EMiT - Emerging Technology Conference 2014, Manchester, April 11th 2014

High Performance Computing in Finance
Why HPC? Why now?

Erik Vynckier Chief Investment Officer — Insurance EMEA

AllianceBernstein

Why High Performance Computing in Finance? Why Now?

- Daunting numerical challenges in risk management
  - Value-at-Risk and coherent **tail risk measures for complex businesses**
  - **Stress testing** and preparedness for crisis with a living will
  - Regulators and politicians hammering firms to post satisfactory **capital**
  - Reporting requirements in **short time-frames** (Basel III, Solvency II)
  - **Stochastic-on-stochastic** “nested” Monte Carlo simulations – a computational bottleneck
- Financial product variety and complexity
  - Products with mix of **financial** and **insurance** features offering clients **discretion**
  - Hybrid or **cross-asset** exposure—equity, bonds, currencies, commodities
- Need for **real-time** pricing, risk, trade and positions metrics
  - **High frequency** trading
  - **Dynamic hedging** of guaranteed products
  - Counterparty modeling and real-time selection of **counterparties**
- Restructuring adds to demand in balance sheet and risk computing

Business accelerates: computational tools need to accommodate the change agenda

Simple Mathematics: How much is $ 1 paid tomorrow, worth today?

- Bootstrapping was considered straightforward, yet it is controversial
  - By far the most important procedure in finance — $$$ monetary impact in settling trades
  - **Interpolation techniques** matter — **discrepancies** frequent and large
- "**Dual-curve**" discounting of $/€/£ Libor products
  - Fed Funds, EONIA, SONIA money market collateral priced off **basis swaps**
  - Foreign currency collateral priced off **cross-currency basis swaps**
  - Corporate debt and other **credit-risky collateral**
- **Cheapest-to-deliver** discounting
  - Easy but wrong: cheapest to deliver on a forward basis
  - Accounting for the **substitution rights on collateral** path dependent
- Maintenance of existing trades
  - **Recouponing** in-the-money swaps
  - A third counterparty steps into in existing trade: **novation**
  - **Trade compression, reducing triangular trades**, meets 3 ISDA/CSA
  - Settlements in cash invariably controversial and adversarial in nature

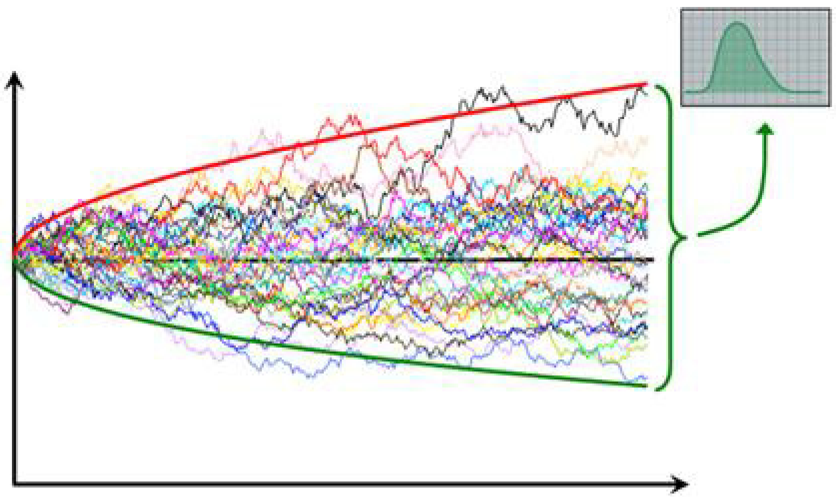
Complicated Mathematics: Stochastic Calculus

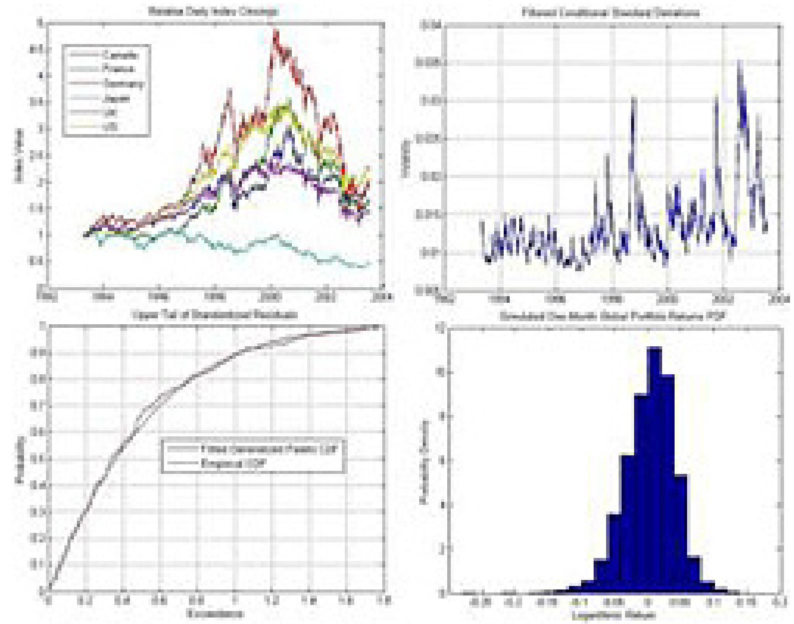
- **Partial differential equations** with multiple risk factors for option value V(S,t)

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

- **Monte Carlo** simulations (equivalence derived from Feynman-Kac theorem)
  - Just too many risk factors … just too many product features … or you don't know any better method
  - Random-number generators for chosen platform?
  - Variance reduction methods for chosen platform, importance sampling, structured quasi-random numbers
  - Least-squares Monte Carlo and replication portfolios
- Discrete or Fast **Fourier Transforms** for known characteristic functions of probability laws
- Even "analytical" solutions rest on transcendental functions — iteratively approximated
- Risk sensitivities — the so-called "**Greeks**" — useful for risk managing a book of derivatives, for hedging and for estimating bid-ask spreads
- **Calibration to market + valuation + Greeks + all strikes + all maturities + all pay-offs on the trading book + ability to implement stress tests** is the real benchmark for calling a valuation method "efficient"

