CLOUD AND BANKING IT

- EACH CAN LEARN FROM ONE ANOTHER

RICHARD CROUCHER

QCON London, 2013

Background and Biases

- Experienced at of all layers of the infrastructure and with devices from SmartPhones to Mainframes.
- Programs in multiple languages, having started with Assembler on his self built computers.
  - Transitioned through Pascal, C, C++, Java and C#
  - Now a strong proponent of Erlang.
- Cloud
  - Lead designer for one of the first Cloud based Grid offerings designed to support Banks
  - Lead DevOps Architect in Cloud DevOps team at Microsoft Live
  - Provided support to several Cloud startups as interim CTO
- Banking
  - Gained experience of Banking IT systems in both consulting and permanent roles with Commerzbank, CreditSuisse, DeutscheBank, HSBC, Flowtraders, Merrill Lynch and RBS.
  - Specialist on low latency and high frequency trading systems
- Technologist not a Banker

Intro

- Banks run massive IT systems and so do Cloud providers so why are so different?
- Why has the uptake of Cloud by the Banks been so low?
- What types of computing are these environments optimized for?
- We describe the characteristics of each; both are pushing the computing envelope in different ways.
- Finally, we'll show how there are lots areas of commonality and both can benefit from leveraging technologies and design patterns from the other.

Macro Differences

Informatix Solutions

Cloud

- Designed for massive scale
- Net neutrality – it's not our fault if there's porn/copyright theft, libel etc.
  - Only just starting to take some small level of responsibility
- Suck it and see
- Lots of the same
- Write own code leveraging and donating to open source
- 24x7 operations – when is a good time to shut down Google, Hotmail.....
- Every thing Internet Accessed – under constant attack
- Willing to sacrifice a few customers to save money
- Cost optimized systems
- # of users difficult to predict – design for 10x scalability

Financial Services

- Designed for high reliability
- Highly regulated industry with seriously large fines leading to potential jail sentences

- Strict Change Control
- Everything is different/special
- Buy most software as products but then customize – making nervous steps with open source
- Daily maintenance windows, regular Change freezes
- Highly secure, multi-level secure environments
- Protective about reputation
- "Gold Plated" systems with no SPOFs
- # of Users predictable
  - Except for Trading systems where need to cope with bursty market conditions

FINANCIALS SERVICES

Financial Services Principle Players

Informatix Solutions

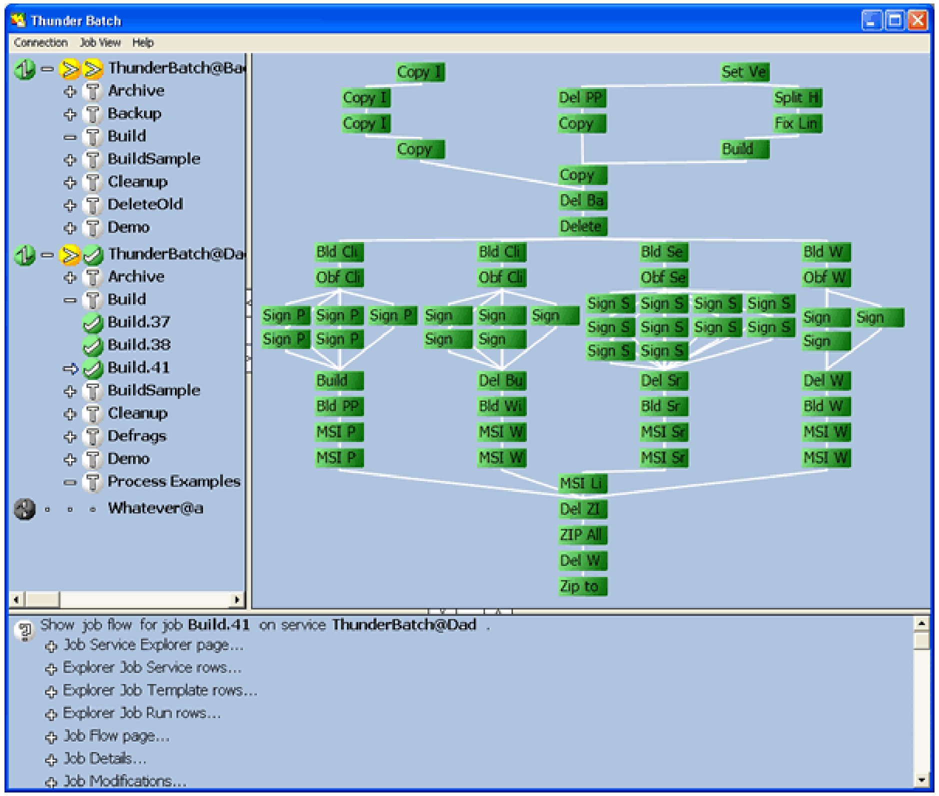
- Banks
  - Retail
  - Investment
  - Corporate
  - Wealth
- Venues
  - LSE, NYSE, Euronext, NASDAQ, DeutscheBorse, SGX, TSE....
  - Multilateral Trading Facilities – BATs, Turquoise
  - Dark Pools - Crossfinder, Barclays LX, Goldmans Sigma
  - + Options, Futures, FX marketplaces
- Hedge Funds
- Pension Funds
- Insurance Companies
- Credit Card

