



Enterprise Architecture  
Conference Europe 2011



Business Process Management  
Conference Europe 2011

**8 - 10 June 2011**

Produced By



# Practical Enterprise Integration

## Realising the Benefits of a Strong Canonical Architecture

Andrew K. Johnston  
Independent consultant at National Grid

[www.andrewj.com](http://www.andrewj.com)  
[www.agilearchitect.com](http://www.agilearchitect.com)

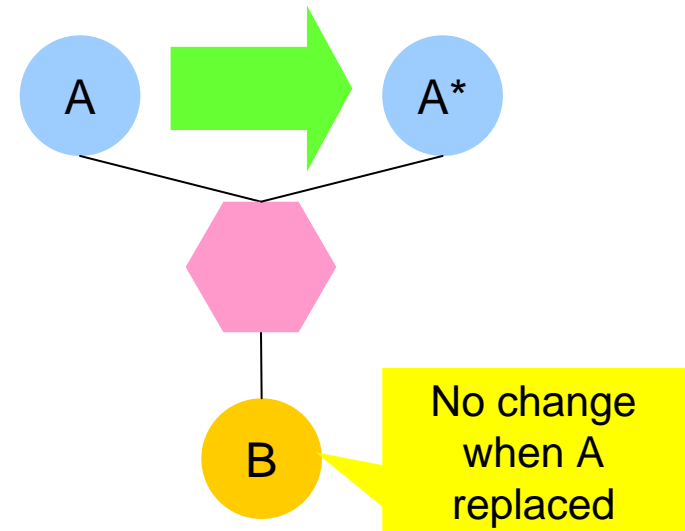
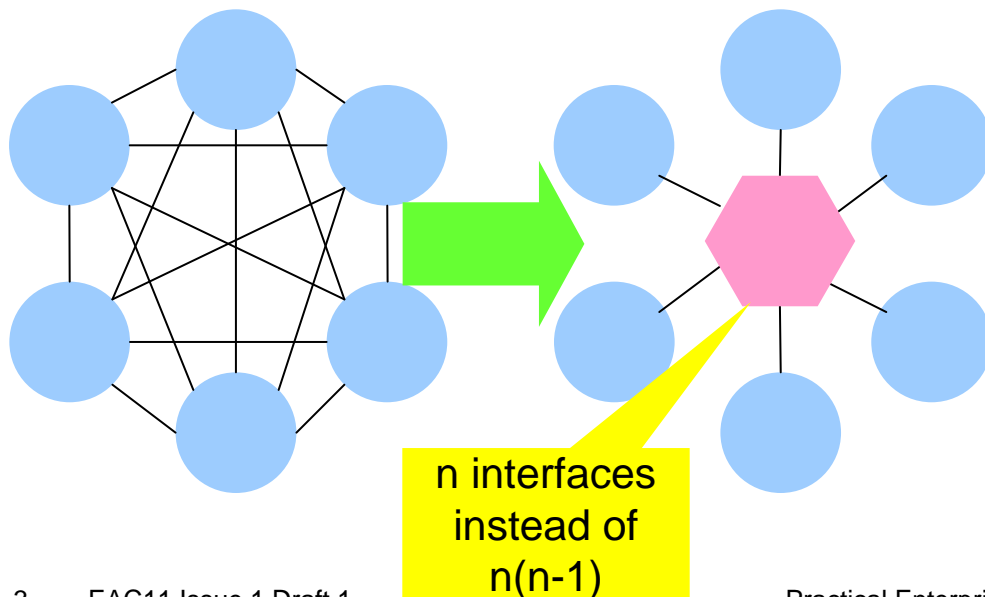
[www.nationalgrid.com](http://www.nationalgrid.com)

**nationalgrid**

The power of action.™

# What's This About?

- ◆ We've all heard of EAI
- ◆ We all know the theoretical benefits
- ◆ We haven't all seen evidence of actually delivering multi £M benefits
- ◆ This is the multi-year story of a real, enterprise-scale example
  - ◆ An example of "Pace Layering" in action!

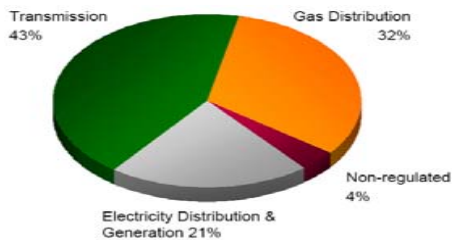
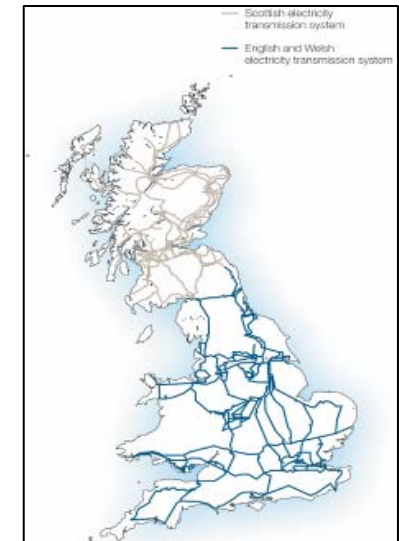


