

Post Trade Solutions

Quant Summit London 2026

LSEG



Welcome Keynote

Emily Prince

Group Head of Enterprise AI, D&A, Analytics

LSEG

Post Trade Solutions

13:00



LSEG's Internal Adoption of ORE



Joey O'Brien

Principal Quant Consultant

LSEG



Richard Bareham

Pricing and Analytics Lead

LSEG



Mohamed Dergueche

Director, Head of SwapClear

Risk Change

LCH



Sanja Hukovic

Head of Model Risk

LSEG

Topics

- Across LSEG, different businesses recognise the significant modelling benefits and cost efficiencies enabled by the open-source ORE framework. SwapAgent and SwapClear share their experiences using ORE and how it has reduced reliance on third-party risk analytics vendors.
- First-hand perspectives on the ORE learning curve, available support resources, reconciliation challenges, and other key planning considerations.
- A discussion on the cost savings and additional benefits that could be achieved through the adoption of ORE.

Performance Optimisation and Run-time Considerations for ORE



Peter Caspers

Quant Development Lead

LSEG



Alexander Soreff

XVA Trader

DZ Bank AG



Lewis Wilkins

Senior Risk Engineer

Trafigura



Roland Stamm

Partner, Quantitative Services

LSEG

Topics

- ORE Performance: Multithreading, AMC, CG, AAD and GPU.
- Optimising run-time for intra-day pricing and market risk management.

Performance / Overview

Multithreading

Exposure, Sensis, Stress, HistSimVar, BT, ...

(MultiThreadedValuationEngine, **productive**)

AMC

Exposure Frameworks based on American MC

(AmcValuationEngine, MT training, vectorized, **productive**)

AMC-CG

Rewrite of AMC based on Computation Graph

(vectorized, experimental)

AAD

T0 Sensis Exotics

XVA Sensis

DynamicSIMM

(CG, **productive**)

(AMC-CG, experimental)

(AMC-CG, experimental)

GPU

T0 Analytics Exotics

XVA Sensis / Stress

(CG, experimental)

(AMC-CG, experimental)

Multithreading

Introduces `MultiThreadedValuationEngine` as a drop-in replacement for `ValuationEngine`

Applicable to all analytics that use `ValuationEngine`:

- MC Exposure Simulation
- T0 Sensis, Stress, HistSimVar, BT, ...

Parallelization in trades:

- use T0 pricing timings to distribute trades in `nThreads` buckets
- every bucket has ~ same total pricing time

AMC and AMC-CG

Use **regression models** instead of full revaluation for fast NPV cube population.

Only the **model state** has to be evolved, no need to construct full conditional market curves (as in MC).

Fast for **Exotics**, since conditional expectations via regression are also available to the pricing engine.

AMC-CG: Rewrite of AMC that separates **computation graph** building and calculation

- enables AAD where applicable (dynamic SIMM, xva sensitivities)
- enables GPU

Both AMC and AMC-CG use vectorization.

GPU

All CG – based calculations can be offloaded to GPU:

- CG is transformed to GPU **Kernel source code** during runtime (OpenCL or CUDA)
- Kernel is compiled
- Input Data is copied CPU -> GPU
- Kernel is executed
- Result Data is copied GPU -> CPU

Calculation Model assumes that one calculation is **repeated** several times on **modified input data**:

- Kernel has to be built only once
- Applies to T0 Sensis, Stress, etc. as well as XVA Sensi / Stress scenario calculation
- Allows to combine AAD for Dynamic SIMM with XVA Sensi via Bump (avoid 2nd order derivatives)

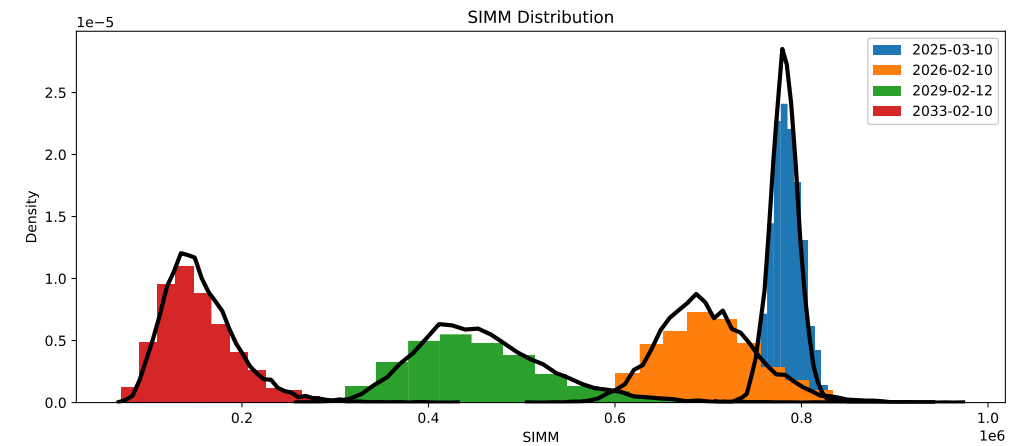
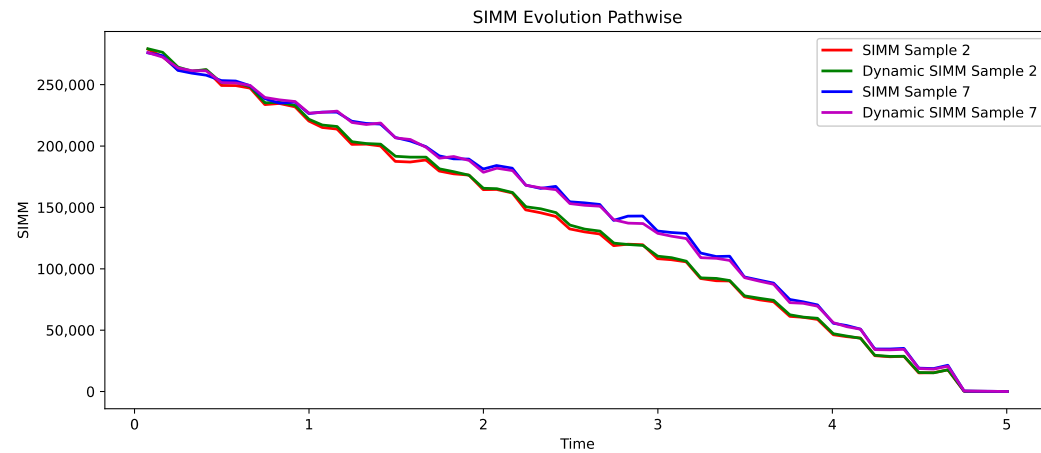
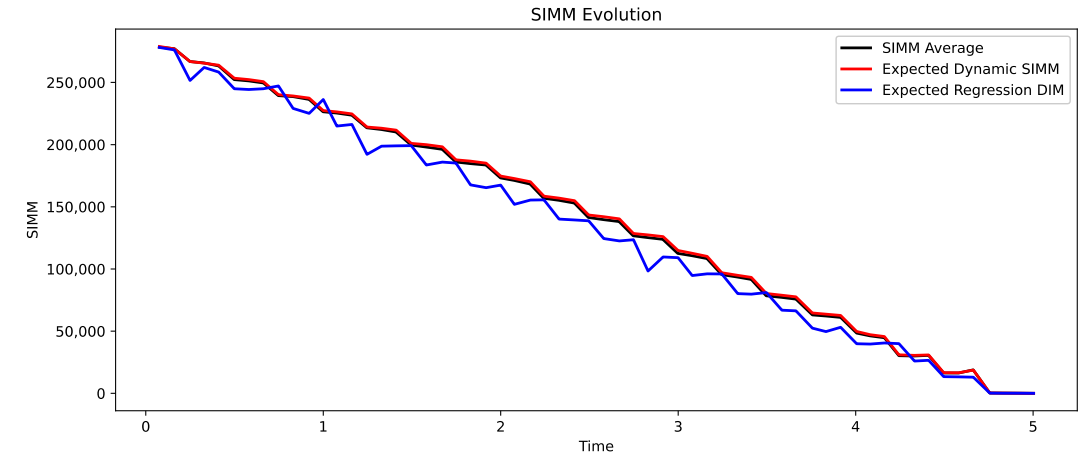
AAD – Bermudan Swaption EUR/SEK SIMM Calc

