Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 2921429 C 2023/01/17

(11)(21) 2 921 429

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(86) Date de dépôt PCT/PCT Filing Date: 2014/08/14
(87) Date publication PCT/PCT Publication Date: 2015/02/19
(45) Date de délivrance/Issue Date: 2023/01/17
(85) Entrée phase nationale/National Entry: 2016/02/16
(86) N° demande PCT/PCT Application No.: GB 2014/052499
(87) N° publication PCT/PCT Publication No.: 2015/022540
(30) Priorité/Priority: 2013/08/16 (GB1314732.7)

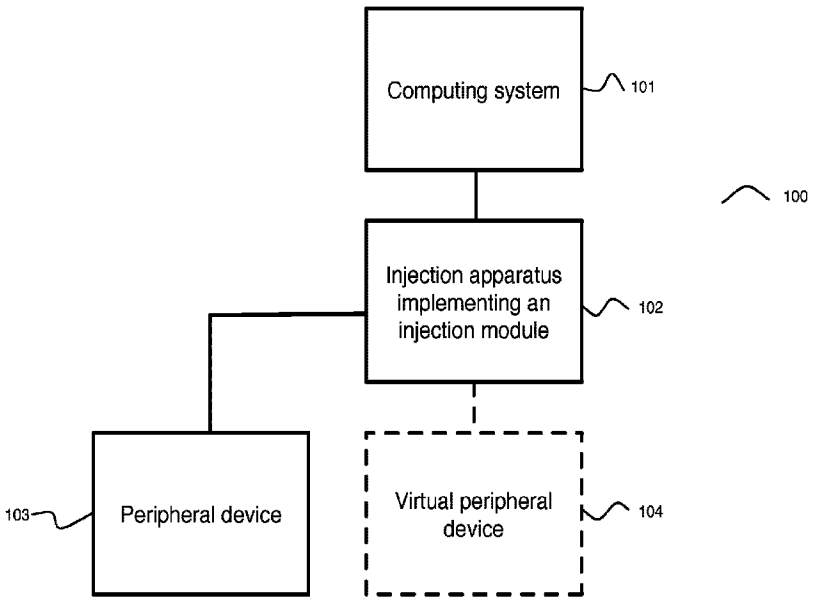
(51) Cl.Int./Int.Cl. *H04L 67/52* (2022.01), *H04L 67/561* (2022.01), *H04L 69/321* (2022.01)

(72) Inventeurs/Inventors:
BRIGHTON, MARK, GB;
FERRER, JUDD, GB

(73) Propriétaire/Owner:
SPARKLE CS LTD, GB

(74) Agent: PERRY + CURRIER

(54) Titre : PROCEDE ET SYSTEME DE TRANSMISSION DE DONNEES
(54) Title: A DATA TRANSMISSION METHOD AND SYSTEM



(57) Abrégé/Abstract:
The present invention relates to a data transmission method including injecting signals into a communications channel between a predefined hardware device for a computing system and an application executing on the computing system, for receipt by the application. A data transmission system is also disclosed.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 February 2015 (19.02.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/022540 A1

(51) International Patent Classification:
H04L 29/08 (2006.01)

(21) International Application Number:
PCT/GB2014/052499

(22) International Filing Date:
14 August 2014 (14.08.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
1314732.7 16 August 2013 (16.08.2013) GB

(71) Applicant: **SPARKLE COUPON SERVICES LTD** [GB/GB]; Hart House, Priestley Road, Basingstoke Hampshire RG24 9PU (GB).

(72) Inventors: **BRIGHTON, Mark**; 67 Oakley Lane, Oakley, Basingstoke Hampshire RG23 7JT (GB). **FERRER, Judd**; Elsmar Lodge, Sleaford Road, Beckingham Lincolnshire LN5 0RF (GB).

(74) Agent: **RATIONAL IP LIMITED**; 81 Rivington Street, London Greater London EC2A 3AY (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) Title: A DATA TRANSMISSION METHOD AND SYSTEM

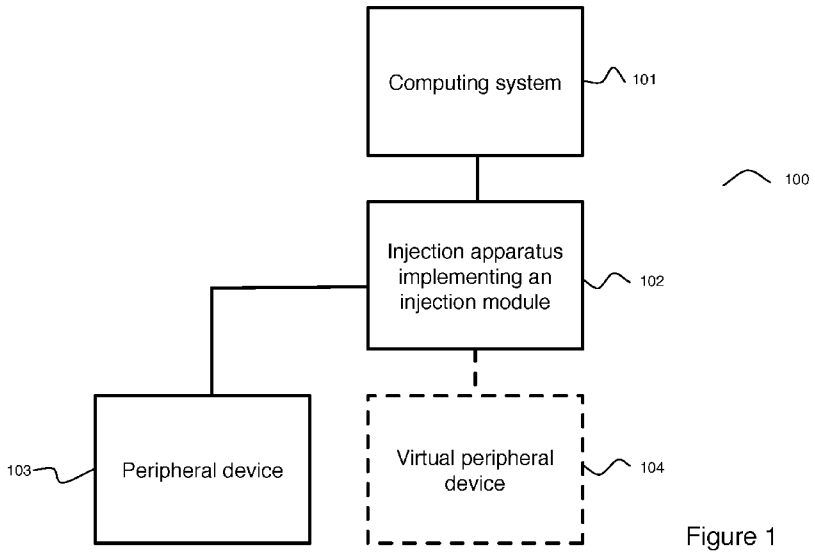


Figure 1

(57) Abstract: The present invention relates to a data transmission method including injecting signals into a communications channel between a predefined hardware device for a computing system and an application executing on the computing system, for receipt by the application. A data transmission system is also disclosed.

WO 2015/022540 A1

A Data Transmission Method and System

Field of Invention

The present invention is in the field of data transmission. Particularly, but not exclusively, the present invention relates to the injection of data into a hardware device communications channel to enable the transmission of data to, for example, a legacy computing system.

Background

It is often desirable to add functionality to deployed technology architectures. However, introducing new functionality often requires substantial modification to the already deployed technology. Modification is generally undesirable as it can introduce unexpected effects within interoperating technology and increase the chances of errors occurring within complex technology.

One field which finds it challenging to improve functionality is the retail sector, particularly in regard to their point-of-sale terminals which tend to be installed and operated by monopoly technology providers.

Retail is, conversely, a field which would derive tremendous advantage from new processes. For example, coupons in new forms have attracted the interest and engagement of consumers. Retailers would like to implement new and intelligent processes for managing coupons in their stores.

Accordingly, there is a desire for additional functionality deploying new processes to be added to legacy computing systems, in particular in retail.

With couponing, there are several improvements that would usefully improve the functionality of existing retail systems. For example, it may be desirable for

a retailer to easily apply coupons, such as digital coupons, that cannot be scanned.

To implement these improvements, an improved data transmission method and system which is easy to deploy into legacy environments is required.

It is an object of the present invention to provide a data transmission system which overcomes the disadvantages of the prior art, or at least provides a useful alternative.

Summary of Invention

According to a first aspect of the invention there is provided a data transmission method, including: constructing a communications channel between a peripheral device for a computing system and an application executing on the computing system; and injecting signals into the communications channel for receipt by the application.

According to a first aspect of the invention there is provided a data transmission system, comprising: a processor configured for constructing a communications channel between a peripheral device for a computing system and an application executing on the computing system and injecting signals into the communications channel for receipt by the application; and a communications module configured for communicating with an application executing on the computing system via the communications channel.

Other aspects of the invention are described within the claims.

Brief Description of the Drawings

Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1: shows a data transmission system in accordance with an embodiment of the invention;

Figure 2: shows another data transmission system in accordance with an embodiment of the invention;

Figure 3: shows a communications architecture in accordance with an embodiment of the invention;

Figure 4: shows another communications architecture in accordance with an embodiment of the invention;

Figure 5: shows a flow chart illustrating a method in accordance with an embodiment of the invention;

Figure 6: shows a block diagram illustrating a message format of signals used within an embodiment of the invention;

Figure 7: shows a block diagram illustrating a software injection module within an operating system in accordance with an embodiment of the invention;

Figure 8: shows a sequence diagram illustrating message flow within a method in accordance with an embodiment of the invention; and

Figure 9: shows a data processing system in accordance with an embodiment of the invention.