# nationalgrid

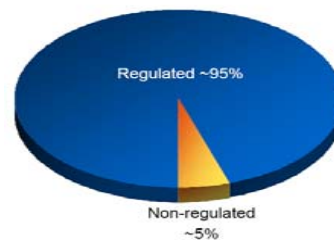
The power of action.™

- ◆ Largest investor owned utility in the UK, second largest in the US
- ◆ Electricity & Gas
- ◆ Generation, Transmission, Distribution & Retail Supply
- ◆ US & UK
- ◆ UK Transmission run both the UK's high voltage electricity transmission grid, and the high pressure gas transmission system

## UK Electricity (T)



**Asset Base**



**Revenue**

## US Electricity (T & D Network)



# Our Scope: Enterprise Within an Enterprise

- ◆ These slides describe what has been done for UK Transmission
  - ◆ UKT manages, maintains and operates UK's high voltage electricity grid, and national high pressure gas transmission network
  - ◆ EAI development focused on Asset and Work Management systems, but supporting links to operational systems and shared services such as supply chain
  - ◆ Model originally developed for electricity, now applies almost equally to gas
- ◆ This is an “Enterprise within an Enterprise” - Line of business focus, but enterprise-scale size & complexity
  - ◆ Significant numbers of users and supply chain partners
  - ◆ ~ 1 million maintained assets
  - ◆ At least 100 work and asset management systems before rationalisation
- ◆ National Grid has single IS function across all regions and lines of business. However:
  - ◆ There is considerable variation in core systems due to history
  - ◆ Strategic consolidation on SAP and “best of breed” systems in progress but not complete
- ◆ A key challenge is to leverage experience and solutions across different parts of National Grid

# Key Players in EAI Implementation

- ◆ Very much a collaboration between multiple parties partnered with National Grid
- ◆ “We couldn’t have done it without...”
  - ◆ AMT-Sybex
    - ◆ Suppliers of MIMS/Ellipse and integration expertise
    - ◆ Designed and built the original version
    - ◆ Continue to manage the design
  - ◆ Accenture
    - ◆ Developed and maintain the integration around FFE
  - ◆ Wipro and TCS
    - ◆ Developers of integration code since 2008
    - ◆ Operate and support the system
- ◆ My role as Solution Architect
  - ◆ Enterprise architecture: develop and maintain the “big pictures”
  - ◆ Solution architecture: ensure designs are consistent and of high quality
  - ◆ Innovation: originating improvements and solutions to specific problems
  - ◆ Co-ordination: trying to hold it all together!

AMTSybex<sup>®</sup>

accenture

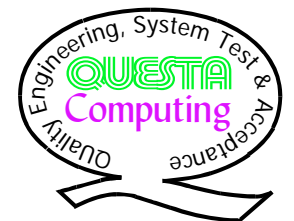
*High performance. Delivered.*



**WIPRO**  
*Applying Thought*



**TATA**



**nationalgrid**

*The power of action.*

# Where Did It Start?

- ◆ Pre-2000: Significant system fragmentation, lots of bespoke “integration spaghetti”
  - ◆ 64 Asset Management Systems, and that’s excluding Gas Transmission!
- ◆ 2000-3: Business consolidation and asset systems review drove investigation into role of EAI in systems rationalisation
  - ◆ Identified potential future core systems, and role of an EAI backbone
  - ◆ Highlighted SeeBeyond as most likely technology
- ◆ 2003: Acquisition of Transco provided UK experience of EAI, and SeeBeyond eGate as incumbent product set
- ◆ 2003-5: “Staying Ahead” programme to provide key new business capabilities for UK Transmission, reduce workforce by 20%: £30M IS investment in new & rationalised systems
  - ◆ Consolidation of asset systems
  - ◆ Field force mobile system
  - ◆ New document management system
  - ◆ Data warehouse and decision support tools
  - ◆ **EAI backbone to link it all together!**

