MobileAge

D2.4: Behaviour Analytics & Workflow Software Components

Report

Project acronym:	Mobile-Age
Project full title:	Mobile Age
Grant agreement no.:	693319
Responsible:	ULANC
Contributors:	ULANC
List of Authors:	Christopher Bull, Nigel Davies (ULANC-SCC)
Document Reference:	D2.4
Dissemination Level:	PU
Version:	Final
Date:	09/10/17

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 693319

History

Version	*Date*	*Modification reason*	*Modified by*
0.1	14/12/16	Initial draft	Christopher Bull
0.2	20/02/17	Final draft for internal review	Christopher Bull, Nigel Davies
0.3	23/02/17	Minor revisions based on internal reviews from AUTH+UPM+TT	Christopher Bull
0.4	27/02/17	Quality check	Christopher Bull, Nigel Davies
1.0	28/02/17	Final reviewed deliverable	Christopher Bull
1.1	27/06/17	Revisions to meet reviewers' requests	Christopher Bull
1.2	09/10/17	Revisions to meet reviewers' requests	Christopher Bull, Nigel Davies

Table of contents

List of figures

List of tables

List of abbreviations

Abbreviation	Explanation
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CRM	Customer Relations Management
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
JS	JavaScript
JSON	JavaScript Object Notation
OSCPSEP	Open Senior Citizen Public Service Engagement Platform
PaaS	Platform as a Service
RCM	Project Partner: Region of Central Macedonia
REST	Representational State Transfer
SaaS	Software as a Service
SDK	Software Development Kit
UI	User Interface
UUID	Universal Unique Identifier
ZGZ	Project Partner: Zaragoza

Executive summary

This deliverable reports on work carried out in Work Package 2 (WP2), as part of Task 2.5, which aims to develop an innovative end-user analytics module to enable councils, application developers, and other organisations to improve the quality of services they offer to aging citizens. The Mobile Age analytics module is a software-as-a-service (SaaS) and associated mobile components capable of capturing interaction event data (e.g. clicks, views, dwell time), supplemented with context information exchanged during end-user sessions with public services and IoT data. This provides an opportunity to explore a novel approach to support new integrated workflows between disparate public services.

Current analytics systems focus on collecting data within single web domains or, at best across related domains. In this deliverable, we show how we can extend the idea of analytics to include a rich set of data relating to activities in both the digital and physical domain – enabling entirely new insights into how aging citizens interact with council services.

This deliverable describes a series of scenarios for potential analytics collection that include data sources (e.g. mobile, web, IoT) from a range stakeholders (e.g. citizens, various government services). These scenarios are followed by a discussion of design considerations for analytics synthesis through the novel analytics module. Technical implementation details of the Mobile Age analytics module, including a conceptual architecture describing how analytics events are processed in an end-user's domain (e.g. on a mobile device) are provided. The deliverable includes specific details of the technical innovation demonstrated.

1 Introduction

Understanding how aging citizens interact with services is crucial to developing solutions that improve the current provision. However, developing this understanding is far from trivial – end-users may interact with services in a wide range of ways (modalities) including via the web, dedicated applications, phone calls and informal conversations. Moreover, such interactions may cause observable events in the physical world such as hospital appointments, bin collections or home visits.

In the context of the web, analytics systems such as Google Analytics have been instrumental in transforming the way web site owners monitor visitor interactions. In particular, the detailed traces of individual end-user interactions – aggregated so as to preserve anonymity – provide detailed insights into the usability of a web site. However, in the context of Mobile Age we require substantially more sophisticated analytics that can capture interactions that occur spanning multiple modalities and that include events in the physical world.

Central to our work in Mobile Age is the concept of *analytics synthesis*, i.e. combining analytics data from multiple stakeholders to provide new insights into the way senior citizens engage with services.

In this deliverable, we describe in detail the Mobile Age analytics module.

1.1 Placement and Objectives

Deliverable 2.4 is part of Work Package 2, which is responsible for the design and implementation of the Open Senior Citizen Public Service Engagement Platform (OSCPSEP). More specifically, the objectives of Work Package 2 are as follows:

- Identify generic user-oriented and system-related, functional, as well as non-functional requirements for the implementation of the system modules.
- Provide technical specification and design the overall architecture of the OSCPSEP infrastructure.
- Define the guidelines for the integration of modules.
- Stress test and integrate all modules developed.
- Implement an early OSCPSEP platform release and develop the final OSCPSEP platform.
- Test and deploy the pilot case applications.

The main objectives of this deliverable are:

1 To present analytics scenarios, developed with public authorities, that outline the need for a new approach to analytics.
2 Derive design considerations.
3 Present the design of the analytics module used within the OSCPSEP.

1.2 Scope and Relationship with Other Deliverables

As illustrated in Figure 1.1, Work Package 2 consists of five tasks. This deliverable is a result of work performed within Task 2.5 – Behaviour Analytics & Service Workflow Engine Development, but also as part of Task 2.6 - Overall System Integration, as it provides endpoint definitions and concrete examples of information exchange with the analytics module.

This deliverable is primarily a counterpart to deliverable D2.3 'SaaS OSCPSEP Software Components', as the analytics module is another SaaS component that will integrate into OSCPSEP. This deliverable re-caps the OSCPSEP requirements that are relevant to the Mobile Age analytics module (section 4.3.3), initially introduced in D2.1 ('OSCPSEP requirements').

This deliverable also relates to D2.2 ('Interim OSCPSEP technical specification') due to its future integration with the platform detailed in that deliverable.

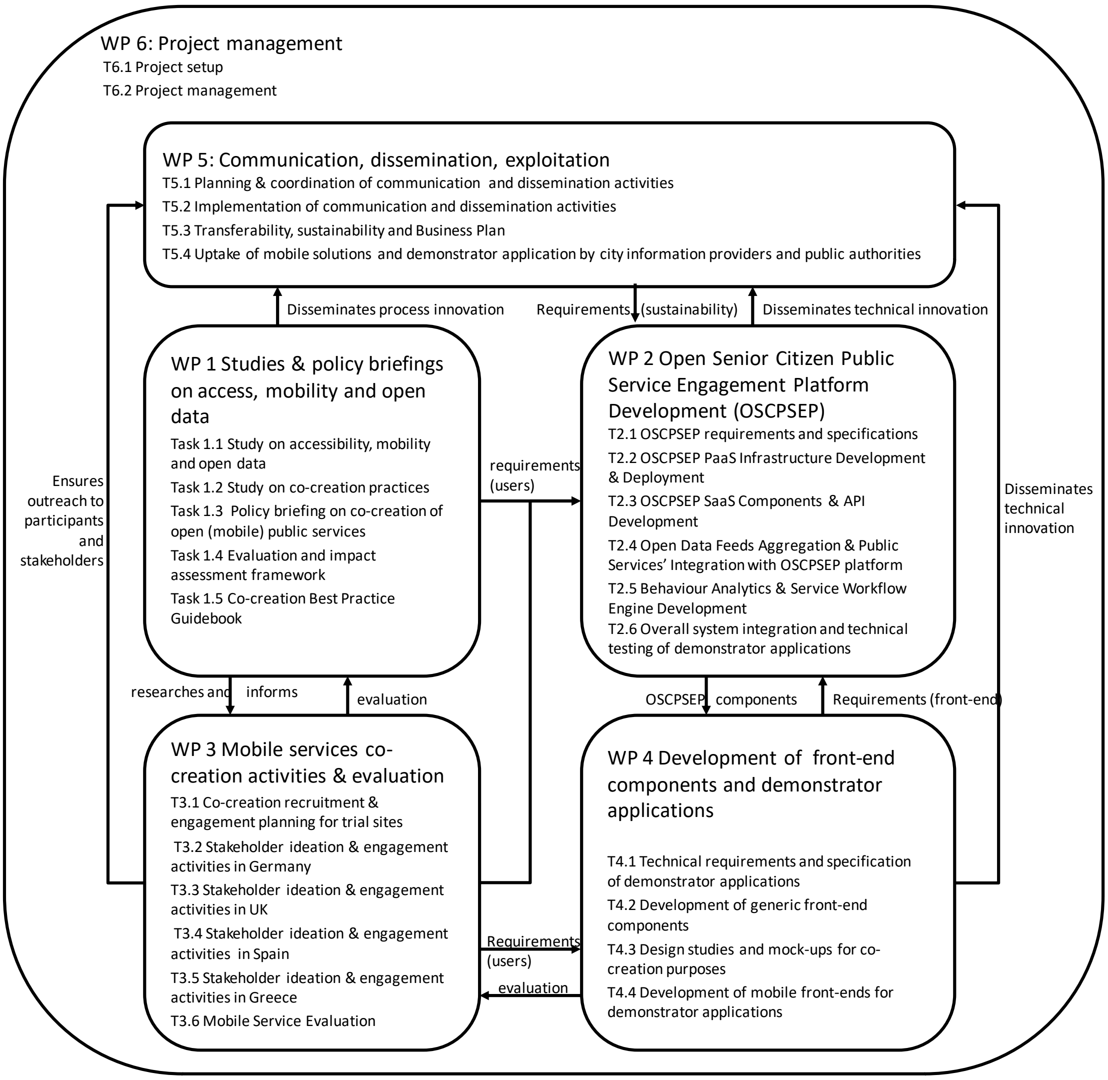

Figure 1.1: Mobile Age Work Package structure and interrelationships.

This deliverable (D2.4) is linked to other Work Package 2 deliverables, in particular D2.2; by summarising the architecture, functionality and interoperability of the Mobile-Age OSCPSEP infrastructure and modules, D2.2 constitutes a core document for WP2. It is a living document and reports on the first part of the second phase of the requirements engineering process of Mobile-Age, as depicted in Figure 1.2. The process is expected to be finalized in month 17 with the release of *D2.5 - Final OSCPSEP technical specifications.*

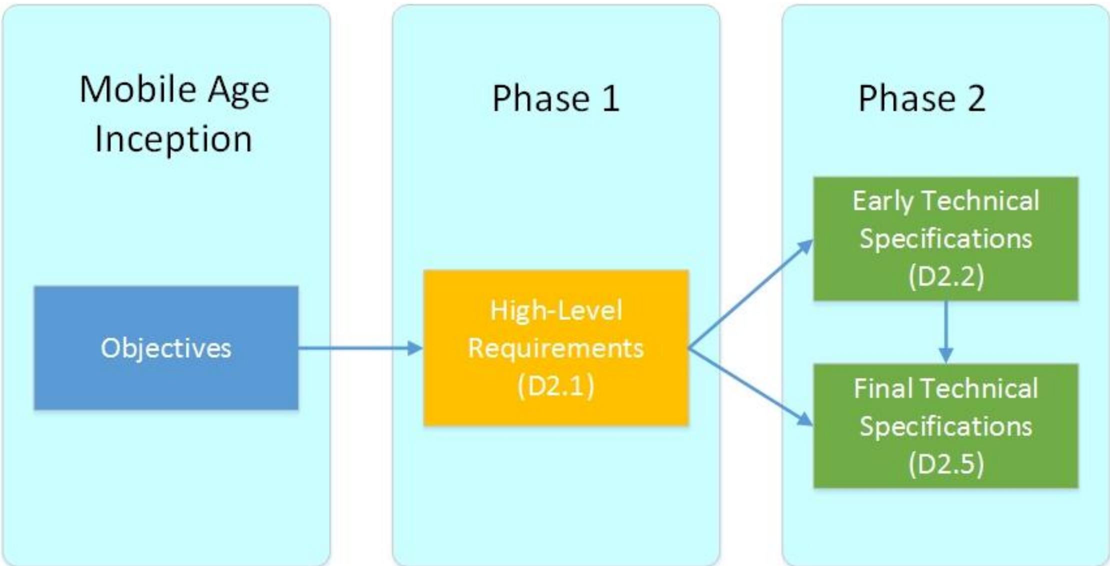

Figure 1.2: Phases of OSCPSEP requirements process, from project objectives to technical specifications

This document has close connections to other deliverables of the same Work Package:

- The module requirements included in this document originate in the corresponding set found in *D2.1 – OSCPSEP Requirements*.
- The work described here will be reflected on the development of the early version of the OSCPSEP, which will be reported in *D2.6 - Early OSCPSEP release* (Month 18).

Figure 1.3 provides a diagrammatic representation of the links of D2.2 to other deliverables in the project.

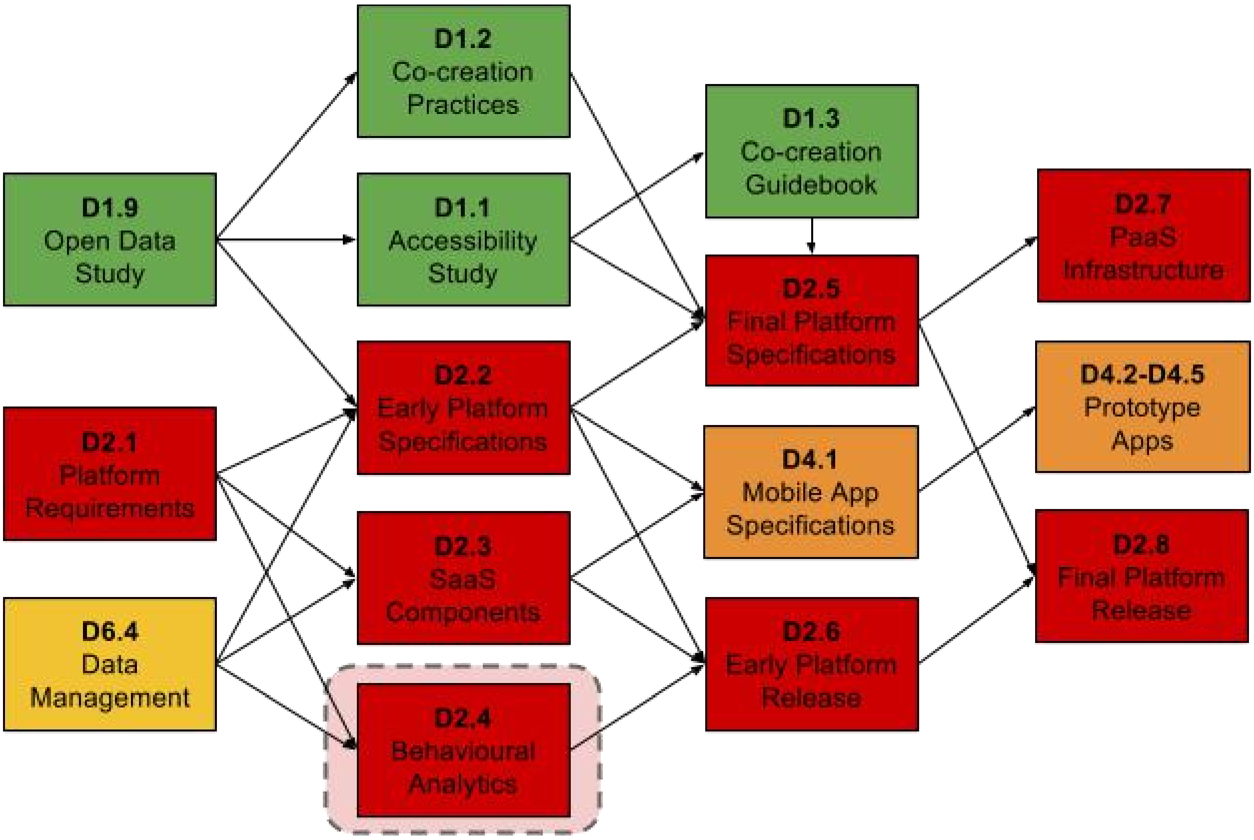

Figure 1.3: Relationship with other deliverables

1.3 Contributors

Table 1.1 lists the contributors for each chapter and section in this document, along with their affiliation.

