

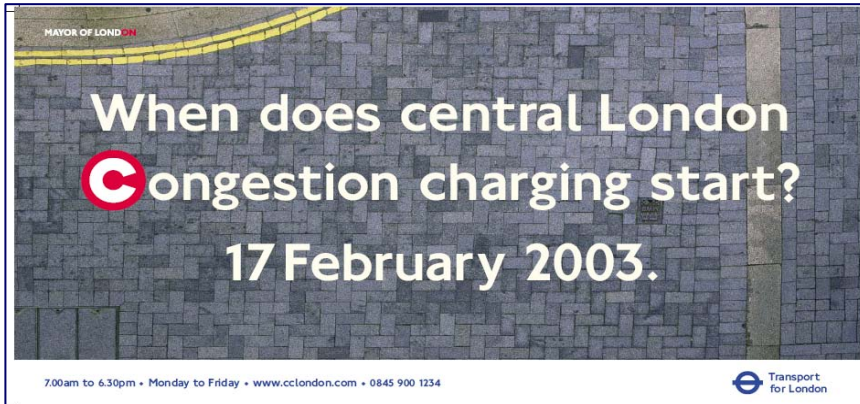


**Enterprise Interoperability Through SOA.**  
**Case Study:**  
**The Central London Congestion Charge.**

# Agenda

- Background to the London Congestion Charge
- What's a service?
  - Contextual view
  - Solution Elements
  - Service Catalogue
  - Example
- What is needed of a service for interoperability?
  - Technical delivery
  - Service levels
  - Governance
- Some lessons learned

