



CONTRACT NUMBER 508830 DEISA DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR SUPERCOMPUTING APPLICATIONS

European Community Sixth Framework Programme

RESEARCH INFRASTRUCTURES Integrated Infrastructure Initiative

Final Report for JRA2

Deliverable ID: D-JRA2-6.2 Due date: October, 31, 2006 Actual delivery date: November, 1, 2006 Lead contractor for this deliverable: EPCC, University of Edinburgh, UK

Project start date : May 1st, 2004 Duration: 4 years

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)				
Dissemination Level				
PU	Public	Х		
PP	Restricted to other programme participants (including the Commission Services)			
RE	Restricted to a group specified by the consortium (including the Commission			
00	Confidential, only for members of the consortium (including the Commission			
00	Services)			

Project and Deliverable Information Sheet

	Project Ref. №:	508830)	
DEISA Project	Project Title:	: Distributed European Infrastructure for		
	Supercomputing Applications		computing Applications	
	Project Web Site:	ite: http://www.deisa.org		
	Deliverable ID:	D–JRA2-6.2		
	Deliverable Nature:	e: Report		
	Deliverable Level:		Contractual Date of Delivery:	
	PU		31 / October / 2006	
			Actual Date of Delivery:	
			2 / November / 2006	
	EC Project Officer:			

* - The dissemination level are indicated as follows: PU – Public, PP – Restricted to other participants (including the Commission Services), RE – Restricted to a group specified by the consortium (including the Commission Services). CO – Confidential, only for members of the consortium (including the Commission Services).

Document Control Sheet

Document	Title: Final Report for JR	A2
	ID: D-JRA2-6.2	
	Version: 1.0	Status: Draft
	Available at: http://www.o	deisa.org/
	Software Tool: Microsoft W	ord 2002
	File(s): D-JRA2-6.2	.doc
	Written by:	Gavin Pringle
Authorship	Contributors:	Mario Antonioletti
	Reviewed by:	Dietmar Erwin
	Approved by:	DEC

Document Status Sheet

Version	Date	Status	Comments
0.1	October, 31, 2006	Draft	
0.2	November, 1, 2006	Draft	
1.0	November, 2, 2006.	Draft	
1.1	November, 17, 2006.	Final version	

Project and Deliverable Information Sheet, Document Control Sheet, Document Status Sheet and Document Keywords and Abstract are internal handling information, which will be suppressed from the document prior to the submission of the deliverable to the EU.

Document Keywords and Abstract

Keywords:	Final report for JRA2, GADGET, portable binary data formats, VO, FLASH, Code migration
Abstract:	This short deliverable contains an overview of the work done as part of JRA2. JRA2 was a 2 year project, jointly funded by DEISA and VirtU. All deliverables were completed and submitted on time.
	The work by JRA2 will benefit the cosmologists of the Virgo Consortium as a number of their production software suites have now been ported and optimised on the DEISA infrastructure. This will allow the consortium to perform more precise simulations of the evolution of the Universe than ever before. This has been made possible due to the work done by EPCC to port and tune their codes, and by the very nature of the DEISA infrastructure.
	EPCC ported an optimised version of GADGET-2 on to two DEISA platforms and investigated its effectiveness in a metacomputing environment. The code was found to be unsuitable to operate in a metacomputing mode when using the installed version of PACX-MPI. In the future, the improved network and co-scheduling may go some way to addressing some of the performance issues, however, metacomputing is no longer part of DEISA.
	All the associated tools were parallelised, where necessary, and ported to two of the IBM platforms. An investigation into various portably binary data formats demonstrated that HDF5 was fastest of the formats considered. Each tool was adapted, where necessary, so that they can now read and write using the HDF5 format. HDF5 interoperates seamlessly with VOTable, the data format of choice for Virtual Observatories.
	EPCC, with Virgo, have ported a version of FLASH, using a particular dark matter simulation as initial conditions for testing purposes, with HDF5 I/O, to several of the DEISA platforms. The code has been profiled and optimised, and now runs around 6 times faster. For one particular configuration, the speed up is almost 9-fold. Code migration has also been introduced, allowing for long simulations to run to completion, seamlessly utilising several of the DEISA platforms if required.

Project and Deliverable Information Sheet; Document Control Sheet; Document Status Sheet; and Document Keywords and Abreact are internal handling information, which will be suppressed from the document prior to the submission of the deliverable to the EU.

