

Enterprise Architecture and the IT Industry

Steve Craggs

Vice-Chairman, Integration Consortium

www.integrationconsortium.org

President, Saint Consulting Limited

www.saintconsulting.com



- The 'opportunistic architecture'
- Enterprise Architectures
- Architectures and the IT Industry
- The practical gap
- Examples
- Summary

- IT systems grow over time
 - Heavier usage
 - Wider access
 - Greater functionality
 - New processes and procedures
- Business pressures focused on completing the next task
 - Find the quickest way to make the next change
 - Little thought to the future
 - Solve immediate problems as they arise

➔ Opportunistic Architecture

- **Lack of visibility**
 - Difficult to get end-to-end visibility of
 - Individual processes
 - Overall business performance
- **Lack of control**
 - Hard to implement corporate and business policies
 - Difficulties in marshalling resources to support varying demand
- **Hostility to change**
 - Inflexible and brittle systems
 - Slow to get new offerings to market
- **Uncompetitive cost structure**
 - Change is expensive, often requiring highly priced, rare skills
 - Redundancy is high, raising project development and support costs

- A simplistic view of enterprise architecture
 - Consistency with a degree of formality
 - A set of definitions and rules
 - Supported by tools and procedures
- **Scope**
 - Interfaces
 - Technologies
 - Policies
 - Techniques
- **Benefits**
 - Addresses all the issues of an opportunistic architecture

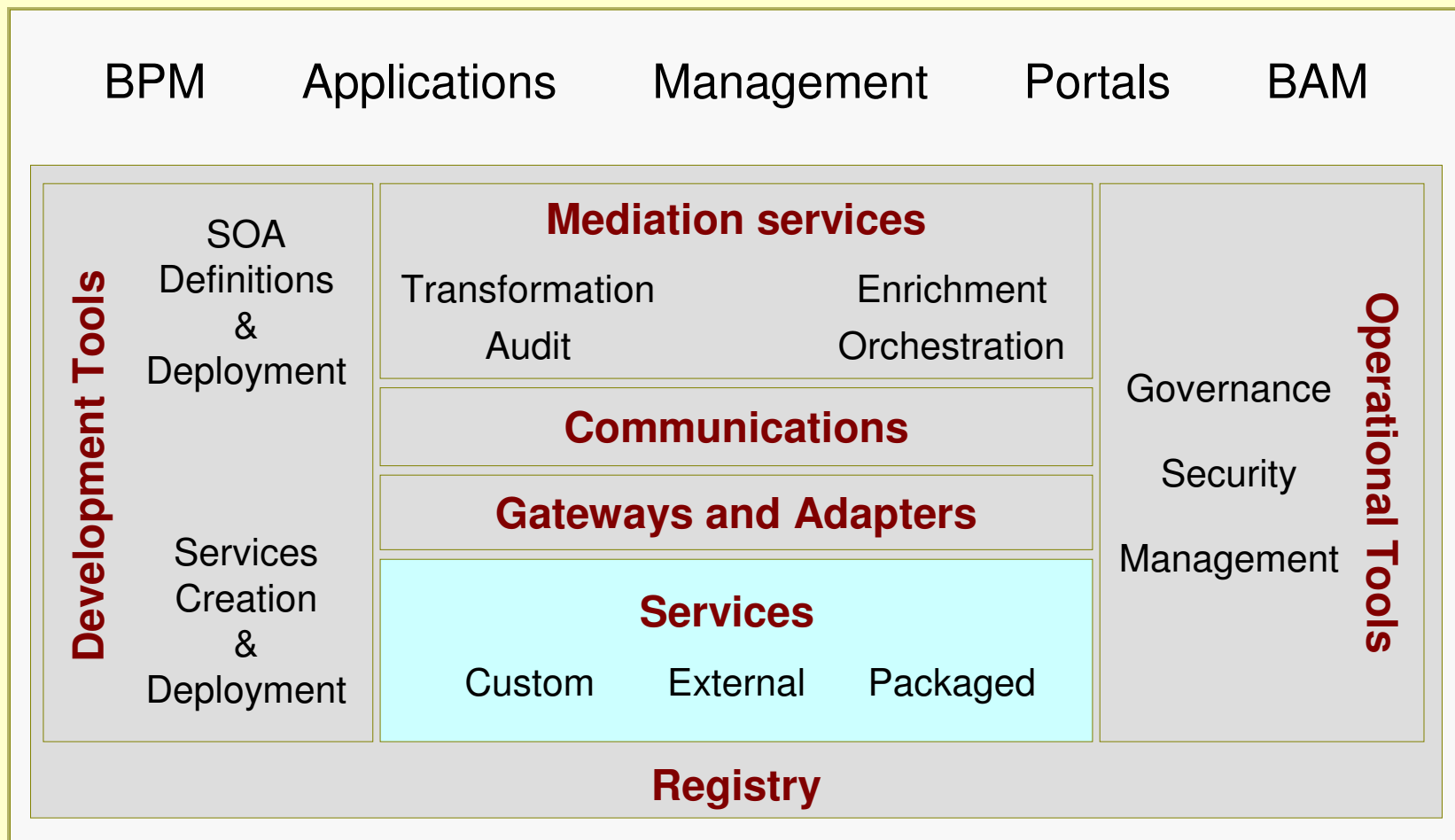
- **Service-Oriented Architecture**
 - IT components orchestrated into business services
 - Piece of business functionality
 - May map to a single IT component or multiple ones linked together
 - Standard means of invocation of services
 - Services can span multiple technologies, platforms and locations
 - Services can themselves be orchestrated into new services

