Z-Risk Engine®
Unlocking Credit Cycles

FEBRUARY 2022

Automating a centralized IFRS9 model architecture using open-source Python to reduce BAU operating expenses by **up to 40%**

- *Our previous Insights article explained how banks could reduce IFRS9 model development and implementation costs by about 30% using a centralised credit model architecture for wholesale portfolios*
- *Centralised model architectures improve substantially on the two-stage IRB-IFRS9 approach common to banks today*
- *In this follow-up article we highlight the IFRS9 BAU operating expense reductions available from utilising an E2E open-source Python approach to IFRS9 implementation*
- *For an illustrative European bank with a £900 bil balance sheet these total expense reductions could reach roughly £22 million over 5 years:*

Reduced IFRS9 Credit Model Development and Implementation Expenses	*Roughly £13 million*
Reduced IFRS9 BAU Operating Expenses	*Roughly £9 million*
Total 5-Year IFRS9 Expenses Reductions	***About £22 million***

The centralised credit model architecture outlined here can support IFRS9, Stress Testing and Climate Risk analysis in a single E2E integrated systems platform.

[1] As we discuss in more detail below, these IFRS9 credit model expense reductions focus on the credit model and BAU operating expenses from reduced staffing and the use of open-source Python that does not require external vendor license fees. The budget required to migrate bank's current two-stage approach to the centralised approach is excluded but these costs would be a fraction of the illustrative expense savings of about £22 million over 5 years.

In this follow-up article we highlight the IFRS9 BAU operating expense reductions available from utilising an E2E open-source Python approach to IFRS9 implementation. Reduced IFRS9 operating expenses, combined with reduced development and implementation expenses therefore in the aggregate, **can save banks a substantial amount of money**. A centralized IFRS9 solution also provides a more accurate and dynamic credit model ecosystem to support **an uncertain risk environment**, that today is complicated by a **combination of covid, climate and regulatory compliance.**[2]

> Re-thinking IFRS9 implementation utilising a centralised credit model architecture substantially reduces IFRS9 costs – providing the impetus for implementing much better integrated credit risk systems

To estimate potential operating expense savings, like the first Insights article we undertake an illustrative benchmark exercise for a large, hypothetical European Bank. Like all banks in Europe, the EBA Repair Initiative currently requires all IRB models to be redeveloped. This effort therefore provides an interesting opportunity to **re-think both IFRS9 methodology and implementation,** providing enhanced risk management benefits to go with the substantial IFRS9 expense reductions. There is a clear evolution in bank's implementation of complex credit models, which is moving away from more expensive, external vendor solutions and toward open-source internally managed platforms like Python.

Here we:

- **Summarise key concerns with implementation of the two-stage IRB-IFRS9 approach,**
- **Review a centralized IFRS9 credit model approach implemented in an open-source, Python architecture,**
- **Provide an illustrative benchmark for 5-year BAU operating expense budgets for each approach, and,**
- **Summarise the overall IFRS9 budget expense reductions available across both development and operating budgets.**

[2] Beyond IFRS9 and stress testing, assessing climate risk and all its added complexity will require a whole new set of models and systems for banks to figure out how to implement while also managing their risk modelling and implementation costs. **To support bank's long-run portfolio climate risk assessments, we are currently designing a climate module for Z-Risk Engine and will be publishing a follow-on article, outlining our approach to climate risk modelling.**

A. Brief Overview: 'Tactical' Approaches to Implementing IFRS9:

Implementing IFRS9 under short compliance timeframes for wholesale portfolios is not easy for a large, global bank with complex credit risk systems designed to support multiple business, risk management and compliance initiatives. In the end, most bank's **wholesale IFRS9 (and stress test) credit model systems** can reasonably be described as **a 'hodgepodge' mixture of the following**:

- vendor-licensed proprietary software,
- analytic credit models, for commercial and corporate IRB models,[3]
- IFRS9 credit models for converting IRB models
- integration with bank strategic credit risk and exposure data warehouses,
- multi-stage MI data systems including managing data staging, processing and cleaning at various processing points,
- technical IT and systems BAU support and ongoing testing,
- management model overlays, and,
- risk and finance IFRS9 reporting layers.[4]

> Banks are faced with ever complex and changing compliance and risk management drivers, making adaptive, tactical changes is usually the only near-term systems option for satisfying compliance deadlines.