Retail Banking

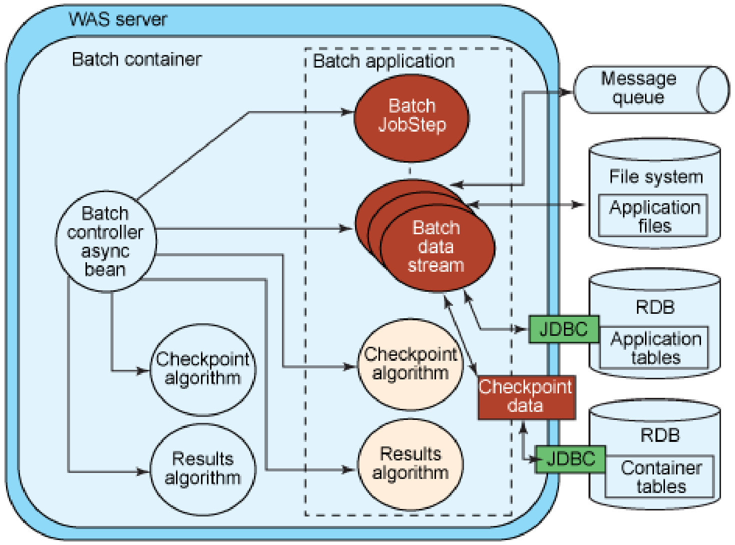
- Mass market banking servicing
- ATM Networks
- Branches
- Offering checking accounts, savings accounts, mortgages, personal loans, debit cards, credit cards insurance, Internet Banking, Telephone Banking
- Checking accounts (in UK) are a loss leader to attract clients to then sell other profitable products to
- Back office is still mostly Mainframe, batch based
- Mobile apps increasingly becoming a differentiator

Batch Processing

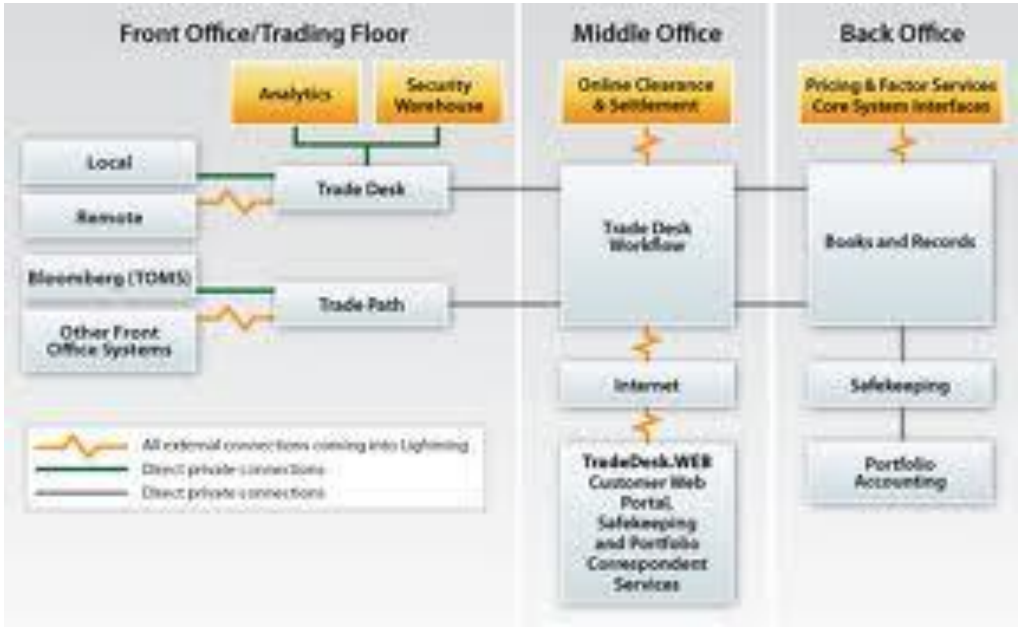
Thousands of batch jobs run every night
Must all finish within set time



Show job flow for job Build.41 on service ThunderBatch@Dad .
Job Service Explorer page...
Explorer Job Service rows...
Explorer Job Template rows...
Explorer Job Run rows...
Job Flow page...
Job Details...
Job Modifications...



Huge number of interdependencies
Integration mostly mostly MQseries messaging and file deposit/wait for


Front Office/Trading Floor
Middle Office
Back Office
Analytics
Security Warehouse
Online Clearance & Settlement
Pricing & Factor Services Core System Interfaces
Local
Remote
Trade Desk
Bloomberg (TOMS)
Other Front Office Systems
Trade Path
Trade Desk Workflow
Books and Records
Internet
Safekeeping
Portfolio Accounting
TradeDesk.WEB Customer Web Portal, Safekeeping and Portfolio Correspondent Services
All external connections coming into Lightning
Direct private connections
Direct private connections

Front Office

•Traders sit on Trading rooms, each with several screens and a fast dialler.

•Punctuated by shouts when someone makes a good deal

Middle Office

•Accountants, Economists, Mathematicians create strategies, watch exposure and risk on client portfolios.

