DOCTOR OF PHILOSOPHY

Business data integration framework for small to medium enterprises (BDIFS) : a service-based framework to support eBusiness data interoperability for small to medium enterprises (SMEs)

Kirkham, Thomas

Award date:
2007

Awarding institution:
Bangor University

Business Data Integration Framework for Small to Medium Enterprises (BDIFS): a service-based framework to support eBusiness data interoperability for Small to Medium Enterprises (SMEs)

Thesis submitted in candidature for the Degree of Doctor of Philosophy

May 2007

Thomas David Kirkham

School of Informatics University of Wales

Bangor

CONTENTS

Acknowledgements **10**

Abstract **11**

Chapter 1 ..... **15**

1.1 The Integration Challenge 15

1.2 Aims and Objectives 17

1.3 Thesis Structure 18

1.4 Contributions 21

1.5 Publications 24

1.6 Summary 25

Chapter 2 ..... **26**

2.1 Distributed Computing in Research 26

2.2 Distributed Computing in Industry 27

2.3 The Emergence of Service Orientation 30

2.4 The Evolution of the Grid 32

2.5 SOA relationship to eBusiness and SMEs 35

2.5.1 SME Integration Needs in North Wales 37

2.5.2 The SMEs this thesis is aimed at 39

2.5.3 Integration pressure from above 41

2.6 Integration Technology 46

2.6.1 Legacy Solutions 46

2.6.2 Middleware 47

2.7 Data Management 51

2.7.1 Standards evolution 51

2.8 Integration Technology 52

2.8.1 Software designs 53

2.8.2 Enterprise Application Integration (EAI) 53

2.8.3 Application Service Provision (ASP) 55

2.8.4 Business integration to suit SMEs 58

2.8.5 Building a solution to suit SMEs 60

2.9 Summary 66
Chapter 3 68
3.1 Grid Design Toolkits 68
3.1.1 The Web Services Resource Framework 70
3.1.2 Other Relevant WS Standards 72
3.2 Peer-to-Peer integration 76
3.3 Service Discovery 77
3.5 Workflows 80
3.6 Business Use 82
3.6.1 Service Level Agreement (SLA) 82
3.7 Portals 85
3.8 Security and Trust 86
3.8.1 Security 86
3.8.2 Trust 88
3.9 Summary 90
Chapter 4 92
4.1 Project JXTA 93
4.2 The Data Transfer Process 96
4.2.1 Data Transfer Structure 96
4.3.1 User Transactions in BDIFS 99
4.3.2 Using a Reporting Peer. 104
Security in the BDIFS P2P Design 105
4.5 Hardware and Software 107
4.6 Evaluation 110
4.6.1 Performance 110
4.6.2 Limitations of JXTA 111
4.6.3 Data Integrity 111
4.7 Summary 113
Chapter 5 115
5.1 Introduction 115
5.2 BDIFS Web Service Components 117
5.2.1 VO Management 117
5.2.2 Services 121

5.2.3 User Registration, Service Discovery, Brokering, and SLA. 124
5.2.3.1 User Registration 124
5.3 Service agents 130
5.4 Portal, User Registration, Token Base 132
5.5 Workflow Management 135
5.6 Security 139
5.6.1 Identity 139
5.6.2 Authorisation and Authentication 141
5.7 Deployment Architecture 146
5.8 Summary 148
Chapter 6 150
6.1 Company A: Use Case 1 151
6.2 Registering with BDIFS 152
6.2.1 User Registration 152
6.2.2 Using BDIFS 155
6.2.3 Service Provider Registration 156
6.3 OpVO Population 160
6.3.1 Authentication 160
6.3.2 OpVO Set Up phase 162
6.4 Workflow Execution Phase 167
6.5 Service Failure 170
6.6 Service Destruction 175
6.7 Testing Methodology 177
6.8 Compatibility Issues 178
6.9 Summary 180
Chapter7 182
7.1 Akogrimo 182
7.1.1 Mobility a challenge for BDIFS 183
7.2 Operational VO integration 184
7.3 Supporting mobile services in BDIFS. 185
7.4 eScience Application 188
7.4.1 Using XSLT 191
7.5 Commercial Applications 193

7.6 Summary 194
Chapter 8 197
8.1 Conclusions 197
8.2 Future Work 201
8.2.1 Service Development 202
8.2.2 Agent Development 203
8.2.3 Customer Requirements 205
8.2.4 Security 205
8.2.5 Other 206
8.2.6 Design Technology 207
APPENDIX 1 209
1.1 Company A 209
Appendix 2 212
References 228

LIST OF FIGURES

Fig. 1: Common Integration Scenarios for SMEs in North Wales 43

Fig. 2 Application integration using the Enterprise Application Integration server approach 61

Fig. 3 ASP approach to integration. 63

Fig. 4 Three areas of integration control in the ASP model. 65

Fig. 5: BDIFS P2P Peer Groups. 95

Fig. 6: Data Transfer Process and Structure. 97

Fig 7: Joining the BDIFS Framework 101

Fig 8: Downloading and Using the Reporting Peer 105

Fig 9: Software and Hardware Architecture. 108

Fig. 10: VO Management in the Base and Operative VOs 119

Fig. 11: Simple Order Process 123

Fig. 12: Service Discovery 127

Fig. 13 Application Specific Service Invocation 131

Fig. 14: Initial User Portal – Base VO Interaction 134

Fig. 15: BPR File Structure 137

Fig 16: BDIFS Domains and Policy Flow 142

Fig 17: Inter Domain Security 143

Fig 18: Token Structure 145

Fig 19: BDIFS Testbed Architecture 146

Fig 20: Example Input and Output Data 148

Fig 21: User Registration Screen 153

Fig 22: User Registration Process 154

Fig 23: Service Registration. 159

Fig 24: User Authentication. 161

Fig 25: User Authentication Process 161

Fig 26: Authentication Failure 162

Fig 27: Workflow Deployment. 163

Fig 28: OpVO Instance Creation 164

Fig 29: A Deployed Workflow 166

Fig 30: Workflow Execution Process. 168

Fig 31: SLA 1 Service Failure. 171
Fig 32: SLA 2 Service Failure. 173
Fig 33: Service Destruction. 176
Fig 34: Operational VO and Base VO Linkage 185
Fig 35: BDIFS Meets Akogrimo. 186
Fig 36: Original Data Format 189
Fig 37: Original Data Format With Values Marked in Regular Format 189
Fig 38: Final Data Mapping, Based on Desired Format with the Positioning of Appropriate Values Included. 190
Fig 39: XSLT Template 192

LIST OF TABLES

Table 1: Base VO Interfaces 118
Table 2: OpVO Interfaces 120
Table 3: User Registration Service (userReg) Interfaces 125
Table 4: Broker Service Interfaces 126
Table 5: Discovery Service Interfaces 128
Table 6: SLA Manager Interfaces 129
Table 7: Service Agent Factory Interfaces 132
Table 8: Service Agent Interfaces 132
Table 9: Portal Registration 135
Table 10: Workflow Engine Interfaces 137
Table 11: Workflow Manager Interfaces 138
Table 12: TokenStore 145
Table 13: BDIFS Deployment Description 147

Acknowledgement

Thanks to all at Bangor and RAL who have supported and inspired me over the years during the creation of this work.

The thesis is dedicated to my family and friends both old and new and no longer here.

Abstract

This work presents a platform to aid multi-vendor and multi-industry eBusiness integration for Small to Medium Enterprises (SME) by the creation of specialised Virtual Organisations (VOs). Central to the management and creation of the VOs is the relationship between users and service providers. A framework of VOs and management structures for users and service providers is presented in this thesis within the context of the Business Data Integration Framework for the SME (BDIFS) project. The work has also been applied in the EU Framework 6 project Akogrimo, and is a significant contribution to both work and research in the area of VO management and composition.

The motivation for this project is directly related to the need for improvement in current practice within common eBusiness integration solutions that tend to focus on single vendor and industry specific applications. Using emerging distributed computing standards and technology, the VO based solutions presented in this work demonstrate multi-vendor and industry-wide support. The BDIFS application is designed to grow through community involvement as opposed to targeted designs which dominate the evolution of commercial Grids. The BDIFS design has evolved from a Sun JXTA (Juxtapose) design to the main design using WSRF (Web Services Resource Framework) Web Services.

The argument and thought behind the integration platform presented in this work is a result of both academic research into the application of P2P and Web Services in eBusiness solutions, and practical research conducted with SMEs in North Wales. The research in the BDIFS project has illustrated that both current research and solutions provided to aid business-to-business integration by both academia and industry prior to BDIFS are not suitable for many SMEs.

In the future there are likely to be other integration problems with other forms of eBusiness integration, for example mobile devices, when included in Service Orientated Architectures (SOA), are likely to pose physical integration problems such as the expression of mobile data which can be done in various vendor specific ways. To address this, the thesis presents a revised BDIFS Web Service architecture applied for use with mobile Web Services, to demonstrate that the BDIFS Web Service architecture can be easily adjusted to handle emerging integration challenges.

In summary, the thesis addresses a key issue in the enablement of complex eBusiness partnerships for SMEs with few technical resources. This is achieved by investigations and use of dynamic VOs, workflow composition and JXTA peer groups. The focus on the dynamic creation of environments to aid eBusiness integration is the key contribution of this work in the field of VOs. Using SMEs as the targeted users, the BDIFS solution demonstrates a viable and highly significant integration framework. The BDIFS framework has the potential to change the way SMEs integrate and it introduces a new

approach to application design in this problem area.

Chapter 1

Introduction

1.1 The Integration Challenge

This thesis is centred on the integration of Information Technology (IT) systems in a business context that, at its most basic, involves the establishment of communication and understanding between at least two computing environments. This integration in commercial reality (particularly amongst SMEs) often stalls on the integration of legacy systems where data is presented in non standard forms and logic. The types of partnerships which present this challenge to the SME are often led by large businesses (a SME can be defined as a company with less than 250 employees). In these cases it is the larger businesses (often via the strength of their resources both financial and technical) that can dictate the decision about how the SME integrates in the eBusiness partnership. This lead is enabled by software (frequently used to enhance supply chain management) that is often technically and financially

out of reach of many SMEs. This creates an environment that fuels an integration divide that is increasing with globalisation. The ability of SMEs to integrate electronically with larger suppliers and customers is essential in order for them to remain competitive.

As a result of these factors, SMEs often have no choice but to achieve automated business to business integration to suit the needs of larger business partners, which can damage the business of the SME. This damage is caused by the adoption of software and technology that can negatively alter the internal workflow of the company: software that can remove resources from business critical functions within SMEs, and software that can expose an SME's data to competitors. This work presents evidence to support these arguments, proving that SMEs are integrating in ways which are potentially causing damage to their business. Often the technical and internal infrastructure/workflow of the business, along with financial and IT resources, are jeopardised in the integration process.

The key motivation behind the research presented in this work is to produce a solution to remove these integration barriers and reduce the risks facing SMEs. This is achieved by directly addressing the issues behind the power that the larger integration partners exert over the smaller companies within business-to-business trading partnerships [1,2]. This power often dictates how the SME integrates and this work examines it in a technical sense. The result of the examination is the Business Data Integration Framework for the Small to Medium Enterprise (BDIFS) project. This is the first international research

project to simultaneously raise, address and explore these specific issues and offer a practical solution [3].

BDIFS is a framework to aid and increase access to standards based integration between SMEs and larger partners, regardless of the legacy systems involved. This is achieved by initially using a Peer-to-Peer software design, which is superseded by a dynamic service and workflow composition approach using a Service Oriented Architecture (SOA) implemented using Web Services (WS). Due to strong industrial links within the project, this work builds on specific practical examples with particular reference to Welsh SMEs.

1.2 Aims and Objectives

The BDIFS framework supports SMEs in their business integration needs. The solution builds upon state of the art emerging resource focused Web Service and P2P technology presented to users using dynamic VOs. The BDIFS solution provides an SME centric, multi-vendor and non-industry specific view on the business integration process, and is a viable example of a business use of both Web Services and P2P technology. The objectives leading to this aim stem from the creation of a suitable software framework to enable this application. This framework has to provide the following functions:

1. Explore and present innovative methods for providing a remotely hosted complex service to a user with little technical knowledge by the development and use of specific Virtual Organisations.

2. Create a message exchange mechanism which guarantees data privacy, and is secure, robust and requires minimal configuration and maintenance effort.
3. Present a multi vendor integration platform that can adapt to include new functions without any changes needed to be made at the local site of the SME.
4. The software will present exploitation opportunities, leading to the support of a business model that will encourage both use and collaboration in the framework. In order to aid open source development and free support alongside this business model, it is envisaged that a free to use community forum to encourage SMEs to discuss integration challenges and share knowledge and experiences related to integration will be developed.

1.3 Thesis Structure

The thesis presents two designs of BDIFS, one using Sun JXTA P2P technology and the other using WSRF based Web Services. These designs address eBusiness integration issues facing SMEs in North Wales and beyond and are introduced in the chapters before the software architecture. Future integration and the application of the framework, with respect to use within large scale science facilities and mobile devices, are described in the final software application chapter.

EBusiness Integration Technologies, the technical and business context of the research elaborating on the eBusiness integration pressure and challenges

facing SMEs, is set in Chapter 2. Using North Wales as a case study, the chapter cites local and national statistics to illustrate the trend towards automated eBusiness integration and how SMEs are failing to adopt this technology.

The chapter also examines current integration technologies, giving special attention to the composition of an ideal integration architecture to aid SMEs. This is done by investigating the key elements in current software architectures that are addressing eBusiness integration challenges.

Building on the elements from the integration architectures discussed in Chapter 2, Chapter 3, Related Work, presents various technologies that are available to implement the desired remotely hosted type of integration architecture. Although this section examines P2P design technologies, it focuses largely on the current state of the art emerging from the Web and Grid communities in which the main design of BDIFS follows.

Chapter 4, BDIFS P2P, initially discusses and presents the first BDFIS design using P2P software. The choice of technology, architecture design and evolution of this P2P messaging approach is covered. Use cases and results of tests into the application of the P2P design are recorded and discussed, and the limitations of the design are also outlined.

Chapter 5, BDIFS Web Services, describes how the Web Service design builds on the BDIFS aims and objectives in Chapter 4, focusing in more detail

on workflow and VO management in the creation of a web service framework. The design is a demonstration of how Web Services can build dynamic VO and Workflow Management frameworks to aid real world business processes. The Dynamic Application Service Provision within the Grid presented in this work, specifically in the application domain of BDIFS, is a unique approach and has broken new ground. Chapter 6 details each component that makes up the framework and its key functions and technological make-up.

As the Web Service design of BDIFS is the more detailed prototype, the use cases for it are discussed in more detail in Chapter 6,, Use Cases and Testing. This outlines the use cases of user and service interaction within the framework. The use cases portray the process from user or service registration through to service execution, along with Service Level Agreement (SLA) failure. This demonstrates a thin slice of the whole BDIFS application lifecycle, and future areas of development are discussed.

The discussion in Chapter 7 explains the BDIFS applications. The use of BDIFS in eScience facilities and the Akogrimo project moves the integration needs of SMEs away from message integration in a business process sense to the integration of new types of business specific data, such as structures of purchase orders and their content. The thesis uses the example of data from emerging mobile devices and other types of legacy system. This is demonstrated as a new integration challenge for SMEs, whilst illustrating how BDIFS is flexible enough to react to new integration challenges.

Chapter 8, Conclusions and Future Work, brings together the work from the previous chapters and matches it against the original aims and objectives of BDIFS. An assessment is made of the achievements of BDIFS, and future areas of work and future research opportunities highlighted by the project are discussed.

1.4 Contributions

The work is a significant contribution to research into the development of VOs and applications in eBusiness. The implementations are focused around the creation of dynamic environments and VOs, and thus they are the key technical contribution. In summary, the contributions that this work has made are listed below:

1. The work in this thesis has developed a unique and specialised VO relationship model for both dynamic and static VOs within the context of information integration. The Web Service implementation of the thesis explores the concept of dynamic VOs in business integration and implements a dynamic VO model for the application in the thesis. This application of dynamic VOs is extended to the mobile support services and has been deployed in the Akogrimo project. Through both these dynamic VOs the thesis has developed a unique contribution to this traditionally Grid computing area of research.

2. This is the first research project to practically address power control issues in real SME eBusiness partnerships, power control being the control of the power exerted by the larger partner in the partnership. In a technical sense, the power is illustrated in a one-way direction from the larger partner in the business partnership. This partner, often with more advanced IT resources, is able to dictate to the smaller partners with few IT resources how to integrate. This integration is usually via a method friendly to the larger business as it can exert its power, a power that is routed in the smaller suppliers' needs for business from the larger partner, for example a scenario with a large supermarket and a small scale sauce manufacturer.

   To date, papers have been written and research has been conducted on the emerging patterns of power control. But none of this work has been able to offer a technical solution; often the solutions are economic or business oriented. This thesis looks at the next generation of power control and how it is enhanced by the divide in IT resources many businesses have with their larger partners. Using Web Service technology this thesis proposes a solution to narrow this divide.

3. The concept of non technical, user-driven workflow and service creation in a Grid computing environment is presented. Users create and store both data translation and workflow settings for use and sharing by multiple users. This is unique in an eBusiness environment and links to previous work such as the Morpheus project which has applied a similar approach in the field of Geographical Information systems [4]. Also, this approach

has been applied in more general terms in the OGSA-DAI project, in particular within the OGSA-DQP architecture[5]. The thesis mirrors this architecture, to the extent that it creates resources to match transaction specific information mapping. Data, for example, can vary in terms of how price is calculated or how an address is expressed, depending on how the original computing systems handle this information. Nevertheless, the thesis takes this data in its simplest form and aims to perform a set of basic mappings. The possibility of added functionality to calculate price in different ways could also be integrated into these mappings, or in services that support them. In the thesis, this concept is focused specifically on the case of SME based eBusiness data, and the transformation of text based data and its application with workflow rules, both added by non technical users; no other research project has attempted this combination.

4. The work is the first to present a solution to address the eBusiness integration needs of SMEs in North Wales. Apart from the eCommerce State of the Nation Report at the University of Cardiff and national statistics, few small scale local case studies into eBusiness in North Wales SMEs have been conducted. The work at Company A is a significant and unique case study into the evolving eBusiness needs of a SME in this locality.
5. The work presents a framework that can save development organisations millions of pounds in funding grants for businesses wishing to integrate. Typically a business integrating, using commercial or open source based EAI type software, would have to invest in hardware support and purchasing costs. This figure can be seen to start at £1000 for a good

server and £100 a month plus for network access and server/network support costs. Software costs on top of this would vary from the cost of customisation of open source software to licences for a proprietary integration server; these costs can run into thousands of pounds. The main outlay for the BDIFS system would be the custom scripts needed to integrate and export data from the local database. These have been priced at approximately £1000 in the case of Company A, a substantial saving. More information can be found in the business plan within the Appendix.

1.5 Publications

The research publications that have stemmed from this work are listed below:

1. A business service network to aid collaboration between small to medium enterprises Proc. 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services, San Francisco, date?,
2. Providing location sensitive eBusiness integration for the small to medium enterprise Proc. IEEE WETICE, Manchester, 2006,
3. Supporting E-Business integration in North Wales Proc. ECHALLENGES 2006
4. Orchestration and workflow in a mobile Grid environment Proc. WSGE, Beijing 2006
5. Introducing BDIFS Proc. EEE Las Vegas , 2006,
6. Migrating BDIFS Proc. 8th International Conference on Enterprise Information Systems 2006 Cyprus

7. Providing reliable distributed Grid services in mobile environments in Self-Organization and Autonomic Informatics (I), eds. H Czap, R Unland, C Branki, H Tianfield, Frontiers in Artificial Intelligence and Applications 135, ISBN 1-58603-577-0 (IOS Press), place of publication p246-255 (2005)
8. A Framework to Provide eBusiness Support for African SMEs, IST Africa 2007 Maputo
9. Using User Driven Dynamic Grid Services and Temporary VOs to integrate new users and old systems in e-science facilities, eChallenges 2007 The Hague

1.6 Summary

The focus of the BDFIS application presents a new and viable research area that is in need of significant support in the face of increasing eBusiness automation in order to improve the competitiveness of SMEs. The software presented in the thesis contributes to the research behind the deployment and development of applications using stateful Web/Grid services in dynamic environments and VOs. The application of the BDIFS software is a significant contribution to the use of Web Services and P2P technology in the formation of dynamic business computing environments. The work has also contributed to the EU Framework 6 sponsored Akogrimo project design and use of Dynamic Virtual organisations in mobile Grid computing environments. The future application of the work in this thesis in commercial and science domains is very real.

Chapter 2

eBusiness integration and the SME

In order to place this research in a technical context, the next section will set the scientific and industrial background to the work. Initially the emergence of Web Services is discussed, with High Performance Computing (HPC) as the starting point. Next the business environment facing many SMEs in North Wales is presented, along with the technical barriers they face. The discussion moves on to the solutions available to aid SMEs. An assessment of the key features of these solutions is then made. As a result of that investigation, the key elements in a software solution to aid SMEs are presented.

2.1 Distributed Computing in Research

A major area of research in distributed computing is focused on resource sharing across machines in local environments; it is an area of research that has developed from the use of computers in high processing environments. This type of computing is commonly referred to as cluster computing [6] and

requires parallel processing techniques between machines. In order to manage machine resources, and more specifically to aid job execution and scheduling applications along these lines, research areas have begun to emerge. A major research area is the actual creation of the resource, which is being tackled in projects such as Beowulf [7], where large amounts of data are processed by various machines linked together to form an HPC resource. The use of these resources is often kept within the local environment, but increasingly their use is evolving to reflect a wider distribution across organizational boundaries. This research area is commonly referred to as Grid computing [8,9] and is significant for the BDIFS project.

Work in distributed computing techniques has developed two key research areas in Grid computing which are central to this thesis; these are concepts and use of VOs [10] and Workflows [11]. VOs will be discussed in more detail later on, but can essentially be defined here as a distributed group of participants sharing resources to achieve a common goal; workflow in the area of computing is commonly described as the orchestrated execution of distributed services in order to deliver a greater goal. Essentially VO and Workflow Management enable a single application to use resources from multiple sources in an orchestrated way.

2.2 Distributed Computing in Industry

The emergence during the 1970s and early 1980s of desktop computers [12] and networking technology [13], enabling client-server environments, led to

growth and real take-up of the commercial use of shared data across wide areas. These computing environments used proprietary software and fixed networks, which were expensive and restricted to specific industries and large companies such as banks.

For banks, the use of shared data resources has always appealed; for example, the transfer of cash from bank to bank was an early commercial adopter of telecoms equipment [14]. Currently the relatively rich banking sector is the main industrial exponent of commercial high performance Grid computing [15]. In order to use these resources, Grid toolkits have emerged into the commercial domain such as those provided by companies like Platform Computing [16], and in the Grid community by organisations like the Globus Alliance. The Globus toolkit created by the Globus Alliance is used by organisations like the National Grid Service [17] in the UK to enable distributed HPC resource sharing to support large scale science experiments. This area of research is commonly referred to as eScience. With respect to banking, trading floor computing infrastructure is commonly used to process financial algorithms overnight. In other industries the distributed resource sharing model similarly follows the development of the HPC approach to computational process sharing. Applications using distributed resources range from image rendering in the animation world [18], to the use of the technology to aid the examination of geological data in the exploration for oil [19], often drawing in resources from the Wide Area Network (WAN) and not just the Local Area Network (LAN).

In the case of SMEs, this model of resource sharing across distance has potential, with respect to the ability of a resource hosted outside the organisation to provide some of the computation and program logic to solve a problem that a SME would not typically have the resources to solve. Current SOA approaches that enable the SME to compete above their resource level are present in the uses of Web Services by companies such as Amazon. The Amazon API allows businesses to link to the Amazon marketplace and trade using the computing resources and business resources available. These enable the SME to present more advanced interfaces to the consumer and the trader, and also potentially reach more customers [20].

However, more significantly, Amazon have introduced Amazon Elastic Compute Cloud (Amazon EC2) [21], which is a Web Service interface providing flexible computing capacity designed to make flexible large scale computing resources available for a larger number of users. This is similar to the Amazon Simple Storage Service (Amazon S3) [22] that like the EC2, enables flexible access to computing (in this case storage) resources from Amazon. Overall, combined Amazon EC2 and S3 aim to reduce the time required to obtain and boot new server instances, producing flexible usage for business and a new economics of computing by allowing the user to pay only for capacity that is used.

However, on a business to business scale, the sharing of more complex information is an area often solved by academic and commercial Grid based solutions. The problems to be solved by Grids traditionally have been associated with computational resource sharing, but in this work the resources shared provide the skills and knowledge to perform the translation and controlled transmission of the data. To date there is no commercial Grid technology specifically designed to address this type of data resource sharing. Commercial Grid solutions are still largely focused on a more fixed and process resource sharing model. But the development of Grids using a service-oriented approach has the potential to break this barrier, as the Amazon approach is demonstrating. Using the emerging open standards it is possible to present various different types of information management resources to be shared in VOs, not just processing and storage power.

2.3 The Emergence of Service Orientation

Web Services are behind the development of SOA and have quickly emerged as part of the development of the World Wide Web [23]. The W3C defines a web service as a software system designed to support interoperable machine-to-machine interaction over a network [24]. Web Services are more advanced in the use of data than simply using the Web as a method to browse text. To meet the demands of advancing machine-to-machine integration potential, Web Service technology has combined with Grid computing techniques to produce Web Service toolkits and standards for the Grid (discussed later).

A key driver in the development of Web Services to share both data and computation resources has been the development and adoption of open standards. The development and use of the transportation protocol SOAP [25] is a key factor in enabling Web Services to communicate, but the creation of Extensible Mark-up Language (XML) [26] is the heart of the Web Service. The XML standard has given the Web a standard by which data can be expressed in standard interoperable forms, thus allowing data ported between distributed applications (or Web Services) on the net to be expressed in a uniform way.

XML is not only to be seen as one of the first significant open standards that the mass adoption of Web technologies has produced, but also as a significant change in the way distributed computing applications communicate. There had been attempts to introduce open standards to aid application interoperability before the mass adoption of computer data sharing across distance was fuelled by the Web. For example, the work around business messaging standards, albeit a valid attempt to integrate using standards, is an event that happened too early (pre-World Wide Web) and has therefore suffered, as it has been specialised for various applications and software vendors [27]. In contrast, the use of XML based methods for expressing data benefit are more interoperable between designs, as XML based standards adhere to the basic XML constructs, and fuelled by large scale use on the Web, XML schema has support in large amounts of software from parsers to editors.

Early integration attempts and the emergence of applications, which were designed for LAN based integration, has led to a complex choice of methods to express data from legacy systems, although the success and strengths of XML, in particular at expressing metadata in a uniform way, has quickly led to standards which have the potential to take this data (once extracted) and use it to support more specific and interoperable data exchanges [28]. This has enabled the Internet to become attractive as a viable space for collaboration of businesses, and has created a sea change in ideas and attitudes to business-to-business integration by moving the resource sharing model from the Local Area Network (LAN) – Value Added Network (VAN) – Wide Area Network (WAN) to the Web [29].

2.4 The Evolution of the Grid

Grid technology has embraced, and is still embracing, new and emerging web service standards, along with developing its own standards to aid advanced resource sharing. Grid specific Web Service toolkits are an advance on common stateless web service toolkits, as they are designed to aid the representation of the resources state in service interaction. This is enabling the use of Grid computing techniques to evolve from a niche market that solely addresses the HPC domain towards a framework that is useable for various types of computing resources within a wide variety of business and knowledge management contexts [30,31,32].

The flexibility presented by SOAs therefore has the potential to remove the Grid computing model away from a reliance on a fixed networking infrastructure for resource sharing, and has given the concept of the Grid the potential to embrace the reality of more dynamic service use. Initially, the development of SOA, fuelled by Web Services, has presented the concept of flexible stateless resource sharing to the mainstream of computer users in the form of Grid enabled Web Services. The emergence of standards to support the Grid inspired stateful service use, and therefore has potential to add the same level of reach to a new generation of Grid computing applications. The Grid is no longer simply the byword for distributed process sharing. The Grid, aided by the new emerging standards, has begun to emerge as a business application layer, where standards like ebXML [33] and BPEL [34] are developing to present various types of resources for use over the Grid in business applications [35].

Environments consisting of services that are able to present state and form applications based on service state will create new types of service-oriented business applications. Significantly, this gives services the potential to compete in economic models, as the choice to be used in specific distributed computing applications, aiding the development of an Application Service Provider (ASP) Grid concept [36].

An ASP is a business that provides computer-based services to customers over a network [37]. Essentially outsourcing the computing resource, the use

of ASP services has taken off with the use of Web Services that allow the services to be made available on the Web. The use of Web Services has the potential to enhance the ASP approach, making the range of services available to the ASP Grid, and the ways in which they are used, more powerful and pervasive. This will be illustrated in this work and is of huge significance to SMEs.

The future of Grids and supporting ASP models is tied to the fortunes of the development of Web/Grid Service supporting standards. Recently the Web and Grid Service communities have begun to work together in the form of the OGSA (Open Grid Services Architecture) [38,39] and the OGSI [40] (Open Grid Services Infrastructure). Under these frameworks and architectures, Grid computing is developing as an SOA, becoming more accessible and lightweight in the process. The design in this work can be viewed as a genuine test of the results of this collaboration.

Thus, using SOAs and the standards that are increasingly emerging from the combined Grid/Web service community, a next generation of Grid-based distributed computing applications is emerging. These applications, building on many Web Service strengths, are adaptable to Grid applications such as resource discovery and execution management through the representation of service state in flexible environments, allowing SOA's to manage data better with respect to business applications [41] and scenarios that focus on less abstract data such as those that support mobility [42]. This has potential for

new investigations into the use of the technology in various applications, and significantly within distributed systems integration [43].

2.5 SOA relationship to eBusiness and SMEs

Due to cost and learning time, business applications to exploit new and emerging technologies have traditionally been adopted by the larger and more resourceful businesses [44]. However, using emerging Web Service sets of standards, a new generation of business computing applications can be created: applications that present business computing solutions that are fast to learn, cheaper to implement, and are more scalable and flexible to suit SMEs and individual consumers.

The ability of Web-based services to present common standard-based interfaces has separated them from the LAN-based application logic from which they originate. Furthermore, the use of standards opens the use of the services and underlying applications to a greater population. Remotely hosted services using open standards are accessible to anybody (with the appropriate authorisation), and this is the real potential in the new generation of eBusiness applications. In addition to this accessibility, increased flexibility of the services is presenting new business models: for example, new per use pricing and charging mechanisms are possible by using remotely hosted services [45].

In traditional enterprise computing systems the business would have to invest both in hardware and often annual or one off software licences, forming both a local technical commitment and on-going financial investment to the application, a risky process that is on a par with other large scale IT projects such as Enterprise Resource Planning (ERP) installations in SMEs [46]. Remotely hosted services by third parties therefore could potentially save on the technical investment and potential cost of using the software. Furthermore, the ability to pick and choose between services which provide the same functions and standard interfaces can reduce costs in such applications. This is of benefit to SMEs, as complex processes can be broken down and service-to-service competition can reduce costs.

Despite the potential that these emerging standards present to SMEs in terms of new eBusiness opportunities, the use of distributed computing applications is yet to create a set of easily accessible applications that are affordable in terms of price and skill for SMEs. On a simple transactional level, the sources for much of the business data that are exchanged in many business-to-business communications are sourced locally from legacy proprietary components with non-standard interfaces [47,48]. Open standards can only be applied to these applications if the vendor or business can wrap them, presenting the interfaces to the legacy systems as standards-based. Thus SMEs are often bound to integrate using proprietary means at the local level. Vendors see this as a hook to entice the customer into more elaborate products, as integration of data in terms of business transactions is often only

presented by the vendor as part of wider software solutions which take up both financial and physical resources in the organisation.

2.5.1 SME Integration Needs in North Wales

North West Wales is a rural and picturesque place; however, it is deemed to be an area in need of economic regeneration within the United Kingdom (UK) and European Union (EU) [49]. Once the industrial mainstay of the area was agricultural and raw material production, but the decline in the slate quarries and in agriculture has seen much of the local economy begin to depend on the service industry [50]. In order to make the area competitive in the global marketplace and to encourage the growth of more skilled industrial sectors, the area is currently benefiting from EU Objective 1 funding. The object of the funding is to "promote harmonious development" and aims particularly to "narrow the gap between the development levels of the various regions, thus the funding is a key indicator of an economically underprivileged area." [51] A key aim of this regeneration project is to develop the IT skills and technology of businesses and people. The funding is thus centred on the stimulation of eBusiness and e-commerce practice by SMEs in the area.

However, evidence gathered in this work and other studies suggests that many SMEs in the region have yet to embrace eBusinesses, either at all or at least on a successful level. Thus, there is a real danger and huge probability that businesses which attract funding to aid eBusiness choose the wrong solution for themselves and waste resources.

EBusiness can be defined as the conduct of business on the Internet, not only buying and selling but also servicing consumers and collaborating with business partners [52]. Automated business-to-business integration for purchasing and sales is a more valuable and established area of eBusiness when compared to business-to-customer revenue volume in the UK. In 2004, within the UK, sales over information and communication technologies (ICTs) other than the Internet (such as Electronic Data Interchange (EDI)) were still nearly three times the value of sales over the Internet [53]. In 2005 this figure rose by 11.5 percent, from £183bn in 2004 to £204.1bn in 2005 [54].

EDI in this context is seen as automated business-to-business electronic commerce. About the same time as these figures were found, 55.3% of SMEs in the Objective 1 areas in Wales reported they had a Website compared to 60.8% in non-Objective 1 areas [55]. On face value, this suggests that the majority of SMEs in Objective 1 Wales are technically skilled or have access to skills to establish a Web presence. However, the same set of statistics shows that 27% of Welsh SMEs conduct online sales and 10% receive payments online, while only 2% of Welsh SMEs in the survey used ICT-type EDI commerce [56,57], a form of eBusiness that accounts for most e-commerce revenue in the UK. It is therefore fair to say that the majority of Welsh SMEs have not adopted ICT-type EDI despite the UK trend to do so.

For many businesses, common barriers to EDI are a lack of trust in the security of eBusiness mechanisms, the cost of implementing a solution, and the complexity of integrating legacy systems [58]. The investment in support

and development to build these types of system has made ICT-type EDI e-commerce available to larger, better resourced multi-national UK companies. Therefore, the majority of electronic trade in the UK is between small areas of large world-wide businesses. On initial examination, this is as an advantage that benefits larger companies in terms of efficiency of business process as opposed to their small scale competitors. However, this integration need is now filtering down the supply chain to smaller companies which are trading with the larger multi-national organisations.

2.5.2 The SMEs this thesis is aimed at

Typically, a SME is defined as an enterprise with less than 250 employees [59]. However, this definition does not define the characteristics of the SME and if it has a highly skilled staff in terms of ICT or a staff with little knowledge and skills of ICT. In terms of competitiveness in global marketplaces SMEs have been further categorised into groups by various researchers depending on the SME's capabilities to compete [60]. This work commonly yields a group of SMEs in various sectors which are seen as lacking resources to allow them to compete on a sufficient level [61].

Much of the research into the issues handicapping SMEs within global trade has been conducted before the popular development of eBusiness. EBusiness has accelerated the globalisation of trade and introduced a significant barrier to the SME wishing to compete in global marketplaces, as

an additional technological resource level has often to be met for SMEs to both join and compete with other businesses within eBusiness communities.

It is thus important to distinguish what types of SME the BDIFS application is aimed at; this thesis is aimed at the SME with few ICT skills, and a case study of one such SME is in Appendix A. In order to reach a formal definition, the SME in Appendix A,is used as a model, one that outsources its IT support or has minimal IT infrastructure including an absence of a Management or Enterprise Information System. These factors are a big enough barrier to preventing the business from integrating their data into an automated eBusiness system using their resources on site. This definition, therefore, includes businesses where basic desktop PC knowledge is present as well as enterprises that can support typical office based applications on site.

2.5.3 Integration pressure from above

As discussed, more and more SMEs have to adopt their IT structure and internal workflow to incorporate the automated needs of a larger customer whose main concern appears to be the protection of their workflow, often to the detriment of SMEs. This project has shown that this negative effect of eBusiness integration on SMEs is present in North Wales and has a negative effect on the internal business workflow and efficiency of an organisation [62]; therefore stemming this tide, reducing this pressure from above in the integration process, and presenting SMEs with a platform on which they can exert influence on the eBusiness integration task, is the eBusiness integration challenge addressed in this work.

The initial analysis of business data workflow in collaborating partner sites as part of this work has revealed a clear direct link between business successes and integration. The research was conducted with SMEs in the North Wales area though links with the University, dissemination events and the projects Website [63]. In all, 34 SMEs took part in the survey, which was supported by a detailed case study of Company A. This research is also supported with other statistics from the Welsh Development Agency, which supports the view that the companies which were better integrated had formed reliable and strong links with customers through long-term agreements. Integration on these sites was usually in the form of business-to-business automated transaction data integration with their longer term partners, stable in nature and in place for a long period.

Conversely, in the cases where the businesses had shorter term relationships with customers, various degrees of integration were achieved. Here, the main factor behind working towards integration was often to satisfy requirements that would enable these businesses to win contracts with larger customers. Unsurprisingly, it was discovered that many companies were prepared to cut corners in order to achieve this integration. As with legacy systems, integration in the messaging exchange process was a major factor stunting design. Fig 1 below demonstrates the four most common non-seamless integration approaches that were encountered in the project.

