Consume

Wetzels, M.H.; Ayoola, I.; Bogers, S.; Peters, P.J.F.; Chen, W.; Feijs, L.M.G.

Published in:
Pervasive and Mobile Computing

Document license:
CC BY

DOI:
10.1016/j.pmcj.2017.11.002

Published: 01/01/2018

Document Version
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 29. Dec. 2018
Consume: A privacy-preserving authorisation and authentication service for connecting with health and wellbeing APIs

Mart Wetzels a,*, Idowu Ayoola a, Sander Bogers b, Peter Peters a, Wei Chen c,d, Loe Feijs a

a Designed Intelligence Group, Department of Industrial Design, Eindhoven University of Technology, Eindhoven, LaPlace 32, 5612AZ, Netherlands
b Designing Quality in Interaction Group, Department of Industrial Design, Eindhoven University of Technology, Eindhoven, LaPlace 32, 5612AZ, Netherlands
c Center for Intelligent Medical Electronics, Department of Electronic Engineering, School of Information Science and Technology, Fudan University, Shanghai, China
d Shanghai Key Laboratory of Medical Imaging Computing and Computer Assisted Intervention, Shanghai, China

A R T I C L E   I N F O
Article history:
Available online 15 November 2017

MSC:
00-01
99-00

Keywords:
Data acquisition systems
Authentication
Privacy
Healthcare
eHealth

A B S T R A C T
The growth of the Internet of Things (IoT) application within the health- and wellbeing domain enables individuals to monitor their health. Acquired data can be used privately, contribute to clinical databases, or for research. The amount of health and wellbeing tracking devices introduces complexity in data aggregation and scattered overviews. Few services exist to aggregate health data. Current services raise privacy concerns. Consume is a service for aggregating authentication and authorisation for Application Programming Interfaces (APIs). Consume aims at research and allows to add existing and custom APIs on-the-fly without restarting services.

© 2017 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Introduction

Publicly accessible health APIs are increasing as companies are releasing new health trackers. In most cases, a single API will enable data extraction from a range of devices from the same vendor [1]. Bluetooth- and WiFi-enabled devices often provide a dedicated smartphone application for communication of data and a dashboard for the user to create insight on their behaviour. The increase in the type of trackers – activity, sleep, heart rate, respiratory – results in various applications being installed on the smartphone and resulting in a scattered overview of the consumer's data. Health Mashups integrate multiple data sources into a uniform interface and possibly create new insights with data fusion [2]. Related commercial Health Mashups and services are Apple HealthKit, Microsoft Health, GoogleFit, QuasiModo, Shimmer, Fitabase, and HumanAPI. HumanAPI targets companies, medical institutions, and other platforms instead of directly to end-users.

Within the domain of pervasive healthcare, privacy is an important value [3]. The use of these services and individual devices raises the following issues concerning the user's or participant's privacy. Firstly, ownership of data often remains at the vendor, or they reserve the right to use and commercially exploit or share with affiliated partners [4]. Secondly, the
location of data stored from European Union (EU) citizens is becoming more regulated [5]; restricting storage in the US in some cases. Thirdly, commercial data aggregation services also hold the right to use the data as the individual vendors. The latter causes an increased privacy risk for consumers because the service holds data from various input streams; enabling a richer knowledge on the consumer’s behaviour.

**Consume** facilitates the combination of ease of integration and preserving of privacy through pseudonymization to existing research that requires data aggregation through health and wellbeing APIs. Ease of integration is achieved by enabling expansion of supported APIs on a live system by prioritising universal logistics not bound by a single API. **Consume** lowers the threshold for researchers to experiment with a range of data sources and data generating devices. Preservation of privacy is achieved by the pseudonymization of Personally Identifiable Information (PPI) and by separating the concerns of authorisation/authentication and aggregation of data.

**Personalised healthcare systems**

The self-monitoring of weight using a scale is one of the oldest self-tracking devices adopted by the general public. Nowadays, the range of measurable body metrics has increased beyond weight and activity but the data extraction of these new devices are not always accurate, and different trackers show a divergence between measuring the same actions and behaviour [6]. A validation study on the accuracy of the Fitbit One (Fitbit) [7] concludes the device to be reliable for measuring step-count but warns about the accuracy of calculated distance travelled. Geolocation, collected by the Moves(ProtoGeo Oy) [8] application, can be used to complement the accelerometer-based step-detection from a wearable device. The iOS-application Gyroscope(Gyroscope Innovations Inc.) is a recent example of a system that uses sensor data from different devices to provide a richer overview of health- and wellbeing data.

