

What Do Developers Discuss about Biometric APIs?

Zhe Jin[†], Kong Yik Chee[†], Xin Xia[‡]

[†]School of Information Technology, Monash University Malaysia

[‡]Faculty of Information Technology, Monash University Australia

{jin.zhe, xin.xia}@monash.edu

Abstract—With the emergence of biometric technology in various applications, such as access control (e.g. mobile lock/unlock), financial transaction (e.g. Alibaba smile-to-pay) and time attendance, the development of biometric system attracts increasingly interest to the developers. Despite a sound biometric system gains the security assurance and great usability, it is a rather challenging task to develop an effective biometric system. For instance, many public available biometric APIs do not provide sufficient instructions / precise documentations on the usage of biometric APIs. Many developers are struggling in implementing these APIs in various tasks. Moreover, quick update on biometric-based algorithms (e.g. feature extraction and matching) may propagate to APIs, which leads to potential confusion to the system developers. Hence, we conduct an empirical study to the problems that the developers currently encountered while implementing the biometric APIs as well as the issues that need to be addressed when developing biometric systems using these APIs. We manually analyzed a total of 500 biometric API-related posts from various online media such as Stack Overflow and Neurotechnology. We reveal that 1) most of the problems encountered are related to the lack of precise documentation on the biometric APIs; 2) the incompatibility of biometric APIs cross multiple implementation environments.

Index Terms—Biometrics, APIs misuse, Empirical software engineering

I. INTRODUCTION

Biometrics as a mean of identity authentication refers to verify or identify an individual using human physiological or behavioral traits, such as *face*, *iris* and *fingerprnt* [1]. The advancement of biometric technologies facilitates the identity management systems in an accurate and efficient manner. Such evolution of biometric technologies drives the increasing demand for biometric-based identity management software. However, biometric recognition is a complicated task indeed. For instance, different modalities of biometrics possess different characteristics. Specifically, majority of fingerprint matching is actually to match minutia points set extracted from the fingerprint image [2] while iris matching is performed in binary domain [3]. Unfortunately, most of the developers are not experts in the field of biometrics and therefore the software developers have to heavily rely on the forums or questions and answers (QA) sites such as Stack Overflow, Neurotechnology forum and SourceForge for more detailed explanation and discussion. Despite there being many biometric Application Programming Interfaces (APIs) available that provide solutions to cater for individual/organization tasks, it is still challenging

for biometric software developers to select the right APIs for their applications as well as in implementing these APIs.

We know the fact that developers often query for solutions or search for explanations on unfamiliar terminologies. Xia et al. [4] conducted an empirical study on what developers search for on the web. Such study illustrates an comprehensive understanding on developer's behaviors and the problems that they are facing throughout the project development process. Inspired by [4], we perform a domain-specific search (i.e. Biometric APIs) in this paper. To be specific, we investigate publicly available data from different developer forums and QA sites and determine the most common issues faced by biometric software developers. We expect that, from our study, the developers can obtain a more clear understanding of the difficulties and gain useful recommendations. Because as stated in [5], it is rather time consuming for developers to resolve API questions when they are unsure which API to use or troubleshoot the error encountered when using the API. Using our study as a guideline, the developers may have a lead on how to resolve their difficulties. On the other hand, this study also acts as a guide for future developers in understanding requirements, challenges in developing biometric-based software. In addition, this study can serve as user's feedback to the biometric APIs providers in understanding the users difficulties and therefore improve the existing products such as providing clearer documentation or new APIs [6]. In this study, we formalize the data obtained into two fields of research:

- What type of questions developers often ask?
- What challenges the developers are facing over time?

Conventionally, developers often utilize online forums or QA sites to share information, ask for support, search for solutions and learn new knowledge. It is not limited to specific or common programming or application problems [7]. Our study specifically examines the posts that are closely related to biometric APIs and hence our data are collected not only from Stack Overflow (SO) but mainly from biometric APIs providers discussion forum, such as Neurotechnology forum and National Instrument Labview forum. That is unique to other studies that focus on SO as the main source of data. It is worthy to note that development of biometric-based applications is relatively new compared to those classic topics such as Java development using cryptography APIs

[8]. Therefore, our study serves as an early-stage of work to provoke thoughts and discussions in this topic. Nevertheless, the results of this study advise that biometric APIs users suffer from obtaining sufficient and appropriate guidance from the API documentation. Moreover, it remains a challenging task for the biometric APIs users to configure/import the suitable libraries or SDK to implement an efficient biometric system.

