



Ecosystem infrastructure for smart and personalised inclusion
and PROSPERITY for ALL stakeholders

D201.4 Working online version of Unified Listing with Mainstream Product Feature capability

Project Acronym **Prosperity4All**
Grant Agreement number **FP7-610510**

Deliverable number **D201.4**
Work package number **WP 201**
Work package title **System Architecture and Unified-**

Listing/Marketplace

Authors **Tony Atkins, Gregg Vanderheiden**

Status **Final**

Dissemination Level **Public**

Delivery Date **26/09/2016**

Number of Pages **25**

Keyword List

assistive technologies, mainstream products, manufacturer participation, community participation.

Version History

Revision	Date	Author	Organisation	Description
1	03/06/2014	Tony Atkins	RtF-I	Initial draft document.
2	10/07/2015	Tony Atkins	RfF-I	Second version
3	17/07/2015	Gregg Vanderheiden	RtF-I	Edits, updates and additions
4	29/07/2015	Tony Atkins	RtF-I	Incorporated feedback from reviewers.
5	14/09/2015	Gregg Vanderheiden	RtF-I	Edits, updates, Chapter 1
6	05/07/2016	Tony Atkins	RtF-I	Updates for M30
7	13/09/2016	Gianna Tsakou	Singular Logic	Requested various clarifications, and suggested the addition of a "summary" and "future work" chapter.
8	14/09/2016	Tony Atkins	RtF-I	Updated based on internal review.
9	20/09/2016	Gregg Vanderheiden Gianna Taskou Matthias Peisner	RtF-I Singular Logic Fraunhofer IAO	Requested title update and clarification on integration with C4A autopersonalisation.
10	21/09/2016	Tony Atkins	RtF-I	Updated based on second internal review.

Table of Contents

- Executive Summary 1**
- 1 Contribution to the global architecture..... 3**
- 2 Background information on the Unified Listing 5**
 - 2.1 What is the Unified Listing?..... 5
 - 2.2 Who does the Unified Listing serve? 5
 - 2.3 What does the Unified Listing help users do?..... 5
 - 2.4 Who contributes information to the Unified Listing? 5
 - 2.5 How is the information in the Unified Listing used? 6
 - 2.5.1 Federated Databases..... 6
 - 2.5.2 Third-party integrators..... 6
 - 2.5.3 End users 6
- 3 Adding the ability to add and edit records in the Unified Listing..... 7**
 - 3.1 The special role of manufacturers 7
 - 3.2 Balancing manufacturer and community input..... 7
 - 3.2.1 Manufactures as a distinct data source 7
 - 3.2.2 Data sources and the unified record..... 7
 - 3.3 The Manufacturer workflow..... 8
 - 3.3.1 Adding a Record 8
 - 3.3.2 Manufacturer review of an entry..... 9
 - 3.4 The Community Member workflow 9
 - 3.5 The Review workflow for Curators 9
 - 3.6 What data are in an entry?..... 10
 - 3.6.1 Basic information 10
 - 3.6.2 Data regarding certifications or other country specific data 10
 - 3.6.3 Data using GPII specific terminology..... 11
 - 3.6.4 Feature data 11
 - 3.6.5 Pricing data..... 12
 - 3.6.6 Application Data 12

4	Coordinating and communicating changes.....	13
5	Moving into Production.....	14
5.1	Peer Reviews.....	14
5.2	Testing.....	16
5.2.1	Code Coverage	16
5.2.2	Integration Tests	17
5.2.2.1	Browser tests that interact with server components.....	17
5.2.2.2	Server tests that require browser control and awareness.....	17
5.2.3	Browser navigation	18
5.2.4	Keyboard navigation	18
5.3	Continuous Integration.....	19
5.4	Preparing for Scale.....	19
6	Conclusions and Future Work	21
7	Key Links	21

List of Tables

Table 1:	Sample status values.....	10
Table 2:	Libraries reviewed and distributed as part of this project.....	16

List of Abbreviations

Abbreviation	Full form
API	Application Program Interface
AT	Assistive Technology
C4A	Cloud4All
D	Deliverable
GUI	Graphical User Interface
GPII	Global Public Inclusive Infrastructure
ICT	Information and Communications Technology
IDE	Integrated Development Environment
ISO	International Organization for Standardization
IT	Information Technology
P4A	Prosperity4all
UI	User Interface
UL	Unified Listing
UX	User Experience
WP	Work Package

Executive Summary

Update for Month 30:

If you are already familiar with the work submitted in Month 18, the updates related to the Month 30 deadline are mainly at the end of the document, beginning with "Moving into Production". The working online version of the Unified Listing can be found at: <http://ul.gpii.net>

In this document we will briefly review the current state of the Unified Listing, and cover the upcoming work that will allow manufacturers and community members to directly contribute new information.