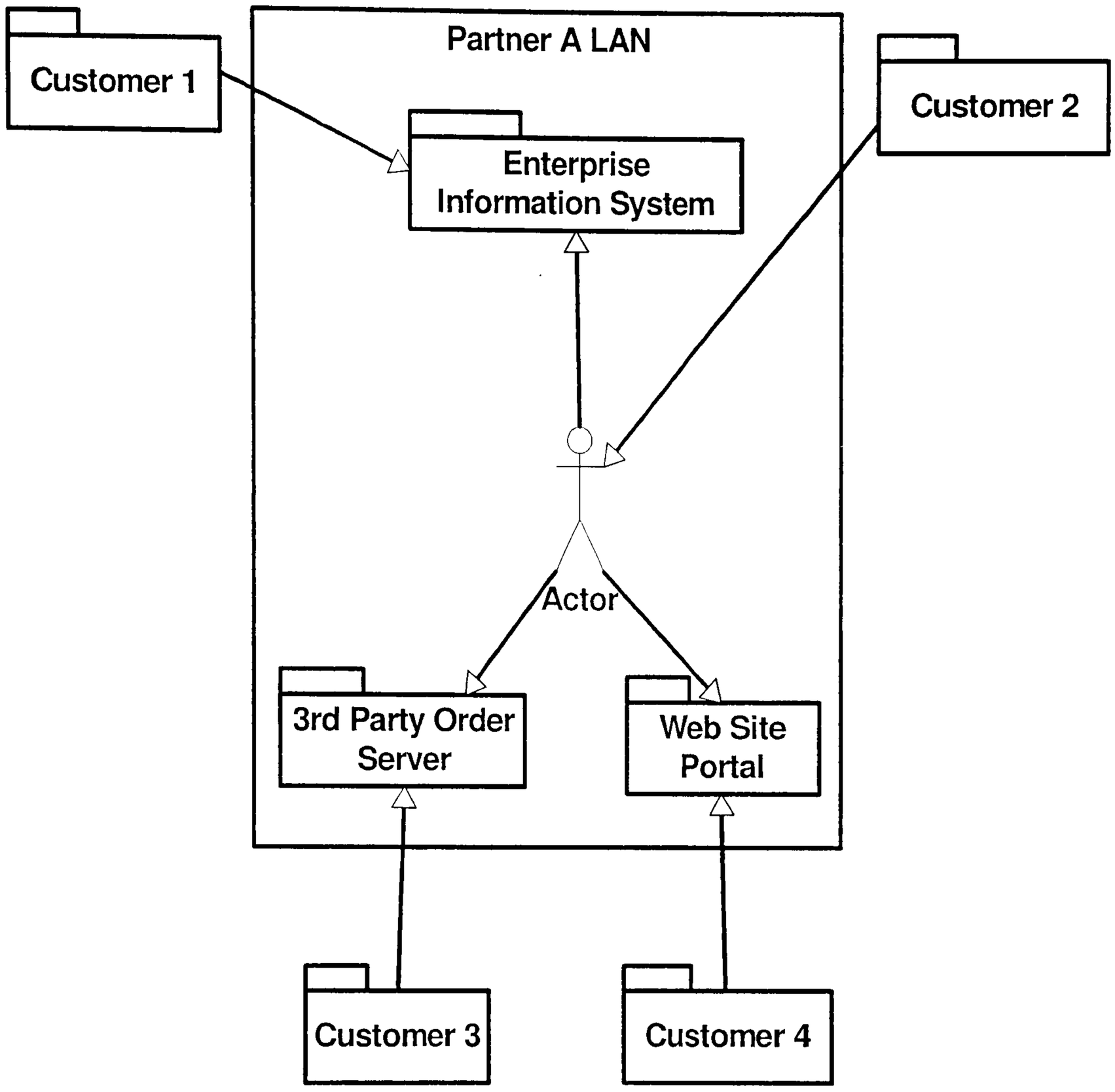


Fig. 1: Common Integration Scenarios for SMEs in North Wales

For each of these cases, the following identifies the process by which the customer business initiates data exchange with the Partner A business.

Customer 1 scenario: the customer has been granted access to the company LAN, usually via a dial-in link, to take remote control of Partner A's legacy system, allowing the customer to query the current stock and add orders to the company's legacy system. This is a very dangerous process in terms of

security, yet the BDIFS project identified a number of businesses following this approach because it was both simple and cheap.

Customer 2 scenario: the most popular manual integration scenario. Here the customer places an order through direct contact with a member of Partner A's staff. The order is sent via e-mail, fax or over the phone. Once received, it is then keyed manually into the business's legacy system. Although the process is labour intensive, it does not compromise the host company's LAN security. However, managing the data that has been sent in various formats is a security consideration. As larger businesses force smaller companies to integrate electronically with them, this type of integration is now in decline.

Customer 3 scenario: this is commonly encountered when the contract is obtained from a large multi-national company with established eBusiness systems. This integration trend, alongside the expansion of the global market place and outsourcing, has been increasing. Such businesses usually require seamless integration and many customer companies invest in a module or server to comply with such a requirement. The costs of this investment are often offset by the revenue generated by the contract. With such a system in place, the order is placed on a cut-down version of the customer's legacy system. The data is then retrieved and keyed manually into Partner A's legacy system. Businesses which cannot afford the initial investment in this integration scenario may not qualify to work with the partner. Businesses that adopt this approach often are left with extra burdens on resources. They often

have to support the new software along with adding an extra step in the internal workflow.

Customer 4 scenario: this is a typical approach observed in the BDIFS project for many of SMEs which receive orders on a Website. Here data is retrieved online and then manually keyed into the legacy system at Partner A's site. This is a labour intensive process and does not provide the real-time information that many online consumers need.

IT systems which are badly adopted or adapted to enable automated eBusiness processing in SMEs can have detrimental impacts on a company's overall performance. Ninety one percent of small businesses rate IT failure as having the greatest ability to compromise their company's good performance. A further twenty five percent of those surveyed opined that IT failure is the most common cause of businesses falling flat [64]. However, in order to compete in a marketplace where the vast majority of trade is conducted in automated eBusiness systems, SMEs have to adapt.

Thus the pressure on SMEs to integrate is large, and the financial impact for SMEs with a poorly integrated system is serious. If an SME changes customers then it is likely to be faced with a new integration challenge. For SMEs with multiple customers on short-term or seasonal contracts this is a huge problem. As the more complex architectures presented in Fig 1 develop, such SMEs become less efficient and at the same time their potential collaboration base starts shrinking. These are SMEs targeted by this work. To

overcome this problem, SMEs need to use a solution that is suited to their specific needs. To date, no solution exists. In order to create such a solution it is necessary to make an assessment of the potential technologies.

2.6 Integration Technology

In order to design a system to address the integration pressures faced by SMEs, current technology in the EAI area will now be discussed. This investigation will look at the core elements and areas of functionality in the EAI process, with the aim of extracting the key functionality needed for a system to suit SMEs, and addressing the problems faced by many SMEs in the previous chapter.

2.6.1 Legacy Solutions

Central to integration of distributed computing applications is the issue of making applications and machines understand each other despite differences in hardware, software and program logic [65]. In order for an eBusiness system to function, distributed and separate computing applications and hardware need to communicate with each other to achieve specific goals. Various methods exist for application integration in respect of the software engineering process, and range from more hardware level languages such as CORBA [66] to application level technology like Enterprise Java [67].

On the whole, however, integration is achieved via methods that are specific to the legacy software that is storing the data, therefore integration has to involve some use of the choice of legacy application technology that is associated with the data. Not only do the data formats vary between legacy systems, but also the logic differs in the way data is expressed. Thus at a local level, integration has no silver bullet and is out of reach of the emerging distributed computing standards.

This picture is not good for SMEs that need a solution to integrate various vendors. Currently the only means of achieving legacy integration is via software supplied by the vendors or by customisation via consultancy. There are no solutions or forums that address the issue of multi-vendor integration. The investigation and provision of such means to aid local legacy system integration is a key aim of the BDIFS project.

2.6.2 Middleware

The software technology that performs the process in-between the legacy integration is commonly known as Middleware. The designs in this work fit into the Middleware role within the business integration scenario, with support also given to legacy application integration. In order to create this technology, a choice is needed between the various technologies and architectures that are capable of presenting effective Middleware.

Over the years computing architectures have evolved with significant advances in making distributed computing applications available to a wider audience. For example, the sharing of data between machines has evolved from the use of specialist thin terminals/mainframe architectures to mass-produced client-server technology [68], opening the technology up to a larger number of people and businesses as the cost of the technology and the skill levels required to use it reduces. As the design technology behind applications changes, so does the range of functions and business models using it; for example due to advances in computing technology, file sharing is going through a transformation in how files are perceived and shared.

Socially and in some business environments the file sharing process is beginning to embrace Web technologies. A good example of this is the explosion in recent years of the use of P2P networking topologies to share data over the Internet [69]. The integration of business data needs a similar injection of new technology to move it away from the LAN-based client-server model towards a model that is more suitable to use the flexibility of the Web.

In business computing (like most other environments), software development has been largely limited until recent years to the Local Area Networks. This localization has led to the development of software that did not have to look as far as the LAN. Creating various sets of standards for business data specific to vendors and individual businesses, within the applications that handle business data in environments such as warehousing, ERP, etc, is the major block at integrating business systems using Web technology.

Application servers are one approach to join distributed systems, and can be viewed as Middleware components that subscribe to the server-based design. These servers often providing distributed integration technologies around technology such as Enterprise Java Technology. A popular product in the Sun family, enhancing the Java technology, is the Enterprise JavaBeans (EJB) specification [70]. EJB is one of the several Java APIs in the Java Platform, Enterprise Edition. EJBs are designed to act as server-side elements that contain business logic to aid integration. Use within application servers is their ideal application. Here EJB can interact with other Java services including Java Cryptography Extension (JCE) [71] and Java Authentication and Authorisation Service (JAAS) [72] for security, naming and directory services via Java Naming and Directory Interfaces (JNDI) [73] and also the Java Messaging Service [74]. This use of EJB and associated technologies brings a mix of application server technologies business logic elements such as transaction processing and currency control, but within server environments.

P2P architectures for use in helping business communication have been available since the early days of computing, before the development of protocols to support multi-user networking environments, and they were present in the early days of distributed computer applications in forms such as the Usenet news server [75]. It is as an architecture where peers have an egalitarian relationship and communicate via direct interactions between peers [76]. Past methods of Peer-to-Peer (P2P) integration are present in

business application technologies such as one-to-one designs of EDI connected by fixed networks. However, as previously mentioned, the accessibility of the internet and information on the World Wide Web has revitalized the use of P2P technology.

The P2P approach to computing is now often home to some of the more radical distributed computing applications, that are emerging to challenge the control of information sharing enabled by centralized client-server architectures [77]. The strength of P2P computing frameworks is that the data passed within them can be secured to be shared only by the sender and receiver, which is ideal for business messages, and makes traditional EAI a potential target for new applications using P2P architectures. Business applications have emerged in this new generation of P2P technology supporting often lightweight marketplace applications [78]. A testimony to the extent to which this architecture stands separate to the client server model, is illustrated by SUN's effort in addition to the EJB, has establishing a separate P2P application technology JXTA [79].

SOA's, as touched upon earlier, go hand in hand with Web Services [80]. The strength of the partnership is its ability to abstract the computing frameworks that present the services to the Web. Within service-oriented architectures, the use of services in a loosely coupled way is an attractive change to the P2P or client-server networks, where the nodes have to be members of a specific framework in advance.

Middleware, therefore, sits firmly in the middle of the business integration process and can be expressed in a variety of software architectures. The aim of this work is the creation of Middleware to specifically address the challenges facing the SME, as outlined in Chapter 1.

2.7 Data Management

2.7.1 Standards evolution

The key function of the Middleware layer in a business integration environment is essentially a value added message routing between the local environments that host the legacy systems. The way that business messages are expressed is central to the integration challenge, and the automation of business process between companies. Various business messaging standards exist and are used to present data to suit specific target applications, thus making the job integrating different systems a complex process.

This is because the initial business software, and therefore integration technology, predated the popular emergence of the World Wide Web. This led to standards being developed at local levels at a time when integration between LANs was rarely an issue. Initial standards for Electronic Data Exchange emerged, but were not fully adopted. In 1979 the American National Standards Institute (ANSI) Accredited Standards Committee (ASC) X12 was the first group to look into a standard for EDI [81]. Initially the use of EDI was within fixed WAN and VAN networking environments that used

expensive leased lines, an approach popular with large companies, such as multinational manufacturers, to increase efficiency and save money [82]. However, within these environments EDI soon evolved and mutated (in some cases away from the standards guidelines) into similar standards to suit specific legacy systems that were originating the business data.

The development of the World Wide Web has driven down the cost barriers associated with the supporting network of a traditional eBusiness infrastructure. No longer are costly leased lines and fixed network infrastructures needed, thus increasing the accessibility of many localized systems. On a message level, unlike the issue with legacy systems, where local integration is key, messages of different formats can be translated from one format to the other, prior to local integration by specialist servers on the wider network. A process that is being supported by the emergence and development of stronger messaging standards is based around XML to express specific business communications [83,84]. Commercially, the main means of achieving this is via the use of the business process server, an example of application integration products such as Microsoft's BizTalk [85].

2.8 Integration Technology

Many products exist that aim to make the automated integration process a reality; however, they only reach up to the legacy application integration part of the integration process. Furthermore, these solutions available to aid business-to-business integration have begun to embrace Web technology and

standards. This is presenting SMEs with an opportunity to use the resources of the internet, to incorporate levels of service orientation in their integration.

2.8.1 Software designs

In terms of commercial application, the software most used for business-to-business integration in eBusiness environments, is either systems developed in-house for companies with plenty of resources, both technical and financial, or off-the-shelf packages by companies like Microsoft (BizTalk is a Microsoft product). These products present solutions in the client-server mould, whilst the others work around the client-server architectures. Whilst custom solutions can achieve the last hop of local legacy system integration, they are often inflexible, unscalable and not suitable to meet the needs of SMEs. However, products like BizTalk are able to group multiple partner messages and queue them for an application to perform the last hop of integration when ready. Servers like BizTalk have had technical success integrating with similar systems in terms of messaging. This has been achieved by taking the emerging WS technologies and making them behave in a uniform and deployable way within the proprietary frameworks.

2.8.2 Enterprise Application Integration (EAI)

Often proprietary integration servers (under the banner of EAI technology) use open messaging technologies like XML to aid the integration potential of the technology. However, as mentioned, this is achieved locally through the servers' rules within the server itself. This makes the translation and workflow integration process between businesses less transparent and flexible, reducing the ability of SMEs to modify or customise the process. For SMEs without the financial investment and skilled staff, implementing such a solution is out of reach, as solutions like these still involve licensing costs, consume local computing resources and rely on local support. Furthermore, these types of servers often have the ability to do far more than is needed and therefore are complex and expensive to support, thus rendering the EAI server as nothing more than a black box [86,87].

EAI systems that are available for SMEs with low resource levels are scarce. The main accessible options for SMEs are to integrate in non-standard means by either using unsuitable technology or workflow, often to the detriment of the businesses (as in Fig 1), or by investing in custom adaptation of the IT infrastructure of the business to suit a specific integration need. When integration needs change, or there is a change in the company's IT infrastructure, the custom integration may have more costly consequences down the line.

Open source EAI options are often cited as possibly cheaper and more accessible, increasing the SME's ownership over the software. For example, in the areas of e-mail and file sharing, there are open source servers available

which have helped some SMEs develop infrastructures at lower cost and with greater ownership [88]. The advantage of open source is that the cost is often a fraction of a proprietary alternative; in addition, due to the open nature of the product, the code is accessible to anyone and can therefore be modified by any skilled software engineer. An example of this in the EAI area is the OpenEAI project [90].

However, despite the use of open source software in the area of EAI, which initially sounds appealing for SMEs, it is argued that the degree of support needed in the whole process makes these products not a simple issue of purchase price [89]. For SMEs this type of EAI requires a local server presence and this complexity still remains a problematic point for many SMEs, as local deployment requires hardware costs, design costs, ongoing support, and greater training of staff. They can also still be black boxes.

2.8.3 Application Service Provision (ASP)

A solution to the local deployment issues is to host the types of services off site in the ASP deployment model. Common ASP services tend to focus on industry, but attempts have been made at producing wide scale, remotely hosted projects like the LAURA project [90], which is designed to encourage collaboration and business-to-business co-operation in specific trading zones. This research provides two means of controlling integration servers: either the server is controlled by SMEs, or the data from SMEs is sent to openly

accessible, remotely hosted integration networks led by organisations outside the company. The problem with these projects, however, like many other large scale EU projects, for example TrustCom [91], GRASP [92] etc, and large scale design models like the Digital Business Ecosystems [93], is that they fail SMEs by trying to do too much, and focusing on the more complex technical challenges. Within these projects are large software vendors and EU pressure to break research barriers. Usable solutions for SMEs are therefore a long way off and are likely to be bundled in packages by the larger industrial partners.

Outside of research projects, ASP has failed SMEs by having few, but frequently changing trading partners. The only example of frameworks which have been developed that offer the ability to host business services remotely are often categorised as Business Service Networks (BSN)[94]. These Web Service based BSNs are closely linked to the development of markets and are therefore largely industry or vendor-specific [95]. As computing resources realize their potential in a global e-commerce marketplace, existing BSN designs reflect this in their focus on business trading communities. A good example here is the eHealth industry, where the user community, applications, users and profits are easily visualized and developed [96]. This application presents promise to single industry SMEs with stable partners who wish to use ASP style integration solutions. But for SMEs that just need basic help integrating with frequently changing partners, in ways that are often non-specific with regards to industry or software vendor, BSNs are yet to meet the challenge [97]. Current BSNs are therefore no use to SMEs which have a

flexible customer base and need to adapt to basic but varied integration needs, rather than detailed business functions of the market place.

Even so, the BSN approach has its plus points, such as producing new integration business models, often resulting in cost savings for the individual business, whilst providing opportunities for specialists to develop saleable applications to add to the network. BSNs have real potential for a pay-per-use or monthly subscription incentive for service development and the use of BSNs as an incentive to aid vendor-specific development is a very attractive proposition for SMEs. However, another problem for these businesses in harnessing the power of BSNs is the interface to the network. This has to be easy to integrate with and often SMEs do not have this automated integration capacity. The technical skills to call on the Web Services are often not in place in SMEs and the BSN often does not have the ability to adjust its range of services to SMEs. A BSN needs to be created that is both dynamic and flexible, so that it can present the required interfaces for SMEs and a flexible integration workflow to achieve SME integration.

A possible way to achieve this is via dynamic service management which is part of emerging distributed computing design within the Web Service community. A BSN that supports SMEs will be quick to evolve, based on collaboration and ease of use with open membership policies. Such designs of BSNs using P2P technology and Web Services can create a network where negotiation can be made using multiple nodes, enabling service selection to

aid SMEs and create BSN environments specific to the business workflow needs [98, 99].

The weakness within business service networks, as they are emerging with existing applications, is that they are developed along commercial guidelines, appealing to specific industries and data types. As mentioned earlier, this presents a vertical solution that is hard to apply to a wide range of SMEs in varying industries.

From the above discussion, it is clear that both the Web Service type and EAI type projects have failed the typical underprivileged SME, like the ones in North Wales, in not producing a simple, easy to implement business-to-business integration environment. This is partially in their specification and partially through their complex handling of various transactions and business processes. They fail to address the specific needs of a business that needs wide ranging integration format support, within a solution that presents simple workflow and transactions. Therefore the goal of this work, to present a practical framework to address the needs of SMEs, is both significant and important.

2.8.4 Business integration to suit SMEs

The key challenge of the integration process is the merger of different business data, with varying data formats, and business logic in a way achievable by SMEs.

The main EAI types of software are, as discussed, the only real successful means of achieving multi-partner forms of integration. As explained, though, for SMEs this approach requires too many resources. The possible solution of remotely hosted ASP EAI type solutions has yet to evolve and currently only Business Service Networks are looking to possibly help SMEs, although they tend to be industry specific and don't support multiple partners. Both approaches seem to be too detailed in the direction each one is pointing: a merger of the two methods is the only practical way to solve the problem. In order to make the EAI more accessible, it needs to be moved away from its use of the fixed LAN-based client-server approach to integration.

In order to begin the design process, closer examination is needed of the needs of SMEs. With respect to the functionality required in an integration platform (to suit the basic SMEs integration needs), the solution really needs to be quite simple. However, complexity is a real issue in the integration process, as the temptation in designing integration systems is to focus on various possibilities within an integration workflow. This distraction is the problem with attempted solutions like the LAURA project. Such projects are a missed opportunity to present SMEs with a real solution, as the interaction with these interfaces at the end of SMEs does not need to be complex to meet the demand of many SMEs; they just need the ability to perform simple transactions. An SME with, for example, a single manufacturing facility, but a

constantly changing group of customer partnership may only need the ability to be able to exchange simple purchase orders and invoices.

As a starting point, SMEs need simple message exchange and functionalities when it comes to the integration. Although it may initially appear that this approach follows one of the simplest paths, the enablement of SMEs in the business workflow is essential. Even a simple business-to-business messaging solution that SMEs can use will give SMEs community the possibility of not only greater trading potential, but also a real seat at the integration technology table and a louder voice in the future direction of the technology.

2.8.5 Building a solution to suit SMEs

The most suitable integration architecture that has been discussed so far for the SME is the ASP approach. It appeals as it is both cheap for SMEs to implement and scalable. It takes the strengths from EAI and moves them out of the LAN, and has the possibility of reducing complexity by splitting the EA functionality into multiple separate services. Thus the BDIFS prototype systems will aim to create an ASP-type solution using the strengths of the popular EAI solutions.

As discussed, the EAI approach is the most popular means of business-to-business integration. However, it is not suitable for use in the type of SME the

project is working with, so it is necessary to identify its key functionality and move this into a solution that is more suitable for SMEs. This functionality is shown in the simple integration process diagram below (Fig. 2).

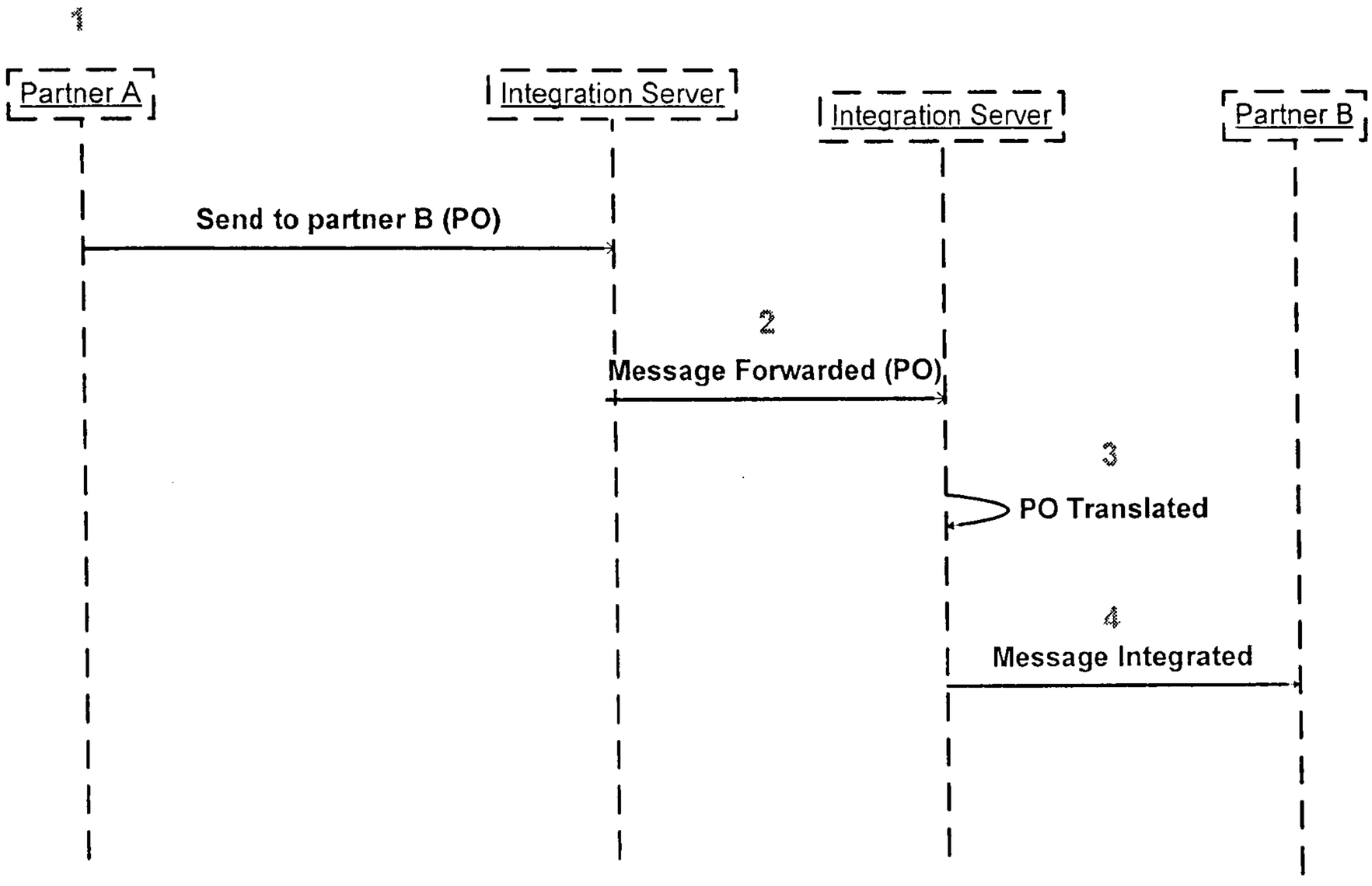


Fig. 2 Application integration using the Enterprise Application Integration server approach.

Within Fig. 2, four key areas of functionality are presented using the integration server approach.

1. Human interaction to join the framework: this stage can easily be overlooked but the procurement and configuration of the software to interact with the local legacy software is a major task.

2. Data transport: here steps 1 and 2 are concerned with the transfer of data; step 1 inside the LAN and step 2 outside the LAN.
3. Data transformation: steps 3 and 4 take place at the partner site's LAN and are concerned with the translation, transferral and integration of the message in the partner's local system. In the EAI system this process can be repeated in reverse.
4. Data integration: this stage would occur locally at the partner site when the message arrives from the BDIFS framework, which will use an integration script to upload the data to the local database.

These areas of functionality illustrate how the EAI approach has essentially four main parts to it. Thus in order to design a distributed computing system that reflects the same basic functionality, it is likely that the four areas of focus will be used in the new design. With this in mind, the ASP approach to integration will now be looked at. The ASP approach to integration is reflected in many of the BSNs that do a more industry-specific and limited job when compared to EAI (Fig. 3).

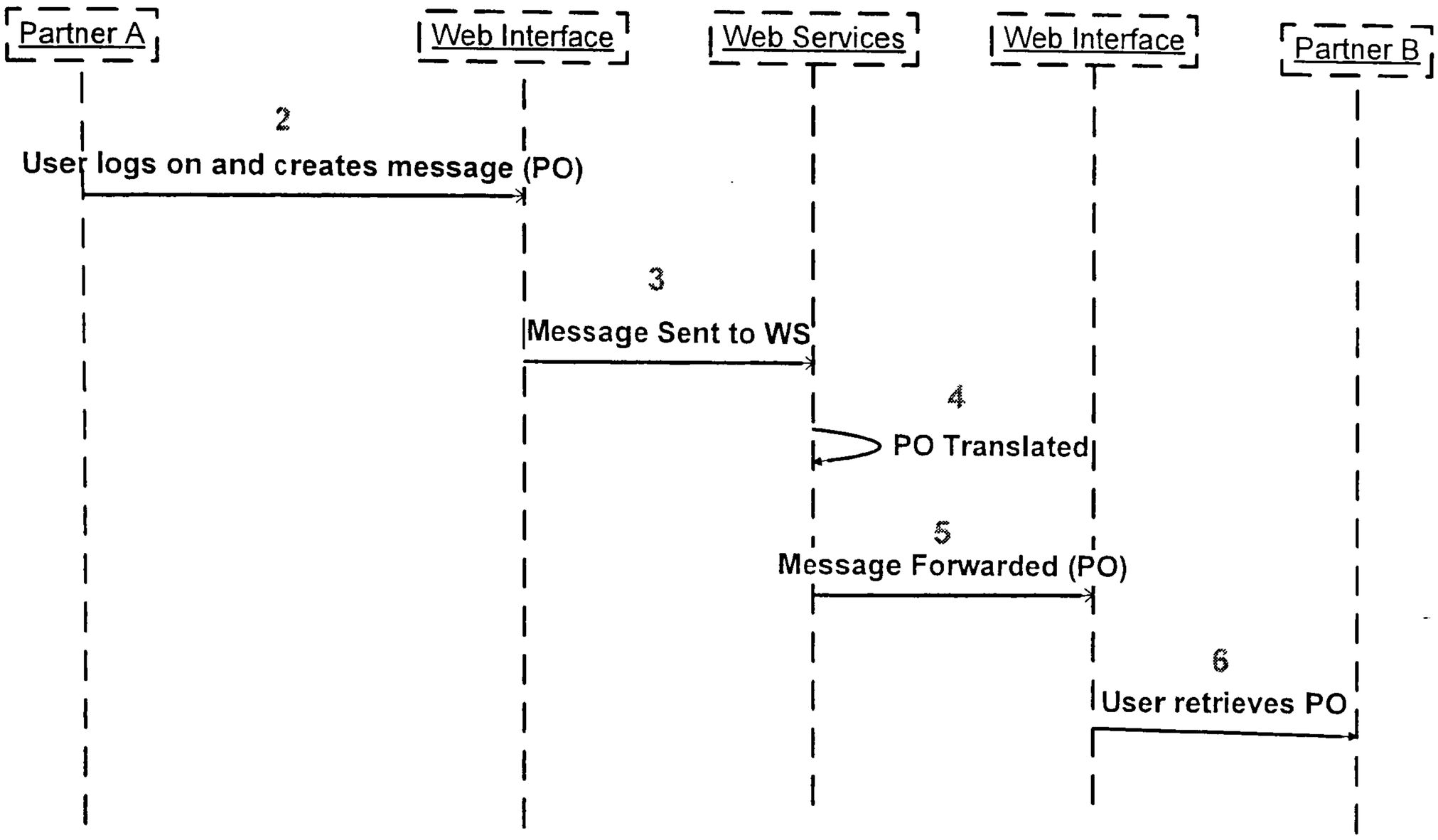


Fig. 3 ASP approach to integration.

As Fig. 3 illustrates, to date common Web service-enabled approaches to EAI have followed a similar messaging structure. Often the Service provider creates a community, usually within a specific industry, to share data. This data could be used for reporting, transaction processing or job acquisition. As discussed, the initial weakness in the system for an SME is that the approach is essentially industry specific and therefore cannot be a general solution for all SMEs.

Looking at Fig. 3, from the decision to exchange data made at point 1, technically the stages 2 and 6 are left to the individual company to resolve. Some ASP services have ways to link popular back end systems, but often this is no use to the integration needs of the legacy systems that many SMEs

rely on. As is illustrated in Fig. 3, steps 2 and 6 are often performed by human interaction with a Website. However, the messaging and translation steps 3-5 are outside SMEs, and thus do not require any IT infrastructure changes, as opposed to the EAI approach.

The main areas of focus in the Web Service system are the four previously mentioned in relation to Fig. 2, with the exception of the message transport and transportation that occurs within the same area of control i.e. the Web Service. This leaves three main areas of focus.

1. Human interaction to join the framework: involving human configuration / procurement of the system and extraction of data.
2. Data transport / transformation: as opposed to the EAI approach the services that translate, send and receive the data are part of the same network.
3. Message integration: in the same process as before, the user needs to take the data that is received and integrate it into the target database with assistance.
4. The model in Fig. 4 illustrates how the three solution areas can be grouped into separate areas of control:

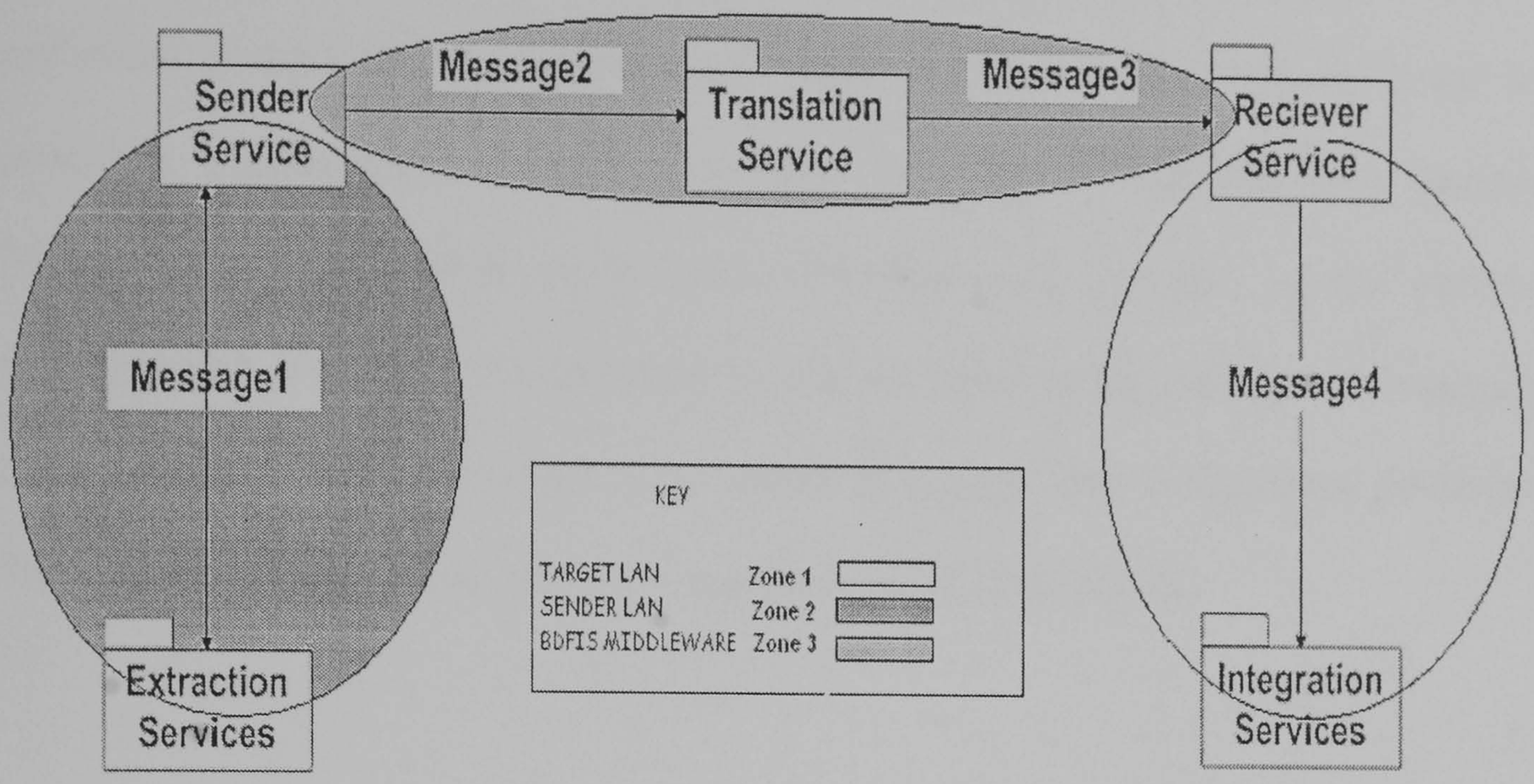


Fig. 4 Three areas of integration control in the ASP model.

The three areas of control in Fig. 4 will be elaborated on when discussing domains of control later on, in the design chapters in this work. Essentially, however, the Web service approach illustrates the burden of transportation and translation of data that can be coupled to, and moved away from, SMEs to an external Service Provider situated in zone 2.

From an internal IT perspective, this seems to be attractive as the approach will require minimal changes to the IT structure of SMEs, since the EAI approach requires both server acquisition and administration. The messaging between the EAI servers will also require secure networking to allow data between the EAI servers in a manner that both protects the data and the network in which it originates. Using a simple Web client in the ASP model and secure HTTP the problem is reduced.

By focusing on these three integration challenges and attempting to move the technical design focus of them away from the individual SME as much as possible, a potential solution is possible. This solution will require a system that is more flexible than current Web Services can provide, as the service relationships and relations between zones will need to be created dynamically on demand to support short term resource usage and integration demand. This design will form the technical basis of the BDIFS design.

2.9 Summary

The weakness in current integration technologies is that they initially require major upheaval at LAN level. For SMEs this has the potential to stifle existing business workflow and compromise the current IT infrastructure. The use of remotely hosted services to aid integration has the potential to aid SMEs by removing the integration task from the LAN of the SMEs. However, to date, there are no solutions to present a framework that helps SMEs to integrate in multi-industry and multi-vendor formats. The LAURA project and other distributed research projects looking to create scalable environments have failed in producing practical solutions for SMEs.

SMEs need wide ranging integration solutions, but often only to support the more basic transactions. Thus the design of the integration software does not have to be overly complex as many current solutions are. SMEs need software that is simple to use and acquire: ideally software that can be

administered and hosted remotely away from the LAN of the SMEs. This chapter has presented this first practical step in the initial contribution to science that this thesis makes. This contribution is the initial design presented here to aid SME integration, and this will be built upon in the coming chapters. The approach adopts a view that a VO can be created to aid business integration and, using this model, the thesis is the first project to address power control issues in real SME eBusiness partnerships. The project is also the first to present a solution to address the eBusiness integration needs of SMEs in North Wales.

Chapter 3

Related Work

The designs in this work will use Web-based services. These will be presented in two separate design architectures. The design architectures are P2P and Web Service design, with the latter being the final and more in-depth design of BDIFS. The web service design is based on emerging Web Service standards, as illustrated within the OGSA development architecture. The Globus toolkit [100] has been used to create the Web Service design. For the BDIFS P2P testbed the Project JXTA P2P toolkit is used. In this chapter the reasoning behind these choices is given along with a description of the use of technology and the position it has in respect to the BDIFS application.

3.1 Grid Design Toolkits

There are various Grid design technologies available. These range from Grid solutions-based provided commercial companies like IBM [101], to open source community-based technologies such as provided by the Globus

Alliance [102]. For the purposes of this project the choice of Grid toolkit was based on two main criteria. The first of these was business viability; in order to assess this, selection was based on the stability of the product both in technical and future development: for example, if the technology has a strong support base and user community. The other criterion was the support of open source technology, which is important because it gives SMEs more ownership and control of the technology, and also because the aim is to release BDIFS as an open source project.

At the time of writing, the latest set of standards for the development of Web Service applications is the Web Service Resource Framework (WSRF) [103]; this is based on the OGSA. WSRF is supported by the leading industrial and academic computing companies as an open means of future Grid technology development using service-oriented architecture. The two main designs of WSRF are supported by both Microsoft and Globus. The Microsoft design is available freely as a download that works with the Microsoft.net [104] framework and is aptly named WSRF.net [105]. Development of WSRF.net is sponsored by Microsoft and includes academic institutions such as the University of Virginia [106]. The other design of WSRF is provided by the Globus group. Globus have released a WSRF compliant toolkit in their latest Globus release, which is Globus toolkit 4 [107].

Whilst the WSRF.net design of the WSRF framework uses the IIS Web server [108] provided by Microsoft, the Globus design of the WSRF framework uses the open source server Apache Axis [109]. GT4 provides software libraries

that support the main infrastructure needed to construct a set of services in order to provide Grid Middleware applications, supporting resource discovery, security and representation of state within the WSRF framework. Largely due to its open source server base as opposed to WSRF.net, GT4's compliance to WSRF compatibility, and researchers' familiarity with the technology, the Globus Toolkit version 4 was chosen as the design technology for this prototype.