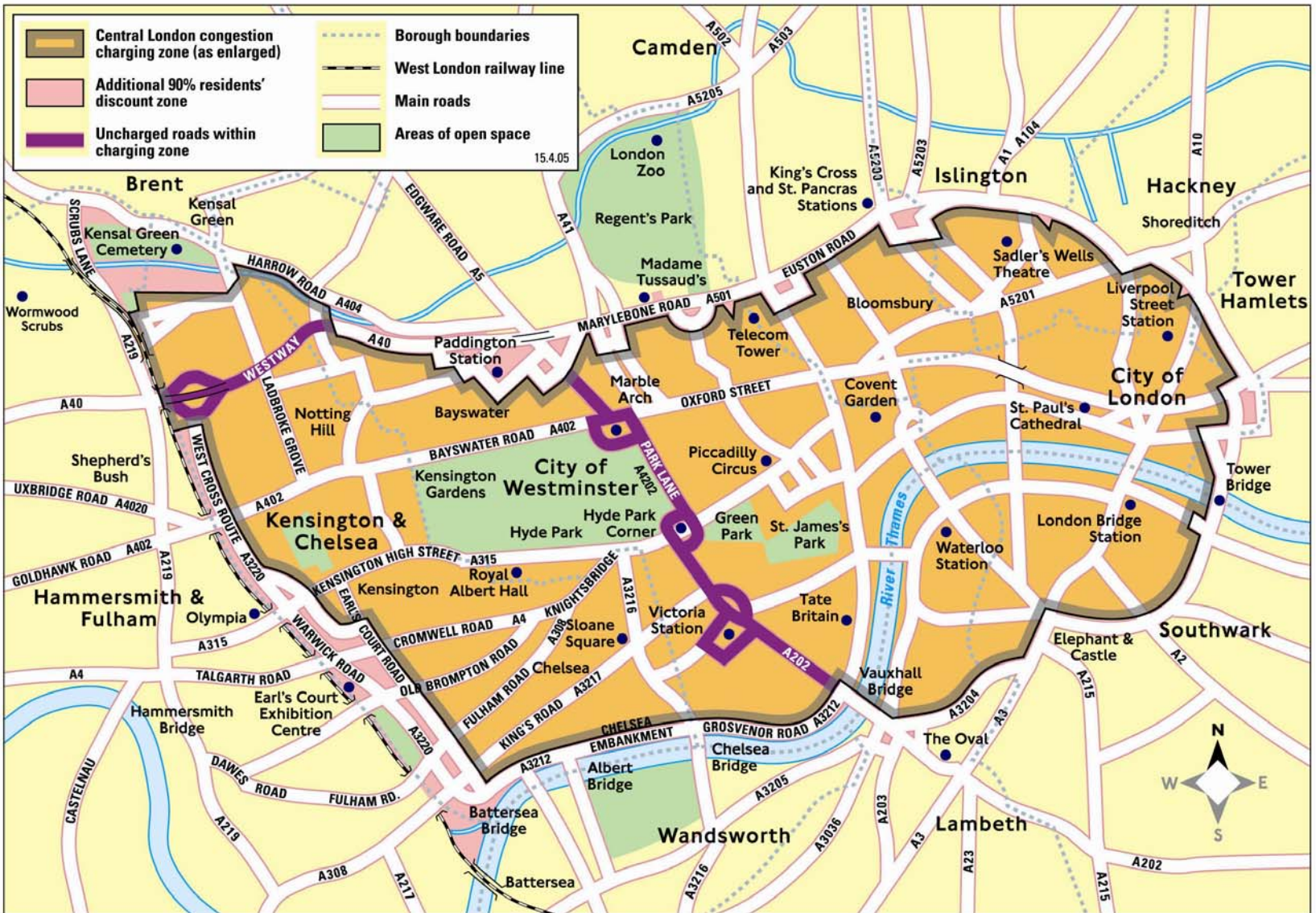
# Background



- Went live in 2003 then extended in 2007
  - More than 100,000 people pay the charge each day.
  - Reduction in vehicles entering the zone of c15%
  - Net revenue of £137m reinvested in London's transport
  - Enforced by a network of more than 1000 cameras.

- Low Emission Zone started last year.
  - Aim is to deter the most polluting heavy diesel vehicles from driving within Greater London.
  - Affected vehicles must meet a minimum emissions standard of Euro III for Particulate Matter.
  - Compliance to date is high with most affected vehicles either fitting abatement devices or diverting from London.





## Original implementation was successful

***“Probably the one shining example of a 'big bang' implementation that went okay was congestion charging in London which was 'big bang'. Yes, there were some teething problems, but compared to other 'big bangs' in the public sector it was a massive success“***