Bermudan Swaption EUR / SEK, 170 sensitivities								
Side	RiskClass	MarginType	bump1k	bump10k	bump100k	bump1000k	aad10k	
Call	InterestRate	Delta	4,599,058	308,487	1,156,085	207,764	146,169	
		Vega	935,464	133,953	136,932	7,405	1,596	
		Curvature	-	25,363	70,162	150	-	
	FX	Delta	1,072,303	624,096	456,021	495,122	483,673	
		Vega	341,280	226,573	251,024	248,635	248,359	
		Curvature	78,564	63,430	65,532	64,867	63,611	
	All	All	6,095,225	1,129,946	1,732,852	889,255	846,178	
	Post	InterestRate	Delta	4,599,058	308,487	1,156,085	207,764	146,169
			Vega	935,464	133,953	136,932	7,405	1,596
Curvature			406,377	33,385	-	4,134	831	
FX		Delta	1,072,303	624,096	456,021	495,122	483,673	
		Vega	341,280	226,573	251,024	248,635	248,359	
		Curvature	-	-	-	-	-	
All		All	6,454,412	1,077,277	1,627,022	828,302	783,910	
runtime / s				27	257	3,763	49,019	1.5

AAD - Dynamic SIMM

Successor of the simple regression approach in ORE

- ISDA SIMM along MC paths for PFE and MVA
- IR/FX and SIMM 2.7 so far
- **Fast Sensitivities** and CRIF via **AAD**
- Validation against “brute force” ISDA SIMM, bump & reval CRIF
- Performance: 10k IM paths for the price of 1 benchmark path
- See Examples → InitialMargin → Dim2



Appendix: Details on Exposure Framework Performance

Exposure Frameworks

1 MC:

- Generic Framework using T0-Pricers
- Generates full market data landscape, amortizes only for large # trades
- **No vectorization, no GPU**

2 AMC:

- Uses Regression Models
- Generates model state only
- Dedicated pricers to generate flows and exercise decisions
- **Partial vectorization (path generation no, training / pricing yes)**
- **GPU for Regression Model training (single calls are routed to GPU)**

3 AMC-CG:

- **Full vectorization**
- Separation of computation graph building and calculation
- **AAD enabled, GPU doing calculation**

AMC Performance Python NumPy vs. ORE

- Bermudan Swaption 4 fixed, 8 float, 3 exercise
- Timings in 1E-6 seconds:

method	pricing	path generation	other
py_lsmc	40,778	2,000	
py_backward	1,415	na	
ore lsmc	2,340	1,920	10
ore backward	76	na	

- Path generation is comparable for ORE vs. NumPy
- LSMC is 10x faster in ORE
- Backward convolution is 20x faster in ORE

ORE MC Performance Improvements QPR-13958

- Caches for Scenario and ASD generation
- Avoid map lookups for risk factor keys
- Introduce `FxForwardOptimized`, `SwapOptimized` as PoC for faster Instruments / Pricers

```

FxForwardOptimized::FxForwardOptimized(const Real& nominal1, const Currency& currency1, const Real& nominal2, const Currency& currency2,
                                       const Date& maturityDate, const bool& payCurrency1, const bool isPhysicallySettled,
                                       const Date& payDate, const Currency& payCcy, const Date& fixingDate,
                                       const QuantLib::ext::shared_ptr<QuantExt::FxIndex>& fxIndex,
                                       const Handle<YieldTermStructure>& currency1DiscountCurve,
                                       const Handle<YieldTermStructure>& currency2DiscountCurve, const Handle<Quote>& spotFX)
: nominal1_(nominal1), currency1_(currency1), nominal2_(nominal2), currency2_(currency2),
  maturityDate_(maturityDate), payCurrency1_(payCurrency1), isPhysicallySettled_(isPhysicallySettled),
  payDate_(payDate), payCcy_(payCcy), fxIndex_(fxIndex), fixingDate_(fixingDate),
  currency1DiscountCurve_(currency1DiscountCurve), currency2DiscountCurve_(currency2DiscountCurve),
  spotFX_(spotFX) {

    if (payDate_ == Date())
        payDate_ = maturityDate_;

    if (fixingDate_ == Date())
        fixingDate_ = maturityDate_;

    if (!isPhysicallySettled && payDate_ > fixingDate_) {
        QL_REQUIRE(fxIndex_, "FxForwardOptimized: no FX index given for non-deliverable forward.");
        QL_REQUIRE(fixingDate_ != Date(), "FxForwardOptimized: no FX fixing date given for non-deliverable forward.");
        registerWith(fxIndex_);
    }

    registerWith(currency1DiscountCurve_);
    registerWith(currency2DiscountCurve_);
    registerWith(spotFX_);
}

bool FxForwardOptimized::isExpired() const {
    ext::optional<bool> includeToday = Settings::instance().includeTodaysCashFlows();
    Date refDate = Settings::instance().evaluationDate();
    return detail::simple_event(payDate_).hasOccurred(refDate, includeToday);
}

void FxForwardOptimized::performCalculations() const {
    NPV_ = -nominal1_ * currency1DiscountCurve_>discount(payDate_) +
          nominal2_ * currency2DiscountCurve_>discount(payDate_) * spotFX_>value();
}

```

ORE MC Performance Improvements QPR-13958

- 1000 FX Forwards
- 5 currencies, 1000 paths, 81 timesteps

				Before:		Speedup
Loop	:	1.17E+07	mus	9.34E+07	mus	7.99
UpdateDate	:	2081	mus	4.90E+06	mus	2.73
UpdateScenario	:	766645	mus			
ScenGen	:	613264	mus			
ApplyScen	:	153381	mus			
Refresh	:	30707	mus			
Asd	:	227739	mus			
Pricing	:	1.07E+07	mus	8.80E+07	mus	8.26
Fixing	:	1282	mus	200000	mus	
Calibration	:	1912	mus	2000	mus	
Residual	:	16342	mus			
Trade Pricing / sec	:	3.85E+06		4.66E+05		

ORE MC Performance Improvements QPR-13958

- 100 Vanilla Swaps
- 5 currencies, 1000 paths, 81 timesteps

				Before:		Speedup
Loop	:	2.17E+07	mus	8.60E+07	mus	3.96
UpdateDate	:	2765	mus	7.20E+06	mus	4.23
UpdateScenario	:	721716	mus			
ScenGen	:	569823	mus			
ApplyScen	:	151893	mus			
Refresh	:	27213	mus			
Asd	:	228832	mus			
Pricing	:	1.96E+07	mus	7.90E+07	mus	4.03
Fixing	:	1.10E+06	mus	1.00E+06	mus	
Calibration	:	2054	mus	2000	mus	
Residual	:	12022	mus			
Trade Pricing / sec	:	2.09E+06		5.19E+05		

The missing bit in the MC Framework: Vectorization

- Example: Calculation of LGM Discount Factors
- **Vectorized version** takes array x of n states and returns array of n discount factors:

```

RandomVariable LgmVectorised::discountBond(const Time t, const Time T, const RandomVariable& x,
                                           const Handle<YieldTermStructure>& discountCurve) const {
    if (QuantLib::close_enough(t, T))
        return RandomVariable(x.size(), 1.0);
    QL_REQUIRE(T >= t && t >= 0.0, "T(" << T << ") >= t(" << t << ") >= 0 required in LGMVectorised::discountBond");
    Real Ht = p_->H(t);
    Real HT = p_->H(T);
    return (discountCurve.empty() ? p_->termStructure()->discount(T) / p_->termStructure()->discount(t)
        : discountCurve->discount(T) / discountCurve->discount(t)) *
        exp(-(HT - Ht) * x - 0.5 * p_->zeta(t) * (HT * HT - Ht * Ht));
}

```

- Quantities that are path-independent are only **calculated once** (Ht, HT, zeta, T0 discount factors)

The missing bit in the MC Framework: Vectorization

- Non-Vectorized version takes scalar x as state and returns a single discount factor:

```

inline Real LinearGaussMarkovModel::discountBond(const Time t, const Time T, const Real x,
                                                const Handle<YieldTermStructure> discountCurve) const {
    // we use a cache (if enabled) specifically in this method for usage in CrossAssetModelScenarioGenerator
    if (QuantLib::close_enough(t, T))
        return 1.0;
    if (discountBondCacheReady_) {
        auto tmp = discountBondCache_[discountBondCacheCounter_].first *
                  std::pow(discountBondCache_[discountBondCacheCounter_].second, x);
        if (++discountBondCacheCounter_ >= discountBondCacheSize_)
            discountBondCacheCounter_ = 0;
        return tmp;
    }
    QL_REQUIRE(T >= t && t >= 0.0, "T(" << T << ") >= t(" << t << ") >= 0 required in LGM::discountBond");
    Real Ht = parametrization_>H(t);
    Real HT = parametrization_>H(T);
    Real c1 = (discountCurve.empty()
              ? parametrization_>termStructure()->discount(T) / parametrization_>termStructure()->discount(t)
              : discountCurve->discount(T) / discountCurve->discount(t)) *
              std::exp(-0.5 * (HT * HT - Ht * Ht) * parametrization_>zeta(t));
    Real c2 = std::exp(-(HT - Ht));
    if (enableCache_) {
        discountBondCache_[discountBondCacheCounter_] = std::make_pair(c1, c2);
        if (++discountBondCacheCounter_ >= discountBondCacheSize_) {
            discountBondCacheCounter_ = 0;
            discountBondCacheReady_ = true;
        }
    }
    return c1 * std::pow(c2, x);
}

```

- Path-independent Quantities are either **recalculated on each path** or have to be **cached manually**

The missing bit in the MC Framework: Vectorization

- C++ Compilers have **auto-vectorizers** using SIMD instructions for loops, operating on **up to 8 doubles simultaneously** in the recent avx512 standard on x86-64

ADDSD

Add Scalar Double-Precision Floating-Point Values

Opcode	Mnemonic	Description
F2 0F 58 /r	ADDSD xmm1, xmm2/m64	Add the low double-precision floating-point value from xmm2/m64 to xmm1.