3.1.1 The Web Services Resource Framework

The WSRF guidelines are designed to define the necessary means to provide standardised access to and management of resources that are exposed via Web Service standards. The key to the aims of the WSRF framework design is the new standards that they are promoting to support the development of Web Services.

The standards include mechanisms for getting and setting values of one or more properties, as well as querying across these values. It is possible that a resource can be destroyed on request or after it has exceeded a certain lifetime (which may be extended at any time). Interested parties may be informed of changes towards the resources by a notification event.

The specifications also allow for joining Web Services with the WS-Resource into groups, similar to a registry. Such groups can be restricted to what services are allowed to join, meaning whether certain Web Service interfaces exist and/or certain (resource) parameters are exposed. A service group can

act and communicate on behalf of its members, i.e. the services may be addressed via the service group they belong to.

The Web Services Resource Framework comprises the following specifications:

WS-ResourceProperties: this standardises the operations of a WS-Resource, as well as the structure of the resource properties document, which represents a view on the WS-Resource's state.

WS-ResourceLifetime: management of a WS-Resource in terms of immediate and delayed destruction. Extension of lifetime is described within the specifications.

WS-ServiceGroup: specifies the standard resource properties defining contents of a service group and how to access details of an entry. Note that it is not an objective to represent the function of a member in a group. The constraints for membership are expressed by intension using a classification mechanism.

WS-BaseFaults: the purpose of these specifications is to define a common form for error messages. Each BaseFault specifies a reference to where the fault was generated, and a timestamp of when the fault occurred.

WS-RenewableReferences: the WS-RenewableReferences specification defines the mechanisms to renew a WS-Resource endpoint reference that becomes invalid.

As will be explained in more detail later on, the use of standards in the WSRF toolkit are central to the implementation in this work. The use of WS-Resource Properties in particular enables the most significant contribution in this thesis, as this standard allows the creation of dynamic VOs, and therefore allows the development of the VO structure in this work based around dynamic and stable (base) VOs.

3.1.2 Other Relevant WS Standards

In addition to the standards above, WSRF also relies on [WS-Notification] [110] for event notification and [WS-Addressing] [111] for referencing a WS-Resource and the appropriate resource properties document. These standards are the latest and most promising standards to emerge in recent years for messaging between distributed applications, and illustrate how the Web Service Resource Framework and other Web Service (WS) standards work together.

The significance of WS Addressing is in the two constructs that it defines, an endpoint reference and an associated message information header. These constructs have become typical elements of various messaging protocols

within the Web Service community. WS-Addressing provides a well-defined way to do asynchronous one-way messaging, with the ability to correlate messages and many other specifications, as well as WSRF build on and leverage endpoint references and message information headers.

WS-Notification is a collection of guidelines that define a topic-based/subscribe pattern. This is of use to distributed stateful systems in many ways and significantly improves the management of service state monitoring in these environments. As the traditional approach to such monitoring often was via a polling mechanism that creates a large amount of unnecessary traffic in the network and listening ports, this does not scale very well in distributed computing models. The WS-Notification standard is outlined in more detail in a series of documents including a white paper called "Publish-Subscribe Notification for Web Services" [112]. There are other specifications of WS-Notification, including WS-Base Notification [113] that links the notification pattern to WS-Resource Properties, and WS-Resource Lifetime, which are core WSRF standards that have already been explained.

The use of topics in notification is also a significant advance on the notification mechanisms in distributed computing models; for instance, to supplement the WS-Base Notification the standard WS-Topics [114] exists to describe specific elements of interest of which subscribers wish to be notified. Topics therefore enable notification mechanisms to express in more specific and wide ranging terms, actions, etc., of which they wish to be notified. The topics can be

described and placed in hierarchies, introducing a concept of the development of greater service-to-service understanding, and using metadata to create service semantics in relation to understanding and communication.

With respect to brokering, WS-BrokeredNotification [115] exists to define interfaces for a NotificationBroker and a PublisherRegistrationManager. The NotificationBroker is a middle man type of service, linking messages from elements that are not service providers into the messaging framework. The PublisherRegistration manager links the services to the notifications that they are subscribed to.

The WS-Eventing [116] standard is similar to notification, but instead of being based on service actions, it can be used to harvest more information to support the notification and service-to-service understanding. Eventing lets applications know of events occurring in other applications, based on a registration of interest. The key elements of the application of the standard are the event sink (source service) and the process of subscription of the source service to another service termed the event source. WS-Eventing is not used in BDIFS but is of potential interest in future designs.

Finally, another significant WS standard is WS-ReliableMessaging [117] specification. This is to ensure delivery of messages designed in the case of software and network failures. Using an RM Source and an RM Destination, the protocol tracks messages and helps ensure delivery. Interoperability using

a SOAP binding in the standard allows integration with other WS standards. This standard is particularly relevant in systems where application reliability is not guaranteed, for example in a mobile network environment where the service can be subject to unreliable network connections. This is therefore of potential interest to the Web Service BDIFS design, with respect to management of unreliable service behaviour.

Other WS standards are in development and emerging outside WSRF but directly relevant to applications that use state. A good example of this is the WS-Agreement [118] standard which enhances dynamic VO formation by aiding the establishment of SLAs between services within frameworks like BDIFS.

However, the BDIFS Web Service prototype initially relies largely on the functionality in the WS-ResourceProperties to aid the dynamic creation of business computing environments, along with the destruction, brokering and notification capabilities in the WS standards. This is due to the focus of the project in this design being on the formation of dynamic workflows around VOs formed on current services stateful resource properties. This is a use that creates a network of services in the model of the ASP Grid described at the beginning of this work. As discussed, future investigation would include practical use and evaluation of standards to aid negotiation, greater event management and service composition which would involve many of the standards discussed above.

3.2 Peer-to-Peer integration

Another means of designing a distributed computing system that presents a simple and flexible computing environment is by using P2P messaging. Technology such as JXTA is particularly suited for the standardized development of P2P messaging systems. Here, the peers communicate securely over the network using JXTA security; they can be split into groups and rules assigned to different peers [119]. The movement of data in such networks differs from the Web Service-based approaches, as in a P2P environment the peers can be more dynamic and exist with no central services. The JXTA security set of standards is built into the JXTA toolkit and ensures the encryption of data between peers. This is done using the peer group and pipeID mechanisms described in more detail in the implementation section.

Current research into P2P computing and business-to-business integration has yet to focus on the needs of SMEs and how P2P software can help. In fact there is little evidence of the new crop of emerging P2P technology embracing the Internet, being developed and implemented for SMEs in an eBusiness integration solution [120]. Thus it was an attractive goal to design a P2P BDIFS system as a practical design of a P2P-based business-to-business integration framework.

3.3 Service Discovery

Regardless of the choice of underlying software architecture for BDIFS, the most important aspect is the functionality that the architecture supports. Central to service use is discovery and workflow composition. Technologies to aid this will now be discussed.

Essential to the successful creation of a BDIFS application is service discovery. Traditionally in computing, directories are commonly used to store information about services and are usually the main means of discovering them in Grid systems. Various means exist to discover and store service specific data, and the two different BDIFS designs reflect this in their choice of discovery mechanism.

Within common WS designs that are available today, the Universal Description, Discovery and Integration (UDDI) [121] protocol is the central discovery standard. The UDDI specification defines a standard method for publishing and discovering the network-based software components of a Service-Oriented Architecture (SOA). A UDDI registry's functional purpose is the representation of data and metadata about Web Services. Protocols like WS Notification are designed to complement and work with UDDI registries which are seen as core-to-many WS Frameworks.

A UDDI registry can be set up either for use on a public network or within an organisation's internal infrastructure. It offers a standards-based mechanism to classify, catalogue and manage Web Services so that they can be discovered and consumed by other applications. UDDI is flexible and allows discovery to be carried out in a way to suit the organisation. For example, specific criteria can be applied to searches made on local services, looking for specific services that support certain protocols and supporting the ability to find new services in case of service failure.

Other registry mechanisms exist in the Web Service world and Electronic Business using extensible Mark-up Language (ebXML) is one standard that has its own registry. Using ebXML, businesses are discovered via their Collaboration Protocol Profile (CPP) presented by the business as part of the ebXML standard [122]. This CPP is used to establish communications with the service. CPPs are stored in the ebXML Registry Service. By comparison with UDDI, ebXML also provides information about, for instance, business processes, business documents and business profiles, making it potentially more appealing in some applications as opposed to UDDI. However, both systems complement each other; and organisations can continue to use UDDI to inquire about businesses in the global UDDI Registry. Those entries can then be used in referring to ebXML Web Services in the ebXML Registry.

From the WS community WS-Discovery [123] is being developed to provide discovery in SOA environments, using concepts associated more with the

networking community, such as its use of multicast discovery protocol to locate services. The primary mode of discovery is a client searching for one or more target services. To find a target service by its type, a scope in which the target service resides, or both, a client sends a probe message to a multicast group; target services that match the probe send a response directly to the client. To locate a target service by name, a client sends a resolution request message to the same multicast group, and again, the target service that matches sends a response directly to the client.

To minimize the need for polling, when a target service joins the network it sends an announcement message to the same multicast group. By listening to this multicast group, clients can detect newly-available target services without repeated probing. To scale to a large number of endpoints, this specification defines multicast suppression behaviour if a discovery proxy is available on the network. Specifically, when a discovery proxy detects a probe or resolution request sent by multicast, the discovery proxy sends an announcement for itself. By listening for these announcements, clients detect discovery proxies and switch to use a discovery proxy-specific protocol. However, if a discovery proxy is unresponsive, clients revert to using the protocol described herein. The strength of WS-Discovery is its compatibility with other WS standards, but on the whole it is not as detailed as UDDI can be or as flexible as P2P.

P2P discovery in JXTA uses a similar network-based discovery format to WS-Discovery. peers send out messages across the network looking for specific peers. To aid this type of search it is common for P2P applications to use multiple search algorithms, and various designs exist for P2P searches. But essentially the discovery of appropriate services occurs via discovery messages that are sent out amongst other peers. Thus the use of other peers acting as proxies/routing peers to find others is a key part of a P2P design. Whilst the search facility in a P2P network may be more creative, the peers are less likely to represent standardised information and management of reliability is a challenge.

3.5 Workflows

Workflows are central to providing the services into applications using distributed computing. Within distributed architectures, workflows are moving from robust mechanisms for the timing of service execution to more dynamic service compositioning and execution providers.

Service composition is central to forming the flexible and dynamic environments in BDIFS that allow new business models to be presented. This type of service composition requires business collaborations driven by an explicit process model. This explicit process model is presented within the BDIFS workflow in both the Web Service and P2P designs. Various standards exist and are well documented in research publications to present such workflows in Grid computing systems.

Receiving a large amount of industry support, and acknowledged as the main standard that is both emerging and holds great promise for the future, is Business Process Execution Language (BPEL). This provides a specification in which business processors and interactions can be expressed. The standard has quickly become adopted as a vital business integration language for the expression of workflows, and multiple enactment engines now exist to execute BPEL scripts. BDIFS uses such an engine to process the BPEL scripts that express the two main workflows developed for the Web Service BDIFS testbed.

Within the Web Service world, choreography focuses on the flow of messages exchanged by a Web Service and can enhance the business processing of an application as opposed to BPEL, which explains the execution process of a service from a partner perspective. The Web Service Choreography Interface (WSCI) [126] is an XML-based interface description language that describes the flow of messages exchanged by a Web Service participating in choreographed interactions with other services. In BDIFS, such a close knit of services is not required and the orchestrated execution of services matching basic criteria in relation to functionality is enough; hence, we just use the orchestrated workflow approach to service execution.

Within P2P networks, the workflows are specific to the individual client that the user uses to join the network. Often they are simple single or multi-hop message exchanges. It is possible to develop more central peers to link peers in an orchestrated fashion to produce applications, but this is not suited to the

P2P approach. As in P2P networks, unlike static service-based frameworks, the behaviour of peers is harder to predict and the structure of the frameworks is designed for flexibility and allowance for peers to drift in and out of range. This is not good for a workflow engine which needs to map the locations and guaranteed behaviours of services in the workflow before execution.

3.6 Business Use

In order to make the BDIFS system a success, users and service providers need to be attracted to use it. A key element of this attraction is the computer human interface functionality of the system/ease of use and the business assurances and incentives that BDIFS presents. Central to the presentation of a business case is the issue of service reliability and charging. Human interaction starts at the systems portal, which in BDIFS is the central point at which users and services join the framework.

3.6.1 Service Level Agreement (SLA)

Essential to the provision of business use of the BDIFS system is the ability for the system to use Service Level Agreements (SLAs). The use of an SLA allows BDIFS to present applications in specific ways in relation to issues such as reliability, and to charge accordingly. Furthermore, the use of SLAs can be used to ensure that external service providers in the application are motivated to make sure their services behave in the way they are supposed to [124].

In relation to SLAs in Grid systems, reliability can be measured and be presented in various ways. But reliability can only be measured when based upon past or current performance, which is only an indicator as to how the service may perform and not a guarantee; therefore SLAs are important in systems where service reliability is needed. Within the majority of computing environments, as in many other types of service provision environments, the required level of reliability is associated with some pre-defined protocols that need to be kept above certain thresholds. Using a contractual agreement between user and provider ensures that service owners aim to maintain a service to a specific level of quality, as it would be in their best interests to fulfil the contracts. It also allows the user to define what they mean and want in terms of reliability when negotiating this agreement. Reliability therefore becomes a protocol approach similar to Quality of Service (QoS) on a network, but is defined by both the supplier and user of a service.

Reliability expectations can be formed and expressed within SLAs, which also include penalties for the provider or user to accept if the SLA is broken. The use of an SLA and negotiation would encourage a marketplace within the computing environment that would lead to greater effort to meet agreements and compete on terms. This negotiation and contract provision, coupled with robust, context-aware Middleware design, are the key elements in the provision of reliability in the BDIFS Grid Middleware.

However, the use of SLAs only encourages service providers to increase their own (bilateral) reliability. In an application exploiting many services, the overall (multilateral) reliability will be at best that of the weakest link, and generally will be far below that. To improve on this, the application requires Middleware support to detect and adapt to changes in the connection to, performance of and context of the underlying services. Adaptation may be as simple as switching to an alternative network provider or as radical as changing the application to achieve the business goals in a different way. The Middleware must be aware of and provide support for such changes, for example by providing solutions for handling data in an application when these changes occur. In such events new services need to be found, and finding these services in an automated way is a key topic in many research projects in the area of SLA and trust [125]. In addition, in the Web Service community several languages have been defined to aid this automated service selection process (WSLA specification language, for example, is one of them [126]). These languages all complement the service description implemented by WSDL (Web Services Description Language) [127]. This allows in general such languages to be used within a framework, allowing the management of Web Services and their compositions; user management Service Level Agreements are dealt within the WS context, and are also present in the OGSI-Agreement specifications (WS-Agreement).

3.7 Portals

Within many distributed computing systems, users interact with the software behind the system using portals. Portals are usually Websites that provide access to tools and information about interacting with and using the distributed computing system. There are many portal designs today in the worlds of academia and industry that provide a resource for day-to-day administration and interactions of large systems. Within BDIFS the portal is envisaged to provide the same functionality.

When designing the BDIFS framework, it was important to take into account emerging standards to aid the design and creation of portals. The Web Services for Interactive Applications (WSIA) [128] and Web Services for Remote Portals (WSRP) [129] OASIS Technical Committees have developed the WSRP protocol for use in portal development. WSRP is designed to aid the process by which content providers may publish their content as remote portals and publish them as WSRP services in public registries.

A goal of the WSRP standard is to make it easy to implement simple services (e.g. just provide mark-up fragments), but also to allow for more complex services that require consumer registration, support complex user interaction and operate based on transient and persistent state. WSRP will enable interoperability of portals by allowing them to consume remote portals provided by other portals or content/application providers, as well as the

sharing of local portals for remote access by other portals in a standardized manner.

This type of approach is attractive to BDIFS, where the users and services can be increased or enhanced by the joining of the portal with other similar portals. Furthermore, this interoperability has the potential, if used to make BDIFS interchangeable with other similar portals, making the use of BDIFS an enhancement to these existing frameworks, which is worthwhile to SMEs.

3.8 Security and Trust

3.8.1 Security

Distributed systems, due to their nature, are incredibly vulnerable to security violations. The degree of protection that surrounds the typical home desktop in terms of virus protection and firewall configuration is symptomatic of the risks a computer opens itself up to when connecting to distributed systems such as the internet. Simply, the main way of securing a device is by blocking intrusion and permitting valid applications to access the interfaces presented to the system. However, when resources are presented to a distributed system, extra levels of control may be needed to prevent issues such as rouge usage; therefore a security solution in distributed computing needs to consist of both physical security provision and also policy based usage of resources.

Physically, the main way of securing a resource in both Grid and P2P computing is via identity management techniques culminating in authentication processors. In such a scenario a potential user of a resource has to provide some credentials to gain access to a resource. In the simplest sense these credentials are checked against a list of allowed users of the resource. Both JXTA and common Grid architectures have mechanisms by which this process of authentication occurs, and common ways of expressing identity tokens using standards such as the x509 certificate have been developed [130]. These standards are developing to present federated identity systems such as Shibboleth [131], where credentials can be checked against trusted third parties.

In order to support identity expression in distribution systems, Transport Level Security (TLS) is also essential to the security infrastructure of a distributed system. TLS can be seen as the process of message encryption when sent between two separate and distributed nodes. For example, the data that you input when logging onto an internet banking site is sent to the bank's authentication servers in an encrypted fashion, thus ensuring that if the message is intercepted no vital security information is compromised. Methods of using this vary, for example JXTA uses its own PeerID-based encryption mechanism on the tunnels its sends information down, whilst the Web Service implementations often use WS-Security based encryption. Public Key Infrastructures (PKI) are also commonly used to encrypt any unencrypted keys used to secure messages.

Finally, the other key component of a secure distributed system is the method of authorisation of resource usage. The commonly adopted idea of a distributed system is that access to the system is not 100% secure, as users can loose tokens or have them cloned by potential intruders; also users can become undesirable users on the system when exhibiting bad behaviour, and therefore a method for usage policy and monitoring can be seen as essential to a distributed system. This policy element in a distribution system is often represented in pre-agreed Service Level Agreements with the user that outlines acceptable behaviour and policy manager components in the system. A well developed instance of such a policy enforcement engine can be seen as PERMIS, which allows fine grain access usage and authorized control of resources in distributed systems.

3.8.2 Trust

In order for a distributed security framework to work, the users and service providers have to provide, prove and protect their identities. However, central to this process is the element of trust. Despite the policies and processes that a service provider or a user may have in place to protect themselves, access to a distributed system implies the expectation that both parties will behave in agreed ways. Traditionally, this behaviour has only been subject to SLA backed policy agreements, but increased dynamicity of distributed frameworks often leads to partnerships being built in small timeframes and

using various users. Thus the process of agreement needs to be flexible, and the expression of trust needs to be a central part of this agreement.

For example, when purchasing something online, the guarantee that our credit card is protected provides us with the facility to repair any damage done if a transaction goes wrong. However, often shoppers only select suppliers with good reputations as the chance of trouble is deemed less likely. The same process can be applied to users and providers of Web based services in distributed systems. Projects such as TrustCOM [132] have been looking into the way that trust is moving from the conceptual to the contractual, and can be seen as filling many of the gaps left by SLA as it moves away from the more fixed traditional notions to the more flexible and dynamic, as discussed.

It is difficult concept to achieve a consensual definition of trust. The English word 'trust' is heavily overloaded with usages, most of which are synonyms for 'confidence'. Such synonyms do not express the unique concept that is itself trust. The computing community has used the term in the last few years in a specialised way to describe authorities whose certificates are accepted as authority of identity or access authority. Within social science, trust has been researched for many years in the context of both personal relationships and economic theory.

Trust plays a role in social exchange where it decreases the need for regulation and institutions, and reduces the cost of both transactions within relationships, and the frequency of monitoring the state of a relationship in

order to maintain it. Such a role would be beneficial within VO management and the development of next generation business models.

A common view is that trust is a psychological state, comprising the intention to accept vulnerability (often inherent in VO), and based upon positive expectations of the intentions or behavior of another. Trust exists when one party of a relationship believes the other party has an incentive to act in the interests of the first party, or to give weight to his or her interests when making choices. This is sometimes called the Encapsulated Interest Model of Trust Relations, where the importance of interest in maintaining the relationship into the future is the primary foundation of the trustworthiness of each party in the relationship. A trust relation emerges out of mutual interdependence and the knowledge developed over time of reciprocal trustworthiness. From this view of trust, it is argued that distrust can lead to the development of institutions which limit exploitation and protect those without established reputations or trust relationships [133]. That is, interpersonal trust is not as imperative in corporate business relationships as has been previously argued in the social science literature.

3.9 Summary

This chapter has outlined the current state of the art with respect to emerging standards in these communities, presenting Middleware capable of giving a distributed business infrastructure. The three zone system presented in the previous chapter requires flexible and adaptable Middleware to join and provide services to users and service providers. The technology that has been

discussed in P2P and Web and Web Service communities can create such a Middleware.

Using emerging P2P and Web Service-based technology, it is envisaged that this design will be implemented in both these ways. At the centre of the implementation will be the creation of VOs/environments to support and aid eBusiness integration. The Peer-to-Peer architecture will provide an example of the application of a lightweight framework to enable business to business communication between peers, aided by application specific services which can enhance the communication. The use of the emerging Web and Web Service technology offers the chance to create an infrastructure of services supported by standards that can implement dynamic membership, negotiation and service discovery. This bases service use on current service state, allowing the creation of business models to increase the development and use of BDIFS.

Chapter 4

BDIFS P2P

The original BDIFS prototype implemented using a P2P approach had the primary goal of providing a simple messaging structure for SMEs to use through the P2P design. Thus the BDIFS P2P system deals with the main goal of the data exchange process in a typical integration scenario. The solution can be deployed quickly, and is easy to manage and customize. Within the P2P design the BDIFS system developed the idea of a new community-centred approach for SMEs wishing to embark on business-to-business integration. The BDIFS P2P framework deals with the main goal of the data exchange process in EAI (the seamless secure transfer of data); it can be deployed quickly, and it is easy to manage and develop new functionality around this core. With respect to the aims of this work set out in the Introduction, this implementation addresses the main aims of:

1. Facilitation of a mechanism for message exchange between two separate partner sites.
2. Support for multi-vendor integration of various software platforms.

As discussed, P2P software can offer a flexible and lightweight means by which messages can be sent and users can join the network. The data transfer process is the core functionality in the demo and is aided by the automated configuration of the P2P network. The user is left with the single requirement of developing scripts to integrate the data to and from the back-end system – a process which will be discussed later. The technology behind the data transfer in the network is Project JXTA.

4.1 Project JXTA

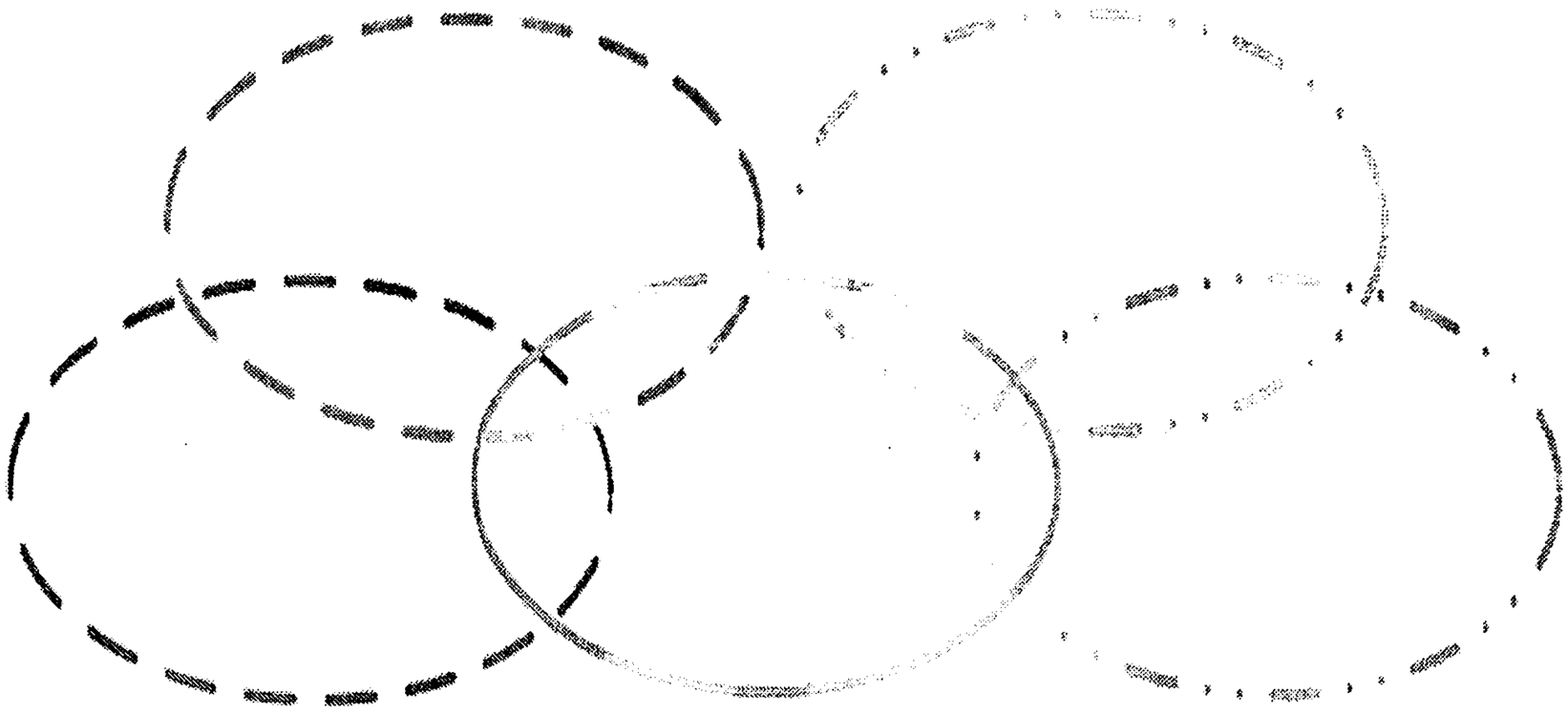
Project JXTA was formed by Sun Microsystems in 2001 to present an open source-based P2P application development platform. P2P computing is commonly associated with resource discovery in file sharing systems such as Napster [134] and Gnutella [135]. However, P2P messaging systems like Jabber [136] are increasingly being used, and illustrate the strengths of, and trust in, P2P computing in the workplace. The ability of Peers to traverse networks, firewalls and NATs in a secure fashion over the Internet and build peer groups is central to the proposed BDIFS framework.

The reason why JXTA was chosen over Gnuttella, Tapestry and other widely available P2P frameworks was because of a number of reasons. The first of these was that JXTA is an open source Java based design of a P2P toolkit.

This makes JXTA unique: it is not tied to any specific commercial vendor and the code is freely available to a community of developers and users. Java is also a programming language I am familiar with and this made the choice of JXTA appealing. Finally in 2001, when this research commenced, JXTA was released as a neutral framework, in the sense it was not tied to a specific application such as file sharing, as the Gnutella framework is. At the time, this was appealing as the software being developed needed to be flexible.

In Project JXTA, the key components in passing messages are the unique Peer ID values. Each JXTA Peer is given a unique ID, as are Peer Groups and Pipes. Peer Groups represent collections of Peers that can communicate with each other; the Pipes are the secure tunnels down which data is sent [137,138]. A combination of these IDs is used within JXTA to secure the network, and the Pipes are used to transmit data from network to network. Like VPNs, this tunnel is encrypted and unlocked by the secure key. Within JXTA, this is the Pipe ID.

These elements, therefore, need to be matched between Peers to enable collaboration and to maintain a secure set of P2P links. The BDIFS architecture is based around this security and has a unique way of generating, sharing and deploying this information.



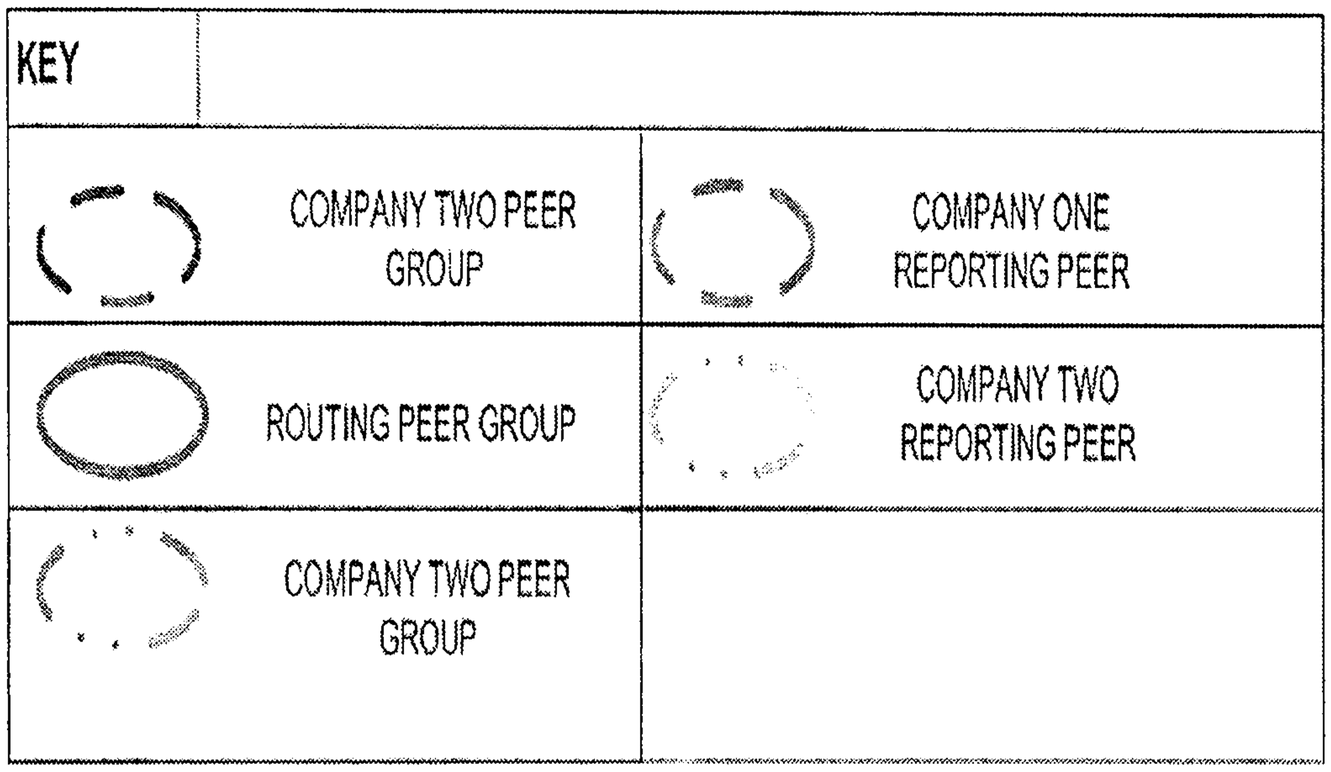


Fig. 5: BDIFS P2P Peer Groups.

As illustrated in Fig. 5, the framework uses Peer Groups grouped by Peer Group ID to determine which Peers can exchange data. In the BDIFS framework, the Routing Peer Group is central and a member of all the company Peer Groups. Each of the Company Peer Groups only knows of the members in its own respective Peer Group. This puts the Routing Peer in a unique position, and it is used as the central group to control the flow of messages, enabling the storage of information whilst the end company set of Peers are unavailable. The Routing Peer also allows the broadcasting of messages across all Peers to aid the design of enhancements and dissemination of network information from a single point.

Data can also be siphoned off from the Routing Peer Group to allow a rich variety of reports on the data exchange. This data is sent to the Reporting Peer Group. The Routing Peer Group is central to the messaging in BDIFS, and therefore BDIFS is not a pure decentralised P2P network as central services are used in the system to manage the network, create Peers and plan the network development.

BDIFS dynamically creates the JXTA Peers from this Web server, based on information provided by the user, with the aid of dynamically created Peer IDs and Peer Group IDs. In turn, this information is stored in the server to allow the creation of new Peers as members of existing Peer Groups from the server. The process is essentially an automated JXTA configuration process and will be discussed further in Section 5.4.

4.2 The Data Transfer Process

4.2.1 Data Transfer Structure

The data transfer process is based around the P2P network and its generation from the central server. The main rules and structure of the network are based around a generic JXTA P2P file sharing network. This is present in existing JXTA projects such as the JXTA content management system [139]. However, the concept of centrally creating Peers and the collaboration around the central server is unique to BDIFS. The use of Reporting Peers to monitor the process remotely is a further enhancement to the basic sharing model.

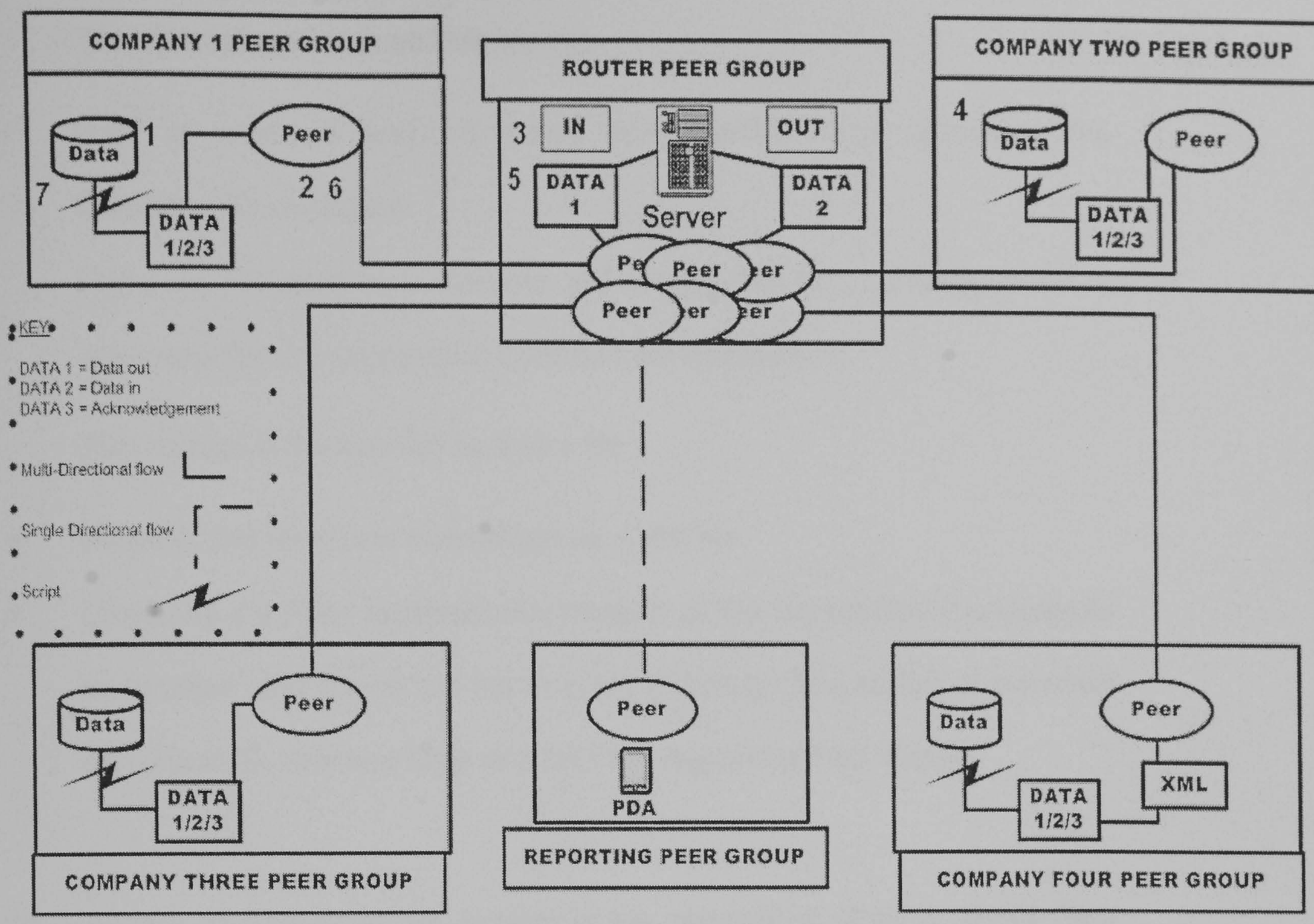


Fig. 6: Data Transfer Process and Structure.

Using Fig. 6 as a reference, the BDIFS framework works as follows:

- Data is created for output from one company to another; for example, a purchase order is raised for ten pairs of pallets of sheep dip from Company 1 on their resource planning system.
- The purchase order is extracted from Company 1's database and translated into ebXML format by a script developed by that company. This file is then sent to the Routing Peer, from Company 1's Peer Group. The data is now in ebXML format.
- The Routing Peer receives the file and checks to see if its destination is available. A message containing a record of the event is forwarded to the Reporting Peer to act as both a log and secure reporting tool.

The data is now in an ebXML format.

- Data is received and translated from ebXML, and uploaded into Company 2's database.

  Company 2's system is updated and a receipt is sent to Company 1 to acknowledge the successful/unsuccessful transaction.

  This receipt is transferred as a text file.

- Stage 3, this receipt is transferred as a text file.
- Company 1's Peer receives confirmation of the successful/unsuccessful transaction and saves the file in a log directory. The status of transfers can either be checked here or within the Reporting Peer Group.

In summary, the basis of the system is the concept of sharing, using Peer Groups as an aid to direct traffic. File sharing in P2P networks is already well established. The main benefit to SMEs of using the system is that with the introduction of an automatically configured set of Peers, a business with minimal IT expertise can effortlessly implement a JXTA secured business-to-business exchange framework.

As the Peer can be downloaded and used on existing IT infrastructure (i.e. a workstation on the LAN), no changes need to be made to the local computing environment. Administration of the system is done from the Routing Peer and the server applies centrally implemented rules. These rules affect the messages passing through the Routing Peer as ebXML data. The target Peer ID is extracted from this data and can be checked against a set of

configurable rules. Only when all rules are satisfied is the message forwarded from the Routing Peer. This Peer also allows the data to be stored if the destination Peer is temporarily unavailable.

