



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
61/418,507	12/01/2010		220	RALEP040+		

CONFIRMATION NO. 1493

21912
VAN PELT, YI & JAMES LLP
10050 N. FOOTHILL BLVD #200
CUPERTINO, CA 95014

FILING RECEIPT



Date Mailed: 12/10/2010

Receipt is acknowledged of this provisional patent application. It will not be examined for patentability and will become abandoned not later than twelve months after its filing date. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Applicant(s)

Gregory G. Raleigh, Woodside, CA;

Power of Attorney:

Michael Schallop--44319

If Required, Foreign Filing License Granted: 12/09/2010

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 61/418,507**

Projected Publication Date: None, application is not eligible for pre-grant publication

Non-Publication Request: No

Early Publication Request: No
Title

APPLICATION SERVICE PROVIDER INTERFACE SYSTEM

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international

page 1 of 3

patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and

Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

Doc Code: **TR.PROV**

Document Description: Provisional Cover Sheet (SB16)

PTO/SB/16 (04-07)

Approved for use through 06/30/2010 OMB 0851-0032

U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

Provisional Application for Patent Cover Sheet

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c)

Inventor(s)

Inventor 1

Remove

Given Name	Middle Name	Family Name	City	State	Country i
Gregory	G.	Raleigh	Woodside	CA	US

All Inventors Must Be Listed – Additional Inventor Information blocks may be generated within this form by selecting the **Add** button.

Add

Title of Invention	APPLICATION SERVICE PROVIDER INTERFACE SYSTEM
Attorney Docket Number (if applicable)	RALEP040+

Correspondence Address

Direct all correspondence to (select one):

☒ The address corresponding to Customer Number ☐ Firm or Individual Name

Customer Number 21912

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are:

Entity Status

Applicant claims small entity status under 37 CFR 1.27

- ☐ Yes, applicant qualifies for small entity status under 37 CFR 1.27
- ☒ No

Warning

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

Signature

Please see 37 CFR 1.4(d) for the form of the signature.

Signature	/Michael J. Schallop/			Date (YYYY-MM-DD)	2010-12-01
First Name	Michael	Last Name	Schallop	Registration Number (If appropriate)	44319

This collection of information is required by 37 CFR 1.51. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **This form can only be used when in conjunction with EFS-Web. If this form is mailed to the USPTO, it may cause delays in handling the provisional application.**

Privacy Act Statement

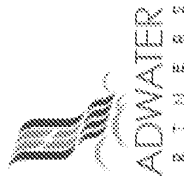
The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that : (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to a n other federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.



Part 1: Motivation and High Level Embodiments



Motivation for Application Developer Services Interface System

There has been a proliferation of wireless applications and application services.

Until now, applications are available to users who pay for a connection service and are billed by an access network carrier for application access usage.

There are application services for which it is beneficial to allow the application service provider (e.g. application developer, web site host, cloud service host, email host, on-line shopping host, ad service host, location service or driving directions service host, M2M service such as vending machine/home power meter/automobile connect/etc, etc) to pay the carrier for some or all of the access services necessary to operate the application service.

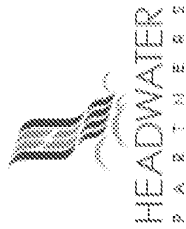
There are also application services for which it is beneficial to allow the application service provider to specify an access service policy and in some embodiments, to also be billed differently for the application access services depending on the access service policies selected by the application services provider.

For large application service provider partners, a carrier may be willing to invest the human resources necessary to negotiate an access service business deal and use the methods previously disclosed by some of the patent authors for creating and publishing the access services required to enable application services providers to specify, pay for and/or control policy for application services (cite previous patent filings??).

When there are many smaller application service provider partners, it is often impractical for the carrier to manually conduct the business processes required to create the access service policies and/or service plans to enable application services providers to pay for and/or control policy for application services. In such cases, an automated Application Services Provider Interface System is valuable to enable many application service providers, and/or device manufacturers, M2M providers, etc to specify, pay for and/or control policy for application services.

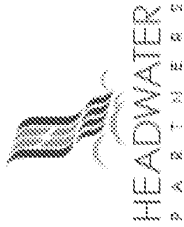
This is one of the objectives of the present invention.

Overview of High Level Embodiments I Through VI



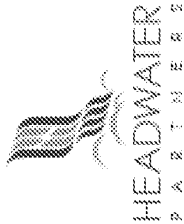
- I. Application Service Provider Interface (ASPI) system embodiments applied to an access network wherein the application services are limited to a restricted set of pre-defined network destinations that are provisioned in the access network gateway apparatus.
- II. ASPI system embodiments applied to an access network wherein the application services are limited to a restricted set of pre-defined network destinations that are provisioned in the access network gateway apparatus and a device Service Processor client is included to provide one or more of the following functions:
 - A. A real time application services status, usage or service option selection notification system for the end user
 - B. Assistance in service usage accounting for application services
 - C. Assistance in service usage transaction support for application services
- III. ASPI system embodiments applied to an access network wherein a set of service policies that allow applications to gain access beyond pre-defined network destinations by provisioning adaptive rules in a proxy server/gateway cloud to enable an application to gain access for service policy conditions that are more sophisticated than simply allowing or blocking access based on a pre-defined list of network destinations.
- IV. ASPI system embodiments applied to an access network wherein a set of service policies that allow applications to gain access beyond pre-defined network destinations by provisioning adaptive rules in a proxy server/gateway cloud in combination with a DAS device Service Processor client is included to provide one or more of the following functions:
 - A. A real time application services status, usage or service option selection notification system for the end user
 - B. Assistance in service usage accounting for application services
 - C. Assistance in service usage transaction support for application services
 - D. Assistance in service usage measurement or service transaction measurement
- V. ASPI system embodiments applied to an access network wherein the network implements a device Service Processor client to implement Device Assisted Services as previously disclosed in prior USPTO patent applications by these patent authors (cite?).
- VI. ASPI system embodiments applied to an access network wherein the network implements a device Service Processor client to implement Device Assisted Services as previously disclosed in prior USPTO patent applications by these patent authors, wherein a third party (e.g. an app store provider and/or an OS system provider) provides an intermediary ASPI function to re-distribute carrier access services provided by one or more carrier networks to application service providers.

ASPI System Process – App Service Provider Side Embodiments



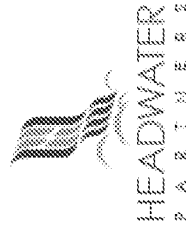
- App service provider acquires ASPI user account log-on – get from carrier or get via automated ASPI process that confirms identity and/or confirms credit or pre-pay payment
- App service provider logs in, applies for credit. If credit is granted, proceed to following steps.
- App service provider uses app service provider SDC UI to select (or design) a service plan policy set for the App Group
 - Select service plan template from available options and fill in the variable service policy parameters in the template (see following discussion on next slide for details).
 - Specify variable entries for service control policy, service charging policy and/or service notification policy.
 - Service plan policy and/or billing options, allowed/disallowed network destinations, service usage limits, effect of network policy state on access control policies and/or billing policies, types of devices to allow for service, application credentials, application authentication processes or procedures, device credentials, device authentication processes or procedures.
 - Network state variables: time of day, network busy state (i.e. network congestion state), foreground/background network conditions, type of network, roaming/home conditions, QoS availability, etc).
 - Optional: Select how to identify services to be associated with the App Group (e.g. specify certain network destinations (e.g. app service provider servers), specify application authentication processes)
- Optional embodiment: app service provider may also use the app service provider SDC to select a placement option (e.g. placement in app store, placement on device, placement on carrier web site, etc.). Premium placement may come at increased cost to app service provider.
- Optional embodiment: Register App credentials so that network and/or device Service Processor can confirm that the proper app software is being used to gain access for the service.
- Optional embodiment: Upload APP (which is pushed into the store and/or devices)
- Optional embodiments: App service provider creates a set of application images that have the access service policies that correspond to information provided via the ASPI and in accordance with the App Group service plan provisioning information provided by the carrier. Examples:
 - Program the application with a set of pre-determined network addresses for the carrier network APNs, gateways or servers
 - Program the application with a set of pre-determined app service provider server destinations
 - Program the application with a set of pre-determined policy controls that correspond with the network state dependent service charging or service control policies selected for the App Group (e.g. software download only during low network busy state or during background service network state, allow or disallow service during roaming conditions, etc)
- App service provider configures app server system to receive usage billing information for each app and/or device that uses the app service provider service, and for the aggregate of all apps/devices that use the app services (the "App Group").
- Carrier programs ASPI system with the rules that determine how app service provider is to pay for service usage.
- Carrier ASPI system provides real time billing status to app service provider for all devices with apps that use the app service provider services (the "App Group"). App service provider is issued an amount of credit and the carrier can elect from multiple billing options to provide to the app services provider. Examples:
 - App service provider receives a certain amount of credit based on automated credit check, pays in real time before this credit is exceeded
 - App services provider must pre-pay and stay ahead of usage (possibly with grace allocation)
- When total charges for app service provider App Group goes beyond credit limit, the app services can be shut off, suspended, or throttled back (perhaps after a series of warnings are sent to the app services provider via the ASPI and/or other communication means)
 - App services provider pays bills via electronic transfers or pays in advance via other means.
- App services provider business model creates revenue via one or more of the following:
 - Charge user for services
 - User ad viewing
 - User purchase transactions
 - Other embodiments will now be obvious to one of ordinary skill in the art.
- Optional - monitor and respond to possible fraud events that come from Service Controller, fraud detection system
 - App service provider portal includes simplified UI for selecting fraud detection trigger events
 - Usage caps, caps per unit time, rate of usage, app destination white/grey/black lists, good customer feedback, etc.
- Optional - provide billing information to aggregated carrier billing engine/store for monthly service, pre-pay service and/or transaction events
- Optional - Provide customer usage feedback information to Service Controller
 - Optional - Need data base to associate app/device pairs (or user) with transactions and customer usage feedback or transaction feedback information
 - E.g. Certain credentials (MDN,IMEI, ESN, MEID, SIM, user id, etc) is included in transaction information from device/app pair and the same credentials are used to identify usage information reported to app service provider

ASPI System Process – Carrier Side Embodiments



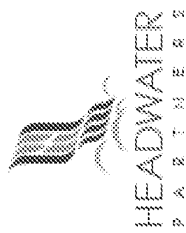
- Carrier personnel uses a Carrier SDC UI to define service plan template options (control, charging, notification, - include network state policies such as time of day, network busy state, type of network, home/roaming, background/foreground availability, QoS availability).
- There are often certain aspects of the service plan template options policy sets that are not pre-determined before the app service provider (or app credentials, app approved destinations, app approved service usage activities, etc) are known - these variable (e.g. not pre-configured) service plan policy parameters are not pre-configured because they are programmed by the app service provider when the app service provider creates the service with the ASPI. In many embodiments, the carrier uses the Carrier SDC UI to specify most of the access control, network state, charging and notification policies and only leaves a subset of the parameters needed to define the service plan policy set as variable service plan policy parameters.
- Example embodiments for variable service plan policy parameters:
 - Service plan policy and/or billing options, allowed/disallowed network destinations, service usage limits, effect of network policy state on access control policies and/or billing policies, types of devices to allow for service, application authentication credentials, application authentication processes or procedures, device authentication processes or procedures,
 - Network state variables: time of day, network busy state (i.e. network congestion state), foreground/background network conditions, type of network, roaming/home conditions, QoS availability, etc).
- The ASPI SDC UI is then pre-configured with the service plan templates. The variable service plan policy parameters are then programmed into the ASPI UI as "fill in" options for the application service providers to specify at the time they log into the ASPI and select their access service policies.
- The ASPI UI is, in some embodiments, a secure web services UI and the app services provider must obtain a log in credential from the carrier or create one to log in and use the ASPI.
- When an app provider logs into the ASPI and applies for credit, the carrier credit check system performs a credit check on the app service provider entity's ability to pay. The carrier system can then either issue a credit through the ASPI UI, or may elect to inform the app service provider entity that they must pre-pay and stay ahead of service usage billing to continue service.
- Once the app service provider entity is approved for credit and specifies the variable service plan policy parameters, then the SDC uses the service plan policy template selected by the application service provider, with the specified entries for the variable service plan policy parameters to create a network (and in some embodiments device service processor) provisioning file (the "App Group Provisioning File") that specifies how each element in the network (and in some embodiments device service processor) is to be provisioned for all application services that use the services that belong to the App Group.
- The SDC then passes the App Group Provisioning file to the network provisioning system (and in some embodiments the Service Controller) to program the network service policy enforcement rules to enable the App Group services.
- A portion of the App Group Provisioning File that is created by the SDC specifies the charging record identifiers that are to be programmed into the network (and/or in some embodiments the device) generated Charging Data Records (CDRs) so that the network charging system may associate the App Group service usage with the App Group account by generating CDRs with the proper charging identifiers, and so that the CDRs thus generated and passed to the billing system may then be properly rated and associated with the app service provider entity's billing account.
- The Service Controller (or other suitable network elements such as the network charging system or the billing system) may also be programmed in some embodiments to transmit real time or slightly delayed service usage, content transaction or service billing information through the ASPI to the application service provider servers. As discussed above, it is often the application service provider's responsibility in some embodiments to ensure that sufficient credit is in their account to pay for service charges accrued by their App Groups.
- If the app developer falls behind on service credit compared to App Group service usage billing, then the Service Controller (or some other suitable network element) can issue a throttle, suspend or stop service policy instruction to the network provisioning apparatus, to the network gateways or to the network home agent or to other suitable network elements specific to any given carrier or network.
- In some embodiments the carrier desires to make sure that a service shut down policy is not deemed as a bad carrier service, network service or device experience by the end user. In such cases a device notification client may be included wherein the user is informed of the service capability status of one or application services that are offered to the device user. Notification messages may include one or more of: this application service is not available at this time due to a condition with the application service provider, this application service is not available at this time due to the service policies employed by the application service provider, this application service is available at this time, this application service has the following service options for you to choose from, etc. The application service notification agent may also include service usage allowance or remaining balance or usage information. The application service notification client may also provide the user with contact information for the application service provider entity responsible for the application so that the user does not call help resources of the carrier.

Example Service Plan Policy Template Options for App Service Provider



- Bill (/control/notify) usage by device
- Bill (/control/notify) aggregate usage for all devices
- Bill (/control/notify) differently based on network state (network type, roaming/home, NBS/TOD, background status, QoS status, geography, etc)
- If background service is chosen by app service provider (or other network state dependent service definition), then there are a couple of choices on how to manage background service control and billing:
 - Option 1: cut off or throttle app when background is not allowed – app has to deal with that without API signal
 - Option 2: cut off or throttle app when background is not allowed – app receives API signal, app has to deal with this
 - Option 3: app is told that background service is either allowed or not allowed, app is responsible to not transmit when background service is not allowed, if app violates this rule then app service provider is charged at a higher rate for non-background data
 - This can all be generalized to network state dependent services – not just background services
- All designed by carrier personnel with carrier SDC UI and provided as options to app services provider via ASPI UI (aka App Service Provider SDC UI)

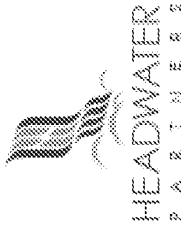
ASPI System Process – Additional ASPI Optional Embodiments



- Provide aggregated/centralized billing and/or transaction system - establish billing rules for the consumer side of the aggregated billing engine
- Provide aggregated app/content store
- App service provider uses analytics to optimize device experience, Ad service sales opportunities
- App service provider can choose app placement & pay accordingly
- App service provider can choose to use centralized billing functions
 - Bill on behalf of
 - Bill with main bill for carrier services

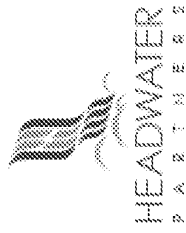
Part 2: Descriptions of Embodiment Elements for the 4 Basic Approaches

High Level Embodiment I. ASPI System With Network Destination Path Control And No Device Service Processor Client



- App service provider portal for credit check & plan selection
- Network address provisioning (pre-defined IP address, host name, etc.)
- Application address provisioning (pre-defined IP address, host name, etc.)
- Billing rate engine limited to portal configuration (plan selection)
- app service provider pays for everything that goes to their address (not just APP traffic, no APP awareness)
- Drawbacks:
 - No general purpose Internet access
 - No sponsored search
 - No add injection
 - No NBS awareness and rating (at least less awareness-harder to do)
 - Centralized/scaling issues
 - Roaming issues
 - Different network issues (2/3/4G, and WiFi)
 - Network box hardware roadmap and service time to market issues

System Operation –Network Controlled Destination Without Device Service Processor Client



- Login to ASPI, confirm credit, create App Group and select authentication options, select app service provider service plan set
- Enforce Policy set
 - Provision allowed network destinations into the core network
 - Provision charging records
 - Carrier can over-rule app policies depending on policy state variables (e.g. TOD, NBS, active networks, background traffic, etc.)
- Fraud check
 - Monitor service usage to defined destinations and look for over usage, over usage given good customer feedback, or usage patterns being inconsistent with expected service usage
 - Send fraud flags to app service provider
 - Or, send usage to app service provider and let them determine fraud events -- claim is for fraud checker in app service provider network
 - Surf-out usage exceeds surf-out usage limits
- Report usage stats & billing info to app service provider
 - app service provider pays to keep services going
 - When services shut down, notification system must notify user that only this App is shutdown and not all others services on the device, e.g. other Apps
 - app service provider monitors for proper device service usage

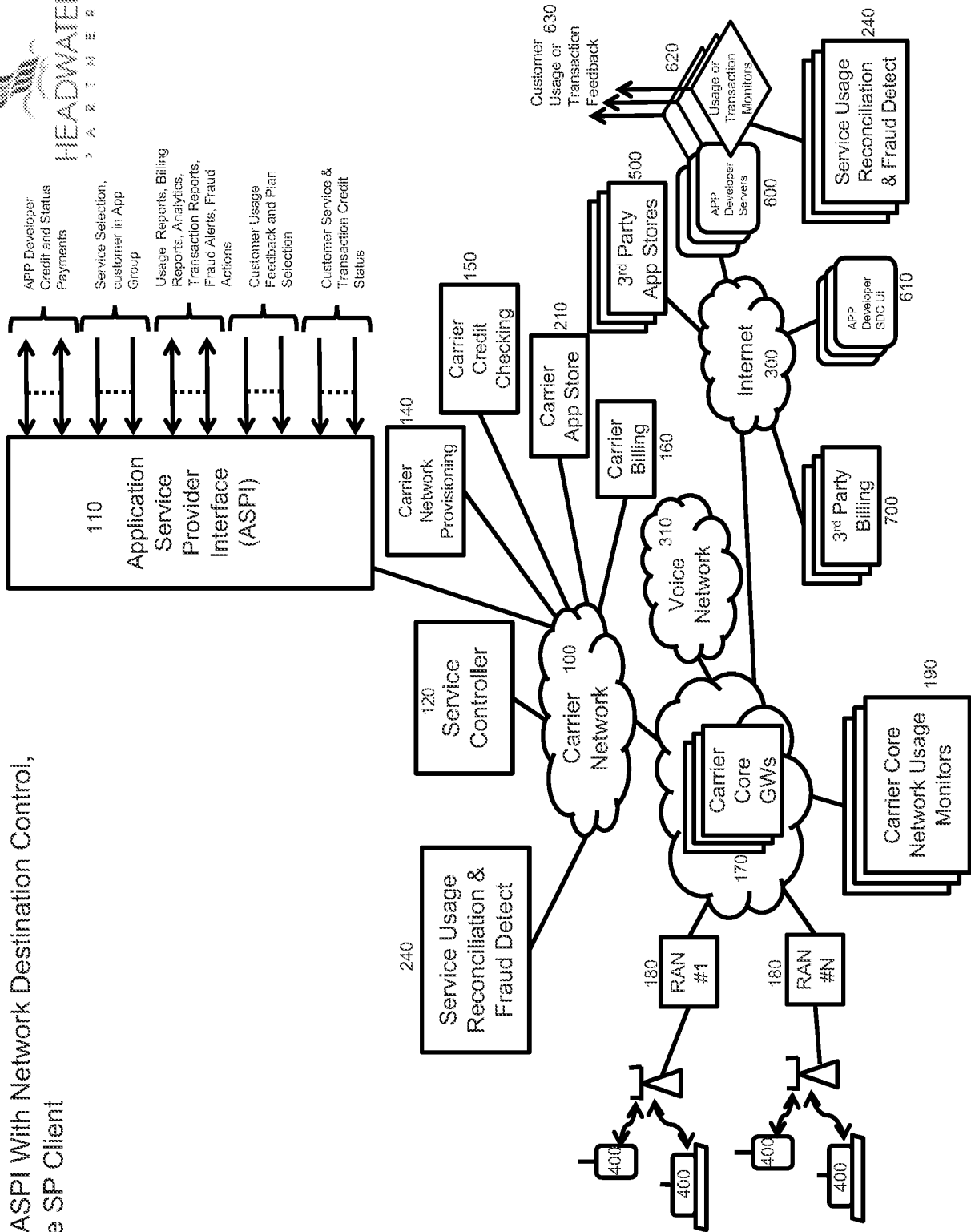
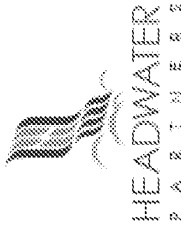


Figure 1: ASPI With Network Destination Control,
No Device SP Client

High Level Embodiment II. ASPI System With Network Destination Path Control And Device Service Processor Client



- * App service provider portal for credit check & plan selection
- * Assign unique gateway/proxy server flows to APP
 - * Unique APN with SSL
 - * Secure with fraud reconciliation
 - * Unique tagged traffic flow
 - * Tagged (e.g. header) & secured by APP
 - * Service include fraud reconciliation
- * Billing rate engine limited to portal configuration (plan selection)
- * app service provider can pay only for APP traffic as APP can go anywhere
- * Need to have secure login/authentication from APP to GW/Proxy server
- * Could set up APP API in proxy server to inform APP of service status and/or allow APP access to services
- * Drawbacks
 - * No real time device client for notification and service plan selection
 - * Less network busy state awareness and rating on device
 - * Centralized/scaling issues
 - * Roaming issues
 - * Different network issues (2/3/4G, and WiFi)
 - * Network box hardware roadmap and service time to market issues

