



(51) International Patent Classification:

H04L 29/02 (2006.01) H04L 12/16 (2006.01)
G06F 17/00 (2006.01) G06Q 30/00 (2012.01)

(21) International Application Number:

PCT/CA2013/000579

(22) International Filing Date:

17 June 2013 (17.06.2013)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant: **CONTAC SERVICES INC.** [CA/CA]; 6600 Dennett Place, Delta, B.C. V4G 1N4 (CA).

(72) Inventors: **PISANI, Riaz**; c/o Contac Services Inc., 6600 Dennett Place, Delta, B.C. V4G 1N4 (CA). **MALCOM, Paul**; c/o Contac Services Inc., 6600 Dennett Place, Delta, B.C. V4G 1N4 (CA). **GRIJZENHOUT, Dan**; c/o Contac Services Inc., 6600 Dennett Place, Delta, B.C. V4G 1N4 (CA).

(81) Designated States (unless otherwise indicated, for every kind of national protection available):

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available):

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

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR GENERATING PERSONALIZED WEBSITES

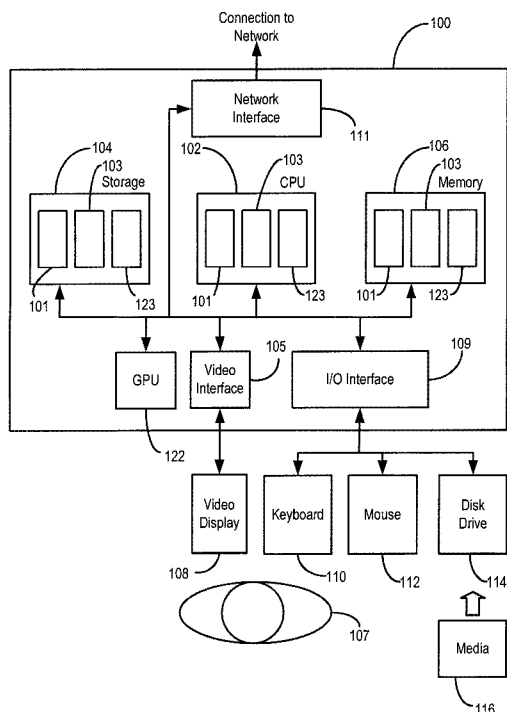


FIG. 1

(57) Abstract: The present disclosure relates to a system and method for generating a personalized universal resource locator (PURL) for a customer. In an aspect, the method comprises: identifying a unique customer; collecting content data from a plurality of data sources; pre-caching the content data for access by the unique customer; and generating a PURL with one or more links to the pre-cached content data prepared for access by the unique customer. The content data is collected in one or more landing databases, and pre-cached in one or more staging databases prior to access by the unique customer based on the unique customer's profile. One or more stages of a life-cycle may be identified for a product or service in relation to the unique customer to determine the content data to be pre-cached and to be rendered in the PURL.

WO 2014/201537 A1

Published:

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

SYSTEM AND METHOD FOR GENERATING PERSONALIZED WEBSITES

FIELD OF THE INVENTION

The present disclosure relates generally to a system and method for generating personalized websites.

BACKGROUND

Presently, companies with an online presence may make attempts to directly market to Customers by collecting information about a Customer's preferences, and providing suggestions for related goods or services that correlate to the user's preferences. For example, some online retailers will make suggestions for related goods or services as a Customer browses an item on a website, based on what the Customer has purchased before or based on what other Customers have purchased after browsing the same item online.

While attempts to customize a website for various users have been made, such customization has been limited as it has been impractical to make a website that is truly personalized for each individual Customer.

Therefore, what is needed is an improved system and method for addressing at least some of these limitations in the prior art.

SUMMARY

The present disclosure relates to a system and method for generating a personalized uniform resource locator (PURL) for a Customer that is regularly updated. In an embodiment, the PURL is regularly updated over an extended period of time for the life-cycle of products or services delivered to a Customer.

In an aspect, there is provided a computer-implemented method for generating a personalized universal resource locator (PURL") for a customer, comprising: identifying a unique customer; collecting content data from a plurality of data sources; pre-caching

the content data for access by the unique customer; and generating a PURL with one or more links to the pre-cached content data prepared for access by the unique customer.

In an embodiment, the computer-implemented method further comprises collecting the content data in one or more landing databases.

In another embodiment, pre-caching the content data comprises preparing the content data in one or more staging databases prior to access by the unique customer.

In another embodiment, pre-caching the content data further comprises preparing the content data based on the unique customer's profile.

In another embodiment, the computer-implemented method further comprises retrieving the pre-cached content data from the one or more staging databases in dependence upon interaction by the unique customer with the one or more links to the pre-cached content data in the PURL.

In another aspect, there is provided a computer-implemented system for generating a personalized universal resource locator (PURL") for a customer, the system adapted to: identify a unique customer; collect content data from a plurality of data sources; pre-cache the content data for access by the unique customer; and generate a PURL with one or more links to the pre-cached content data prepared for access by the unique customer.

In an embodiment, the system is further adapted to collect the content data in one or more landing databases.

In another embodiment, the system is further adapted to pre-cache the content data by preparing the content data in one or more staging databases prior to access by the unique customer.

In another embodiment, the system is further adapted to pre-cache the content data by preparing the content data based on the unique customer's profile.

In this respect, before explaining at least one embodiment of the system and method of the present disclosure in detail, it is to be understood that the present system and

method is not limited in its application to the details of construction and to the arrangements of the components set forth in the following description or illustrated in the drawings. The present system and method is capable of other embodiments and of being practiced and carried out in various ways. Also, it is to be understood that the phraseology and terminology employed herein are for the purpose of description and should not be regarded as limiting.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a schematic block diagram of a system architecture in accordance with an embodiment.

FIG. 2 shows a schematic block diagram of a generic computer device in accordance with an embodiment.

FIG. 3 shows a schematic block diagram of a macro level depiction of the system from a Customer facing perspective.

FIG. 4 shows a schematic block diagram of an architecture for rendering personalized URLs to a Customer website.

FIG. 5 shows a schematic block diagram of a PURL skin architecture in accordance with an embodiment.

FIG. 6 shows a schematic flow chart of a PURL generation system flow in accordance with an embodiment.

FIG. 7 shows a schematic block diagram of a "services bus" architecture in accordance with an embodiment.

FIG. 8 shows a schematic flow chart of a booking e-mail system flow in accordance with an embodiment.

FIG. 9 shows a schematic flow diagram that illustrates a content data rendering system and method in accordance with an embodiment.

FIG. 10 shows a schematic flow diagram that illustrates a content data normalizer and learning engine in accordance with an embodiment.

FIG. 11 shows a schematic flow diagram that illustrates how third party content is gathered and rendered to a Customer PURL with an embodiment.

FIG. 12 shows a schematic flow diagram of an illustrative airline booking process for a travel excursion life-cycle in accordance with an embodiment.

DETAILED DESCRIPTION

As noted above, the present disclosure relates to a system and method for generating a personalized website or URL ("PURL") for a Customer, and more particularly to a PURL that is regularly updated over the life-cycle of a product or service.

For the purposes of the present discussion, the "Client" refers to the client of a system vendor providing the present system and method. The "Customer" or Customers" refers to their buyers or end users of the Client's services. It will of course be appreciated that in an alternative embodiment the Client may itself be the system vendor.

In an aspect, there is provided a system and method for generating personalized websites that are highly customized for a particular identified Customer based on a Customer profile. This is made possible by a system architecture which significantly reduces direct end user load upon a core database system, and instead utilizes a series of Landing or Staging databases or Data Marts that are separate from the core system. These Landing or Staging databases are used to pre-stage a PURL for a Customer and can be dispersed across different clustered servers to be able to harness more power to serve up data and web pages to many more concurrent users without appreciable performance degradation. Thus, the system and method effectively separates components of the database by function point so as to reduce the I/O footprint to the core database system itself – reducing direct end user loads upon it.

In another aspect, the system and method collects most third party data to be served up locally from local servers, as opposed to having to serve up this data remotely from third parties directly to the user while they are in an on-line session. This may drastically

reduce screen rendering time to a few seconds to display third party data on personalized websites.

The system and method will now be described in more detail with reference to a number of illustrative drawings.

Referring to FIG. 1, the present system and method may be practiced in various embodiments. A suitably configured computer device, and associated communications networks, devices, software and firmware may provide a platform for enabling one or more embodiments as described above. By way of example, FIG. 1 shows a generic computer device 100 that may include a central processing unit ("CPU") 102 connected to a storage unit 104 and to a random access memory 106. The CPU 102 may process an operating system 101, application program 103, and data 123. The operating system 101, application program 103, and data 123 may be stored in storage unit 104 and loaded into memory 106, as may be required. Computer device 100 may further include a graphics processing unit (GPU) 122 which is operatively connected to CPU 102 and to memory 106 to offload intensive image processing calculations from CPU 102 and run these calculations in parallel with CPU 102. An operator 107 may interact with the computer device 100 using a video display 208 connected by a video interface 105, and various input/output devices such as a keyboard 110, mouse 112, and disk drive or solid state drive 114 connected by an I/O interface 109. In known manner, the mouse 112 may be configured to control movement of a cursor in the video display 108, and to operate various graphical user interface (GUI) controls appearing in the video display 108 with a mouse button. The disk drive or solid state drive 114 may be configured to accept computer readable media 116. The computer device 100 may form part of a network via a network interface 111, allowing the computer device 100 to communicate with other suitably configured data processing systems (not shown). One or more different types of sensors may be used to receive input from various sources.

The present system and method may be practiced on virtually any manner of computer device including a desktop computer, laptop computer, tablet computer or wireless handheld. The present system and method may also be implemented as a computer-readable/useable medium that includes computer program code to enable a computer

device to implement each of the various process steps in a method in accordance with the present invention. It is understood that the terms computer-readable medium or computer useable medium comprises one or more of any type of physical embodiment of the program code. In particular, the computer-readable/useable medium can comprise program code embodied on one or more portable storage articles of manufacture (e.g. an optical disc, a magnetic disk, a tape, etc.), on one or more data storage portions of a computing device, such as memory associated with a computer and/or a storage system.

Now referring to FIG. 2, shown is a schematic block diagram of a system architecture 200 in accordance with an embodiment. As shown, the overall operational system 200 contains a number of inter-relating Abstraction Layers that cohesively and seamlessly communicate with each other to deliver the desired functionality at the desired levels of performance and scalability. For purposes of presentation and discussion, these layers can be categorized as follows: (i) Integrated Systems Architecture; (ii) Application Architecture; (iii) Enterprise Data Environment; and (iv) Enterprise Computing Environment.

Key components of this logical architecture include a "Client Configuration" Module which creates Client profile settings in a Client configuration database. Settings in this profile create the rules for accessing third party data stores for that Client, its sales agents and its Customers.

In an embodiment, processing rules for all queries are stored as callable stored procedures in the system's Rules Engine. The system may further include a Configuration/Setup module which Clients will be accessing to perform various functions.

In a travel or event application context, many of the processes in the system are triggered by the receipt of booking event information. In an embodiment, the system includes a "WEB API" to receive booking information in the formats preferred by the system. Alternatively, custom retrieval and initial processing may be designed and coded into a custom front-end retrieval process.

Personalized URL or PURL screen activity also triggers system activities, and the Rules Engine is a repository of common "Stored Procedures" with backend database interfaces and "Processing rules" that are triggered as PURL events occur. Although there are certain PURL related activities that do not require use of the Rules Engine (such as queries made via hyperlink to certain third party services such as weather information, or screen changes, etc.), it is expected that many queries and business transactions will require internal systems decision making and sorting of data to be retrieved. Much of this is driven by Client configurations, so the Client Configuration profiles drive which stored procedure contained within the Rules. Engines are empowered to service the related query, as described in detail further below.

Working in concert with the "Rules Engine", the Content Server manages the final selection and rendering of information to a PURL when an end user event is triggered. One important function of the Content Server is in its relationship with Booking Notifications. When the system receives a Booking Notification from a Client, the system will send out a marketing e-mail to the end user Customer. The Customer receiving the e-mail then has the opportunity to click on a link that will take him to a custom generated PURL specific to him/her. The Content Server will generate the custom PURL and will render the appropriate content to the created PURL. This related content to the Home PURL page will be "fetched" from internal databases. This promotes a very fast loading to a PURL "Template/Skin" for the Customer, and will also help to minimize overall interactions with internal databases thus helping to maintain higher levels of system performance for all users. The Content Server also manages the retrievals, sorting, selecting and rendering of content back to the PURL screens on user initiated queries.

The mail server manages the distribution and tracking of all e-mails sent out by the system to Customers of the system. Beyond standard distribution capabilities as found in COTS (Common Off The Shelf Systems), this server processes outbound marketing messages created by the system. It also tracks message "Opens" for reporting purposes and further contact with Customer sales messages that can be sent out. One solution deployable by the system is discussed in more detail further below.

The system is a multi-language system at the PURL level. Initial deployment incorporates language translation capabilities for all deployed web pages using available tools (e.g. Google Translate) but the underlying Architecture Database has been designed for multi-language deployment which can be turned on at any time as desired. This database supports the changing of text at the PURL levels – both for Customers and for sales agents. At the PURL Level, an end user can thus change the language, and the web page will re-display itself as per the choice made by the end user.

In an embodiment, the system module takes revenues earned from Customers via both internal and external shopping carts and, based on revenue sharing rules built with Clients (see Client Configuration Module), allocates commissions to all revenue sharing parties. This system “renders” in a reporting fashion, revenue share information to sales agent dashboards, Client Activity web page and an internal reporting web page so all parties can view sales results and their share of revenues. In an embodiment, this also interfaces with the “Account Reconciliation” module to reconcile when purchases are modified – such as when there are hotel cancellations, or stays are shorter than originally booked/paid for, etc. This revenue splitting feature also directly ties the Agent to an integrated, third party payment service capability which they can use at their request within the system to trigger physical payouts to themselves globally.

In an embodiment, the platform includes an internal Shopping Cart interfaced with a payment mechanism and setup to also interface with a payment authorization system, such as *Authorize.net*, as needed for merchant account processing. From this shopping cart, a service provider can process financial transactions for third party purchases and can even push financial transactions, hotel booking confirmations and third party shopping cart purchase to external business partners where business rules warrant it. This shopping cart is able to obtain Client purchase information that is then analyzed for “gross profit margin” amounts so that these amounts can be distributed among the stakeholders when revenue sharing procedures built into the system take place.