Data is translated in the system by configuration scripts, either developed by the company or based on scripts shared on the central Web server of the BDIFS system. This aids collaboration and also encourages the development of new scripts. The two main translations envisaged in the system are translating the purchase order from its native format to ebXML format, and vice versa.

Users join the system and download the software to be part of the BDIFS framework from the portal hosted on the BDIFS central Web server; users will be encouraged to use the Web server to share user translation script contributions.

4.3.1 User Transactions in BDIFS

To participate in the BDIFS framework, an SME will be encouraged to provide a script which can translate a transaction from its server into an ebXML file and repeat the translation in reverse. Script creation is the main investment the company needs to make in order to participate in the BDIFS framework. The sharing of the script, once created within the system, allows modification at less of a cost by other companies running similar systems in order to create their scripts. Furthermore, to help companies wishing to develop the scripts in

the BDIFS system, all scripts are made available to subscribers of the service. This will allow, for example, existing users of the enterprise resource planning software MFG/PRO [140] version 8 to modify an existing script to suit their needs, and thus save on development time and cost. The open source software movement and ERP vendors will also be free to contribute to the database of available scripts. The use of ebXML will simplify the process and hopefully build a knowledge base of converting legacy standards to the ebXML.

New businesses participating in BDIFS will have to follow one of two interaction scenarios in order to download the software to start collaborating within the BDIFS framework. This is because the server dynamically generates Peer ID, Pipe ID and Peer Group ID for new members of the BDIFS framework, with existing Group IDs being used for transactions with existing members. New users only wishing to connect with existing members can do so by specifying the set of IDs required, identifying the business they want to exchange data with. The idea is that they register with the site to enable a central record of activity and business data within the system, and also to give them a central area for collaboration.

However, if the user had some degree of technical knowledge in JXTA, he or she could build a Peer based on this group ID information alone. Although in this model it is assumed that in a great majority of cases it will be easier for the user from SMEs to register and generate the code automatically, further enhancements to the system could also make this type of short circuit

impossible, although it is still a use of SME collaboration for which the system is designed.

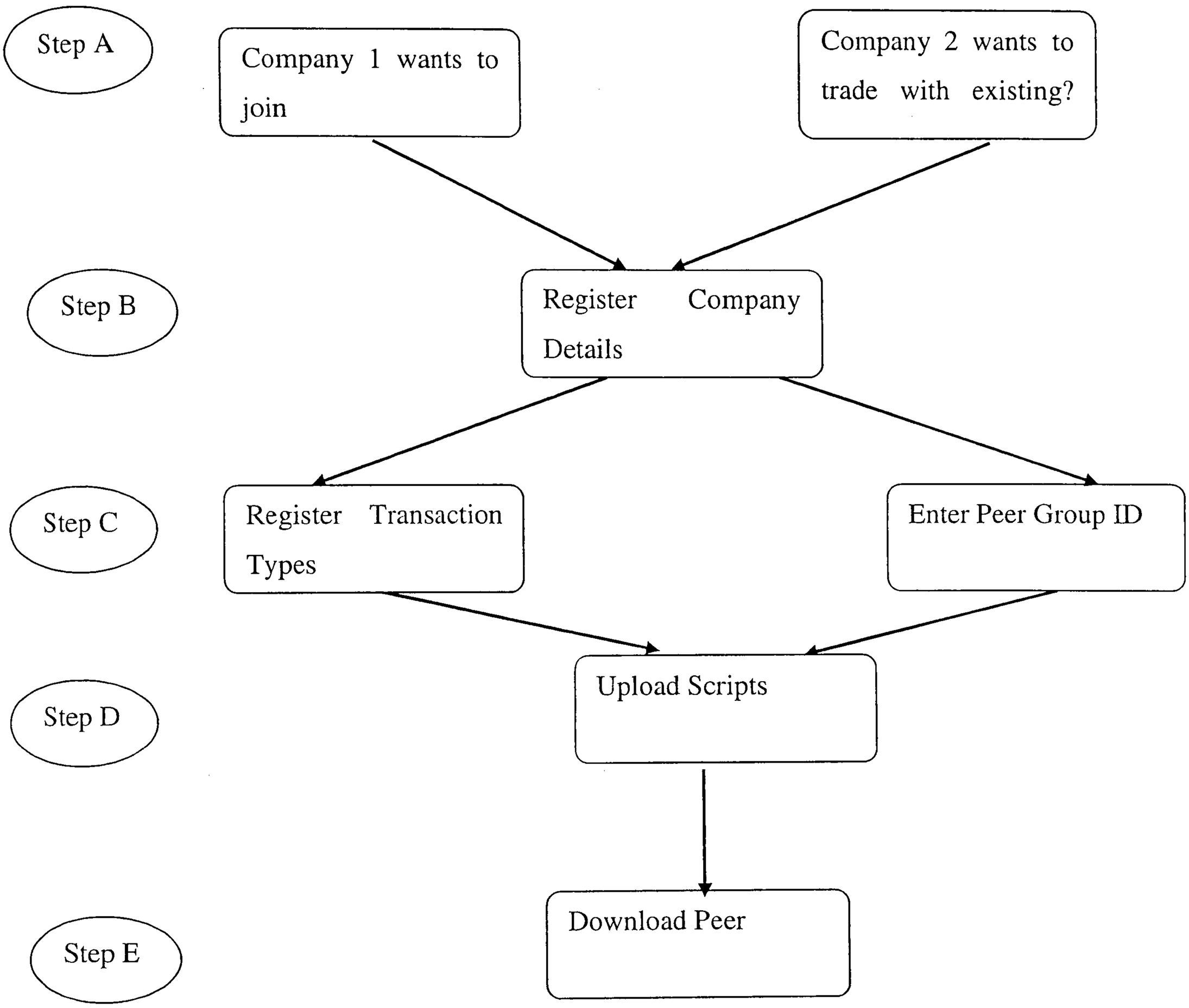


Fig 7: Joining the BDIFS Framework

In Fig. 7, the scenarios discussed above are split into two cases:

Case 1 is for the business wishing to join the BDIFS framework as a member with no immediate links. Case 2 is how a link between two companies can be achieved.

Case 1: Business (Company 1) wanting to join the BDIFS framework:

A. The business moves straight on to point B.

B. In order to participate in the BDIFS framework, the business must provide its details on the Web form. The Web form is the only interaction the user will have in generating the Peer for his or her business. It submits details including name, address, nature of business, back-end system, and other characteristics of the company to the database.

C. Company 1 must register the types of transactions its integration script can handle. These are selected from a dropdown menu and include options like sales orders, purchase orders and so on.

D. The integration scripts used to make these processes happen are uploaded to the database.

E. Based on the information provided above, a Peer is created for the user to download and deploy within their organisation. The Peer Group ID is given for reference, letting the user share this key with other members who want to send messages to their Peer.

The integration script which is specific to their system is made available

____to the BDIFS community for reference and customization.

Case 2: Business (Company 2) wanting to exchange information with Company 1.

A. The Peer Group ID recorded in part E of Case 1 is passed to Company 2 to allow generation of a Peer capable of exchanging information with Company 1.

B. As in Case 1.

C. The Peer Group ID of Company 1 is provided in order for the system to find the transaction details for the company's Peers. This ID will be used to generate a Peer which can establish a pipe via the Routing Peer to Company 1.

D. The integration script which is specific to Company 2 is uploaded to the database for the community to reference. Based on the information provided above, a Peer is created for the user to download and deploy within their organisation. The Peer Group ID for Company 2's Peer and Group are given for reference; the Peer has a route specified to Company 1's network within the Routing Peer. As in Case 1, the ID lets the user Share this key with other members who want to send messages to their Peer.

The integration script which is specific to their system is made available to the BDIFS community for reference and customization.

The two cases above illustrate the collaboration that BDIFS has at its heart. The automated generation of a Peer, which forms the transportation framework for the information exchange, is a major advantage for SMEs. The only effort they must make is the translation and upload of information from

and to their database, and this is aided by the sharing of scripts from other businesses that use this process on the site; with this process SMEs have a standard reference workflow and place for collaboration. The automated generation of Peers and also the linking of trading groups through the Routing Peer are achieved by the interaction with the server.

4.3.2 Using a Reporting Peer.

The use of a Reporting Peer in the BDIFS network uses the same basic principles as the use of the main data exchange Peers in the BDIFS network. This is because the Partner wishing to use a Reporting Peer has to input the specified peerGroupID in order to download an appropriate client.

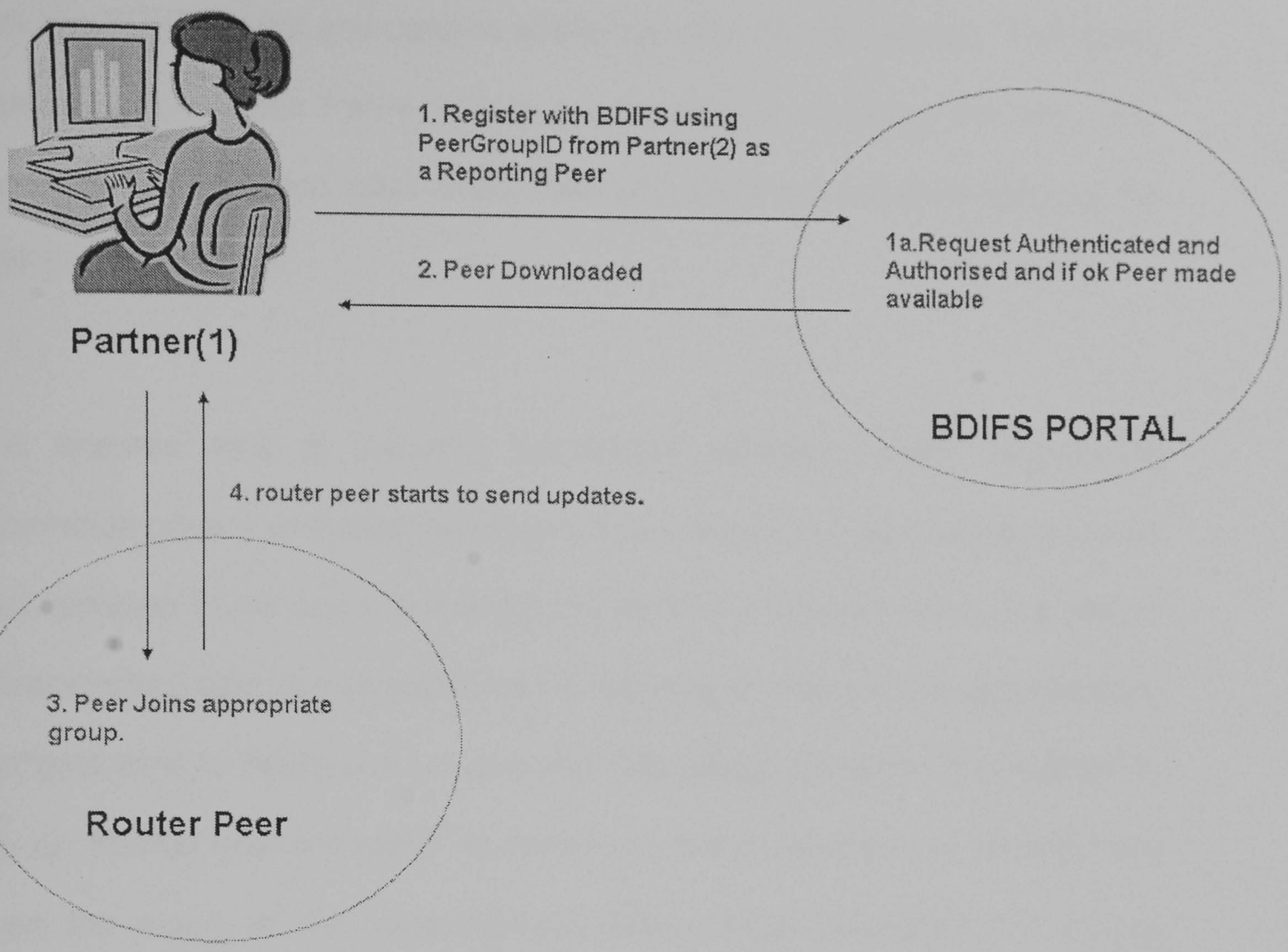


Fig 8: Downloading and Using the Reporting Peer

The client gives the user access to reporting updates from the central Routing Peer; these updates are information specific to the information being exchanged within this group. This is possible as the Router Peer is a member of all Peer groups and traffic has to travel through it in the process of BDIFS message exchange, as outlined in 4.2.1.

Security in the BDIFS P2P Design

Within the BDIFS P2P system, the security mechanism is provided by the existing Project JXTA framework of Peer Group membership. This encrypted key-based information is part of the JXTA Peer that the partner downloads

from the BDIFS portal and uses to exchange data on the network. This gives protection from other Peers joining the network by copying the key. The distribution of keys and rules associated with them are administrated by the BDIFS administrator.

This ensures data is securely transmitted between Peers, maintaining information privacy and also the privacy of the users. The framework relies on trust between Peers and the sharing of PeerID's is central to this. The action of transmitting data to a specific Peer is made by the user in the set up phase and data sent to Peers can be ignored. This design therefore is designed to rely on trusted user behaviour to determine certain degrees of security and given the nature of the transmissions being simple messages this seems appropriate. The security of the central Peers is maintained by the JXTA service providers, who conceivably will be bound by legal terms of agreement to handle data appropriately as in other Web based service communities. Compared to the latter Web Service design of BDFIS, this JXTA design defines a simple level of security for a demonstration that is quite basic. The issue of security will be examined in more depth within BDIFS in Chapter 6.

For the BDIFS P2P framework to be developed further, a more comprehensive identification and security system will be needed. This would include an authentication-based system on the P2P network to validate the user as well as the Peer Group ID, thus removing the large aspect of trust present in the system. Also the downloading of Peers from the portal would

have to be managed as multiple rogue Peers on the network could create denial of service attacks. Currently the deployments are small and the central administrator for the Portal and Router Peers can detect antisocial behaviour on the network and act accordingly.

4.5 Hardware and Software

A Linux (Suse version 9) server was used in the project, running PHP and MySQL over the Apache Web server. The server also ran Java in order to support and generate Peers for the JXTA P2P network.

MySQL and PHP were chosen as they can be used both to generate the user interface and also execute the Java to create new IDs for the system. Furthermore, these technologies are open source, and it is hoped using them will encourage and aid the open source community to get involved with the project. The key technologies on the server can be seen in Fig. 8.

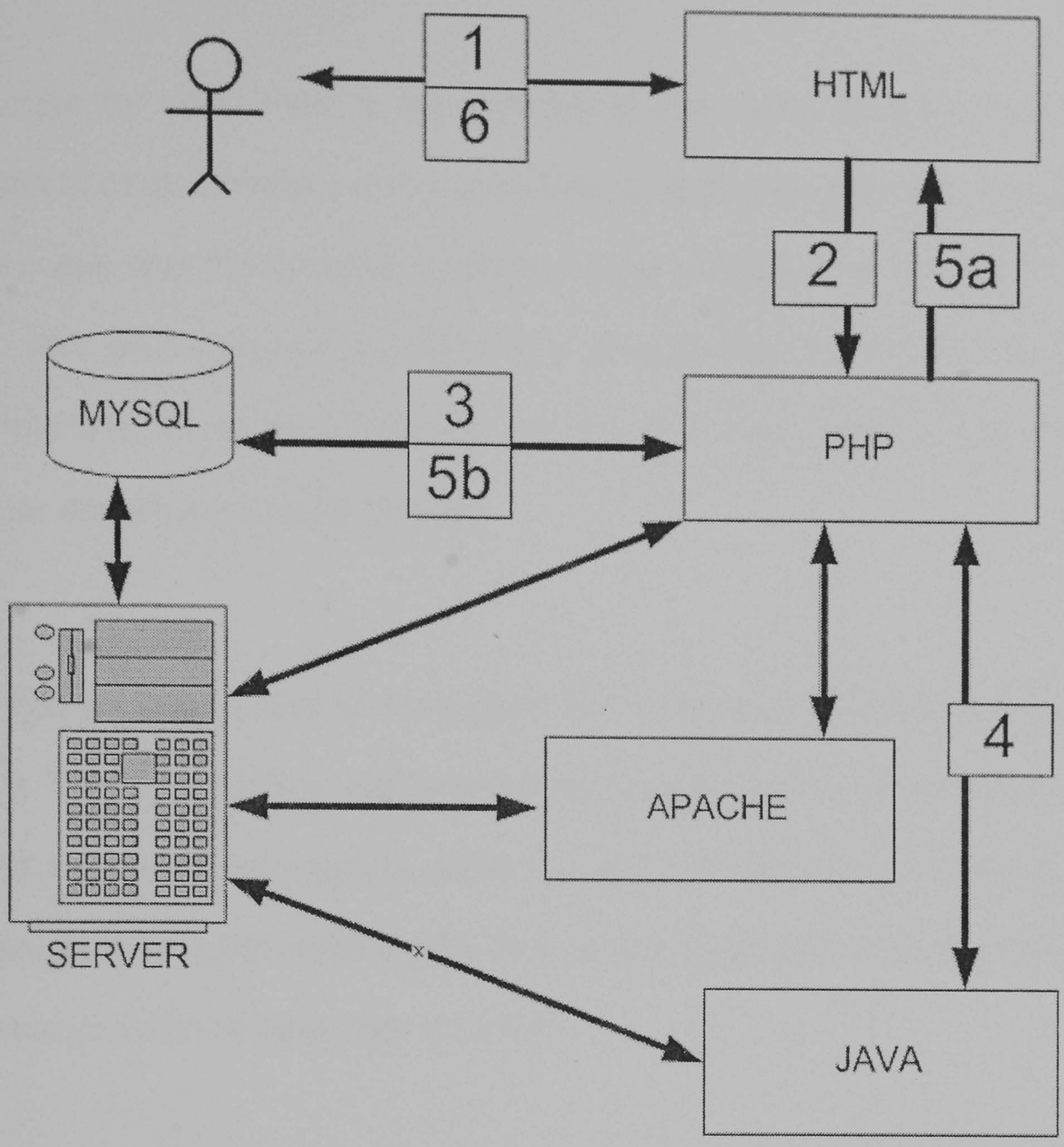


Fig 9: Software and Hardware Architecture.

Processing in the BDIFS framework is as follows:

1. User interacts with Web form to provide details. The form is presented in HTML and generated from PHP running on Apache.
2. The user's details are submitted by form and forwarded to a PHP script.
3. The PHP script writes the details to the MySQL database.
4. The PHP script then executes a Java applet to provide Peer and Peer Group IDs, which are embedded in a Peer for download.
5. Peer is presented to Web for download.
6. User downloads Peer with relevant IDs.

The code for each Peer in the network is the same, as they essentially connect to other members of the same Peer Group and share files. The key to the process was the dynamic creation of a Peer Group and Peer ID for each Peer. This was achieved by executing a Java applet on the server to create the Peer IDs. These were then read into the Java Peer code for output to the user for download from the Website.

The Routing Peer is also on the system but is not illustrated in the workflow in Fig. 6. This Peer stores routing paths established by companies joining the BDIFS framework who specify which Group IDs they want to connect to and, essentially, share information with. This update again is done by the execution and manipulation of Java code via PHP.

The hardware and software in the model consist of open source technology to aid collaboration on the project. This was the main reason for using this technology. As previously mentioned, the collaboration of the community in improving the BDIFS framework is at the centre of the project, making it both unique in the subject area and extending its potential considerably more.

4.6 Evaluation

4.6.1 Performance

Early tests using the P2P architecture have illustrated that it is effective for the transportation of messaging, but the architecture will need to be changed to integrate more functionality. In particular, the P2P framework support for the request of new features requires more functionality to be added to the system to support multiple workflows. In addition, the provision of one-to-many communications has raised issues in respect of thread control. Both these issues can be initially resolved by adding more Peers and rules to the routing on the network. Generally, in the BDIFS application the amount of Peers on the network does not affect performance in tests performed using the lab set-up described in the previous chapter. This test was conducted using simple message exchange between Peers in the set-up described in the previous chapter, whilst increasing the number of Peers, a test that was only deemed to see if the performance was visibly affected in the BDIFS application by the increase of the amount of Peers to 20 in increments of 5.

However, this test was only small scale and contributed little to the actual likely performance of the real world use of the application. More detailed studies have illustrated that the performance of the JXTA network is affected in a negative fashion by the amount of load placed on each peer and the synchronisation of larger networks. Also other factors affect performance of Peers including the version of Java virtual machine [141].

4.6.2 Limitations of JXTA

The JXTA software is also hard to work with, and this is a limitation that needs to be considered when building distributed applications from it. Tests in the project reveal and confirm issues relating to the complexity of JXTA, for example there exists various protocols in JXTA, and the pipes that carry these messages can be defined in various ways. Furthermore, in terms of Peer discovery, JXTA is limited in that it does not support in a reliable fashion the discovery of Peers in different subnets [142].

JXTA also encapsulated IP addresses of Peers and Peer Groups to achieve location transparency, which limits the interaction that is possible with JXTA and other non JXTA based technologies. When compared to the Web Services set of standards and technologies, JXTA is rather specific and less rapid in its development. Only two really stable implementations of it exist in C and Java, and also there is only a limited set of tools to use with JXTA. Experience of developing using JXTA in this project was hindered by this smaller user base and lack of resources to aid development.

4.6.3 Data Integrity

Despite the P2P design having an ability to provide a method for sharing information in a rich and effective way, the BDIFS system needs further

investigation in methods of checking data integrity. This is particularly the case when Peers drift on and offline. Whilst it is conceivable that these issues could be solved by adding increasing complexity to the Router Peer again, the potential would be that the Router Peer could become an isolated vital part of the system. This is going to create another single point of failure issue, thus greater resilience is needed in the routing structure. To solve the problem, the existence of extra and back up Routing Peers is a viable option.

The P2P design's use of the ebXML standard demonstrates that a complex integration process need not always be approached in complex ways, but on the other hand it is a demonstration of the framework's inflexibility: it does not support any other method of data transfer other than ebXML. This has discouraged commercial collaboration in the framework, as the ability of data to appear in native format within the system is still a major desire for users of integration frameworks. This had led to a scrapping of the idea that the system could force standards adoption; instead a more gentle approach is needed.

The P2P framework has demonstrated that lightweight information exchange can be enabled between eBusiness partners using BDIFS JXTA clients – data exchange that can be enhanced by the support of a forum to support the use of the technology and to share integration ideas. In order to move the framework on to support more complex business processes surrounding the data exchange process, there is a need to make a more flexible framework. BDIFS could provide, alongside provision of business-to-business messaging,

dynamic service selection against criteria and service state to aid the integration process. For instance, there is a need for negotiation in the system during the service selection phase, in order to support business models that can give the user a choice of use of BDIFS based on criteria such as price.

To incorporate this more complex set of services and Peers, the network would have to be regulated more and workflows would need to be represented more clearly on the system. To date there are no leading workflow systems for decentralized P2P networks, and the key decentralized concepts in P2P design make it unlikely that reliable and scalable workflows could be implemented. Service discovery in the network, for example, is dominated by the Router Peer which does not hold a static up-to-date set of guaranteed routes to services; also, more information would be needed to be transmitted on the network. This conflict between lightweight design and increased functionality in the system increases as businesses are consulted on the application of BDIFS. In addition, the current service state information is hard to represent in standard ways in a P2P system; where services are found by address criteria only, this makes it hard to run reports on the status of the messaging applications.

4.7 Summary

The BDIFS P2P design is a proof-of-concept to aid integration for SMEs with little IT skills. In the design of this P2P framework the issue of legacy system integration was approached via an innovative community collaboration forum

of discussion and skill sharing. Whilst these exist on the Web with respect to various vendors and industries, to date there are no specific forums and online communities designed to aid integration from a neutral and wide ranging standpoint. The implementation also introduced the concept of a specialised virtual integration environment for the SME.

In addition to this Portal, the development and design of a JXTA based Peer to Peer application for use in business integration partnerships is another innovation that this chapter has presented. This solution has provided a simple design of a P2P framework building on infrastructure provided by JXTA technology. However, further development is needed for the BDIFS P2P system in order to really cater for SMEs needs, particularly around the support of more complex workflows. This can create a conflict between the concept of a P2P framework and the business friendly system BDIFS needs to be.

This conflict in the project was reconsidered early on, and then it was recognised that emerging Web and Web Service standards in Service Oriented architectures could be a better solution to match the SME's needs. Therefore, to increase the depth of the investigation started by the P2P design of BDFIS, work into a Web/Web Service design was started, and this will be discussed in the next chapter.

Chapter 5

BDIFS Web Services

5.1 Introduction

The BDIFS web service architecture has the same goals of standardization, collaboration and simplicity as the P2P architecture. However, the Web Service approach has a greater goal of providing not only messaging but other business functions using BDIFS agents. These agents are designed to invoke remotely hosted core services which are available to all users on the BDIFS network. The core services each provide specific functionality within BDIFS and are a development on the concept of the Peer providing standalone functionality in the P2P design; for example, the creation, storage and checking of identity tokens is provided by separate services in the BDIFS Web Services architecture.

The Web Service design as a whole initially incorporates the aims of the work (outlined in the Introduction) that have been addressed by the P2P design, and in addition this section addresses aims of the work outlined below:

1. Support of Message exchange between two partners with low technical ability.
2. Support of multi vendor data types within the message exchange framework.
3. Support of a business model in the software to encourage both use and collaboration in the framework.

The major strength in Web Services is the ability of the services to maintain state, for example, a stateless Web service when invoked by a user performs the same operation each time. A stateful service can perform the operation based on the number of times the user has invoked it, thus maintaining knowledge of the state of past invocations. This enables the services to provide a larger range of functionality when compared to traditional Web Services and supports business models. It also makes the use of this type of service more flexible as multiple service instances can be created out of one service.

This instance creation allows environments of service instances to be created to carry out specific tasks. This is particularly significant in business applications when compared to static Web Services or P2P approaches where the peers/services are represented once. The significance and application of this functionality is factored into the design and will be explained later.

5.2 BDIFS Web Service Components

5.2.1 VO Management

The key difference between a WS-based business service network using Web Services and an ASP type BSN using Grid Services is in the use of the VO. A Virtual Organisation (VO) can be defined as a group of users and resources collaborating in one or more tasks across organisational boundaries [143]. WS BSNs can be defined as Internet business communities where businesses collaborate through a set of loosely coupled business services [144]. In the VO environment the inter-service communication and understanding is greater to support greater collaboration. The dynamicity of Web Services adds greater and wider application functionality, related to better service resilience and specification within the VO. For BDIFS, this is particularly significant where the network of services can be moulded to suit the requirements by using service resource properties to form dynamic/operative VOs.

BDIFS employs a two-tier VO approach to ensure both secure and effective resource usage. A static Base VO is placed at the core of the implemented BSN. The main function of this element is to record the details of the services registered with the VO and to supply and enforce security policy. The interfaces to the Base VO in this design of BDIFS are shown in the table below.

Interface	Function	Input Value	Return Value
Register Base VO	Registers, Users and Services with the Base VO.	userID, Token.	Integer expressing pass or fail.
Start Base VO	Starts Dynamic VO creation phase prior to application execution.	Token, WorkflowID, SLAID.	EPR of WF.

Table 1: Base VO Interfaces

The Base VO functionality helps illustrate the Base VOs' importance at the top of the hierarchy of the VO services. The Base VO has the two most important interfaces that BDIFS presents: those for registering the user and starting the application. Without these BDIFS would not work. The Base VO manager is the central point of membership and VO management.

When external services are instantiated, Operative VOs from a separate permanent Operative VO Manager (that acts as a factory service) are explicitly formed for this purpose. The main Base and Operative VO Manager have one-to-many relationships with the Operative VO Manager instances within the VO structure. The Services invoked in the Operative VOs are not known to the Operative VO calling them and can reside in any global location that supports the BDIFS service infrastructure. Each Operative VO has its own instance of an Operative VO Manager, and the main Operative VO Manager that has created these in the Base VO is a member of all Operative VOs. Fig. 10 illustrates these relationships.

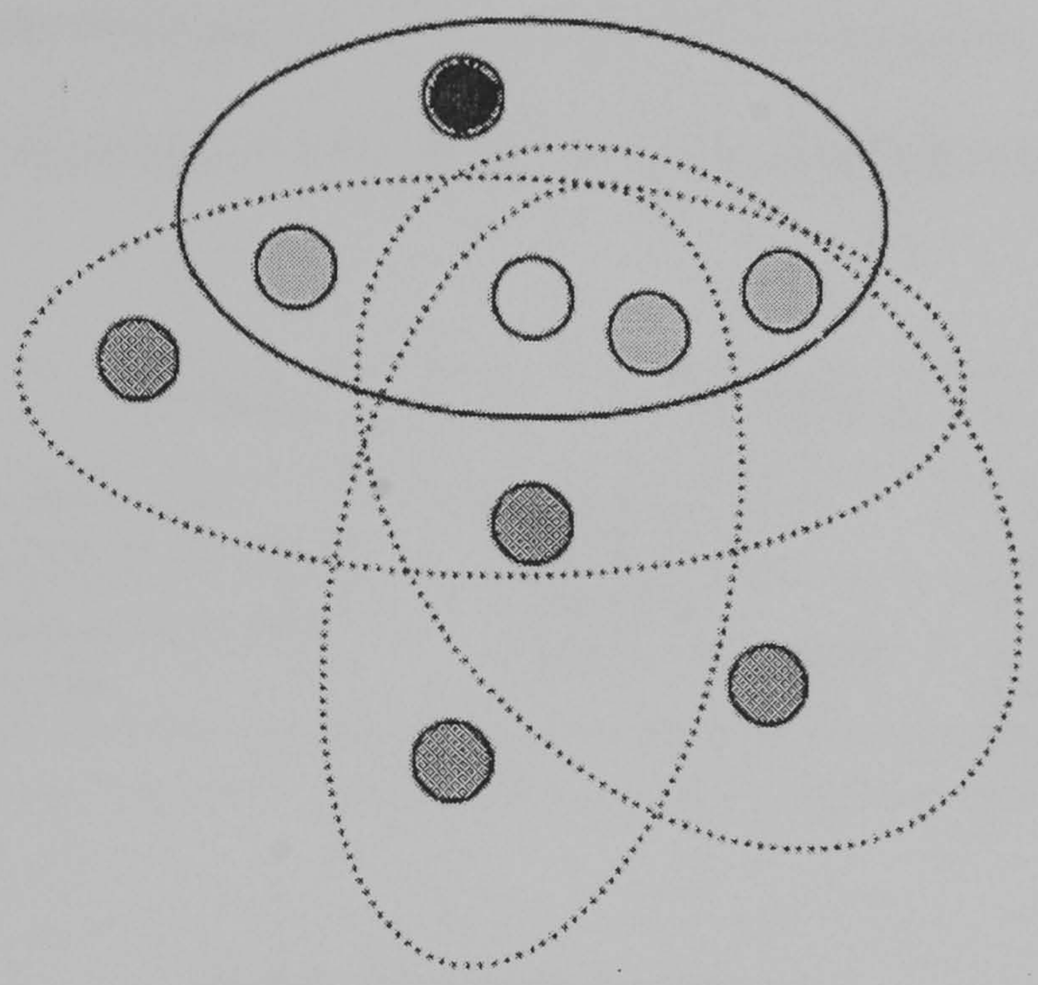


Fig. 10: VO Management in the Base and Operative VOs

The generation of a dynamic VO/Operative VO (OpVO) creates another management domain within the service network which covers the externally invoked services; the service that manages this domain is the Operative VO Manager Service. The Operative VO is specific to the management of the

services for the workflow it has been created for. This allows for workflow security to be based on Operative VO tokening, and for separate workflow threads per Operative VO. This is important when partners may be exchanging confidential business information. The interfaces to the Operative VO Manager in this design of BDIFS are shown in the table below.

Interface	Function	Input Value	Return Value
Create OpVO	Creates a new instance of the OpVO.	Base VOToken, WFID, SLAID	Integer expressing pass or fail.
Destroy OpVO	Destroys the OpVO instance.	Base VO Token, OpVO EPR.	Integer expressing pass or fail.

Table 2: OpVO Interfaces

The functionality in the OpVO Manager is the simple creation and destruction of OpVO instances. The use of the OpVO introduces a degree of flexibility within the architecture for the discovery and invocation of services. As services are invoked from within the Operative VO, extra services can be invoked without needing to send messages outside the VO. Once the workflow is complete, the Operative VO is destroyed. This ensures the Operative VO is managed within a clearly defined starting and ending point. As a result, security is enhanced as all tokens issued for the VO are made invalid when this process terminates. The scheme also presents the possibility for greater analysis of the specific workflow session as a block.

Both VO Managers are developed using Java Web Services WSRF libraries present in the GT4 Grid toolkit. The WS Resource properties are exploited to make the OpVO manager service act as a service factory, and can therefore be invoked to create multiple instances of it. It is these instances which present the Operative VO environment for the application. Furthermore, this dynamicity also allows multiple Operative VOs to be in existence at the same time, but ultimately administered at the one point of contact in the OpVO manager factory, which has the ability to destroy/terminate instances using WS-Destruction.

The approach is well suited for SMEs as it is based on transactions; when a workflow is not being used, there are no resources dedicated to SMEs. As well as being secure and flexible, Operative VOs are therefore economical too and are more suitable for SMEs, compared with the use of business services within static environments. Lastly, the single presence of the Base VO Manager ensures a single point of focus for integration between the client site and the BDIFS interfaces.

5.2.2 Services

Service use can best be visualized in terms of the service's relationship with the Base and the Operative VOs. The Base VO is supported in its key tasks by a set of permanent core services which are not used in business integration. The other services are the dynamic services that are hosted externally to the base VO and are application-specific. The core services differ from the application-specific services as they are permanent in presence and

do not have direct contact with the business data. They are essential in order to enable a centralized control and administration of the network.

The relationship between core and application-specific services is present in a typical workflow. Fig. 11 shows a simple order processing workflow where Partner A wishes to send data to Partner B. The request is authenticated by the Base VO Manager and is then passed on to the application-specific Business Messaging Services. These services reside in the Operative VO and are selected depending on the type of transformation needed on the data. In a typical transaction, this is expected to be the transformation of the source data into a file format that Partner B can accept.

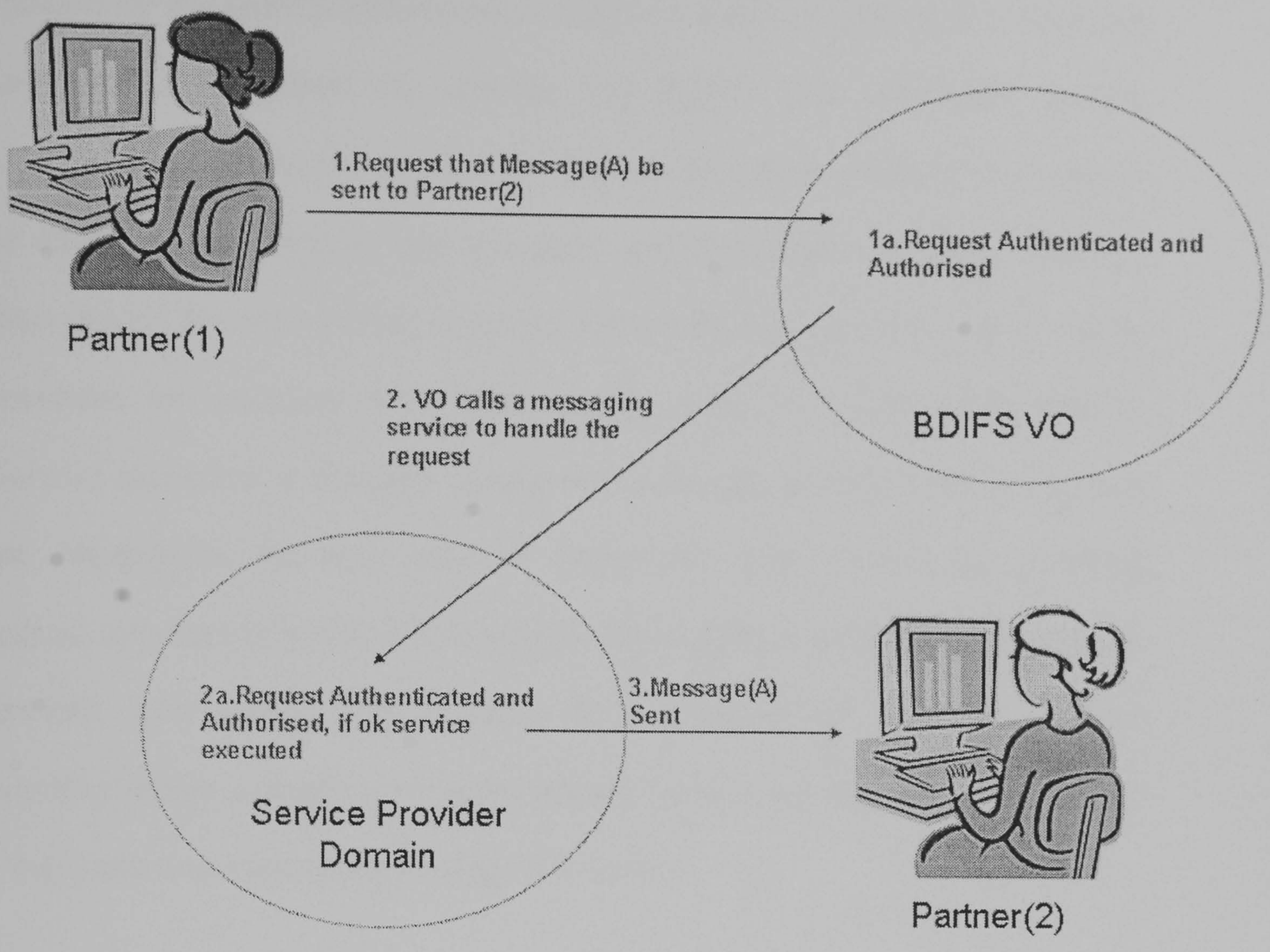


Fig. 11: Simple Order Process

In an extension of Fig. 10, the inclusion of a Workflow Manager and other core services are linked to the process of service discovery and execution. The application-specific services are already pre-registered with the Base VO and are selected against specific criteria in the discovery process. Within BDIFS, the Endpoint References (EPRs) of these services are passed into the Operative VO for instantiation as part of the workflow execution process.

The divide between the two sets of services is necessary at both a conceptual and practical level. This is because the provision of application-specific services in the network will be done by a third-party. The base services will be

provided by the BDIFS administrator. Different levels of control and trust can therefore exist between the services and BDIFS. The application-specific services could be owned by external Service Providers profiting from use in the BSN. These services are managed and provided externally, although when joining the network they have to comply with the core SLA criteria. Other incentives for providing these services can range from community spirit to financial reward, if a charging mechanism is introduced into the VO for their use. Meanwhile, the base services ensure the SLA function by providing support services to monitor SLA specific information from application-specific services, and alerting the workflow when breaches of SLA occur. Such reporting needs a single point of control and ownership, and hence is situated in the Base and belongs to the Base VO owner.