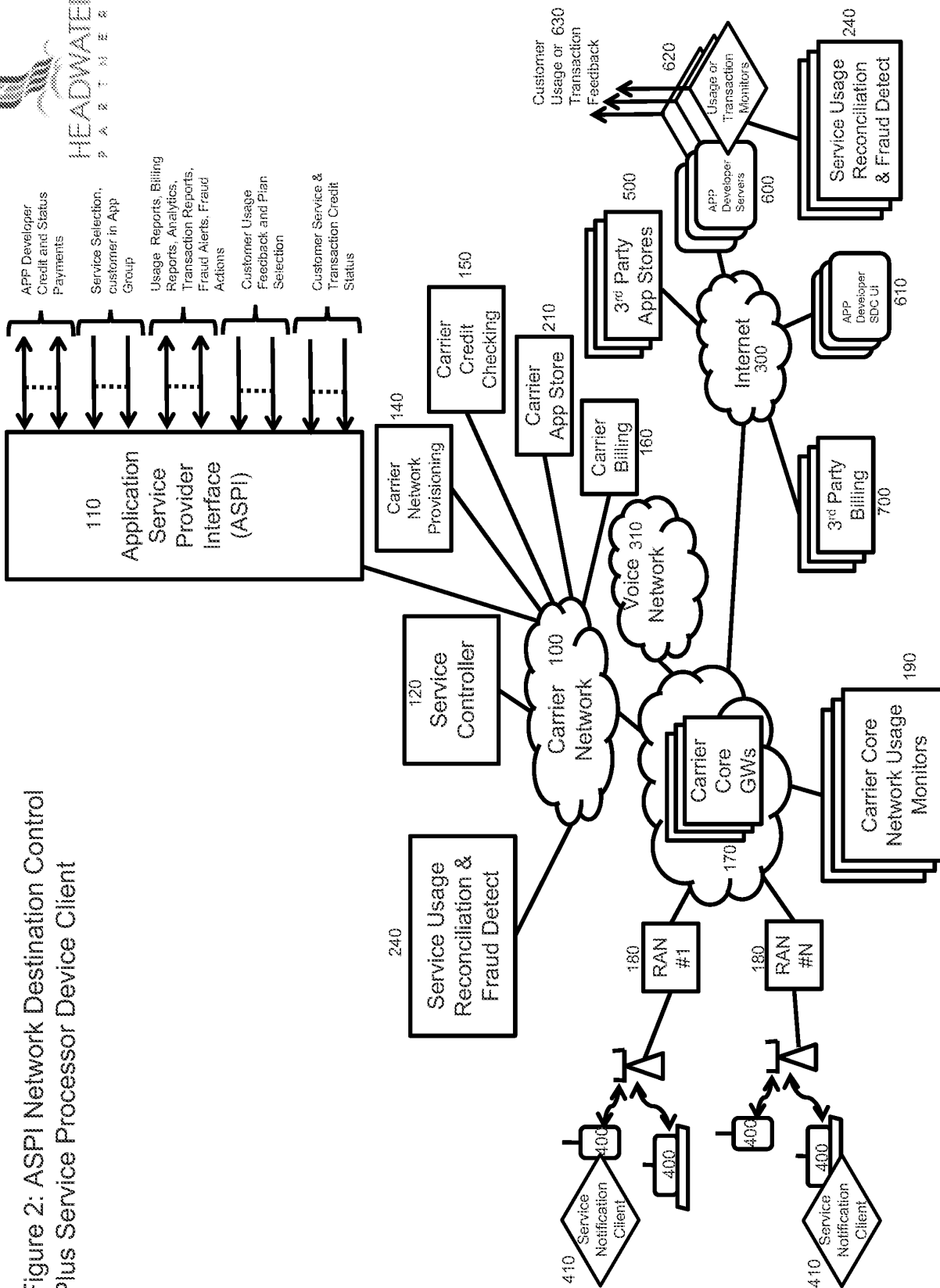


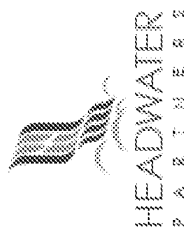
Figure 2: ASPI Network Destination Control Plus Service Processor Device Client

High Level Embodiment III. ASPI System With Proxy/GW Server And No Service Processor Client



- Service Controller and/or network provisioning apparatus can map app service provider service plan template choices and variable service policy parameters entered through ASPI into access control and service usage accounting policies in proxy server/gateway cloud traffic processing elements
- App service provider can specify a service plan that allows the app to go to destinations that are less limited than with strict network destination control
 - e.g. Use previously disclosed USPTO embodiments on associative traffic for apps, surf-out for apps, customer usage and/or transaction feedback (“good customer feedback”), etc
- App can have secure login/authentication to GW/Proxy server
- Can set up APP API in proxy server to inform APP of service status and/or allow APP access to services
- Drawbacks
 - No real time device client for notification and service plan selection
 - Less network busy state awareness and rating on device
 - Centralized/scaling issues
 - Roaming issues
 - Different network issues (2/3/4G, and WiFi)
 - Network box hardware roadmap and service time to market issues

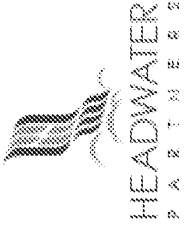
Embodiments on Who Owns the Proxy and Resulting Carrier Network Provisioning Via ASPI



Carrier can own proxy cloud and programs via ASPI

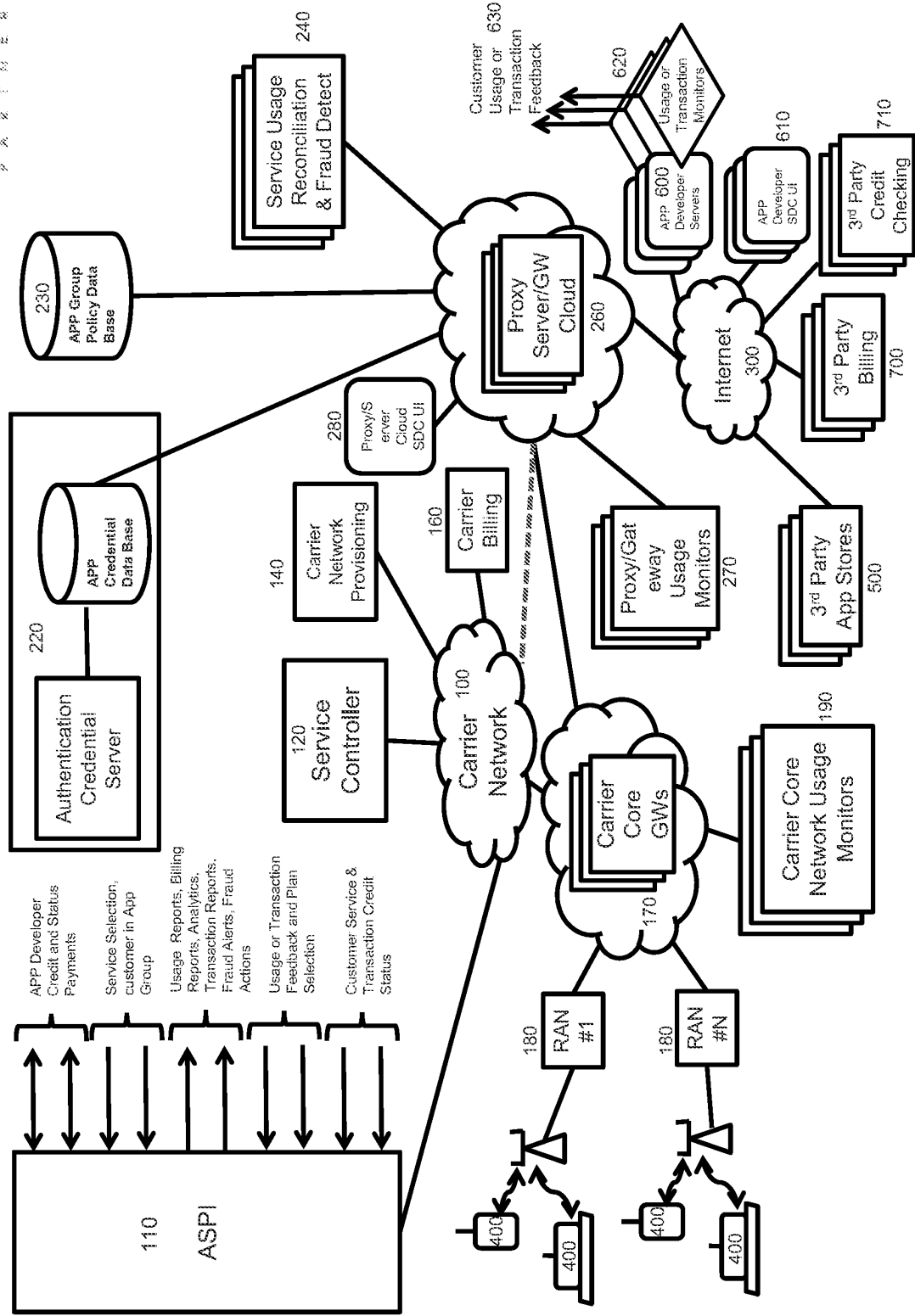
Developer can own proxy server and programs only path to
proxy through ASPI

System Operation – Proxy With No Service Processor Client

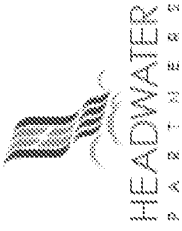


- Login to ASPI, confirm credit, create App Group and select authentication options, select app service provider service plan set
- Assign and provision unique proxy server/gateway routes for app, provide destination pointer to app service provider, app service provider compiles pointer into app
- Provision proxy server/gateway with app service plan policies and/or billing plan policies
 - Unique APN (optionally with SSL-like encryption)
 - Secure with fraud reconciliation
 - Optional app authentication process for proxy server/gateway (e.g. individualized credentials for each app -- e.g. sw license #)
 - Unique tagged traffic flow
 - Tagged (e.g. header) & secured by APP
 - Proxy server could also be controlled by app service provider
 - Provision proxy server with fraud detection flag parameters
 - Carrier can over-rule app policies depending on policy state variables (e.g. TOD, NBS, active networks, background traffic, etc.)
 - Proxy server can determine what links to highlight and what links to de-emphasize
- Enforce Policy set
 - In proxy server/gateway
- Fraud check
 - App authentication error (if app authentication is used)
 - Exceed usage caps (program app to stop or tell app to stop when it keeps going)
 - Usage caps exceeded without good customer feedback
 - Usage behavior does not match expected behavior
 - Surf-out usage exceeds surf-out usage limits
- Report usage stats & billing info to app service provider
 - app service provider pays to keep services going
 - When services shut down, notification system must notify user that only this App is shutdown and not all others services on the device, e.g. other Apps
 - app service provider scans for proper device service usage

Figure 3: ASPI With Proxy Server/Gateway And No Service Processor Client

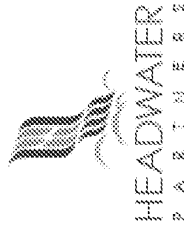


High Level Embodiment IV. ASPI System With Proxy/GW Server And Service Processor Client



- * Assign unique gateway/proxy server flows to APP
 - * Unique APN with SSL
 - * Secure with fraud reconciliation
 - * Unique tagged traffic flow
 - * Tagged (e.g. header) & secured by APP
 - * Service include fraud reconciliation
- * Service Controller and/or network provisioning apparatus can map app service provider service plan template choices and variable service policy parameters entered through ASPI into access control and service usage accounting policies in proxy server/gateway cloud traffic processing elements
- * App service provider can specify a service plan that allows the app to go to destinations that are less limited than with strict network destination control
 - * e.g. Use previously disclosed USPTO embodiments on associative traffic for apps, surf-out for apps, customer usage and/or transaction feedback ("good customer feedback"), etc
- * App can have secure login/authentication to GW/Proxy server
- * Can set up APP API in proxy server to inform APP of service status and/or allow APP access to services
- * APP on-device API (APP does not need to reach out to proxy for API)
- * Secure APP credential check
- * Notification agent for app services
- * Can do much of Full DAS functions except the following
 - * Lots of chatter traffic between DAS client and proxy
 - * Centralized solution/scaling issues
 - * Roaming issues
 - * Different network issues (2/3/4G, and WiFi)
 - * Network box hardware roadmap and service time to market issues
 - * Notification sequences can be long unless notification policy enforcement is fully under client control

Basic Embodiments for Service Processor Client And Proxy/Gateway Communications and Policy Enforcement



Client provides for notification connection to inform user of proxy server/gateway traffic control actions, to provide user with description of service plan configuration and capabilities, to provide user with service selection platform, to provide user with options to upgrade/downgrade/acknowledge actions or notifications, to provide user with real time usage and/or billing status, etc.

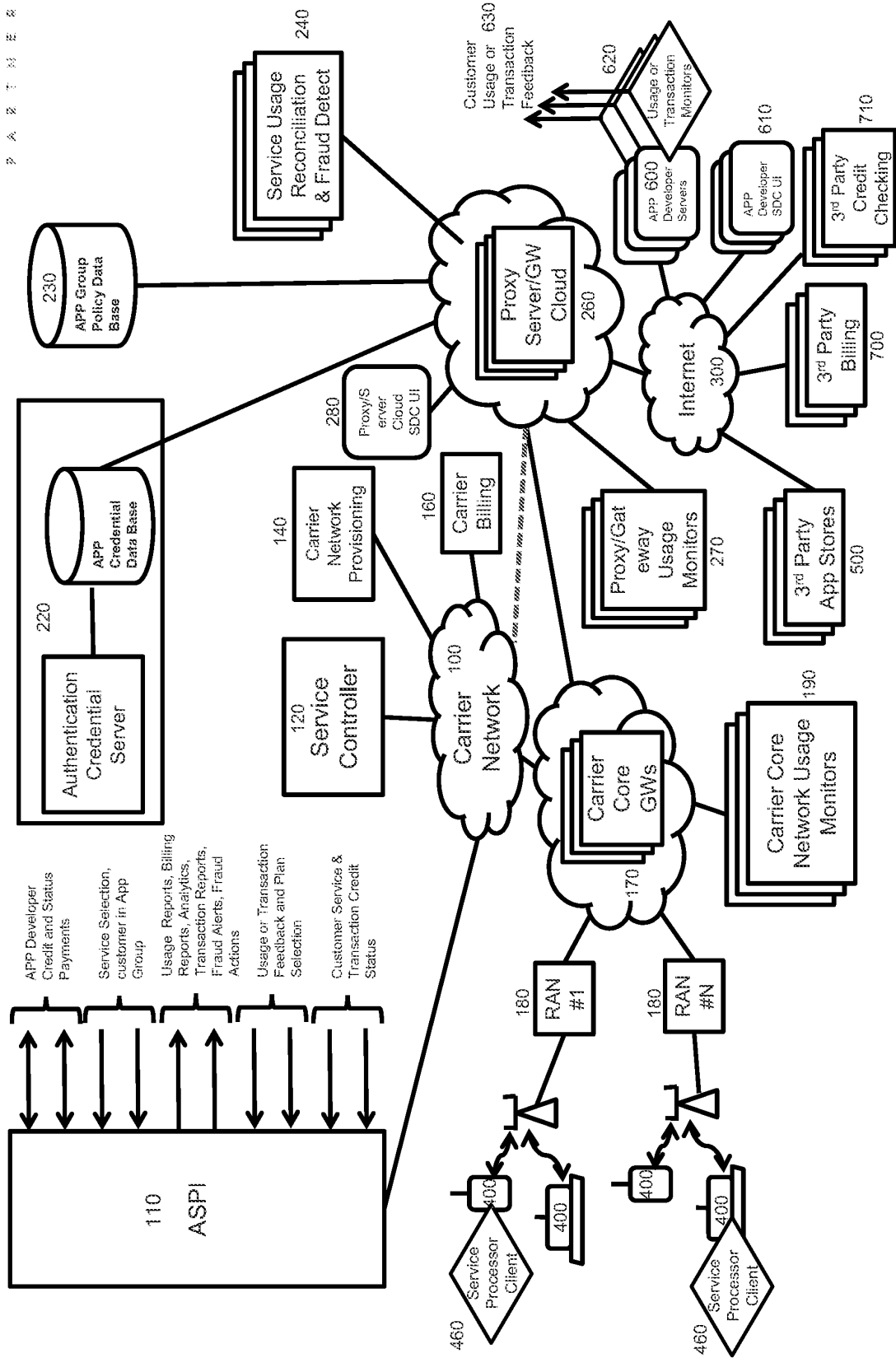
- * Options for gateway and client communications link management and programming:
 - Gateway sends service activity enforcement information messages directly to client
 - Client sends responses directly to gateway
 - Gateway sends enforcement information messages to Service Controller that then formats gateway message into user notification message and sends messages to client
 - Client sends responses to Service Controller which then formats responses into new gateway service policy commands
 - Service controller formats information messages to client UI and converts client selection choices into new gateway service policy commands

Carrier can own proxy cloud and programs via ASPI

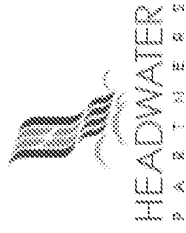
Developer can own proxy server and programs only path to proxy through ASPI

Service Processor client can also perform application credential check and identity confirmation function to ensure that app that is receiving application specific access services is the correct app version and is not another app fraudulently seeking access service (see embodiments for confirming app credentials/identity)

Figure 4: ASPI With Proxy Server/Gateway And Service Processor Client

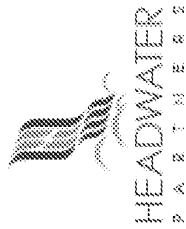


High Level Embodiment V. ASPI System With Full DAS Service Processor Client



- * App service provider portal for credit check & plan selection
- * APP embeds policy rules for
 - * Access control
 - * Charging (charged to; user account, app service provider or APP sponsor)
 - * Notification UI messages
- * DAS performs secure APP credential check
- * DAS verifies (crosschecks) APP policies against carrier established policies
- * DAS tracks APP service usage
- * DAS performs access control
- * Fraud detection performed using
 - * DAS based usage measure against APP usage measure
 - * NAS based usage measure against APP usage measure
 - * DAS & NAS based usage measures against APP based usage measure
- * DAS APP API to provide network states; NBS, TOD, QoS, active networks (2/3/4G, WiFi), background traffic, etc. for optimum APP usage rating
- * DAS Provides Analytics to app service provider in exchange for Ad services placement or for a fee
- * Flexible billing
 - * Carrier billing consolidation
 - * app service provider billing
 - * APP sponsored Billing
- * Can provide advanced service plans, access control, usage charging, notification on roaming networks
 - * Addition of previously disclosed USPTO secure hardware DDR embodiments strengthens fraud prevention in this case

System Operation – ASPI With DAS Approach



- * Login to ASPI, confirm credit, create App Group and select authentication options, select app service provider service plan set
- * Upload App credentials to SC (carrier network database)
- * Download App credentials to SP (device)
- * Check App credentials
 - * At load time
 - * At run time
 - * Ongoing
- * Service Processor/Service Enabled OS: Associate App credentials with service policy set
 - * Report credentials to server & get policy set
 - * Get policy set from pre-stored data base
 - * Get Policy set from 3rd party, e.g. app service provider server or app service provider certificate
- * Enforce Policy set
 - At Device
 - At Device & Network
 - At Network
- * Fraud check
 - * Check with standard DAS techniques in SC
 - * App service provider selects fraud flags then push fraud flags to app service provider, push usage to app service provider and app service provider determines fraud flags (e.g. usage limits exceeded, usage patterns, good customer behavior)
 - * Network CDR/FDR/RTR reconciliation
 - * SP report reconciliation
 - * Secure DDR reconciliation
 - * Secure SP
 - * App reporting
- * Report usage stats & billing info to app service provider
 - * App service provider pays to keep services going
 - * When services shut down, notification system must notify user that only this App is shutdown and not all others services on the device, e.g. other Apps
 - * App service provider monitors for proper device service usage

Figure 5: ASPI With Full DAS
Service Processor Client

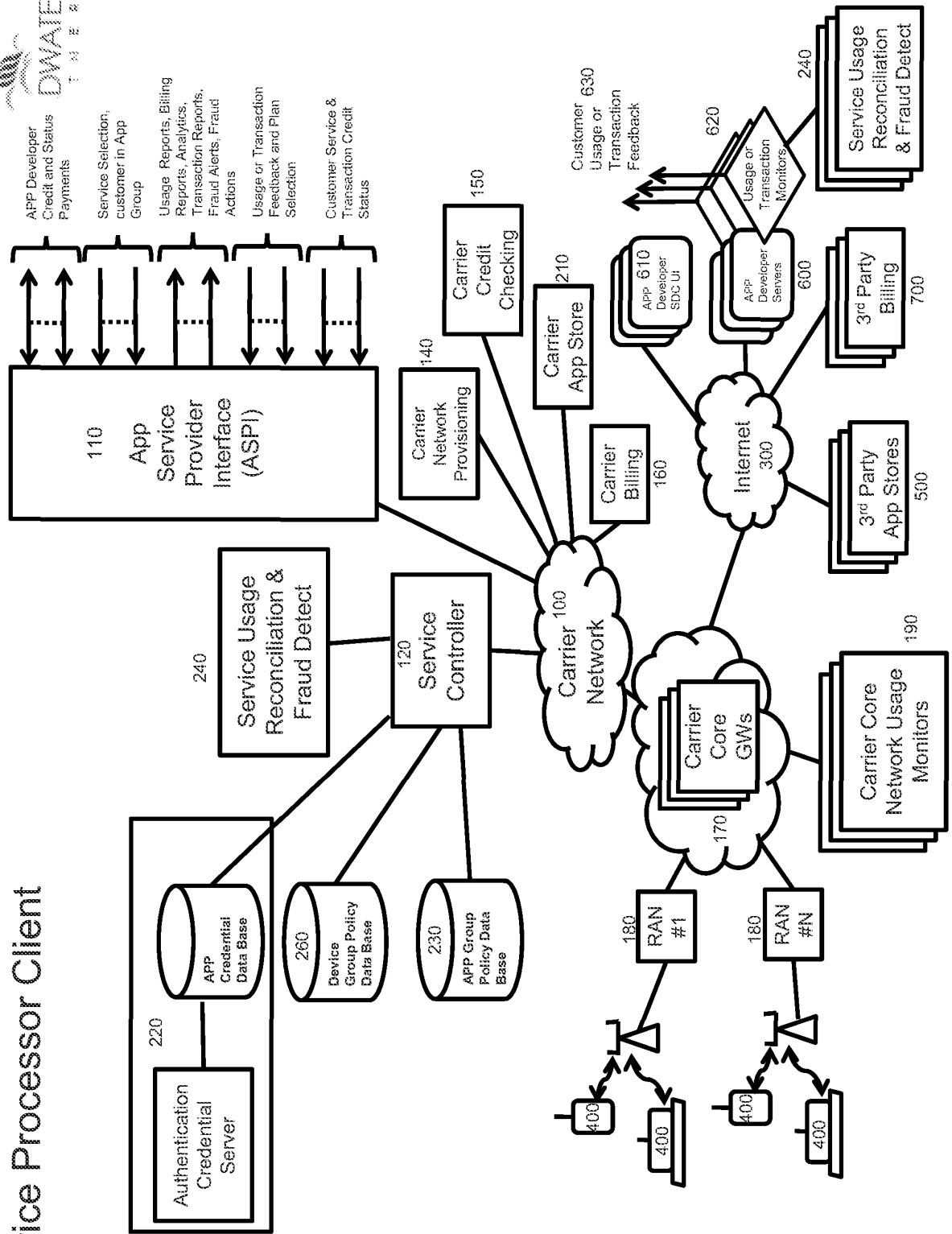


Figure 5 Description

- App service provider interfaces with ASPI (App Service Provider Interface) to do the following
 - Applies for carrier credit in order to publish its App Service
 - Carrier checks app service provider credit status and issues appropriate credit for the App service to go online
 - Carrier conveys its business rules to the app service provider and obtains agreement/signature before proceeding with the service offer
 - Carrier provides service plan selection offers to the app service provider to choose from
 - app service provider provides the App credential associated with selected plan and policy-set
 - app service provider can also connect to the App Authentication Credential Server directly to deliver the App credential
 - app service provider selects plan, App group, devices where the App can operate on, and also sets fraud parameters for carrier to notify
 - app service provider gets a web-portal interface to the Carrier SDC in order to create plans, assign policy-set, set fraud parameters and also selects if it wants to use network state information (e.g. NBS, TOD, QoS, background traffic, etc.) delivered by the device API in order to optimize App service usage
 - Carrier provides ongoing usage reports, transaction reports, analytics, fraud detection alerts to the app service provider to manage its App service
 - app service provider can provide Ad placement to carrier and/or to the App store for a nominal fee or in exchange for analytics
 - app service provider provides "good customer" feedback to the carrier indicating potentially bump-up on the service usage for a given App, device credential (MEID) and potentially user credential combination
 - Carrier provisions the App service in its network elements
 - Carrier configures Service Controller data base (SDC) with plan selection, plan policy-set (e.g. control, charging/billing, and notification) and fraud trigger parameters
 - app service provider can assign billing responsibility to Carrier, a 3rd party (App store) or directly to the user
 - Informs Service Controller of the selected App Group and potentially the devices (or device groups) that the App can operate under
 - Carrier performs usage monitoring and reconciliation and fraud detection
 - Service Processor delivers ongoing App service usage reports to Service Controller
 - Carrier network elements (GW, AAA, HA, etc.) delivers CDR/FDR to the Service Controller for App service usage reconciliation
 - app service provider provides fraud trigger parameters
 - app service provider provides "good customer" feedback as the mean to overrule potential fraud and/or usage overage
 - Service Processor performs App validation using various techniques including code signing, code hash verification and/or certificate based.
 - App validation can be done during download, launch and/or during service usage
 - App validation can be done locally in SP
 - App validation can be done with help of SC
 - App validation can be done via App store (e.g. 3rd party)
 - SP provide App API to inform App service with Network State information such as NBS, TOD, QoS, Background traffic, etc.

