

H2020-ICT-688712



Project: H2020-ICT-688712

Project Name:

5G Applications and Devices Benchmarking (TRIANGLE)

Deliverable D5.2

Report on App Lab Integration with FIRE testbed

Date of delivery: 24/05/2017

Version: 2.0

Start date of Project: 01/01/2016

Duration: 36 months



Deliverable D5.2

Report on App Lab Integration with FIRE testbed

Project Number:	ICT-688712
Project Name:	5G Applications and Devices Benchmarking
Project Acronym	TRIANGLE

Document Number:	ICT-688712-TRIANGLE/D5.2
Document Title:	Report on App Lab Integration with FIRE testbed
Lead beneficiary:	University College London
Editor(s):	Niall Roche (UCL)
Authors:	Niall Roche (UCL), Pedro Merino (UMA), Alberto Salmerón (UMA) Janie Baños (DEKRA)
Dissemination Level:	PU
Contractual Date of Delivery:	31/08/2016
Work Package Leader:	Niall Roche (UCL)
Status:	Final
Version:	2.0
File Name:	TRIANGLE_Deliverable_5.2v2-FINAL

Abstract

App Lab platform is currently in production at UCL to support real user based testing of mobile applications. PerformNetworks is a FIRE testbed deployed at UMA for experimentation in 4G/5G technologies. TRIANGLE is a new FIRE testbed coming from the extension of PerformNetworks, new components and App Lab platform. This deliverable discusses the potential approaches for the integration of the two platforms and presents the final decision to be implemented.

Keywords

Testing Mobile apps; Testing Portal; Testbed integration



Executive summary

App Lab is a private app store which functions as an app distribution platform for a user community on UCL campus. This allows students and staff to test early versions of software developed by SMEs pre-release on public App Stores. For the TRIANGLE project App Lab plays a part in the testing of apps on humans to determine their feedback on the app and their perceived Quality of Experience in using the app under different network conditions. Physical integration between App Lab and the TRIANGLE Testbed components was discussed during the project proposal stage, but on discussion it was felt to be unnecessary as any testing would not be possible as not all elements of the network path would be controllable and latency would invalidate any benefit from doing this.

A functional integration and lightweight interconnection for exchanging uploaded mobile applications over a VPN connection will however be implemented. This approach will make it possible to take advantages of both platforms, App Lab at UCL, and the FIRE testbed at UMA. Future potential integrations with TRIANGLE Testbed elements have been identified when UCL's testing lab is at a further stage of development later in the project.



Document: ICT-688712-TRIANGLE/D5.2**Date:** 24/05/2017**Dissemination:** PU**Status:** Final**Version:** 2.0

Revision History

Version	Date of Modification	Revision Details
Version 1.0:	31 st August 2016	Initial Deliverable Submission
Version 2.0:	24 th May 2017	Revised submission based on review feedback and progress since original submission



Contents

1	Introduction	1
2	App Lab Overview	2
2.1	App Lab Background.....	2
2.2	User Testing and feedback through App Lab	2
2.3	Exporting Users sessions for use in Triangle.....	3
2.4	Audio and Video application trialing.....	4
2.5	App Lab Architecture.....	4
2.6	App Lab Portal	5
2.7	Installation and operation of App Lab	6
3	Approaches to integrate App Lab in TRIANGLE.....	7
3.1	Deployment of App Lab University Malaga.....	7
3.2	VPS integration with App Lab.....	7
3.3	VPN integration with App Lab.....	7
3.4	App Lab Portal as Testbed UI	7
3.5	Remote App Testing on devices.....	8
3.6	Other uses of App Lab	8
4	Final App Lab - TRIANGLE Testbed Integration.....	9
4.1	Planned use of App Lab.....	9
4.1.1	App Lab to record user interactions for TRIANGLE	9
4.1.2	App Lab using the TRIANGLE Testbed for device testing and validation	9
5	App Lab TRIANGLE Testbed Integration Summary	11
6	References.....	12
7	Annex 1. App Lab iOS Screenshots	13
8	Annex 2. App Lab Portal Screenshots.....	16
9	Annex 3. App Lab Software Architecture	19
10	Annex 4. VELOX VPS Engine	20
11	Annex 5. TRIANGLE REST APIs	22
12	Annex 5. UXCam Data Format.....	24



Document: ICT-688712-TRIANGLE/D5.2

Date: 24/05/2017

Dissemination: PU

Status: Final

Version: 2.0

List of Figures

Figure 1 – UXCam Data Capture example	3
Figure 2 - App Lab High- Level Architecture	5
Figure 3 - App Lab App Submission Process.....	6
Figure 4 - App Lab and TRIANGLE Integration Flow	10



Document: ICT-688712-TRIANGLE/D5.2

Date: 24/05/2017

Dissemination: PU

Status: Final

Version: 2.0

List of Tables



List of Abbreviations

API	Application Programming Interface
E2E	End-to-End
JSON	JavaScript Object Notation
JEE	Java Enterprise Edition
LTE	Long Term Evolution
Mbps	megabits per second
Mo	Month
MA	Multiple Access
MAC	Medium-access Control
QoE	Quality of Experience
QoS	Quality of Service
SLA	Service Level Agreement
SME	Small and medium sized enterprises

TCP	Transmission Control Protocol
UCL	University College London
UE	User Equipment
UL	Uplink
UMA	University of Malaga
UUID	Universally Unique Identifier
UX	User eXperience
VPS	Virtual Path Slice
VPN	Virtual Private Network
WLAN	Wireless Local Area Network
WP	Work Package
WPAN	Wireless Personal Area Networks



Document: ICT-688712-TRIANGLE/D5.2

Date: 24/05/2017

Dissemination: PU

Status: Final

Version: 2.0

1 Introduction

This document provides an overview of App Lab and its relationship to the TRIANGLE Project, in particular how it fits into the overall certification process and the Testbed and its components. The consortium had a number of conversations and ideas about how App Lab can benefit the project, and where it could fit architecturally with the rest of the Testbed Components, both at present, and at later stages of the project. This document outlines the various ideas that were proposed and may be considered at a later stage in the project and plans for the current integration between the functionality of App Lab and the TRIANGLE Testbed.



2 App Lab Overview

This section contains an overview of App Lab [1], what is used for and the benefits that it provides and how it will benefit the TRIANGLE Project and TRIANGLE Testbed application users. Some screenshots of the App Lab web admin portal and the iOS Client application are included in Annex 1.

2.1 App Lab Background

App Lab is a mobile app distribution platform developed by UCL. Advances [2] (now renamed to Innovation and Enterprise), UCL's center for entrepreneurship, which was formed to develop new links between UCL and businesses of all kinds, with a particular focus on entrepreneurs and small and medium sized enterprises (SMEs). UCL Advances ran a number of programs to deliver a unique, in-depth service to help SMEs to test new products and services, gain customer feedback and ensure technical feasibility.

App Lab was one of these projects to create a "living-lab" testbed that provides controlled conditions for SMEs to test and validate their innovations, through access to UCL's 45,000 staff and students. This project supports innovation and early product development in a mobile ecosystem environment.

App Lab was developed as an app testbed and is a high-level platform for distributing applications to a large- scale testbed for pre-commercial testing and validation services. To accomplish this, App Lab provides a private (enterprise) app store, offering in-the-wild user rapid field testing and feedback.

The system is independent of the Apple iOS App Store [3] and Google Play [4], limited to authorized users of the University community, under the Apple Enterprise Agreement for iOS [5], which for universities allows distribution to its students and staff under limited conditions. Google does not provide restrictions for their App Distribution.

2.2 User Testing and feedback through App Lab

In addition to app distribution, App Lab collects valuable data for evaluation to provide fast iteration of releases for app improvement cycles. The data is useful to see how users are engaging with the applications, what features are used and which ones are not, and how user journeys occur with real users. App Developers can use this data to tweak elements such as the User interface, or to try different experiments such as A/B Testing [7] to see if changes have an effect on users in a subsequent release.

App Lab uses a third-party tool called UXCam [6] to capture data about how users interact with the applications. UCL is negotiating an agreement with UXCam to allow free usage for the consortium members and the experimenters that will participate in TRIANGLE through the Open Call mechanism.

UXCam allows visualization of hotspots where users are interacting with the UI and the entire user journey can be played back as a set of animations from a web console. This is very useful for SMEs to visualize how real users are interacting with the applications, as opposed to how they think users should interact with their applications. This often leads to discovering usability issues and necessitating rework of the User Experience.

