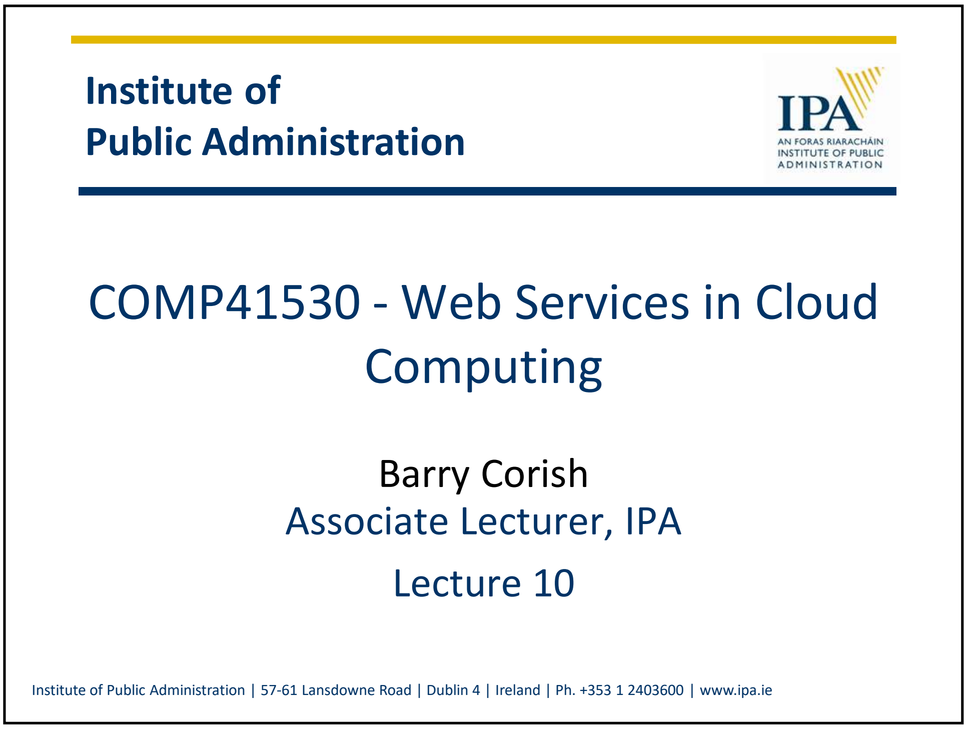


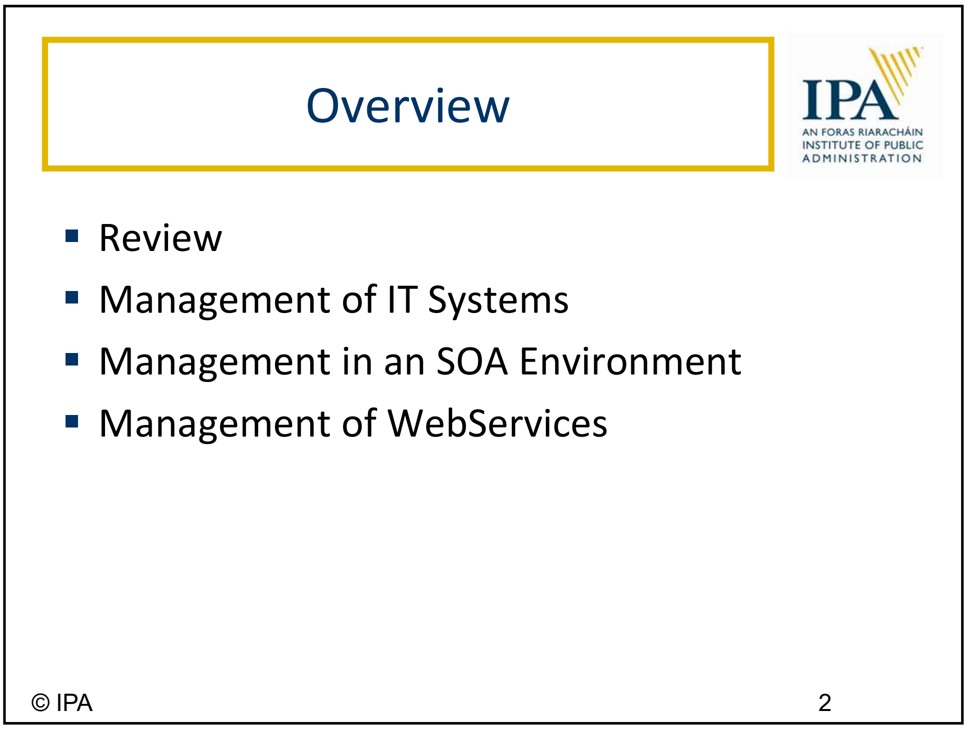
Institute of
Public Administration

COMP41530 - Web Services in Cloud Computing

Barry Corish
Associate Lecturer, IPA

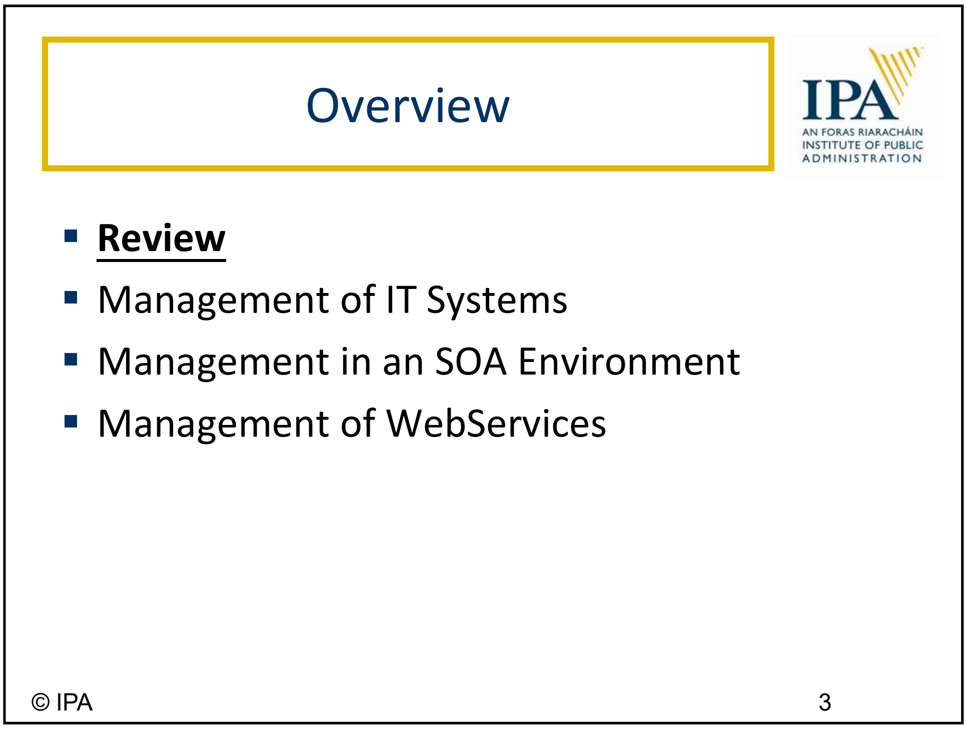
Lecture 10

Institute of Public Administration | 57-61 Lansdowne Road | Dublin 4 | Ireland | Ph. +353 1 2403600 | www.ipa.ie



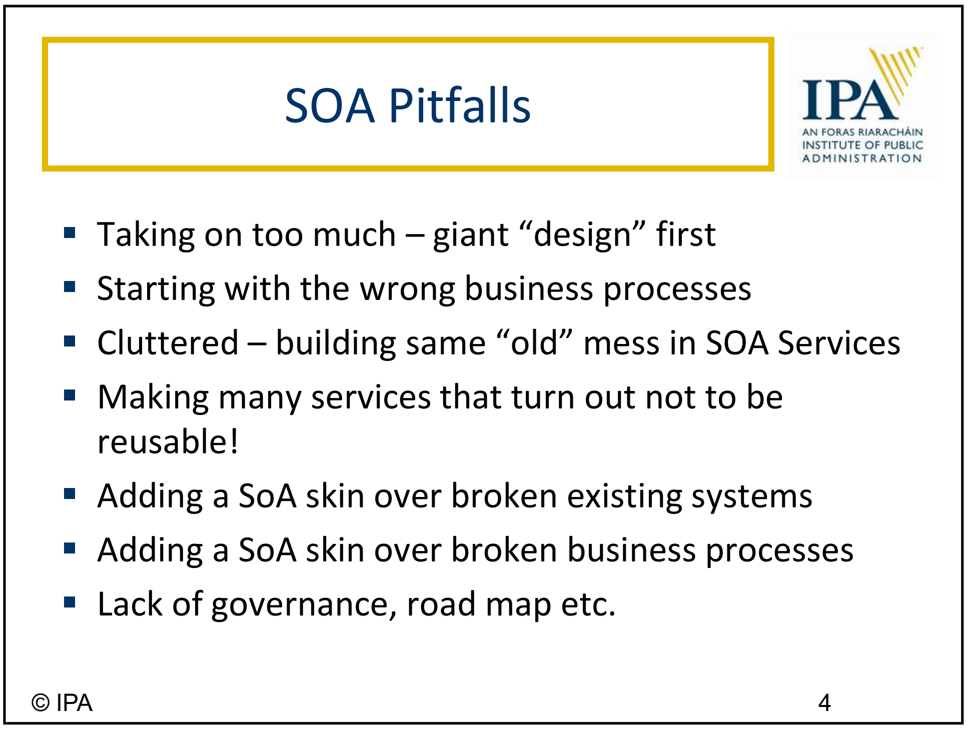
Overview

- Review
- Management of IT Systems
- Management in an SOA Environment
- Management of WebServices