Figure 6: Platform Component Extensions to DAS to Implement App Service Provider Interface (ASPI)

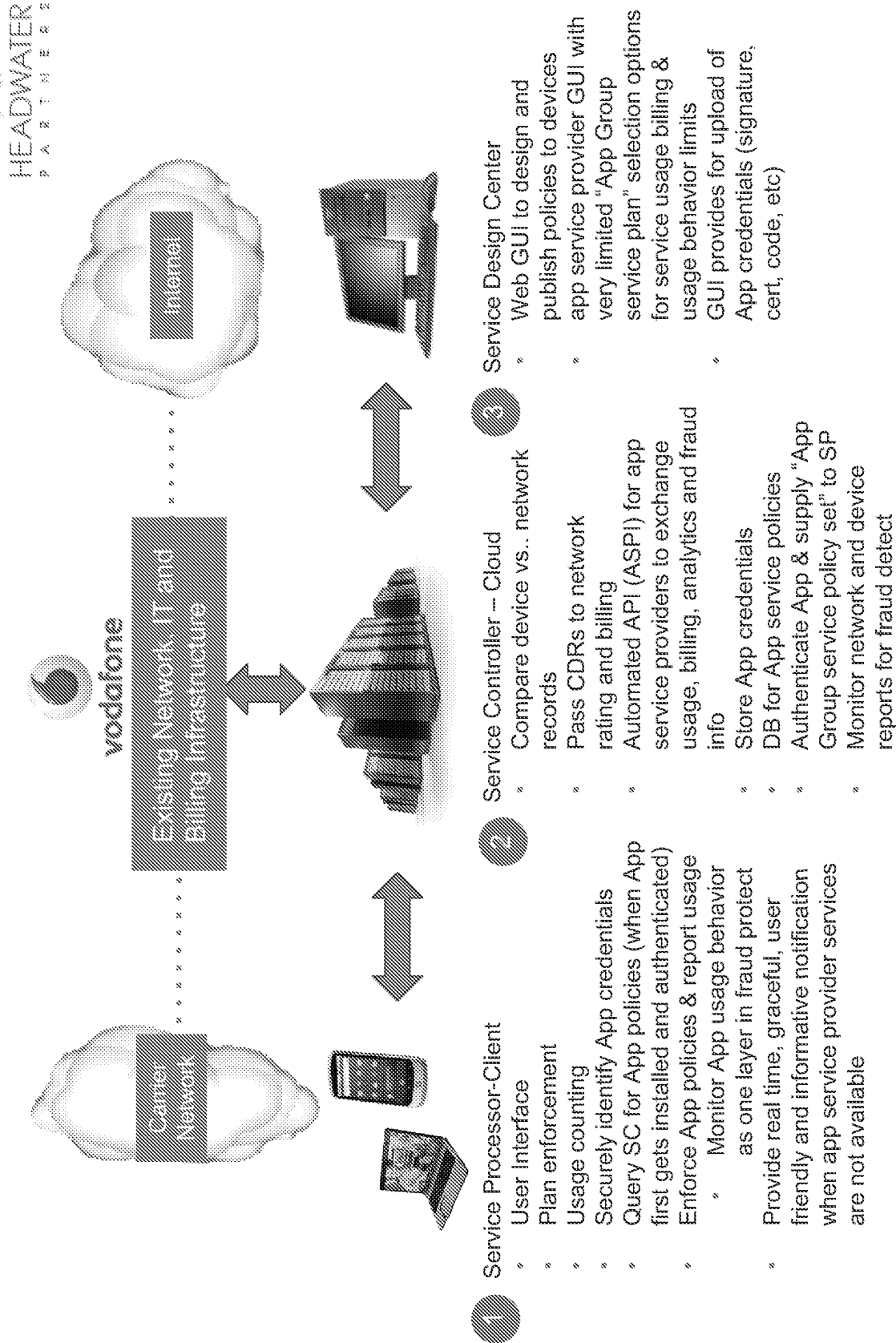
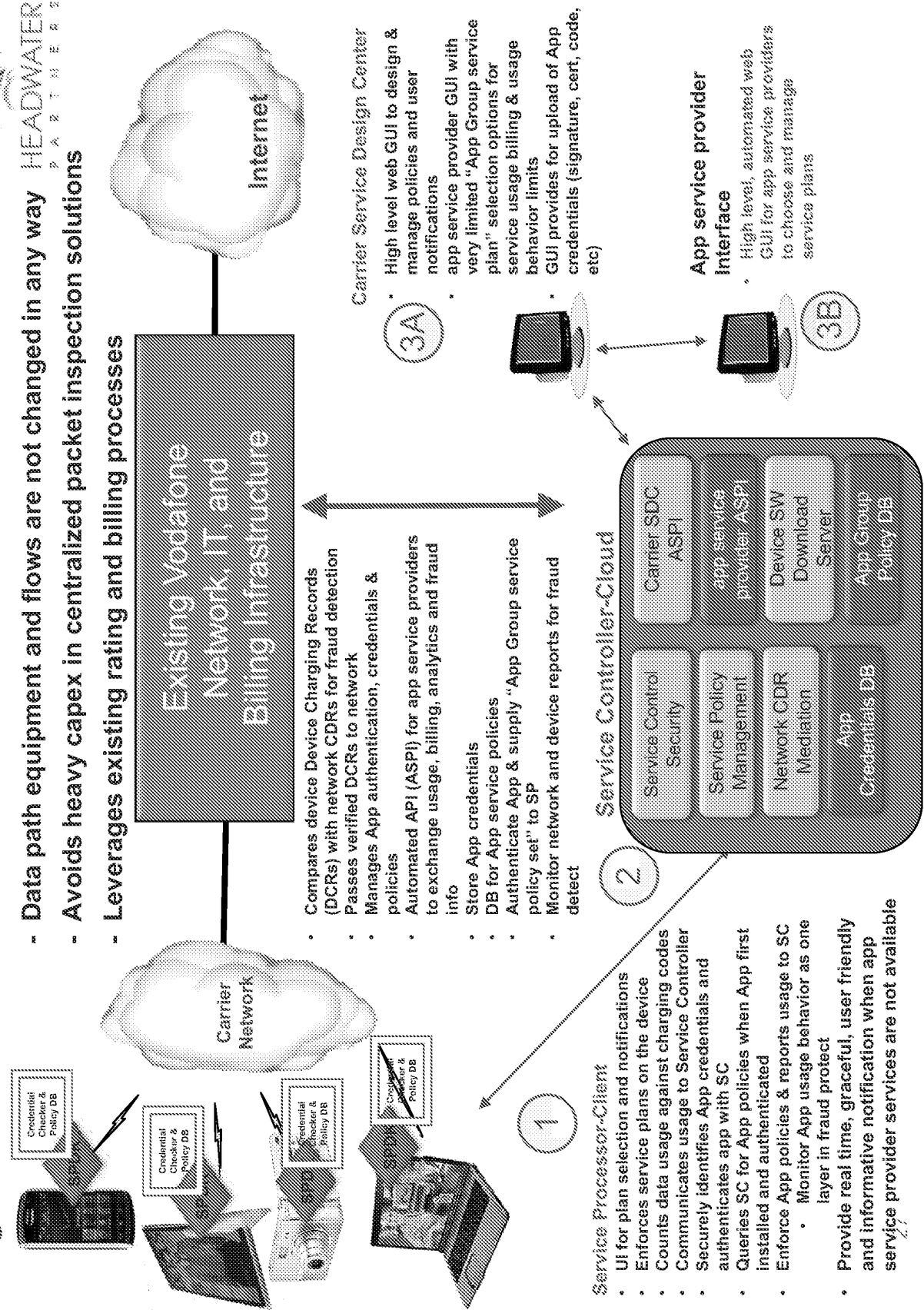
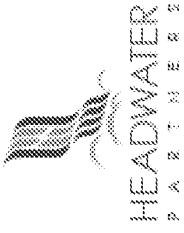


Figure 7: ASPI Extensions to DAS

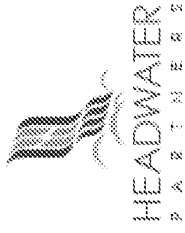


Top Level Summary Of DAS Carrier Embodiments



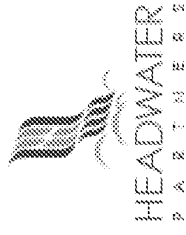
- * App service provider is simply a highly restricted sponsored services partner
- * Small and restricted subset of SDC capabilities & screens provided to app service provider
 - * Service plan selections
 - * Service plan cycle
 - * Service plan billing/charging policy selection
 - * Prepay, post-pay, plan duration, etc.
 - * Fraud detection parameter settings
- * Carrier offers “bulk (open access)” plans and large partner “ala cart” plans
- * app service provider is responsible for fraud
 - * User notification is key when credit status system protects carrier
 - * app service provider is shutdown
- * How the app service provider sets up and manages their app access services:
 - * Credit check is carried out separately by carrier
 - * app service provider receives credit for service, can not go beyond that credit
 - * Default for new unknown app service provider is pre-pay with guaranteed payment (e.g. wire transfer)
 - * Pre-pay and/or post pay is available for app service provider
 - * Shut down app service provider services for their app when they exceed their credit limit or run out of pre-pay credit
 - * NOTE: Critical to have device notification system that explains App service not available but device/network/other apps are fine
- * app service provider gets real time feedback on service usage stats and remaining credit for App Groups
 - * Can also sell analytics for real time ad and transaction optimization by app service provider
- * Can also provide app placement options as part of what app service provider pays for
 - * Highlighted in store, placed on device, placed with high visibility on device
- * Can also provide centralized transaction billing system and/or app store for app service providers

Additional DAS Carrier Embodiments



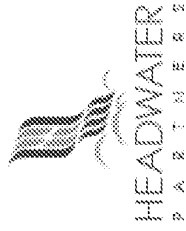
- Carrier can offer ASPI for app service provider services on any network even if network assets are not controlled or owned by carrier since access control and accounting are carried out by Service Processor in conjunction with Service Controller
 - Previously disclosed hardware secured DDR also makes this fraud resistant/proof without carrier network usage reports in real time
- World wide, WiFi, 3G/4G, roaming/home, etc.
 - No backhaul issues
- App can control it's own usage and go where ever it likes: app service provider services are unrestricted (not only App services allowed), any service possible with no changes to the existing APN provisioning, e.g. sponsored search with click-out, supports current internet ad model (arbitrarily inserted reference URL to any ad server)
- app service provider takes fraud risk for App services
- Graceful way to shutdown app service provider services and notify user when app service provider gets behind on service payouts
 - Again – device notification UI is KEY for making sure user understands that it is an app service provider service problem, not a device service or network service problem, when the app service provider runs out of credit or is shut down due to fraud events.
- Highly scalable with zero carrier touch

Embodiments for Verifying That App Credentials Belongs To An App Group With A Specific App Services Access Policy or Service Plan



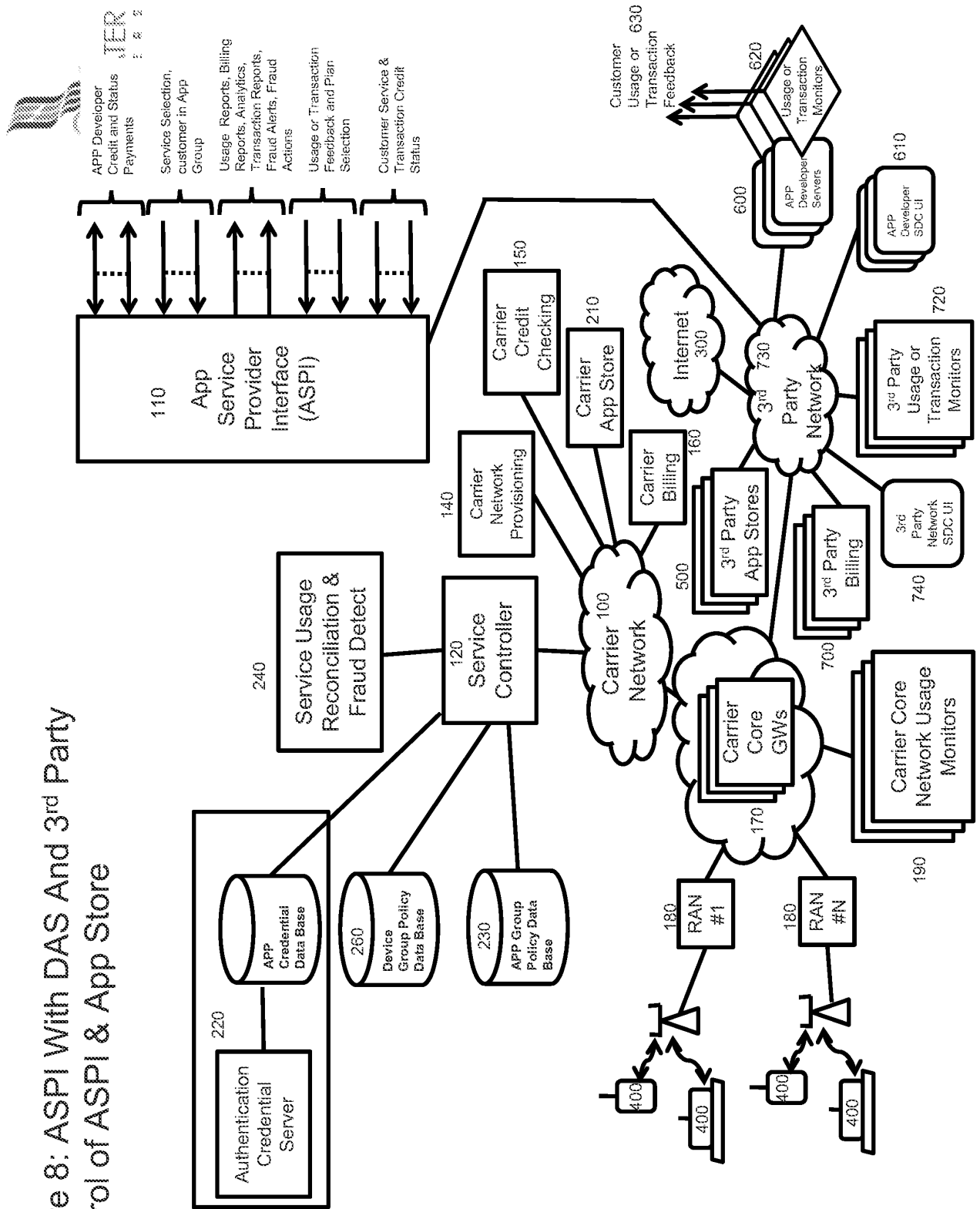
- * Device embodiments:
 - App credential checker - signature checker/hash checker for app that is part of the service processor, part of the OS or sits in secure OS execution -- first fraud detection layer
 - Confirm app signature/hash with known signature/hash stored in: service controller, download file on device, central authority
 - Check app when it is loaded to confirm that it is the right app
 - Possibly also check app each time it is launched and/or during app operation
 - Report results to Service Controller
 - If app signature/hash is not correct, then suspend, kill, block app
 - If app signature/hash is not correct, then notify service controller
- * Network embodiments:
 - Service controller or equivalent on carrier network maintains data base of valid signatures/hashes and corresponding service policies
 - Distributes to device checker via push or pull
 - Evaluates device checker hash result sent to server
 - App credentials data base or equivalent maintains data base of valid signatures/hashes and corresponding service policies
 - Distributes to device checker via push or pull
 - Evaluates device checker hash result sent to server

High Level Embodiment VI. ASPI System With 3rd Party Service Distribution And Control of ASPI



- Similar to previously disclosed MVNO DAS embodiments, but this embodiment extension includes ASPI system
- Provides for 3rd parties to create virtual networks using either proxy server/gateway approach or DAS approach
 - Example approach A: 3rd party owns and/or controls the proxy server/gateway cloud network as described herein and in other USPTO disclosures, negotiates wholesale access service deal with one or more carriers who own/control access network assets, and provides ASPI interface to set up app service provider system as described herein.
 - Example approach A: 3rd party owns and/or controls the DAS service controller and Service Processor cloud as described herein and in other USPTO disclosures, negotiates wholesale access service deal with one or more carriers who own/control access network assets, and provides ASPI interface to set up app service provider system as described herein.
- Example 3rd party provider scenarios (i.e. party that provides service and is not the party that owns the access network assets) :
 - Global carrier with wholesale partnerships with other carriers
 - App store providers (e.g. Google, Apple)
 - OS providers (e.g. Google, Microsoft)
 - Device OEMs (e.g. Apple, RIM, Samsung, Nokia)
 - M2M service providers (e.g. car connection services provider, vending machine connection services provider, home 2-way power meter connection services provider, etc)
 - Any other 3rd party connection services provider

Figure 8: ASPI With DAS And 3rd Party Control of ASPI & App Store



Additional Embodiments

• SDC

- Create the plan profiles via SDC (this is slightly different than creating a plan for a device group, because the device group (i.e. "App Group") has not yet been defined -- the association of the plan profile with the "App Group" happens every time an app logs in.
- Define options for app service providers who have elected to pay for premium placement on device

• Service Controller

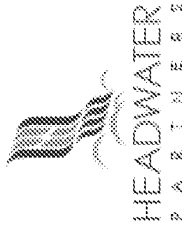
- Service Controller includes an App Group data base with app credentials and policy set that goes with each app that is part of the program;
- App seeks access, service processor asks service controller if app is part of app service provider program, if no then app is treated under bulk plan (if any), if yes then app service plan policy set is downloaded to device service processor
- When Service Processor logs in with possible app id, Service Controller coordinates checking of app id against data base of apps that are in plan, associates app with device -- device now belongs to the group of devices that have that managed service app
- Service controller sends usage reports to carrier CDR system (and possibly partner CDR system)
- Service Controller downloads service policy set to service processor when valid app seeks service
- Track real time usage of app service provider App/Device pairs (or user credential), implement fraud detection rules in usage reconciliation engine
- Track real time App Group usage and if it exceeds credit, begin app group notification or shut down processes
- Optional: Push app to devices for app service providers who have elected to pay for premium placement on device

Additional Embodiments

- **Service Processor**
 - * Service processor checks credentials of app (hash, signature, certificate) to identify it
 - * Service processor initiates the handshake with Service Controller to check app status (app service provider sponsored plan or not), share app credentials for authentication, accept service policy set for app, enforce policies
 - * Service processor reports usage for app
 - * Service processor performs ongoing fraud detection on app
 - * When app is shut down, provide notification to user that app service has a problem and device/other apps are fine
 - * Service Processor provides App API for App to exploit network state modifiers and optimize its service charge, e.g. TOD, NBS, QoS, background traffic, active networks, etc.
- **Analytics and Ad optimization**
 - * Use device service processor to collect information on device usage state, apps being used at present time, what user is doing with the apps (e.g. what user is searching for or purpose of app, device location, network busy state, etc
 - * Inject Carrier ads on APP that are optimized to present device usage state
 - * Select ad content that is appropriate for current network busy state (e.g. select high bandwidth content when high bandwidth is available or less expensive, select low bandwidth content otherwise)
 - * Provide analytics information to app service providers or ad providers via ASPI for a fee