In the context of TRIANGLE UXCam can be used as a data source for extracting user behavior models or single specific behavior for scripting the App testing tools available in the testbed.



UCL is also in discussion with other analytics tracking providers, one of whom, TestFairy [9], may also be useful for tracking purposes. One of the important aspects of the data captured is the meta data capture and looking at how this data can be used to represent user journeys for the purposes of automating common or problematic user journeys for testing and validation purposes.

2.3 Exporting Users sessions for use in Triangle

UCL and Quamotion [12] have been investigating the feasibility of using data captured from UXCam for the purposes of creating user journeys based on real user interactions. The data captured by UXCam is in a proprietary format, but can now be exposed via an API. This API has only become available in May 2017, so this was not available during the original version of this document. There are some limitations with the data captured that mean that user interface control ids are not always present and can vary from each Android application build. This would mean a new set of recordings would be needed for each application build. An example of a subset of the output from a UXCam session for a radio button click is provided below.

```
"screenActions": [{  
  "actualClass": "RadioButton",  
  "t": 0,  
  "v": "on",  
  "a": "toggled",  
  "n": "New RadioButton",  
  "i": "2131558532",  
  "h": 96,  
  "w": 421,  
  "y": 1584,  
  "x": 295,  
  "e": 3  
}]
```

Figure 1 – UXCam Data Capture example

Similar to Quamotion, when frameworks such as Unity [12] are used to build the application UXCam cannot identify individual control ids. Where control id identification is not possible, x&y coordinates can be used, provided the capture and playback devices have the same screen size and resolution, or an appropriate scale factor can be computed and adjusted. Quamotion UCL and UMA are in discussion with UXCam to identify strategies for managing the automatic creation of test cases in a WebDriver compatible format that would be suitable for playing back scenarios in Quamotion. Appendix 12 contains a more detailed overview of the data format exported from UXCam.



2.4 Audio and Video application trialing

App Lab collects valuable data for evaluation fast iteration of releases for app improvement cycles and includes audio and video capability as part of the government sponsored Innovate UK Digital Testbed. UCL have signed an agreement with Innovate UK [9] to use their Digital Testbed trial management software solution to facilitate trials of multimedia content on the student body, to determine commercial viability and user feedback for different content and delivery and charging models.

UCL and Innovate UK believe this Digital Testbed facility will allow Multimedia Content Providers to bring high value content for use by the test community and provide a strong justification for 5G technologies for high bandwidth and potentially use of the content in non-traditional delivery methods enabled via 5G, such as Augmented and Virtual Reality. This could be a great asset for some of the SMEs that will use the TRIANGLE Testbed, in particular as they try to find out how to leverage the opportunities to leverage the new technologies and presentation methods in a way that is appealing to users.

2.5 App Lab Architecture

App Lab is comprised of client applications for Android and iOS mobile devices and a JEE server portal for application upload, management and distribution. The platform has a Web Management Console that carries out the management tasks of the store including:

- upload of new applications or updates (applications developed by the company, public applications or purchased applications),
- adding meta data such as app descriptions and screenshots,
- test case definitions for testing, approval and publication of applications which will can be adapted for the TRIANGLE pilot cases,
- workflow of app approval lifecycle including reviewer and tester comments,
- managing multiple app versions as new releases are live and in testing,
- targeting app variants for specific device operating system versions, useful when testing apps for beta releases (such as iOS 10 at the time of writing),
- client software for the disabling of apps should there be an issue with the app or app content.



2.6 App Lab Portal

The App Lab Portal supports the lifecycle from app submission, review & testing and publishing. Figure 1 below shows the flow for submitting apps into App Lab. Screenshots of the App Lab Portal can be seen in Annex 2. Initially there was some thoughts given by the TRIANGLE Consortium to adapt the App Lab Portal for use by the end user TRIANGLE App and Device Submission Portal, however, after some discussion and evaluation, it was felt that this was probably not the best course of action as explained further in Section 3.1.