Description
Adds the low double-precision floating-point values from the source operand (second operand) and the destination operand (first operand), and stores the double-precision floating-point result in the destination operand. The source operand can be an XMM register or a 64-bit memory location. The destination operand is an XMM register. The high quadword of the destination operand remains unchanged.

Operation
<pre>Destination[0..63] = Destination[0..63] + Source[0..63]; //Destination[64..127] remains unchanged</pre>

ADDPD

Add Packed Double-Precision Floating-Point Values

Opcode	Mnemonic	Description
66 0F 58 /r	ADDPD xmm1, xmm2/m128	Add packed double-precision floating-point values from xmm2/m128 to xmm1.

Description
Performs an SIMD add of the two packed double-precision floating-point values from the source operand (second operand) and the destination operand (first operand), and stores the packed double-precision floating-point results in the destination operand. The source operand can be an XMM register or a 128-bit memory location. The destination operand is an XMM register.

Operation
<pre>Destination[0..63] = Destination[0..63] + Source[0..63]; Destination[64..127] = Destination[64..127] + Source[64..127];</pre>

Status GPU Exposure Frameworks

- MC : requires **major rewrite**, not planned
- AMC: route **single regression training** to GPU:

```
    } else if (regressionMethod == RandomVariableRegressionMethod::QR) {  
#ifdef ORE_ENABLE_CUDA  
    res = gpuQrSolveMultiStream(A, b);  
#else  
    res = qrSolve(A, b);  
#endif
```

There are general several calls to qrSolve() per trade and timestep (!) though, so that the speedup is diluted by the **data copying CPU -> GPU and GPU -> CPU** for each call.

- AMC-CG: encode the **whole calculation**, i.e.,
 - from calibrated curves and models to NPV Cube entries or
 - from calibrated curves and models to XVA numbers

as a computation graph and generate a GPU kernel from the cg at runtime.

Break

14:15 – 14:30

LSEG Models-as-a-Service (MaaS) and Model Context Protocol (MCP): Spotlights on Risk Analytics Lab and Anthropic Partnership

Post Trade Solutions

14:30



Aysegul Erdem

Head of Modelling Solutions,
D&A - Analytics
LSEG



Samuel Schwalm

Director, Enterprise Pricing
Analytics Proposition
LSEG



Jake Ullman

Senior Business Development Associate
LSEG

Contents

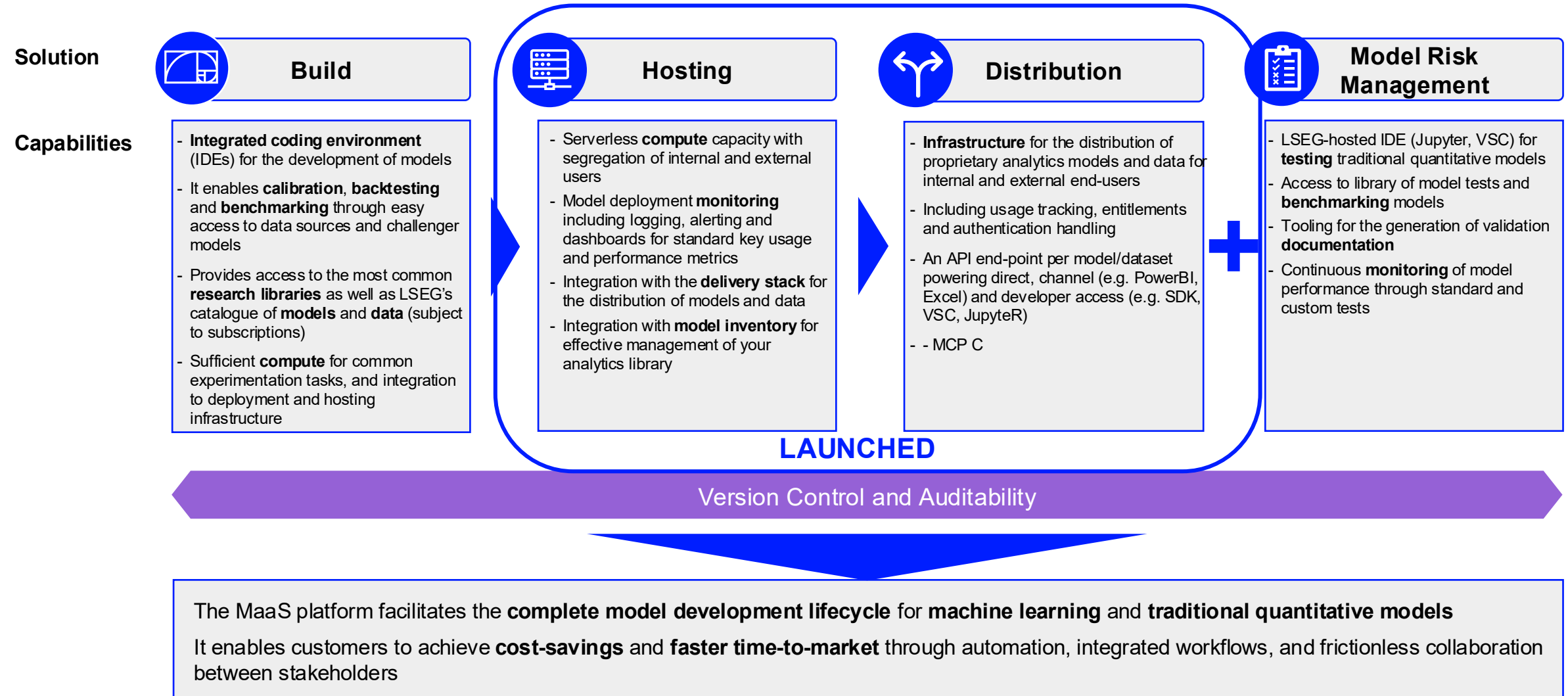
- What is Models-as-a-Services (MaaS)?
- Risk Analytics Lab
- Open Risk Analytics

What is Models-as-a-Service (MaaS)?

Aysegul Erdem (Head of Modelling Solutions)

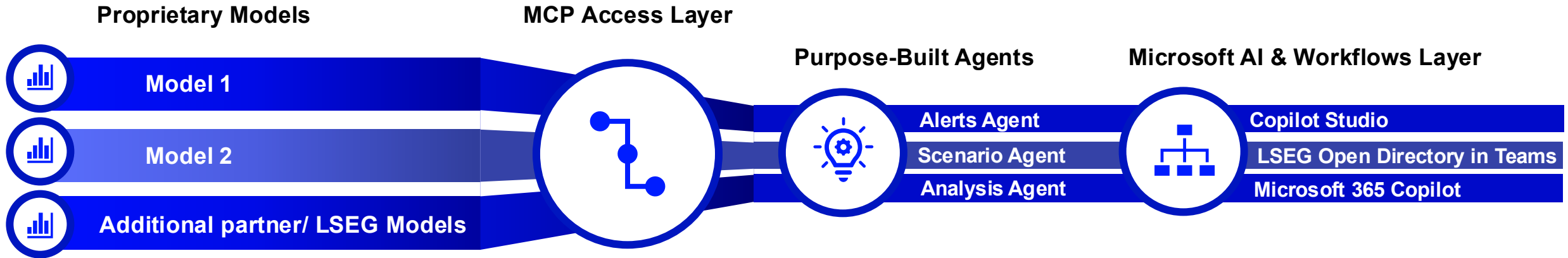
Overview of Modelling Solutions Vision

With the Modelling-as-a-Service ecosystem, LSEG will standardize the process of building, testing, deploying and storing financial and AI models faster, more efficiently and critically, in a risk-managed way.



How Modelling Solutions Power AI Workflows

Connect proprietary models to AI agents and deliver insights directly into enterprise workflows



Model-as-a-Service provides a secure and governed way to host, distribute and analyse proprietary models. Using MCP, these models can be connected to AI tools and surfaced through Microsoft environments such as Copilot Studio and Teams, making insights easier to access, share and act on within day-to-day workflows.

