CLAIMS

What is claimed is:

1. A system for supervising usage of software comprising:

a software vendor producing instances of software;

a tag server producing a plurality of tags, one tag per instance of software, each tag uniquely identifying an instance of software with which it is associated; and

a user device receiving and installing an instance of software and securely receiving a tag uniquely associated with that instance of software, the user device including a supervising program which detects attempts to use the instance of software and which verifies the authenticity of the tag associated with the instance of software before allowing use of the instance of software.

2. The system of claim 1, wherein the supervising program on the user device verifies the authenticity of the tag and maintains the tag in a tag table and maintains the instance of software if the tag is authentic and rejects the instance of software if the tag associated with the software is not authentic.

3. The system of claim 2, wherein the supervising program verifies a hash function value in the tag to determine if the tag is authentic and is properly associated with the instance of software.

4. The system of claim 2 wherein the tag is digitally signed and the supervising program verifies the authenticity of the tag by verifying a digital signature of the tag.

5. The system of claim 1 wherein each of the plurality of tags created by the tag server comprises at least one of a name of software, a unique number of an

instance of software and a hash function value on portions of an instance of software.

6. The system of claim 5 wherein the unique number of the instance of software is selected from a sparse set of numbers.

7. The system of claim 5 wherein each tag further comprises a unique identifier of the supervising program.

8. The system of claim 7 wherein the supervising program verifies that the unique identifier of the supervising program in a tag is the same as an identifier of the supervising program on the user device.

9. The system of claim 1 wherein each tag includes at least one fingerprint computed on portions of the instance of software associated with the tag.

10. The system of claim 9 wherein the supervising program verifies that the software instance associated with a tag satisfies a same-location fingerprint check against the at least one fingerprint included in the tag associated with the instance of software.

11. The system of claim 10 wherein the same-location fingerprint check is performed by the supervising program at at least one time of before, during, and after use of the instance of software.

12. The system of claim 9 wherein each tag further includes at least one list of locations containing values from which the at least one fingerprint is computed and the supervising program verifies that the software instance associated with each tag satisfies a same-location fingerprint check against the at least one fingerprint associated with the software at locations specified in the at least one list of locations.

13. The system of claim 1 wherein whenever any data file is accessed by the instance of software, information associated with the instance of software performing the access is stored in a location associated with the data file.

14. The system of claim 13 wherein the information associated with the instance of software is the tag associated with the instance of software.

15. The system of claim 13 wherein the information associated with the instance of software is the time of modification performed by the instance of software .

16. The system of claim 13 wherein the information associated with the instance of software performing the access is written to a secure location which the supervising program alone can access.

17. The system of claim 16 wherein the supervising program verifies that when an instance of the software attempts to access a data file having associated information stored in the location associated with that data file, the supervising program verifies that the associated information stored is information associated with the instance of software currently attempting access.

18. The system of claim 16, wherein the supervising program uses an unaliasable hash function to verify the associated information stored in the location associated with the data file for which access is currently being attempted.

19. The system of claim 1 further comprising:

a guardian center including:

a tagged software database; and

a verification program;

the guardian center periodically communicating with the user device via a call-up procedure to receive tags from the user device, said tags associated with instances of tagged software used on the user device, the verification program examining each tag received from the user device against the tagged software database to ensure that the tags are in compliance with at least one usage supervision policy, and the verification program returning a continuation message to the user device, the continuation message indicating for the instance of software associated with each tag on the user device an action to follow; and

the supervising program on the user device verifying the continuation message for authenticity and if authentic, performing the action to follow indicated in the continuation message.

20. The system of claim 19 wherein at least one of the software vendor, the tag server, and the guardian center are combined with another of the at least one of the software vendor, the tag server and the guardian center.

21. The system of claim 19 wherein the maximum allowed time interval between successive call-up procedures is determined by at least one of a combination of the time elapsed in the user device, a number and duration of uses of instances of software, a number of times the user device is powered on, and a measure of use of the user device.

22. The system of claim 21 wherein when a user device fails to perform a call-up procedure with the guardian center before the end of a maximum allowed interval since the last call-up procedure, the user device is disabled for a period of time.