Detailed Description of Preferred Embodiments

The present invention provides a data transmission method.

The inventor has discovered that data can be transmitted to a legacy computing system, such as a POS terminal system, by injecting data into a communications channel with a hardware device and the computing system.

For example, a POS terminal may be configured to receive barcode information from a barcode scanner hardware input device to the POS terminal computing system. Data can be spoofed to appear to come from a barcode scanner and injected into the communications channel to the POS terminal application software.

In Figure 1, a data transmission system 100 in accordance with an embodiment of the invention is shown.

A computing system 101 is shown. The computing system 101 is configured to execute a software application. The computing system 101 may be a point-of-sale (POS) terminal configured to execute an electronic point-of-sale (EPOS) software application.

The computing system 101 includes a processor and a memory.

An injection apparatus 102 implementing an injection module is shown. The injection apparatus 102 may be connected to the computing system 101, for example, by a port on the computing system 101. The connection permits signals to be transmitted and received between the injection apparatus 102 and computing system 101.

The injection apparatus 102 includes a processor and a memory.

A peripheral device 103 is shown. The peripheral device 103 may be an input hardware device, such as a QR scanner, a NFC reader, or a camera.

In an alternative embodiment, the data transmission system 100 does not require the peripheral device 103.

The peripheral device 103 may be connected to the injection apparatus 102 such that signals can be transmitted and received between the injection apparatus 102 and the peripheral device 103.

A virtual peripheral device 104 is shown. The virtual peripheral device 104 may be an input hardware device, such as a barcode scanner, or any other type of input device. The virtual peripheral device 104 is virtual in the sense that it does not exist, but its existence is spoofed to the application on the computing system 101 by the injection apparatus 102.

The injection apparatus 102 may be configured to create a communications channel which appears to come from the virtual peripheral device 104 and the application executing on the computing system 101. The injection apparatus 102 may be further configured to inject signals into the communications channel between the virtual peripheral device 104 and the application on the computing system 101.

The injection apparatus 102 may be configured to process signals received from the actual peripheral device 103 and generate signals in response for injection into the communications channel.

In Figure 2, a data transmission system 200 in accordance with another embodiment of the invention is shown.

A computing system 201 is shown. The computing system 201 is configured to execute a software application 201a. The computing system 201 may be a point-of-sale (POS) terminal configured to execute an electronic point-of-sale (EPOS) software application.

The computing system 201 includes a processor and a memory.

A peripheral device 202 is shown. The peripheral device 202 may be an input hardware device, such as a QR scanner, a NFC reader, or a camera.

In an alternative embodiment, the data transmission system 200 does not require the peripheral device 202.

The peripheral device 202 is connected to the computing system 201 such that signals can be transmitted and received between the computing system 201 and the peripheral device 202.

A virtual peripheral device 203 is shown. The virtual peripheral device 203 may be an input hardware device, such as a barcode scanner, or any other type of input device. The virtual peripheral device 203 is virtual in the sense that it does not exist, but its existence is spoofed to the application 201a on the computing system 201 by an injection module 204.

The computing system 201 may be further configured to execute the injection module 204.

The injection module 204 may be configured, in turn, to create a communications channel which appears to come from the virtual peripheral device 203 and the application 201a executing on the computing system 201. The injection module 204 may be further configured to inject signals into the communications channel between the virtual peripheral device 203 and the application 201a on the computing system 201.

The injection module 204 may be configured to process signals received from the actual peripheral device 202 and generate signals in response for injection into the communications channel.

In Figure 3, an exemplary deployment of a data transmission system 300 within a network architecture in accordance with an embodiment of the invention is shown.

A data transmission system 300 as described in relation to Figures 1 or 2 is shown.

The data transmission system 300 may be further configured to communicate with a server 301 via a network 302.

In Figure 4, an alternative exemplary deployment of a data transmission system within a network architecture in accordance with an embodiment of the invention is shown.

A plurality of data transmission systems 400, 401, and 402 as described in relation to Figures 1 or 2 are shown.

The data transmission systems 400, 401, and 402 may be connected to a network 403. The network 403 may be connected to a second network 404 via a gateway 405.

The data transmission systems 400, 401, and 402 may be further configured to communicate with a server 406 via the first network 403, the gateway 405, and the second network 404.

It will be appreciated that alternative deployments of the data transmission system may be envisaged. For example, with cloud computing architectures,

functions may be distributed amongst a plurality of devices or servers, which may be interconnected by various network and communication configurations.

In Figure 5, a method 500 in accordance with an embodiment of the invention is shown.

In step 501, a communications channel is constructed between a first peripheral and an application executing on the computing system by an injection module. The communications channel may be spoofed, in that the first peripheral may not exist.

In step 502, signals may be received from a further peripheral device by the injection module. The signals may be related to data captured by the further peripheral device. For example, the further peripheral device may capture and encode QR data, NFC information, or photographic information. The further peripheral device may not be directly supported by the application software on the computing system.

In one embodiment as shown in step 503, signals may be received from an external server. The signals may be received in response to signals received in step 502 and transmitted on to the server, or in response to events triggered at the server.

In one embodiment, the injection module may as part of the processing generate the display of a prompt to the operator of the computing system. The prompt may be displayed on a display of the computing system or the prompt may be displayed on an external display connected, for example, to the injection apparatus. The operator response to the prompt can be received by the injection module and determine the content or type of signals to be injected.

In step 504, the injection module injects signals into the communications channel between the first peripheral device for a computing system and an application executing on the computing system, for receipt by the application.

The signals injected may be those received, or new signals may be created in response to those signals.

For example, the injection module may reprocess the received signals into a message format that the application software is expecting, and/or the injection module may capture the received signals and generate new signals in response at times and in a sequence that is expected by the application software.

Signals may be received from multiple peripheral devices and be processed by the injection module to generate one or more new signals for injection into the spoofed communications channel.

In one embodiment, the injection module may be also configured to intercept signals from a third peripheral device, such as a peripheral configured for use with the application software (i.e. a barcode scanner for use with a POS terminal), that forms a communications channel with the application software. The injection module may be configured to process those signals in conjunction with the signals received from the further peripheral to generate new signals in the spoofed communications channel.

Figure 6 shows the format of signals expected by the application software from the virtual peripheral device in one embodiment of the invention.

Within that figure:

PID = Packet ID which identifies the type of packet being sent;

ADDR = Address field of device for which the packet is intended;

ENDP = Endpoint;

CRC = Cyclic Redundancy Check; and
EOP = End of Packet.

Token packets 600 are transmitted from the application software for receipt by the virtual peripheral device and are intercepted by the injection module.

In response to receipt of the token packets 600, data packets 601 are constructed and sent from the injection module to the application software.

In response to receipt of the data packets 601, status packets 602 are constructed and send from the application software for receipt by the virtual peripheral device but are intercepted by the injection module.

It can be seen from the above that the injection module may be configured to generate signals formatted in accordance with a message format, such as that described above. In this way the application software receiving the data believes that it comes from an existing peripheral and the existence of the injection module is transparent.

A software injection module in accordance with an embodiment of the invention will now be described with reference to Figure 7.

The method will be described in relation to the use of a software injection module 700 executing on the computing system and in relation to the use for improving a computing system executing a point-of-sale (POS) software application 701.

The software injection module 700 is installed at the operating system level on the same device (the computing system) that is hosting the driver software 702 for the peripherals 703 expected by the POS 701.

To create and install the software injection module 700:

1. A software object 'wedge' 704 is created which presents the same interface to the POS application 701 as the original peripheral class 705 for a peripheral type which is known to the POS 701 but whose existence 705a will be spoofed to the POS 701.

2. The POS software 701 is configured to load the wedge object 704 instead of the original class 705.

3. The software injection module 700 in turn loads a service object 706 for a peripheral 706a of a type not known to the POS 701. A basket builder 707 component may, for example, load the service object and manage communication to/from the new peripheral 706a.

The following description is for POS for .NET but, with suitable modifications, the invention of substituting one object's registration with another would equally well work with other peripheral driver standards such as OPOS and JPOS.

