

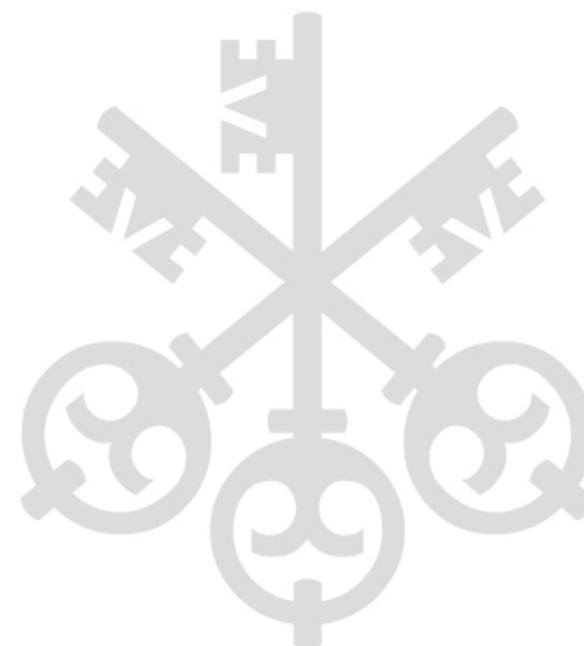
Futures Trade Flow

Ian Bond

Exchange Trade Derivatives

Prime Services IT

9 March 2011



Introductions

- Who is Ian Bond ?
 - Last 10 years working on client facing reporting & connectivity software
 - Part developer, part architect, part dev manager
 - Work here was done as a team.

- Who is UBS ?
 - Client focused financial services firm of 65,000 people.
 - Wealth Management
 - Investment Banking
 - Asset Management
 - Retail Bank in Switzerland

Agenda

Building a new front to back trade flow architecture

- Background, Futures and Options
- Where we started from
- Project drivers, business and IT
- Overall approach
- Understanding the domain, getting the words right
- Wiring it together
- Managing the workflow, Event driven architecture
- The challenge of OTC regulatory change

Section 1

Background

What is a future ?



Financial futures

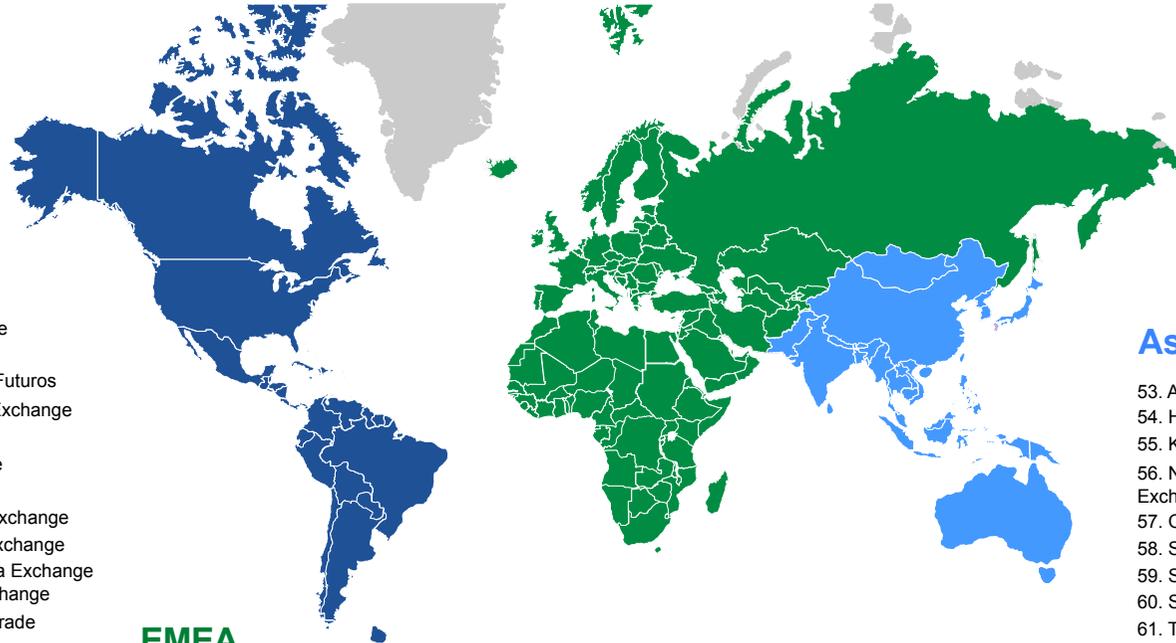
- A futures contract
 - is traded on an exchange
 - is based on the price of a commodity, stock, stock index, bond, or currency, referred to as the "underlying"
- Specifies a
 - Price, Time, Amount
- For
 - the delivery of the underlying
 - or settlement in cash
- Option - right, but not the obligation
- Examples
 - S&P500 Index Future
 - Heating Degree Day Futures
 - Index goes up as the temperature goes down, interest to businesses that supply heating oil for example.