The Unified Listing is a database of assistive technologies (AT) and mainstream products with accessibility features. It brings together AT and mainstream product databases internationally into a single federated database that both draws from and feeds the other databases. The Unified Listing software currently consists of:

1. Database structures and associated views.
2. An API to manage adding, editing, and retrieving information.
3. Import scripts that pull information from EASTIN and GARI, our first federation members.
4. A mechanism to notify federated database managers of new information from other vendors that is not already reflected in their dataset.
5. A front-end UI provided by the Semantic Alignment Tool: <http://sat.gpii.net/>

In Prosperity4All we will add:

1. An improved editor's UI for adding, editing, and retrieving information
2. A mechanism for identifying manufacturers and identifying their special relationship with products in the Unified Listing.
3. A web UI to allow manufacturers to view, add and edit their products.
4. A mechanism to notify manufacturers when other parties (community members, database vendors) update information related to their products.
5. Import scripts for other databases that affiliate
 - a. For example ABLEDATA (US) and The ARC (focusing on cognitive disabilities)
6. An improved mechanism for notifying federated database managers of new information from other vendors that is not already reflected in their dataset (based on feedback from them going forward.
7. The mechanisms for user feedback and feedforward to product developers and vendors – and feed-sideways to other users.

We will also be coordinating with work by other (non-Prosperity4All) partners on enhancements to the Unified Listing

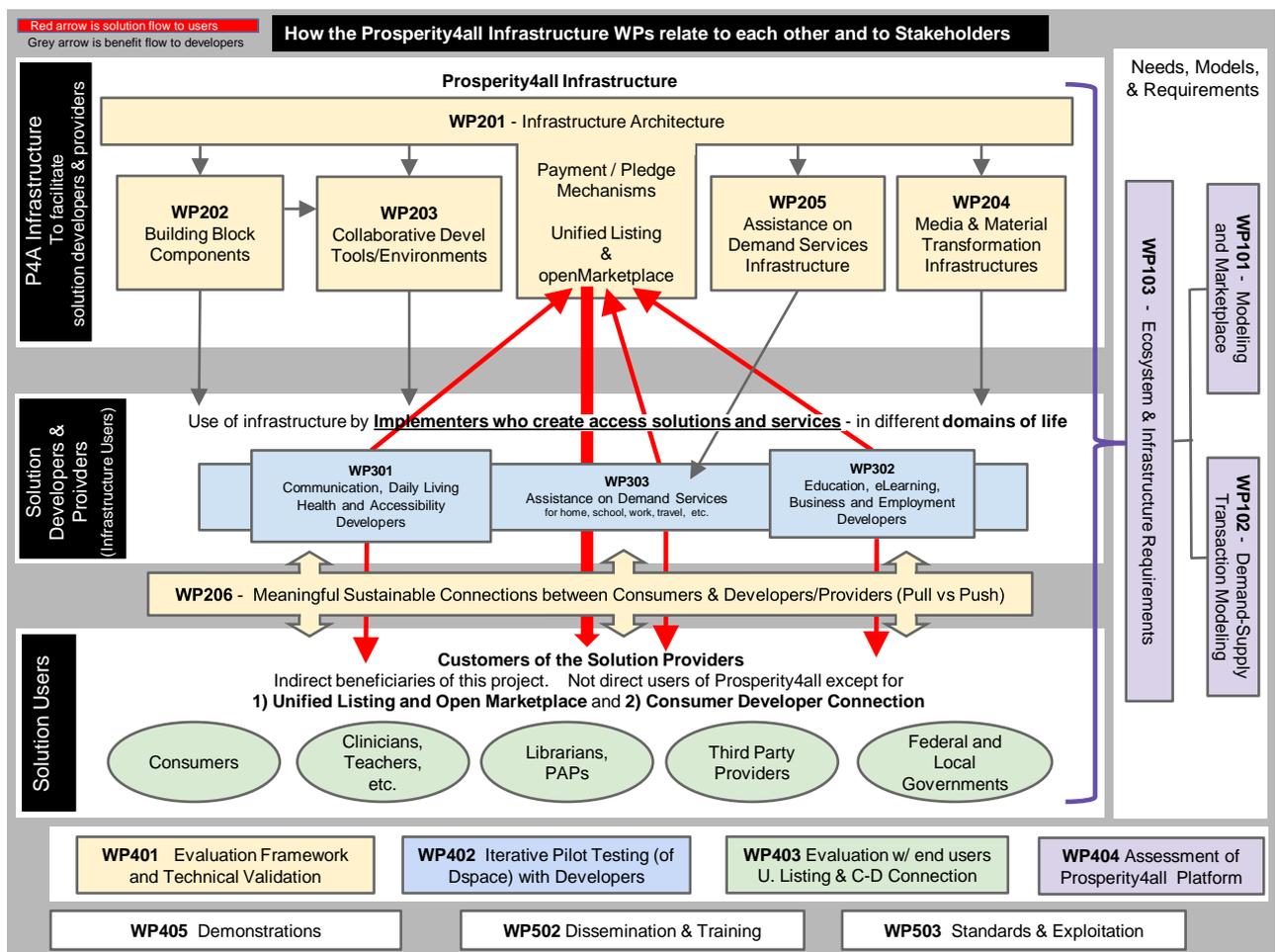
1. The Shopping / Alerting interface for the Unified Listing being developed by the Universal Interface and Information Technology Access Rehabilitation Engineering Research Center.
2. The Consumer Electronics Association Foundation efforts to enhance mainstream products in the Unified Listing

1 Contribution to the global architecture

The Unified Listing is a core element in the Prosperity for All infrastructure. First, it is core to the autoperpersonalisation components since it contains all of the information on accessibility features and mainstream products and assistive technologies. It is from the Unified Listing that the solutions registry used by the matchmaker is distilled.