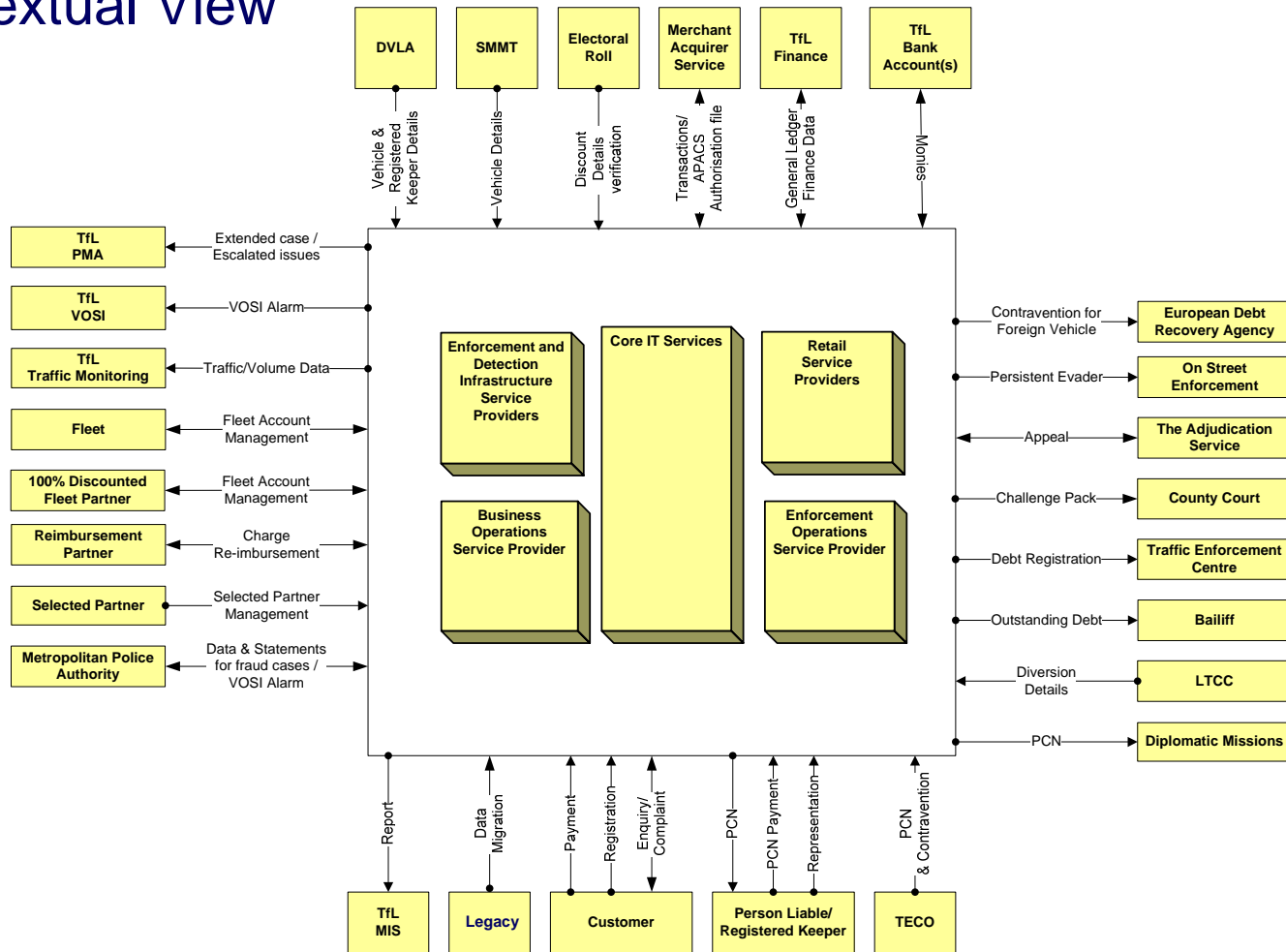
**Sir Peter Gershon CBE**

**Minutes of Evidence taken before the Public Administration Select Committee,  
Civil Service effectiveness 3 February 2005.**

