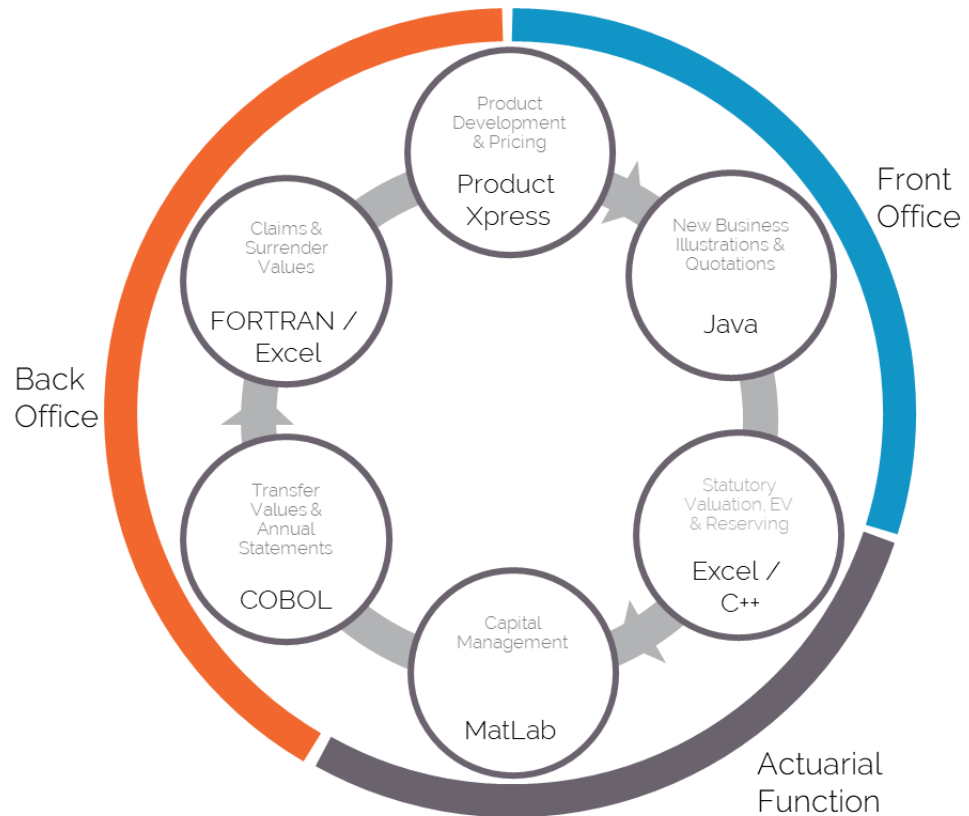


Figure 1 – Typical End-to-End Insurance Product Lifecycle

These calculations have often been implemented using the most appropriate (or fashionable) technology / language at the time, ranging from Java in the front office, through to proprietary tools / code in the financial reporting function, and FORTRAN or COBOL in the back office, all invariably supported by a modest collection of spreadsheets.

Despite the range of systems in play, the calculations themselves may actually be quite similar, albeit using different approaches (e.g. commutation functions vs cashflows) and consuming a variety of formats / granularity of data & assumptions.

## The Problem

There are a number of potential issues & concerns with the current end-to-end calculation environment. These include, but are not limited to:

- Inconsistency of calculations, approach, data and technology leading to significant reconciliation challenges / overheads
- Limited and often declining knowledge base, especially relating to systems / calculation modules developed in the 1970s
- Little or no transparency of calculations – black boxes requiring “shadow” calculations in more accessible systems, such as spreadsheets
- Poor performance, concerns over quality / current of data and therefore calculation results
- Complexity of environment – calculations tightly coupled / integrated with consumer systems
- Ability to change calculations quickly and effectively
- Significant IT debt associated with the end-to-end product lifecycle; little or no ability to integrate with other parts of the enterprise
- Regulatory change burden – ability to extend
- Appropriateness for new product features / designs

The most obvious implications of the challenges above are a high cost of support / change, and a high risk of failure / outage.

## The Opportunity

Considering the fundamental business support requirements of the end-to-end product lifecycle and the shortcomings / challenges of current solutions, a more appropriate target state solution might reasonably consist of the following features:

- Consistency of calculation cost base – both in terms of approach and implementation approach – potentially allowing a single calculation kernel for a given product throughout its end-to-end lifecycle
- Transparency and accessibility of calculations requiring a modest set of commonly available skills to develop / support
- Optimised calculations that can support all modes of operation throughout the lifecycle – on demand or in batch
- Highly reusable / modular calculation components making change / new developments as simple & efficient as possible
- Integration with any part of the insurance enterprise – calculations accessible from anywhere using a loosely-coupled architecture

## Modelling as a Service

Financial modelling platforms have traditionally been constrained by the walls of the actuarial or risk function, with little or no possibility of integration with other parts of the insurance enterprise. This has been largely due to the monolithic, closed architectures of most legacy modelling platforms.