Overview

- **Review**
- Management of IT Systems
- Management in an SOA Environment
- Management of WebServices



SOA Pitfalls

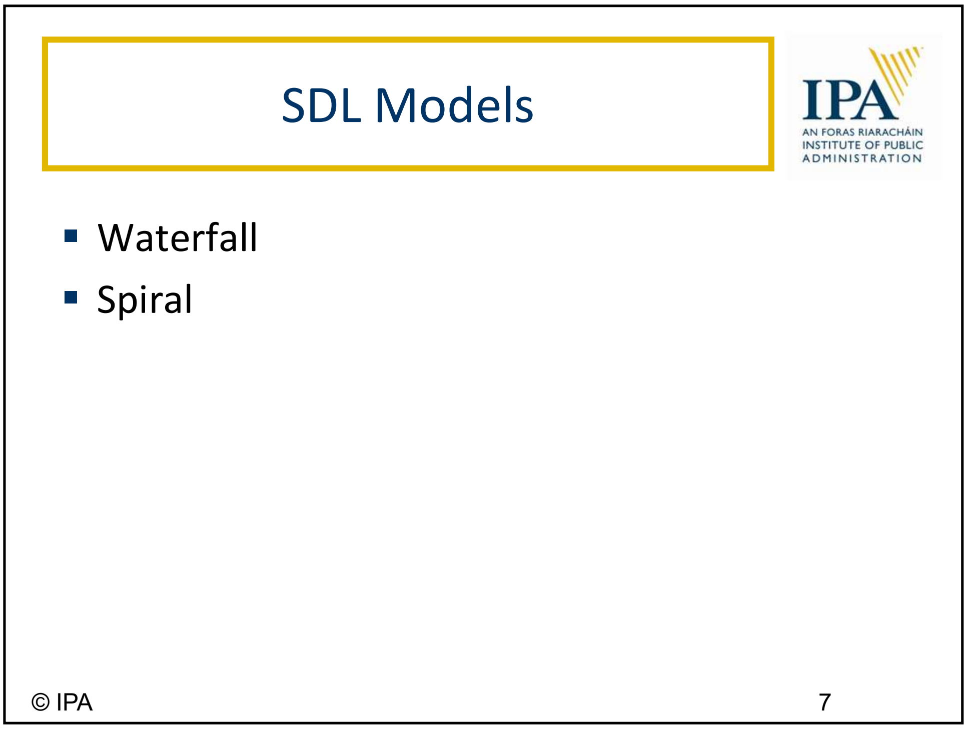
- Taking on too much – giant “design” first
- Starting with the wrong business processes
- Cluttered – building same “old” mess in SOA Services
- Making many services that turn out not to be reusable!
- Adding a SoA skin over broken existing systems
- Adding a SoA skin over broken business processes
- Lack of governance, road map etc.

SOA Development Livecycle

- How to start? First questions:
  - What are the core business processes?
  - What services form part of these processes?
  - Which of these services will add immediate value?
  - How can these services be assembled into useful business processed?
  - Are the business processes you are going to work on working correctly before you start?

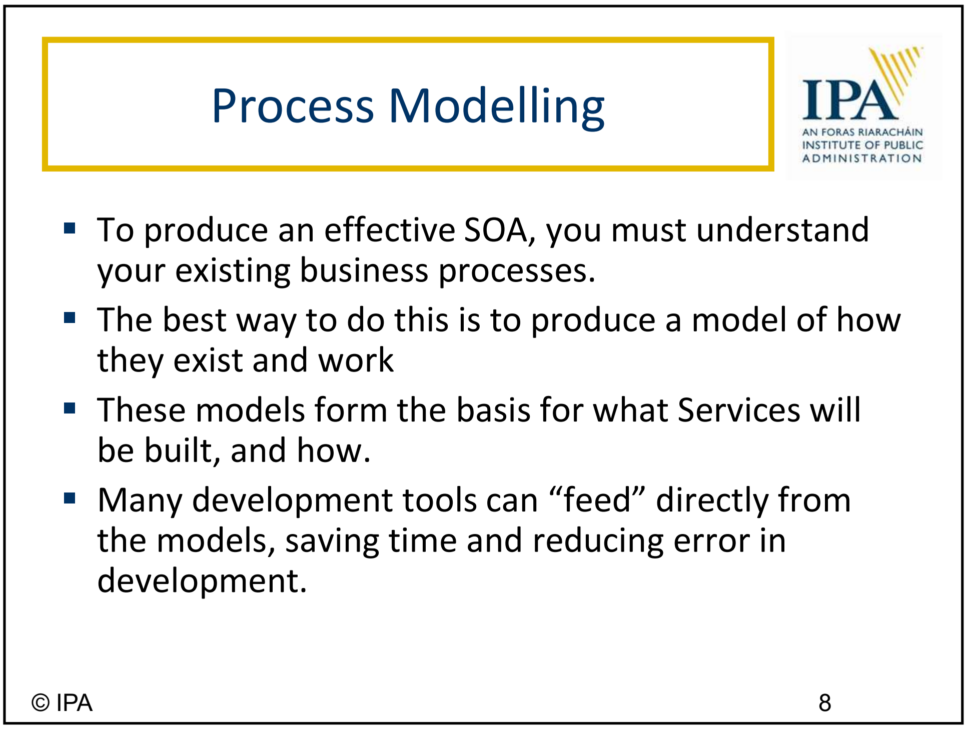
Software Development Livecycle

- Process of designing, building and implementing software systems:
  - Analysis
  - Design
  - Implementation
  - Maintenance