# The case for a more architected approach

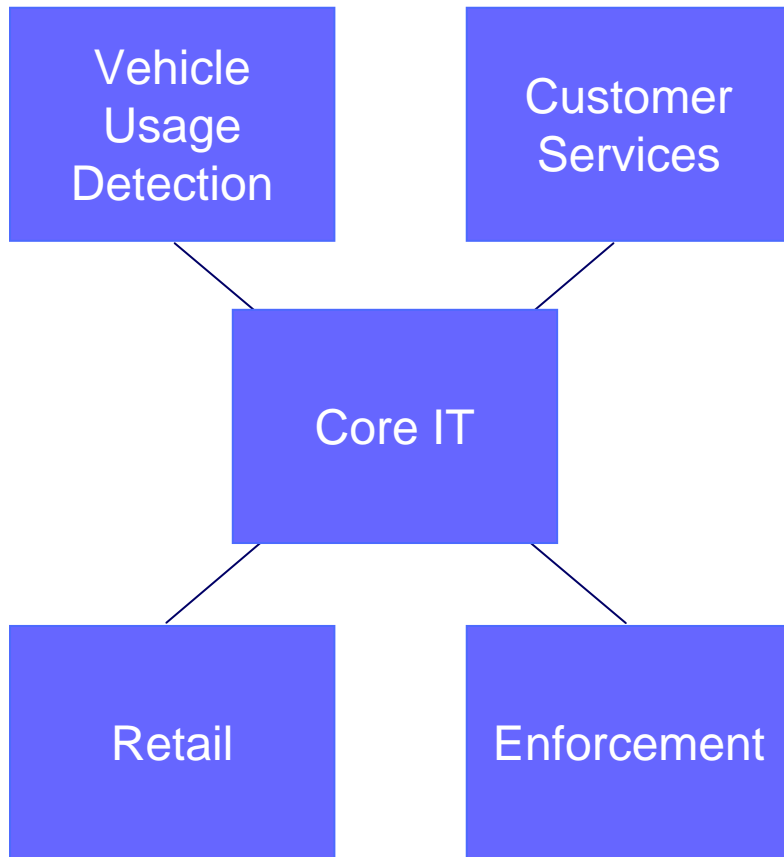
- Original contracts were mostly output based with a single supplier:
  - Had to give suppliers maximum flexibility to meet **aggressive implementation time** line
  - Wanted suppliers to own and manage the **delivery risks**
- However as the scheme evolved TfL wanted to be more control of the service and the solution:
  - To reduce tie in to a single supplier **reducing cost** of change
  - The future plans became more uncertain (politically driven) demanding a **more flexible** solution
  - Experienced business users had increasingly sophisticated requirements for the provision of services and wanted **more control**
  - Potential for **shared services**

# The Contextual View



Re-let Solution Architecture		<b>Key</b> Third party	
Author: Technology & Commercial Paul Johnson	Version: 003	View: Context Diagram	Release / option: Baseline

# High level solution elements – The course grained services



- Vehicle Usage Detection
  - Systems for detecting an individual's usage of the scheme.
- Retail
  - Suppliers to provide a network of convenient payment facilities.
- Customer Services
  - Systems for registration, enquiry and/or complaint handling, payments, billing and/or account management.
- Enforcement
  - Systems to chase those who fail to comply.
- Core IT
  - Provides custom functionality for pricing and identifying contraventions.



# Services catalogue

S01: AccountService  
S02: ChargingEngineService  
S03: ContraventionCandidateService  
S04: CustomerCommunicationService  
S05: PaymentAccountService  
S06: DetectionEventService  
S07: EnforcementService  
S08: ERService  
S09: MessageService  
S10: MISService  
S11: PaymentService  
S12: RejectedContraventionCandidateService  
S13: ShadowPaymentAccountService  
S14: SubSystemStatusService  
S15: TagAnomalyService  
S16: TagVRMMappingService  
S17: ValidationService  
S18: VehicleListManagementService  
S19: VehicleLookupService  
S20: VoISIService  
S21: PartnerAccountService  
S22: ChargingEvidenceService  
S23: AssociatedVehicleService  
S24: AssociatedTagService  
S25: DiscountService  
S26: ImageService  
S27: AuthoritativeTagVRMMappingService  
S28: AuthoritativeDSRCActionListManagementService  
S29: DSRCActionListManagementService  
S30: AuthoritativeVehicleListManagementService  
S31: ProxyPaymentService

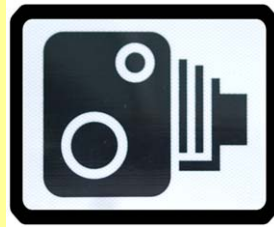
- Solution architecture defined 31 services.
- Each Solution Element implements a subset of these services.
- Each service implements one or more operations. 130 defined in the solution architecture.
- For each operation the data types consumed and produced are defined. We have 278 of these..
- The usage to be implemented is then defined. We specified 50 interactions.