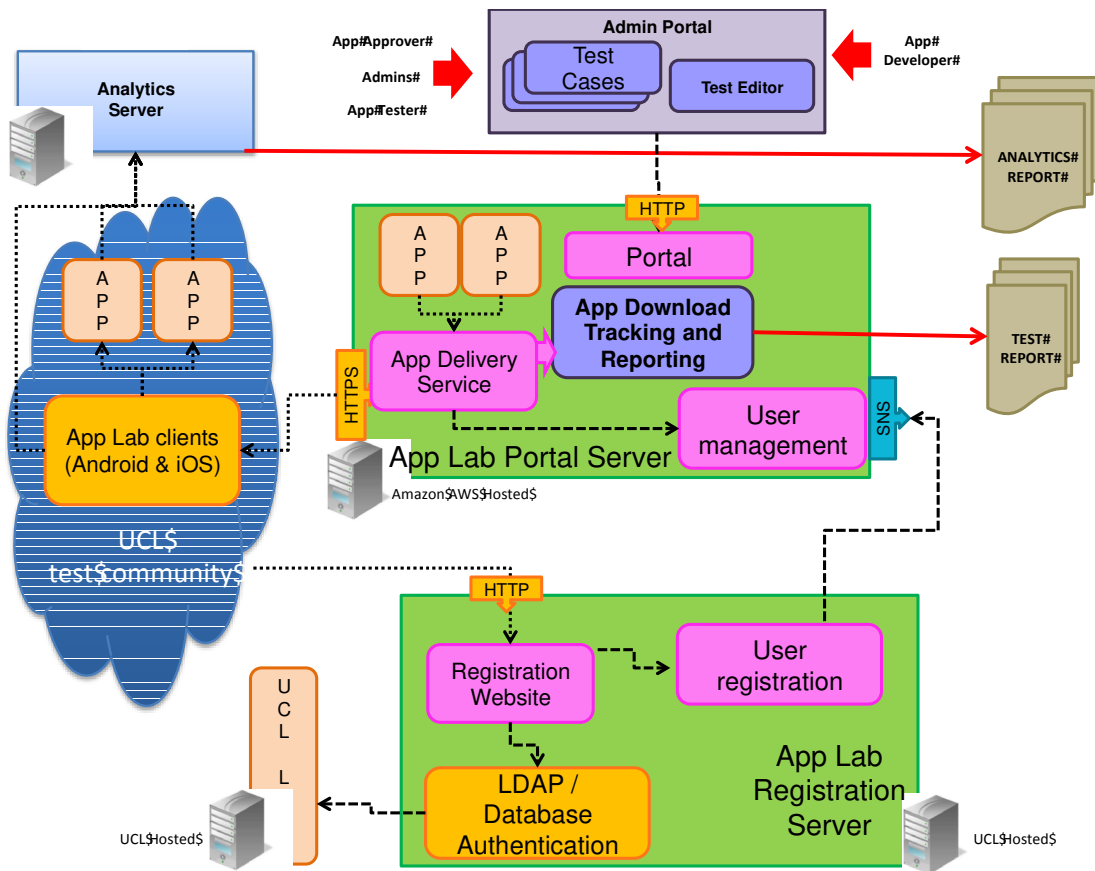


Figure 2 - App Lab High- Level Architecture

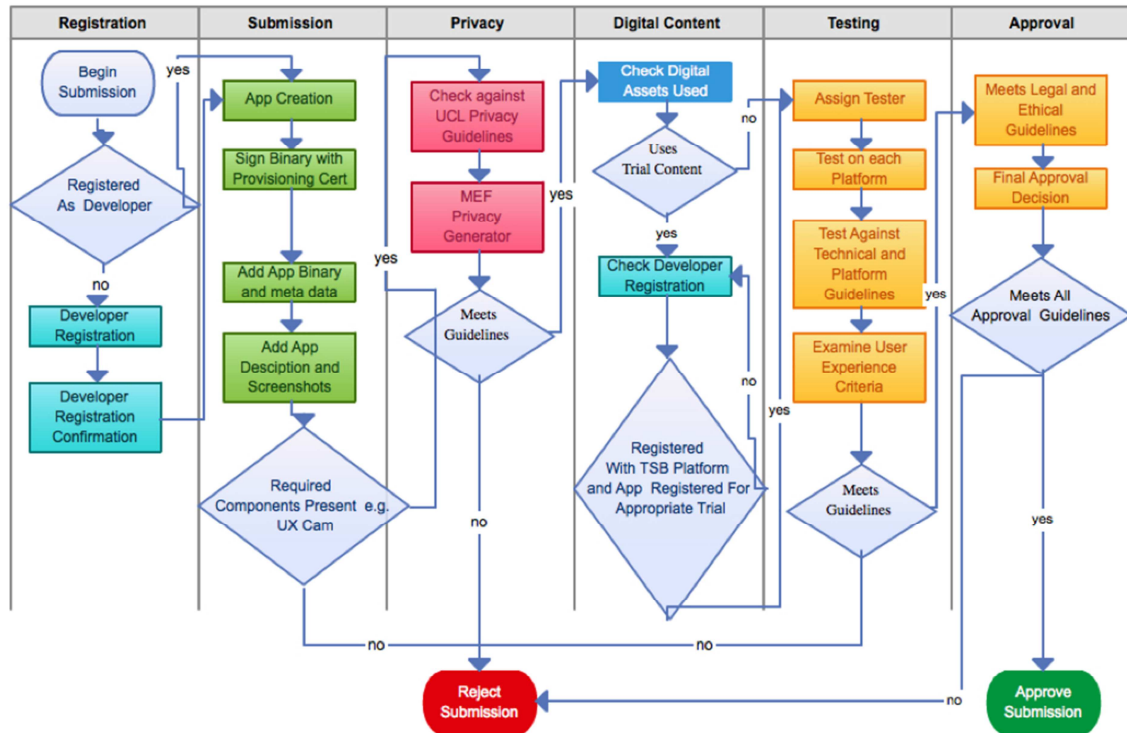


Figure 3 - App Lab App Submission Process

2.7 Installation and operation of App Lab

App Lab requires integration with an authentication system to identify users; for a university deployment the users are staff and students, for a corporate, these are the company employees. App Lab can use two authentication mechanisms, a database storing user information, or using LDAP. In UCL, App Lab integrates with the University's LDAP authentication mechanism to identify if users are part of the University. This is important as it is necessary to stop users who are not members of the University from accessing the system. This is an important point as distribution iOS apps outside of the normal iOS Apple App Store has restrictions on use and the Agreement between Apple and Universities covers distribution of apps strictly to those within the staff and students of the University.

In order to integrate with the LDAP authentication mechanism, App Lab needs to be hosted within the University network as external servers are not permitted to access the LDAP servers directly, for security reasons. This makes the installation of App Lab outside of a University network more difficult in practice, as it would not be able to easily authenticate users, without the export of large amounts of user data outside of the University controlled environment, which violates the data protection requirements of the University.



3 Approaches to integrate App Lab in TRIANGLE

The consortium has evaluated several options to integrate App Lab with the current testbed in Malaga as part of TRIANGLE.

3.1 Deployment of App Lab University Malaga

One option that was considered would be a deployment of App Lab inside the University of Malaga with the student population. However, it was decided that an App Lab deployment would be of limited value without a sizeable user Testbed population. This would normally involve a large promotion activity to engage the student and staff bodies to download App Lab and also integration with UMA authentication mechanisms. For these reasons, there are no plans to install App Lab at UMA presently.

3.2 VPS integration with App Lab

This was an early consideration and still one that we may explore later in the project. The idea was that individual apps deployed on App Lab would have a dedicated virtual path on the network with a SLA for certain bandwidth and network performance capabilities. This would use the VPS engine component from RedZinc, integrated with App Lab. In practice however, deploying an App in an environment where there is no real control over the underlying network as users in the UCL network would connect using commercial operators and campus-wide Wi-Fi networks. Therefore, implementing any level of path slicing is practically impossible, as these networks would not be capable of being configured.

In the medium term UCL is working on creating a controllable network environment for Wi-Fi and LTE environments where it may be possible to have some type of path slice implementation. Once that is in place, it may be reasonable to revisit the VPS integration.

Details of the API that applications can use to integrate on the UMA Testbed and request quality of service are given in Annex 4.

3.3 VPN integration with App Lab

Another option considered was a VPN connection between the App Lab installation at UCL and the Malaga Testbed. This was considered for a number of reasons including:

Enabling interconnection between apps distributed from App Lab and running in London and the Testbed in UMA. As mentioned in the previous section, this was not practical due to the latency involved and the lack of control of the underlying network.

There will still be a VPN connection as App Lab and the UMA Testbed components for the purposes of transferring uploaded apps from App Lab and into the Testbed and vice versa. This process and the reason for needing this is described in Section 3.7.1

3.4 App Lab Portal as Testbed UI

One of the early ideas for a potential integration was to utilize many of the existing function of App Lab around managing app submission, upload testing distribution and approval. These high-level functions had some perceived overlap with the main TRIANGLE Portal, but on more detailed discussion it was apparent that there were a number of differences between the aims and requirements of the two portals. Also, the App Lab portal while quite flexible in regards to creating new kinds of tests to run as part of an app review and approval process, is not very flexible for workflow changes, though this is possible if needed.

At the London meeting in May 2016 it was decided by the consortium to proceed with developing a new web portal as a basis for the main TRIANGLE Portal as it would better suit the requirements that were identified, rather than involve an additional system.



The TRIANGLE Portal is the Testbed front-end for end-users, offering a simple web interface that hides the complexities of the underlying Testbed capabilities. App developers upload their Apps to the TRIANGLE Portal in order to perform experimentation and certification test campaigns. The Portal Backend stores the Apps uploaded by its users, the declaration of the App features and capabilities, and the definitions of the test campaigns to be performed on the Testbed. The Backend and the Testbed orchestration each offer a REST API, which are used when the user runs a test campaign from the Portal. These APIs are described in Annex 5.

One possible integration point between the App Lab Portal and the TRIANGLE Portal/Testbed would be through these REST APIs, so that App Lab users could run test campaigns from the same website. However, this would require significant work in the App Lab Portal, both to implement the required REST API, and to adapt the UI to show TRIANGLE related functionality. Again, this integration has been deemed too invasive for App Lab, as each Portal offers a different UI better suited for their use case. Later we will explore a more lightweight integration between the two Portals.

3.5 Remote App Testing on devices

One idea that was explored was investigating whether it would be useful for testing Apps that are running remotely on devices in the UMA testbed, from external devices in London or elsewhere. This is technically possible but in practice of limited use. It may be useful to see what an app looks like on another device, from the form factor of a mobile device, as opposed to a mobile browser as is the usual case. However, to get an accurate feel of how the app performs and a realistic experience of response time, it is not of use due to the overhead and latency involved. Nevertheless, this remains a future option if a suitable use case is identified for doing it. One potential use is recording the user actions when the app is interacting with components only deployed at UMA location.

3.6 Other uses of App Lab

App Lab may still prove to be useful for the Deployment of apps inside the UMA Testbed as it is a distribution mechanism and performs features such as enterprise resigning that are likely to be useful in deploying apps in an automated fashion to the Testbed devices and also to small user groups for testing, this may help with obtaining Mean Opinion Scores (MOS) for determining QoE.

As outlined in 2.3 App Lab, in conjunction with UXCam, can help to get useful feedback on QoE of applications and also to record and playback user sessions to find out more information about how the app was used. This can be of further benefit by using this usage information to generate test cases that can be automated, and are based on real-world usage of the app by actual users.



4 Final App Lab - TRIANGLE Testbed Integration

4.1 Planned use of App Lab

At the kick-off meeting in London in May 2016 it was decided that the main value of App Lab to the TRIANGLE project was in the user community and how real users actually use applications and their perceptions on how the apps performed. Real users are also beneficial where automated testing is not practical, so where automation is not possible certain tasks could be performed on apps by real users. As it was not practical to measure network behavior on apps running outside of the UCL testbed in London, there was still value in capturing information of the steps the user took inside the running application.

As a result, the consortium has decided to integrate App Lab in the following ways:

- To record user interactions with the apps in order to produce user app flows as input to the experimentation and certification process in TRIANGLE testbed
- To use TRIANGLE in order to perform additional testing and validation of the apps submitted for analysis to App Lab

4.1.1 App Lab to record user interactions for TRIANGLE

One benefit of App Lab is to use the information recorded by UXCam on how users have used the app and to gather that data and to use it as a mechanism to create automated scripts that will execute on Devices in the TRIANGLE Testbed in UMA. This is important, as sometimes app developers will not already have automated test scripts created when they submit apps to TRIANGLE for certification, or they may have automated scripts that may not be compatible with the Quamotion system or the popular WebDriver protocol.