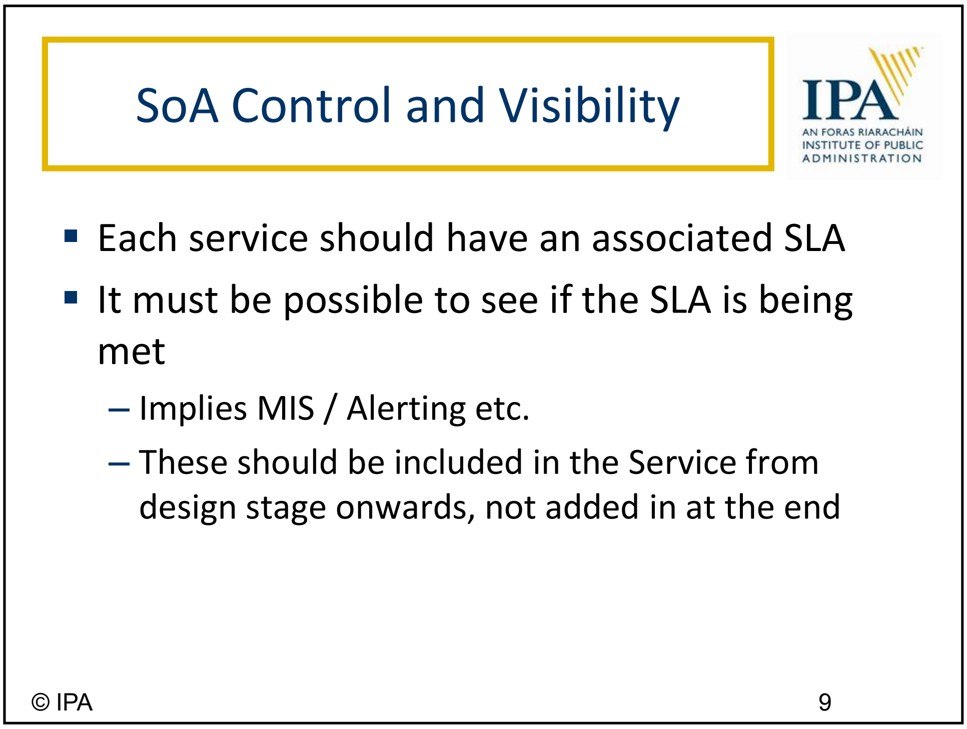
SDL Models

- Waterfall
- Spiral



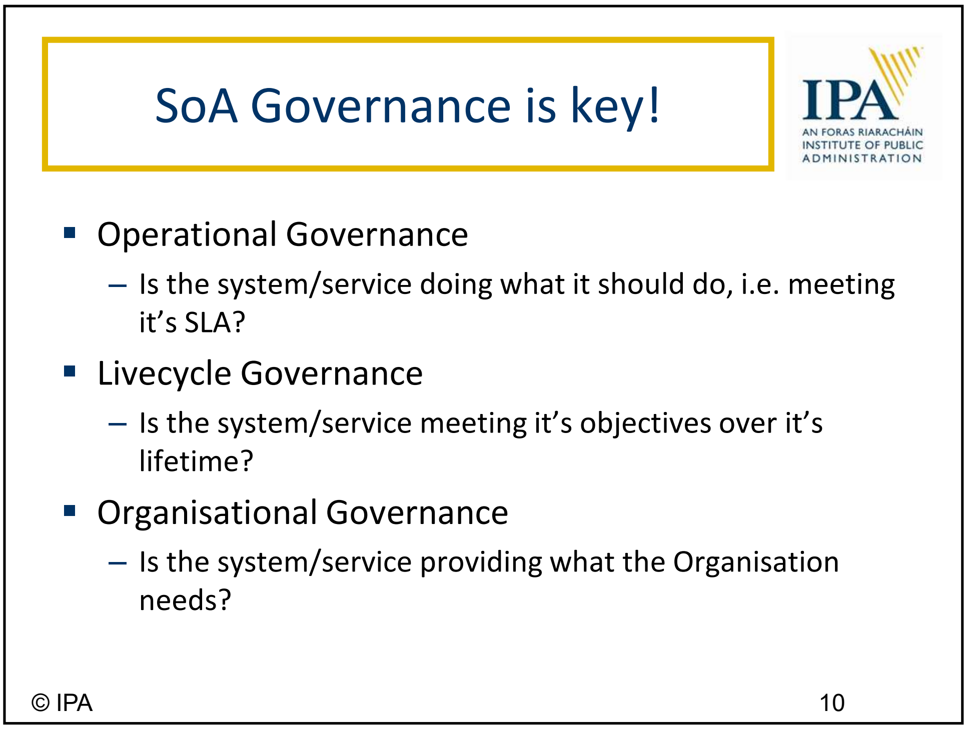
Process Modelling

- To produce an effective SOA, you must understand your existing business processes.
- The best way to do this is to produce a model of how they exist and work
- These models form the basis for what Services will be built, and how.
- Many development tools can “feed” directly from the models, saving time and reducing error in development.



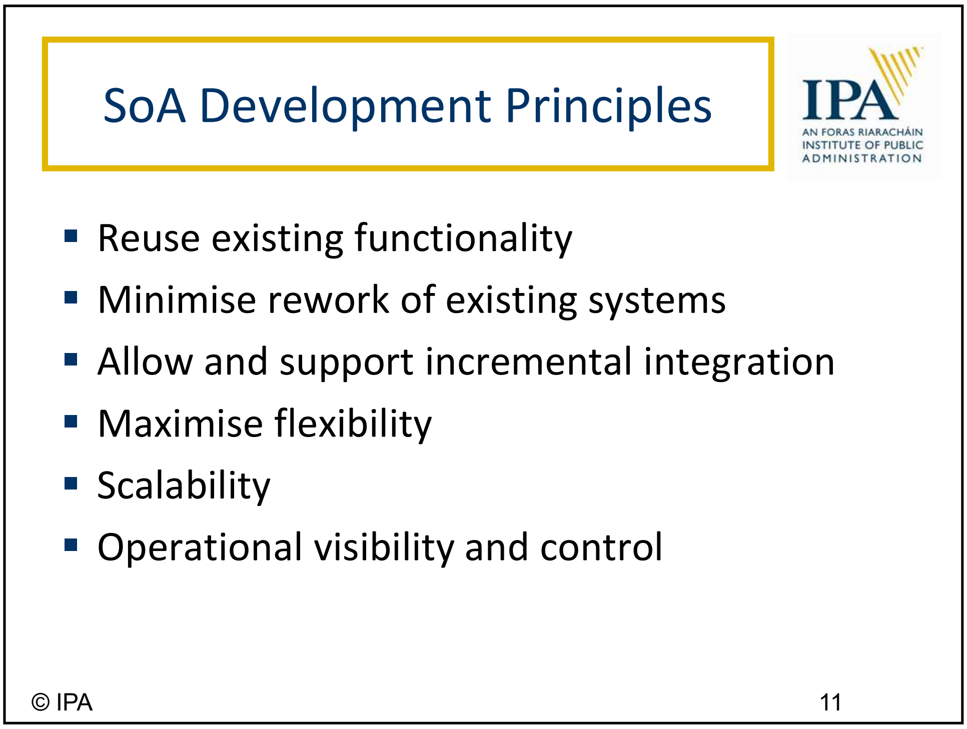
SoA Control and Visibility

- Each service should have an associated SLA
- It must be possible to see if the SLA is being met
  - Implies MIS / Alerting etc.
  - These should be included in the Service from design stage onwards, not added in at the end



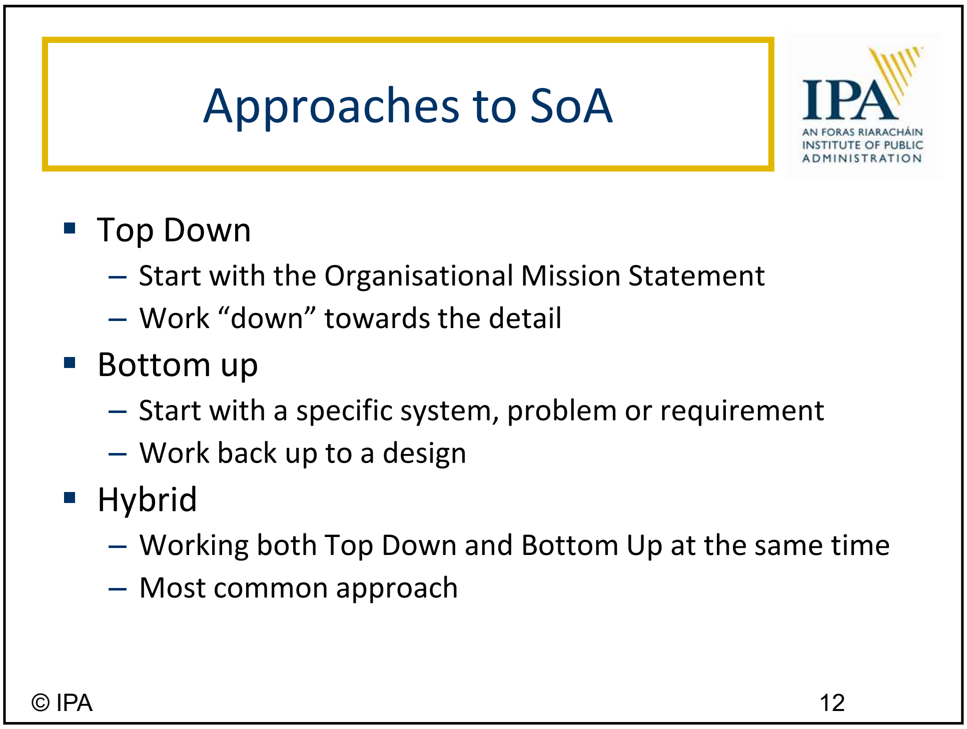
SoA Governance is key!

- Operational Governance
  - Is the system/service doing what it should do, i.e. meeting it’s SLA?
- Livecycle Governance
  - Is the system/service meeting it’s objectives over it’s lifetime?
- Organisational Governance
  - Is the system/service providing what the Organisation needs?



SoA Development Principles

- Reuse existing functionality
- Minimise rework of existing systems
- Allow and support incremental integration
- Maximise flexibility
- Scalability
- Operational visibility and control



Approaches to SoA

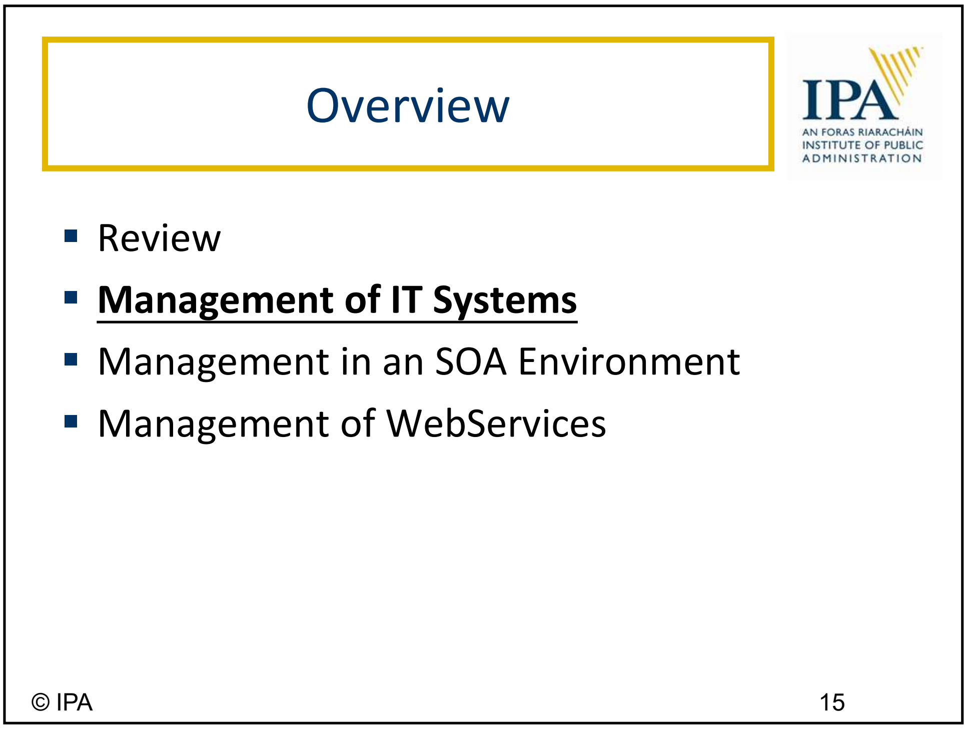
- Top Down
  - Start with the Organisational Mission Statement
  - Work “down” towards the detail
- Bottom up
  - Start with a specific system, problem or requirement
  - Work back up to a design
- Hybrid
  - Working both Top Down and Bottom Up at the same time
  - Most common approach