# Basic Congestion Charge process through to contravention

## Vehicle Usage Detection



Vehicle enters zone



Vehicle captured on camera



Automatic number plate recognition technology used to identify vehicle

Customer pays the charge by phone, by SMS text message, at retail outlets or by post



VRM added to database



## Customer Service

## Core IT

Image matched to payment



# Service definition

## **S06: DetectionEventService**

Implemented by: Core

Used by: Detection and Enforcement Infrastructure Solution

Element to inform the Core Solution Element of Vehicles detected by the road-side infrastructure.

<b>Interface ID</b>	<b>Operation ID</b>	<b>Operation Name</b>
DExCS0606	OP0606	SubmitMatchData
DExCS0607	OP0607	SubmitSessionRecord
DExCS0608	OP0608	SubmitSR
DExCS0609	OP0609	SubmitTagRecord
DExCS0612	OP0612	SubmitSRRejectionResponse

# Operation definition

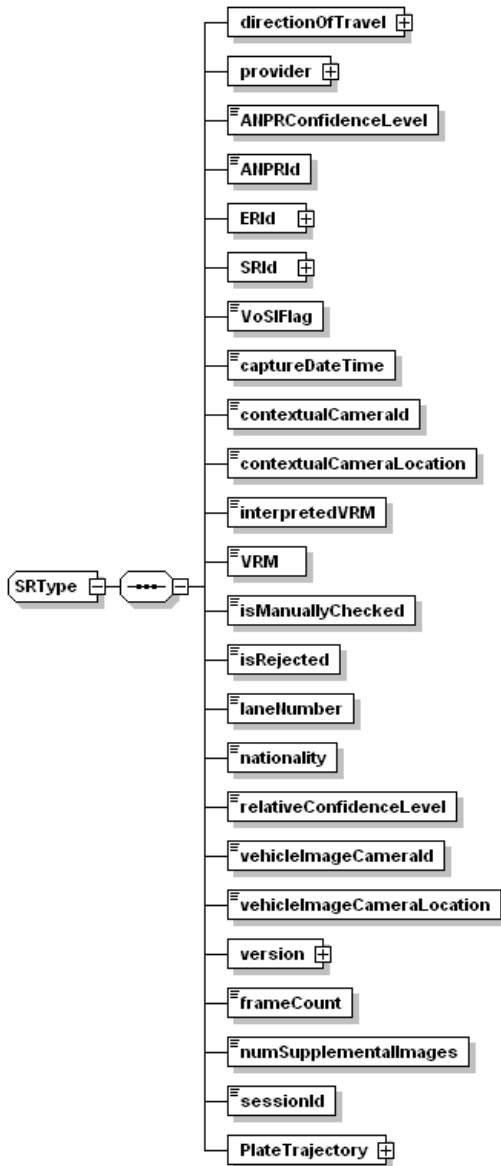
## OP0608: SubmitSR

This Operation is used to submit Summary Records (SRs - text-based summaries of Evidential Records) to the Core Solution Element.

When a Summary Record is first created the initial version number sent to the Core Solution Element shall be "0", and the updated date shall be omitted. The Core Solution Element shall then store the record as version 1 with the modification date set to the time the record was persisted

Operation					
<b>Operation ID</b>	OP0608	<b>Operation Name</b>	SubmitSR		
<b>Expected Usage</b>	The Detection and Enforcement Infrastructure Solution Element is expected to transfer all Summary Records to the Core Solution Element using this Interface.				
<b>Input Data Type ID</b>	D3.08.11	<b>Input Data Type</b>	SRType		
<b>Output Data Type ID</b>	D2.07.01	<b>Output Data Type</b>	MessageBodyType		
<b>Operation Type</b>	A				
Associated Interfaces					
<b>Interface ID</b>	DExCS0608	<b>From</b>	D&EI	<b>To</b>	Core