23. The system of claim 21 wherein when a user device fails to perform a call-up procedure with the guardian center before the end of a maximum allowed

interval since the last call-up procedure, usage of certain instances of software is denied for a period of time.

24. The system of claim 19 wherein a call-up occurs when an instance of software is used a first time on a user device.

25. The system of claim 19 wherein a call-up occurs due to a request by the guardian center.

26. The system of claim 19 wherein the supervising program tests an authenticity of the continuation message by verifying that a hash function value of a tag table in the continuation message is the same as a hash function value of a tag table sent in a call-up message from the user device.

27. The system of claim 26 wherein the authenticity of the continuation message is tested by the supervising program by verifying that a digital signature in the continuation message is produced by the guardian center.

28. The system of claim 19 wherein a user device that receives no continuation message following a call-up message to the guardian center resends a call-up message with a cancellation command for a previous call-up message.

29. The system of claim 19 wherein at least one usage supervision policy is associated with at least one individual instance of software with which at least one tag is associated.

30. The system of claim 19 wherein at least one usage supervision policy is associated with the entire user device with which the guardian center communicates during the call-up procedure

31. The system of claim 19 wherein at least one usage supervision policy is associated with an individual user of the user device with which the guardian center communicates during the call-up procedure.

32. The system of claim 19 wherein at least one usage supervision policy is associated with a usage supervision history of the user device with which the guardian center communicates during the call-up procedure.

33. The system of claim 19 wherein the guardian center maintains a tag data structure in the tagged software database for each tag associated with each instance of software on each user device.

34. The system of claim 33 wherein each tag data structure includes a tag of an instance of software, a usage supervision policy associated with the instance of software, and a collection of references to call-up records.

35. The system of claim 34 wherein each call-up record in the collection of call-up records represents information concerning one call-up procedure and the continuation message associated with the call-up procedure includes at least one of a call-up time, a header of a tag table transferred to the guardian center during the call-up procedure, a last call-up time indicating a time stamp of a former call-up procedure, a hash function value of the tag table transferred to the guardian center during the call-up procedure, and actions to follow on the user device.

36. The system of claim 1 further comprising:

a guardian center including a verification program;

the guardian center periodically communicating with the user device via a call-up procedure to receive a unique identifier for the user device's supervising program from the user device, the verification program examining the unique identifier to ensure that at most one supervising

program has that identifier, and the verification program returning a continuation message to the user device, the continuation message indicating an action to follow upon attempted use of the instances of software associated with each tag on the user device,

the user device's supervising program verifying the continuation message for authenticity and if authentic, performing the action in the continuation message.

37. The system of claim 36 wherein the supervising program identifier is generated a first time that the supervising program is invoked, based on a rarely duplicated number.

38. The system of claim 37 wherein the rarely duplicated number is a very precise clock value occurring when the supervising program is first invoked in the machine.

39. The system of claim 37 wherein the rarely duplicated event is a number provided by a guardian center.

40. The system of claim 1 further comprising:,

an untagged instance of software used on the user device;

wherein the supervising program detects the use of the untagged instance of software and performs a fingerprinting process on the untagged instance of software and stores fingerprints resulting from the fingerprinting process on the user device.

41. The system of claim 40 where the user device's supervising program further performs a fingerprinting process on a tagged instance of software used on the device and stores the fingerprints resulting from the fingerprinting process in a fingerprint table on the user device.

42. The system of claim 41 wherein the supervising program stores locations from which the fingerprints are computed.

43. The system of claim 41 wherein the fingerprints are based on contents of the instance of software.

44. The system of claim 41 wherein the fingerprints are based on known sequences of behavior of the instance of software.

45. The system of claim 41 further comprising:

a guardian center including:

a fingerprint data structure; and

a verification program;

the guardian center periodically communicating with the user device via a call-up procedure to receive all fingerprints from the user device for an instance of software used on the user device, the verification program comparing every fingerprint received from the user device against the fingerprint data structure to determine if an instance of software used on the user device is an infringing instance of software.

