Best Practices in Biometrics Performance Monitoring Programs

Version 1.0

September 2011
Irvine, California, USA
Karlskrona, Blekinge, Sweden

Authors:
Babak Goudarzi Pour
Victor Adolfsson

Optimum Biometric Labs
www.optimumbiometrics.com

A co-founding member of Swedish National Biometric Association
www.biometricassociation.org

Table of Contents

Foreword

Since its inception in 2003, Optimum Biometric Labs has been pioneering (see milestones) the research and development of one essential and multi-disciplinary (yet highly interconnected) area within biometrics. In the current ISO standardization work, most parts of this convergence area have been outside the scope and hence lacked a dedicated standard, although ISO/IEC FCD 19795-6.2 encourages some parts of it.

We have gathered this multi-disciplinary area in a holistic framework in this **Best Practices in Biometrics Performance Monitoring Programs** which focuses on the use of standards, methods, processes, and IT tools to support end-users' and businesses' real-world expectations associated with Reliability, Availability, Maintainability, and Performance of biometric-based verification and identification systems and applications.

This Best Practices is our contribution to the biometric industry and end-user community and is intended for everyone who works or is involved with biometrics. It is recommended to diverse types of organizations:

- End-customers
- Maintenance & support centers
- Prime contractors and system integrators
- Biometric vendors
- Standardization bodies
- Biometric fora and associations

And key role functions:

- CIO, CSO
- IT / System Administrator, Operator, First-line support
- System Architect, Developer
- CTO, Product Manager, Sales Manager

This Best Practices is a "living" document and we are committed and will do our best to improve its structure and content as the discipline and its community move forward. We encourage review, comment, criticism, correction, and contribution by the biometrics industry and end-users. You can find "Leave a comment" field on every page of its dedicated site, www.BiometricsPerformanceMonitorin.com**,** and also a dedicated category for Guest Authors' posts and a Feedback page.

For future versions and expansion of this Best Practices, other important and adjacent areas whose addition should be valuable include:

- Security and vulnerability
- Usability/User experience (user experience is one central thread in this Best Practices. Perhaps, it would be beneficial to develop and apply more dedicated measurement framework and methods)

Methods and philosophies presented in this first version of Best Practices are developed by Optimum Biometric Labs. We intend to seek international consensus to collectively develop this Best Practices into an ISO standard. We will initiate this consensus building by formulating and sending a New Work Item Proposal, NWIP, via the Swedish National body SIS to ISO/IEC JTC1/SC37 - Biometrics.

Introduction

This Best Practices in Biometrics Performance Monitoring, BPM, programs is concerned with the use of standards, methods, processes, frameworks, and IT tools to support end-users' and businesses' expectations associated with **Reliability**, **Availability**, **Maintainability**, and **Performance** of biometric-based verification and identification systems and applications.

Biometrics Performance Monitoring, which in essence is comparable to [Application Performance Monitoring (APM)] and [Event Correlation and Analysis (ECA)], is using real-time data to **detect**, **diagnose**, **report**, and **recover** issues, or potential issues, in order to ensure that end-customers' business goals and requirements are met or exceeded.

The ultimate goal with establishing an efficient Biometrics Performance Monitoring, BPM, program is to ensure that biometric end-customers' business goals and requirements are met or exceeded. The three distinct goals of the discipline is to:

- Measure, analyse, and improve operational Reliability and Performance metrics
- Minimize downtime (Availability)
- Minimize maintenance and service needs and thereby their costs, and to minimize the mean time to detect/ isolate and resolve issues (Maintainability)

These objectives are formalized as end-user goals, requirements, and expectations in a Service Level Agreement (SLA). The positive side effects of these distinct goals are, among others, improved customer/user satisfaction and minimized Total Cost of Ownership (TCO) of your biometric system/service.

Section 3 lists and analyses the relevance of international standards to this Best Practices.

Section 4 defines essential metrics and terms in Biometrics Performance Monitoring, BPM.

Section 5 describes what Biometrics Performance Monitoring is and why it is needed; it lists some of the real-world factors that influence reliability, availability, maintainability, and performance of biometrics systems. It presents the three distinct goals of Biometrics Performance Monitoring.

Section 6 presents five steps for implementing and adopting a successful Biometrics Performance Monitoring, BPM, program. It explains Service Level Agreement (SLA) and why it is important. Further, it presents a method (and a calculator) on how to identify and estimate operational costs that are related to management of Reliability, Availability, Maintainability, and Performance in biometric-based applications.

Section 7 describes functions and properties of general Biometrics Performance Monitoring, BPM, tool.

Section 8 defines the relationship between units/elements, events, alerts, and metrics in a Biometrics Performance Monitoring tool.

Section 9 is normative, and specifies how to integrate biometric-based systems and applications with a BPM tool. It defines BPM tool's Application Programming Interface (API).

Annex A is informative, and answers to some sample questions centered around Reliability, Availability, Maintainability, and Performance. These questions are formulated in a way to assist Biometrics Operations management team and personnel.

Annex B is informative, and lists relevant metrics in a Service Level Agreement.

Annex C is informative, and when developed for the upcoming versions will give an example of a Service Level

Agreement.

Annex D is informative, and when developed for the upcoming versions will give a sample of symptoms and their possible causes.

Annex E is informative, and describes how Biometrics Performance Monitoring added value in an enrollment application (an actual case study).

Annex F is informative, and is a supporting part to Section 8. It outlines and explains details of BPM tools for buyers.

Annex G is related to Section 9. G.1 is normative and specifies List of Status Codes for BPM tool's APIs (web services). The annex also specifies Example of Event Constants.

Best Practices in Biometrics Performance Monitoring Programs

1 Overview and Scope

This document is an introduction to the best practices in Biometrics Performance Monitoring, BPM, Programs. It presents the following parts:

- Informative standards and their relevance
- Terms and Definitions
- What is Biometrics Performance Monitoring?
- Why is Biometrics Performance Monitoring needed?
- Three distinct goals of Biometrics Performance Monitoring
- Five steps towards implementing and adopting a successful Biometrics Performance Monitoring, BPM, program
- What is Service Level Agreement (SLA) and why is it important?
- How to estimate biometrics operational costs
- Functions of general Biometrics Performance Monitoring, BPM, tool
- Properties of general Biometrics Performance Monitoring, BPM, tool
- Relationship between units/elements, events, alerts, and metrics
- Interface between the biometric system and the Biometrics Performance Monitoring, BPM, tool
- BPM Application Programming Interface (API)
- Q&A related to Reliability, Availability, Maintainability, and Performance in the context of BPM
- Example of metrics in a Service Level Agreement
- Sample Service Level Agreement
- Sample of symptoms and their possible causes
- Case studies
- How to evaluate IT tools for Biometrics Performance Monitoring?

2 Conformance

Section 9 and Annex G specify the conformance requirements for Units/Elements claiming conformance to this Best Practices in Biometrics Performance Monitoring Programs.

Currently, there is no other conformance requirement to this Best Practices. Optimum Biometric Labs aims to seek enthusiasm, consensus, and collaboration among the organizations and individuals in the biometric industry and end-user community to develop this Best Practices into a useful and vital ISO-standard. We believe the most appropriate environment for this endeavor would be the technical committee ISO/IEC JTC1/SC37 - Biometrics.

3 Informative standards and their relevance

There are a number of international standards that are relevant and useful with regards to Biometrics Performance Monitoring. See also the section **Useful resources** for other relevant national and international guidelines.