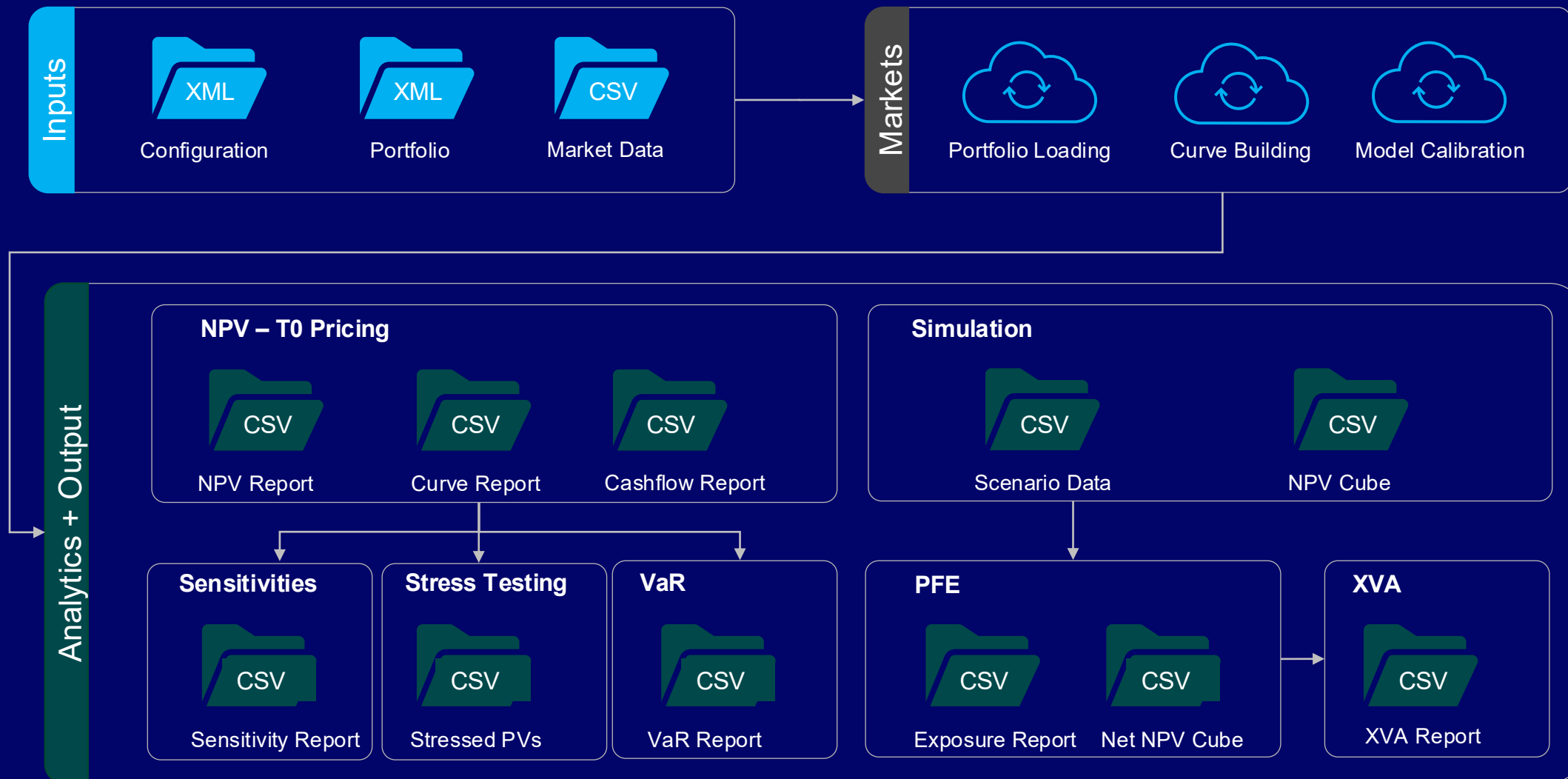
- **Secure model distribution at scale** through a governed marketplace.
- **Agent-based access to models and analytics** through Microsoft AI delivery surfaces.
- **Cross-firm collaboration** in Teams with Open Directory for discovery and communication.

Risk Analytics Lab

Jake Ullman (Senior Business Development Associate)

LSEG

ORE Workflow



What is Risk Analytics Lab?



Open-Source Risk Engine

- Open-Source software
- Unlimited flexibility
- Client led projects



Risk Analytics Lab

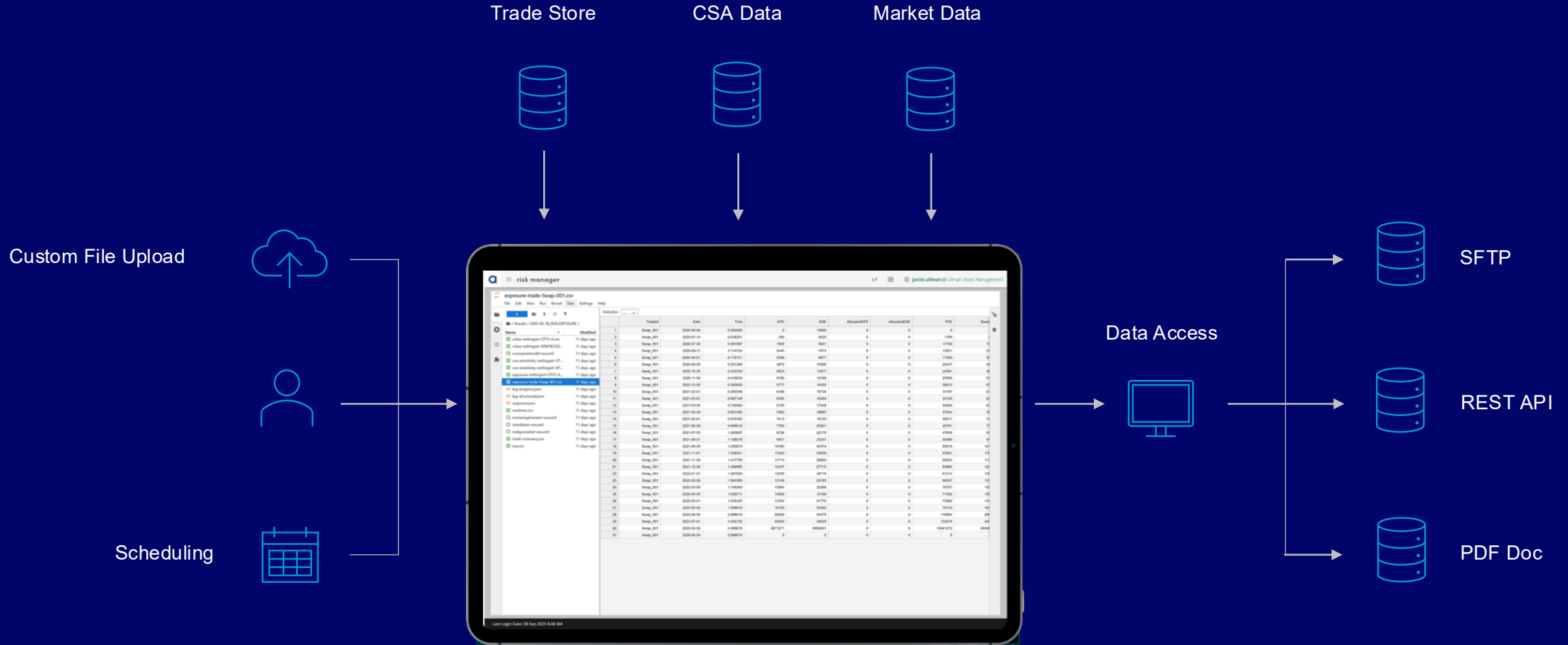
- SaaS delivered solution
- Integrated market data
- Flexible scripting & workflow



AcadiaPlus

- SaaS delivered
- Dedicated workflows
- Outcome based solutions
- 24x5½ support

Functional Overview



User Interaction

Extend the use of existing SFTP connections to support the secure upload of:

- Daily trade data
- Supplemental market data
- Other custom data types

User Specific Scripting Environment:

- High Quality Base Examples
- Fully Customisable Environment
- Access Powerful Acadia APIs

Schedule events based on the Acadia Event Bus

- Daily Trade Upload Complete
- IMRG Calculations Complete
- Timed Daily Schedule

Custom File Upload



Scheduling

Data Interaction

Access Agreement Data Store

- Agreement data for simulation of margin flows
- Unique 2-sided data model
- Inherit data from your counterparty

Access trade data from Initial Margin Risk Generator (IMRG)

- Access existing integrated trade data
- Make Additional use of adaptor investment
- Reactive Job execution based on trade upload