In an embodiment, the system defines certain business rules for business events. An example is – who does a Customer belong to if he is using the PURL system if several

Clients send him PURL triggering "bookings" resulting in e-mails being sent to him? In an embodiment, the Customer becomes the Customer of the Client from whom he receives an e-mail which he opens and initiates a transaction.

The User Directory registers the different classes of users of the system and assigns/documents their access permissions. Users are attached to Clients and sales agents within this system, depending on if they are Customers or sales agents themselves. Note: Customers can belong to both sales agents and Clients – and sales agents can belong to Agencies, Customers and Clients.

In an embodiment, the "meta data" environment is now managed and controlled by a data modeler, such as the "CA ERwin Data Modeler" provided by Computer Associates. This means that the system can immediately generate, update and replicate database schemas to replicate points of functionality and create additional "Data Marts", and the system can even physically deploy to other database engines as well. ERwin, used as a "Meta Data" repository within the architecture, allows technical staff members to quickly understand how the system works at a database level, and it provides "Point of Truth" information on all database elements and attributes, provides important VARCHAR, key and foreign key knowledge to the developer, is a repository for all "Stored Procedure" code used within the system, and provides database management services and "change control" support to developers working with the system. What this all means from a development and support perspective is that the system now has complete centralized "Control" of the data and developers can learn what they need to from this repository to quickly make changes to systems functions where required and add new features to the system much more rapidly than they were able to before.

In the system, these databases represent for the most part "Landing Databases/Data Marts" resident on internal servers to initially receive third party content prior to this content being made available to Customers and other users of the platform. These databases do not attempt to change any information/data received from third parties. Rather, they "pull" this data from the third parties and then store it in the same definitions as provided from the third parties. "Data Transformation" then occurs as this

landed data is passed across to the “Staging Databases/Data Marts discussed in the Internal Database section below.

Internal databases (Staging Databases/Data Marts external to the core Architecture Database) are utilized throughout the system for a variety of purposes. These databases in most instances mirror the database schemas of various parts of the primary Architecture database. In relation to downloaded third party data, they are the receivers of “Transformed Data” to be used within the primary Architecture Database. Programmed interfaces between these and the Primary database load “Changed and Updated” data (including additions and deletions) from the last interfaced data transfer to the primary Database. These interfaces are tied to a “Scheduler” so these data transfers take place according to rules defined by the system – in some instances, they run during non-peak business hours – such as for third party data and in some instances, batches run every few minutes – such as when Booking Record details are loaded into the system after initial PURL notification e-mails are initiated for release. Note: Initial Booking Notification e-mails sent out to Customers bypass the primary Architecture Database entirely so as to ensure that their generation has no impact on the primary DB. These notifications are sent out from the “Booking Notification Staging Database” – a completely separate DB that in production is housed on a second clustered server separate from where the primary Architecture DB is resident – this to maximize performance of both databases.

Application Architecture

In an embodiment, the system is designed to be a sophisticated marketing engine whose core characteristics include: (i) Importation of Booking Records; (ii) Bulk E-mail and SMS/IM Marketing system capable of high-volume delivery; (iii) A web experience facility that is “Personalized” for each Customer; (iv) A sophisticated “Content Rendering” workflow system tied in to multiple third party content providers; (v) “Target Marketing” capabilities based on “Profiling” algorithms built into the system for each Customer; (vi) A subscription/membership signup system built for “Sales Agencies”, “Sales Agents” and repeat “Customers”; and (vii) A purchasing and shopping cart system usable globally.

Now referring to FIG. 3, shown is a macro level depiction of this system from a "Customer" facing perspective. As shown, delivery of content requires process, architecture, presentation and management services in addition to availability of the content. Here, the core Client Facing systems components of the system are primarily delivered by the Internet. To achieve architectural goals of fast rendering of content for a user, several architectural strategies have been employed by the inventors, including some of the following:

Web pages should load quickly, so they need to be built thin. That is, content rendering logic would not be contained within the web page as this slows down the page loading process. Rather, the strategy is to decouple this programmed logic so it is only called via a "web service" when specifically requested by a Customer or when an internal rule requires it.

As far as is possible, third party data will be stored and rendered from internal data servers as opposed to live third party interfaces – this to save web page loading and content rendering speeds – to improve the Customer experience and to as a result, increase sales conversions.

"Cache" storage of web pages to be rendered will be used as a strategy as opposed to "physically" stored web pages resident on web servers – again, to improve the web page rendering speeds.

Web page development strategies must employ the latest strategies in technical innovations. This strategy is important to improve greater "Openness" to wider ranges of user computer configurations, to improve rendering speeds, and also to improve longevity of the system code and designs.

Now referring to FIG. 4, in an embodiment, the system may utilize web development technologies such as Windows Communication Foundation (WCF), and others, to significantly reduce lines of code per screen. For example, in a possible scenario, the lines of code may be reduced to about a quarter of the lines of code of web pages using previous technology. The framework shown in FIG. 4 is physically instantiated into the design through some further architectural design strategies that have been employed.

In an embodiment, the system separates the “Design Skins” used for each “Client” and even “Agencies” from the web rendered content and associated programming logic so as to be able to setup individual “Skins”/CSS folders by business entity thus allowing ease of customization and reuse of “Skin” designs. Thus, instead of having to change each web page if a Client or Agency requests a design change, it is only necessary to change each instance of a “Skin – Type”, which greatly reduces the amount of work required to make changes to a website design.

In an embodiment, the “Web Server” farm (see physical topology documentation) will manage the skins and the rendering of the PURLs and their related content to “Cache” memory.

In an embodiment, throughout the deployed solution, as users of it traverse from screen to screen, “selection” and “user “profile” parameters are passed along to waiting “Content Rendering” programs that accept these parameters and then use them to calculate and render the most appropriate content to that end user, interacting with the databases as required to “get, sort and render” the desired content. Enterprise Data Environment

This section discusses some of the key architectural design strategies and capabilities of the new “Enterprise Data Environment” (EDE).

The key to the target data architecture is the definition and implementation of an integrated enterprise wide database. It provides a common data architecture for all business functions reducing data redundancy, promoting shared data, and eliminating inconsistencies. This common data architecture with the appropriate data management technologies facilitates the integration of data from multiple sources, guarantees consistency of data, maintains end-user transparency, supports a flexible transaction framework, is extensible and dynamic, and supports standards-based APIs.

Now referring to FIG. 5, shown is a schematic block diagram of a Customer PURL skin architecture in accordance with an embodiment. In an aspect, a key to the target data architecture is the definition and implementation of an integrated enterprise wide database. It will provide a common data architecture for all business functions reducing

data redundancy, promoting shared data, and eliminating inconsistencies. This common data architecture with the appropriate data management technologies facilitates the integration of data from multiple sources, guarantees consistency of data, maintains end-user transparency, supports a flexible transaction framework, is extensible and dynamic, and supports standards-based APIs.

In an embodiment, program logic in the system has been decoupled as far as possible from core system driving business objects/attributes. Now instead of attributes being instantiated in code, the system calls key attribute values from database tables instead – allowing for much greater flexibility and extensibility in operational implementation. This has allowed the inventors to construct very sophisticated, parameter driven “Rules Engines” within the design of the new system. Shown in example below are just a few of the key attributes to this new system that are now supporting the new “Parameter Driven Design”.

Parameter Attributes	Business Value Examples
Third Party Type	Hotels, Cars, Limos, Shows, Activities, etc.
Order of Preference	1 st 3 rd Party Choice by Type, 2 nd 3 rd Party Choice, etc.
Currency	Now a Multi-Currency System
Clients	Incremental Numerical Parameter in the System
Insurance Providers	RBC, Manulife, etc.
Agency Membership Types	Iata, etc.
Product Types	Travel, Brochures, Guidebooks, etc.
Customer Groups	Define marketing initiatives against defined Customer Groups – define rules for the groups
Content Types	Define content types and render based on defined rules

Culture	Multi-Language Capable – English, French, Japanese, etc.
Rule Request Number	Repository of Client, Agency, etc. Driven System Processing Rules
PURL Template Number	Manages which PURL Screen is Rendered
Additional Information Types	Parameter Driven – Select the Content You Want to a PURL by the Rule You Want for This

In an embodiment, the entire EDE environment setup for the system is geared towards managing the delivery of content in relation to Client, Agent and Customer generated “Events”. Clients “push” booking record “events” to the system – which as a result of the event, sends out related marketing messages to Customers. Customers trigger “PURL Generation” events by responding to messages sent to them. Once initial PURL’s are requested by a Customer, they then request additional “Content” which the servers and

systems provide to them in such a manner that the Customer is enabled and hopefully interested to generate a product or service Transaction” which the system then manages for them, supported by the EDE.

Key features/capabilities constructed for the EDE Architecture include the following:

- Simplified Business Setup – The new parameter driven system allows for much easier new “Client” setup and configuration or reconfiguration.
- The EDE “Data Model” is clear and intuitive: Business and data relationships are easily understood.
- The databases are modularized – designed by function point so the system can be easily modified and new or additional functionality to the system can be easily implemented.
- The environment can be easily replicated for multiple instances to maintain high performance and availability as volumes of concurrent usage and transactions grow. Business requirements at that time will define the amount and levels of replication required.
- ERwin, used to design and deploy this system into production is now a fully documented “Meta Data” environment for the system even down to and including the “Stored Procedures” used within it.
- Points of Truth for data attributes are easily found – helping to ensure data accuracy and integrity through the system.
- “Web” and other developers building interfaces to or accessing the database to insert or retrieve data from it, interact only with the published stored procedures built to manage the system. They are shielded by these programs from having to learn and write programming logic to apply business rules. The stored procedures in place “shield” them from this requirement – making their jobs easier and as an added benefit, this helps to preserve the “Referential Integrity” of the database itself.

- Data is stored in its most utilitarian forms – it is in effect modeled to support the business requirements that need it. Clear naming standards and table designs further support the system's ease and speed of use.

Enterprise Computing Environment – Infrastructure

The scope of the Enterprise Computing Environment (ECE) is to provide a suite of common services and consistent access to those services across all components of the enterprise. The intent of the ECE is to make these enterprise services available to authorized users and processes within the context of a single systems image. The ECE environment provides these enterprise available services through a series of “technology layers” and is comprised of-

- Server environments which provide the platforms for enterprise and data services;
- Workplace environments that allow users (staff, Customers, Clients, Agents, etc.), developers and applications to access enterprise services;
- The “Network” which provides connections to all enterprise services and servers to all users and applications both locally and remotely;
- The Network and Systems Management (NSM) environment which ensures consistency, manageability, and availability across all system components for all user groups and applications.

A multi-tiered architecture model (both physical and logical) has been designed to provide location transparency of data and processing resources. This approach allows for differing degrees of centralization and decentralization of processing and information based upon efficiency and effectiveness considerations. This framework utilizing standard API's provides benefits from advances in technology, rapidly adapt to a changing enterprise and evolving operations concept, and support a variety of deployed configurations. A common enterprise information model will support different but logically integrated views for each level of operations: Global, Regional and Local. Support for higher level views is provided through replication, summarization, or

transformation of certain data, while other data is accessible transparently from source levels as necessary or upon demand.

Enterprise Level E-Mail and Message Delivery Services

In an embodiment, the system is adapted to send out e-mail messages to prospective Customers, and monitor the success of messages sent while simultaneously collating user usage statistics.

Reliably sending a high number of e-mails monthly requires implementation of an effective Sender Policy Framework (SPF) to reduce the possibility of e-mail being blocked as unwanted e-mail. This framework has operationally been instantiated and configured in our third party purchased and deployed message management system.

Third Party Data Topology

The core business and technical strategy of downloading third party data so that it can be served up to Customers from internal servers as far as is possible brings with it a number of technical challenges to be overcome. The nature of some third party content requires it to be downloaded daily and even hourly while other content can be downloaded monthly or at even longer periods between downloads. Also, the "Maturity" of the third party content providers vary – some are sophisticated and content can be downloaded easily and it is possible to easily obtain changes, updates and deletions from them through "data refreshment" interfaces that have been constructed. Other providers require constant downloading of subsets of their full data in background as they do not have the ability or policies in place that allows the system to easily download what is needed. Some cannot even get the data to the system except through individual information requests through an online interface – this last group causes the greatest challenges – as the Customers are then at the mercy of the third party's "Speed of rendering" capabilities in an online session.

The system therefore implements individual interfacing and downloading strategies with each third party provider. Downloading data can be time consuming – and once it has been downloaded, it needs to be "Transformed" before it can be served up to

Customers from within the system. Core strategies employed to get this data to where it is usable from within the system include the following mechanisms:

- A separate server from the production server(s) is employed to initially capture third party data. Data is initially downloaded in “Native” formats as defined by the third parties so as to minimize the downloading times from the third parties. Each third party’s data is stored initially in a “Landing Database” designed specifically for their data.
- A second set of tables in the same instance of this “Landing Database” have been setup that contains the same table and attribute definitions as the production system for this data – this is called the “Staging Database”.
- The Landing Database is “Mapped” to the new data definitions required for use in the production system and data transformation rules are instantiated into event triggered stored procedures setup in the Landing data base so that new or changed/updated records that appear in the Landing Database are automatically transformed and loaded across to the Staging Database.
- Scheduled programs/routines run against the Staging Database to load changed (including notified deletions) or new records only across to the production database in “Batch” uploads. The timing of each of these upload interface programs run using a system “Scheduler” according to business rules/requirements that have been defined for each third party.

Booking Notifications and Initial Marketing E-mails

Just as for above third party content providers, “Booking Notifications” often need to be “Transformed” before they can be processed through the system. Core strategies employed to get this data to where it is usable from within the system include the following mechanisms:

- A separate server from the production server(s) is employed to initially capture native booking record data so as to minimize the downloading times from Clients.

Each Client's data is stored initially in a "Landing Database" designed specifically for their data.