Table 1.1: List of contributors

Document Chapter	*List of Sections*	*Contributors' Information*	
		Name	*Affiliation*
1. Introduction	All	Christopher Bull Manolis Falelakis	SCC AUTH
2. The Mobile Age Ecosystem: Stakeholders and Users	All	Manolis Falelakis Michail Papamichail	AUTH AUTH
3. Methodology	All	Christopher Bull Nigel Davies	SCC
4. Results	4.1 - 4.2	Christopher Bull Nigel Davies	SCC

		Mateusz Mikusz	
4. Results	4.3	Christopher Bull	SCC
5. Innovation Aspects	All	Christopher Bull Nigel Davies Mateusz Mikusz Mike Harding	SCC
6. Conclusions	All	Christopher Bull Nigel Davies	SCC
APPENDIX I – Installation: Full Code Samples	All	Christopher Bull	SCC
APPENDIX II - Installation and Usage	All	Christopher Bull	SCC
APPENDIX III - Public Authority Responses: ZGZ	All	Christopher Bull	SCC

1.4 Document Structure

The remainder of the document is structured as follows: we first describe the stakeholders and users of the Mobile Age ecosystem (section 2), clearly defining terms such as 'User'. We then set out the methodology used (section 3) in relation to this deliverable, including how Public Authorities are used in the co-creation of the analytics module.

Next, we introduce several illustrative scenarios (section 4.1) to frame the problem-space and motivate our new approach to analytics. This is followed up with a section on design considerations for this new approach to analytics (section 4.2).

Next, a technical description is given of the Mobile Age analytics module (section 4.3), including project requirements being satisfied, description of an existing analytics technology being extended (Pheme [1]), architecture, APIs, and how the module interacts within OSCPSEP. Installation and usage instructions are provided in the appendix, to further illustrate how analytics are reported to the analytics module. This includes registering for the service, code snippets, and examples. This is complemented by appendices, which contain some complete minimal working examples (MWE), showing code a developer would need to interact with the analytics module.

Finally, we describe the innovation aspects of this work (section 5), and conclude the deliverable with key findings (section 6).

2 The Mobile Age Ecosystem: Stakeholders and Users

Here we provide an overview of the roles of Mobile Age users. Please note that their roles may be overlapping.

Local/regional governments: These can be managing the co-creation activities, define features of the applications, and serve as experts for a specific service domain. In many cases, local governments are also the data owners.

Software developers: These can be independent developers or companies, or working for IT-departments in public authorities or civil society organisations such as the Open Knowledge Foundation. They develop the applications using the platform and they participate in the co-creation activities, adjusting the applications to accommodate for the participants' requests and demonstrating the results in an iterative process.

Older adults: They are a key stakeholder of Mobile Age and the main users of the mobile applications being developed. They may participate in the core project group or engage in the broader co-creation activities.

Service providers such as government, social welfare organisations, religious congregations or NGOs may be part of the core project group or engaged for specific input. Some of the service providers may also provide (open) data.

Intermediaries include professionals and non-professionals that may support the co-creation activities by providing input for specific tasks in the co-creation process. They may become users of the applications developed.

Facilitators are experienced individuals in the work with older adults and/or groups. They support the co-creation activities through e.g. running workshops, focus groups, interviews.

Other organisations & individuals comprise for example senior citizen organisations, senior citizens' clubs (e.g. computer clubs) but also media and journalists that may report about the co-creation activities, and thereby support engagement as well as dissemination.

Figure 2.1 provides an illustration of these roles.

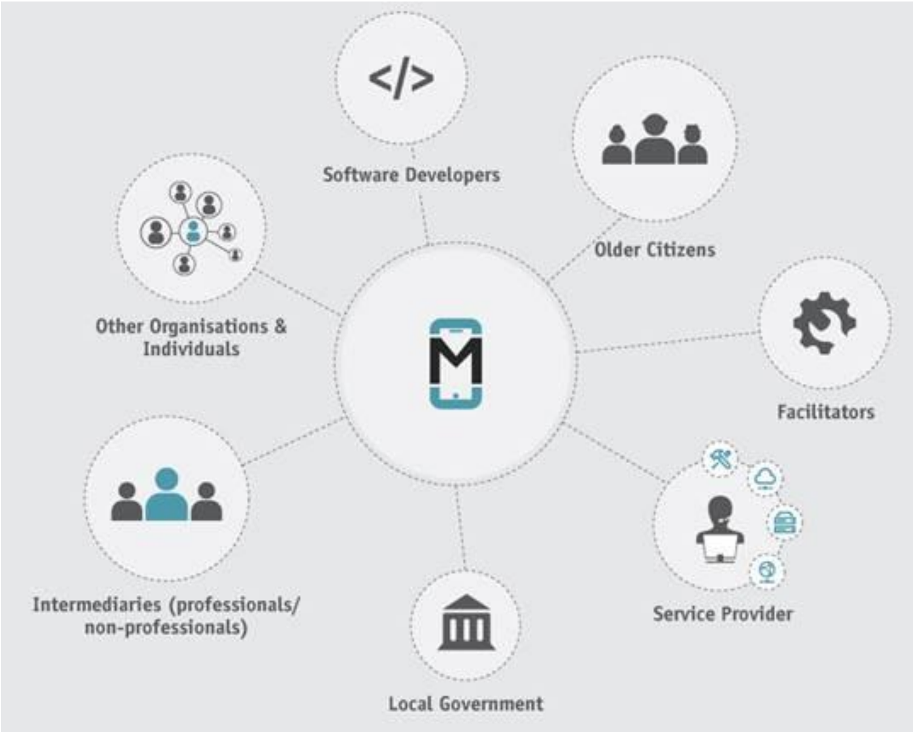

Figure 2.1: Overview of the Mobile Age users and stakeholders in co-creation of open data-based public services

Based on how these entities interact with the Mobile Age ecosystem, they can be categorized in two main groups:

- ***Platform Users***: These are users that make direct use of the platform and consist of the following:
    - **Software Developers**
    - **Service Providers**
- ***End Users***: These are users of the mobile applications developed using the Mobile Age platform.
    - **Older adults**
    - **Local /regional governments**
    - **Intermediaries (professional and non-professional)**
    - **Facilitators**
    - **Other Organizations and Individuals**

Figure 2.2 illustrates this categorization.

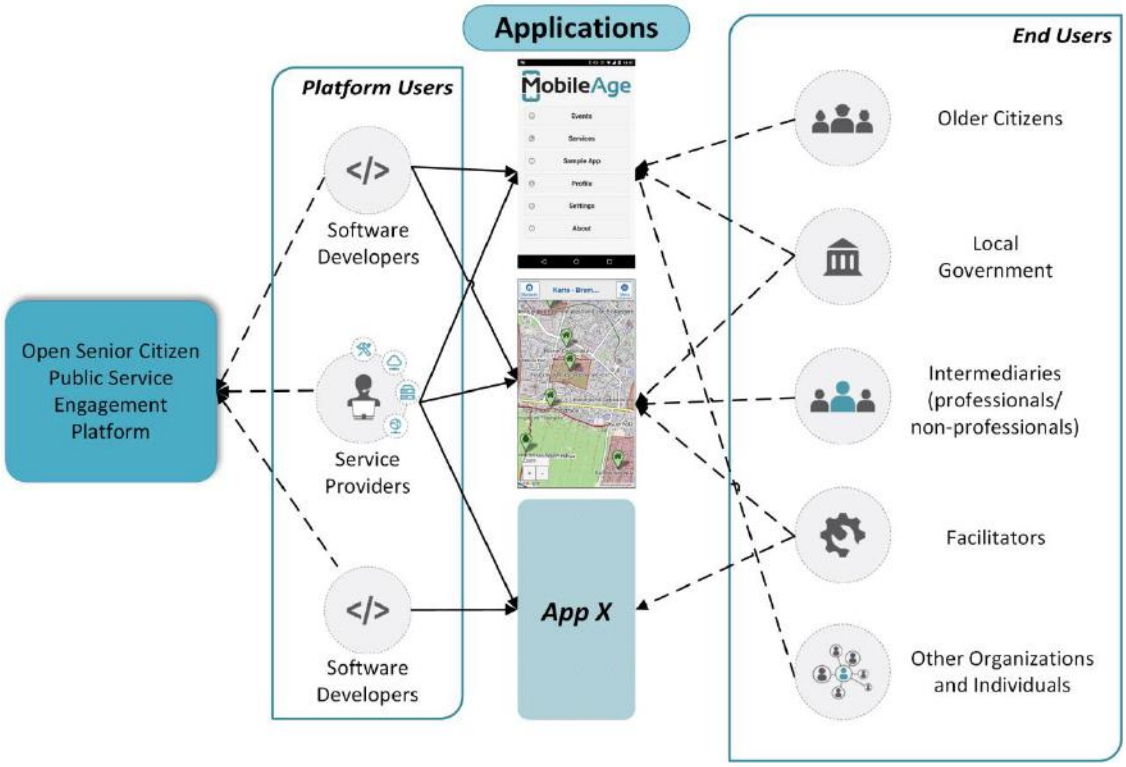

Figure 2.2: Overview of the Mobile Age users and stakeholders in co-creation of open data-based public services

3 Methodology

3.1 Approach

Illustrative stakeholder scenarios were created to frame the problem of multi-stakeholder analytics when synthesising analytics and user-relevant data from multiple data sources (see section 4.1).

We then identified several significant design considerations (section 4.2) that need to be addressed to satisfy requirements of all involved stakeholders. The idea of analytics synthesis, and the design considerations, have also been applied by the authors to a publication exploring novel analytics for pervasive displays [10]. These considerations are transferable due to the multi-stakeholder nature of the analytics.

The development of the analytics module extends an existing analytics technology: Pheme [1].

3.2 Co-creation Aspects

The stakeholder scenarios used for the design of the analytics module, presented in section 4.1, were derived through a mix of brainstorming internally amongst project members (listed in Table 1.1), meetings and interviews with local council employees (South Lakeland District Council representatives), and responses to questions from written communication with international local authorities (facilitated through project partners).

Written Communication

A set of questions were identified to build an understanding of analytics usage by public authorities, and, where possible, identify how a public authority might utilise synthesised multi-stakeholder analytics. The following questions were sent to representatives of public authorities, and expanded upon in the following subsections:

1. What are the various ways you capture how people interact with the council?
2. What kind of analytics and reports do you currently collect regarding an individual's (anonymous) user experiences?
    1. …do you collect these just within a single website?
    2. …do you collect these across websites within your domain (e.g. tracing a single user's experience from one of your websites to another)?
    3. …do you collect analytics for individual user experiences across partner websites (e.g. a user starting from SLDC website and moving to the AgeUK website)?
    4. …thinking beyond the websites, …….
        1. do you correlate phone calls or physical visits with web activity?
        2. do you collect any cyber-physical analytics (e.g. Internet of Things, sensors, RFID)?
3. Of the above (2.1-2.4), would you like to be able to collect these types of analytics?
4. Why would you like to collect these types of analytics?
5. Can you think of any concrete scenarios that could benefit from access to these types of analytics?

3.2.1 Co-creating with a Local Public Authority

The primary public authority (PA) was the South Lakeland District Council (SLDC). The representatives from SLDC were Paul Mountford (Principal Performance & Intelligence

Officer) and Lola Denwood (Web Manager). The above questions were sent to them in advance, and their responses were given and discussed face-to-face at South Lakeland House. They discussed how they use analytics to optimise the provision of their services. Various sources of analytics data were described, which included web analytics (primarily for optimising service delivery), and social media engagements.

When we discussed the potential for including analytics from multiple stakeholders, one of the conversations was about how this could inform the council whether their advice to go the council's website was actually followed - which relates to scenario 4.1.5 ("Scenario: Council Telephone Query").

One scenario explicitly co-created with this local PA was collecting physical analytics related to refuse collection – scenario 4.1.3 ("Scenario: Refuse Collection"). Of particular interest to this PA was optimising telephone queries from residents, regarding refuse collection and assisted collection (for residents with mobility restrictions). Analytics from multiple stakeholders could be synthesised to provide better service mappings – identifying people who could benefit from available services, especially for those that do not know the service exists.

3.2.2 Co-creating with International Public Authorities

International public authorities were also contacted with the above list of questions, facilitated by our international project partners. Representatives from Zaragoza (ZGZ) and Region of Central Macedonia (RCM) responded through partners of Mobile Age (Víctor Morlán and Manolis Falelakis, respectively). It was clear that the utility of multi-stakeholder analytics was not obvious to the PAs, given that the responses to the final question, for example, focused on using the analytics to prioritise a mobile-first design of their website. However, they understood the potential benefits, and some insights were drawn from their responses:

Zaragoza:

The email response from ZGZ was translated by Víctor; the verbatim textual responses are attached in *"APPENDIX III"*.

This PA reported capturing a variety of traditional and impersonal sensor-based analytics (such as air quality sensors), but does not currently synthesise data from multiple sources and stakeholders. However, they did state that they would like to do so to:

1. Give a personalized answer to their needs
2. Detect their actual and future needs.

The responses from this PA did not inform any scenarios, but did highlight a desire to use multi-stakeholder analytics with heterogeneous data sources.

Region of Central Macedonia:

The responses from RCM were relayed to us in discussion with Manolis Falelakis after he met with RCM to discuss their use of analytics.

This PA collects a variety of traditional analytics data, including crowd-sourced data about local issues. As part of Thessaloniki's focus on open government, they have a service[1] (and associated app[2]) for visualising and reporting issues in the local city called Improve My City

[1] http://opengov.thessaloniki.gr/imc

[2] https://itunes.apple.com/us/app/improve-my-city-thessaloniki/id1067808924

(shown in Figure 3.1). This service “*empowers users to improve their city in collaboration with local administration*”[2], and can be used to report a variety of issues. This service was suggested to be considered as part of cross-service or cross-stakeholder analytics, but they did not have any concrete suggestions.

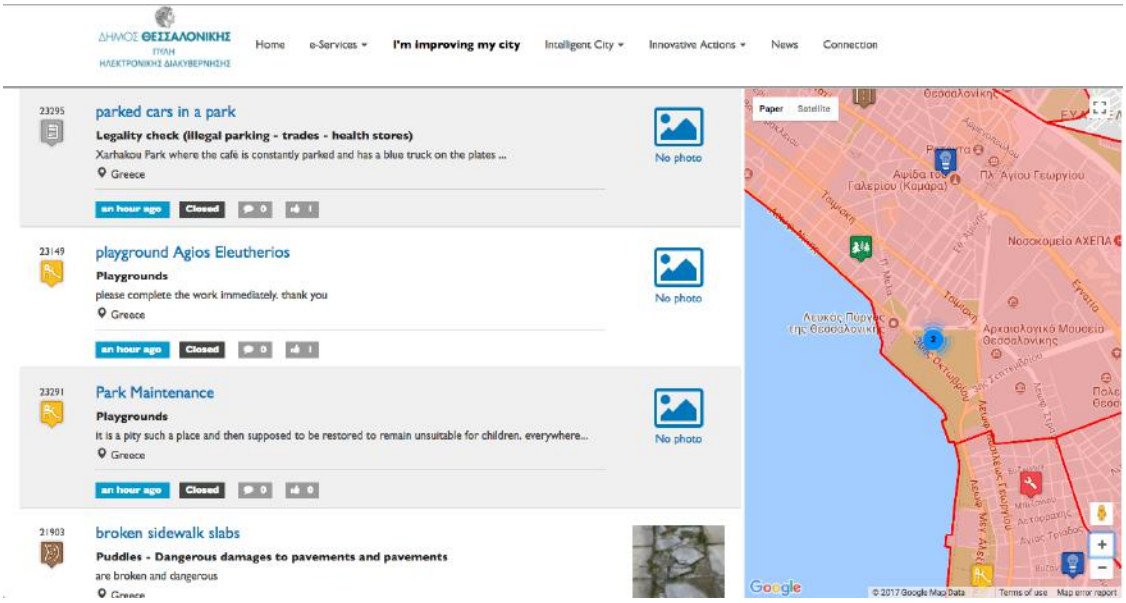

Figure 3.1: Improve My City website (Thessaloniki)

RCM highlighted their access to various data sources (e.g. web analytics, IoT sensors, and the crowd-sourced data). However, despite no new scenarios being developed with this PA directly, their input did reinforce scenarios that mixed sensor data across various stakeholders, for example ‘Scenario: Refuse Collection’ (section 4.1.3).

