Infoq

QCon 北京

Beijing
18th April 2011

早上好

欢迎来到QCon北京2012大会

我很希望用普通话演讲

但是很抱歉我只会说英语

John Davies

- An ageing "Über-geek"
  - Hardware, Assembler, C, Objective-C, C++, OCCAM, SmallTalk, Java
  - Worked mostly in trading systems, FX & Derivatives
  - Head of trading systems at Paribas, head of architecture at BNP Paribas, global head of architecture at JP Morgan
  - Author of Learning Trees Enterprise Java courses & co-author of several Java & architecture books
- Co-founder of C24 Solution in 2000
  - Sold to Nasdaq's Iona Technologies in 2007, Iona sold to Progress Software in 2008, Technical Director of both companies
- Co-founded Incept5 in 2008, re-acquired C24 from Progress in April 2011
  - CTO of Incept5 & C24
  - Original technical architect behind Visa's V.me (pre-public release)

Incept5

What we're going to look at today...

- SQL or NoSQL - Good or Bad?
  - If Java better than SQL?
  - What can't SQL do?
- Today's need to scale
  - Too much data to store on spinning disks of iron
  - Too many calculations
- Compute Grids
  - Distributed computing, grids, private and public cloud
- Look at a few grid technologies

Incept5

SQL is just a language

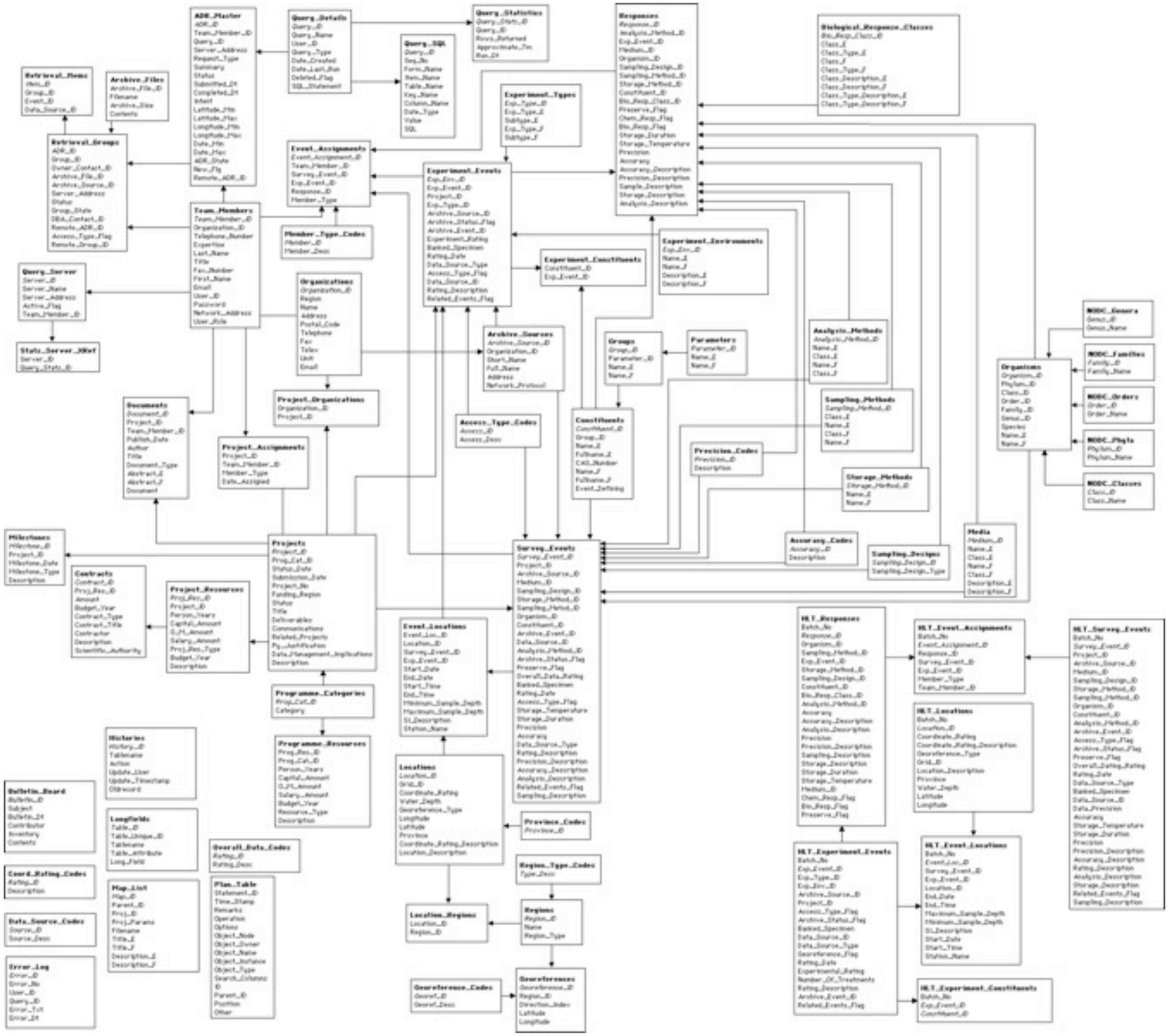
- SQL is a language that allows you to manipulate relational databases - RDBs

- If the data you're storing is relational then SQL is a pretty good fit

- If fact, for dealing with lists (as tables) it's a great language, dynamic and relatively fast
  - Sure it has a few problems but give me a language that doesn't

Incept5

How complex can it be?



NoSQL - No What?

- Did we really need a name for it?
  - We'd worked for years without needing a name for it

- Later No SQL became Not Only SQL
  - The vendors chickened out :-)
  - NoSQL vendors support NoSQL because they can't support SQL

- NOJ - No Java
- NOSS - No Shell Scripts
- NOW! - No Windows!