Main Steps in SOA Development

- Plan
- Analyse
- Design
- Build
- Test
- Provision
- Deploy
- Manage
- Govern

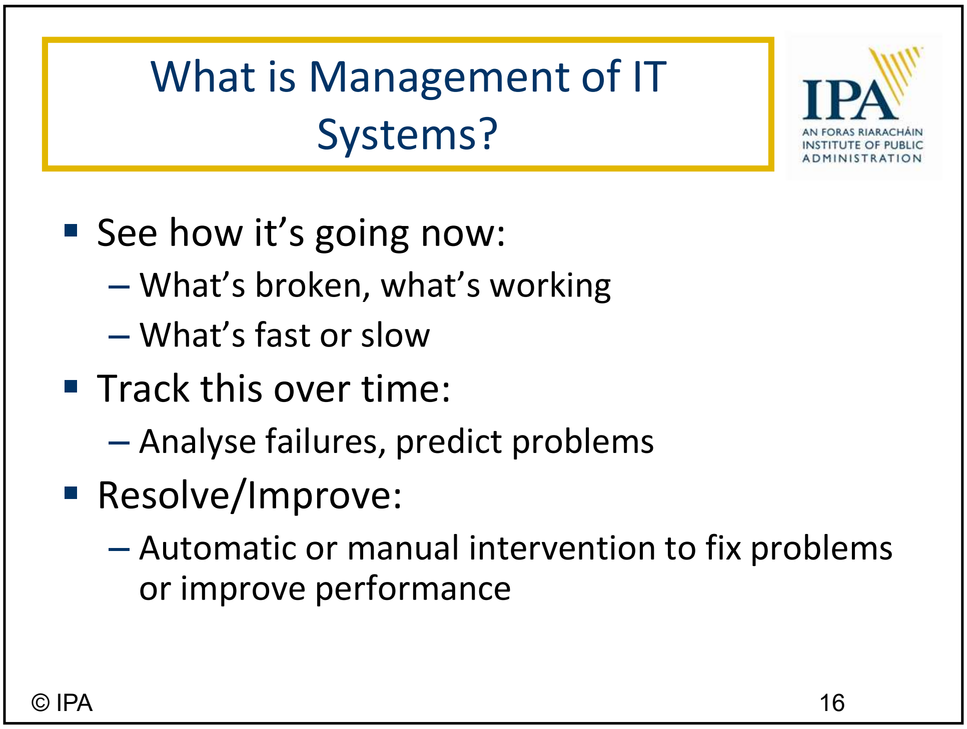
Key points

- SoA is as much a Business methodology as a set of technologies
- SoA must be implemented within a plan and a methodology to avoid chaos
- Governance is critical!
- SoA != WebServices



Overview

- Review
- **Management of IT Systems**
- Management in an SOA Environment
- Management of WebServices



What is Management of IT Systems?

- See how it’s going now:
  - What’s broken, what’s working
  - What’s fast or slow
- Track this over time:
  - Analyse failures, predict problems
- Resolve/Improve:
  - Automatic or manual intervention to fix problems or improve performance

Management of Legacy Systems

- Architecturally Simpler Systems
- Highly staffed / high skill levels
- Generally limited change in infrastructure
- More centralised
- Fewer components
- Easier to manage?

Problem investigation in Legacy Systems

- Is it working?
  - Sign on, try it...
- If not, what's wrong?
  - "Unified Console"
- Tracking system resources, failures etc.
- Very highly evolved management infrastructure!

Managing Distributed Systems

- Distributed Computing Systems
- Loose Coupling
- Heterogeneous Computing Systems
  - Implies different Development and Infrastructural teams
- All the stuff we love…
  - …but much harder to Manage!

You know you have a distributed system when…

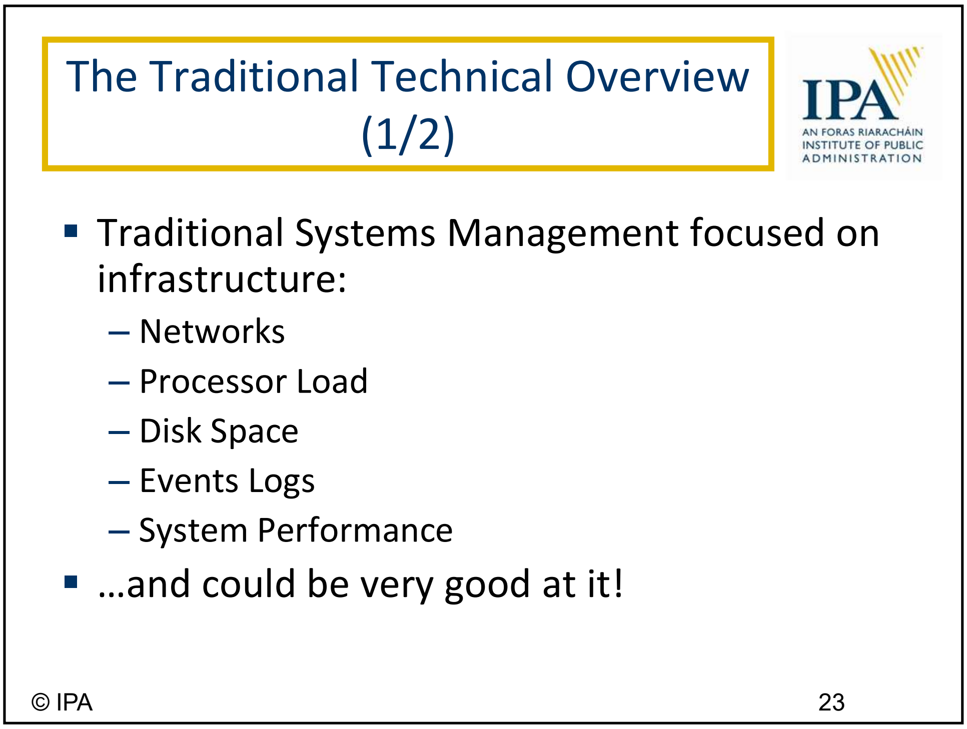
- You know you have a distributed system when the crash of a computer you've never heard of stops you from getting any work done.
  - Leslie Lamport

Distributed Management Systems (1/2)

- The answer?
  - Promises to be just that
  - Designed to address these problems
- Big promises…
  - Big money!
  - Requires management/staff/expertise etc. on and ongoing basis

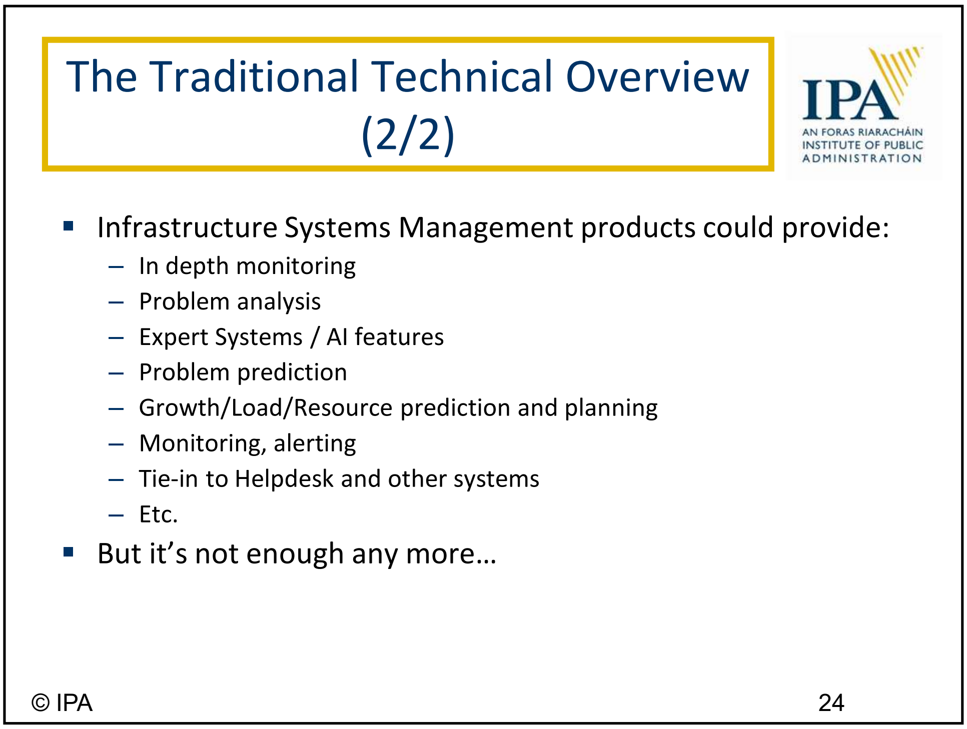
Distributed Management Systems (2/2)