3.2.3 Co-creating with Older Adult Participants

Co-creating the analytics module with older adults was not part of the original plan. However, the analytics module was discussed with the participants during the fourth co-creation workshop – in part because we endeavour to make the entire project and co-creation process transparent with our co-creators.

The feature that was co-created (further described later in section ‘4.3.6 Privacy’) was the ability to opt-out of any or all analytics collection. Originally the participants expressed vehement dislike for tracking and being tracked, which was based in privacy concerns, and often related to the advertising industry. The researchers followed this up with a conversation of where analytics and data collection and sharing is not intended to make money, but instead aims to benefit users. The co-creators still generally disliked the idea of tracking, but acknowledged that, in context, the potential for direct benefits to them. As such they suggested an ability to opt-out of analytics on the mobile device.

4 Results

4.1 Stakeholder Scenarios

To understand the requirements for behavioural analytics we developed a series of scenarios to help frame the problem-space. These scenarios were derived through brainstorming internally amongst project members (listed in Table 1.1 for this section), meetings and interviews with local council employees, and written communication with international local authorities through project partners. The communication with the pubic authorities is summarised in Appendix II.

Each of the scenarios is presented in the following format:

- A description of a scenario from the perspective of a citizen, to set the scene.
- A description of the data that is, or could be made available, using the analytics module.
- A description from the perspective of someone wishing to use the analytics data (e.g. a council employee or app developer)
- Examples of the opportunities that are afforded by using this new analytics module.

Each scenario is further tagged with the types of analytics that the scenario uses:

Digital Analytics is a foundational aspect of the analytics module, with its ability to collect usage statistics for applications and websites. These can be collected within a Mobile Age app, and reported to the analytics module. In essence, digital analytics represents a combination of both web and mobile analytics.

Examples of logged data, in its most basic form, include app open times, app open duration, activities/views initiated, search queries/filters input, buttons pressed etc. End-user location could also be periodically reported (with permission, likely obfuscated), which could give a general insight into frequently visited locations or rare/one-off locations. There is also potential for later extensions gathering mobile specific data from (e.g. phone sensors, which would distinguish it from more traditional analytics).

Digital analytics data can be used to determine an end-user's pathways through an app, their exit points, and provide indications of end-user behaviour – helping developers refine their apps.

Inter-App Digital Analytics are another core aspect of the analytics module. The inclusion of analytics across apps utilises common end-user identities and profiles. Analytics across apps, rather than just per-app analytics, is typically not available to app developers in traditional systems. Their addition can be used to identify common behaviour across apps.

Cyber-Physical Analytics are interesting for combining data captured through the analytics module with data reported for service use, both digitally and physically. For example, we might combine digital analytics data with usage logs from council service providers to understand interactions with council services and, furthermore, track physical events to understand actual use of services in the system. This is could also enable non-analytics data (for example, open government data) to be pulled in to the analytics module, and used for additional processing of analytics events.

Digital Analytics	*Inter-App Digital Analytics*	*Cyber-Physical Analytics*
✓	✓	

Table 4.1: Analytics types in scenario - Example scenario

4.1.1 Scenario: Public Transport

Transport is a key service for many elderly citizens and through behavioural analytics we hope to provide new tools to enable councils to improve their services.

> *Barbara is using a Mobile Age app to find social events in her local area. She discovers an afternoon tea session, held in the town hall, that she is interested in attending. Whilst viewing the details of the event in the app, she selects an option to pre-book a bus ticket – this opens a separate Mobile Age app for booking public transport. Barbara searches for buses to and from the event's location; she sees that there is a direct bus during the day, but for the journey back at night there is only one route available that takes longer and requires changing buses in an adjacent town. Barbara decides not to attend the event and finds a different one she is less interested in. She purchases a bus ticket, closes the app, then puts her tablet away. She shortly receives an email confirming the purchase.*

Available data (): In this scenario, there is potential to collect data about local social events that were browsed in an app. Within the same app there is data on intent to purchase transport to the event (selecting the option to pre-book a bus ticket). In the separate public transport app, there is data about journeys (pickup location, destination, and times), whether a bus route is available, and whether a purchase was made. There is also external data available, provided by the local bus company, on ticket sale stats, and bus route delays.

> *Council person Joe is interested why certain council run events have better participation than others. Some of the organisers know anecdotally that some dates, times, and places are better for attendance than others – Joe would like to explore and better understand why this may be. Joe looks at analytics reports and sees patterns of people using the social events app, navigating to the separate transportation app, but then not following through with a ticket purchase.*

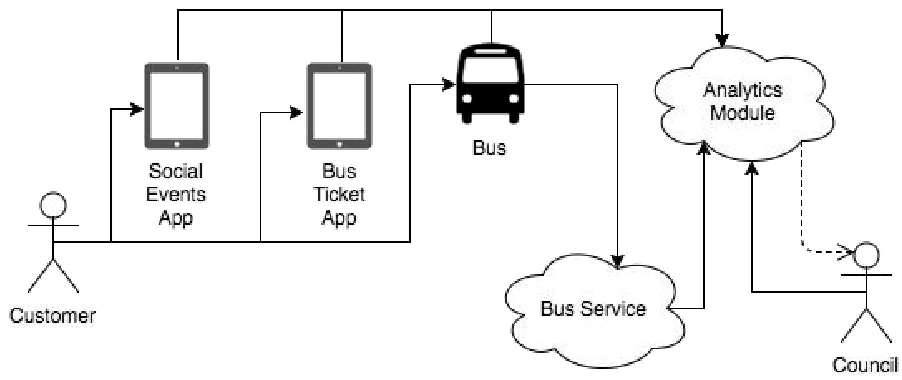

Figure 4.1: Scenario Diagram – Public Transport

Opportunities for analytics: The council person could request data on whether end-users using a 'local events' app also used a 'travel' app for organising transportation to events they viewed in the former app. Having end-user profiles/identities that span across the different apps will enable analytics of the wider ecosystem. There is also a potential to use this to verify data provided by private transport companies (e.g. ticket purchase numbers, delays/timeliness, etc.).

Digital Analytics	Inter-App Digital Analytics	Cyber-Physical Analytics
✓	✓	

Table 4.2: Analytics types in scenario – Public Transport

4.1.2 Scenario: Pharmacy Prescription Collection

This scenario describes a citizen purchasing a prescription online from a local pharmacy.

> *George had a GP's appointment this morning, where he was given a prescription. He is moderately mobile, but must travel to this pharmacy on the bus as it is too far otherwise. He opens a Mobile Age pharmacy app and he pays for his prescription in advance, ensuring it will be available when he goes to collect it. He then opens the local bus app on his phone and purchases a bus ticket for tomorrow morning (a Friday). He wants to ensure he gets to the pharmacy early as it is closed on weekends, and does not want to have to wait until Monday to collect it.*

Available data (): George has data available in the pharmacy app (when the prescription was made, when the order was placed, and which pharmacy he will be collecting it from). The pharmacy can record time and place of pickup; this includes George's app knowing when he is in the pharmacy. George has public transport usage that can be logged (time of purchase, journey information). Additionally, external data can be pulled in, such as data volunteered by bus companies (about bus routes, frequencies, and delays).

> *Council worker Tess wishes to better understand delays in picking up prescriptions from pharmacies, and ultimately explore options to improve the well-being of citizens in her area. She looks at analytics reports to attempt to identify pharmacies and demographics that have a higher frequency of delayed prescription collections than others, and what the contributing factors are. With the provided information, she then looks at other external data sources (such as bus routes and availability) to see if there is a relationship between prescription collections in an area and availability of public transport links. It turns out that delayed pickups are clustered around end-users who have limited transport options.*

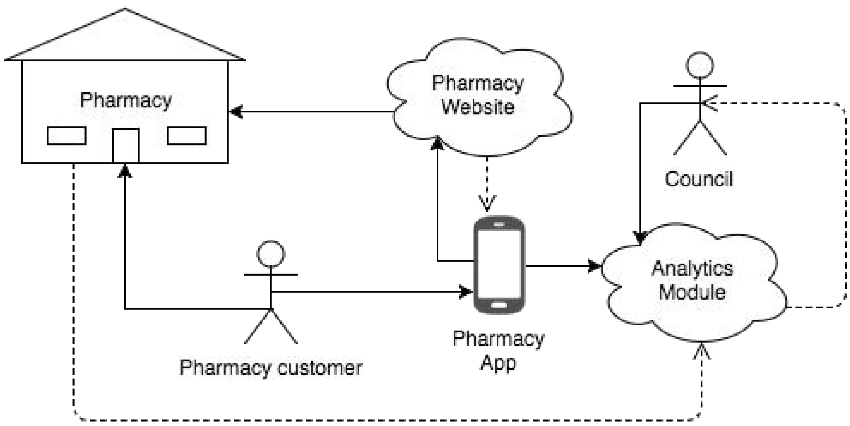

Figure 4.2: Scenario Diagram – Pharmacy Prescription Collection

Opportunities for analytics: Using digital analytics we can track end-users through the pharmacy app and know if they place an order at a pharmacy. Using cyber-physical analytics, we can see when (anonymised) prescriptions were physically collected from a pharmacy, and the time difference between prescriptions being written, ordered, and collected. Thus, we can track aggregated behaviour in both the digital and physical domains.

Digital Analytics	*Inter-App Digital Analytics*	*Cyber-Physical Analytics*
✓	✓	✓

Table 4.3: Analytics types in scenario – Pharmacy Prescription Collection

4.1.3 Scenario: Refuse Collection

This scenario is about a citizen that regularly struggles with emptying his household bin.

> *Phil is an elderly person with recent mobility issues, and lives in a terraced house. Each week the bins of all the houses on his street are collected and emptied from the alley at the rear of their respective houses. The refuse collectors take each bin to the end of the street, empty it, and then they leave all the streets' bins at the end of the alley. Phil finds it difficult to walk down the cobbled street at the rear of his house, and relies on a neighbour, relatives, or visitors to bring his bin back for him. Some days Phil even struggles to move the bins from his garden into the alley. One week Phil's daughter finally encourages Phil to call the council to talk about this situation; as a result, Phil is now signed up to the 'assisted collection' service for his bins.*

Available data (): In this situation, the refuse collection lorry has sensors that automatically scan each bin as it is emptied. There is also external data from the council about refuse collection queries and complaints, for example, from telephone calls to the council.

> *Abigail, who works at the district council, is looking to identify early potential recipients of the 'assisted collection' service. She looks for frequent but irregular missed refuse collection – this could indicate that a household may require assistance. She also looks to correlate queries and complaints about refuse collection, as this may show some interesting patterns. By combining refuse collection analytics and contact with the council (which initiated enrolling someone onto the 'assisted collection' service), Abigail hopes she may be able to find predictive indicators for when the assisted bin collection service may potentially be beneficial to someone.*

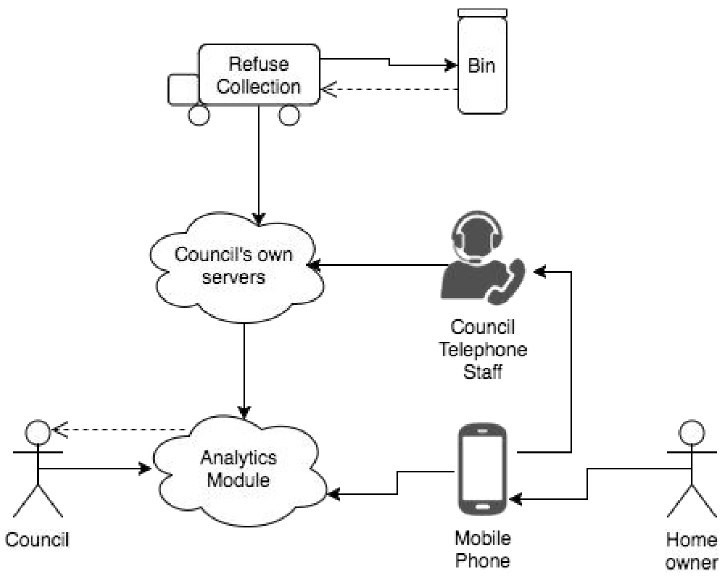

Figure 4.3: Scenario Diagram – Refuse Collection

Opportunities for analytics: By collecting data on typical bin weights of households, analytics could be used to find derivations from existing weight patterns. For example, if bins of a specific household were filled each week but suddenly had not been filled one week, this derivation from the pattern could potentially be useful to flag a house as this could indicate that someone requires assistance that they are not yet receiving (e.g. assisted bin collection service) – though it should be noted that this is a potential privacy concern, so the analytics should either go directly to the citizen, or as aggregate data to the council. Another opportunity would be to determine at what point citizens complain to the council, and try and pre-empt those circumstances. Further, if a household is known to have at-risk individuals (e.g. elderly, disabled, but mobile), then a single missed bin collection could indicate more significant possibilities (e.g. a fall in the elderly); being able detect this could flag this with the necessary people, for example a pre-registered trusted intermediary (e.g. family member). A separate opportunity for the council could be that they use the sensors in the bins to determine how full the bins are as they are emptied. Analytics could then be used to optimise refuse collection routes and intervals.

Digital Analytics	*Inter-App Digital Analytics*	*Cyber-Physical Analytics*
	✓	✓

Table 4.4: Analytics types in scenario – Refuse Collection

4.1.4 Scenario: Demographic Impact of Hospital Appointments – Beyond Wait Times

This scenario is about a person's journey to a hospital appointment – how much time it takes to attend an appointment, not just the time spent in the waiting room.

> *Edith has an appointment at the hospital. Her hospital is known for its wait times, which makes the long journey to the hospital less appetising. She sets off early to ensure she is not holding anyone up, but due to problems beyond her control (traffic, or late buses), she still misses her appointment.*

Available data (): Public transport data captured through transport apps. Patient waiting times in hospitals, either through sensors or pushing analytics events from the hospital when patients arrive and sign-in.