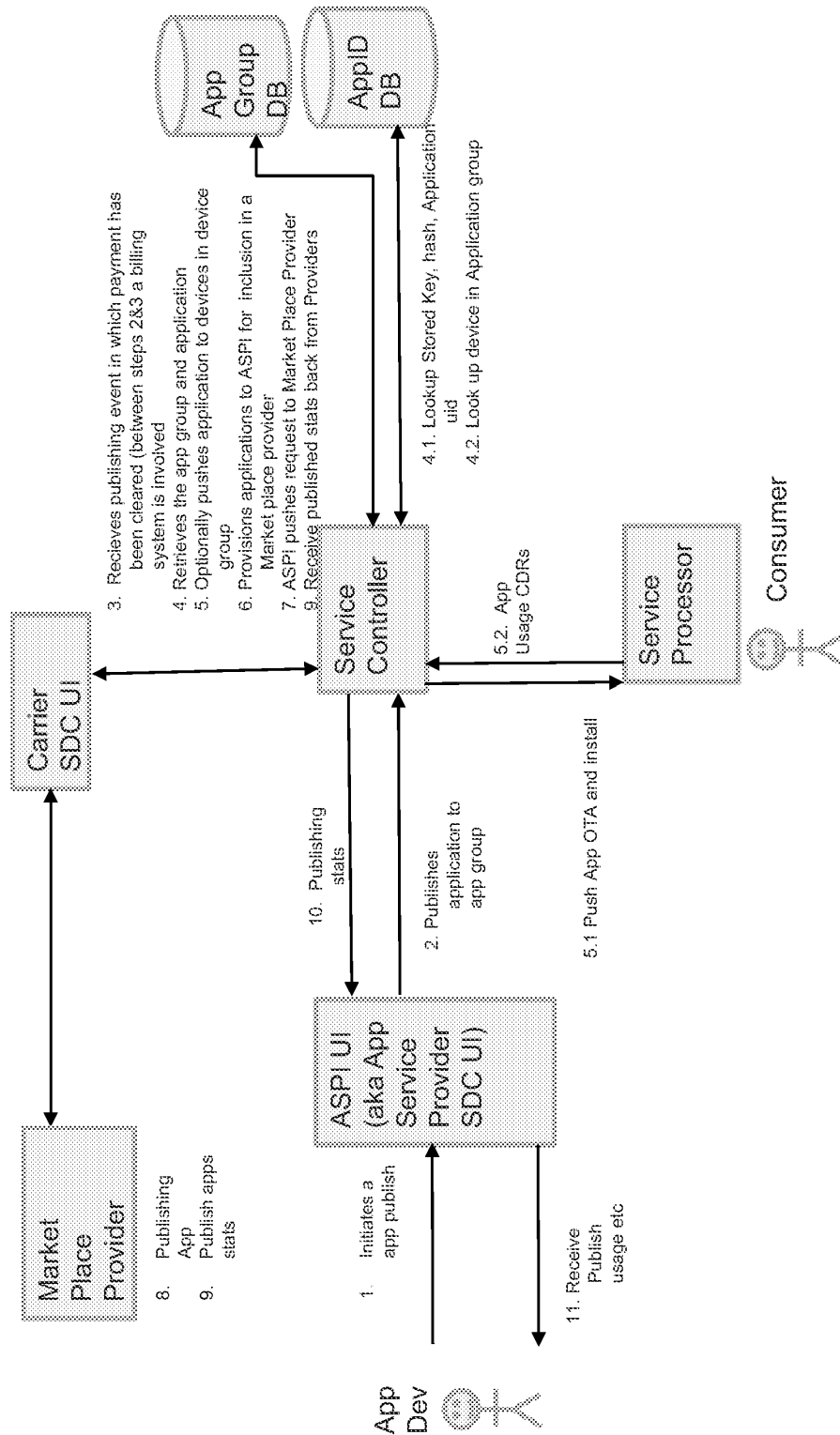
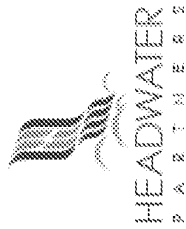
Additional Embodiments

- ◆ Individual version of app created and signed at time of download and verified via app authentication process – prevents mass attack

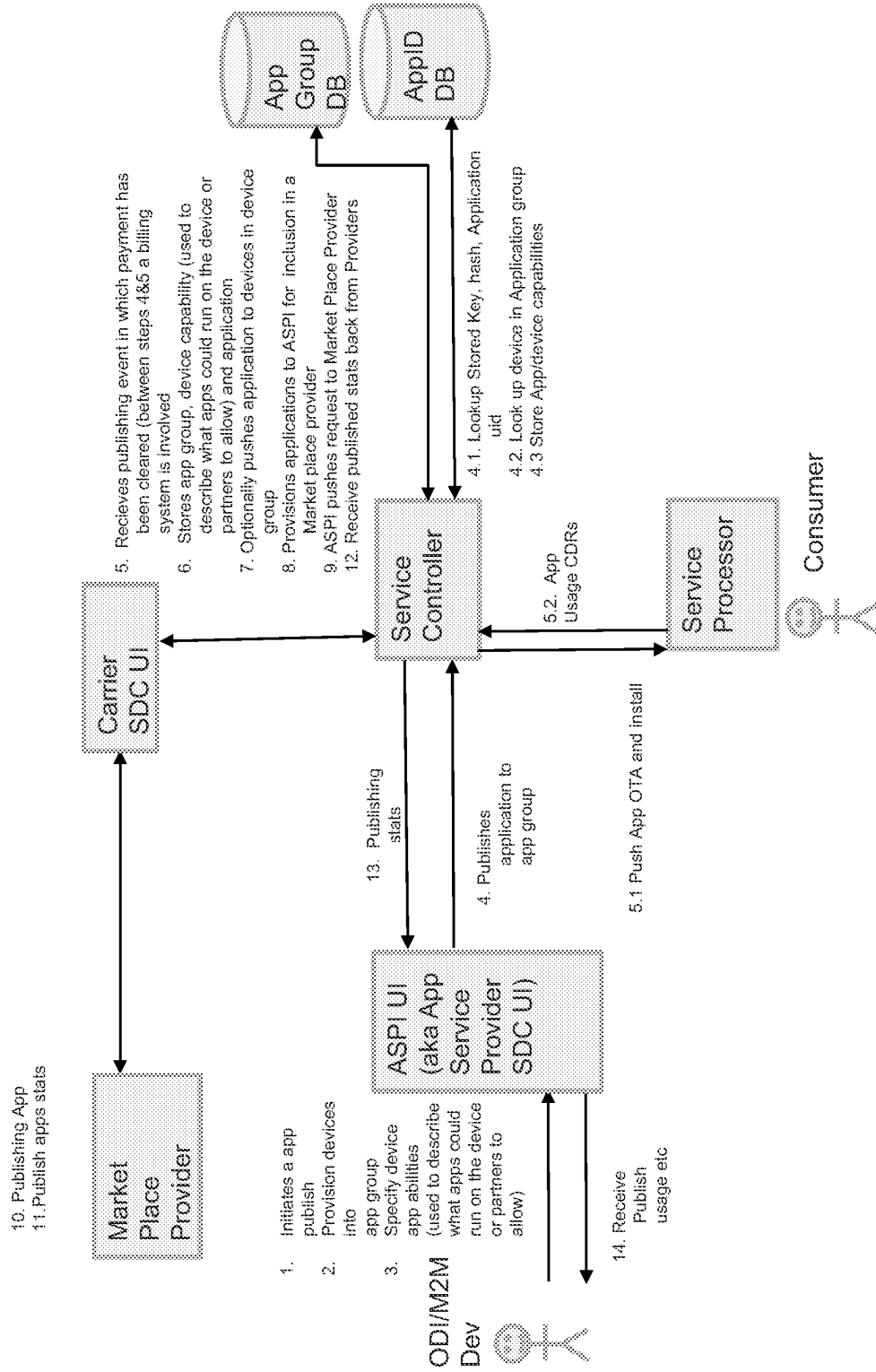
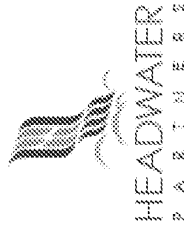


Part 3: Select Detailed Embodiments & Figures

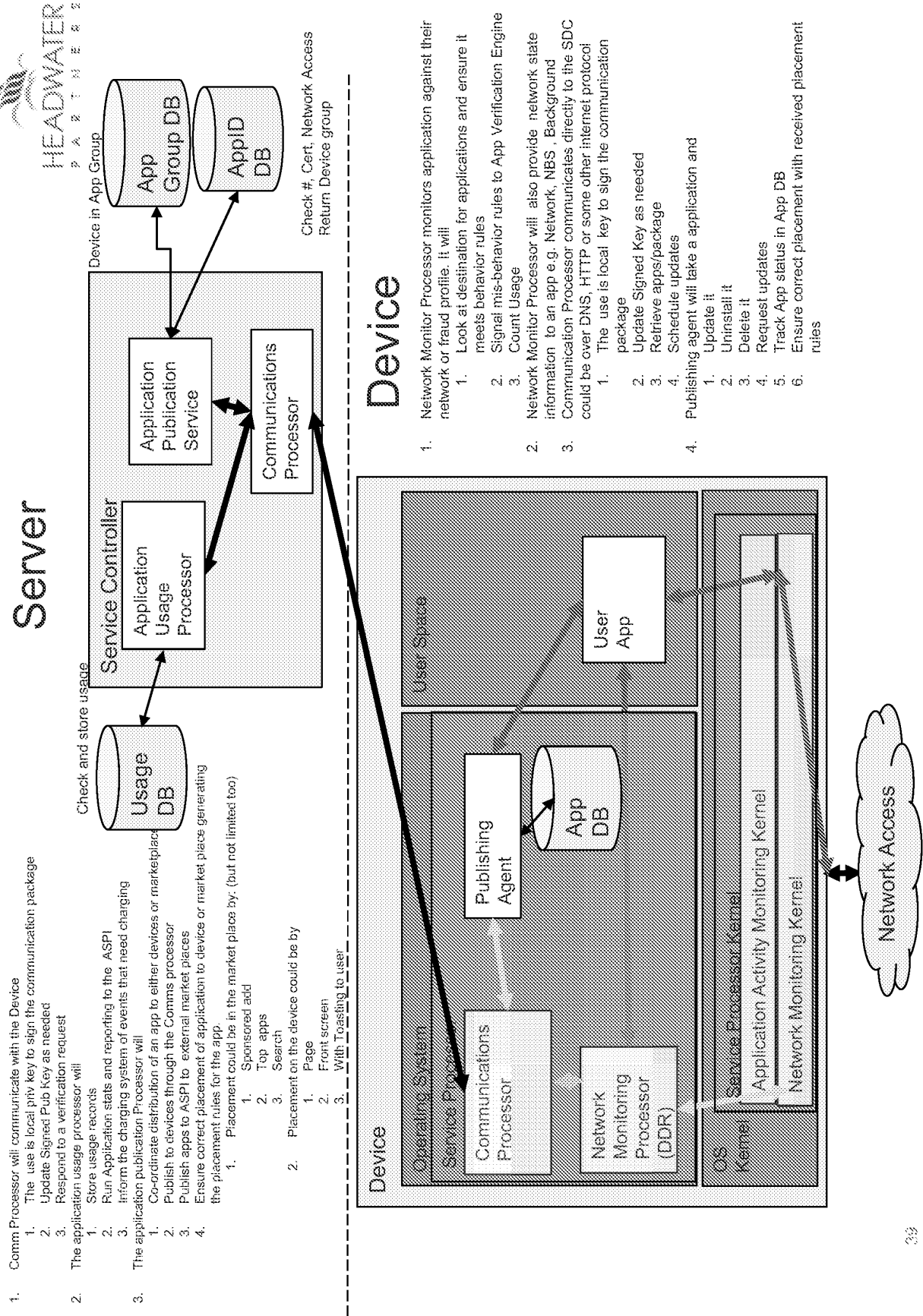
Publishing Apps Using ASPI System



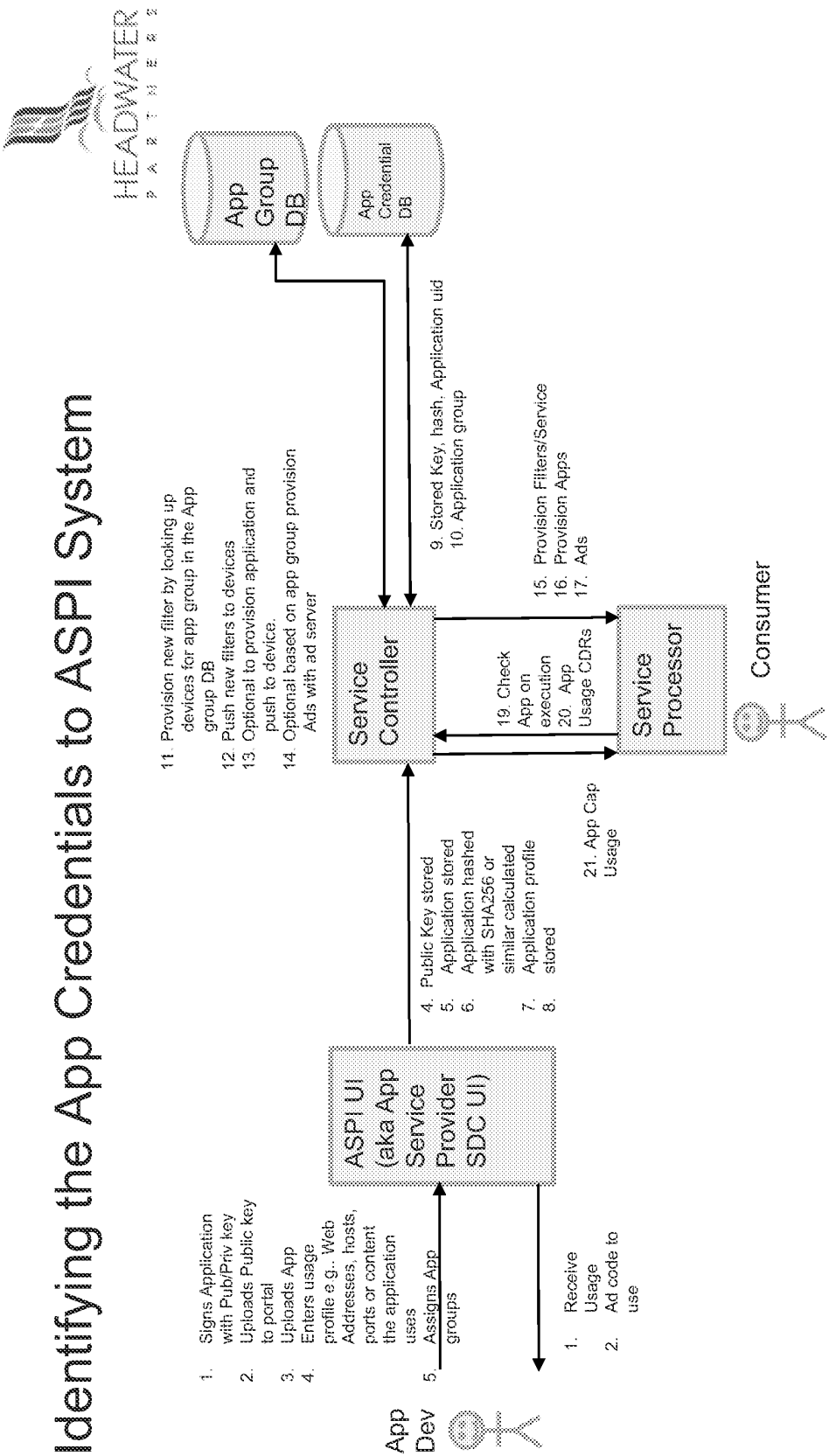
Publishing Apps/Devices(ODI) Using ASPI System



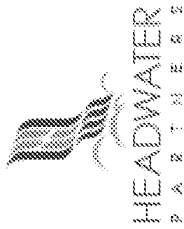
Provisioning apps with ASPI (publishing could also reside in SEE too)



Identifying the App Credentials to ASPI System

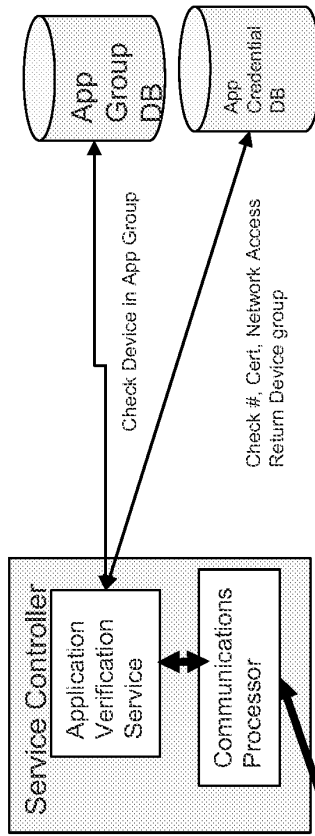


Identifying the Apps to ASPI System (Embedded OS Enhanced Functionality)

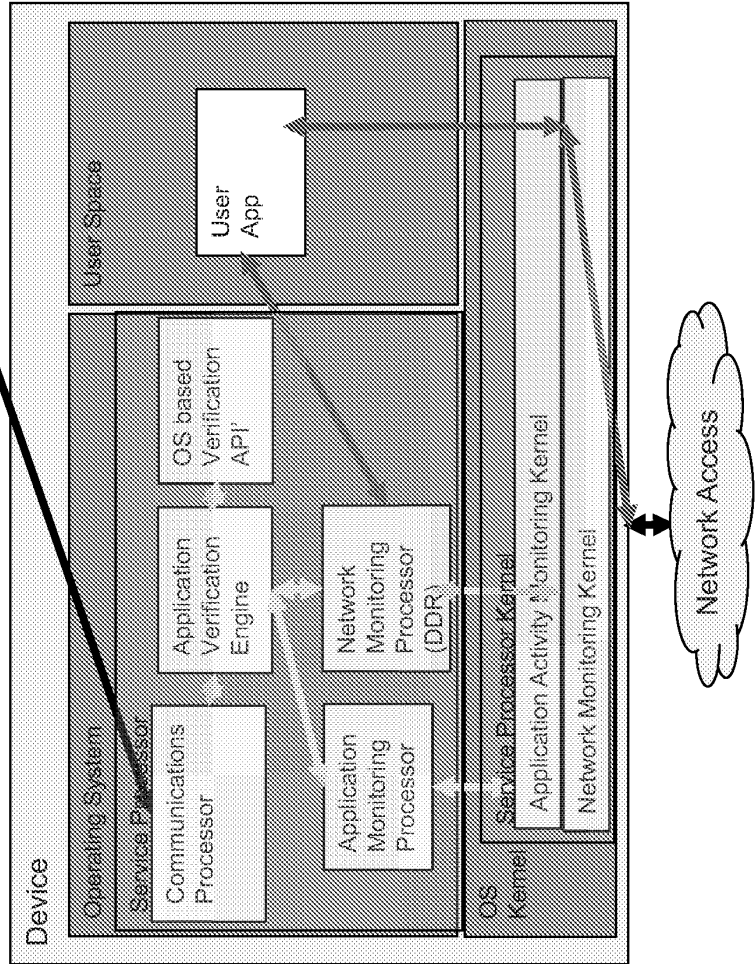


Server

1. Comm Processor will communicate with the Device
 1. The use is local priv key to sign the communication package
 2. Update Signed Pub Key as needed
 3. Respond to a verification request
2. The application service will
 1. Receive a verification request
 2. Validate the App in the App DB by key, has, UID etc
 3. Take the Device ID and check it in the app group db
 4. Validate that the app group contains device



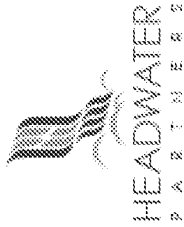
Device



Device

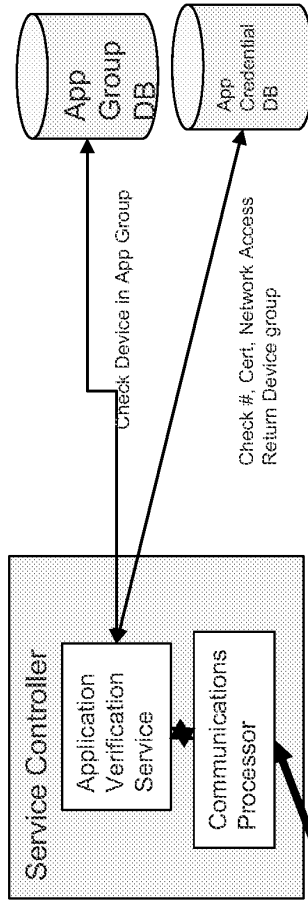
1. Network Monitor Processor monitors application against their network or fraud profile. It will
 1. Look at destination for applications and ensure it meets behavior rules
 2. Signal mis-behavior rules to App Verification Engine
 3. Count Usage
2. Network Monitor Processor will also provide network state information to an app e.g. Network, NBS, Background such as launch etc
3. Application Monitor Processor monitors application activity
4. Communication Processor communicates directly to the SDC could be over DNS, HTTP or some other internet protocol
 1. The use is local key to sign the communication package
 2. Update Signed Key as needed
5. App Verification engine will :
 1. Create hash such as SHA256
 2. Get the applications public key, UID
 3. Gather network events
 4. Communicate to the Comms Processor

Identifying the Apps to ASPI (Chip based, VM based, etc)



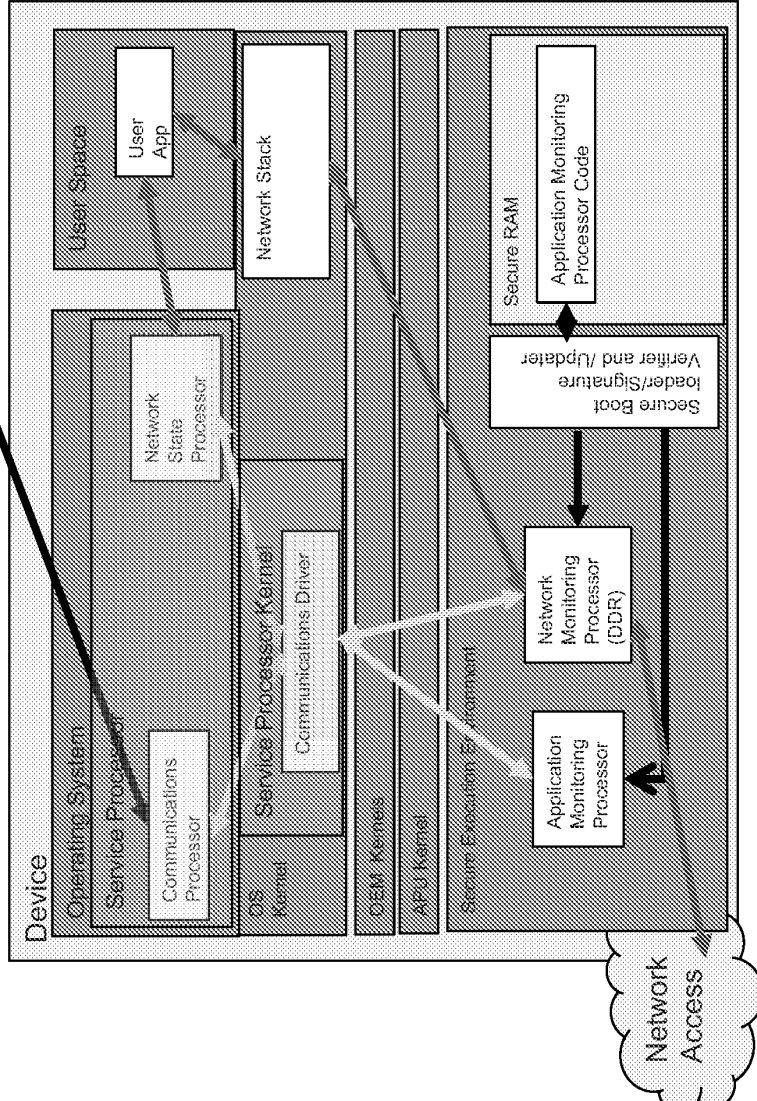
Server

1. Comm Processor will communicate with the Device
 1. The use is local priv key to sign the communication package
 2. Update Signed Pub Key as needed
 3. Respond to a verification request
2. The application service will
 1. Receive a verification request
 2. Validate the App in the App DB by key, has, UID etc
 3. Take the Device ID and check it in the app group db
 4. Validate that the app group contains device



Device

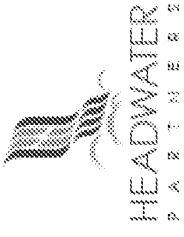
1. Network Monitor Processor monitors applications against their network or fraud profile. It will
 1. Look at destination for applications and ensure it meets behavior rules
 2. Signal mis-behavior rules to App Verification Engine
 3. Count Usage
2. Network State Processor will also provide network state information to an app e.g. Network, NBS, Background
3. Application Monitor Processor monitors application activity such as launch etc
4. Communication Processor communicates directly to the SDC could be over DNS, HTTP or some other internet protocol
 1. The use is local pub key to sign the communication package
 2. Update Signed Key as needed
5. App Verification engine will :
 1. Create hash such as SHA256
 2. Get the applications public key, UID
 3. Gather network events
 4. Communicate to the Comms Processor



Network Access

Part 4: Reconciliation and Fraud Detection For ASPI System

Motivation



In DAS systems, there is a potential for device users to spoof or hack the device Service Processor to fraudulently obtain access services at incorrect lower service usage billing rates.

