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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 September 2009 (24.09.2009)

PCT

(10) International Publication Number
WO 2009/115921 A2

(51) **International Patent Classification**:
G06F 9/44 (2006.01)

(21) **International Application Number**:
PCT/IB2009/005387

(22) **International Filing Date**:
23 February 2009 (23.02.2009)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
449/CHE/2008 22 February 2008 (22.02.2008) IN

(71) **Applicant** *(for all designated States except US)*: **IPATH TECHNOLOGIES PRIVATE LIMITED** [IN/IN]; 79 Bazulla Road, Chennai 600017 (IN).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: **SRINIVASAN, Ramakrishnan** [IN/IN]; 79 Bazulla Road, Chennai 600017 (IN). **VAITHEES, S., Kumar** [IN/IN]; No. 24 Dhanasekaran Street Kodambakkam, Chennai 600024 (IN).

(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, BR, BW, BY, BZ, CA, CH, 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, KM, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (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, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

WO 2009/115921 A2

(54) **Title**: TECHNIQUES FOR ENTERPRISE RESOURCE MOBILIZATION

(57) **Abstract**: An enterprise mobilization system having an EUS which receives user requirement and translates the requirement into a content component and platform independent delivery component. A DSIM receives an information tree based on the content component and translates it into requests for data from at least one data source. The information is contextualized. The system provides an ability to take actions based on the context. All user experiences related to the mobilization are achieved through the concept of end-user services.

TECHNIQUES FOR ENTERPRISE RESOURCE MOBILIZATION

Background

[01] One can say that this is the age of Enterprise Mobility. In today's world, mobility has become more of a necessity than an option and information is what makes companies run. And as workers become more mobile, there is a high demand for companies to provide information that travels with people as they move around – in other words, information to follow people as they move.

[02] Today, there is an abundance of information and this information is stored in different formats across different sources. Further, due to the rapid progress of technology, there is an abundance of channels across which this information can be supplied to an end-user. For example, information regarding the current points tally of a country in the Olympic games can be supplied to an end-user via e-mail, short message service (SMS), etc. Simply put, technology enables information to be delivered via any channel.

[03] However, the delivery channel through which information is delivered is not always as per the end user's wish. Further, the information that is provided to the end-user is not necessarily based on what the end-user needs most. For example, the CEO of a company does not need a detailed report of what happened in each department every day. In fact the CEO may not even have the time to go through each report. The CEO would probably prefer to receive only an executive summary of the report, rather than the complete report itself. The CEO would also prefer that when he/she is in the office, the executive summary should come to the PC instead of his PDA. If the CEO is traveling, the executive summary should be even more precise and should be delivered to his PDA, because the experience that an end user gets for the same information on a 17 inch monitor is very different from the experience the end-user gets on a small PDA screen. Hence, what is needed is a solution that will

understand and identify the needs of the end-user (the CEO in the above example) and give the end-user the information that he/she wants, in the format that the end-user wants, and on the delivery channel that the end-user wants. Hence, the delivery of the information to the end-user should be independent of the source of the data, and should rather be guided by the wishes and needs of the end-user.

[04] Such a solution would ensure that better decisions can be made efficiently because the information that is needed is available at the user's fingertips and not in the office, in the data center or constrained by wired networks.

[05] Further, what is needed is a Unified Information Execution and Delivery Platform that provides rich context-specific information delivery with relationship between content, people and actions anytime, anywhere and to anyone through any device. What is also needed is that actioning tools be made available when the information is delivered, i.e., when information is made available to the end-user, the end-user would most likely want to take actions based on the delivered information. For example, a business manager may periodically receive sales reports of various regions on his PDA. The business manager would most likely want to discuss these reports with the regional heads. Hence, a solution is needed which would provide the business manager with the capability of connecting to his regional heads (possibly by a simple click on the PDA or by a simple voice command), instead of searching through the telephone directory for the telephone numbers of the regional heads. Along with the above capability, the solution should provide the regional head whom the business manager is calling, relevant information (sales report) so that the business manager and the regional head can be on the same page and come up with an action plan.

List of Figures

Fig. 1 shows an overview of context based information.

FIG. 2 gives an example of a system that provides an enriched mobility experience on various communication channels.

Fig. 3 shows further details of RTE and its interaction with other components.

Detailed Description

[06] To achieve one or more of the above objectives, the disclosed teachings provide a system and method for managing an enterprise and providing a unique mobility experience. An exemplary implementation of the disclosed teachings provide a middleware platform that can provide mobility experiences which includes providing highly context specific actionability using the communication/mode of choice.

