What is claimed is

1. An electronic meeting tool for communicating arbitrary media content from users at a meeting comprising:

a node configuration means adapted to operate a display node of a communications network, the display node being coupled to a first display, the node configuration means being adapted to receive user selected arbitrary media content and to control display of the user selected arbitrary media content on the first display; and

at least one peripheral device adapted to communicate the user selected arbitrary media content via the communications network, wherein the peripheral device is a connection unit comprising:

(c) a connector adapted to couple to a port of a processing device having a second display, a memory and an operating system; and

(d) a transmitter for communicating with the communications network,

a program adapted to be loaded onto the processing device and to run on the operating system of the processing device, said program being adapted to obtain user selected arbitrary media content, said program leaving a zero footprint on termination, and

an input device to allow the user to carry out a user action that triggers transfer of said user selected arbitrary media content to said transmitter through said port.

2. The electronic meeting tool of claim 1 wherein the program is stored on the peripheral device.

3. The electronic meeting tool of any previous claim wherein the transmitter is a wireless transmitter or transceiver.

4. The electronic meeting tool of any previous claim wherein the peripheral device is a plug-and-play device.

5. The electronic meeting tool of any previous claim wherein the program is adapted to screen scrape content of the second display.

6. The electronic meeting tool of any previous claim wherein the input device is physical actuator coupled to the peripheral device.

7. The electronic meeting tool of claim 6 wherein the physical actuator has a surface area of between 100 and 14,400 square mm.

8. The electronic meeting tool of any previous claim wherein the input device is a key displayed on the second display.

9. The electronic meeting tool of claim 8 wherein the key displayed on the second display is not screen scraped by the program.

10. A method for connecting a processing device to a communications network, the processing device having a memory, a display and an operating system with at least one pre-installed generic driver providing a generic communications protocol for communication between processing device and a standard class of peripheral devices, the method comprising the steps of:
a) coupling a peripheral device to the processing device, the peripheral device having a transceiver;
b) setting up, by means of the pre-installed generic driver of the operating system, a means for communication between the peripheral device and the processing device;
c) coupling the processing device to a communications network via the peripheral device;
d) routing screen scraped data between the processing device and the communication network via the means for communication, wherein the generic communication protocol is used for transferring the screen scraped data between the processing device and the peripheral device.

11. The method of claim 10 wherein Step b) comprises presenting the peripheral device to the processing device as a human interface device and wherein the pre-installed generic driver is a human interface device driver.

12. The method of claim 10 wherein Step b) comprises presenting the peripheral device to the processing device as a mass storage device and wherein the pre-installed generic driver is a mass storage device driver.

13. The method of claim 10 wherein Step b) comprises presenting the peripheral device to the processing device as a composite device and wherein pre-installed generic drivers drive different device interfaces independently.

14. The method of any of claims 10 to 13 wherein a client application is stored on the peripheral device which when run on the processing device obtains the screen scraped data.

15. The method of claim 14 wherein the client application is a portable application.

16. A peripheral device for providing communication connectivity to a processing device which is provided with memory, a display and an operating system with at least one pre-installed generic driver providing a generic communication protocol for communication between the processing device and a standard class of peripheral devices, the peripheral device comprising a memory in which executable software code is stored for execution on the processing device, said executable software code comprising:
a first software code portion for setting up, by means of the pre-installed generic driver of the operating system, a means for communication between the peripheral device and the processing device;
a second software code portion for connecting the processing device to a communications network via the peripheral device, the peripheral device having a transceiver;
a third software code portion for screen scraping data from the processing device; and
a fourth software code for routing screen scraped data between the processing device and the communications network over the means for communication; wherein the first software code portion is adapted to use the generic communication protocol for transferring the screen scraped data between the processing device and the peripheral device.

17. The peripheral device of claim 16, wherein the first software code portion is adapted to present the peripheral device to the processing device as a human interface device and wherein the pre-installed generic driver is a human interface device driver.

18. The peripheral device of claim 16, wherein the first software code portion is adapted to

present the peripheral device to the processing device as a mass storage device and wherein the pre-installed generic driver is a mass storage device driver.

19. The peripheral device of claim 16, wherein the first software code portion is adapted to present the peripheral device to the processing device as a composite device and wherein pre-installed generic drivers drive different device interfaces independently.