Monte Carlo Simulation in Finance often Embarrassingly Parallel



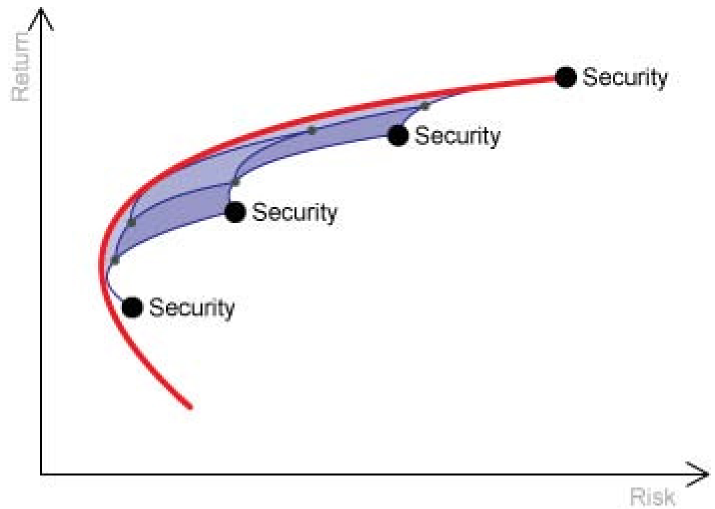


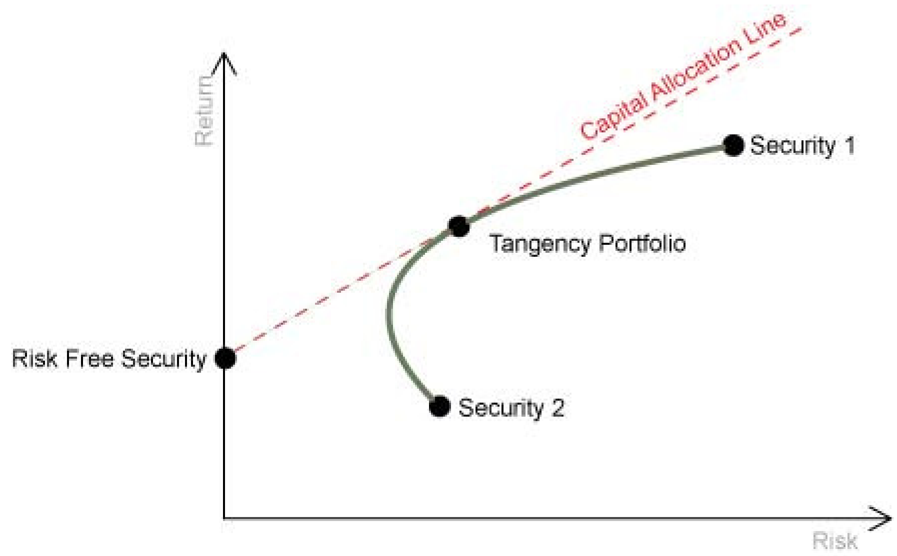
Entrenched method for modelling financial risks and deriving metrics, such as option prices, risk sensitivities and Value at Risk

Explicit and Hidden Markowitz in Portfolio Optimization

- Portfolio optimization trading off risks and returns
  - Capital Asset Pricing Model: **Minimize T(weights) x Covariance x weights** subject to return constraint
    - Mythical investors investing according to **mythical** utility functions
    - “Arbitrage” of the past record or a flawed numerical treatment
  - Tricking the Covariance matrix
    - Time-weighting
    - Bayesian prior
      - + You bet less aggressively on a flawed approach
      - – You do not bet with better odds
      - – Single horizon implies no time dimension, so no cash flows in or out, portfolio rebalancing
  - Robust prediction of expected returns is virtually impossible
- Variations on a theme by Markowitz: risk and returns **relative to benchmark**
  - Black-Litterman optimization
  - Run active alpha versus tracking error as a risk measure
  - Long-short optimization

Efficient Frontier Optimization with Sequential Quadratic Programming





One-for-one porting of mathematical techniques from the physical sciences into investment practice has often produced catastrophic outcomes

2nd Wave of Portfolio Optimization: Dynamic Stochastic Programming

- Newer metrics and newer optimization techniques
  - **Risk parity** instead of market cap weighting
  - Identify, invest in and mix and match **risk premium from diverse risk factors**
  - Rely on the **central limit theorem** to reduce skew and kurtosis
  - **Entropy pooling** for optimization of asset allocation
  - **Time is of essence:** lifecycle savings and drawdown, dynamic strategies
- **Attribution** of risk, return, capital, liquidity …
  - Decompose aggregate metric into its **marginal contributions**
  - **Copulas** for aggregating risk factors **across risks and business lines** — admits **fat tails**
- Multiple time-steps as intermediate experience matters to the investor
  - Cash in- and out-flows need to reflected
  - Dynamic asset allocation, reflection trading opportunities and cost of illiquidity

Dynamic stochastic programming is in my view the most promising approach

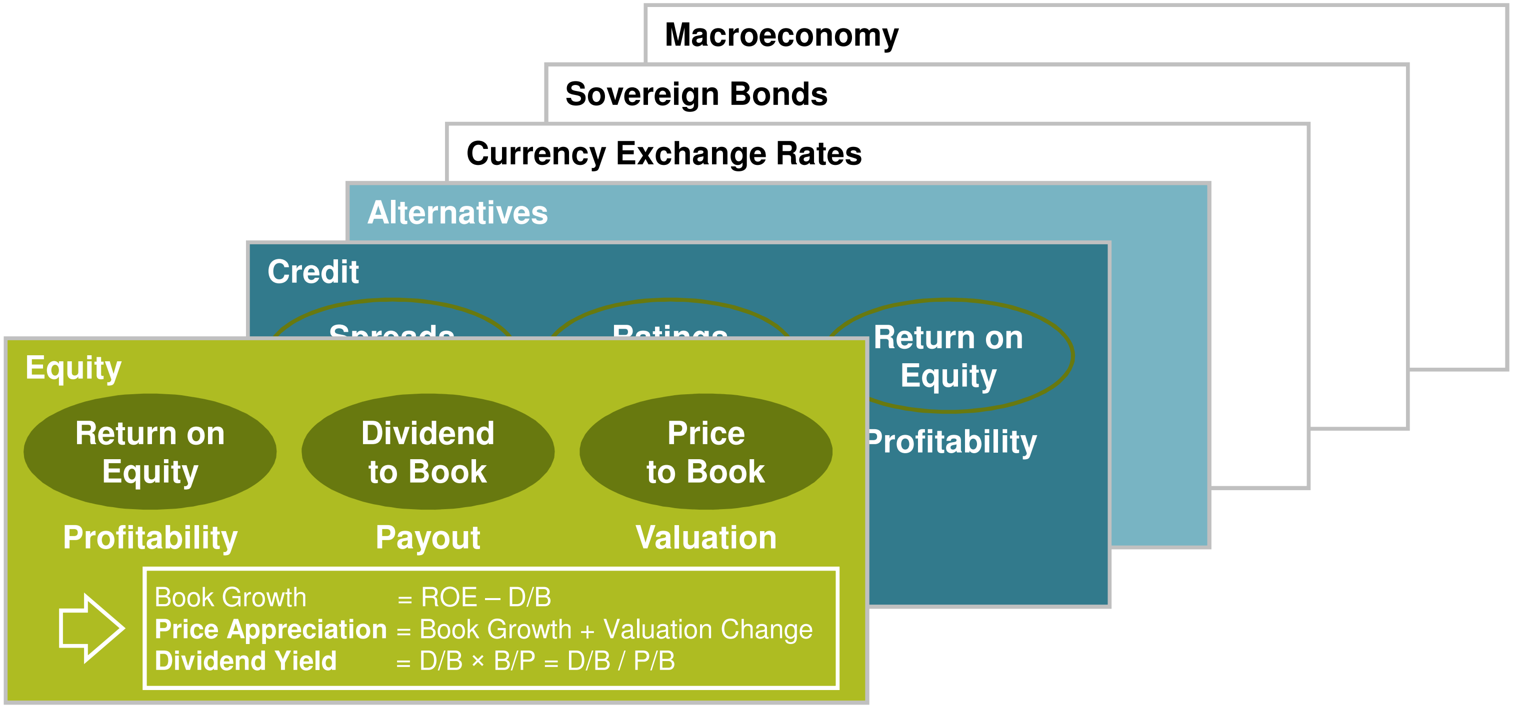
What's Special About Our Capital Markets Engine?