46. The system in claim 45 wherein if the verification program detects more than a specified number of matches between fingerprints in the guardian center's fingerprint data structure and fingerprints received from the user device, the verification program specifies a punitive action to be performed, and the verification program returns a continuation message to the user device, the continuation message indicating the punitive action to be performed on the user device.

47. The system in claim 46 wherein the fingerprint matching process is at least one of general location or same location fingerprint matching.

48. The system in claim 46 wherein the fingerprint matching uses an inverted guardian center fingerprint table.

49. The system of claim 46 wherein the punitive action specifies that the user device be disabled for a specified length of time.

50. The system of claim 46 wherein the punitive action specifies that the instance of software associated with the fingerprint that was matched to a fingerprint in the fingerprint data structure of the guardian center should be disabled for a specified length of time.

51. The system of claim 46 wherein the punitive action depends on at least one of a combination of the history of the behavior of the user device, the history of the behavior of a particular user on the user device, and the collection of other software on the user device.

52. The system of claim 45 wherein the software vendor transmits a copy of an infringing instance of software to the guardian center and the guardian center computes fingerprints on the copy of the infringing instance of software and incorporates and stores the fingerprints into the fingerprint data structure on the guardian center.

53. A tag table data structure encoded on a user device's readable medium, the tag table data structure including at least one tag that is uniquely associated with one instance of software and including at least one field associated with the tag in the tag table, and including at least one field indicating a usage status associated with the tag associated with the instance of software.

54. The tag table data structure of claim 53 where the at least one field indicates use statistics for the one instance of software associated with the tag.

55. The tag table data structure of claim 53, further including a tag table header that uniquely identifies the tag table.

56. The tag table data structure of claim 53 wherein the tag table header includes information concerning user device use statistics and includes a continuation message.

57. A software vendor comprising:

a software production mechanism creating instances of software each having at least one of a name and software content;

each instance of software being usable only in conjunction with a tag that is unique to that instance of software, the tag being a unique unforgeable collection of information concerning the instance of software with which the tag is associated and including at least one of the name of the software, a unique number of the instance of software and a hash function value on portions of content of the software

58. The software vendor of claim 57 wherein the tag includes an identifier of the supervising program associated with a user device upon which the instance of software is to be used.

59. The software vendor of claim 57 wherein the tag includes a list of fingerprints of portions of the instance the software with which the tag is associated.

60. The software vendor of claim 57, further comprising:

an infringing software detection mechanism detecting software that is infringing on the vendor's rights and transferring a copy of the infringing software to a guardian center so that usage supervision can be implemented to detect attempted use of an instance of the infringing software on a user device.

61. The software vendor of claim 60, further comprising:

an infringing software detection mechanism detecting software that is infringing on the vendor's rights and transferring a copy of the infringing software to a guardian center, the guardian center invalidating any tag associated with an instance of the infringing software and sending a punitive action to any user device detected by the guardian center to have used the instance of infringing software.

62. A user device comprising:

an input port receiving an instance of software and receiving a tag uniquely associated with that instance of software and receiving a request to use the instance of software;

a processor executing a supervising program, the supervising program detecting the request to use the instance of software and verifying the authenticity of the tag associated with the instance of software before allowing use of the instance of software by the user device.

63. The user device of claim 62, wherein the supervising program verifies the authenticity of the tag and stores the tag in a tag table and maintains the instance of software if the tag is authentic and rejects the instance of software if the tag associated with the software is not authentic.

64. The user device of claim 63, wherein the supervising program computes a hash function value on the instance of software and compares the computed value with a hash function value in the tag to determine whether the tag is authentic and is properly associated with the instance of software.

65. The user device of claim 63 wherein the tag is digitally signed and the supervising program verifies the authenticity of tag by verifying a digital signature of the tag.

66. The user device of claim 63, wherein the tag table is a data structure stored in storage on the user device and contains at least one tag that is uniquely associated with an instance of software and includes at least one field associated with the tag in the tag table, the at least one field indicating a usage status for the instance of software associated with the tag.

67. The user device of claim 62 wherein the supervising program determines that a call-up procedure is required as defined by a call-up policy and the supervising program performs the call-up procedure to update the usage status of tags stored in the tag table.

68. The user device of claim 62 wherein the supervising program verifies that each data file used by tagged software is produced by a legitimate instance of software.