- 3 aspects of Management:
- Monitor
  - Capture and record stats. and events over time
- Track
  - Watch single "units of work" flow through the system
- Control
  - Change and manage the system while live



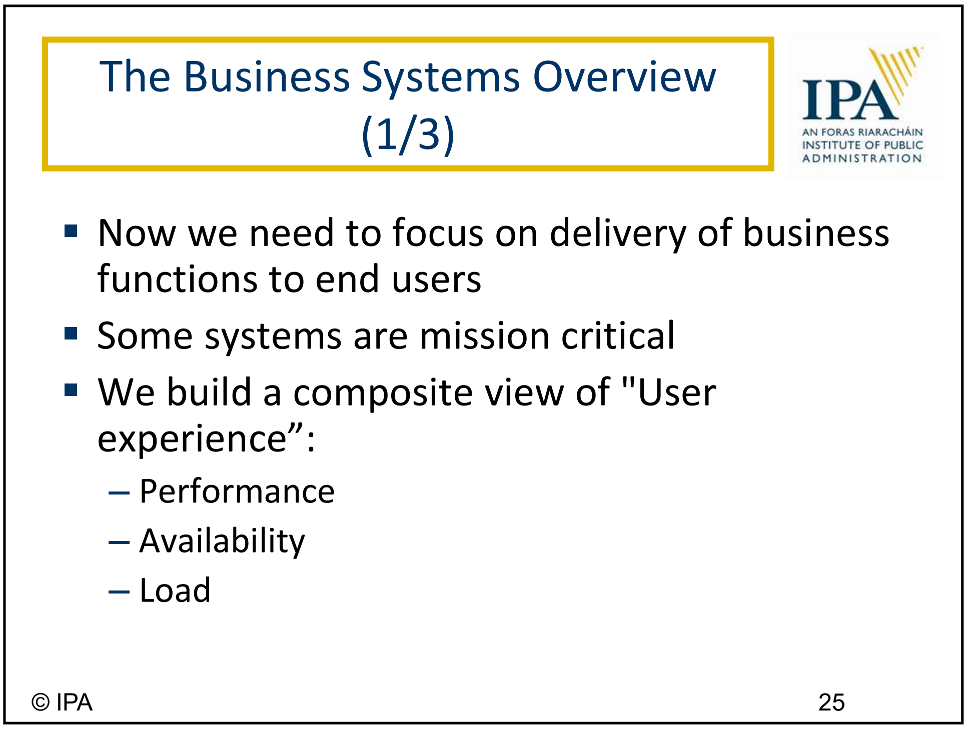
The Traditional Technical Overview (1/2)

- Traditional Systems Management focused on infrastructure:
  - Networks
  - Processor Load
  - Disk Space
  - Events Logs
  - System Performance
- …and could be very good at it!



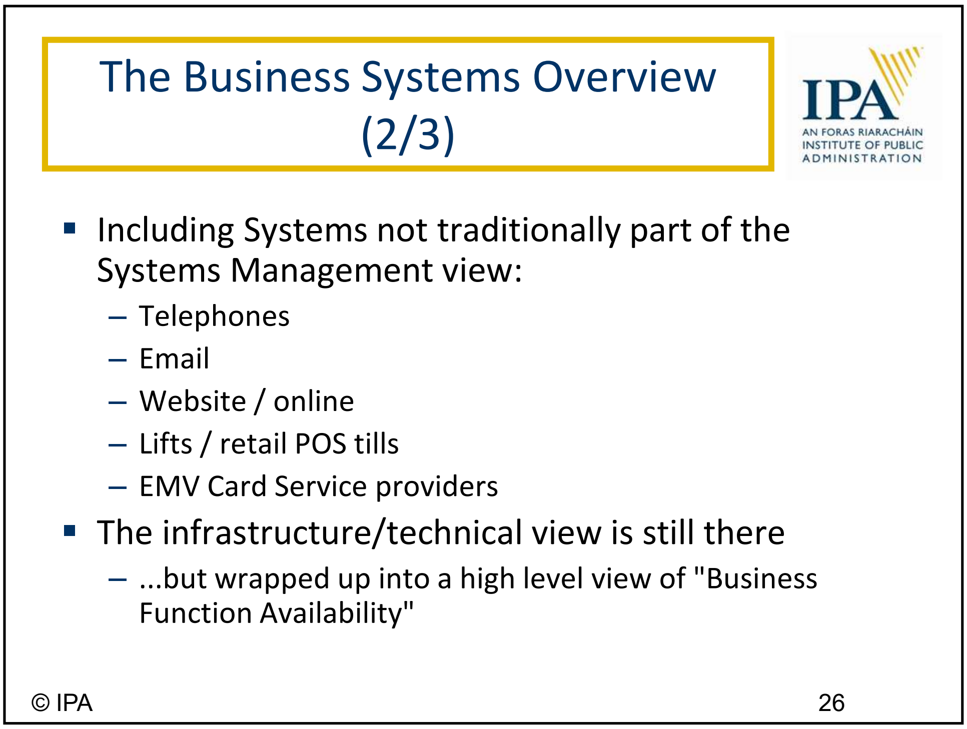
The Traditional Technical Overview (2/2)

- Infrastructure Systems Management products could provide:
  - In depth monitoring
  - Problem analysis
  - Expert Systems / AI features
  - Problem prediction
  - Growth/Load/Resource prediction and planning
  - Monitoring, alerting
  - Tie-in to Helpdesk and other systems
  - Etc.
- But it’s not enough any more…



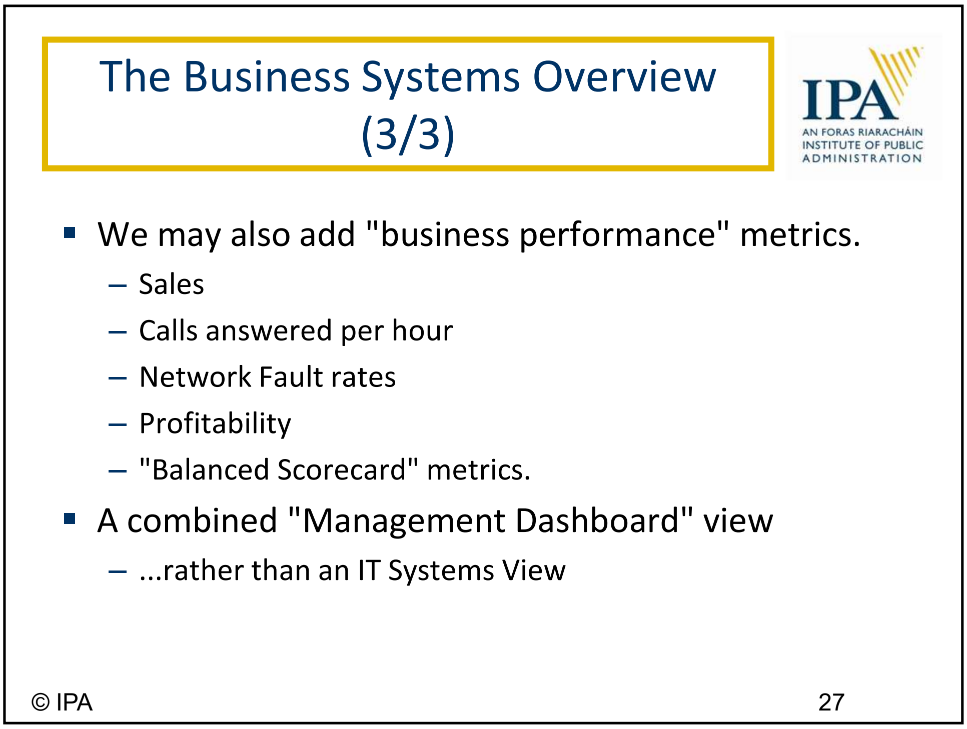
The Business Systems Overview (1/3)

- Now we need to focus on delivery of business functions to end users
- Some systems are mission critical
- We build a composite view of "User experience”:
  - Performance
  - Availability
  - Load