These types of credit model implementations require extensive operating budgets for external software, coupled with internal staff to, manage complex risk data MI, provide ongoing data clean-up across a multi-stage process, and periodically update credit model parameters. **'Tactical' credit risk systems generally lead to ongoing, system adaptations that preclude ever getting to true strategic solutions.** This usual hodgepodge credit system approach is in essence potentially more like the opposite of automated, efficient, flexible E2E processing for risk models for large wholesale credit portfolios.[5] But unfortunately for complex banks, tight compliance timelines can lead to implementing tactical solutions.

IFRS9 operating costs can be substantially reduced by implementing an efficient, centralised credit model system that applies an open-source, micro-services architecture. This approach can be more flexible, more dynamic and provide more accurate MI for complex risk decision-making. Nearly all banks utilise variations of periodic macro-economic scenarios like GDP and other macro variables to undertake their IFRS9 PIT credit model adjustments and project ECLs and provisions. A centralised, automated batch architecture can use detailed industry and region credit cycle factors derived from

[3] Our focus here is only on commercial and corporate portfolios as retail and small business are usually designed in different credit risk systems.

[4] For this discussion we focus on the key IFRS9 credit models and their implementation – as IFRS9 is accounting focused, we exclude discussion of the core reporting components of the E2E solutions bank's use, to simplify the narrative.

[5] As IFRS9 is a recent accounting and regulatory standard, our general assumption is that bank's have had to adapt by adding new credit models on top of their IRB models and therefore current IFRS9 systems implementations are adaptive rather than strategic re-builds. IFRS9 then in either the two-stage or centralised approach has added a 'PIT adjustment' layer between IRB models and ECL calculations.

For IFRS9 and Stress Testing, CROs and CFOs can achieve substantial expense savings for combined development and operating budgets – while also improving credit risk model accuracy and decision making

public-firm default models in the PIT adjustment layer. Utilising much more dynamic risk signals can support **re-running portfolio-wide ECL assessments daily, weekly or monthly instead of the usually less frequent quarterly approach.** This allows the PIT adjustments and the ECL logic to be a single E2E process.

To motivate IFRS9 and stress testing operating expense savings and the enhanced credit risk analysis that is possible, we calculate an illustrative benchmark for BAU operating expenses. This benchmark compares the two-stage IRB-IFRS9 model approach, to the holistic, centralised approach.[6] This benchmark assumes bank's current IFRS9 systems approach uses an evolving, Tactical architecture as described above that also utilises a substantial amount of external, proprietary software components.[7]

In comparison, we contrast bank's two-stage IFRS9 approach with a centralised credit model solution implemented in a single E2E, micro-services open-source Python architecture. We then present illustrative BAU operating expense benchmark budgets. This usual bank 'Tactical' IFRS9 systems implementations, developed over the last 2-3 years – is then compared to the centralised model architecture approach.

B. Benchmarking Bank's Operating Expenses for 'Tactical' IFRS9 System Implementations:

Like many evolving, complex regulatory compliance challenges faced by banks, IFRS9 combined with stress testing requires banks to integrate in some way, their IRB credit models, scenario analysis, and portfolio-wide risk projections. This is on 'top' of their legacy credit risk platforms. See, Venter and Pertsinis, Deloitte Risk Advisory, Deloitte Financial Services Blog (2018) and Protasiewicz (2020) for a discussion of complex credit risk measurement system challenges banks currently face and the use of open-source Python. Developing open-source tools for credit risk models, is helping reduce expenses, support fully internal system 'ownership', rationalise business value and manage implementation risk.

Our illustrative IFRS9 operating expenses benchmark exercise presented here, is focused first on how banks have generally implemented their IFRS9 ECL and provision systems under the two-stage model approach. Then secondly, we juxtapose, the current common approach with a holistic, centralised model architecture.