69. The user device of claim 67 wherein during performance of the call-up procedure, the supervising program securely transmits the tag table from the user device via an interconnection mechanism coupled to the user device and awaits reception of a continuation message returned to the user device, the continuation message indicating actions to be performed for each tag in the tag table.

70. The user device of claim 67, wherein during the performance of the call-up procedure, the supervising program securely transmits a tag table header from the user device via an interconnection mechanism coupled to the user device and awaits reception of a continuation message returned to the user device that indicates an action to be performed for each tag in the tag table.

71. The user device of claim 62 further comprising:

an untagged instance of software used on the user device;

wherein the supervising program detects the untagged instance of software and performs a fingerprinting process on the untagged instance of software and stores fingerprints resulting from the fingerprinting process in a fingerprint table on the user device.

72. The user device of claim 71 wherein the supervising program determines that a call-up procedure is required as defined by a call-up policy and the supervising program performs the call-up procedure to update the usage status of untagged instances of software stored on the user device.

73. The user device of claim 72, wherein during performing the call-up procedure, the supervising program transmits a portion of the fingerprint table from the user device via an interconnection mechanism coupled to the user device and awaits reception of a continuation message returned to the user device that indicates actions to be performed for each untagged instance of software stored on the user device.

74. A guardian center comprising:

a tagged software database; and

a verification program executing on a processor in the guardian center;

the guardian center periodically executing a call-up procedure to receive, via an interconnection mechanism, tags for instances of software, the verification program examining each tag received against the tagged software database maintained on the guardian center to ensure that the tags are in compliance with at least one usage supervision policy, and the verification program transmitting a continuation message via the interconnection mechanism indicating actions to follow upon attempted use of the instances of software associated with each tag received by the guardian center during the call-up procedure.

75. The guardian center of claim 74 wherein at least one usage supervision policy is associated with each instance of software with which at least one tag is associated.

76. The guardian center of claim 74 wherein at least one usage supervision policy is associated with a user device with which the guardian center communicates to receive tags.

77. The guardian center of claim 74 wherein at least one usage supervision policy is associated with an individual user of the user device with which the guardian center communicates to receive tags.

78. The guardian center of claim 74 wherein the guardian center maintains a tag data structure in the tagged software database for each tag associated with each instance of software on each user device and receives newly created tags associated with instances of software from a tag server and further receives tags associated with instances of software used on a user device in a tag table transmitted from the user device.

79. The guardian center of claim 78 wherein each tag data structure includes at least one of a tag of an instance of software, a name of the instance of software, a unique number of the instance of software, a hash function value on the instance of software, a usage supervision policy associated with the instance of software, and a collection of references to call-up records associated with the tag associated with the said instance of software.

80. The guardian center of claim 79, wherein each call-up record in the collection of call-up records represents information concerning one call-up procedure and includes at least one of a call-up time, a header of a tag table transferred to the guardian center during the call-up procedure, a last call-up time indicating a time stamp of a former call-up procedure, a hash function

value of the tag table transferred to the guardian center during the call-up procedure, and the action to follow on the user device contained in the continuation message associated with the call-up procedure.

81. A guardian center including:

a fingerprint data structure; and

a processor executing a verification program;

the verification program periodically executing a call-up procedure with a user device to receive, via an interconnection mechanism, fingerprints for instances of software used on the user device, the verification program examining each fingerprint received against the fingerprint data structure to determine if an untagged instance of software used on a user device is an infringing instance of software, and if so, the verification program preparing a punitive action to be executed on the user device.

82. The guardian center of claim 81 wherein all vendor software is fingerprinted and infringements of one vendor's software upon another vendor's software are detected based on at least one of same location or general location fingerprint checking.

83. The guardian center in claim 81 wherein if the verification program detects a sufficient number of matches between a fingerprint in the fingerprint data structure and a fingerprint within the fingerprints received, the verification program specifies punitive action to be performed, and the verification program transmits a continuation message, the continuation message indicating a punitive action to be performed on a receiver of the continuation message.

