



Project co-funded by the European Commission within the ICT Policy Support Programme

ReAAL

CIP ICT PSP – 2012 - 325189

www.cip-reaal.eu

Platform official releases

[Deliverable 2.3b, Revision 1.0]

Key Information from the DoW

Due Date	15-Sep-2015
Type	Software
Security	Public

Description:

D2.3 represents the official releases of universAAL software within ReAAL. Two versions of this deliverable are planned: D2.3a in early stages for the start of work in WP3, and then, after some intermediate releases, D2.3b as the final official release within ReAAL timeframes.

Lead Editor: Álvaro Fides (UPVLC)

Internal Reviewer: Michele Girolami (CNR-ISTI)

Versioning and contribution history

Version	Date	Author	Partner	Description
0.1	07-Sep-2015	Álvaro Fides	UPVLC	First un-reviewed proposal of document
0.2	15-Sep-2015	Álvaro Fides	UPVLC	Final content ready for review
0.3	16-Sep-2015	Michele Girolami	CNR	Reviewer-approved version
1.0	22-Sep-2015	Saied Tazari	Fh-IGD	Official release

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.



Executive Summary

Official administrative representation of the last version of the platform software release within ReAAL. All links to the software are included and explained, and finally listed as references for quick use.



Table of Contents

1.	About This Document	5
1.1.	Deliverable context.....	5
1.2.	Release management	5
1.3.	Version-specific notes.....	6
1.3.1.	General information	6
1.3.2.	Release features	7
2.	Conclusion.....	9

1. About This Document

This document is the reference to the actual software of the platform official release, version B (last version). It provides links to the software as well as a brief explanation about the management of the universAAL releases. The structure of this document differs from that of D2.3a, and instead follows the refined structure that has progressively been adopted for the intermediate platform releases in the internal deliverables ID2.2a/b/c, with updates concerning the last release 3.4.0.

1.1. Deliverable context

Project item	Relationship
Objectives	O3: The objective refers to setting up the runtime platform components, which are the actual software that composes this deliverable.
Exploitable results	Res 1: Requires a common open platform. This deliverable is its third intermediate version. Res 4: “A specific release of the universAAL platform ...” is exactly this deliverable.
Work plan	WP2 T2.3: Task to which the deliverable belongs WP2 T2.1: Identifies the requirements to be satisfied by the software in this deliverable. WP2 T2.2: Sets up framework to fix problems found in the software of this deliverable. WP3 T3.2: Adapts applications to the platform software of this deliverable. WP4: Pilot operations use the platform software.
Milestones	MS4 Final corrective actions: As a regular update of the platform, it does not impact on the milestone itself, but helps the “Mature application releases” that are part of MS4.
Deliverables	ID2.2a,b,c: Intermediate releases of the platform software in this deliverable. D2.3a: First release of the platform software, first version of this deliverable.
Risks	Rk1, Rk3, Rk4, Rk5: These are risks that have the platform software as the probable causative of such risk.

1.2. Release management

Task 2.3 defines two key roles for preparing and packaging the platform release.

Release Master. Its role is to prepare the software to be released. The master must perform the following operations:

- Tag the code in the source code server
- Increase the version number of the software related to the release
- Deploy all the software artifacts on the universAAL Nexus server (<http://depot.universaal.org/nexus/>).

Release Manager. Its role is to package the platform release with the latest software prepared by the master. The role of the manager is to:

- Checkout the source code repositories for all the different universAAL sub-projects (i.e. universAAL middleware, universAAL UI, universAAL LDDI etc.)
- Configure and run the OSGi container in order to obtain a customized universAAL OSGi distribution (namely the universAAL distribution)
- Install the sub-projects on the OSGi container and perform some preliminary checks
- Package the OSGi distribution and upload it to the official universAAL web-site.

1.3. Version-specific notes

1.3.1. General information

The release source code is available in GitHub (<https://github.com/universAAL>) across different subproject-specific repositories. The compiled binaries are deployed in Nexus Maven repository (<http://depot.universaal.org/nexus/content/repositories/>) and can be browsed from the Nexus interface (<http://depot.universaal.org/nexus/#view-repositories;releases~browsestorage>)¹.

In addition to these, a pre-packaged release of the universAAL platform embedded in its own running environment is made available at the AALOA hosting site (<http://universaal.aaloa.org>). The universAAL release consists of a compressed archive, prepared for Windows or Linux, which contains:

- A ready-to-run Karaf 2.2.9 OSGi Environment that includes:
 - Customized universAAL OSGi console and shell commands
 - Pre-configuration for access to the universAAL Nexus repositories
 - Pre-configuration for access to basic universAAL Karaf features

The links to the packaged release is <http://universaal.aaloa.org/media/universaal-3.4.0.rar>.