In this case, the data captured from UXCam could be used to generate test scripts capable of being transformed into models that are executed in the Quamotion automation platform. Even where automated tests exist, these normally test the expected path through an app. In practice, these flows often differ from how real users interact with apps and also experience realistic delays in between steps as user's pause to look at screen options and decide what action to take next. One planned task is to use the recorded data from UXCam to build realistic models of the user interaction in order to automatically generate the scripts.

4.1.2 App Lab using the TRIANGLE Testbed for device testing and validation

One immediate use of the Testbed and App Lab is to perform device and functional testing in a number of characteristics of apps that are submitted to App Lab. In the process flow illustrated below, in Figure 4, it can be seen that users who submit their apps to App Lab for use on the UCL user community can utilise the TRIANGLE Testbed for verifying that the app functions as expected in different network and device configurations.

This is a logical connection between the two Portals. App Lab Portal users would get additional credentials to use the TRIANGLE Portal to perform additional testing. However, no direct integration would be done between the backends of both Portals, i.e. the apps uploaded to each Portal, and the data generated in both Testbeds, would be physically separated, and accessible through the respective Portals. The initial step of uploading an App from the App Lab Portal into the TRIANGLE Portal could be automated in the future, requiring integration of the authentication schemes of both Portals, but it is not considered for the initial integration.

This is a good pilot of the Benchmarking as a service BaaS concept that has been outlined in the TRIANGLE Deliverable '5G BaaS – Benchmark as a Service'. As other systems may want to integrate the TRIANGLE Testbed as part of their testing and validation or possibly certification processes.

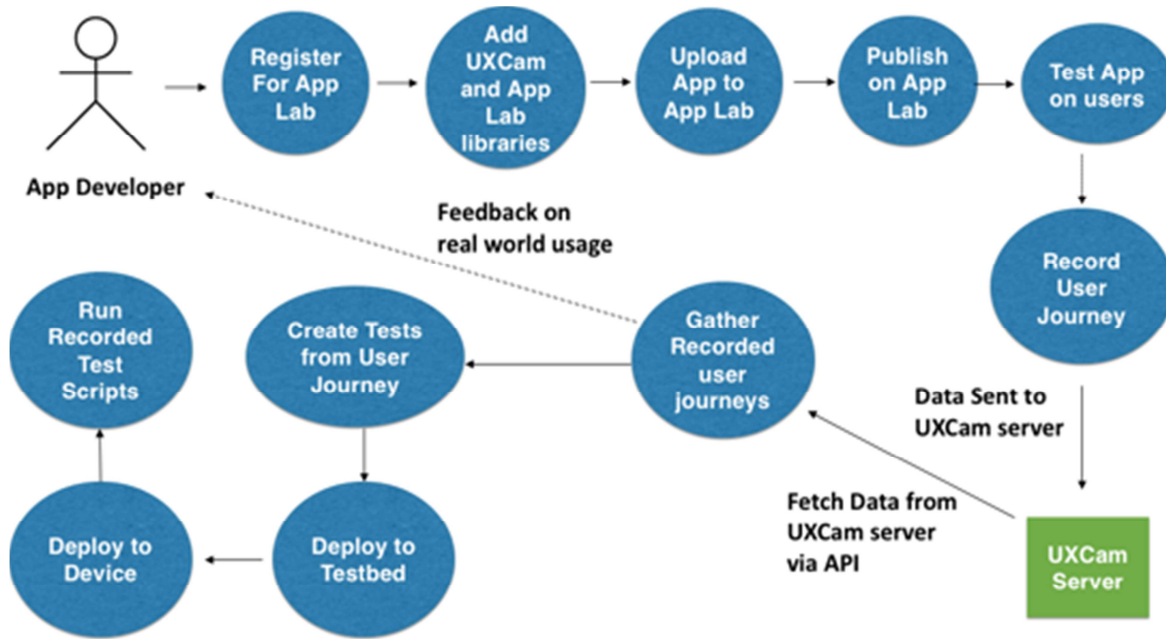


Figure 4 - App Lab and TRIANGLE Integration Flow

In addition, recordings generated from UXCam as described in the previous section could be used in the TRIANGLE Portal, to perform test campaigns with these realistic user interactions.

The flow steps of the integration between App Lab and TRIANGLE would be as follows. These steps could be repeated by the user as necessary, to perform additional test cycles and refine the App.

1. App developer uploads app to App Lab
2. Creation of app user flows in App Lab
 - a. App distributed to real users in App Lab
 - b. User journeys recorded with UXCam from real users
 - c. App model created from recorded user journeys
3. Advanced testing on TRIANGLE Testbed
 - a. User uploads app to TRIANGLE Portal
 - b. User uploads app model from recorded user journeys [optional]
 - c. User configures test campaign: reference handset, network scenarios, etc.
 - d. Test campaign is executed in lab
 - e. User get campaign results; relevant KPIs, metrics computed; certification
4. Deployment onto App Lab for human feedback, HCI testing



5 App Lab TRIANGLE Testbed Integration Summary

An overview of App Lab and the benefits of the platform were outlined.

The App Lab TRIANGLE Testbed integration proposal has been presented as the process to use both TRIANGLE and UCL App Lab testbeds with the same app, but limited physical integration is expected to facilitate secure app transfer between App Lab and the TRIANGLE Portal as outlined in Section 4.

Some aspects of App Lab for iOS distribution may be used by Testbed as part of app distribution and to assist with human interaction with determining QoE.

Apps on App Lab may use a controllable Wi-Fi and LTE network at UCL that would justify a VPS engine integration to allow more accurate device testing under different network conditions and to examine user's reaction to the UX and provide feedback to App Developers.

No deployment in Malaga is expected at this time as a physical installation does not make sense for a number of reasons illustrated in Section 3.1.

App Lab will use the services of the Testbed for some parts of the App Testing flow.

The key point is that real user use of applications captured by the UXCam tool used by App Lab will be used to track user behaviour that can be used to potentially create accurate user journeys that can be automated. We are currently investigating the feasibility of automatic data extraction from the UXCam feeds and translation into test scripts to be executed on devices in the UMA testbed. This may prove to be as good as, if not better than, any existing test cases that may have been supplied by the app developers.