•Trades are matched, cleared and settled

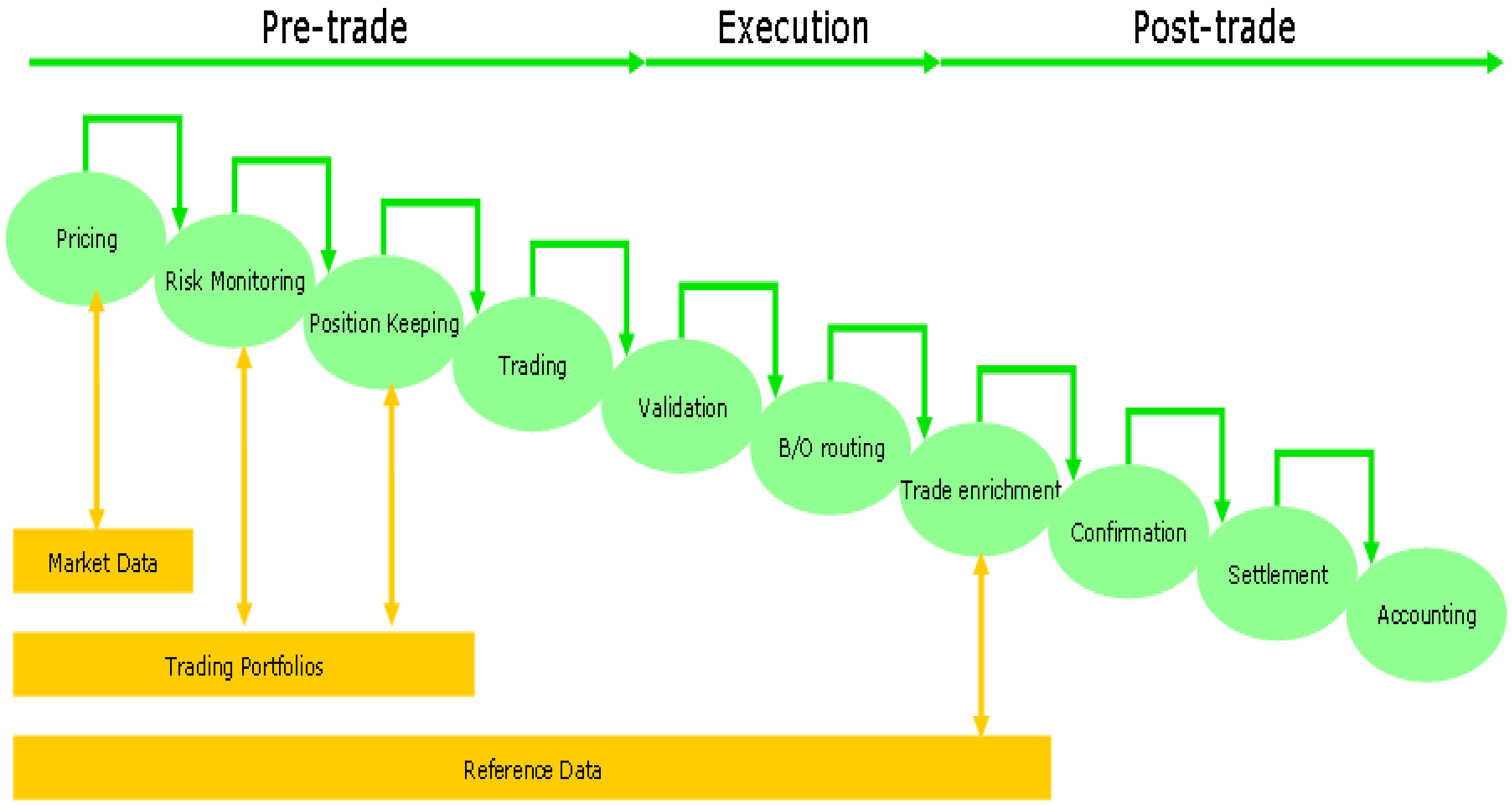
Back Office

•The day's cash movements are aggregated and payment instructions sent out

•The Banks overnight positions are re-calculated and updated

Trade Flow

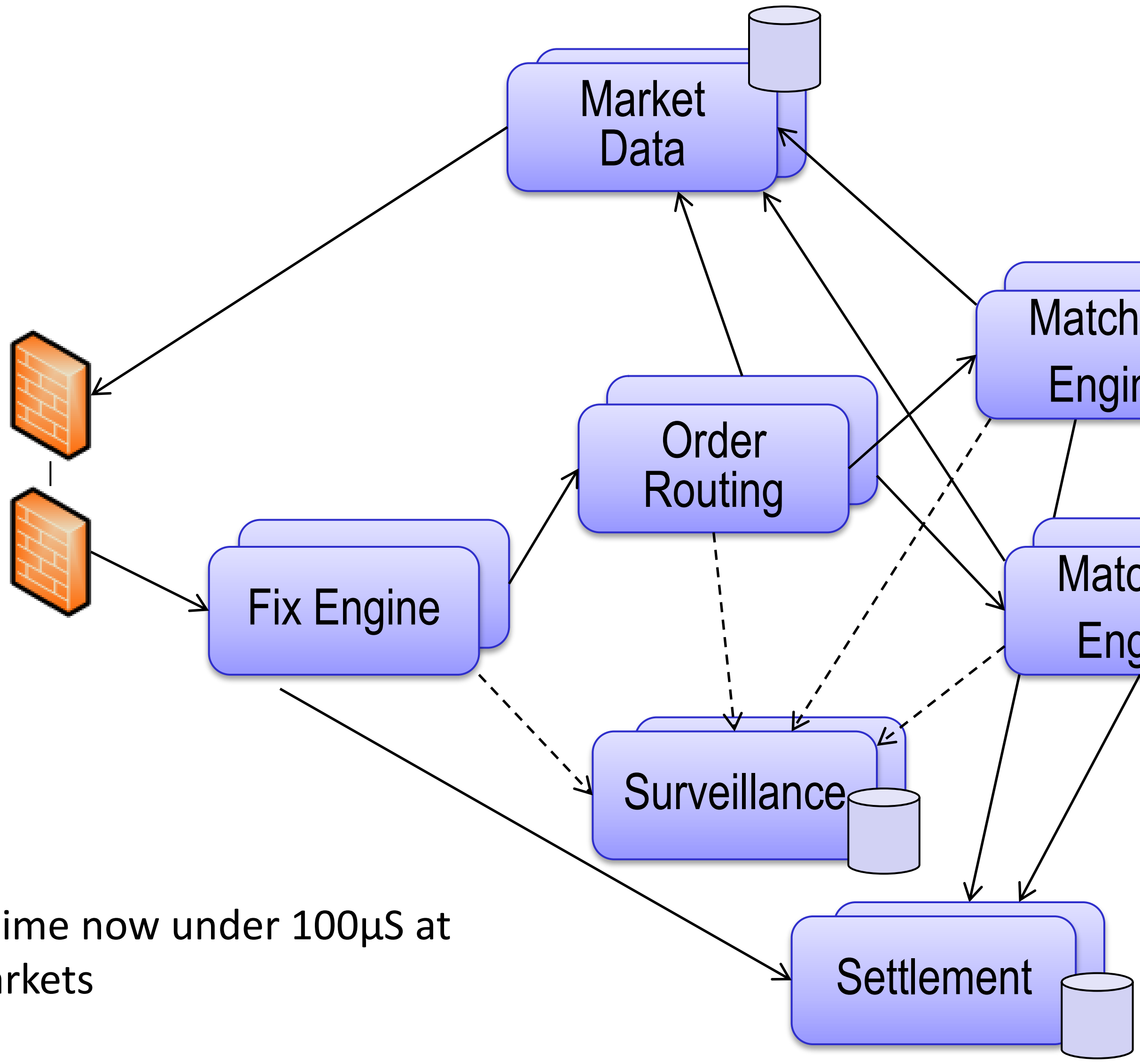
Informatix Solutions



Risk Analysis

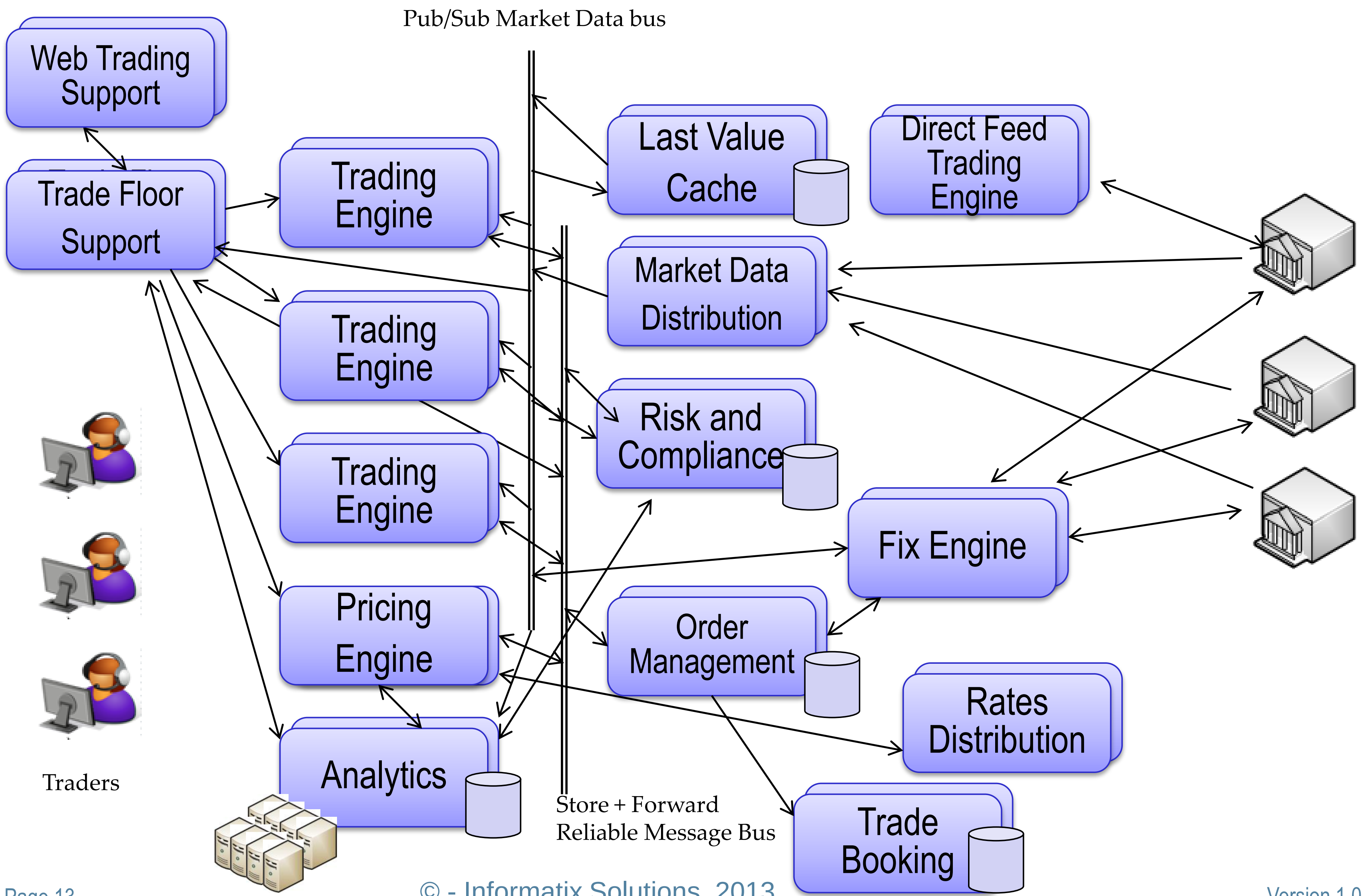
- Before Trade
  - Test strategies with Quant models, often Monte-Carlo simulation on Grids. Backtesting with historical market data.
  - Value at Risk (VAR) estimation, Mark to Market (prices)
  - Implied alpha
  - Long/Short balance
- During Trade
  - Client Reporting and performance attribution
  - Transparency
  - Simplicity
  - Absolute returns
- After Trade
  - Post trade analytics
  - P&L Stop Loss
  - Sensitivity limits
  - Impact and delay costs