To prevent this, a variety of fraud detection and Service Processor protection embodiments have been disclosed in previous USPTO filings.

This filing provides additional embodiment details in the area of service usage reconciliation from one or more reporting sources for the purpose of detecting and mitigating potential fraud

Service Usage Reporting Sources



Device

- Service Processor Device Charging Reports (DCRs) – generally not trusted
- Secure DDR processor Device Data Records (DDRs) – can be trusted with secure design of DDR processor and secure DDR report sequencing
- Applications (Apps) – generally not trusted

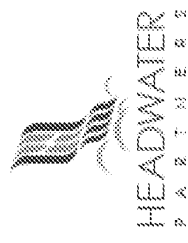
Carrier (Or MVNO) Network – generally trusted

- Gateways (GGSN, SGSN, PDSN, Routers, Switches, etc)
- Home Agent
- Proxy servers
- Charging gateway system
- Mediation
- Reconciliation
- Billing
- AAA
- Etc

- Can be “bulk measures” of general usage, can be FDRs that indicate destinations, can include network state records at time of usage (e.g. Time of Day (TOD), network busy state or congestion state (NBS), type of network (e.g. 2G, 3G, 4G, Wi-Fi, home/roaming, etc), etc)

3rd Parties – can be trusted with correct processes in place to verify reports via sampling etc

- Usage reports, site visits, transaction reports, ad view reports, economic benefit reports, usage credit increase instructions, etc from:
 - Partner service destination servers (e.g. search site, ad site, application service site, shopping site, content site, gaming site, email site, etc)
 - Enterprise customer network
 - Etc



Basic Fraud Prevention Embodiments

Make sure that when device is on network that Service Processor is properly authenticated and reporting usage

Make sure service processor is properly responding to access control commands

Make sure the service processor code integrity is maintained – that the code has not been tampered with

- * e.g. Hash or signature checks, execute in secured partitions, etc – see USPTO previous filings for additional embodiments
- * e.g. e.g. Hash or signature checks, execute in secured partitions, etc – see USPTO previous filings for additional embodiments

Make sure that an application that has an application specific service has not been tampered with

Make sure usage reports from trusted source(s) indicate usage behavior that is within usage control policies

Make sure usage reports from trusted source(s) match usage reports from Service Processor

Make sure usage does not exceed caps

- * Bulk caps
- * Service component (e.g. application services, sponsored services) caps

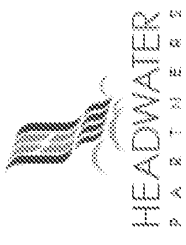
Make sure usage patterns fit within expected usage patterns

- * Based on access control policies that should be in place
- * Based on statistical assessment of device group population
 - ... Can be updated dynamically by Service Controller

Make sure network destinations and usage (e.g. reported via FDRs) are consistent with the service plan that should be in place

Make sure that access performance achieved by device (or app) is consistent with service plan policy that should be in place

What to Do When Fraud is Detected



Notify network administrator or network resource for further evaluation

Increase billing rate

Increase billing rate and notify user via one or more communication media (e.g. service processor or device notification client UI, text, email, voice mail, phone call) of service agreement violation and increased billing rate.

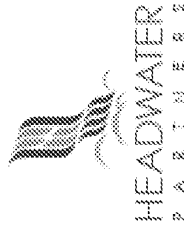
Throttle device access (or app access)

Suspend device access (or app access)

» Verify the service usage measurements the device is reporting:

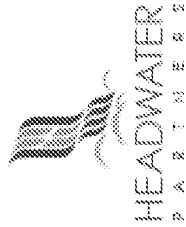
- 43

Basic Fraud Detection Procedure Embodiments 2



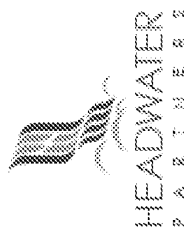
- Basic methods for reconciling device usage reports and algorithms for reconciling device usage reports to detect fraud
 - Only bulk usage is available
 - Cap and verify cap is not exceeded
 - If multiple services, then verify sum of caps is not exceeded
 - Throttle (e.g. max rate) and verify throttle policy is not exceeded
 - Compare bulk usage or sum of component usage that device reports with 2nd measure (Device, Network or app service provider Server based)
 - Observe rate of bulk usage (e.g. usage velocity) and compare to policy given the set of services that are enabled
 - Observe usage breakdown of individual usage categories reported by device to determine if each is within plan and/or expected behavioral limits
 - Limits set by plan parameters
 - Limits set by expected user behavior
 - Limits set by observing, collecting and compiling population statistics of user device group
 - Compare component usage vs. real time network state reported by other devices on same BTS or sub-nets
 - Observe usage breakdown of network state (e.g. TOD, NBS, type of network, roam/home, etc) dependent controls (caps, speeds, background/foreground, QoS, etc) reported by device to determine if each is within plan and/or expected behavioral limits
 - Limits set by plan parameters
 - Limits set by expected user behavior
 - Limits set by observing, collecting and compiling population statistics of user device group
 - Compare component usage vs. real time network state reported by other devices on same BTS or sub-nets
 - Send limiting command (e.g. cap, stop, throttle commands) to device and see if usage is stopped or controls are obeyed}
 - Bulk plus network state is available
 - Observe bulk usage behavior vs. network state reported by 2nd source (e.g. network or DDR) and compare to policy
 - Observe bulk usage behavior vs. population statistics for bulk usage behavior and flag if outside of norm

Basic Fraud Detection Procedure Embodiments 3



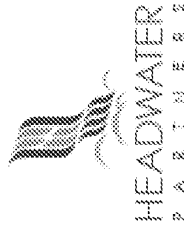
- Basic methods for reconciling device usage reports and algorithms for reconciling device usage reports to detect fraud:
 - FDR level information is available
 - Classify FDRs to known service components, rack up credit of classified usage for each service component, ensure that service component usage does not exceed limits and/or matches with device reports for component, ensure that sum of components matches bulk
 - If not all service component usage can be classified with known sources (e.g. ad servers, content servers, allowed click-out, etc), then create a unclassified usage policy allowance (e.g. proportional to or a percentage of the classified service usage for the component) and "credit" that amount of unclassified service usage to the service component. Then as long as the bulk service usage that remains after each service component policy allowance "credit" is deducted is under the policy limit (if any) for bulk and/or the remaining bulk usage matches what the device is reporting then no fraud alert is generated.
 - The pre-determined usage policy allowance can be based on a set of unclassified usage policy allowance rules determined by various methods including pre-testing such as using the service component and classifying the resulting usage as "classifiable" and "not classifiable", or observing device/user population statistics.
 - Policy allowance can include an amount for "click-out" and/or good customer feedback (see what follows).
 - Algorithm:
 - Accumulate all FDR usage reports that can be classified into known service component usage.
 - For each service component, based on the amount of service usage that could be classified for that component, apply the unclassified usage policy allowance based on the amount of classified usage for that component and application of the unclassified allowance usage policy rules. This amount can be added to the classified service usage for each component to form an estimate of the total service usage (classified plus unclassified allowance) for each service usage component.
 - In the case of multi-service App, the classified traffic includes unshared white-listed services as well as shared services. Shared services will have its own allowance as part of classified traffic
 - Subtract the classified plus unclassified allowance usage for each service component from the total service usage recorded by the second measurement source (e.g. network, DDR, good customer feedback, etc). The amount of service usage remaining after these subtractions from the total service usage is the unclassified or "bulk" service usage.
 - Ensure that the classified plus unclassified allowance service usage for each component is below the policy limits and/or matches device usage reports for that component.
 - Ensure that the remaining "bulk" service usage is below the policy limits and/or matches device usage reports for the device/user "bulk" usage category.
 - For each service component, can also have the device record the network state that existed at the time of usage (and possibly the policy controls that are in place for the usage), then compare this to service policy and/or a second measure (e.g. network measure or DDR) of network state. Ensure that the usage and/or policy controls for each service component are within policy limits
 - FDR level information is available and network state information is available
 - Perform the above algorithm with the appropriate service limits and/or traffic control policy checks for each network state that is recorded with the second usage measure. Classification now includes not only service components, but service components for each network state – run second measurement source classified and unclassified allowance usage checks as described above across all service components for all network states and compare results to usage policy and/or device reports.

Basic Fraud Detection Procedure Embodiments 4



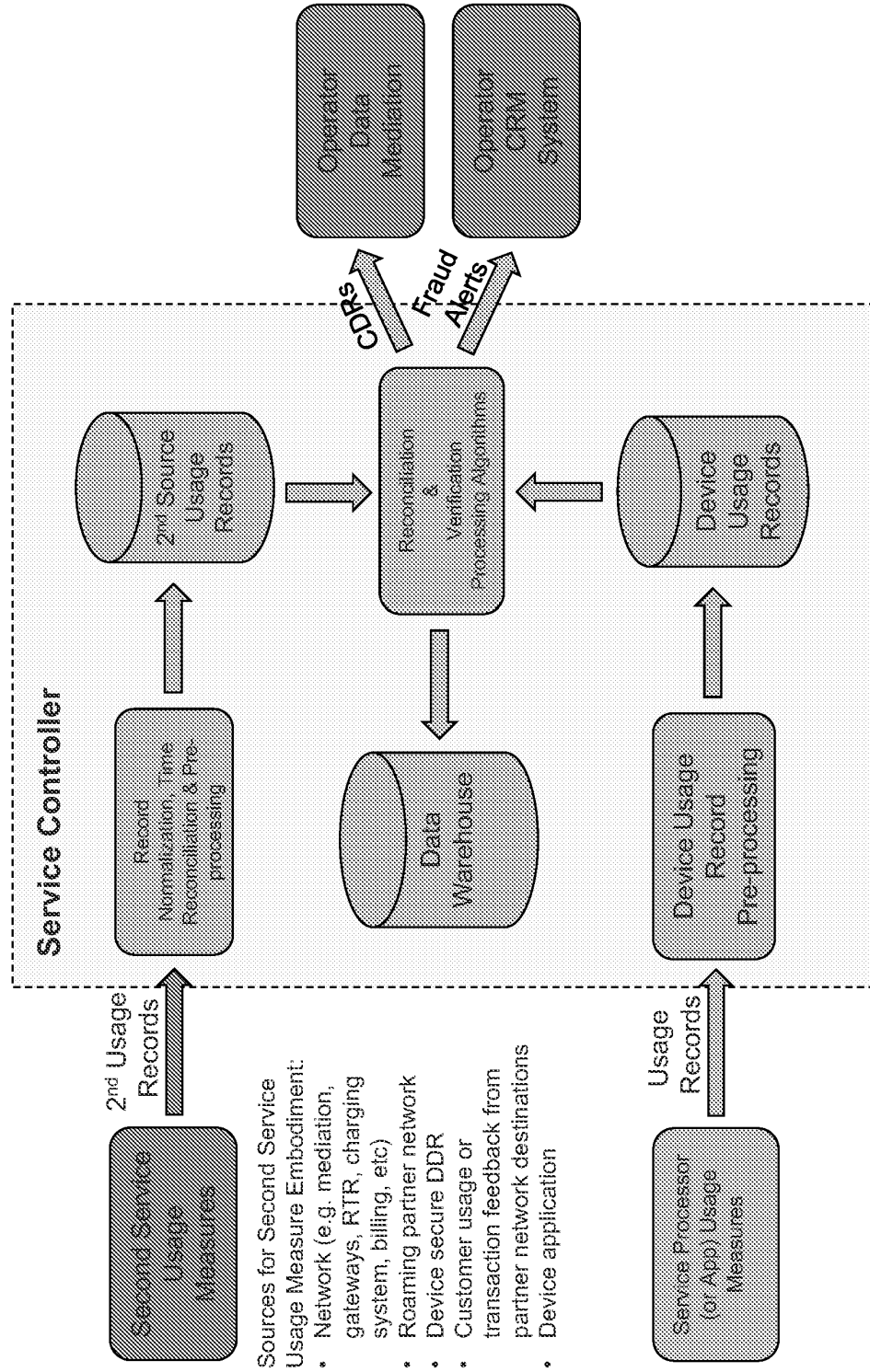
- * How to make sure that App credential is valid:
 - * Service processor validation of App credential (e.g. hash, signature, certificate based verification)
 - * Device OS based validation of App credential (e.g. hash, signature, certificate based verification)
 - * Service Controller assisted App credential validation (e.g. hash, signature, certificate based verification)
 - * Authentication Credential Server based App credential validation (e.g. hash, signature, certificate based verification)
- * **How to use feedback from component service servers, ad sites or stores be used to increase allowances:**
 - * Device credential (e.g. http header info, special side channel, app passes credential, etc) is passed to component service server so that server can log visits, usage and/or transactions
 - * Apply rating rule to convert visits, usage and/or transactions to good customer points
 - * Pass good customer points to Service Controller (or possibly directly back to Service Processor or service component App)
 - * Service Controller (or Service Processor/App) uses rating rule to convert good customer feedback to usage allowance increase-
 - * Usage allowance can be for specific component service, more than one component service, or for general purpose usage
 - * Example: each time a user clicks on an ad (or buys something or visits a web site to view something), they get so much free search, browsing, text, voice, or other services...
- * **How to make sure a user does not exploit free services such as book or music clip browsing:**
 - * Cap usage per unit time
 - * Reduce cap if customer abuses service
 - * Reduce cap if customer abuses service and does not get good customer feedback
 - * Get good customer feedback
 - * Real time feedback
 - * Every so often feedback
 - * Increase limits if good customer feedback is received
- * **How to verify "click-out" usage:**
 - * Good customer feedback
 - * Make sure the "home" site is being accessed between surf outs and not more than a certain amount of surf out is occurring between home site visits.

Basic Fraud Detection Procedure - 5

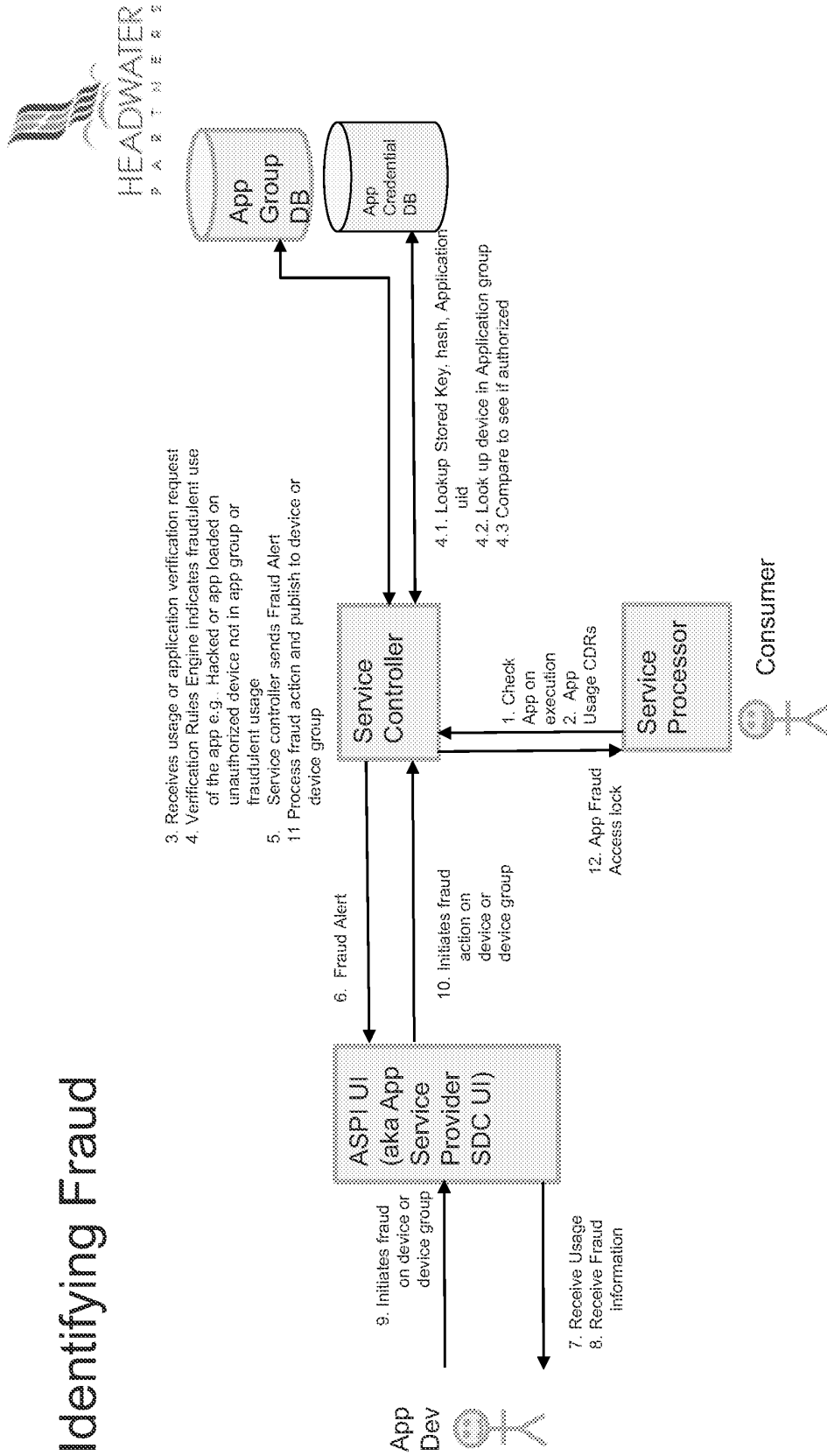


- Basic methods to determine if the Service Processor is in place and working correctly:
 - Flag possible error when SP does not log in but off-hook, network CDR/FDR or secure DDR report is received
 - Flag possible error when Service Processor authentication procedure is not correctly completed
 - Flag possible error when network counts and/or DDR counts do not match device service processor counts (see additional material herein for more details)
 - To minimize possible fraudulent usage between time device is first used and time that first network usage reports are received, use accelerated secure DDR reports
 - Use DDR processor access controller to prevent general purpose device usage before Service Processor and DDR are securely logged into Service Controller
 - Use securely sequenced DDR reports with maximum timeout between reports to make sure period of time where usage can be fraudulently acquired is minimized
 - When SC has issued "service stopped command" to service processor but then still receives usage report for the device from network, flag error or instruct network to suspend device or charge device user at a higher rate and notify device user

Service Controller CDR and DCR Reconciliation Processing for Fraud Detect



Identifying Fraud

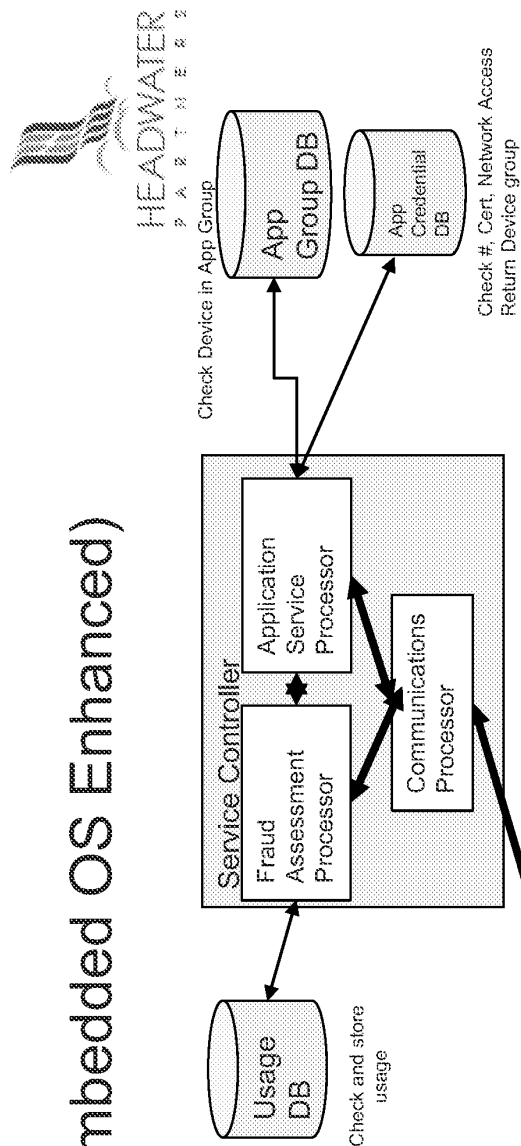


App Dev could choose to have the fraud detection automated in which case steps 9-10 would be skipped

