

# The Universal Biometric System

H. M. N. Dilum Bandara, S. M. Ravindra P De Silva, P. W. H. Dasun Weerasinghe  
Department of Computer Science and Engineering  
University of Moratuwa  
Sri Lanka  
077-7154210, 033-2288308  
dilumb@cse.mrt.ac.lk

## ***Abstract***

*The Universal Biometric System is a biometric enabled third-party authentication system. It tries to address most of the major issues faced by current biometric industry. This paper describes a system that hides all the complexities of biometrics and provides biometric technology, vendor and platform independent authentication. It also introduces two novel ideas: many-to-many mapping reduces the Total Cost of Ownership (TCO) while Device-Hierarchy enforces in-depth security.*

*It achieves biometric vendor, technology and platform independence through the BioAPI specification (the Defacto standard for Biometrics). However it sets its sights far beyond the BioAPI. The system is able to overcome the issues relating to integration of biometrics in an enterprise level network which is one of the biggest problems faced by the biometric industry. It is also designed to provide a simple development environment that does not require complex data structures, pointers and memory management inherent to the BioAPI. This is a Proof of Concept effort.*

**Keywords - Biometric, BioAPI, Biometric Service Provider, Device Hierarchy, Networking, Security.**

## **1.0 Introduction**

Biometrics is an open-ended set of technologies based on the measurement of some unique physical characteristics of human beings (or even animals) for the purpose of identifying an individual or verifying identity. Simply saying “*your body is your password*”. Biometrics is today’s prime technology when it comes to access

control, especially in medium to large-scale organisations. At present technology is matured enough and has proven it is the current best when tight security is the main concern. It is convenient to use, publicly accepted (up to a certain level) and more importantly affordable. Users have all forms of biometrics technologies (Fingerprint, Iris, Retina, Facial, etc.) to choose from, based on the required level of security and available budget constraints.

With such a value proposition it is disheartening to see, the low level of deployment of biometrics. The reason is the difficulty of integration with in a networked environment at a low cost. Thereby this paper describes a concept and its implementation to integrate biometric technology at low cost with less effort while enforcing in-depth security [1].

## **2.0 Biometric integration**

The complexity of integration is the major factor withholding the market for biometrics and its large-scale deployment. Most of the time it is too hard, too costly and some times impractical as well. It does not easily fit into today’s complex enterprise level networks. There are very few solutions that meet up this challenge, even those solutions are either limited to a specific biometric technology, vendor or platform.

If any organization moves into biometrics for access control, according to the current standard practice it is required to install same type of device from the same vendor all over the organization. This raises three concerns: It is required to have a dedicated device for each and every doorstep and on each host (PC/Server). Secondly, organizations have to stick with the same type of biometric devices regardless of required level of security. Third, organizations are in a dilemma when they scale up (i.e. integration problems, tight dependence on a particular vendor, inability to go forward with latest technology developments due to backward compatibility issues, etc.).

From the application developer point of view they need to master a specific SDK (Software Development Kit) provided by its device vendor. Typically this requires thorough knowledge of C/C++ or even Assembly, which is very inconvenient for an average level developer.

### 3.0 Previous work

According to a web based survey it seems that the Universal Biometric System (referred as **Universal BioSys**) is unmatched. There were several security solutions that are related and worth mentioning here.

#### The Independent Security Server – by Info Data, Inc.

Provides means to identify a person based on any biometric characteristics. They have developed BSP (Biometric Service Provider) libraries for all the popular products therefore it is compatible with all major biometric scanners. Hence users have to rely on Info Data, Inc. [3] to provide compatibility with what ever the biometric device they buy.

#### The WhoIsIt biometric server for e-Commerce

This is basically an application server hosted in the Internet where a client system sends a biometric template to be verified. On success any secret (payload) that is stored for that particular user can be retrieved. WhoIsIt biometric server needs to be aware of the underlying technologies and the users are restricted to the vendors in commercial agreement with them [4].

Even in these systems it is obvious that the problems of vendor, platform and technology dependence are still present up to a certain level.

The BioAPI specification [2,5] could be considered the Defacto industry standard framework for biometrics. It defines how application developers and device vendors communicate with each other through a standard framework. The BioAPI is defined to overcome the problems of technology, platform and vendor dependencies. There is a freely available reference implementation of the BioAPI as well.