# Data type definitions



## **SRType (D3.08.11)**

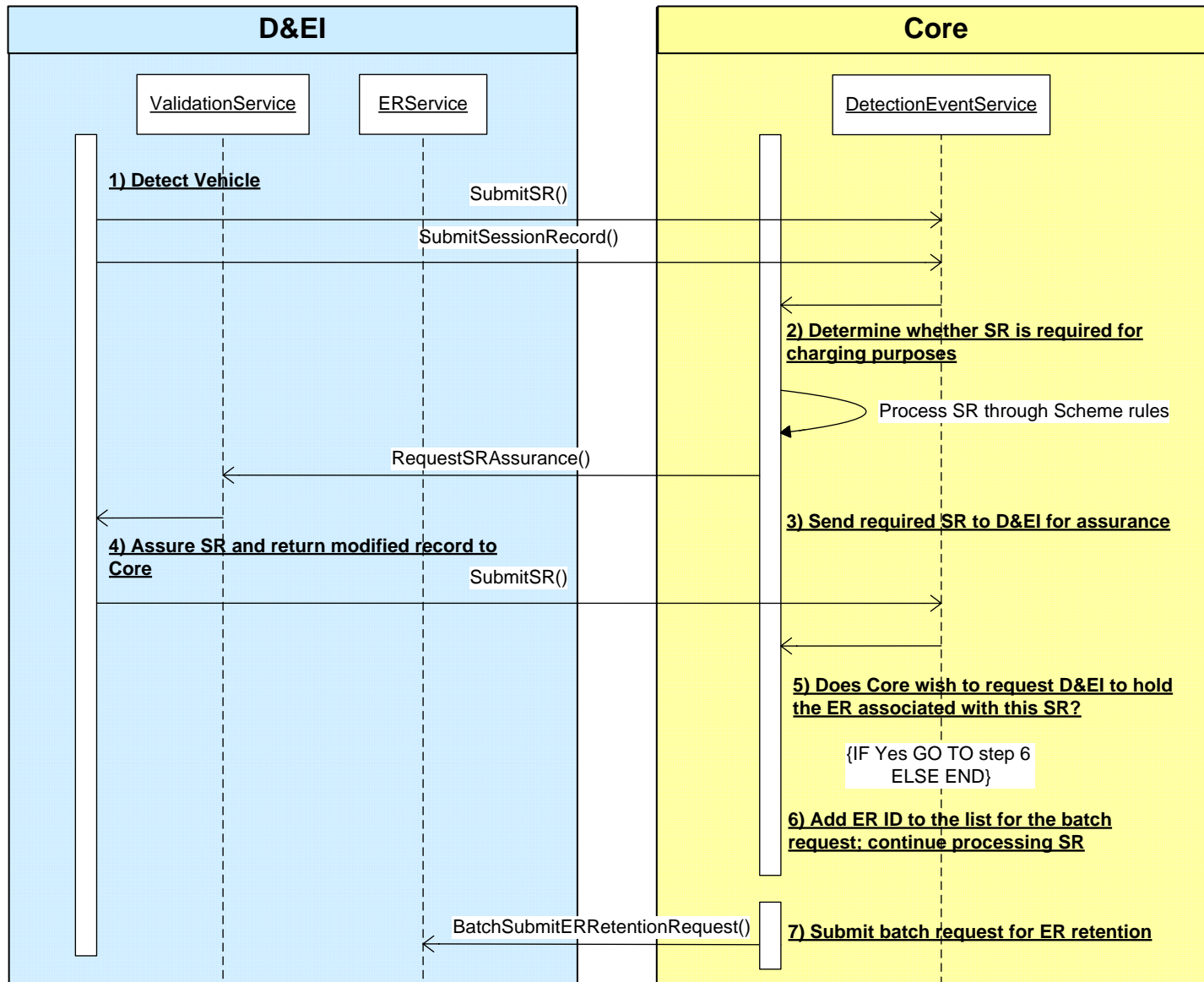
Where a Vehicle is captured by a camera, an SR (Summary Record) is produced. This is a textual summary of the image, together with information about where and when the record was captured and any manual checking the image may have gone through. This Data Type is used by Core to determine whether this detection resulted in any form of Chargeable Item. If the Vehicle is suspected of having contravened any of the Schemes, the ER corresponding to this SR is then requested from the appropriate provider of the D&EI Solution Element for enforcement purposes.