20. The peripheral device of any of the claims 16 to 19 wherein said executable software code comprises fifth code for providing a means for connecting to the communications network including a base node.

21. The peripheral device of any of the claims 16 to 20 wherein said executable software code comprises sixth code for providing means for capturing video frame buffers of the processing device.

22. The peripheral device of any of the claims 156to 21 wherein said executable software code comprises seventh code for providing a means for encoding, compressing and optionally encrypting the screen scraped data and sending the screen scraped data the communication network.

23. The peripheral device of any of the claims 16 to 21 wherein said executable software code comprises eighth code for providing a means for handling the peripheral device.

24. The peripheral device of any of the claims 20 to 23 wherein said executable software code comprises ninth code for providing means for initiating connection to the base node.

25. The peripheral device of any of the claims 16 to 24 wherein said executable software code comprises tenth code for receiving inputs from an input device on the peripheral device.

26. The peripheral device of any of the claims 16 to 25 wherein said executable software code comprises eleventh code for providing a means for sending state changes to the visual indicator on the peripheral device.

27. The peripheral device of any of the claims 16 to 26 wherein said executable software code comprises Twelfth code for providing a means for presenting to the user a GUI.

28. The peripheral device of any of the claims 16 to 27 wherein said executable software code comprises thirteenth code for presenting a GUI for administration of the said executable software code when executed as a portable application.

29. The peripheral device of any of the claims 16 to 28 wherein said executable software code comprises fourteenth code for providing a means for displaying and activating a key on the display of the client processing device for allowing a user to input the start the transfer of data from the processing device to the base node.

30. A peripheral device for providing communication connectivity to a processing device which is provided with memory, a display and an operating system with at least one pre-installed generic driver providing a generic communication protocol for communication between the processing device and a standard class of peripheral devices, the peripheral device comprising a memory in which executable software code is stored for execution on the processing device, said executable software code comprising:
a first software code portion for setting up, by means of the pre-installed generic driver of the operating system, a means for communication between the peripheral device and the processing device;
a second software code portion for connecting the processing device to a communications network via the peripheral device, the peripheral device having a transceiver; and
a third processing software code for routing data between the processing device and the communications network over the means for communication; wherein the first software code portion is adapted to use the generic communication protocol for transferring the data between the processing device and the peripheral device, and
an input device coupled to the peripheral device, the input device being adapted to react to a user action to trigger the transfer of the data from the processing device to the peripheral device.

31. The peripheral device of claim 30, wherein the first software code portion is adapted to present the peripheral device to the processing device as a human interface device and

wherein the pre-installed generic driver is a human interface device driver.

32. The peripheral device of claim 30, wherein the first software code portion is adapted to present the peripheral device to the processing device as a mass storage device and wherein the pre-installed generic driver is a mass storage device driver.

33. The peripheral device of claim 30, wherein the first software code portion is adapted to present the peripheral device to the processing device as a composite device and wherein pre-installed generic drivers drive different device interfaces independently.

34. The peripheral device of any of the claims 30 to 33 wherein the input device is physical actuator coupled to the peripheral device.

35. The peripheral device of claim 34 wherein the physical actuator has a surface area of between 100 and 14,400 square mm.

36. The peripheral device of any of the claims 30 to 35 wherein the input device is a key for display on the display.

37. The peripheral device of any of the claims 30 to 36 wherein said executable software code comprises fourth code for providing a means for connecting to the communications network including a base node.

38. The peripheral device of any of the claims 30 to 36 wherein said executable software code comprises fifth code for providing a means for capturing video frame buffers of the processing device.

39. The peripheral device of claim 38 wherein said executable software code comprises sixth code for providing a means for encoding, compressing and optionally encrypting the video frames and sending them over a secure link to the base node.

40. The peripheral device of any of the claims 30 to 39 wherein said executable software code comprises seventh code for providing a means for handling the peripheral device.

41. The peripheral device of claim 39 or 40 wherein said executable software code comprises eighth code for providing a means for initiating connection to the base node.

42. The peripheral device of any of the claims 30 to 41 wherein said executable software code comprises ninth code for receive inputs from an input device on the peripheral device.