84. The guardian center of claim 83 wherein the sufficient number is one.

85. The guardian center of claim 83 wherein the sufficient number is greater than one.

86. The guardian center of claim 85 wherein the sufficient number is computed as a weighted sum of matches where the weight of each match depends on a fingerprint that matches.

87. The guardian center in claim 83 wherein the fingerprint matching technique is general location fingerprint checking.

88. The guardian center of claim 83 wherein the punitive action specifies disablement of the receiver.

89. The guardian center of claim 83 wherein the punitive action specifies that the instance of software associated with the fingerprint that was matched to a fingerprint in the fingerprint data structure should be disabled.

90. The guardian center of claim 81 wherein the verification program receives, via the interconnection mechanism, a copy of an infringing instance of software and computes fingerprints on the copy of the untagged infringing instance of software and incorporates and stores the fingerprints in the fingerprint data structure.

91. A tag server accepting a copy of specific vendor software and producing a plurality of tags, one tag per instance of the software, each tag uniquely identifying an instance of software with which it is associated, and each tag comprising at least one of the name of the software associated with the tag, a unique number of the instance of software associated with the tag, and hash function values computed on portions of the instance of software associated with the tag.

92. The tag server of claim 91, further including a digital signature mechanism used to digitally sign the tags and to securely transmit the tags to an intended receiver.

93. A method for supervising usage of software comprising the steps of:

creating an instance of software;

creating a tag that is uniquely associated with the instance of software;

distributing the instance of software and securely distributing the tag to a user device and receiving the instance of software and the associated tag at the user device;

detecting an attempt to use the instance of the software on the user device;

determining if the attempt to use the instance of the software is allowable by determining a status of the tag that is associated with the instance of software to be used.

94. The method of claim 93 wherein the step of creating a tag includes the steps of:

assigning a unique number to the instance of software;

computing a first hash function value on portions of the content of the instance of software;

computing a second hash function value for the instance of software, the second hash function value combining the name of the software, the unique number of the instance of software, and the first hash function value.

computing a tag that is uniquely associated with the instance of software, the tag including the name of the software, the unique number of the instance of software and the second hash value.

95. The method of claim 94, wherein the step of computing a tag creates a digitally signed tag by applying a digital signature function to the second

hash function value to produce a signature and including the signature in the tag.

96. The method of claim 93, wherein the step of distributing the tag to a user device includes the step of securely distributing the tag to a software vendor and user device using a public key encryption technique.

97. The method of claim 93 wherein the step of receiving the instance of software includes the step of:

obtaining the instance of software at the user device; and

wherein the step of receiving the tag at a user device includes the steps of:

securely obtaining the tag associated with the instance of software at the user device;

determining if the tag associated with the instance of software is signed, and if so, verifying a signature on a hash function value in the tag and if the signature on the hash function value is verified, installing the software on the user device, and if the tag associated with the instance of software is not signed, installing the instance of software on the user device.

98. The method of claim 93 wherein:

the step of detecting an attempt to use the instance of the software on the user device includes the steps of:

invoking a supervising program on the user device to intercept a user request for use of the instance of software; and

wherein the step of determining if the attempt to use the instance of the software is allowable includes the steps of:

determining if a call-up procedure is needed based on a call-up policy and if so performing the next three steps:

performing a call-up procedure to verify the authenticity and to determine the usage supervision policy of the tag associated with the instance of software;

updating tag information in the user device based upon an outcome of the call-up procedure; and

examining status information associated with the tag to determine if use of the instance of software associated with the tag is allowed.

99. The method of claim 98, wherein the step of performing a call-up procedure includes the steps of:

transmitting a tag table storing the tag associated with the instance of software from the user device;

awaiting reception of a continuation message returned to the user device that indicates an action to be performed for each tag in the tag table.

100. The method of claim 98, further including the step of verifying that the continuation message is directed towards this device and that the event history corresponds to the event history at this device.

101. The method of claim 98, wherein the step of performing a call-up procedure includes the steps of:

receiving a tag table including the tag associated with the instance of software;

examining each tag received in the tag table against a tagged software database to ensure that tags in the tag table are in compliance with at least one usage supervision policy; and

transmitting a continuation message indicating an action to follow at the user device upon detecting an attempted use of the instances of software associated with each tag.