Incept5

SQL works

- SQL might be the wrong language for hierarchical data but it's superb for tables

- Imagine the telephone directory - in a single table
  - Tell me how many "Smith's"
  - select count(*) from directory where surname="Smith"

- Now in XPath (for example)
  - count(/directory[surname='Smith'])

Incept5

SQL is useful - when things are flat

- Easy, so now I'd like to see the 10 most popular names and order them by popularity...
  - Select surname,count(*) from directory group by surname order by count(*) desc limit 10

- If you had the list as a CSV then you could do the same with a little shell magic - assuming you're on a decent OS of course
  - cut -d "," -f 1 | sort | uniq -c | sort -r | head -10

- Both will execute, even on a million rows in around a second

Java instead of SQL?

- A few years back we needed a cache for 1TB of data
  - Coherence was perfect so I called Oracle for a price
  - It would have been over $500k!

- So we used MySQL distributed over 20 machines with a query aggregator
  - Data was stored in a columnar format with everything indexed in memory

- It worked incredibly well but...

- Joins and aggregate functions took another year to write
  - avg, max, min, count etc.

Incept5

A question for you...

- I have 1 million rows of CSV (Comma-Separated-Values), each row roughly 1k in size and some 50 columns
  - So 1GB of data, 50 million fields

- I now want to count the number of rows where the 7th column contains the word "Hello"

- How long SHOULD it take to return the answer?
  - A - Around 30 seconds to a minute
  - B - Around 20-30 seconds
  - C - Around 10-15 seconds
  - D - Sub 10 seconds

Incept5

Java NIO

- Using ~~Sun's~~ Oracle's NIO Grep example it threw OOM with -Xms4G -Xmx4G
  - The 100MB version took 1.2 seconds so let's estimate 1GB will take 12 seconds as it's a linear search

- The following however took under a second...

```
grep -c Hello BigFile.csv | cut -d "," -f 7
```

- Java is not the "silver bullet" (yet)

Incept5

ORM - OMG!

- Probably the biggest waste of programming time, lines of code and source of bugs and latency is ORM

- The effort of trying to convert something inherently hierarchical into something relational

- It's probably the main source of complaints about SQL
  - People start with a nice XML document, map it with ORM into a RDBMS and then end up writing custom SQL because the ORM layer is too slow

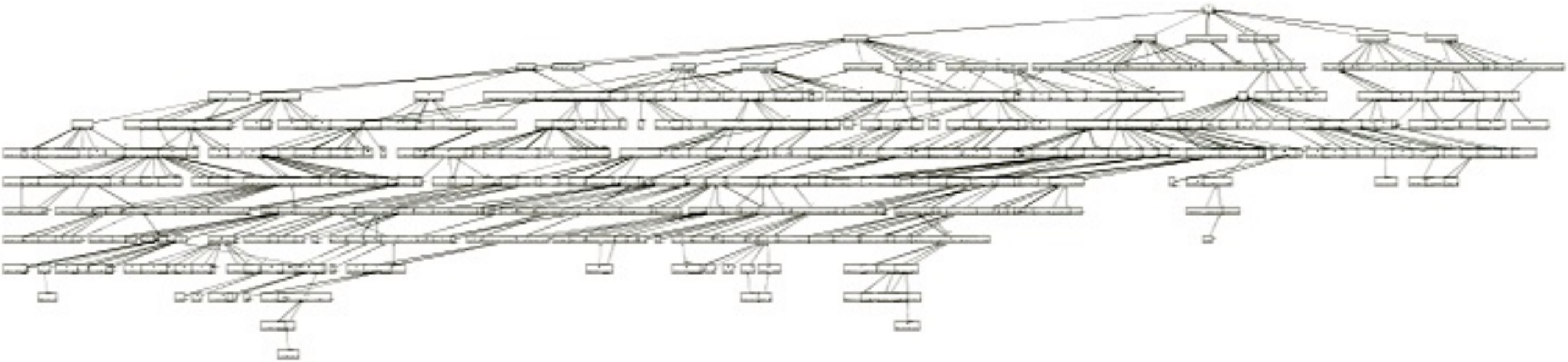
- GOOD LUCK!

Incept5

FpML into a classic database?

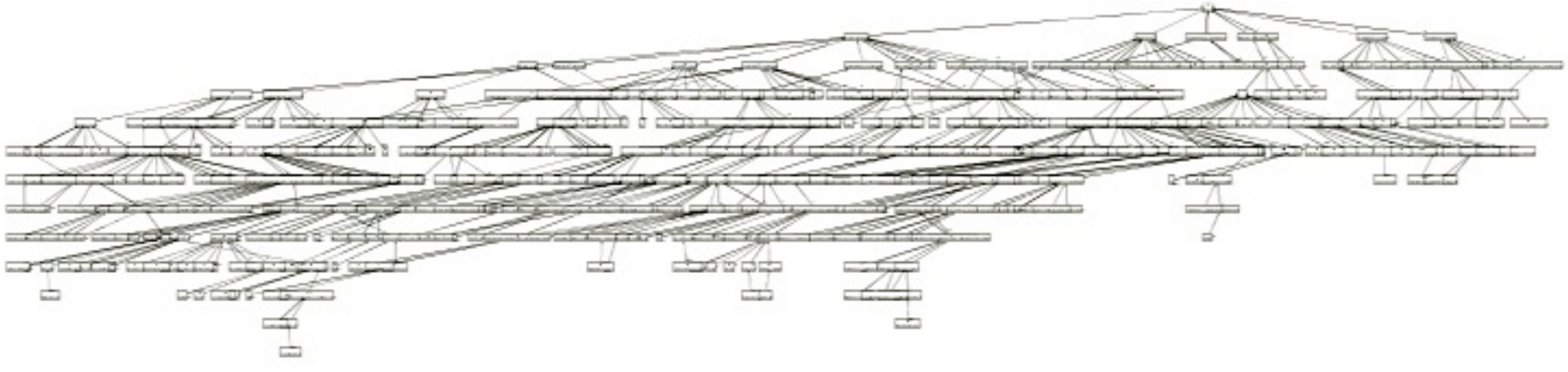
- The fuzzy patch below is the complete model of and FpML Swap from the IRD (Interest Rate Derivative) schema