- A second set of tables in the same instance of this "Landing Database" have been setup that contains the same table and attribute definitions as the production system for this data – this is called the "Staging Database".
- The Landing Database is "Mapped" to the new data definitions required for use in the production system and data transformation rules are instantiated into event triggered stored procedures setup in the Landing data base so that new or changed/updated records that appear in the Landing Database are automatically transformed and loaded across to the Staging Database.
- Scheduled programs/routines run against the Staging Database to load changed (including notified deletions) or new records only across to the production database in "Batch" uploads. The timing of each of these upload interface programs run using a system "Scheduler" according to business rules that have been defined for Booking Records.
- Also from this Staging Database, the "Booking Marketing" E-mail may be initiated that goes out to Agents and Customers letting them know that a Customer PURL site has been created. Templates for these e-mails are constructed and stored in our Lyris message management system whereas the business rules relating to the use of these templates and mail outs are stored within the Staging Database as stored procedures. Please be aware that this strategy has been designed so as to minimize performance impacts against the primary Architecture Database – which processes and manages all Customer PURL requests and purchases. The system may send out 1 million PURL e-mails from this Staging database and by utilizing this strategy, all these PURL's do not have to be individually loaded to the primary Architecture Database – which saves considerable I/O against that DB. By introducing a slight latency between when the e-mails go out and when "batches" of records are uploaded to the Primary, the performance impacts on the Primary can be minimized.

Now referring to FIG. 6, shown is an illustrative schematic flow diagram of an architectural system flow for PURL generation.

Stage 0 – Importing Booking Records to the Staging Database

Currently, there are three methods envisioned for obtaining initial booking information from Clients (airlines or airline aggregators):

- Method one is to first “parse” either the body of an e-mail or the subject line of an e-mail and then go back to these Clients to pull down a complete booking record (see Booking Record Mapping spreadsheet for a full booking record definition and how it maps to the Architecture Database system)
- The second method is to receive a booking notification via an XML message and then go back to their system to get the full booking record.
- The third is to have a full booking record sent via the new Web Service/Booking Record Receipt API that asks the “Clients” to use when sending booking records to us. This will be the preferred method of receiving booking records moving forward.

An important design consideration for this interface is to ensure that as this procedure inserts booking records received from a Client into the Staging database that the related Client_No attribute is attached to it as an added attribute to each booking record inserted. This is needed as it is used as part of the Booking E-mail Notification process later in the Architecture DB.

This means that for each booking record import process created, part of the setup procedures for new Clients include defining this parameter for each Client setup.

Stage 1 – Proc02 – Import Booking Records to Architecture DB

The PURL Generation process is triggered when Proc02 inserts a booking record or a batch of booking records (it can do both) into the Architecture Database. The mapping of data from the Staging Database to what is inserted into Arch DB should be the same

as all data transformation takes place in the prior processes of loading data to the Staging Database from Client data.

Still referring to FIG. 6, shown is a schematic system flow diagram of a PURL generation system flow. As shown, Booking Records are imported to the Architecture Database and remain there ready for use until such time as a Customer clicks on a PURL link received via e-mail through the Create Booking Marketing E-mail process. There are two types of content rendering activities that take place once this link is triggered:

- Initial PURL Generation: This is the first screen that the Customer sees. From this first screen view, he/she then makes choices that lead to subsequent "Content Renderings".
- Subsequent PURL Generation: This is a Customer triggered event where a Customer selects a secondary screen from which he/she can make purchase decisions or get more information. Some screen selections lead to additional direct viewable content (text and images directly rendered from the Architectural Data Base. Some screens pass the Customer directly to a third party website, passing certain parameters with the query to the new site so that financial interests are protected and so that searches can be completed using the "Data parameters" passed; and finally, some events bring the Customer to a "Search Screen" wherein they can enter search parameters so a search can be completed relating to third party data stores managed by Vendor internally.

This design specification documents in overview how the system will manage the retrieval of all third party content to the PURL screens.

Initial PURL Rendering

There are Clients using the Vendor Marketing/Architecture System that have unique business requirements that need to be managed. There are two such requirements that have impact at the very front end of rendering the initial PURL. These are:

Certain Clients desire a refresh of the booking record prior to rendering the first PURL. To accommodate this requirement, on their clicking the link, the system needs to create a web-server based application that checks to see if this is the first time this PURL is rendered and if it is, the related "Web Service" will go back to the Client system to download a replacement booking record and insert it into the Architecture database prior to rendering the first PURL screen.

The second requirement is that, on rendering the first PURL, some Clients desire that hotels and/or car rentals, etc. be rendered on the first PURL even though the Customer may have already purchased these from the Client prior to the Customer using the system – and other Clients do not wish to render these items in the first PURL in such a situation.

Booking Record Refreshment Program

When a PURL link is clicked, the Customer is brought to the Web Server which manages all PURL rendering. Here there is resident a WCF web service that calls the Architecture DB to see if there is an existing record in the PURL_Activity_Log for this Customer. If there is a record in the log that contains a PID value that is the same as the PID value for the related call/event, then it is possible to determine that this is not the first time through the system for this Customer. If so, the Record Refreshment program does not need to be accessed to refresh the booking record - the Customer can instead be taken directly to Proc04 to render his/her pages according to the appropriate business rules and selection/rendering parameters.

If there is no record in the PURL_Activity_Log for this PID number, then the program checks the PURL_Parameter field in the Primary_Client_PURL_Parameters table to see if this attribute has an attribute value of "PURL_Refresh". If it does, then the program knows that the program needs to access the Client booking system to "Refresh" the booking record details for this PID/event prior to rendering the first PURL.

In the table below, the XX represents a numeric value that denotes the time "Delta" between when the booking record was first created in the system and when the system will require booking record refreshment based on when the Customer is accessing the

initial PURL via the link he/she received in the Booking Notification/Marketing E-mail. Only one of these records will be resident in the table below – on setup the record that best fits a particular Client’s defined rule will be implemented. For example, if the requirement is that if the Customer is accessing the initial PURL via the link within 5 minutes of initial record creation then no refreshment is necessary, then the record in this table will read Min05. If the requirement is no refreshment within an hour, it would read Hour01. If the requirement is no refreshment within an hour and a half, then the record would read Min90 – etc.

Core Table Attribute Found In: Primary_Client_PURL_Parameters
 Attribute: Parameter_Value

Attribute Value	Attribute Business Name
MinXX	Minutes where XX is a 2 digit numeric
HourXX	Hours where XX is a 2 digit numeric
DayXX	Days where XX is a 2 digit numeric
WeekXX	Weeks where XX is a 2 digit numeric

If it is found after going through the above system checks that refreshment needs to occur, then the system will follow the following system flow.

Booking Record Refreshment on “Initial PURL Rendering” - System Flow Steps:

- Get parameters required to retrieve Booking Record: PNR, Customer Name (First and Last Name) and Agent Name.
- Call Client interface “sub-routine” to retrieve Booking Record and retrieve record to cache/session (One of these sub-routines will need to be constructed for each Client that wishes Booking record refreshment feature to be enabled in the system).
- Read Landing DB record (Version Number/Last Date Changed field available?) If no change to original booking record has occurred, run Proc04 and continue to render PURL.

- If Booking Record has changed, update Landing DB and trigger Landing DB to Staging DB to Arch DB program (to be defined and written) and then finish rendering PURL for Customer via Proc04 – making sure that Proc04 grabs new record information instead of retaining old booking information. Note: This has to be a parallel program to the mainstream Booking Record capture programs. This all has to work at one time within one program instead of the 3 programs written for the mainstream system process as it has to happen quickly – to enable as fast as possible initial PURL rendering.

PURL Rendering Parameter Capability – in Proc04

In an illustrative example, the PURL Rendering “Parameter” capability is included as part of Proc04 illustrated in FIG. 7. As mentioned above, on rendering the first PURL, some Clients desire that hotels and/or car rentals, etc. be rendered on the first PURL even though the Customer may have already purchased these from the Client prior to the Customer using the system – and other Clients do not wish to render these items in the first PURL in such a situation.

The way this is handled in the system is as follows. During Proc04, on being passed the initial parameter “Client_No”, and the system checks for that Client in the *Primary_Client_PURL_Parameters* table for attribute “PURL_Parameter” and value “Always_Show”. If this value is found, then Proc 04 includes a rendering of hotels and car rentals when rendering the initial PURL even though a hotel and/or car rental has already been booked by the Customer through the Booking Record. If this “Always_Show” value does not appear here for this Client, then it does not render a selection of hotels/and or car rentals on the initial PURL if they have already been purchased through the booking record.

Proc04 – Get Client Configuration & Get third Party Rules

Used for all PURL third party data generation requests and third party queries. First step in process for a PURL third party content generation request or callout for third party information to retrieve ancillary purchase options - cars, hotels, events, etc.

As per the illustration above, if this is an “initial PURL request (triggered by a Customer clicking of a link containing the GUID information), then the procedure first goes back to the booking record loaded into the Architecture Data Base to determine if either a hotel or car rental or both have already been purchased by the Customer. If both have already been purchased, then the procedure sends the Customer to Proc05 – Request Other Content. If either a hotel or car rental or both are still to be purchased, then Proc04 passes this callout request on to Proc06 – Request Hotel or Car Content after first getting all required Client Configuration and Preferences parameters so they can be utilized in Proc06 to search for, select, sort and render content to the web page.

Now referring to FIG. 8, shown is a schematic system flow diagram of a booking e-mail notification design flow.

Stage 0 – Importing Booking Records to the Staging Database

Currently, there are three methods for obtaining initial booking information from Clients (airlines or airline aggregators):

(i) Method one is to first “parse” either the body of an e-mail (current method employed with Travelport), or the subject line of an e-mail and then go back to these Clients to pull down a complete booking record.

(ii) The second method is to receive a booking notification via an XML message and then go back to their system to get the full booking record.(iii) The third method is to have a full booking record sent via the internally constructed Web Service/Booking Record Receipt API. This is our preferred method of receiving complete booking records from our clients. Important design principles contained within this interface include the following:

- There must be a Client identifier within the Staging Database attached to each record. In an embodiment, Client_No attributes are set up in the Arch. DB as the system sets up new Clients and that as each new Client is added to use this system, the program that inserts new booking records into the Staging Database must be updated/kept in alignment with each new Client set up in the Arch DB

system. For example, if a new Client is entered as Client_No "6", then all booking records received from that Client and inserted into the Staging DB must be assigned to Client_No "6" in order for the system to work properly.

- As each new record is entered into the Staging DB from a "Client" interface, it is assigned a unique "PID" number which is also entered. It also receives a "GUID (Global Unique Identifier) number that is also inserted with the record. The GUID is a system generated 36 character random multi-character type data attribute.

Stage 1 – Proc03 – Create Booking Marketing E-Mail

This is a process that is managed by the server's "Scheduler" system and runs very frequently (the frequency may be set to run every 1-3 minutes to allow prior batches sent through to complete), picking up new batches of bookings for which e-mails are to be sent out, grabbing them from the "Staging Database".

The Staging DB contains database triggered programming logic contained within it that generates e-mail requests which are sent to the message management system to be merged with e-mail templates resident there for the purpose of sending out the appropriate PURL link e-mail message to the Customer. The Client_Nodata attribute registered on the insertion of the booking record into the staging database defines which template is to be used for the e-mail being sent out to the Customer. When the Client is known, it then checks the "Select_To_Do settings for both PURL_Event_No attributes 1 and 2 to see if e-mails are to be sent out to both the agent and the Customer or to the Agent only – so they can forward it to the Customer later using the TAD feature for this purpose.

The template is first grabbed by the procedure, and then is populated with the "personalized" content contained in the booking record and then sent it off to the messaging system to generate the required e-mail. Note, to determine whether or not the e-mail is to be sent to just the agent or to the agent and the Customer by the system, a "Client Profile" table in the Staging Database is kept and maintained.

Information that is needed to create a booking marketing e-mail from the Staging DB are listed below.

Staging DB

Table	Attribute	Description
PrimaryClient	Client_No (Attribute=2 for Travelport)	Relates to Client generating the booking record
Client_PURL_Options	PURL_Event_No (Attribute=1 for Booking Marketing E-Mail to Agent)	Attribute number defines which e-mail is going out.
Client_PURL_Options	PURL_Event_No (Attribute=2 for Booking Marketing E-Mail to Customer)	Attribute number defines which e-mail is going out.
Client_PURL_Options	Select_To_Do = Y	(Attribute =Y in relation to above 3 attributes means an e-mail is to be sent out)
Users	First_Name	Customer first name – “token” to be used to insert customer’s first name into the salutation line in the e-mail (see attachment “A”
Flight_Base_Information	Destination_Name	Destination City - “token” to be used to insert customer’s first name into the salutation line in the e-mail (see attachment “A”
PrimaryClient Additional Info Table??	PURL_WebPage1 ??(lan to Name)	So that webpage can be displayed on e-mail for cut and paste into browser.
PURLForCustomer	PURL_Unique_ID	So that webpage can be displayed on e-mail for cut and paste into browser.
PURLForCustomer	PID	So that webpage can be displayed on e-mail for cut and paste into browser.
??? Where to store access to this?	Weather Object + Destination Parameter	Cello working on this part – connect with him for details.
Users	e-Mail_Address	Customer E-mail address
Users	Agent_Email_Address	Agent E-mail address

A stored procedure in the Architecture DB passes information across to the Staging DB for instances when Client information is updated within that system. This procedure captures and inserts to the Staging DB the following data attributes relating to a Client when new records are created or values are updated.

Architecture DB

Table	Attribute	Description
PrimaryClient	Client_No	A new client is added – also insert record to Staging DB
Client_PURL_Options	Select_To_Do	If a Y changes to a N or vice versa, this change needs to be also inserted into the Staging DB plus if a new record for a new client is created, this needs to be in Staging DB as well.
PrimaryClient Additional Info Table	PURL_WebPage1	If this updates or a new record is inserted, it also needs to be inserted in Staging DB

Stage 2 – Proc02 – Import Booking Records to Architecture DB

To ensure that a PURL can be created, the booking record plus related PID plus related GUID are also passed to the Architecture DB through another procedure.

The mapping of data from the Staging Database to what is inserted into Arch DB is the same as all data transformation takes place in the prior processes of loading data to the Staging Database from Client data stores as per booking record designs.