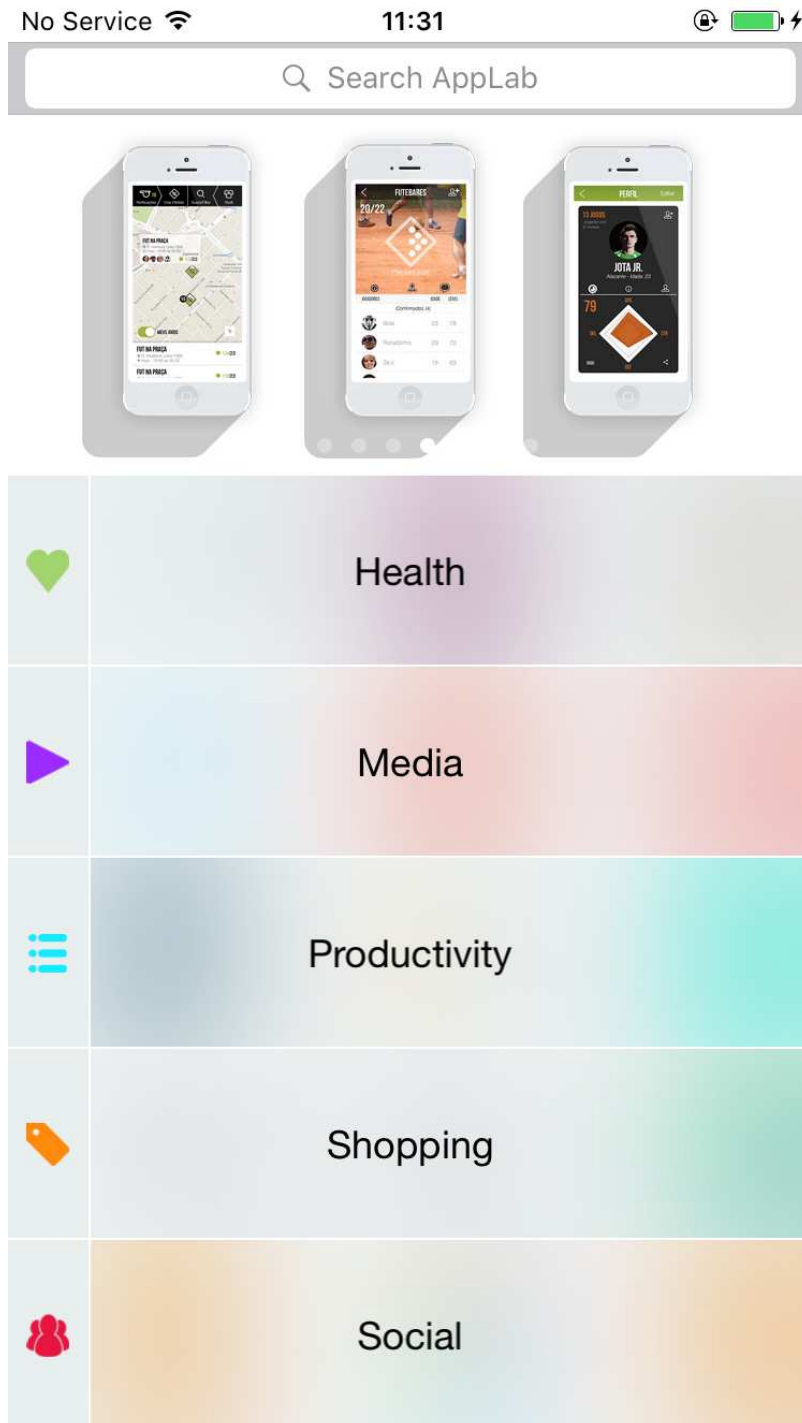


6 References

- [1] App Lab website <https://applab.ucl.ac.uk/>
- [2] UCL Innovation and Enterprise (formerly UCL Advances) <http://www.ucl.ac.uk/enterprise>
- [3] Apple iOS App Store <http://www.apple.com/itunes/>
- [4] Google Android Play Store <https://play.google.com/store/apps?hl=en>
- [5] Apple Developer Enterprise Program <https://developer.apple.com/programs/enterprise/>
- [6] UXCam User Experience improvement tool <http://uxcam.com/>
- [7] A/B Testing Wikipedia Entry https://en.wikipedia.org/wiki/A/B_testing
- [8] Quamotion mobile app test automation website <http://mwc.Quamotion.mobi/>
- [9] TestFairy iOS and Android Beta Testing solution <https://testfairy.com/>
- [10] Innovate UK <https://www.gov.uk/government/organisations/innovate-uk>
- [11] LabWiki <https://github.com/mytestbed/labwiki>
- [12] Unity <https://unity3d.com/>