Market Data

- Access to Post Trade Solutions (formerly Acadia) exceptional market data solution
- Access Live Data from a range of snaps
- Access Cleansed Historical data from 2008 – Present day

Trade Store



CSA Data



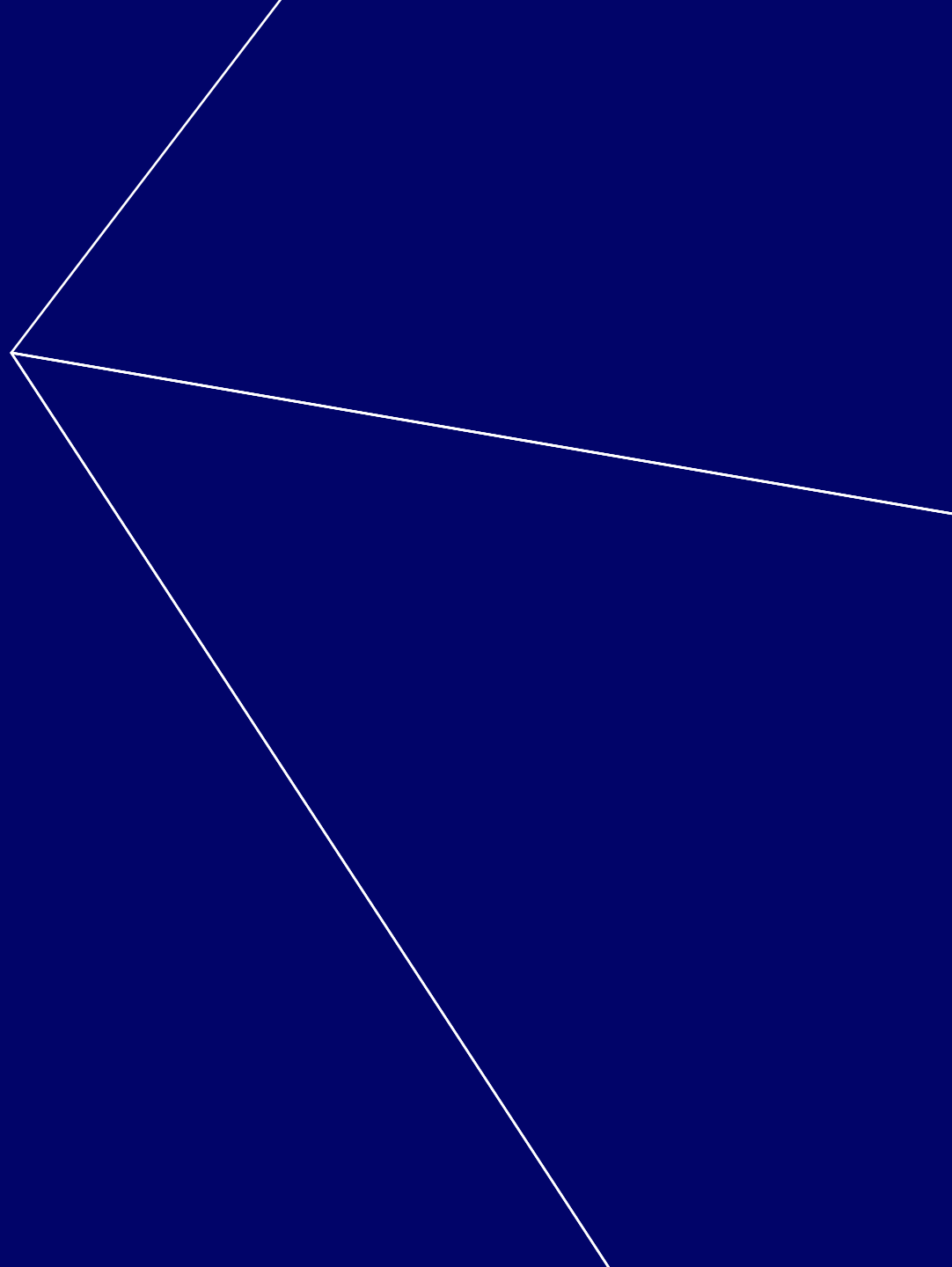
Market Data



Demo

Let's switch to the interface live!

LSEG



Open Risk Analytics

Jake Ullman (Senior Business Development Associate)

Samuel Schwalm (Director, Enterprise Pricing Analytics Proposition)

What is Open Risk Analytics?

Open Risk Analytics delivers powerful risk analytics – empowering teams to focus on risk insights, not infrastructure.

Open Risk Analytics is a SaaS-hosted risk analysis API designed to bring the power of the Open-Source Risk Engine (ORE) to a broader audience, without the complexity of in-house implementation.

Available through LSEG's Models-as-a-Service, customers benefit from a unified marketplace for quantitative datasets and analytics. This offering enables customers to access models from both LSEG and multiple third-party partners (model providers) securely and at scale without additional integration effort.

Key Features:

Broad set of key risk models

Covering a key set of standard risk calculations such as Historical VaR & PnL Decomposition

Prepopulated datasets

Out of the box risk model, with pre-configured market data, correlations and calibrations

SaaS delivery

Fast, secure, and scalable access backed by the strength and stability of LSEG

API integration

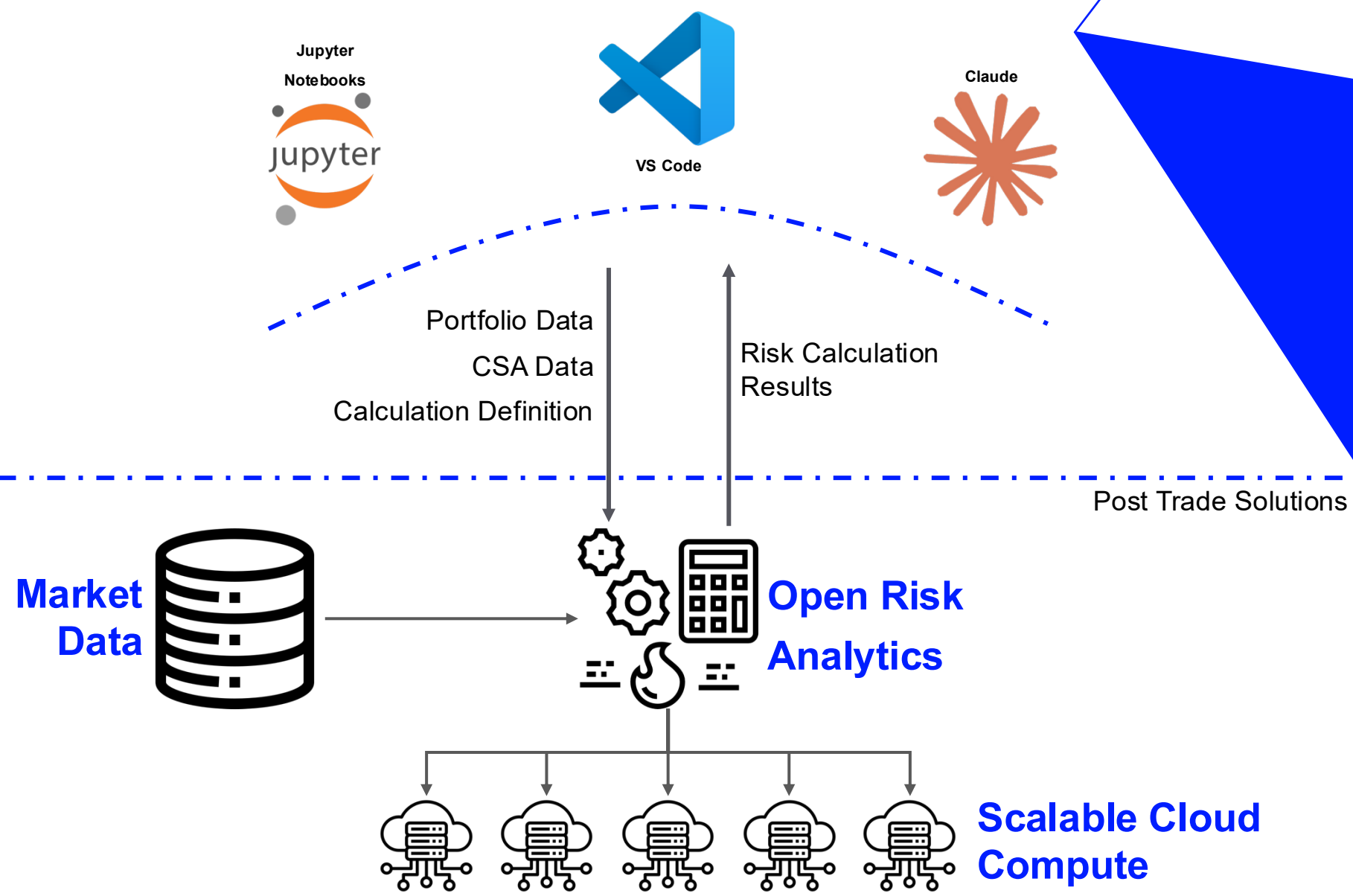
Fast and convenient API integration available through LFA, enabling tools such as visual studio code, and jupyter notebooks

Model Context Protocol Enabled

Accessible through the models context protocol allowing AI engines to call the APIs, building risk analytics and analysis dashboards on the fly



Open Risk Analytics – Diagram



☰ 🔍 ⬅ ➡

☰ - ☐ ✕

Chat `</> Code`

- + New chat
- 📁 Projects
- 📦 Artifacts
- 🏢 Ask your org
- 🛠 Customize

Pinned

Recents

☀ Coffee and Claude time?