- It's one of several dozen financial models in FpML

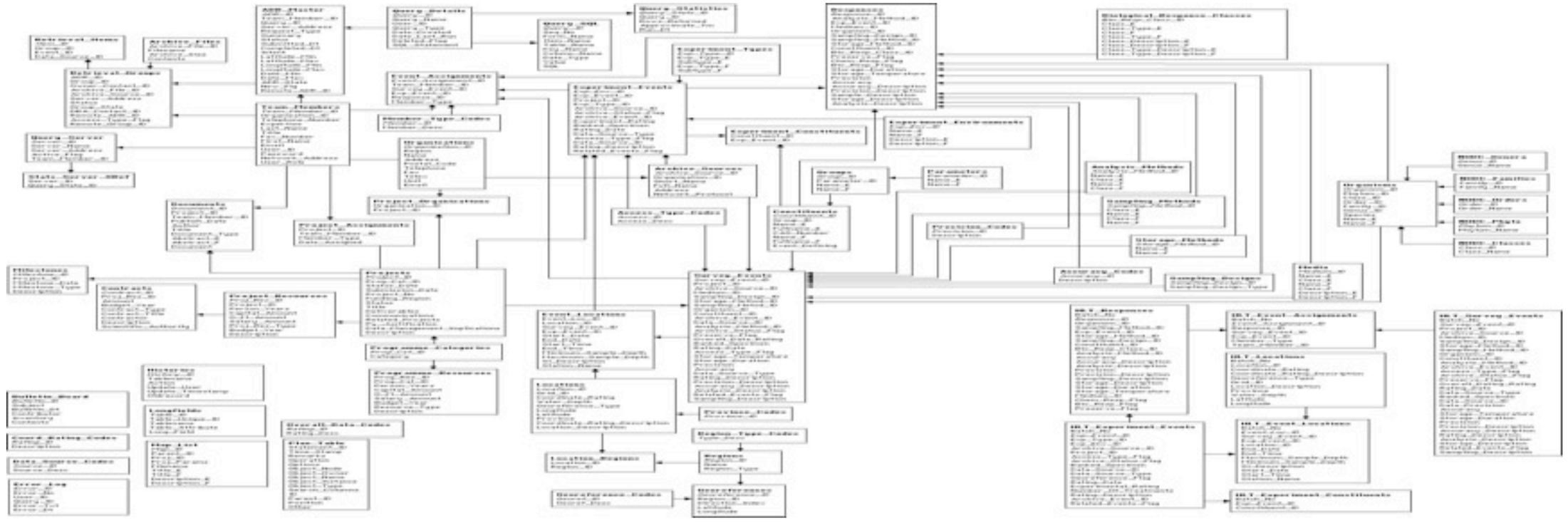


Incept5

ORM - The root of much evil



- Hierarchical does not fit nicely into relational :-(



FpML has to be persisted

- **This still needs to be stored...**

```
<?xml version="1.0" encoding="UTF-8"?><!--
  == Copyright (c) 2002-2007. All rights reserved.
  == Financial Products Markup Language is subject to the FpML public license.
  == A copy of this license is available at http://www.fpml.org/license/license.html
  -->
<FpML xmlns="http://www.fpml.org/2007/FpML-4-4" xmlns:fpml="http://www.fpml.org/2007/FpML-4-4" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" version="4-4" xsi:schemaLocation="http://www.fpml.org/2007/FpML-4-4 ../fpml-main-4-4.xsd http://www.w3.org/
2000/09/xmldsig# ../xmldsig-core-schema.xsd" xsi:type="DataDocument">
   <trade>
      <tradeHeader>
         <partyTradeIdentifier>
            <partyReference href="party1"/>
            <tradeId tradeIdScheme="http://www.chase.com/swaps/trade-id">TW9235</tradeId>
         </partyTradeIdentifier>
         <partyTradeIdentifier>
            <partyReference href="party2"/>
            <tradeId tradeIdScheme="http://www.barclays.com/swaps/trade-id">SW2000</tradeId>
         </partyTradeIdentifier>
         <tradeDate>1994-12-12</tradeDate>
      </tradeHeader>
      <swap><!-- Chase pays the floating rate every 6 months, based on 6M USD-LIBOR-BBA,
            on an ACT/360 basis -->
         <swapStream>
            <payerPartyReference href="party1"/>
            <receiverPartyReference href="party2"/>
            <calculationPeriodDates id="floatingCalcPeriodDates">
               <effectiveDate>
                  <unadjustedDate>1994-12-14Z</unadjustedDate>
                  <dateAdjustments>
                     <businessDayConvention>NONE</businessDayConvention>
                  </dateAdjustments>
               </effectiveDate>
               <terminationDate>
                  <unadjustedDate>1999-12-14Z</unadjustedDate>
                  <dateAdjustments>
                     <businessDayConvention>MODFOLLOWING</businessDayConvention>
                     <businessCenters id="primaryBusinessCenters">
                        <businessCenter>GBLO</businessCenter>
                        <businessCenter>JPTO</businessCenter>
                        <businessCenter>USNY</businessCenter>
                     </businessCenters>
                  </dateAdjustments>
               </terminationDate>
```

NoSQL?

- No RDBMS or NoSQL?

- We can still store the XML or hierarchical data in a RDBMS but SQL is relatively useless to access it

- Now we need something like XQuery

- We have however found a good reason to move away from SQL

Enough of SQL

OK, so SQL has its uses

Hierarchical data is not one of them

Let's look at another problem...

Lots of data

- In the banking world we have a lot of data
- Today 50-100,000 quotes a second isn't unusual
  - We recently hit 350,000/sec from just one source (CME)
- Writing this to a database is possible but not usually practical or necessary
  - Storing 100,000 objects is pretty simple, the data is relatively flat
  - But the problem is rarely so simple
- It gets more complex...