Initially a software object 704 is created which presents the same interface to the POS as the original bar scanner class. In POS for .NET these objects are called Service Objects or SOs. The POS software 701 must then be configured to load the created SO instead of the original class 705. How this happens will be dependent upon the client software 701. For example, if the client software 701 has a configuration file containing the name of an SO to load then this may be altered to load the wedge 704 instead. If no such configuration file is available and the SO is being loaded by name directly then the replacement object 704 could be given the same name as the original 705 and copied in its place on the filing system.

A POS for .NET service object browser object can be used to retrieve a list of available SOs which may be queried according to the device type they represent. A registry key contains values representing directories where service object binaries reside. This is used by the browser to find SOs. The wedge object's directory is added to this key. A mechanism exists in POS 701

for .NET to redefine the hardware device with which an SO is associated, and this may be used to dynamically link the wedge SO 704 with a particular hardware device ID at installation time. A config file is written which maps the hardware ID to a particular SO's name, and this is placed in a folder which is added to another registry key used by POS for .NET to locate such config files. At device instantiation time POS for .NET will load any config files it finds and dynamically use the mapping contained within to load the appropriate SO for the device with the referenced hardware ID being opened. The ID of the hardware to be replaced can be extracted from metadata exposed by the device's hardware manager.

The wedge SO 704 may be created so that it fully implements the class hierarchy which is being spoofed. For example the POS for .NET standard defines a class for scanners which has at its top level a class called ScannerBase. This in turn inherits from multiple other classes, all of which can be implemented in order to maintain the inheritance hierarchy and polymorphic nature of the class.

Once this process is complete, when an interface method is called by the POS 701, it is thus calling the associated method on the wedge 704 and when an event is triggered by new data, it is similarly being fed back to the POS 701 via the wedge 704. As long as the wedge object 704 and basket builder 707 manage the methods and events from the POS Software 701 and repackage any messages from new peripheral 706a, the POS Software 701 will operate as if the wedge object 704 did not exist and will believe it is interfacing with a known peripheral 705a.

The wedge 704 and basket builder 707 can include further processing logic to redirect the method and event data to other software modules 708 to 710 to process the received data and, based on instructions from those modules, replace, discard or inject new data into the original data travelling to/from new peripheral 706a.

A sequence diagram showing message flow within a data transmission system in accordance with an embodiment of the invention will now be described with reference to Figure 8.

A code is scanned by, for example, a QR or RFID scanner and the data is sent in step 800 to the application software for an electronic point-of-sale (EPOS) terminal. The QR or RFID scanner may not be known to the EPOS terminal.

The code may be presented for scanning during the scanning of a series of goods for a customer. The series of goods, and any associated coupons, personal identifiers and payment tokens, will be termed a basket.

An injection module receives the code and sends the code to a basket builder and evaluator module in step 801.

The basket builder and evaluator module adds any goods it identifies within the code to the basket it is tracking for the customer in step 802. The injection module may also receive codes to build the basket from other peripherals such as barcode scanners.

The basket builder and evaluator module may then make a lookup request using the code from a remote server in step 803. For example, if the code relates to a presented coupon rather than a particular good, the basket builder and evaluator may need to verify if the coupon is valid.

The basket builder and evaluator module may evaluate the current basket for the customer to determine if the goods within the basket qualify for a particular action (e.g. a discount) in step 804.

The basket builder and evaluator module may prompt for input from the EPOS user/operator before applying an action in step 805. For example, there may be local conditions that the user is required to verify (e.g. is the customer is over the age of 65 or is female, etc.) These local conditions may be required, for example, in relation to a presented coupon or to the particular discounts presently in effect. Alternatively, the prompt may inform the EPOS operator that a coupon appears to be invalid and giving them the option to accept or reject the coupon; or the prompt may request a code for a one-time CAN (Coupon Authentication Number) protected coupons, which the customer or operator enters (the code may then checked for previous use).

The basket builder and evaluator module may prompt for input at different times during the process.

The basket builder and evaluator module may then generate instructions, in step 806, for the injection module based on one or more of the constitution of the basket, the validity of any presented coupon and the verification of the required elements.

The instructions may result in the injection module generating data in step 807 which would instruct an action known to the EPOS (for example, applying a discount of a specified amount or percentage). The data may be sent in step 808 to the EPOS software by, for example, injecting the data into a spoofed communications channel created by the injection module between a virtual peripheral and the EPOS software.

Although Figure 8 shows a RFID scanner as the source of a scanned code, the injection module can equally intercept data feeds from one or more further peripherals including standard EPOS peripherals (e.g. cash drawer, receipt printer), proprietary peripherals, and commodity peripherals.

Instructions returned to the injection module by the basket builder and evaluator module can be multiple instructions. So a QR code scanned by a proprietary optical scanner could lead to multiple injected barcodes sent as if from a barcode scanner.

The injection module can be either hardware or software based with messaging being the same.

An injection module implemented within an injection apparatus in accordance with an embodiment of the invention will now be described with reference to Figure 9.

The injection apparatus may comprise a processor and memory within a communications hub 900 or within an attached interactive display 901 (a display and an input).

Communications routed through the hub 900 may be received by the injection module.

The communications hub 900 may have USB connections or other communications inputs. Adaptors may be used to connect non-USB inputs/outputs to the USB connections.

Input and output peripherals 902 to 907 may be connected to the communications hub 900.

The communications hub 900 may be connected to an electronic point-of-sale (POS) terminal 908 through one or more communication connections. For example, the POS terminal 908 may utilise industry standard input/output connections such as “scanner in”, “printer out”, and “cash drawer out”.

The communications hub 900 may include a wifi dongle 909 and/or an Ethernet connection 910 to communicate with an external server 911.

Where the injection module injects a new barcode, it will 'hijack' a token packet sent from the POS 908 and return its own data packet containing the injected code. After receipt of the resulting status packet from the POS, and assuming no other configurations would prevent it, normal communications between the peripheral and POS are then resumed.

Note that in an alternative configuration, the injection module may never block any communications between a standard peripheral 905 to 907 and host 908. In this configuration, a prompt will be triggered after a coupon barcode has already been passed to the POS 908 (potentially relying upon the POS operator to manually remove an invalid coupon) and a new code will be injected (for example) by spoofing an extra barcode scanner. This configuration may be required for increased reliability in a retail environment.

A potential advantage of some embodiments of the present invention is that data can be transmitted to an application on a computing system where the application is configured to interoperate with an expected external peripherals without requiring the application to be modified. Accordingly, a data transmission system in accordance with the invention results in an improved computing system which can execute legacy software whilst remaining reliable.

While the present invention has been illustrated by the description of the embodiments thereof, and while the embodiments have been described in considerable detail, it is not the intention of the applicant to restrict or in any way limit the scope of the appended claims to such detail. Additional advantages and modifications will readily appear to those skilled in the art. Therefore, the invention in its broader aspects is not limited to the specific details, representative apparatus and method, and illustrative examples

shown and described. Accordingly, departures may be made from such details without departure from the spirit or scope of applicant's general inventive concept.

Claims

1. A data transmission method, including:
   a) A software injection module constructing a communications channel between a peripheral device for an electronic point-of-sale (EPOS) terminal and an EPOS application executing on the terminal; and
   b) The software injection module injecting signals into the communications channel for receipt by the EPOS application; wherein the software injection module includes a software wedge object, the software wedge object presents the same interface to the EPOS application as an original peripheral class for a peripheral type which is known to the EPOS application but whose existence will be spoofed to the EPOS application, the EPOS application loads the wedge object instead of the original class, and the software injection module loads a service object for a peripheral of a type not known to the EPOS application.

2. A data transmission method as claimed in claim 1, wherein the peripheral device is a virtual peripheral device and the communications channel is constructed by spoofing the existence of the virtual peripheral device.

3. A data transmission method as claimed in any one of claims 1 to 2, wherein the peripheral device is an input device.

4. A data transmission method as in any one of claims 1 to 3, wherein the signals are created in response to preceding signals intercepted from a further peripheral device to the application.

5. A data transmission method as in any one of claims 1 to 4, wherein the signals are created, at least in part, in accordance with a stored set of instructions.