How can I help you today?

+ Sonnet 4.6 🗣

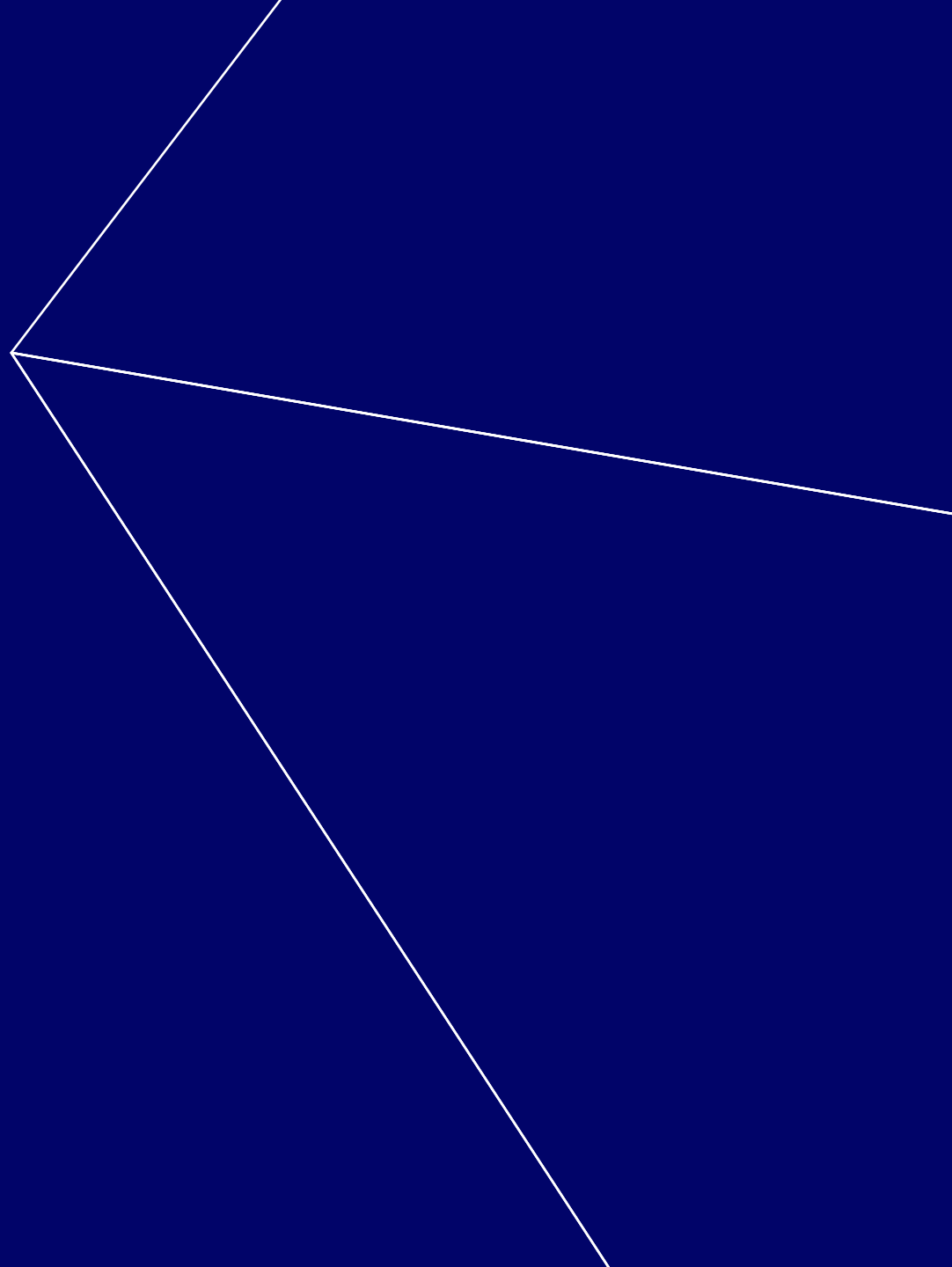
[Write](#) [Learn](#) [Code](#) [Career chat](#) [Claude's choice](#)

5 Samuel · LSEG

Demo

Let's switch to a live preview!

LSEG



Future Vision

How can we evolve the solution?

Open Risk Analytics intends to bring substantial new functionality to the marketplace, the approach to that will depend on the sales feedback from the Core Analytics package.

Integrated VaR Calculation

Leveraging the best-in-class analytics from Yield Book to deliver a comprehensive VaR calculation, covering listed instruments, structured credit and bilateral derivatives

Expansion of xVA Analytics

Bringing to market the full breadth of xVA calculations, including CVA, DVA, MVA, FVA delivered with high performance American Monte Carlo Techniques

Basel III (Endgame) compliance package

Now that Basel III has been announced globally many firms will look to adopt the standardised approaches, we can provide a compliance pack for; SA-CCR, FRTB-SA and BA-CVA

Integration with other data partners

Integrating the data outputs of the Open Risk Analytics models to existing data partners such as Data Bricks, can unlock new capabilities such as real time VaR and intraday limit checking

Improved support for listed instruments

While Open Risk Analytics is focused on bi-lateral derivatives today, it has the capability to perform calculations across exchange traded derivatives and listed asset as well, expanding this coverage will provide a more complete risk solution

ORE in the Era of Agentic AI

Post Trade Solutions

15:15



Xabier Anduaga

Partner, Quantitative Services
LSEG



Joey O'Brien

Principal Quant Consultant
LSEG



Gordon Lee

Senior Director,
Head of Markets Quants,
BNY

Topics

- What are LLMs genuinely capable of doing today? Where do they fall short? Can LLMs write pricing/risk models?
- Real examples where new AI agents are creating value.
- How ORE benefits from this new technology?
- How's the quant role evolving with AI?
- How might a risk system look like in a few years time?

Benchmarking Counterparty Credit Risk Capital Models: Recent work with ISDA and the PRA



Xabier Anduaga

Partner, Quantitative Services

LSEG



Paola Rensi

Head of Capital Models
Benchmarking

ISDA

Post Trade Solutions

16:00



Noemi Nava Morales

Director – Quantitative Risk Analytics

Bank of America

Topics

- Feature the latest ISDA white paper and its key takeaways.
- Address practical considerations for backtesting, model validation, and securing regulatory approvals.

Scan to read the full publication



New Features in ORE release v15/16 and Roadmap

Post Trade Solutions

16:25



Peter Caspers

Quant Development Lead

LSEG



James Mac Hale

Head of Risk Operations

LSEG

Topics

- Overview of Updates in v15 / v16
- Future Roadmap

Updates v15

Pricing & Simulation

- American Monte Carlo (AMC) model support for GenericBarrierOptions, FXEuropeanBarrierOptions, FXDoubleBarrierOptions, FXBarrierOptions, and FXKIKOBarrierOptions
- Flat-forward interpolation for commodity price curves
- Hull White n-Factor model calibration to historical rate curve moves
- Exposure cube NPV overlay

Instruments & Pricing Engines

- Support subtrades for composite trades
- Volatility smile outputs in additional results for variance swap replication engines
- Rate curve cashflow report containing the cashflows of rate curve instruments
- Constituent calibration to index level for index CDS options
- Configurable usage of indexed and par IBOR coupons respectively for curve building and trade pricing
- Support for conventional spreads in credit curve builder
- Start delay for inflation rate helpers
- New pillar choices (StartDate, StartAndEndDate, None) in rate curve configuration
- Penalty for curve smoothness in global rate curve builder

Analytics

- Sticky SABR sensitivity calculations

Misc

- QuantLib v1.41, released January 2026

Updates v16

New Instruments & Features

- Swaption Straddles as a single trade
- Cross-currency Flexi-Swap and Cross-currency Bermudan Swaptions
- New range accrual variation
- Rate digital option
- Dual European Binary Option
- Cross-Currency support for CommoditySwap

XVA

- LocalVol support for FX in exposure simulations
- Speed up of MC exposure simulation, scenario generation and fast fx forward, vanilla swap engines
- Review of inflation simulation framework
- Speed up of cube reading / writing