7 Annex 1. App Lab iOS Screenshots





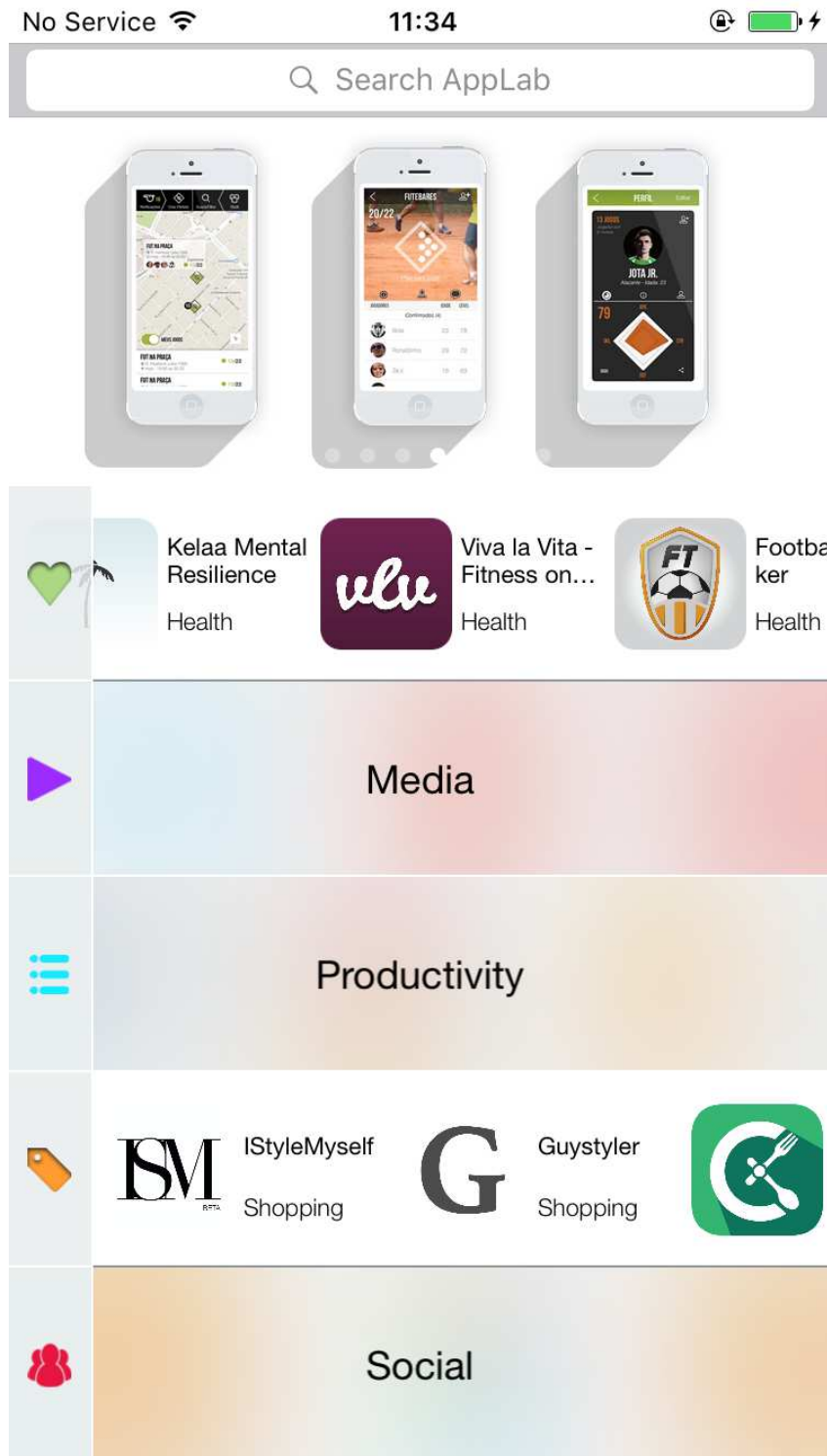
Document: ICT-688712-TRIANGLE/D5.2

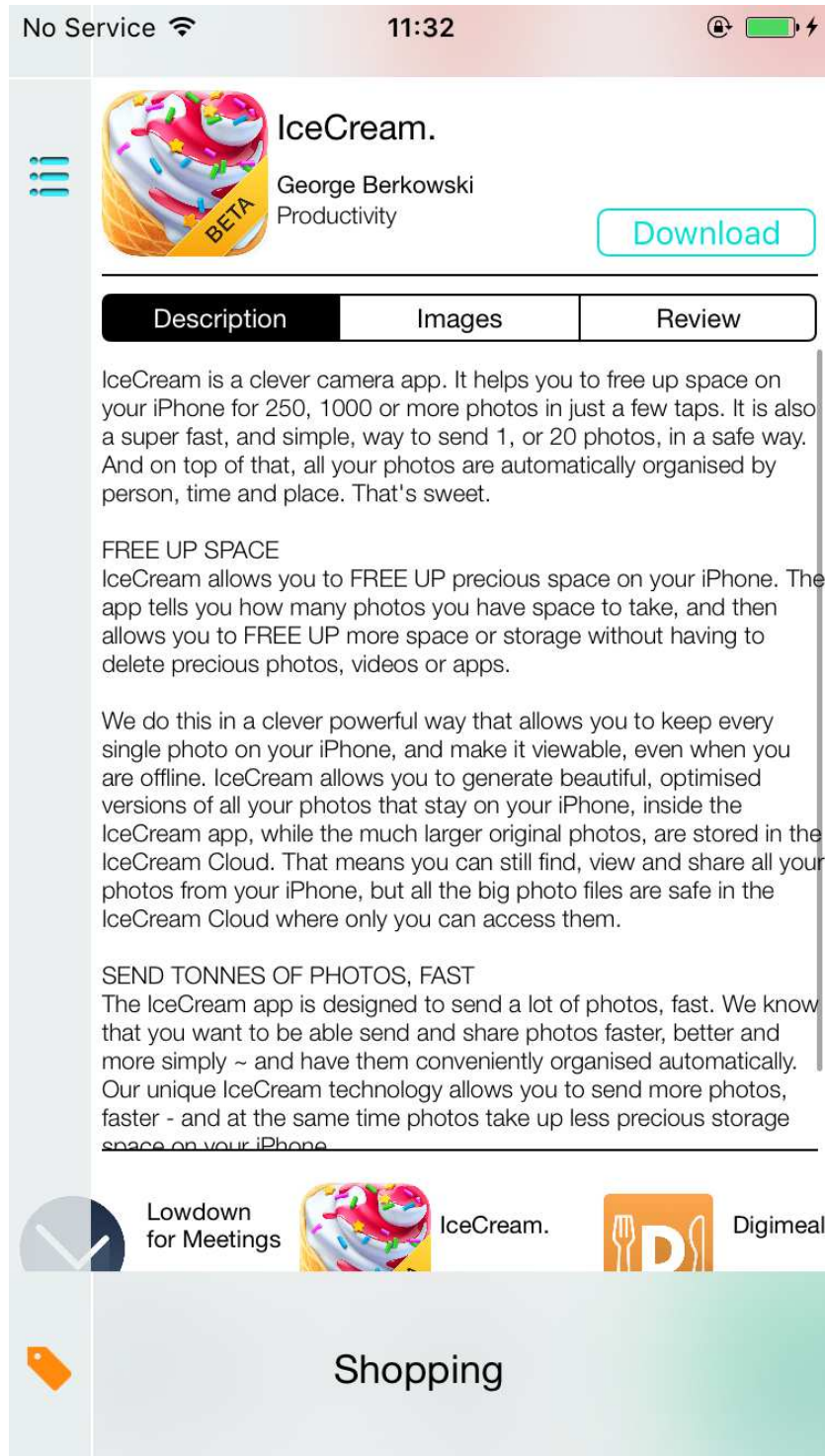
Date: 24/05/2017

Dissemination: PU

Status: Final

Version: 2.0







8 Annex 2. App Lab Portal Screenshots

The screenshot displays the App Lab Portal interface. At the top, there is a navigation bar with links: Users, Apps, Add App, Approve for testing, Approve for publication, Create Test, and Tests. Below this, a user status bar shows 'User: UCL Decide', 'Developer Docs', and 'Log out'.

The main content area is divided into two sections:

- Not ready for approval:** This section contains two app entries. Each entry shows an app icon, name, category, a 5-star rating, '(0) Comments', a 'Description' link, a 'Binaries' link (showing '0 Active Installs'), and a 'Media' link.
- Waiting for approval:** This section contains five app entries. Each entry shows an app icon, name, category, a 5-star rating, '(0) Comments', a 'Description' link, a 'Binaries' link (showing '0 Active Installs' or '17 Active Installs' with a 'Published' status), and a 'Media' link.

At the bottom of the page, there is a pagination bar showing '3 / 9' and a copyright notice '© UCL 2014'.



Document: ICT-688712-TRIANGLE/D5.2

Date: 24/05/2017

Dissemination: PU

Status: Final

Version: 2.0

Screenshot of the App Lab interface showing the 'Approve for publication' screen. The browser address bar shows the URL: <https://applab.ucl.ac.uk/atosmarket-1.0.9/approver/pendingApps.zul>.

The interface includes a navigation bar with the following links: Users, Apps, Add App, Approve for testing, Approve for publication (active), Create Test, and Tests. Below the navigation bar, there are buttons for 'User: UCL Decide', 'Developer Done', and 'Log out'.

The main content area displays a table titled 'Approve for publication' with the following columns: App, Test, Finished tests, Trials, and Reject / Approve.

App	Test	Finished tests	Trials	Reject / Approve
UCL Audio... UCL Apps	2 Total test	27/09/2014 0 test(s) failed	No Trials	<input type="button" value="✖ Reject"/> <input type="button" value="✔ Approve for publication"/>
UCL Go! UCL Apps	2 Total test	27/09/2014 0 test(s) failed	No Trials	<input type="button" value="✖ Reject"/> <input type="button" value="✔ Approve for publication"/>

© UCL 2014



Document: ICT-688712-TRIANGLE/D5.2

Date: 24/05/2017

Dissemination: PU

Status: Final

Version: 2.0

https://applab.ucl.ac.uk/atosmarket-1.0.9/approver/assignTests.zul

App Lab
Delivered by UCL

Users Apps Add App Approve for testing Approve for publication Create Test Tests

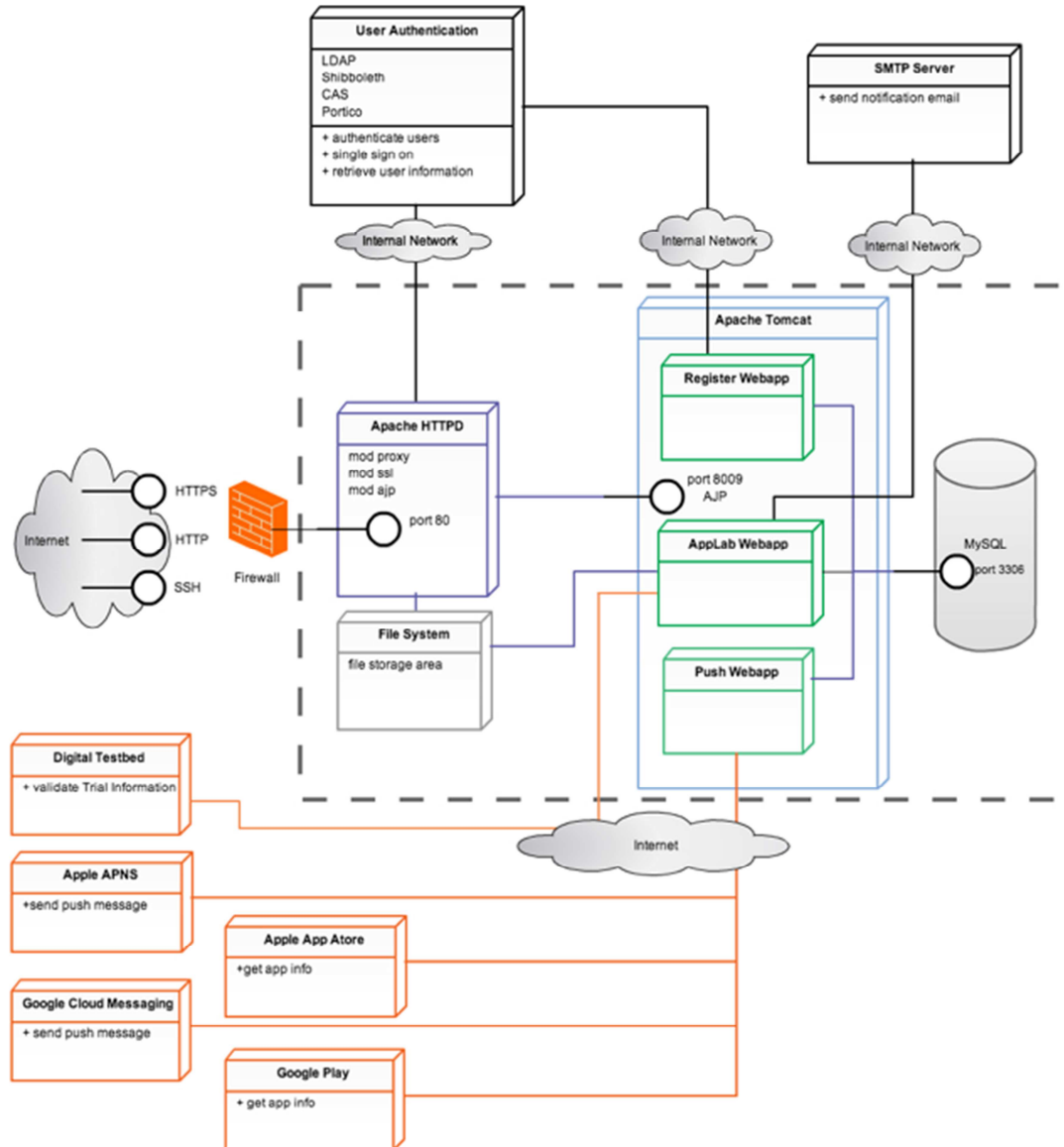
User: UCL Decide Developer Docs Log out

Approve for testing

Select tests groups Drag and drop over the version to add test case group to version	Versions Click on the arrow to see the assignment	Assign testers Drag and drop over the test case group dropped
Android Technical Test ▶	flashbackr 0.0.1 (iPhone) ▶ <input type="checkbox"/> Reject	Ann Chan
Android UX & UI Test ▶	rendeevoo 2.0.1 (iPhone) ▼ <input type="checkbox"/> Reject <input checked="" type="checkbox"/> Start test	Daniel Fozzati
IOS Technical Test ▶	IOS Technica... Niall Roche...	John doe
IOS UX & UI Test ▶		Jules Spa
		Niall
		Niall Roche
		Nicolas Arrellaga

© UCL 2014

9 Annex 3. App Lab Software Architecture





10 Annex 4. VELOX VPS Engine

The VELOX VPS Engine API provides developers with a set of resources that allow the listing and control of VPS services. The API is a set of JSON encoded requests and responses sent over a TCP connection. All service availability is bound to the API Key used.