Sell Side System - Venue



Tick to fill time now under 100µS at leading Markets

Buy Side System



Market Data Rates

- All bids, Trades and cancels are multicast from the venues
- Market Data Rates continue to grow
- Peaks are when you make (or lose) a lot of money so must keep up
- Volumes can vary x100 and peaks continuously increase

- **Top three by volume (as of Feb 2013), 1 second peak**
- SIAC OPRA 4,999,610 msg/s at 11:24 on 7th Nov 2012
- BATS Options MCASTPITCH AGG 1,843,665 at 15:59 on Jan 3rd 2013
- NYSE AMEX options 1,618,370 at 90:30 on Jan 11th 2013

Most venues are 10Gbps attach, NASDAQ now offering 40Ge

Plans by some venues to offer RDMA access

Banking Application Ecosystem

People
Process
Technology

Each Application has its own ecosystem consisting of:

People

- Developers, Test, Support, Project Mngt , Change Mngt Application vendors, Business Mngt

Process

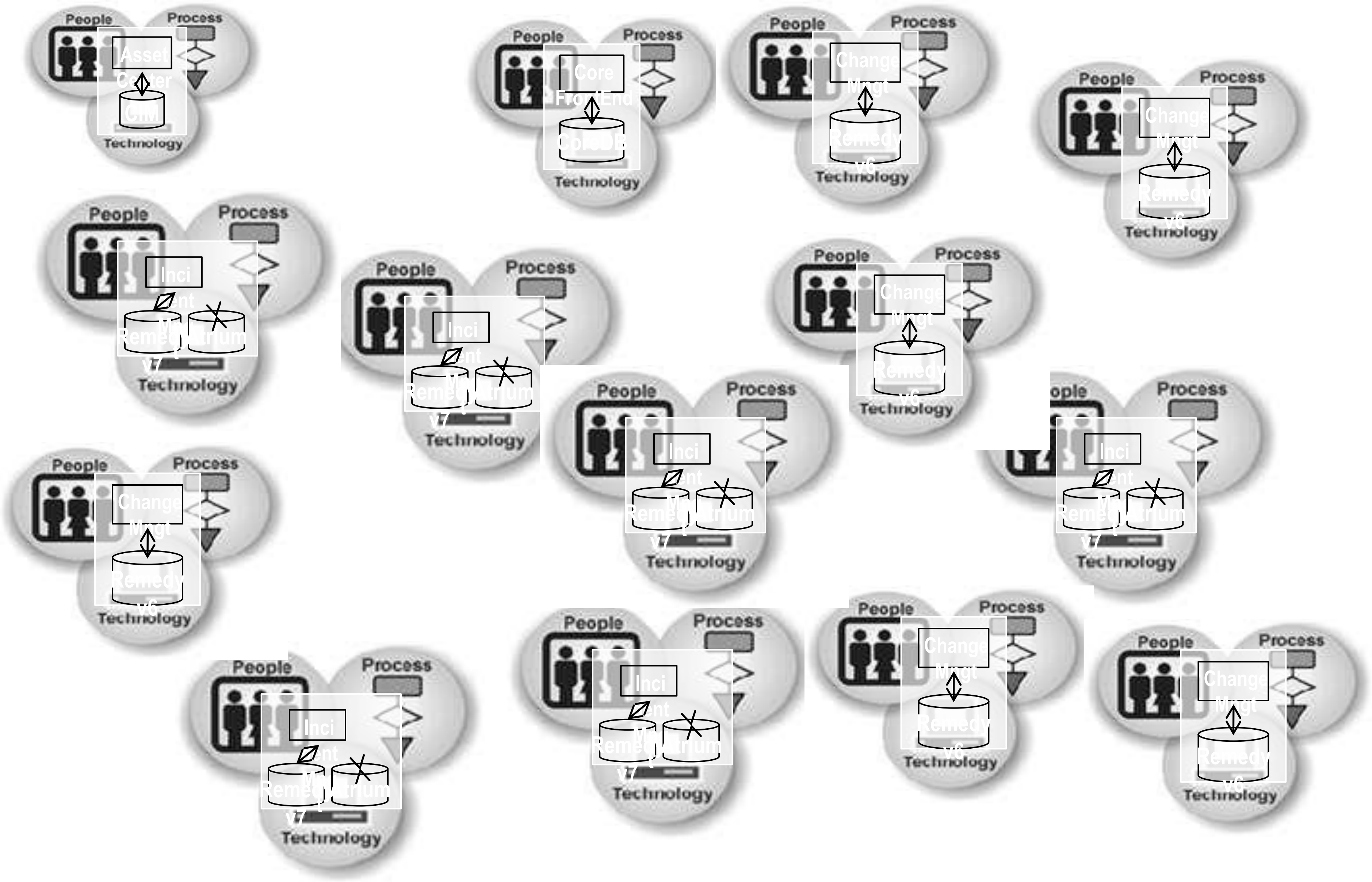
- Purchase, Change, Operate

Technology

- Prod, Test, DR

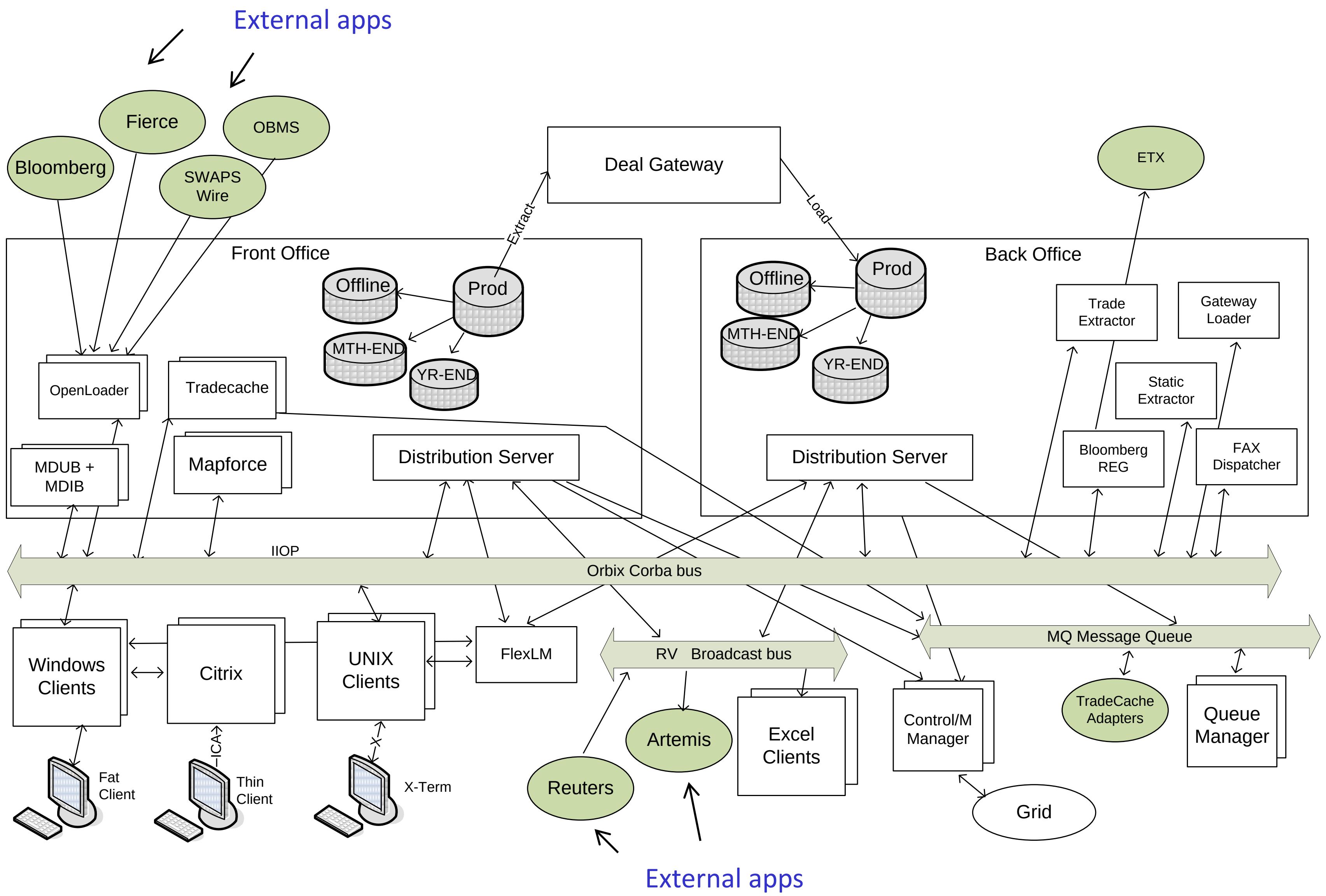
Hundreds of Application Ecosystems in every Bank

Informatix Solutions



Example Application Architecture

Informatix Solutions



Current best practice for Trading

- Proximity hosting
  - CoLocating with the venue to minimize propagation latency
- Low latency raw feeds
- Increasing NIC speed – 1/10/40G
  - Reducing serialization latency and improving capability to handle microbursts
- Monitoring and measurement improvements
  - Microburst detection
  - Packet capture and reporting
- Tuning to minimise latency and jitter
  - Measuring in nanoseconds, every microsecond make a difference
- Exotic technologies

Exotic Technologies used for trading systems