Date reçue/date received 2021-11-04

6. A data transmission method as in any one of claims 1 to 5, wherein the signals are created, at least in part, with the assistance of an external server.

7. A data transmission method as in any one of claims 1 to 6, wherein the signals are injected at an operating system level.

8. A data transmission method as in any one of claims 1 to 7, wherein the signals are injected by an intermediary hardware device into which the peripheral device is connected and which in turn is connected to the EPOS terminal.

9. A data transmission method as in any one of claims 1 to 8, wherein the peripheral device is a barcode reader.

10. A data transmission system, comprising:
a processor configured for constructing a communications channel between a peripheral device for an electronic point-of-sale (EPOS) terminal and an EPOS application executing on the terminal and injecting signals into the communications channel for receipt by the application using a software injection module; and
a communications module configured for communicating with an EPOS application executing on the terminal via the communications channel;
wherein the software injection module includes a software wedge object, the software wedge object presents the same interface to the EPOS application as an original peripheral class for a peripheral type which is known to the EPOS application but whose existence will be spoofed to the EPOS application, the EPOS application loads the wedge object instead of the original class, and the software injection module loads a service object for a peripheral of a type not known to the EPOS application.

11. A data transmission system as claimed in claim 10, further comprising:
the terminal; wherein the terminal comprises the processor.

Date reçue/date received 2021-11-04

12. A data transmission system as claimed in claim 10, further comprising an interceptor apparatus; wherein the interceptor apparatus comprises the processor.

13. A data transmission system as claimed in any one of claims 10 to 12, further comprising:
a display configured for displaying prompts to a user of the terminal; and
an input configured for receiving responses to the prompts from the user;
wherein the processor is further configured for generating the display of prompts and receiving the responses, and wherein the signals are generated, at least in part, on the responses.

14. A data transmission system as claimed in any one of claims 10 to 13, further comprising:
one or more peripherals, wherein the processor is further configured for receiving data from the one or more peripherals and wherein the signals are generated, at least in part, on the data received from the one or more peripherals.

15. A data transmission system as claimed in claim 14, wherein at least one of the one or more peripherals transmits data in a format unknown to the EPOS application.

16. A data transmission system as claimed in any one of claims 11 to 15, wherein the peripheral is a virtual peripheral.

17. A computer medium configured for storing a computer program configured for performing the method of any one of claims 1 to 9.

Date reçue/date received 2021-11-04

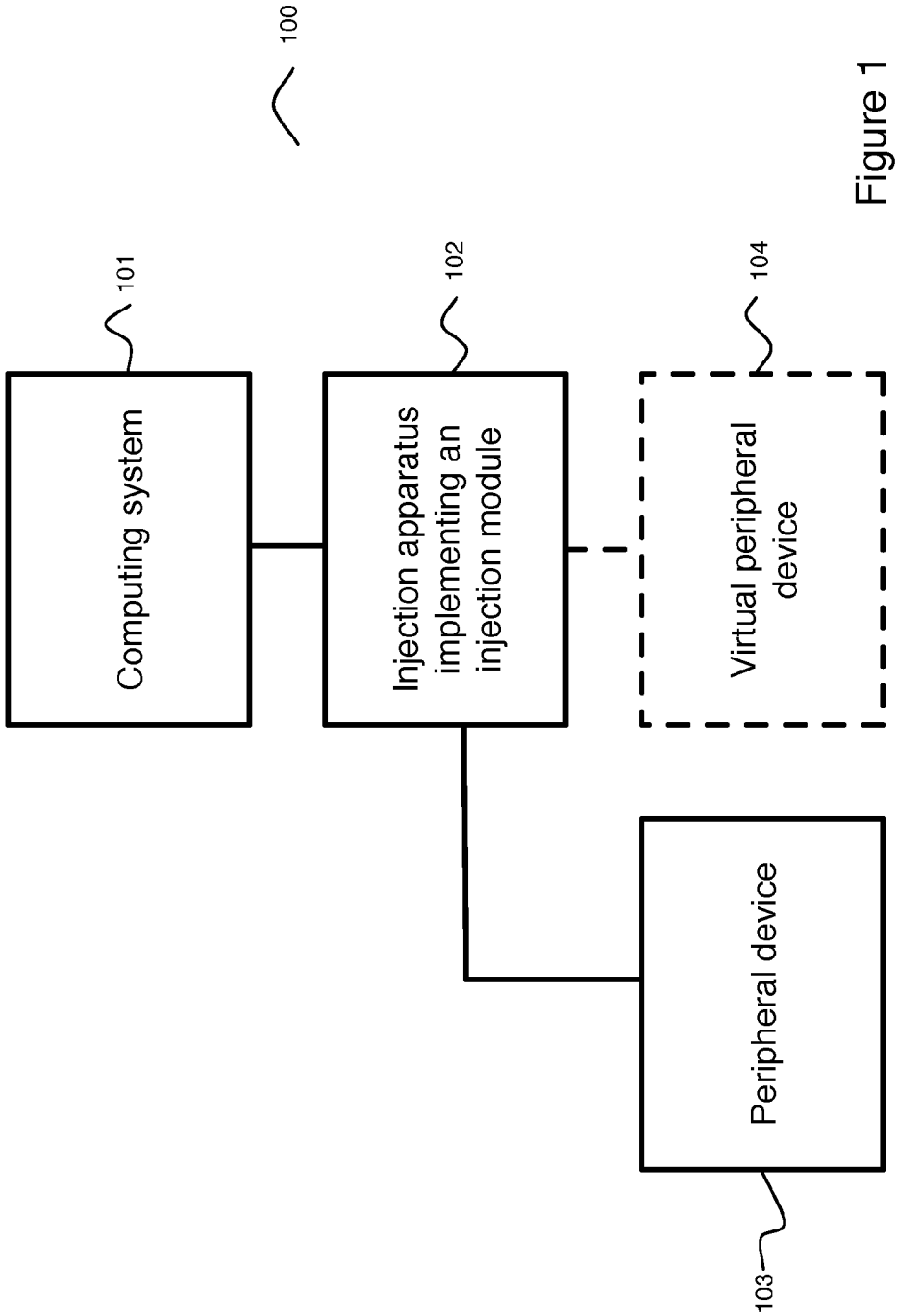

100
101
Computing system
102
Injection apparatus implementing an injection module
104
Virtual peripheral device
Peripheral device
103

