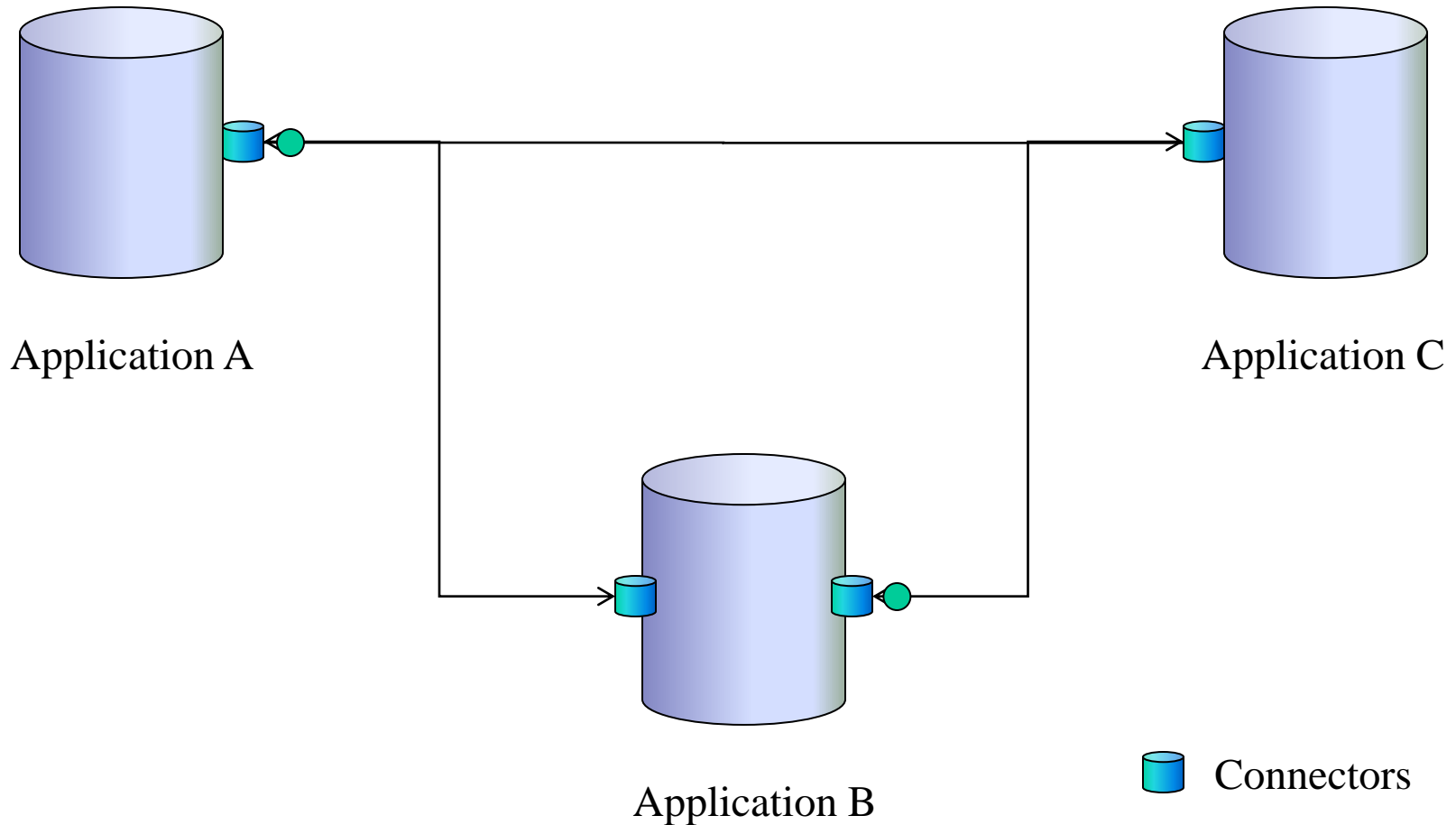
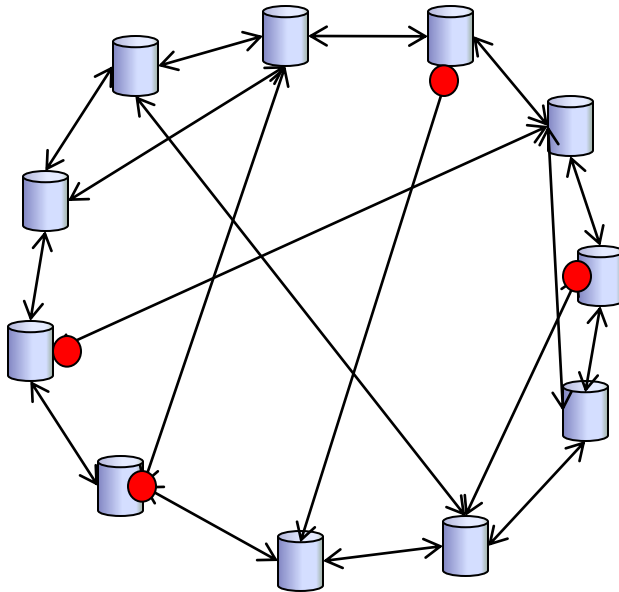