- **Real-world scenarios** crystallizing financial risk premiums, characterize the **risk and return universe** for investments
- Projections are ranges and probabilities, not point forecasts
- Multiple horizons, cash in- and out-flows can be modelled
- Global & multi-currency ($, €, £, …)
- Grounded in the present
- Economically underpinned
- Consistent across asset classes

Typically 10,000 scenarios are needed for converging, reliable outcomes

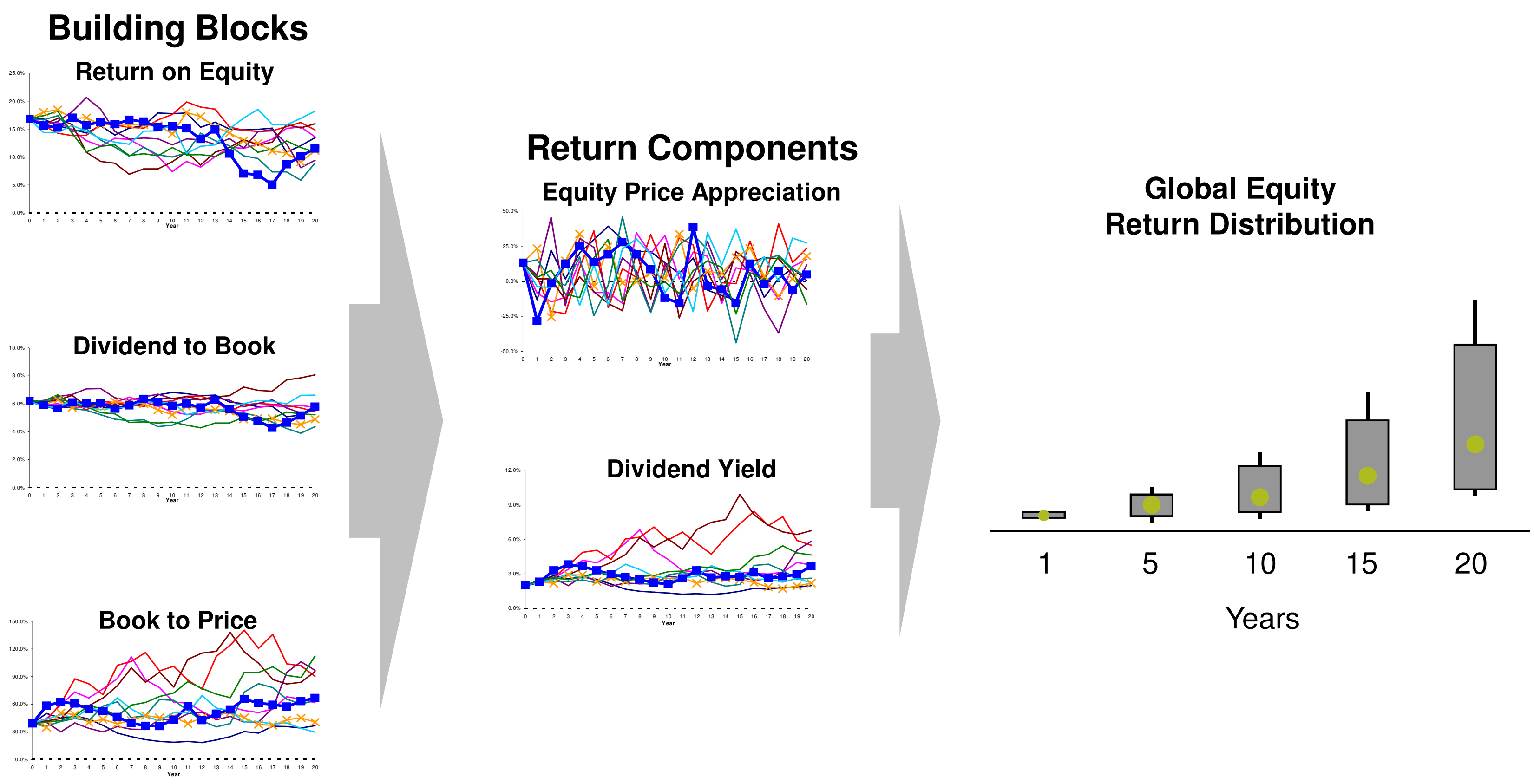
AllianceBernstein's Capital Markets Engine is US patent pending.

Why We Focus on Modelling the Economic Building Blocks of Return



- **Linkages across markets arise from common underlying economics:** corporate profitability, economic growth, yield levels, returns in stock and credit markets.
- We glean how the system might evolve from its historical time series, plus economic and accounting logic
- **We can project returns for a wide array of asset classes** — equities, credit, real estate and alternatives — consistently