Personal Healthcare Systems often focus on maintaining or increasing the user’s health by providing insights and persuading them to be more active. The **Principle of Tailoring** [9] describes that contextual information, in combination with personal information, can increase the personalisation of tailoring technology for an effective persuasion strategy. For example, a running application that provides motivational support can take into account the weather, in relation to the user’s performance, to determine the type of motivation that is required. Data acquisition systems need to be able connect to other APIs that can provide contextual information to enrich the personal information obtained from trackers. **Consume** enables researchers to access other APIs, following OAuth2 protocol, in addition to popular health and wellbeing APIs.

In addition to fitness and activity trackers, research is done on new applications within tracking of health. **Smart Cup** allows patients with fluid retention to measure their water intake [10] to prevent over-hydration. **SocialHue** utilises the HUE-lights (Philips) to increase social connectivity between elderly and their caregivers [11]. The last example uses the Philips HUE API to change the ambience in a living environment. **Consume** supports researchers to utilise these APIs for new development in personalised healthcare system. New devices and contextual information services, with their APIs, will be developed more rapidly than before. These developments emphasise the need for data validation models [12,13] to ensure the validity of aggregated data.

**System**

**Consume** uses Node.js (Node.js Foundation, 2016), an event-driven I/O server-side Javascript environment based on Google V8-engine, and utilises several popular dependency listed in Table 1; minor dependencies are not listed.

Fig. 1 illustrates the system architecture of **Consume**. In production environment **Consume** redirects to an encrypted connection between client and server when an SSL-certificate is provided. Authentication- and user-related data is stored in MongoDB collections named *users* and *usertokens*. API configurations are stored in the collection named *apis*. All collections have a Schema defined using Mongoose. The current version of **Consume** provides an interface, using Twitter’s Bootstrap, for participants to register and link accounts. Other efforts addressed in this paper will include the removal of user interface, leaving the back-end, in a separate version of **Consume**. This version can be used for integration in existing systems.

**Architecture**

Figs. 2, 3, and 4 depict the Data Flow Diagrams (DFDs) of different data acquisition systems. The DFDs separate external and internal services. External services include the APIs where data is to be fetched, external systems, and possible databases. Internal services include the system for fetching data and local database. Fig. 2 represents a system with several APIs where each API requires dedicated code within the system to authorise with an API. Fig. 3 represents a service that aggregates APIs externally and enables internal services to acquire the data. Fig. 4 represent the **Consume** service where universal code is used to aggregate from different APIs within the internal service.

The benefit of using a data aggregation system, Fig. 3, over a custom data collection system, Fig. 2, is the reduction of complexity in the internal services.

The privacy issues raised by data aggregation services are caused by the aforementioned right to use and ownership of data by external services. These services are not transparent in how they (re-)use data. For the individuals sharing data it is not communicated what happens to their data and what other parties have access.
Consume facilitates the authorisation and authentication of APIs on a designated server of choice. At the same time it increases the privacy protection of research participant’s data. Consume itself does not store any data, only access tokens, but provides end-points to make calls to other APIs. Depending on the need of the platform, data can be fetched on request or aggregated within a dedicated database. Considering the amount of contextual data needed to identify health-related behaviours, and provide personalised healthcare, Consume provides the flexibility to grow with current health technologies by removing the complexity of authentication and authorisation [14].
API integration

Consume uses the Passport dependency to authenticate users and authorise data access for APIs following the OAuth2 framework [15]. Popular APIs are supported by specific Passport-modules. For systems that include few APIs, combining multiple specific Passport-modules can provide sufficient functionality. However, the code-base will increase per added module. Often, similar code is used with different end-points. The minimal amount of settings required contain the following parameters: Credentials are issues by the server that provides the API (Client-ID and Client-Secret). Authentication Schemes like Basic or Bearer [16]. Authorisation URL to authorise the personal information to be used by the application. Token URL for exchanging an authorisation grant for an access token. Callback URL for the authorisation server to respond to with authorisation credentials. Profile URL for the user’s profile to obtain the unique identifier as defined by API-provider.