The Unified Listing is also core to the Developer Space. It is a primary mechanism for developers to reach their customers internationally. There is no cost to list products in the unified listing or to use it. It is also the most comprehensive listing internationally. It is therefore a primary source for any developer, especially new developers, to be able to reach a broad audience effectively and economically. In addition, the unified listing is combined with technology such as the Shopping and Learning Aid which can take an individuals needs and preferences and use them to help find all of the relevant products and features in the database. In addition, users can ask it to alert them to any new products or features that would appear and meet their needs.

Figure 1: Overall Picture of Prosperity4all



A variant of the Unified Listing will also be used within the developer space to list all of the components, frameworks, service infrastructures etc. available to developers.

The openMarketplace, which is not discussed here, is separate from, but attached to, the Unified Listing. The openMarketplace is simply a place where developers can actually sell their products if they do not have any other mechanism for selling their products and handling international purchases already. There will be a commission for selling products through the openMarketplace but there is no commission for having them listed in the Unified Listing.

This work is part of WP 201. The Unified Listing would be used by all of the SP3 projects in that they are either assistive technologies or mainstream technologies with accessibility features. It also relates to SP1 (meeting identified needs) and SP4 where it will be evaluated. Finally, the consumer connection and feedback aspects are being carried out in connection with WP206.

2 Background information on the Unified Listing

2.1 What is the Unified Listing?

The Unified Listing is a central database that stores information about AT solutions and mainstream products with built-in accessibility features. It brings together AT and mainstream product databases internationally into a single federated database that both draws from and feeds the other databases. The work in progress is available on the web at: <http://ul.gpii.net/>

2.2 Who does the Unified Listing serve?

The Unified Listing has a number of different types of users:

1. End users of AT solutions and mainstream devices with accessibility features.
2. Caregivers of the above.
3. Therapists and other AT professionals.
4. Manufacturers of AT solutions and mainstream devices with AT features.
5. Integrators and third parties building solutions on top of the Unified listing.
6. Developers and vendors

In addition to serving a broad range of communities, the Unified Listing is also intended to help users from around the world, and supports multiple languages.

2.3 What does the Unified Listing help users do?

The users described above have two broad use cases:

1. They want to locate existing solutions.
 - a. For themselves,
 - b. To understand the field and what is available,
 - c. For market and competition research
2. They want to provide information about solutions.
 - a. About their products (developers/vendors)
 - b. About products (users)

2.4 Who contributes information to the Unified Listing?

The Unified Listing currently contains information pulled from database vendors like EASTIN, the EASTIN federation members, GARI and soon ABLEDATA. To continue growing and updating both the Unified Listing and its federated databases, the Unified Listing will also enable manufacturers and community members to contribute directly.

2.5 How is the information in the Unified Listing used?

There are a number of projects working to meet the two key use cases outlined above, entering solution information and using solution information.

2.5.1 Federated Databases

The Federated Databases are both producers and consumers of the data in the Unified Listing. First, the databases provide source records, which are regularly synchronized with the Unified Listing. The data coming from EASTIN and GARI provided the initial dataset we have available.

The Unified Listing is both broader (includes AT and Mainstream products) and narrower (it focuses on ICT related products while the federated databases often cover all types of AT). Although the foci of the databases are different we have worked out a way for the flow of information to be two-way. The Unified Listing provides federation members with services and tools that make them aware of new solutions as well as updated information about existing solutions, so that they both feed and are fed by the Unified Listing.

The Unified Listing also introduces some unique features such as the Universal Product ID that allows the other database federations to consolidate search results, and a one-stop update function for developers/vendors that can provide greater incentive for them to update their product data rather than requiring database managers to do so.

2.5.2 Third-party integrators

The open-API of the Unified Listing allows others to use the information in still new and expanded ways.

The Universal Identifier of the Unified Listing also provides way of uniquely identifying a specific solution even if described in different ways using different names in different databases. Integrators can use the Universal Identifier in the Unified Listing to collect information like features, settings, and user reviews without collecting duplicate information about the solution itself.

2.5.3 End users

Today, end users can search and view the data in the Unified Listing via the Semantic Alignment Tool, which is integrated with the Unified Listing. The Semantic Alignment tool organizes the raw data found in the Unified Listing using a hierarchical ontology, and allows users to browse the data set. The Semantic Alignment Tool can be found at <http://sat.gpii.net>. As the other interfaces come online (such as the Shopping/Alerting Aid) end users will have other options as well.