Table of Content

Table c	of Content	1
1 Int	troduction	1
2 WF	P 1: Grid-enablement of GADGET for metacomputing environments:	1
3 WF	P 2: Development of a Grid-enabled toolset	3
4 WF	P 3: Grid-enablement of FLASH through platform-independent optimisations	;
and co	de migration:	1
5 WF	P 4: Distributed data management and information extraction, and WP 5:	
Grid-er	nabled visualisation:	3
6 WF	P 6: Project management	3
7 JR	A2 Presentations	7
8 JR	A2 Publications	7
9 Lis	st of deliverables	7

1 Introduction

The objectives of this activity were two-fold:

- To avail the Virgo consortium, Europe's leading consortium in Cosmology, with the most advanced features of Grid computing by porting their production applications to effectively use the DEISA infrastructure;
- To lay the foundations for a Theoretical Virtual Observatory (TVO) complementing the approach of the Global Virtual Observatory.

To perform the work of JRA2, EPCC worked in close partnership with the Virgo Consortium. All of this work was only progressed after gathering clear and detailed user requirements for each workpackage. These user requirements were then published as public DEISA deliverables, as listed below.

There was a slight modification to the planned start date, partly due to the change in DEISA's own start date. JRA2 technical work was originally planned to begin at DEISA Project Month 7 (PM 7), which would have been November 2004. The start date was moved to DEISA PM 6 (October 2004) so that it coincided with the start of the matching funding from the associated VirtU project. Deliverable dates have been adjusted slightly from the original Technical Annex (TA) to complement the dates of the other Joint Research Activities' (JRAs) deliverables. This had no impact on the actual work being done; it was purely a reporting issue.

The DEISA Technical Annex describes the six work packages in JRA2 and this document gives a brief summary of the work achieved.

2 WP 1: Grid-enablement of GADGET for metacomputing environments:

This workpackage is split into two parts:

- 1. Replacing the generic MPI environment with a Grid-aware MPI implementation (e.g., PACX-MPI, MPICH-G2 or similar); and
- 2. Developing adaptive algorithms to make the application environment aware, i.e., typically to reduce the impact of the high latency between systems by adapting key algorithms to the topology of the distributed system. These run-

time optimisations will ensure an optimal performance for any given metacomputing environment¹.

The GADGET code was replaced by GADGET-2 for WP1, as GADGET-2 was found to be more robust and could model more physical processes.

GADGET-2 was profiled on HPCx and a bug in IBM's qsort was located, reported and circumvented.

With a view to testing GADGET-2 in a metacomputing environment, we set-up a small, two-node Grid built out of two heterogeneous Sun multi-processor platforms (E3500 and E15000) with PACX-MPI. GADGET-2 was run over this small Grid using PACX-MPI, and it run well. This Sun Grid consisted of one small, "slow" machine, and one large, "fast" machine. Within GADGET-2, each particle has a notion of work associated with it, which GADGET-2 then uses to load-balances its workload. In order to take into account the distribution of the application over different heterogeneous platforms the work associated with particles was extended to take this into account. In addition to the existing mechanism, we have introduced actual execution times to this measure of work. Thus, when the code performs its load balancing, it takes into account the less powerful platform to the more powerful one, thus balancing the work-load over this small heterogeneous test Grid. Our first initial implementations of this self tuning were found to be unstable, and further work to stabilise this method is required.

This small Grid was successfully extended to include HPCx on which jobs were successfully run, but the performance of this extended Grid was not investigated.

This work was presented at the Sun HPC meeting in Heidelberg, 20/21 June, 2005.

We successfully employed the platforms at IDRIS and RZG, running a gas-only simulation using GADGET-2, employing PACX-MPI across both systems.

We then introduced the load-balancing code running both Dark Matter only simulations and Dark Matter plus Gas simulations, using the platforms at IDRIS and RZG. Once this had been done, we attempted to mask the latency incurred by running over multiple nodes.