5.2.3 User Registration, Service Discovery, Brokering, and SLA.

Central to service selection in BDIFS is a negotiation process. This process is conducted on SLA protocols that the Service Providers supply about their services when they join them to the BDIFS VO. Users specify certain criteria about their needs when registering, which is matched to workflows that they invoke when contacting BDIFS for an application to be executed. More detail on this process will be supplied, dealing with each process separately.

5.2.3.1 User Registration

When services and users are registered with the BDIFS system, they must become part of the Base VO. This registration process is done via a portal service; the key to the process is the population of the userReg service. This

is essentially a database that is used as a point of reference for user and service properties, and usage history. The interfaces in the table can be seen in table 3 below.

Interface	Function	Input Value	Return Value
Add user	Add a user to the BaseVO	userInfo, BVO Token	Integer as acknowledgement
Remove user	Remove user from the BaseVO	userID, BVO Token	Integer as acknowledgement
UpdateuserInfo	Update user Information, such as user rights and usage history.	userID, BVO Token, userData	Integer as acknowledgement
getuserProfile	Retrieves users profile including access rights etc.	BVO Token, userID	String ProfileInfo

Table 3: User Registration Service (userReg) Interfaces

As the table illustrates, authentication is related to the Base VO token assigned to the Base VO manager. The only BDIFS service that can call the external application-specific service is the Base VO manager. This enables the integrity of the BDIFS service to be maintained and can be used as a point of metering of use, a process central to charging for use of the Base VO and therefore BDIFS system.

5.2.3.2 Brokering

Service Discovery is essential in the population phase of the Operative VO and for the formation of the workflow composition. Within the BDIFS Web Service design, the user invokes the VO Manager for the execution of a specific workflow. The workflow requested has a template which contains the

service requirements for the workflow to be executed. Based on this information, the broker service is invoked for the discovery and selection of application-specific services. The interface to the broker service in this design is shown below in table 4.

Interface	Function	Input Value	Return Value
GetServices	Retrieves WF SLA info from the SLA Manager, and uses this to retrieve services from the Discovery Service	Service Requirement as String, BVO Token	ServiceEPR

Table 4: Broker Service Interfaces

The Broker populates the workflow template with appropriate services by matching them against level of SLA; in future designs this can be exploded and more complexity and factors added. The basic relationship of these services and the divide between core and application-specific services is demonstrated in Fig 12 below. In this diagram the process of populating a workflow with application-specific services is demonstrated. In this design SLA ID is specified by a specific numeric value such as SLA ID 1 or SLA ID 2. At this stage of development the ID is generated when the SLA is added to the SLA repository where it will receive its unique key.

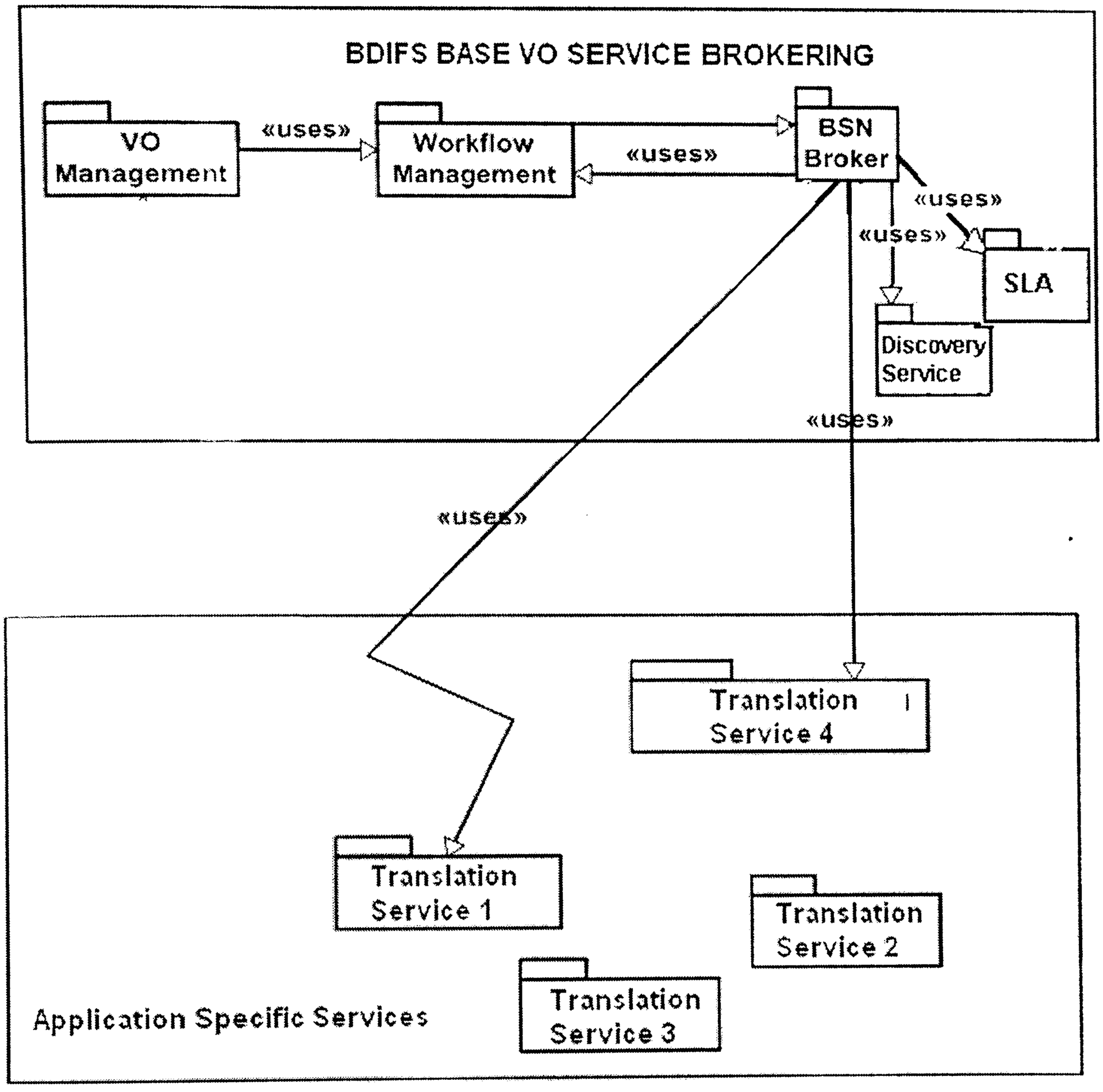


Fig. 12: Service Discovery

5.2.3.3 Discovery

Fig. 12 depicts the BDIFS service discovery process. The Translation Services are specific to the type of translation they perform but the interfaces they present are generic. In Fig. 12 all the Translation Services in the figure could be the same, in which case the SLA they offer could influence selection; the SLA and location of services are listed in the Discovery Service and SLA

Services. The EPRs of the services that are discovered are stored in a registry that is part of the Discovery Service; this information is added during registration in BDIFS by the service provider. The service provider is the human owner of the service. It is envisaged that service providers will register their services for use in BDIFS via the Portal, a process that will be explained in more detail later on. The design of the registry at this stage uses a standard MySQL[145] database and presents a Java-based WSRF Web Service set of interfaces using GT4 libraries and WS Resource properties. The interfaces to the Discovery Service are shown in table 5 below.

Interface	Function	Input Value	Return Value
RetrieveService	Gets service from database	ServiceID, BVO Token.	EPR of Service
AddService	Adds service to database	Service EPR, ServiceID, BVO Token.	Integer expressing pass or fail

Table 5: Discovery Service Interfaces

The two strands in the Discovery Service are relatively straightforward, either the addition or retrieval of a service. The process of service discovery also needs to use the SLA information and to be present in the SLA registry. Once it is present, the Service will have a serviceID assigned to it. This serviceID is used for the population and retrieval of services from the Discovery Service, and ensures that the service provider has signed up for an SLA to be associated with the service, as the information allows the SLA Manager to group sets of services according to an SLA template that they are assigned to.

5.2.3.4 Service Level Agreements (SLA)

The SLA template is associated with the service when the service provider registers the service with the Portal and selects the SLA for the service. The SLA Manager service stores the information linking the services to a predefined SLA template. When a request is made for a service with a specific SLA, the SLA Manager can be used to give a list of suitable IDs. In this design, a service is selected from the discovery node from one of these IDs. In the future these IDs could be used to broker more on the services and add other selection choice to the final service chosen for the workflow. The SLA Manager is implemented in a very similar way to the Discovery Service and uses the same technology. The interfaces to the SLA Manager are shown in table 6.

Interface	Function	Input Value	Return Value
Register SLA.	Allows addition of new SLA templates.	SLA Template, BVO Token .	SLA_ID.
Register Service.	Registers Services associated with SLA templates.	SLA_ID, Service EPR, BVO Token	ServiceID.
Retrieve Service.	Gets Service list according to SLA criteria.	SLA_ID, BVO Token.	List of ServiceID's.

Table 6: SLA Manager Interfaces

SLA is implemented as a proof of concept in the Web Service prototype and its basic functionality exists behind the SLA templates. The user has the

choice of SLA Template A or SLA Template B when registering. SLA Template A is the cheaper, and offers a basic level of service where only one instance of each needed service is assigned to the workflow when the workflow is populated. SLA Template B creates two service instances for each service in the workflow, so that if a service fails then the back-up second service can take its place.

The ability to discover services at the start of each workflow is a key advantage to SMEs, as this process can be elaborated so they can take advantage of the latest state of available services such as current load and cost of use of the service. This information may also be used to define items in virtual marketplaces, where services in addition to SLA provide product details such as stock amount and price .

5.3 Service agents

When services are retrieved from the Discovery Service they are invoked from BDIFS using a Service Agent (SA). These agents are created during workflow population and associated with the service EPRs gathered from the Discovery Service. This approach is useful as it allows binding with the Base VO service and application-specific service on the fly, using instances of the core service existing in the Operative VO. This allows the service EPRs to be invoked without joining the OpVO directly, thus making the system more secure as multiple service providers can be in the same OpVO, providing different function. This enhances privacy in the OpVO and creates a clear secure

boundary between the Operative VO, Base VO and the Service Provider domain, illustrated in Fig 13.

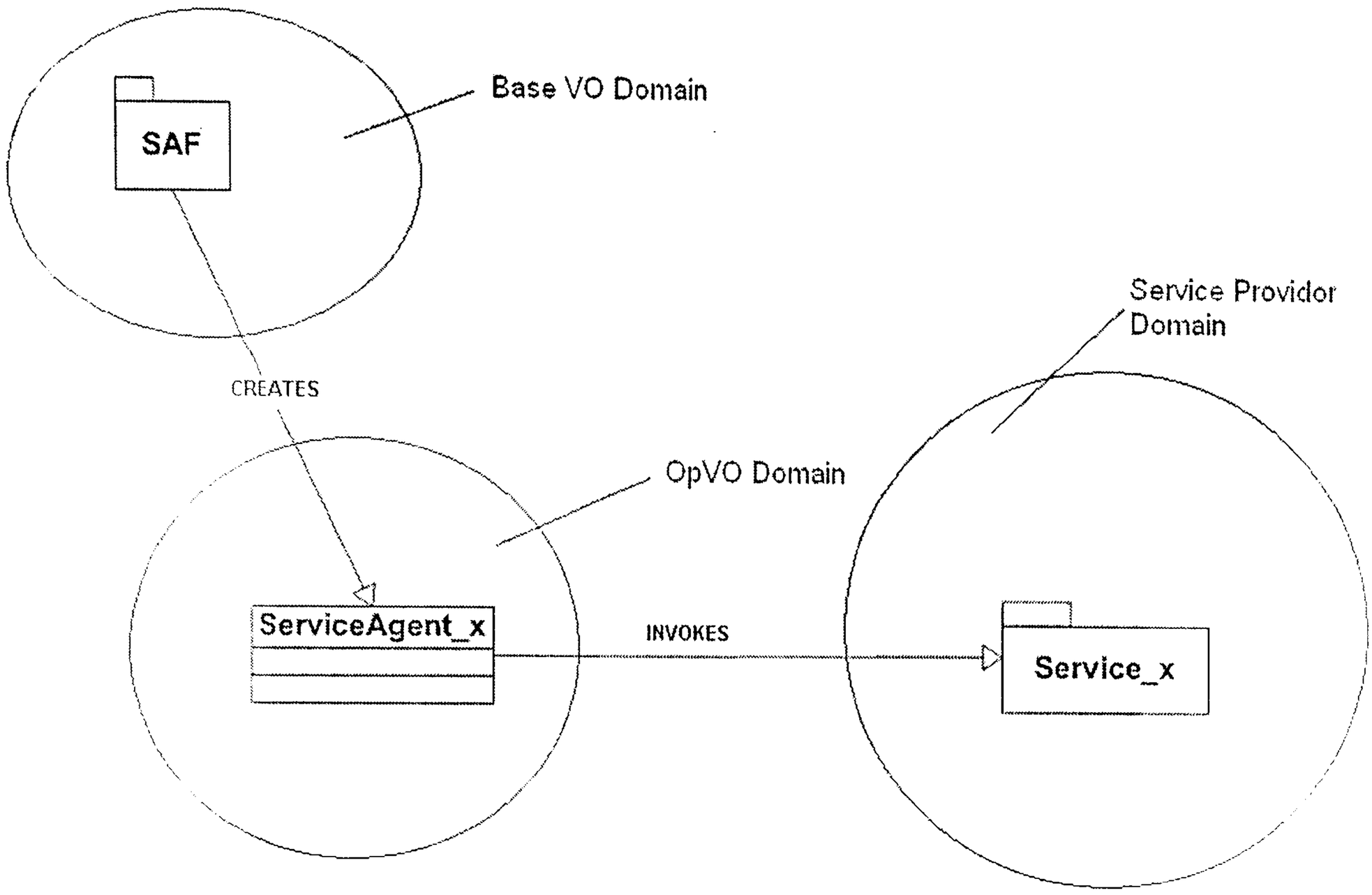


Fig. 13 Application Specific Service Invocation

As illustrated, the Service Agent Factory (SAF) creates an instance of an SA service agent to call Service x in the Service Provider domain. The service in the Service Provider domain will be a Web Service, interoperable with BDIFS, presenting a commonly defined interface in BDIFS for execution. In BDIFS the requirements of the service, its interfaces and so on will be enforced via SLA. Thus, if the invocation fails on an error from the Service Provider domain, regardless of what it is, the SLA penalty for service failure can be enforced. This approach uses three domains which are directly linked to the three zones discussed at the beginning of this work. The interfaces for the Service Agent and Service Agent factory are shown below, in tables 7 and 8.

Interface	Function	Input Value	Return Value
CreateSA	Creates an Agent for a specific Service	EPR of Service, BVO Token	SA EPR
DestroySA	Destroys an Agent for a specific Service	EPR of Service, BVO Token	Destruction Result as String

Table 7: Service Agent Factory Interfaces

Interface	Function	Input Value	Return Value
ExecuteService	Executes Service that the Agent is representing in the OpVO	BVO Token	String ExecutionResult / Service Return
Stop Execution	Stops communication / call to service.	BVO Token	String Result

Table 8: Service Agent Interfaces

As these tables illustrate, within BDIFS both processes can be started and stopped. In the latter case, it is conceivable that when the SA is stopped the process of execution could be started again, but the SAF would have to recreate an SA if it had destroyed the last one.

5.4 Portal, User Registration, Token Base

All participants in both of the BDIFS designs, including Service Providers and users, must register through a central Portal. This lets the VO Manager collect data about participants and their requirements from a central resource – the

portal used to issue the users with agents to access the BDIFS frameworks. The use of the Portal is a direct link to the BDIFS P2P design.

The user agents are different from the service agents as they are not created by a factory. The service agents, as already discussed, are created and call out to the external services in the Service Provider domains. The interfaces that these services present are governed by SLA, whilst the user agents essentially only call the Base VO to request certain operations. The validation is done by a token embedded in the call to the Base VO from the downloaded user agents. This token is checked against a token stored against the userID in the TokenBase service. The TokenBase is simply a service that stores the tokens and users that they are associated with.

The user agents have belonged to neither one of the domains of the BVO or OpVO during operation, but rather to a separate user domain. Fig. 14 shows this.

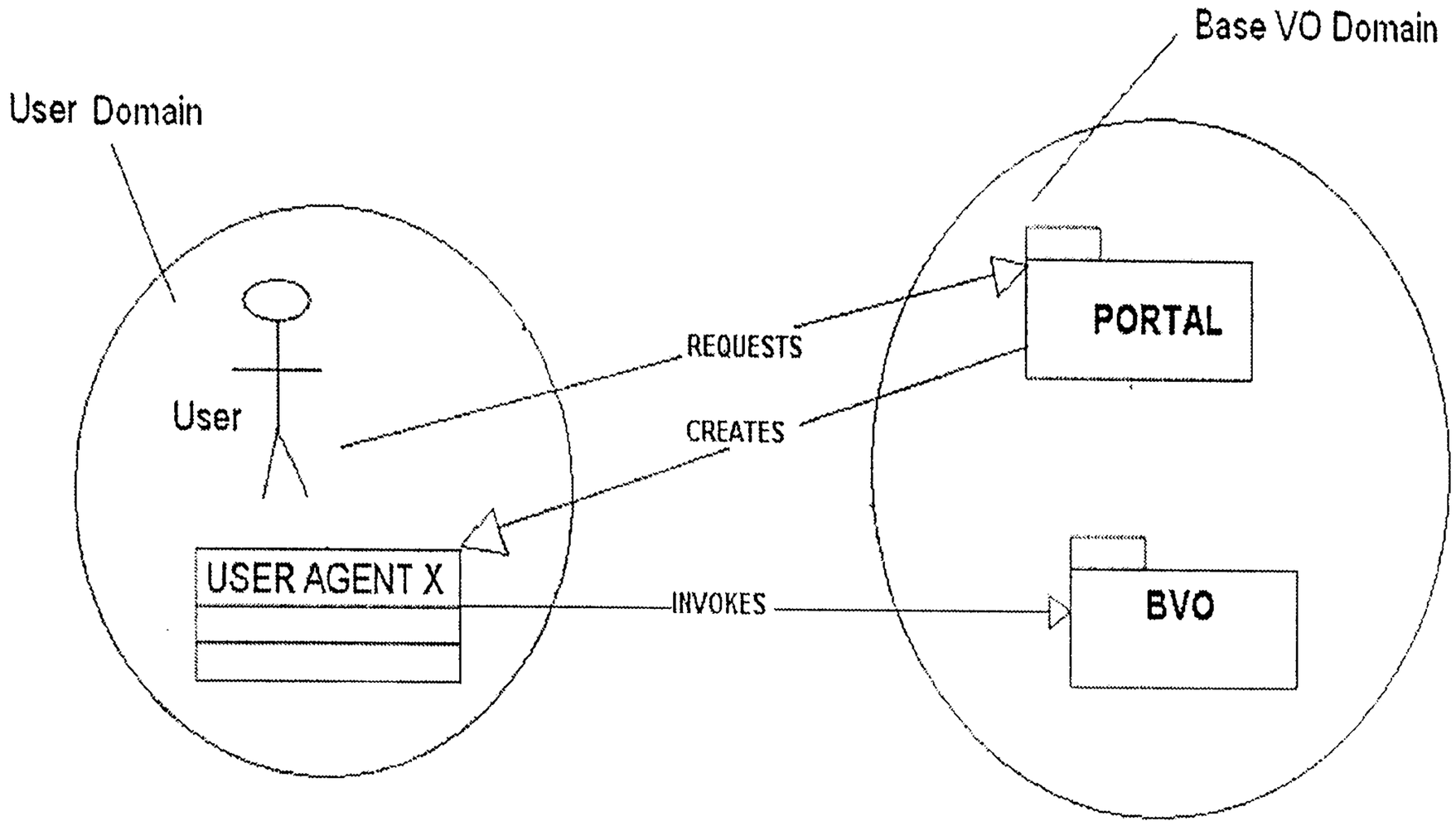


Fig. 14: Initial User Portal – Base VO Interaction

As illustrated, the user requests an agent from the portal in the Base VO via a Web page and downloads the agent if the request is a success. At the same time the Portal service will populate a user registry of the information provided by the user during the registration phase. The agent can then be used to invoke the Base VO Manager to request an application. This initial request for an application to the Base VO from the agent, if successful, will return to the agent a list of workflows that can perform the requested application. These workflows will vary in cost in relation to the level of service they provide. The interface to the Portal service called by a user's Web page is shown in table 9:

Interface	Function	Input Value	Return Value
JoinBDIFS	Registers user with BDIFS, populating SLA and user Registry	Info from Web page.	String to client Download

Table 9: Portal Registration

Aside from obtaining the user agent, the BDIFS portal maintains some of its original focus from the previous P2P design. To help the process of retrieving and formatting the return data from the agents, the Web front-end of the Portal is also designed to be a forum for the exchange of integration information. SMEs can then look on the Portal for information about integrating their agent for simple order processing with their type of database and legacy system. It is envisaged that by encouraging community collaboration, this information will soon accumulate and become a valuable resource for both the community and the Business Information Service (BIS) alike.

5.5 Workflow Management

Workflows in BDIFS are the central strand to the delivery of the BDIFS applications. The Workflow Manager and the Workflow Engine act together for the delivery and execution of the workflows. The Workflow Manager is a Web Service like most of the others in the BDIFS designed using WSRF-compliant Web Services. However, the Workflow Engine is a third-party piece of software under an open source licence. An engine was chosen as this removes the effort required in the workflow design and execution process. As in BDIFS, Web Service workflows are expressed within Business Process

Execution Language (BPEL) templates. In order to provide a rich range of functionality when running the BPEL workflows, an engine with sufficient functionality was selected.

The engine chosen is Active Endpoints BPEL engine [146]. This was based on the fact that the author of the thesis has worked with the product; it is open source, and mature compared to one of the main open source rivals (at the time of writing), Twister, which has recently incubated with Apache and has yet to come forward [147]. The nature of the BDIFS project made the need for an open source engine the more attractive option as this allowed development without having to pay licence fees, and the product has good access to community support forums. Other engines exist and have been evaluated in other projects [148]. With this and the needs of the BDIFS project in mind, the widely used Active Endpoints engine has the necessary functionality.

In order to generate the BPEL template with the correct information during the service discovery process, these templates are populated with EPRs of services. The Active BPEL engine has the ability to update the workflow as it is in execution phase with new services; for example, this can be done in the case of a service exception. This updating, along with the deployment of the workflow and any termination before workflow completion, is done by the Workflow Manager. The key interfaces that are used in the Workflow Engine are shown in table 10.

Interface	Function	Input Value	Return Value
deployBPR	Deploys Business Process/ Workflow	BPR file.	String address of deployed workflow.
terminateProcess	Terminates running Process / Workflow	ProcessID.	String acknowledgement
startProcess	Starts the Business Process / Workflow	ProcessID.	
setVariable	Updates the variables set in the Process/Workflow	ProcessID, variable name, new data.	String acknowledgement

Table 10: Workflow Engine Interfaces

The workflow is deployed by passing a Business Process Resource (BPR) file to the engine when the deploy BPR method is invoked. The BPR file is made from deployment Web Service Description Language (WSDL) for the services, a process deployment descriptor that is populated with the EPRs of the services and workflow of the BPEL. The WSDLCatalog contains the references of other WSDL needed for the deployment and execution of the workflow. The key elements in the BPR file are shown below in Fig. 15.

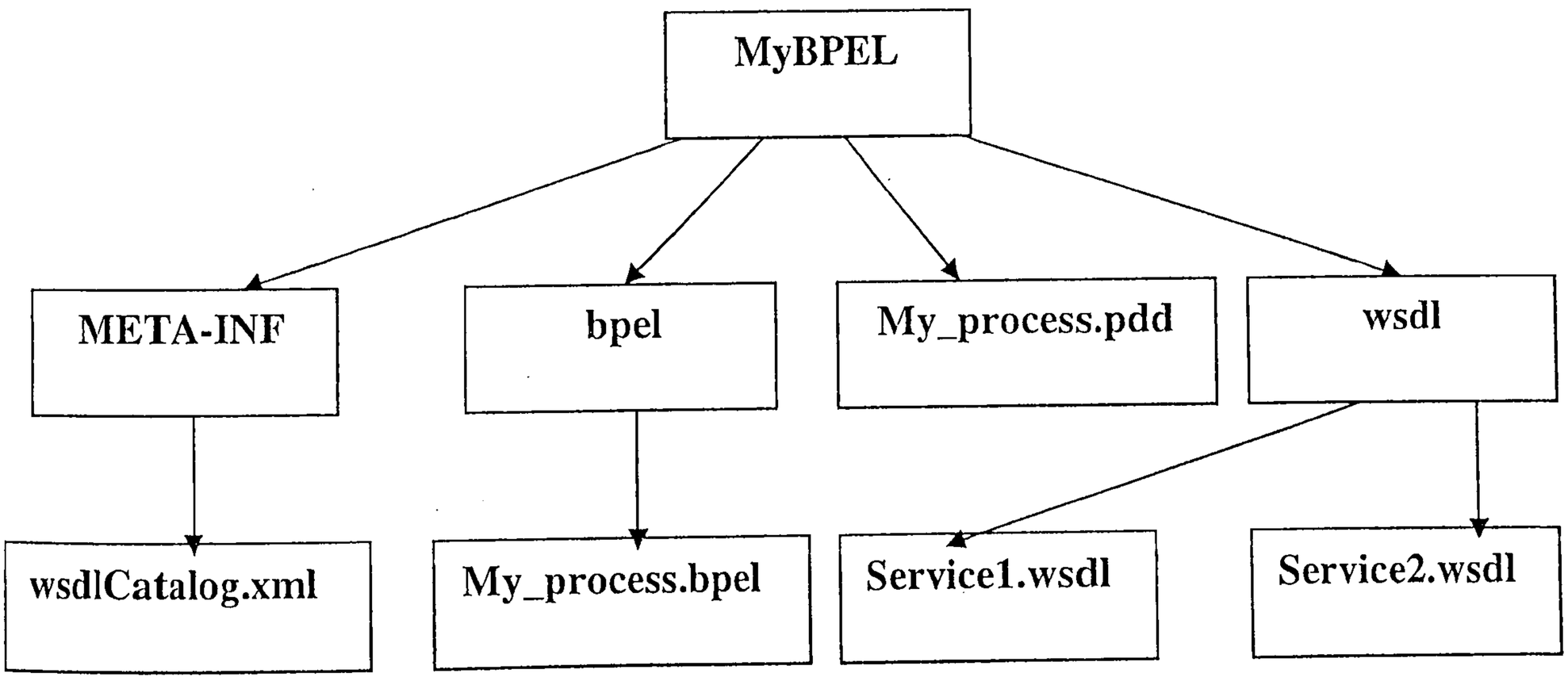


Fig. 15: BPR File Structure

Once the BPR file is successfully deployed, the workflow is ready to be used via calls to the interfaces shown in table 10. These methods are invoked, and the engine is effectively controlled by a BDIFS developed service, the Workflow Manager. The Workflow Engine does not authenticate requests and its address is only made available to the Workflow Manager which provides it with authentication and authorisation of requests. The Workflow Manager also stores the Workflow Templates that are retrieved and populated with service EPRs prior to workflow deployment. The interfaces to the Workflow Manager are shown below in table 11.

Interface	Function	Input Value	Return Value
DeployWF	Registers user with BDIFS, populating SLA and user Registry	BPR file, BVO token	String address of deployed workflow.
UpdateProcess	Terminates running process	ProcessID, BVO token	String acknowledgement
Terminate Process	Updates variables in deployed process	ProcessID, variable name, new data, BVO token	String acknowledgement
StartProcess	Start deployed workflow	BVO Token	String acknowledgment
getWFTemplate	Retrieves a workflow template from the Workflow Manager	BVOToken, WorkflowID	Workflow Template as String

Table 11: Workflow Manager Interfaces

The *deploy, update* and *terminate* methods are all directly linked to the methods in the enactment engine. The Workflow Manager acts as a BDIFS buffer around the WFE before these calls get invoked. This allows the calls to be monitored, and in some cases authorized, before they reach the workflow. These functions are linked to the security and usage policies in the VO. These policies are enforced by individual service rules. Currently in BDIFS, there is

no VO-wide policy rules produced from and enforced by a central Policy Manager service. In larger Grid systems such policy engines sit like the Workflow Manager in BDIFS as a central source for the processing of policy on a service-to-service basis.

5.6 Security

There are two main ways of securing a distributed computing system: preventing access or allowing controlled secure access [149]. The perimeter version of security is not so practical, especially in systems where the users and services are constantly changing. In this controlled, secure access type of scenario, security is often limited to message security via methods like encryption, usage restriction and monitoring. BDIFS is such a system, and the way that this security is ensured will now be discussed.

5.6.1 Identity

The main way to secure and control user access is via the creation of user groups and roles with specific privileges, and building up trust relationships with users from other domains. At the point of entry users can be authenticated and then their behaviour can be monitored and controlled when they are active within the system.

The mechanism essentially relies on the use of identity tokens that are not only essential for authentication and authorisation but are also vital, in order to implement pricing and charging schemes within BDIFS. Tokens allow users to

be identified and associated with various other metadata relevant to the VO. User Identity can be defined as: "The collective aspect of the set of characteristics by which a thing is definitively recognizable or known." [150]. Within BDIFS, identity management is the core mechanism by which the VO is secured.

There exist technologies to aid in the creation of identity tokens, and the Security Assertion Mark-up Language (SAML) [151] is the main standard in the Web Service community that can provide a format for the transmission of user data. Both the Liberty Alliance [152] and Shibboleth [153]use SAML. SAML makes use of widely known technologies such as XML and SOAP to build a basis upon which user-related information can be suitably exchanged. Different kinds of assertions can be distinguished within SAML. These include:

- Authentication assertions: declarations about a user's identity.
- Attribute assertions: contain particular details about a user.
- Authorisation decision assertions: dictate what a user is authorized to do at a particular site.

With respect to future development of identity and policy in BDIFS, other policy systems also exist in distributed systems provided by projects such as PERMIS [154]. This product provides a framework for creating and managing a distributed authorisation infrastructure using X.509 [155] standard attribute certificates (ACs) to hold users' roles.

With respect to standards, WS-Federation exists as an alternative to using SAML, and is supported in the authorisation system PERMIS. The standard is still relevantly new and has three functional parts, including the Web Services Federation Language, which defines how different security realms broker identities, user attributes and authentication between Web Services.

The Web Services Federation specification is another component of the Web Services Security Model. The Web Services Security Model outlined in the Web Services Security Roadmap [156] published by IBM, includes other WS standards to enhance SOAP messaging to enable secure Web Service communication. WS-Federation is part of this, and is significant as it defines trust mechanisms based on federated identity management.

Thus, two emerging approaches from the Web Service and Grid world exist to identity management. However, unlike the past where standards have developed separately, it is fair to say that the standards are likely to converge in the future, and applications are likely to support each other. A good example of this is in the plans for the development of the Grid Security Infrastructure [157] which supports Grid and WS security innovations.

5.6.2 Authorisation and Authentication

As touched upon in the scenario, where the user agent requests an application from the Base VO Manager, BDIFS uses an authentication and authorisation procedure to establish identity during the access from the user domain to the Base VO domain. This is then replicated within service-to-

service access in the VOs and access to services from the OpVO domain to the Service Provider domain. The flow of policy in relation to the service and domain relationships can be seen in Fig. 16.

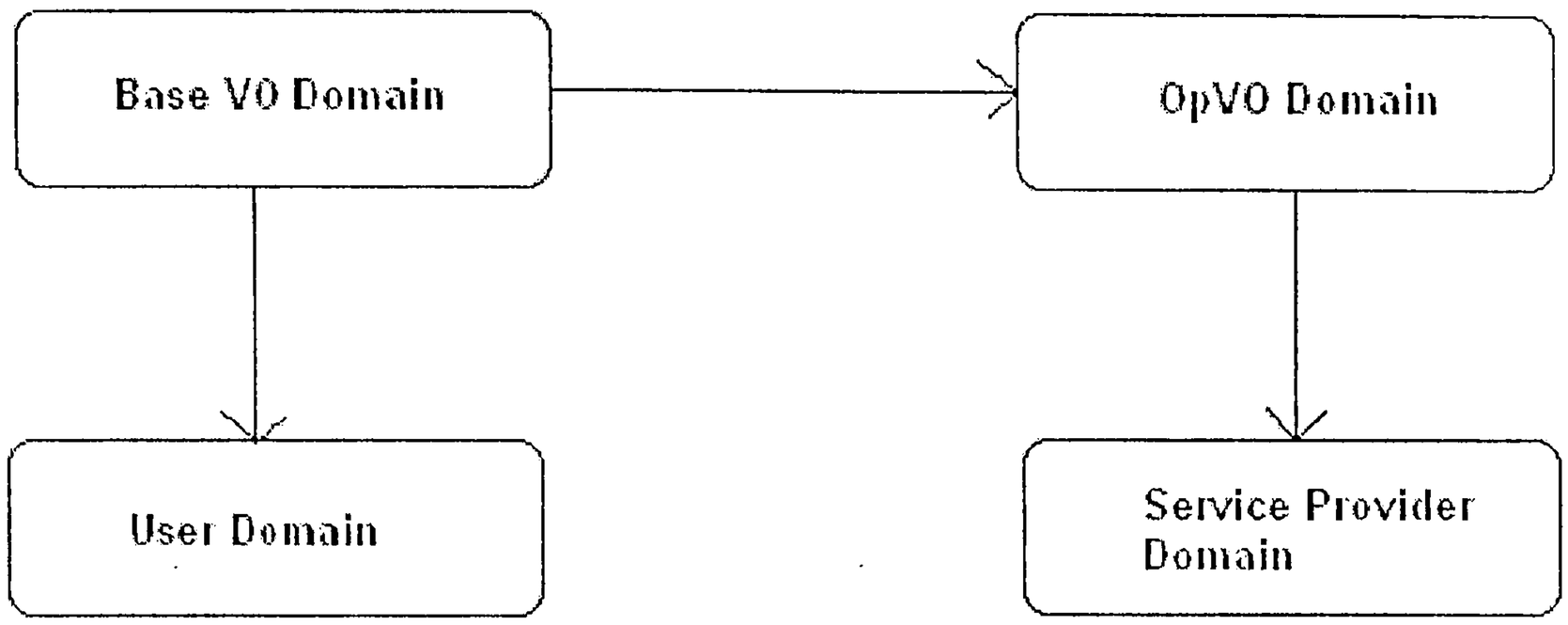


Fig 16: BDIFS Domains and Policy Flow

The process of joining a VO can be achieved through the authentication of certificates; for example, Globus uses x509. Applications exist to provide this functionality in packages that also assign roles and manage other tasks. The VO Membership Service (VOMS) [158] is an example of such a package and its architecture uses the authentication and delegation mechanisms provided by the Globus Toolkit Grid Security Infrastructure (GSI).

The authentication and authorisation is implemented within BDIFS at the areas illustrated in Fig. 17. Each exchange uses a specific token associated with the target domain where it is checked. The process can be made more secure by adding encryption using the public and private keys.

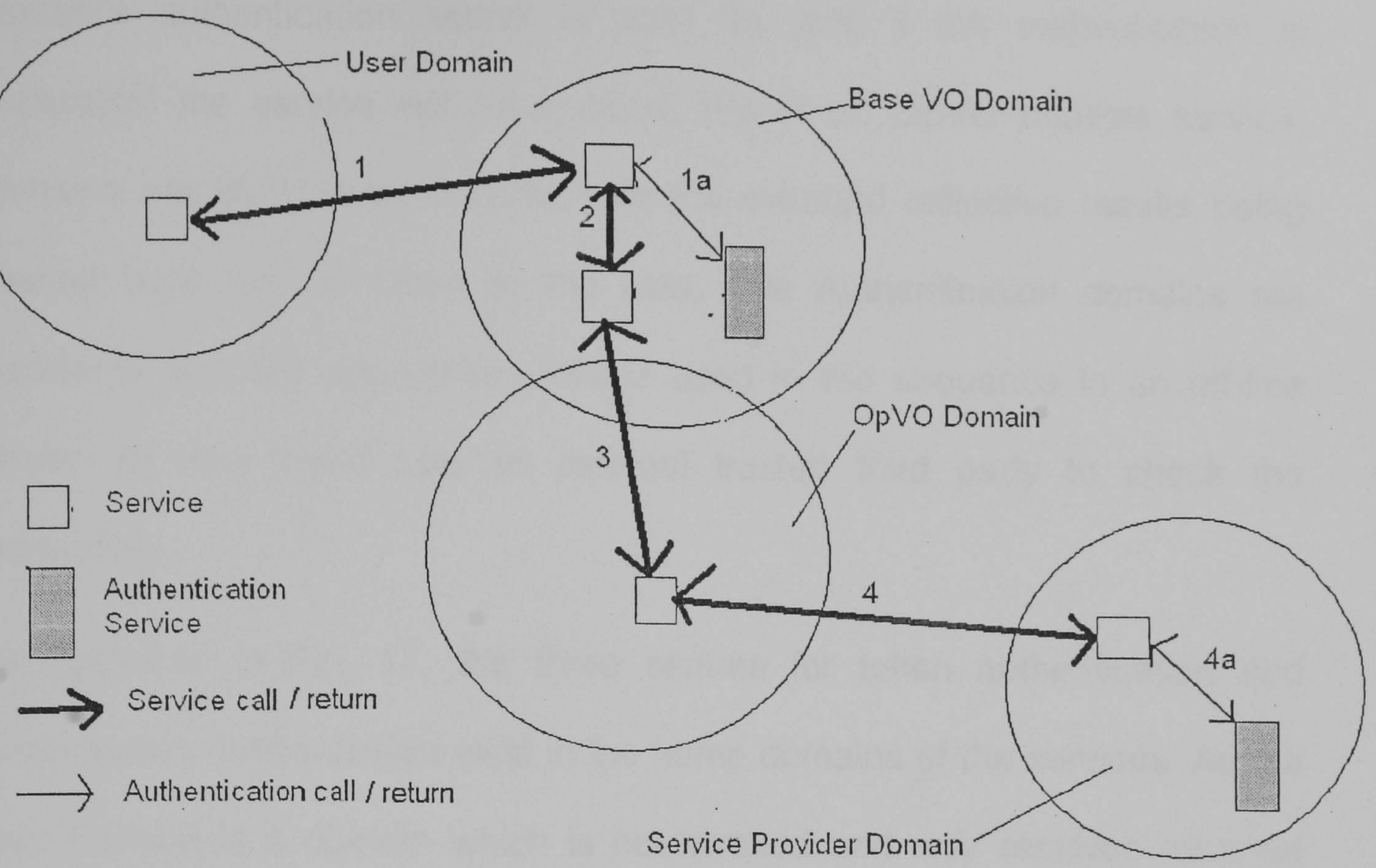


Fig 17: Inter Domain Security

Within Fig 17, the call marked 1 is the initial contact from the user (passing his or her ID Token) to the Base VO Manager to request an application from the VO. In 1a of this call the token passed is authenticated against the authentication service and also the authentication service. Call 2 is made if the authentication is accepted and the OpVO Manager is invoked. The result of the invocation of the OpVO Manager is to create another OpVO Manager instance that is specific to a OpVO domain (as illustrated in point 3). Within this OpVO, other instances of services are added from the BaseVO domain and are used to invoke the services in the Service Provider domain.