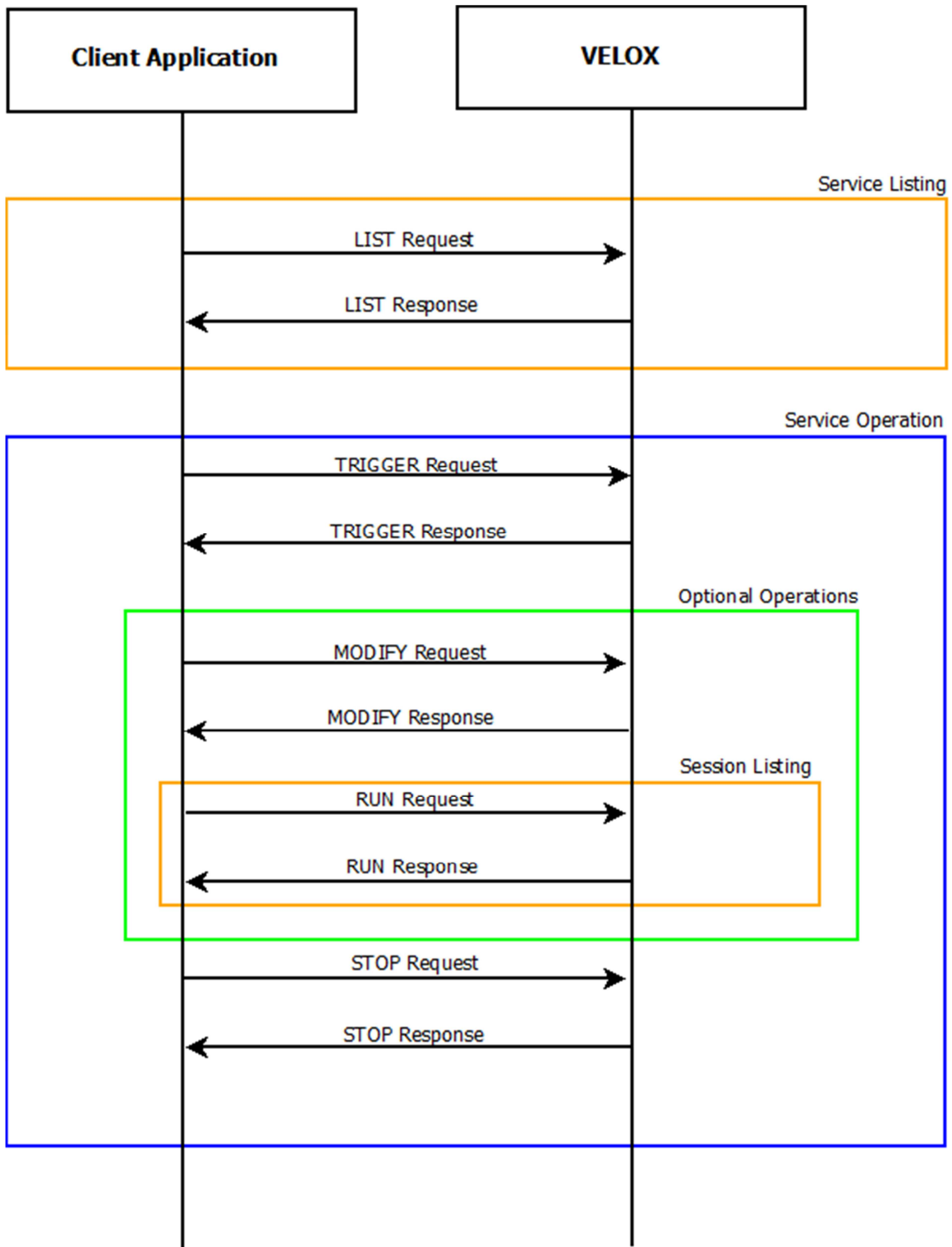
In order to use the VELOX API an application must:

1. Create a TCP connection to known IP address/port (provided by local operator)
2. Write Request (as a single text line, new line ends a request)
3. Read Response (sent as a single line) Connections are terminated on the VELOX side after sending the response All Requests must use the API key generated by the local operator VELOX.

In this scenario, the client application is considered to be any application that uses the VELOX API to access VELOX services.

Modify and Run requests are considered optional since not all circumstances will require their use.

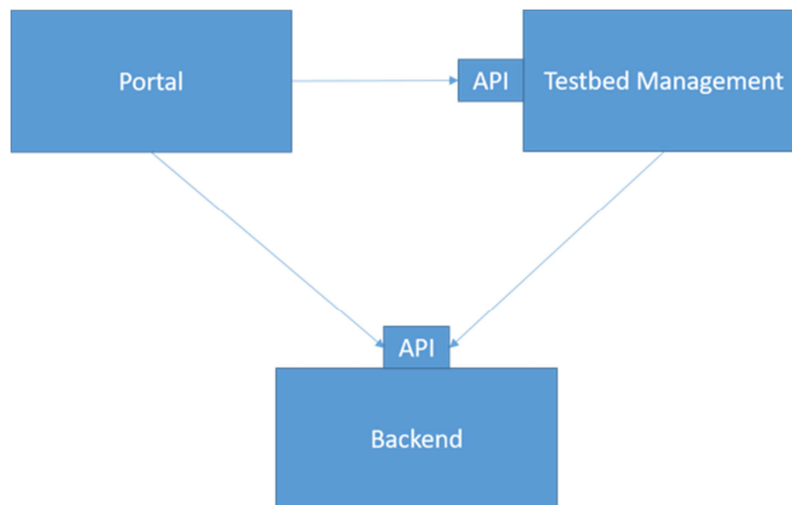
While the List request is the first to be executed it is not mandatory before every Trigger request, the use of the List request should be done as deemed necessary in order to check for any service library changes.