Basically, we found that GADGET-2 did not perform well when using the DEISA infrastructure in a metacomputing fashion, i.e. with the code running between two different machines in the DEISA infrastructure as one application. This was due to a number of reasons:

- The PACX-MPI library employed was not the latest version available and, theoretically, would not perform as well. However, the more recent version proved to contain bugs and was unusable;
- The DEISA infrastructure did not then have co-scheduling, which meant that the sites had to co-schedule 'by hand' with the co-operation of the local system administrations, which greatly increased the time it took for tests to
- run: The code itself, GADGET-2, is not as loosely coupled as first thought and is probably not a good candidate code for running in a metacomputing environment;
- GADGET-2's own load-balancing routines proved to be include a great deal of communication, which slowed the code down immensely during the initialisation phase. Moreover, the new load-balancing routine which took the

¹ In August, 2005, the DEISA Executive Committee decided that the DEISA infrastructure would not be used for metacomputing

distribution of the algorithm over heterogeneous platforms that we introduced proved to be unstable.

If GADGET-2 is ever to be used over the DEISA infrastructure in a metacomputing environment, and this applies to any other similar application, we suggest using the upcoming release of MPICH-G4 used over DEISA's future 10 GB/s network might give an acceptable performance. In addition it is essential that co-scheduling be supported else such applications will not be able to operate in a production mode. It should be noted however that, towards the very end of WP1, the DEISA Executive Committee decided that the DEISA infrastructure would not be used for metacomputing, thus one of the objectives of this work package became somewhat academic.

3 WP 2: Development of a Grid-enabled toolset

A number of serial pre- and post-processing tools have been developed for GADGET and Hydra-MPI. Grid-enabled versions of these tools were developed to cater for distributed datasets in heterogeneous environments, hence laying the foundations for WP 4. These tools will be able to take advantage of the DEISA global filesystem (see SA2) as well as operating within a "conventional" distributed environment for interoperability outside of the DEISA framework. More specifically this work had three objectives:

- Grid-enablement of group finders: we proposed to develop and Gridenable a series of tools known as "group finders". Their purpose is to identify gravitationally bound structures in the dark matter. A code called Subfind has already been developed by Volker Springel to identify bound structures (or substructures) which are located inside larger bound structures. The work undertaken as part of this WP would enable users to apply the Subfind code to large distributed datasets in order to identify dark matter halos and their substructure. Once found, a number of existing analysis tools could be applied to calculate properties of these
- 2. Parallelisation and Grid-enablement of codes for making halo merger trees: the primary objective was to build structures called merger trees for each halo in the final output of the simulation. This task required us to find all the groups at all times. The Virgo consortium has codes which allows one to determine for a given halo at a given time to track which halos in the past merged together to form it. Having found the merger trees for all halos one can then use these trees to model galaxy formation. The codes for building merger trees already existed but, in general, were serial codes, and were parallelised and were extended to utilised HDF5, where necessary, and then ported to the DEISA infrastructure.
- 3. Development and Grid-enablement of tools for evaluating basic properties of the dark matter (e.g., density field, correlation functions, power spectra).

The public DEISA-JRA2-D2.1 'specifications' document, which was submitted to the commission ahead of schedule, describes the work which was to be attempted during the remaining months of WP2. DEISA-JRA2-D2.1 included porting "FoF", another so-called Group-Finder code akin to SubFind, which is not referenced in the DEISA Technical Annex.

All the tools described in DEISA-JRA2-D2.1 have been delivered and installed on HPCx and IDRIS.

An investigation into the performance of different portable, binary data formats, namely HDF5, BinX, FITS, VOTable and DFDL, was carried out. This investigation was performed in order to determine which data format performed the best and would then be used to Grid-enable the toolset. By Grid-enable we mean, in this

context, to ensure that the tools can be used within DEISA's GPFS-MC and in a heterogeneous environment. Employing such a format allows codes to run over a heterogeneous Grid, such as DEISA. Based on these results, VirtU and Virgo both decided to adopt HDF5 as the portable format from now on. This is because the I/O rates were so much faster than all the other data formats, including the default vendor specific I/O. Furthermore, HDF5 can interface with seamlessly with VOTable, which has become the data format of choice for Virtual Observatories. This investigation into portable binary data formats was partially carried out by an MSc student at EPCC who not funded by DEISA. The results of this investigation are reported in "Comparison of Portable Binary Data Formats within a Cosmological Simulation", Wei, W., M.Sc. Dissertation, The University of Edinburgh, 2005, and key results referenced in D-JRA2.1 for completeness.