The SRType includes an "interpretedVRM" which is the VRM as interpreted by ANPR technology and a "VRM" field which is the latest working value of the VRM. The "VRM" will be the same "interpretedVRM" initially, however as the SR goes through checking processes, only the "VRM" field will be updated.

The SR includes a version number, which must be incremented each time the record is modified.

AttributeName	Type	Cardinality	Description
directionOfTravel	directionOfTravelEnumeratedType	1..1	Direction in which Vehicle was travelling at the time of detection.
provider	providerEnumeratedType	1..1	The provider of the D&EI Solution Element that submitted this SR.
ANPRConfidenceLevel	string	1..1	The Confidence Level as reported by the ANPR.
ANPRId	string	1..1	The ID of the ANPR unit producing this evidence.
...			

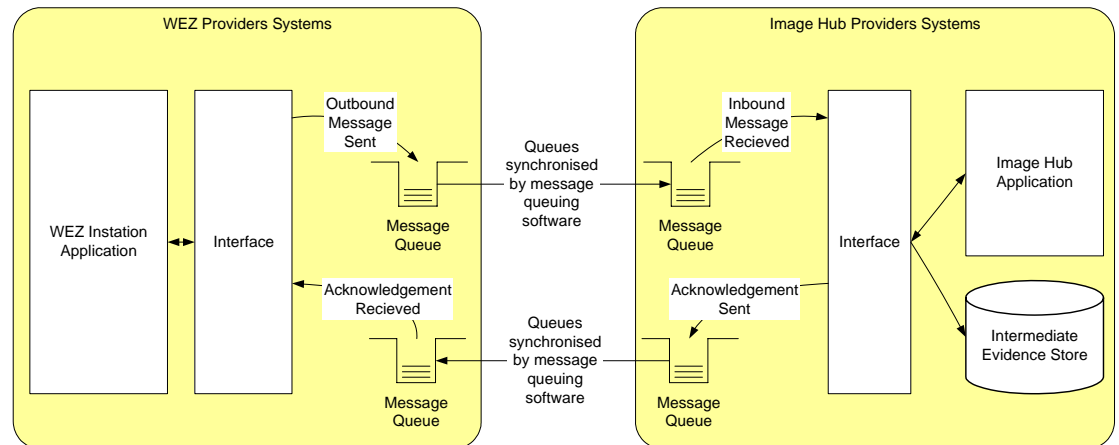
# Sequence diagram



# Pragmatic delivery of services

- Want loose coupling therefore tend towards:
  - Asynchronous message based (MQ ubiquitous)
  - Batch FTP (old fashioned, but still very effective)
- For synchronous (preferably non transactional):
  - Web services