102. The method of claim 101, wherein the continuation message includes:

a supervising program identifier of the supervising program to which the continuation message is to be sent;

the time when the continuation message was prepared;

an encoding of the tag table header that accompanied the call-up from the device.

103. A method for supervising use of software comprising the steps of:

detecting use of an untagged instance of software on a user device;

creating and storing fingerprints associated with the untagged instance of software on the user device;

detecting an attempt to use the untagged instance of the software on the user device; and

determining if the attempt to use the instance of the software is valid by comparing the fingerprints associated with the untagged instance of software with a fingerprint data structure of infringing fingerprints and disabling use of the untagged instance of software if a fingerprint match is found.

104. The method of claim 103 further comprising the steps of:

detecting use of a tagged instance of software on a user device;

creating and storing fingerprints associated with the tagged instance of software on the user device;

detecting an attempt to use the tagged instance of the software on the user device; and

determining if the attempt to use the instance of the software is valid by comparing the fingerprints associated with the tagged instance of software with a fingerprint data structure of infringing fingerprints and disabling use of the tagged instance of software if a fingerprint match is found.

105. The method of claim 103, further including the steps of:

detecting, by a software vendor, an instance of infringing software;

submitting a copy of the instance of infringing software to a guardian center; and

computing fingerprints at the guardian center on the infringing instance of software and incorporating and storing the fingerprints in a fingerprint data structure.

106. A method for uniquely identifying instances of software comprising the steps of:

obtaining an instance of software;

assigning a name to the instance of software;

assigning a unique number to the instance of software, the unique number being different from any unique number assigned to another instance of the same software;

computing a hash function value on portions of the instance of software;

computing a second hash function value on a concatenation of the name of the instance software, the number of the instance software, and the first computed hash function value to produce an unsigned hash function value unique to that instance of software;

signing the unsigned hash function value using a key to produce a signed hash function value for the instance of software; and

creating a tag associated with the instance of software that uniquely identifies that instance of software, the tag including the signed hash value of the instance of software, the name of the instance of software, the unique number of the instance of software, and the unsigned hash value of the instance software.

107. The method of claim 106, wherein the steps of obtaining the instance of software and assigning a name to the software are performed by a software vendor and the steps of assigning a unique number to the instance of

software, computing the first and second hash function values, signing the second hash value, and creating the tag are performed by a tag server.

108. A computer readable medium encoded with instructions that when read and executed on a processor perform the following steps:

detecting a request to use an instance of software;

determining if a tag corresponding with the instance of software has an associated status that allows the instance of software to be used; and

periodically performing a call-up procedure to validate the authenticity of the tag and to ensure that the instance of software corresponding to the tag is used in accordance with an usage supervision policy.

109. A propagated signal transmitted via a carrier over a communications medium, the signal carrying an encoded tag table data structure which includes at least one tag that is uniquely associated with one instance of software and includes at least one field associated with the tag in the tag table, the at least one field indicating a use control status for the one instance of software associated with the tag.

110. A propagated signal transmitted via a carrier over a medium, the signal carrying an encoded continuation message, the continuation message containing an indication of actions to be performed at a receiver of the propagated signal when an attempt to use an instance of software associated with the actions is detected at the receiver.

111. A method for ensuring that a software program hasn't been altered comprising the steps of:

computing an unaliasable hash function value on contents of the software program; and

comparing the result of the unaliasable hash function with a result of a previously held hash value to determine if the results are the same, thus indicating if a software program has been altered.

112. The method of claim 111 wherein the operating system computes the unaliasable hash function value and the software program is the supervising program.

113. A method for ensuring that data has not been altered by means of computing an unaliasable hash function value on the contents of that data and comparing the said value with a previously computed hash function value.

114. The method of claim 113 wherein the supervising program computes the unaliasable hash function value and the data used by the supervising program.

115. The system of claim 19 wherein all messages between the guardian center and the user device are sent in a secure fashion.

116. The system of claim 115 wherein the secure fashion involves public key encryption.

117. The system in claim 38 wherein the rarely duplicated number is further based on the values of at least one memory location.