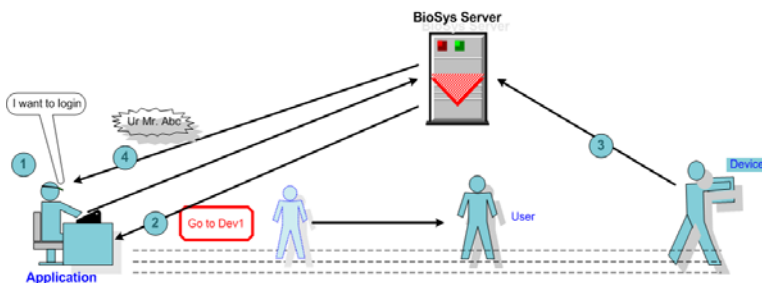


Fig 1 : Authentication steps involved with Universal BioSys.

BioAPI is two fold. The application developer needs to comply with Application Program Interface (API).

Device manufacturer needs to comply with the Service Provider Interface (SPI). That is how platform, technology and vendor independence is achieved. However the issues are not as simple as they sound. These developers need to have some basic idea of biometric technology and they should master C/C++. The BioAPI consists of complex data structures, pointers in average with three levels of indirections and memory management. These required skills are far from the skills of an average developer.

### 4.0 Related work

The Universal Biometric System is a Proof of Concept that is intended to overcome the above-mentioned issues relating to biometric technology, vendor and platform independence.

The BioAPI is the core of the Universal BioSys, however it sets its sites far beyond what BioAPI is intended to do so. It introduces a simple development platform hiding the complexities of the BioAPI. It also introduces two novel concepts.

#### 4.1 many-to-many mapping

The Universal BioSys offers seamless *many-to-many* ( $m$ -to- $n$ ) mapping between biometric devices and hosts. This is one of its novel concepts. Current standard practice is to install a dedicated device for each and every host. So if an organization has 50 machines it has to have 50 devices. This is one of the major reasons that bar the heavy deployment of biometrics.

The Universal BioSys will map  $m$  number of devices into  $n$  number of hosts where  $m$  is much less than  $n$  ( $m \ll n$ ) or  $m$  could even be 1 ( $m=1$ ). However in real practice it has to install several devices due to physical boundaries such as rooms, floors or buildings and mainly for better user convenience and fault tolerance. In simple terms the BioSys can share biometric devices among multiple hosts.

Consider a high quality software development company, with a lucrative business marketing a proprietary product. Therefore it is essential no one else other than the development team has access to its source code.

If the development room has 50 PCs it needs 50 devices plus few more at door steps. With the many-to-many concept organization may need only 5 to 10 biometric devices. It allows group several hosts together and assigning them to one or two devices. It could even share all the installed devices among all the hosts.

#### 4.2 How it is seen by the user

Many-to-many mapping approach share devices among each other through a network. Fig 1 illustrates steps involved in while a user gets authenticated:

*Step 1* : User informs the application that he/she needs access and that request will be send to the server.

*Step 2* : The BioSys Server informs the user a device where user can submit his/her biometric credentials (this decision is given based on user's current location, nearest device and its availability).

*Step 3* : User submits his/her credentials and it is sent to the server where it get processed.

*Step 4* : Server carry out identify or verify functions and make sure that user is either authenticated according to the predefined policies or rejected.

The reader may ask; what if some one else use that machine while the user is still coming back after submitting credentials. This is similar to a situation where a user moving away from a computer while already logged in. There is no real solution to this problem (even without the BioSys) and this is a compromise between the how much owners are willing to pay and the level of security they get. Possible solution would be not to place devices far away from the host or use of double authentication (i.e. first with biometric than possibly by means of a password).