- SOA characteristics:
 - Transparency of local IT environments
 - Connectivity and orchestration to form business services
 - Ability to reuse services
- SOA benefits:
 - Change is easier and cheaper, improving business agility
 - IT becomes more closely aligned to business activities
 - Reuse improves quality of service and lowers costs
 - Skills requirements are less and more freely available
 - **Business risk is reduced**

- The IT industry loves architectures
 - Introduces the architectural blueprint / reference architecture = what tools you need
 - Allows vendors to position themselves
 - Against each other
 - Matching with 'user needs'
 - Bridges technology to customer requirements
 - Enables new 'differentiations' to be created (often artificially)
 - ➔ **Brilliant for marketing!**
- Watch out for hidden agendas
 - The 'one stop shop' play
 - New architecture to support a new product
 - Morphing of architectural component definitions

The SOA Ecosystem

An architectural blueprint in action



You need the whole blueprint to deliver your architecture

- Some vendors sell you components
- Others claim to provide a one-stop shop
- Continuous morphing of the blueprint and the component definitions occurs
 - Vendors / Analysts struggling to differentiate themselves
 - Eg The Enterprise Service Bus (ESB) has a myriad of definitions
 - Eg Some vendors claim a repository and a registry are two components
 - Eg Some sources say Business Activity Monitoring (BAM) is essential
- But remember, YOU have to implement the architecture...

Industry voices get locked into the blueprint, often forgetting the practical issues