The main contributions of this paper are as follows:

- 1) To the best of our knowledge, we are the first to perform an empirical study of biometric API-related posts across different online media.
- 2) We report several interesting and valuable conclusions. We investigate the categories of various biometric API-related posts which provide some implications for future research.

The remainder of this paper is organized as follows. We present our empirical study setup in Section II. We present our empirical study results in Section III. We discuss additional points on the benefits and limitations of our study in Section IV. We discuss related work in Section V. We conclude and mention future work in Section VI.

II. EMPIRICAL STUDY SETUP

A. Data Collection

The objectives of our study are to identify what are the issues encountered by developers and make a recommendation or suggest plausible solutions. To collect the data, we visited several Q&A sites, namely SO, Neurotechnology Forum and other relevant discussion forums. To search for the relevant posts, we conducted the searches based on biometric API tags. We were able to gather a total of 500 posts, in which 300 posts from Neurotechnology Forum, 187 from SO, 7 from MARF, 3 from National Instrument Labview, and 1 each from Cisco Community, SourceForge, and Recognito. Due to the fact that biometric is still a relatively new research topic, there is limited information that can be found on these Q&A sites.

B. Card Sorting

We manually analyze all the 500 posts, understanding their problems and solutions. We use open coding and since our study is of exploratory nature, we do not use pre-defined categories. We created a category for each new post encountered. Each category is labelled in such a way that it is easily understood and is self-explanatory. The posts are analyzed according to the questions asked and the response given by other users. From the response collected, we then grouped and assigned the post with a similar root cause to the respective category. The detailed steps are as below:

- **Sorting.** We manually inspect each of the selected post and sort the post into distinct sets according to their root cause. Those posts with unclear root cause are first omitted from the sorting process. Next, we discuss each set and name them based on the characteristics of each set.

TABLE I
INTERPRETATION OF KAPPA VALUES

Kappa Value	Interpretation
<0	poor agreement
[0.01, 0.20]	slight agreement
[0.21, 0.40]	fair agreement
[0.41, 0.60]	moderate agreement
[0.61, 0.80]	substantial agreement
[0.81, 1.00]	almost perfect agreement

TABLE II
CLASSIFICATION OF THE PROBLEM CATEGORIES FOR THE 500 POSTS COLLECTED

Problem Category	Number of Count
Lack of information on documentation	112
Configuration	74
Misuse of API	49
Module selection	48
Template conversion	43
Constant update of SDK	38
Device/Framework not supported	38
Communication with devices	26
Modification of source code	20
Lack of knowledge in biometrics	24
API bugs	18
Passing parameter	6
Using component in other platforms	4
Grand Total	500

- **Labeling.** The first two authors independently label the posts. The agreement between the two labelers are measured using the Fleiss Kappa [9] and the interpretation of the Kappa value is as shown in Table I. The overall Kappa value between the two labelers is 0.62 which suggest that there is a substantial agreement between the two labelers. After the labeling process, the labelers discuss their disagreement and reach a common decision. For those posts that do not provide sufficient information or with the unclear root cause, the labelers have made a consensus to not include them in the classification as this may affect the credibility of the study.

III. EMPIRICAL STUDY RESULTS

A. Categories of Biometric API-related Posts

We categorize the difficulties into 13 problem categories as shown in Table II.

Table II shows that more than 22.4% of the developers had difficulties in obtaining the information from the documentations of the related biometric API. Most of the questions involve the functions of a particular API or asking for an updated version of documentation. For instance, one of the users questioned the error message returned, but the documentation of the API did not provide a detailed explanation on the error message.

The next popular type of questions that developers asked is regarding the configuration of the SDK (14.8%). Many developers encounter errors when setting up the SDK or misplacing certain files into the wrong folder during configuration. For example, one of the questions posted was regarding the configuration of a method call in application and changes

to be made in the config file in order to communicate with the database server. About 9.8% of the developers had misused the APIs due to misunderstanding the documentation or passing the wrong parameters to the API. For example, a user was using the wrong approach to generate one multi-face template consisting of different people where the SDK used was intended for face template generation of the same individual. Other than that, we also notice that a similar number of posts (9.6%) that developers asked are regarding the module selection, in which most of these developers are new to biometrics and want to develop an API but not sure where to start. Given that there are many publicly available APIs for different biometric modalities, developers will have difficulties in choosing the right or suitable APIs to use. In Neurotech forum, many of the users questioned on the suitable fingerprint SDK to be used with the appropriate biometric scanner.

