**Response to “CBP CEE Biometric Excellence Review of the OASIS WS-Biometric API Draft 01”**

Ross J. Micheals & Kevin Mangold, Co-Editors, WS-Biometric Devices 1.0

On behalf of the TC, the co-editors thank CBP for their comments on WS-Biometric Devices 1.0 Committee Draft 01.

1. The WS-BD specification was developed in concert with an implementation of that specification to ensure that effective and efficient biometric sensor command and control was possible. In the development of the specification, NIST (the original developers of the specification) did not experience any unacceptable latency attributable specifically to Wi-Fi. In addition, the specification can be used anywhere web services can operate; the quality of service for the underlying networking is beyond the scope of the standard.
2. Service lifetime is outside the scope of the specification. This is because of the design principle (as stated in the Service Lifecycle Behavior section) that “The lifecycle of a service (i.e., when the service starts responding to requests, stops, or is otherwise unavailable) must be modeled after an integrated implementation.” If a WS-BD service is running on a dedicate device, then that device has the sovereignty to decide when the WS-BD service starts and stops; perhaps it is via a button press, perhaps it is always running as long as the device is powered on. The following was added to “Service Lifecycle Behavior” to address this concern.   
     
   “Consequently, this specification does NOT make any specific recommendations on how a WS-BD service should be started, stopped, or reset. This (a) reflects the connectionless nature of HTTP but also (b) allows the host environment maximum flexibility on how to implement service availability. For example, a manufacturer of an embedded device might elect to have the device run a service as long as the device is powered on.”
3. See (1).
4. It is the responsibility of the server to clean up the stored biometric data. See Appendix B, Storage Parameters for information on how a service advertises some of its storage capabilities.
5. The specification does indeed allow for different parameters and threshold ranges. Different services are permitted to define these ranges differently. The conformance section describes a framework in which “normalized” parameters might be used to ensure a level of known interoperability among specific subsets of WS-BD capable devices (e.g., fingerprint scanners). Additional conformance profiles are potential future work items for consideration by the TC.
6. Text has been added to describe how to initiate and stop live preview.
7. Note that lock operations are not sensor operations. Lock requests are not queued, nor do they need to be resubmitted. In addition, in CSD02 we added clarifying language to “Locking: Overview and General Behavior”
8. NIST offers two public domain reference implementations of the server—one is Java and one in .NET. NIST also offers a public domain reference application compatible with those servers called *wsabi*. More information on the reference implementations can be found at <http://bws.nist.gov>
9. See (8). In addition, these reference implementations includes a variety of unit tests that cover a variety of failure cases. Clients would have to handle the situation in which a service is not available or unexpectedly ends in the same manner that applies to any HTTP request.
10. The following was added to “Locking : Overview and General Behavior.”   
      
    “WS-BD only offers the core locking, unlocking, and lock stealing operations. Any other lock coordination is outside of scope of this specification and is the clients’ responsibility.”
11. See (6). The decision on what to show (or not show) in the live preview is left completely to the service.
12. The specification contains a variety of diagrams that show data flow; for example, Figures 7, 8, 9, 10, 11, 12, and 13 all show how data and commands flow between a client and a service.
13. As stated in “Design Concepts and Architecture” WS-BD is “is influenced heavily by the REST architecture; deviations and tradeoffs were made to accommodate the inherent mismatches between the REST design goals and the limitations of devices that are (typically) oriented for a single-user.” That is, strict adherence to REST is not claimed. The comment is correct that some of the endpoints are more “operations” than they are “resources”. Users of the specification are encouraged to use creativity when assigning semantic meanings to particular uses of URL & verb pairs.
14. See (13).
15. The section HTTP Request-Response usage discusses the use of HTTP status codes.  
      
    In addition, the phrase “If a service successfully receives a well-formed request, then the service mustreturn the HTTP If a service successfully receives a well-formed request, then the service mustreturn the HTTP status code 200 indicating such.” was corrected to “status code 200–299 indicating such”
16. JSON may be considered for future versions of the specification
17. Base64 encoding is allows the biometric data to be embedded uniformly within any metadata that describes it within the same XML ‘document’. In future versions of the specification, there may be an option to replace the base64 encoded biometric data with a link that would point directly to the target biometric data.
18. “A service that controls multiple biometric devices simultaneously (e.g., an array of cameras with synchronized capture) should be exposed via the same endpoint.” has been changed to “A service that controls multiple biometric devices simultaneously (e.g., an array of cameras with synchronized capture) should be exposed via the same endpoint; this SHOULD NOT be the preferred architecture if the sensors would need to be addressed or controlled separately. “
19. Example 2 was changed to: “Figure 5 shows a physically separate implementation where a single host machine controls a pair of cameras used for stereo vision. The cameras act together as a single logical sensor and are both exposed via the same service, http://wsbd/camera\_array. The left and right camera are not individually addressable because the service is exposing both by a single endpoint. If the left and right camera needed to be separately addressable, then the host should expose two services—http://wsbd/left\_camera and http://wsbd/right\_camera—one for each camera.”
20. See (4).
21. The specification does not support any ‘in band’ way of determining what client holds the lock and for how long; but may be considered in future versions of the specification.
22. An LED status light might be controlled by a ‘set configuration’ operation, or (more likely) through a custom command. Where appropriate, language was added to the speciation to better accommodate this kind of use of ‘set configuration’.
23. See (2).
24. There is no particular recommendation for how a client should handle not getting a response back. However, as suggested by the comment, a timeout might be an appropriate way to handle this.
25. See (2).
26. See (2).
27. See (2).