# Early Successes and Failures

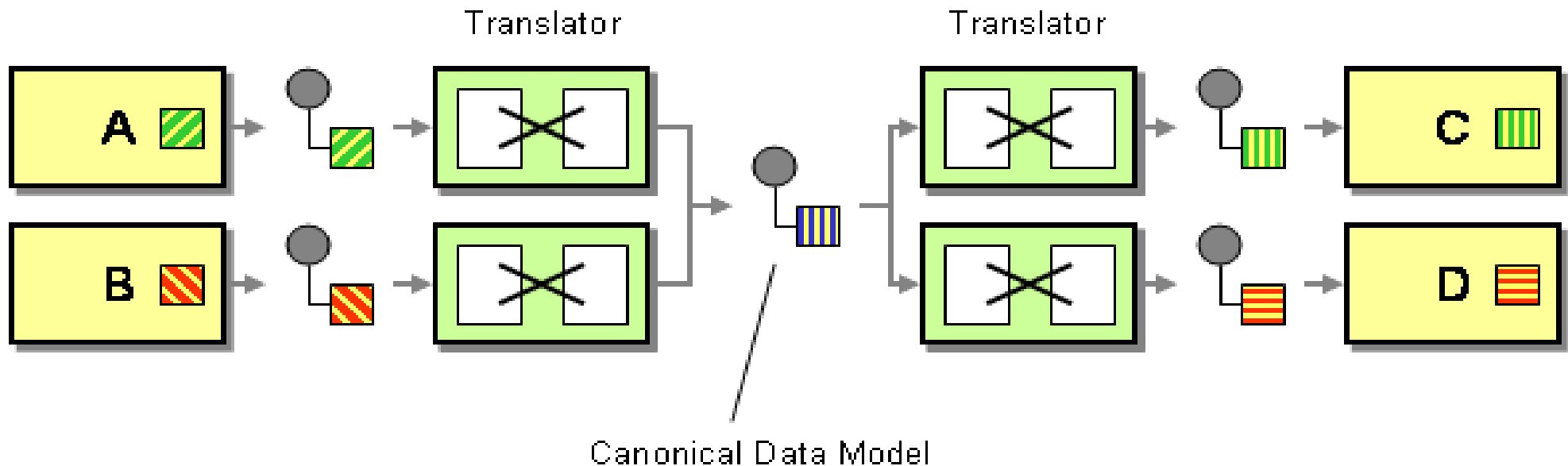
---

- ◆ What we got right
  - ◆ “Core plus satellite” model for asset systems
  - ◆ The Common Message Model
  - ◆ Re-use and change isolation capabilities
- ◆ What wasn't so good...
  - ◆ Fragmented integration responsibilities
  - ◆ Multiple hand-offs in key integration chains
  - ◆ Varying integration models driven by different supplier preferences
  - ◆ Performance and reliability problems, exacerbated by complex responsibilities



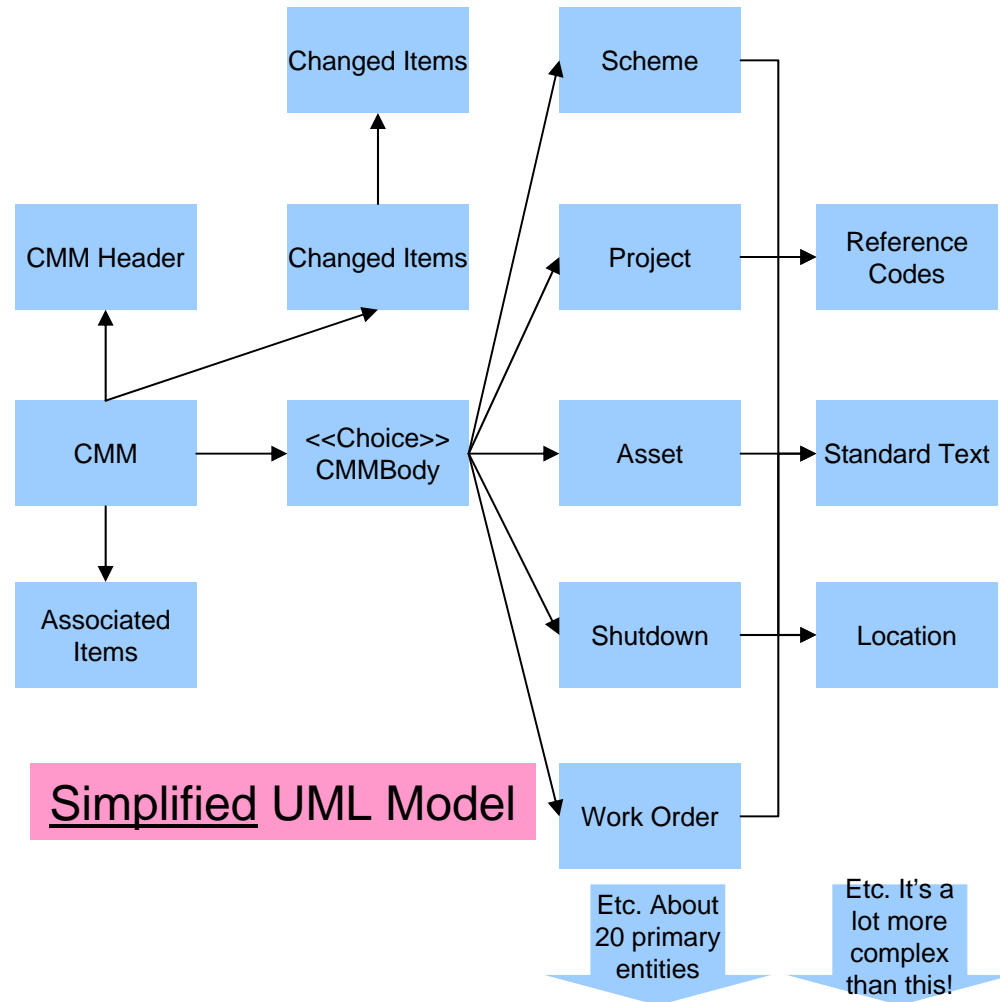
# The Canonical Data Model Pattern

- ◆ Problem: Many-many message-based integration
  - ◆ Many/all systems have different data formats
- ◆ Solution: Use the “Canonical Data Model” pattern
- ◆ Delivers “hub and spoke” benefits at the logical level, as well as the physical