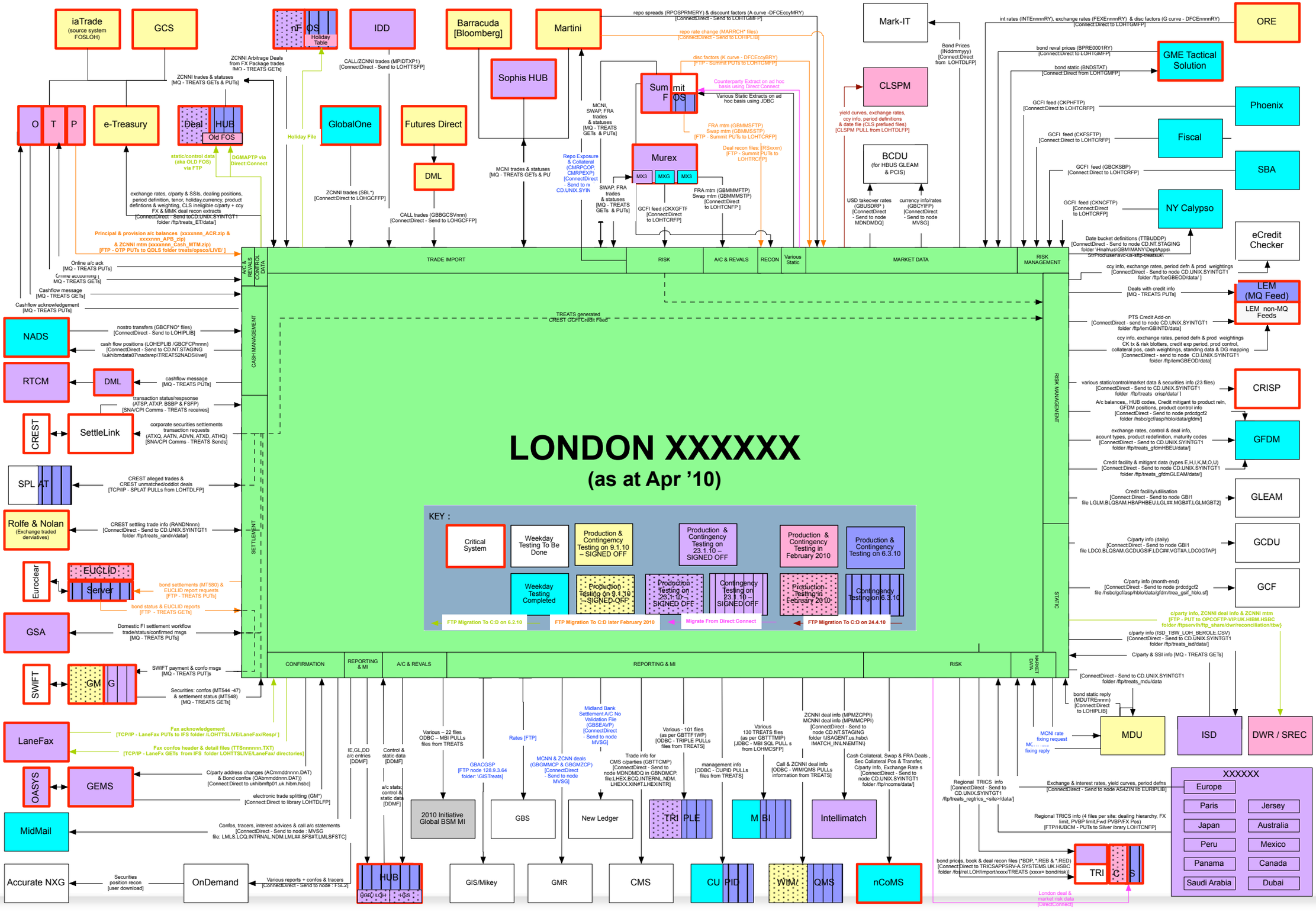
Incept5

Adding complexity

- 10,000 portfolios, each with 1,000 buy/sell orders at specific prices
  - For example one portfolio might contain someone's investment, partly held in Hong Kong equity, one of those equities might have a sell order (for the 500 shares) at HKD $85
- We now have 100,000 prices coming in every second and 10 million orders to watch
  - Technically 1 trillion (10^{12} or 1,000,000,000,000)
  - Fortunately there are optimisations that can be made
- Then repeat this across 20 exchanges
  - Oh yes and if you get a match we need sub milli-second triggers

Incept5

System Architecture View



Functional Landscape

Front Office

- Sales
- Trading
- Client Relationship Management
- Marketing & Advertising
- Product Development
- Research

Middle Office

- Order/Trade Matching
- Trade Allocation & Booking
- Trade Desk Support
- Pricing & Mark-to-Market
- Prime Brokerage Support
- Risk Control