# SOA, Interoperability and the Need for Advanced Information

Clive Longbottom,  
Service Director, Quocirca Ltd

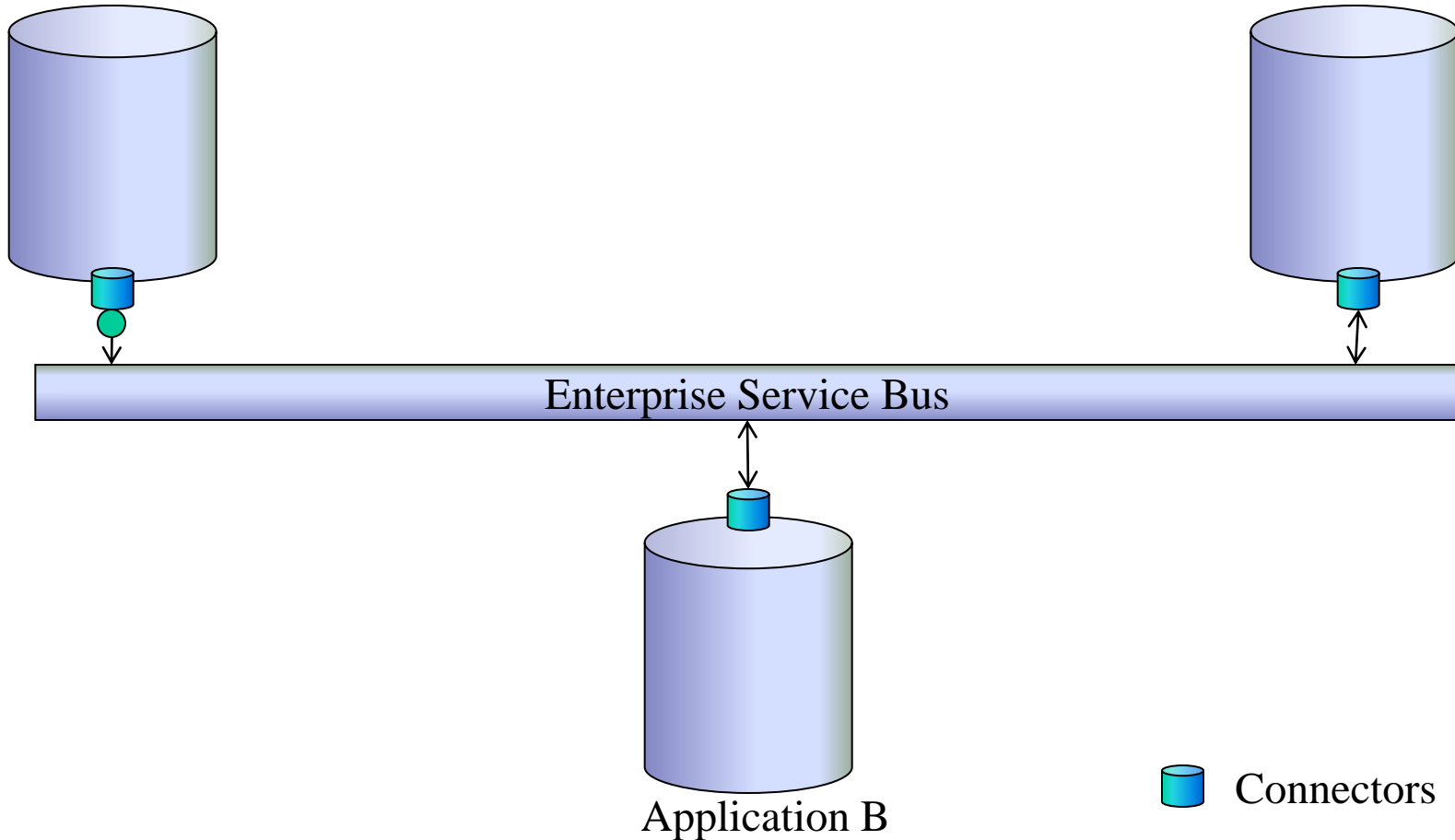




- Too many instances of applications needing too many connectors
- Any changes required all dependent connectors to be changed
- Companies tended to only connect those applications that were perceived as being necessary

Application A