How We Combine Building Blocks into Asset Returns: Equity

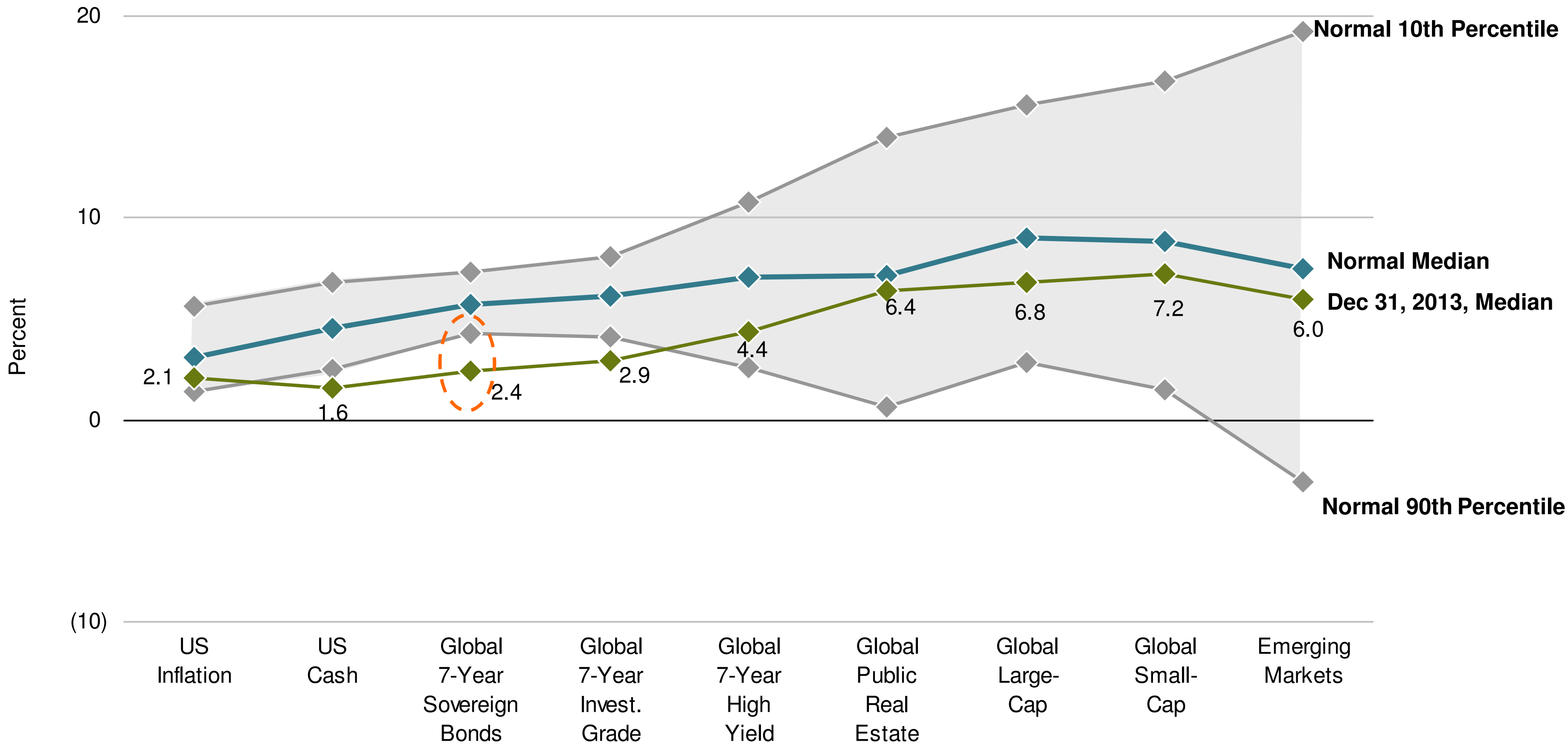


Note: Return on equity (ROE) cycles along each path; dividend to book (D/B) is more stable, and book to price (B/P) is more volatile. On the blue path, when ROE falls sharply, D/B dips as well.

Grounded in the Present: Low Yields a Challenge for Bond Investors

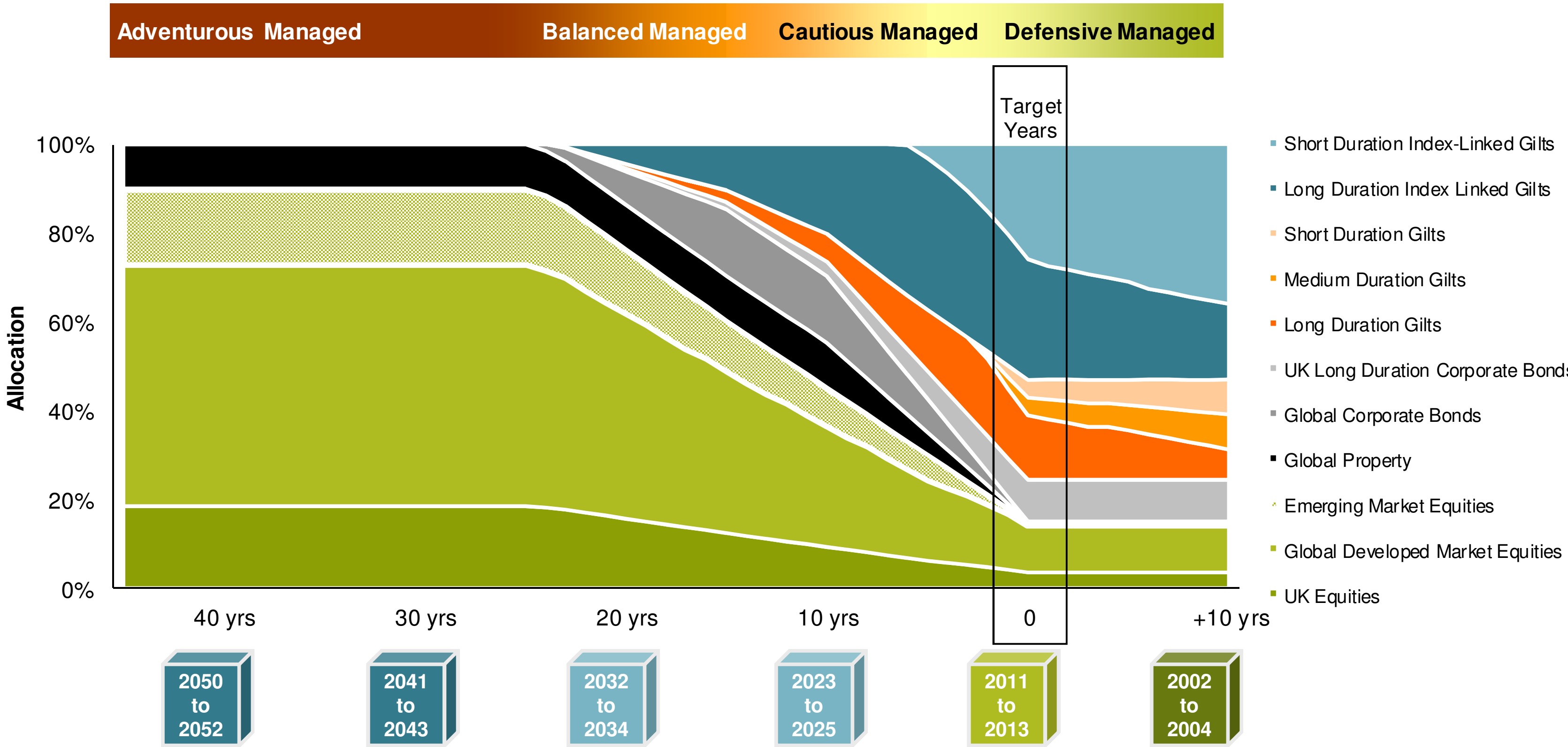
In US dollars

Range of Compound Growth Rates over 10 Years



As of Dec 31, 2013
Returns hedged into US dollars and reported in US dollars
Data do not represent past performance and are not a promise of actual returns or range of future results.
Source: AllianceBernstein