[07] FIG. 2 gives an example of a system that provides an enriched mobility experience on various communication channels. In FIG. 2, the Access Layer 100 establishes communication between the Run Time Engine (RTE) 200 and the destination devices such as a PDA, a personal computer, etc. The Access Layer may include a web server such as an IBM WebSphere, or a voice gateway such as a CISCO 5300 or Vegastream. For example, this layer could include Message Queues as implemented by MS or IBM, Audio and Video Streaming Server, FTP, SMPP and SMTP Gateways. However, the components of the Access Layer are not limited to these components and many more components that can establish communication between various destination devices and the RTE 200, may be used.

[08] FIG. 2 also includes a RTE 200 that is responsible for the execution of the end user services (more information on the composition of the RTE 200 is described later). The RTE 200 represents the primary component of the mobilization platform. The RTE is implemented as a web application. The RTE receives requests from interactive channels like

PC/Voice/Mobile browsers, PC/Mobile applications, Scheduler/Polling Engines 300 and Non-Interactive Service Gateway 400. The RTE has implementations of functionality for communication, personalization, metadata management.. RTE uses DSIMs 700-1,2.. as a means to integrate to external data sources. RTE's primary job is to process EUS execution requests, get it processed by either On-Server Actions or by DSIMs. RTE also interfaces with delivery components/templates to transformation information for delivery/presentation. The RTE 200 is also connected to the Scheduler/Polling Engine 300 and the Asynchronus Gateway 400.

[09] The Scheduler/Polling Engine 300 periodically polls/triggers notifications and causes execution of a certain periodic end user service (EUS). The RTE 200 maintains a database of EUS execution schedules. The EUS execution scheduled provide the information from back-end systems (500) to perform notifications. The job of scheduler is to periodically run the scheduled EUS' The Asynchronous Gateway 400 is responsible for triggering an EUS asynchronously, i.e., due to an asynchronous/random event. The asynchronous event may be generated by the back end system 500.

[10] The back end system 500 is the source of the core data for the enterprise. The back end system 500 may include databases, standards based web-services, file systems, legacy back end-systems, etc. While the back-end system itself is not configured the disclosed system integrates to it. In that sense the IPath Configurator, is able to identify and present all data sources (and associated functionality exposed by them) existing in a business domain. The RTE 200 connects to the multiple heterogeneous sources of the back-end system 500 via DSIMs (Data Source Integration Modules) 700-1...700-n. DSIM refers to Data Source Interface Modules that are "thin layers of integration" between iAllWays App and third party data stores or information services (standards based web services for example). Requests to DSIMs are in terms of the Information Tree (which itself is a representation of all

information in solution domain from a business perspective). The DSIM translates this to a native request based on the specific type of DS it connects to (for example, convert to a database request if DSIM is interfacing with an RDBMS). DSIM uses BER (refer later comments for how EUS' are executed) to process request; is responsible for transforming output of BER to a iAllWays standard XML. DSIMs may be thought of as interpreters, all dealing with English from an end-user perspective but interfacing in the background with German, French, Greek, information sources The data is delivered to the destination device being used by the end-user via the access layer 100. The RTE 200 provides this data to the access layer 100 thru delivery components 800.

[11] A delivery component 800 may be a dynamic template or templatized applications. A delivery component is a block of software logic that generates either a browser experience (HTML, VXML or xHTML page) or generates code set (in this case it may be thought of as a templatized application. The template evaluation is based on primarily 3 types of information – user session information, state of EUS execution information and configuration information (which is the iAllWays application data). Connection to access layer is outside delivery component. Either the delivery component execution is happening from within an access layer connection context (say, a user request from a PC browser that comes through a Web Server; the template itself represent a dynamic web page) or is used as a parameter to an access layer connection (say, the template output determines a HTML body of an email and is used by an On-Server Action that establishes connection with an SMTP server to push mail)

[12] A security platform 600 may also be connected to the RTE 200.

[13] The disclosed teachings will now be described in detail with reference to exemplary scenarios. A first exemplary scenario is now described. Let us say, Bob (an example for an end-user) wants to track e-mails from XYZ. Inc. Bob can use his Personal

Workspace (may be available as a portal on a device Bob is using) for the above service request. The tracking email service can be provided as a link in Bob's Personal Workspace. The above service request could be a one time request or it could be a personalized request that is run periodically, i.e., it could be a scheduled service. The personalized request can be stored in Bob's Personal Workspace for Scheduled Services. As seen from FIG. 3, Bob can make a one time request for tracking emails through Portal Services available in Bob's Personal Workspace or the Scheduler/Polling Engine can run Bob's personalized service request periodically (for example daily or every two hours) by polling the RTE 200. It should be noted that if Bob makes a one-time request from his Personal Workspace, which he may be accessing on his PDA, the one-time request is received by the RTE 200 through the Access Layer 100.