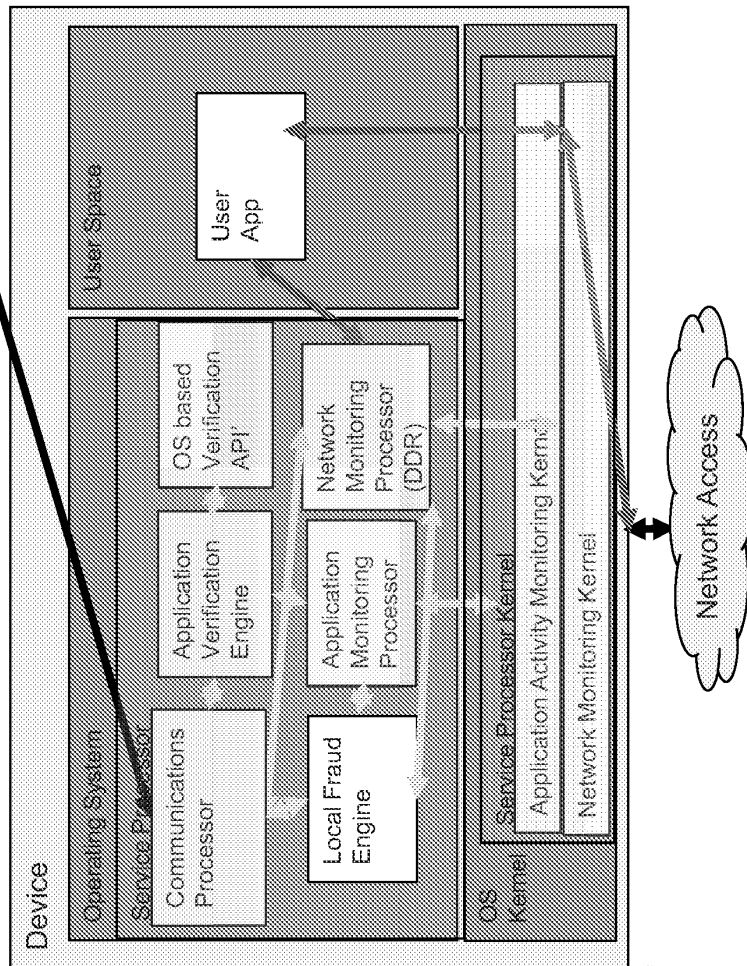
Identifying Fraud (Embedded OS Enhanced)

Server

1. The Comm Processor will communicate with the Device package
 1. The use is local priv key to sign the communication package
 2. Update Signed Pub Key as needed
 3. Respond to a verification request
2. The Fraud Assessment Processor will
 1. Store usage records
 2. Run Application Fraud rule set
 3. Perform advance data analytics and data mining to spot Fraud
 4. Alert the App Dev through the portal etc of Fraud alerts
3. The application Service Processor will
 1. Notify the Fraud Assessment Processor of a application usage event that is invalid



Device

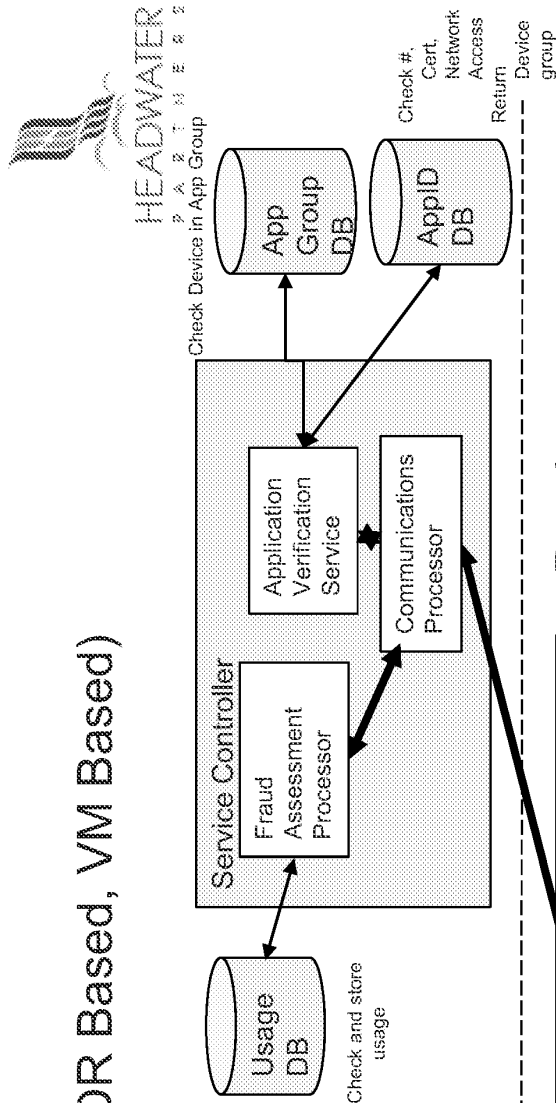


1. Network Monitor Processor monitors application against their network or fraud profile. It will
 1. Look at destination for applications and ensure it meets behavior rules
 2. Signal mis-behavior rules to App Verification Engine
 3. Count Usage
2. Network Monitor Processor will also provide network state information to an app e.g. Network, NBS, Background such as launch etc
3. Application Monitor Processor monitors application activity
4. Communication Processor communicates directly to the SDC could be over DNS, HTTP or some other internet protocol
 1. The use is local key to sign the communication package
 2. Update Signed Key as needed
5. App Verification engine will :
 1. Create hash such as SHA256
 2. Get the applications public key, UID
 3. Gather network events
 4. Communicate to the Comms Processor
6. Local Fraud Engine will:
 1. an used heuristic or similar based event monitoring on events sent to the Comm Processor
 2. Drive throttling or blocking events into the application and Network monitoring Processor to prevent/restrict fraud

Identifying Fraud (Chip DDR Based, VM Based)

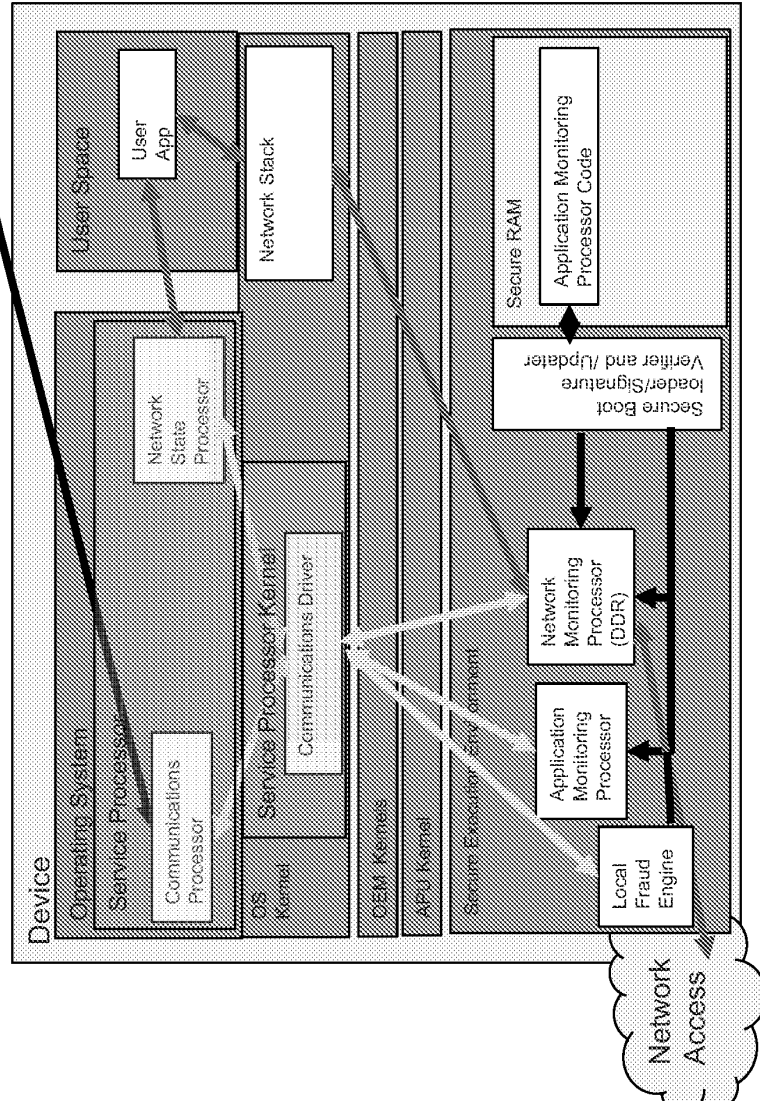
Server

1. Comm Processor will communicate with the Device
 1. The use is local priv key to sign the communication package
 2. Update Signed Pub Key as needed
 3. Respond to a verification request
2. The Fraud Assessment Processor will
 1. Store usage records
 2. Run Application Fraud rule set
 3. Perform advance data analytics and data mining to spot Fraud
 4. Alert the App Dev through the portal etc of Fraud alerts
3. The application Service Processor will
 1. Notify the Fraud Assessment Processor of a application usage event that is invalid

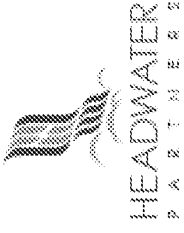


Device

1. Network Monitor Processor monitors application against their network or fraud profile. It will
 1. Look at destination for applications and ensure it meets behavior rules
 2. Signal mis-behavior rules to App Verification Engine
 3. Count Usage
2. Network Monitor Processor will also provide network state information to an app e.g. Network, NBS, Background such as launch etc
3. Application Monitor Processor monitors application activity
4. Communication Processor communicates directly to the SDC could be over DNS, HTTP or some other internet protocol
 1. The use is local key to sign the communication package
 2. Update Signed Key as needed
5. App Verification engine will :
 1. Create hash such as SHA256
 2. Get the applications public key, UID
 3. Gather network events
 4. Communicate to the Comms Processor
6. Local Fraud Engine will:
 1. an used heuristic based or similar event monitoring on events sent to the Comm Processor
 2. Drive throttling or blocking events into the application and Network monitoring Processor to prevent/restrict fraud



Fraud Scenarios



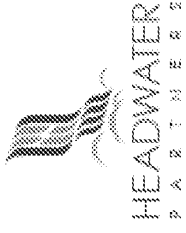
- * Service Processor (SP) doesn't attempt authentication within acceptable window after the Service Controller (SC) receives "data session started" indication from the network
 - * Cause: SP disabled, SIM in a non-Service Processor enabled device (non-fraud)
- * SP fails authentication with SC
 - * Cause: Invalid credentials, device spoofing
- * SC receives Usage Data Records (UDR) from SP after receiving "data session stopped" trigger from the network
 - * Cause: Device spoofing
- * Service Controller receives CDRs but does not receive UDRs
 - * Cause: SP disabled, SIM in a non-Service Processor enabled device (non-fraud, unless SP has authenticated with SC during this data session)
- * SC receives CDRs and UDRs but the usage counts don't align
 - * Cause: SP tampering
- * SC receives CDRs and UDRs but the UDRs indicate usage over Charging Policy (CP) limit(s)
 - * Cause: SP tampering
- * SC receives UDRs but charging codes do not correspond to CP(s) for current active services
 - * Cause: SP tampering, device spoofing

Fraud Scenarios (cont'd)



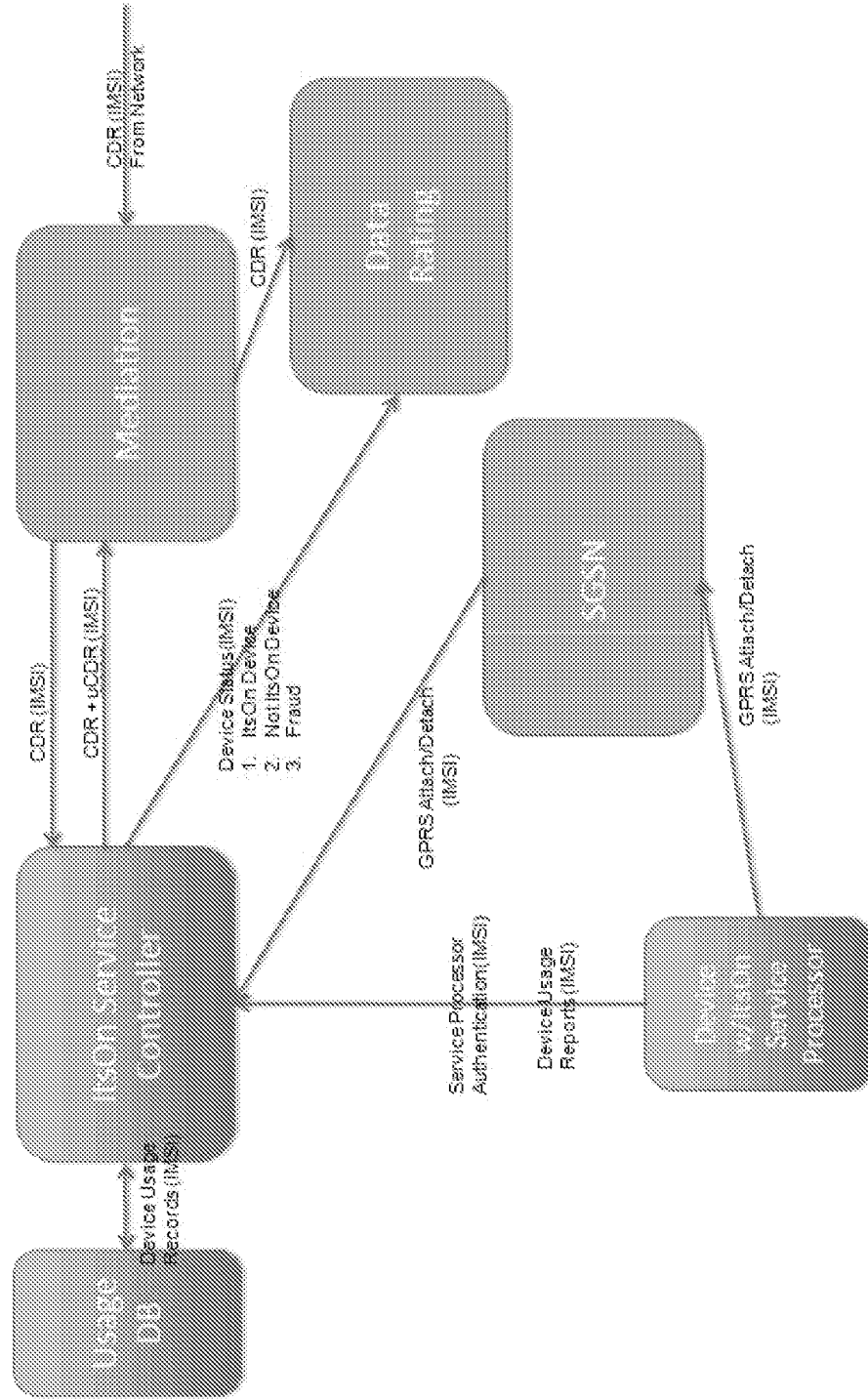
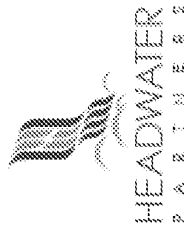
- SC receives CDRs and UDRs, counts align, charging codes are correct, but data is mis-categorized (needs FDRs to corroborate)
 - Cause: SP tampering
- SC receives CDRs and UDRs, counts align, but usage velocity within a Service Component or Service Activity is greater than the rate limit(s) set via the Control Policy
 - Cause: SP tampering
- SC receives CDRs and UDRs, counts align, but usage velocity at the Service Activity or Service Component level deviates "significantly" from average user usage velocity
 - Cause: SP tampering, Service Activity/Component usage patterns changing (e.g., service adds streaming content) (non-fraud)

Active Service Processor Verification

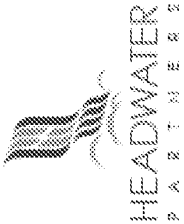


- Service Controller receives a start session message from the network
 - Could be a "Start Accounting" RADIUS message from a GGSN, AAA or PDSN, or successful "GPRS Attach" notification from an SGSN
- Service Controller sets an "Expecting Service Processor Login" timer
- If Service Controller successfully authenticates the Service Processor within the timer interval, Service Processor is deemed to be running and valid.
- If the Service Controller does not successfully authenticate the Service Processor within the timer interval, it is assumed that: 1) the device does not contain a Service Processor; or 2) the Service Processor on the device has been disabled.
 - When this event occurs, the Service Controller can either 1) trigger to the network to charge for usage at "standard" bulk rates; or 2) specify a "standard" bulk rate charging code in the CDRs sent to data mediation
- Following slides depict sample high-level implementation options of this functionality in a GSM/GPRS core data network

SGSN Notification of Start/Stop Data Session

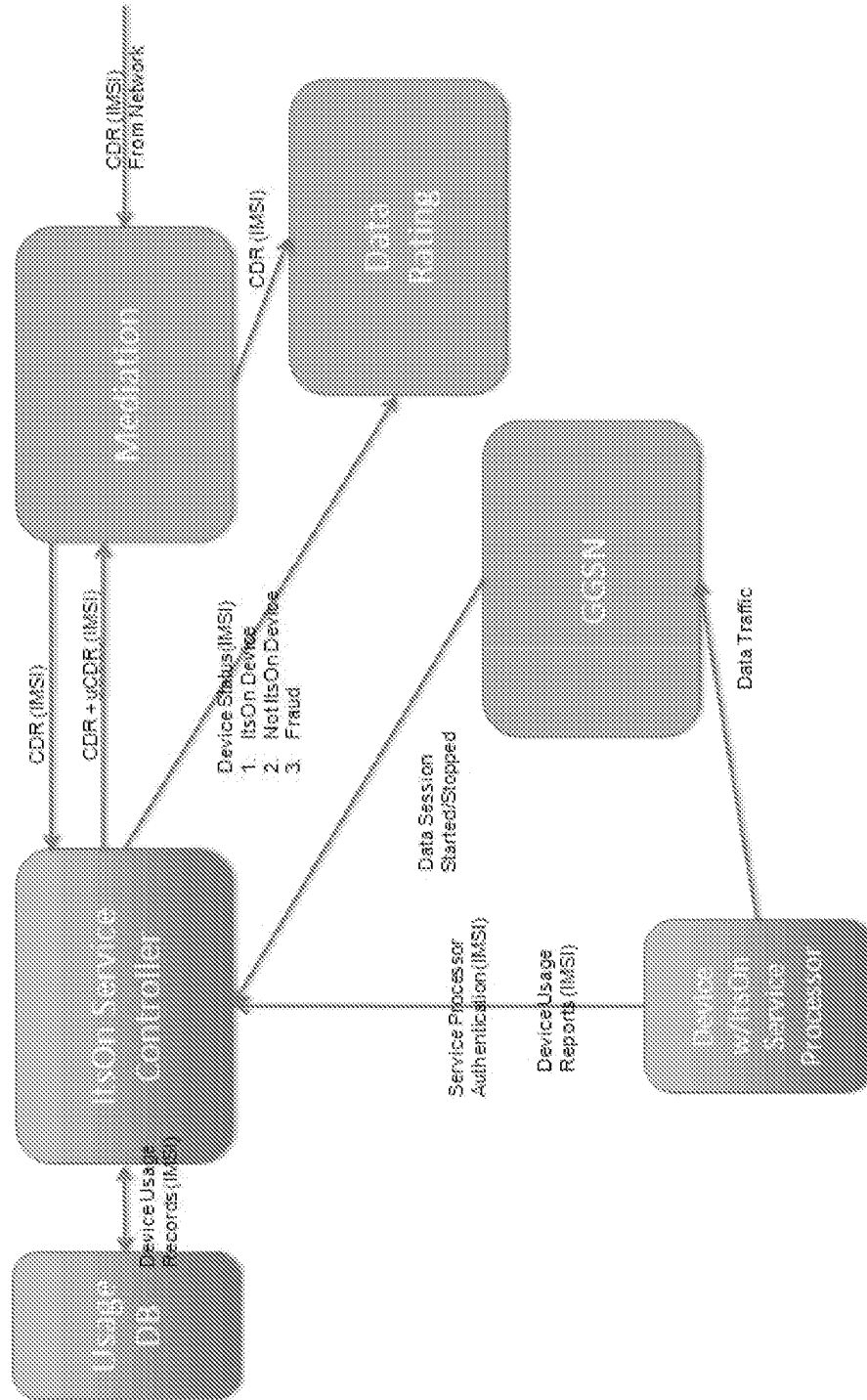
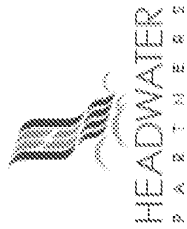


SGSN Notification of Start/Stop Data Session – Process



- Device attempts data session and sends GPRS Attach to SGSN
- SGSN notifies Service Controller that the device has started a data session
- Service Controller waits for a pre-determined time (e.g., one minute) to receive a login/authentication request from the Service Processor on the device
 - If the Service Controller sees the request, it sends a notification to Data Rating to indicate that the device has an Service Processor and expect Device Charging Record (DCR) reports via Data Mediation.
 - If the Service Controller does not see the request, it sends a notification to Data Rating to indicate that the device does not have an Service Processor and to process usage based on carrier CDR reports.
- Service Controller receives usage reports from the Service Processor
 - If Service Controller does not receive usage reports from the device based on time/usage intervals, it sends a notification to Data Rating to indicate that the device is in a Fraud state and usage should be billed based on carrier CDR reports.
- Data Mediation sends CDRs to Service Controller.
- Service Controller queries it's device usage database for DCRs for the device.
- Service Controller compares carrier CDR data usage byte counts vs.. DCRs.
 - If Service Controller determines that the carrier usage counts don't match the DCR usage counts (e.g. Fraud), it sends a notification to Data Rating to indicate that the device is in a Fraud state and usage should be billed based on carrier CDR reports.
- Service Controller sends carrier CDR and Service Processor DCRs to Data Mediation.
- When GPRS Detach is received by the SGSN, SGSN sends a notification to the Service Controller that the device data session is closed.

GGSN Notification of Start/Stop Data Session

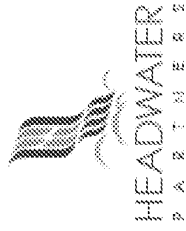


GGSN Notification of Start/Stop Data Session – Process



- * Device starts data session.
- * GGSN recognizes start of a new data session.
- * GGSN notifies Service Controller that the device has started a data session
- * Service Controller waits for a pre-determined time (e.g., one minute) to receive a login/authentication request from the Service Processor on the device
 - * If the Service Controller sees the request, it sends a notification to Data Rating to indicate that the device has an Service Processor and expect CDR reports via Data Mediation.
 - * If the Service Controller does not see the request, it sends a notification to Data Rating to indicate that the device does not have an Service Processor and to process usage based on carrier CDR reports.
- * Service Controller receives usage reports from the Service Processor
 - * If Service Controller does not receive usage reports from the device based on time/usage intervals , it sends a notification to Data Rating to indicate that the device is in a Fraud state and usage should be billed based on carrier CDR reports.
- * Data Mediation sends CDRs to Service Controller.
- * Service Controller queries it's device usage database for CDRs for the device.
- * Service Controller compares carrier CDR data usage byte counts vs.. CDRs.
 - * If Service Controller determines that the carrier usage counts don't match the DCR usage counts (e.g. Fraud), it sends a notification to Data Rating to indicate that the device is in a Fraud state and usage should be billed based on carrier CDR reports.
- * Service Controller sends carrier CDR and Service Processor CDRs to Data Mediation.
- * When GGSN determines that the data session has closed, it sends a notification to the Service Controller that the device data session is closed.