Moving down the list, approximately 8.6% of the questions are regarding the template conversion. As some biometric APIs only allow certain format of template, hence it would be challenging for the biometric developers to convert the biometric template from one to the other. For instance, some API only accept The Wavelet Scalar Quantization (WSQ) and developers are required to convert the templates of the images they generated into WSQ image else they are not able to proceed.

Out of the 500 posts, 38 posts are regarding the constant update of the SDK and another 38 posts are about the compatibility of the APIs in other devices or framework. Due to the fact that biometric technologies are evolving rapidly, some of the biometric APIs have already outdated, although the API provider has provided an updated version of the APIs, many developers are not aware of the updated version. While some of the functions from the older version had depreciated, developers will face issues when using these old APIs. For instance, users had faced issues such as their existing API is not functioning due to the updates that they are not aware of.

Table III lists the posts collected from different platforms. Most of the posts collected from Neurotechnology Forum are regarding the configuration and the documentation of the biometric APIs, while in SO we observed similar questions on the API documentation being asked while not many questions on the configuration of biometric APIs. As Neurotechnology Forum are mainly for those who use Neurotechnology products, we have observed that the questions asked are often answered faster and with more accuracy as compared to SO where SO is more open to any biometric questions and hence it would take a longer reply time.

B. Trends of Biometric API-related Posts

Fig. 1 shows the breakdown of the problem categories by year. In the earlier years, the problems most revolve around improper documentation and configuration issues as well as constant updates of the biometric APIs. As time goes on, there are fewer questions regarding configurations and updates as the APIs deployment are more stable. However, issues around getting the correct information from the API documentation

TABLE III
CLASSIFICATION OF THE PROBLEM CATEGORIES BASED ON PLATFORMS

Source	Problem Category	Count of No
Neurotechnology	Configuration	60
	Lack of information on documentation	59
	Misuse of API	35
	Constant update of SDK	34
	Framework/Device not supported	26
	Template conversion	24
	Module selection	18
	API bug	15
	Communication with devices	12
StackOverflow	Modification of source code	11
	Lack of information on documentation	49
	Module selection	24
	Lack of knowledge in biometrics	24
	Template conversion	18
	Communication with devices	14
	Configuration	14
	Misuse of API	13
	Framework/Device not supported	12
	Modification of source code	9
	Constant update of SDK	4
	API bug	3
	Passing parameter	2
	Using component in other platforms	1
MARF	Module selection	4
	Lack of information on documentation	2
	Misuse of API	1
NI Labview	Lack of information on documentation	1
	Module selection	1
	Passing parameter	1
Recognito	Template conversion	1
SourceForge	Lack of information on documentation	1
Cisco Community	Module selection	1

persist. With more and more biometric APIs being developed, more questions centered around the compatibility of the biometric devices as well as cross-platform usage of biometric API.

From Table IV, there is also an increasing trend in the problems regarding the communication with biometric devices, in particularly the sensor to obtain the raw biometric information from the user. This is mostly caused by the licensing issue or misconfiguration of the connection with the local machine and the biometric devices. The increasing trend suggests that the drivers for biometric devices are not up to date while in some cases the local machine is not able to detect the supported biometric devices due to the missing files.

Furthermore, the increase in the popularity of biometric usage has led to many new users to explore more about biometrics. However, these new users might not have a solid foundation of biometric itself, hence we can observe a slight

Problem Category	2005	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	Trend
Module selection	1	0	2	2	5	8	4	4	4	5	6	5	2	
Lack of information on documentation	0	1	4	19	9	8	14	6	6	8	5	10	22	
Communication with devices	0	0	1	0	0	1	0	3	3	1	1	7	9	
Configuration	0	0	2	18	12	16	4	2	3	3	5	4	5	
Passing parameter	0	0	1	1	1	0	0	0	0	0	2	1	0	
Framework/Device not supported	0	0	5	1	9	5	2	0	0	4	2	5	5	
Misuse of API	0	0	3	4	3	7	5	0	4	3	4	12	4	
Constant update of SDK	0	0	0	2	5	14	9	0	1	1	1	3	2	
Modification of source code	0	0	0	1	4	4	1	1	1	1	0	2	5	
Template conversion	0	0	0	3	6	2	4	5	5	4	4	5	5	
Lack of knowledge in biometric	0	0	0	1	3	0	3	3	1	4	1	4	4	
API bug	0	0	0	7	1	4	3	0	0	0	0	1	2	
Using component in other platforms	0	0	0	0	0	0	0	0	0	0	1	1	2	

Fig. 1. Analysis of Problem Category Over Time

increase in trend on the issues regarding the template conversion and questions on the basic concepts of biometric authentication system.

