Trusted and QoS Aware Provision of Application Services
(TAPAS)

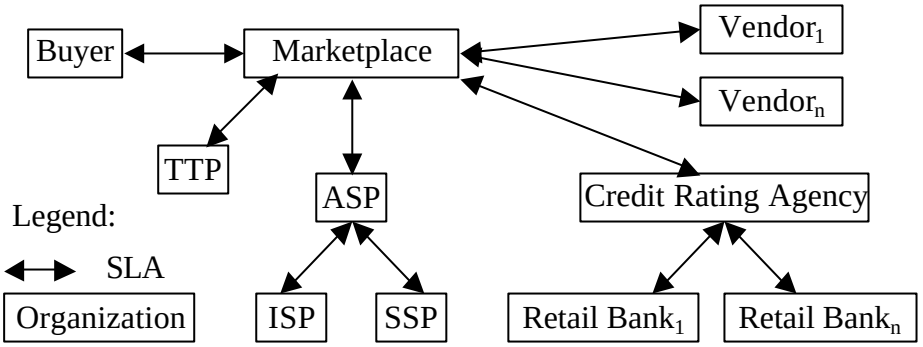
University of Newcastle, UK
University College London, UK
University of Cambridge, UK
University of Bologna, I
Adesso GmbH, D

It is well known that organisations are increasingly focusing on their core businesses and streamlining their operations by 'outsourcing' non-core businesses to external organisations. In particular, many organisations find it cost effective to outsource their IT applications to *Application Service Providers* (ASPs). An ASP typically uses middleware and component technologies for deploying, hosting and managing applications of an organization from a centrally managed facility. However, as organisations become global and distributed, such centrally managed hosting solutions will need to be replaced by multi-site, distributed hosting solutions. As we argue below, many research problems in enterprise distributed computing will need to be solved before such hosting solutions can be realised.

The TAPAS partners believe that an ASP will increasingly be called upon to host distributed applications that make use of a wide variety of Internet services provided by different organisations. This naturally leads to the ASP acting as an intermediary for interactions for information sharing that cross organisational boundaries. However, despite the requirement to share information and services, autonomy and privacy requirements of organisations must not be compromised. Organisation will therefore require their interactions with other organisations to be strictly controlled and policed. This creates two major challenges. Firstly, contractual relationships between multiple organisations for information access and sharing will need to be governed by *service level agreements* (SLAs), which will need to be defined and agreed between the organisations and then enforced and monitored by the ASP. Secondly, the ASP will have to establish appropriate *trust relationships* with the organisations and implement corresponding security policies before organisations will permit the ASP to act as an intermediary for inter-organisational service invocations. Unfortunately, middleware services for inter-organisational interactions as outlined above do not yet exist; indeed, development of such services is very much a research problem. Thus ASPs currently lack tools and techniques for offering hosting facilities for advanced Internet based applications. We use a simple example to focus on application hosting issues and limitations of current technologies.

Our example considers the hosting of a Marketplace application that matches buyers and vendors. We might imagine that the Marketplace provides services through a component that manages requests for proposals (RFP). This component might use a service from a credit rating agency in order to obtain a credit rating of the buyer that is then forwarded alongside the RFP to vendors. The credit rating agency itself may implement its services using data obtained from account history services provided by retail banks. The Marketplace may need the services of one or more trusted third parties (TTPs) to meet the security requirements of buyers and vendors. We assume that the ASP hosting the Marketplace is obtaining its communication and storage resources from an ISP (Internet Service Provider) and an SSP (Storage Service

Provider) respectively. The figure below shows the various organisations and the corresponding SLAs involved. The company providing the Marketplace has set up SLAs with buyers and vendors etc. and to meet the obligations implied by these SLAs, it will have a comprehensive SLA with the ASP specifying overall processing, storage and communication requirements, as well as availability responsiveness and security requirements.