Informatix Solutions

- GPU – Used within Grid computing
- NPU – network processors, allow processing at the packet level e.g. Netronome, Tilera
- FPGA – Processing as the packet arrives
  - allowing simple trading algo's to fire immediately
  - Filter for specific symbols or book changing ticks
- FPGA - Compute
  - Hard coding algos, some Banks using for Risk or Pricing
- Kernel bypass technologies, e.g. Open Onload
- InfiniBand - high performance, low latency/jitter networking
- RDMA – Remote Direct Memory Access (InfiniBand and RoCE)
- Huge memory servers e.g. 4TB RAM to host entire DB's in memory
- Overclocked servers
- Lot's of appliances brought in from specialist vendors, featuring all of the above

CLOUD

Recovery Oriented Computing (ROC) Project - Berkeley/Stanford

- **Assume everything will fail**
  - S/W & H/W can fail at any time
  - Build redundancy at all levels in system
- **Scale out rather than up**
  - Typical Cloud server (2P, 16GB, 2x250GB): <$6k
  - Typical Enterprise server : ~$33k (many near $60k)
  - "Gold Plated" machines not much more reliable
    - S/W failures dominate H/W
    - More workload on a single system increases potential failure impact
    - Even good H/W still fails more frequently than typical SLAs allow
- **Reduce operations costs**
  - Inexpensive hardware slice
  - 1 to 2 orders of magnitude fewer systems engineers
  - Increase reliability through redundancy and less operator interaction
- **Goal: 24x7 availability with 8x5 operations**
  - "Lights out" operation is more reliable
  - Write system such that S/W quality problems are reported but don't show to customers. It should take many failures to miss SLA