Figure 1

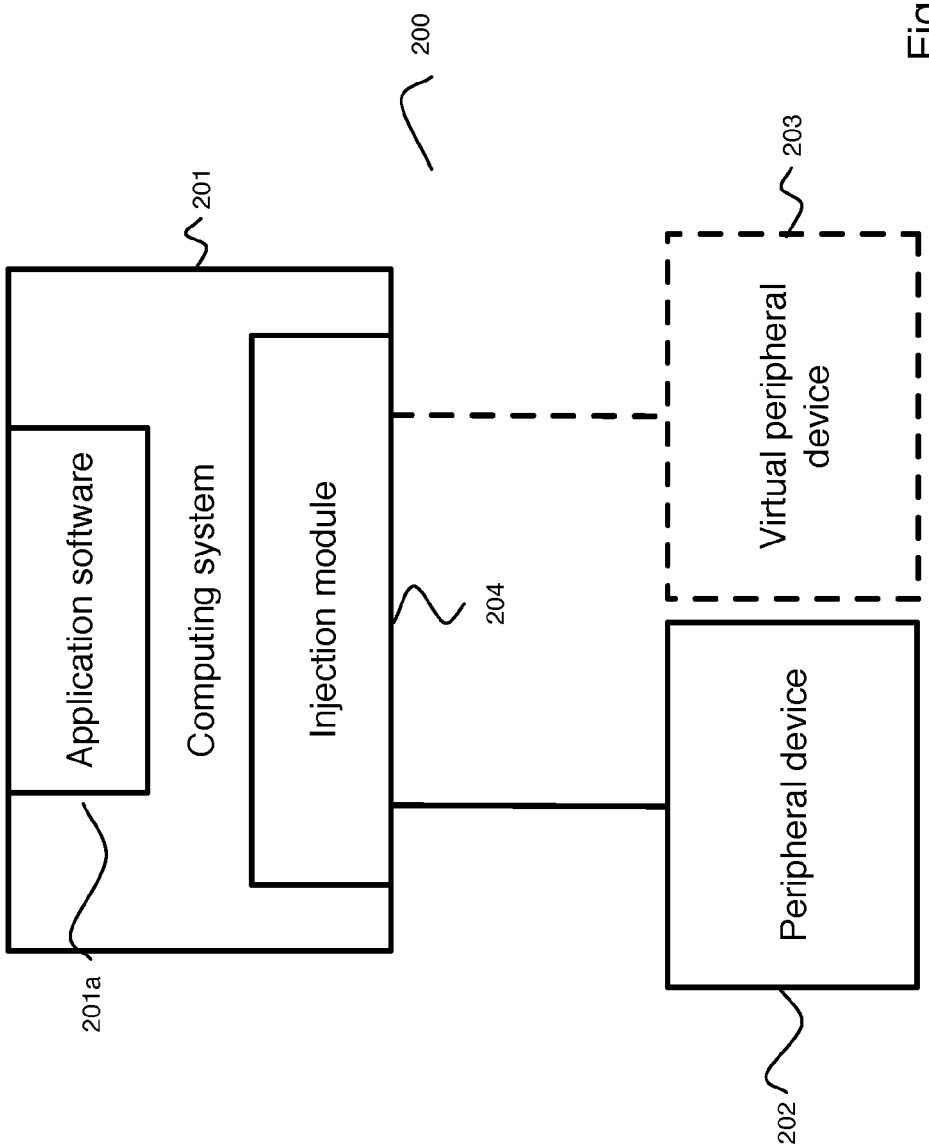

200
201
203
Application software
Computing system
Injection module
Virtual peripheral device
204
Peripheral device
201a
202

Figure 2

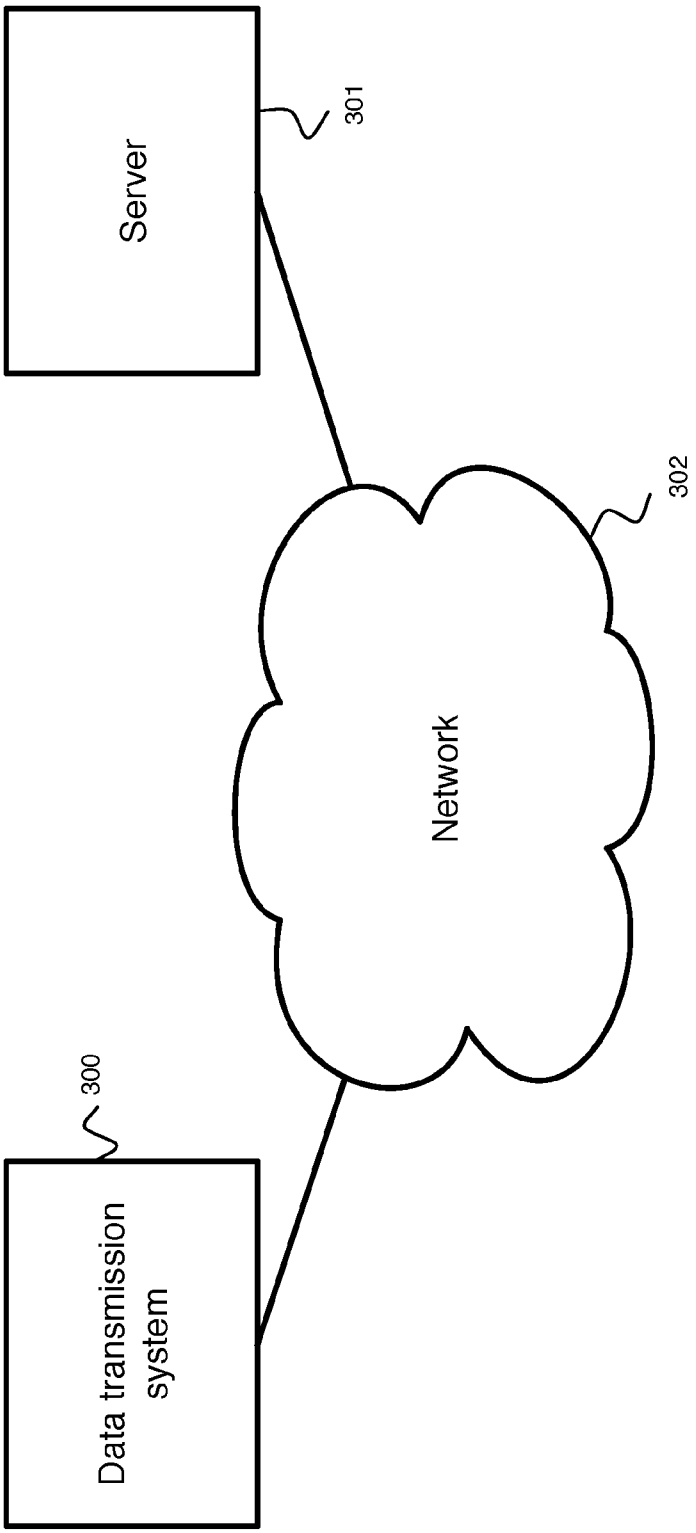


Figure 3

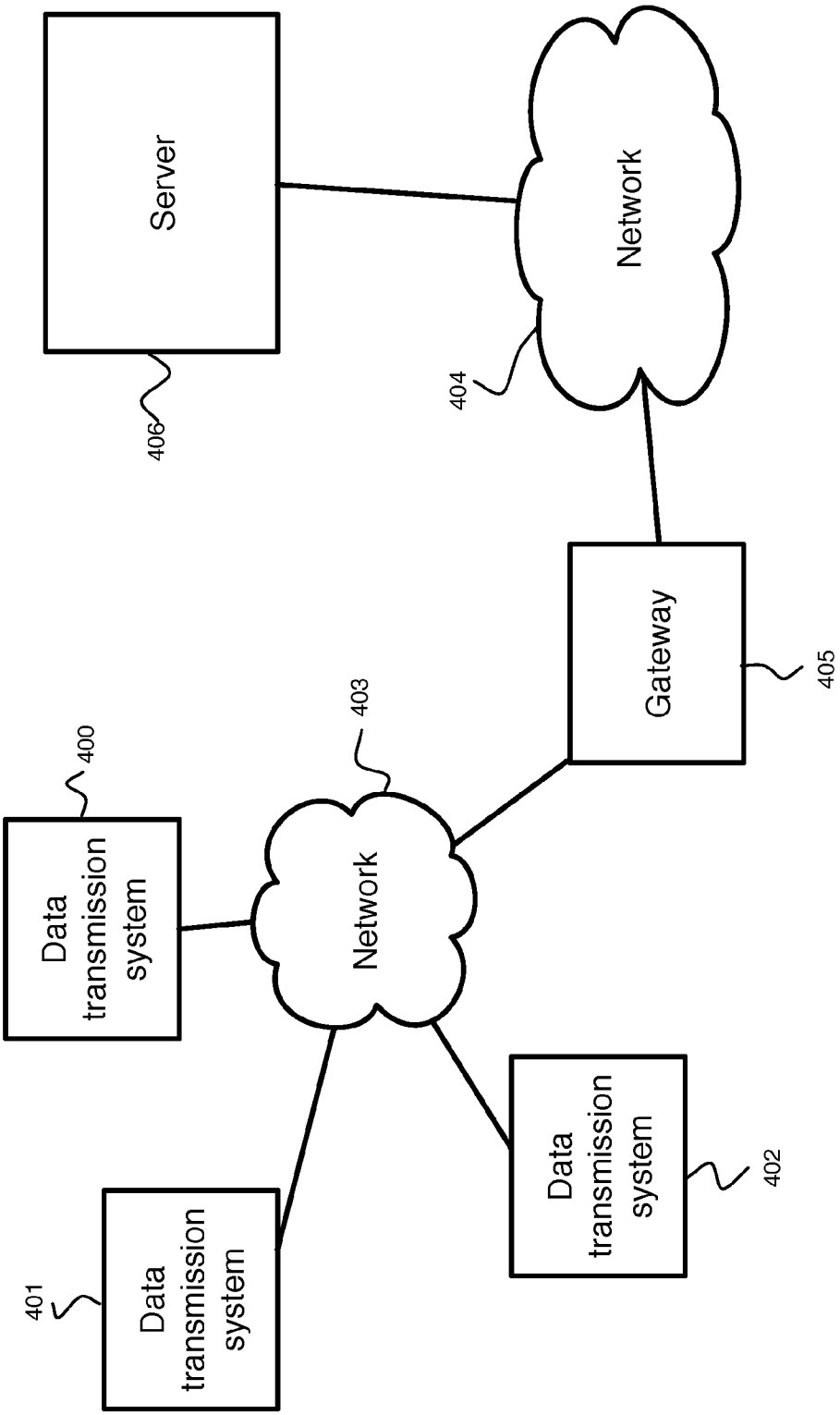

Server
Network
406
404
Gateway
405
403
400
Data transmission system
Network
Data transmission system
402
401
Data transmission system