3 Adding the ability to add and edit records in the Unified Listing

For the most recent Cloud4All review, the initial REST API for the Unified Listing was created, and was used to import data from EASTIN, and to integrate with the Semantic Alignment Tool.

In Prosperity4All we are proposing to create an editing interface that can be used by all users to add and edit data, but which provides a special interface designed and optimized for manufacturers.

3.1 The special role of manufacturers

Manufacturers play a special role in describing their products. They can provide the clearest guidance about what a product is intended to do, what its features are, and where to find additional information. However, theirs is not the only voice. To provide a truly inclusive community, we need to allow and encourage everyone to contribute information about solutions they are aware of.

3.2 Balancing manufacturer and community input

We need to recognize the special role of manufacturers and the data they provide without preventing the larger community from participating. We propose to do this by establishing a special relationship between a manufacturer and the solution they provide. This relationship would initially be managed on a case-by-case basis by site administrators. Eventually, we would build tools to manage this relationship through the web interface. Once the relationship has been established, a manufacturer would be able to create and edit records for products they own in a new way.

3.2.1 Manufactures as a distinct data source

The Unified Listing already has the concept of data sources, which we use to distinguish data coming from individual databases. For the products they produce, manufacturers add entries where they do not already exist. This information would then be moderated as with all entries and also fed to the other federated databases.

3.2.2 Data sources and the unified record

Source records are each associated with a Unified Listing base record, which is record unique to the Unified Listing and created by curating data from one or more source records from federated databases, manufacturers, and/or the community. This "unified" record is created

by a group of UL moderators, whose job is to take into account and balance the input of all contributors, while removing any superlatives such as "best", "only", "fastest" etc. that are not supportable or are temporal.

This model allows and encourages manufacturers to monitor and contribute to information about their products, but also serves a wider set of use cases. As an example, a manufacturer who stops working in a given market may wish to withdraw their entry. We would rather keep the unified record as a reference point for existing users, and to assist people in moving from an older product they may have used to newer alternatives. Also people often spend much time looking for something that has been dropped by a manufacturer – and it is important they can find it and see that it is no longer available. To accommodate both audiences, we preserve the product in the database but mark it appropriately (e.g. 'no longer sold by manufacturer', 'no longer supported by manufacturer', etc.)

3.3 The Manufacturer workflow

Let's talk through the workflow manufacturers would use, beginning with adding a record.

3.3.1 Adding a Record

When we originally imported data from the EASTIN databases and GARI, we began with source data from one or more of these sources, and then created the "unified" record. The original "source" records are preserved in their entirety, and are associated with the "unified" record such that we can present all the different listings from different federated databases at once.

Individual contributors (including manufacturers) would follow a similar process. They would contribute an entry for a new product or an update for an existing product. These are reviewed, and incorporated into the "unified" record by moderators. The only difference is that the companies or community member's original un-curated submission is not viewable by all database viewers as are the curated entries from the federated databases. This is done because we do not want to be responsible (or liable) for claims made by companies or raw descriptions from community members. Uncurated entries are therefore treated as "leads" and "input" to the Unified Listing curators construction of a new entry in the Unified Listing. Submissions by companies or individuals can be in any language – though English is preferred when possible to maximize the ability of curators from different countries to read it directly.

Each federated database may contribute in whatever language(s) they choose, and these entries, in their original language, will be visible to all users. In addition, authorized translators can also go through the database and create translations of the entries there, if

there is a desire to have the entries in another language directly (and not rely only on the auto-translator function in the Unified Listing). The end result is a web of related records describing the same product, which are visible to the end user (sometimes directly in different languages, and sometimes only through auto-translators).

Here is the proposed workflow for a manufacturer adding a record:

1. A manufacturer adds a new "lead" record describing a product not in the database.
2. The curators create a "unified" record for the product.
3. Database vendors are notified that a new record has been created (see below).
4. The federated database managers create new records in their databases as appropriate.
5. On the next sync, the federated database record is detected and using the Universal ID, is matched to the "unified" record in the Unified Listing, and added as another "source" record associated with the same "unified" record.

3.3.2 Manufacturer review of an entry.

Much of the data we have imported from EASTIN or GARI will initially lack manufacturer review. In these cases, the workflow will be something like the following:

1. A manufacturer encounters an existing record for their product.
2. They review the record and submit any updates or suggested edits
3. Curators of the Unified Listing are notified of the update (see below).
4. Curators incorporate any new information into the "unified" record as needed.
5. Manufacturers are notified of the update to the "unified" record.
6. Federated databases are notified of the update to the "unified" record.

3.4 The Community Member workflow

A community member's workflow is the same as that outlined for manufacturers.

3.5 The Review workflow for Curators

The information from the different sources (manufacturers, consumers, practitioners, etc.) within the Unified Listing is curated by moderators (also referred to as curators) to ensure accuracy, quality and decorum.