Futures & Options

Chicago Board of Trade, 1993



Exchanges



The Americas

1. AMEX American Stock Exchange
2. BOX Boston Options Exchange
3. BM&F Bolsa de Mercadorias & Futuros
4. CBOE Chicago Board Options Exchange
5. CBOT Chicago Board of Trade
6. CCE Chicago Climate Exchange
7. CFE CBOE Futures Exchange
8. CME/IMM Chicago Mercantile Exchange
9. COMEX New York Mercantile Exchange
10. CSCE Coffee, Sugar and Cocoa Exchange
11. ISE International Securities Exchange
12. KCBOT Kansas City Board of Trade
13. MGE Minneapolis Grain Exchange
14. NQLX Nasdaq Liffe Markets, LLC
15. NYBOT New York Board of Trade
16. NYMEX New York Mercantile Exchange
17. ONEC OneChicago
18. PBOT Philadelphia Board of Trade
19. PHLX Philadelphia Stock Exchange
20. PSE Pacific Stock Exchange
21. WCE Winnipeg Commodities Exchange

22. ME Montreal Exchange
23. MEXDER Mexican Derivatives Exchange

EMEA

24. AEX Euronext (Amsterdam)
25. Bclear Euronext.Liffe-Wholesale Equity Derivatives
26. BELFOX Euronext (Brussels)
27. DME Dubai Mercantile Exchange
28. ECX European Climate Exchange
29. EDX Equity Derivatives Exchange (London)
30. EEX European Energy Exchange
31. ENDEX Netherland Energy Exchange
32. EUREX European Exchanges
33. ICE Intercontinental Exchange
34. IDEM Italian Derivatives Market
35. LCE Euronext.liffe (Commodities)
36. LCH – LCH Freight
37. LIFFE Euronext.liffe
38. LME London Metal Exchange
39. MATIF Euronext (Paris)
40. MEFF MEFF

41. MONEP Euronext.liffe Paris (MONEP)
42. Nordpool Nord Pool ASA
43. OMX The Nordic Exchange (Stockholm / Denmark)
44. PWN Powernext SA
45. SAFEX South African Futures Exchange

46. ADEX Athens Derivatives Exchange
47. BDP Euronext (Lisbon)
48. BSE Budapest Stock Exchange
49. OTOB Wiener Boerse AG (Vienna)
50. TASE Tel Aviv Stock Exchange
51. TurkDEX Turkish Derivatives Exchange
52. WSE Warsaw Stock Exchange

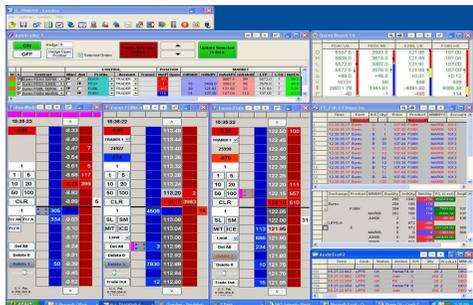
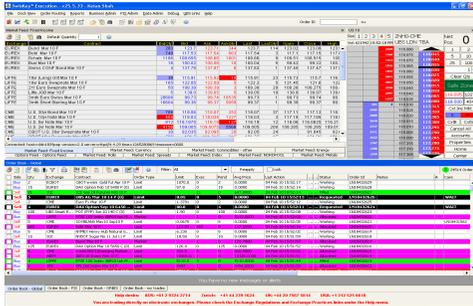
Asia/Pacific

53. ASX Australian Stock Exchange
54. HKFE Hong Kong Exchanges & Clearing Ltd
55. KRX Korea Exchange
56. NZFOE New Zealand Futures & Options Exchange
57. OSE Osaka Securities Exchange
58. SEHK Hong Kong Exchanges & Clearing Ltd
59. SFE Sydney Futures Exchange
60. SGX Singapore Exchange Derivatives Trading
61. TAIFEX Taiwan Futures Exchange
62. TFE Thailand Futures Exchange Plc
63. TFX Tokyo Financial Exchange (formerly TIFFE)
64. TSE Tokyo Stock Exchange
65. ISE Indian Stock Exchange

65. CJCE Central Japan Commodity Exchange
66. KANEX Kansai Commodity Exchange
67. MDEX Bursa Malaysia Derivatives
68. TGE Tokyo Grain Exchange
69. TOCOM Tokyo Commodity Exchange
70. JADE Joint Asian Derivatives Exchange (JADE)

Direct Memberships	Agent Broker Relationships
55	15

Trading



- The vast majority of Futures and Options trading today is done via Direct Market Access (DMA) tools.
- Most markets no longer have Pit trading, even on the CME in 2008 82% of volume¹ was done via Globex its electronic platform.
- There are a large number of diverse trading tools across the industry providing access to exchanges, including both vendor platforms and inhouse built systems from Investment Banks.

Background

Conclusion

- Futures and Options are standardised contracts traded on exchanges
- Large number of exchanges
- Diversity in trading platforms
- Lower volume than cash equities, but much higher than OTC products
- Exchanges mitigate credit exposure
- Marked to market each day. Participants must lodge funds with an exchange (in a margin account) and profit/loss is booked to that account.
- To support this process records of trades done flow from trading systems into settlement systems. The settlement systems calculate the margin payable each day.