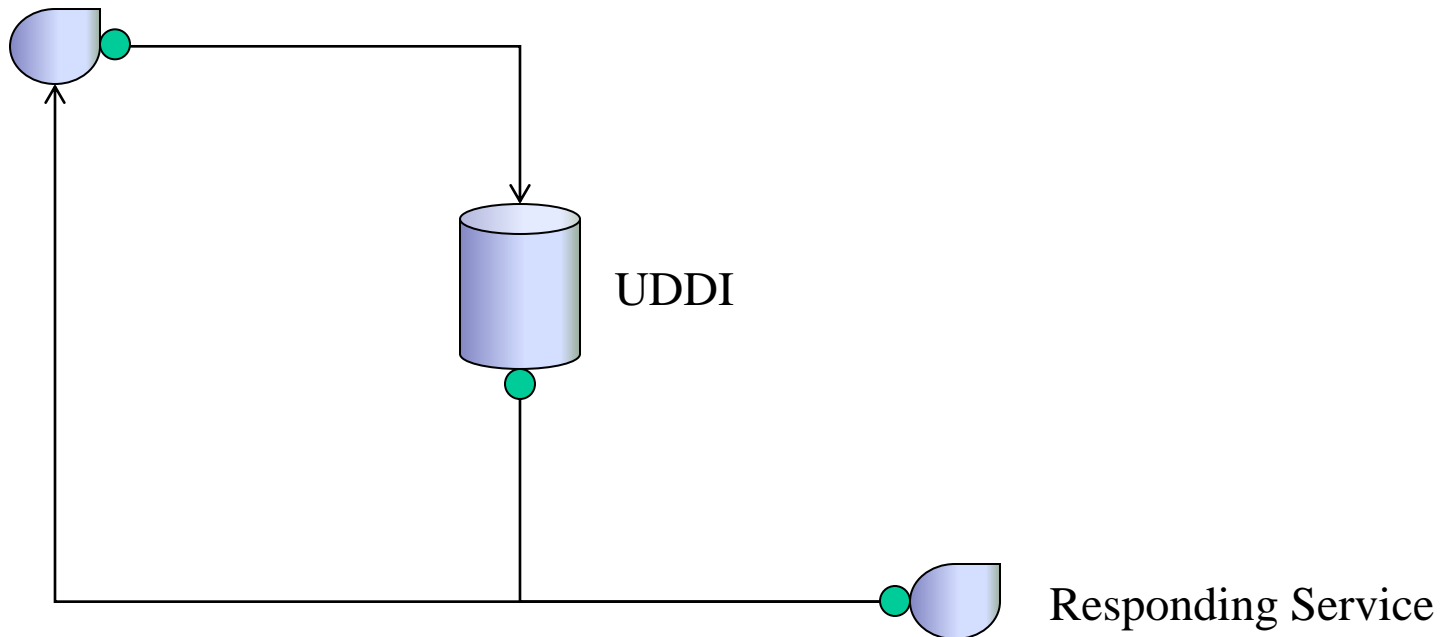
Application C



- Interoperability is dependent on multiple standards
  - ...Or on none at all
  - Hard coded interactions tended to be the standard
- Monolithic
  - Little chance for functional re-use
  - Change is a major issue
- Scalability issues
  - Tended to be one server, one application
- Resilience issues
  - Tended to be mirrored via clustering