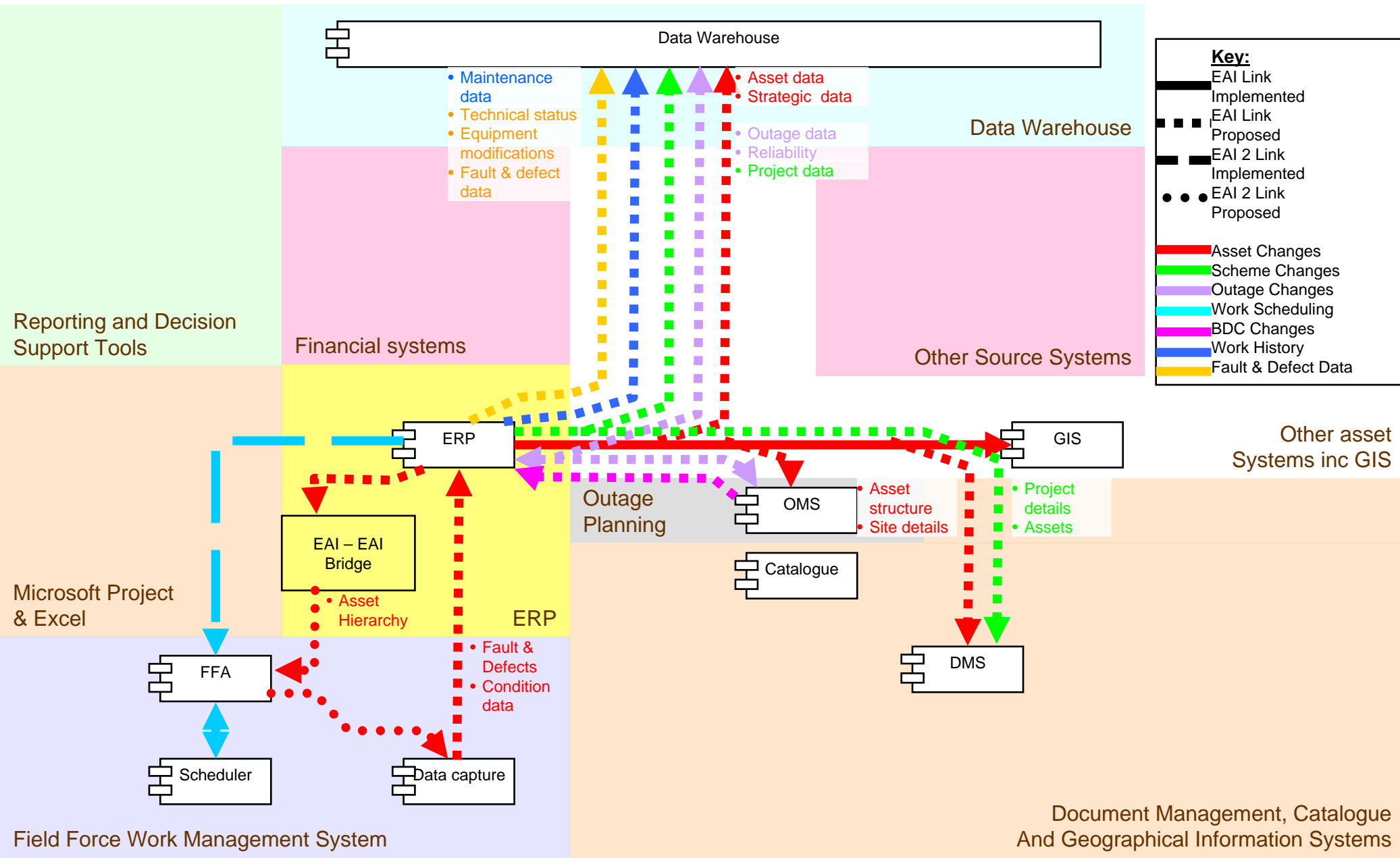


# UK Transmission's Common Message Model

- ◆ Canonical message model used to intermediate between system-specific formats
- ◆ Used for all except a few very high volume, low complexity links
- ◆ Business meaningful structure, rather than “meta model”
- ◆ Modelled in UML
- ◆ “First cousin” to IEC CIM: CIM wasn’t mature when we started, but provided key concepts and formats
- ◆ Early implementations suffered from errors in manual coding. Now use Sparx Systems Enterprise Architect to generate XSD schema direct from UML