ISO/IEC 2382-14 Information technology – Vocabulary – Part 14: Reliability, maintainability, availability
Note: defines and explains general terms in Reliability, Maintainability, and Availability

ISO/IEC 19795 (all parts) Information technology - Biometric performance testing and reporting

- **Part 1: Principles and framework**
  Note: among useful and relevant parts are 1) Terms and definitions 2) General biometric system including description of components, functions, and performance measures 3) factors influencing performance 4) fundamental performance metrics

- **Part 2: Testing methodologies for technology and scenario evaluation (ISO/IEC 19795-2:2007)**
  Note: among useful and relevant parts is what specification details (system information) to collect and log about the biometric system/product. Furthermore, Biometrics Performance Monitoring, BPM, tool can be customized and used for Technology and Scenario evaluation if it enables a test to claim conformance to the set of the specified clauses in the standard.

- **Part 3: Modality-specific testing (Technical Report) (ISO/IEC TR 19795-3:2007)**
  Note: 1) When designing evaluation tests, this TR's consideration of performance influencing factors for a given modality should give fast and straightforward hints in narrowing down the selection process (inclusion/exclusion) of which parameters/metrics to monitor based on a given modality 2) The Robustness test is presented as a useful method to determine the amount of change in performance (performance sensitiveness) as a function of change of a modality-specific influential factor. This, in Biometrics Performance Monitoring, seems to be an appropriate notion to define adaptive baseline performance level as a function of (triggered by) the change of an influential factor (e.g. temperature) for a certain biometric modality.

- **Part 4: Interoperability performance testing (ISO/IEC 19795-4:2008)**
  Note: 1) normative requirement of 'Measuring component failure' (beside transactional error rates) by introducing and defining the properties 'failure to process' and 'component-level failure rate' 2) Examples given regarding component-level failure, their root-cause, and in what phases/functions they may occur.

- **Part 5: Access control scenario and grading scheme (ISO/IEC 19795-5:2011)**
  Note: 1) Centered around access control applications (and truly inspirational for other biometric application types) this standard introduces one important aspect which is a central topic in this Best Practices and that is to enable the customer to state its performance requirements on error rates and transaction times (Grade levels and their corresponding metrics). The framework establishes a grading scheme (level of performance with statistical significance) that can be applied when setting the corresponding metrics in both Service Level Agreements and Baseline Performance Levels of BPM tools (presented in this Best Practices). 2) Similar to Part 2, logging detailed information/specification of the biometric system is required.

- **Part 6: Testing Methodologies for Operational Evaluation (ISO/IEC FCD 19795-6.2**, Target publication date: 2012-06-17**)**

Note: among useful and relevant parts are 1) Terms and definitions 2) The highly relevant ‘Operational system monitoring’ is presented in an annex with one particular detail: to graph performance metrics as a function of time in order to detect potential abnormalities and tendencies 3) Due to unknown ground truth (identity claim), BPM is comparable to Operational Evaluation with regards to relevant performance measures which are presented in this FCD 4) Presented 'Sub-goals of operational testing' are highly relevant to this Best Practices; e.g. one can forward/input the obtained performance data from pilots to BPM programs when setting the corresponding metrics (benchmark levels) in both Service Level Agreements and Baseline Performance Levels of BPM tools (presented in this Best Practices). 5) logging detailed information/specification of the biometric system and relevant environmental data 6) ‘Non-mandatory performance metrics’ presented in an annex are useful for defining and measuring additional metrics in BPM programs.

- **Part 7: Testing of on-card biometric comparison algorithms (ISO/IEC 19795-7:2011)**
  Note: In realm of BPM, what is relevant here is the absence (stated as outside of scope) of methods for evaluating the performance of IC cards readers along with ruggedness or durability of the card.

ISO/IEC 29197 Introductory element — Evaluation Methodology for Environmental Influence in Biometric Systems (ISO/IEC WD 29197, Target publication date: 2013-08-11)
Note: this is highly relevant for this Best Practices when it is published as a Standard (see also notes on Part 3: Modality-specific testing).

ISO/IEC 19784 (Parts 1, 2, 4), Information technology - Biometric application programming interface
Authors' note: we need to study these standards in more details in order to identify their relevance to this Best Practices.

- **Part 1: BioAPI specification (ISO/IEC 19784-1:2006)**
  Note: Back in 2003, when Optimum Biometric Labs initiated its work with BPM, it defined all biometric-related components (and any related subsystem component) as Units. Incidentally, this is exactly how Units (BioAPI Units) are defined in this standard (hardware or software or a combination of both e.g. Sensor, Archive, Matching algorithm, Processing algorithm). This notion helps to make this Best Practices more familiar to users and system integrators as BioAPI is commonly used in the industry.

ISO/IEC DIS 2382-37 Information technology -- Vocabulary -- Part 37: Harmonized biometric vocabulary (ISO/IEC DIS 2382-37, Target publication date: 2013-01-24**)**
Note: when published, this standard should be relevant with regards to the Terms and Definitions within its scope.

SP 500-288 - Rev 0 - Draft 2 (WS-BD is a specification describing how to expose a biometric sensor to various clients via web services.)
Note: Back in 2003, when Optimum Biometric Labs initiated its work with BPM, it chose to base the interface of its BPM tool on Web Services; predicting that biometrics devices and services will eventually choose the standard as interface. Three years later, in 2006, NIST initiated the Biometric Web Services (BWS) project. Thus, this standard initiative is an invaluable support for this Best Practices as web services (WSDL) is also the interface (API) between the biometric system and Biometrics Performance Monitoring, BPM, tool.

4 Terms and Definitions

This section defines some of the essential metrics and terms in Biometrics Performance Monitoring, BPM, programs. It is structured in the following categories:

- Reliability metrics
- Availability metrics
- Maintainability metrics
- Performance metrics
- Environmental metrics
- Maintenance strategy terms
- Service Level Agreement terms
- BPM tool related terms
- Abbreviated terms

4.1 Reliability metrics

Reliability can be defined as the probability that a system, a sub-system or a component will operate successfully at a given time.

Mean-Time-Between-Failure (MTBF) is the average time duration between failures of a functional entity under given conditions. It is typically applicable to entities that are “repairable”, i.e. supposed to be fixed and then returned to operation.

Mean-Time-To-Failure (MTTF) is the average time duration to the first failure of a functional entity under given conditions. It is only applicable to “non-reparable” entities where the entity is replaced with a new one.

> *Example: A supplier claims that its fingerprint sensor has a MTTF of 7300 hours. MTBF and MTTF are many times estimated using predictive analytical models. However: True measures on MTBF or MTTF are always computed using collected data from real-world operations.*

4.2 Availability metrics

Downtime is the total (accumulated) time in an observed or predetermined period of time that the biometric device/service is unavailable to users. Note that downtime, during the intended operating period, is calculated independently of its underlying reasons such as planned or unplanned shut-downs (due to e.g. maintenance, updates, power failures etc.).

Uptime is the total (accumulated) time in an observed or predetermined period of time that the biometric device/service is available to users.

Operational Availability (A_o) is the actual availability that the customer experiences. It is easily calculated by: uptime / (uptime+downtime)

> *Assume that your biometric service must serve 9 hours per day for 365 days per year and you allow for a maximum of 10 hours downtime annually. Then: The calculated A_o is 99.70% which can be used to set the required operational availability in a SLA contract.*

4.3 Maintainability metrics

Mean-Time-to-Recovery (MTTR) is the average time it takes to restore an entity to operational status after it has failed to function. MTTR is also interpreted as Mean-Time-to-Repair/Replace/Resolve/Restore. Note that with a MTTR of e.g. 1 hour the supplier does not guarantee an uptime (or even notification of the problem) within 1 hour. What the supplier states is that the recovery action is estimated to approach around 1 hour. Therefore, MTTR should not be used if the end-user requires guaranteed uptime, after a failure, within a given timeframe.