Important design features for this interface include:

- This is a timed server resident procedure that reads the Staging Database for new records and then inserts them into the Architecture Database. Timing for the procedure to be run is set externally by the server's "Scheduling" system.
- The interface is designed to create "batches" of new records and it imports all new records at one time/by batch to the Architecture Database. So an indicator of last "Date/Timestamp" is kept in a log and records captured there ensure subsequent running(s) of the program only capture new records inserted into the Staging Database.
- As each record is inserted into the Architecture DB, the PID and GUID (PURL_Unique_ID) also are passed across.
- We also bring along the Client_No attribute and insert this value into the PURLForCustomer table as received from the Staging DB.
- This procedure is written in such a way that it can be set by a programmer to run against a single, defined Client or for several at a time, or for all Clients at one time.

In an embodiment, the present system and method delivers a personalized, consumer targeted website containing real time and near-real time gathered, sorted and rendered content delivered from both internal databases and external third party content suppliers.

In order to achieve these objectives, the inventors have implemented a number of features. For example:

- The presentation layer of the system, built on the newest of technologies, programming languages and databases, is dynamically rendered to the user in

“Cache” Memory to speed up page rendering to the consumer. In addition, each page rendered is custom “skinned” to reflect the business entity or entities that offer the services. For example, in the Travel Industry, skins reflect the airline and/or the agency relating to the initial consumer purchase.

- Programming logic between the web page constructs and the database are written with a view to maximizing the separation between Abstraction Layers so that the system can easily be reconfigured as usage volumes increase, ensuring a highly scalable user interface environment.
- The database layer of the architecture is very complex and utilizes a series of “Landing” and “Staging” databases that continually receive, process and transform “disparate” content data from external content providers so that this data can be imported to the core processing system to be rendered to consumers as freshly and as accurately as it is possible to render it. To add to this sophistication, depending on the business rules required by the user triggered business process, internally derived data is often coupled with real-time imported and delivered external data (within a user session) and delivered to the consumer rapidly enough so that the user experience is not appreciably compromised.
- The infrastructure providing these end user services makes use of a number of deployment strategies geared also to supporting the maximization of rendering speeds to the consumer, and its deployment design ensures that user entered data is kept safe and that systems are quickly recovered and again usable to a consumer should some form of system failure occur.

Data Rendering

FIG. 9 shows a schematic flow diagram that illustrates a content data rendering system and method in accordance with an embodiment.

Content when rendered to the consumer needs to be accurate and timely. As an illustrative example, for a Travel Industry Client, showing a hotel with availability at a certain price in real-time involves a number of steps.

If obtaining the same hotel content for a user and searching multiple third party hotel providers simultaneously to find the user the best deal, the system and method needs to match this disparately received data from different source points prior to rendering to the web page, otherwise, the same hotel might appear several times on the same screen for the user. To avoid duplication, the system and method utilizes a filtering module to match data from disparate source points to ensure only one purchase option for a particular hotel is rendered to the user viewable web page. The system and method thus gathers, sorts and selects data prior to rendering including. For example, the system and method executes one or more of (i) name splitting to improve searches for content (various content attributes searched), (ii) parsing algorithms strategies, and (iii) absolute and partial match calculation routines including Soundex matching for indexing names by phonetic sound.

In an embodiment, it is desirable to filter data to send the most appropriate and desirable content to the targeted user – content that would show him/her purchase options that would interest them and possibly motivate them to make a purchase. To do this, the system and method utilizes an internal Learning Engine that utilizes context to determine what type of data is most appropriate to render. For example, if a tickets purchase indicates that a traveler is travelling with a family, the system and method renders family oriented activities rather than adult oriented hotels, shows, activities, etc. If the traveler purchased a first class airfare, then the system renders 4 and 5 star hotels rather than 2 and 3 star hotels, etc. Thus, in an embodiment, the Learning Engine uses one or more of “Weighted Averaging”, Aggregate Match Scoring, Lookup Matching and other “Fuzzy logic” approaches to filter and then render appropriate content to the targeted user.

Now referring to FIG. 10, shown is a schematic flow diagram that illustrates a Content Data Normalizer and a Learning Engine in accordance with an embodiment.

Preferably, the system and method should obtain and process third party data to web pages fast enough such that end user Customers are not frustrated waiting for the web page to appear with the appropriate, targeted content inserted. Interfacing to this data directly from code written in web pages, as is done on most websites generally does not work as the complexities introduced to the rendering rules and filtering logic cause speeds to slow.

In an embodiment, to achieve quick rendering of timely, accurate and filtered data, the present system and method renders web pages to many concurrent users by rendering content from internal databases to cached web pages, rather than physically storing web server resident web pages.

In another embodiment, the system and method utilizes internal systems logic (Content Data Normalizer, Learning Engine, Rules Engine, etc.), then stages, sorts and matches data at different points in the rendering process and at various levels of data abstraction to render “timely” selected data quickly.

Some data may be pre-staged into new caches, inheriting the related business rules of the obtained data without corrupting (i.e. forgetting/changing) any element of the original rules. The system then flattens the cache and work with only one set of combined rules to deal with at only one Abstraction Layer level – this all done to help minimize negative performance impacts.

Showtime Tickets Example

Below is a third party content provider example that illustrates the data rendering described above.

Showtime tickets require a composite match-up of pre-staged and rendered content to content obtained during the user session through a real-time interface. Some of this real-time data is downloaded and cached within the session prior to a user requesting it, and some of the real-time data is rendered as a result of a query or transactional process initiated by the end user. For example, the following data access layers may be involved in a system event:

Data Access Layer (DAL) 1: Nightly download of events, locations, times, standard pricing, etc.

Data Access Layer (DAL) 2: Real time query to Show Time within the user session to get and book available seats and up to the minute pricing involves several sequential interactive queries to Showtime Tickets:

- Ticket Category
- Ticket City
- Ticket Type
- Content Type (eg., Music Type)
- Event Listing (eg., Artist)
- Date & Time
- Seat Pricing, Availability & Seat Options
- Select, Book & Pay

Pre-caching and Flattening Objects

With multi-layer mixed source content acquired from multiple architectural levels, conventional rendering of web pages takes too long. Therefore, the present system and method significantly flattens the process by retrieving and pre-caching beforehand a significant amount of data in the hopes of minimizing the communications between Abstraction Layers on user request. By having a significant amount of data flattened, the rendering times may be greatly reduced.

As an illustrative example, assume that from a Booking Record, the system can learn things about the traveller, including Airline, destination, age, family, etc. Using this information, the system can query a local content database and the third party content database to get content in advance of the end user Customer accessing the site. Thus, pre-caching the data and having it ready to transmit may greatly increase the speed with which a Customer query may be rendered when he/she logs on or enters a query at the site.

Now referring to FIG. 11, shown is a schematic flow diagram that illustrates how third party content is gathered and rendered to a Customer PURL with an embodiment.

When a TAD (Travel Agent Desktop) or Customer PURL screen is requested by an agent or Customer, a "Skin" for that screen is pulled from the system (i.e. web server resident webservice) and the data is pulled from internal and external data stores as defined by related business and systems processes. When a request is made for a screen, the system knows the following, as certain parameters have been passed along with the screen request: (i) The Client Profile (in the case of a PURL generation request); (ii) The Customer and his/her/their trip details (in the case of a PURL generation request); (iii) The booking details define which third party "Types" are to be included and the Client Profile tells us which third parties to use along with their "Priority" settings as related to that Client; and the agent involved when a B2B event exists. These parameters define what third party data will be selected and rendered to the screen within the PURL generation procedure.

In an embodiment, the definitions/rules within the system and method further include how many records of each third party type are to be rendered to the screen, and preference rules relating to third party data selection. In an illustrative embodiment, the process flow for this event may be as follows:

- Receive screen request event from Agent or Customer – including initial knowledge parameters (see above)
- Is Client involved? Get relevant Client profile/settings information
- Using Client information (if Client involved) – otherwise use default parameters, select third parties and data according to rules and within the required date ranges where applicable and retrieve data
- Sort and parse data according to Client rules and preference settings
- Render data to screen/Skin
- Present screen to requester from "Cache Memory"

After a screen has been rendered, the Agent or Customer when choosing to purchase an item or service clicks on the “Add to Cart” button to initiate a purchase request. The following details are then passed to the Shopping Cart to manage/complete the purchase process for the request:

- The Customer and related profile information (DB search is done to see if Customer Profile exists)
- Details on the item to be purchased
- When the Customer subsequently presses the “Go to Cart” button, he/she is taken to a screen where what is known about him/her is rendered to the first Shopping Cart Screen.

Upon selection by the Customer of an item to purchase, an internal shopping cart procedure then manages the remaining purchasing steps.

Third Party Data

In another aspect, the system and method may be configured to render initial queries for third party data from within the system where possible. This requires that third party data will be loaded and refreshed to the system database regularly – preferably during non-peak hours. In an embodiment, a parameter driven server resident program manages all data retrievals for third party data – except for when a shopping cart purchase is being made.

“Planning PURL” Feature

In an embodiment, the system allows a travel agent to manually request and create for a currently non-travelling Customer, a “Planning PURL” – which is to say a created PURL site for the Customer to visit without the Customer having to book a flight first. This PURL will appear on the Agent’s screen and he/she can then forward the access link for the PURL to the Customer via e-mail.

On receipt of this e-mail, the Customer will be able to click on a “linked URL” button that will take the Customer to his or her created PURL website. Once there, the Customer

may choose to use a “Forward to a Friend” feature that sends an invitation to a friend to come and view the Customer’s PURL site. In this way, Customers may review a possible travel excursion before purchasing, or multiple Customers can review and plan a travel excursion together based on a PURL created for one Customer.

Use Case – Travel Booking and Itinerary

Now referring to FIG. 12, shown is a schematic flow chart of a Travel Industry use case in accordance with an embodiment. The platform is a complete “end to end” enhanced itinerary system that provides travel agencies with a suite of next generation communication and 1:1 marketing tools to enhance their traveler experience and drive incremental revenues. The platform dynamically creates a highly personalized website (PURL) as previously described, based on any number of available variables from their profile, itinerary or destination. Additionally, the platform provides the ability to access/edit their Universal Profile and manages the traveler’s itinerary before, during and after travel. At each stage of the journey, the platform customizes and personalizes each traveler website including the design, branding, messaging, graphics, e-commerce vendors, content, destination information and merchandising.

Planning Travel – The platform provides travel agencies with the industry’s first interactive and digital planning sales tool. The purpose of the Planning PURL is to enable travel agents to professionally present travel options and improve their close ratios. The platform offers the ability to present itinerary options, selected through Client business events, to their Customers via a personal website. Each “Planning PURL” will couple the detailed itinerary options with customized destination content (e.g. destination information, excursions, merchandise etc.) to build excitement and engagement. In addition, the agency can select a call to action or the ability to book directly from the platform under their credentials.

Booked Travel – Once a travel booking has been confirmed and ticketed by the agency, the traveler will receive a completed enhanced itinerary. At this stage in the journey, the PURL acts as the traveler’s final travel documentation and includes the Customer’s comprehensive itinerary, accounting, e-ticket and expense data. All other content is

now customized to this itinerary/ destination and the e-commerce and merchandising vendors are triggered and embedded in the PURL. Several other features are also activated including a “Communication Corner” to enable direct notifications of schedule change, alerts or other travel information and the mobile version is available for access via any smartphone. The mobile application enables in destination purchases and communication via the same platform.

After Travel – The platform automatically manages the “after travel” experience with a “Welcome Home” site timed for guest return. This provides the travel agent an enhanced Customer service opportunity to survey the guest, enrol them in a loyalty program and most importantly to generate the next Customer inquiry/booking. Again all content, advertising and messaging is customized to this phase of the trip and offers the Customer the opportunity to upload photos and share with friends and family via their social media network.

Dashboard -Each enhanced itinerary (PURL) is both emailed to the travel agent and/or traveler and deposited to the travel agent’s wallet within the TAD dashboard. This dashboard can be embedded into the Client platform and provides the agent with the full status of itineraries, send final PURLs, track Customer open rates and view commission statements. The dashboard also provides administrative access to travel agency owners for the management of any configurable components of the Customer email and itinerary platform as determined by the Agency or Client.

The 1:1 marketing capabilities of the platform offer travel agencies and their Customers one or more of the following advantages:

- More emotive presentation user interface
- A new degree of contextual personalization and customization
- A platform to manage the full journey with products and services from before, during and after travel
- New incremental revenue streams from e-commerce and merchandising
- Ability for the traveler to access and edit their Universal Profile
- A leading advertising platform to highly target and maximize ROI

- Single itinerary platform for all Client core provided services

The platform provides airlines with a suite of next generation itinerary, merchandising and 1:1 marketing tools to enhance the Customer experience and drive incremental ancillary revenues throughout the travel journey.

The itinerary is a sophisticated and dynamic web platform providing Airlines traveller's with a customized website for each trip.

An itinerary can be initiated by passing information electronically via API or any other system.

Each itinerary provides Airlines the ability to personalize and customize the design, messaging, graphics, content, offers and information by the business rules chosen Ability to dynamically customize each site based on Enrich Frequent Flyer Status or any other Customer segment variable Airlines can self-configure and control the branding, user interface and design elements through simple configurability tool available in the TAD dashboard The wallet acts as a single repository for all Customer bookings with Airlines and can feed valuable information to the Client's CRM platform

In an embodiment, an itinerary never expires and can be updated several times throughout the journey...before, during and after travel Full destination information via Lonely Planet or Guimatic including videos Calendar integration with Outlook, i-cal etc. Flight status and tracking Enhanced hotel images, text and video The Communication Corner allows Airlines to stay in constant communication with the traveller for schedule changes, alerts, notifications etc. 60% of those canvassed define the user experience as very appealing with information easy to find and purchase

The platform provides an enhanced tool to drive new incremental revenues... Airline seats, bags, meals, mileage etc. Hotels, Cars, Insurance, Rail, excursions, events, ski, golf, limo etc. and guides Dynamic messaging can be automatically delivered at pre-defined intervals to drive purchasing behaviour: 48 hours prior upgrade to business class Hotel / car booking reminders In destination activates and excursions Check in 24 hours prior to departure.

Customer Retention

In an embodiment, a welcome home site is automatically generated and timed for guest return. Furthermore, an automated email from travel consultant may provide the opportunity for feedback via a Customer survey and promote the next purchase with repeat guest promotions.