43. The peripheral device of any of the claims 30 to 42 wherein said executable software code comprises tenth code for providing a means for sending state changes to the visual indicator on the peripheral device.

44. The peripheral device of any of the claims 30 to 43 wherein said executable software code comprises eleventh code for providing a means for presenting the user a GUI.

45. The peripheral device of any of the claims 30 to 44 wherein said executable software code comprises twelfth code for presenting GUI for administration of the executable software code when executed as a portable application.

46. The peripheral device of any of the claims 37 to 45 wherein said executable software code comprises thirteenth code for providing a means for displaying and activating a key on the display of the processing device for allowing a user input to start the transfer of data from the processing device to the base node.

47. A peripheral device for providing communication connectivity to a processing device which is provided with memory, a display and an operating system with at least one pre-installed generic driver providing a generic communication protocol for communication between the processing device and a standard class of peripheral devices, the peripheral device comprising a memory in which executable software code is stored for execution on the processing device, said executable software code comprising:

a first software code portion for setting up, by means of the pre-installed generic driver of the operating system, a means for communication between the peripheral device and the processing device;

a second software code portion for connecting the processing device to a communications network via the peripheral device, the peripheral device having a transceiver;

a third software code portion for receiving media content from the network and for displaying the media content on the display in accordance with a set of rules; wherein the first software code portion is adapted to use the generic communication protocol for transferring the media content between the peripheral device and the processing device.

48. The peripheral device of claim 47 wherein said executable software code comprises fourth code for providing a means for correctly configuring a receiver as an access point.

49. The peripheral device of claim 47 or 48 wherein said executable software code comprises fifth code for providing a means for listening for output of connection units trying to connect on a specific port.

50. The peripheral device of any of the claims 47 to 49 wherein said executable software code comprises sixth code for providing a means for a GUI for administration purposes.

51. The peripheral device of any of the claims 47 to 50 wherein said executable software code comprises seventh code for providing a means for publishing its presence over the network using the zeroconf protocol.

52. The peripheral device of any of the claims 47 or 51 wherein said executable software code comprises eighth code for providing a means for accepting and installing software updates.

53. The peripheral device of any of the claims 47 or 52 wherein said executable software code comprises ninth code for providing a means for providing facilities for pairing of connection units to the processing device.

54. The peripheral device of any of the claims 47 or 53 wherein said executable software code comprises tenth code for providing a means for auto-composing of different incoming arbitrary media streams and rendering of composited image on display.

55. The peripheral device of any of the claims 47 or 54 wherein said executable software code comprises eleventh code for providing a means for receiving, decrypting and decoding incoming arbitrary media content.

56. The peripheral device of any of the claims 47 or 55 wherein said executable software code comprises twelfth code for scaling of incoming arbitrary media streams.

57. The peripheral device of any of the claims 47 or 56 wherein said executable software code comprises thirteenth code for providing a means for displaying incoming arbitrary media content in accordance with a set of rules.

58. A method for communicating arbitrary media content from users at a meeting comprising:
operating a display node of a communications network, the display node being coupled to a first display, to receive user selected arbitrary media content and to control display of the user selected arbitrary media content on the first display; and
connecting a peripheral device to a port of a processing device and communicating the user selected arbitrary media content via the communications network,
loading a program onto the processing device and running the program on the operating system of the processing device to obtain user selected arbitrary media content, said program leaving a zero footprint on termination, and
triggering transfer of said user selected arbitrary media content to said transmitter through said port after a user action on an input device.

59. A peripheral device comprising:
a base
a connector for connection to a plug and play port of a host processing device,
a flexible connection between the base and the connector for transferring data signals and power,
an actuator on the base for actuating a signal and for transferring the signal to the connector for transfer to the port, wherein the base has electronics comprising permanent storage for storing a portable application, a processing engine, a transceiver and a visual indicator.

60. A method for providing communication connectivity from a processing device,
setting up a communications network between a base node of the communications network and a peripheral device coupled to the processing device;

transferring media content between the processing device and the peripheral device, transmitting the media content from the peripheral device to the communications network, receiving media content from the communications network at the base node and displaying the media content on a display in accordance with a set of rules.