The invocation to the Service Provider domain is point 4 in Fig. 17 and it uses the identity token of the BaseVO. This is checked against the Service Provider

domain's authentication server in point 4a, and if the authentication is successful the service will be invoked. Within an OpVO multiple service, domains are likely to be invoked with the eventual collective results being passed back up the chain to the user. The Authentication domains are populated with the appropriate tokens used in the sequence in an off-line phase, or they could use an external trusted third party to check the credentials.

As illustrated in Fig. 17, the three centres for token authentication and authorisation in this design exist in the home domains of the services. As the user Domain is a domain which is not invoked and only receives returned values, no point of authentication is needed in the current design. The OpVO domain is created by the Base VO and uses the Base VO authorisation service to check tokens. Within the Base VO, this function is performed by the TokenStore service, which also stores and issues tokens to users.

Interface	Function	Input Value	Return Value
CreateToken	Creates a user token Association.	userID, BVO Token	String token.
CheckToken	Checks token	BVOToken, userID, Token	String result
RevokeToken	Revokes token associated with user / Service	BVOToken, userID, Token	String acknowledgement
CreateGroup	Creates user Group	BVO Token, GroupDetail	String Acknowledgment
AddServiceToken	Adds a token from a Service Provider domain to be associated with an application-specific service.	BVO Token, ServiceID, ServiceToken	String Acknowledgment

Table 12: TokenStore

The TokenStore in the Base VO, as table 12 demonstrates, associates tokens with users and groups. This is a basic approach to the development of such a service, and applications like VOMS have a far better range of functionality. The approach was chosen because at this stage of the application development it suits the need for BDFIS to focus on its core functionality areas. The tokens in the current BDIFS design are expressed as regular strings. Use of a common standard such as SAML is planned at the next update to the framework to express identification. The information in a Token and its structure is shown in Fig 18 below.

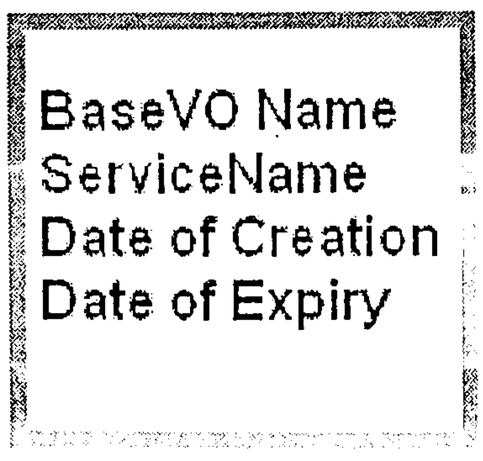


Fig 18: Token Structure

This structure can be made much more secure by using public key infrastructure (PKI) and encryption techniques. In order to do this, the keys and signatures would need to be distributed amongst the services that need them. In the current BDIFS design this level of security is not implemented, but in order for the project to increase its scalability the use of PKI would be essential. The authentication process could also be increased in terms of speed by the introduction of single sign on capabilities and token/ID caches.

5.7 Deployment Architecture

The main hardware in the BDIFS testbed is shown below in Fig 19.

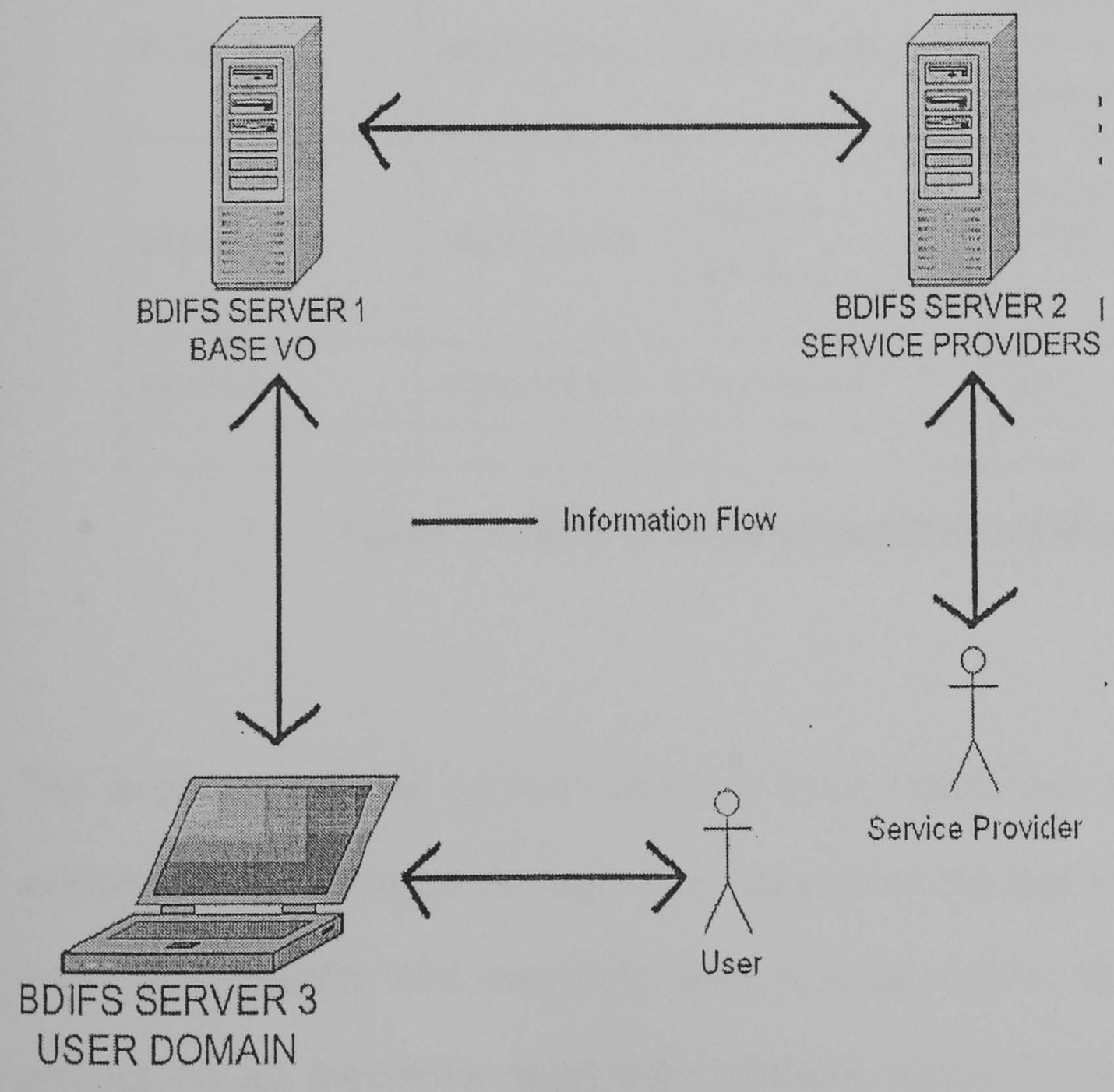


Fig 19: BDIFS Testbed Architecture

Table 13 illustrates the services, hardware and software deployed on each server.

Server Name	Operating System	Services	Software	Address/Port
BDIFS1	Ubuntu Linux	Workflow Manager. Service Discovery OpVO Manager TokenBase Broker Service Agent Factory SLA Manager Portal Service BaseVO	Globus WS-Core, Java, Apache Axis.	192.168.0.1:8081 192.168.0.1:8084 192.168.0.1:8082 192.168.0.1:8086 192.168.0.1:8087 192.168.0.1:8088 192.168.0.1:8089 192.168.0.1:8085 192.168.0.1:8080
BDIFS1	Ubuntu Linux	Workflow Engine	Apache Tomcat, Active Endpoints BPEL Engine	192.168.0.1:8090
BDIFS2	Ubuntu Linux	Servicex Servicey	Globus WS-Core, Java, Apache Axis.	192.168.0.2:8080 192.168.0.2:8082
BDIFS3	Windows XP	User agent	Java	192.168.0.3

Table 13: BDIFS Deployment Description

The deployment was carried out using three machines, due to the limited availability of resources. All machines, apart from the laptop that was BDIFS3 (running Microsoft Windows XP), were running Linux. Web Services were present on all machines apart from BDIFS3, which hosted the Java client. ServiceX is the translation service used in the demo applications. The service

performs the translation of data from a specific text format in this case, to a simple XML format. The input and output documents can be seen below and are very simple. The ServiceY is a duplication of ServiceX used in the service failure tests outlined later on in the thesis.

date: 22/10/06

item: 5 cars

cost: £45500

delivery address: 7 new rd, new town, UK

orderID: 000449393

Original Format

```
<GarageOrder>
<OrderID>000449393<OrderID>
<Product>5 cars</Product>
<Price>45500</Price>
<Currency>GBP</Currency>
<ShipTo>7 new rd, new town, UK</ShipTo>
<RecieverID>Automated</RecieverID>
</GarageOrder>
```

Transformed Format

Fig 20: Example Input and Output Data

MySQL databases were used by the Portal userReg and TokenBase services to contain the information they used and created. The Tokens generated by the tokenbase in this design were basic strings; in the future the TokenBase could be replaced by a SAML authority.

5.8 Summary

The Web Service design presented in this work uses state of the art Web and Web Service technology to create an application that can support the eBusiness integration needs of SMEs. Message exchange of multi vendor

formats supported by a business model is illustrated in the Web Service design of BDIFS. The services in the framework enable the exploitation of service states, allowing the dynamic population of workflows with services. This is enabled by the use of dynamic VOs, which is a key contribution to this thesis within this particular application area. Also, the user driven workflow creation via user preferences set in the Portal is unique in this type of computing environment. The Web Service design proves that the increased dynamicity provided by Web Service standards, and in particular the WSRF set of specifications, can provide a new generation of SOA that can be drilled down to suit specific needs of SMEs or individual users.

In BDIFS this VO approach is enhanced by the supporting service infrastructure. The brokering and services associated with security and SLA based resilience in the system ensure that secure business models and processes can be supported in the framework. This innovation is not specific to the BDFIS project, but the application of a Web Service based market place presented in the BDIFS model is the first of its type to encourage commercial service development to aid business to business integration. This business model is uniquely strengthened, when coupled with the community support of users of the system around the Portal.

Chapter 6

Use cases and testing

As described, the main aim of BDIFS is to present SMEs with an infrastructure to aid business–to-business integration in an eBusiness environment. In this chapter the fruits of the discussion, development and design of the BDIFS Web Service architecture are presented in a series of use cases. To do this in an orderly way, BDIFS functionality is broken up into stages. These stages are illustrated in the four main use cases outlined below;

- User and Service Registration – here the use cases surrounding user and service registration in order to gain BDIFS membership will be described.

- OpVO Set Up – this phase can simply be seen as the workflow population phase. This is the stage when the VO environment is set up for the BDIFS application and appropriate services for the workflow are chosen.
- Workflow Execution – this is the stage when the workflow process starts and begins to execute the application-specific services populating the OpVO. This phase culminates either in the termination of the workflow by natural means (i.e. completion) or by administrative control.
- OpVO destruction – once the workflow is complete and the user unsubscribed from the service, the OpVO destruction phase is processed to destroy all used service instances within the OpVO. This process should also include the delivery of appropriate billing to users of the system.

6.1 Company A: Use Case 1

The workflow demonstrated is specific to a user requesting a translation to be performed on some data received, or to be sent to an integration partner. The translation is outlined in the previous section and is conducted by serviceX. This is linked to the transmission of data already presented in the previous section on P2P business messaging, which is one way the data could be exchanged. The other means by which the data can be transported is by the selection of a service in this workflow to integrate with a target partner service. As the workflow demonstrated only takes this to the service execution level, the functions of these services and the added development that can be added

to the framework will be discussed in a future work section at the end in section 9.

6.2 Registering with BDIFS

Registration into BDIFS is done via the BDIFS portal and will involve human interaction. The person registering a user or service to become part of BDIFS should be the person with administration rights over the user domain or Service Provider computing domain as described in previous chapters. This is because the user registering will be the service that receives the bill for the use of BDIFS, payment if their service is used in a BDIFS workflow or penalty if the service provided breaks the SLA it is signed upto.

6.2.1 User Registration

Users are the main customer for BDIFS and invoke the Base VO for the execution of a specific BDIFS workflow / application. In the use case of Company A, administrator Ian registers with the BDIFS portal. The page he uses asks him the following information:

Personal Details - name, address, position etc.

Business Details - company name, address etc.

Type of membership - this signs the user up to the types of workflows on offer in BDIFS; currently there is one business integration workflow.

Level of SLA – in this design the user gets to choose between two levels of SLA. On top of a registration and use fee, the level of SLA will be a significant factor in determining the price of using the client to invoked BDIFS. Currently there are two levels of SLA on offer: SLA 1 application-specific services selected for the workflow with no back up and SLA 2 services with backup (i.e. two services) for each service required in the workflow.

Integration format – data format that the user wants to receive data in.

Add to code base – this allows the user to upload any integration scripts they may be using to integrate data from BDIFS into their own system.

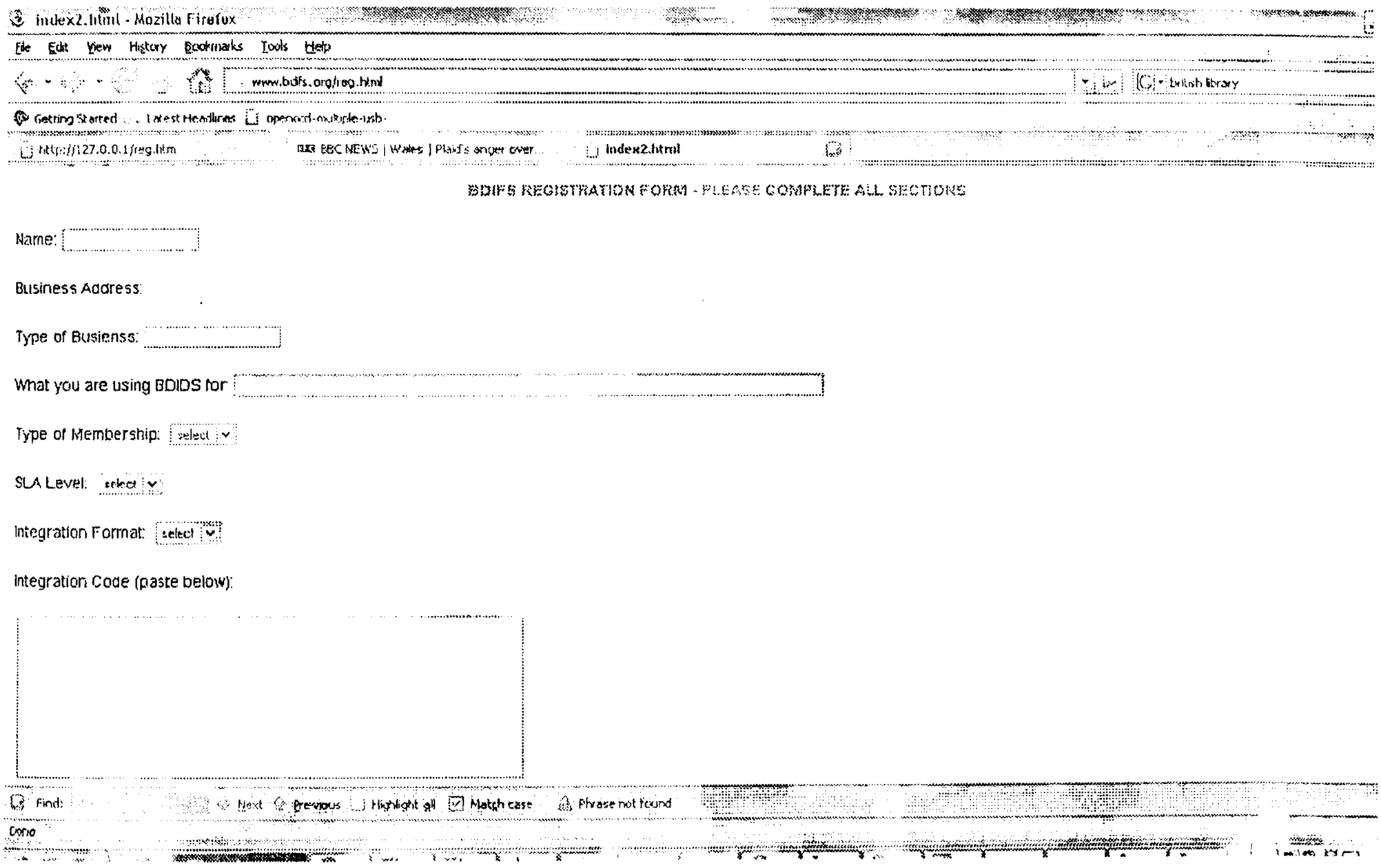


Fig 21: User Registration Screen

Once this form is completed the BDIFS portal will generate a BDIFS java client for the user to communicate to the BDFIS framework with. The client as already described will contain a token with the user's information and access rights, the user group to which the user is added and the type of data they wish to receive. Currently the BDIFS framework supports ebXML and some mfg/pro specific transformations, though it is possible to add more services and workflows.

The Registration process for a user is shown in the diagram below.

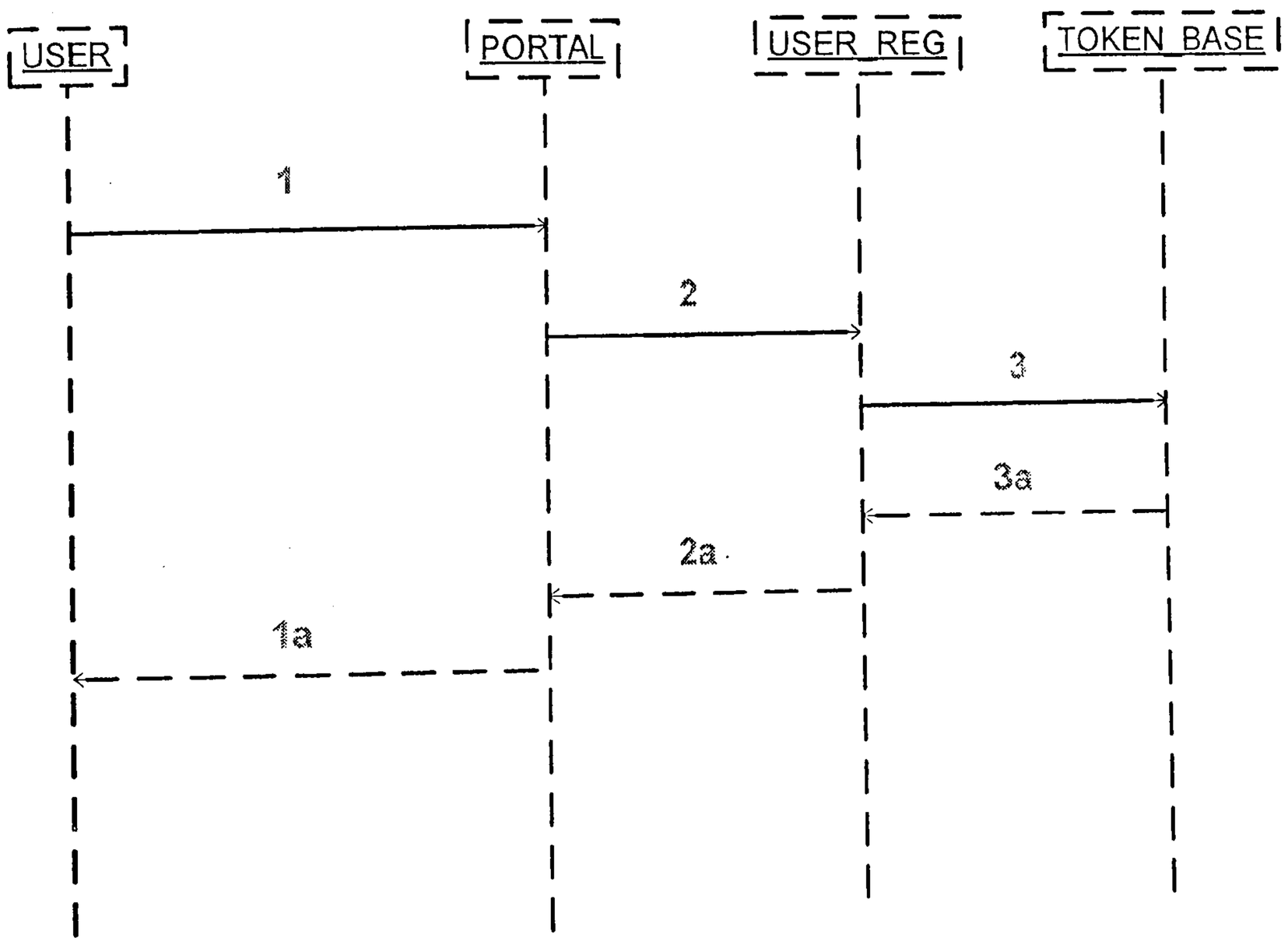


Fig 22: User Registration Process.

Following the numbers in Fig 22, the use case is as follows:

Step 1. User fills in the form on the portal to request a BDIFS client. In this case the user name is Tom Kirkham and the membership is for a user, the requested SLA is SLA1 and return data type is ebXML.

Step 2. The portal adds the user details to the userReg database, associating the user with a group. (The EPR of the userReg service is hardwired into the portal.).

Step 3. The userReg service passes the information about the new user to the TokenBase where a token is generated for the user. (The EPR of the TokenBase service is hardwired into the userReg.)

Step 3a. This token is returned to the userReg database and associated with the user.

Step 2a. The Portal is informed of the token to be associated with a user and generates a client for the user.

Step 3a. A notification is sent to the user on the Portal that the client is ready for downloading and use.

6.2.2 Using BDIFS

Once the user downloads the client they are ready to use BDIFS. No other activity takes part on the system. The other nodes / services in BDIFS are not notified of the new user, as the user has agreed to the terms of conditions of use by downloading the client. These terms are agreed by all participants of BDIFS. With respect to privacy, the user and Service Providers remain anonymous on the application-specific services, as the user only ever

communicates with the Base VO and application-specific services with their respective service agents. No direct user-to-service communication or application-specific service to application-specific service communication is possible within BDIFS. If there is a failure in the chain, the error message will be picked up by the invoking service and will be passed as a return value to the Portal.

6.2.3 Service Provider Registration

As already discussed, the services which make up the BDIFS system are divided between BaseVO services and Operative VO services. The Base VO services are permanent and already in place in the BDIFS system, providing functionality like token generation, VO Management and Workflow Management. It is the services that make up the Operative VO which are open to third-party developers that are separate from the user and Base VO owner roles.

These third-party services are referred to as application-specific services and provide the application functions central to the workflow. These functions in the current BDIFS design include data transformation and partner matching. The ability of third-party vendors to manufacture services to aid integration into their specific ERP type system and advertise them on BDIFS is central to the aim of the system. The incentive for services to join BDIFS is that they will be paid every time their service is used.

Service providers join their services to BDIFS through the same portal as the users of BDIFS. However the form they complete is slightly different. The key features of the form are listed below:

Personal Details - name, address, position etc.

Business Details - company name, address etc.

Type of membership – this is defaulted to Application Specific Service Provider.

SLA – here the Service Provider must agree to uphold certain SLA requirements reflecting availability of service and reliability when invoked. Also the SLA will cover the appropriate return value of the service. It is important that this can be controlled by BDIFS as it is essential to BDIFS understanding of the invocation result.

The Service Provider will also have to select from a series of options to best describe the service availability. This will go into the SLA template and aid service discovery during the workflow execution process.

Operation Supported – the Service Provider fills in the details of the operation that the service supports. This is done by selecting functions from a list and enables the service to be added into a specific group of services for use in specific slots in the workflow.

Service Provider Token – needed by the service agent when invoking the service on the Service Provider domain so it can authenticate.

Service EPR – is the address of the service that the BDIFS system has to invoke to call the service.

InterfaceName – the name of the interface that BDIFS should call when invoking the service.

Arguments to pass – aside from the token, here go any other arguments the BDIFS service agent should pass to the service when invoking it.

Add to code base – just like with the user, this allows the Service Provider to upload any integration scripts that may aid integration of the end data that the service produces.

When the form is completed, the portal offers the client a token to download. This token is used to authenticate on the VO side responses from the service when called in a workflow execution process. The Service Provider is then left with the job of keeping the service available according to the SLA selected. The sequence of events in this use case is shown in the diagram below:

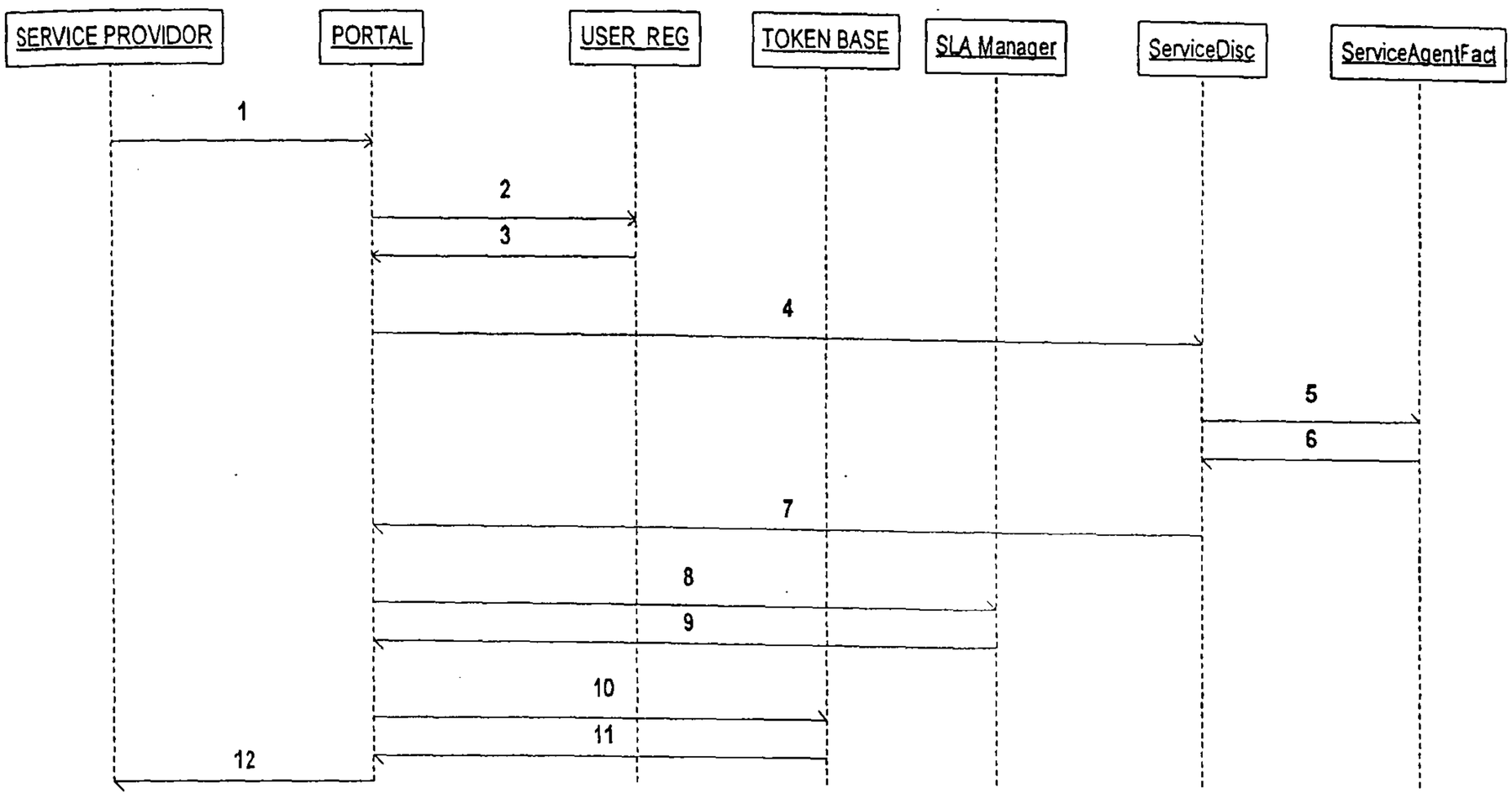


Fig 23: Service Registration.

Step 1. The Service Provider fills in the form on the portal to register the service with the BDIFS framework.

Step 2. Once this is complete the portal populates the userReg database service with the details of the Service Provider.

Step 2a. The userReg returns an acknowledgment and userID for the Service Provider.

Step 3. The portal makes a call to the service discovery passing the userID, service address and serviceEPR to the ServiceDiscovery service (EPR of service discovery is hardwired into the Portal.)

Step 3a. The Discovery Service informs the Portal of the result of the Service Discovery Service population and possible invocation test of the service by the Service Agent factory.

Step 4. Once the service is registered in the Discovery Service, the SLA Manager is updated with the SLA information of the service (EPR of the SLA Manager is hardwired in the ServiceDiscoveryService.)

Step 4a. The result of this operation is returned back to the Portal.

Step 5. The Portal service invokes the TokenBase to generate a token for the service.

Step 5a. A token is generated for the specific service and Service Provider, and is passed back to the Portal.

Step 1a. The token is made available to the Service Provider via the portal and, if needed, a service client is made available for download. The client can be used to wrap the existing service on the Service Provider domain in order to make it compatible with BDIFS.

6.3 OpVO Population

6.3.1 Authentication

As already explained the OpVO population phase is a central part of the BDIFS functionality. The population of an OpVO is carried out when a user agent requests a BDIFS workflow. The request is made using the client obtained by the portal and communication is made with the Base VO. This initial phase involves the BaseVO authenticating and authorizing the request from the Agent. If this stage is a success then the population phase begins.

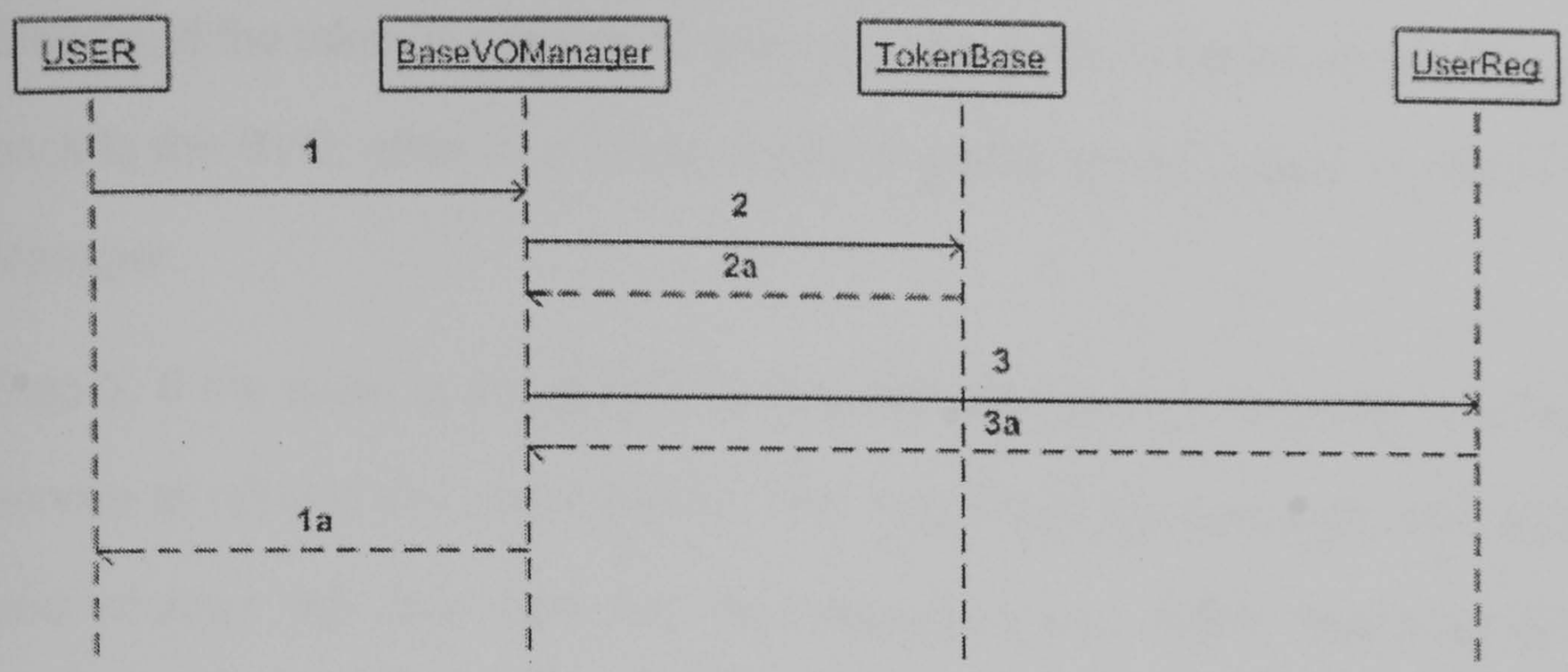


Fig 24: User Authentication.

Step 1. The user requests an application from the BaseVO, passing the token embedded in the agent.

Step 2. BaseVO checks the token is valid with the TokenBase; the validity is determined by the token being a proper token and also a token that still is live i.e. has not expired or been revoked (the EPR of the TokenBase is hardwired into the BaseVO.)

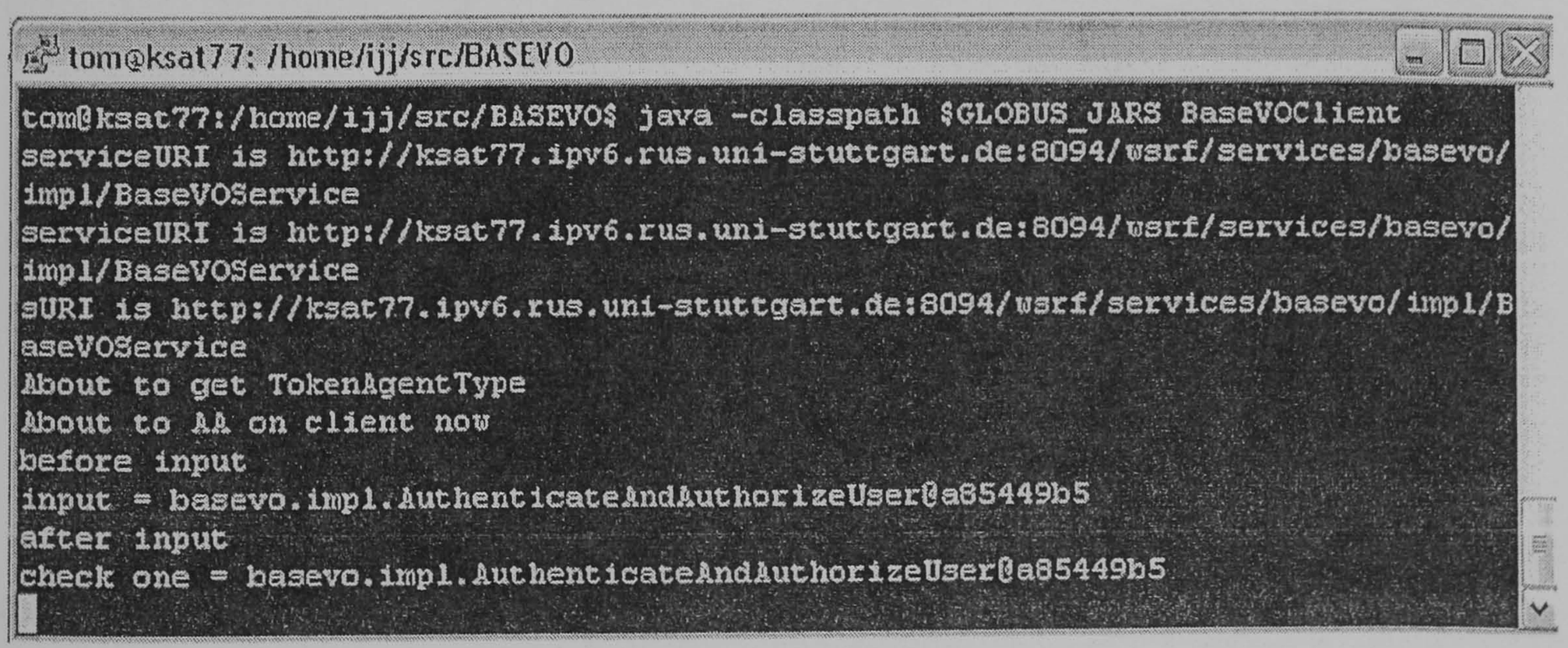


Fig 25: User Authentication Process

Step 2a. If the token is not a valid token then an error message will be passed back to the BVO; otherwise an ok message then it will be passed to the BVO Manager.

Step 3. If the token is accepted, the BaseVO Manager contacts the userReg service to retrieve the user's profile. This determines the user's access rights and whether the user can use the application they have requested (the UseReg EPR is hardwired into the BaseVO.)

Step 1a. The result is passed back to the user agent. If the profile and token are validated then the user is given a choice of SLA for the application they have requested.