[6] Our focus generally in this Insights article like the previous one is primarily on IFRS9 models, architecture and systems costs – but the same logic of substantially reduced costs and more accurate ECL projections also applies to stress testing. We see IFRS9 and stress testing compliance and related cost reductions as going hand-in-hand as they both involve the same models, processes and systems – with both running various what-if scenarios.

[7] While our focus is more narrowly on IFRS9 and related stress testing, there is a general evolution in the banking industry that is moving away from less flexible, more expensive external software applications toward development of in-house, open-source applications for most risk model development and implementation.

IFRS9 is the latest compliance objective where banks must satisfy new regulatory requirements, most likely by adapting or augmenting their current 'As Is' risk systems IT architecture instead of developing a strategic solution

Bank credit risk systems usually have a fairly common set of standardised software components and business processes. For IFRS9 recently banks have faced an adaptive process, building IFRS9 models along with the more mature IRB credit model implementations.

More specifically, IFRS9 'Tactical' systems developed over the last 2-3 years to satisfy IFRS9 compliance most likely have had to take a **'hodgepodge' approach that may include the following more specific characteristics:**[8]

1. *Internally developed IRB PD/LGD/EAD credit models implemented in (i) proprietary, (ii) hybrid open-source or (iii) full open-source system platforms*[9]
2. *Externally licensed proprietary credit models validated for IRB use or used as secondary decision-support or benchmarking together with internal IRB credit models*
3. *IFRS9 internal or external consultant-developed second-stage models that adjust IRB models using macro-economic data to make them better reflect forward-looking PIT risk conditions*
4. *'Tactical' Integration with various strategic portfolio data and exposure systems*
5. *Multiple, partially fragmented standalone 'silos' for these models, processes, data and applications that are not fully integrated with the bank's strategic operating systems*
6. *Partial and tactical approaches to 'what if' analysis for either model parameters or undertaking scenario analysis*

In actuality, each bank will have taken their own specific approach to implementing IFRS9, so these comments remain general to support the high-level operating expense benchmark analysis presented here.

The **Figure** below provides a graphic to illustrate the general implementation approach common to bank's using two-stage IFRS9 model. For their wholesale portfolios, banks will have various IRB credit models usually implemented in proprietary software platforms that are linked to internal portfolio MI and exposure data warehouses. A second-stage model that adjusts IRB model outputs especially PD is usually organised in a second set of models. Broadly, these model architectures will **'knit' together different internal and external code components** in usually somewhat **less efficient ways** that can require intervention at various stages to manage accurate processing and data quality.

[8] We use the term 'Tactical' to describe data, model, processing and IT implementations that augment or adapt existing strategic platforms 'on the margin'. This usually occurs in strategic bank systems development life cycles when a complete 'rethink' of the overall architecture isn't possible – this is especially true when new requirements like IFRS9 have urgent deadlines and new types of models and their implementation are developed in parallel.