Mobile Platform

The itinerary is mobile enabled for the following: Blackberry, i-Phone, Android Tablet Bookable Mobile Content Insurance Hotels Car, Limos, Transfers Shows, Sporting Events, Concerts Excursions Ability to drive pre-trip and in-destination messaging: Upgrade to business class prior to departure Limousine transfers Destination excursions Traveler services Check in online Flight Status Destination content and weather Share a PURL with friends and family via any social media site.

Communications

The communication platform will automatically deliver a number of rules based consumer communications...

- Email with the itinerary to the Customer, can be co-branded if booked by a travel agency.
- Reminder emails to open the Itinerary after specified number of days
- Timed emails (7, 14, 21 days prior to departure) promoting the purchase of intelligent merchandising including insurance, hotel, car, limo, excursions, shows, concerts etc.
- Pre-trip email sent 27 hours prior to a flight offering check in services, upgrade to business class, merchandising, limo or transfer to the airport Mobile SMS and Email messaging once arrived in destination to promote local limos, excursions, shows, events, concerts etc.

- Pre-trip email sent 27 hours prior to a flight offering check in services, upgrade to business class, merchandising, limo or transfer to the airport
- Welcome Home communication including personalized message, survey, frequent flyer enrolment, promotions for next Airline ticket etc.
- Home city promotions advising Customers of shows, theatre, concerts and sporting events in their home city and purchasable via their travel agent.

Advertising

In an embodiment, the advertising platform is adapted to significantly improve targeted advertising in order to increase revenue generation. An advertising capability is built into the system and advertising space on web pages is being sold to approved advertisers.

Reporting and Metrics

The system generates a significant number of statistical reports, metrics and key performance indicators including but not limited to the following:

General throughput statistics

Number of PURLs generated manually by agents for each Customer.

Number of PURLs automatically generated for each Customer.

Open Rate Percentage

Open Timeline

Conversion Rate

Commercial Success

Revenues per PURL

Cost per PURL

Time per PURL

Impressions

Productivity

Metrics such as CTR per content provider & ad impressions.

CRM Metrics

Demographics - age, income, gender, ethnicity, education level, etc.

Lifestyle - interests, lifestyles, purchasing behavior, attitudes

Social Media – frequented sites, usage, visits

Survey Data - based on data available from welcome home surveys

Marketing Metrics

Sales Metrics

Service Metrics

In an embodiment, when the "Forward to a Friend" feature is used by the Customer, an e-mail is created and automatically sent out by the system to invite the Friend to create their own Planning PURL with a link provided in the e-mail to do so. The originating Travel Agent for the Customer is then notified that this has occurred by being copied on this e-mail sent to the Friend so that the process has now also generated a new opportunity for the Agent to be of service to the Friend as well. Consequently, a planning PURL can be easily created and shared between different users, who may then use the Planning PURL as the basis for their travel excursion.

Thus, in an aspect, there is provided a computer-implemented method for generating a personalized universal resource locator (PURL") for a customer, comprising: identifying a unique customer; collecting content data from a plurality of data sources; pre-caching the content data for access by the unique customer; and generating a PURL with one or more links to the pre-cached content data prepared for access by the unique customer.

In an embodiment, the computer-implemented method further comprises collecting the content data in one or more landing databases.

In another embodiment, pre-caching the content data comprises preparing the content data in one or more staging databases prior to access by the unique customer.

In another embodiment, pre-caching the content data further comprises preparing the content data based on the unique customer's profile.

In another embodiment, the computer-implemented method further comprises retrieving the pre-cached content data from the one or more staging databases in dependence upon interaction by the unique customer with the one or more links to the pre-cached content data in the PURL.

In another embodiment, the computer-implemented method further comprises: identifying one or more stages of a life-cycle for a product or service in relation to the unique customer; pre-caching content data relevant to the current life-cycle of the product or service in relation to the unique customer; and rendering personalized

content data relevant to the current life-cycle of the product or service for the unique customer.

In another embodiment, the computer-implemented method further comprises retrieving pre-cached content data from one or more staging databases in dependence upon interaction by the unique customer with the one or more links to the pre-cached content data in the PURL.

In another embodiment, the product or service relates to a travel excursion, and the life-cycle of the product or service relates to the stages of the travel excursion.

In another embodiment, the life-cycle of the travel excursion includes planning travel, booked travel, and after travel.

In another embodiment, the computer-implemented method further comprises generating a personalized itinerary of all products and services booked through the PURL by the unique customer.

In another embodiment, the computer-implemented method further comprises selecting a PURL skin associated with a product vendor or service provider; and rendering personalized content data corresponding to the product vendor or service provider.

In another embodiment, the personalized content data corresponding to the product vendor or service provider is connected to products or services offered by the product vendor or service provider.

In another embodiment, the computer-implemented method further comprises generating a planning PURL for access by the unique customer prior to purchase of a product or service.

In another embodiment, the computer-implemented method further comprises sharing the planning PURL such that others can review the PURL prior to purchase of the product or service.

In another aspect, there is provided a computer-implemented system for generating a personalized universal resource locator (PURL") for a customer, the system adapted to:

identify a unique customer; collect content data from a plurality of data sources; pre-cache the content data for access by the unique customer; and generate a PURL with one or more links to the pre-cached content data prepared for access by the unique customer.

In an embodiment, the system is further adapted to collect the content data in one or more landing databases.

In another embodiment, the system is further adapted to pre-cache the content data by preparing the content data in one or more staging databases prior to access by the unique customer.

In another embodiment, the system is further adapted to pre-cache the content data by preparing the content data based on the unique customer's profile.

In another embodiment, the system is further adapted to retrieve the pre-cached content data from the one or more staging databases in dependence upon interaction by the unique customer with the one or more links to the pre-cached content data in the PURL.

In another embodiment, the system is further adapted to: Identify one or more stages of a life-cycle for a product or service in relation to the unique customer; pre-cache content data relevant to the current life-cycle of the product or service in relation to the unique customer; and render personalized content data relevant to the current life-cycle of the product or service for the unique customer.

In another embodiment, the system is further adapted to retrieve pre-cached content data from one or more staging databases in dependence upon interaction by the unique customer with the one or more links to the pre-cached content data in the PURL.

In another embodiment, the product or service relates to a travel excursion, and the life-cycle of the product or service relates to the stages of the travel excursion.

In another embodiment, the life-cycle of the travel excursion includes planning travel, booked travel, and after travel.

In another embodiment, the system is further adapted to generate a personalized itinerary of all products and services booked through the PURL by the unique customer.

In another embodiment, the system is further adapted to: select a PURL skin associated with a product vendor or service provider; and render personalized content data corresponding to the product vendor or service provider.

In another embodiment, the personalized content data corresponding to the product vendor or service provider is connected to products or services offered by the product vendor or service provider.

In another embodiment, the system is further adapted to generate a planning PURL for access by the unique customer prior to purchase of a product or service.

In another embodiment, the system is further adapted to share the planning PURL such that others can review the PURL prior to purchase of the product or service.

It should be understood that further enhancements to the disclosed system, method and computer program are envisioned.

CLAIMS:

1. A computer-implemented method for generating a personalized universal resource locator (PURL™) for a customer, comprising:

identifying a unique customer;

collecting content data from a plurality of data sources;

pre-caching the content data for access by the unique customer; and

generating a PURL with one or more links to the pre-cached content data prepared for access by the unique customer.

2. The computer-implemented method of claim 1, further comprising collecting the content data in one or more landing databases.

3. The computer-implemented method of claim 2, wherein pre-caching the content data comprises preparing the content data in one or more staging databases prior to access by the unique customer.

4. The computer-implemented method of claim 3, wherein pre-caching the content data further comprises preparing the content data based on the unique customer's profile.

5. The computer-implemented method of claim 4, further comprising retrieving the pre-cached content data from the one or more staging databases in dependence upon interaction by the unique customer with the one or more links to the pre-cached content data in the PURL.

6. The computer-implemented method of claim 1, further comprising:

identifying one or more stages of a life-cycle for a product or service in relation to the unique customer;

pre-caching content data relevant to the current life-cycle of the product or service in relation to the unique customer; and

rendering personalized content data relevant to the current life-cycle of the product or service for the unique customer.

7. The computer-implemented method of claim 6, further comprising retrieving pre-cached content data from one or more staging databases in dependence upon interaction by the unique customer with the one or more links to the pre-cached content data in the PURL.

8. The computer-implemented method of claim 7, wherein the product or service relates to a travel excursion, and the life-cycle of the product or service relates to the stages of the travel excursion.

9. The computer-implemented method of claim 7, wherein the life-cycle of the travel excursion includes planning travel, booked travel, and after travel.

10. The computer-implemented method of claim 7, further comprising generating a personalized itinerary of all products and services booked through the PURL by the unique customer.

11. The computer-implemented method of claim 1, further comprising:

selecting a PURL skin associated with a product vendor or service provider; and

rendering personalized content data corresponding to the product vendor or service provider.

12. The computer-implemented method of claim 9, wherein the personalized content data corresponding to the product vendor or service provider is connected to products or services offered by the product vendor or service provider.

13. The computer-implemented method of claim 1, further comprising generating a planning PURL for access by the unique customer prior to purchase of a product or service.

14. The computer-implemented method of claim 13, further comprising sharing the planning PURL such that others can review the PURL prior to purchase of the product

or service.

15. A computer-implemented system for generating a personalized universal resource locator (PURL") for a customer, the system adapted to:

identify a unique customer;

collect content data from a plurality of data sources;

pre-cache the content data for access by the unique customer; and

generate a PURL with one or more links to the pre-cached content data prepared for access by the unique customer.

16. The computer-implemented system of claim 15, wherein the system is further adapted to collect the content data in one or more landing databases.

17. The computer-implemented system of claim 16, wherein the system is further adapted to pre-cache the content data by preparing the content data in one or more staging databases prior to access by the unique customer.

18. The computer-implemented system of claim 17, wherein the system is further adapted to pre-cache the content data by preparing the content data based on the unique customer's profile.

19. The computer-implemented system of claim 18, wherein the system is further adapted to retrieve the pre-cached content data from the one or more staging databases in dependence upon interaction by the unique customer with the one or more links to the pre-cached content data in the PURL.

20. The computer-implemented method of claim 15, wherein the system is further adapted to:

identify one or more stages of a life-cycle for a product or service in relation to the unique customer;

pre-cache content data relevant to the current life-cycle of the product or service

in relation to the unique customer; and

render personalized content data relevant to the current life-cycle of the product or service for the unique customer.

21. The computer-implemented system of claim 20, wherein the system is further adapted to retrieve pre-cached content data from one or more staging databases in dependence upon interaction by the unique customer with the one or more links to the pre-cached content data in the PURL.

22. The computer-implemented system of claim 21, wherein the product or service relates to a travel excursion, and the life-cycle of the product or service relates to the stages of the travel excursion.

23. The computer-implemented system of claim 21, wherein the life-cycle of the travel excursion includes planning travel, booked travel, and after travel.

24. The computer-implemented system of claim 21, wherein the system is further adapted to generate a personalized itinerary of all products and services booked through the PURL by the unique customer.

25. The computer-implemented system of claim 15, wherein the system is further adapted to:

select a PURL skin associated with a product vendor or service provider; and

render personalized content data corresponding to the product vendor or service provider.

26. The computer-implemented system of claim 15, wherein the personalized content data corresponding to the product vendor or service provider is connected to products or services offered by the product vendor or service provider.

27. The computer-implemented system of claim 15, wherein the system is further adapted to generate a planning PURL for access by the unique customer prior to purchase of a product or service.

28. The computer-implemented system of claim 27, wherein the system is further adapted to share the planning PURL such that others can review the PURL prior to purchase of the product or service.