Market Risk

- Support multithreading in HistSimVar analytic
- Optional trade and risk factor breakdown in HistSimVar analytic
- Theta in sensitivity calculation

Pricing

- Heston model for scripted trade pricing
- SVI for equity, fx, commodity, credit vol surfaces
- Convexity adjustments for EquityTRS and BondTRS
- Optionally use effective volatilities in market cap surfaces
- Include conditional fee paid on exercise for swaption pricing

Updates v16

Market Data

- Support for rebasing of inflation indices
- Update single name CDS options to current conventions
- Quoted commodity calendar spread options
- Wildcards for zero curve spreads

Misc

- QuantLib and version 16 will use the recent QuantLib version v1.42
- Various bug fixes

Global Curve Building for rate curves and caps

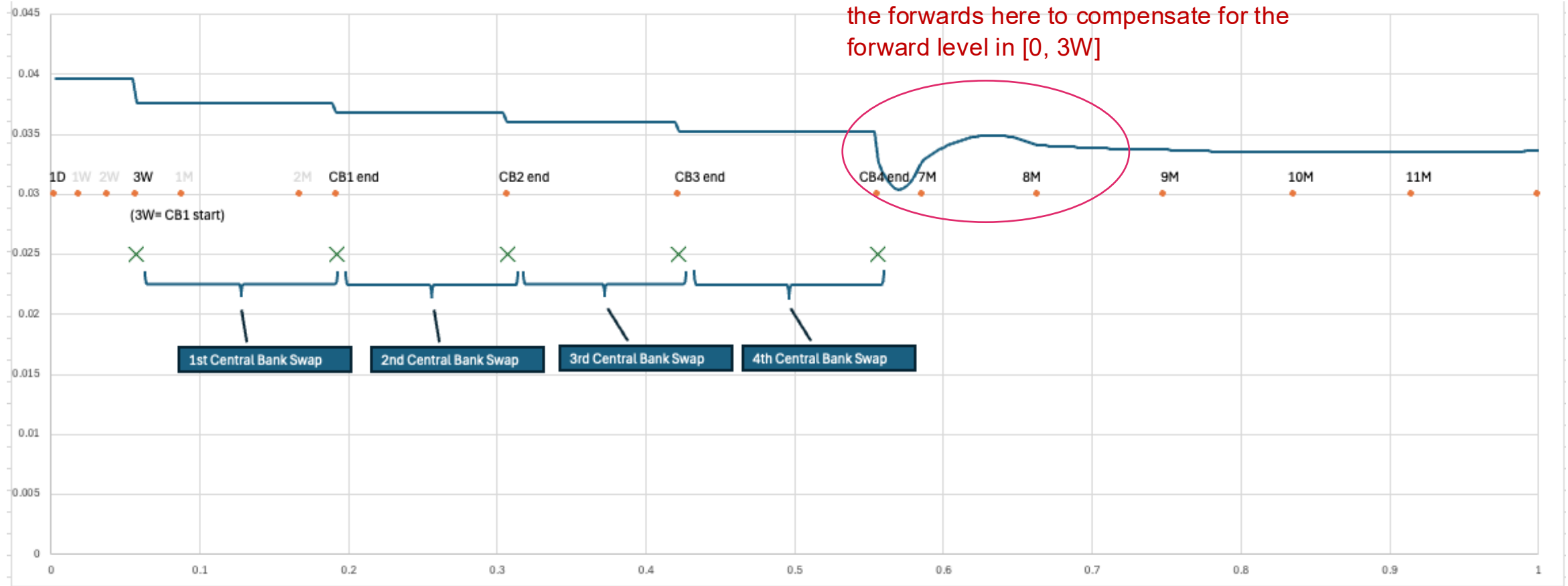
Iterative Bootstrap

- Limited to local interpolations + “last relevant date” interpolation nodes
- Workaround for other cases: Convergence Loop. Does not work in all cases!
- Replace Iterative Zero Search with Global Optimization.

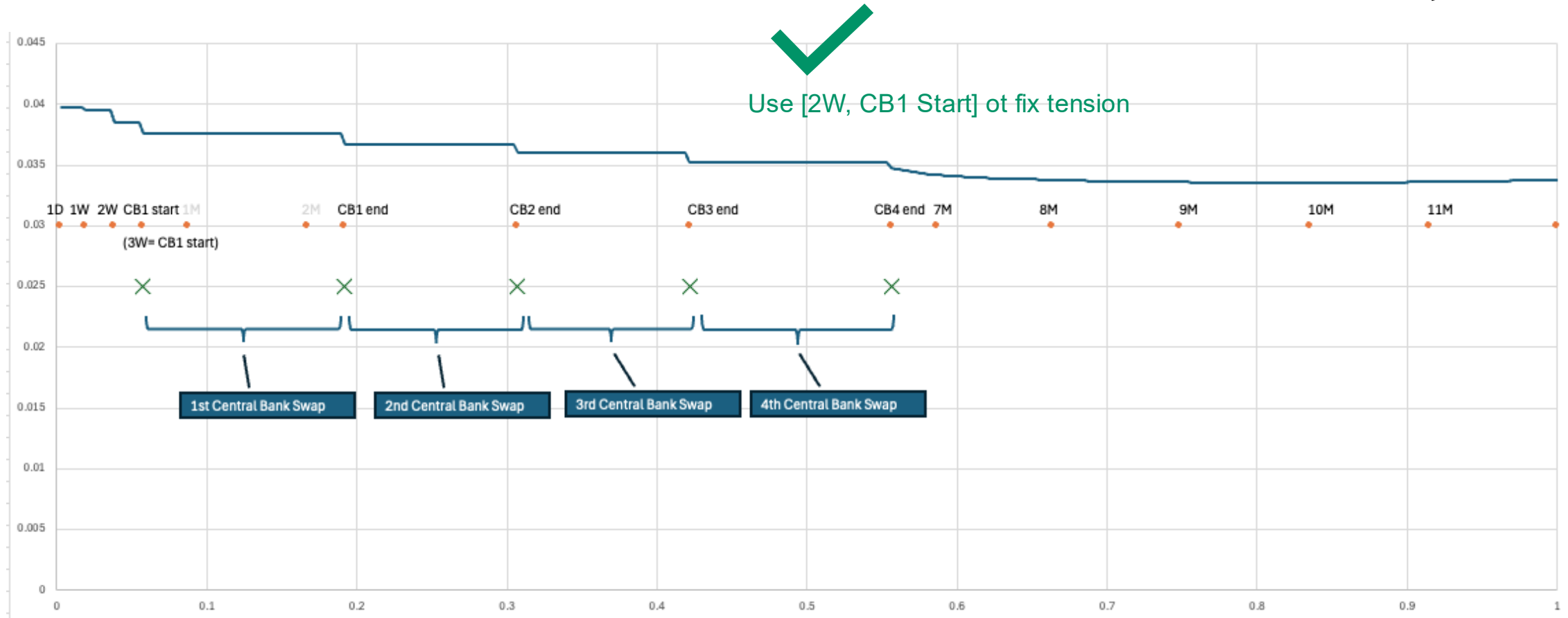
Global Bootstrap

- Can also handle overdetermined curves ($\#$ instruments $>$ $\#$ interpolation nodes)
- Exact repricing can be relaxed in favor of curve smoothness
- Applies to rate curves, but also to cap surfaces (use additional interpolation node at $t = 0$ to minimize oscillations)

Global Curve Building Example



Global Curve Building Example



We have $n+1$ interpolation nodes for n instruments. We add penalty terms for curve smoothness to use the additional degree of freedom.

```
<BootstrapConfig>
  <Accuracy>1E-14</Accuracy>
  <Global>true</Global>
  <SmoothnessLambda>0.1</SmoothnessLambda>
</BootstrapConfig>
```

Rate Curve Instruments Cashflow Report

In addition to [todaysmarketcalibration.csv](#) we produce [todaysmarketcalibration_cashflows.csv](#):

- Cashflow report for all instruments used in curve building
- Can be used to effectively reconcile details of instruments against other systems.