Maximum-Time-to-Recovery which is the maximum allowed time that includes taking a corrective maintenance action until the operational status is achieved.

> *In a maintenance contract, where budget allows and nature of mission is critical, we recommend to use Maximum-Time-to-Recovery.*

4.4 Performance metrics

This section divides performance metrics in three main categories: enrollment, matching, and transaction time.

4.4.1 Enrollment performance metrics

Failure-to-enrol rate (FTE)

Failure-to-acquire rate (FTA)

Image quality score

4.4.2 Matching performance metrics

Unit/System acceptance rate

Unit/System rejection rate

Unit/System identification rate

Technology and Scenario evaluation related performance metrics:
False non-match rate (FNMR)
False match rate (FMR)
False reject rate (FRR)

False accept rate (FAR)
Generalized false reject rate (GFRR)
Generalized false accept rate (GFAR)

False-negative identification-error rate
False-positive identification-error rate
Cumulative match characteristic (CMC)

4.4.3 Transaction time performance metrics

Throughput for enrollment Transaction (Transaction time)

Throughput for recognition Transaction (Transaction time)

Some other performance related metrics and stats:
Size of database
Number of enrollees
Total background database
Total number of samples

4.5 Environmental metrics

Temperature

Relative humidity

Pressure

Illumination

Noise

4.6 Maintenance strategy terms

Corrective (reactive) maintenance is a recovery action performed after failure of a functional entity in order to restore it to its operational status.

Preventive maintenance (PM) is performed periodically in order to reduce the probability of failure or deterioration of a functional entity.

Scheduled (routine) maintenance is a preventive maintenance performed according to an established time schedule or a predetermined frequency of usage.

Predictive maintenance (PdM) is to use past trends to predict failures. Also called Condition-based monitoring.

Proactive maintenance (Proactive monitoring) is the latest, most popular and probably the most cost effective of all maintenance strategies since it selects the best of several maintenance methodologies with a holistic view of vigilance in mind. It is often interpreted and structured in slightly different ways depending on who advocate it and in which industry. The bottom line is: Proactive maintenance focuses on pinpointing and eliminating the root-causes rather than symptoms of failure. It makes use of e.g. continuos monitoring, analyzing, detecting, diagnosing and responding actions.

4.7 Service Level Agreement (SLA) related terms

Service Level Agreement (SLA) is an agreement document commonly used in the IT and other industries which is often negotiated and mutually agreed between two organizations or departments.

Baseline Performance Level: a level of (point of reference) performance at which the system’s (or a sub-system's) performance is either as expected, minimum required, or satisfying.

4.8 BPM tool related terms

Units/elements are all supporting parts (including abstract processes) that enables a biometric system to function. Examples of units/elements are capture device, biometric algorithm, algorithm licensing expiration date, server, hard drive, operating system, database, network, user interface, any end-to-end service/business process, etc.

Event is an occurrence of communication between the monitored unit/element (client) and the BPM tool. An event can be an alert as well.

Alert: An alert refers to a critical event. It is triggered based on events or absence of them.

Metric: a metric (also known as Key Performance Indicator, KPI, a measurement system, or simply Stat/Statistic) is anything quantifiable and measurable. Metrics are all things that are variable and can be measured; and therefore improved.

BPM Client/Agent is a monitoring agent that sends Installation Events, System Events, and User Events to, and fetches Command Events from, the BPM server.

4.9 Abbreviated terms

The following abbreviated terms apply here.

API	Application Programming Interface
APM	Application Performance Monitoring
BPM	Biometrics Performance Monitoring
ECA	Event Correlation and Analysis
MTBF	Mean-Time-Between-Failure
MTTF	Mean-Time-To-Failure
ROI	Return on Investment
SOA	Service Oriented Architecture
SLA	Service Level Agreement
TCO	Total Cost of Ownership
WSDL	Web Services Description Language

5 What is Biometrics Performance Monitoring?

Biometrics Performance Monitoring or Management, BPM, is the use of standards, methods, processes, frameworks, and IT tools to support end-users' and businesses' expectations associated with Reliability, Availability, Maintainability, and Performance of biometric-based verification and identification systems and applications.

Biometrics Performance Monitoring, which in essence is comparable to Application Performance Monitoring (APM) and Event Correlation and Analysis (ECA), is using real-time data to detect, diagnose, report, and recover issues, or potential issues, in order to ensure that end-customers' business goals and requirements are met or exceeded.

5.1 Why is Biometrics Performance Monitoring needed?

The quote "you can't manage what you don't measure" is especially true for biometric-based identification and verification systems and applications since they are particularly more sensitive to human and environmental-based factors than other IT-systems. Here is a list of some of the real-world factors that influence reliability, availability, maintainability, and performance of biometrics systems:

Human factors:
- User age
- User demographics (ethnicity, gender, and occupation)
- User acceptance and motivation
- User habituation and familiarity
- User physiology (dry or worn fingers, skin tone, etc.)
- User behaviour and interaction (pose, expressions, etc.)
- User appearance (glasses, contact lenses, etc.)
- User's biometric template aging
- Fraudulent activities such as spoofing
- Accidents
- Configuration/maintenance caused issues

External environmental conditions:
- Light condition (*e.g.* face and iris)
- Noise and vibration (*e.g.* voice)
- Dirt, temperature, humidity (*e.g.* fingerprint)

System (hardware, software) related issues:
- Capture device related issues (e.g. camera lens out of focus, sensor deterioration, etc.)
- Software failure (UI or backend system not accessible, etc.)
- Algorithm license expiration
- Server downtime
- User Interface (UI) related issues (e.g. failure, user training and instruction, system feedback)
- Other subsystem component breaches and deterioration

5.2 Three distinct goals of Biometrics Performance Monitoring

The ultimate objective with establishing an efficient Biometrics Performance Monitoring program is to ensure that biometric end-customers' business goals and requirements are met or exceeded. The 3 distinct goals of the discipline is to:

- Measure, analyse, and improve operational Reliability and Performance metrics
- Minimize downtime (Availability)
- Minimize maintenance and service needs and thereby their costs, and to minimize the mean time to detect/ isolate and resolve issues (Maintainability)

These objectives are formalized as end-user goals, requirements, and expectations in a Service Level Agreement (SLA). The positive side effects of these distinct goals are, among others, improved customer/user satisfaction and minimized Total Cost of Ownership (TCO) of your biometric system/service.

6 Five steps towards implementing and adopting a successful Biometrics Performance Monitoring, BPM, program

When designing Biometrics Performance Monitoring programs, the following sequence should be applied to ensure a successful implementation and a sustainable adoption:

1. Plan your Biometrics Performance Monitoring, BPM, program
    a. Define and establish your Service Level Agreement (see: What is Service Level Agreement (SLA) and why is it important?)
        i. Define your biometric services based on/aligned with your business objectives
        ii. Select the associated metrics and set their baseline/target levels,
        iii. Specify the duties of the parties (e.g. Determine actions such as corrective measures),
        iv. Specify financial penalties (if applicable)
    b. Define details and tasks of your BPM tool
        i. Identify and list all the supporting units/elements (subsystem parts)
        ii. Identify, list, and select appropriate units'/elements' events associated with reliability, availability, maintainability and performance (see vendor's API) and/or define your own events
        iii. Define critical events/alert rules that best indicate that your defined business objectives are at risk (see Alert types and alerting methods)
2. Build your biometrics operations support team and assign each individual with clear responsibilities
3. Identify and quantify all costs and risks related to reliability, availability, maintainability and performance (see How to estimate biometrics operational costs)
4. Select and evaluate your IT tool for Biometrics Performance Monitoring, BPM
5. Implement the above and periodically review and evaluate the results (e.g. use the ROI-calculator) and if necessary make adjustments to steps 1-4

Tips:

- Start small and simple: narrow down your planning to only cover the core business value you want to deliver to the user; **Focus only on the essential units/elements that make that possible** (e.g. only monitor your end-to-end biometric service)
- Define, implement, and log as many Event types as you want but minimize the type of alerts in the beginning, you can always add more later (too many alerts, or duplicate of alerts, jeopardize the purpose with a BPM tool)

6.1 What is Service Level Agreement (SLA) and why is it important?

Service Level Agreement (SLA) is an agreement document commonly used in the IT and other industries which is often negotiated and mutually agreed between two organizations or departments. SLA is often established between the biometric end-user (company/organization/department/user group) and the biometric service supplier/operator (which can be an outside vendor/supplier our an in-house IT department). The goal is to measure, meet, and exceed the agreed expectations and requirements in the Service Level Agreement (SLA).