These pre-packaged release is a convenience method for publicly downloading the Karaf distribution available at <https://github.com/universAAL/distro.karaf> without having to deal with GitHub or its accounts.

¹ The hyperlink in this document may lead to a missing page depending on the program used to view the document or the browser of choice. If so, go to <http://depot.universaal.org/nexus/>, click on “repositories” to the left, and then browse the “Releases” folder.

Once downloaded, universAAL can be set up and started as usual as it is described in https://github.com/universAAL/platform/wiki/DG-Quick-setup-and-start-guide#Just_run_in_OSGI.

1.3.2. Release features

Below we report the major changes for each of the building blocks of the universAAL platform:

- **Middleware:** Bug fixes and updates to make it more debug-friendly. Included new feature:
 - **Distributed MW:** Allows to manage the MW from connected instances.
- **Context and Ontologies:** Bug fixes to make ontologies more cohesive and the CHE more memory-efficient and tenant-aware. Added new ontologies:
 - **Multi-factor authentication**
 - All profile-related ontologies are now identified as PrivateResource, which will help in the definition of permissions.
- **LDDI:** Added a helper bundle for device abstraction that should help create future Exporters.
- **Remote Interoperability:** Several bug fixes and a new feature:
 - Added an Apache ACE target Karaf feature, which allows a central Apache ACE server to deploy and update bundles remotely
- **Android:** Several fixes and updates on the middleware and the examples, and now apps can access deeper data through property-paths in the data received from the middleware.

For a comprehensive, detailed list of all bug fixes and new features in this release 3.4.0, please refer to the Release Notes:

<https://github.com/universAAL/platform/wiki/RD-Release-History>. Developers can check which updates require code changes in the Transfer Instructions: <https://github.com/universAAL/platform/wiki/Transfer>.

Being the last release within ReAAL, this version of the platform was supposed to cover all remaining requirements defined in D2.1. Due to time constraints and after the focus on the development of new features was toned down as a consequence of the last review, some of the requirements were reduced in scope and others were halted (taking into account that they would not have been used in the current pilot deployments anyway):

N_PRIV_1 - Privacy Protection: *Completed.* The middleware feature of enabling Functional Manifests allows apps to declare what type of data they are going to handle. This represents a Permission-based framework where applications can be allowed or denied based on the type of data they manage. This feature was already available in earlier releases but was not advertised to developers nor completely documented. To ease things further, a new concept was added to ontologies that allow directly marking some concepts as “private”.

N_SECU_1 - Multi-factor authentication: *Completed.* The inclusion of ontologies for Multi-Factor authentication allow the creation of uAAL services for this matter.

The universAAL platform cannot provide the infrastructure for M.F.A. itself because it is an aspect that is the responsibility of each deployer.

N_SCALA_1 - Remote management: *Completed.* Rather than developing a new remote management framework from scratch, Apache ACE was used and included as an installable feature in uAAL. Together with a standalone server tool, it can be used to deploy and update collections of bundles across several remote instances.

N_INTEROP_4 - SPS Exporter: *Partially Completed.* The SPS Exporter was developed as a private endeavour for a specific pilot and cannot be made open source. Instead, the module that was used to bridge this Exporter to universAAL has been added to LDDI in order to ease the creation of future Exporters.

N_INTEROP_2/3 - HL7/Continua Exporters: *Not completed.* Several reasons led to the decision of not implementing these exporters: In addition to time constrains, they are communication standards rather than network technologies, which makes them a less obvious to map to Exporters. Also, the pilots using them are already using their own specific universAALized solutions.

N_CONF_3 - Remote configuration: *Not completed.* Due to time constrains this could not be implemented, and has to be performed with a combination of existent tools, such as Karaf's own remote management features, the local configuration framework, Multitenancy (to some degree) and the new Remote Management feature.

2. Conclusion

The platform official and intermediate releases were scheduled for task 2.3 to be reported in deliverables D2.3 and ID2.2. Because the objectives of all these deliverables and versions are the same (reporting the software of each platform version) the structure of the deliverables would be the same for all of them – even if ID2.2 is technically an “Internal Deliverable” – but the actual structure has been updated version after version.

Now the features of the platform have been completed in the scope of the ReAAL project. But to the extent of available resources, the developers involved in the universAAL platform within ReAAL will maintain and support the platform until the end of the project with continuous bug fixes and, if possible, new features.