



CoherentPaaS

Coherent and Rich PaaS with a
Common Programming Model

ICT FP7-611068

Holistic Transaction Management Architecture D4.1

February 2014

Document Information

Scheduled delivery 28.02.2014
Actual delivery DD. MM.YYYY
Version 1.5
Responsible Partner UPM

Dissemination Level:

PU Public
PP Restricted to other programme participants (including the Commission)
RE Restricted to a group specified by the consortium (including the Commission)
CO Confidential, only for members of the consortium (including the Commission)

Revision History

Date	Editor	Status	Version	Changes
7.03.2014	Ricardo Jimenez	Draft	0.0	Toc and introductory contents
25.03.2014	Iván Brondino	Draft	0.1	2 nd draft
2.04.2014	Marta Patiño	Draft	0.2	3 rd draft
6.04.2014	Ricardo Jimenez	Draft	0.3	4 th draft
15.04.2014	Iván Brondino	Revised	0.4	1st peer review
23.04.2014	Iván Brondino	Revised	0.5	2nd peer review
27.04.2014	Ricardo Jimenez	Final	1.0	Final version

Contributors

Ricardo Jimenez-Peris
Iván Brondino
Marta Patiño-Martínez

Internal Reviewers

Pavlos Kranas
Jose Pereira

Acknowledgements

Research partially funded by EC 7th Framework Programme FP7/2007-2013 under grant agreement n° 611068.

More information

Additional information and public deliverables of CoherentPaaS can be found at: <http://coherentpaas.eu>

1. Executive Summary

This deliverable describes the holistic transaction management architecture. The holistic transaction management provides a set of components that implement transaction management: log, snapshot counter and conflict detection. These components are decoupled so that they can be scaled-up independently. The holistic transaction management implements snapshot isolation, therefore all data stores accessed in a transaction need to provide multiversion data and write-write conflict detection. The data stores either use their own log and functionality to redo transactions, or provide the logging information to the holistic transaction management. Applications can bracket transactions directly or through the common query language. In both cases a proxy client to the transaction manager is used at the client side. At each data store a local transaction manager must be deployed.