### 4.3 Device-Hierarchy

Organizations may install multiple devices (either same or different biometric technology) on different locations based on the required level of security while having a good balance in TCO. These devices automatically create different security levels. These security levels can

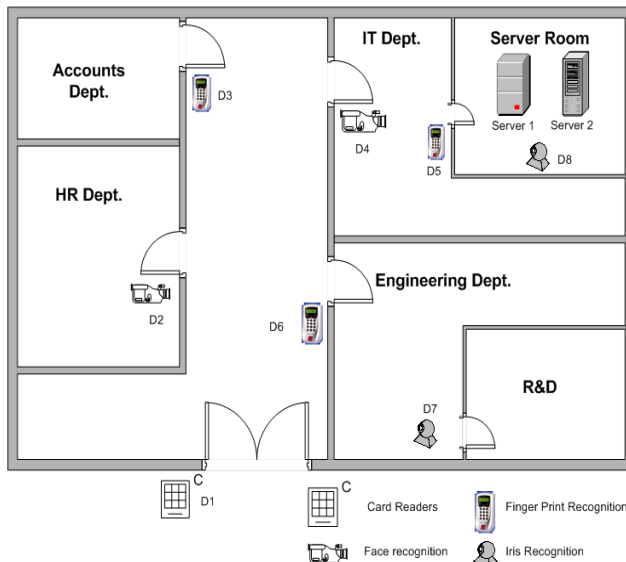


Fig 2: Device arrangement of an organisation

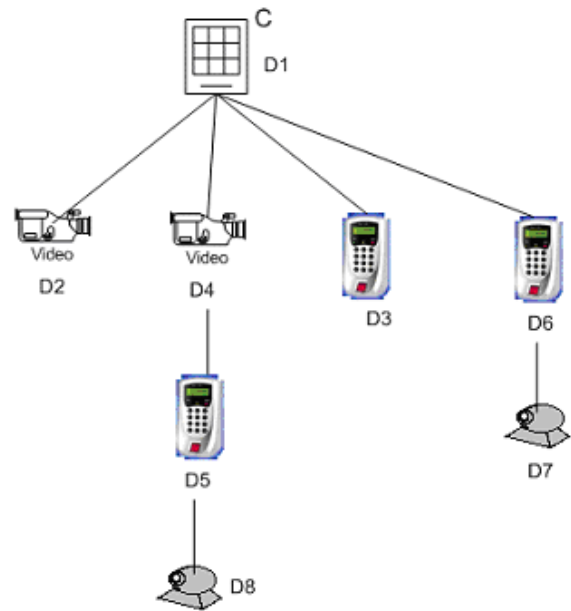


Fig 3: Device hierarchy for the organization given in Fig 2

be represented in a hierarchy.

Fig 2 illustrates a hypothetical organization that has installed different devices based on their security requirement. They have installed card reader at the entrance and placed more secure fingerprint and iris systems at places like Server room, Engineering and R&D. Such an arrangement will result in different security levels which could be mapped into a logical hierarchy such as Fig 3. Knowledge of this hierarchy (referred as Device-Hierarchy by the authors) could be used to gain in-depth security.

In reality, Device-Hierarchy is automatically created when multiple devices are installed with different access levels. It is already there but the problem is no one sees it; therefore no one makes use of this hierarchy to gain tighter security.

According to the defence in-depth approach an organization should have security from its doorstep to the server room. Today all these security precautions are there with different access levels. However the problem is these security measure are independent so bypassing one layer is possible. Device-Hierarchy tries to integrate all these levels together in order to enforce tighter security.

When multiple levels of security measures exist a user has to get authenticated through several devices in a specific order. Consider an example where the system administrator is going to the server room starting from the main entrance. First he/she has to get authenticated using the card reader at the entrance. Then he/she is required to use the facial recognition system at the IT department. Then if the administrator needs to go into the server room he/she has to get authenticated through the fingerprint device as

well. The path followed by the system administrator can be represented by a specific branch in a tree which represents the Device Hierarchy (Fig 3). The branch includes device D1, D4 and D5.

From the system administrator's point of view he/she does not need to remember any of these devices or specific paths. These things happen naturally when people move around within an organization.