Consume re-uses common functions and flows by storing API settings in the database and removing them from the code-base. The correct settings are retrieved when a correct universal end-point is called. One of the authorisation end-points is /:.provider/auth; where :provider is a parameter for the name of the API. When the API is included in this Consume instance, the authentication process will continue. Otherwise, an error will be returned stating that this API (with name) is not supported (yet). Besides the code-base reduction, another benefit of pulling API settings from a database is the ability to add APIs without restarting the Consume-service or resubmitting code.

Aggregation of accounts

Two user-authentication use-cases are supported by Consume: individual APIs and aggregated APIs. When the study only requires data from a single API, the participants can authorise for that vendor. If multiple devices from different vendors are used, a Consume account needs to be created to act as an Identity Provider (IdP) to link tokens from several APIs; future efforts will include the feature of using an external IdP like Auth0. The participant will have a unique Consume-identifier that is used to link UserTokens from the same participant. Fig. 5 illustrates the possible relationships.

Consume supports the scenario where a participant already has a UserToken registered for an API but wants to register another. When the participants creates a Consume-account and re-authorises with the former API, the old UserToken will be updated and coupled to the User.
Endpoints

The endpoints for Consume are restricted to authentication and pass-through-requests. Authentication endpoints include: sign-up, login, logout, API-authentication, and API-callbacks. The pass-through-request endpoint, or request, is used to make the call to the APIs using access tokens stored within Consume. The request endpoint is secured by a combination of JSON Webtoken [17] and whitelisted services.

Fig. 6 illustrates making requests to an API through Consume. Without Consume, the client needs to supply access tokens, safely store tokens, and keep tokens up-to-date when expired. With Consume, the client makes the request to Consume, API settings are fetched, access tokens get added, and the response is returned to the client. Consume takes care of maintaining the validity of access tokens that expire. The current feature description of Consume includes the participants to make requests for their own data as well. This feature will be consolidated into a separated version because it is not desirable for all research studies.

Security aspects

Consume stores pseudonymised identifiers, access-, and refresh-tokens. Although the Consume-User identifier does not contain any personal information, the connected UserTokens contain pseudonymised user identification from connected APIs. Some APIs provide public user profiles that can be found using the APIs user identifier. Several security features, or best-practices, are included in the development of Consume:

- Storing least amount of data to operate. The only data-element related to personal information is the user identifier for the external API. In order to obtain that identifier, a call for the user's profile is needed. All data but the identifier are removed before storing to the database.
- In production environment, always redirects to HTTPS using a mandatory SSL-certificate to be provided.
- Rate-limiting is applied for authentication, successive failed authentication attempts will increase, an exponentially growing, waiting time for another attempt.
- Libraries and session information are restricted to the server-side.
- External services, connected to Consume, are whitelisted so no other service can use the access tokens for other APIs. The restricted callback-url contained within the API-settings prevents using the access-tokens in the case of a security breach.
- Other unlawful access or security issues are reported, using Helmet, through a webhook.

Dependency

Consume depends on the effort of the authors, and module names, listed in Table 1.
Table 1
List of dependencies of Consume.

<table>
<thead>
<tr>
<th>Dependency</th>
<th>Github</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Express</td>
<td>expressjs/express</td>
<td>Web framework</td>
</tr>
<tr>
<td>Passport</td>
<td>jaredhanson/passport</td>
<td>Authentication</td>
</tr>
<tr>
<td>Mongoose</td>
<td>Automattic/mongoose</td>
<td>MongoDB object modelling</td>
</tr>
<tr>
<td>Moment</td>
<td>moment/moment</td>
<td>Date library</td>
</tr>
<tr>
<td>Helmet</td>
<td>helmetjs/helmet</td>
<td>Security</td>
</tr>
<tr>
<td>Request</td>
<td>request/request</td>
<td>HTTP request client</td>
</tr>
<tr>
<td>Bootstrap</td>
<td>bower/bootstrap</td>
<td>Front-end framework</td>
</tr>
</tbody>
</table>