Figure 4

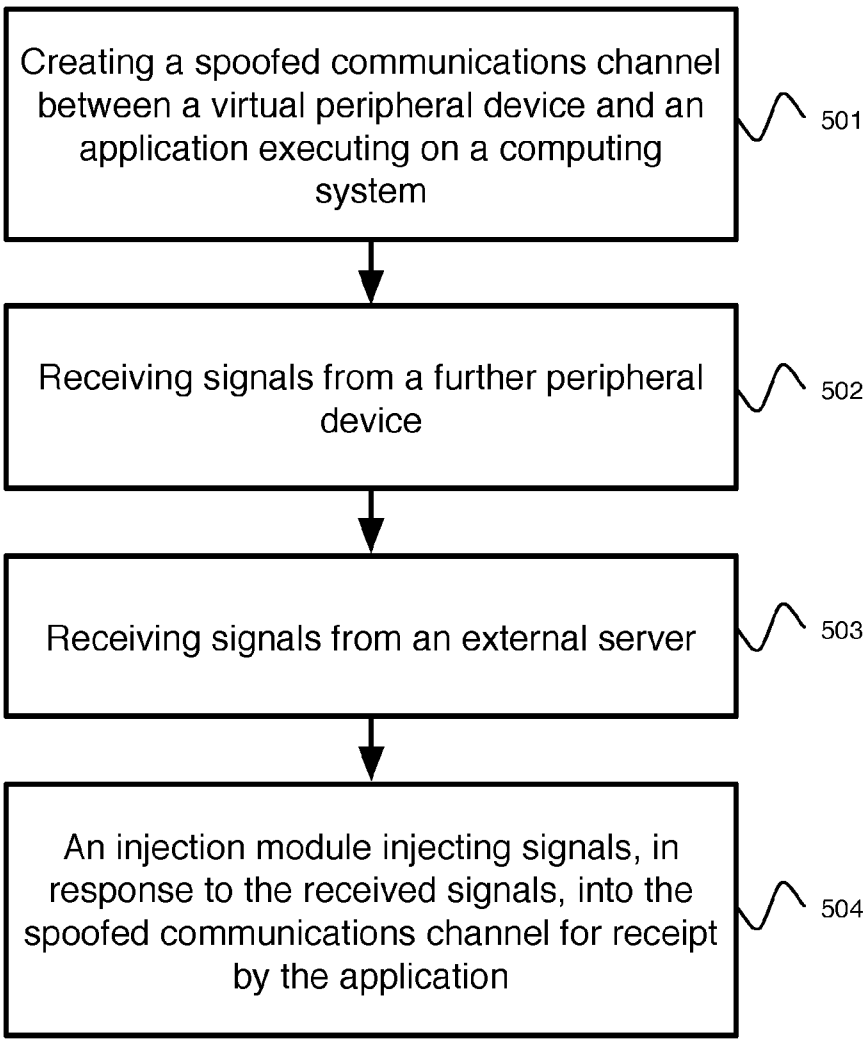

Creating a spoofed communications channel between a virtual peripheral device and an application executing on a computing system
501
Receiving signals from a further peripheral device
502
Receiving signals from an external server
503
An injection module injecting signals, in response to the received signals, into the spoofed communications channel for receipt by the application
504

Figure 5

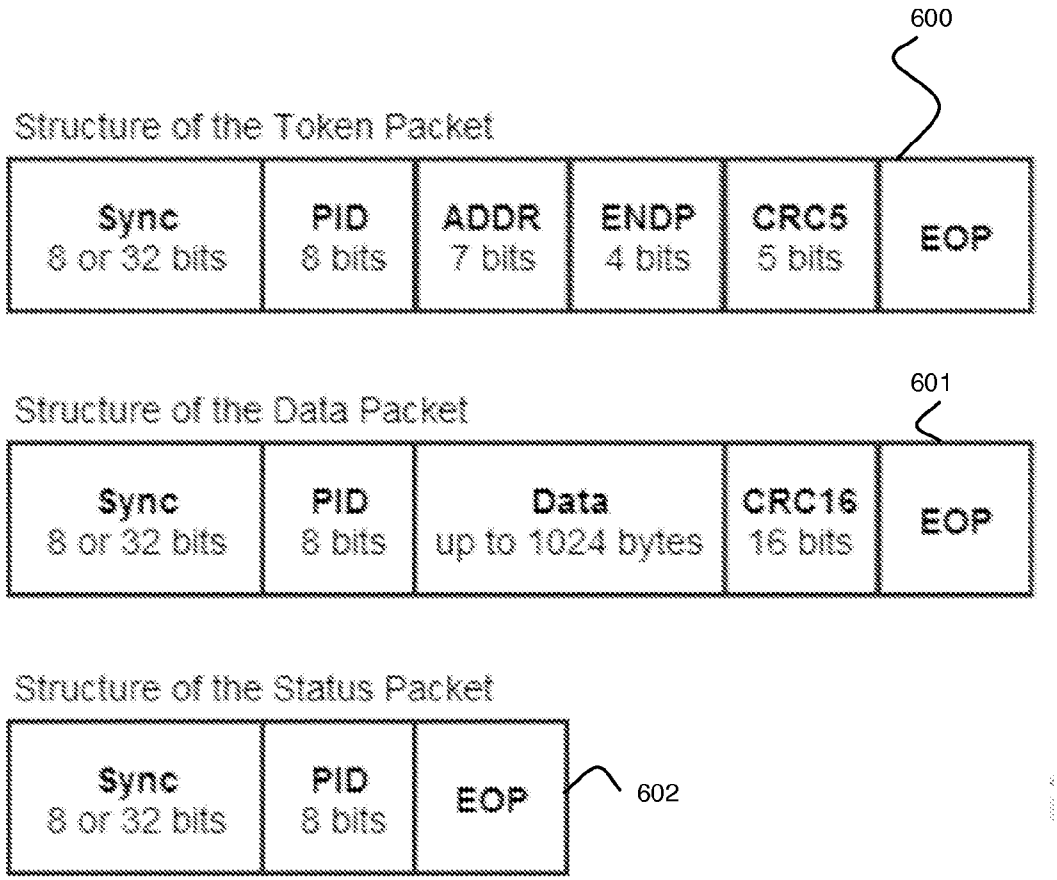

600
Structure of the Token Packet
Sync
8 or 32 bits
PID
8 bits
ADDR
7 bits
ENDP
4 bits
CRC5
5 bits
EOP
601
Structure of the Data Packet
Sync
8 or 32 bits
PID
8 bits
Data
up to 1024 bytes
CRC16
16 bits
EOP
Structure of the Status Packet
Sync
8 or 32 bits
PID
8 bits
EOP
602