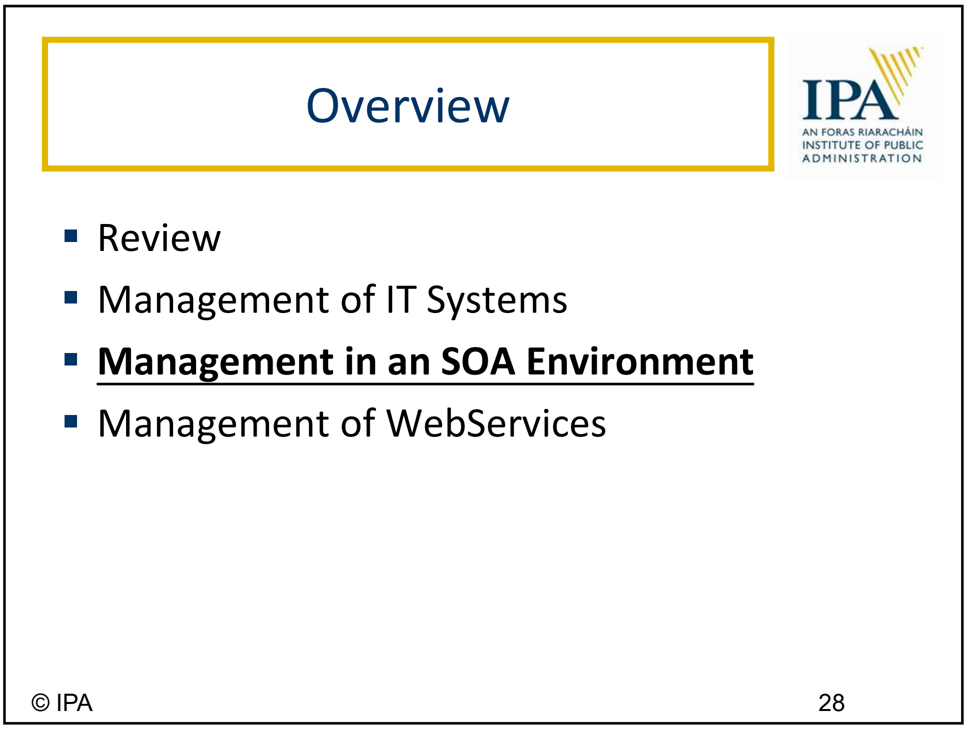
The Business Systems Overview (2/3)

- Including Systems not traditionally part of the Systems Management view:
  - Telephones
  - Email
  - Website / online
  - Lifts / retail POS tills
  - EMV Card Service providers
- The infrastructure/technical view is still there
  - ...but wrapped up into a high level view of "Business Function Availability"



The Business Systems Overview (3/3)

- We may also add "business performance" metrics.
  - Sales
  - Calls answered per hour
  - Network Fault rates
  - Profitability
  - "Balanced Scorecard" metrics.
- A combined "Management Dashboard" view
  - ...rather than an IT Systems View



Overview

- Review
- Management of IT Systems
- **Management in an SOA Environment**
- Management of WebServices

Management of SOA Systems (1/2)

- Is there a problem?
  - Even this can be hard to determine
- What is the problem?
  - Hard to tell...
- What is the cause of the problem?
  - Ummm...

Management of SOA Systems (2/2)

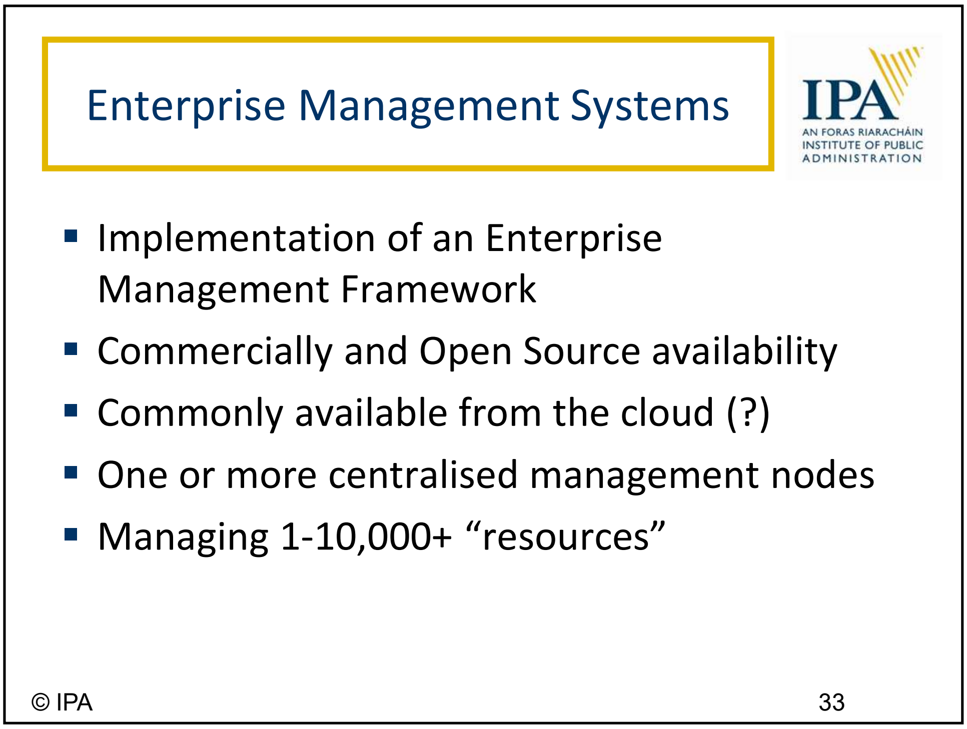
- No single console any more
  - Each system and component has it's own
- No single system overview
- Huge number of points of possible failure
- No single "expert" who can trace problem
  - System is too complex and distributed

Enterprise Management Frameworks (1/2)

- Software Systems
- Designed to manage complex IT Systems
- One or more "Management Servers"
- Trying to centralise and unify:
  - Reporting
  - Logging/recording/analysis
  - Alerting

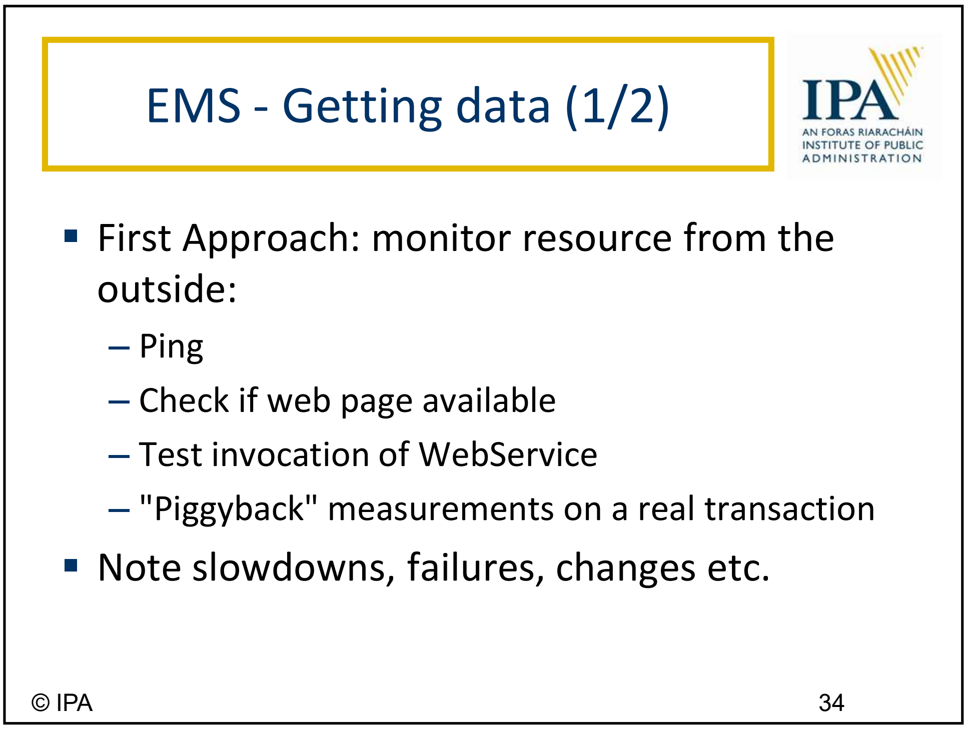
Enterprise Management Frameworks (2/2)

- Trying to manage:
  - Faults/Events
  - Configuration
  - Accounting
  - Performance
  - Security
  - …and many more facets of the systems.



Enterprise Management Systems

- Implementation of an Enterprise Management Framework
- Commercially and Open Source availability
- Commonly available from the cloud (?)
- One or more centralised management nodes
- Managing 1-10,000+ “resources”



EMS - Getting data (1/2)

- First Approach: monitor resource from the outside:
  - Ping
  - Check if web page available
  - Test invocation of WebService
  - "Piggyback" measurements on a real transaction
- Note slowdowns, failures, changes etc.

EMS - Getting data (2/2)

- Second approach: monitor from inside the resource
  - Install an agent on the monitored system
  - Agent "hooks into" various aspects of the system
  - Reports data back to central monitoring server.