Back Office Operations, Services & Control

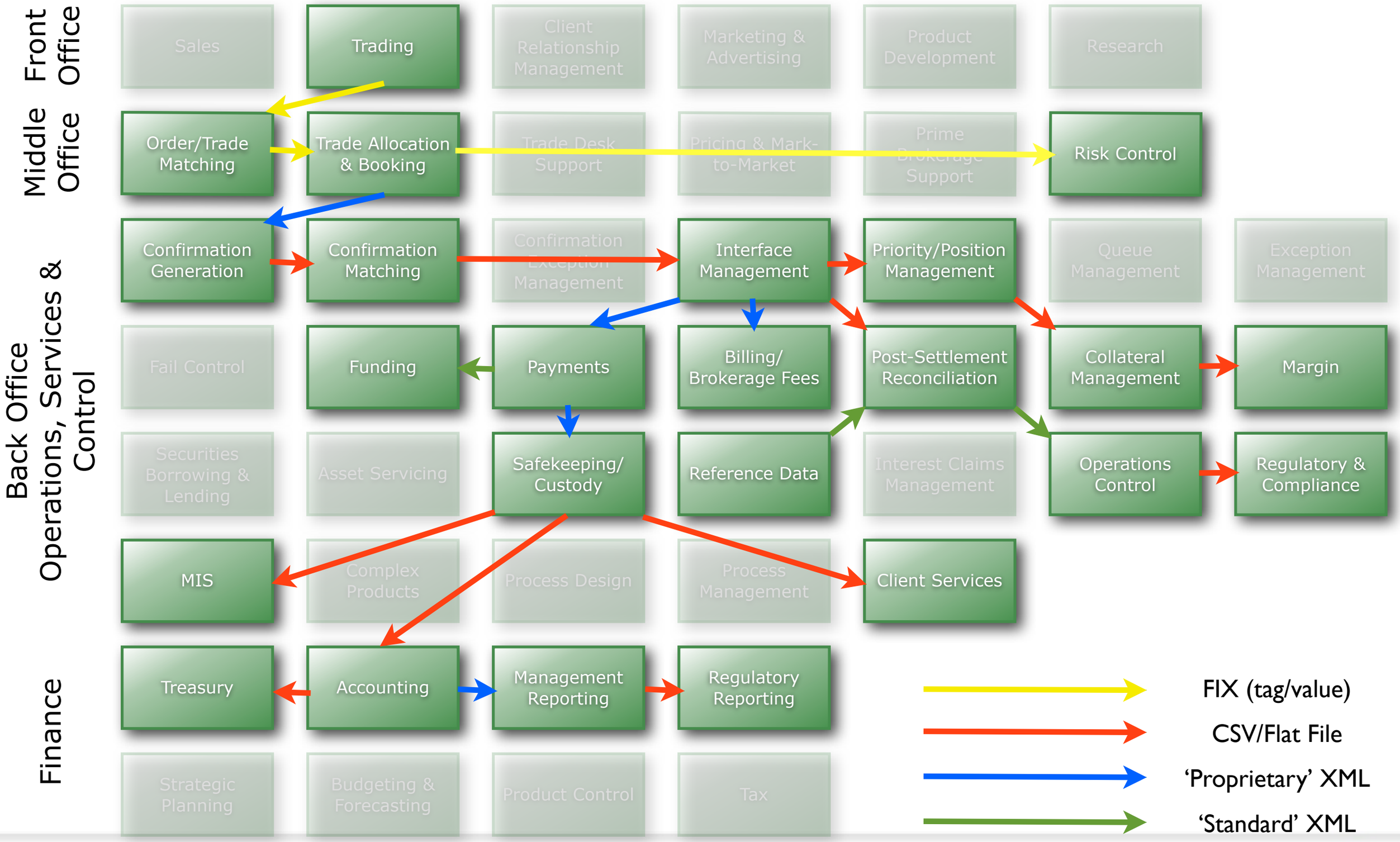
- Confirmation Generation
- Confirmation Matching
- Confirmation Exception Management
- Interface Management
- Priority/Position Management
- Queue Management
- Exception Management
- Fail Control
- Funding
- Payments
- Billing/ Brokerage Fees
- Post-Settlement Reconciliation
- Collateral Management
- Margin
- Securities Borrowing & Lending
- Asset Servicing
- Safekeeping/ Custody
- Reference Data
- Interest Claims Management
- Operations Control
- Regulatory & Compliance
- MIS
- Complex Products
- Process Design
- Process Management
- Client Services

Finance

- Treasury
- Accounting
- Management Reporting
- Regulatory Reporting
- Strategic Planning
- Budgeting & Forecasting
- Product Control
- Tax

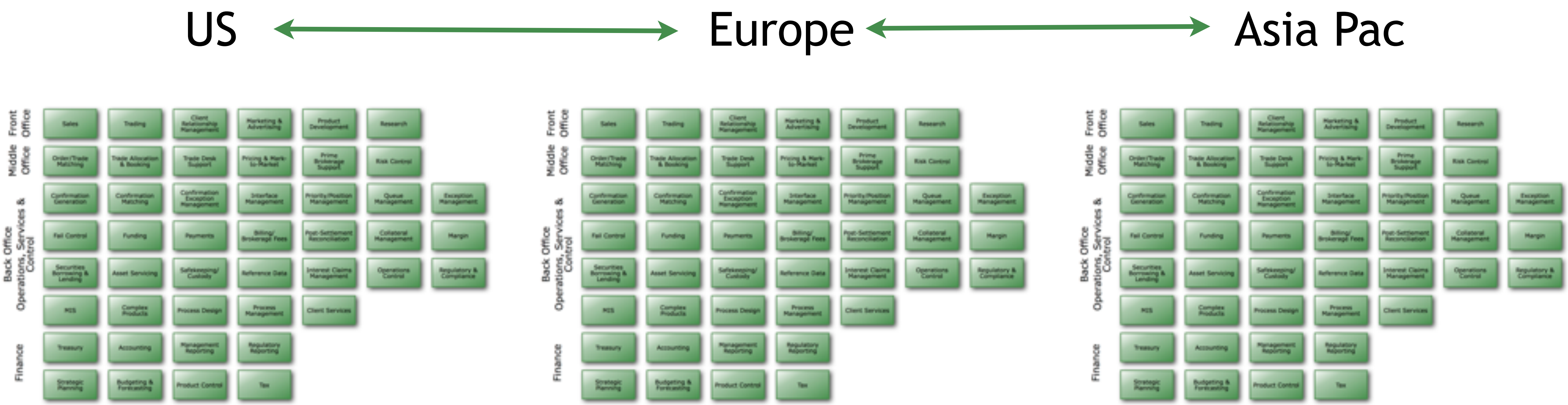
Incept5

Just one transaction...