[14] The above request for tracking e-mails from XYZ.com is supplied to the RTE 200, which then executes the EUS "Check Mail". RTE comprises of a service execution gateway, a service execution manager, application data managers (for handling configuration data of different kinds – EUS, BER, IR, DS, TR...), On-Server Actions (OSA), DSIM Selection unit, OSA selection unit, Environment Manager, Personalization Manager. The EUS Execution Unit 200-1 is responsible for executing a service request from the end-user. The EUS Execution Unit 200-1 may be embodied as a hardware unit or may be a software code that defines the various EUS' that the system has to offer. For example, the EUS Execution Unit 200 may store the software definitions for each of the multiple EUS' that the system has to offer. .

[15] A technique similar to "if" or "switch case" blocks exits to process a request. However, they do not bear a one-to-one relation with the number of EUS definitions itself. The conditional happens around EUS type because there are some nuances to how it is

processed based on whether it is an information request for interactive/portal processing, notification processing, On-Server Action execution.

[16] As seen from the description above, the service request from Bob comes to the EUS execution unit 200-1 via the access layer 100. The EUS execution unit 200-1 first checks if an input is required from the user. This check is performed based on the definition of the EUS being requested. If no inputs are required (in this case it has already been specified that e-mails from "XYZ.com" need to be tracked), the EUS execution unit 200-1 forwards the request to the DSIM determination unit 200-2

[17] An EUS execution is being requested by user or scheduler. EUS has content/back-end info and delivery/presentation info. Content/Back-end info comprises of a Information Request. This in turn comprises of one or more user input blocks, a Visual Representation (a combination of information details) and a back-end request(BER). The BER has the Data Source (DS) associated with it. The DS has a type information that is tied to a particular DSIM.

[18] Now, the DSIM determination unit 200-2 determines the DSIM and the back end request (BER) to use for executing the EUS "Check Mail". For the present exemplary scenario, the DSIM determination unit 200-2 determines that it needs to connect to the Mail Server (determined based on back-end integration details in the service definition for "Check mails" from Mail server access based on a specific protocol like PO3 or IMAP. The particular IMAP_POP3 DSIM that is asked to execute the Information Request. The Information Request has an associated Back-End Request and the Back-End Request (BER) has an associated Data Source or DS (address and connectivity details for mail server, in this case). The DSIM uses DS information to establish connection to mail server (the connection itself may not be established and disconnected on a request-to-request basis!) and then the information in BER is used to make appropriate protocol request and obtain data from mail

server and uses an appropriate DSIM 700-1 to communicate with the Mail Server, which is part of the back end system 500. The DSIM 700-1 fetches the data from the Mail Server and may convert it to a format that is being used by the RTE 200.

[19] Once the data has been fetched from the Mail Server, a delivery component to be used for delivering the fetched data is determined. The delivery component to be used for delivering the fetched data is selected by a Delivery component selection unit 200-3. The Delivery Component selection unit 200-3 has a plurality of delivery components at its disposal.

[20] Personalized notification services have preferences on specific channels of delivery – email, SMS. Today it is a simple "toggle channel of delivery" capability available; With minimal extension to data stored, we can provide dynamic delivery decision making (such as, before 9am, sms me; between 9am and noon email me..) In the current exemplary scenario, if the data has to be delivered on a phone that Bob is using, the Delivery component selection unit 200-3 selects a delivery component 800 that delivers the data on a phone from a plurality of delivery components that may be available. After determining the delivery component, the Delivery component selection unit 200-3 transfers control to the selected delivery component and initiates delivery action of the EUS that is being executed.

[21] According to the current exemplary scenario, the selected delivery component 800 corresponds to Bob's phone. The selected delivery component 800 determines the context relations and presents actioning around people and information, i.e., - Bob's phone will ring and he may get a message such as "This is your system calling. You have a mail from paul@xyz.com. Received at 9:30am this morning.". The selected delivery component 800 determines the necessary information that needs to be presented to Bob. The determined information may be for example, links to related services such that Bob can take actions using those links. An action that Bob may take using those related links would trigger

another EUS in the RTE. For example, Bob may be provided with an option to check project status, which is a related action. The delivery component is able to determine this related action based on the context of the original EUS "Check Mail". "Check project status" is a related service because when mails are being accessed, project status information may be requested.