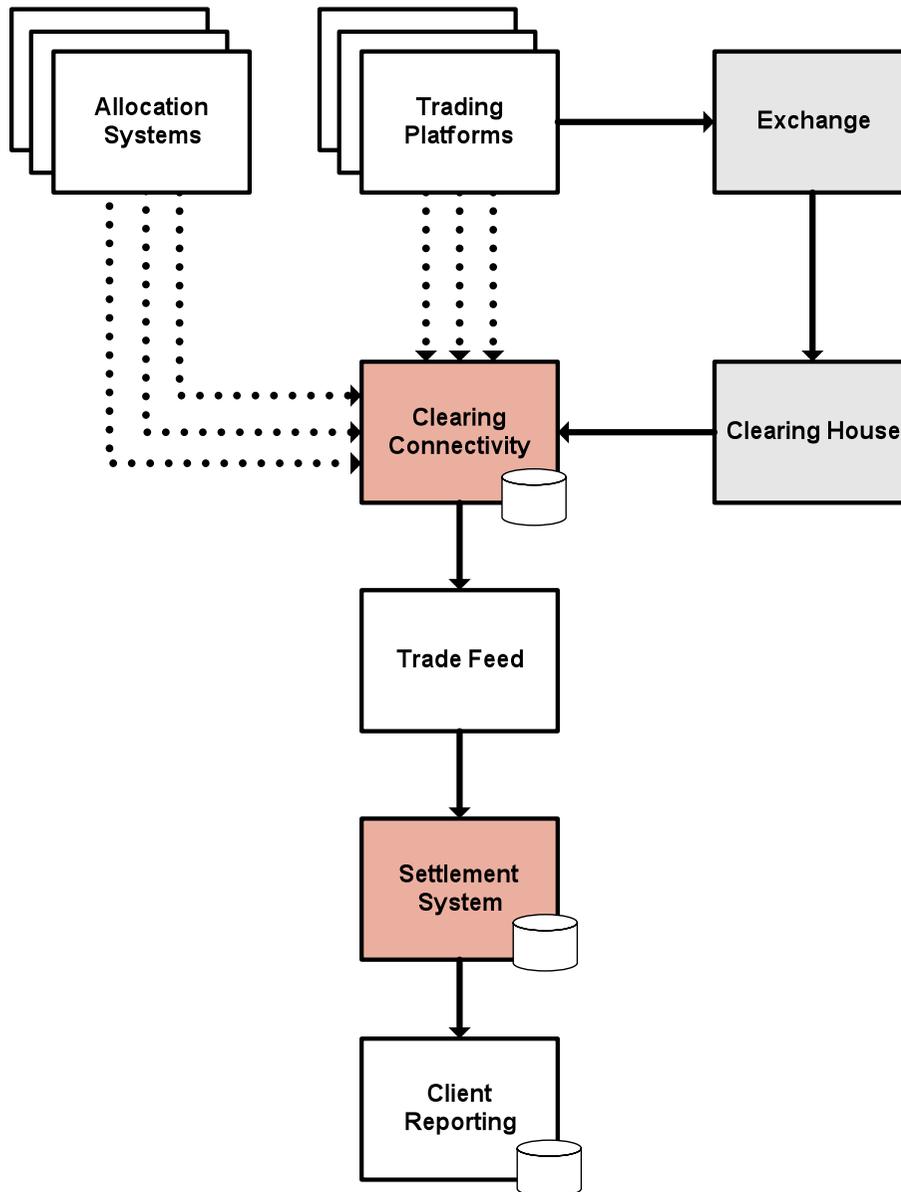
Section 2

Where we started from

Where we started

- High number of trade flow changes required
 - New exchanges, new products, new regulations
- Change overhead high
- The challenge of large vendor systems. One written in RPG
 - Inflexibility

High Level Trade Flow



- High number of bespoke batch (PERL, Java & RPG) “trade” feeds directly into vendor supplied clearing database.
- Matching of cleared trades with internal systems frequently did not maintain information.
- Required additional information passed in vendors “free” fields.
- Multiple trade repositories
- Monolithic batch-based feed component between clearing and settlement with high technical debt incurring v. slow change velocity.