The loosely coupled, open architecture of Mo.net together with the modest footprint of user developed models enables these calculations to be deployed "as a service" into each stage of the insurance lifecycle above, using a simple but rich API.

An Application Programming Interface ("API") is a means by which secondary applications can access exposed functions / features of a primary application without the need to understand underlying functionality. In modern systems this is often achieved via a web service.

## Mo.net Enterprise Services

Mo.net Enterprise Services, which include Mo.net Quotations Service (see Figure 2) and Mo.net Execution Service allow compiled models to be published to either or both enterprise services and consumed by one or more applications, either on demand using the Mo.net Quotations Service, or in batch using the Mo.net Execution Service.

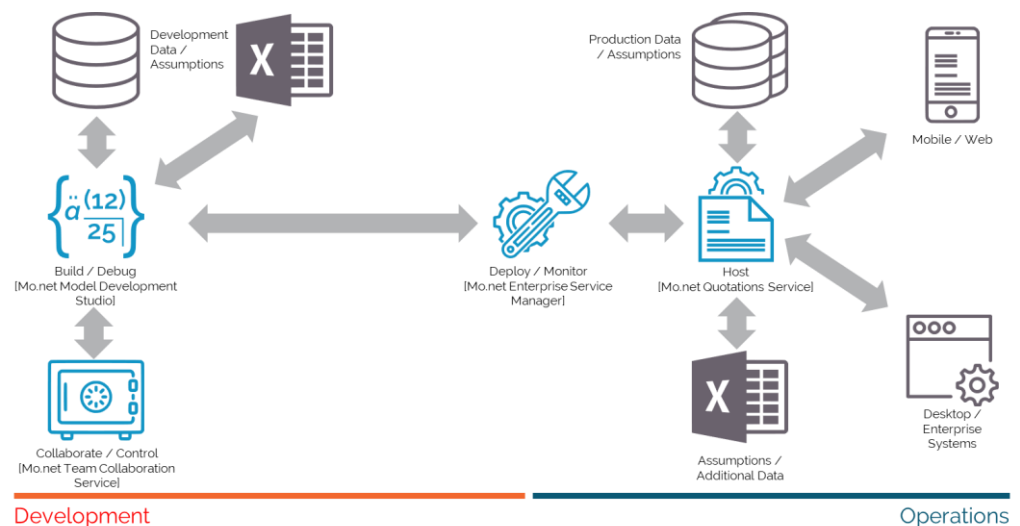


Figure 2 – Publishing a Mo.net Model to the Mo.net Quotations Service

## The Quotations Service

The Mo.net Quotations service allows any user developed model or calculation that has been published, to be consumed on demand (policy by policy) by any consumer application – typically a customer / IFA-facing website, mobile application or back-office administration system.

The service implementation allows for native workload distribution across many cores (to address any scalability considerations), and also includes built-in load-balancing functionality through an associated "router service" to ensure a responsive service to consumer applications.

Windows Communication Foundation ("WCF") is a framework that enables developers to write service code once and allows that code to run in a number of ways (SOAP/XML, REST/JSON, TCP Socket, etc.) via configuration

Net TCP is a binary-encoded transport over TCP for WCF generally used for intranet applications (inside the corporate firewall). RESTful services are typically used for communication between services over the intranet.

## Typical Implementation

A typical on-premise implementation at a client site with one or more consumer applications would normally include the following components:

- One Mo.net Model Development Studio (Enterprise Edition) instance to enable model / calculation development, testing and publishing
- One instance of Mo.net Enterprise Service Manager to promote the published models to the Quotations Service, monitor active workload, and download live calculation / model requests for analysis / review back in the development environment
- One or more Mo.net Quotations Services (SOAP or RESTful service) installed as Windows Services on one or more servers. If more than one Quotations Service is being used, the Quotations Service Router Service would also be installed.

The Quotations Service itself can be accessed either using SOAP / net.tcp (across the intranet, inside the corporate firewall) or RESTful (across the internet) protocols.

## Benefits

The benefits of adopting the Mo.net Quotations Service are potentially huge. The most obvious benefits would typically include:

- Reuse of existing, granular and performant calculation logic – ability to consume approved calculations outside the actuarial community
- Speed of deployment, operation and change
- Transparency of calculations
- Ability to review & analyse or debug any live calculation request

## Further Information

For further information regarding Mo.net, the Mo.net Quotations Service, or to see a live demonstration of the Mo.net Quotations Service in action, please contact us.

Software Alliance Limited  
30 Stamford Street, London, SE1 9LQ  
Tel: +44 (0) 20 3964 2755  
[www.softwarealliance.net](http://www.softwarealliance.net)

Author: Guy Shepherd

© Copyright 2020 Software Alliance Limited. All rights reserved.

Mo.net is a registered trademark of Software Alliance Limited.

All brand names and product names used in this document are trade names, service marks, trademarks or registered trademarks of their respective owners. Microsoft, Visual Studio and Windows are registered trademarks of Microsoft Corporation in the United States and other countries.