SLAs can contain parts such as service definition, associated metrics and their baseline and target levels, and actions such as corrective measures, duties of the parties, and even financial penalties in case of consistent non-compliance with the SLA. For a list of metrics in an SLA see Annex Example of metrics in a Service Level Agreement.

6.2 How to estimate biometrics operational costs

This step, among others, is important for comparing/benchmarking the targeted cost saving against the Total Cost of Ownership of your investment in your BPM program (BPM tool's costs such as licenses, professional services, etc.).

In biometric-based systems and applications, as in all technology-based solutions, there are different types of direct and indirect operational costs involved. Here, the focus is to identify and estimate those that are related to management of **Reliability**, **Availability**, **Maintainability**, and **Performance**.

> The goal with Biometrics Performance Monitoring is to reduce those costs by applying the Best Practices and using the right tools.

This simple yet advanced calculator is developed (access it here) to help biometric customers to identify relevant information such as requirement and metrics in order to estimate the total operational cost of biometric projects and installations. The calculator consists of the following parts:

- Step 1 - Enter # of your subsystem components that need to report their health and performance
- Step 2 - Enter metrics on your availability requirement
- Step 3 - Enter metrics on your acquisition device maintenance
- Step 4 - Enter metrics on your root cause analysis and diagnosing
- Step 5 - Enter metrics on your operational performance testing (Baseline-Performance Comparison), Continuous Monitoring and Analysis (if applicable)
- Step 6 - Enter metrics on your Service Level Agreement, SLA (if applicable)

The calculator can be accessed here: https://docs.google.com/a/optimumbiometrics.com/spreadsheet/ccc?key=0AmnvEm-lBBZNdHBNQV85dDRZRjJhNXhuZk9lWjVaREE&hl=en_US#gid=0

7 Biometrics Performance Monitoring, BPM, tool

7.1 Functions of general Biometrics Performance Monitoring, BPM, tool

The following are the functions and capabilities of the general Biometrics Performance Monitoring tool:

- Creating and keeping detailed records of biometrics installations (e.g. address, contact list for each organization, installation type and location, time zone, unit/element names and types, serial numbers, comments) and easy drill-down navigation to an individual unit/element view (the lowest level of hierarchy) from an bird's eye view of all sites and installations
- Collecting, consolidating, and logging events generated from all units/elements in a biometric system (e.g. capture device, biometric algorithm, algorithm licensing expiration date, server, hard drive, operating system, database, network, user interface, any end-to-end service/business process)
- Processing/sorting/organizing/filtering collected events
- Detecting, diagnosing, and presenting/notifying critical events (alerts) and root-causes to biometrics operations personnel
- Generating reports on operational reliability, availability, maintainability, and performance metrics (e.g. goals and parameters defined in a Service Level Agreement)
- Proposing and performing corrective actions

7.2 Properties of general Biometrics Performance Monitoring, BPM, tool

This section aims to give an overview of diverse features and properties of a Biometrics Performance Monitoring tool which should give a good idea on how to look for and evaluate available BPM tools in the market:

7.2.1 Features

- **Event logging/consolidation/grouping:** e.g. organizing events based on units/elements, sites/installations, business processes/units, departments/organizations/geographic regions, global (bird's eye view), or any variant of user-defined filtering
- **Event/alert analytics:** capable of detecting and diagnosing critical events (alerts) and root-causes
- **Event/alert user interface (UI):** presenting events and instant notifications via web/desktop interface (dashboard, console), mobile/tablet app, email, etc.
- **Reporting:** e.g. using visualization/analytical tools to generate automatic or on-demand reports in diverse formats e.g. graphs/numerical values on e.g. SLA metrics or other properties and distributions
- **Corrective actions:** e.g. capable of setting/changing a parameter or triggering an action in the monitored unit/element
- **Inventory/record of biometric installations:** keeping lists of geographically spread biometric sites and installations with detailed information about them for reasons such as archiving or remote troubleshooting (detail record such as address, contact list for each organization, installation type and location, time zone, unit/element name, type, serial number, comments, etc.)

- **Drill-down navigation:** capable of drilling-down to an individual unit/element view (the lowest level of hierarchy) from the bird's eye view of all sites and installations

7.2.2 Configuration/customization

- **User-defined events:** capable to define, add, and modify new type of events
- **User-defined alerts:** capable to define, add, and modify new type of alert rules and logic
- **Event update frequency:** capable of setting/changing how often units/elements report their events to the BPM tool (e.g. every five minutes, etc.)
- **User Interface:** support for developing the UI, or parts of it, for different Eco-systems and platforms (e.g. iOS, Android, etc.)
- **Site administration:** adaptable to your requirements such as creating and adding: New organizations, installations, and units, New type of installations and units

7.2.3 Flexibility and compatibility

- **Biometric modality/type:** Biometric agnostic/independent or can only work with certain biometric types
- **Biometric sample acquisition sensor type:** Sensor-agnostic/independent or can only work with certain capture devices
- **Unit/element type:** ability to monitor any user defined unit/element
- **Vendor independent:** capable of working with hardware and software from any vendor (vendor agnostic/ independent) or can only work with certain vendors HW & SW (e.g. if biometric or sensor agnostic then most likely vendor independent too)

7.2.4 Scalability and Performance

- **Multi-unit/element management:** e.g. capable of monitoring # of units/elements with acceptable performance (based on a certain BPM server hardware configuration)
- **Multi-site management:** e.g. support of unlimited # of user-defined sites/installations, business processes/ units, departments/organizations/geographic regions
- **BPM tool users:** e.g. support of unlimited # of users

7.2.5 BPM tool user management

- **User Roles and Access Management:** support different types of user roles with different access rights (e.g. super admin, admin, operator, etc.)

7.2.6 Security

- **Data encryption:** Encrypted data transfer (e.g. SSL) for every event transmission between a monitored unit/element (or its agent) and the BPM Server
- **Agent-BPM server authentication:** Authentication for every event transmission between a monitored unit/element (or its agent) and the BPM Server

7.2.7 Technical requirement for BPM Server

- Minimum HW requirements: Type of processor, available disk space and RAM, network interface
- Supported OS (e.g. Windows, Mac OS X, Linux)

7.2.8 Technical requirement for BPM tool front-end

e.g. Internet connection, web browser (e.g. Firefox, IE, Safari, Chrome, etc.), etc.