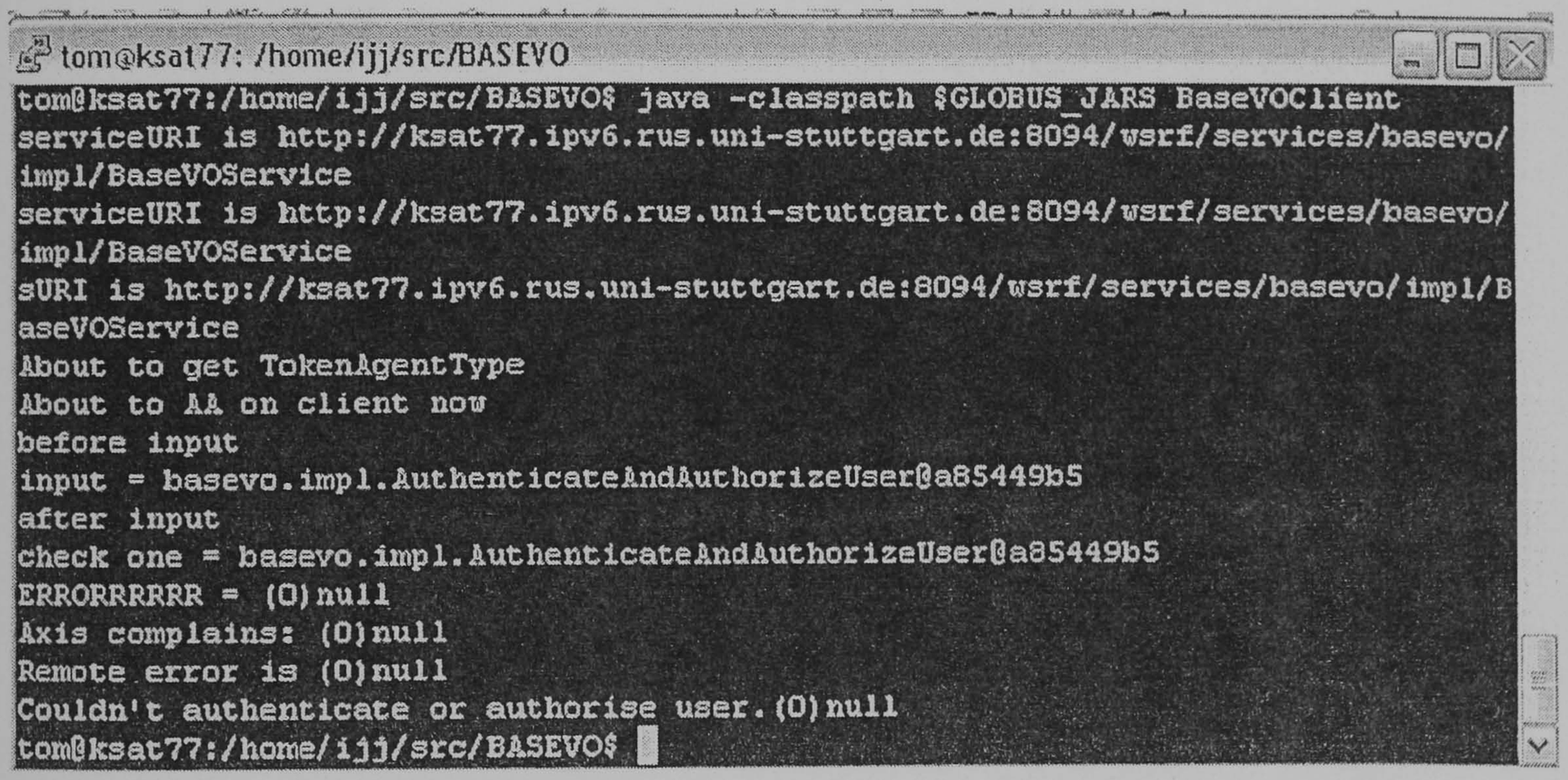


Fig 26: Authentication Failure

6.3.2 OpVO Set Up phase

The SLA choice is returned to the user agent, and new SLAs can be added without upgrading the client. Once the user selects the level of SLA for the

application they wish to run and notifies the Base VO of it, the OpVO set-up phase begins. This phase is illustrated in the diagram below.

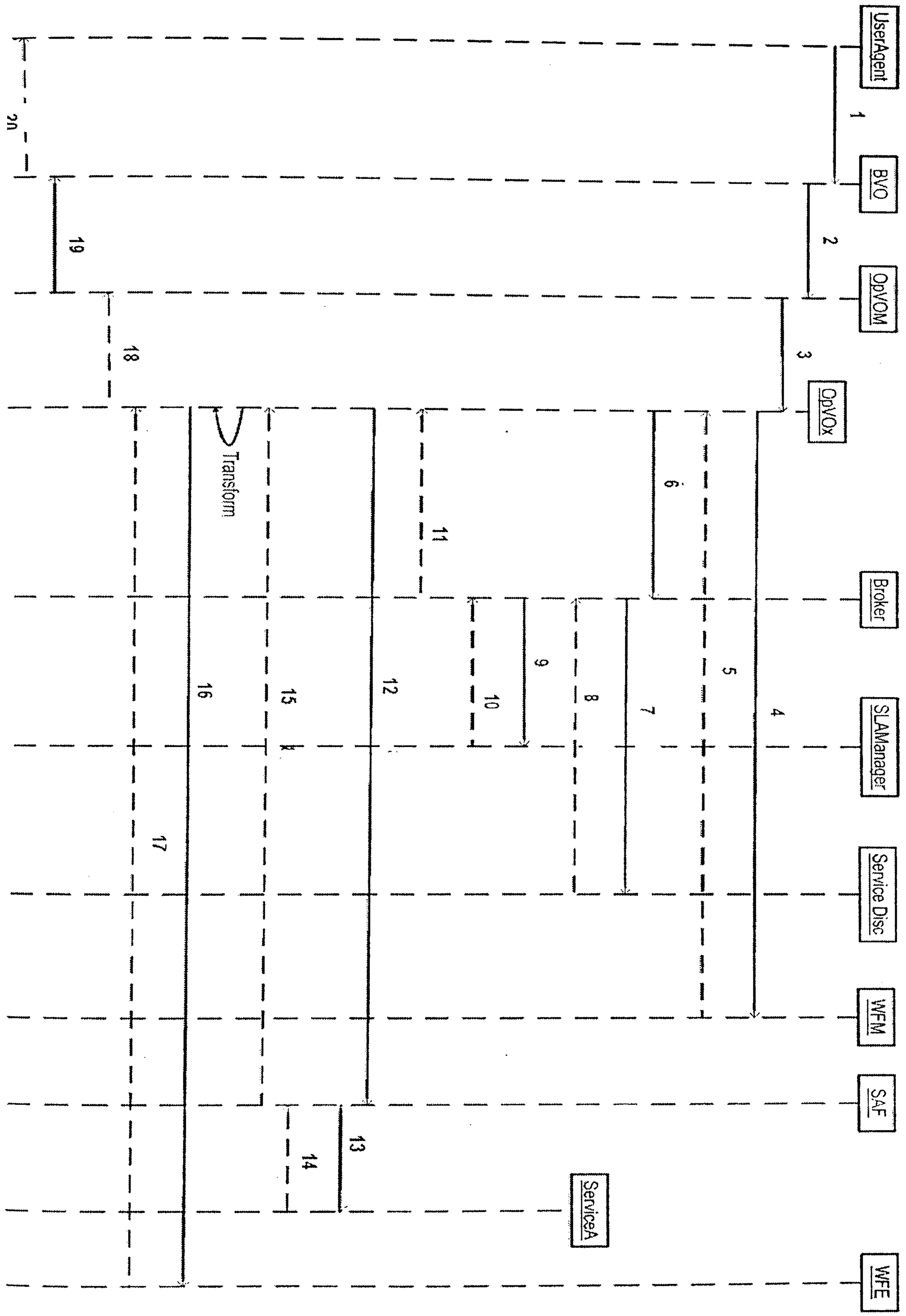


Fig 27: Workflow Deployment.

Step 1. The user requests an application from the BaseVO at a specific level of SLA.

Step 2. Base VO matches the application to a workflowID and forwards the request to the OpVO manager passing the workflowID (the EPR of the OpVO manager is hardwired into the BaseVO Manager.)

Step 3. The OpVO manager responds to the call from the BaseVO by creating an instance of itself and passing the workflowID into the new OpVO manager.

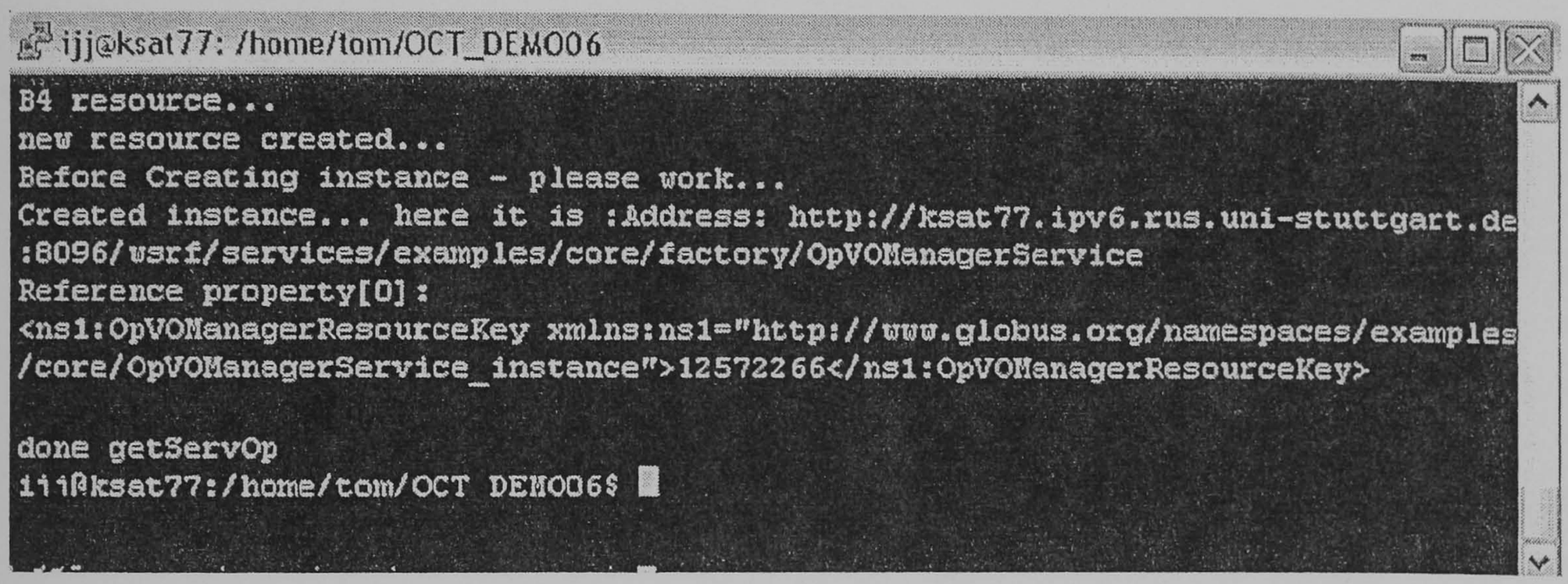


Fig 28: OpVO Instance Creation

Step 4. Once created, the OpVO instances start the OpVO population by retrieving the workflow template from the Workflow Manager. (The EPR of the WorkflowManager is hardwired into the OpVO Manager.)

Step 5. The Workflow Template is returned from the Workflow Manager and the OpVO Manager scans the information for the required services.

Step 6. The OpVO instance contacts the Broker to retrieve the EPRs of the services required for the workflow. (The EPR of the Broker is hardwired into the OpVO Manager.)

Step 7. The Broker contacts the SLA Manager with the SLA ID of the relevant services. (The EPR of the SLA Manager is hardwired into the OpVO Manager.)

Step 8. The SLA Manager returns a list of services matching the SLA criteria in this design. Service-level SLA is not implemented, so the list will be all relevant services regardless of SLA.

Step 9. The Broker selects from the list of services (which can be based on criteria such as pricing) and uses the Service Discovery Service to retrieve the EPRs.

Step 10. The EPRs of services are returned to the Broker.

Step 11. The Broker returns the ServiceAgent EPRs to the Operative VO Manager.

Step 12 The OpVO Manager contacts the Service Agent Factory passing the service EPRs to create the Service agents.

Step 13. The Service Agent factory creates the Service Agent. An extra step here could be to use the agent to call the service to check it is theirs.

Step 14. The Service Agent responds to creation from the Service Agent Factory.

Step 15. The Service Agent Factory returns the address of the new ServiceAgent to the Broker.

Transform: The OpVO Manager instance uses these returned ServiceAgents EPRs to populate the workflow template.

Step 16. The OpVO Manager deploys the Workflow to the Workflow Engine.

Step 17. If the deployment is a success, the Workflow Engine will return the address of the deployed service to the Operative VO Manager instance.

Step 18. The result of the OpVO population phase is returned back to the main OpVO manager. If it is a success, this will include the address of the OpVO Manager instance and the deployed workflow.

Step 19. The result of the OpVO population phase is returned back to the BaseVO manager.

Step 20. If the result of the OpVO execution is a success in step 19 an address of the OpVO Manager instance and the deployed workflow will be sent this is passed to the User Agent to end the process of population.

Fig 29: A Deployed Workflow

The process of population is purely a Base and OpVO process once the call has been made from the user. In the first BDIFS testbed many of the Base

Components like the Discovery Service and OpVO Manager are on the same machine so there is no security between the components. Therefore, to make the process more scalable, security in the form of token passing and building trust relationships needs to be added between all services.

The instance of the OpVO Manager and Service Agent are the only WS resources that are created. This adds a degree of scalability as the process could be running several OpVOs in separate dynamic environments. Conversely the Workflow Manager is a bottleneck to the control of the workflow. This workflow control process will be discussed later, but making the Workflow Manager a WS resource may further increase the flexibility of the system.

6.4 Workflow Execution Phase

The workflow instantiation phase leads on directly from the workflow population phase in the BDIFS testbed. The process is started by the BaseVO as soon as the population phase is completed. Once the workflow is deployed, it is possible to execute it at a later time. However as BDIFS envisages a system where services sign up to SLA and availability, it is likely that if the time between workflow deployment and execution is too long then some of the services could become unavailable. The process is shown in Fig 30.

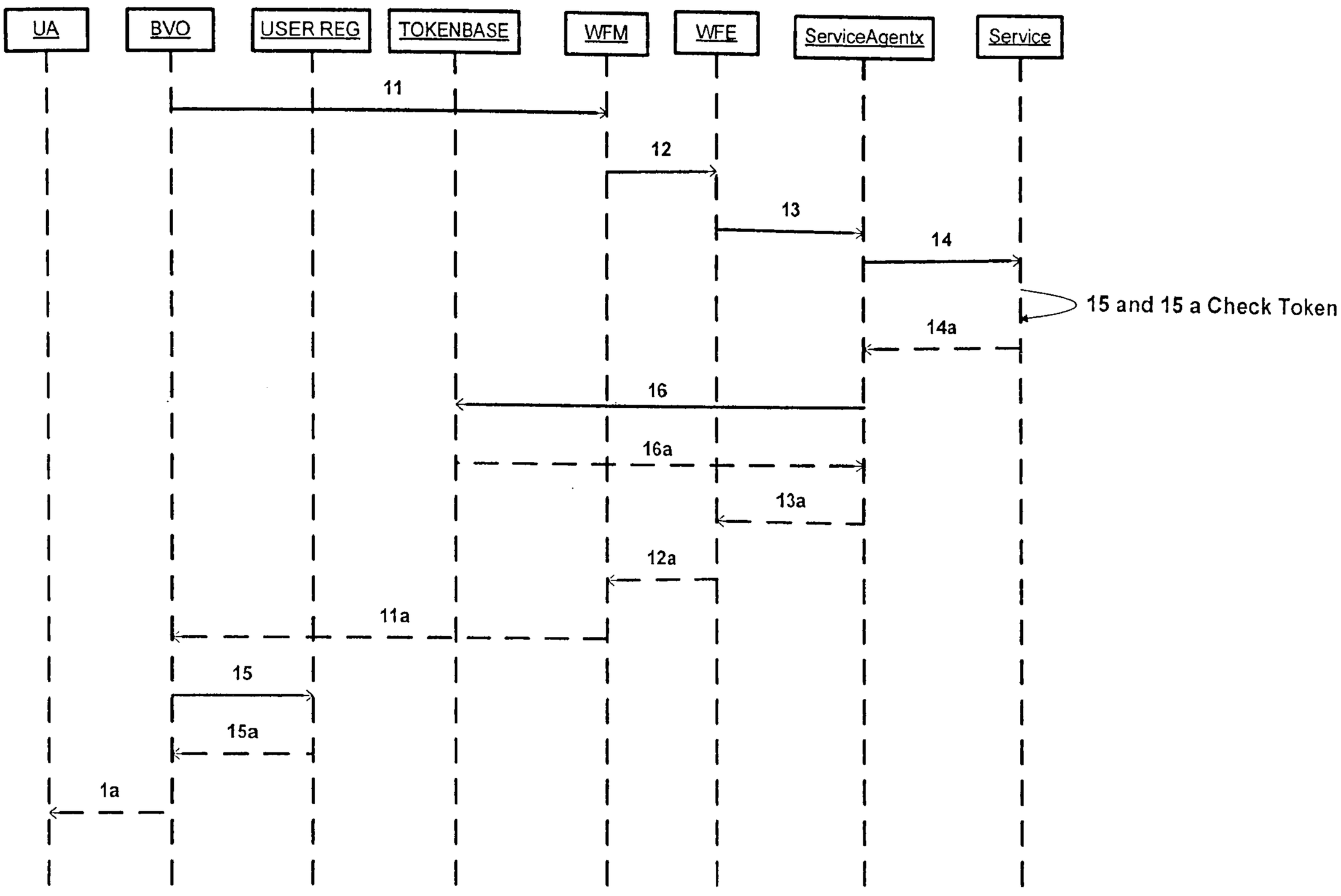


Fig 30: Workflow Execution Process.

Step 11. BVO calls the Workflow Manager execute workflow method, passing the address of the deployed workflow.

Step 12. The Workflow Manager calls the workflow engine's deployed service asking to execute it.

Step 13. The Workflow Engine executes the workflow, calling each service listed within it. In this cast there is only one service represented by ServiceAgent_x.

Step 14. ServiceAgent_x calls the service in the Service Provider domain it is associated with, passing the Service Agent domain token.

Step 15. The Service checks the token passed by the Service Agent using the Service's local TokenBase.

Step 15a. The result of this check is sent back to the Service from the TokenBase.

Step 14a. The externally hosted service replies to the call from the ServiceAgent_x, passing the Base VO token associated with the Service's domain.

Step 16. The Base VO token sent from the Service is checked with the TokenBase in the Base VO domain.

Step 16a. The result of this token check is sent back to the ServiceAgent.

Step 13a. If the token is ok, the ServiceAgent_x returns the result of the call from the externally hosted service to the workflow engine.

Step 12a. The workflow engine, once complete or faulted due to error, informs the Workflow Manager of the result.

Step 11a. The Workflow Manager passes the result of the workflow execution back to the Base VO Manager.

Step 15. The BaseVO Manager adds to the user record the result of the workflow execution.

Step 15a. The userReg confirms it has received this data and adds information to the Service Providers of the services used in the workflow execution that their services have been used. This data and the user's data in relation to the use of the workflow can be retrieved periodically to perform the billing and payment mechanism in BDIFS.

Step 1a. The result of the BDIFS application execution is returned to the user agent.

6.5 Service Failure

During the service execution phase, when the workflow is executing, the reliability of services is not guaranteed. The only guarantee the system has of the services behaving correctly in the Service Provider domain is the word of the Service Provider in the SLA agreement that they sign up to when joining BDIFS. Thus the SLA templates penalties must ensure the SLA is not broken, and monitoring of SLAs in the system must be sufficient.

Service breach of SLA is an official example of service failure, as the SLA that the Service Provider agrees to provide services to suit covers various unexpected service behaviour. These can include but are not limited to service unavailability, service failure during execution and the return of wrong information from a service. Within the BDIFS testbed the example of service failure during execution is illustrated. This is reflected in the two levels of SLA cover that the user is offered in case such an occurrence happens. These two levels of SLA are workflow-based and are reflected in the two SLA failure cases shown in Fig 31 below. The diagram deals with the failure with SLA 1 provision, which has no reserve services.

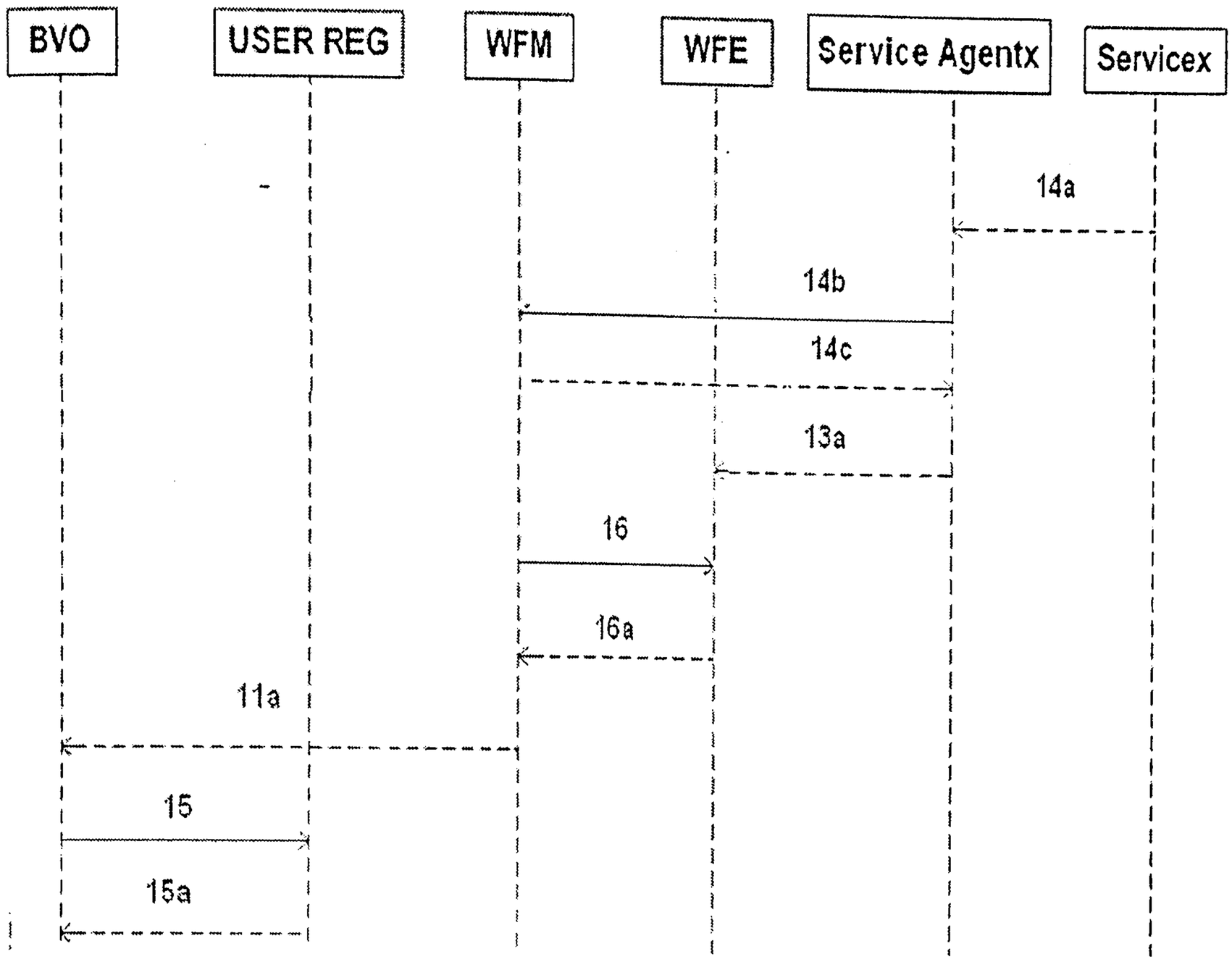


Fig 31: SLA 1 Service Failure.

The service failure process is as follows;

Step 14a. Service failure passed to the Service Agent.

Step 14b. The Service Agent informs the Workflow Manager of the service failure.

Step 14c. The Workflow Manager makes a decision on the failure and its effect on the workflow, and informs the ServiceAgent of the decision.

Step 13a. If the service failure is not essential and the workflow can carry on then the result is returned to the workflow engine. If it is essential then this step is skipped.

Step 16. Depending on the type of service failure, the Workflow Manager can ignore the failure, suspend the workflow or terminate the workflow altogether.

Step 16a. The Workflow Engine sends an acknowledgement to the call of the Workflow Manager, based on the call made in step 16.

Step 11a. In all cases of failure, the Workflow Manager informs the Base VO of the service failure and the status of the workflow engine, even if the workflow completes with the failure in place. If the failure caused the termination of the workflow, this notification allows the Base VO to start the service discovery phase again and restart the workflow.

Step 15. In all cases the Base VO adds the record of the service failure to the records of the Service Provider so that action can be taken according to the terms of the SLA.

Step 15a. The userReg acknowledges that this information has been updated.

The service failure of the application-specific service in the Service Provider domain is detected by the Service Agent invoking the Service. As discussed, common failures include wrong data returned to the service becoming unavailable. In this case there is no backup service in the workflow so the error is returned to the Workflow Manager that either suspends or cancels the workflow. The Base VO is informed, which populates the userReg to assign the failure to the specific Service Provider so that penalties can be issued. It is likely that the OpVO population phase would need to be repeated before the workflow is invoked for the user again. In the case of SLA 2, the following use case is observed.

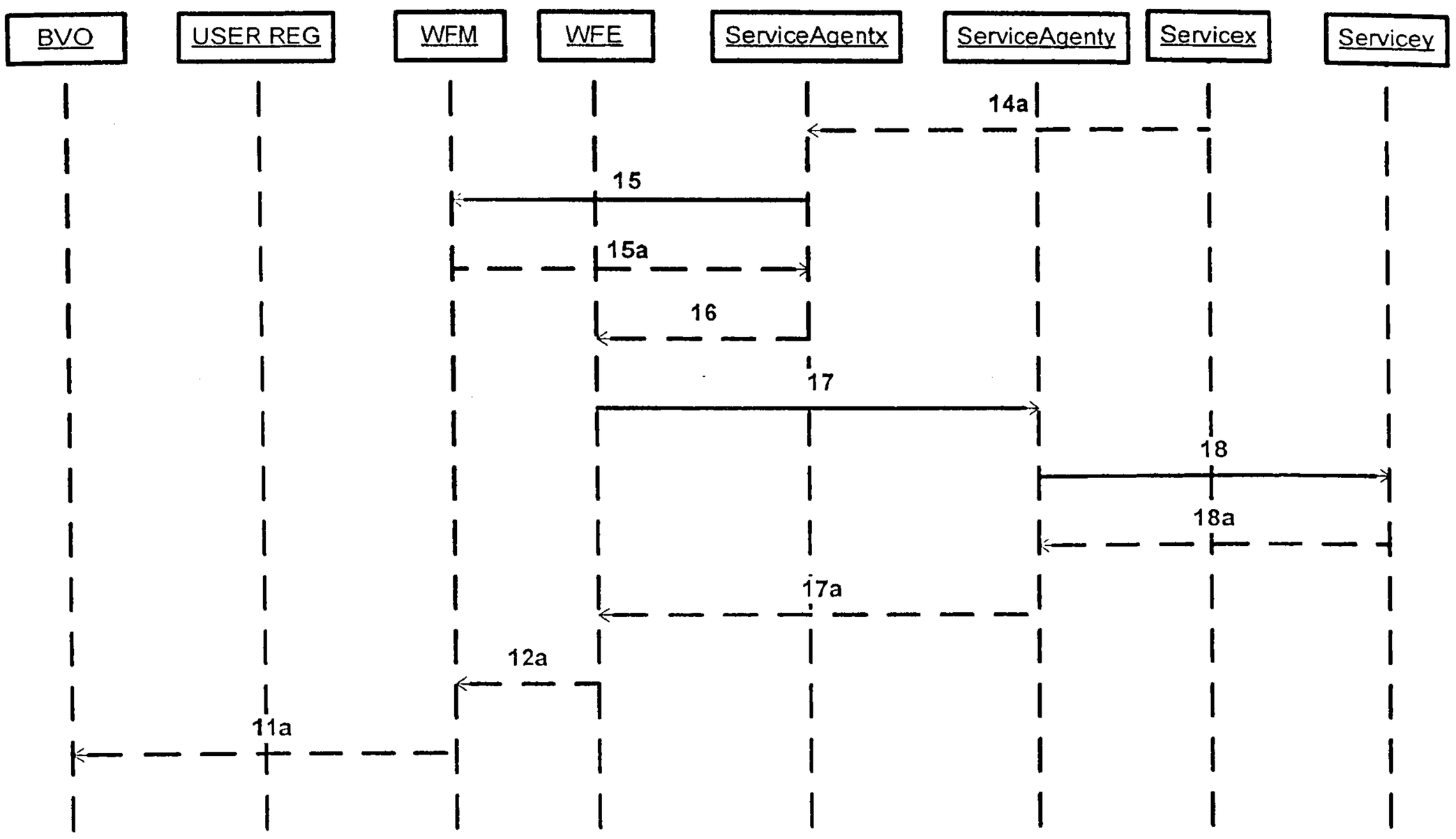


Fig 32: SLA 2 Service Failure.

Step 14a. Service (Servicex) returns a failure to the Service Agent (ServiceAgent_x).

Step 15. Service Agent informs the Workflow Manager of the failure.

Step 15a. The Workflow Manager acknowledges the failure notification and makes a decision.

Step 16. If the decision is that a new service is needed, the Service Agent notifies the workflow engine that the service has failed.

Step 17. The Workflow Engine then executes the back-up Service by calling ServiceAgent_y.

Step 18. ServiceAgent_y calls the back-up service (Service_y).

Step 18a. The Service returns its result to the ServiceAgent.

Step 17a. The ServiceAgent returns the result to the Workflow Engine. If this is another failure, the process in the SLA 1 scenario will be executed. Otherwise the process will continue as outlined in the diagram.

Step 12a. The Workflow Engine notifies the Workflow Manager when the workflow cycle is complete.

Step 11a. Workflow Manager informs the Base VO of the completed workflow execution and any service failures that were encountered.

Step 15. The Base VO updates the userReg for both the Service Providers and user involved in the execution of the workflow.

Step 15a. The userReg acknowledges the update and informs the Base VO. The Base VO can then enter the destruction phase if the workflow was a success.

In the SLA 2 scenario, for each service used in the workflow a back-up service is also reserved within the Service Provider domain. As the scenario illustrates, when the workflow engine is informed of the failure by the Service Agent the back-up service agent is invoked. The Service Agent on which the service failed also informs the Workflow Manager of the service failure.

If the back-up service works, the process of workflow execution will finish as illustrated in the normal workflow execution scenario. However, when the Workflow Manager passes the details of the execution to the Base VO Manager it will also pass information on the failed service. The Base VO will

then add this information against the Service Provider's information in the userReg, as in the previous case.

The example of SLA provision demonstrated is workflow centric. The failure of services is solved by the workflow engine instantiating a back-up service or the Base VO starting the workflow population process again in order to start another workflow. A service centric SLA is also possible where for example replacement services can be selected in the Service Provider domain, a process that can be hidden from the workflow. The Service Provider could be encouraged to do this by strong SLA penalties. However, the best way in which reliability can be ensured via the Base VO is by this workflow provision of SLA. The Base VO could charge the user for this on top of the fee for the application execution, thus ensuring a quality of service.

6.6 Service Destruction

Once the process is complete, the infrastructure created for the workflow execution needs to be destroyed. This can be done easily by using the destroy mechanism in WSRF. The Operative VO and Service agents created for it can be destroyed by issuing this command – a process that is started from the Base VO and illustrated below in Fig 33.

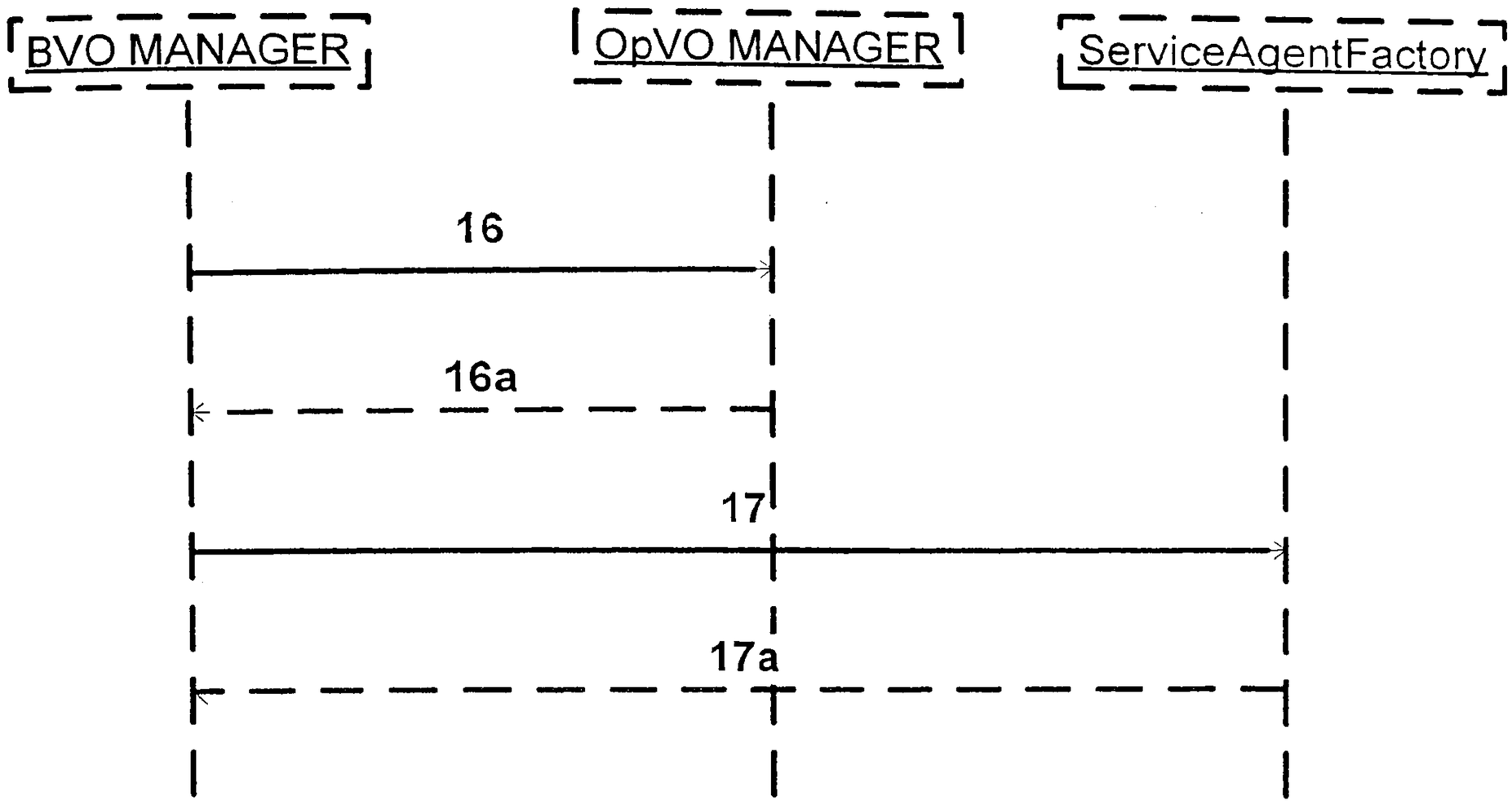


Fig 33: Service Destruction.

The process of service destruction is as follows;

Step 16. The Base VO calls the OpVO Manager to destroy the instance of the OpVO Manager created for the application that has executed. It uses the OpVO instance address passed back to the Base VO in the initial OpVO instantiation phase.

Step 16a. The OpVO Manager confirms that the instance has been destroyed and returns the Service agents EPRs to the Base VO.

Step 17. The Base VO calls the Service Agent Factory to destroy the specific service agent instances used in the application. The Base VO passes in the OpVO Manager instance as a reference.

Step 17a. The Service Agent Factory confirms in the return that the instances have been destroyed.

The destruction phase is simple and is led by the main point of authority in BDIFS, the BaseVO. The Base VO Manager uses the instance address of the OpVO created for the application as the reference by which the services that are to be destroyed can be linked to.

6.7 Testing Methodology

Testing was carried out using the Java tool TestNG [159]. This was used to detect faults in the Java code during run time. The testing scenarios followed the use case that has been discussed. The main problems encountered during testing were synchronisation of containers with respect to the individual services all being alive. In retrospect it would have been an advantage to have developed a GUI tool to display the availability of all BDIFS services.

Various test clients were also developed to isolate service execution without having to go through the whole set of use cases. These clients were developed using Java. On the whole the services worked well and the use cases executed with no problem. The development of the Portal and generation of clients also works in a satisfactory manner. Future tests will be conducted with Company A with the aim of delivering a prototype that is robust for use commercially.

6.8 Compatibility Issues

A key issue with the design of BDIFS is the support for other Web Services not designed with GT4 libraries or Java. A good example of this is the communication made to the Enactment Engine. This service does not use Axis or GT4 libraries and communication with the service was a problem, caused by trying to call the service from an Axis container. Possibly the container was unable to process the return from the Tomcat container. As a workaround, the workflow was contacted by a Java Client on BDIFS2. The service called the client by starting a shell script.

A major issue addressed in this work was GT4 compatibility with WSRF.net. The key issue with compatibility is the WSDL2Java conversion tool which is part of Apache Axis. The process works by contacting the WSRF service and converting the WSDL to Java stubs for use in generating a Java client for the service. This issue was found when integrating the OpVO Manager Service with the Akogrimo gird middleware. The problem here is that there are various anomalies between addressing types as defined in the WSRF.net WSDL as opposed to the Java GT4 version. In order to make these two types of services compatible, the process has to be broken down into two steps.

The first step is to generate the WSDL from the WSRF.net service. This was done using a WSDL grabber tool. Once the WSDL is generated, the second step takes place. By having the WSDL locally it can be manually edited and

the references to the addressing versions etc that cause the errors can be manually removed when using the WSDL2Java generation process. This is something that can't be done when the process is pointing at an externally hosted service out of reach.

The other error encountered when integrating with the Akogrimo project, is the use of WSRF.net EPRs within GT4. Within BDIFS, the framework has not yet been tested with WSRF.net services but it will need to use them. In the use case the EPR of the service will be provided to the Service Agent factory so the service can be called. However in tests using this during the Akogrimo integration with BDIFS the EPR was again incompatible with the GT4 services using the Axis server. Consequently there was a need to modify the EPR to reflect libraries and formats that we could use via Axis. This was done using another bit of customised code that the author helped develop, that works by reading the XML of the WSRF.net EPR and stripping it of the core values to form a GT4 compatible EPR out of it.

The main problems that have been encountered with the design have been in the compatibility of the Globus design of WSRF.net with other Web servers. Whilst this cannot be seen as a problem with WSRF directly, it is a significant problem that may hamper the development of WSRF services, even though in this project no services have been developed using WSRF.net which may integrate better with other forms of WSRF and Web Services.