Manufacturers, community members, and database vendors may all supply information to the Unified Listing, which are incorporated appropriately and fed to the other federated databases as per the federated databases wishes and needs.

Table 1: Sample status values.

Status	Description
Unreviewed	A record has been created but has not been reviewed.
Incomplete	These may be displayed if felt to be largely complete and appropriate, but will clearly be flagged.
Active	The record has been reviewed as complete and safe for use and is displayed without warnings or other qualifications.
Needs Review	The record is visible to the public, but is flagged as potentially needing update in some manner – which is noted.
Invisible	The record has been marked invisible and will only be visible to editors. This is likely to only be used during the initial review or when managing duplicate records.

Each status is a step in the workflow, and changing the status indicates how reliable the record is. The full set of statuses will be expanded upon by the review committee as they begin regularly reviewing records.

3.6 What data are in an entry?

3.6.1 Basic information

The minimum data required to construct a Unified Listing entry are as follows:

1. The name of the solution (its title).
2. The description of the solution.
3. One or more images that represent the solution. (If missing, the entry would be listed but marked as incomplete)
4. The language in which this record is written.

In addition, to facilitate the collection of new data like pricing and features (see below), for this milestone we will add a "notes" field, which can be used to enter additional information not covered above.

3.6.2 Data regarding certifications or other country specific data

There may be information that varies from country to country such as certifications (safety, funding, etc.) Initially we will handle these all in the text field. Later we will look to see if some types of country specific fields are warranted or not.

3.6.3 Data using GPII specific terminology

Manufacturers would enter this basic information for the products they produce, and we would work with them over time to add additional data that uses specific terminology such as features and settings. For settings we will also work with them to make sure all of their settings are registered in the Preference Term Dictionary and equated with their corresponding common-format terms.

3.6.4 Feature data

One of the key challenges in the Unified Listing is to take the various ways that people have used to describe solutions, and standardize that so that we can help users find solutions across the whole Unified Listing.

We are supplied with a range of feature data by the federated databases that contribute to the Unified Listing. ISO 9999 is used by the EASTIN federated database to categorize all AT solutions it lists. Although this information is helpful to AT professionals, it is not fine grained enough, and is not categorized in terms of user needs. For example, screen readers are categorized either as "special output software" or "alternate input device".

GARI, on the other hand, uses its own structured vocabulary to map user needs to product features. For all entries, this information is separated into five broad categories:

1. Dexterity
2. Vision
3. Hearing/Speech
4. Cognition
5. Hardware

The first four categories are described in terms of the help they provide. The fifth concerns features that may be of interest, but which do not generally help in one of the other areas. A good example is the device's battery life.

GARI has a full list of features that are tracked, which are each categorized into one of these five categories. Each mobile device has metadata to indicate whether it has a feature, doesn't have a feature, or whether no data is available.

Although GARI's feature data is fine-grained, it is largely specific to phones and tablets. These are divided into broad categories according to what kind of support they provide (dexterity, vision, hearing, speech, cognition). This type of coarse-grained categorization is less specific, but can more easily be applied to a range of solutions

We need something which is detailed enough to be useful in finding solutions to specific problems, but general enough that we can expect manufacturers to enter the data. As we do not have anything like this, for this milestone we are simply collecting information about

features in free text form. This problem is being explored as part of a parallel project at the UIITA-RERC based out of the Trace Center in the US as part of the Shopping and Alerting Aid project (a companion search and alert interface to the Unified Listing). As this work is completed it will be tapped and incorporated as an alternate interface to the current "Semantic Alignment Tool" interface.

3.6.5 Pricing data

Pricing data is an important aspect of the buying decision. Unfortunately, it is difficult to collect this data in a way that is both accurate and timely.

For software, this is complicated enough. Many of the solutions we list operate in a number of markets, some based on location, some based on industry (educational versus commercial, for example). There may be a dozen prices for a single solution, and they may be based on criteria that are difficult to compare (for example, bulk licensing arrangements).

Pricing for physical devices is also problematic. Devices such as phones may be available at reduced cost with a phone contract, but the up-front cost is not the full cost. Devices you can purchase outright may only be available from a single country, in which case shipping and customs can add significant costs when purchasing from elsewhere in the world. There are also discount schemes, reseller programs, and general retail markup to consider, each of which can add a lot of variety in the final price a consumer pays. Prices are even used as an experimental variable in "A/B testing", and may not even be the same for two visitors buying the same exact product from the same store on the same day.

Even if you somehow managed to collect data that is accurate for a single point in time, pricing data changes rapidly. Rather than attempt to solve all of this, for this milestone, we will collect whatever pricing data manufacturers choose to provide and place it in the free text field. As with feature data, we will review and add structure to this once we see the patterns that emerge.