4 WP 3: Grid-enablement of FLASH through platformindependent optimisations and code migration:

The FLASH code, as modified by the Virgo Consortium, is to be employed to simulate the evolution of the universe. FLASH is written in FORTRAN90 and MPI and employs HDF5. The use of HDF5 ensures that binary data files are portable across a heterogeneous DEISA infrastructure. FLASH is used under license and cannot be redistributed, otherwise it is freely available. The software used in this work was in part developed by the DOE-supported ASC / Alliance Center for Astrophysical Thermonuclear Flashes at the University of Chicago.

This task necessitated a number of modifications to the existing code:

- Optimise the use of MPI use within the dark matter calculation of FLASH, to ensure communications scale to large numbers of processors to obtain a fast performance across heterogeneous platforms and topologies;
- Replace the iterative multigrid dark matter Poisson solver with a noniterative FFT-based version, and ensure that this routine scales to large numbers of processors to ensure fast performance across heterogeneous platforms and topologies;
- The development of additional routines to induce and perform code migration;
- Consider the introduction of multi-time-stepping, to allow for more clustered simulations to be carried out. Any new multi-time-stepping code was to be produced as collaboration between EPCC and Virgo members.

The plan was to port the FLASH code, as modified by the Virgo Consortium, to a number of different DEISA architectures and to perform some basic profiling, which would indicate which routines were computationally expensive, irrespective of which platform the code was running on. The results of this investigation forms the content of D-JRA2-3.1

The gathering of the requirements for D-JRA2-3.1 was straightforward. However, porting the code proved to be problematic, as, within the first few months, the Virgo Consortium changed their modified version of FLASH a number of times. Thus, once we had ported their code to a number of DEISA platforms, we were instructed to replace that code with another version which was either due to a bug fix, a new feature which Virgo wanted to be included, or a different version of FLASH itself, plus porting modifications were not re-introduced into the original code base. This made porting a difficult and time consuming process. This process was made more difficult as the process of ensuring the port was correct which, itself, was not initially straightforward.

However, a definitive version of the FLASH algorithm and a definitive simulation were finally decided upon and the associated code has now been ported and profiled on the IBM clusters at EPCC (HPCx), IDRIS, RZG, the IBM Linux cluster (Mare Nostrum) at BSC.

This code was also ported and profiled on the NEC SX-8 at HLRS by an MSc student at EPCC, however, this work was not funded by DEISA. The results of this investigation is reported in "Porting, Profiling and Optimising an AMR Cosmology Simulation", Radoswlaw Hubert Ostrowski, MSc in High Performance Computing, The University of Edinburgh, 2006, and key results are referenced in D-JRA2-3.3 for

completeness. The process of porting the simulation on the SGI at SARA was not completed as, after extensive investigations hampered by an unstable code base and a restrictive working environment, the code did not produce the correct results. This is most likely due to currently unknown errors in both the HDF5 and the FFTW libraries. The results of these investigations are reported in D-JRA2-3.3.

We implemented, along with Virgo's assistance, some platform-independent optimisations to the code, as suggested by these profiling studies. These optimisations included ameliorating the multigrid bottleneck with an FFT procedure. In effect, this replaced the iterative Poisson solver with a non-iterative FFT-based Poisson solver. The porting of this new optimised code proved to be problematic, as the Virgo Consortium updated the code a number of times during the porting/profiling process, typically due to bug fixes, and the necessary suggested porting modifications had to be reintroduced each time.

The new code was ported to the IBM clusters at HPCx, IDRIS, RZG and the IBM Linux cluster at BSC. The new code's performance was not profiled at HPCx nor IDRIS since, as reported in D-JRA2-3.1, the old code's performance characteristics were similar to those on RZG and, as such, further profiling would not offer new insight.

The port of the optimised version, to the NEC at HLRS was successfully completed by the MSc student at EPCC, however, this work was not funded by DEISA (see above) but key results are referenced in D-JRA2-3.3 for completeness.

For a given platform, the optimised version of the Virgo Simulation is now around 6 times faster. For a particular configuration on RZG, the speed up is 8.9 over the previous version of the code.

A method of code migration was also investigated and implemented. In this context, if a code starts running on one particular platform and finishes on another, different platform of a potentially different vendor, then the code is said to have migrated. This is not to be confused with job migration, where a job is taken from one batch queue and migrated to another platform's batch queue. This form of code migration was successfully introduced to the optimised Virgo simulation, within the DEISA infrastructure, and can be utilised using either the UNICORE Workflow or a workflow script invoking the DESHL.

