IST-2001-32133
GridLab - A Grid Application Toolkit and Testbed

D3.2 Grid-Aware Triana Prototype

Author(s):	Ian Taylor, Matt Shields, Ian Wang, Roger Philp and Shalil Majithia

Document Filename:	GridLab-3-D3_2-0001-TrianaPrototype
Work package:	WP3 Work-Flow Application Toolkit (TGAT)
Partner(s):	University of Wales, Cardiff
Lead Partner:	University of Wales, Cardiff
Config ID:	GridLab-3-D3_2-0001-1.0
Document classification:	INTERNAL

Abstract: This document outlines the main features of the first Grid-Aware working Triana prototype

Contents

1 Introduction

Triana [1] provides a visual programming interface with functionality represented by units connected by data flow pipes. Conceptually, Triana is split into three components that provide the GUI (controller), then local and distributed remote services. The Triana controller provides the interface to the command server process that acts as a client service to locally and distributed server services. Groups of Triana units may be distributed across the Grid. Figure 1 shows a simple spectral analysis network with a schematic of the relationship between the user view within the Triana controller and its distribution amongst the Triana services.

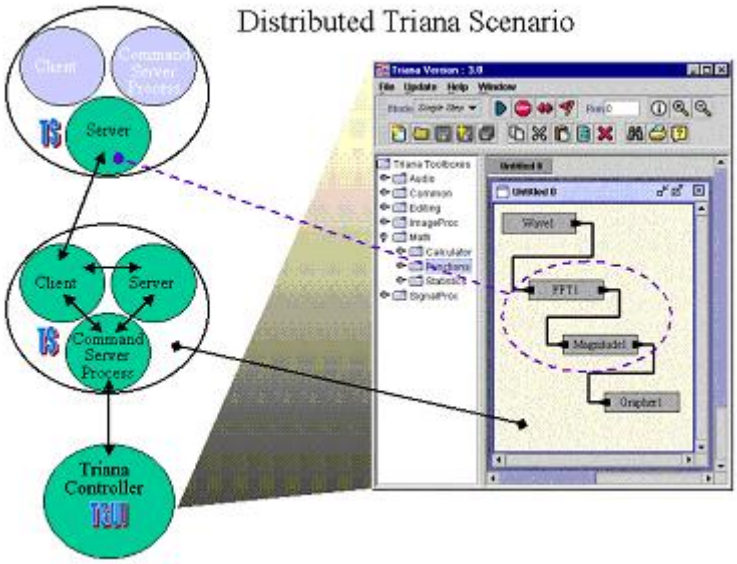

Figure 1: Distributed Triana Scenario

2 JXTA Implementation

The first grid-aware working prototype of Triana uses JXTA [2] via a high-level application API called JXTAServe. This prototype was designed to work within the Consumer Grid scenario [3]. JXTAServe is used directly at present but it has been designed so that we can incorporate it into GridLab as soon as we negotiate the GAT API. The architecture for such a model is given in Figure 2.

The Triana user interface creates a WSFL-like taskgraph which is sent to a Triana service for execution. Results from this execution are then sent back to the user.

We can look at JXTA P2P networking with respect to Triana and distributing Triana task graphs at varying levels of granularity. At the lowest level, each Triana Unit (or Group Unit) is itself a peer communicating with other units/peers via some mechanism, using JXTA as an implementation this would be a JXTA pipe. The units or peers would organise in peer groups with communication to the Triana client peer and collaboration and communication with other peer groups of units being controlled by a Triana Server peer. However, this model has implementation overheads due to the level of granularity; each unit communicates via the underlying P2P communication mechanism which could be implemented on top of many different underlying layers. Our approach therefore is based around a Triana Server peer. This peer is

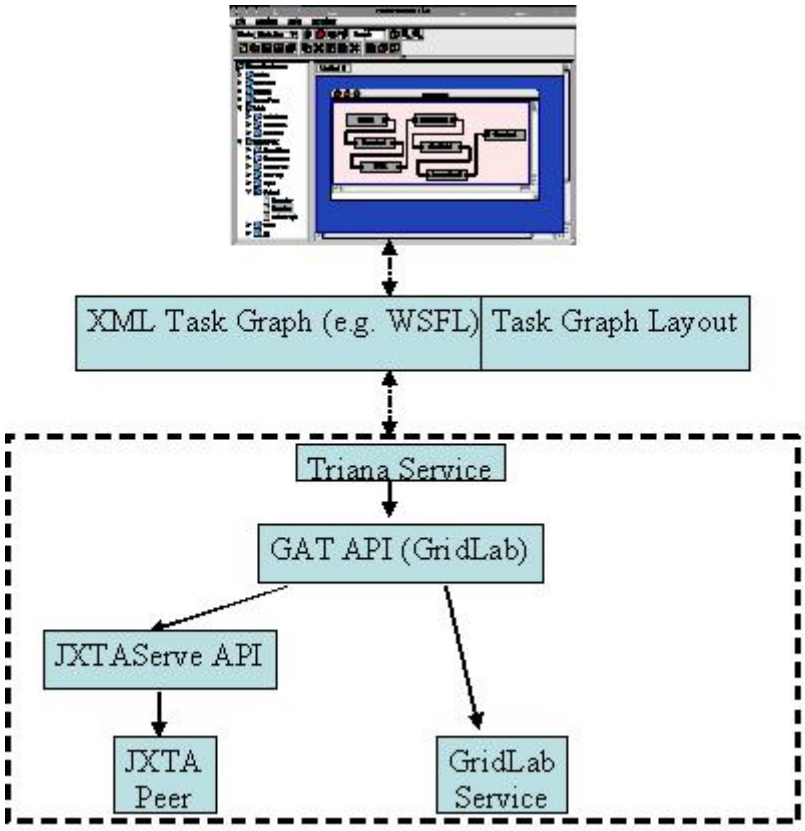

Figure 2: The architecture of JXTAServe and GridLab for the the distributed Triana prototype

capable of running a Triana task graph or sub-section of a task graph. This peer would also act as a rendezvous or relay peer able to forward messages, other sub-sections of a task graph or control instructions onto other Triana Server peers. Communication is more course grained than the previous model, communication between units within a server are internal to that server. Triana services discover each other via Triana-service advertisements and communicate via JXTA pipes.

At the time of writing, we have integrated Triana with JXTAServe and it is capable of distributing any group (i.e. compound) unit onto a consumer grid. We have tested the system and it can not only achieve the distributed scenario outlined in Figure [**?**] but also distribute one taskgraph amongst several servers e.g. machine 1 can send data to machine 2 which can send data to machine 3 which can then return the data back to machine 1 for displaying. We will be demonstrating this capability live in GGF-5 (July 2002) in the Service Management Framework/Jini session.

References

[1] `http://www.triana.co.uk`

[2] `http://www.jxta.org`

[3] Taylor I.J., Philp R., Rana O., Schutz B.F. & Shields, M. 2002. The Consumer Grid. Global Grid Forum, Feb, 2002, Toronto, Canada.