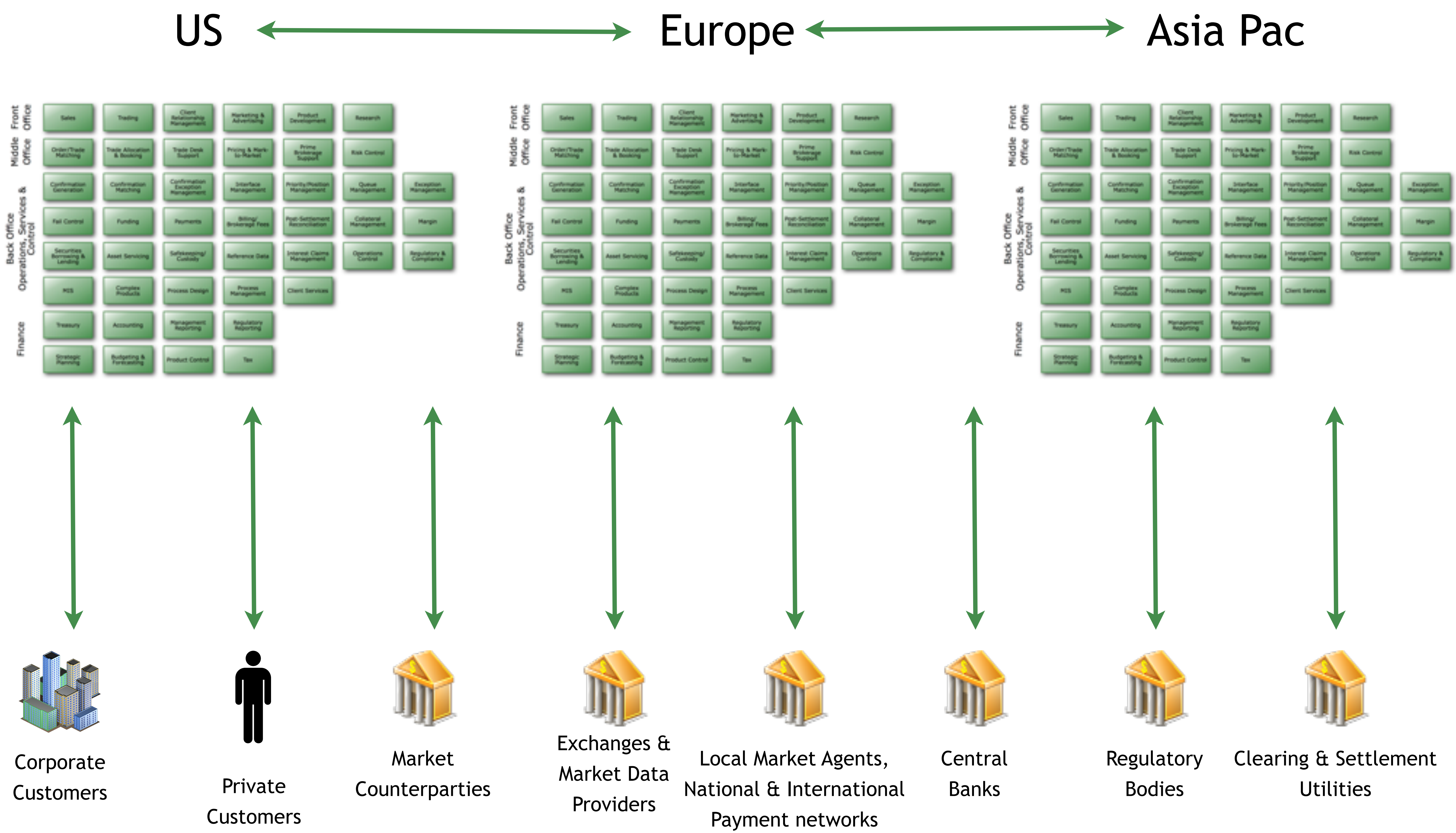


Incept5

Now add the geography...