http://roc.cs.berkeley.edu/

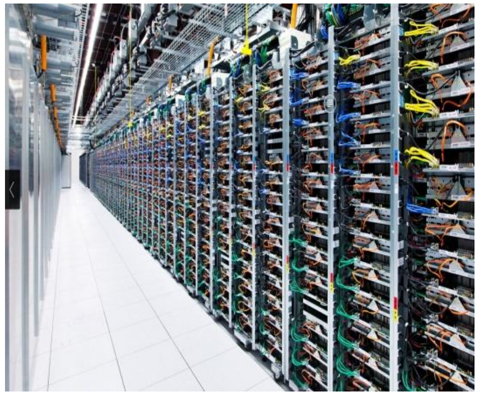
Three principles for Running Cloud Apps

Informatix Solutions

- **“Expect failures.** Your component might be stopped, or crash, at any time. Components you use might also be stopped at any time. The network will have problems. Disks will run out of space. Handle all failures gracefully.
- **Keep things simple.** Complexity breeds problems. Simple things are easier to get right. Avoid unnecessary dependencies. Installation should be simple. Failures on one box should have no impact on the rest of the data center.
- **Automate everything.** People make mistakes. People need sleep. People forget things. Automated processes are testable, fixable, and therefore ultimately reliable. Automate wherever possible”

Bill Hoffman – Hotmail , Operations

- Cloud Servers
- Bank Servers



Courtesy of Google

Cloud Storage example – Open Storage Pod 3.0

http://blog.backblaze.com/2013/02/20/180tb-of-good-vibrations-storage-pod-3-0/

180TB under $2000 parts cost using
Low cost commodity part
Resilience provided across Pods
Developed and used by Backblaze
(Cloud Backup Service Provider)

Design published online
Adopted by many others

Requires Application level resiliency
e.g. Amazon Dynamo and its decedents – Cassandra, Riak, S3 or other object storage systems

Approx 2-10% of the cost Banks typically pay for their storage

Banks and Cloud

- Banks reluctant to use Public Cloud offerings
  - Jurisdiction, liability and security issues caused by the regulatory environment they operate under
- Most Cloud providers have been targeting easier customers
  - Banks are extremely demanding and difficult customers
- Most Banks have sufficient size e.g. Typically 20-60K servers that there are fewer economies of scale achieved by going to Cloud
- Banks are experimenting with “caged” Cloud and building their own “private” Clouds
  - Some overflowing to the Cloud for some Grid computing

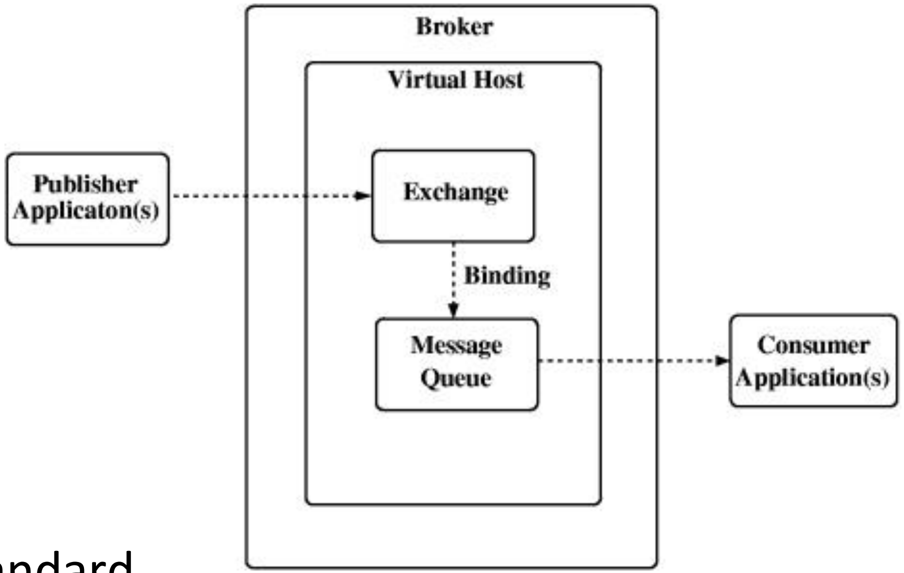
Technology Adoptions

- Both Cloud and Trading systems are pushing the technology envelope
- Both starting to adopt technologies first used elsewhere
- Whilst they started in different places, the problems faced by both are converging in some areas

Banks -> Cloud - Reliable Messaging

Informatix Solutions

- Messaging - Banks got together and defined AMQP as a reaction to buying expensive bespoke products
- Provides reliable asynchronous messaging, including Pub/Sub and topic based subscription



- Now an Oasis standard
- Several implementations including Apache Qpid and VMwares RabbitMQ
- Cloud deployments must also survive Internet disconnections
  - AMQP enables asynchronous messaging patterns
- RabbitMQ gaining large footprint in new Cloud designs

Banks -> Cloud – low latency and high performance

- Same FPGA cards already being used by some Cloud companies for Firewalls, Intrusion Detection and regulatory monitoring
- TCP kernel bypass such as OpenOnload delivering higher throughput and lower latency than the kernel stack
  - Preload library, requires Solarflare 10Ge cards
- Low latency tuning applied to Linux kernel pushed back into mainstream and providing performance benefits

  traditionally Linux TCP/IP behaviour has been optimized for the web server Use Case