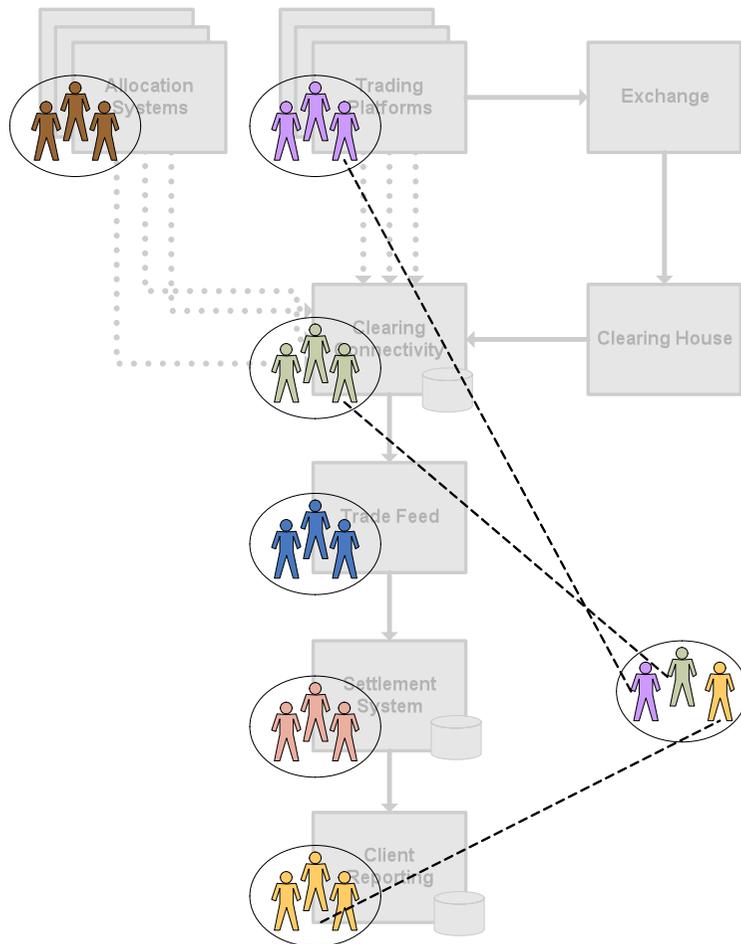
Project Drivers

- Business wanted the ability to recognise how a trade is done, through which trading platform, which desk, user etc
- Ability to enrich a trade, with calculated information
- Lower the cost of change or rather make the cost of change *proportionate* to the scale of change.
 - Reduce the complexity in the environment.
 - Better abstraction from exchange specifics.
- Business case for building a new trade flow architecture was based on combining a high priority business requirement (almost impossible to do with current architecture) along with selling the longer term benefits of putting in place the architectural building blocks required for both the immediate requirement and longer term strategy.
- Latency was less of a driver versus trading systems, but transactional throughput scalability was more important. 100s of thousand trades a day.

Approach

- Feature team
- Language
 - Understanding the domain and business processes
- Appropriate Architecture “Agile versus Architecture”
- Simplifying how components are developed
- Event-based system
 - when a change happens allow immediate relevant processing to follow
 - make trade flow extension points simple to add
- Wrapping vendor systems

Feature Team



- Most front-to-back trade flow changes had to go through multiple component teams, sometimes up to 5.
- Handoff, testing & release scheduling, limited cross component learning. Lean Thinking.
- Conways Law. Software tends to mirror the structure of the organization that built it.
- Craig Larman & Bas Vodde, Scaling Lean and Agile Development
- Mary & Tom Poppendieck, Lean Software Development
- John Seddon, Systems Thinking
- First decision – form a new team with members from the old component teams.
- Scrum – low tech whiteboard stories & tasks, spreadsheet for backlog

Language

Trade

Black line

Blue line

Intra-Day Trade

Language

“Getting the words right”

Language

Order

Execution

Allocation Instruction

Cleared Trade

Booking Instruction

Transaction

Language

“Commit the team to exercising that language relentlessly in all communication within the team and in the code. Use the same language in diagrams, writing and especially speech”

Eric Evans

Section 3

Wiring it together

Message Formats

Decisions

- Message Format
 - FIX
 - FpML
 - Swift

- Message Bus Architecture
 - Standard topics
 - Standard approach

FIX

Financial Information eXchange

- FIX protocol was first developed for equity trading between Fidelity and Salomon Brothers in 1992.
- It's goal was to simplify connectivity between buy-side and sell-side firms involved in equities trading. Its now widely adopted there.
- Tag=Value message format
- It also defines a TCP/IP based session protocol
- Message formats support business processes – we learnt a lot from FIX in this respect.

FIX

FIX Execution Report

8=FIX.

4.2 9=321 35=7 49=UBSABC 56=UBS 34=1157 50=123456 52=20101024-07:54:07 30=
96.135 60=20101024-03:54:07 150=1 1=ABCDEF 31=96.1350 151=78 32=2 6=96.135 3
7=9187497239473291234 38=100 9998=132963 39=1 40=2 100=LIFFE 14=23 75=2010
1024 76=4351 17=20101024000354060001 167=FUT 48=ERH10
229=20101024 20=3 200=20070319 22=1 54=1 55=ERH10 9506=Comdty 9896=11 10
=123

FIX

FIXML Execution Report

```

<ExecRpt OrdID="123456" OrdID2="CMEORD123" ExecID2="CMEEEXEC54321" ID="CLIORD_ABC1"
  ExecID="123456_1_1" ExecTyp="F" Stat="1" Acct="ABCDEF-2Z009" AcctIDSrc="100"
  Side="1" CustCpcty="4" LastMkt="XCME" LeavesQty="80" CumQty="20" TrdDt="2010-05-18"
  TxnTm="2010-05-18T15:08:00-00:00" MLegRptTyp="1">
  <Hdr SSub="UBS:ETD:KES"/>
  <Pty R="1" Src="C" ID="UBS"/>
  <Pty R="4" Src="C" ID="BCD">
    <Sub Typ="10" ID="G Up Ref 123"/>
  </Pty>
  <Pty R="73" Src="g" ID="XCME"/>
  <Pty R="76" Src="d" ID="PPM"/>
  <Pty R="10001" ID="2"/>
  <Instrmt Sym="ESU0" ID="90464079" Src="a" CFI="FFICS-" MMY="201009"
  MatDt="2010-09-17" Exch="XCME" Desc="S&P500 EMINI FUT Sep10">
    <AID AltIDSrc="b" AltID="ES"/>
  </Instrmt>
  <OrdQty Qty="100"/>
</ExecRpt>

```

FIX

FIXML Allocation Instruction

```

<AllocInstrctn ID="58874451" TransTyp="0" Typ="2" ID2="pc_t63" NoOrdsTyp="0" Side="1"
  Qty="1" LastMkt="ICEU" TrdDt="2010-03-15" TxnTm="2010-03-23T08:23:00-00:00"
  MLegRptTyp="1">
  <Instrmt Sym="COJ1P" ID="1234567" Src="a" CFI="OPAFPS" MMY="201012"
    StrkPx="50" Exch="ICEU" Desc="CRUDE OIL OPT IPE Jan11C 50">
    <AID AltIDSrc="b" AltID="BC"/>
  </Instrmt>

  <Alloc Acct="ABCDEF-2E001" ActIDSrc="100" Px="163" Qty="1"
    IndAllocID="2374682368687" IndAllocID2="abc63" AllocPosEfct="0">
    <Pty R="1" Src="C" ID="UBS"/>
    <Pty R="3" Src="a" ID="Clnt Shrtm"/>
    <Pty R="16" Src="b" ID="ABC"/>
    <Pty R="59" Src="c" ID="Algo"/>
  </Alloc>
</AllocInstrctn>

```

FIX

Why did we choose FIXML ?