EMS Reporting

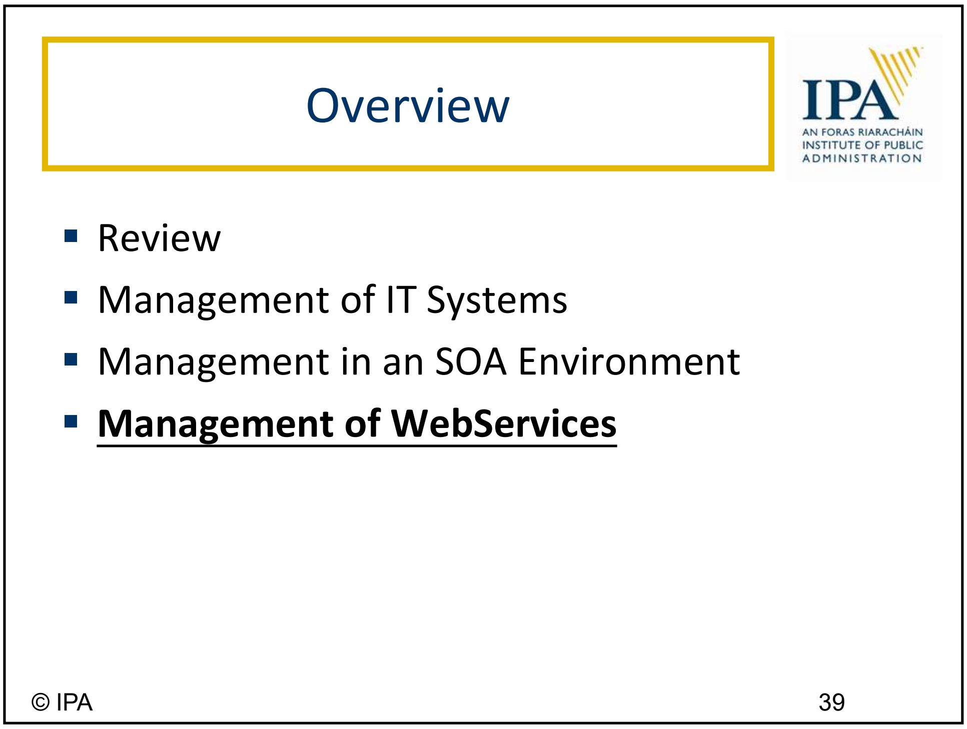
- External and Agent based monitoring are commonly used together
- Data from two sources are merged to give a single view of the resource
- Data from multiple resources combined together to give:
  - “Infrastructural” view, e.g. all LAN components
  - “Business System” view, e.g. every resource required by the Call Centre

Standard Frameworks: SNMP

- **S**imple **N**etwork **M**anagement **P**rotocol
- Agents are internal or external to monitored component.
- Agents perform tests, collects and stores data
- SNMP is a protocol for sending data and results from Agents to Management Systems
- Commonly used:
  - Fine for managing "IT Infrastructure"
  - Limitations in protocol for managing "business systems"

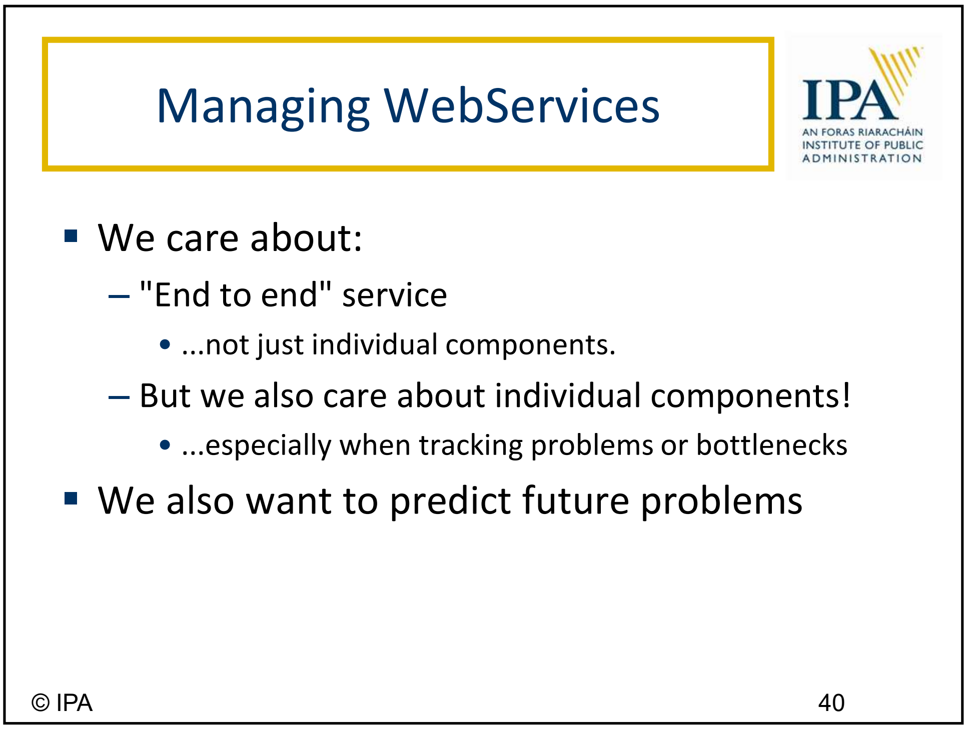
Standard Frameworks: CIM

- **C**ommon **I**nformation **M**odel
- Set of schema for Management Information
- Messaged con be "reporting" or "control"
- Implemented as CIM-XML
- Windows Management Information (WMI) is CIM compliant
- Not taking off (yet?)
  - Too open?
  - Not prescriptive enough?



Overview

- Review
- Management of IT Systems
- Management in an SOA Environment
- **Management of WebServices**



Managing WebServices

- We care about:
  - "End to end" service
    - ...not just individual components.
  - But we also care about individual components!
    - ...especially when tracking problems or bottlenecks
- We also want to predict future problems

Typical Metrics for a WebService (1/2)

- Number of requests received
  - ...number of successes
  - ...number of failures
- Performance
  - Avg processing time
  - Max processing time
  - Min processing time

Typical Metrics for a WebService (2/2)

- We also care about the relationships between these metrics
- What happens to the system under load?
  - Nothing?
  - Gradual slow down?
  - Sudden tipping point?
  - Crash?

Management of WebServices (1/3)

- So, we need to monitor both:
  - User Experience
    - Availability and performance
  - Infrastructure Monitoring:
    - Servers, networks, databases, software

Management of WebServices (2/3)

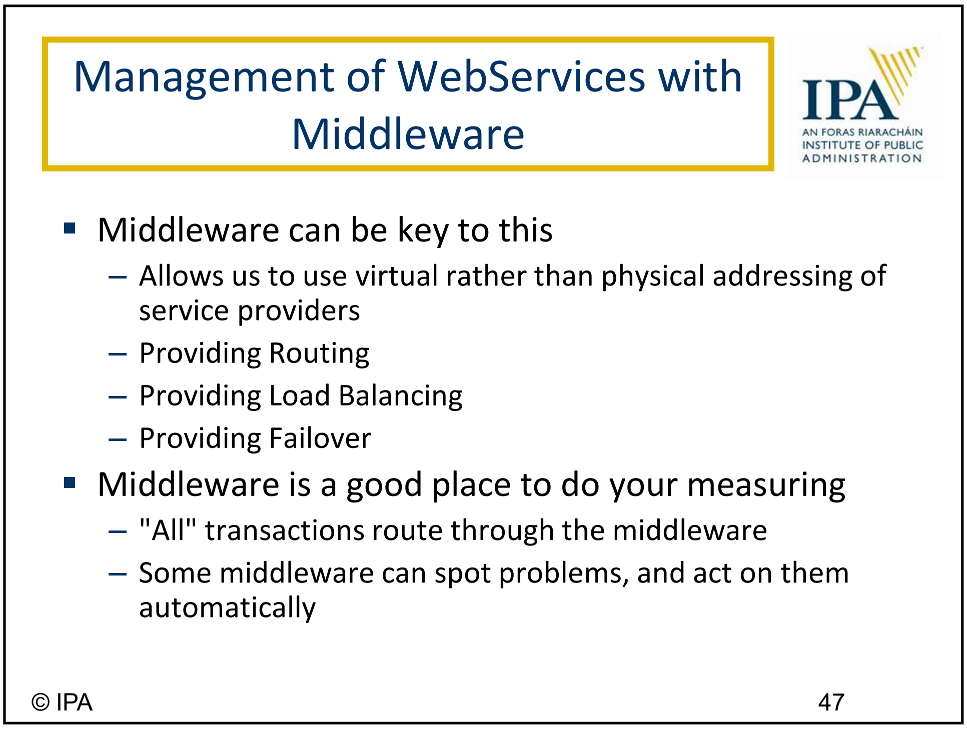
- Resource Provisioning:
  - Particularly in Virtualised environments
  - Feed in more CPU, or cluster members where infrastructure is under load.
- Transaction Monitoring:
  - Throughput, volume, trends
  - Track single transactions

Management of WebServices (3/3)

- All wrapped up into:
  - SLA Monitoring:
    - Are we meeting our SLA?
- From these we need to:
  - Record (and allow analysis)
  - Alert (on exception)
  - Fix (manual or automatic?)