MarketObjectId	PillarDate	RateHelperType	MdLabel	CashflowNo	LegNo	PayDate	FlowType	Amount	Currency	Coupon	Accrual
GBP1D	2025-11-28	Deposit	MM/RATE/GBP/0D/1D	1	0	2025-11-28	Interest	109.2448178726	GBP	0.0398743585	0.002739726
GBP1D	2025-11-28	Deposit	MM/RATE/GBP/0D/1D	2	0	2025-11-28	Notional	1000000.0000000000	GBP	#N/A	#N/A
GBP1D	2025-11-28	Deposit	MM/RATE/GBP/0D/1D	3	0	2025-11-27	Notional	-1000000.0000000000	GBP	#N/A	#N/A
GBP1D	2025-12-04	OIS	IR_SWAP/RATE/GBP/0D/1D/1W	1	0	2025-12-04	Interest	765.8824657535	GBP	0.0399353000	0.01917808
GBP1D	2025-12-04	OIS	IR_SWAP/RATE/GBP/0D/1D/1W	1	1	2025-12-04	InterestProjected	-766.0158263065	GBP	0.0399422538	0.01917808
GBP1D	2025-12-11	OIS	IR_SWAP/RATE/GBP/0D/1D/2W	1	0	2025-12-11	Interest	1530.2306849314	GBP	0.0398953000	0.03835616
GBP1D	2025-12-11	OIS	IR_SWAP/RATE/GBP/0D/1D/2W	1	1	2025-12-11	InterestProjected	-1529.1866272340	GBP	0.0398680799	0.03835616
GBP1D	2026-02-05	OIS Dated	IR_SWAP/RATE/GBP/20251218/1D/20260205	1	0	2026-02-05	Interest	5086.5087671232	GBP	0.0378893000	0.13424657
GBP1D	2026-02-05	OIS Dated	IR_SWAP/RATE/GBP/20251218/1D/20260205	1	1	2026-02-05	InterestProjected	-5082.9701464099	GBP	0.0378629409	0.13424657
GBP1D	2026-03-19	OIS Dated	IR_SWAP/RATE/GBP/20260205/1D/20260319	1	0	2026-03-19	Interest	4260.7906849315	GBP	0.0370283000	0.11506849
GBP1D	2026-03-19	OIS Dated	IR_SWAP/RATE/GBP/20260205/1D/20260319	1	1	2026-03-19	InterestProjected	-4258.3353380599	GBP	0.0370069619	0.11506849
GBP1D	2026-04-30	OIS Dated	IR_SWAP/RATE/GBP/20260319/1D/20260430	1	0	2026-04-30	Interest	4175.4098630138	GBP	0.0362863000	0.11506849
GBP1D	2026-04-30	OIS Dated	IR_SWAP/RATE/GBP/20260319/1D/20260430	1	1	2026-04-30	InterestProjected	-4172.7662542415	GBP	0.0362633258	0.11506849
GBP1D	2026-06-18	OIS Dated	IR_SWAP/RATE/GBP/20260430/1D/20260618	1	0	2026-06-18	Interest	4764.8539726028	GBP	0.0354933000	0.13424657
GBP1D	2026-06-18	OIS Dated	IR_SWAP/RATE/GBP/20260430/1D/20260618	1	1	2026-06-18	InterestProjected	-4761.1282719300	GBP	0.0354655473	0.13424657
GBP1D	2026-06-29	OIS	IR_SWAP/RATE/GBP/0D/1D/7M	1	0	2026-06-29	Interest	21717.0717808219	GBP	0.0370408000	0.58630136
GBP1D	2026-06-29	OIS	IR_SWAP/RATE/GBP/0D/1D/7M	1	1	2026-06-29	InterestProjected	-21771.3920795775	GBP	0.0371334491	0.58630136
GBP1D	2026-07-27	OIS	IR_SWAP/RATE/GBP/0D/1D/8M	1	0	2026-07-27	Interest	24427.6126027396	GBP	0.0368433000	0.66301369
GBP1D	2026-07-27	OIS	IR_SWAP/RATE/GBP/0D/1D/8M	1	1	2026-07-27	InterestProjected	-24440.4605858770	GBP	0.0368626782	0.66301369
GBP1D	2026-08-27	OIS	IR_SWAP/RATE/GBP/0D/1D/9M	1	0	2026-08-27	Interest	27394.0915068494	GBP	0.0366258000	0.74794520
GBP1D	2026-08-27	OIS	IR_SWAP/RATE/GBP/0D/1D/9M	1	1	2026-08-27	InterestProjected	-27391.5841505756	GBP	0.0366224477	0.74794520
GBP1D	2026-09-28	OIS	IR_SWAP/RATE/GBP/0D/1D/10M	1	0	2026-09-28	Interest	30440.0863013699	GBP	0.0364283000	0.83561643
GBP1D	2026-09-28	OIS	IR_SWAP/RATE/GBP/0D/1D/10M	1	1	2026-09-28	InterestProjected	-30436.9184889690	GBP	0.0364245090	0.83561643
GBP1D	2026-10-27	OIS	IR_SWAP/RATE/GBP/0D/1D/11M	1	0	2026-10-27	Interest	33204.4498630137	GBP	0.0362863000	0.91506849
GBP1D	2026-10-27	OIS	IR_SWAP/RATE/GBP/0D/1D/11M	1	1	2026-10-27	InterestProjected	-33201.2753911493	GBP	0.0362828309	0.91506849
GBP1D	2026-11-27	OIS	IR_SWAP/RATE/GBP/0D/1D/1Y	1	0	2026-11-27	Interest	36165.8000000000	GBP	0.0361658000	1.00000000
GBP1D	2026-11-27	OIS	IR_SWAP/RATE/GBP/0D/1D/1Y	1	1	2026-11-27	InterestProjected	-36169.8009958191	GBP	0.0361698010	1.00000000
GBP1D	2027-11-29	OIS	IR_SWAP/RATE/GBP/0D/1D/2Y	1	0	2026-11-27	Interest	35351.7999999999	GBP	0.0353518000	1.00000000
GBP1D	2027-11-29	OIS	IR_SWAP/RATE/GBP/0D/1D/2Y	2	0	2027-11-29	Interest	35545.5084931506	GBP	0.0353518000	1.00547945
GBP1D	2027-11-29	OIS	IR_SWAP/RATE/GBP/0D/1D/2Y	1	1	2026-11-27	InterestProjected	-36169.8009958191	GBP	0.0361698010	1.00000000

AMC Support for FX Barrier Option Variants

Extend Coverage of **AMC Exposure Simulation**:

GenericBarrierOption, FxEuropeanBarrierOption, FxDoubleBarrierOption, FxBarrierOption, FxKIKOBarrierOption

Normally, this requires an implementation of a new amc-enabled pricing engine per trade type.

More economic approach that can be applied in this case: **Internal representation as scripted trade.**

```
<Product type="FxBarrierOption">
  <Model>ScriptedTrade</Model>
  <ModelParameters/>
  <Engine>ScriptedTrade</Engine>
  <EngineParameters/>
</Product>
```

```
QuantLib::ext::shared_ptr<ore::data::Trade>
FxBarrierOptionScriptedEngineBuilder::build(const Trade* trade,
                                             const QuantLib::ext::shared_ptr<EngineFactory>& engineFactory) {
    GenericBarrierOptionData data;
    if (auto fxKiKoBarrierOption = dynamic_cast<const ore::data::FxKIKOBarrierOption*>(trade);
        fxKiKoBarrierOption != nullptr) {
        data = parseFxKIKOBarrierOptionData(fxKiKoBarrierOption);
    } else if (auto fxBarrierOption = dynamic_cast<const ore::data::FxOptionWithBarrier*>(trade);
               fxBarrierOption != nullptr) {
        data = parseFxBarrierOption(fxBarrierOption);
    } else {
        QL_FAIL("FxKIKOBarrierOptionScriptedEngineBuilder::build(): trade is not a FxKIKOBarrierOption");
    }
    auto barrierOption = QuantLib::ext::make_shared<GenericBarrierOption>(
        data.underlying, data.optionData, data.barriers, data.barrierMonitoringDates, data.transatlanticBarrier,
        data.payCurrency, data.settlementDate, data.quantity, data.strike, data.amount, data.kikoType);

    barrierOption->build(engineFactory);
    return barrierOption;
}
```



Roadmap

What's next?

LSEG

ORE Roadmap – 2026 & Beyond

ORA and Risk Analytics Lab

- Historical VaR Optimization
- Corporate Treasury Support
- Analytics (e.g. Stressed SIMM)
- Documentation

Advanced xVA models

- More factors: QGM, n-factor HW
- Beyond IR...
- Local vols
- Stochastic Vols (Heston+...)

xVA Optimization

- RFE specific pricers
- Bypass some QuantLib structures
- Refactoring to reduce repricing & devirtualization
- GPU (long-term)

Scripted Trades

- Deeper debugging
 - Paths outputs
 - Variable logging
- More functions
- GPU Pricing

Trade Coverage

- Listed Derivatives
- Enhancement of commodity features and new instruments
- Classic exotics (e.g. lookbacks)
- Expansion of Trade Generator

Analytics Extensions

- CVA sensis with AAD
- More SA-CCR coverage
- More FRTB-SA coverage

Closing Remarks

Scott Sobolewski

Co-Head Quantitative Services

LSEG

Post Trade Solutions

16:55



Networking Reception