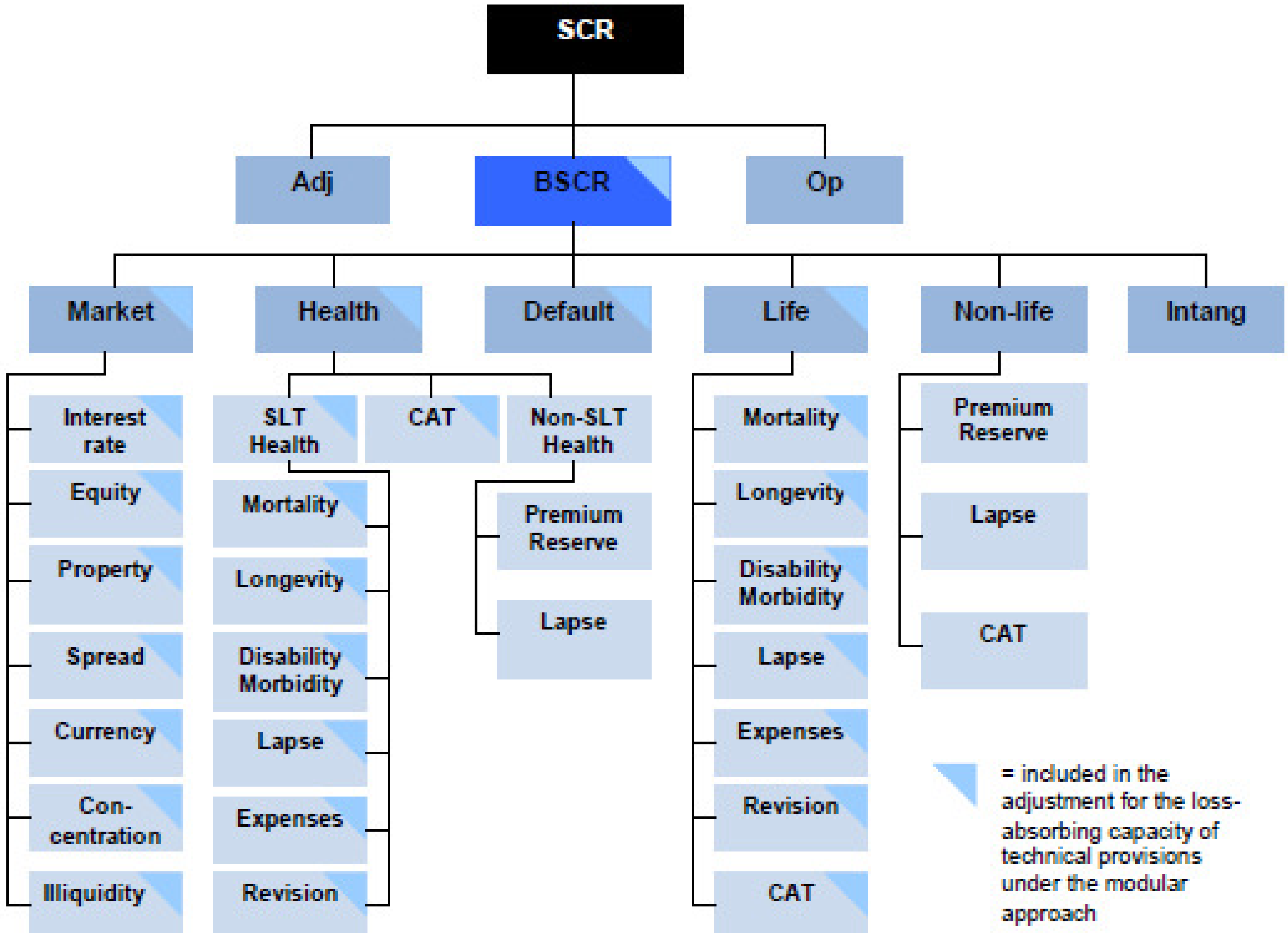
Applying Scenarios: Defined Contribution Glide Path



How should an investor allocate money to investments before and in retirement to increase wealth for consumption and reduce the risk of running out of savings?

Applying Scenarios: Internal Model for Solvency II/Basel III Capital

- **Standard model** has a modular build for solvency capital requirement in view of each material risk
  - Market risks
  - Insurance risks
  - Calibrated stresses
  - Calibrated correlations to combine modules
- **Internal model**
  - Scenario generator
  - Use high performance computation
  - Businesses and assets which are poorly covered in the standard models



Accurate tail risk metrics require a massive number of scenarios

Trading & Investment: Market Infrastructure and Market Conduct
Dodd-Frank[1] (US) & EMIR[2] (EU)

- Clearing for selected derivatives
  - Initial & variation margin
- Collateral (cash or eligible securities) will be tight
  - Increased demands on the repo markets
  - Feeds back into portfolio allocation and returns
- Transparent markets through SEF – swap execution facilities

Computational challenge of planning collateral for a book of assets and derivatives

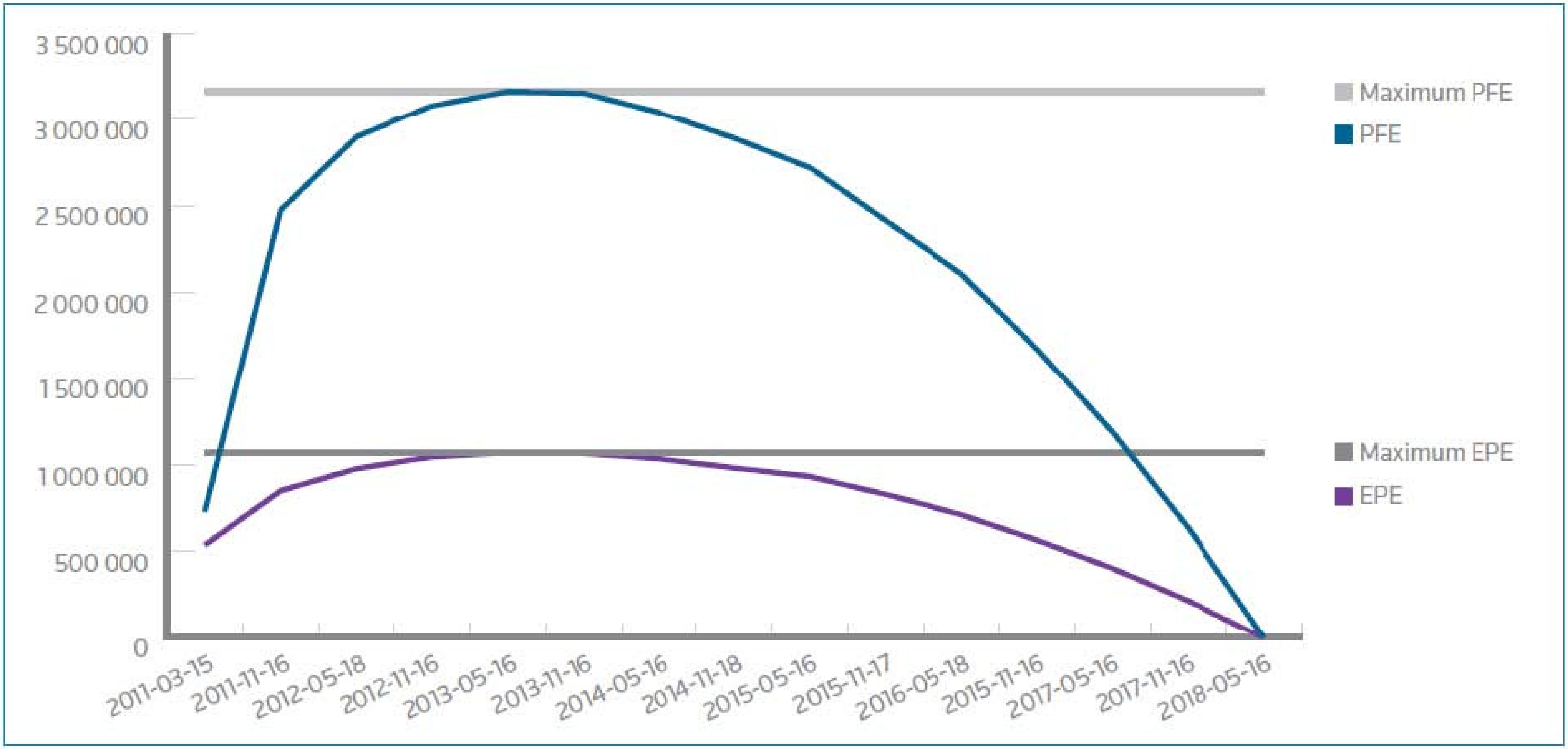
(1) Dodd-Frank Wall Street Reform and Consumer Protection Act
(2) EMIR = European Market Infrastructure Regulation

Counterparties and the Credit Crunch

- Counterparty risk is real, particularly so in the absence of collateral
  - At asset-liability driven institutions, equity and interest rate risks order of magnitude higher!
  - We need to **manage counterparty risk, not necessarily eliminate it**
  - Start with reviewing ISDA specifications and Credit Support Annexes
- Metrics are different for different institutions
  - For uncollateralized trades: Credit and Debt Value Adjustments (CVA, DVA)
  - For collateralized trades: Potential and Expected Future Exposure
  - Potential impact on liquidity of the institutions (MetallGesellschaft)
- Netting off derivatives positions and collateral across legal entities on both sides