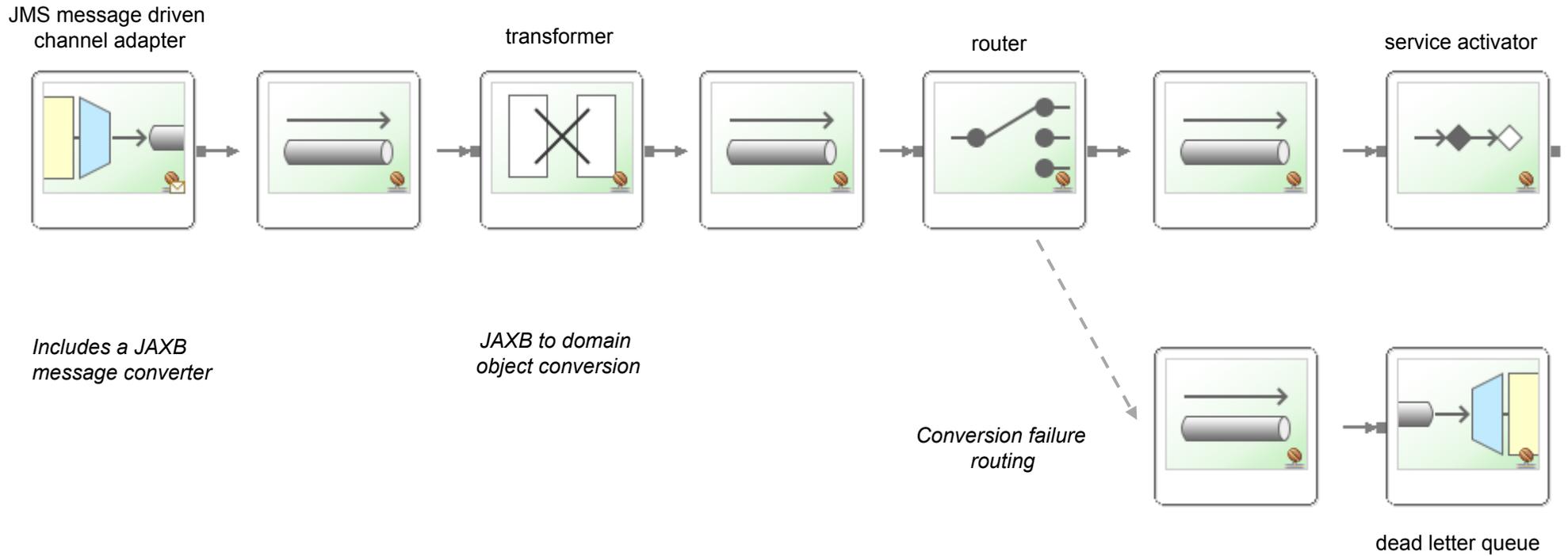
- The message formats support the business process lifecycle e.g. execution, allocation, cleared/trade capture.
- Adopting a standard brings with it supporting tools i.e FIXimate, documentation and emotional buy in from other parties.
- Despite being in an unsophisticated use of XML, it is XML.
- FIXML as XML was easier to encourage other teams to adopt
 - Tooling for parsing more mainstream

Messaging

Requirements and overview

- How do we not end up with another type of mess, a messaging mess ?
- Spring Integration
 - to have a simple declarative base for our gateway components, encouraging loose coupling.
 - Uses the language of *Enterprise Integration Patterns* Book - Gregor Hohpe and Bobby Woolf
- Pipes and Filters
- Message Channels
- Transformation
- Routing
- End Points

Spring Integration



Spring Integration

Inbound JMS message

- Message Driven Channel Adapter
 - The starting point receiving messages from a queue or topic and putting them on a channel.
 - Has a message-converter attribute that we use for JAXB unmarshalling

```
<jms:message-driven-channel-adapter auto-startup="false"  
    container="allocationsTopicMessageListenerContainer"  
    channel="allocations-gateway.from-allocations-topic"  
    message-converter="jms-converters.fixml-jaxb-message-converter"/>
```

Spring Integration

Transformation

- Transformer
 - Converts the previously unmarshalled JAXB FIXML object to our true domain object
 - Conversion code is discretely testable as a standalone component

```
<si:transformer id="allocInstructionTransformer"
ref="allocationInstructionTransformerBean"
    input-channel="allocations-gateway.from-allocations-topic"
    method="convertFIXAllocationInstructionToDomain"
    output-channel="allocations-gateway.toRouter"/>

<bean id="allocationInstructionTransformerBean"
class="com.ubs.etd.tma.gateway.allocation.AllocationInstructionTransformer" />
```