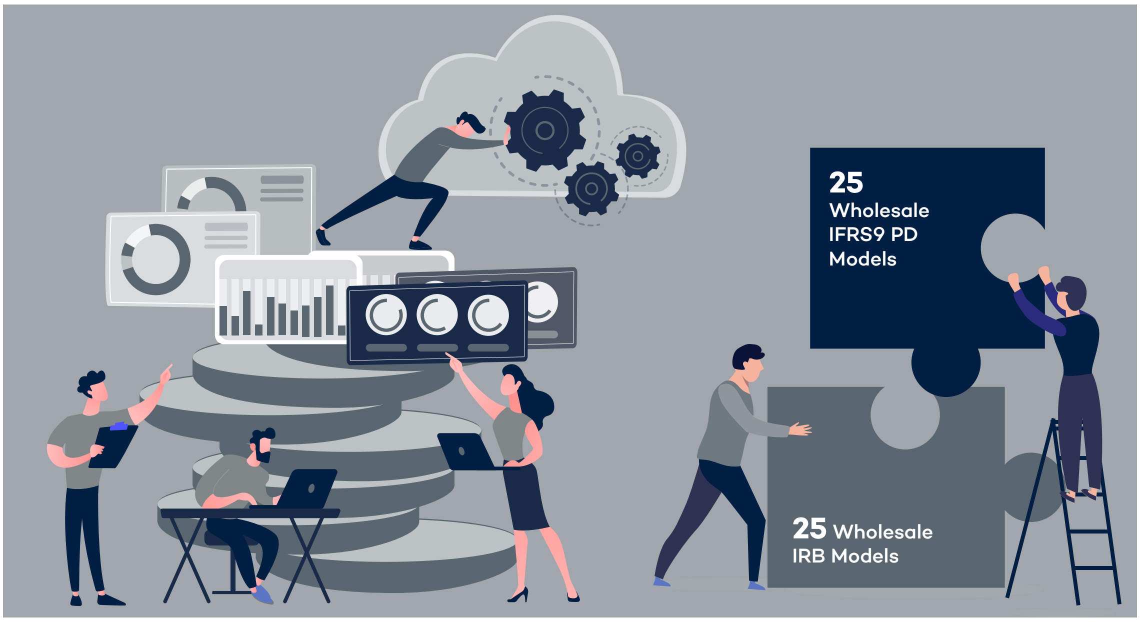
[9] See Deloitte 2018 for a more detailed outline of these three types of credit model system solution approaches.

The goal here is not to outline this IFRS9 architecture in detail but to highlight the various components in simple terms to be able to then illustrate for a benchmark hypothetical bank, what the broad operating expenses might look like.

In the previous article, the substantial expense savings stemmed from not needing to develop a full second stage IFRS9 model for each individual wholesale IRB PD model. Here the operating cost savings will derive from implementation of a streamlined, E2E architecture and the use of open-source tools, reducing external software license costs. Additional expense reductions accrue from using a coherent E2E single code platform that is easier to maintain and enhances processing efficiency, reduces staff costs and improves DQ.

> IFRS9 system implementations that are 'knitted together' with disparate applications and internal and external components - cost more money to develop and operate

IFRS9 HIGH LEVEL ARCHITECTURE DIAGRAM



C. An Alternative, Holistic, Centralised IFRS9 Architecture Developed Using Open-Source Python:

As an alternative we outline a centralised implementation approach for IFRS9 that uses a single, holistic PIT/TTC methodology that applies credit cycle adjustments to all of a bank's IRB model outputs in a single batch process. As described in our previous article, this approach utilises a single IFRS9 model for all wholesale PD models together instead of a separate second-stage model for each individual IRB PD model.

We will use the same generic system components outlined above to support our benchmark operating cost analysis for this alternative, centralised approach. Compared to the usual two-stage IFRS9 model approach, this alternative centralised architecture:

- Can be developed in E2E open-source, Python, and better integrated database management tools,
- Reduces reliance on proprietary external software vendor applications and licenses,
- Implements a single IFRS9 model to adjust all IRB PD models in a single process, and,
- Improves ECL projection accuracy, reduces staff costs related to data cleansing and staging, and supports the ability to run multiple and frequent batch processing.

To implement a centralised, micro-services IFRS9 architecture, Python modules are used to develop a set of integrated components that are shown in the Figure below. Instead of a second stage model to adjust IRB PD model outputs on an individual model basis, a separate central batch creates up-to-date credit cycle factors for each borrower's industry and region. These so called 'Z' credit cycle factors adjust primarily TTC PD model outputs to fully reflect PIT credit conditions.[10] These adjustments are undertaken in a single batch process E2E within Python.