1/12

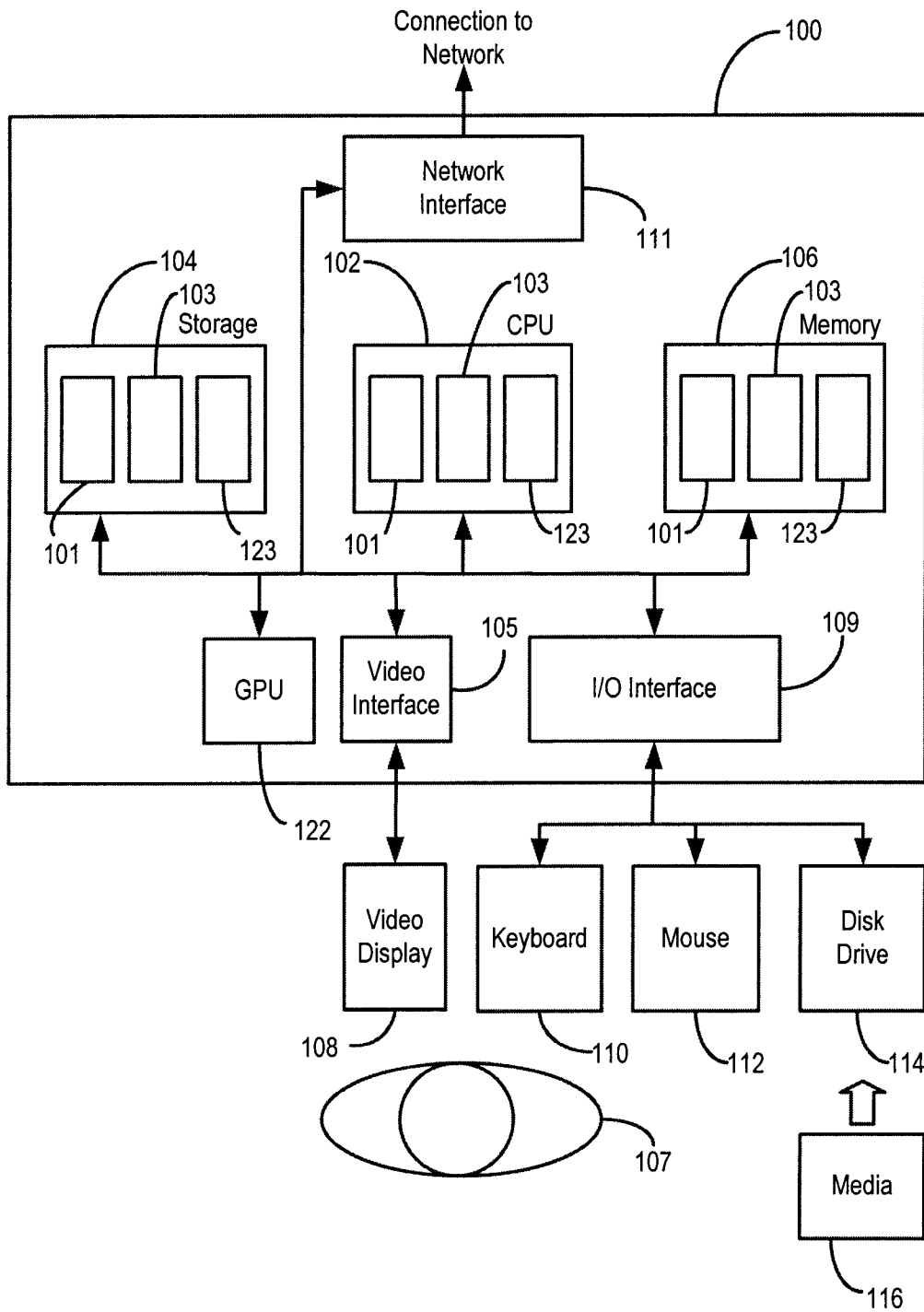


FIG. 1

2/12

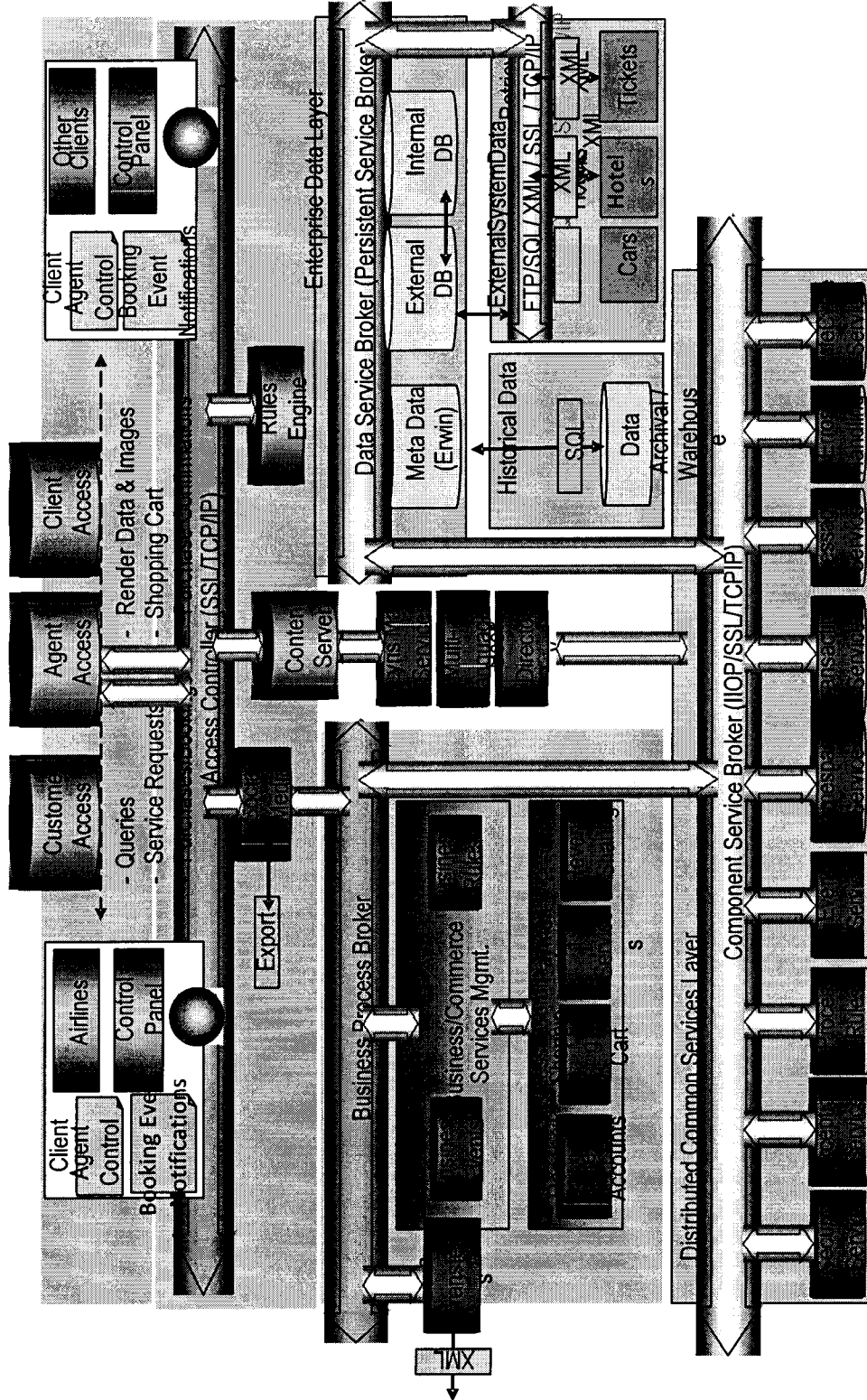


FIG. 2

3/12

Content: Requires Process, Architecture, Presentation, Management Services and Availability

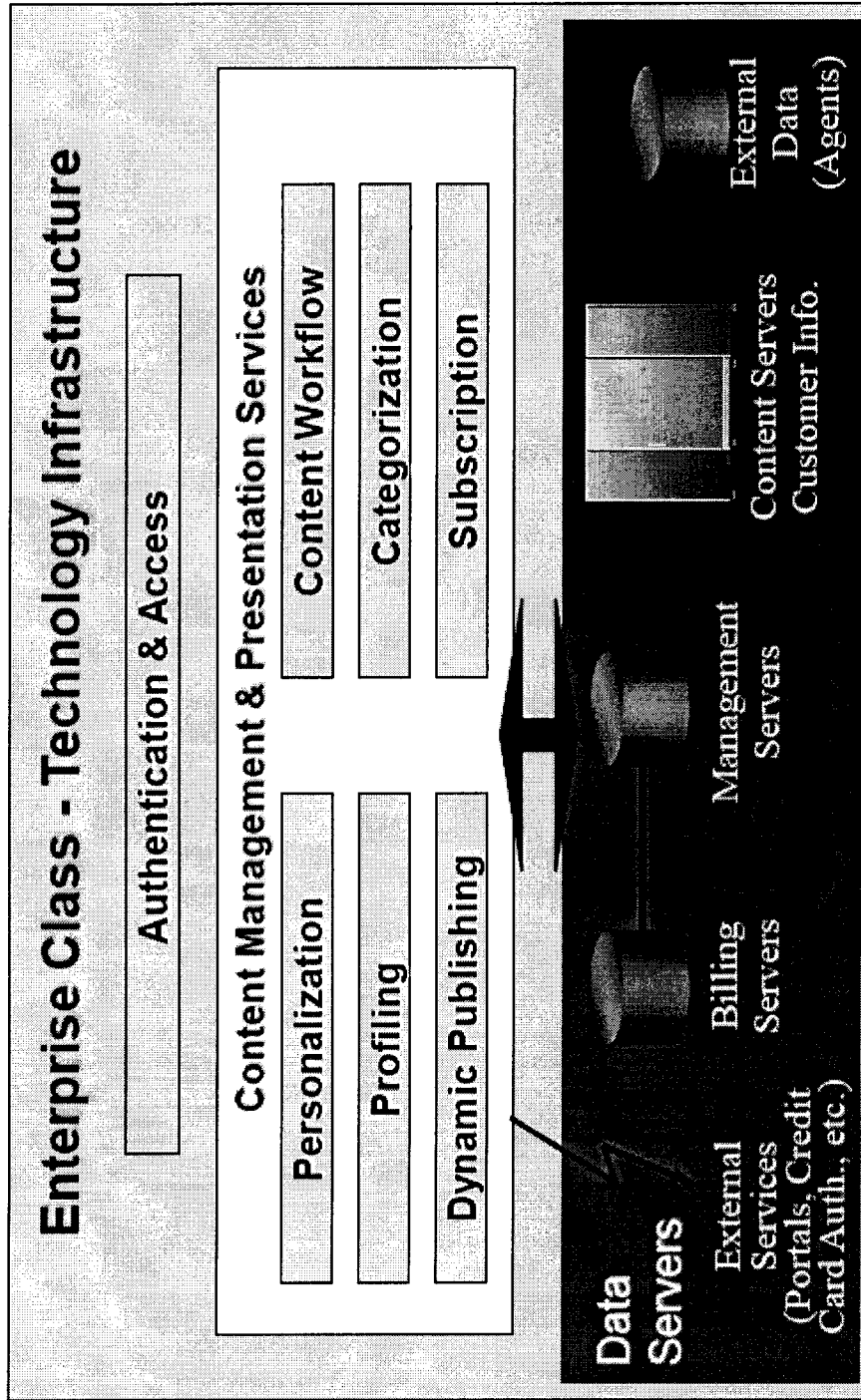


FIG. 3

4/12

WCF Architecture Strategy – R1 (Rendering PURL's to User Websites)

MS ASP.NET MVC implementation

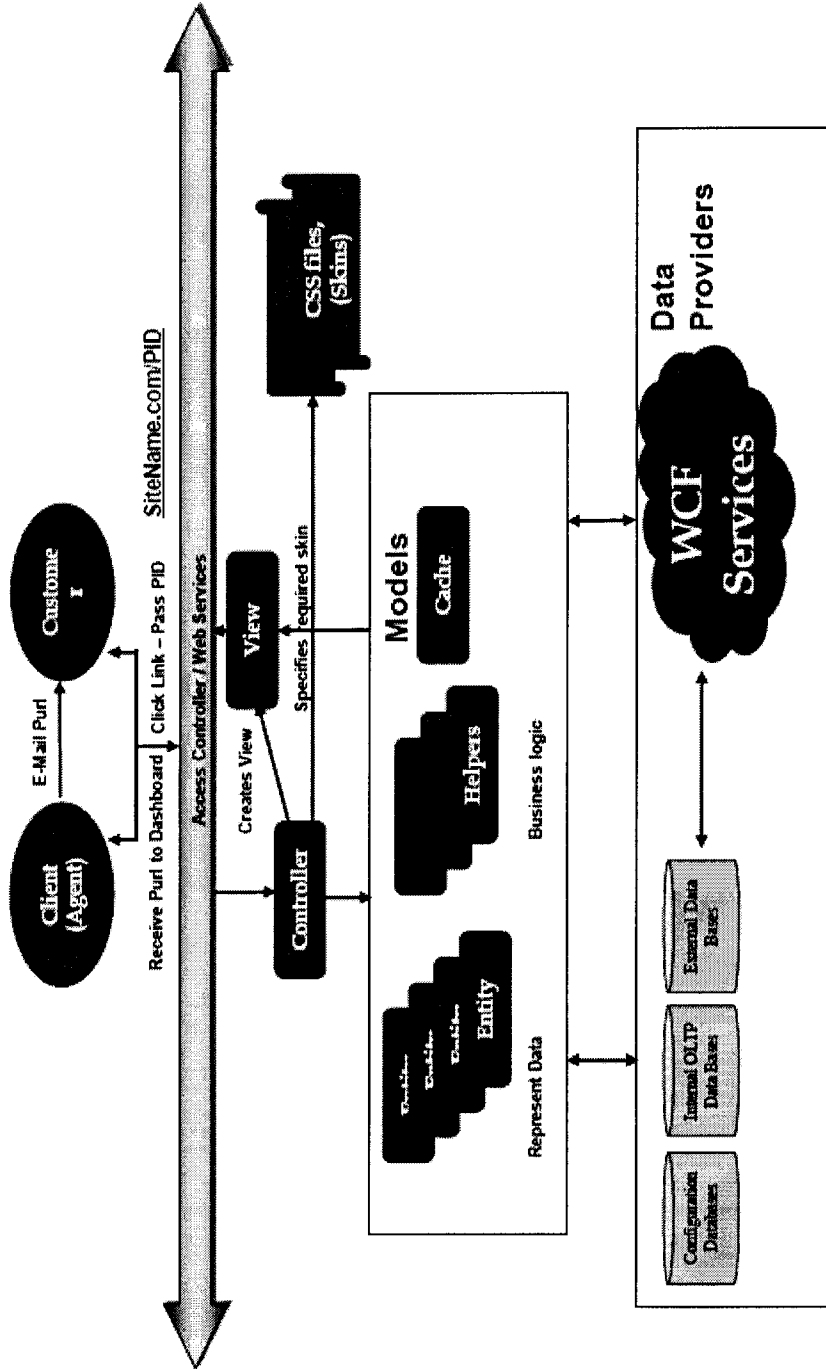
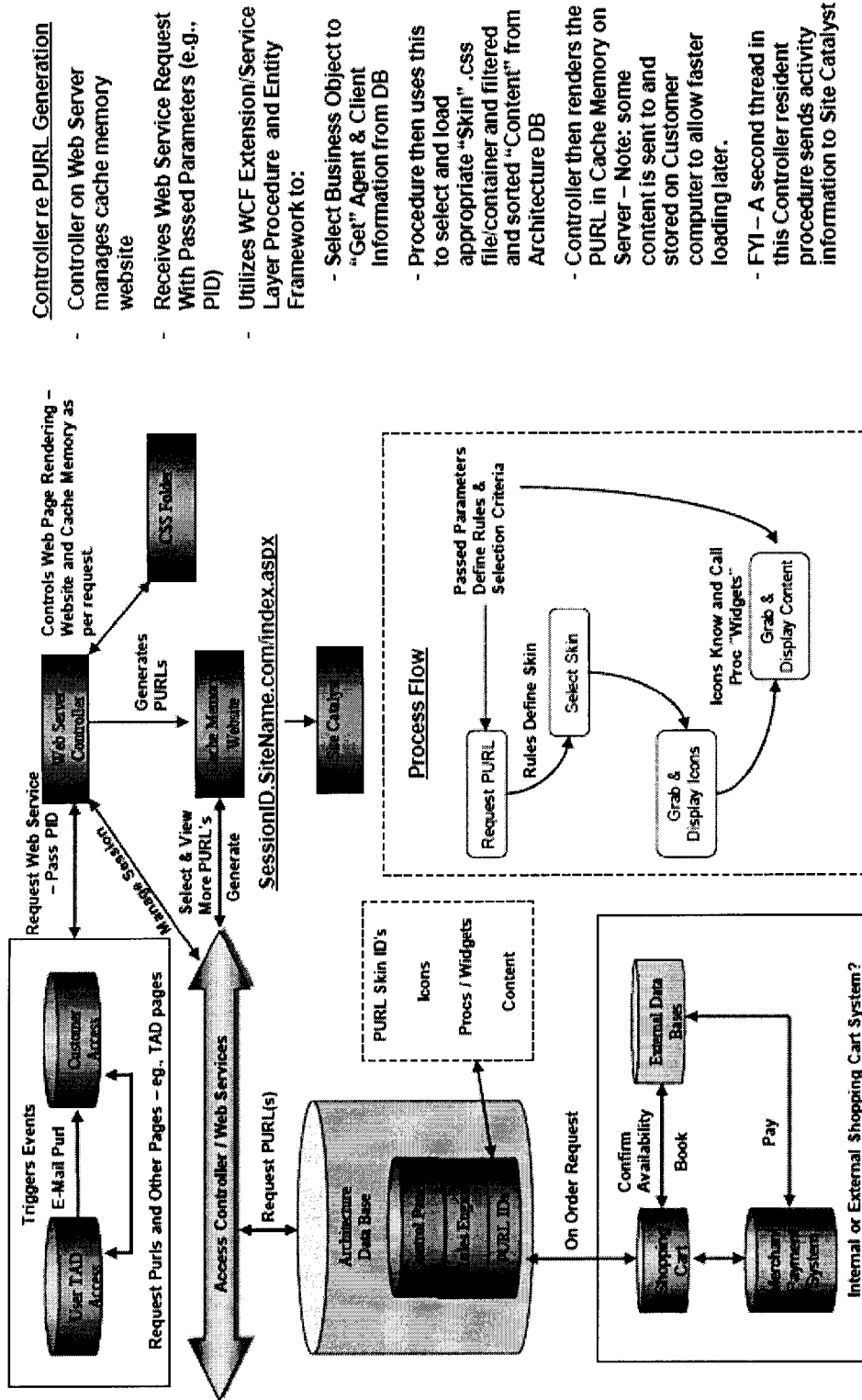