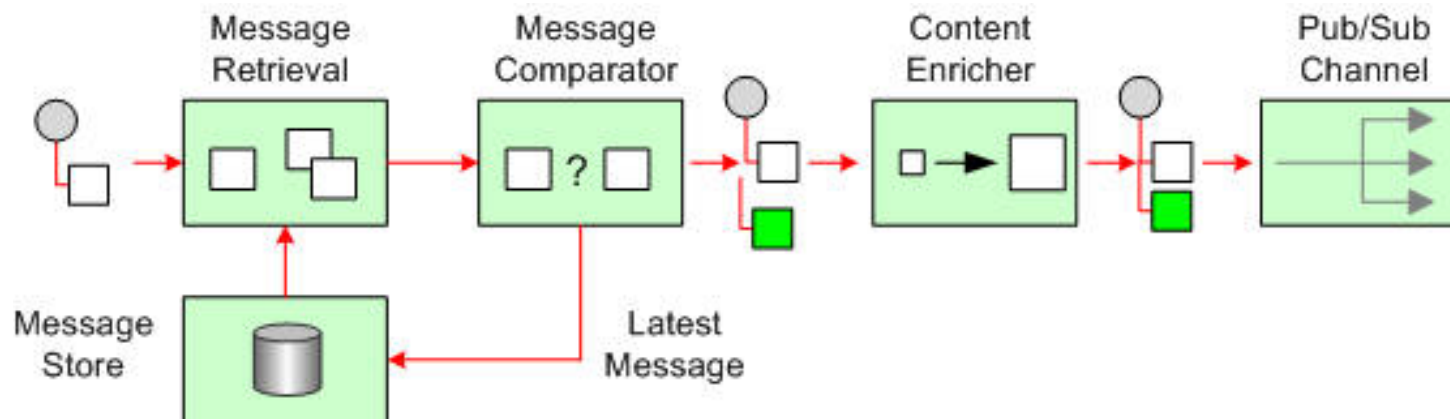


# Interface and Data Reuse: the EAI "Bus Map"



# Asset Feed Problems and Solutions

- ◆ Envisaged a “trickle feed” of asset updates from Asset Inventory
- ◆ Turned into a flood, because of bulk updates to e.g. account codes, not relevant to downstream systems
- ◆ EAM adapter couldn’t identify “what has changed” – just sent whole record every time
- ◆ Solution exploits integration layer:
  - ◆ Stores last message per asset
  - ◆ Compares content to identify changes, and enriches messages with “changed items” info
  - ◆ Integration layer then filters records per system based on relevance of changed items
- ◆ Solution later exploited to rationalise similar interfaces, and provide auditing features



# Adding the “Point of Work” Solution

- ◆ Problem: PC-based field force solution working well, but physically too large & heavy for use “at point of work”
  - ◆ Impractical for overhead line surveys and other inspection work
  - ◆ Resulted in data being captured manually, with costly & error-prone transcription back at office
- ◆ Solution: add a PDA version of the Field Data Capture Solution, as a “satellite” device to the PC
- ◆ Challenges: limited funding, strong desire not to change field force system itself (now stable after initial problems)
- ◆ Design mantra: exploit existing interfaces, **zero** change to FF system
  - ◆ PoW solution “transparently” uses and updates same files as PC solution
- ◆ **Outcome: success! Zero change required to FF or back end systems. Initial prototype delivered in about 10 weeks and immediately exploited in the field**