3.6.6 Application Data

In addition to the general product information, we will also be adding features to allow users to attach application information. These notes, which will be moderated for appropriateness, spam, and decorum, can vary widely from compatibility information, to alternate uses for the product, to tricks for implementing special functionality. This work will be carried out in conjunction with WP206 on maintaining the user connection with manufacturers and the development and application of new and existing solutions

4 Coordinating and communicating changes

As outlined previously, many parties can provide new information about a single solution. To assist each of these groups in coordinating with each other, and to help ensure that data is kept up to date, we are proposing to extend the "updates" report and notification mechanism developed during the Cloud4All grant.

For all edits through the web interface or updates made via an import script, the date at which the record was updated is recorded. We also track the source for each record. With this information, we can do things like:

1. Notify manufacturers when someone else adds new information about the solutions they provide.
2. Make moderators aware of new records that need to be reviewed.
3. Make federated database owners aware of new solutions as they are added by manufacturers, the community, or other database vendors.
4. Make federated database owners aware when a manufacturer provides new information about products the database vendor already knows about.
5. Make moderators aware of changes that may need to be incorporated into the "unified" record.
6. Allow users to be able to track certain types of products and be alerted when new ones appear or existing ones change.

This information is available as a live report whose parameters (which sources, what time period, etc.) can be updated through the web interface. There is also a script that runs nightly and emails a summary of recent updates. Clicking the link within the summary email opens the full detailed report on the web.

For the initial Cloud4All deliverables, the focus was on federated AT databases. This Prosperity4All deliverable includes work to extend this concept to cover mainstream devices and manufacturer and community input. For this deliverable, the process of configuring update reports will be manual. Beyond the scope of this deliverable, this feature would likely evolve into a preference that can be controlled directly by manufacturers in their site settings.

5 Moving into Production

For month 30, the key goal was preparing the prototype delivered in M18 for production use. Although the Unified Listing is a key resource, our expectation is that it will mainly be used indirectly through interfaces like the Semantic Alignment Tool and the Shopping and Alerting Aid. So, when we say "production" in this context, we do not mean a service that has to handle millions of requests daily, or one that has to be available year-round with no outages.

In this context, "moving into production" means:

1. The system should be available online at a well-known URL.
2. The approaches used and the specific implementation should be peer reviewed.
3. We should be able to test changes before deploying them to production.
4. We should be able to easily deploy changes to production.
5. We should have good options to scale in the future.
6. We should have backups and a good understanding of how to setup the service and restore from backups if needed.
7. The availability of the hosting server should be monitored and we should have a plan for responding to outages.

5.1 Peer Reviews

The work submitted in Month 18 consisted of tens of thousands of lines of original code. Between Month 18 and Month 30, we broke up the work created to date into cleanly reviewable pieces, and reviewed it in depth. This has involved discussing and explaining underlying approaches as well as meticulously reviewing individual lines of code.

In some cases, the code reviews highlighted best practices in use elsewhere in the GPII that needed to be applied to the Unified Listing. The existing code was refactored to use common code and approaches wherever possible. This included lessons learned from an ongoing high-level review of the overall GPII architecture and technologies being conducted by IBM as part of the APCP grant (see the "Key Links" section for details).

In other cases, the code reviews highlighted "gaps" for which we had no well-understood approach within the larger GPII effort. In these cases, we had longer discussions with the wider GPII architecture team and evolved the work done to date into new libraries. Our goal is to close these gaps not only in this specific project, but in future efforts that build on this work.

As a result of the review process and the work done leading up to Month 30, we now have well-reviewed libraries like the following:

Title	Description
gpii-binder	<p>A library to "bind" model variables to form elements and relay changes between them. A basic requirement for dynamic forms.</p> <p>https://github.com/GPII/gpii-binder</p>
gpii-express	<p>A series of components that allow the use of express 4.x conventions and libraries within the GPII.</p> <p>https://github.com/GPII/gpii-express</p>
gpii-express-user	<p>A series of fluid components that provide a user-management library backed by CouchDB.</p> <p>https://github.com/GPII/gpii-express-user</p>
gpii-handlebars	<p>A series of Fluid components that add support for rendering Handlebars template content within the browser, on server-side pages, and in email messages.</p> <p>https://github.com/GPII/gpii-handlebars</p>
gpii-json-schema	<p>A series of Fluid components that add the ability to validate JSON content against a schema. Used both for browser form input validation as well as to provide clear feedback to integrators using our APIs.</p> <p>https://github.com/GPII/gpii-json-schema</p>
gpii-location-bar-relay	<p>A fluid component to relay model changes to and from the browser history. Adds the ability to bookmark the current state and to navigate between changes. Used with the search in the Unified Listing.</p> <p>https://github.com/GPII/gpii-location-bar-relay</p>
gpii-mail-test	<p>A test mail server used to verify that an outgoing mail message is sent, and to examine the message's contents.</p> <p>https://github.com/GPII/gpii-mail-test</p>

Title	Description
gpii-pouchdb	A library that provides a workalike implementation of the CouchDB REST API. Used to test our views and other approaches used with CouchDB. https://github.com/GPII/gpii-pouchdb
gpii-pouchdb-lucene	A library that integrates a PouchDB instance with couchdb-lucene, so that we can test the search integration used in the Unified Listing. https://github.com/GPII/gpii-pouchdb-lucene
gpii-webdriver	A library that uses the Webdriver API to test code using real-world browsers. Critically, this adds the ability to perform end-to-end tests of our support for keyboard navigation. https://github.com/GPII/gpii-webdriver

Table 2. Libraries reviewed and distributed as part of this project.