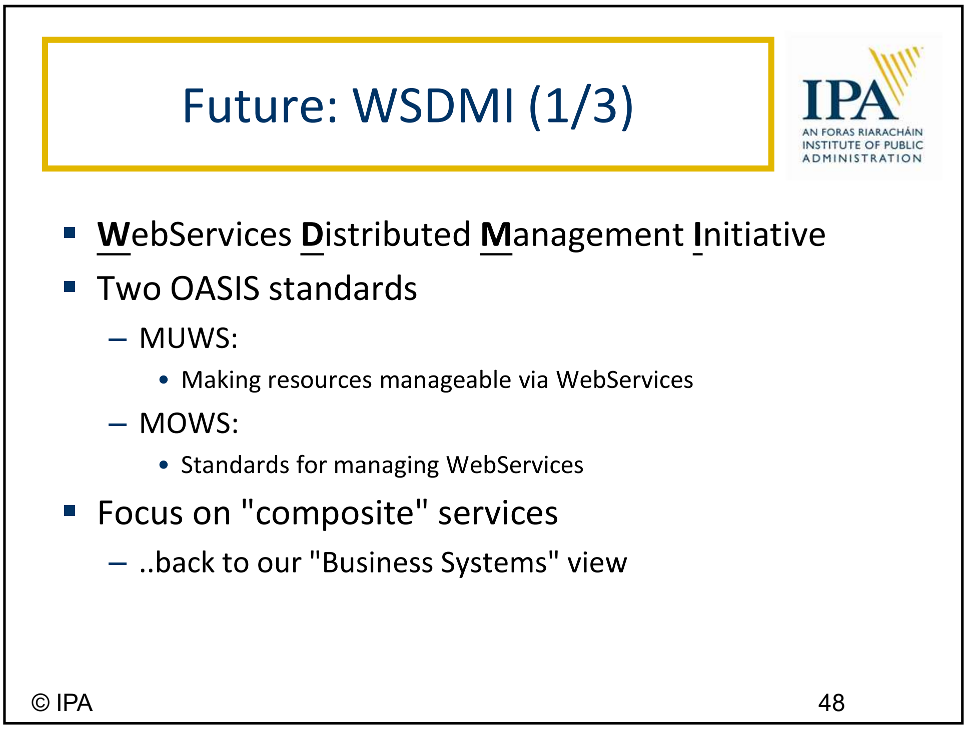
Active Management of WebServices

- We need to be able to:
  - Bring them up or down
  - Balance the load between multiple providers
  - Launch new versions of a service
  - Deprecate old versions of a service
  - Upgrade or enhance versions of a service
- ...all non-disruptively.



Management of WebServices with Middleware

- Middleware can be key to this
  - Allows us to use virtual rather than physical addressing of service providers
  - Providing Routing
  - Providing Load Balancing
  - Providing Failover
- Middleware is a good place to do your measuring
  - "All" transactions route through the middleware
  - Some middleware can spot problems, and act on them automatically



Future: WSDMI (1/3)

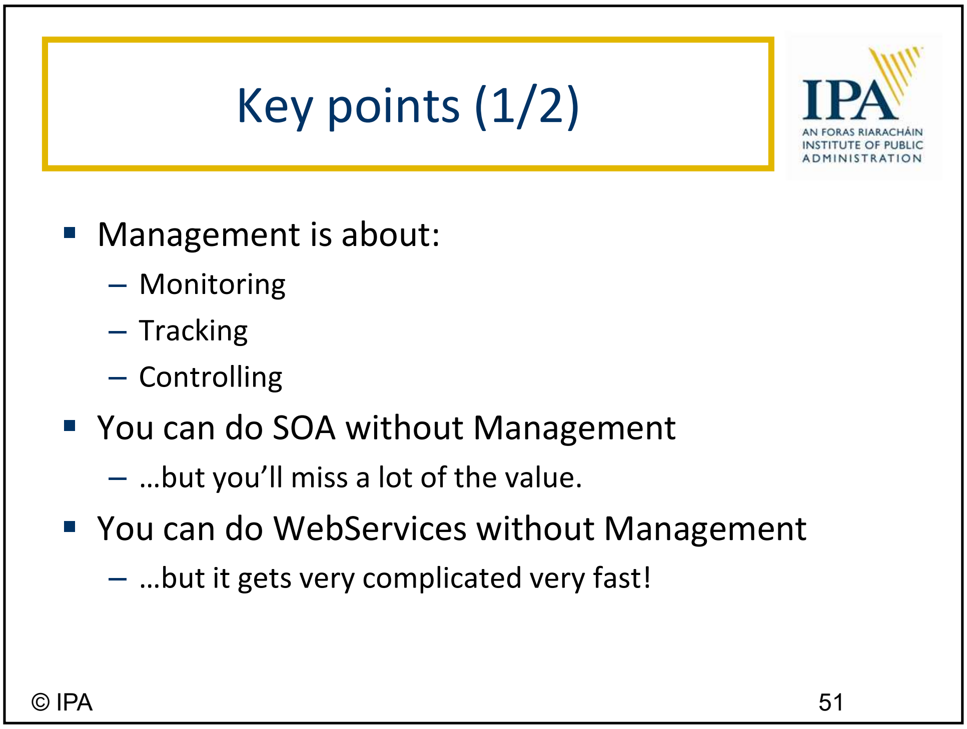
- **W**ebServices **D**istributed **M**anagement **I**nitiative
- Two OASIS standards
  - MUWS:
    - Making resources manageable via WebServices
  - MOWS:
    - Standards for managing WebServices
- Focus on "composite" services
  - ..back to our "Business Systems" view

Future: WSDMI (2/3)

- All based on WebServices
  - All based on SOAP
- Uses existing SOAP based schemas:
  - WS-Notification
  - WS-Topics
  - etc.
- Specifies more schemas on top / beside these existing ones

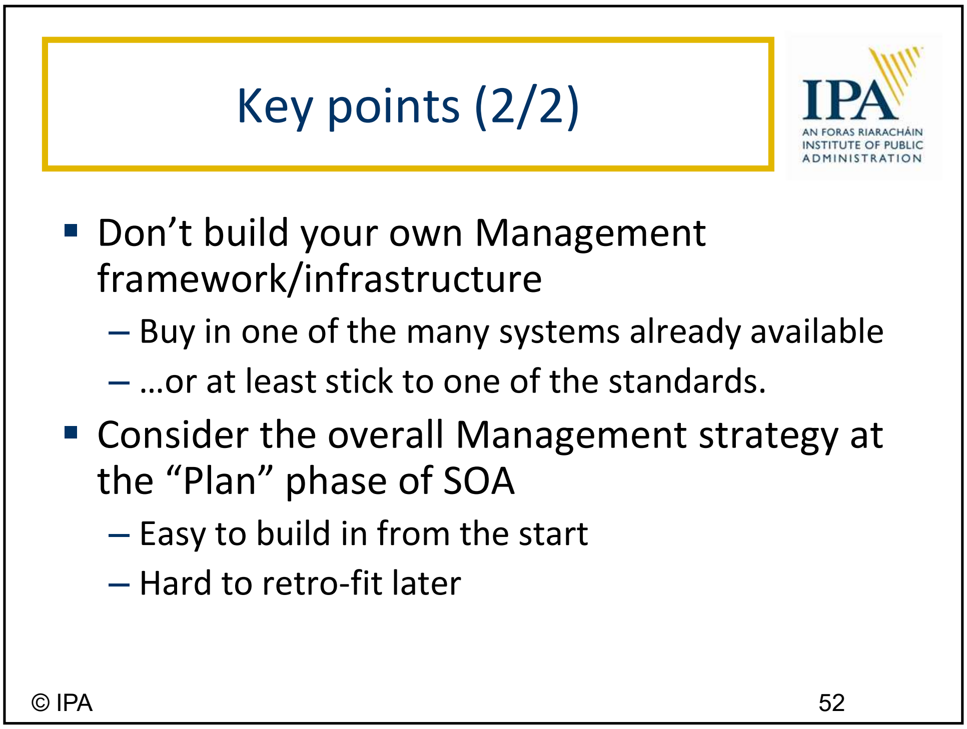
Future: WSDMI (3/3)

- Limited success
  - ...again.
  - Few implementations
  - Too complicated?
  - Not prescriptive enough?
  - Discuss!



Key points (1/2)

- Management is about:
  - Monitoring
  - Tracking
  - Controlling
- You can do SOA without Management
  - …but you'll miss a lot of the value.
- You can do WebServices without Management
  - …but it gets very complicated very fast!



Key points (2/2)

- Don't build your own Management framework/infrastructure
  - Buy in one of the many systems already available
  - …or at least stick to one of the standards.
- Consider the overall Management strategy at the "Plan" phase of SOA
  - Easy to build in from the start
  - Hard to retro-fit later

Overview

- Review
- Management of IT Systems
- Management in an SOA Environment
- Management of WebServices