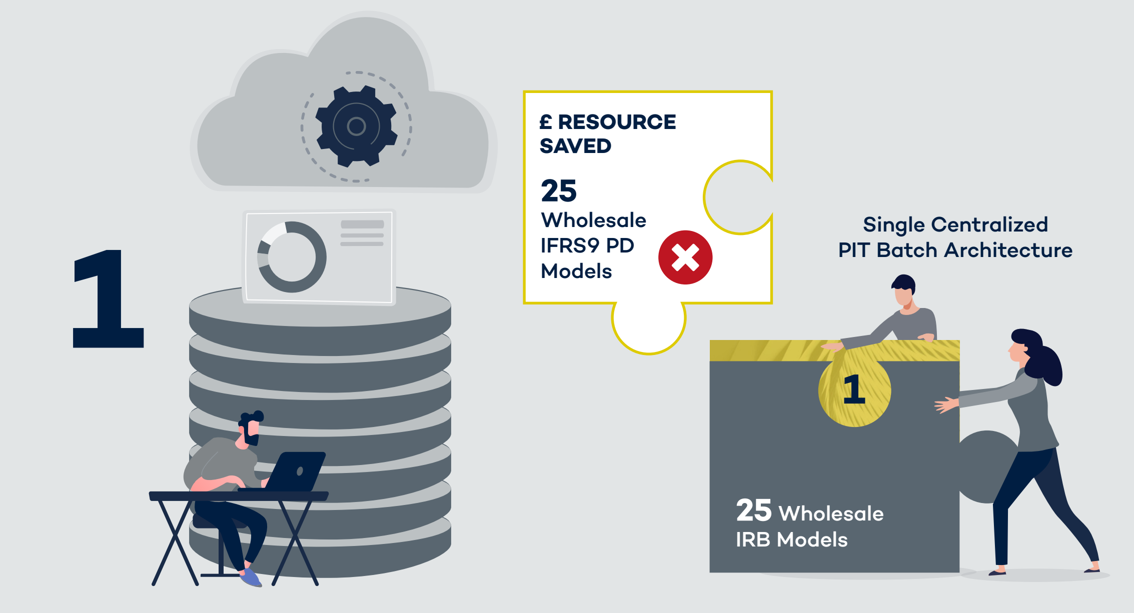
This Z-model can be run E2E on a fully automated basis and the factors also saved to be reused with alternative macro-economic scenarios to develop IFRS9 ECL estimates for each facility, borrower, or any portfolio segment using various what-if scenarios. The adjusted IFRS9 PIT PD, LGD and EAD estimates are then processed in a fully integrated ECL batch module across the full-term structure for all exposures and for all borrowers.

The fundamental difference with this approach compared to Bank's 'Tactical' models and architecture is as follows:

1. *There is a single set of custom Python modules that can be run fully automated on an E2E basis,*
2. ***The architecture provides a centralised code base that can be supported by a single team of BAs and coders,***
3. *The micro-services architecture also supports parallel processing, and flexible 'what-if' parameter updates and scenario analysis,*
4. *Open-source code solutions like Python in a fully internal implementation environment can be fully customised in contrast to less flexible external vendor licensed software*
5. *Developing and implementing bank-driven open-source systems saves costs both in licensing and training and ongoing maintenance when additional vendor modules are potentially required, and,*
6. *An automated, E2E process on one consistent code base can substantially reduce data quality (DQ) issues and therefore reduce the need for extra staff to be cleaning and massaging data before it can be analysed.*

[10] The Z credit cycle factors are usually derived from any of the vendor provided public-firm default models that are derived daily for predicted one-year 'EDFs' from roughly 37,000 publicly traded firms.

CENTRALISED PYTHON IFRS9 HIGH LEVEL ARCHITECTURE


1
£ RESOURCE SAVED
25 Wholesale IFRS9 PD Models
Single Centralized PIT Batch Architecture
1
25 Wholesale IRB Models

D. Reducing IFRS9 BAU Credit Model Operating Costs – Benchmarking Bank's Two-Stage Model Approach to a Centralised Python Architecture

We illustrate the potential BAU operating expense savings available to banks that utilise a centralised open-source IFRS9 model architecture. To support this we present 5-year (2021-25) cost estimates for an illustrative benchmark bank for each of the implementation architectures. **These expense savings are derived from the centralised implementation architecture that:**

1. **reduces external software license costs and,**
2. **reduces staff costs due to the automated nature of the batch processing.**