This sort of path tracking and path enforcement will make sure that users are not allowed to access any resources unless he/she has entered what ever the place according to the accepted route (branch in the tree). Consider a case where an authorized person is being able to get into the server room through the roof of the organization (or by any other means) and trying to access one of the servers in the server room. If the unauthorized person is able to provide a valid username password combination or is able to forge the biometric device attached to the server there is nothing to stop him/her from accessing the server. However according to the concept of Device-Hierarchy the unauthorized person has violated the hierarchy (i.e. not gone through the accepted path). He/she has directly used device D8 without getting authenticated through devices D1, D4 and D5. In this case the Universal BioSys will not allow any access to the server, although the submitted credentials for the device D8 is correct, since the Device

Hierarchy is being violated.

Enforcement of such a policy would result in-depth security from the doorstep to servers.

Tracking employees (specially the IT support staff) in a large organization could be real problem. Being aware of the Device-Hierarchy will enable the possibility of tracking users. Based on the last authenticated device, a probable location within the organization can be identified. All this is possible since the administrator can configure the Universal BioSys with a map of the organization's floor arrangement (something similar to Fig 2). Several maps could be used if the organization spans several building, multiple floors or if it is too dense to put everything in a single map.

Whatever technologies come and go passwords will remain so many years to come, although it is easily forgettable by the users or guessable by others. Therefore the BioSys also supports password-based authentication.

Although having all those features the Universal BioSys will not be complete if standard practices of network management, administration are not combined with security. Its design highly encourages such security precautions.