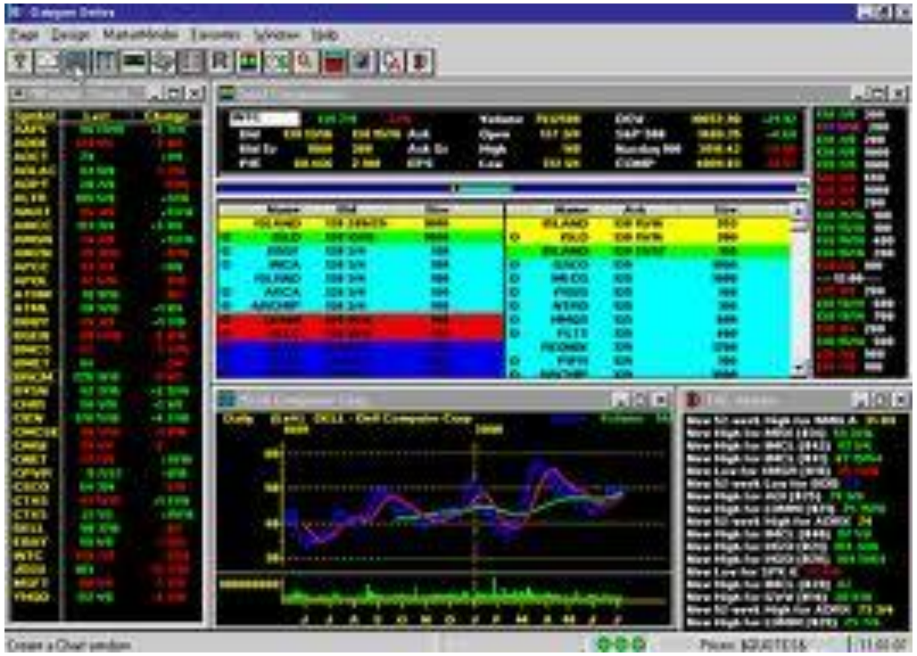
  Lots of low performant connections

  throttle to minimize memory footprint per connection

  Severely limits throughput of individual connections for high performance application to application connections

Banks -> Cloud

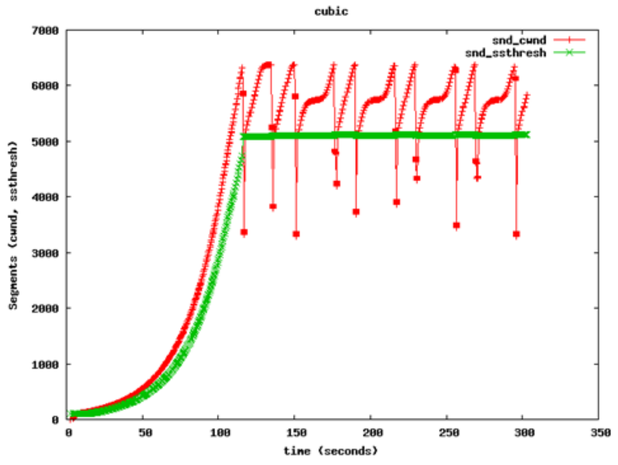
- Brokers developed Internet Push technologies to support Internet browser based stock trading
- Prices are pushed continuously to the Traders Screen



- HTML5 websockets now enable similar capabilities
- Enabling the next generation of web apps

Banks -> Cloud InfiniBand

- Most Trading venues and many Banks use InfiniBand today
- InfiniBand provides the lowest latency and the highest throughput of the non proprietary interconnects
- Reliable data transfers without TCP by using RDMA
  - Currently 56Gb/s
    - Measured 6GB/s server to server, single thread transfer using RDMA
    - Measured 2µS User Space -> User Space memory transfer between servers
  - TCP single thread throughput limited
  - InfiniBand costs same of less than 10G Ethernet
  - Long haul solutions available
  - Now also being adopted by High Performance storage devices



TCP Congestion Control limiting bandwidth

Cloud -> Banks noSQL

- noSQL
- Developed by the cloud since existing databases were no longer scaling or meeting the performance requirements
  - a single login could result in several SQL queries, some involving multiple joins
- Caching helps but adds complexity and cost
- Most large Cloud systems now designed using noSQL
- Banks are just starting as a way of handling the increased message rates and potentially lowering costs
  - Time series for market data
  - DevOps performance data

Cloud -> Banks Functional Languages

- Scalability, particularly concurrency is too hard with imperative programming languages
  - Servers supporting 1000 concurrent threads are already available.
  - Trend will increase with Moore's law doubling this every 18 months
- Functional languages have a simpler concurrency model
  - Erlang is widest deployed
  - Haskell, OCaml in some Hedge Funds
  - Erlang is core language used in RabbitMQ, Riak, CouchDB
- Functional Languages are a better match to create code which can run efficiently on FPGA's
  - Converts to VHDL more efficiently than imperative languages
  - Exploits native parallelism better
  - e.g. NovaSparks develop their FPGA based Feed handler in Lisp

Cloud -> Banks Functional Languages - Erlang

- Benefits of Erlang
  - Erlang OTP provides comprehensive runtime support including Event Managers, Watchdogs, FSM, in-memory DB, Distributed DB, HA, Unit test, Docs, Live Update
  - Big **Int** – no need to deal with overflows
  - Built in support for HTML and SNMP
  - Powerful bit level processing – useful for protocol encode/decode
  - Code is more powerful – achieve more in fewer lines
  - Vast ecosystem of Erlang based communications handlers including FIX engines (www.ieiss.com), PCAP processing
- Challenges with Erlang
  - Relies on automated restart – fails early e.g. OOM (Out of Memory)
  - Native String handling inefficient
  - Overhead of typed data reduces e.g. **Int**'s
  - Virtual Machine, GC overheads and jitter
- Great for devOps
- For higher performance use Erlang as a Control Plane to achieve concurrency then drop down into C/C++ code for performance critical
  - Code as Ports, or Native Functions; Allows these to be single threaded

- Questions?
- Feedback
- Qcon speaker evaluation

- Richard.croucher@informatix-sol.com
- www.informatix-sol.com