Figure 6

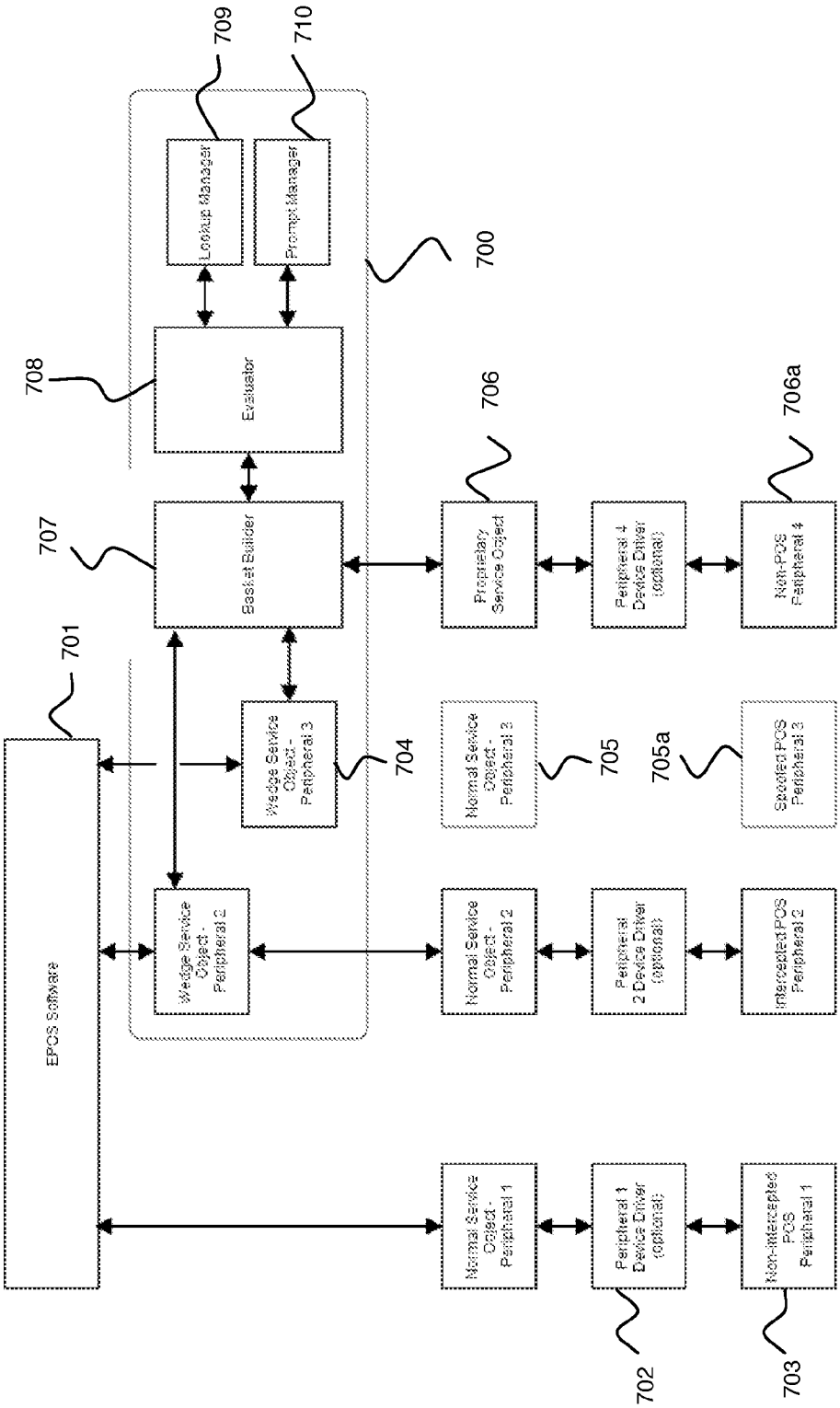

709
710
Lookup Manager
Prompt Manager
700
708
Evaluator
706
706a
707
Basket Builder
Proprietary Service Object
Peripheral 4 Device Driver (optional)
Non-POS Peripheral 4
701
Wedge Service Object - Peripheral 3
704
Normal Service Object - Peripheral 3
705
705a
Spoofed POS Peripheral 3
EPOS Software
Wedge Service Object - Peripheral 2
Normal Service Object - Peripheral 2
Peripheral 2 Device Driver (optional)
Intercepted POS Peripheral 2
Normal Service Object - Peripheral 1
Peripheral 1 Device Driver (optional)
Non-intercepted POS Peripheral 1
702
703

Figure 7

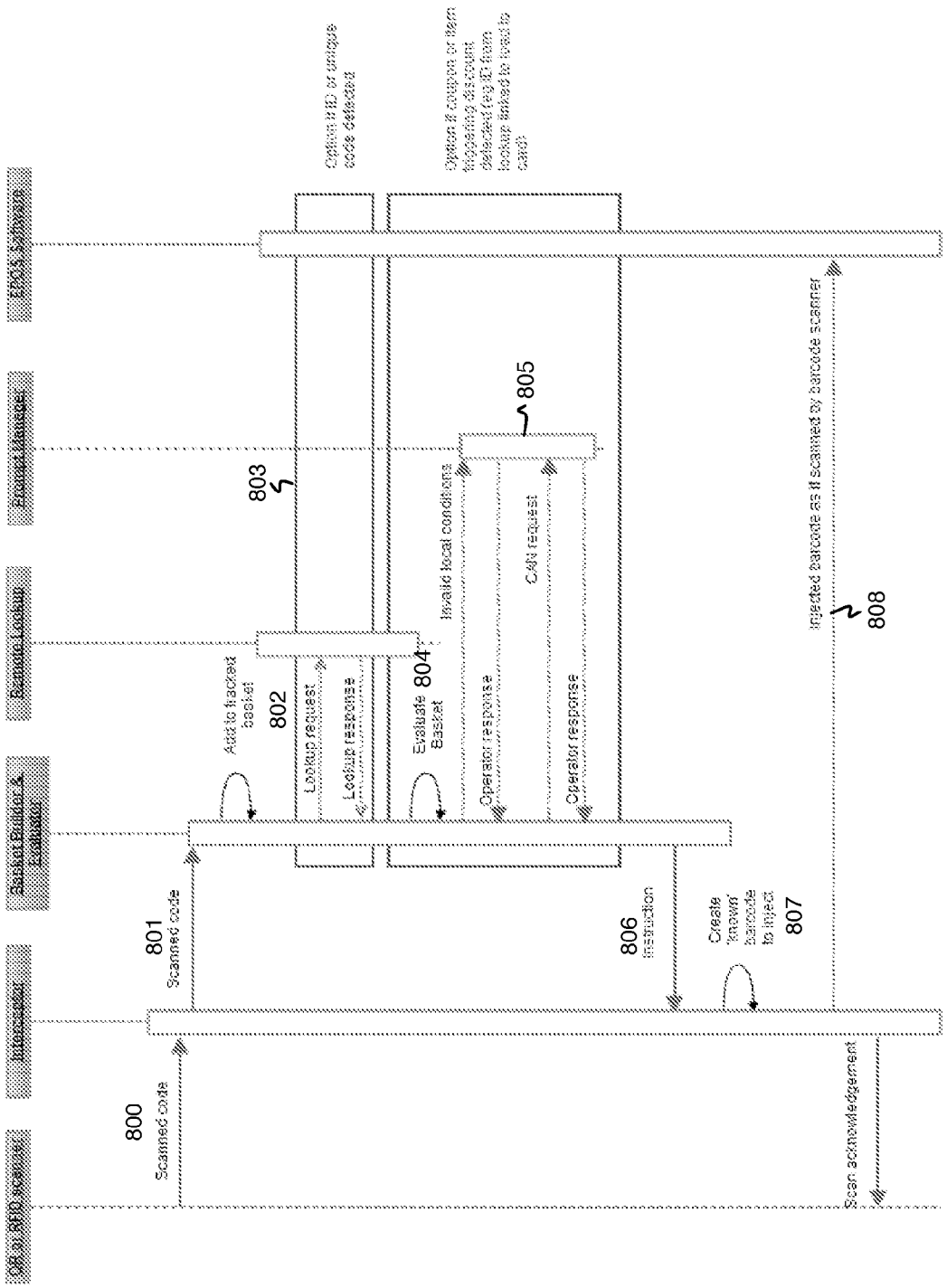

OS or RFID scanner
Interpreter
Basket Builder & Estimate
Remote Lookup
Product Master
EPOS Express
800
Scanned code
801
Scanned code
802
Add to basket
Lookup request
Lookup response
803
804
Evaluate Basket
Invalid local conditions
805
Operator response
CAN request
Operator response
806
Instruction
807
Create known barcode to inject
808
Injected barcode as if scanned by barcode scanner
Scan acknowledgement
Option HID or unique code detected
Option if coupon or item triggering discount detected (eg ID from lookup linked to item to cash