FIG. 4 400

Customer Purl "Skin" Architecture - R1



- Controller re PURL Generation
- Controller on Web Server manages cache memory website
- Receives Web Service Request With Passed Parameters (e.g., PID)
- Utilizes WCF Extension/Service Layer Procedure and Entity Framework to:
 - Select Business Object to "Get" Agent & Client Information from DB
 - Procedure then uses this to select and load appropriate "Skin" .css file/container and filtered and sorted "Content" from Architecture DB
 - Controller then renders the PURL in Cache Memory on Server - Note: some content is sent to and stored on Customer computer to allow faster loading later.
- FYI - A second thread in this Controller resident procedure sends activity information to Site Catalyst

5/12

FIG. 5 500

6/12

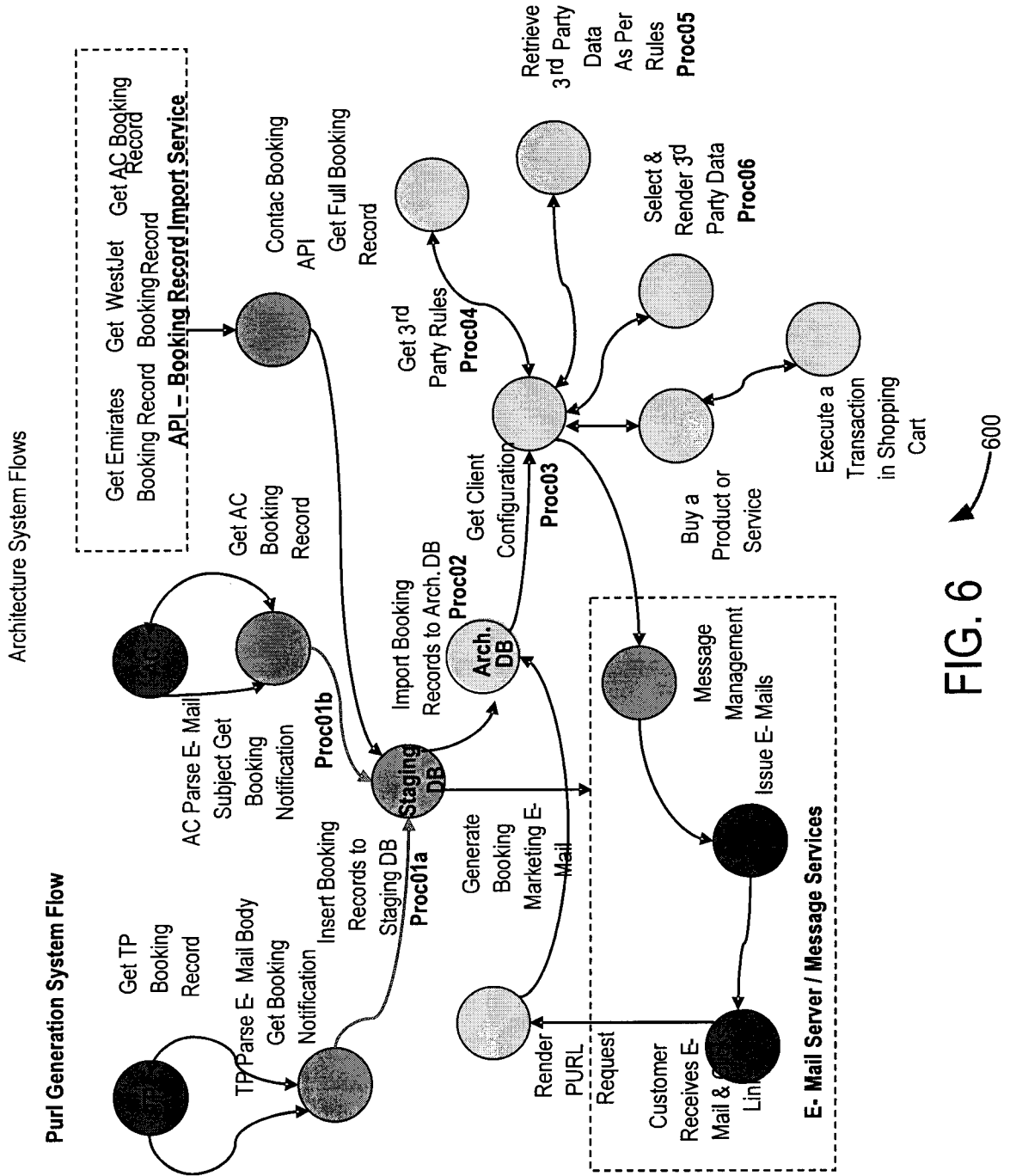
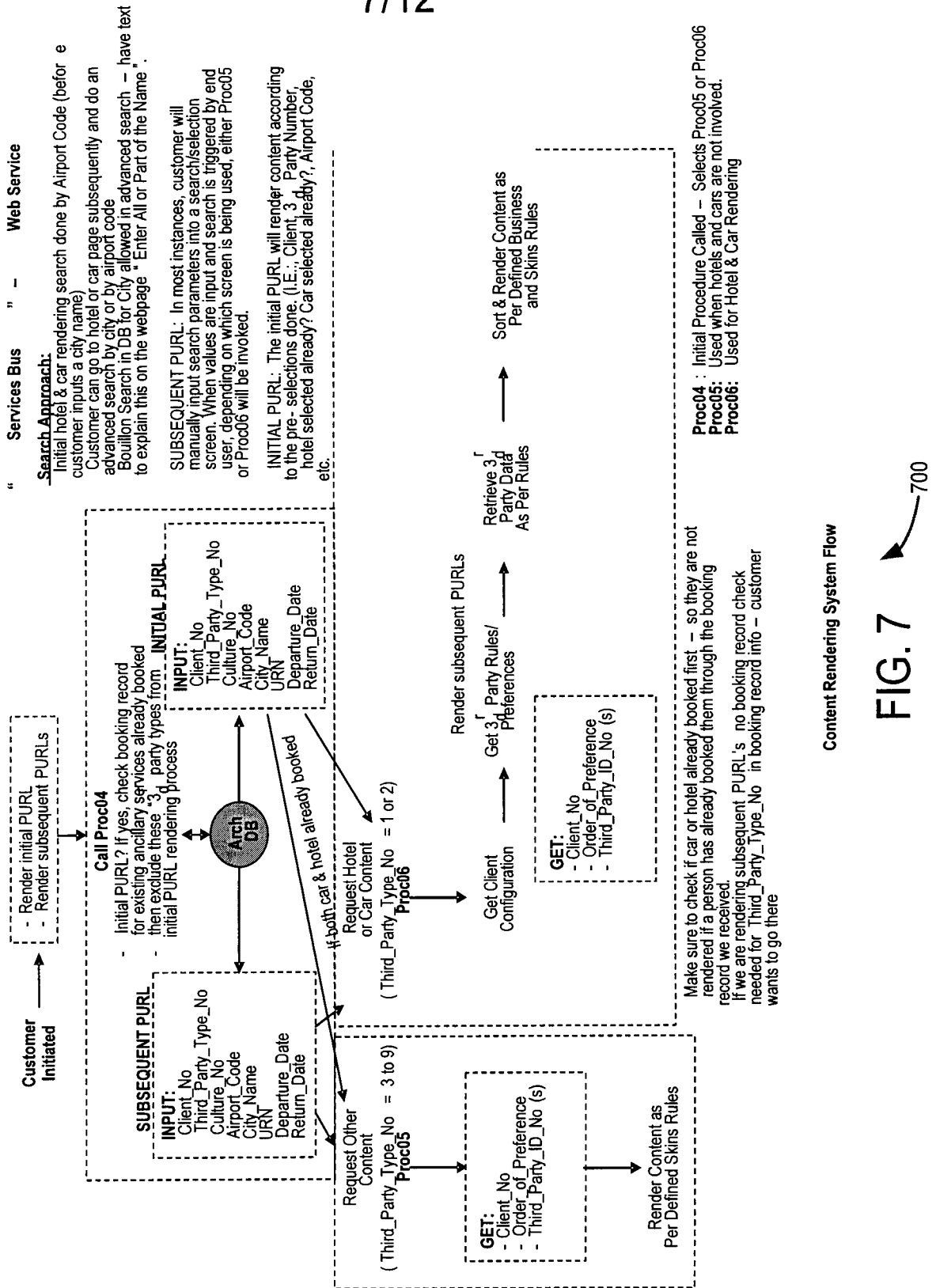


FIG. 6

7/12



8/12

- Pull in Destination Name
- Pull in Weather
- Pull in first name of traveller
- Pull in Agency name, Agent ID and Agent Name
- Access Your site link
- Share on facebook link
- Share on Twitter link