> *Trevor is exploring process improvements for hospital waiting times. By looking at analytics reports he can identify demographics that trend towards higher wait times, or even lengthy journey times. Trevor sees that certain people with longer travel times also have higher wait times in this hospital (due to arriving earlier than necessary). Trevor now uses the available data to set appointment times that reflect patients transport options, minimising the overall time that a patient needs to set aside to attend a hospital appointment.*

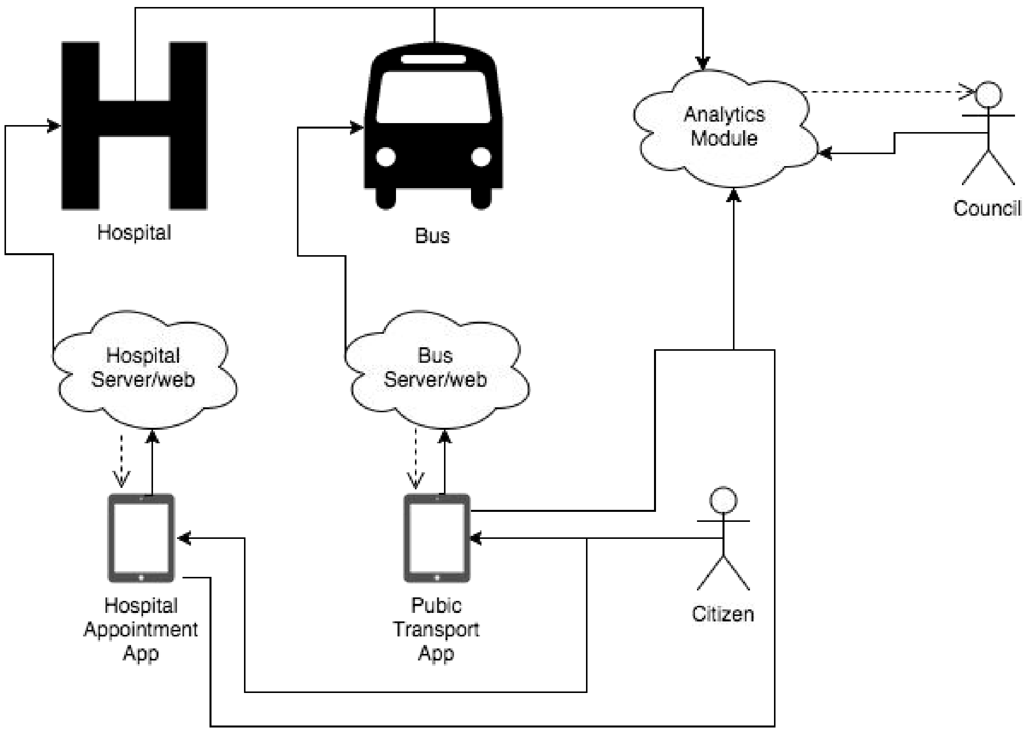

Figure 4.4: Scenario Diagram – Hospital Appointments

Opportunities for analytics: This scenario explores whether some demographics are more likely impacted by longer overall journey and wait times. This data can be used to more intelligently set hospital appointment times that reflect journeys: broadly where someone comes from, and probability of transport patterns and delays. A future possibility would be to use these analytics in the creation of a dynamic scheduling system, whereby appointments could be automatically adjusted or rescheduled – if someone misses their appointment, other appointments could get shuffled to optimise journey and wait times. This type of prescriptive analytics could provide a more efficient scheduling experience for hospitals. This scenario could also identify problems with bus routes for certain hospitals.

Digital Analytics	*Inter-App Digital Analytics*	*Cyber-Physical Analytics*
✓	✓	✓

Table 4.5: Analytics types in scenario – Hospital Appointment Wait Times

4.1.5 Scenario: Council Telephone Query

Citizens have a wide range of possible ways to contact the local council and it can be interesting to understand how these can be used in combination.

> *Margaret is moving to a new house and wishes to know when her bins will be collected at her new property. Without hesitation, she picks up her phone, searches for the local council's telephone number and makes the call. She is met with courteous service, and at the end of the call she is informed that the information she required could be found on the council's website – she shrugs this off with "but I don't know where to look for it".*

Available data (): Margaret's phone has the potential to log analytics of phone calls made, and activities that led up to that call. Analytics can also be collected from the council's website.

> *Jane works at the council, and is considering customer experiences of people who call the council. She is particularly interested in exploring if information for common queries can be easily found on the council's website, or why people fail to find these answers before making a call. She can view aggregated reports that show that many people act as Margaret did – calling the council first without ever exploring the web site.*

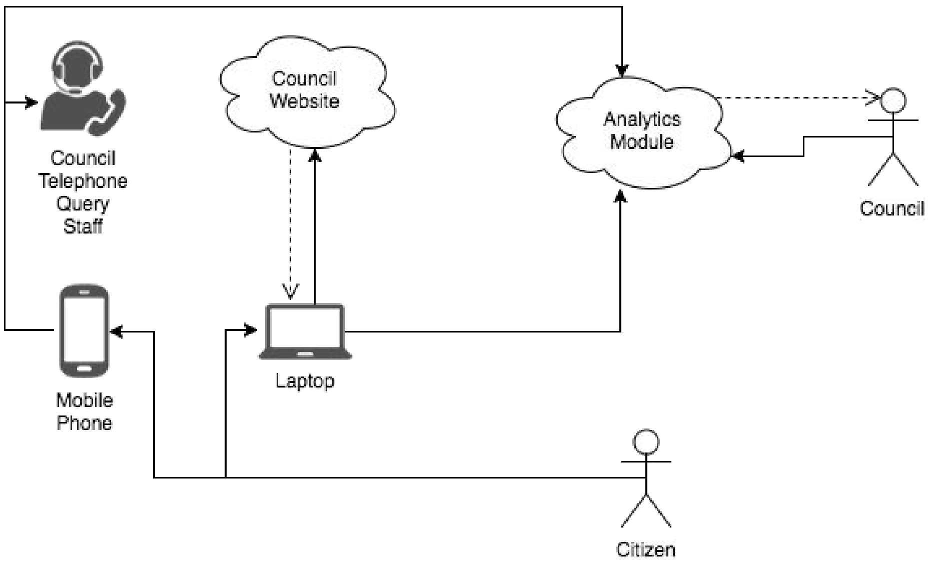

Figure 4.5: Scenario Diagram – Council Telephone Query

Opportunities for analytics: Determine how many people call the council after first looking for an answer to their query on the council's website. This could be achieved by correlating analytics captured between apps and services (phone analytics, website analytics). This could show if people use the council's website prior to making a call, and what pages they were on, and what they may have searched for. This could lead to improved website design and navigation, or frequently asked questions covered whilst people are on hold with the council. Furthermore, you could determine if someone visited the council's website after a phone call took place, which pages they went to, and if this led to a reduction in calls to the council.

Digital Analytics	*Inter-App Digital Analytics*	*Cyber-Physical Analytics*
✓	✓	

Table 4.6: Analytics types in scenario – Council Telephone Query

4.2 Design Considerations for Analytics Synthesis

To synthesise analytics and user-relevant data from multiple data sources including council websites, we need to address several significant design considerations to satisfy the requirements of all involved stakeholders.

The idea of analytics synthesis, and the design considerations described in this section, have also been applied by the authors to a publication exploring novel analytics for pervasive displays [10]. These considerations are transferable due to the multi-stakeholder nature of the analytics.

4.2.1 Protecting End-User Privacy

In traditional analytics, no personalised information of (Web) end-users is captured [3] as request logs are anonymised (e.g. hashed or shortened IP addresses or randomised end-user identifiers are used to identify the source of requests). Hence, such systems are typically compliant with legislation governing personal data collection (e.g. EU data protection rules) and raise few concerns amongst visitors. However, with the increased focus on analytics relating to individuals contacting council services through different means regarding specific requests, new privacy related concerns emerge that are compounded when analytics data from multiple stakeholders is aggregated. As the council could potentially track very detailed behaviours of their citizens, such data needs to be additionally protected by design—also to increase the trust of end-users in such systems and their willingness for data sharing.

For example:

- Data synthesis beyond consent. By providing user identifiable data to councils, insights gained could be combined with additional services without the consent of the end-user, potentially revealing information about individuals that was supposed to remain private—for example about their medical history.
- Tracking of complaints to the individual. Certain complaints to the council, e.g. about not emptied bins, should be handled in care. For example, by passing detailed information on to collection crews, individuals could be exposed to accusations from on-site crews.
- Building up end-user profiles. Generally, synthesising information from various domains of the council with additional end-user navigation traces and behavioural patterns can expose individuals and enable the collection of very detailed insights into individuals.

As a result, we believe that a substantial challenge for future analytics platforms will be the need to address issues of end-user privacy, and that fundamental work is required to understand the level of guarantees that will be possible in future environments that support synthesis of analytics data. Concretely, we believe that the need of privacy mediators [4] will be an essential component of such systems.

4.2.2 Protecting Commercially Sensitive Data

While there are clear benefits to be derived from analytics synthesis, certain stakeholder data sets in the previously described scenarios are likely to contain commercially sensitive information. For example, tracking customers in pharmacies to understand if certain prescriptions were picked up could also reveal further insights into the number of sales and profits of businesses. Similarly, by providing information on bus ticket sales, bus providers would, for example, reveal profitable bus routes to other stakeholders and potentially also to competitors. Indeed, it is likely that all analytics data is commercially sensitive to at least one of the stakeholders because it provides insights into either a physical space, behaviour or navigation patterns of end-users across devices or locations, or details on profits or sales statistics in retail.

The presence of commercially sensitive data might suggest insurmountable obstacles to sharing. However, precedents for sharing analytics data exist in many other areas of business. For example, in business-to-business relations (e.g. along a supply chain) it is common to share information with business partners – providing insights and benefits to all those involved [8, 9]. Inventory levels and point-of-sales statistics can be automatically transmitted to suppliers with the goal of better demand prediction for certain products and performance improvements [7]. Such information sharing typically requires written agreements and contracts between all involved parties [6] and as a most basic requirement, benefits from data sharing must be clear for all partners [7]. Finally, we observe that while it might be expected that less commercially sensitive data would be more likely to be shared—for example, visitor statistics for shopping malls – we found that, even for this type of data, stakeholders typically only publish aggregated figures (e.g. [5]) suggesting that the need for contracts and agreements is inevitable.

In summary, we believe that there are clear benefits to be obtained by combining analytics data from multiple stakeholders and clear precedents exist for data sharing in commercial environments but providing appropriate controls will be critical to success.

4.2.3 Aggregation Rules and Policies

Future analytics aggregation platforms will need to provide functions that allow end-users to control which data is aggregated, how the data is aggregated and with whom the results are shared. The first of these functions, i.e. controlling which data is aggregated, is likely to be amenable to standard access control and filtering approaches in which stakeholders can specify how data should be shared and any redaction (e.g. reduction in the sampling rate) that is needed prior to release. Indeed, it is crucial that "users should be able to control the release of their own data" [4] – this resonates with frequent comments from our co-creation participants in South Lakeland. Providing the capability for data aggregation itself is more complex. Our initial work suggests that aggregation is likely to be extremely specific to domain and data-type, and that support is likely to be needed for custom code fragments that perform the aggregation (rather than, for example, through generic queries).

Control of the aggregation performed will also be necessary and may cover a range of options such as specifying the analytics data feeds involved, the level of aggregation, combinations of policies across datatypes and stakeholders, additional sharing limitations and permissions. For example, stakeholders could specify that statistics on picked up prescriptions could be combined with other datasets but no personally identifiable data should be released. Finally, stakeholders will wish to control the sharing and distribution of processed datasets, e.g. by specifying rules that describe the future use of their data and combined datasets, and stating access and permission rights.

4.2.4 Architectural Models

Initially it may appear that a data sharing architecture in which each stakeholder can control access to their own analytics data would be appropriate to provide support for analytics synthesis. A simple example with our stakeholders illustrates the drawbacks in this approach. Consider the case in which the pharmacy holds statistics on all prescriptions that have been picked up, and hospitals or medical practises track which prescriptions have been issued and are due to be picked up. The aim is to produce a report that shows which proportion of patients have failed to pick up their prescription, and what the average delay between the expected and actual time was in which patients picked up their prescription—without risking the disclosure of full datasets to either of the stakeholders given the sensitive nature of the data. It is not possible for either of the stakeholders to independently decide which data they need to release. Only by merging both data sets can such a report be produced. As a result, any future analytics platform will most likely need to include a logically common aggregation component across multiple stakeholders.

We note that a logically common component does not necessarily imply a centralised implementation—a distributed implementation could be developed if required. In addition, we note that it is not necessary for there to be one single analytics service—multiple such services could exist but a single service would need to be employed by any group of stakeholders that wished to collaborate. Deploying an analytics system as a logically centralised service with common components potentially introduces new trust concerns, particularly as analytics data may be commercially sensitive. However, there is increasing confidence in the use of cloud services backed by appropriate service level agreements and we do not consider that this will be a significant issue. For example, companies already trust providers such as Google to maintain clear boundaries between data from different customers.

4.2.5 Data Integration

An analytics system must make the ingestion of raw data as easy as possible as each stakeholder will typically maintain many data sources. To simplify processing and storage of such information, and to allow easy integration of future data sources, a standard reporting format such as the Universal Measurement Protocol (UMP) [2] could be used. Such formats typically provide a wealth of appropriate data models (e.g. page impression, location information), and can be extended to support domain-specific data types such as retail statistics and end-user demographics. However, no existing analytics reporting protocols can cope with the diversity of data types and suitable extensions will be required.

4.3 Behavioural Analytics Report Module

To address the scenarios depicted in section 4.1, keeping in mind the design consideration in section 4.2, we have formulated a novel architecture for collecting and reporting analytics events. We describe the analytics module's technical details throughout this section, including how it will integrate into OSCPSEP (section 0).

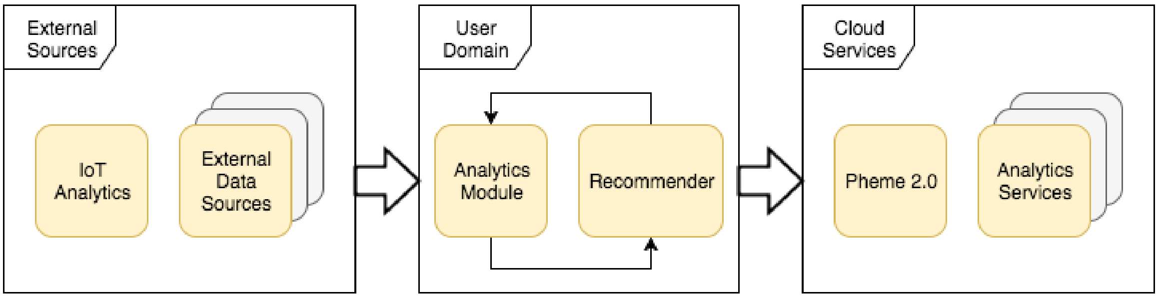

Figure 4.6: Mobile Age Analytics Module – Conceptual architecture

The Mobile Age analytics module is conceptually spread across three domains (Figure 6.1): External sources, end-user domain, and cloud services. The end-user domain is the primary location where analytics are collected and processed. This provides end-users the potential to moderate their analytics collection and sharing. The end-user domain can accept data from external sources, such as IoT and open data, enabling cyber-physical analytics. The cloud services domain can receive analytics events from an end-user domain for further processing. Pheme 2.0 in the below diagram is an existing technology called 'Pheme' that is being extended for this analytics module (detailed in section 4.3.1).