Figure 8

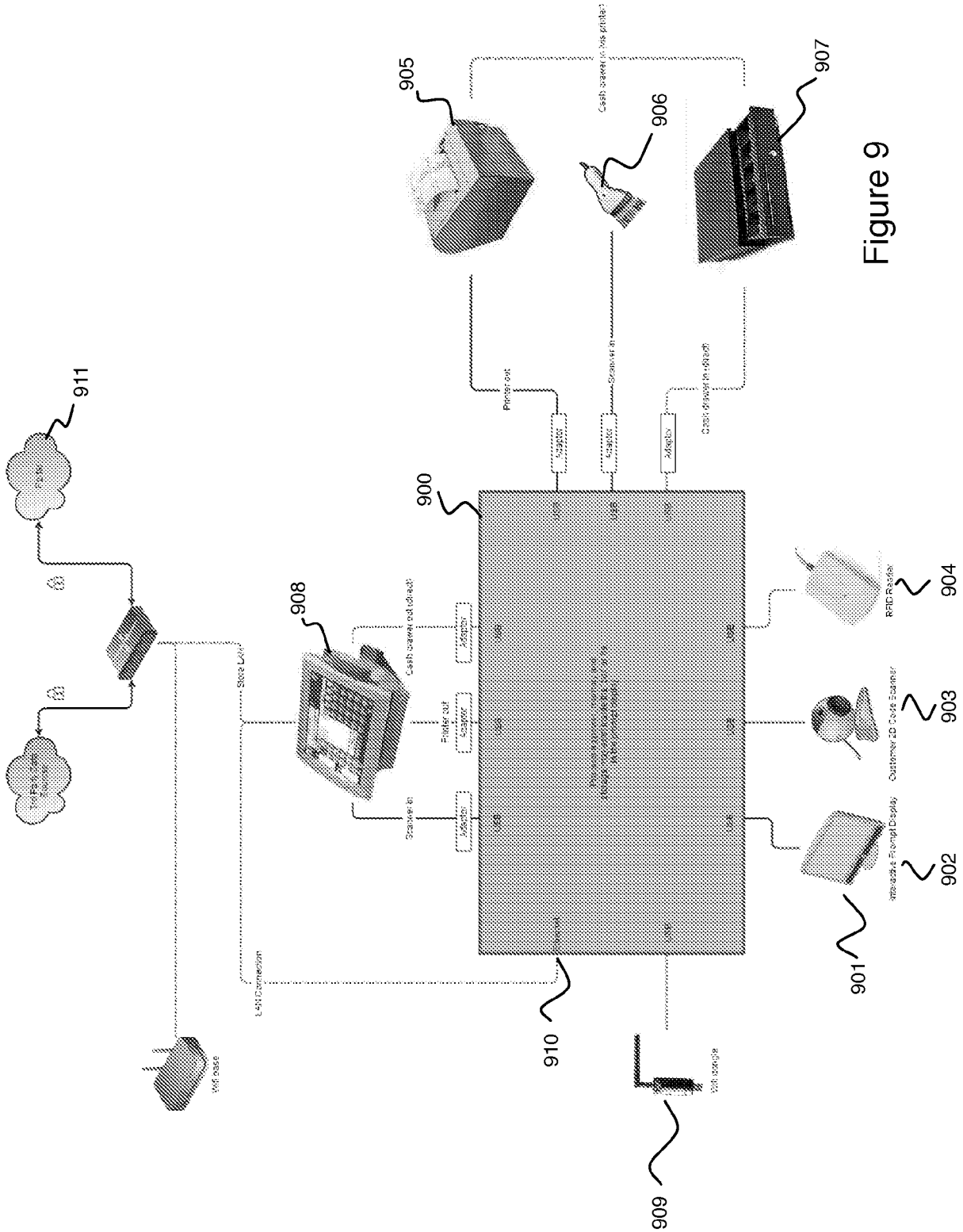
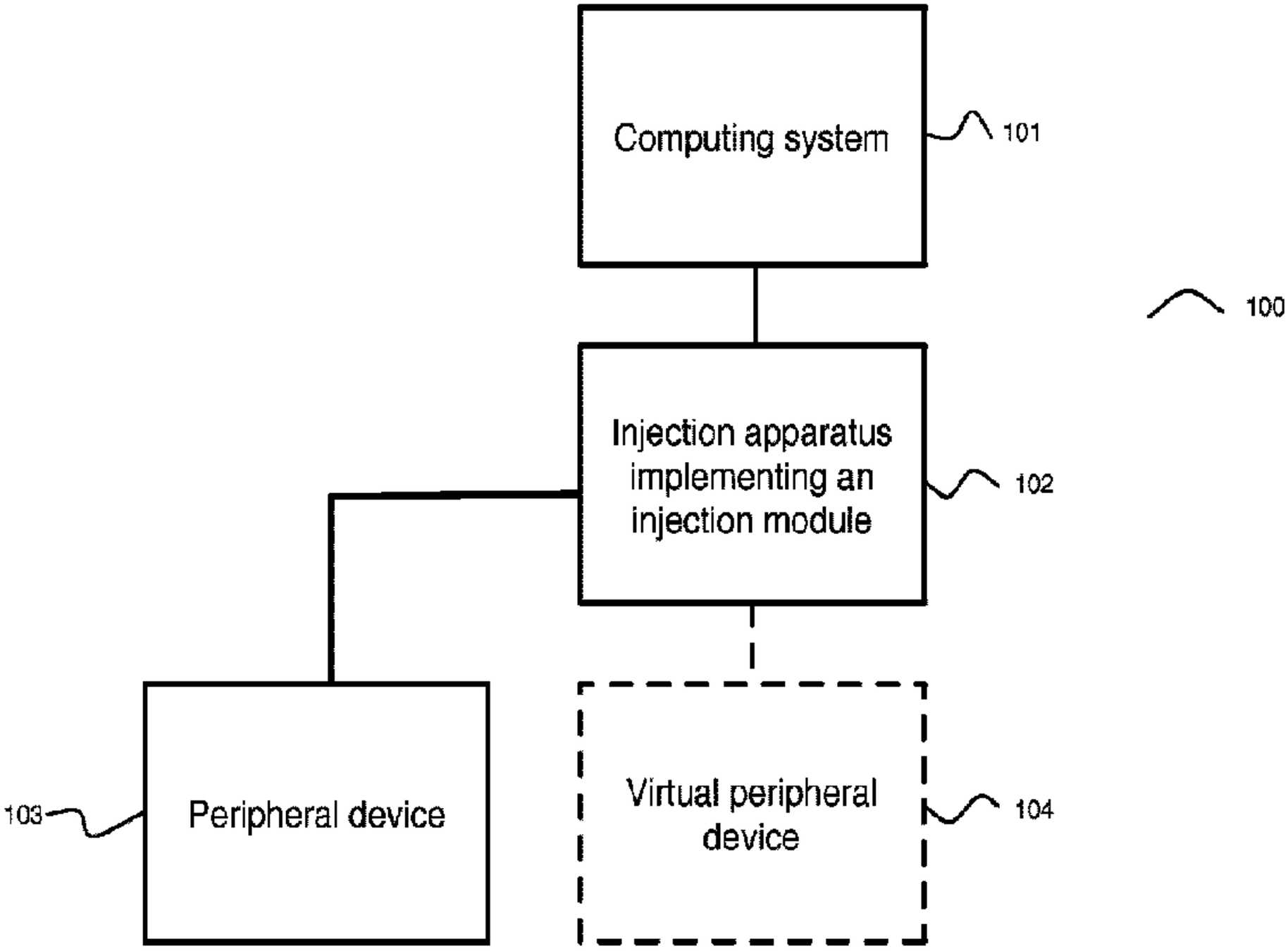


Figure 9


Computing system
101
100
Injection apparatus implementing an injection module
102
Peripheral device
103
Virtual peripheral device
104