Multiple time-step, path-dependent simulations across the complete book of assets, derivatives and counterparties over the entire life of the trades

Potential Expected Future Exposure for a Range-Accrual Swap



Ticks, Time-Series, Confidentiality and Crime

- Large data
  - Financial databases
  - Real-time data, transaction data — **kdb** tool
  - Data-mining consumer databases for consumer behavior
  - **Customize** marketing and pricing of insurance and retail financial products
  - **Protocols**: SQL, XML, FPML, LEI, corporate actions…
- Size is the smaller issue of large data
  - Large data varies in **source**, it's **unstructured, contains errors and N/As**
  - The industry is really struggling to find an approach to **mine noisy large data** for **info**
- **Crime!**
  - Detection — **cross-checking** large data sets originating from independent and novel sources
  - From detection to prevention — with a **real-time** toolkit?
  - Encryption of data — encryption standards as of yet, unbreakable

RSA Encryption Algorithm – under threat by Shor's Algorithm

Key Generation

1. Choose two very large random prime integers: p and q
2. Compute $n = pq$ and $\varphi(n) = (p-1)(q-1)$
3. Choose e, $1 < e < \varphi(n) : \gcd(e, \varphi(n)) = 1$
4. Compute d, $1 < d < \varphi(n) : ed \equiv 1 \pmod{\varphi(n)}$

the ***public*** key is (n, e) and the ***private*** key is (n, d)

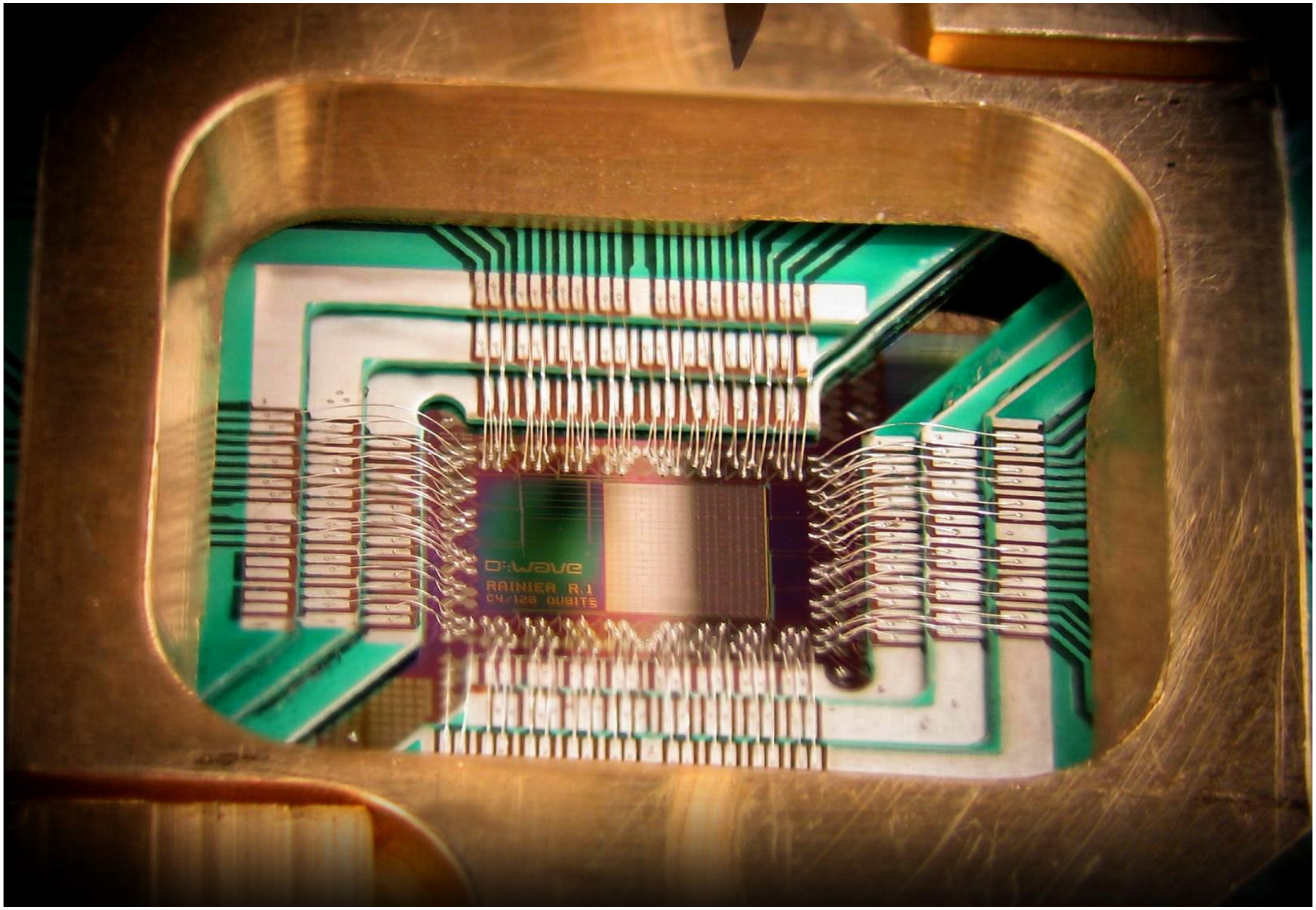
Encryption

Cyphertext: $C = M^e \bmod n$, where M is the original message.

Decryption

Plaintext: $M = C^d \bmod n$

D-Wave 128-Qubit Superconducting Adiabatic Quantum Processor



Why Now?

- High-performance computing is **no longer esoteric**:
  - **Moore's "bet"**: a new chip design for a new foundry every two years
  - Limits in cooling and quantum effects at very small transistor sizes have halted progress
  - **Many** cores: NVIDIA (very many), Intel XeonPhi
  - New platforms — **use all of the silicon all off the time**: Altera, Xilinx
  - High performance computing is now established in many other sciences and technologies
- High performance computing has become **affordable**!
- Financial engineers are still betting on Moore's law
  - Higher integrated circuit density and clock speed are not forthcoming
  - As of old we write "correct" code and expect the compiler to handle it

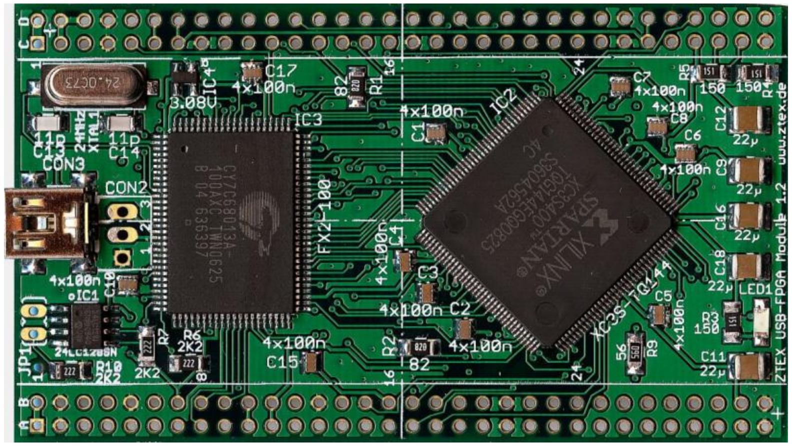
Optimal mix of platform + algorithm + numerics + code new game in quant development