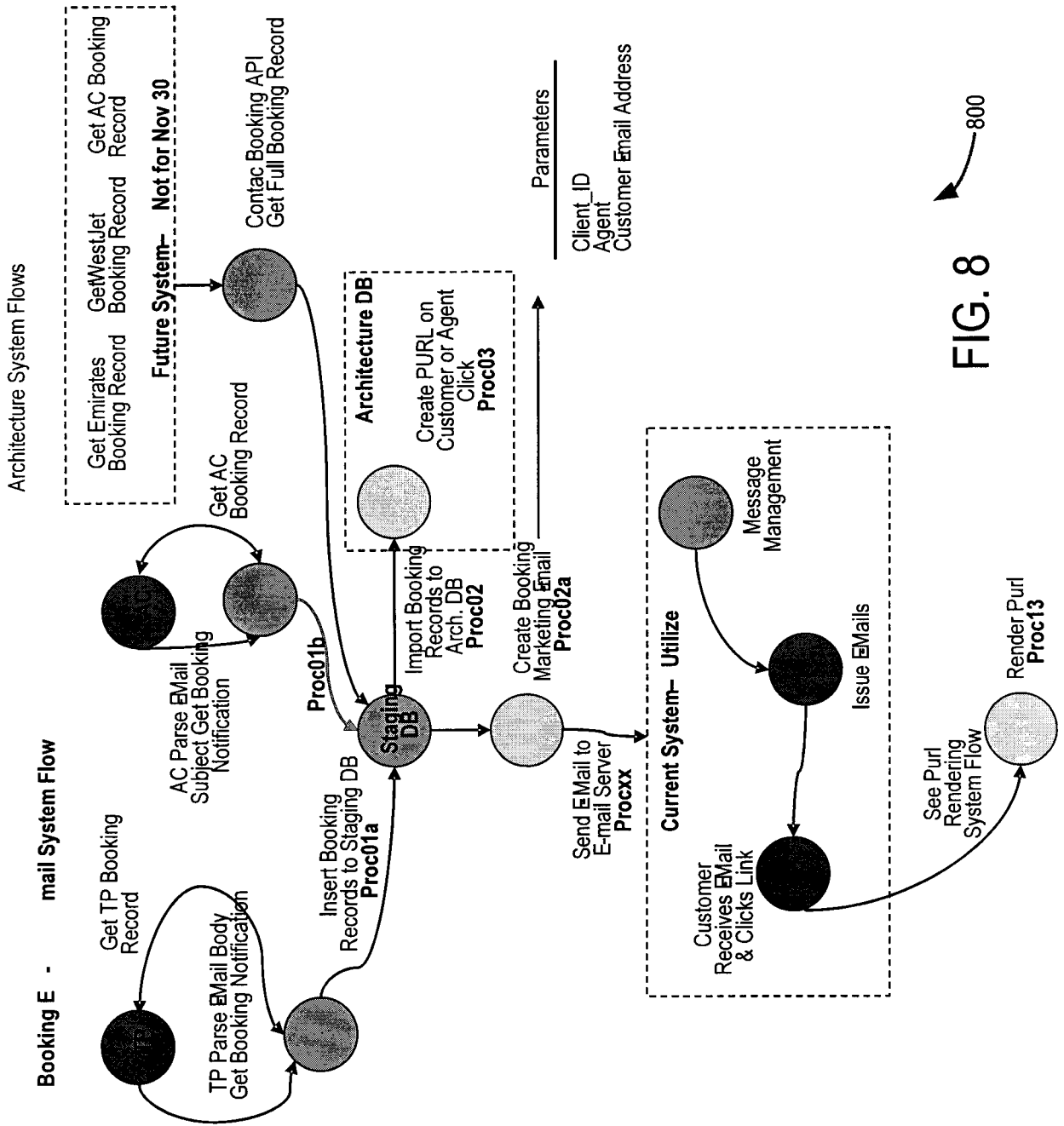


FIG. 8

800

9/12

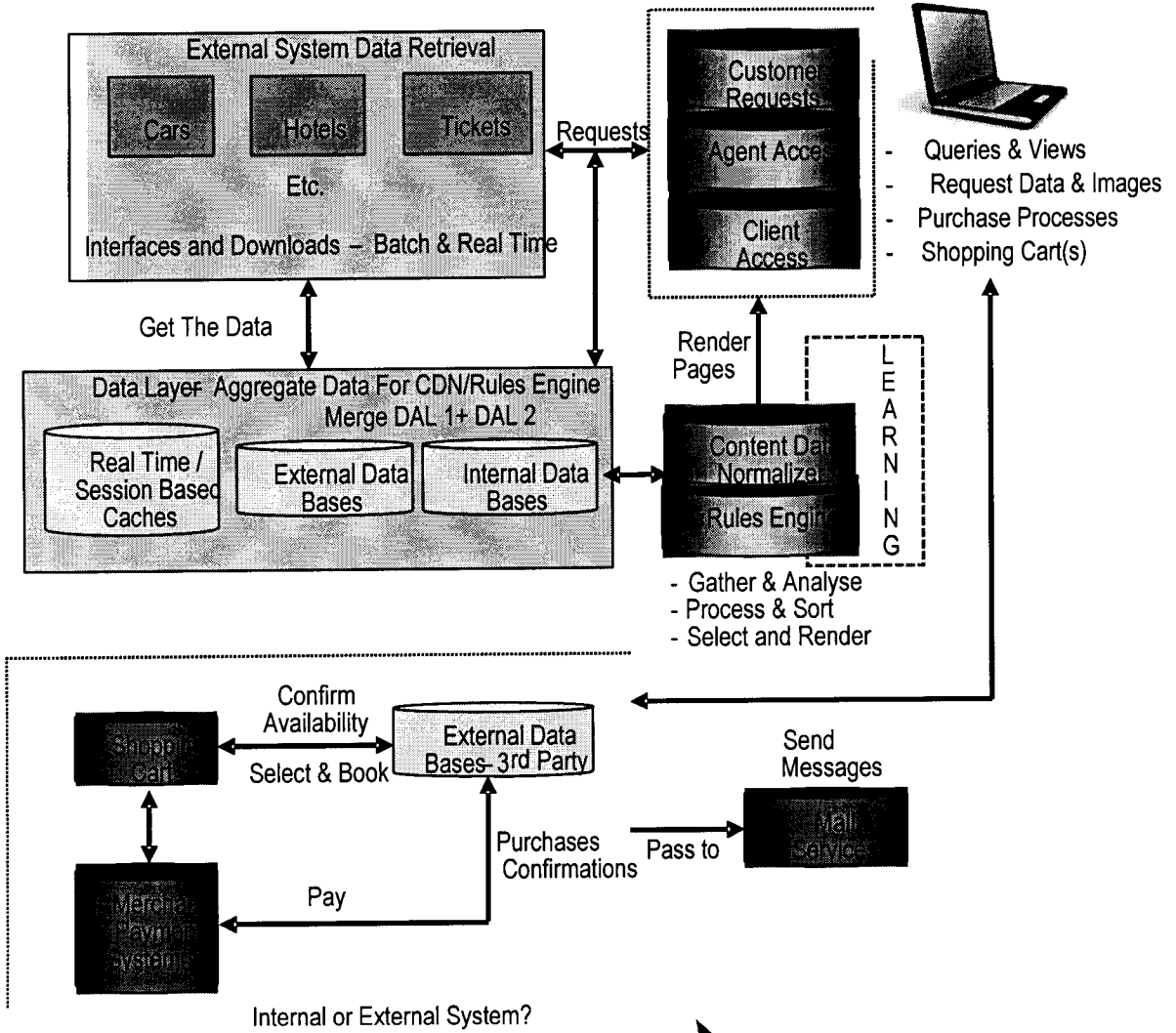


FIG. 9

900

10/12

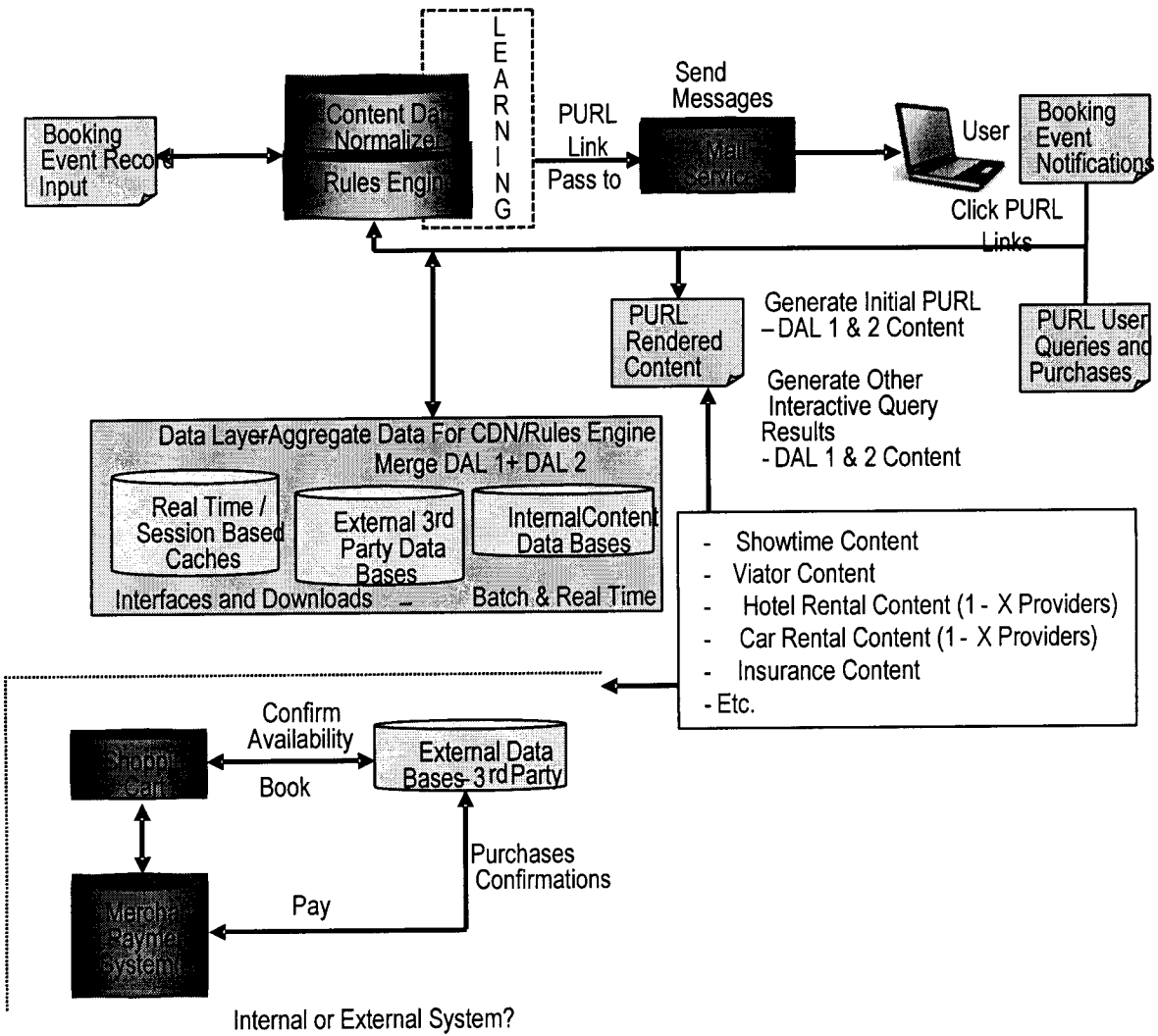


FIG. 10

11/12

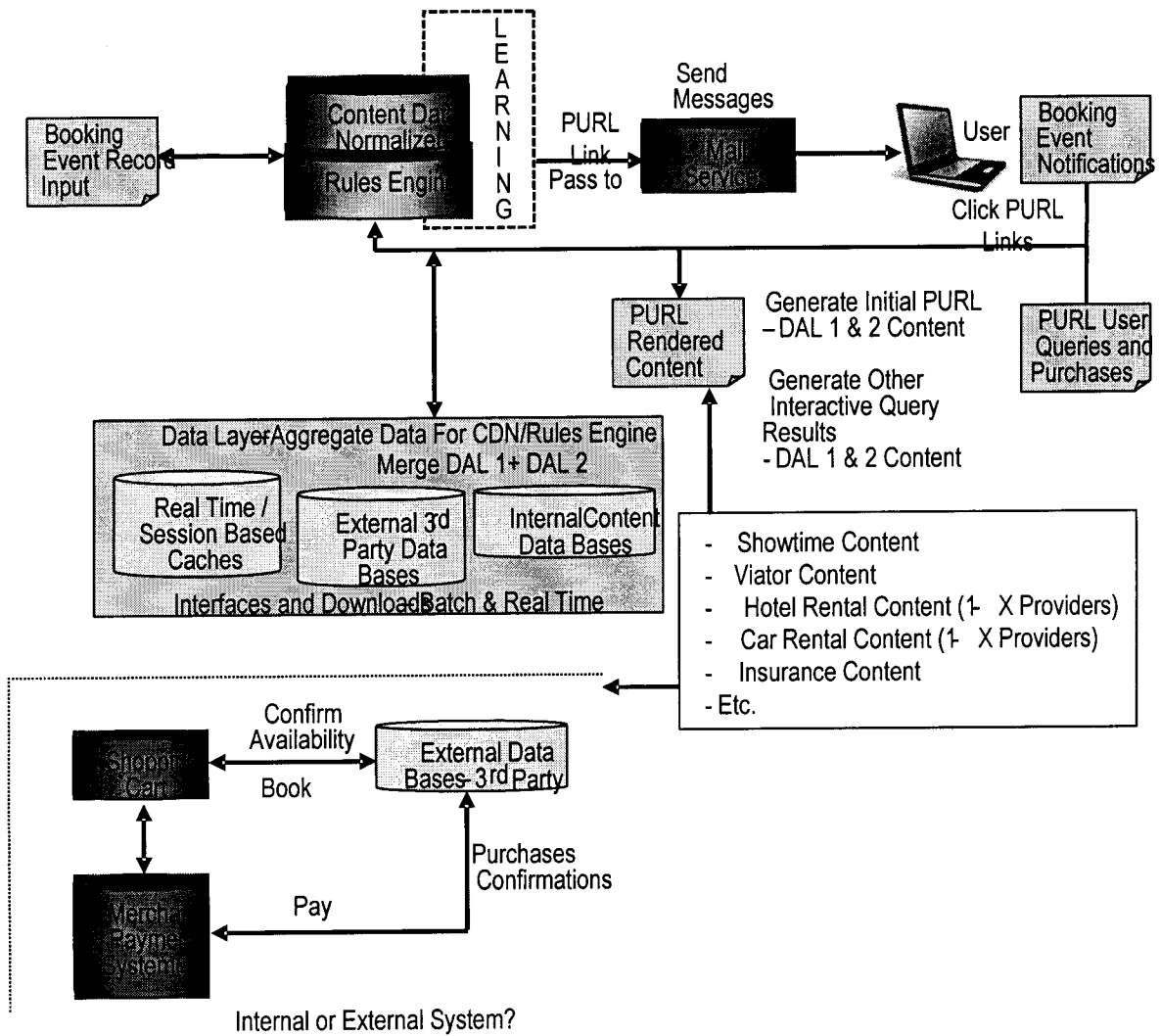


FIG. 11

1100

12/12

myworld™ Airline Booking Process



FIG. 12 1200

CONTACT

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2013/000579

<p>A. CLASSIFICATION OF SUBJECT MATTER IPC: H04L 29/02 (2006.01) , G06F 17/00 (2006.01) , H04L 12/16 (2006.01) , G06Q 30/00 (2012.01) According to International Patent Classification (IPC) or to both national classification and IPC</p>		
<p>B. FIELDS SEARCHED</p>		
<p>Minimum documentation searched (classification system followed by classification symbols) IPC: H04L* (2006.01) , G06F* (2006.01) , G06Q* (2012.01)</p>		
<p>Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched</p>		
<p>Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used) Canadian Patent Database, TotalPatent, Google Keywords: pURL, personal* URL, client, customer, database, rules engine, website, profile, cach*, content data, pre-cach*</p>		
<p>C. DOCUMENTS CONSIDERED TO BE RELEVANT</p>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2009/0271259 A1 (FISH), 29 October 2009 (29-10-2009) **see para. [0008], [0015] and [0018]	1-5, 13-19, 27 and 28
Y	US 2011/0131341 A1 (YOO et al.), 2 June 2011 (02-06-2011) **see para. [0013], [0018], [0026] and [0033-0037]	1-5, 13-19, 27 and 28
A	US 2010/0211448 A1 (BEENAU et al.), 19 August 2010 (19-08-2010) **see entire document	1 and 15
<p><input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.</p>		
*	Special categories of cited documents :	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier application or patent but published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means	
"P"	document published prior to the international filing date but later than the priority date claimed	
<p>Date of the actual completion of the international search 12 February 2014 (12-02-2014)</p>		<p>Date of mailing of the international search report 18 February 2014 (18-02-2014)</p>
<p>Name and mailing address of the ISA/CA Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476</p>		<p>Authorized officer Jamie Hayami (819) 934-2670</p>

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CA2013/000579

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US2009271259A1	29 October 2009 (29-10-2009)	None	
US2011131341A1	02 June 2011 (02-06-2011)	None	
US2010211448A1	19 August 2010 (19-08-2010)	US2010161493A1	24 June 2010 (24-06-2010)