7.2.9 User/customer support

- Guideline and tutorials such as UI manual, installation manual, troubleshooting help, white papers, etc.
- Online support (e.g. web portal)
- Telephone support

7.2.10 Integration support

- **Agent/client interface/mechanism:** has its own agent description/method and/or is agent-less (e.g. SNMP) for event collection
- **API technology:** e.g. open API for developing customized agents. e.g. based on Service Oriented Architecture (SOA), Web services-based platform (WSDL interface), SOAP for integration with biometric-based systems, rapid and easy to integrate with, etc.
- **API documentation and sample code**

7.2.11 Business/delivery model

- Open source
- Proprietary
- Software-as-a-Service (SaaS), Cloud-based
- Appliance (software bundled and shipped with dedicated hardware)

7.2.12 Pricing

- Attractive price point
- Flexible licensing policy (e.g. license based on # of units/elements)
- Updates, maintenance, and support
- Amount of necessary/needed related professional services (business case analysis, planning, installation, integration, testing, optimization, etc.)
- Total Cost of Ownership (TCO) analysis
- Available Return on Investment (ROI) analysis and calculator for e.g. benchmarking cost savings against Total Cost of Ownership (TCO)

7.2.13 Documented customer case study

Tried-out and used by verifiable customers

7.2.14 Standards

Compliant with international, national, and industry standards

7.2.15 Independent test/audit

Evaluated by independent test organizations for e.g. scalability, performance, usability, interoperability, reliability, security and vulnerability, etc.

8 Relationship between Units/Elements, Events, Alerts, and Metrics

The relationship between events and alerts is that an event refers to an occurrence of communication between the monitored unit/element (client) and the BPM tool whereas an alert refers to a critical event. Thus, an event can be an alert as well.

In most cases an event is sent from the unit/element (client) to the BPM tool. However there may be cases (e.g. controlling a property) when the unit/element (client) receives some communication from the BPM tool as well, but this communication will always be triggered from the unit/element (client) itself (i.e. unit/element (client) will pull information from the BPM tool).

For a detail description of Units/Elements, Events, Alerts, and Metrics please see Terms and Definitions.

8.1 What is a Unit/Element?

Please see Terms and Definitions.

8.1.1 Unit/element types

Units/elements are all supporting parts (including abstract processes) in or outside of a biometric system such as capture device, biometric algorithm, algorithm licensing expiration date, server, hard drive, operating system, database, network, user interface, any end-to-end service/business process, etc.

8.2 What is an Event?

Please see Terms and Definitions. More than often, a software or hardware vendor has defined its own set of specific events for its products which means there is already a strong support those events be utilized via a BPM tool. For a list of events please see Event Constants in Appendix BPM Application Programming Interface (API).

8.2.1 Event categories (main types)

There are 4 categories of events that support different aspects of a Biometrics Performance Monitoring, BPM, tool:

- **Installation Event:** an Installation Event is used in order to communicate to the BPM tool that a particular unit/element is running and wants to register itself.

- **System Event:** The most frequently used Event category is System Event which is used to report in a particular data to the BPM tool. The monitoring is performed on basis of this data. A System Event may carry one or more data fields (e.g. transaction time) which set the foundation for generating threshold-based alerts. To generate this type of alert the received value must be compared with a defined baseline/threshold.

  Example: FreeHardDriveSpace should never be below 1GB, if it is, a ”Disk running low” alert should be generated

- **User Event:** A User Event is used for reporting in those type of data which depends upon the users' interaction with the system being monitored (e.g. user login/logout, authentication rejected etc.)

- **Command Event:** A Command Event is used for setting a particular property of the monitored unit/element.

  Example: Set Frequency

8.3 What is an Alert?

Please see Terms and Definitions.

8.3.1 Alert types and alerting methods

An alert refers to a critical event. It is triggered based on events or absence of them. A BPM tool shall have a rule-driven alerting system where alerts can be generated in two ways:

- **Condition/State-based:** an alert is triggered when an event is changed to an another event (occurrence of two consecutive events; state change).

- **Baseline Performance Level/Threshold-based:** an alert is triggered when value of an event is higher or lower than a threshold/baseline.

Further, a BPM tool must be capable of filtering alerts in active alerts (alerts that are not yet resolved) and resolved alerts (alerts that are resolved).

8.4 What is a Metric?

Please see Terms and Definitions.

9 Interface between the biometric system and the Biometrics Performance Monitoring tool

The purpose of this section is to provide guidelines on how to integrate biometric-based systems and applications with Biometrics Performance Monitoring (BPM) tools. It describes the working of BPM client/agent.

9.1 BPM Client/Agent

A BPM Client/Agent is a monitoring agent that sends Installation Events, System Events, and User Events to, and fetches Command Events from, the BPM server.

9.2 BPM Client/Agent functional architecture

BPM is implemented using **Service Oriented Architecture (SOA)**. Communication between the BPM client/agent and BPM server is done by using web services. This allows for clients developed in any language on any platform that has libraries for interacting with web services to be supported by BPM tool. Every unit that is being monitored reports its BPM declared Events via **WSDL (Web Service Definition Language)** exposed webservice functions together with a set of predetermined status codes (see Annex G).

9.3 BPM Application Programming Interface (API)

This section describes the BPM web services that are exposed to the BPM client/agent.

9.3.1 SendInstallationEvent ( )

Before starting to report any data to the BPM tool, the very first web service that should be executed is the SendInstallationEvent ( ).

SendInstallationEvent ( ) is used in order to communicate to the the BPM server that a particular Unit is now running. In response to this web service the BPM Client/Agent running for that particular Unit receives a status code and a unitId. A negative status code represents an error (see Annex G for the list of status codes). If there is no error then status code 0 is returned. The other value returned along with the status code is the unitId. This needs to be stored persistently.

Function / Service	**SendInstallationEvent ( )**	
Input Parameter Name	**Description**	**Parameter Type**
UnitUserName*	Username for authentication	String

UnitPassword*	Password for access authentication	String
installationId	Installation Id for the installation	Integer
unitTypeID	Unit type ID for this unit	Integer
ClientVersion*	Client Version	String
systemEventTimeIntervall	Time interval between regular events. Measured in seconds.	Integer
Description	Unit Name	String
Location Installation	Location	String
Comments	Any additional comments	String
dateTimeStamp	Date and time	Date Time
Return Value Name (InstallationEvent)	**Description**	**Return Value Type**
statusCode	Status code 0 signals that the web service executed without errors.	Integer
unitID	Id for the particular Unit	Integer
Comments	This web service is the first one that should be executed when a new unit is about to be monitored.	

** Optional vendor specific BPM parameters.*

9.5.2 SendSystemEvent ( )

SendSystemEvent ( ) is used to report in data to the the BPM Server. The monitoring is performed on basis of this data. This web service is used most frequently and it must be called at least every “systemEventTimeIntervall” seconds. The arguments passed to this web service may partly be fetched from persistent storage. The Unit’s current status is reported in at every SendSystemEvent ( ). See Annex G for a list of existing events.

Function / Service	**SendSystemEvent ( )**	
Input Parameter Name	**Description**	**Parameter Type**
UnitUserName*	Username for authentication	String
UnitPassword*	Password for access authentication	String
installationId	Installation Id for the installation	Integer
unitId	Unit ID for this unit	Integer
ClientVersion*	Client Version	String

simpleSubEvents	SimpleSubEvent is an array of SimpleSubEvents. Each SimpleSubEvent has two properties. i.e propertyID and propertyValue. This allows the BPM client/agent to send in the status of the unit being monitored based on several different criteria. Valid values for propertyIds can be found in Annex G Example of Event constants.	Array of Objects
dateTimeStamp	Date and time	Date Time
Return Value Name (SystemEvent)	**Description**	**Return Value Type**
numberOfCommandEvents	Signaling to the BPM client/agent the number of command events that is bound for it. Most often 0.	Integer
statusCode	Status code 0 signals that the web service executed without errors.	Integer
Comments	This web service is used most frequently. Monitoring that takes place is actually based on the data sent by this web service.	