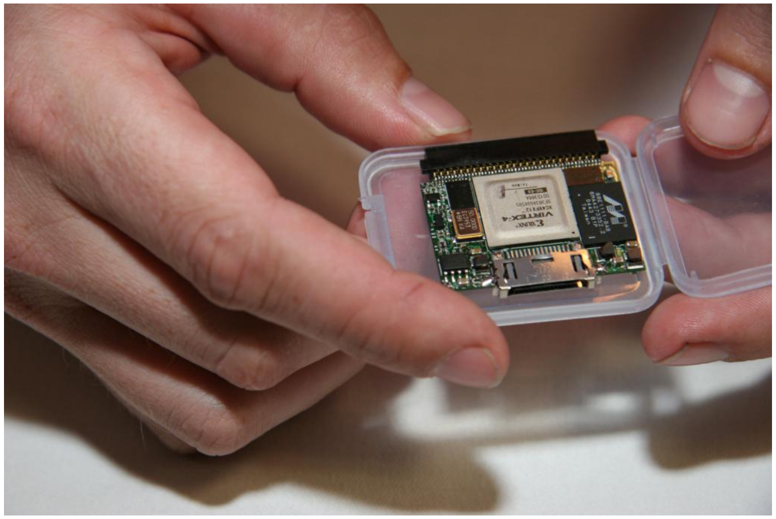
HECToR (EPCC): Cray Supercomputing

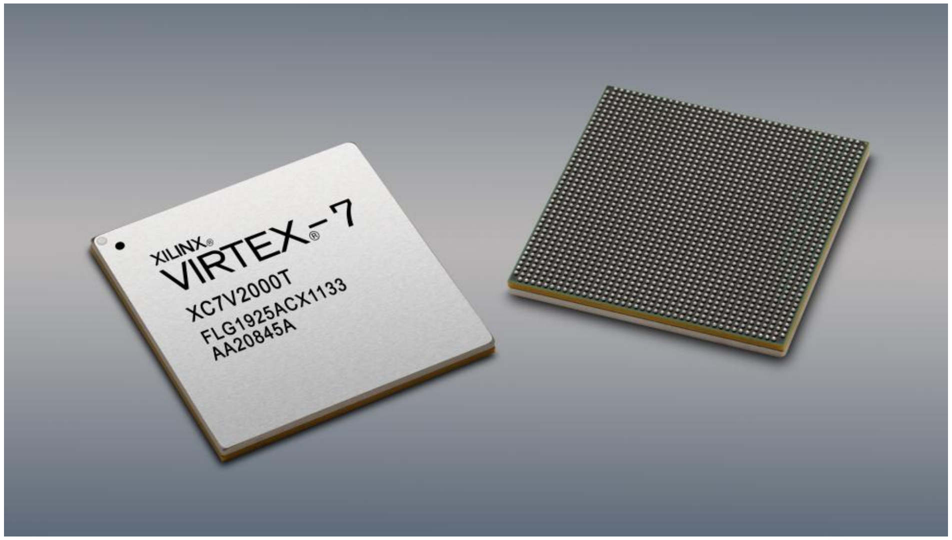
What Are the Options?

- **Cheap** high performance computing has arrived!
- **Multi-core** lends itself well to embarrassingly parallel tasks
  - nVidia Tesla, Kepler, Maxwell…, Xeon Phi
  - OpenCL, OpenACC, OpenMP, CUDA
  - Bandwidth considerations on data movement & saturating the cores
  - Libraries (NAG, nVidia, MathWorks), domain specific languages (SciComp)
- **Reconfigurable gate arrays**
  - Verilog High Definition Language a daunting task for financial engineers
  - Standard and portable OpenCL for Xilinx FPGA
  - MaxCompiler with a high-level language for dataflow computing on FPGA
  - MATLAB VHDL toolkit
- Grids are now called **clouds**
  - On-demand **bursting** to needed capacity fills punctual requirements
  - Outsourced, low capital cost at the risk of dependency on provider
  - Azure, amazon, peer one…outsourced service organizations operating under a Service Level Agreement
  - **Hosting the data in the cloud**

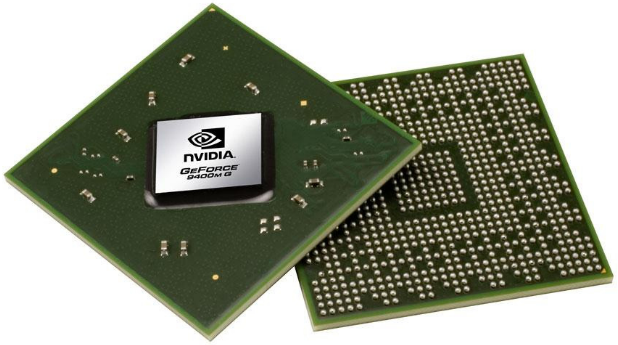
Field-Programmable Gate Arrays – Computational Pipeline for Dataflow