IV. DISCUSSION

A. Implications

From this study, there are two directions for moving forward. Firstly, we would suggest API developers to not provide additional unnecessary information/detail on the background of the biometric but to provide more task-based/functionality details to API users. From some of the solutions we gathered from different posts, many of the users do not necessarily have the biometric background and their main concern will be to get the desired output from the respective function/ method call. Secondly, from the solutions we gathered, we recommend a guideline for developers to follow when developing biometric API:

- Provide a brief explanation on the function, with a detailed explanation on the input parameters and expected output.
- Provide sample codes or examples so that users have a brief idea on the API usage.
- Provide limitations of the biometric API and any alternatives/references for users if there are any shortcomings of the API.
- Provide the API design as well as the algorithm of the API in such a way that the user can have a better understanding of how the API works.

B. Threats to Validity

Internal Validity. As we rely on the biometric API tags when collecting data, there might be some posts that we missed out that does not have biometric API tags. In addition, there are some posts that are wrongly tagged as biometric API but does not have anything to do with biometric API. In these cases, we do not account them into our data as this will affect the overall accuracy of the analysis. Furthermore, to mitigate the biasness when categorising the posts, we have two different individuals to perform the categorization independently and result will only be recorded when consensus is made.

External Validity. To avoid biasness in selecting data from only one source, our data collected are from seven different source ranging from commercial forum (i.e. Neurotechnology) to open forum (i.e. SO) and regardless of programming languages. Hence, we are confident that the data collected and our findings are applicable to the general biometric API questions.

V. RELATED WORKS

To develop the software efficiently, developers often use APIs of frameworks and libraries to access to the built-in functionality [10]. However, many APIs suffer from incomplete/incorrect documentation, outdated source codes, incompatibility with external libraries or operating system. In order to resolve the above issues, developers often visit Q&A sites such as SO and Quora where developers can ask and share various issues they have encountered. Hence, these Q&A sites will provide us an insight into what are the trending technologies and what are the most popular questions the developers frequently ask. Through the data collected from these sites, API providers or developers can obtain a comprehensive understanding of the users requirements and develop more sophisticated or more user-friendly APIs.

Rosen and Shihab [11] focused on using Latent Dirichlet Allocation (LDA) method to study mobile related questions asked in SO. They reveal that mobile developers concern on app distribution, input and user interface development. On top of that, they also show that mobile developers are keen to obtain the working examples from respective SDK documentations. Sadowski et al. [12] conducted studies on how developers ask questions in Google. Their studies provide insights into what developers are trying to understand when performing a search, the search context, query properties and the query scope. Their studies also reveal that developers often seek answers about API usage, error in code and code location.

Nadi et al. [8] investigated the obstacles faced by Java developers when using cryptography API. Through examining questions from GitHub, SO and two surveys from 48 respondents, the study had shown that Java developers often require these APIs to perform authentication, establish secure network connections and safeguard different forms of data stored. The

TABLE IV
COMPARISON OF BIOMETRIC APIS VS OTHERS

Related Works	Topic of discussion	Major challenges faced/ Main discussion	Implications of study
Rosen and Shihab	Mobile Applications	App distribution, user interface and input	Help research community understand the challenges faced and further improve the current documentation practice to include more working examples.
Nadi et al.	Java Cryptographic APIS	Determining the correct sequence of method call and lack of domain knowledge	Assist API providers to improve their current documentation, encourage API designers to hide unnecessary details and provide more task-based method calls.
Ventakesh et al.	Web APIS	Known Issues/ Bugs	Assist web APIS providers to address dominant issues efficiently and suggest programming community platform to highlight and create custom tags to enable faster responses to programmers.
Jin et al.	Biometric APIS	Insufficient information from API documentation	Suggest API providers to provide more task-based/functionally details to API users and recommendation of guideline to follow when documenting the APIS.

study also indicated that the existing cryptographic APIS are too low level and developers often asked for more task-based solutions.