The benchmark expense savings outlined here are in addition to the expense savings outlined in the previous *Insights* article that focused on various modelling and BA function staff reductions deriving from not having to develop a second stage IFRS9 model for each IRB PD model.[11] This benchmark focuses on the BAU operating costs of running either IFRS9 approach. To be clear, for the purposes of this benchmark exercise, we have excluded the strategic system development costs of migrating bank's current two-stage IRB-IFRS9 approach to the centralised python-based approach for brevity. This allows us to combine the model development expense savings outlined in the previous article with this BAU operating expense savings illustrative benchmark to aggregate the full expense reduction benefits.

Like the previous benchmark discussion, we use here an illustrative hypothetical European bank with a global footprint, with a substantial wholesale portfolio and an assumed total balance sheet of £900 bil.

[11] See November 2021 article

Table 1: Benchmark Illustrative IFRS9 BAU Operating Costs – Two-Stage IRB-IFRS9 Model Approach

Annual Average Software License and BAU Operating Staffing Requirements

	2021	2022	2023	2024	2025	2021-25
External Software Costs* **	£1 MIL	£1 MIL	£1 MIL	£1 MIL	£1MIL	£5 MIL
Quant Modellers	4	4	4	4	4	20
BAU Support/ Testing BAS	4	4	4	4	4	20
System Operating Coders	5	5	5	5	5	25
Data Analysts (Reporting and DQ Cleanup)	8	8	8	8	8	40
Project Managers	3	3	3	3	3	15
Total	**24**	**24**	**24**	**24**	**24**	**120 FTE**
Avg Staff Cost - £90k/Year						£10.8 MIL
Total 5 Year Software And Support Operating Budgets Required						**£15.8 MIL**

*Each bank will be using a wide mix of various external vendor supplied software licenses to support their current IFRS9 implementation, so this annual software cost assumption is an 'educated guess' excluding inflation adjustments.

** Banks also usually contract for global vendor licenses to support multiple applications. The assumed license saving is focused just on reducing the use of the external vendor licenses for IFRS9 specifically.

Table 2: Benchmark Illustrative IFRS9 BAU Operating Costs – Centralised Open-Source IFRS9 Model Approach

Annual Average Software License and BAU Operating Staffing Requirements

	2021	2022	2023	2024	2025	2021-25
External Software Costs*	0	0	0	0	0	0
Quant Modellers	3	3	3	3	3	15
BAU Support/ Testing BAS	3	3	3	3	3	15
System Operating Coders	3	3	3	3	3	15
Data Analysts (Reporting and DQ Cleanup)	4	4	4	4	4	20
Project Managers	2	2	2	2	2	10
Total	**15**	**15**	**15**	**15**	**15**	**75 FTE**
Avg Staff Cost - £90k/Year						£6.75 MIL
Total 5 Year Software And Support Operating Budgets Required						**£6.75 MIL**

*The zero-cost external vendor software license assumption is narrowly focused on IFRS9 model development and implementation specifically and is undertaken in Python – this assumption excludes the use of external software license costs incurred related to IRB models directly or indirectly, or support for any other risk or regulatory objectives

Assumptions:

- Average annual staff costs assume:
  - They are fully loaded for benefits
  - Represent a mix of senior and junior, contract and permanent staff that are somewhat lower on average than the first benchmark presented in the earlier Insights article
  - A mixture of higher and lower cost geographic regions

- The bank's regulatory and modelling approach:
  - The bank is an IRB bank for regulatory capital purposes
  - As a European bank, all IRB models will be re-developed in 2020-2022
  - The illustrative benchmark bank has a total balance sheet of about £900 billion

As shown in this illustrative benchmark example, this hypothetical bank could potentially save roughly £9 mil in BAU IFRS9 operating expenses. The previous benchmark estimate of saved roughly £13 mil in model development and implementation costs over 5 years by moving away from the two-stage IFRS9 model approach.