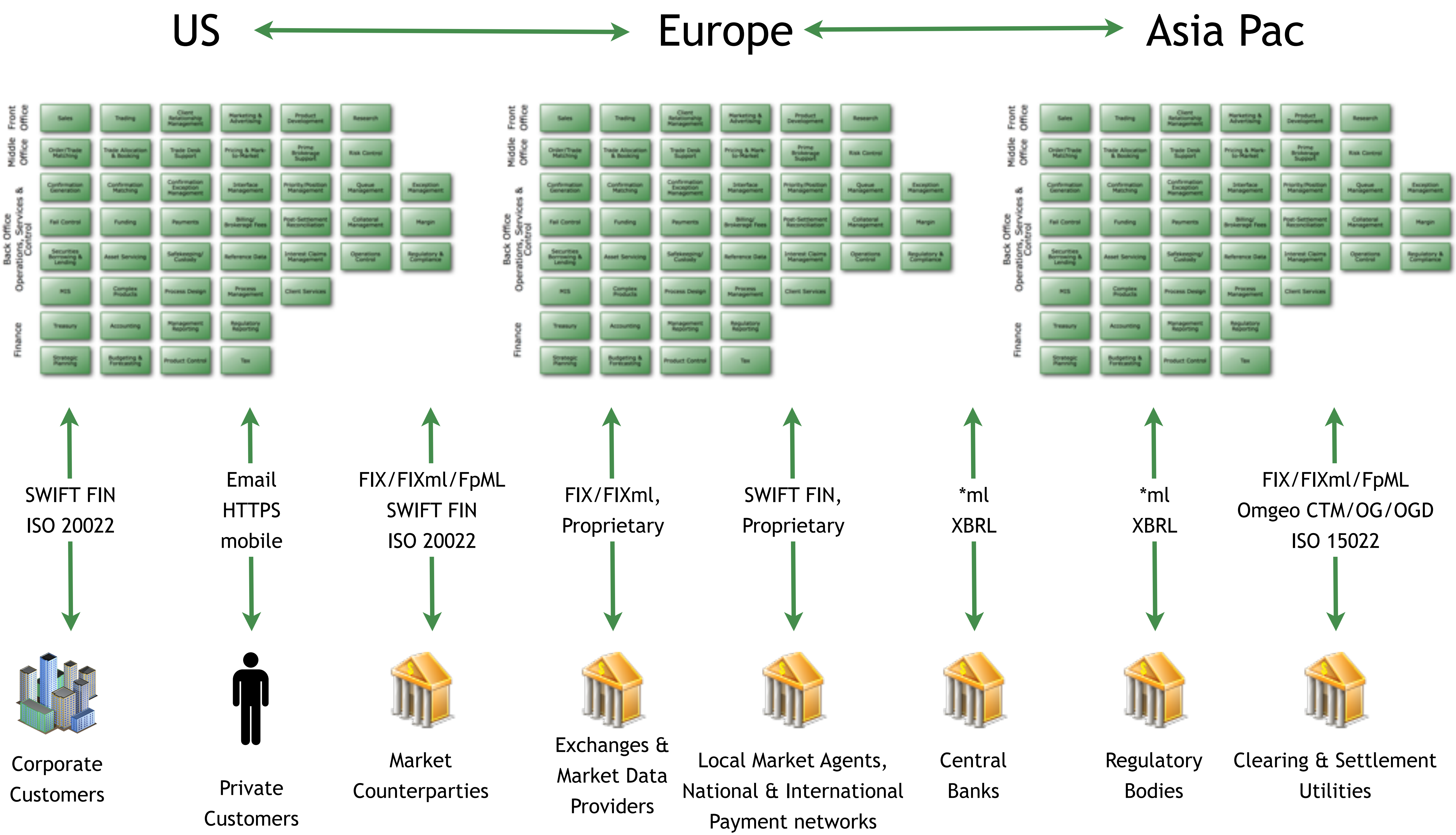


And the rest of the world...



Incept5

Add a few standards...



Incept5

Risk - The new driver

- Everything presents a risk...
  - The equity / asset could de-value
  - The seller could go under (bankrupt)
  - The currency of the asset could de-value
  - The political regime of the seller could change
  - The political regime or currency of the parent bank could change
  - The broker or counter-party who brokered the deal could go under

- All these may appear small risks but they are very real
  - Remember Sub-prime (US), Enron (US), Northern Rock (UK)?

- It's like a Tokyo or San Francisco earthquake, a very small chance it will happen tomorrow but it will happen one day

Lots of data, lots of calculations

- There are two main flavours of distributed computing
  - Data
  - Computation

- Often they are closely related but not always

- To achieve either we usually need lots of memory and CPUs

- We don't stack them or put them in clusters these days, we distribute them
  - Usually in a rack - but you don't need to know that

Incept5

We need to scale

Huge amounts of data
Vast amounts of computations

We need scaleability

Distributed Computing - Ideal case

- In an ideal world we code without having to know about the deployment architecture
  - We assume a machine powerful enough and with enough memory to perform our task(s)
  - For a long time this was the case and it often still is

- Sadly in the enterprise this doesn't usually work so we need to design to scale, scaling both memory and CPU power
  - Computers on their own don't scale, good architecture does

- We code for a distributed environment, we hand out tasks to be distributed and access & persist data through APIs

Incept5

Change the programming model

- Probably the hardest thing to drill into programmers and worse, their managers, is to program for a distributed architecture
  - It all seems like an overhead at first
  - Abstraction of location
  - Storage though APIs
- The EJB model was a start but scalability was limited to the server or cluster of servers
  - JNDI lookup
  - Object life-cycle manage by the container
  - Spring extended this

Grid - Local vs Cloud

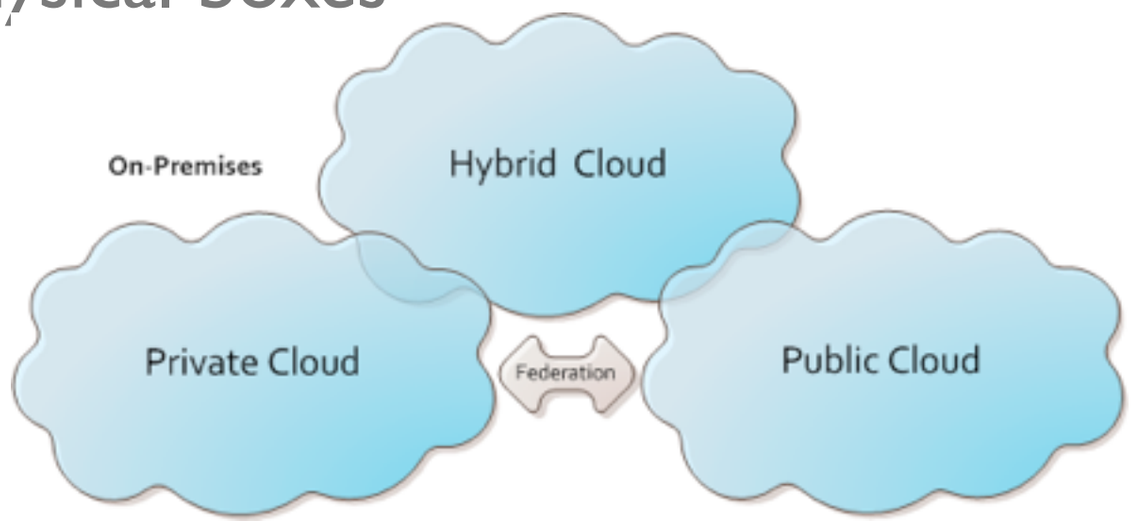
- If we can distribute to local VMs we're most of the way there
- Move the VMs to other machines on the network and we have "Grid"
  - Also known today as "private cloud", these can be physically local or remote but usually on your network or VPN
  - Today's investment banks and hedge-funds have anything from 200 to over 20,000 CPUs in their grids
- Use someone else's hardware and you have "Cloud"
  - Amazon's EC2 is a perfect example

Local vs Grid vs Cloud

- Local
  - Very fast but limited by the physical size/power of your box
  - Perfect for developing and testing

- Private Cloud / Grid
  - Very secure - perfect for banks & governments
  - Very scalable but slight latency
  - Costly - as you have to buy the physical boxes

- Cloud
  - Pay for what you eat
  - Extremely scalable
  - Latency and security can be an issue



Incept5

Cloud Foundry

- Perhaps one of the best ways to get started

- Open Source and secure

- You can work locally on the "Micro Cloud Foundry" and them upload to the Cloud

- Get started here: **www.CloudFoundry.com**

Incept5

Grid technologies

- Let's look at some of the Grid technologies, there are dozens but we'll take a slightly closer look at a few...
  - GemFire
  - Terracotta (BigMemory)
  - GigaSpaces
  - Coherence
  - Neo4J

GEMFIRE

TERRACOTTA

GIGASPACES

ORACLE Coherence

Neo4j the graph database

- Many other technologies overlap in areas, predominately the caching side, these too are viable alternatives
  - EHCache, Memcached, JCache (JSR-107) etc.