Service Processor/Service Controller Authentication



- * Device starts data session
- * Service Controller receives "Start Data Session" notification from Operator network and starts "Expected Login" timer for device
- * Service Processor on the device establishes data session with Service Controller
- * Service Controller cancels "Expected Login" timer
- * Service Controller attempts to authenticate Service Processor (algorithm TBD)
- * If authentication fails, Service Controller either 1) sends trigger to charge device data usage at "standard" bulk rate (implementation TBD based on operator network implementation); or 2) specifies a charging code for "standard" bulk rate on DCRs sent to operator data mediation in response to CDRs received from data mediation
- * If authentication succeeds, Service Controller marks the device as "Successfully Authenticated".
- * If "Expected Login" timer interval expires, Service Controller either 1) sends trigger to network to charge device data usage at "standard" bulk rate (implementation TBD based on operator network implementation); or 2) specifies a charging code for "standard" bulk rate on DCRs sent to operator data mediation in response to CDRs received from data mediation

SC Receives UDRs From SP After Receiving “Data Session Stopped” Trigger From Network

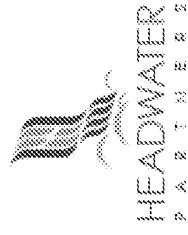
- Authenticated SP active on operator network
- SC receives “Data Session Stopped” trigger from operator network
- SC marks the SP as “inactive”
- SP attempts session with SC via operator network
- SC detects device status as “inactive”
- SC closes data session
- SC marks the device a “fraudulent”
- SC generates “potential fraud alert” to operator CRM
- How to handle UDR from WiFi and not 3G/4G?

Service Controller Receives CDRs But Does not Receive UDRs



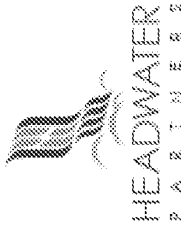
- * SC receives CDR from operator data mediation
- * SC attempts to verify usage by requesting all UDRs for the time period indicated in the CDR
- * No UDRs with charging codes associated with 3G/4G access are returned for the time period specified by the CDR
- * SC marks the device a “fraudulent”
- * SC generates “potential fraud alert” to operator CRM

SC Receives CDRs and UDRs But the Usage Counts Don't Align



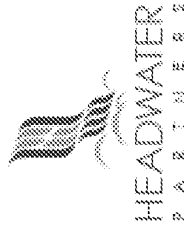
- SC receives CDR from operator data mediation
- SC attempts to verify usage by requesting all UDRs for the time period indicated in the CDR
- SC aggregates all of the usage counts associated with charging codes assigned to 3G/4G access
- SC verifies that UDR totals vs.. CDR total for the same time period are within the "fraud discrepancy" threshold.
- If the discrepancy is outside the "fraud discrepancy" threshold, SC marks the device a "fraudulent" and generates a "potential fraud alert" to operator CRM

SC Receives CDRs and UDRs but SC Detects Usage Over Charging Policy (CP) Limit(s)



- * SC receives CDR from operator data mediation
- * SC attempts to verify usage by requesting all UDRs for the time period indicated in the CDR
- * SC aggregates all of the usage counts associated with charging codes assigned to 3G/4G access
- * SC verifies that UDR totals vs.. CDR total for the same time period are within the "fraud discrepancy" threshold.
- * Usage counts verify
- * SC computes "total usage" per active Charging Policy
- * SC verifies that aggregate usage within each active Charging Policy is under the usage cap defined by the charging policy
- * If SC detects usage within an active Charging Policy that is over the usage cap, SC marks the device a "fraudulent" and generates a "potential fraud alert" to operator CRM

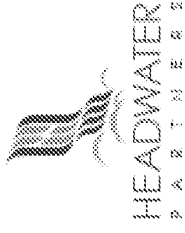
SC Receives UDRs but Charging Codes do not Correspond to CP(s) for Current Active Services



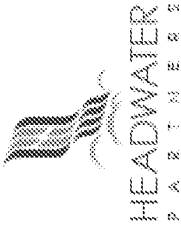
- * SC receives CDR from operator data mediation
- * SC attempts to verify usage by requesting all UDRs for the time period indicated in the CDR
- * SC aggregates all of the usage counts associated with charging codes assigned to 3G/4G access
- * SC verifies that UDR totals vs.. CDR total for the same time period are within the "fraud discrepancy" threshold.
- * Usage counts verify
- * SC detects charging codes not associated with CP(s) for current active services on the device
- * SC marks the device a "fraudulent" and generates a "potential fraud alert" to operator CRM

SC receives CDRs and UDRs, counts align, but usage velocity within a Service Component or Service Activity is greater than rate limit(s) set via CP

- * SC receives CDR from operator data mediation
- * SC attempts to verify usage by requesting all UDRs for the time period indicated in the CDR
- * SC aggregates all of the usage counts associated with charging codes assigned to 3G/4G access
- * SC verifies that UDR totals vs.. CDR total for the same time period are within the "fraud discrepancy" threshold.
- * Usage counts verify
- * SC detects usage associated with charging codes that have rate limits set is greater than the maximum rate limit for a continuous stream over the time period specified by the UDR (check performed on a UDR-by-UDR basis)
- * SC marks the device a "fraudulent" and generates a "potential fraud alert" to operator CRM

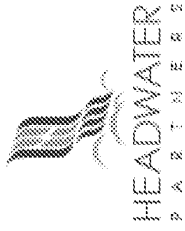


SC receives CDRs and UDRs, counts align, but usage velocity at the Service Activity or Service Component level deviates “significantly” from average user usage velocity



- * SC receives CDR from operator data mediation
- * SC attempts to verify usage by requesting all UDRs for the time period indicated in the CDR
- * SC aggregates all of the usage counts associated with charging codes assigned to 3G/4G access
- * SC verifies that UDR totals vs.. CDR total for the same time period are within the “fraud discrepancy” threshold.
- * Usage counts verify
- * SC detects usage velocity (bps) associated a Service Activity or Component over the time period specified by the UDR (check performed on a UDR-by-UDR basis) is “significantly” greater than average user usage velocity (bps)
- * SC marks the device a “fraudulent” and generates a “potential fraud alert” to operator CRM

CDR-based Verification Algorithm



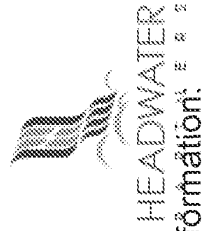
- * Assumes CDR input contains: start time, end time, MO bytes, MT bytes
- * Collect Service Processor Charging Reports from the device, order them by network time, store on the server
- * Collect CDRs from carrier for each device, order them by network time, store on the server
- * Verify device usage is consistent with the sum of all active service usage allowances (caps)
- * Verify that usage within services is within service-level caps
- * If a service has an "associative traffic" filter
 - * Verify that the ratio of associative traffic vs.. non-associative traffic is within service policy limits
 - * Verify that the ratio of traffic between the filters in the service are within policy limits
- * For the time window specified in a CDR:
 - * Sum device usage based on all of the charging reports received from the device for the same period
 - * For windows where the CDR and the charging reports don't match exactly, compute "average usage/minute" for the charging interval and add that to the device usage total
 - * Compare totals
 - ... If comparison is within <x>% everything is OK
 - ... If Charging Report < CDR by <y>% then either Service Processor reporting bug or potential fraud
 - ... If Charging Report > CDR by <z>% then Service Processor reporting bug, carrier reporting error or failed fraud attempt
 - * Generate micro-CDRs to the carrier based on time window specified in the original CDR

FDR-based Verification Algorithm

- Assumes FDR input contains: start time, end time, remote IP/port, MO bytes, MT bytes
- Perform CDR algorithm on FDRs to get totals
- If total usage is within policy, everything is OK so far
- For each “non-bulk-allow” service reporting during that period
 - Get service policy definition
 - Translate hostnames to IP addresses
 - Count all FDR matching Policy’s IP addresses
 - Count all device usage records for the service
 - If the service has an “associative traffic” filter
 - Verify that the ratio of associative traffic vs.. non-associative traffic is within service policy limits
 - Verify totals of non-associative filters vs.. FDR counts
 - Verify that the ratio of traffic between the filters in the service are within policy limits
 - Compare totals
 - If comparison is within <x>% everything is OK
 - If Charging Report < FDR by <y>% then either Service Processor reporting bug or potential fraud
 - If Charging Report > FDR by <z>% then Service Processor reporting bug, carrier reporting error or failed fraud attempt
- If there is a “bulk-allow” service
 - Sum the charging report totals for all non-bulk services (including associative traffic)
 - Subtract charging report total from carrier FDR total
 - Compare previous result with charging records for bulk service
 - If comparison is within <x>% everything is OK
 - If Charging Report < FDR by <y>% then either Service Processor reporting bug or potential fraud
 - If Charging Report > FDR by <z>% then Service Processor reporting bug, carrier reporting error or failed fraud attempt

Behavior-based Verification Algorithm

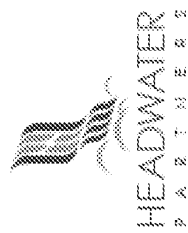
- * Service Controller keeps a database, based on service and filter that contains the following information:
 - * Service Name
 - * Filter Name
 - * Average use per time period (e.g. per day, actual value TBD)
 - * Average ratio of non-associative vs.. associative traffic for the service
 - * Expected ratio of filter matches vs.. total service usage
 - * Allowed percentage of variance (before indicating potential fraud event)
- * Service controller keeps a data warehouse to track both aggregated usage within a service and usage at the filter level to compute the above information
- * Periodically (e.g., daily), Service Controller compares user usage within a service and/or filter to the “average” usage data in the database
- * If user usage within the service/filter is outside the parameters specified for the service/filter, a “potential fraud” alert is generated
- * Example:
 - * Amazon service:
 - * 2 filters: amazon.com and images-amazon.com + associative
 - * Service cap of 10 MB/day
 - * Ratio of usage within the service is: amazon.com: 20%, images-amazon.com: 70%, associative: 10%
 - * Ratio of non-associative vs.. associative is 9:1 (or 90% non-associative)



Production Deployment Considerations

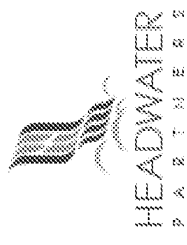
- * CDR-based Verification can be accomplished on all devices all of the time
 - * CDR feed delivered to Service Controller via operator Data Mediation
- * Behavior-based Verification can be accomplished on all of the devices on a daily basis
 - * Requires Service Processor usage reporting records only (no operator dependency)
 - * Can be augmented by FDR to validate the non-associative traffic counts within a service
- * FDR-based Verification should be used when there is a discrepancy in one or more of the following checks:
 - * Service Processor reported service-level usage is statistically higher than defined caps (e.g., Amazon Sponsored Service has a 10 MB daily cap, but the Service Processor is reporting 15 MB of usage within the daily period)
 - * CDR reported usage is statistically higher than Service Processor reported usage
 - * Average ratio of non-associative vs.. associative traffic within a service is statistically higher than expected
 - * Behavior-based Verification indicates usage within a service does not follow statistical norms
- * FDR-based Verification can also be used to randomly verify Service Processor usage reports at the service level (e.g., Amazon Sponsored Service usage reported by Service Processor matches FDR-based usage reports)
- * Ideally, retrieval of FDR's by Service Controller would be via a request mechanism (e.g., UserID, start time, end time)

Fraud Analysis Considerations



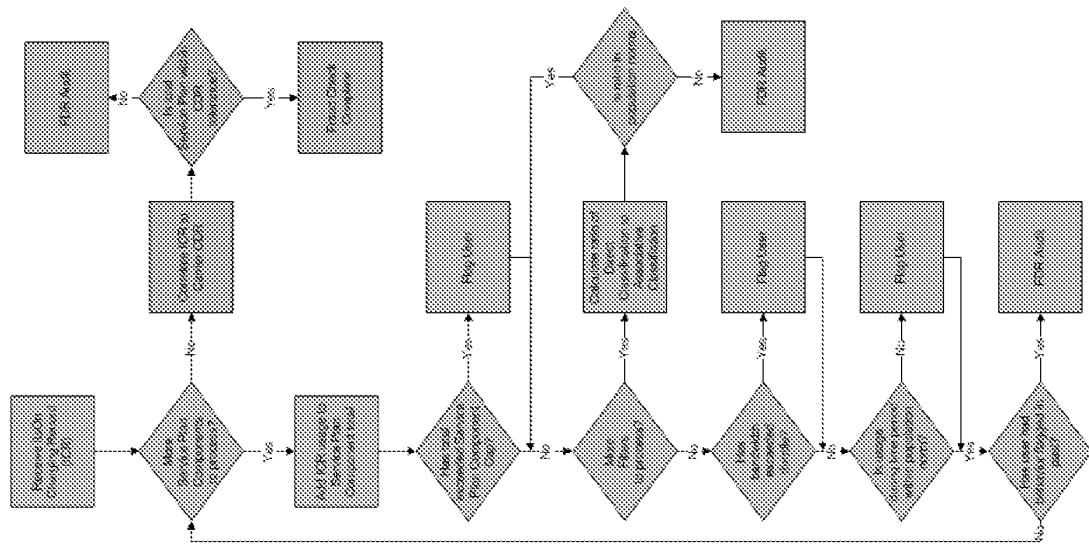
- Device Charging Record (DCR) verification is sufficient to detect the following fraud
 - Singly Defeating the Service Processor Controls
 - Service cap violation detected by comparing total usage to service maximum
 - Service throttle violation detected by comparing usage over time to expected bandwidth
 - Singly performing Charging Record Spoofing
 - Per filter bucketing of traffic classification method allows comparison to expected classification ratios
 - Service level ratio of non-associative vs.. associative traffic
 - Filter level ratio of filter matches as a percent of overall service usage vs.. expected percentage
 - Behavioral analysis allows usage over time period comparisons to user population statistics (at the service level)
- Carrier CDR-based verification is sufficient to detect the following fraud
 - Disabling the Service Processor
 - Carrier CDR's without Device Charging Records (DCR) signal DAS service interruption
 - Defeating Service Processor Controls & Performing Charging Record Spoofing
 - Carrier usage will show discrepancy from DCR reported usage
- Carrier FDR-based verification is sufficient to detect the following fraud
 - Substituting Sponsored Service charging for bulk service usage
 - Per filter totals of non-associative classification must match FDR totals otherwise fraud has occurred

Fraud Sampling

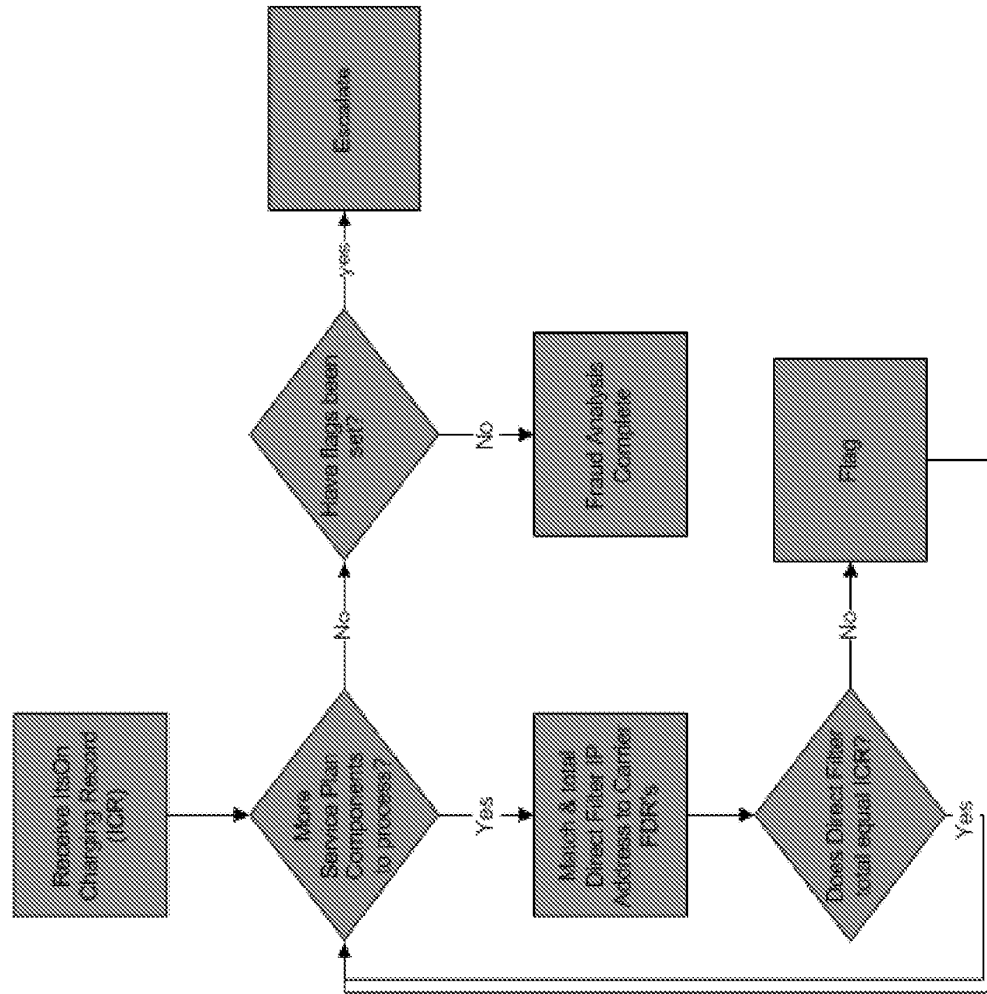


- Analysis of every DCR must be performed on every DCR received
 - If Service Processor is running in secure method/area on device this is sufficient.
- Carrier CDR validation must be performed if Service Processor not secure
 - Based on the specific Service Plan it may be a post process activity
 - Low caps on Services limit liability
 - User will be charged for data, the fine grain mediation will be lost
 - Can be addressed in the contract
 - CDR's may not be available in near/real time
 - Roaming
 - Near-real time CDR validation can be used when DCR validation detects problem
- Carrier FDR validation only needed when fraud audit is required
 - DCR & CDR validation sufficient for all but the most sophisticated fraud attempts

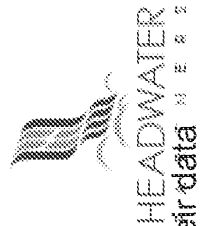
DCR & CDR Fraud Analysis Flow



FDR Fraud Analysis Flow

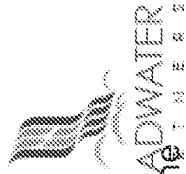


Reconciliation Algorithms for Fraud Detection



- ~ Traffic based profiling – looking at the average user (can be by demographic and profiling their data usage by app.
 - ~ Could use a SVN based implementation with boundary analysis to determine if the data and/or app usage is invalid
- ~ App to host based profiling
 - ~ Looking at average app to host/port based profiling. Look at usage of app to host destinations use known fraud and known good profiles to train some kind of AI algorithms to detect future fraud
- ~ Profiling based on connection times and applications for a specific user. For instance if a user is active 9am-1pm and then 6-10pm as connection between 2-5 might be suspicious
- ~ Profiling of data usage and variance statistically across a large user base of same demographic and doing abnormality filtering
- ~ On open web browsing could use Markov chains or similar approaches to predict likely application usage patterns therefore data usage around those apps and see if there is fraud going on
- ~ Could perform inactivity profiling on the device to see if the data usage is similar to a human profile vs. a machine. Same could be performed on the server using CDRs or FDRs, or Start/Stop sessions.
- ~ Analysis on access across devices. If a number of devices are access obscure or lesser know hosts could be fraud or just a set of compromised devices

Additional Embodiments to Secure Sponsored SIM With DAS Enabled



1. Don't zero-rate SIM, require pre-paid or credit card or method for billing to receive a SIM and the service agreement says that if you tamper with device you are billed at a relatively high (less favorable) rate, only provide credit for sponsored or specialized (non-bulk) services via valid SP usage reports, reconciliation of valid SP micro-CDRs against network CDRs occurs in service controller and "benefit of doubt" goes to user, service usage that can not be reconciled is charged to user at agreed upon contract rate. Can offer service this way for non-SP and SP devices. Can still have an experience with non-SP device, but experience is better with SP device.
2. Give out SIM with small usage cap, up cap as micro-CDRs come through, cap increase is determined by reconciliation of valid micro-CDRs against network CDRs in the service controller and "benefit of doubt" goes to user. Usage cap system should be able to respond in a time that is relatively small with respect to minimum time required to burn through cap at max modem data rate. Does not address box of free SIMs problem. Provide credit in reconciliation by either increasing cap or reducing charge against cap. Can offer service this way for non-SP and SP devices. Can still have an experience with non-SP device, but experience is better with SP device.
 - * Additional embodiment detail: can increase cap as soon as we confirm a valid MCDR flow from SP instead of waiting for MCDR usage to offset CDRs – this allows you to stay ahead of usage report delays and make sure a small cap does not shut the guy off before we can mediate out the usage. Detail: can send in the reconciled usage after getting network CDRs, or send credit to increase cap or decrement usage before receiving network CDRs.
3. User agreement that goes with sponsored SIM says that you can only use a certain number (e.g. one) free or sponsored SIM in a given device. Monitor device IMEI or other device credential and SIM credential. Use off-hook or first CDR received condition from network for SIM credential to determine if SP has authenticated and a valid MCDR sequence has been initiated. If a given IMEI or other device credential is burning through more than one free service usage cap with multiple SIMs, then send network a suspend or deny command based on IMEI or other device credential. Can also associate with user if user is providing user credentials somehow as part of device authentication sequence.

App Based Services Embodiments Outline

1. Motivation and top level embodiments

- a. Problem set definition: Service policy enforcement on access connection networks has traditionally been limited to network equipment centric solutions. More recently technology is being developed to enforce access network service policies on devices. This patent describes various embodiments to tie access network service policy enforcement to applications. App developers increasingly want to offer application based access network connection services that are adjuncts to or even independent of the access network carrier service plan status of a device or user. Examples: sponsored services, subscription services, shopping services, news services, ereader services, music services, camera/photo services, location services, social networking services, enterprise services, roaming services, etc. Presently can not get to user if user does not have a sufficiently large data plan. Many applications are profitable for the app developer and/or carrier if the app developer pays for service – providing an opportunity to be adopted by a much larger world wide consumer base. Many applications need much lower service plan usage levels and pricing than general purpose internet plans. Application developers are typically too small or not experienced enough with carrier network services to create MVNO solutions. Some applications benefit from billing splits between user and sponsor (e.g. enterprise split services). Access service awareness at the application level and/or application participation in policy enforcement at the application level can greatly enhance the scope of service policies that can be implemented and enforced. Need: way to allow application developers, carriers, application sponsors to define services as a bundle with applications. Need: application level service awareness and policy enforcement that can be verified. How to tie access network service policy enforcement to an app. By tie service policy enforcement we mean classify the service usage as belonging to the application, enforce policies to account for or charge for the service usage, enforce policies to control access connections and/or traffic on the access network and/or enforce notification policies for service usage, service status, service plan selection options or service fraud. By verify we mean to prevent tampering or fraud, or detect tampering or fraud so that action may be taken. When service policy enforcement is tied to and/or partly implemented by an application, it is crucial to secure the service against fraud techniques involving modifying, tampering with or re-writing the application software.
- b. Basic services model for distributing access services integral to applications: When a user chooses to install an app, or an OEM or carrier chooses to install an app on the device, the app comes with a pre-defined set of access network service plan access policy allowances bundled with the app. **Network system is able to identify a specific app and associate it with the correct access network service policies for one or more of access control, charging and/or service usage notification.** Different apps can have different service policies. The service payments can be embedded in the app purchase agreement or the service can be sponsored.
- c. In some embodiments, the carrier network service policy enforcement is able to automatically classify access network connections for a specific application on a

device and differentially control, charge for or notify the user about access network usage for that application.