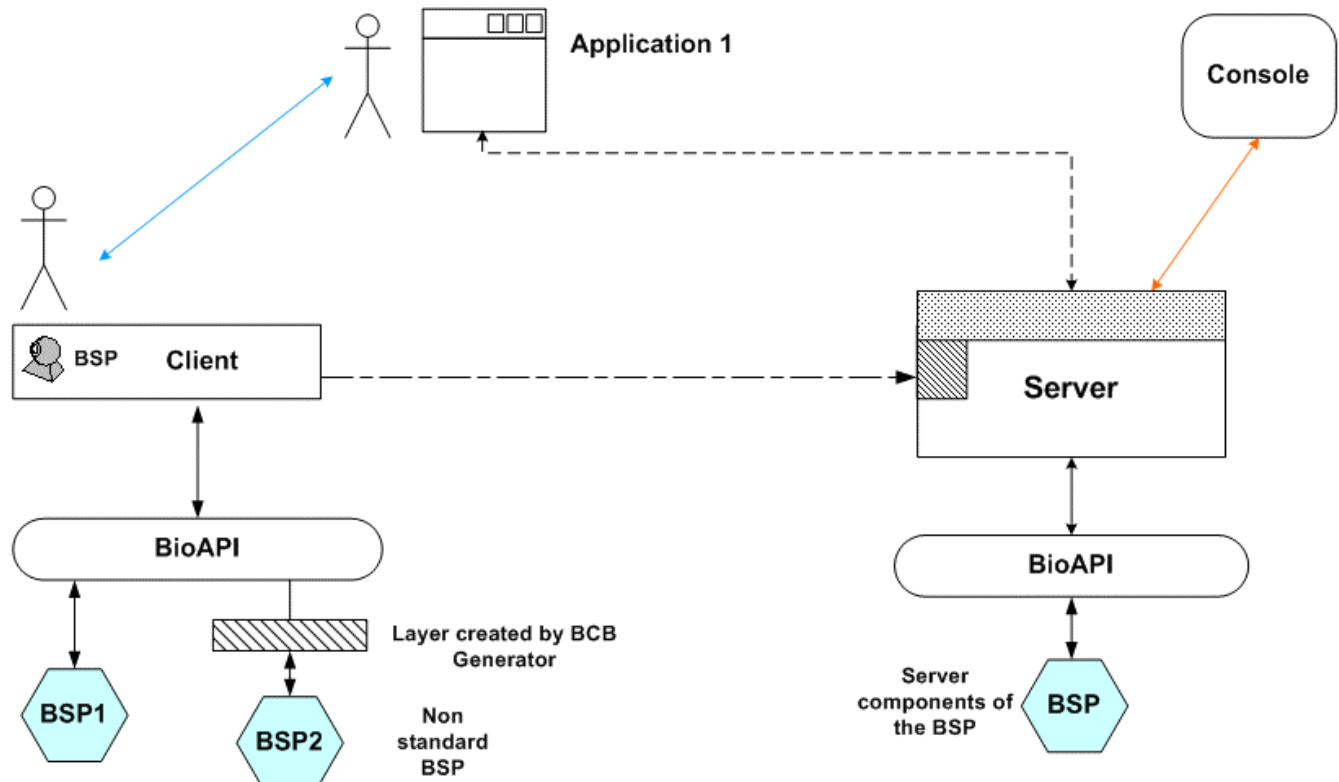


Fig 4: Components of BioSys and their intercommunication

## 5.0 Design and implementation

The Universal BioSys consist of several components that are interconnected to each other (Fig 4). The BioSys Server is the central point of communication and it performs all the administrative, management, policy enforcement and image processing tasks [6].

The Universal BioSys make use of BioAPI and it enables plug & play biometric components. The BioAPI was wrapped by adding another layer in-between the BioSys Server and the BioAPI.

This in-between layer (referred as the BioAPI Wrapper) was essential since the BioAPI reference implementation is written in C/C++. It was not directly accessible through Microsoft .Net C# as the BioAPI deals with multiple levels of indirections of pointers and union types. Therefore it was essential to have such a layer. By doing so it did made the task of the BioSys development much easier and the authors were able to make it even simple for the application developer to work with the BioSys rather than with the BioAPI.

Considerable interest and effort was put onto make development as simple as possible. The decision was to make use of web services [5, 7-8]. Web services allow high level programming module with platform independence, centralised control with sufficient scalability. Use of web services (referred as the BioSys Service) makes it suitable for an enterprise level networked environment, which only requires application clients to support SOAP and HTTP messaging. Therefore any programming language supports SOAP and HTTP can be used to develop applications that make use of the BioSys. Use of web services allows the BioSys to extend beyond a LAN or Intranet into the public Internet. This is useful in cases where remote users and mobile users want to get authenticated using biometrics.

Use of web services allows applications and Console (is a separate application where all the administrative stuff is done) to be of any platform but only the server to be limited to a specific platform. Web service was developed using Microsoft .Net C# and currently will only work with Microsoft IIS.

Microsoft .Net C# was selected not just because it supports Web services, there were two other concerns. First: it was not possible to use other languages (other than C++) to access complex data structures and pointers that the BioAPI extensively requires. Even support of C# is limited up to a certain level. The authors could have used C++, but writing web services in C++ is tedious and nothing much will be gained since this is a research prototype. The second concern was the performance.

It is unrealistic to ask a biometric device to support web service and send whatever it captures to the server for the processing. It is highly encouraged that image

processing be carried out in the BioSys Server since it is secure doing it at the server and it will reduce the processing overhead of the device (most devices do have limited processing power). Device make use of socket connection when communicating with the server. System should also support RS232/485 protocols if it to be commercially successful.

The Console is a separate management station, which can either reside on the same machine as the server or in a different host. It is the place where all the policies are defined and monitoring is done. All the user information, management policies, login and user biometric records are stored in a centralised database and could be extended to distributed databases if required.

## 6.0 The future

The developed system only supports some of the very basic network and security practices. It should be redesigned addressing security from bottom-up to manifest high-level of security. This is not because the current design is bad it is because security should never be a separate layer; it should be an integral part of the whole system. These things were left out in the prototype due to resource limitations.

Current biometric infrastructure of an organization consists of lots of non-BioAPI compliant devices. If these devices can be transferred to become BioAPI compliant it would reduce a lot of reinvestment. BCB Generator (BioAPI Compliant BSP Generator) is such an approach where it tries to automate the processes which could transform non BioAPI device to become BioAPI compatible.

The Universal BioSys can be easily extended to the public Internet with enhanced security since it is already exposed as a web service. As with many biometric systems the Universal BioSys can also be used as a time and attendance system that could directly integrate with a payroll system. Necessary data is already available within the system and it is just a matter of organizing them.

The BioSys could also extend into a ticketing system like Kerberos or integrate with Domain management system such as Microsoft Active Directory [9]. In order to enhance security certain vendors uses a combination of biometric technologies (example: face, lip movement and voice combined recognition system). The Universal BioSys could support such mechanisms as well.