118. The guardian center of claim 80 wherein the guardian center tests whether the last call-up time recorded in the continuation message from the device matches the call-up time of the most recent call-up record recorded on the guardian center for this device.

119. A system for supervising usage of software comprising:

a software vendor producing instances of software,
a user device receiving and installing an instance of software,
the user device including a supervising program,
an untagged instance of software used on the user device;
wherein the supervising program detects the use of the untagged instance of software and performs a fingerprinting process on the untagged instance of software and stores fingerprints resulting from the fingerprinting process on the user device.

120. The system of claim 119 where the user device's supervising program further performs a fingerprinting process on an untagged instance of software used on the device and stores the fingerprints resulting from the fingerprinting process in a fingerprint table on the user device.

121. The system of claim 120 wherein the supervising program stores locations from which the fingerprints are computed.

122. The system of claim 120 wherein the fingerprints are based on the contents of the instance of software.

123. The system of claim 120 wherein the fingerprints are based on known sequences of behavior of the instance of software.

124. The system of claim 120 further comprising:
a guardian center including:
a fingerprint data structure; and
a verification program;
the guardian center periodically communicating with the user device via a call-up procedure to receive all fingerprints from the user device for an instance of software used on the user device, the verification program comparing every fingerprint received from the user device against the

fingerprint data structure to determine if an instance of software used on the user device is an infringing instance of software.

125. The system in claim 124 wherein if the verification program detects more than a specified number of matches between fingerprints in the guardian center's fingerprint data structure and fingerprints received from the user device, the verification program specifies a punitive action to be performed, and the verification program returns a continuation message to the user device, the continuation message indicating the punitive action to be performed on the user device.

126. The system in claim 125 wherein the fingerprint matching process is at least one of general location or same location fingerprint matching.

127. The system in claim 125 wherein the fingerprint matching uses an inverted guardian center fingerprint table.

128. The system of claim 125 wherein the punitive action specifies that the user device be disabled for a specified length of time.

129. The system of claim 125 wherein the punitive action specifies that the instance of software associated with the fingerprint that was matched to a fingerprint in the fingerprint data structure of the guardian center should be disabled for a specified length of time.

130. The system of claim 125 wherein the punitive action depends on at least one of a combination of the history of the behavior of the user device, the history of the behavior of a particular user on the user device, and the collection of other software on the user device.

131. The system of claim 124 wherein the software vendor transmits a copy of an infringing instance of software to the guardian center and the guardian center computes fingerprints on the copy of the infringing instance of software and incorporates and stores the fingerprints into the fingerprint data structure on the guardian center.

132. A software vendor comprising:

a software production mechanism creating at least one instance of software incorporating a device identifier inside a test and

a user device receiving and installing the instance of software,

the test comprising the comparison of the incorporated identifier with the identifier of the device upon which the software instance is to be used;

if the incorporated identifier equals the device identifier then the software instance can be used, otherwise punitive action is taken by the supervising program on the device.

133. The software vendor of claim 132 wherein the software vendor sends a digital signature of the hash of the instance of software and

a second test determines whether the digital signature is authentic,

a third test determines whether the value signed is equal to the hash of the instance of software,

wherein if the digital signature is not authentic or the signed value is different from the instance of software, then the supervising program in the device takes punitive action.

134. The software vendor of claim 131 wherein the device identifier is incorporated at the beginning or at the end of the contents of the software instance.

135. A method for supervising usage of software comprising the steps of:

creating an instance of software incorporating a device identifier inside a test, the test comprising the comparison of the incorporated identifier with the identifier of the device upon which the software instance is to be used;

distributing the instance of software to a user device;

determining if the attempt to use the instance of the software is allowable by performing the test and allowing use if the incorporated identifier equals the device identifier then the software instance can be used, otherwise performing punitive action

136. The method of claim 135 comprising the additional steps of:

sending a digital signature of the hash of the instance of software;

determining whether the digital signature is authentic,

determining whether the value signed is equal to the hash of the instance of software,

wherein if the digital signature is not authentic or the signed value is different from the instance of software, then the supervising program in the device takes punitive action.

137. The method of claim 135 wherein the device identifier is placed at the beginning or at the end of the software instance.