** Optional vendor specific BPM parameters.*

9.5.3 SendUserEvent ( )

This web service is used for reporting in that data which depends upon the users' interaction with the system being monitored. For instance, user login/logout, authentication rejected etc.

Function / Service	**SendUserEvent ( )**	
Input Parameter Name	**Description**	**Parameter Type**
UnitUserName*	Username for authentication	String
UnitPassword*	Password for access authentication	String
installationId	Installation Id for the installation	Integer
unitId	Unit ID for this unit	Integer
ClientVersion*	Client Version	String
Application	The user application being monitored.	String

UserIdentifier	User Identifier, for instance the login name or user name.	String
simpleSubEvents	Data that has to be reported in to the BPM tool	Array of Objects
dateTimeStamp	Date and time	Date Time
Return Value Name (UserEvent)	**Description**	**Return Value Type**
statusCode	Status code 0 signals that the web service executed without errors.	Integer
Comments	This web service is used for reporting in that data which depends upon the users' interaction with the system being monitored.	

** Optional vendor specific BPM parameters.*

9.5.4 GetCommandEvent ( )

This web service is used for allowing the BPM client/agent to fetch commands from the BPM server. This should be executed by the client whenever it has sent a systemevent which has resulted in numberOfCommandEvents being larger than zero. Typical use cases for command events can be:

- Update the systemEventTimeInterval for a unit
- Update the driver for a large set of fingerprint sensors

Function / Service	**GetCommandEvent ( )**	
Input Parameter Name	**Description**	**Parameter Type**
UnitUserName*	Username for authentication	String
UnitPassword*	Password for access authentication	String
installationId	Installation Id for the installation	Integer
unitId	Unit ID for this unit	Integer
ClientVersion*	Client Version	String
dateTimeStamp	Date and time	Date Time
Return Value Name (CommandEvent)	**Description**	**Return Value Type**
commandEventId	Valid values for commandEventIds can be found in Annex G Example of Event constants.	Integer

statusCode	Status code 0 signals that the web service executed without errors.	Integer
subCommandEvents	SubCommandEvents is an array of SubCommandEvent. Each SubCommandEvent has two properties. i.e propertyID and propertyValue. This allows the BPM tool to send to the client a set of new properties or actions it should perform. Valid values for propertyIds can be found in Annex G Example of Event constants.	Array of objects
unitId	Unit ID for the unit that the CommandEvent is directed to	Integer
Comments	This web service is used for fetching command events, allowing the BPM tool to update properties at the client, given that this feature is supported by the client.	

** Optional vendor specific BPM parameters.*

Annex A (informative): Sample Q&A related to Reliability, Availability, Maintainability, and Performance in the context of BPM

With regards to Reliability, Availability, Maintainability, and Performance this section aims to formulate some sample questions to be asked by the Biometrics Operations management team and personnel.

A.1 Reliability-related questions

How can we make the best use of the vendor's claimed Mean-Time-Between-Failure (MTBF) about its unit/element (e.g. capture device)?
Measure unit/element's duration of operation; Set its scheduled (routine) maintenance before the claimed MTBF; Repair if broken and place back in operation. See Terms and Definitions for description of Mean-Time-Between-Failure (MTBF).

How can we make the best use of the vendor's claimed Mean-Time-To-Failure (MTTF) about its unit/element (e.g. capture device)?
Measure unit/element's duration of operation; Set its scheduled (routine) maintenance before the claimed MTTF; Replace with a new one. See Terms and Definitions for description of Mean-Time-To-Failure (MTTF).

How can we utilize by the vendor specified operational conditions (e.g. temperature, noise, etc) for a unit/element (e.g. capture device)?
Measure those conditions in real-time, Set by vendor specified conditions as baseline values, Compare real-time/measured values by baseline values. Trigger alert when a real-time/measured value exceeds its baseline value (threshold).

A.2 Availability-related questions

How do we set our Operational Availability requirement?
For how to set your target on Operational Availability of your biometric system/device/service see Terms and Definitions.

How do we know that our systems are up and running? how can we spot problems before our user do?
By applying real-time Availability monitoring (part of the Biometrics Performance Monitoring)

How early are we notified that a device/service/application is down or performing poorly?
By properly setting the frequency of real-time Availability monitoring. More frequent check will notify you earlier, e.g. one every minute compared with one every 30 minutes. Higher frequency of real-time Availability monitoring (i.e. earlier notifications) improve your targets on Mean-Time-to-Recovery (MTTR) and Maximum-Time-to-Recovery.

How often (with which frequency rate) should we perform real-time Availability monitoring of our biometric system/device/service?
See the above question. Other factors that may influence your decision are 1) cost of the real-time Availability monitoring; e.g. the higher frequency the higher price due to ISP/SaaS provider business model (e.g. bandwidth usage) 2) the possible effect (e.g. response time) that the real-time Availability monitoring has on the system/device/service it monitors (in most cases this will not be an issue as this side effect is non-existing/too small and therefor negligible).

A.3 Maintainability-related questions

How do we set our maintainability requirement?
For how to set your targets on maintainability of your biometric system/device/service see Mean-Time-to-Recovery (MTTR) and Maximum-Time-to-Recovery in Terms and Definitions.

How can we minimize the time to recovery?
When a problem occurs, how fast are we able to resolve it? By applying real-time Availability monitoring and properly setting its frequency so that time See Terms and Definitions

What is our Maintenance and Support requirement, strategy and budget?
See Maintenance strategy terms under Terms and Definitions

A.4 Performance-related questions

How do we know that our systems are performing as expected?
By applying real-time Availability and Performance monitoring (part of the Biometrics Performance Monitoring)

A.5 Operational (real-world) feedback-related questions

Our products have previously undergone technology and scenario testing with rock-solid and top-score results. How do we know how they actually perform in real-world cases?
By applying Biometrics Performance Monitoring.

Our R&D unit would make great use of consolidated performance and user experience statistics from our customers' sites. Where do we start?
By applying Biometrics Performance Monitoring.

A.6 Feasability of BPM program and tool

How do we know that our biometric installation is "Mission Critical"-enough to justify and motivate the need for applying a BPM program?
Basically, any biometric system/service that serves a need (even for one hour a day of operation time) whose operator needs to know if it is working properly or not should be considered as mission critical.

Annex B (informative): Example of metrics in a Service Level Agreement

Here are some examples of metrics in an SLA.

Reliability metrics
Mean-Time-Between-Failure (MTBF)
Mean-Time-To-Failure (MTTF)
Availability metrics
Operational availability (Uptime)
Maintainability metrics
Mean-Time-to-Recovery (MTTR)
Maximum-Time-to-Recovery
Performance metrics
Maximum verification/identification response time
Maximum rate of rejections
Maximum rate of Failure-to-Acquire
Maximum rate of Failure-to-Enroll

Annex C (informative): Sample Service Level Agreement

Authors' note: This is a place holder for a sample Service Level Agreement

Annex D (informative): Sample of symptoms and their possible causes

Authors' note: This is a place holder for Sample of symptoms and their possible root causes.