Spring Integration

Routing

- Router
 - Payload type routing
 - Use to redirect conversion failures to a dead letter queue

```
<si:payload-type-router input-channel="allocations-gateway.toRouter"
    default-output-channel="allocations-gateway.toPersistor">
    <mapping type="com.ubs.etd.utils...DeadMessage" channel="dealLetterChannel"/>
</si:payload-type-router>
```

Spring Integration

End Point

- Service Activator
 - The end point to connect a Spring-managed bean to an input channel
 - Used to pass the converted domain object to a persistence service.

```
<si:service-activator input-channel="allocations-gateway.toPersistor"
    ref="alloc-gateway.persistor"
    method="receive"/>

<bean id="alloc-gateway.persistor" class="com.ubs.etd...AllocationInstructionPersistor">
    <property name="nimbusDao" ref="tma.nimbus-dao"/>
</bean>
```

Section 4

Managing the workflow

Goals

- Single responsibility
 - Each processing unit should be responsible for doing just one thing
- Easily testable
 - Helped by the above, but each unit in the system should be easily testable
- Open for extension
 - It should be obvious when a new requirement for processing along the trade flow comes how to hook it in.
- Constraints
 - Keep 2 days of trade data “online”
 - The maximum time a trade might be relevantly looked up
 - 300,000 cleared trades + Orders, Executions, Allocations per trading system. Transactions also back from the settlement systems.
 - Feeding a GUI blotter