It was discovered that the code required more memory on HPCx than expected. Poor memory management was located in MPI run-time libraries as provided by IBM, which have since been improved and replaced. However, the code continues to require more memory than expected. It is suspected that the HDF5 output routines are to blame. These are serial and currently employ HDF5 v1.4.4. We suggest that the memory issue may be eradicated by employing the parallel version of HDF5

16.4 The optimised version of the FLASH-based simulation, along with the code migration procedures, is documented in D-JRA-3.3, along with the results of the profiling exercises.

It is stated in the TA that the optimised version of the FLASH-based simulation, along with the code migration code, forms the content of public report D-JRA2-3.2,

however, FLASH is used under license and cannot be redistributed by DEISA (otherwise it is freely available), therefore, as such, D-JRA2-3.2 will not be publicly available as planned. However, and most importantly, the modified code will be available to the Virgo consortium for their use. Further, we expect that our optimisations will incorporated into the free, publicly available version of FLASH by the FLASH team in Chicago.

5 WP 4: Distributed data management and information extraction, and WP 5: Grid-enabled visualisation:

In addition to the funding from the Commission for JRA2, WP 2 and WP 6 were funded in part, and WP 4 and WP 5 were funded in total, by VirtU. As such, work done on WP4 and WP5 is not reported here as deliverables D-JRA2-4.* and D-JRA2-5.* are not subject to reviews by the Commission.

6 WP 6: Project management

The activity was managed by EPCC following its standard management procedure for Industrial projects, where these working practices ensured the timely delivery of high quality deliverables.

Note that JRA2 started on one month earlier than originally planned – specifically, JRA2 started on DEISA PM 6 and not on DEISA PM 7. Deliverable dates have thus been adjusted slightly from the original TA to complement the dates of the other deliverables belong to the other DEISA JRAs. This has no impact on the actual work being done, as this is purely a reporting issue. The JRA2 deliverables are now submitted inline with the other JRAs.

It is stated in the TA that the grid-enabled version of GADGET-2 forms the content of public report D-JRA2-1.2, however, GADGET-2 was employed in cooperation with Virgo and cannot be freely redistributed, therefore, as such, D-JRA2-1.2 is not publicly available as planned. However, and most importantly, the code will be available to the Virgo consortium for their private use. Further, the optimisations to GADGET-2 are expected to be incorporated into future releases of GADGET-2 and, hence, will benefit all GADGET-2 users.

There were only minor deviations from the WP2 work program, wherein some of the tools ported to DEISA, namely Power and Correl, were *not* formed in the suggested manner but in a manner which provided more powerful versions of the same tools.

As previously suggested in the first DEISA annual report, there was a major deviation from original WP3 work plan, where we replaced the original code, Hydra_MPI with FLASH. It was noted by EPCC and the Virgo Consortium that Hydra_MPI produces `very similar science' to GADGET-1.1 and it was suggested by Virgo that we replace Hydra_MPI with an Adaptive Mesh Refinement, or AMR, code, namely FLASH. Furthermore, Virgo wanted EPCC to grid enable GADGET-2, which has more "science" than GADGET-1.1.

There is another deviation from the WP3 work program, albeit rather minor, wherein rather than porting FLASH to only the core DEISA platforms, it was decided by Virgo and EPCC, that it would be more interesting port and profile FLASH to a collection of heterogeneous platforms, rather than just IBM Power5 clusters, which make up the core DEISA platforms. This would ensure that any platform-independent optimisations are exactly that: independent of the platform on which the code runs.

The original plan, as stated in the current TA, was to perform an investigation into the possibility of introducing multi-time-stepping into the optimised Virgo simulation. It was decided by Virgo and EPCC to replace this month's work with an investigation

into the apparent necessity for the code to require more memory on HPCx than expected.

It is stated in the TA that the optimised version of the FLASH-based simulation, along with the code migration code, forms the content of public report D-JRA2-3.2, however, FLASH is used under license and cannot be redistributed (otherwise it is freely available), therefore, as such, D-JRA2-3.2 will not be publicly available as planned. However, and most importantly, the code will be available to the Virgo consortium for their private use. Further, WP's optimisations to FLASH are expected to be incorporated into FLASH by the FLASH team in Chicago and, hence, will benefit all registered FLASH users.