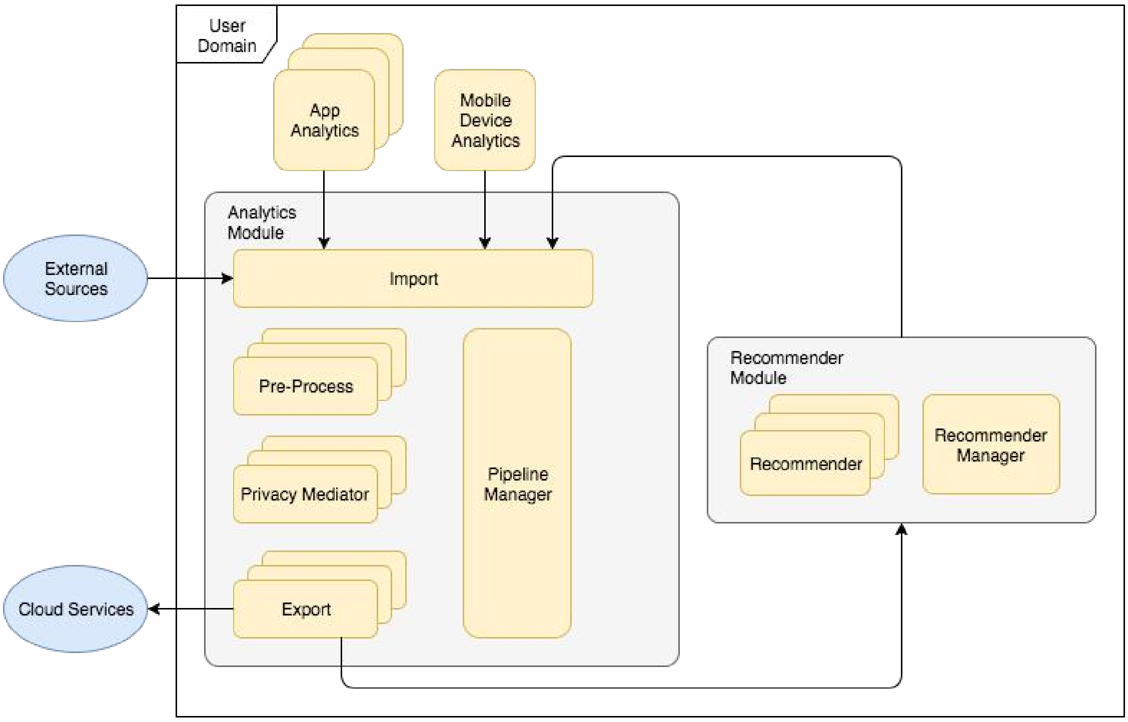

Figure 4.7: Mobile Age Analytics Module – Conceptual end-user domain

The end-user domain conceptually runs on an end-user's Mobile Age device, but in practice not everything may run on a mobile device – there can be server-side processing, whilst the data remains in the end-user's domain. A more in-depth look at this user domain is given in .

This conceptual architecture has several key properties, which include:

1. Capturing analytics and pre-processing in a mobile context.
2. Capturing device and cross-app analytics
3. Ability to receive IoT analytics to the end-user's domain (as an external source).
4. Using analytics directly for recommender components (without leaving the end-user's domain)

Processing in a mobile context: Mobile Age apps will be installed on end-user's mobile devices through a trusted launcher (the "Mobile Age Launcher") – this is an app to download, manage, and launch Mobile Age apps. This launcher will contain a built-in analytics manager which will enable offline pre-processing of analytics events, on-device privacy mediation of events, and message queuing to the analytics server. This allows collected analytics to conceptually remain within the end-user's domain, which is reflective of privacy and trust design considerations. Pushing the processing of captured analytics onto the mobile devices, instead of the cloud, also strengthens the idea of data ownership. The analytics manager will also utilise a preferences user interface, that could, for example, allow end-users to moderate what data is collected (or which apps collect data) and where their data is shared/exported to.

Cross-app analytics: Further to the on-device processing of analytics, apps can benefit from analytics collected for other Mobile Age apps. For instance, one app can collect its own analytics of how an end-user interacts with the app, and a second app can collect its own analytics too, however, the first and second app can share their analytics with each other (with the end-user's permission). This potentially enables richer recommender components, and greater contextual understanding of a system – e.g. what an end-user doing in the previous app before navigating to the current app. Privacy is important here, so there will not be a repository of analytics that any app can read. Instead, app developers will make their analytics events selectively accessible to other apps.

IoT analytics to mobile context: Another property of the above conceptual architecture is that it makes IoT analytics available in the mobile context, so on-device recommender components can directly benefit from IoT analytics.

Recommender components: Each installed Mobile Age app could specify their own recommender components. This would enable prescriptive analytics by utilising analytics data, captured in the end-user's domain, and making recommendations to the parent Mobile Age app. A recommender component could take input from the end-user's domain alone, making personalised recommendations, or accept multiple end-users' analytics to make general recommendations; these would preserve privacy by passing through a privacy mediator before reaching a recommender component.

4.3.1 Extending an Existing Technology: Pheme

The Mobile Age analytics module conceptually extends an existing cloud-based service called 'Pheme' [1], a generic end-user analytics platform previously developed by ULANC, to support on and offline interactions with public displays. Specific implementation details of the Mobile Age analytics module are given in section 4.3.4 and onwards. Pheme (section 4.3.1) repurposes web analytics for IoT data collection, and is being extended to support new interaction types required to capture and store end-user interactions across both on and offline mobile and web-based public services. Such novel interaction analytics information will be applied to drive simplified end-user workflows between public services and provide insightful analytics about service use and performance.

Pheme uses a modified version of the Universal Measurement Protocol (UMP) [2] for describing and structuring the data, which has been extended with fields relevant for IoT and mobile analytics, such as for describing proximity and location data of passers-by.

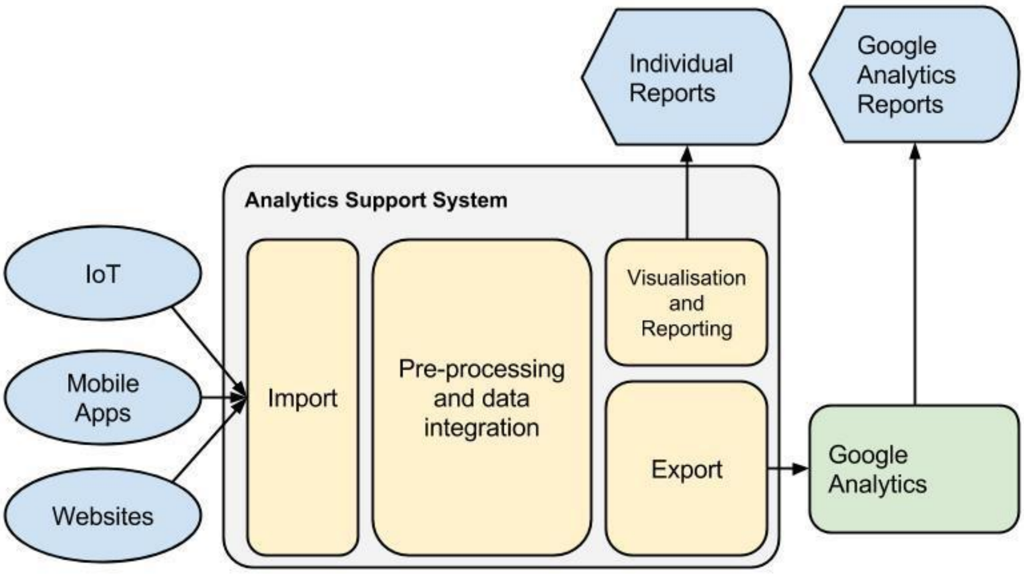

Figure 4.8: Pheme's server architecture

4.3.2 Integration with OSCPSEP

The analytics module will be loosely connected in OSCPSEP as a SaaS. The analytics-specific requests and events from Mobile Age apps will be communicated directly to the analytics service, not through OSCPSEP – this is to reduce overheads, and therefore latency, of communicating with the analytics server.

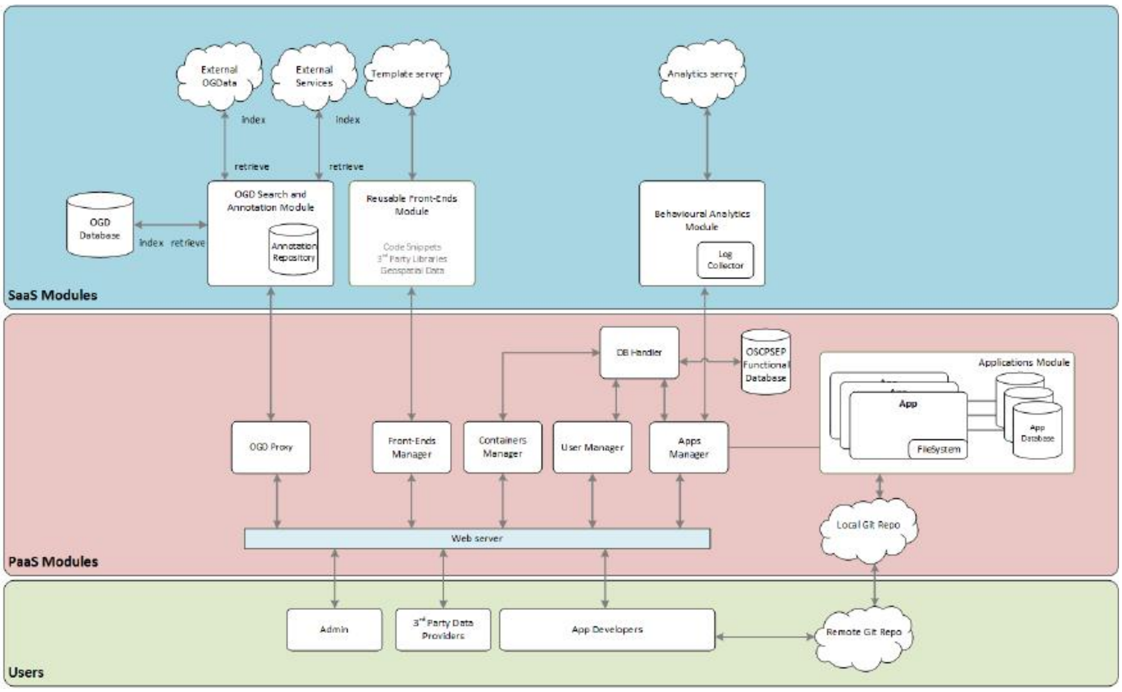

Figure 4.9: OSCPSEP overall architecture

The analytics module instead integrates into the OSCPSEP ecosystem by providing developers of Mobile Age apps the ability to register their app to use analytics. This is presented to developers through a web front-end on the OSCPSEP website. If they choose to include analytics in their apps, they will be shown a webpage to register for a Tracking ID. They will then have access to API documentation and installation and usage guides (described in Appendix II).

4.3.3 Requirements

Requirements for OSCPSEP, which includes the analytics module, were identified in D2.1 'OSCPSEP requirements'. , below, is a reminder of the requirements from that deliverable that pertain specifically to the analytics module. For full details of these requirements, and the elicitation process, please refer to 'D2.1 OSCPSEP requirements'.

ID: Title	*Description*
SSC-FR-22: Front-end interaction monitoring	OSCPSEP must be able to perform behavioural analysis by monitoring the interaction of the applications front-end components with end-users. (e.g. clicks on every element, unused buttons etc...).
SSC-FR-23: Behavioural models export format	OSCPSEP must be able to export behavioural models created by monitoring front-end interaction behaviour. Those models must be exported in XML/JSON format to be reportable and reusable.
SSC-FR-24: Public services analytics	OSCPSEP must collect analytics referring to public services usage (e.g. profiling on users and performance).
SSC-FR-25: Public service usage analytics reporting and visualization	OSCPSEP must provide reports and visualization tools regarding public services usage analytics.
PER-NFR-09: Privacy – Anonymous Reporting	The SYSTEM must ensure that it is possible to provide anonymous reporting of analytics data such that the privacy of individual end-users is not compromised.

Table 4.7: Relevant OSCPSEP requirements

4.3.4 Architecture

The system architecture of the Analytics Module, that will reside on the server-side (earlier referred to as Pheme 2.0), consists of three main components (): *Import*, *Analytics Pipeline*, and *Reporting*.

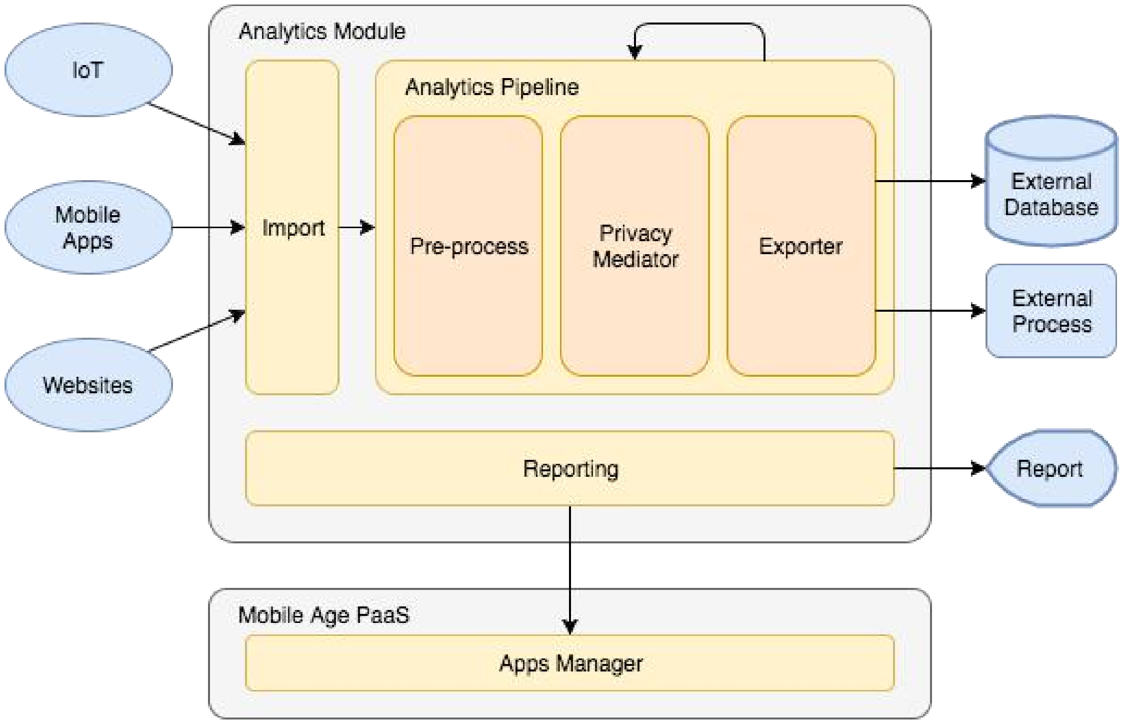

Figure 4.10: The overall architecture of the Analytics Module

The analytics module fits into OSCPSEP by connecting to the PaaS component 'Apps Manager'.

4.3.4.1 Import

Client devices and IoT sensors report events and data to the import component through the analytics module's RESTful API. The data is formatted using an extension to the Google Universal Measurement Protocol providing a richer set of data types, for example, to describe end-user interactions across IoT devices; this satisfies the design consideration "Data Integration" (section 4.2.5). Each request must include a tracking ID (unique per application) and can be extended by a unique client identifier (end-user identifier) allowing to map requests from multiple sources to the same individual.

When an analytics event is received in the Import component, the Tracking ID received with the event is used to determine which analytics pipeline should process this event.

4.3.4.2 Pipeline

The analytics pipeline is a collection of classes that implement a Pipe interface (i.e. Pre-process, Privacy Mediator, Exporter), which receive and process an Analytics Event object. There can be multiple implementations of each pipe, allowing other developers to specify their own custom implementations if they desire; which ones to use and the order in which they are used is specified by the owners of Tracking IDs in a pipeline schema.

A pipeline schema is a simple list class names that implement the Pipe interface (Figure 6.6). An analytics event is then passed sequentially to each pipe in that list. A more complex branching pipeline can consist of nested pipelines, which for example can duplicate an Analytics Event at specified points and then process the event along a parallel pipeline (). This is useful, for example, if an event is to be exported to multiple external entities.