- Web Services
  - Basic components of functionality
  - WS-DL as the underlying language
  - WS-\* as extensions/further definitions
- Reusable
- Flexible

Calling Service



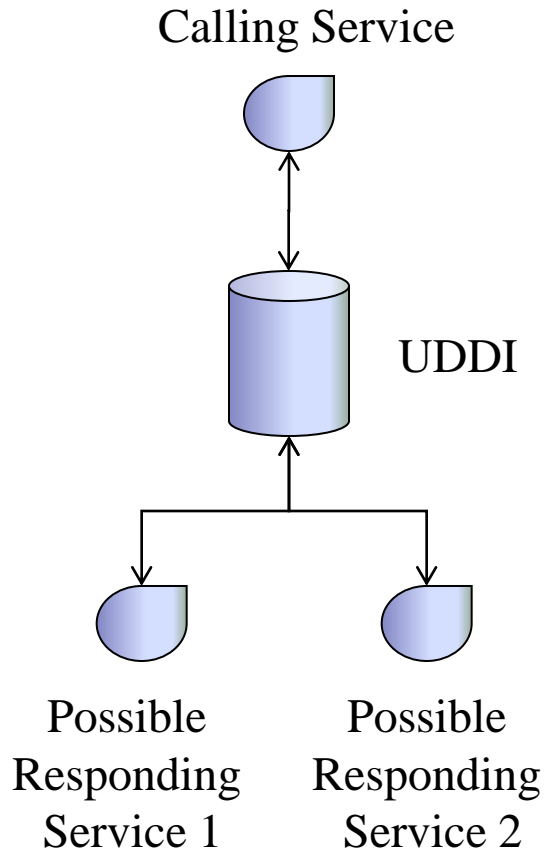
- Universal Description, Discovery and Integration
  - An xml-schema repository
  - Standardised via OASIS, now on V3.0
  - Acts as the middle-man between calling and responding services
- Calls carried out via SOAP
- Interoperability would be through xml documents
  - But what should be in the document?

- Simple Object Access Protocol
  - Uses XML as a message format
  - Tends to use http/https as a transport protocol
- XML verbosity can lead to slow responses, but gives richness
  - Message Transmission Optimisation Mechanism (MTOM) gives acceleration

- SOA takes Web Services to the next level
  - Composite applications
  - Highly flexible
  - Highly interoperable through SOAP/xml

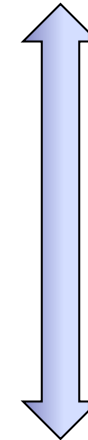
- Correct granularity an issue
  - Is SAP a service?
  - ...Or is one line of code?
- Contract negotiation not a strong point
  - Lots of services – which one is right?
- Richness of xml a two-edged sword
  - You can do anything – but should you?
- Dynamic nature can be a problem
  - Where is the instance of a service running?

- Just how do we deal with interoperability in an SOA world?
  - Xml is the basis
  - SOAP/UDDI provide the intermediaries
- Requirements:
  - Reliable transport (message queuing)
  - Service run time governance
  - Contracts



“I have to have...”

“I want to have...”



“I have...”

“I don't have”

- SLAs are easy – but essentially useless
- SVMs are hard – but have strong business value
  - Through use of semantic contract negotiations, services can be provided based on:
    - Speed of response
    - Cost
    - Throughput/scalability
    - ...

- Businesses need greater flexibility
  - SOA based on WS provides that
- Existing constructs can be too simple, with little business value
- There is a need for dynamic contract negotiations between services
- Xml provides a standardised and proven means of structuring such contracts
  - But the semantics have to be agreed