# The Next Big Challenge: Core EAM System Upgrade

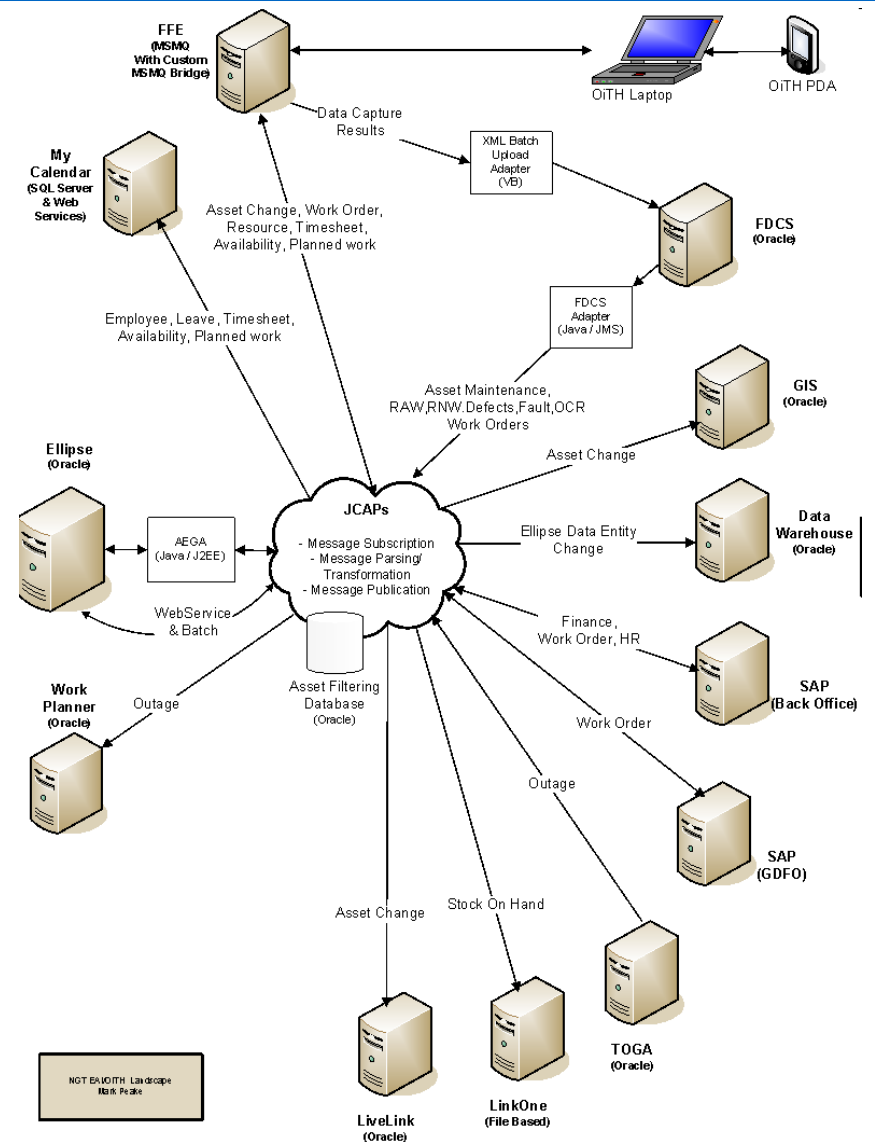
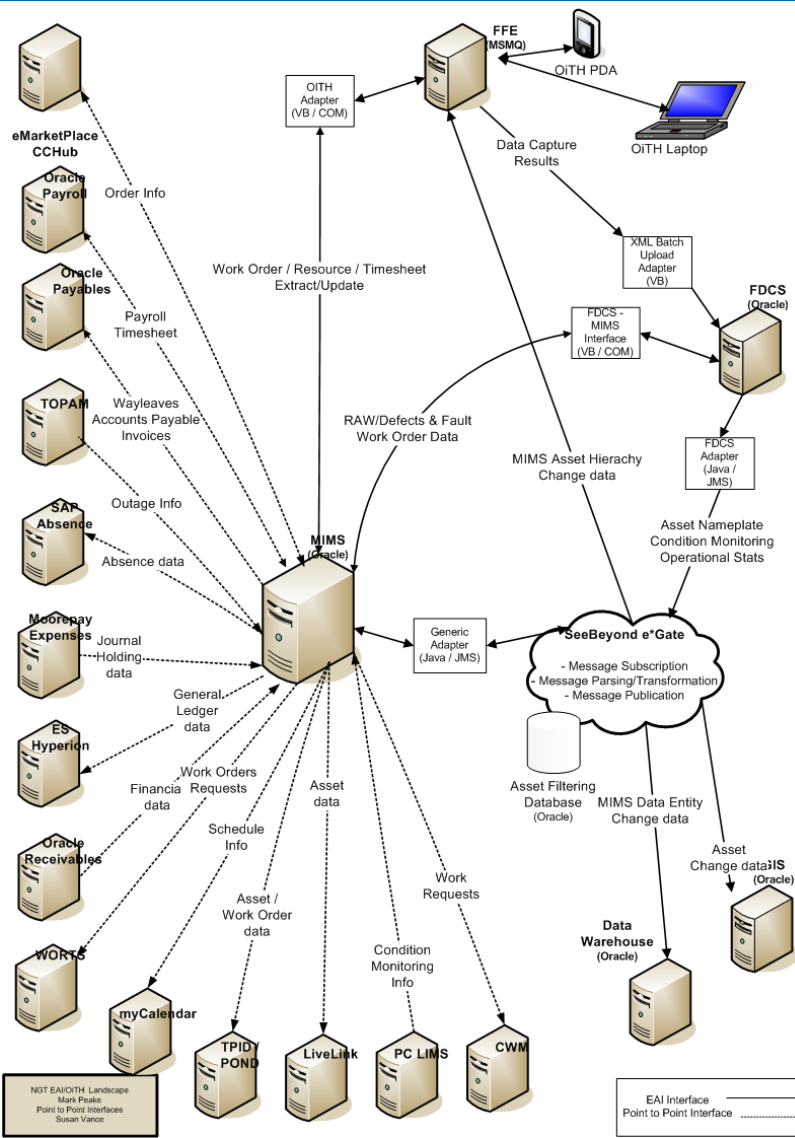
Having just about got things stable, we embarked on another major change...

- ◆ Replaced core Work and Asset Management system (MIMS) with much newer version (Ellipse)
- ◆ Completely new hardware, operating systems & database
- ◆ Changed “back office” system from Oracle to SAP
- ◆ “Boundary change” moved key back office functions previously in MIMS (e.g. materials management) to new SAP system
- ◆ Replaced SeeBeyond eGate integration layer with new version (Sun JCAPS)
- ◆ Significantly rationalised the integration model, got rid of a lot of “spaghetti”
- ◆ Replaced custom integration adapters with standardised flows

And...

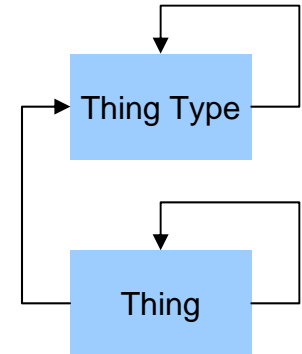
- ◆ Largely avoided knock-on impacts the other core systems, through strength of integration model

# Rationalising the Integration Architecture



# The Transformation Engine

- ◆ MIMS / Ellipse has a powerful integration model, but it's based on a meta-model of the data (e.g. the payload is an object which other payload data describes as an asset)
- ◆ Our CMM is based on a “business meaningful” model of the data (e.g. the payload is an asset, so the “asset” node is populated)
- ◆ Prior to the upgrade, each transformation was a complex hand-coded mapping, with separate “request” and “enrichment” stages
- ◆ In the Ellipse world, we would have >50 of these!
- ◆ Enter “The Transformation Engine”
  - ◆ Two generic transformations (one in each direction)
  - ◆ Request and mappings defined in a common, configurable rule table