Symptom	Possible root cause
Reliability-related	
Availability-related	
Maintainability-related	
Performance-related	

Annex E (informative): Case studies

Use case study, one: BPM in an enrollment application

Case study snapshot

Project: Biometric Identification on the Move (BIMS)
Project leader: AFIS and Biometrics Consulting Inc. (Newport Beach, CA)
Project Sponsor: Department of Homeland Security (DHS)
Sponsor of Biometrics Performance Monitoring tool and know-how: Optimum Biometric Labs with BioUptime

Type of application: Enrollment module for Automated Border Control (ABC)

Units/elements monitored:

- GreenBit optical fingerprint sensor (DactyScan84)
- BIMS Enrollment Software
- IriTech IriTerminal MD-300 Iris camera
- Canon EOS 50D Digital SLR camera 15.1MP
- Symbol LS1203 General Purpose Bar Code Scanner

Frequency of monitoring: between 10 seconds to 600 seconds, based on individual units/elements

Total no. of device specific Event types: more than 108

The Biometrics Performance Monitoring, BPM, tool BioUptime has added value in these areas:

Consolidating/displaying primary site information on three different levels:

1. Organization information:
    - organization name, type, address, contact persons, alert setting, list of installations, installation type, alert setting for individual installations
2. Installation information:
    - installation name, installation type, address, location, time zone, user who created it, and the date it was created, alert setting, alert subscribers, list of units/elements and their types, alert setting for individual units/elements
3. Unit information:
    - Consisting of three main sections: Unit Information, Alert Setting, and Latest Event
        i. Unit Information: The Unit Information contains the useful details: Unit name, Unit type, Location, Installation and organization it belongs to, Comment, Serial number, and frequency of monitoring (poll interval)
        ii. Alert Setting and alert subscribers
        iii. Unit's last event and its corresponding data field and time stamp

Collecting and archiving more than 108 events and transactions for analysis of trends, behaviour and abnormalities.

Computing and presenting performance-related variables (metrics and stats):

- Number of enrollees
- Total background database
- Total number of samples
- Transaction duration
- Comparison score
- Image quality score
- Failure to enroll rate (FTE)
- Failure to acquire rate (FTA)

Remote performance and availability (health status) monitoring on the following levels:

- Capture device
- Application/service components:
  - Database Service
  - Enrollment
  - Template generating app
  - Data management service
- System components such as:
  - Enrollment server
  - Database server

Displaying and sending (via email) alerts and diagnosis information

Accessing the above features via a web-based UI

Annex F (informative): How to evaluate IT tools for Biometrics Performance Monitoring?

Property	Description (example)	Checklist
Features		
Event logging/consolidation/ grouping	e.g. organizing events based on units/elements, sites/ installations, business processes/units, departments/ organizations/geographic regions, global (bird's eye view), or any variant of user-defined filtering	
Event/alert analytics	capable of detecting and diagnosing critical events (alerts) and root-causes	
Event/alert user interface (UI)	presenting events and instant notifications via web/desktop interface (dashboard, console), mobile/tablet app, email, etc.	
Reporting	e.g. using visualization/analytical tools to generate automatic or on-demand reports in diverse formats e.g. graphs/numerical values on e.g. SLA metrics or other properties and distributions	
Corrective actions	e.g. capable of setting/changing a parameter or triggering an action in the monitored unit/element	
Inventory/record of biometric installations	keeping lists of geographically spread biometric sites and installations with detailed information about them for reasons such as archiving or remote troubleshooting (detail record such as address, contact list for each organization, installation type and location, time zone, unit/element name, type, serial number, comments, etc.)	
Drill-down navigation	capable of drilling-down to an individual unit/element view (the lowest level of hierarchy) from the bird's eye view of all sites and installations	
Configuration/customization		
User-defined events	capable to define, add, and modify new type of events	
User-defined alerts	capable to define, add, and modify new type of alert rules and logic	

Event update frequency	capable of setting/changing how often units/elements report their events to the BPM tool (e.g. every five minutes, etc.)	
Event update frequency	capable of setting/changing how often units/elements report their events to the BPM tool (e.g. every five minutes, etc.)	
User Interface	support for developing the UI, or parts of it, for different Eco-systems and platforms (e.g. iOS, Android, etc.)	
Site administration	adaptable to your requirements such as creating and adding: New organizations, installations, and units, New type of installations and units	
Flexibility and compatibility		
Biometric modality/type	Biometric agnostic/independent or can only work with certain biometric types	
Biometric sample acquisition sensor type	Sensor-agnostic/independent or can only work with certain capture devices	
Unit/element type	ability to monitor any user defined unit/element	
Vendor independent	capable of working with hardware and software from any vendor (vendor agnostic/independent) or can only work with certain vendors HW & SW (e.g. if biometric or sensor agnostic then most likely vendor independent too)	
Scalability and Performance		
Multi-unit/element management	e.g. capable of monitoring # of units/elements with acceptable performance (based on a certain BPM server hardware configuration)	
Multi-site management	e.g. support of unlimited # of user-defined sites/ installations, business processes/units, departments/ organizations/geographic regions	
BPM tool users	e.g. support of unlimited # of users	
BPM tool user management		
User Roles and Access Management	support different types of user roles with different access rights (e.g. super admin, admin, operator, etc.)	
Security		
Data encryption	Encrypted data transfer (e.g. SSL) for every event transmission between a monitored unit/element (or its agent) and the BPM Server	
Agent-BPM server authentication	Authentication for every event transmission between	

	a monitored unit/element (or its agent) and the BPM Server	
Technical requirement for BPM tool's Server		
	Minimum HW requirements: Type of processor, available disk space and RAM, network interface	
Supported OS	e.g. Windows, Mac OS X, Linux	
Technical requirement for BPM tool front-end	e.g. Internet connection, web browser (e.g. Firefox, IE, Safari, Chrome, etc.), etc.	
User/customer support		
	Guideline and tutorials such as UI manual, installation manual, troubleshooting help, white papers, etc.	
	Online support (e.g. web portal)	
	Telephone support	
Integration support		
Agent/client interface/mechanism	has its own agent description/method and/or is agent-less (e.g. SNMP) for event collection	
API technology	e.g. open API for developing customized agents. e.g. based on Service Oriented Architecture (SOA), Web services-based platform (WSDL interface), SOAP for integration with biometric-based systems, rapid and easy to integrate with, etc.	
API documentation and sample code		
Business/delivery model		
Open source		
Proprietary		
Software-as-a-Service (SaaS), Cloud-based		
Appliance (software bundled and shipped with dedicated hardware)		
Pricing		
Attractive price point		

Flexible licensing policy	e.g. license based on # of units/elements	
Updates, maintenance, and support		
Amount of necessary/needed related professional services	e.g.business case analysis, planning, installation, integration, testing, optimization, etc.	
Total Cost of Ownership (TCO) analysis		
Available Return on Investment (ROI) analysis and calculator for	e.g. benchmarking cost savings against Total Cost of Ownership (TCO)	
Documented customer case study	tried-out and used by verifiable customers	
Standards	Compliant with international, national, and industry standards	
Independent test/audit	evaluated by independent test organizations for e.g. scalability, performance, usability, interoperability, reliability, security and vulnerability, etc.	