Ventakesh et al. [13] studied the concerns of client developers when using Web APIS. They investigated the most discussed topics related to Web APIS and the evolution of this topics over time. Their studies had shown that "Known issue/bug" is dominantly discussed among developers and majority of the discussions are topics or concerns that disappears quickly which indicates that Web APIS providers often review and address to the issue faced by client developers.

Table IV shows the comparison to identify the commonalities and uniqueness of biometric APIS vs. others.

VI. CONCLUSION

From the analysis of the data collected, we have observed two main issues need to be addressed when developing biometric API:

- Precise documentation. Although most of the biometric API provides documentation, some of the documentation does not provide sufficient information and is not clear on the purposes and concepts of the APIS rather than merely stating what parameters to pass and what outputs are expected [14]. Hence precise documentation on the biometric APIS is vital in guiding the users the correct usage of the APIS.
- Universality and Efficiency. As there are more and more biometric APIS being developed, it is important to take into considerations whether the APIS are compatible with other platforms or framework.

Biometric documentation often acts as the first source of information for developers when using biometric APIS. From our empirical study, it has shown that the obstacles faced by developers often caused by poor documentation. Our findings also show that many developers have trouble assessing the API, for instance, wrong environment configuration, and failure to communicate with devices. With more and more biometric APIS being developed, more issues concerning the compatibility and universality of the APIS will rise. Our study will help to serve as a guide for future biometric development

and to help lessen the problems arise when using biometric APIS.

ACKNOWLEDGEMENT

This research was supported by Fundamental Research Grant Scheme (FRGS/1/2018/ICT02/MUSM/03/3) and NVIDIA Corporation donation of the Titan Xp GPU.

REFERENCES

- [1] A.K. Jain, P. Flynn, and A. A. Ross, "Handbook of biometrics," 1st ed., Springer US, Oct 2007.
- [2] D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, "Handbook of fingerprint recognition," 2nd ed., Springer-Verlag London, Apr. 2009.
- [3] M. J. Burge, K. W. Bowyer, "Handbook of iris recognition," 1st ed., Springer-Verlag London, Apr. 2013.
- [4] X. Xia, L. Bao, D. Lo, P.S. Kochhar, A.E. Hassan, and Z. Xing, "What do developers search for on the web?," *Empirical Software Engineering*, 22(6), pp. 3149–3185, 2017.
- [5] J. Zhang, H. Jiang, Z. Ren and X. Chen, "Recommending apis for api related questions in stack overflow," *IEEE Access*, 6, pp. 6205-6219, 2018.
- [6] C. Rosen, E. Shihab, "What are mobile developers asking about? A large scale study using stack overflow," *Empirical Software Engineering*, 21(3), pp. 1192-1223, 2016.
- [7] C. Treude, O. Barzilay, M. A. Storey, "How do programmers ask and answer questions on the web?," *Proceedings of the 33rd International Conference on Software Engineering*, ser. ICSE11. pp. 804807, 2011.
- [8] S. Nadi, S. Kruger, M. Mezini, and E. Bodden, "Jumping through hoops: why do Java developers struggle with cryptography apis?," 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE), pp. 935-946, 2016.
- [9] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.
- [10] M. Ahasanzaman, M. Asaduzzaman, C.K. Roy, and K. A. Schneider, "Classifying stack overflow posts on API issues," 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp.244-254, 2018.
- [11] C. Rosen, and E. Shihab, "What are mobile developers asking about? A large scale study using stack overflow," *Empirical Software Engineering*, 21(3), pp. 1192-1223, 2016.
- [12] C. Sadowski, K.T. Stolee, and S. Elbaum, "How developers search for code: a case study," *Proceedings of the 10th joint meeting on foundations of software engineering (FSE)*, ACM, pp. 191201, 2015.
- [13] P. K. Venkatesh, S. Wang, F. Zhang, Y. Zou, A. E. Hassan, "What do client developers concern when using web apis? an empirical study on developer forums and stack overflow," 2016 IEEE International Conference on Web Services (ICWS), pp. 131-138, 2016.
- [14] H. Jiang, J.X. Zhang, X.C. Li, Z.L. Ren, and D. Lo, "A more accurate model for finding tutorial segments explaining APIS." *Proc. the 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, March, pp.157-167, 2016.