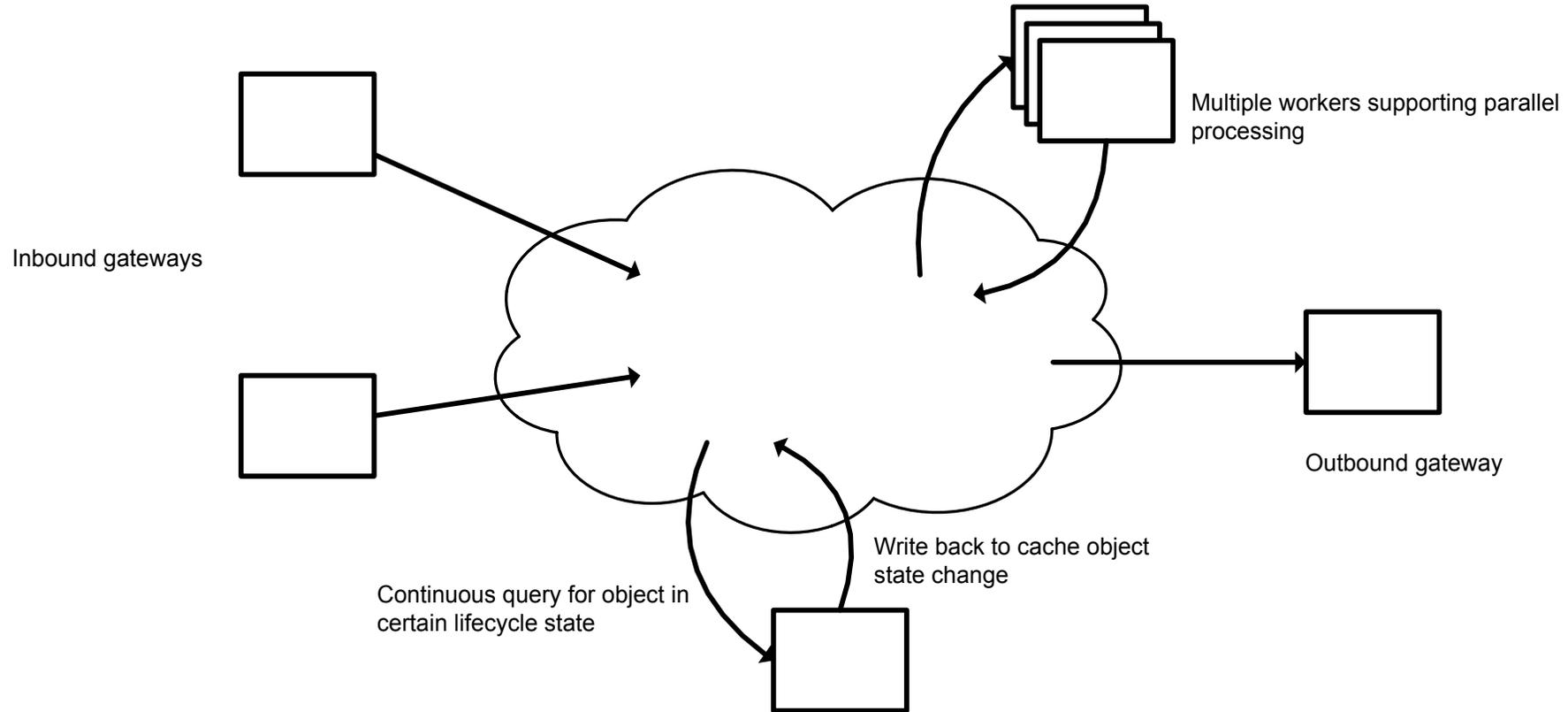
Space Based Architecture

Distributed cache and event processing

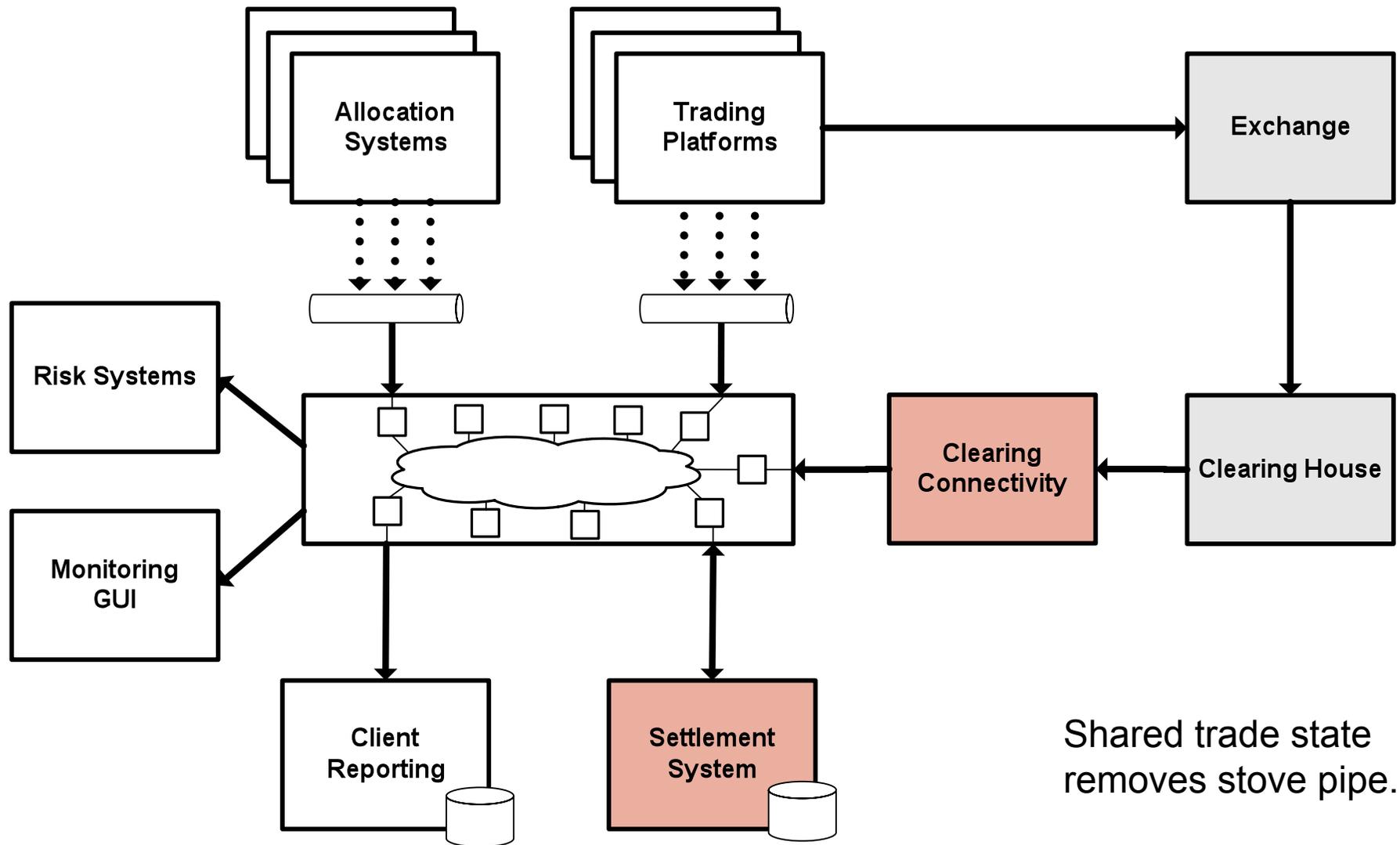
- We use a proprietary in house space based architecture component, essentially a event based distributed cache.
 - Data Grid – a distributed cache
 - Processing Grid – event processing components based on master/worker pattern (aka blackboard architecture)
 - A bit like the old JavaSpaces.
 - Gigaspaces
- We've also briefly explored Coherence...
 - `MapListener / ContinuousQueryCache` – to get the event
 - `Entry Processors (AbstractProcessor)` – to save the result atomically
 - Less transactional support out of the box
 - Alternatively could use the Command Pattern

Event Based

Master Worker Pattern



Blackboard



Shared trade state
removes stove pipe.

Section 5

The challenge of OTC regulatory change

OTC Derivatives

- September 15, 2008, Lehman Brothers filed for Chapter 11 bankruptcy protection.
- The failure of Lehman Brothers highlighted that positions and exposures of firms in OTC derivative markets were not sufficiently transparent to other market participants or to regulators. This led to an unwillingness to trade, and hence a lack of liquidity.¹



Swaps

New products, new flow

- In response the Dodd–Frank Wall Street Reform and Consumer Protection Act becomes effective 15th July 2011
- Credit Default Swaps (CDS)
 - Protection on an instrument from a credit event (for example bankruptcy)
- Interest Rate Swaps (IRS)
 - Exchange of interest payments, most commonly one party paying a fixed rate and the other a floating rate.



"Unless derivatives contracts are collateralized or guaranteed, their ultimate value also depends on the creditworthiness of the counterparties to them. In the meantime, though, before a contract is settled, the counterparties record profits and losses—often huge in amount—in their current earnings statements without so much as a penny changing hands. The range of derivatives contracts is limited only by the imagination of man (or sometimes, so it seems, madmen)."