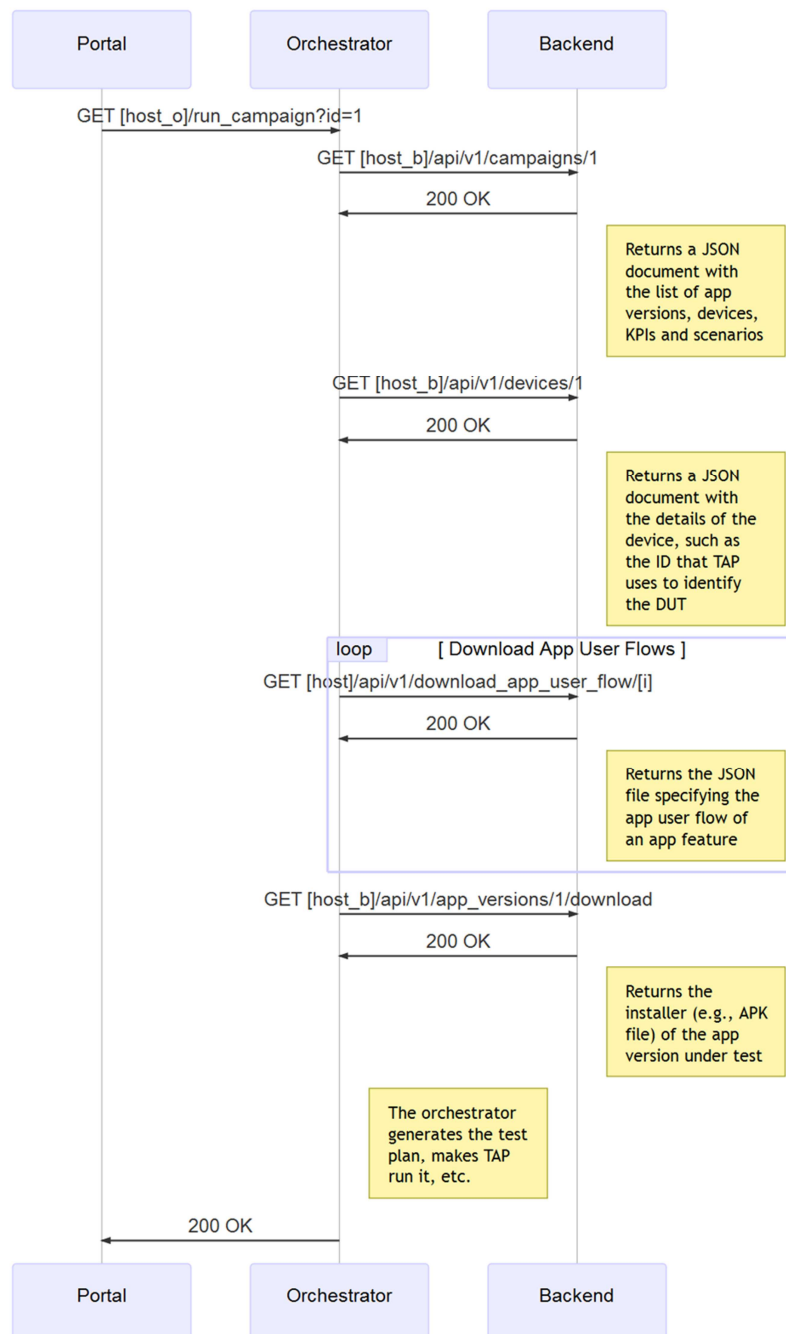
11 Annex 5. TRIANGLE REST APIs

The TRIANGLE Portal and Testbed are distributed and integrated through REST APIs. The Portal is the website for end-users. Data uploaded by users and generated by test campaigns is stored in the Portal Backend. This data is accessible through a REST API. The Testbed also offers a simple REST API that the Portal uses to run test campaigns. In turn, the Testbed uses the Backend REST API to fetch the details of the campaign to be executed, such as the App file, app user flows, and selected network scenarios.



The following sequence diagram shows an example of interaction between the Portal, the Backend and the Testbed orchestrator, to run a test campaign. The interaction is initiated by the Portal, which asks the Testbed to run a campaign. This first request only contains the ID of the campaign to be executed, so the Testbed has to fetch the details from that campaign from the Backend. This is done in a series of requests from the Testbed to the Portal.

In the first request, the Testbed asks for a description of the campaign, including the apps, devices and scenarios to use in the campaign. Then, the Testbed performs additional requests to get the details for each device, the required app user flows, and app files. Once the Testbed has fetched all the details and prepared the campaign to be executed, it returns a success code to the Portal.



There are additional interactions between these three components not shown in this example. For instance, once the test campaign has finished its execution, the Testbed informs the Backend of the result, and uploads any generated measurement data.



12 Annex 5. UXCam Data Format

```
{
  "data": [{
    "_id": {
      "$id": "582025d6c4b502ffff19a998"
    },
    "app_id": "582025d5c4b502ffff19a996",
    "device_id": "d9d04c073d709d2c",
    "device_type": "Phone",
    "os_lname": " M",
    "width": 1080,
    "height": 1920,
    "resolution": 480,
    "crashLog": "",
    "totalTime": 5,
    "isCrash": 0,
    "device_name": "bishwash@gmail.com",
    "isFavorite": 1,
    "deviceModelName": "ONE A2003",
    "date": {
      "sec": 1478501820,
      "usec": 0
    },
    "ymd": "2016-11-07",
    "timestamp": 1478501856,
    "firstActivity": "MainActivity",
    "company_name": "OnePlus",
    "device_run": 1,
    "device_run_name": "Session 1",
    "app_version": "2.0",
    "sdkVersion": "AN.2.5.1",
    "device_run_notes": "",
    "os_name": "6.0.1",
    "os_name_float": 6,
    "events": [{
      "params": {
        "UXCam":
          "uxc.am/s/582025d6c4b502ffff19a997"
        Session
        Recording
        link":
      },
      "time": 0,
      "tag": "verification_success"
    }],
  ]
}
```




```
"unplayed": 0,
"deviceClass": "Android phone Large",
"screenvideo": {
  "bucket": "statictest.uxcam.com",
  "key":
"57a2c2a78968c0f01f7b06a4/582025d5c4b502ffff19a996/582025d6c4b502ffff19a998",
  "file_name": "screen.mp4"
},
"tags": [],
"deviceLocation": {
  "country_name": "United States",
  "country_code": "us"
},
"preferences": {
  "camera": {
    "visible": true,
    "position": {
      "top": null,
      "left": null
    }
  },
  "sharing": {
    "status": 1,
    "links": []
  }
},
"usedAppversion": "v3",
"heatmapTimeline": [{
  "ref": "Launch",
  "an": "MainActivity",
  "vt": 5,
  "at": "0",
  "cor": [
    [
      0,
      0,
      1,
      10,
      1,
      0.023, []
    ],
    [
      363,
```



Document: ICT-688712-TRIANGLE/D5.2

Date: 24/05/2017

Dissemination: PU

Status: Final

Version: 2.0

```
        1718,  
        1,  
        1,  
        1,  
        0.953, []  
    ],  
    [  
        387,  
        1673,  
        1,  
        1,  
        1,  
        1.285, []  
    ],  
    [  
        381,  
        1699,  
        1,  
        1,  
        1,  
        1.619, []  
    ],  
    [  
        383,  
        1691,  
        1,  
        1,  
        1,  
        1.926, []  
    ],  
    [  
        410,  
        1676,  
        1,  
        0,  
        1,  
        2.384, []  
    ]  
],  
"screenActions": [{  
    "actualClass": "RadioButton",  
    "t": 0,  
    "v": "on",
```



```
        "a": "toggled",
        "n": "New RadioButton",
        "i": "2131558532",
        "h": 96,
        "w": 421,
        "y": 1584,
        "x": 295,
        "e": 3
    }, {
        "actualClass": "RadioButton",
        "t": 0,
        "v": "on",
        "a": "toggled",
        "n": "New RadioButton",
        "i": "2131558532",
        "h": 96,
        "w": 421,
        "y": 1584,
        "x": 295,
        "e": 3
    }, {
        "actualClass": "RadioButton",
        "t": 1,
        "v": "on",
        "a": "toggled",
        "n": "New RadioButton",
        "i": "2131558532",
        "h": 96,
        "w": 421,
        "y": 1584,
        "x": 295,
        "e": 3
    }, {
        "actualClass": "RadioButton",
        "t": 1,
        "v": "on",
        "a": "toggled",
        "n": "New RadioButton",
        "i": "2131558532",
        "h": 96,
        "w": 421,
        "y": 1584,
        "x": 295,
```



```
        "e": 3
    }, {
        "actualClass": "RadioButton",
        "t": 1,
        "v": "on",
        "a": "toggled",
        "n": "New RadioButton",
        "i": "2131558532",
        "h": 96,
        "w": 421,
        "y": 1584,
        "x": 295,
        "e": 3
    }, {
        "actualClass": "RadioButton",
        "t": 1,
        "v": "on",
        "a": "toggled",
        "n": "New RadioButton",
        "i": "2131558532",
        "h": 96,
        "w": 421,
        "y": 1584,
        "x": 295,
        "e": 3
    }, {
        "actualClass": "RadioButton",
        "t": 1,
        "v": "on",
        "a": "toggled",
        "n": "New RadioButton",
        "i": "2131558532",
        "h": 96,
        "w": 421,
        "y": 1584,
        "x": 295,
        "e": 3
    }, {
        "actualClass": "RadioButton",
        "t": 1,
        "v": "on",
        "a": "toggled",
        "n": "New RadioButton",
```



```
        "i": "2131558532",
        "h": 96,
        "w": 421,
        "y": 1584,
        "x": 295,
        "e": 3
    }, {
        "actualClass": "RadioButton",
        "t": 2,
        "v": "on",
        "a": "toggled",
        "n": "New RadioButton",
        "i": "2131558532",
        "h": 96,
        "w": 421,
        "y": 1584,
        "x": 295,
        "e": 3
    }, {
        "t": 3,
        "v": "back",
        "a": "tapped",
        "n": "Navigated back",
        "i": "",
        "h": 0,
        "w": 0,
        "y": 0,
        "x": 0,
        "e": 7
    }
  ]
  },
  "__v": 0
},
"pagination": {
  "totalPages": 259
},
"error": {
  "hasError": false,
  "message": ""
}
}
```