- It would be wrong not to mention NoSQL DBs...

Incept5

The list goes on...

- Many of the following are appearing on the scenes
  - MongoDB
  - HBase
  - Cassandra
  - Riak
  - CouchDB
  - Redis

- MongoDB is pretty popular
  - HBase with Hadoop and Cassandra occasionally too
  - Others I've not seen but that doesn't mean they're not being used, many of them have extremely powerful features with considering

Incept5

How to compare?

- I could write a book and talk to you for days
  - But I still can not tell you which is best
- I've seen a lot of money spent on comparing them
  - Each one of them will give you examples where they've excelled
- Most of them will do the job and most of them will tell you bad things about the others
- All I can do is point out a few major differences
  - Anything claim I make would be disproven as things evolve

Incept5

Grid technologies

- GemFire
  - Originally an OOD, now has a pure Java implementation
  - Recently acquired by VMWare
- Terracotta
  - Uses Java VM replication,
  - Recently acquired by Software AG
- GigaSpaces
  - Originally the only viable implementation of Sun's JavaSpaces
- Coherence
  - Formally "Tangosol", now owned by Oracle
- Neo4J
  - The wild-card, a graph database

GemFire

- Connect to the distributed system and get the Cache ref

```
// Create / Find a cache (using the map interface)
DistributedSystem ds = DistributedSystem.connect();

// Get the Singleton instance of the cache
Cache cache =  CacheFactory.create(system);
```

- Instantiate your object and simply put into a Map

```
// Create / Find Data Region "Prices" in the cache
Map prices = (Map ) cache.getRegion("Prices");

// Write the Price to the cache...
prices.put( price.getKey(), price );
```

- As you can see this couldn't be easier

Incept5

Reading a Price from GemFire

```
// Get Access to the Data Region "Prices" and cast as a java.util.Map
Map map = (Map)cache.getRegion("Prices");

// Retrieve the latest spot price for GBP/NOK
Price myPrice = (Price ) map.get("GBP/NOK-SPOT" );
```

- All GemFire Regions are indexed on the key used in put()

```
// Get Access to the Data Region "Prices"
Region prices = cache.getRegion("Prices");

// If the retrieval is not based on primary key, you can use OQL
// Retrieve the latest spot price for GBP/NOK
SelectResults results = prices.query("getKey() = 'GBP/NOK-SPOT'");
for (Iterator iter = results.iterator(); iter.hasNext(); ){
   Price myPrice = (Price) iter.next();
}
```

- All Regions can be indexed on fields and/or methods

Incept5

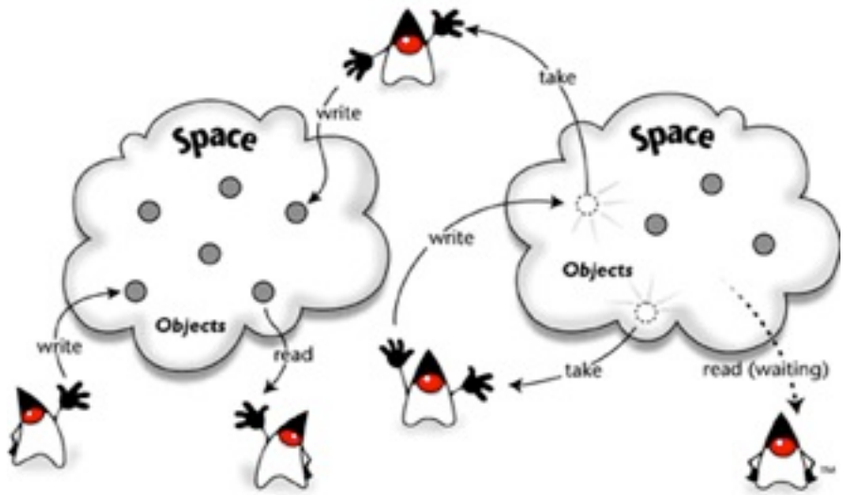
GigaSpaces

- GigaSpaces is the perfect implementation of a Master/ Worker pattern
  - But they don't really push this sadly

```
public String url = "jini://*/*/C24Space";
space = new GigaSpaceConfigurer(new UrlSpaceConfigurer(url)).clustered(true).gigaSpace();

// CacheItem is our own POJO to host an ID & SWIFT POJO from C24's Integration Objects
ci = new CacheItem();
MT513Element mt513Element = new MT513Element();
MT513Message mt513 = ci.parseMT513FromString(rawSwiftMT513);
id = extractID(mt513);

ci.setId(id);
ci.setMessageData(mt513);
space.write(ci);
```



- Once again really easy to use
  - Notice the interface is not a Map here
  - In "classic" JavaSpaces we use a template to retrieve data
  - But GigaSpaces have added new search APIs now

Incept5

We do all this today

- This is everyday technology
- The technology is not complex
  - The implementations are but the architecture is not
- If you think big when you architect you will go far
- Architect for huge, design for simplicity and don't pre-optimise!
- More in my talk this afternoon!

Thank you!

谢谢

Incept5

ArchSummit

中国·深圳 2012.08

INTERNATIONAL ARCHITECT SUMMIT

全 球 架 构 师 峰 会

详情请访问：architectsummit.com

- 3天 • 6场主题演讲
- 3场圆桌论坛 • 9场专题会议
- 国内外30余家IT、互联网公司的50多位来自一线的讲师齐聚一堂

主办方：InfoQ 战略合作伙伴：Tencent 腾讯 特别支持：

http://architectsummit.com


QCon
杭州站 · 2012年10月25日~27日
www.qconhangzhou.com（6月启动）
QCon北京站官方网站和资料下载
www.qconbeijing.com