Use-cases

To elaborate on different uses of Consume, several (possible) applications are described in relation to the positioning of Consume within a service. A study on human activity recognition (AR) [18] utilises several APIs, Twitter/Instagram/FourSquare, to gather personal activities. When using Consume, it would be possible to more easily implement other activity-based APIs, like Moves, to enhance the activity classifier and concat aggregation elements of the system architecture. Similarly, a life-logging of personal health data system [19] could benefit from Consume by including more health-related APIs to facilitate a larger range of users who might prefer a brand of trackers over others. The system architecture from the ipHealth project [20] used HumanAPI to aggregate data from different sources. With Consume the research of ipHealth would have increased control over the data flow and privacy concerns. Other ongoing and future implementations of Consume by the authors are discussed in the following paragraphs.

Education

Within the Department of Industrial Design, Eindhoven University of Technology in the Netherlands, students focus on developing intelligent products, services, and systems within a societal context that often require data from existing devices. Within this use-case an implementation of Consume including a front-end is provided for students to easily access their own data and from a small amount of participants to experiment with and use for the demonstrator of their final prototype. For these students Consume takes care of the authentication and authorisation, enabling them to focus on the actual design work in an earlier part of the project. Performance of Consume in an educational setting will be measured by observing the coverage of projects using Consume versus other implementation approaches.

Personalised health and behaviour change mobile application

An upcoming study follows the Experiential Design Landscape (EDL) [21] methodology to design and develop data visualisations and behaviour change prompts to trigger a healthier lifestyle. Participants are given several consumer-grade health trackers and a mobile application. The mobile application changed over time based on the participant’s feedback from within the application. A limited amount of participants are asked to use additional trackers, resulting from other studies, from which the results are integrated in the application. Consume will enable to gather data from the consumer-grade health tracking APIs and private APIs from the additional trackers of other studies. Performance of Consume will be evaluated by the ease of expanding data sources, amount of errors in requesting data, and overall stability.

Do CHANGE ecosystem

Within the EU Horizon2020 Do Cardiac Health: Advanced New Generation Ecosystem (Do CHANGE) project, an ecosystem for behaviour change of cardiovascular patients is being developed. The previously mentioned mobile application and tools will be used in the clinical trials in Spain, Taiwan, and The Netherlands. For these trials, Consume, and an accompanying data aggregator, are used to store sensitive data from various sources into a dedicated database on private servers within the consortium; no external service stores aggregated data. In addition to the evaluation criteria mentioned in the previous paragraph, the use of Consume in the clinical trial will conclude the overall performance for a longer duration and higher load due to the increase of participants.

Conclusion

The growth of connected monitoring devices enables consumers to obtain a richer overview of their health and provides new opportunities for research. In order to obtain health- and wellbeing data from various sources, privacy and data aggregation services are becoming more important. Before the IoT-era, research within personalised health was limited to questionnaires, manual recording of vital signs, or restricted to post-study analysis of sensor data. Technological developments enable researchers to perform near real-time analysis and obtain more data over longer periods of time with limited burden for the participants.
Research studies that utilise these devices need to facilitate data aggregation systems. These systems require additional effort for the preparation of the study. Studies with a limited, low amount, of devices might prefer a custom system as presented in Fig. 2. The benefits of using an external data aggregation system like HumanAPI, presented in Fig. 3, is the reduction of effort for the research team. The down-side of using such a system are the privacy issues resulting from an external party owning a copy of collected data. In addition, custom integration of APIs is not possible.

Consume, see Fig. 4, provides the ground-work for data aggregation systems by removing the complexity of setting up authentication and authorisation for multiple APIs. By excluding an external data aggregation service the research data is consolidated within a dedicated environment which reduces privacy risks. The exact privacy risks are influenced by governmental policies and regulations. The approach within Consume to write universal code for authorising with other APIs increases the flexibility for supporting new vendors or experimenting with new devices. As with any other system, the development of Consume will require effort to stay up-to-date for security reasons but can also grow in features that are desired within the research community.

Acknowledgement

This work is supported by the European Commission Horizon2020 funded Do Cardiac Health: Advanced New Generation Ecosystem (Do CHANGE) project (Grant number 643735).

References