5

61. Any of the above software code stored on a non-transitory storage medium.

CLAIMS

1. A method for connecting a processing device to a communications network, the processing device having a memory, a display and an operating system with at least one pre-installed generic driver providing a generic communications protocol for communication between processing device and a standard class of peripheral devices, the method comprising the steps of:

a) coupling a peripheral device to the processing device, the peripheral device having a transceiver;

b) setting up, by means of the pre-installed generic driver of the operating system, a means for communication between the peripheral device and the processing device;

c) coupling the processing device to a communications network via the peripheral device;

d) routing screen scraped data between the processing device and the communication network via the means for communication, wherein the generic communication protocol is used for transferring the screen scraped data between the processing device and the peripheral device.

2. The method of claim 1 wherein Step b) comprises presenting the peripheral device to the processing device as a human interface device and wherein the pre-installed generic driver is a human interface device driver.

3. The method of claim 1 wherein Step b) comprises presenting the peripheral device to the processing device as a mass storage device and wherein the pre-installed generic driver is a mass storage device driver.

4. The method of claim 1 wherein Step b) comprises presenting the peripheral device to the processing device as a composite device and wherein pre-installed generic drivers drive different device interfaces independently.

5. The method of any of claims 1 to 4 wherein a client application is stored on the peripheral device which when run on the processing device obtains the screen scraped data.

6. The method of claim 5 wherein the client application is a portable application.

7. A peripheral device for providing communication connectivity to a processing device which is provided with memory, a display and an operating system with at least one pre-installed generic driver providing a generic communication protocol for communication between the processing device and a standard class of peripheral devices, the peripheral device comprising a memory in which executable software code is stored for execution on the processing device, said executable software code comprising:

a first software code portion for setting up, by means of the pre-installed generic driver of the operating system, a means for communication between the peripheral device and the processing device;

a second software code portion for connecting the processing device to a communications network via the peripheral device, the peripheral device having a transceiver; and

a third processing software code for routing data between the processing device and the communications network over the means for communication;

wherein the first software code portion is adapted to use the generic communication protocol for transferring the data between the processing device and the peripheral device, and

an input device coupled to the peripheral device, the input device being adapted to react to a user action to trigger the transfer of the data from the processing device to the peripheral device.

8. The peripheral device of claim 7, wherein the first software code portion is adapted to present the peripheral device to the processing device as a human interface device and wherein the pre-installed generic driver is a human interface device driver.

9. The peripheral device of claim 7, wherein the first software code portion is adapted to present the peripheral device to the processing device as a mass storage device and wherein the pre-installed generic driver is a mass storage device driver.

10. The peripheral device of claim 7, wherein the first software code portion is adapted to present the peripheral device to the processing device as a composite device and wherein pre-installed generic drivers drive different device interfaces independently.

11. The peripheral device of claim 7 wherein the input device is physical actuator coupled to the peripheral device.

12. The peripheral device of claim 7 wherein the input device is a key for display on the display.

13. The peripheral device of claim 7 wherein said executable software code comprises fourth code for providing a means for connecting to the communications network including a base node.

14. The peripheral device of claim 7 wherein said executable software code comprises fifth code for providing a means for capturing video frame buffers of the processing device.

15. The peripheral device of claim 14 wherein said executable software code comprises sixth code for providing a means for encoding, compressing and optionally encrypting the video frames and sending them over a secure link to the base node.

16 The peripheral device of claim 7 wherein said executable software code comprises seventh code for providing a means for handling the peripheral device.

17. The peripheral device of claim 15 wherein said executable software code comprises eighth code for providing a means for initiating connection to the base node.

18. The peripheral device of claim 7 wherein said executable software code comprises ninth code for receive inputs from an input device on the peripheral device.

19. The peripheral device of claim 7 wherein said executable software code comprises tenth code for providing a means for sending state changes to the visual indicator on the peripheral device.

20. The peripheral device of claim 7 wherein said executable software code comprises eleventh code for providing a means for presenting the user a GUI.

21. The peripheral device of claim 7 wherein said executable software code comprises twelfth code for presenting GUI for administration of the executable software code when executed as a portable application.

22. The peripheral device of claim 13 wherein said executable software code comprises thirteenth code for providing a means for displaying and activating a key on the display of the processing device for allowing a user input to start the transfer of data from the processing device to the base node.