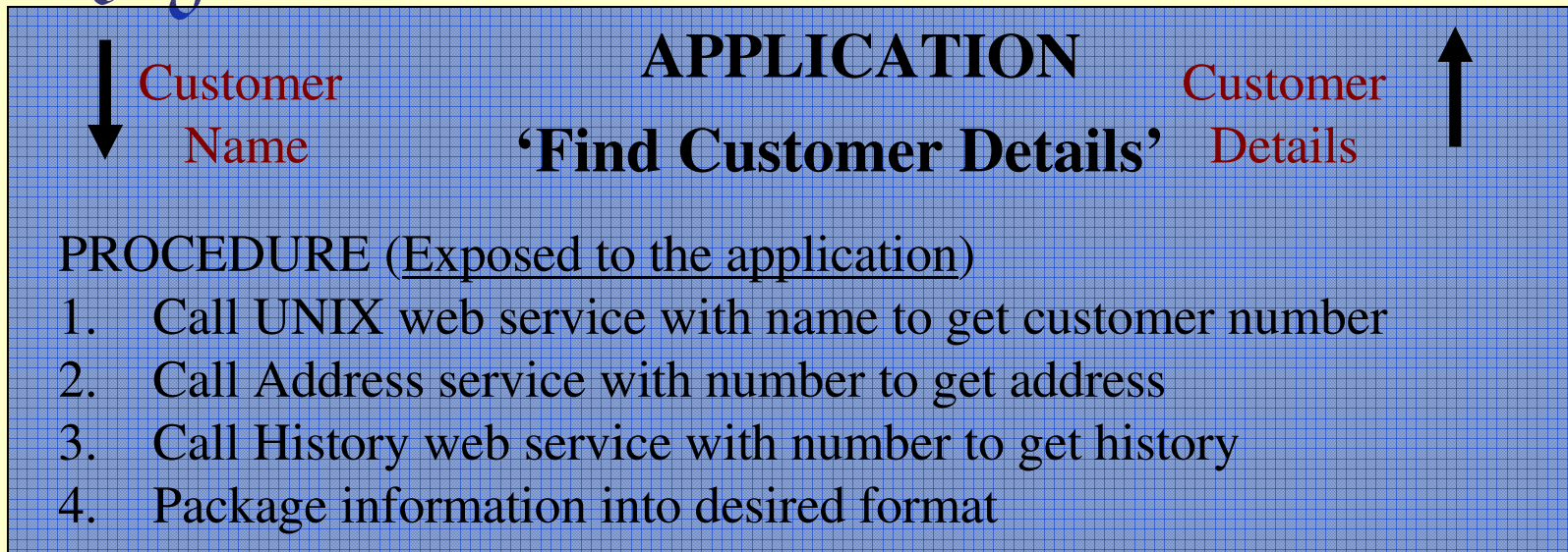
- How should you go about planning for the architecture?
- What new procedures and processes will you need?
- Does the new architecture create organizational issues?
- How should you prioritise architecture adoption and deployment?
- How do you create a winning business case to ensure approval?
- Do you have sufficient corporate backing to enforce adoption?
- How do you measure the returns?