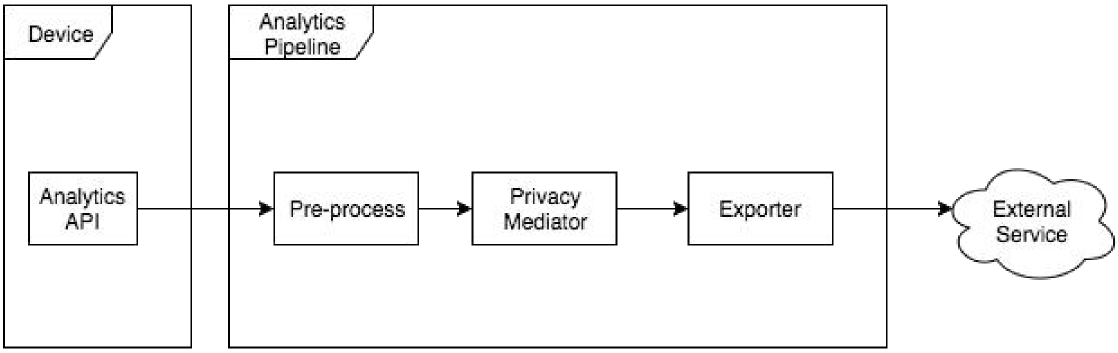

Figure 4.11: Analytics Pipeline – Linear

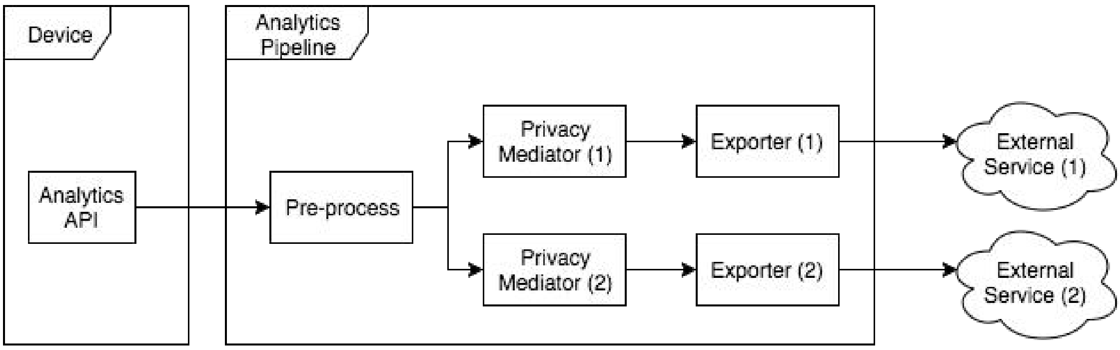

Figure 4.12: Analytics Pipeline – Branching

The default pipeline, if none is specified by the owner of a Tracking ID, uses just a single pipe (*DefaultPreProcess*) – this simply performs some default anonymization (such as anonymising IP addresses) before storing the analytics event. This default pipeline can be overridden by the owner of the Tracking ID, if they desire; for example, if you are doing analytics of public displays, and may wish to not anonymise IP addresses by default. The ability to define custom pipelines provides an architectural capacity to support the design consideration "Protecting Commercially Sensitive Data" (section 4.2.2) and "Aggregation Rules and Policies" (section 4.2.3) – essentially, a platform user could agree to share a subset of their analytics data by defining a pipeline branch that utilised separate and customised privacy mediator and exporter components.

4.3.4.2.1 Pipeline: Pre-process

Each incoming analytics event triggers a set of pre-processing and data integration components used for parsing data into analytics event objects, making data that is integrated into an event usable by other system components. The modular design of the analytics pipeline makes it possible to easily add additional processing components. One example for a pre-processing component could be a 'correlation engine' that combines multiple (historic) analytics events from the same user but from multiple devices, enabling cross-device analytics.

4.3.4.2.2 Pipeline: Privacy Mediator

This component provides various privacy masking and filtering capabilities. Privacy mediator implementations can anonymise, edit, or remove existing fields of an Analytics Event (e.g. location information), or provide more extensive analytics sharing capabilities by changing or assigning a new Tracking or Client ID. These components will satisfy design consideration "Protecting End-User Privacy" (section 4.2.1). This would be useful for events that are exported outside of the analytics module, or to re-import an analytics event recursively into the analytics module. Copies of analytics event may have already been persisted (e.g. after a pre-process component), therefore having a privacy mediator that changed the Tracking ID of an event would allow it to be resubmitted to the analytics module, but then ensuring it is only visible and accessible to the platform users who are registered with the new Tracking ID. This could lead to some data duplication, but allows (modified) events to be shared with other external stakeholders whilst remaining in an analytics context.

4.3.4.2.3 Pipeline: Exporter

Export components can be used to automatically push processed analytics data in real-time to third-party services, e.g. for visualisation purposes or additional processing. Such export components can be specific for an application or tracking ID, and allow the definition of specific mappings for ensuring compatibility with other services. Each instance of a tracking ID can be associated with one or multiple Export components, allowing different mappings to be used, simultaneously providing support for different reporting requirements.

Visualisation will be provided, per requirement SSC-FR-25 (Public service usage analytics reporting and visualization). This is primarily achieved using a Google Analytics exporter component, which takes analytics events and exports them in real-time to the Google Analytics service. This allows us to view analytics reports using powerful visualisations available on an existing platform.

Exporter components can also be used to recursively ingest data (if the pipeline specifies it). For example, this could be used in 1) a pipeline that masks some sensitive data, then resubmits the data under a different tracking or client ID (e.g. for data sharing purposes), or 2) sharing data with a separate and external instance of the Mobile Age analytics software (allowing for distributed collection, sharing, and processing). This capability satisfies the design consideration “Architectural Models” (0) and “Data Integration” (0).

A further example usage of an exporter component would be for exporting certain events directly into a Mobile Age app, and used as input for (e.g.) a recommender system. This could enable end-user specific suggestions and personalised user interfaces.

4.3.4.3 Reporting

The Reporting component provides an output API to obtain data from the internal data store. This provides information on-request.

This reporting component can be used by app developers to retrieve all or parts of the internally stored data relevant to provided Tracking IDs. In addition, visualisation components can be added to visualise internal analytics results instead of using third-party services, providing support for custom reports as needed.

4.3.5 Interface

The interface to the analytics module is split into two main areas: the RESTful API exposed by the web-server (section 4.3.5.1), and functions exposed through developer-ready client library APIs (section 4.3.5.2), e.g. JavaScript, Python. Installation and usage instructions for these client-side APIs are given later in Appendix II.

4.3.5.1 Web-server Interface

Communication with the analytics module will be in the form of HTTP requests, originally described in D2.2 ‘Interim OSCPSEP technical specification’. These requests are wrapped in the developer-ready APIs.

shows the breakdown of the web servers end-point which requests internally stored raw analytics data for a given tracking ID. This API functionality is basic, but is expected to be extended in future to support more complex queries (e.g. allowing applications to self-adapt by requesting top used app features).

Endpoint	*/analytics/:tid/:api-key*
Method	GET
Headers	Content-Type: application/x-www-form-urlencoded
Parameters	**tid {String}:** Tracking ID for querying the analytics data store. **api-key {String}:** Token used to determine access rights to this API.
Data	N/A
Example	/analytics/74786db4-c2ad-4740-baac-92af645dd0f3/sh76ja190olx65g8

Table 4.8: Request most recent analytics data for a Tracking ID

shows the breakdown of the request to log analytics events for a single end-user of a Mobile Age app. This API is based on the Universal Analytics Protocol and was extended to support additional use cases.

Endpoint	*/analytics/*
Method	POST
Headers	Content-Type: application/json
Parameters	N/A
Data	**v {Float}:** API version that is used. **tid {String}:** Tracking ID (app specific and used for authenticating the request) **qt {Integer}:** Queue time in milliseconds; time delta between the time in which the event occurred and the time when the event was reported. **z {Integer}:** Random number to make sure browsers don't cache the request. **cid {String}:** End-user unique identifier used to recognise requests from the same user. **dr {String} [optional]:** Request referrer. **t {String}:** Type of event, e.g. \`event', \`proximity' or \`interaction' – this will determine which of the following parameters are required *<u>'Event' Specific:</u>* **ec {String}:** Event category. **ea {String}:** Event action

	ev {Positive integer}: Event value **el {String}:** Event label *'Proximity' Specific:* **pd {String}:** Relative location of end-user to a device (e.g. display) **pm {String}:** Metric used for distance (m, cm, ft, interaction areas such as \`far away', \`close', \`walking by', \`touch interaction') **dv {String}:** Direction of view of passers-by **pt {Integer}:** Proximity time in seconds, e.g. time spent in the area *'Interaction' Specific:* **it {String}:** Interaction type, e.g. "touch" or "move" **ix {Float}:** x-coordinate of the touch event **iy {Float}:** y-coordinate of the touch event

Example	/analytics/
	{
	"tid": "ca2ca6a5-9ab0-47ff-960d-9ce34f3cb5a6",
	"cid": "b17dd04a-2b85-4741-82e5-52f7e06f8d76",
	"v": 1.0,
	"qt": 0,
	"z": 0,
	"t": "event",
	"ec": "Page view",
	"ea": "click"
	}

Table 4.9: Push analytics data to the server

Note that for pushing an analytics event to the server, as described in , certain parameters in the JSON data will be required dependant on the type of event that is being submitted – a break down is given in . Any parameters for an event not being submitted are automatically ignored.

Event Type	*Required*	*Event JSON Data Parameters*
Event	Required	**ec {String}:** Event category. **ea {String}:** Event action
	Optional	**ev {Positive integer}:** Event value **el {String}:** Event label
Proximity	Required	N/A
	Optional	**pd {String}:** Relative location of end-user to a device (e.g. display) **pm {String}:** Metric used for distance **dv {String}:** Direction of view of passers-by **pt {Integer}:** Proximity time in seconds, e.g. time spent in the area
Interaction	Required	**it {String}:** Interaction type, e.g. “touch” or “move”
	Optional	**ix {Float}:** x-coordinate of the touch event **iy {Float}:** y-coordinate of the touch event

Table 4.10: Analytics Event – Required or optional parameters

Which parameters to use or not is simplified when using the client APIs, and clear in their documentation – summarised in the following subsection.

4.3.5.2 Developer Library API

The client APIs simplify the interaction with the analytics module. Several of the data attributes required by the web-server API will be automatically populated by the developer client-side libraries (e.g. API version, and queue time). In the following sections, there is a description of the main functions exposed to the Mobile Age platform users.

4.3.5.2.1 Initialising the Analytics Library

Before using the analytics libraries, they must be initialised (). The libraries only require a Tracking ID (tid) to be initialised. A Client ID (cid) can be provided to identify an end-user, though if one is not provided, then a UUID is auto-generated.

Object	*Analytics()*
Description	Initialises the client library with the necessary information to identify this instance.
Parameters (Required)	**tid {String}:** Tracking ID

Parameters (Optional)	**cid {String}:** End-user unique identifier used to recognise requests from the same user.

Table 4.11: Client-side API – Initialise the library

4.3.5.2.2 Recording an Analytics Event

The client APIs simplify the reporting of analytic events by providing individual functions for each of the supported event types developers wish to report. Currently, the following three main functions are available: event(), proximity(), and interaction(). After the library is initialised, these functions will automatically populate an event with all the necessary information, place them on a message queue, and submit them when possible. Each event type will require a few additional pieces of data, which are passed to the library as parameters to the following function descriptions (; ;).

Method	*event()*
Description	Sends a generic analytics event to the analytics server.
Parameters (Required)	**ec {String}:** Event category. **ea {String}:** Event action
Parameters (Optional)	**ev {Positive integer}:** Event value **el {String}:** Event label

Table 4.12: Client-side API – Send analytics event (general)

Method	*proximity()*
Description	Sends a proximity analytics event to the analytics server.
Parameters (Required)	N/A
Parameters (Optional)	**pd {String}:** Relative location of end-user to a device (e.g. display) **pm {String}:** Metric used for distance (m, cm, ft, interaction areas such as 'far away', 'close', 'walking by', 'touch interaction') **dv {String}:** Direction of view of passers-by **pt {Integer}:** Proximity time in seconds, e.g. time spent in the area
Note	Whilst all four parameters are optional, at least one must contain a value.

Table 4.13: Client-side API – Send analytics event (Proximity)

Method	*interaction()*
Description	Sends an interaction analytics event to the analytics server.
Parameters (Required)	**it {String}:** Interaction type, e.g. "touch" or "move"
Parameters (Optional)	**ix {Float}:** x-coordinate of the touch event **iy {Float}:** y-coordinate of the touch event

Table 4.14: Client-side API – Send analytics event (Interaction)

4.3.6 Privacy

Privacy is a core requirement of analytics reporting systems, and reflected in the project's requirement PER-NFR-09 ('Privacy – Anonymous Reporting'), identified in D2.1 'OSCPSEP Requirements'. Privacy is initially protected through standard data protection approaches (e.g. secure storage of data). Furthermore, events and reports will only be accessible to relevant platform user accounts, dependent on the Tracking IDs associated with their account.

Another significant aspect of our specific implementation, is the use of privacy mediators. These components will enable the filtering and mediation of analytics events and event fields, allowing data to be shared across apps or for exporting to external partners or entities. Mobile Age developers will be able to define their own privacy mediators, though some default mediators will be available by default; for example, one privacy mediator component will anonymise IP addresses of incoming events.

Furthermore, end-user UI controls for privacy management were co-created with end-user participants in a WP3 workshop in South Lakeland; this also partially satisfies the design consideration "Aggregation Rules and Policies" for end-users (section 4.2.3). During this workshop, we discussed the ability to opt-out of analytics collection and reporting. This broad feature surfaced through a conversation about privacy and security on the internet, and how the participants were uncomfortable being tracked and having their data shared (e.g. by other analytics services for targeted advertising across the web). The general benefits of analytics were further discussed, and what it could mean in this project for end-users specifically. For example, service provision improvements, or personal (even private/anonymous) recommendations. This culminated in the intent to implement analytics privacy controls to give fine-grained control over who is collecting analytics, and even who it could get shared with – providing end-users with a privacy or features trade-off. This feature will continue to be co-created with the end-user participants as the demonstrator applications develop.

4.3.7 Implementation Status

The overall architecture and REST API has been implemented – an early deployment is implemented, with end-to-end tests being performed on the implementation (from data collection, through to processing and visualisation). Further server-side components are currently being implemented, in Python, into a Django[3] web service, using the Django REST

[3] Django: https://www.djangoproject.com/

framework[4]. The current focus is on specifying platform user accounts and restricting access to analytics reports based on Tracking IDs associated to your platform user accounts.

The analytics module is frequently being tested internally to the project, as part of standard development practices, to ensure that the implementation meets the design. The core architecture and example pipeline components have been created and internally tested to demonstrate the architectures end-to-end capabilities for branching and filtering data – mixing privacy mediators with this pipeline, and considering its ability to import and export the data recursively, illustrates a system design that satisfies all of the design considerations. However, acceptance tests will take place once the implementation is complete. The design and implementation of aspects of this analytics module are ongoing, due in part to the co-creation process with the older adults in South Lakeland.

More intricate reporting capabilities are expected to be implemented – currently, reporting relies on using an Exporter component within the pipelines to parse analytics events in real-time and push them to a Google Analytics instance (for visualisation purposes).

The analytics manager (located in the Mobile Age Launcher, or as separate libraries) is not yet complete. Initial development of a standalone JavaScript library is close to completion, but currently it only posts events to the analytics server. The Mobile Age Launcher is being developed with Cordova[5], an SDK for creating cross-platform mobile apps. Therefore, the analytics manager is expected to be developed as a Cordova plugin.

The Mobile Age analytics module will be made available as open source code, and there are no dependencies on external proprietary libraries or services to use the analytics module.