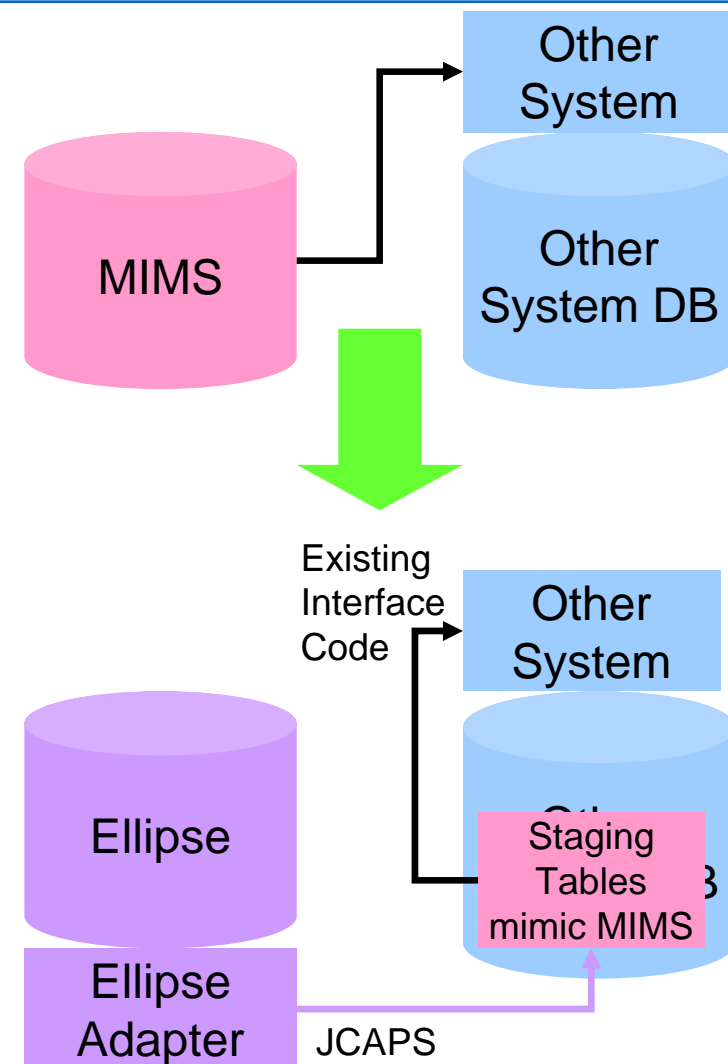


The eTXML Model!  
(somewhat simplified 😊)



# Integration Successes from the EAM System Refresh

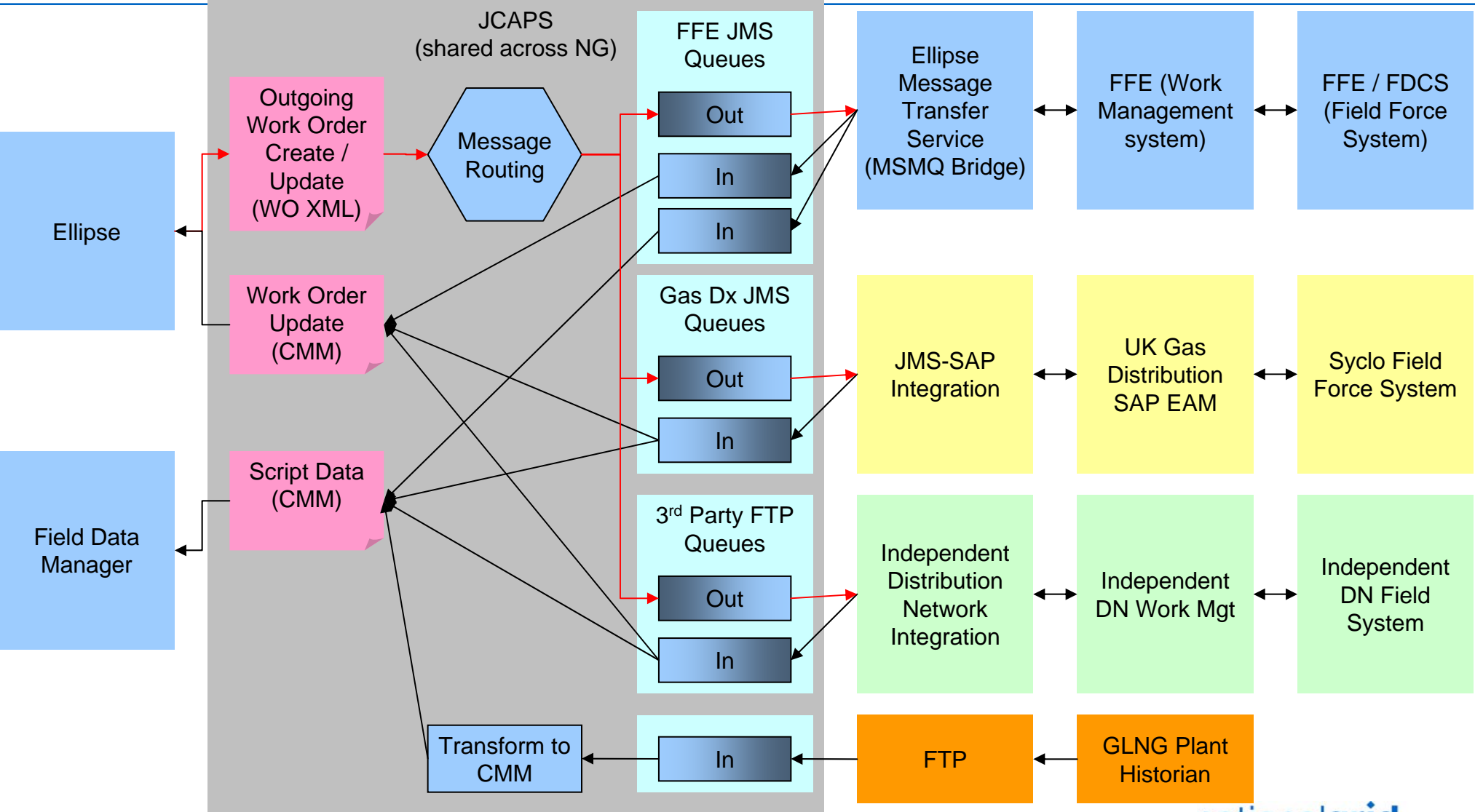
- ◆ All Ellipse interfaces converted to JCAPS, with JMS or FTP interface
  - ◆ Got rid of all database / ODBC links
  - ◆ Avoided downstream changes using “staging table” design pattern (see right)
- ◆ Proper message based interfaces replaced wide variety of file and database links
- ◆ Consolidated several similar EAI flows
- ◆ Web Services used for real-time request / response exchanges between Ellipse & SAP
- ◆ **No significant change to other major systems:**
  - ◆ Field force system
  - ◆ Data warehouse
  - ◆ Geospatial information system
  - ◆ Document management system
  - ◆ Minor work management systems