5.2 Testing

Good tests are a part of our "agile safety net". They allow us to easily make changes, as we can confirm that all existing functionality is preserved, and that no problems are introduced. Many of the "gaps" identified in the peer review process centered on the way in which code was tested.

5.2.1 Code Coverage

Although we certainly had tests, we did not use code coverage tools in the initial work submitted in Month 18. For this deliverable, we started using Istanbul (<https://github.com/gotwarlost/istanbul>) to prepare code coverage reports. As a result, we were able to identify areas where our testing was not adequate. Although there were cases in which tests simply had not been written, more often we were missing "branch" coverage, failing to test one or more possible paths through the code. In particular, there were not enough tests for "failure modes", unlikely but important cases where a clear error should be thrown. Although it is not always practical to test 100% of the code in a given project, at the end of the review process most of the libraries referenced in this chapter have 95% code coverage or better.

5.2.2 Integration Tests

Another key "gap" was with the test frameworks themselves. Previous approaches within the GPII focused mainly on testing server and browser code in isolation. Server code would be tested from within Node, and browser code would be tested from a web page loaded in a browser. Typically, the browser tests would be loaded using multiple browsers and (in the case of Internet Explorer) browser versions, and the results would be collated into a combined report.

For this project, we needed to test browser components and server components in combination. Depending on the code being tested, we used one of two approaches:

1. Browser tests that interact with server components.
2. Server tests that require browser control and awareness.

5.2.2.1 Browser tests that interact with server components

For cases in which we did not need to interrogate both the server and browser components at the same time, we were able to extend the work done previously within the GPII in using QUnit (<http://qunitjs.com/>) and Testem (<https://github.com/testem/testem>).

Qunit is a unit-testing framework used extensively within the GPII. Testem is a harness that launches multiple browsers and runs one or more browser tests. Testem has the ability to detect running QUnit tests and understand the results, and then to collate those into a combined report that is suitable both for developer review and for use in a continuous integration environment.

In the work done between Month 18 and Month 30, we explored a new approach, in which Testem is configured to launch all server-side components and then run the browser tests. This allows us to run integration tests and also to ensure that our front-end components work across a range of browsers.

5.2.2.2 Server tests that require browser control and awareness

Let's start with an example of the type of end-to-end test we needed to be able to conduct as part of this review. We have a "password reset" process, which involves a verification email. This is an incredibly common feature of web applications that require logins, and involves steps like the following:

1. A user fills out and submits a form within a browser.
2. An email is sent from the server.
3. A unique one-time link contained in the email must be opened with a browser.
4. A second form must be filled out and submitted.
5. The ability to log in with the updated password must be verified.

Previous approaches would have been unable to handle the second step, namely intercepting the email that results from the first step, and then using that information as an input to the next step. Although we had node tests for the APIs involved, we would in essence have to stop our browser tests at the end of step 2 and then start step 3 using predefined test data.

An approach in which we were clicking an actual link generated by the server-side APIs would give us much greater confidence. To test this, we needed to be able to confirm the state of both server and browser components at the same time. On the server side, we needed to confirm that an email was sent, and to extract the link contained in the email. On the browser side, we needed to fill out two separate forms and confirm that the appropriate feedback was displayed at each step.

To make this possible, we built the `gpii-webdriver` library (see Table 2. Libraries reviewed and distributed as part of this project.). The `gpii-webdriver` library uses an approach based on the W3C Webdriver standard (<https://www.w3.org/TR/webdriver/>), which makes it possible to control real-world browsers from an external process, for example, from a test script running in node. We can instruct the browser to perform actions such as navigating to a page, hitting a particular key, or clicking a link. We can also pass a function from the "server side" to the browser via an IPC bridge. This code is executed within the browser, and the results of the code's execution in the browser are passed back to the "server side". This function allows our tests running in node to compare the state of the server with the state of the browser. This combination of features is key in testing use cases like the "password reset" mechanism described above.

5.2.3 Browser navigation

With our previous approach to browser testing, hitting the "back" or "previous" buttons in most browsers would cause the tests to behave erratically. This makes sense, as the tests are in essence running in one particular stage in the browser's history, and navigating away changes that state. The new `gpii-webdriver` library allows us to fully test components that manage information based on the browser's history, such as `gpii-location-bar-relay`, which is used in our search (see Table 2. Libraries reviewed and distributed as part of this project.)

5.2.4 Keyboard navigation

Previous approaches also had limitations when testing keyboard navigation. To meet standards like [Success Criterion 2.1.1 of WCAG 2.0](#), we must confirm that all controls and content are reachable with a keyboard. Previous approaches used within the GPII tested a single component's handling of individual key presses (such as hitting tab or the arrow keys). This gives us confidence about the behavior of individual components, but does not help us

confirm that they work in the type of complex combinations found in a modern web application.