[4] Django REST framework: http://www.django-rest-framework.org/

[5] Cordova: https://cordova.apache.org/

5 Innovation Aspects

We believe that our work on behavioural analytics exhibits a high degree of technical innovation as compared to the state of the art (Figure 5.1). In particular, our framework supports:

(i) the collection of novel data sets that have not previously been used to support engagement with older adults,
(ii) new capabilities for integrating heterogeneous data sets across multiple stakeholders,
(iii) new uses of behavioural analytics to support personalisation of services and design of future public services, and,
(iv) novel ways of enabling users to protect their privacy using techniques that are specifically designed to match the expectations of older adults.

Our focus on a framework for behavioural analytics for older adults represents a significant paradigm shift for the sector. Current approaches rely on using generic analytics toolkits, such as Google Analytics, iOS Analytics and Fabric, that are not tailored to the unique needs of the Mobile Age target user group. We are not aware of any other systems that have adopted a framework-based approach to behavioural analytics for older adults or that offer the capabilities of the Mobile Age platform.

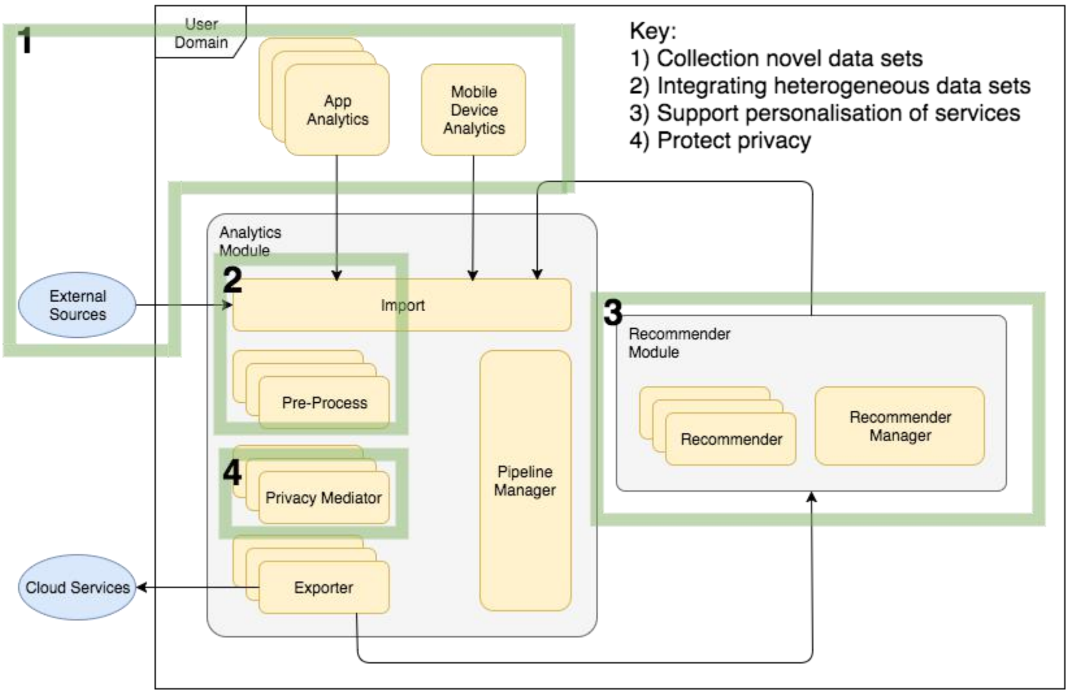

Figure 5.1: Conceptual Architecture - Novelty highlighted

In the following sections, we consider aspects of technical innovation in more detail.

5.1 Novel Data Sets

The state of the art in behavioural analytics is web analytics and IoT analytics. These systems support the ability to collect user interaction logs and IoT sensor data to provide new levels of understanding in the usage of web and pervasive technologies [22]. Currently, public administrations can capture basic events and insights through Web analytics toolkits including raw page visits and in-page behaviours (e.g. button clicks). Typically, such information is reported as aggregated metrics (e.g. total of return visitors or unique visitors) [23]. Furthermore, local authorities are beginning to leverage new IoT analytics data in domains such as transport and logistics. For example, data captured through sensors can report traffic conditions through in-field systems (e.g. inductive loop & video analytics) to better understand transport behaviour. Web and IoT analytics systems can be supplemented with traditional Customer Relationship Management (CRM) software (that can be used to track telephone and face-to-face interactions) and databases of council case records. Our interactions with councils suggest they use a highly limited set of analytics data (see section 3.2).

In summary, their focus is on the use of web analytics to improve the usability of council web sites and crowd-sourced data for understanding local issues. These data sources are all for the general population that a PA governs and there have been no systematic attempts to capture user analytics for specific target groups such as older adults (indeed - most web analytics systems would not enable segmentation of user traces in this way).

Within the context of Mobile Age, we support collecting web analytics and IoT analytics, and supplement these with new forms of analytics data that includes: mobile application usage, specific application analytics events and contextual data including location traces and phone logs.

Contextual data, in particular pervasive events, has typically not been available for PAs through off-the-shelf analytics toolkits. This type of data provides the potential to infer workflows using the mobile apps and interacting with external services, for example, as highlighted in the scenario “Council Telephone Query” (section 4.1.5). These workflows can be further integrated or reported as analytics data.

Supporting the capture of offline-interactions is important for an analytics framework scoped around older adults, as internet connectivity for this demographic is reported lower than younger generations[7], especially if they live in rural locations. This type of analytics capture is not typically supported in existing frameworks as they presume high degrees of connectivity.

Crucially our system also enables the creation and capture of new analytics event types - thus helping to provide a degree of future-proofing for the Mobile Age architecture.

5.2 New Capabilities for Analytics Integration

In order to generate the novel insights and reports that support the scenarios described in section 4.1, the integration of datasets from multiple stakeholders is required. State-of-the-art analytics systems in the area of user interaction and behaviour tracking, however, typically do not integrate datasets across multiple stakeholders [10]. Indeed, privacy concerns often mean that system are specifically designed to reduce the potential for this type of analytics data integration. In Web analytics, datasets are automatically collected either server- or client-side and are owned by a single stakeholder, often the website administrator. Early

7 https://www.ons.gov.uk/businessindustryandtrade/itandinternetindustry/bulletins/internetusers/2017#recent-internet-use-for-those-aged-65-and-over-is-catching-up-with-younger-age-groups

analytics systems focused on data mining website logs that were collected on the server [22], while modern analytics systems such as Google Analytics specifically target "businesses of all sizes to better understand your customers"[8] and collect relevant datasets through applications that are run on the client. In this more modern approach, some effort has been made to support cross-device analytics in which the administrator or developer can provide a user identifier to link relevant datasets from multiple sources (e.g. a website and mobile phone application). However, such identifiers are only used for the correct computation of user sessions, and the sole owner and stakeholder of the data remains the administrator - giving the user limited insights into controlling the kinds of data that is collected and how it is aggregated[9].

In the area of IoT analytics that is specifically focussed around the collection and integration of heterogeneous datasets from a range of IoT-enabled devices and sensors, we observe a similar pattern. Such systems are designed for single stakeholders who are in control of both the devices and the data. IoT support platforms such as Mnubo (http://mnubo.com), SAP IoT Solutions (http://go.sap.com/solution/internet-of-things.html), and Thingworx (https://www.thingworx.com) are targeting single stakeholder and therefore assume their ability of defining integration and aggregation rules. Other analytics systems, such as AGT (http://agtinternational.com/iot-analytics/iot-analytics) and Google Big Data (https://cloud.google.com/solutions/big-data/) support the aggregation across multiple data sources and generation of relevant reports; but represent 'big data' tools and are designed to support application developers instead of decision makers.

Examples for the successful integration of multiple data sources owned by different stakeholders can be found in business-to-business relations and the supply chain management in which it is common to share information across business partners [16, 15]. Typical examples include the automated sharing of inventory levels and point-of-sales statistics across the supply chain to predict the demand of individual products to improve the performance of manufacturing and delivering a product in time [16]. We note that the sharing of such information presupposes existing legal agreements across the involved business partners [17] pointing out the usages and clear benefits of the data sharing to the involved stakeholders [16].

In Mobile Age, we have successfully shown that the approach of integrating datasets can be taken into other application domains such as viewer behaviour analytics in digital signage [10]. By integrating data from multiple stakeholders, we were able to create a novel set of behavioural insights that has not been possible otherwise. To enable and support the synthesis of datasets from multiple distinct stakeholders, we have developed a set of design considerations as part of section 4.2 ("Design Considerations for Analytics Synthesis"), clearly highlighting unique challenges and requirements for future analytics systems. These design considerations have been published as part of peer-reviewed work at an international conference [10] including specific requirements around the data integration and aggregation rules and policies.

Our architecture specifically supports analytics synthesis on the mobile to provide analytical insights into the interactions of older adults within and across applications and council services within a mobile context. This enables service providers to utilise new forms of analytics data to enhance public service provision to older adults through mobile interactions. For example, user profile information captured through one application can be shared with

[8] www.google.com/analytics

[9] https://support.google.com/analytics/answer/3234673?hl=en

other 3rd party mobile applications developed within the context of the Mobile Age framework. Server-side integration is supported using Pheme [1], which provides a novel platform for processing analytics events. Further scenarios are described in more detail in section 4.1 ("Stakeholder Scenarios"). The Mobile Age framework provides a series of supporting modules to facilitate the implementation of services that are sensitive to the actions and needs of older adults. This innovative approach is in contrast to state-of-the-art mobile analytics (e.g. Fabric) that typically support the capture and inference of mobile user analytics within a single application context, and do not provide mechanisms for the integration of analytical insights across applications and services within the wider mobile context.

In summary, we are not aware of any previous efforts into the design and implementation of analytics platforms that support the synthesis of data from multiple stakeholders with the goal of improving user experience and service provision for senior citizens.

5.3 Uses of Data Analytics for Senior Citizens

The collection and synthesis of behavioural analytics data that characterises engagement and interaction with public services offers the potential to deliver new insights that can be utilised to streamline mobile user experience and shape our understanding of designing mobile services for older adults.

Previous mobile and pervasive computing research has explored the application of mobile data analytics to support senior citizen independence with a focus on understanding assisted living technologies [20] that aim to augment daily activities and routines. For example, existing research has been undertaken to understand how data captured through mobile platforms can be applied to support a broad range of older adult mobile applications including cognitive orthotics (i.e. medication reminders), health monitoring (e.g. vital signs monitoring), emergency detection (e.g. fall detection) and emotional well-being (i.e. social connectedness) [21]. However, at present mobile behaviour analytics data has predominantly been explored only within the context of closed, standalone assisted living technologies that aim to support specific senior citizen activities (i.e. daily medication reminders). In contrast, our approach includes an innovative technical framework to explore how a deeper understanding of both individual and combined senior citizen behaviours across heterogeneous public services can be used to effectively personalise senior citizen engagement seamlessly across digital and physical public services.

Our architecture supports three distinct areas of innovative analytics use:

- Access to a shared local data analytics store across heterogeneous mobile applications enables service providers to utilise new forms of analytics data to enhance public service provision to older adults through greater personalisation (see section 4.1 "Stakeholder Scenarios" for more detailed example use-cases)
- Mobile developers can apply new forms of data analytics to inform the future design and development of mobile services through a deeper understanding of user actions and behaviours
- The availability of in-depth data analytics that characterise how older adults engage with public services (e.g. Meals On Wheels) provides new insights managing local authorities can leverage to deliver service improvements, support more effective coordination of resources and better inform decision-making.

The 'Behaviour Analytics and Workflow Software' components of the Mobile Age framework have been designed (see section 4.3) to support a novel collection of development modules (e.g. Recommender/Exporter component see section 4.3.4.2.3) to capture and utilise behaviour analytics data currently unavailable to public service providers. In particular, these components enable Mobile Age end-user applications to capture and characterise specific usage events (e.g. detection of a user stuck filling a form), gain authorised access to a user's personal information (e.g. health information) and synthesise local and external data stores to support greater real-time personalisation (e.g. auto-playback of text in audio for visually impaired) and contextualisation of mobile service interactions.

5.4 Privacy

Privacy in analytics systems is extremely important, especially as the diffusion of private information is increasing [19]. This view is supported by our co-creation engagements with older adults - ownership and control over their personal data was very important to them.

Privacy in user analytics is typically protected by avoiding the collection of Personally Identifiable Information (PII), for example by anonymising data (such as IP addresses, and using IDs instead of real names). Analytics are then usually sent to analytics servers and securely stored, so only those with the adequate authentication credentials have access to them. Analytics systems typically then produce aggregated reports that hide individual identities.

Website cookies, which are used in web analytics to determine unique users, are protected from other websites reading them. This protection is built-in into web browsers (e.g. cookies can only be read by web domains that created them). A further privacy protection is afforded by opt-out mechanisms, a way for a user to explicitly set that they do not want data collected from them or transmitted.

Additional protection of privacy is application-dependent. PII can often inadvertently be sent in typical analytics events. These situations are often governed by best practices; for example, Google Analytics provides a list of best practices[10] which include suggestions such as using IDs, avoiding PII in website title or URL, and collecting fine-grained location data (i.e. GPS).

Mobile Age analytics uses Privacy Mediators [18] to protect users' privacy. A Privacy mediator imposes a privacy policy on a data stream, and will strip or mask certain data (for example, PII). These privacy mediators are integrated into the Mobile Age analytics pipeline, in the conceptual user domain (see section 4.3). Multiple privacy mediators can be placed in a pipeline, which provides support for different privacy policies to be imposed by other data sources or stakeholders. These mediators can also be stacked, which will ensure a data or event stream can be used by multiple stakeholders or recommenders. The key innovation in the use of privacy mediators lies in the recognition that their approach of intercepting (mediating) data at the point of capture directly maps on the requirements of older adults who wish to maintain "ownership" of data and are often uncomfortable with exporting data to the cloud.

The concept of privacy mediators was originally developed by members of the Mobile Age team in 2016 and were developed for use in fixed network environments. We are not aware of any other research teams that are currently working on the use of privacy mediators in the mobile domain.

[10] https://support.google.com/analytics/answer/6366371

6 Conclusions

This deliverable outlines work done in task T2.5 0 Behaviour Analytics and Service Workflow Engine Development. As part of this work we have created a series of scenarios that illustrate the importance of analytics synthesis. Based on these scenarios, we developed a set of design considerations for the Mobile Age analytics module. For example, these include the importance of protecting both end-user privacy and commercially sensitive data. These considerations have been used to inform the design and implementation of the Mobile Age analytics system. This includes novel mobile and server side components that collectively provide a new framework for mobile analytics in the domain of supporting older adults. Our specific technical innovations within the framework include:

(i) supporting the collection of novel data sets that have not previously been used to support engagement with older adults,

(ii) new capabilities for integrating heterogeneous data sets across multiple stakeholders,

(iii) new uses of behavioural analytics to support personalisation of services and design of future public services, and,

(iv) novel ways of enabling users to protect their privacy using techniques that are specifically designed to match the expectations of older adults.

The analytics module described will serve as a foundation for collecting and using analytics throughout the entire lifespan of the Mobile Age project.

References

[1] Mateusz Mikusz, Sarah Clinch, Rachel Jones, Mike Harding, Christopher Winstanley, and Nigel Davies (2015). Repurposing Web Analytics to Support the IoT. *Computer*, *48*(9), 42–49. DOI: 10.1109/MC.2015.260

[2] Google Inc. (2015). Measurement Protocol Parameter Reference. https://developers.google.com/analytics/devguides/collection/protocol/v1/parameters (September, 2015).

[3] Ann Cavoukian. (2011). White Paper: Anonymous Video Analytics (AVA) Technology and Privacy. *Technical Report*. Information and Privacy Commissioner of Ontario, Canada.