Developing services using GT4 and Apache Axis therefore are a potentially unstable mix when trying to create an design like BDIFS. The problems with compatibility need to be solved within the Globus and Axis community before the use of Java developed Globus WSRF services is as a viable option. The use of other forms of WSRF is a possibility, but the problems communicating with the main Java design of the technology has an impact on this use too.

6.9 Summary

In summary this chapter has described how the BDIFS framework provides both its key functionality and innovations. The BDIFS Web Service design is built on the main principles developed in the P2P design and presents them in a more flexible architecture. The main enabler for this flexibility is the ability to create Operator VOs and Service agents dynamically. The design of the Web Service architecture has focused on the development of services using the Globus Toolkit version 4 and this has provided a stable framework.

However testing with services developed using other WSRF toolkits such as WSRF.net and standard Web Services such as provided by the Active BPEL enactment engine has proved to be more difficult. The problems here are related to how the individual servers behind the toolkit designs interpret the EPRs sent and returned by the others. Thus the web servers used in this project, IIS, Axis and Tomcat have issues when integrating using WSRF. In this project workarounds were found to this issue but to date the main problem

resides with the Axis web server. Thus for future commercial development other web servers may need to be addressed if these problems with Axis remain.

Chapter7 Applying BDIFS

The project has been applied within three main sectors branching out from the integration of business data. Elements of the project have been applied in a mobile grid computing environment, a business integration testbed and eScience facilities integration project. Each one of these applications has involved software developed in BDIFS, and has accomplished the integration of data from one source to another. The first of application that will be introduced is the use of the BDIFS Operative VO Manager in the EU Framework 6 project Akogrimo.

7.1 Akogrimo

During the latter stages of the research into the BDIFS project the author has been working on the European Framework 6 Project Akogrimo. The Akogrimo

project aims to bring together the market orientation and pervasiveness of mobile communication technology with the promise of a dynamically concerted use of resources and services provided through Grid infrastructures [160]. In order to achieve this, the project is focused on the use of data from the mobile network level in a Grid application, and presents new challenges for the Middleware designer in a similar way the integration of legacy system data has in BDIFS. The Operative VO manager from BDFIS has been implemented in the Akogrimo project.

7.1.1 Mobility a challenge for BDIFS

Like the challenges in BDIFS, currently the use of mobile data is tied to the network to such a degree that mobile devices present data from this layer in various and often non standard-based ways [161]. For the SME this has the potential to widen the scope of BDIFS from legacy system integration to integration of data from mobile devices. To integrate current business software with devices that measure mobility of staff, equipment etc, involves often integration of propriety and varied standards as the software on these devices tends to be vendor specific [162].The challenge of reacting and using mobile data, therefore is again emerging to be one that larger companies can invest in and take benefit of. Leaving SMEs wishing to integrate data from such applications as vehicle tracking etc with their legacy systems, but are unable to do so without this investment. But, the use of the BDIFS OpVO component in Akogrimo has opened up a means by which this integration can be achieved.

7.2 Operational VO integration

In the initial application of Akogrimo, the BDIFS OpVO manager has been used within the eHealth Monitoring testbed. This work was demonstrated successfully at a major IST event in Helsinki in November 2006 [163].

Integration of mobile data in Akogrimo is achieved by the use of mobile service instances that are registered with the Base VO, but are located in separate hosting environments and only become active within the Base VO infrastructure, when they are incorporated in to an Operational VO. This process is tied to workflow population and execution and is a similar pattern followed by many other Web Service based frameworks including BDIFS. During this execution process the core Base VO services work together to ensure reliable instantiation and execution of services. This flexibility ensures in the case of service movement or failure the application can dynamically adapt and find new services. Supporting this in Akogrimo is the OpVO manager software that what was developed for BDIFS.

In order to support service selection during workflow execution or service reselection in Akogrimo, The OpVO Manager works closely with a broker component that links to the Service reservation services, Akogrimo SLA managers and service agents. Fig 34 illustrates, the Operational VO Manager and Broker both reside in the Base VO and Operational VO.

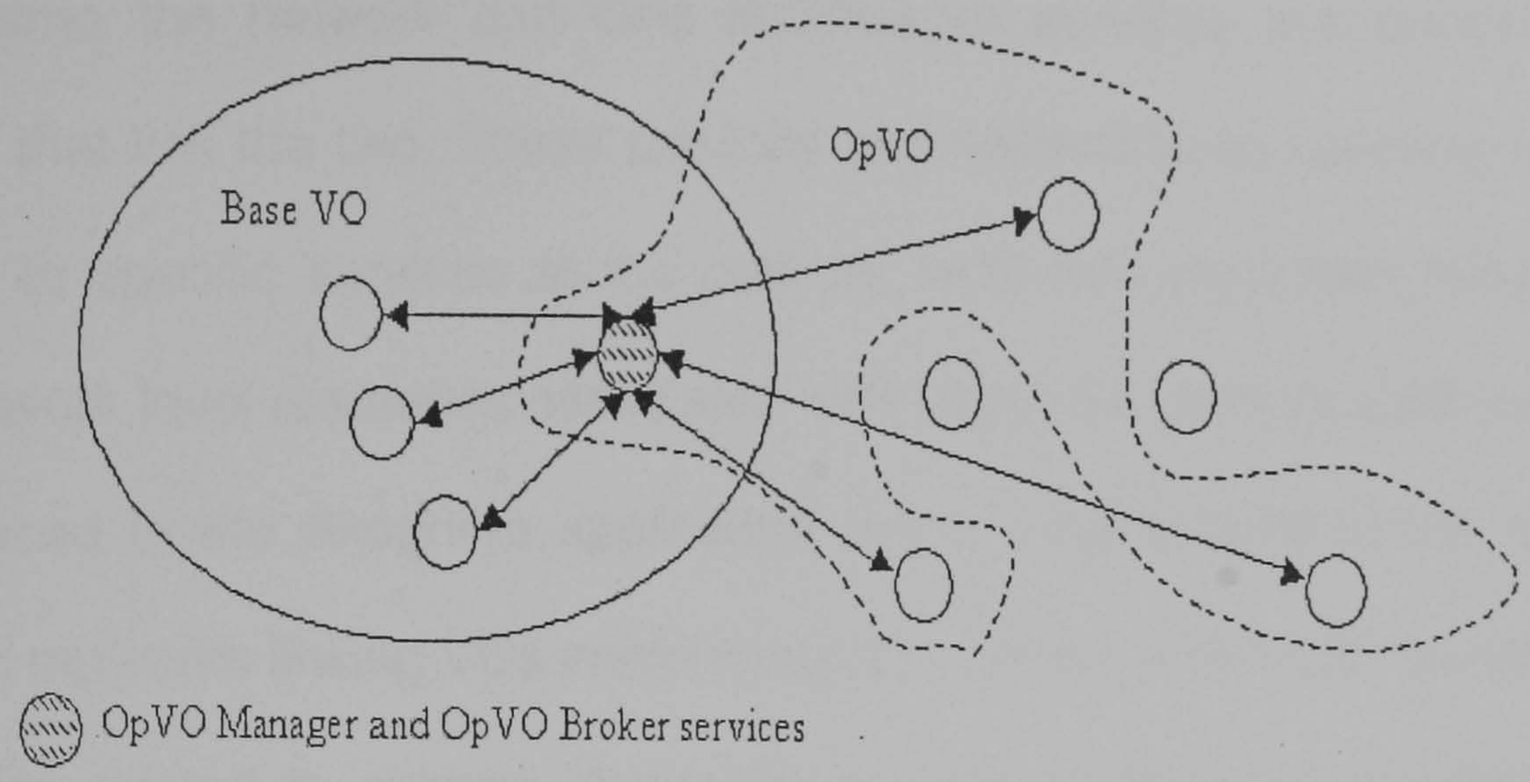


Fig 34: Operational VO and Base VO Linkage

This approach is similar to BDIFS where the Operational VO and Broker are linked. But the difference between the two architectures is the broker's linkage to a more delicate service reservation infrastructure to handle both the instantiation and management of mobile services, while BDIFS brokers and selects services on business specific rules.

7.3 Supporting mobile services in BDIFS.

In order to give the integration of the OpVO manager with the Akogrimo middleware more scope, architecture to integrate mobile service use within BDIFS has been devised. This centres on the way mobility is expressed from the network layer and is captured via a series of services linking it to the Grid Middleware. The positioning of these services is a logical starting point for the support of mobility in the BDIFS project.

In Akogrimo the network and Grid Middleware services are bridged with services that link the two. These services are referred to as gateway service and link to specific services at the network level that represent information from network level resources associated with protocols such as QoS and SIP. SIP is used in the Akogrimo application to link data related to the session between networks linking VOs enabling application such as VOIP transfer The QOS data related to network performance statistics that are monitored like traffic levels are represented at service level, Fig 35 illustrates a simple scenario where the SIP/SOAP service is used to link to the mobile network layer with the BDFIS Grid Middleware.

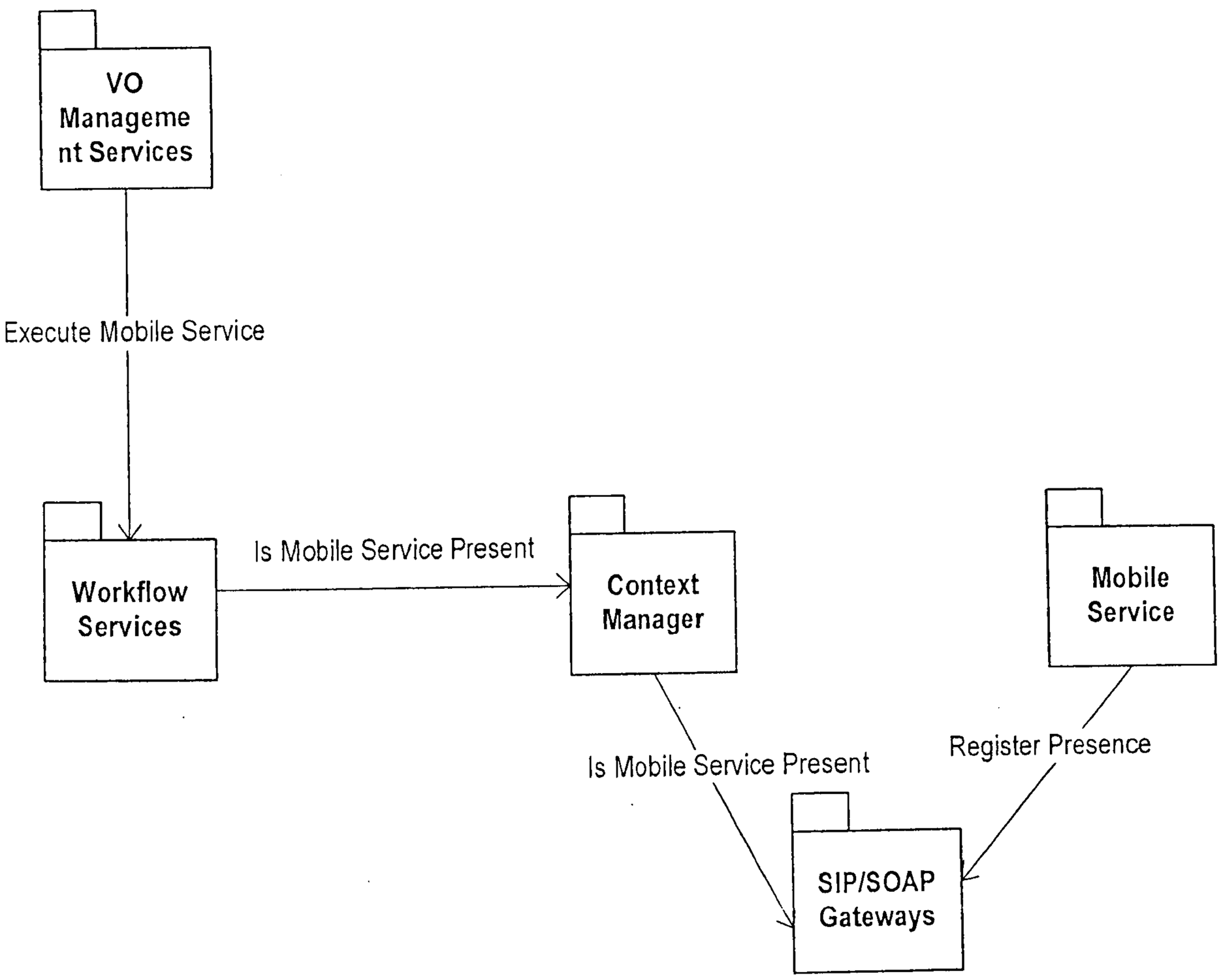


Fig 35: BDIFS Meets Akogrimo.

Within both Akogrimo and the BDIFS frameworks the workflow is designed to execute services or agents presented as services, in an orchestrated fashion, and is the point at which the two architectures can join. The difference in the workflows is that the BDIFS workflow selects and executes services according to their advertised state at time of registration. Whilst, in a mobile environment this is likely to change, and initially a service discovery phase is required prior to execution. This phase is then built on with Middleware services monitoring the service's state up until the point when execution is needed.

Thus the use of Mobile services in existing non-mobile Web Service architectures can be achieved by adding new services at the point of the workflow. These services are needed to specifically link and present data to the Grid Middleware from the Network. This process is relatively simplistic in terms of introducing gateway services and Middleware services to represent elements of mobility such as context. However in respect to integration the increased amount of data and new types of information gained from and required by a mobile network can make the task of workflow design and execution more complex. As the architecture adapts to incorporate mobile services into existing workflows, greater support for the workflow execution process is therefore needed.

7.4 eScience Application

Between January and April 2007, sections of the BDIFS framework were applied to a data management project at the Central Laser Facility (CLF) at RAL [164]. For this project BDIFS was applied to assist in the translation of data sourced from experiments using the facilities lasers. The application of BDIFS in this domain led to the development of a new BDIFS service, called the Translation Mapping Service.

The application of BDIFS on the facility has followed the BDIFS Web Service design, with a BDIFS client collecting data deposited by the laser and sending it to a service to perform the translation. The format of the data extracted from the laser consists of both textual and sometimes binary format data. In order to manage the translations two services were created, one for the binary and the other for the ASCII data.

The design architecture followed the BDIFS design and the two translation services were deployed as Web services, with communication from the machines creating the data to the services via a Java client. The translation of data was achieved by the storage in the services of an array expressing the position of the static data in the translations. This array was created using a new tool developed for BDIFS to aid mapping.

The tool essentially is a graphical user interface (GUI) where users paste in the original format of the data, the original with the variables removed

replaced with a consistent string, and a final desired version of the document, as illustrated in the following figs. 36-38.

```
FNAME:20061109GL0000003N_PUMP_RIG.dat;SECTION:LA3\N_PUMP\RIGOWSKI
DATE:20061109;TIME:13:59:17.0
TIMES:50357
SHOTNUM:3
DIM: 1
ARRAY: 1000
DATASIZE:SNG;FORMAT:1D traces
CTSTIME:OFF
PARENT:
CHANS: 5;CNAMES:RIG_3
RIG_6
RIG_9
RIG_12
RIG_15
DATASTATE: 0;
EOH]
```

Fig 36: Original Data Format

```
FNAME:<VALUE1>;SECTION:<VALUE2>
DATE:<VALUE3>;TIME:<VALUE4>
TIMES:<VALUE5>
SHOTNUM:<VALUE6>
DIM:<VALUE7>
ARRAY:<VALUE8>
DATASIZE:<VALUE9>;FORMAT:<VALUE10>
CTSTIME:<VALUE11>
PARENT:<VALUE12>
CHANS:<VALUE13>;CNAMES:<VALUE14>
DATASTATE: <VALUE15>
EOH]<VALUE16>
```

Fig 37: Original Data Format With Values Marked in Regular Format

```
<NXroot NeXus_version="3.9.0" XML_version="mxml"
file_name="performance-xml.nxs" file_time="">
   <NXentry name="gemini">
    <CNXdaily name="daily">
      <fname>value1/</fname>
      <section>value2/</section>
      <date>value3/;TIME:value4/</date>
      <time>value5/</time>
      <shotnum>value6/</shotnum>
      <dim>value7/</dim>
      <datasize>value9/</datasize>
<format>value10/</format>
<cstime>value11/</cstime>
<chans>value13/</chans>
<datastate>value15/</datastate>
<NXdata name=value1/>
 <NXdata>
 value16/
 </NXdata>
```

Fig 38: Final Data Mapping, Based on Desired Format with the Positioning of Appropriate Values Included.

Once these three formats are input into the GUI, the user can submit them. The BDIFS framework makes a calculation on the exact points of each document where the marked variables occur. This is done by detecting the point at which the first variable occurs and counting back to capture the text before it. This is then used to determine in the transformation at what point the first value starts. The end points are calculated in the same way, so a series of stable reference points are built up in the document.

This information makes up the mapping for that specific translation and is stored within the BDIFS services. It is envisaged that these mappings can be stored and selected by other users.

7.4.1 Using XSLT

As a footnote for the eScience application a similar translation was conducted using XSLT. This standard is W3C ratified and is noted by the web service and wider computing community as the best way to conduct distributed data transformations. This is due to the fact that XSLT defines in common mechanics the means by which a XML based set of data can be transformed. Also well developed parsers in the form of DOM [ref] and SAX [ref] exist to parse and transform XSLT formatted data.

The XSLT approach takes a XML file and transfers the data into a different format using a XSLT defined template that is pivotal to the translation. This XSLT is more desirable than the translations above as it is based on clearly defined namespaces and transformation, it does not depend on white space and text positioning. Thus making the use of XSLT a more stable way of transforming. Part of a example of an XSLT template developed to transform a workflow file used in Akogrimo can be seen in Fig 39 below.

```
<!-- For <process> element, need to ensure we generate all namespaces (otherwise xmlns:bpelns
<pddxsl:template match="pdd:process">
   <pddxsl:copy>
     <pddxsl:apply-templates select="@*|node()" />
   </pddxsl:copy>
</pddxsl:template>

<!-- The attributes of <process> seem to need to be put out one by one.  There must be a bette
<pddxsl:template match="pdd:process/@location">
   <pddxsl:attribute name="location">
     <pddxsl:value-of select="."/>
   </pddxsl:attribute>
</pddxsl:template>
<pddxsl:template match="pdd:process/@name">
   <pddxsl:attribute name="name">
     <pddxsl:value-of select="."/>
   </pddxsl:attribute>
</pddxsl:template>

<pddxsl:template match="pdd:partnerLinks">
   <partnerLinks> <pddxsl:apply-templates/> </partnerLinks>
</pddxsl:template>

<pddxsl:template match="pdd:partnerLink">
   <partnerLink>
      <pddxsl:attribute name="name"><pddxsl:value-of select="@name"/></pddxsl:attribute>
      <pddxsl:apply-templates/>
```

Fig 39: XSLT Template

This application of XSLT was slotted into the OpVO framework as a specific translation service, and like the example above a repository of XSLT templates could be developed to assist in the pooling of translation resources. However XSLT is a complex language and unlike the example in the BDIFS application it would take a large amount of skill to develop a automated mapping tool to place a random text based format into a XSLT template. Thus it is likely that future use of BDIFS may need to support both methods until such a tool is developed in the project.

7.5 Commercial Applications

Company 1 was the original partner in the BDIFS project at the inception of the KTP. Since leaving the company the project has maintained contact with the company and an early version of BDIFS is used within the business. This design is set around the use of web servers to mirror data from the ERP systems in the companies various sites to present a global reporting mechanism. This is done by the translation and transformation in a specialized way at the central web servers, using largely standard technology based around the Linux, Apache, MySQL and PhP model.

Future plans have been created to incorporate third party suppliers into this mechanism and a business data integration mechanism for the manufacturing site of Company 1. This will entail deploying BDIFS clients at the sites of the organization and developing Web Services to communicate with their external customers who are putting integration pressure on the SME.

The development of these services should be generic to the make and version of the Enterprise Resource software that the SME has to talk to. Thus SMEs in similar circumstances can use the same services designed for Company 1. The eventual plan is the establishment of a large number of these services, making BDIFS a rich resource for SMEs worldwide. Rules associated with the message exchange can be applied by Company 1 using the criteria input from

the BDIFS portal. BDIFS is looking to the University in support for providing a translation service resource to SMEs linked to the University.

BDIFS is also looking to a partnership with another company in North Wales to exploit it workflow design tools. The ability for users to modify generic workflows to support their own specification appeals to a local company that provides specialized EAI solutions. At the moment the business has to define specific workflows per business they work with. Using the BDIFS workflow management system it will be possible for the customer themselves to simply modify generic workflows for the task using GUI based tools.

7.6 Summary

The eScience application of BDIFS is the first application of a Web Service framework that couples the ability of the initial BDIFS design for users to set and store workflow settings, with the ability for users to create, store and share translation algorithms. This application is set to be deployed and used in the Central Laser Facility at CCLRC. The Combination of Akogrimo with BDFIS and the use of the OpVO manager in BDIFS is the first time a dynamic VO infrastructure has been applied to support mobile services. The architecture of Akogrimo can be linked to BDIFS in order to allow the support mobile in a BDIFS application, although this investigation has highlights the need for such mobile grid use in BDIFS to have a more intricate and specialized workflow infrastructure.

Finally the original goal and main focus of the project, the deployment of BDIFS in a commercial application has been achieved in a SME in North Wales. This design has demonstrated that the framework can be used by users with little technical knowledge and that the software is able to function in the way this thesis has planned. However more large scale designs are planned for the project and the source code is due for release on SourceForge. It is hoped that interest can be generated in the project and the functionality of the eScience GUI tools can be added to make the framework more user centric. Overall BDIFS has the potential to remove the need for specialized commercial integration servers, specialised consultancy and other software development to aid eBusiness integration for SMEs. This can save the SME's and development agencies money with respect to these costs.

Chapter 8 Conclusions and Future Work

8.1 Conclusions

BDIFS is the first research project to practically address power control issues in real SME eBusiness partnerships within North Wales and beyond. The use of Project JXTA technology and Web services within the context of dynamic VOs is also a first. The use of the dynamic and BaseVO relationship demonstrated both in BDIFS and Akogrimo is the most significant contribution this thesis has made. Using these technologies the development of BDIFS is a solid platform to build a solution to these integration issues. By isolating the individual steps in the integration problems and presenting a mechanism where SMEs can collectively get behind a software solution this work has broken new ground.

BDIFS facilities provide a value-added mechanism for message exchange between two separate partner sites specific to the needs of the SME. This central feature is present in both the P2P and Web Service designs. The application of these frameworks to aid message exchanges as part of a business to business integration framework is a unique feature of BDIFS. The application of this design in the work has been done with the needs and capabilities of the SME's in mind.

Support for multi-vendor integration of various software platforms is achieved by the use of services provided by external service providers either in P2P or SOA environments. This achievement is enhanced in BDIFS with support for the sharing of user-designed services and workflows to aid business-to-business integration. The system also provides users the ability to create and share translations services and customise workflows through simple-to-use graphical user interfaces. The development and sharing of this technology is encouraged around the BDIFS portal, an approach to integration that has never been presented previously on a non vendor specific level.

The software is secure and maintains the privacy of its users. This is achieved by using WS security standards and techniques and with the use of JXTA security. The ease of use by staff with low technical abilities is enhanced in the framework by the peers and services that provide the functionality being maintained, supported and updated off-site. This is enhanced by the portal that is provided by BDIFS which is a neutral resource, and aims to increasing

SMEs influence on eBusiness integration technologies, and aid individual business in theirintegration challenge.

To support a valid business model in the software and encourage both use of the framework and collaboration within it, is a task allocated to the dynamic VOs. These VOs are for direct use in business integration partnerships, and includes services to support the creation of a business model where integration services can be supplied on a free or pay-per-use basis. This approach to integration breaks down the integration specialisation and, when compared to traditional EAI approaches, can significantly reduce the cost of development and licensing of such systems in SMEs.

From a North Wales perspective the BDIFS project has been the first study of this kind. The project has presented an investigation into the problems with current technologies to aid integration and why these are not well-suited to SMEs. Funding supplied by the EU to aid integration in economies like the one of North Wales, when spent on traditional approaches, may not be as effective as possible. In contrast to this, BDFIS focuses on the development of services to build a large non vendor-specific integration base for use by all SMEs in the area; this is clearly a more effective strategy. The development could also push for the skills and expertise in the development of these services to be based in the area where the funding is present, thus increasing its overall impact.

Outside the business application of BDIFS, the project has fed into the Akogrimo project and produced the Operative VO component for its demonstrators. This achievement has also been matched by the application of elements of BDIFS to aid eScience facilities integration at CCLRC. This has demonstrated the strength of the innovations in BDFIS and also flexibility of the software.

The future focus of the project is both technical- and business-oriented. We have really only begun to scratch the surface with regards to supplying viable data integration using the technologies presented in this work. There is real interest in the commercial domain that is surrounding the project is linked to the project's grounding in both industry and academia and will be the main focus of future development in the project. Although from a research perspective the dual investigation toward the end of the work with respect to eBusiness and eScience is an interesting domain.

With respect to supporting trends in industry and academia in this research area, the combination of integration and SOA development looks to continue in large scale research projects and more specific digital ecosystem designs. Within in industry there are emerging ASP solutions to aid SMEs who are starting on the ladder of business computing which have the potential to reduce the local legacy integration mess, although these still have to potential to lock businesses to one supplier. There does not seem to be the effort, ambition or vision to approach the challenge in the same manner as BDIFS.

This could be as the SOA business models have yet to reach maturity in both the industrial and open source / user community

In summary, BDIFS is a project that is before its time and has looked at uniquely combining emerging technologies, computing communities in an application domain that is ripe for change. Through further dissemination and partnership building it is the aim to start a catalyst in the North Wales computing and SME community that will rapidly spread the BDIFS approach to new communities and computing domains. The existing work with local SMEs and the branching out into the eScience community is the first step on this ladder.

8.2 Future Work

Although the initial BDIFS prototype is now finished there are plenty of avenues leading on from this effort that are worthy of exploitation. These ideas will be refined following more investigation and commercial use of the system at Company A and CCLRC. In order to document current issues with the system and future investigation, the future work in the project is now discussed.

8.2.1 Service Development

The BDIFS project started off with the key aim of integrating data between business partners. The two designs do this in two different ways and therefore target different audiences. The initial P2P design of BDIFS was focused on being easy to implement and use. Although lightweight technically, the system presents a means by which a group of SMEs can communicate with each other and exchange eBusiness specific data in an organised and secure way.

The Web Service application was created in order to interface SMEs with more complex integration systems and to remove the reliance of a BDIFS client being present on the other end of a messaging exchange. The BDIFS Web Services system has support for the use of Web Services within its framework. These services are invoked by the service agents created in the BDIFS Operative VO. These services invoked by the OpVO could be provided by third parties or the BDIFS developers to interface with complex ERP type systems such as SAP.

Interfacing with the more complex / costly computing frameworks would be a potentially invaluable use of BDIFS. To achieve this the interfaces of the target services need to be exposed to the BDIFS system. For this to take place there needs to be development of services capable of communicating

with the more popular back end systems. Conversely, the types of return data to the user initiating the integration request need to be expanded too. The business model that can be developed for charging of service use is a future way of achieving collaboration in the framework in order to rapidly develop integration services. This can be combined with the user-driven workflow creation described in chapter 6 and with the translation service creation described in chapter 8.

The project needs to look for investment and potential partnerships with larger businesses, whilst at the same time encouraging community support to aid service development. Part of the way to achieve this is by developing a business plan for the framework where the charging and pricing can be put into place to encourage use and development. To aid this there is a business plan being developed, a summary of which is in Appendix 2. It is also envisaged that the source code and the project will be made available to the open source community.

8.2.2 Agent Development

Despite what may seem a giant leap into the complexity of integrating with more complex eBusiness systems, the simple needs of SMEs community reflected in the P2P design still remain. The use of a client by Partner A to communicate with Partner B in a P2P framework is a simple process, but in the Web Service framework in order for Partner B to communicate to Partner A, one partner needs to present a web service. This is a process that is much

more resource consuming and complex than in the P2P design when a similar Java client to Partner A is downloaded.

Thus the use of the BDFIS framework by SME-to-SME may still need the P2P message exchange functionality within it. It is therefore proposed that the Java clients downloaded from the Portal have dual functionality, with the function chosen by the user when the client is used. For example, if Partner A wants to communicate with the SAP server of Partner C, they can select the appropriate workflow in the Web Service design and use the client to invoke the VO in order to run this request. But if they want to communicate with Partner B who only has P2P functionality, the Java Client can be switched to achieve this simple communication.

The development of a set of agents with dual functionality will make BDIFS more flexible and usable by a wider range of users. Academically it will present the possibility of further investigation into where the Service-oriented Architecture is less suited to tasks than a P2P framework. There may even be larger businesses that choose to use the P2P approach due to issues such as security, as the P2P approach is of a simpler and conceptually easier architecture to secure.

8.2.3 Customer Requirements

Industry tests are expected to reveal further areas of development needed in terms of the functionality of the system to meet business needs. For example, the SLA infrastructure in BDIFS needs to be developed to reflect a closer representation of business goals of the SME's. This can be achieved by more consultation and testing with SMEs and adoption of standards such as WS-Negotiation to aid SLA development during application execution.

Other advances in SLA could include developments associated with the penalties for the breaking of SLAs in the system, and also what type of behaviour they cover. Internally in the architecture the use of SLAs could be improved with greater monitoring and standards used in the expression and use of the SLA data.

8.2.4 Security

Like SLAs in the BDIFS Web Service Framework, the system has been secured in a laboratory environment. The use of non-standard identity tokens can be improved with the use of SAML or a similar technology for their

expression. The tokens could also be encrypted using public and private keys and signed certificates introduced. The information in these tokens could also be enhanced to include more information such as policy and SLA information.

In order to automate the Service Provider domain and Base VO trust relationship, the development of trusted third-party servers may be introduced. These could provide a trusted service for the exchange and validation of tokens when new Service Providers join BDIFS. A trusted third-party could also take on the role of the accounting in the system, to ensure fair billing and charging.

8.2.5 Other

In addition to the points above, it would be good to include more WS standards in BDIFS as they emerge. At the moment the main use of the WS standards in the project is the use of the standards represented in the WSRF framework and the creation of multiple service instances.

The system also needs a good set of stress tests to be applied to determine if it can withstand constant commercial use. To date the design is on personal computers and several runs of the use cases have been as far as the stress testing has gone. Within the P2P design testing was minimal, and the design was basically a proof of concept. Therefore if the use of the P2P network in

BDIFS is going to be used, more testing of it will be needed. This is particularly true in the case of user request validation. To date there is little flexibility in the system. The ability for userID's to be cached to speed up sign on is quite basic and is not likely to scale well. Also there is no provision for users who change their mind about the execution mid workflow, at the moment the only option is to stop and start again. It would be a better model if the workflow could be updated in runtime, this would involve linking such a decision also to the policy infrastructure of the VO.

Finally, the integration of mobile services should be implemented in the system by combining it with an Akogrimo testbed. Upon the testing of this type of application support, future work will be found looking into viable application use. A new business plan may be needed for the support of the task.

8.2.6 Design Technology

The design experience has highlighted the fragility of the Globus Toolkit version 4 in its support of the WSRF standards. This can be linked to the use of the Apache Axis Web server, but it still is a factor that may affect the future development of BDIFS. More research needs to be carried out into the support of other Web service enabling servers. To date BDIFS has a workaround for problems it has encountered with services designed using WSRF.net and other Web Services running on Tomcat.

APPENDIX 1

1.1 Company A

The work in this work originated from a real set of scenarios encountered and still being experienced by SMEs in the Gwynedd area and beyond. Company A is such an SME, classified as an SME based on the number of people that it employs. However this ranking is not an accurate indicator of the use of ICT within SMEs. As the business integration needs of an SME may vary widely between instances, Company A is as a less advanced SME in terms of ICT.

The main business of Company A was warehousing, small scale manufacturing, product assembly and packing in a sterile environment. The environment in North Wales of low wages and cheap land proved a good environment for the operation, and the company has contracts from both national and international customers. Due to the contractual nature of the work the business partnerships that Company A have are often seasonal or short term contracts.

The state of ICT in Welsh SMEs will be explored in greater detail later on. However the compounding factors which make companies like Company A particularly disadvantaged (in a business integration environment) are as common in North Wales as in other disadvantaged economies in Europe and the rest of the world. Some of the compounding factors influencing the disadvantaged nature of Company A and companies like it are listed below.

- No professional dedicated on-site IT support.

- Accounting, warehousing ERP type system, both proprietary and supported by a third-party.
- Reactive IT strategy.
- Business based on many short and medium term contracts.
- Software in second language.

Within Company A, the lack of on-site IT support extended to support on the systems which held the company's data, thus producing a situation where the technical control of Company A's business systems was outside of SMEs. This is compounded by the short term contract nature of the business and the impact of this on the integration needs of the company. The integration pressures placed on the business by customers consume many of Company A's resources, leading to a reactive IT policy, unable to plan and invest to control this situation in the future. This is an issue in many SMEs where IT is only thought about when it is needed to make a change or a crisis or system malfunction occurs. Finally in the case of Company A the majority of staff's first language is Welsh and the lack of internationalization of some systems hampered some of the occasional users of the IT system.

At the core of Company A's IT systems at the start of the project was an Enterprise Resource Planning system that controlled the company's financial reporting and recording along with manufacturing and warehousing operations. The initial scope of the project was to support the company's IT systems using open source software and develop an e-commerce side to the company using the same technology. These measures were requested by higher level management to increase company revenue and improve the IT structure at low costs. However after the establishment of an e-commerce site

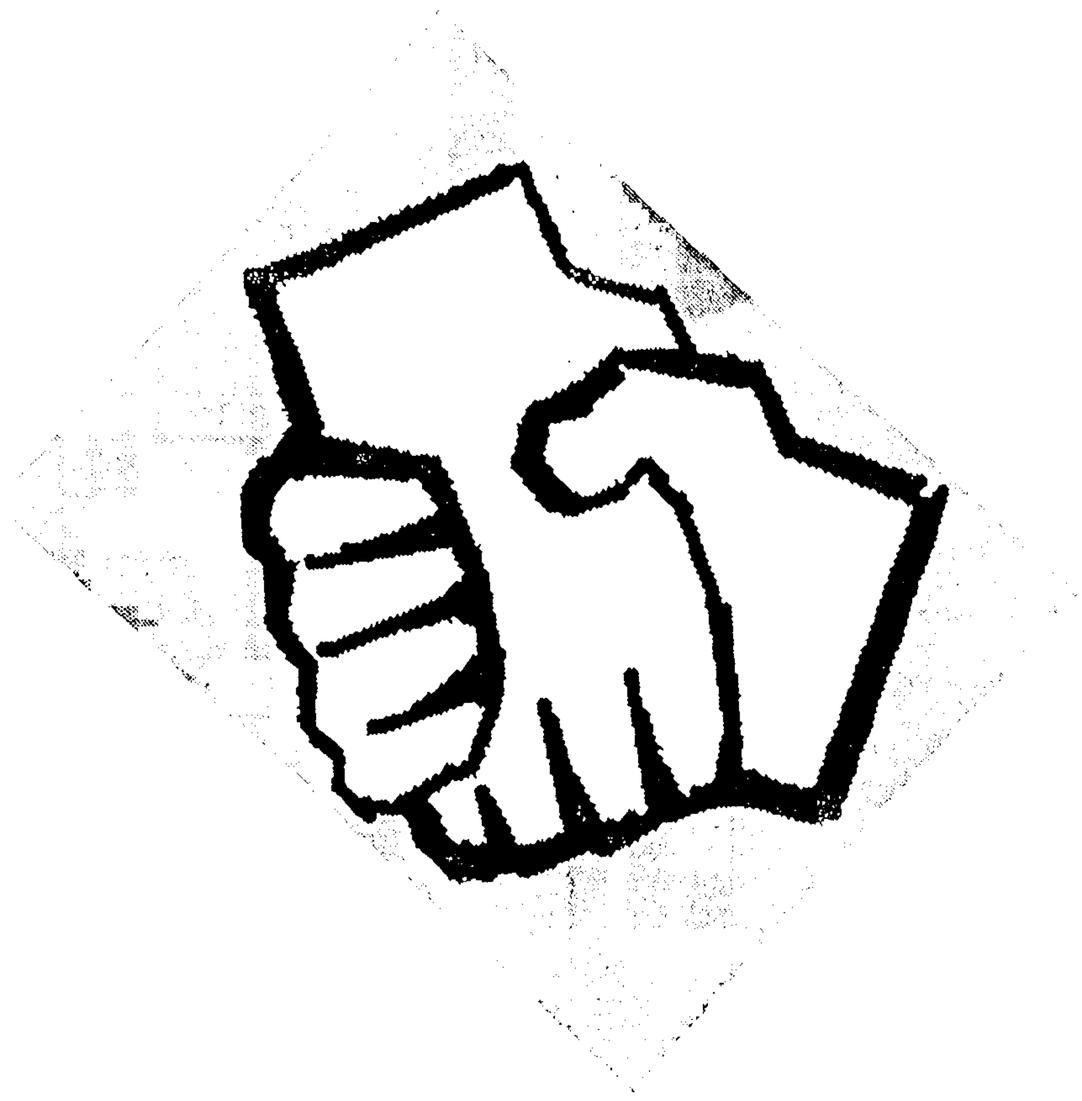
for customer sales and the development of open source fileservers, e-mail servers and corporate data repository, it was obvious that a solution to aid the simple automation of the company's supply chain was key to improving the company's IT wellbeing.

The target for this automation was field sales orders and this was achieved with a sales force reporting and ordering system using the open source backbone technologies of Linux Apache MySQL and PhP (LAMP). This project was to form an MPhil based on research within the company. However the company was only papering over the cracks and the integration of the remote sales order system revealed the need for measures to be made in the MIS to supplier or customer automation process. Thus an alternative project was needed to aid integration between Company A and its business partners. Furthermore, many of the needs of external partners in terms of securing the contract were related to access to data at Company A. This common pressure on SMEs formed the main motivation behind the application developed in this work.

However the difficulty in achieving integration between Company A and its partners remained that of integrating data from legacy application software. This is the heart of the business integration challenge and what the solution documented in this work aims to solve. Thus the applications illustrated in this paper are ultimately motivated and designed for testing within Company A.

Appendix 2

the business data integration Grid



Enabling E-Business for the SME

Business Plan 15-04-2007

t.kirkham@rl.ac.uk

Third Party material excluded from digitised copy.
Please refer to original text to see this material.