- Message form
  - XML
  - Non proprietary schemas

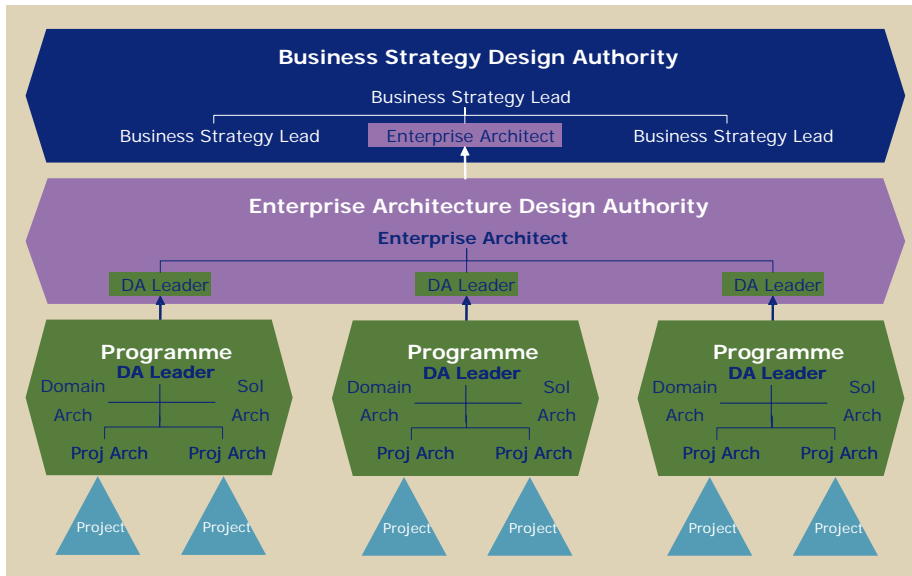


# Service levels

- Need to relate to business need:
  - Is 24 x7 and 99.9% availability really necessary?
  - Does the service need to provide sub second response?
  - Don't measure servers but services.
  - Must be measurable.
  
- Understand implications of service failures across provider boundaries:
  - Who ultimately takes the consequences?
  - Set penalties proportional to impact and/or cost of delivering the service
  - Create margin in service levels between provider and consumer of services



# Governance structure

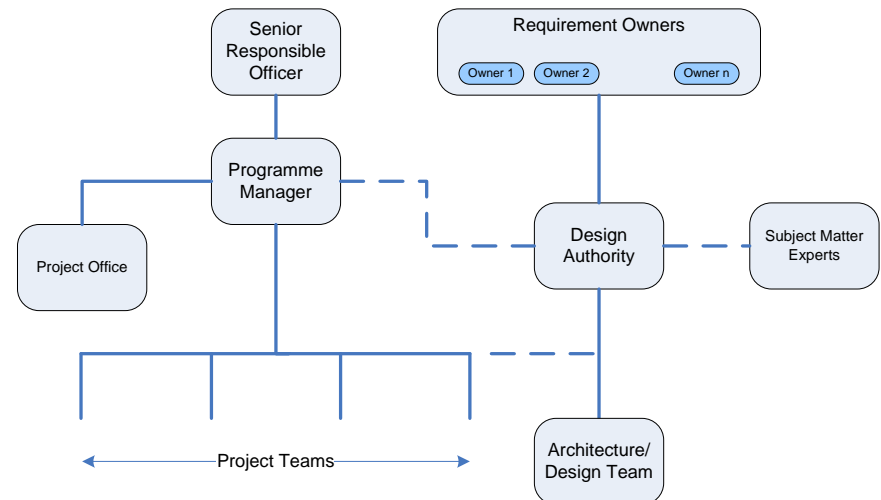


## Typical Problem:

- Solution crossed organisation boundaries
- No end-to-end owner
- Many experts in areas of the solution

## Design Authority:

- Appoint an individual to be the Design Authority
- Established group of requirement owners
- Consult subject matter experts but avoid design by committee



# Lessons learned

- Must have **design authority responsible for end-to-end solution**. (Both technical and business)
- Must establish **clear governance** and responsibilities. Include all interested parties to get buy-in and have escalation process to make decisions where necessary.
- Nailing down requirements is not possible but identifying a **range of scenarios** to architect for is.
- **Minimise likely change impact** through defining services boundaries which encapsulate business logic likely to change or to enable change through differing assembly of services.
- Need **clean well defined interfaces** with well defined and **measurable service levels** between any separable solution elements.
- Need to ensure **shared understanding** through good communication and clear architectural models and artefacts.
- Need to produce **detailed specification of service interfaces** and usage in Architecture/High Level Design phases.

# Questions?

Email: [mburnham@deloitte.co.uk](mailto:mburnham@deloitte.co.uk)

**Deloitte.**