Therefore, the combined expense budget savings would total about £22 million over a 5-year period for this illustrative bank – about £13 million for reduced development expenses and about £9 million for reduced operating and software expenses.

Reduced IFRS9 Credit Model Development and Implementation Expenses	Roughly £13 million
Reduced IFRS9 BAU Operating Expenses	Roughly £9 million
Total 5-Year IFRS9 Expenses Reductions	**About £22 million**

Z-Risk Engine – Providing the Full E2E Holistic Solution:

This follow-on expense benchmark assessment has added estimates of IFRS9 BAU operating expense reductions to the previous benchmark which focused on reducing model development, validation and implementation costs. Assessing the availability of these expense savings is critical for banks using the common two-stage IRB-IFRS9 model approach. While focusing on operating expense reductions overall, **we have also highlighted the general cost inefficiency and architecture concerns that result when new global accounting compliance standards are rolled out with aggressive deadlines in large, complex banking organisations.**

Developing completely new types of wholesale PIT credit models is not easy and these are the types of early cycle system implementations that contribute to a 'hodgepodge' systems architecture and ultimately substantially higher iFRS9 expenses.

However, the holistic, centralised IFRS9 solution described here is available today and has been fully implemented in a Python E2E credit systems platform.

Z-Risk Engine has business benefits that can:

- Achieve the substantial IFRS9 expense reductions outlined here,
- Bypass the costly internal development and migration required to move from the two-stage approach to the centralised architecture,
- Fully support IFRS9 compliance, having already been approved at multiple large wholesale banks

- **Support IFRS9, Stress Testing and in the near future, long run climate risk – all in a single platform,**
- Be fully integrated with a bank's own wholesale IRB credit models, and customized to each bank's specific portfolio segmentations, and,
- Be implemented E2E as a single solution either in Python source code managed by the Bank, or used as a Managed Service
- Utilise custom IFRS9 PIT models tailored to a bank's specific portfolio, industry and region

As a follow-up, we will publish next, an Insights article outlining our long-run portfolio approach to modelling climate risk for banks.

BIBLIOGRAPHY

Aguais, S. D., Forest, L. R. Jr., Wong, E. Y. L., and Diaz-Ledezma, D. (2004). Point-in time versus through the-cycle ratings. In The Basel Handbook: A Guide for Financial Practitioners, Ong, M. (ed). Risk Books, London.

Aguais, S. D., Forest, L. R. Jr., King, M., Lennon, M. C., and Lordkipanidze, B. (2007). Designing and implementing a Basel II compliant PIT–TTC ratings framework. In The Basel Handbook II:A Guide for Financial Practitioners, Ong, M.(ed).Risk Books, London.

Forest, L and Aguais, S (2019), "Inaccuracies Caused by Hybrid Credit Models and Remedies as Implemented by ZRE."

Forest, L and Aguais, S (2019), "Scenario Models Without Point-in-Time, Market-Value Drivers Understate Cyclical Variations in Wholesale/Commercial Credit Losses."

Deloitte Financial Services Blog, 'Open-Source or Proprietary Software for Credit Risk: It's all about asking the right questions', 25/04/2018

Protasiewicz, Jakub, 'How Python is Used in Finance and Fintech', March 5, 2020

Venter, E., and A Pertsinis, Deloitte Risk Advisory, 'Credit risk measurement technology trends – Charting the course from legacy issues to strategic solutions', undated

Developed by Aguais And Associates Ltd, Z-Risk Engine® (ZRE) provides a highly accurate, centralised, and integrated solution supporting global bank's compliance for IFRS9, CECL and Stress Testing regulations.

ZRE is a proven and efficient route to regulatory compliance for CROs and CFOs that also delivers a 30%-40% reduction in IFRS9 modelling operational costs. As an advanced suite of Python or SAS® based software that works with a bank's own IRB wholesale internal credit models, ZRE unlocks complex industry and regional credit cycles to accurately convert TTC PD, LGD and EAD models into PIT measures. Whilst lowering implementation risk, the solution is also highly configurable and customisable to support large bank's detailed portfolio mix of commercial, corporate and bank clients.