Warren Buffet 2002

Issues

- Message formats (FIXML) didn't fully support IRS and CDS trades
- Integration now needed with systems outside the immediate area
- Unplanned when we started the work, architecture needed to respond

FpML

Partial example of IRS swap block

```

<swap>
  <productId productIdScheme="http://...product_id">USD3L1</productId>
  <swapStream id="fixedLeg">
    <payerPartyReference href="clearing_service"/>
    <receiverPartyReference href="clearing_firm"/>
    <receiverAccountReference href="account1"/>
    <calculationPeriodDates id="fixedCalcPeriodDates">
      <effectiveDate>
        <unadjustedDate>2010-10-20</unadjustedDate>
        <dateAdjustments>
          <businessDayConvention>NONE</businessDayConvention>
        </dateAdjustments>
      </effectiveDate>
      <terminationDate>
        <unadjustedDate>2011-10-20</unadjustedDate>
        <dateAdjustments>
          <businessDayConvention>MODFOLLOWING</businessDayConvention>
          <businessCentersReference href="fixedPrimaryBusinessCenters"/>
        </dateAdjustments>
      </terminationDate>
      ...
    <calculationPeriodFrequency>
      <periodMultiplier>6</periodMultiplier>
      <period>M</period>
      <rollConvention>20</rollConvention>
    </calculationPeriodFrequency>
  </calculationPeriodDates>

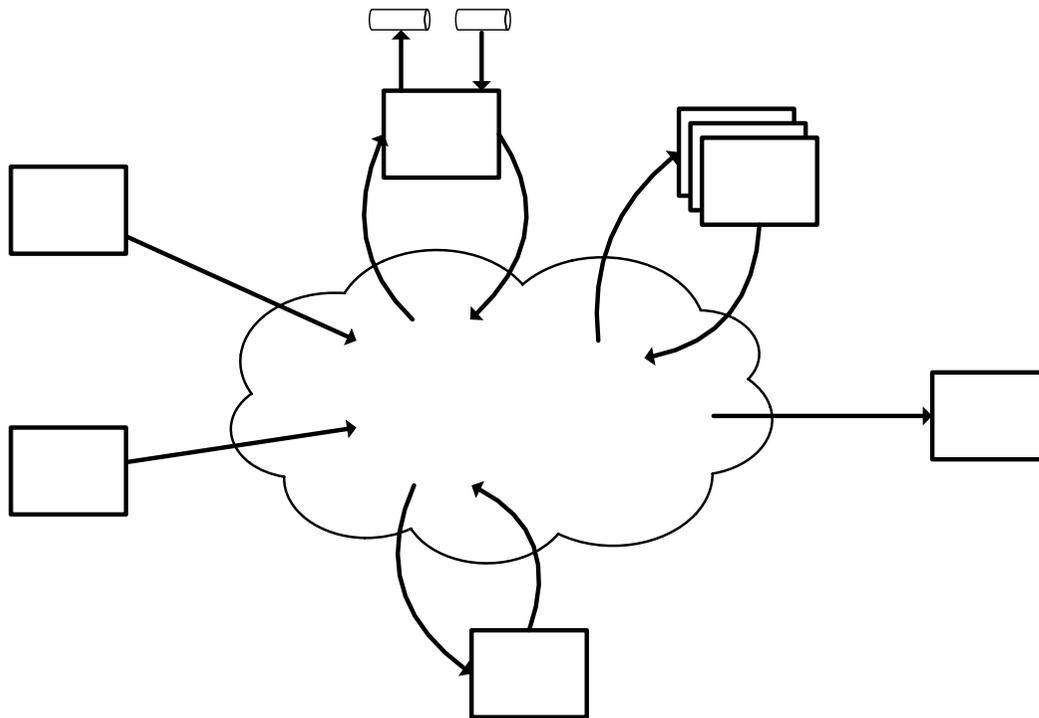
```

FIXML - FpML

Comparison

- FIXML focused on standardised products, FpML on complex customised products.
- FpML focused on product description, FIXML more on business process
- As FpML is focused on the problem of describing bespoke products in a structured way it uses XML structures far more than the simpler FIXML.
- Swift ISO20022 convergence standard.

Cross function integration



- Event based listeners provide fine grained extension points.
- Given we had less control over the outside systems we decoupled via outbound and inbound queues.
- What to cache in the cloud app versus not ?
- We didn't convert the whole message to a domain object rather keep non relevant aspects still in message form
- To save space we only updated lifecycle objects.

What's next

- Spring Outbound Channel Adapters for proprietary cache
 - Save to the cache then would not need the service activator called persistence service
 - Or using the Spring Cache Abstraction in 3.1
- Dynamic transformations
 - Writing a custom DSL combined with Spring Expression Language vs Drools
- Feeding cross product blotters
 - Streaming out to an LCDS / Flex
- Akka framework seems to fit what we do on the event processing side
- The trend is for products to become increasingly more standardised and commoditised, more products will be exchange cleared in response to OTC regulation.

Summary

Lessons Learned

- “Getting the words right”
 - understanding the business process was important. Once we got that right a lot could follow.
- Standardising the messaging on an external standard
 - Got better cross team buy in, but encountering a different product area presented challenges
- Appropriate architecture – event based
 - But could do more to standardise
 - Investing time in the lifecycle model
 - I don’t think I will ever build a transactional system with a relational database
- Simplify development
 - Coupling, testability
- Feature team
 - The concept was slow to get traction elsewhere in immediate department, but large benefit in terms of delivery speed.

Questions & Contact

Ian Bond

ian.bond@ubs.com

Questions ?