Some examples from the SOA use-case

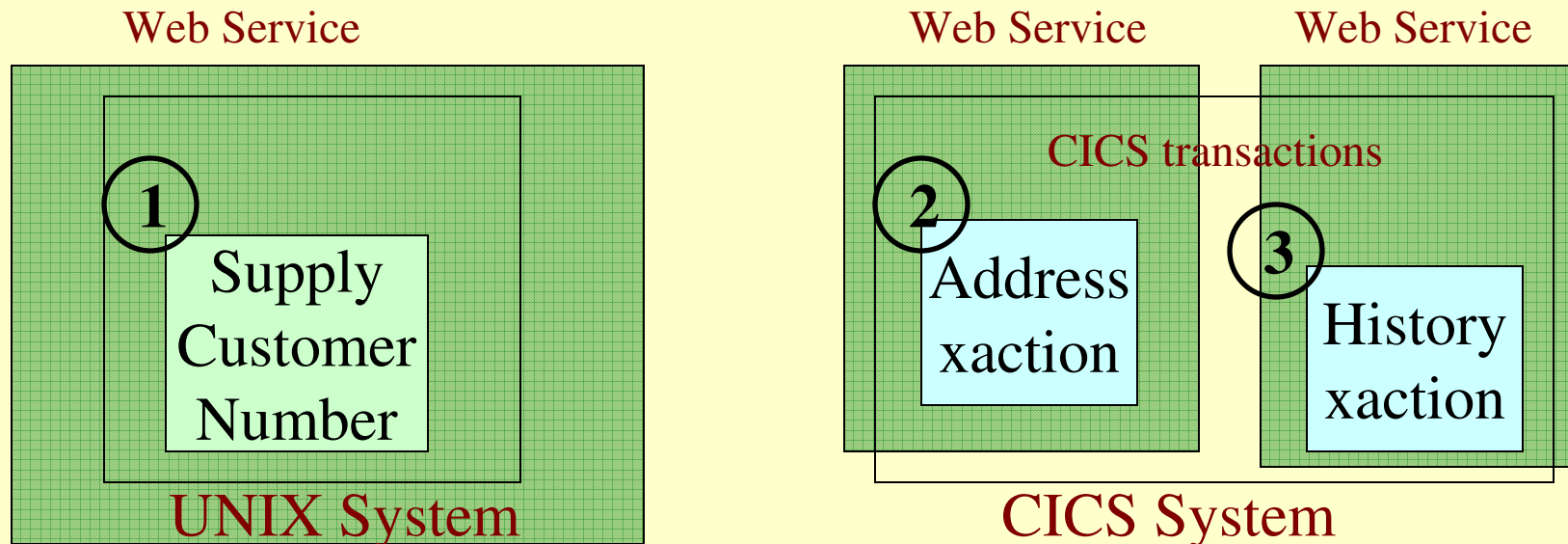
Get the service granularity right

- Turning every program into a service may be inefficient
 - Multiple WSDL copies
 - Potentially multiple conversions to/from XML
 - Multiple trips to/from service location
- Building the service at the wrong level will not achieve loose coupling
- An orchestration component will be needed
 - As provided by the ecosystem
- Remember to think beyond tens of services
 - In a successful SOA implementation, growth may see hundreds or more services created or exposed

NB: See 'Invest in SOA Service Design', www.lustratusresearch.com



Web Services Interface (WSDL)



APPLICATION
Call 'Find Customer Details' web service



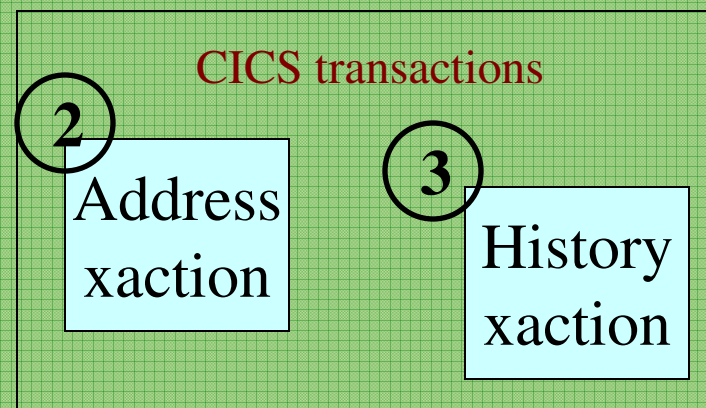
'Find Customer Details' multi-step web service

PROCEDURE (Hidden from application)

1. Input name to UNIX web service to get customer number
2. Run CICS transaction using customer number to get address
3. Run CICS transaction using customer number to get history
4. Package information and return



UNIX System



CICS System

Remember that SOA is not just about technology – it's a transformation of the business

- Cross-departmental process for agreeing service definitions
 - Escalation process to resolve disputes
- Incentive scheme to encourage reuse
- Education and Training requirements
 - Turn technology skills into services skills
- Radically different approach to application creation (composite application development)

- Enterprise architectures address the problems of opportunistic ones
 - ❑ Lack of visibility
 - ❑ Lack of control
 - ❑ Hostile to change
 - ❑ Non-competitive cost structure
- The IT Industry will be quick to join in with reference architectures and blueprints
- But you must still take responsibility to ensure a practical approach
- **With proper thought, planning, adoption and policing, enterprise architectures can deliver real business advantage**