## 7.0 Conclusion

Since the Universal BioSys is a Proof of Concept there are no measurable results. It has proven that biometric integration can be made seamless and effortless while

enforcing management and security practices. It offers a biometric enabled third-party authentication which is platform, vendor and biometric technology independent. It also reduces the TCO considerably and enforces tighter security with two unique features.

This solution would be more suitable for a medium to large-scale organization that has stringent security requirements and need to install many biometric devices.

Authors work could be used as a framework in developing next generation biometric based third-party authentication systems.

## 8.0 Case Study

Consider a hypothetical software development company who is developing high quality products and involved in top class research. Assume that its floor arrangement is something similar to Fig 2. Suppose they would like to have a balance between tighter security and TCO. If they are only concentrating on physical access and servers they may install different type of devices on different locations as shown in Fig 2 with multiple access level.

Altogether they may install 8 devices (D1 to D8) and device D8 will be shared by the 2 servers. If they had 10 servers on the server room they still only need 1 device which can be shared by all the servers. Therefore the total investment is less than having a dedicated device. If someday the organization decide to incorporate all the hosts with the BioSys it is possible as well. If they have 100 hosts which are used for development and R&D they do not need 100 biometric devices they will be able to do it with 5-10 devices. In minimum they only need 2 where one for the IT and another one for the R&D. Since they have already invested for the BioSys they will reduce their cost from 90% (with 10 devices) to 98% (with 2 devices).

According to the Universal BioSys system administrator or security office first has to draw a floor arrangement such as the Fig 2. Then based on that he/she can indicate where different doors, room, departments, hosts, servers, vaults, client applications, etc. are located. Then he/she is required to install biometric devices and register them with the system as well. Thereafter the Device-Hierarchy has to be defined using a graphical interface which is similar to Fig 3.

Finally users have to be enrolled to use applications. Enrolment involves a user, application and a biometric device. Users are enrolled into a particular class of devices rather than to individual devices (i.e. there are

device classes such as fingerprint, facial, iris, etc. and if a user is enrolled to fingerprint class he/she can use any fingerprint device). This is referred as single point of enrolment.

When things are configured correctly system will identify potential users and allow them any access based on predefined policies. However if someone tries to bypass the Device-Hierarchy either; purposefully or by accident it will not allow any access although the given credentials are correct.

## Acknowledgement

Over the last six months many people have contributed this project. In particular, the authors thank Mr. Chathura de Silva who was the project supervisor as well as the final year project coordinator. Authors would also like to thank Dr. Sanath Jayasena, the Head of Department of Computer Science and Engineering and other staff members. Finally authors would also like to thank each and very individual that the authors have not mentioned above, who gave even the slightest support through the lifetime of the project.

## References

- [1] Universal BioSys web site  
[www.biosys.net.tc](http://www.biosys.net.tc)
- [2] The BioAPI Consortium, "BioAPI Specification Version 1.1", 16th March 2004.  
Available: <http://www.bioapi.org>
- [3] The Independent Security Server, developed by Info Data, Inc.  
Available: <http://www.infodatany.com/independentsecurityserver.htm>
- [4] The WhoIsIt biometric server for E-commerce  
Available: [http://www.qvbiometrics.com/E\\_Metrics\\_server.htm](http://www.qvbiometrics.com/E_Metrics_server.htm)
- [5] De Silva S. M. R. P, Weerasinghe P. W. H. D, Bandara H. M.N. D, "Universal BioSys - A literature review"  
Available: <http://www.cse.mrt.ac.lk/~ravids/literal.html>
- [6] De Silva S. M. R. P, Weerasinghe P. W. H. D, Bandara H. M.N. D, "Universal BioSys", final project report.  
Available: <http://www.cse.mrt.ac.lk/~ravids/literal.html>
- [7] Lakshmi Ananthamurthy, "Introduction to Web Service"  
Available: <http://www.developer.com/services/article.php>
- [8] Heather Kreger, IBM Software Group, May 2001, "Web Services Conceptual Architecture (WSCA 1.0)",  
Available: <http://www-306.ibm.com/software/solutions/webservices/pdf/WSCA.pdf>
- [9] Microsoft, "Active Directory Overview", 30/06/1999  
Available:  
<http://www.microsoft.com/windows2000/server/evaluation/features/dirlist.asp>