Ideally, what we want is to be able to pass instructions directly to the browser and then examine the results, without artificially making the component under test aware of what we're trying to do. For example, we would like to confirm that hitting tab actually results in the current element losing focus and a new element gaining it, and that a sequence of tab and arrow key presses results in the focus moving to a particular piece of content. The new `gpii-webdriver` library adds exactly this ability, and is critical in ensuring the quality of our work.

5.3 Continuous Integration

After good tests, the second half of our "agile safety net" is continuous integration. Continuous integration servers run the tests in a server environment and notify developers of the results.

A continuous integration server is especially important in complex projects, where multiple developers are working on a number of branches. A continuous integration server can monitor all branches and rerun the tests whenever changes are committed. This provides a clear indication to all parties involved that a particular branch is in a safe state to merge.

A continuous integration server is also key in ensuring that code works beyond a single machine. A single developer's machine may have particular tools installed or particular settings that allow the tests to run. In a continuous integration environment, all of these settings and dependencies must be clearly expressed in a way that the continuous integration server can interpret.

In preparing for this deliverable, we took advantage of the work being done by the team at the OCADU IDRC team, who have created a new "Quality Infrastructure" as part of their work in Prosperity4All. All of the libraries produced in the runup to M30 are now tested continuously using the QI.

Their approach makes use of Vagrant (<https://www.vagrantup.com/>) to provision a new virtual machine for each test run. This has the added benefit of also being usable on development machines, so that a developer can test provisioning locally. We were able to use this mechanism to resolve provisioning problems that would have been much more difficult to troubleshoot using a cloud-only tool like Travis CI.

5.4 Preparing for Scale

The Unified Listing source database and API are not expected to see large daily usage, because most end users will interact more directly with systems like the Shopping and Alerting Aid, the Semantic Alignment Tool, and the Solutions Registry (see Chapter 2 for Ecosystem infrastructure for smart and personalised inclusion and PROSPERITY for ALL stakeholders www.prosperity4all.eu

more details). Each of these systems periodically pulls updates from the Unified Listing, typically on a daily basis. The primary real-time use is by contributors providing updates and new information. That load is expected to be light compared to the usage of the GPII itself.

However, we do have the means to scale as needed. The most important architectural choice in building the Unified Listing is our use of CouchDB (<http://couchdb.apache.org/>). Out of the box, CouchDB supports robust database replication. Although there has been no need to do this yet, we have already discussed the options this functionality gives us.

If the daily usage of services built on top of the Unified Listing grows beyond what we can support with a single CouchDB instance, we have good options to add load balancing across multiple CouchDB instances. If network latency between our current hosting environment and the rest of the world becomes an issue, replication also gives us the option to set up regional mirrors to reduce network latency.

6 Conclusions and Future Work

The Unified Listing is currently available via the web. It provides a reference user interface that can be used to search through its records. It also provides a user interface that allows manufacturers and end users to contribute new products (or updates to existing products). Most importantly, it provides an API that allows other user interfaces to be constructed to present the Unified Listing data to consumers and other users in a variety of ways. The first example is the Cloud4All Semantic Alignment Tool. The second example is the Shopping and Alerting Aid being developed under a separate project in collaboration with Prosperity4All. See the "Key Links" section for more details.

The 2,500+ records currently in the Unified Listing were imported from EASTIN (Assistive Technologies) as part of Cloud4All and GARI (mainstream products) as part of Prosperity4All. For now, we are displaying the provisional results of the initial imports. The provisional records still need to be reviewed for quality and "unified" to organize information duplicated between the databases we federate. There is a working group being put together to review the existing data, and we expect that work to be complete in the next twelve months. Once this work is complete, we will default to displaying only "reviewed" product records to end users.

The Unified Listing is currently not fully integrated with the rest of GPiI. The product information used by the Cloud4All autopersonalisation infrastructure is not yet pulled directly from the Unified Listing. This integration will be completed in the next six months, and will make it possible for updated information on GPiI-compatible products in the Unified Listing to be automatically passed to the Cloud4All autopersonalisation infrastructure.

Finally, we will be integrating the secure payment system outlined in T201.3 with the Unified Listing and creating the openMarketplace. This work is scheduled for completion in month 46.

7 Key Links

One front-end to the Unified Listing is the Cloud4all's Semantic Alignment Tool, which is available at: <http://sat.gpii.net/>

Manufacturers and end users can search the Unified Listing and contribute changes at: <http://ul.gpii.net/>

Although the Shopping and Alerting Aid (SAA) mentioned in this document is not part of Prosperity for All, it builds on work being done here. The work in progress on that tool will appear at: <http://saa.gpii.net/>

The APCP project mentioned in this document is a new project that will build on the work conducted as part of Cloud4All and Prosperity4All:

<http://gpii.net/apcp/>