[4] Nigel Davies, Nina Taft, Mahadev Satyanarayanan, Sarah Clinch, and Brandon Amos (2016). Privacy Mediators: Helping IoT Cross the Chasm. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications (HotMobile '16)*. ACM, New York, NY, USA. 39—44. DOI: 10.1145/2873587.2873600

[5] Retail Gazette (2012). Westfield Stratford attracts 47m shoppers in first year. (Sept. 2012). https://www.retailgazette.co.uk/blog/2012/09/20444-westfield-stratford-attracts-47m-shoppers-in-first-year

[6] Anupam Ghosh and Jane Fedorowicz (2008). The role of trust in supply chain governance. *Business Process Management Journal*, 14(4), 453—470. DOI: 10.1108/14637150810888019

[7] Susan Cohen Kulp, Hau L. Lee, and Elie Ofek (2004). Manufacturer Benefits from Information Integration with Retail Customers. *Management Science*, 50(4), 431—444. DOI: 10.1287/mnsc.1030.0182

[8] Hau L. Lee, V. Padmanabhan, and Seungjin Whang (2004). Information Distortion in a Supply Chain: The Bullwhip Effect. *Management Science*, 50(12_supplement), 1875—1886. DOI: 10.1287/mnsc.1040.0266

[9] Hau L. Lee and Seungjin Whang (2000). Information Sharing in a Supply Chain. *International Journal of Manufacturing Technology and Management*, 1(1), 79—93. DOI: 10.1504/IJMTM.2000.001329

[10] Mateusz Mikusz, Sarah Clinch, and Nigel Davies (2017). Design considerations for multi-stakeholder display analytics. In *Proceedings of the 6th ACM International Symposium on Pervasive Displays* (PerDis '17). ACM, New York, NY, USA, Article 18. DOI: 10.1145/3078810.3078830

[11] Ramesh Sharda, Daniel Adomako Asamoah, and Natraj Ponna (2013). Business Analytics: Research and Teaching Perspectives. In Proceedings of the 35th International Conference on Information Technology Interfaces (ITI). IEEE, Cavtat, Croatia, pp. 19-27. DOI: 10.2498/iti.2013.0589

[12] Dursun Delen and Haluk Demirkan (2013). Data, information and analytics as services. In *Decision Support Systems*, Volume 55, Issue 1, Pages 359-363, ISSN 0167-9236. DOI: 10.1016/j.dss.2012.05.044

[13] Luigi Atzori, Antonio Lera, and Giacomo Morabito (2010). The Internet of Things: A Survey. In *Computer Networks*, vol. 54, no. 15, 2010, pp. 2787–2805. DOI: 10.1016/j.comnet.2010.05.010

[14] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami (2013). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. In *Future Generation Computer Systems*, vol. 29, no. 7, 2013, pp. 1645–1660. DOI: 10.1016/j.future.2013.01.010

[15] Susan Cohen Kulp, Hau L. Lee, and Elie Ofek. 2004. Manufacturer Benefits from Information Integration with Retail Customers. Management Science 50, 4 (2004), 431–444. DOI: http://dx.doi.org/10.1287/mnsc.1030.0182

[16] Hau L. Lee and Seungjin Whang. 2000. Information sharing in a supply chain. International Journal of Manufacturing Technology and Management 1, 1 (2000), 79–93. DOI: http://dx.doi.org/10.1504/IJMTM.2000.001329

[17] Anupam Ghosh and Jane Fedorowicz. 2008. The role of trust in supply chain governance. Business Process Management Journal 14, 4 (2008), 453–470. DOI: http://dx.doi.org/10.1108/14637150810888019

[18] Nigel Davies, Nina Taft, Mahadev Satyanarayanan, Sarah Clinch, and Brandon Amos. 2016. Privacy Mediators: Helping IoT Cross the Chasm. In Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications (HotMobile '16). ACM, New York, NY, USA, 39-44. DOI: 10.1145/2873587.2873600

[19] Balachander Krishnamurthy and Craig Wills. 2009. Privacy diffusion on the web: a longitudinal perspective. In Proceedings of the 18th international conference on World wide web (WWW '09). ACM, New York, NY, USA, 541-550. DOI: 10.1145/1526709.1526782

[20] Rashidi, Parisa, and Alex Mihailidis. "A survey on ambient-assisted living tools for older adults." IEEE journal of biomedical and health informatics 17.3 (2013): 579-590.

[21] Dai, Jiangpeng, et al. "PerFallD: A pervasive fall detection system using mobile phones." Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on. IEEE, 2010.

[22] R. Cooley, B. Mobasher, and J. Srivastava. “Web mining: information and pattern 28 discovery on the World Wide Web”. In: Proceedings Ninth IEEE International 29 Conference on Tools with Artificial Intelligence. Nov. 1997, 30 pp. 558–567. DOI: 10.1109/TAI.1997.632303.

[23] Jason Burbie, Angie Brown, and WAA Standards Committee. Web Analytics De nitions – Version 4.0. Tech. rep. Web Analytics Association, 2007.

APPENDIX I

Installation: Full Code Samples

This appendix contains complete code samples in JavaScript and Python for including the Analytics libraries into Mobile Age apps. Each example consists of a minimum working example, with a single example of how to report an analytics event.

JavaScript Example

File: maanalytics.js

This file is the JavaScript library which provides the API for communicating with the Analytics service. This file will be provided to the application developers.

File: index.html

```
<html>
    <head>
        <!-- Analytics library -->
        <script type="text/javascript" src="maanalytics.js"></script>
        <!-- Developer's custom code -->
        <script type="text/javascript" src="sample-script.js"></script>
    </head>
    <body>
        <!-- Sample button -->
        <input type="button" id="myButton" value="Click me"></input>
    </body>
</html>
```

File: sample-script.js

```
// Initialise the Analytics library
var maa = new Analytics({
    tid: "YOUR-TRACKING-ID-HERE",
    cid: "YOUR-CLIENT-ID-HERE"
});

// Log an analytics event on a mouse click
var myButton = document.getElementById("myButton");
myButton.addEventListener("click", function(){
    // Variables to be passed as parameters
    var CATEGORY = "routing to location";
    var ACTION = "click";
    var VALUE = 0;
    var LABEL = "This can be used for supplemental information";
    // Log an event to the Analytics service
    maa.trackEvent(CATEGORY, ACTION, VALUE, LABEL);
});
```

Python Example

File: analyticsapi.py

This file is the Python library which provides the API for communicating with the analytics module. This file will be provided to the application developers.

File: sample-script.py

```python
from analyticsapi import MAAnalytics

# Initialise the analytics library
maa = MAAnalytics(tid="YOUR-TRACKING-ID-HERE",
                  cid="YOUR-CLIENT-ID-HERE")

# Log an analytics event
maa.trackEvent(category="my category",
               action="my action",
               value=0,
               label="supplemental information")
```

APPENDIX II

Installation and Usage

This appendix contains a short description and some example usage of the analytics software, to help illustrate how a developer registers and uses the analytics module within their Mobile Age app.

Code snippets are provided in this deliverable, using JavaScript syntax – other languages will be available when the analytics module is released.

Registering to Use the Analytics Module

A registered developer using the Mobile Age platform (OSCPSEP) will be able to additionally register their intent to use the analytics module, described in this deliverable, for their app(s). Whilst at the dashboard of the Mobile Age front-end, there will be a list of options that the developer can access or register. One of these will be the analytics module.

Upon selecting the "Analytics" option in the OSCPSEP UI, the developer will be presented with an option to register for the analytics module – initially this will just be a Yes/No message box, with an option to visualise the data by exporting the analytics data in real-time to Google Analytics for visualisation and reporting purposes. If they choose to export to Google Analytics, the developer will need to copy/paste the Tracking ID provided by their Google Analytics account (which is separate from the tracking ID generated by the analytics module). They will also be able to select "Documentation" to view the documentation and usage examples for themselves.

Upon clicking "Register", the Mobile Age platform user will have their Mobile Age account associated with the analytics module, and will have a Tracking ID generated for them which they can copy there and then. This will be stored and accessible in their account so they can also copy it later. In addition, if the developer is not making an app for the Mobile Age Launcher, the developer will be provided a link to download a library for them to include in their projects. For example, if they require a library in the JavaScript language, they are given file 'maanalytics.js' (Mobile Age Analytics).

The analytics module API works by utilising HTTP requests with JSON content. Whilst the analytics server's HTTP requests will be documented, allowing custom interaction libraries or AJAX requests, initially the above JavaScript library will be provided to streamline installation and requests. This is because Mobile Age demonstrator applications will initially have web-based user interfaces. So, for the purposes of this scenario we will show how to install and use the Analytics module with a mobile application which has a web front-end.

Whilst a direct HTTP/POST message can be sent directly to the server, in a mobile context the request will fail if there is no internet connectivity. This also defeats some of the significant benefits of using the Mobile Age analytics, such as on-device pre-processing, privacy mediation, etc. Otherwise, an immediate benefit to using the provided libraries is the built-in caching and message queueing of logged analytics events that will be sent to the analytics server, until an internet connection becomes available.

Installing Analytics into Apps

Once a developer has registered for the analytics module, retrieved their Tracking ID, and downloaded the JavaScript library 'maanalytics.js', they are ready to start installing and using the module in their codebases. Firstly, the platform user will copy the library into their codebase. Next, they will copy the below code snippet into each webpage that analytics is desired on.

```
<script type="text/javascript" src="maanalytics.js"></script>
```

This element should be appended to the bottom of the `<body>` element, or in the `<head>` element. Note: depending on where the file is on the filesystem in relation to the developer's project, they may need to update the `src` attribute of the `<script>` tag, for example: `src="js/maanalytics.js"`.

A simple and complete JavaScript/HTML code sample is available in Appendix I, which demonstrates the Analytics library being referenced in code and reporting a single event.

Initialising the Analytics Script

At this stage the analytics library is ready for initialising; the analytics script will not log any analytics or communicate with the analytics server until this is done. This is achieved by calling `new Analytics({/*OPTIONS*/})` (described in more detail below), preferably after the DOM has loaded. Below are examples of where to insert the analytics initialisation call.

Plain JavaScript 'DOMContentLoaded' example:

```
document.addEventListener("DOMContentLoaded",
  function(event){
    // initialise analytics here...
  }
);
```

JQuery document 'ready' example:

```
$(document).ready(function(){
    // initialise analytics here...
});
```

When initialising the analytics script, you must pass a JSON object containing options/preferences. A full list of the options and their effects will be detailed in full in the documentation.

The only mandatory option for this library is the Tracking ID (a client ID will be auto-generated if not provided). A developer can initialise the library using the following code snippet (replacing '`YOUR-TRACKING-ID-HERE`' with the Tracking ID that was provided earlier):

```
// Initialise the Analytics library
var maa = new Analytics({
    tid: "YOUR-TRACKING-ID-HERE",
    cid: "YOUR-CLIENT-ID-HERE"
});
```

A tracking ID takes the form of a UUID, and will look similar to this: 2a9a2431-0a04-443d-8fc9-1c66cec60787

Other options that will be available, for example, include `logAllButtonClicks: true`. This allows the library to look for existing and future buttons within the DOM, and then log an analytical event upon each button click event. Each button click reported will ensure that each button is uniquely referenced, to ensure traceability from the analytics reports back to specific end-user interface components. This is a convenience option, but will increase noise in the analytics reports – it is up to the developer to make that decision if it benefits them.

Recording Analytics

Once the analytics library is installed and initialised, the developers can now have finer control over what they desire to be logged. In addition to analytics that are automatically logged upon initialising the library (i.e. a page view), the library offers a further set of customisable API options.

Default Functions

The JavaScript library offers four main methods for recording analytical events:

```
maa.trackPageview( ... ) // page view
maa.trackEvent( ... ); // generic event
maa.trackDProximity( ... ); // proximity event
maa.trackDInteraction( ... ); // interaction event
```

For example, if a developer desires to log when a specific button is clicked, they would add a call to `maa.interaction()` in the click listener for that button, e.g:

```
// Log an analytics event on a mouse click
var myButton = document.getElementById("myButton");
myButton.addEventListener("click", function(){
    // Variables to be passed as parameters
    var CATEGORY = "routing to location";
    var ACTION = "click";
    var VALUE = 0;
    var LABEL = "This can be used for supplemental information";
    // Log an event to the Analytics service
    maa.trackEvent(CATEGORY, ACTION, VALUE, LABEL);
});
```

In this example, the parameters `CATEGORY` and `ACTION` passed to `maa.trackEvent()` are mandatory, and `VALUE` and `LABEL` are optional. However, the content of the parameters can be whatever the developer wishes, as long as they conform to the correct types (string, int etc.). The developer can define their own categories and actions.

Requesting Analytics

Once Mobile Age applications have been registered, a developer will be able to request reported analytics for their applications. This is achieved programmatically, directly to the Mobile Age analytics server (section 0), or visually through a 3rd party service (section 0).

Programmatically Requesting Analytics

The analytics server accepts HTTP/GET requests for information. To make this simpler, libraries are made available to make these requests easier.

A request to the server to retrieve analytics data will return a paginated JSON response; the URI to access each subsequent page of results is contained within the JSON response from the analytics server.

Visually Accessing Analytics

Alternatively, a developer may wish to visually view analytics data. In these situations, the developer can view their data on Google Analytics by associating a Google Analytics Tracking ID, retrieved from their Google Analytics account. This would be pasted into the relevant field when initially registering the analytics module for their Mobile Age application, but this will also be available to do after the initial registration.

After associating a Google Analytics account, you will be able to view the data on the Google Analytics website.

APPENDIX III

Public Authority Responses: ZGZ

The following are the verbatim responses received from Zaragoza public authority regarding their analytics usage.

1. **What are the various ways you capture how people interact with the council?**

   Online:

   - City claims and suggestions service
   - Comments in different services like city Events or Facilities.
   - Social networks: Twitter, Facebook, Youtube, Flickr
   - Web analytics (Google Analytics, Yandex)
   - Usability tests (Focus Group, Heuristic tests)

   Phone:

   - 010 information service

   Presencial:

   - City council spaces (Juntas, Civic centers, Senior centers, Young centers)

2. **What kind of analytics and reports do you currently collect regarding an individual's (anonymous) user experiences?**
   - Annual Report of visits to the complete website (Google Analytics)
   - On-demand reports of the different portals (Google Analytics)
   - Usability reports reported by Yandex (heat maps).
   - Claim and suggestions reports.
   - Web Accessibility reports.
     1. **...do you collect these just within a single website?**

        Yes, all the portals belongs to www.zaragoza.es

     2. **...do you collect these across websites within your domain (e.g. tracing a single user's experience from one of your websites to another)?**

        Yes

     3. **...do you collect analytics for individual user experiences across partner websites (e.g. a user starting from SLDC website and moving to the AgeUK website)?**

        No

     4. **...thinking beyond the websites, ...**
        1. **do you correlate phone calls or physical visits with web activity**

           No

        2. **do you collect any cyber-physical analytics (e.g. Internet of Things, sensors, RFID).**

Air quality or traffic density are obtained through sensors. But they are not focused on personal data.

3. Of the above (2.1-2.4), would you like to be able to collect these types of analytics?

- Yes, in order to know better our users experience we are open to improve our analytics reports.

4. Why would you like to collect these types of analytics

We need to know our clients and define roles to:

1. Give a personalized answer to their needs
2. Detect their actual and future needs.

Ejemplos de perfiles: administrado, ocio, deportista, comerciante, mayor, joven

[*Translation*: Examples of profiles: managed, leisure, sportsman, merchant, senior, young]

5. Can you think of any concrete scenarios that could benefit from access to these types of analytics

1. The analysis of the analytics data that would be collected would be useful for the update of our eAdmin Style Guide.
2. Prioritise the mobile-first design of our website, with responsive designs and technology.
3. Ranking of public services offered to citizens, according to their preferences and patterns of usage of the ones offered by our site.