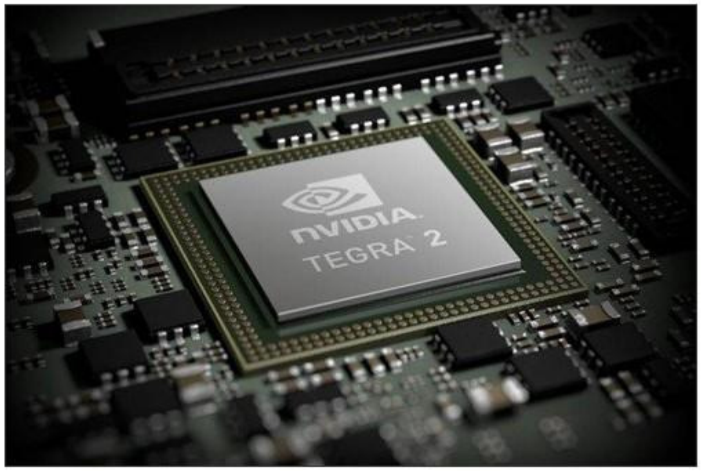




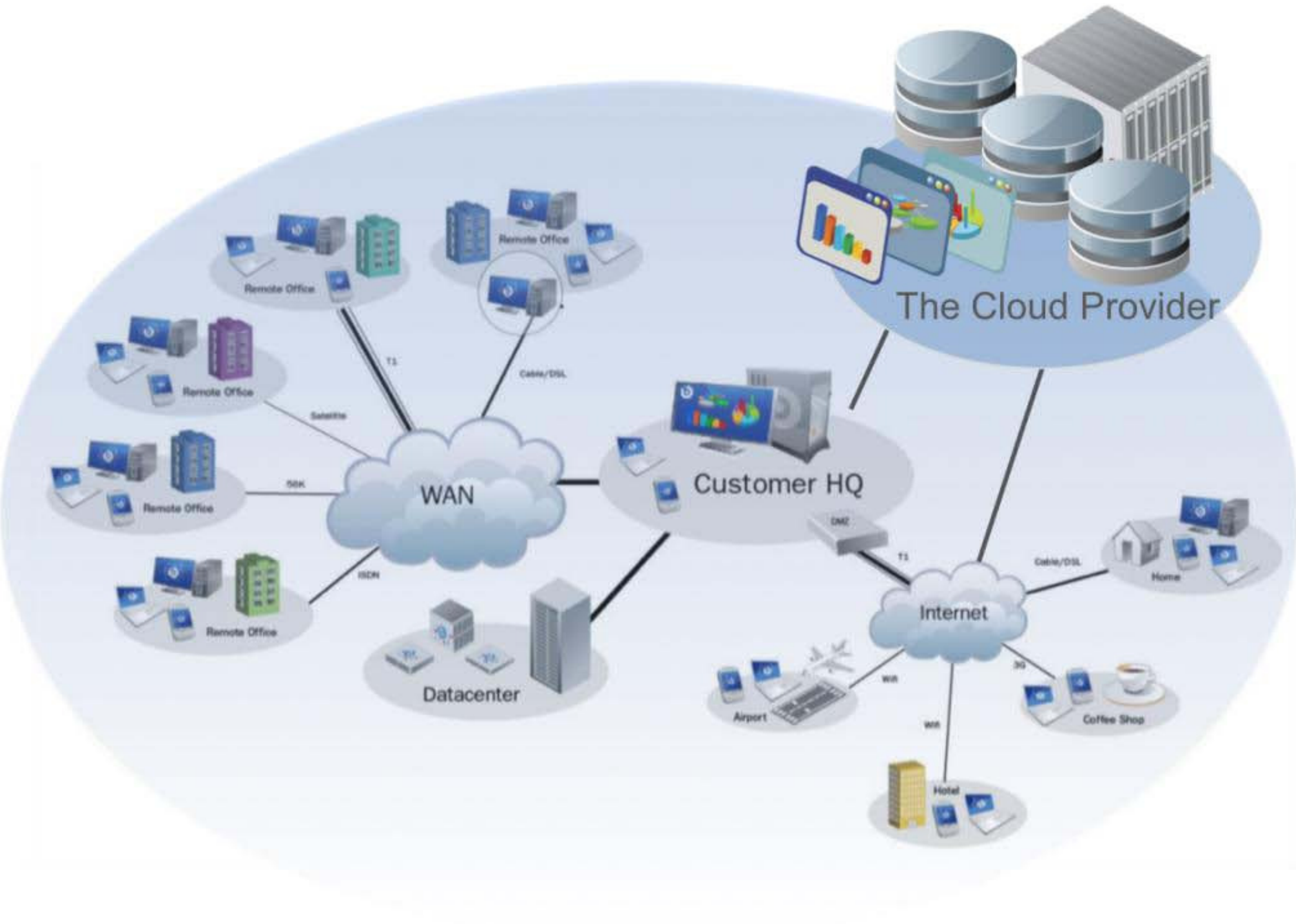


NVIDIA (GeForce, Tesla, Kepler)





The Grid Has Become a Cloud



Platforms Come and Go … but Applications Remain

- When developing, do we know which platform we will run on?
  - Turn code into binary only when the target platform is identified: Java >> intermediate language >> **virtual machine** >> bytes
  - How to manage data, libraries, … and output in this concept?
  - **Compiler directives** to the pre-processor to do the hard work?
  - Defer to **libraries** or **domain specific languages** to do the hard work?
- The **software life-cycle**
  - Rewrite? Re-factor? Port and Run?
  - Initially clean code base starts to diverge into platforms versions
- Application to very large codes and computational tasks (Basel III, Solvency II)?
- Ultimately I want **infrastructure and software that follows and fits my business**

Cost of ownership — static and dynamic — across versions of hard and soft

Bumping to Compute Risk is Slow and Inaccurate

- Calibration of model to market data and valuation of financial contract
- **As well as**
- Greeks: perturbing parameter with a bump or are there better techniques?
- Entire trading book
  - All maturities
  - All tenors
  - All strikes
  - Plain vanilla (linear pay-offs) and exotics (more complicated pay-offs)
- Stress tests — should not be an afterthought of model and code development

Plan the code to do the full job before hacking your first lines

PDEs: Get your Greeks for Free!

- Ultimately linearization (often Finite Differences) and Gaussian elimination
  - Decompose into LU **once**, then **column-band** the matrix of PDE coefficients with all pay-offs, all strikes
  - Add columns with extra **pay-offs as you back-solve** — solve across all maturities in one go
- People don’t apply this economy because:
  - They don’t see through a layered, verbose object-oriented code
  - The object orientation doesn’t allow them to re-factor for it
- **The most important Greeks == balancing terms of the PDE == free**, once you have solved for the pay-off

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

- Alternating Direction Integration for several factors (stochastic volatility, stochastic rates)
- **Boundary conditions** can be tricky to formulate and to achieve sufficient accuracy
- Consider handling **discontinuities** with smoothing techniques (Rannacher, sparse grids)

Tossing a Coin Should Be Carefully Engineered

Generate Monte Carlo scenarios just once

- What range of inputs should the scenarios cover? At what time-step?
- How do you calculate the Greeks and provide for stress tests?
- Is there a role for importance sampling or other advanced techniques?

Adapted numeric methods for stochastic differential equations

- Strong versus weak convergence of numeric SDE solvers
- **Taming** of processes that can run away, such as square-root diffusions
- What is the order of the selected method for SDE?
- Handle accuracy of a solution in **multi-level** approach:
  - Scenarios n, N, time-steps dt, dT (n << N, dt << dT)
  - Accurate path integration: n × dt (minimize discretization error)
  - Diversifying away the random error: N × dT (minimize sampling error)
  - Fine-tune not N x dt but rather **optimize n × dt + N × dT** for cheapest accuracy

Least squares Monte Carlo

- Think through how you obtain Greeks, how you accommodate stress tests, early on
- Inspired choice of **basis functions** which could be “replicating portfolios” — get asymptotic behavior right
- Quantify accuracy of the LSMC, **numerical experiments are insufficient**

Automatic Differentiation: a Smooth Ride

- **Symbolic differentiation** or **interval methods** or **hyper-dual numbers**?
  - **d** (for (i = 0; i < n; **f**(i++)) == for (i = 0; i < n; **(df)**(i++))
  - **f**([a : b]) = [**f**(a) : **f**(b)] (for f monotonically increasing)
  - **d**f(a + b.**i** + c.**j** + d.**k**) (Taylor expansion on hyper-dual arithmetic)
- Forward or backward **adjoint**: input and output dimensions determine approach
- Include calibration?
  - From market data to parametric model: derive and compute **Hessian** through AD
  - From parametric model to financial pay-off: AD
  - From **market data to financial pay-off** with the chain rule: AD × AD = AD
- True benchmark — Automatic Differentiation for full trading books
- The mathematical case is easily proven, yet the implementation may fail
  - Discontinuities and kinks are common in financial pay-offs
  - **Memory storage and retrieval** versus computational cost, even for well-behaved problems

Quo Vadis

- Hardware & Platforms: gates, cores, clouds, compilers?
- Software & Applications
  - Code or re-factor — for one or multiple targets
  - Compiler directives
  - Seek domain-specific libraries or languages
  - A “locked” or an extensible vendor product
  - Aim for target chip/platform or portable design or cloud or hybrid cloud?
- Algorithms & Numerical Methods
  - **Mathematics is more effective than “compute and error”**
- Optimal collaboration between experts

Need for cross-trained financial engineers who span the subject domains of finance, numerical mathematics and high performance computing