# Extending Further Into the Enterprise

- ◆ Through 2009-2010, we have progressively applied the pattern across other parts of National Grid in the UK
- ◆ Liquid Natural Gas Storage and Grain LNG “non-regulated businesses” adopted Ellipse as EAM system
  - ◆ Needed own Ellipse “district” (effectively separate “company” in same instance)
  - ◆ Made integration model “multi-district” with **zero knock-on changes**
  - ◆ Now exploiting existing asset information flows to integrate to Plant Historian Database
- ◆ NG Gas Distribution do some work on behalf of Gas Transmission
  - ◆ New EAM system “tees” into existing work and asset data feeds (see next slide)
  - ◆ **No changes required to Ellipse or OITH**
  - ◆ Same approach can be used for work done by independent Gas Distribution companies
- ◆ Cathodic protection surveys managed in a separate system (Uptime)
  - ◆ Will exploit similar architecture to schedule surveys and confirm their completion
- ◆ **All possible because we are working with a strong, flexible message model!**

# Tapping into the Existing Work and Asset Flows



# A Reduction of Spaghetti

- ◆ System continues to evolve with progressive reduction of “integration spaghetti”
- ◆ Each upgrade / replacement project tries to streamline and standardise interfaces
- ◆ Example: bridging to MSMQ
  - ◆ Originally: complex, unreliable “adapter server” with support responsibilities split 5 ways
  - ◆ After Ellipse: server still existed, but adapter software reduced to simple “transfer service”
  - ◆ Now: JCAPS connects directly to MSMQ, server virtualised and moved under single party control
- ◆ Example: interfaces to “My Calendar” system
  - ◆ Originally: single-purpose HTTP “screen scraping”, with complex proprietary “adapter” software
  - ◆ Late 2011: web services using Common Message Model as native message format



# Looking Forwards

- ◆ What are the future challenges?
- ◆ Promoting the lessons and best practices elsewhere in NG
  - ◆ Can we do the same thing with other technologies, in particular for the strategic SAP footprint?
- ◆ Extending the model for more service exchanges
  - ◆ Can we use the CMM as a basis for true SOA?
  - ◆ What's the right model for a mix of asynchronous messaging and synchronous service exchanges?
- ◆ Supporting Strategic Asset Management
  - ◆ How do we move dynamic asset condition & performance data around for novel analysis and presentation?
  - ◆ How should we bring data from multiple systems together in composite applications and portals?
- ◆ Incorporating industry standards
  - ◆ Can we use IEC CIM for real-time asset data flows?
  - ◆ Can we use IEC CIM as an “external” message standard?



# Looking Backwards

- ◆ Lessons Learned
  - ◆ You need a strong logical architecture as well as technical tools
    - ◆ Otherwise you just produce “technically consistent spaghetti”
  - ◆ Someone has to act as guardian of the architecture
  - ◆ Don't wed yourself to technical perfection
    - ◆ Ideas which look good on paper may not always be the best fit
    - ◆ Remember: No battle plan survives contact with the enemy!
  - ◆ Allow systems to evolve at their own speed – “pace layering”
    - ◆ Design so that the most volatile components are separate from the less volatile ones, and ideally treated as data
    - ◆ Exploit the integration architecture to minimise knock-on impacts of system changes
- ◆ Can we quantify the benefits?
  - ◆ Business value delivered – met original 25% efficiency targets, now supporting growing footprint and business volumes
  - ◆ Dramatic avoided costs – easily £0.5-1.0M per project, probably around £10M total by now
  - ◆ **Well worth the investment in both EAI and CMM**



**Any Questions??**