7 JRA2 Presentations

"DEISA: JRA2: WP1 and WP2: Status Report", Gavin Pringle, Virgo Consortium meeting, Max-Planck-institute for Astrophysics, Garching, Munich, Germany, 14 April, 2005.

"Cosmology using Grid MPI", Kevin Stratford, Gavin Pringle, Volker Springel, SunHPC, Heidelberg, 21 June, 2005.

"DEISA: JRA2: The Results of WP1 and WP2", Gavin Pringle, Virgo Consortium meeting, Institute for Computational Cosmology, Department of Physics, University of Durham, U.K., 18 November, 2005.

The results of WP3 will be presented to the Virgo Consortium in Leiden, The Netherlands, in January, 2007, which is the next available, full consortium meeting.

8 JRA2 Publications

DEISA: JRA2 poster, ISC05, Heidelberg, Germany, 2005.

"EPCC at Sun HPC", Gavin J. Pringle, EPCC News, issue 54, Summer 2005.

"DEISA: a Europe-wide Research Infrastructure", Jeremy Nowell, Gavin J. Pringle, Terry Sloan, EPCC News, issue 57, Summer 2006.

DEISA: JRA2 poster, ISC06, Dresden, Germany, 2006.

FLASH: Modelling the Universe, Gavin J. Pringle, EPCC News, issue 59, October, 2006.

The code migration work will be submitted to a suitable international conference in the near future, after some further refinements are completed

JRA2-related publications

"Comparison of Portable Binary Data Formats within a Cosmological Simulation", Wei, W., M.Sc. Dissertation, The University of Edinburgh, 2005.

"Porting, Profiling and Optimising an AMR Cosmology Simulation", Radoswlaw Hubert Ostrowski, MSc in High performance Computing, The University of Edinburgh, 2006.

9 List of deliverables

The following table contains a complete list of all the JRA2 deliverables.

D-JRA2-1.1: Specifications document for D-JRA2-1.2 D-JRA2-1.2: Grid-enabled implementation of GADGET for metacomputing environments D-JRA2-1.3: Documentation for D-JRA2-1.2 D-JRA2-2.1: Specifications document for D-JRA2-2.2, D-JRA2-2.3, D-JRA2-2.4 D-JRA2-2.2: Grid-enabled "Group finders" D-JRA2-2.3: Grid-enabled tools to build halo merger trees D-JRA2-2.4: Grid-enabled tools for evaluating basic properties of the dark matter D-JRA2-2.5: Documentation for D-JRA2-2.2, D-JRA2-2.3, D-JRA2-2.4 D-JRA2-3.1: Specifications document for D-JRA2-3.2 D-JRA2-3.2: FLASH on the DEISA Infrastructure: System-independent optimised FFT routines, multi-time stepping with support for code migration D-JRA2-3.3: Documentation for D-JRA2-3.2 D-JRA2-4.1: Specifications document for D-JRA2-4.2, D-JRA2-4.3, and D-JRA2-4.4 D-JRA2-4.2: Global data management environment (replica catalogue and metadata catalogue) D-JRA2-4.3: API and reference implementation of an application library to generate metadata from running application. Fortran 90 bindings for FLASH and GADGET D-JRA2-4.4: Basic servlet allowing multi-criteria search of the metadata catalogue from any location using a standard Web browsers D-JRA2-5.1: D-JRA2-3.2 with added support for remote visualisation D-JRA2-5.2: D-JRA2-1.2 with added support for remote visualisation

D-JRA2-5.3: Documentation for D-JRA2-5.1 and D-JRA2-5.2

D-JRA2-6.1: Biannual interim reports

D-JRA2-6.2: Final Report

Repeated remark: in addition to the funding from the Commission for JRA2, WP 2 and WP 6 were funded in part, and WP 4 and WP 5 were funded in total, by VirtU. As such, work done on WP4 and WP5 is not reported here as deliverables D-JRA2-4.* and D-JRA2-5.* are not subject to reviews by the Commission. In addition to the funding from the Commission for JRA2, WP 2 and WP 6 were funded in part, and WP 4 and WP 5 were funded in total, by VirtU. As such, work done on WP4 and WP5 is not reported here as deliverables D-JRA2-4.* and D-JRA2-5.* are not subject to reviews by the Commission.