In effect, to host the Marketplace, the ASP will require a distributed execution environment with a number of core services capable of meeting specific non-functional requirements of fault tolerance, availability, security, and timeliness; we will refer to these as *QoS enabled services* (QoS: Quality of Service). State-of-the-art application services are developed using component-based technologies, such as those provided by the Java 2 Enterprise Edition (J2EE), Microsoft's .NET or the Object Management Group's CORBA Component Model. These technologies support the cost effective creation of services by composing and integrating existing components. To date components can be obtained from component vendors through component catalogues and be deployed in-house. In the future, as our example illustrates, components that implement services will be hosted by vendors or dedicated service providers in application servers and will be invoked from components from other organisations across public networks. Current component technologies support the specification of functional component interfaces. They, however, do not adequately support the definition of the non-functional characteristics of component execution.

With the above observations in mind, the overall objective of the TAPAS project is to develop novel methods, tools, algorithms and protocols that support the construction and provisioning of Internet application services. We will achieve the overall objective by developing QoS enabled middleware services that will enable components to be deployed and interact across organisational boundaries. We have divided the overall objective into three broad categories, which we describe below together with our approach to achieving them.

Objectives related to SLA specification, Service Composition and Analysis Techniques: The TAPAS project will develop notations for expressing SLAs to enable specification of QoS, such as the availability, performance and scalability characteristics of components, as well as trust relationships. Model checking capabilities will be developed to support reasoning about QoS characteristics of components and their composition. This will support the ASP in assessing the qualities of service perceived by end-users prior to developing and deploying an application service. In our example, the composition of complex application services from more elementary services provided by different organizations will thus be governed by multiple SLAs. The project intends to adopt UML as the language for the

description, modelling and analysis and extend it with formally defined stereotypes and properties. This approach is motivated by the large number of users of the UML in industry, who will find it easier to adopt a small UML extension than an entirely new specification language.

Objectives related to Trusted and QoS-aware Services for Application Hosting: Component oriented middleware promotes the use of *containers* to host component instances. Containers are responsible for using the underlying middleware services for communication, persistence, transactions, security and so forth. TAPAS will develop support architectures that provide QoS negotiation, establishment, and adaptation facilities to such services; these will be used by containers to make them QoS enabled. Particular attention will be placed on the development of QoS enabled multi-party communication (e.g., for supporting publish/subscribe communication, dynamic load balancing between replicated containers) that cross organisational boundaries. To this end, SLA trust specifications will be used for deriving service invocation primitives enriched with authentication, non-repudiation mechanisms, with or without the involvement of TTPs.

Objectives related to assessment: The results from the TAPAS projects, in the form of methods, tools and techniques for design and development of component oriented middleware for provisioning of Internet application services will be evaluated by comparing the results with current state-of-the-practice (e.g. an off-the-shelf CORBA or J2EE application server). The TAPAS consortium includes an ASP, who will provide requirements for QoS enabled application service provision, and case studies. Partners will build demonstrator applications, such as hosting of an auction service to demonstrate the effectiveness of TAPAS results.

Deliverables and exploitation: The deliverables of the project by means of which we provide measurable evidences of the extent to which we succeed in meeting the above stated very challenging objectives will include prototype implementations using open source application servers and demonstrators as well as technical reports and papers. The inclusion of an ASP in the consortium will speed up the migration of TAPAS results and technologies to industry. In addition, the project will form an Industry Advisory Board, whose membership will represent a cross-section of technology providers and end-users; the Board will help us in revising, where necessary, the objectives of the project. Thus, even though the TAPAS project has been formed to address the needs of ASPs, the results of the project will be of interest to much wider scientific and industrial communities.

The TAPAS consortium brings together academic researchers from software engineering, middleware and Internet communities to work with an ASP. Success in this endeavour will substantially ease the development of domain specific application services; for example, services for *computational grid* (enabling scientists to move large amounts of data globally and perform long lasting parallel scientific computations), *electronic markets* (enabling art dealers to conduct real-time auctions on a world-wide scale), *collaborative applications* (enabling globally scalable multi-person games) and a variety of other Web Services.