- d. In some embodiments the application access network service policy enforcement is accomplished by the device and/or the device in coordination with the network or the application server. In some embodiments the application access network service policy enforcement is accomplished by the network. In some embodiments the application access network service policy enforcement is accomplished by the app server or the app server in coordination with the network. In some embodiments the app itself participates in service policy enforcement for one or more of access control policy, service accounting/charging policy, service usage notification policy.
- e. Basic services model for app participation in service plan provisioning and/or policy enforcement: Application communicates with, coordinates policy enforcement with or is monitored by one or more of (A) device service processor, (B) carrier network servers and/or (C) application sponsor servers to participate in access network service plan provisioning and implementation in one or more of the following areas: (i) access network service usage classification/accounting/charging, (ii) access network access control enforcement and/or traffic control policy enforcement, (iii) access network service user notification. Means are provided to verify that application is properly participating in service policy enforcement. Application may have programmable service policies that are updated by device, service controller/network or app server.
- f. Services distribution model 1: carrier controlled/offered services. Carrier creates a business model where application becomes an integral component of service classification, control, charging and notification. Application is integral to specialized "sponsored service plans or service plan components", and/or "application specific service plans or service plan components".
- g. Services distribution model 2: app sponsor controlled/offered services. App developer can become "app service sponsor". App service sponsor defines the services that go with an app, agrees to a service payment deal with a carrier. Carrier provides infrastructure that allows app service sponsor to pay for app access services or include app access services as part of app purchase agreement with end user.
- h. Services distribution model 3: app sponsor partner offered services. Partner of app sponsor works with app sponsor on "surf-out" basis. App sponsor offers user service activities that result in "surf-out" to app sponsor partners if user chooses the service activity (e.g. web site click off of sponsored service site, ad click off of sponsored service site, shopping and/or content purchase or other purchase transaction off of sponsored service site, etc).
- i. Services distribution model 4: app store becomes app service distributor to app sponsors – reduces or eliminates need for carrier to deal with all the app developer/sponsors, reduces or eliminates need for app developer/sponsors to create infrastructure to deal with carrier, allows app store to offer same app services across multiple carrier stores.
- j. Carrier provides for app services via pre-load of app or app that belongs to carrier specified service plan with carrier specified policies

- k. Carrier provides for app services via app sponsor belonging to qualified app services program
 - i. App sponsor in control of app policies
 - 1. Defined in app itself, SDC for app
 - 2. Defined in device service processor, SDC for app settings in service processor
 - a. API from service processor to define access policies and policy state for app
 - b. Service processor as primary implementer of service controls, charging
 - c. Service processor allows app to control services and count, service processor monitors service policy implementation for app, counts service usage and report, detects fraud
 - 3. Defined in App server, SDC for app server policies
 - a. Proxy server/gateway function for surf-out
 - b. SDC for proxy server/gateway function
 - ii. Carrier bills based on usage
 - iii. Carrier can also over-rule app policies depending on policy state variables (active network, TOD, NBS, fraud detection, etc)
 - iv. App based service policies implemented in app itself (hard to detect fraud since device and network may not know policies)
 - v. App based service policies are implemented on device
 - 1. App certificate can come with policy list for device programming
 - vi. App based service policies are implemented in network
- l. App store becomes main carrier partner, distributes app based service policies to individual apps in store per agreement with each app store app developer
 - i. App developer does have to deal with carrier infrastructure and app store is just a conduit for disseminating app based services to app store partners
 - ii. App store provider deals with carrier and app developer does not have to deal with infrastructure to work with carrier network
 - iii. NOTE: this is what Google will probably want.
- m. Various embodiments provide for differing levels of app awareness of app based service policy enforcement and various levels of app participation in policy enforcement
 - i. App awareness of app based policy enforcement is limited only limits access attempts to specific service usage required to run app and app usage restrictions are known to device, network or app server
 - 1. Very useful for early adoption of app based services because app developers do not need to change app to accommodate app based services distribution models
 - ii. App interacts with app based services system through API – device service processor app services API or network app services API
 - 1. Useful because apps do not get confused by differential access services available to different apps and apps can directly access service status information to adapt policies and implement user notification
 - iii. App participates in policy enforcement for one or more of charging, access control, service status notification

1. Useful for app developers or app sponsors to tightly control app access service policies

2. Verifying that an app belongs to an app based access service plan

- a. First level app fraud prevention/detection:
 - i. Signature checker/hash checker for app that is part of the OS or sits in secure OS execution – first fraud detection layer
 1. Confirm app signature/hash with known signature/hash stored in: service controller, download file on device, central authority
 - ii. Check app when it is loaded to confirm that it is the right app
 1. Possibly also check app each time it is launched and/or during app operation
 - iii. If app signature/hash is not correct, then suspend, kill, block app
 - iv. If app signature/hash is not correct, then notify service controller
 - v. Service controller or equivalent on carrier network maintains data base of valid signatures/hashes and corresponding service policies
 1. Distributes to device checker via push or pull
 2. Evaluates device checker hash result sent to server
 - vi. App server or equivalent on app sponsor or app store network maintains data base of valid signatures/hashes and corresponding service policies
 1. Distributes to device checker via push or pull
 2. Evaluates device checker hash result sent to server

3. App Based Access Control

- a. As part of an app based service plan or service plan component, app based service policy enforcement system is assigned a set of access control policies (traffic control policies) on device
 - i. App implements access control policies
 1. Policies implemented by app are programmable
 - a. Secure API
 - b. Secure programmable policy set
 - i. Pushed to app from app server, network, device
 - ii. Pulled by app from app server, network, device
 - c. Updated by device
 - d. Updated by network
 - e. Updated by app server (in this case, device charges app sponsor based on agreed upon usage rating rules)
 2. Restrict access to only those network destinations that support app
 - a. URL/Domain restrictions
 - b. White list
 - i. Known specific to app
 - ii. Known multi-use
 - c. Black list
 - d. Unclassified list
 - e. Report list usage counts
 - f. Analyze list usage counts
 3. App may be aware of various policy state variables
 - a. App determines variable state

- b. Device sets app variable state
- c. Network sets app variable state
- d. App server sets app variable state
- e. API informs app of variable state
- f. Active network
- g. NBS
 - i. Device measure
 - ii. Network measure
- h. Time of day
- i. Geographic location
- 4. Apply traffic controls based on:
 - a. Destinations
 - b. Content types
 - c. Protocols
 - d. Active network
 - e. NBS
 - f. Time of day
- 5. Surf-out access leases
 - a. Surf-out depth
 - i. # Domains
 - ii. # URLs
 - iii. # IPs/other address counts
 - iv. # Bytes
 - v. # seconds/minutes
 - vi. Off black list
 - vii. On white list
 - b. App counts surf-out traffic and reports
 - i. For purpose of fraud detection'
 - c. App determines allowed surf-out user click options
 - i. Highlight on web page display or UI display
 - ii. E.g. paid advertiser web site vs general search result
 - iii. Organize search results or surf-out click options based on who is paying for surf-out relationship
 - d. App provides app server or websites with information identifying app based service credentials
 - i. Credential indicates that service is app based
 - ii. I.D.s service configuration, app, app developer, app distributor, app service sponsor, carrier, device type, device/user credentials, active network, service policies, service charging information, etc
 - iii. Credentials identified by:
 - 1. Header
 - 2. Special side channel/packet
 - 3. Which server destination app goes to
 - a. SSL
 - iv. Web site can decide whether or not to accept access service connections and/or service access conditions
 - 1. Agrees to pay

- a. Sends signed credential
 - i. Credential checked by app
 - ii. Credential checked by device
 - iii. Credential checked by network server
 - iv. Credential checked by app server
 - b. Pre-agreed deal to pay if web traffic is served
 - v. Web site chooses optimized content for app based service configuration and/or business arrangements
 - vi. Web site provides good customer feedback
 - vii. Web site provides usage counts
 - viii. Web site provides transaction counts
 - ix. Web site provides new usage policy limits
- e. Bring back to main service UI state after lease expires
 - i. Provide notification of why brought back to main service state
 - ii. Provide option to roll over or purchase service if user desires to continue
- f. Automatically roll-over to user bucket when lease expires
 - i. Just roll over as part of service agreement
 - ii. Provide notification of rollover
 - iii. Provide option to roll over or return to main service state
 - iv. Provide notification of available plan purchase options if no user bucket exists or if another user choice exists
- g. Allow increased surf-out allowance based on good customer standing
 - i. Surf-out points spent during surf-out access
- h. Surf-out controlled by app sponsor proxying service for surf-out lease
 - i. App server becomes proxy server for surf-out service access
 - ii. Proxy server first authenticates or determines app credentials or device credentials as above
 - iii. Proxy server can determine what rules to put in place
 - iv. Proxy server can account for surf-out charges to app sponsor partners
 - v. Proxy server can determine what links to highlight and what links to de-emphasize or remove
 - vi. Proxy server can add header information (or other means) to identify that transaction is sponsored and/or to identify one or more aspects of app, device or user credentials
 - vii. Proxy server can inject advertisements or other content into web pages served back to device
 - viii. Proxy server can determine good customer standing

- ix. Proxy server can receive good customer feedback from app sponsor partner servers to change app surf-out access policies for one or more sponsored services
- 6. Count service usage
- 7. Count content transactions
 - a. To device agent
 - b. To network server
 - c. To app server
- 8. Report service usage or transactions
 - a. To device agent
 - b. To network server
 - c. To app server
- 9. Multi-service application
 - a. Count service usage and associate to correct service based on which service is being accessed – differentiate usage counts
 - b. Count transactions for each service
 - c. Report
 - d. Self-contained service app in multi-service app
 - e. Launch external service app from multi-service app
 - i. External app aware: Count service usage, count transactions, report within launched app
 - ii. External app not aware: Count service usage, count transactions in an agent outside of app
 - 1. Stack API
 - a. API replacement
 - 2. Stack API shim
 - a. API shim plus app wrapper to make app think it is seeing same API instructions that rest of device apps are seeing
 - 3. Route traffic to counter app
 - 4. Kernel space stack sidekick/interceptor/driver
 - 5. Modem bus driver agent
 - 6. Modem agent
- ii. Device implements access control policies
 - 1. Classifies traffic by application and applies appropriate access policy rules for that application
 - a. Capability to provide differential access control policies for different applications
 - 2. Monitors app access behavior
 - a. E.g. FDRs based on domain, url, ip, port, protocol, etc. with time stamp, NBS state, active network, location, etc.
 - 3. Reports app access behavior to service controller
 - 4. Device compares policies against behavior as a second fraud detection layer
 - a. Compare FDRs to white list
 - b. Known app specific destinations
 - c. Known shared app destinations
 - d. Compare app to black list

- e. Compare app access behaviors to known fraudulent detection patterns
- f. Cap app

4. App Based Services Notification

- a. App includes design elements for an integral service usage notification system within app code
 - i. App code designed to track service usage and service activity trigger events that kick off service notification sequences
 - ii. Carrier or app store sponsor publishes app design specs for service usage notification
- b. App includes design elements for an API for service processor service status updates
 - i. API provides app with information that app then displays to user directly or with additional processing
 - ii. Device service processor sends notice of service usage or service status change to app through API
 - iii. App polls device service processor API to determine changes in service usage or service status
 - iv. Carrier or app store sponsor publishes service processor app based services API
- c. App includes design elements for an API for network based service status updates
 - i. API provides app with information that app then displays to user directly or with additional processing
 - ii. Network sends notice of service usage or service status change to app through API
 - iii. App polls network API to determine changes in service usage or service status
 - iv. Carrier or app store sponsor publishes app based services network API
- d. App includes service plan sign up or service plan upgrade or service plan change platform integral to app
- e. Service notification sequences and trigger events:
 - i. Notify at a given point in service usage allowance – example activity trigger: app usage hits X% of app usage allowance for given time window
 - ii. Notify on app cap – example activity trigger: usage hits app service usage allowance for given time window
 - iii. Notify of app usage levels, remaining service, usage velocity meter – example trigger: upon usage update from app, device service processor, secure device monitor, or network usage meter, remaining service meter and/or velocity meter are updated
 - iv. Notify of possible service plan changes – example triggers: if current plan does not suit app usage patterns, or if app is consistently hitting usage limitations due to app usage patterns, or if app is using allowance at a velocity that is better suited to another service plan
 - v. Notify user of service status of app specific service – example triggers: active network change; network availability change; network congestion, performance or busy state change; roaming condition
 - vi. Notify user of service plan options for app specific service – example triggers: user hits service plan cap, user does not have an app service plan in effect and user attempts to use app, user requests service plan option information

- vii. Notify user of billing status for app specific service
- viii. Notify user when fraud is detected
- ix. Notify user input on service plan sign up or changes
 - x. Notify user with self-help screens for access network service trouble shooting
- xi. Notify user with communication to app service support resources or personnel
- xii. Notify user of “good customer service credit standing”
- xiii. Notify of “good customer service credit building opportunities”
- xiv. Notify user of “good customer service credit spending opportunities”

5. Good Customer Feedback

- a. Good customer standing to modify app policies provided by feedback from app server
 - i. App server identifies app/device/user credentials/service plan or plan component configuration and/or charging rules
 - 1. App provides app server or websites with information identifying app based service credentials
 - a. Credential indicates that service is app based
 - b. I.D.s service configuration, app, app developer, app distributor, app service sponsor, carrier, device type, device/user credentials, active network, service policies, service charging information, etc
 - c. Credentials identified by:
 - i. Header
 - ii. Special side channel/packet
 - iii. Which server destination app goes to
 - 1. SSL
 - d. App server can decide whether or not to accept access service connections and/or service access conditions
 - i. App server can agree to pay
 - 1. Sends signed credential
 - a. Credential checked by app
 - b. Credential checked by device
 - c. Credential checked by network server
 - d. Credential checked by app server
 - 2. Pre-agreed deal to pay for served traffic
 - e. App server site chooses optimized content for app based service configuration and/or business arrangements
 - ii. App server can identify app access specific to service plan or plan component
 - iii. App server monitors user purchases and/or transaction counts
 - iv. App server monitors user activities that are beneficial to app distributor and/or other party (carrier, MVNO, 3rd party customer of app developer, etc)
 - 1. Purchases
 - 2. Sponsored usage or viewing activities
 - 3. Ad views, clicks, revenues
 - 4. CRM data to mobile device marketing/advertising platforms

- v. App server monitors usage that is detrimental to use model – can reduce caps and/or access control policy levels
 - vi. API from network to app to modify app policies and/or report customer activity/standing
- b. Good customer standing to modify app policies provided by app
 - i. Same as above under app server
 - ii. API between app and policy controls on device
 - iii. App reports standing to app server
- c. Good customer standing to modify app policies provided by device monitor
 - i. Same as above under app server
- d. Good customer standing can be applied to an individual service based on good customer activity on that particular service, or good customer activity on one or more services can be applied to some other service's good customer standing
 - i. E.g. someone who buys on line for one service may be a good customer for another service to increase access allowances since they are more likely to buy there
 - ii. E.g. an app sponsor who receives good customer feedback for one service may use that credit to sponsor additional surfing for other services
- e. Change app caps based on good customer activity
- f. Change app access policy levels based on good customer activity
- g. Provide good customer access allowance points to app or device based on good customer activity
- h. Provide device user with a notification UI for good customer standing
 - i. Notify of standing
 - ii. Notify of remaining usage allowance
 - iii. Notify of activities that user can conduct to increase good customer standing
 - iv. Allow user to increase standing by using other service allowance
 - v. Allow user to increase standing by paying for additional allowance

6. App Based Service Accounting and/or Charging

- a. App is assigned a set of classification, accounting, charging and reporting policies
 - i. Traffic usage classification
 - 1. Classify usage based on service used by app
 - a. Classify multiple service app usage by each service used by app
 - ii. App reports to service controller/ network charging system
 - 1. Service controller/network charging system API
 - iii. Service controller/network charging system reports to app sponsor server
- b. App server is assigned a set of classification, accounting, charging and reporting policies
 - i. Traffic usage classification
 - 1. Classify usage based on services served to app credentials
 - 2. Classify usage based on services served to device credentials
 - 3. Classify usage based on services served to user credentials
 - ii. App server reports to network charging system
 - iii. App server keeps local records
 - iv. Credit system – device/user account credited for app services that are served by app server – third level of fraud detection

1. App can be configured to only point to app server
 - a. Fraudulent traffic is not credited and is therefore charged to user account
 - b. Reconciliation determines if reported app traffic being used by device does not match app server reports – signals fraud event
- c. Network charging system is assigned a set of classification, accounting, charging and reporting policies
 - i. Traffic usage classification
 1. Classify usage based on device credentials and services communicated with a given network destination
- d. Reconciliation and fraud detection
 - i. Compare one trusted measure vs another measure:
 1. Network vs app
 2. Network vs app server
 3. Network vs device service processor
 4. Secure device vs app
 5. Secure device vs app server
 6. Secure device vs device service processor
 7. Classify usage patterns by
 - a. Known specific to app
 - b. Known used by multiple apps
 - c. Unknown
 - d. Black listed for app
 - e. App usage patterns for unknown
 - f. Black listed usage patterns
 - g. App traffic usage vs traffic control policies that should be in place
 - i. Tag usage records by time of access, access control policy intended to be in place at that time, NBS at that time, active network at that time, location at that time, etc
 1. Device sometimes knows more of this than network or app server, so there is sometimes a need to get a second measure other than service processor or app
 - a. Secure device FDR tags
 - b. Service controller NBS tests via device agent
 - i. Device agent gets traffic priority for test
 - c. Service controller active network testing
 - d. Service controller communication with secure device agent
 - i. Secure APU, modem driver, modem
 - e. Monitor network CDR/FDR patterns
 - i. Record network measures of active network, NBS, etc at time of CDR/FDR measurements
8. Fraud detection methods

- a. Secure usage measure vs policy that should be in place
 - i. Given secure device usage reports and secure measures of network state (TOD, NBS, etc), compare inferred access policies (e.g. destination, allow/block, speed, etc) vs policy that should have been in place given the service plans that are in effect at the time of usage measurement

 - 1. Compare usage by device vs usage that can be credited to valid app services over a given time period
 - a. Monitor patterns of usage by device vs usage that can be credited to valid app services over multiple time periods to detect consistent policy violations
 - 2. Compare patterns in unclassified usage reported by secure measures
 - a. E.g. consistently high levels of unclassified traffic in secure measures or unsecure measures
 - b. E.g. bursty levels of unclassified traffic in secure measures or unsecure measures
 - c. More examples...
 - 3. Analyze black listed usage patterns
 - a. E.g. Existence of black listed usage pattern in secure or other measure when no service plan is in place to support
 - 4. Usage can not be directly correlated between the policy enforcement point and the reconciliation analysis point since there will be a certain error between one usage measure and another.
 - a. Provide allowance or tolerance for usage measures
 - 5. Usage can not be directly compared to policy since there will be a portion of traffic that can not be classified as accurately with one measure as it was with another measure (e.g. usage by app).
 - a. Provide allowance or tolerance for unclassified traffic in one or both measures
- b. Verify app usage measure, compare app usage measure with policies that should be in place ###
 - i. Given app report (possibly with tagging) of device usage, use second measure (e.g. trusted/secure report from network, secure device, app server) to verify app usage report
 - ii. Trigger fraud error if app usage report does not check out
 - iii. If app usage report checks out, then use app usage report to compare inferred access policies (e.g. destination, allow/block, speed, etc) vs policy that should have been

in place given the service plans that are in effect at the time of usage measurement

1. Same sub-bullets as above under ***
- c. Verify device measure, compare app usage measure with policies that should be in place
 - i. Same sub-bullets as above under ### but verify device based measure for app usage instead of app based measure for app usage
 - d. Compare app server measure with second measure
- ii. Additional method to eliminate fraud in app based services
 1. Use app server measure as credit to user account
 - a. User pays for any usage above cumulative credits from app servers.
 - i. Paid for with debit to bulk usage account.
 - ii. Paid for with overage payments from user.
 - iii. Reconciliation for carrier to app sponsor billing purposes:
 1. Carrier charges app sponsor based on reconciled measures of usage
 2. Algorithms examples:
 - a. Choose most trusted measure of app service usage when discrepancy exists
 - b. Choose lowest usage measure of app service usage when discrepancy exists
 - c. Bill to
 - d. Bill to user when fraud is detected

7. Additional network centric embodiment:

- a. App requests service through network API
 - i. API on device
 - ii. API on network
- b. Network instructs device to hash app and confirm that it is valid
- c. Provided app is valid, network instructs device to let it on
- d. Network based fraud embodiments as above

Electronic Patent Application Fee Transmittal				
Application Number:				
Filing Date:				
Title of Invention:		APPLICATION SERVICE PROVIDER INTERFACE SYSTEM		
First Named Inventor/Applicant Name:		Gregory G. Raleigh		
Filer:		Michael J. Schallop		
Attorney Docket Number:		RALEP040+		
Filed as Large Entity				
Provisional Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Provisional application filing	1005	1	220	220
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Total in USD (\$)				220

Electronic Acknowledgement Receipt	
EFS ID:	8942246
Application Number:	61418507
International Application Number:	
Confirmation Number:	1493
Title of Invention:	APPLICATION SERVICE PROVIDER INTERFACE SYSTEM
First Named Inventor/Applicant Name:	Gregory G. Raleigh
Customer Number:	21912
Filer:	Michael J. Schallop/Veronica Pula
Filer Authorized By:	Michael J. Schallop
Attorney Docket Number:	RALEP040+
Receipt Date:	01-DEC-2010
Filing Date:	
Time Stamp:	13:57:12
Application Type:	Provisional

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$220
RAM confirmation Number	106
Deposit Account	500685
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Provisional Cover Sheet (SB16)	RALEP040plus_Provisional_Coversheet.pdf	999993 74182d86945e202dd5e60151b3c4a570d4fd8c32	no	3
Warnings:					
Information:					
2	Specification	RALEP040plus_App.pdf	2190281 fffd135a18c0a0cab946221fac9464597463a6cd	no	94
Warnings:					
Information:					
3	Fee Worksheet (PTO-875)	fee-info.pdf	29173 1cfca3e717788592374a2030b01fd688ef5220e	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			3219447		
This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503. <u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application. <u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course. <u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.					