Annex G (normative): BPM Application Programming Interface (API)

G.1 List of Status Codes

Status Code	Description
STATUSCODE_OK = 0	OK, i.e no errors
STATUSCODE_INVALID_UNIT_USERNAME = -1010	Unit Username was not in the range of min-max lengths
STATUSCODE_INVALID_UNIT_PASSWORD = -1020	Unit Password was not in the range of min-max lengths
STATUSCODE_INVALID_INSTALLATION_ID = -1030	Installation Id was not in the range of min-max Values
STATUSCODE_INVALID_UNIT_TYPE_ID = -1040	Unit type Id was not in the range of min-max values
STATUSCODE_INVALID_UNIT_ID = -1050	Unit Id was not in the range of min-max values
STATUSCODE_INVALID_PARENT_UNIT_ID = -1060	Parent Unit Id was not in the range of min-max values
STATUSCODE_INVALID_SYSTEM_EVENT_TIME_INTE RVAL = -1070	Time intervals was not in the range of min-max values
STATUSCODE_INVALID_COMMAND_EVENT_ID = -1080	Command Event Id was not in the range of min-max values
STATUSCODE_INVALID_UNIT_VERSION = -1090	Unit Version was not in the range of min-max values
STATUSCODE_INVALID_SIMPLE_SUB_EVENT_ARRAY _LENGTH =-2000	Simple Sub Event Array Length was less than 1
STATUSCODE_INVALID_DESCRIPTION = -2010	Description was not in the range of min-max values
STATUSCODE_INVALID_LOCATION = -2020	Location was either null, or not in the range min-max values
STATUSCODE_INVALID_COMMENT = -2030	Comment was either null, or not in the range of min-max.
STATUSCODE_INVALID_SIMPLE_SUB_EVENTS_DATA _PROPERTY =-2040	Simple Sub Event Data Property was either null or not in the range of min-max lengths

STATUSCODE_INVALID_SIMPLE_SUB_EVENTS_ID_PR OPERTY = -2050	Simple Sub Event Id Property was not in the range of min-max values
STATUSCODE_INVALID_SIMPLE_SUB_EVENTS_NULL = -2060	Simple Sub Event-array was null which is forbidden
STATUSCODE_INVALID_USER_IDENTIFIER = -2070	User identifier was either null or not in the range of min-max lengths
STATUSCODE_INVALID_APPLICATION = -2080	Application was either null or not in the range of min-max lengths
STATUSCODE_INVALID_EXECUTION_STATUSCODE = -2090	Execution Status Code was not in the proper range i.e. 0 or less than 0
STATUSCODE_INVALID_DATETIMESTAMP_NULL = -3000	DateTimeStamp was null

G.2 Example of Event Constants

The Event Constants in the table below are examples from Optimum Biometrics Labs. Any vendor or organization can define and set their own Event Constants. In fact, software and hardware vendors often define their own set of Events specific for their products which are usually included in their API documentations.

Event Constant Name	Constant Value	Description
UNDEFINED	-1	Undefined event
UNDEFINED2	0	Undefined event
INSTALLATION	100	Installation Event
SYSTEMEVENT_BEGIN	200	System event
SERVICESTARTED	201	Service started
SERVICESTOPPED	202	Service stopped
AGENTON	203	Agent On
HEARTBEAT	210	Periodic Heart Beat
HEARTBEAT_OK	211	Periodic Heart Beat ok
HEARTBEAT_BIOAPI_NOK	212	Periodic Heartbeat not ok, unable to initialize BioAPI
HEARTBEAT_BSP_NOK	213	Periodic Heartbeat not ok, Unable to load BSP
HEARTBEAT_UNIT_NOK	214	Periodic Heartbeat not ok, Unable to attach unit
Capture Device related events		
SENSORHEARTBEATOK	220	Sensor Heartbeat ok

SENSORHEARTBEATNOK	221	Sensor heart beat not ok
SENSORRESPONDING	223	Sensor Responding
SENSORNOTRESPONDING	224	Sensor not responding
SENSORSTATUSOK	225	Sensor status ok
SENSORSTATUSNOK	226	Sensor status not ok
OPCHEARTBEATOK	230	OPC heart beat ok
OPCHEARTBEATNOK	231	OPC heart beat not ok
Service/Application related events		
SERVICERESPONDING	243	Service Responding
SERVICENOTRESPONDING	244	Service not responding
SERVICESTATUSOK	245	Service status ok
SERVICESTATUSNOK	246	Service status not ok
APPLICATIONRESPONDING	253	Application responding
APPLICATIONNOTRESPONDING	254	Application not responding
APPLICATIONSTATUSOK	255	Application status ok
APPLICATIONSTATUSNOK	256	Application status not ok
SYSTEMRESPONDING	263	System Responding
Website related events		
WEBSITERESPONDING	273	Website responding
WEBSITENOTRESPONDING	274	Website not responding
WEBSITESTATUSOK	275	Website status ok
WEBSITESTATUSNOK	276	Website status not ok
Project specific user events		
USEREVENT_BEGIN	1000	Not in use
FACEQUALITY_ASSESSED	2001	Not in use
TRANSACTIONTIME	5010	Transaction time in (MS)
FingerPrintingTime	5011	Fingerprint Capturing Time
FaceCapturingTime	5012	Face Capturing Time
IrisCapturingTime	5013	Iris Capturing Time
AVAILABLEHARDDRIVESPACE	5101	Available hard drive space
CompleteFingerprintEnrollment	6010	All 10 Flats, 10 rolls and 4 palms captured
CompleteFaceEnrollment	6011	All 9 Faces Captured

CompleteIrisEnrollment	6012	Both Eyes Captured
InCompleteFingerprintEnrollment	6110	Not all 10 Flats, 10 rolls and 4 palms captured
InCompleteFaceEnrollment	6111	Not all 9 Faces Captured
InCompleteIrisEnrollment	6112	Not all irises Captured
FingerprintAverageQuality	7001	Fingerprint captures average quality
FaceAverageQuality	7002	Face captures average quality
IrisAverageQuality	7003	Iris captures average quality
USEREVENT_END	19999	Not in use
COMMANDEVENT_BEGIN	20000	Not in use
COMMANDEVENT_SYSTEMEVENTTIM EINTERVALL	20001	Change systemeventtimeintervall (in ms)
COMMANDEVENT_ACKNOWLEDGES UCCESS	29000	Not in use
COMMANDEVENT_ACKNOWLEDGEF AILURE	29001	Not in use
COMMANDEVENT_END	30000	Not in use

Useful resources

Optimum Biometric Labs, White paper: Reliability, Availability and Maintainability (RAM) in Biometric Applications – Delivering Quality of Service that customer wants (pdf)

TSA, Airport security, The Transportation Security Administration (TSA), “GUIDANCE PACKAGE, Biometrics for Airport Access Control”. (*ref.* 3.2 OPERATIONAL AVAILABILITY). http://www.tsa.gov/assets/pdf

The Biometric Web Services project: http://www.nist.gov/itl/iad/ig/bws.cfm

Information Technology Infrastructure Library (ITIL): http://en.wikipedia.org/wiki/ITIL#Service_Level_Management

The Construction Property Services Industry Skills Council in Australia (CPSISC), “Monitor biometrics equipment/ systems” (*ref.* ELEMENT, PERFORMANCE CRITERIA) http://www.cpsisc.com.au/projects/Biometrics%20Project/

List of Biometric metrics, www.whatmetric.com

Intelligent device management: http://en.wikipedia.org/wiki/Intelligent_device_management

Telematics: http://en.wikipedia.org/wiki/Telematics

Gartner Magic Quadrant for Application Performance Monitoring (APM), 18 February 2010

Gartner Magic Quadrant for Event Correlation and Analysis (ECA), 13 December 2010

Gartner Magic Quadrant for Security Information and Event Management (SIEM), 12 May 2011

Revision History

Version 1.0

More relevant standards and guidelines
Identify more relevant details in other standards (to also include international guidelines such as ICAO etc.)

A table (break-down) of biometric subsystem components
Biometric subsystem components defined as distinctive Units/Elements along with their corresponding events and metrics

A table of applicability of performance metrics in Biometrics Performance Monitoring, BPM
Relative to Technology, Scenario, Operational Testing

Expansion of the annex on symptoms and possible root causes
Possibly also add a corrective action column

Extension of Case Studies
Include more relevant examples and reference cases