[22] Providing links or the capability to take a related action is made possible by storing relationships between different End User Services. In the above exemplary scenario a relationship was stored between "Check mail" and "Check project status", which are both End User Services. These relationships are stored in the Relationship storage unit 200-4. Before delivering the fetched data, the delivery component 800 determines if any relationships are stored for the EUS "Check Mail". Now, the relationship storage unit 200-4 may store a relationship between "Check Mail" and "Check project status". Therefore, the delivery component 800 is able to provide Bob with the ability to execute the "Check project status" EUS.

[23] In response to the related links/actioning capability presented to Bob, Bob may speak "check project status". Thus, Bob has requested execution of another EUS. In other words, Bob has triggered a relation action. The request is again routed through the EUS execution unit 200-1. In this case, the EUS execution unit 200-1 may determine that a project ID is needed and may request the project ID via the Access Layer 100. Once Bob enters the project ID, the EUS execution unit 200-1 proceeds with EUS execution as described above. In this case, the delivery component 800 in conjunction with the access layer 100 proceeds with delivery action and may deliver for example, a VXML output that is used by Voice Browser for speaking out project status details.

[24] Another exemplary scenario is now described. In this exemplary scenario John is a senior portfolio manager, responsible for over 50 different portfolios. John is

watching CNBC market report just after more than one leading vehicle manufacturer has reported serious losses. John estimates a domino effect in coming months on some software portfolios under his management, doing significant development in the auto space. John goes to his PDA to pull out information on number of customer portfolios affected by downturn in auto sector. In the present exemplary scenario, John himself found out that one of the leading vehicle manufacturers has reported losses. However, John could be notified of this issue by the Asynchronous Gateway 400, which is communicating with the back end system 500, i.e., when an event that is of consequence to John occurs in the back end system 500, the Asynchronous Gateway 500 can present this event to John through the RTE 200. The RTE 200 (via the delivery component in conjunction with the access layer) would then provide John with related information such that he can take the necessary actions, if any.

[25] Going back to the exemplary scenario in which John finds out that a vehicle manufacturer reported losses. John goes to his personal work space on his PDA in which John has his personalized "Impact Check Service". The "Impact Check Service" could be a service that lists customers whose portfolios have more than x% invested in a specific set of companies. Further, John could have personalized this service for a particular set of companies such as Ford, Hyundai, etc. Once John requests execution of this EUS, the service request is routed to the EUS execution unit 200-1, which runs the personalized service against the portfolio management DB.

[26] A similar sequence of events takes place as described in the exemplary scenario with Bob. Delivery action is now triggered by the RTE 200. In this case, the delivery component 800 may prepare a presentation of retrieved customer data, determines context, and includes 2 additional EUS' along with the customer list. The two additional EUS' that are delivered or presented for John to execute may be "Connect to Customer Relationship Manager for Portfolio" and "Get Portfolio Details for Customer"

[27] Delivery templates constitute delivery components. "Impact Check Service" defines the context. The 2 additional services listed as part of the delivery are related to "Impact Check Service". As noted earlier, the system provides John with these related EUS' because the relationship storage unit 200-4 stores a relation between different EUS'.

[28] John sees listing of customers impacted. He selects 1 or more customers and clicks link to select related action. John picks one of 2 possible related actions. In fact John triggers a "service orchestration", which is sequenced execution of EUS'. For example, John may trigger an EUS in which the RTE 200 pushes details of the portfolio belonging to customer selected by John. The EUS sequence may further proceed by establishing a live communication connection with the regional manager that is in charge of the selected portfolio.

[29] An orchestration is a pre-defined sequencing of EUS executions. The orchestration definition is itself exposed as a EUS and may be available for execution in the context of any other EUS execution. In this case, the orchestration may involve the following EUS sequencing: (1) For selected list of customers and portfolio holdings of interest, fetch holding information, (2) Fetch the list of RMs (email and phone information included) associated with these customers, (3) push details obtained in (1) to the RMs, (4) trigger a voice conferencing service that connects current user with RMs. Orchestrations are similar to MS Outlook Message Rule – on mail received from x, copy message to folder y, delete original message..

[30] An exemplary implementation of the disclosed teachings may be able to deliver rich context-specific information with relationship between content, people and actions anytime, anywhere and to anyone through any device. The delivery of the context-specific information and related actioning is made possible by storing relationships between different EUS'. As such, an enriched mobility experience can be provided to the end-user.

[31] The key to providing related action links is the storing of the relationships. Relationships may be explicit or implicit. Explicit relationships may Share Scope, Share output details, may be linked, may have shared information context, etc. When two different EUS' share at least one parameter that is input, the relationship is said to have shared scope. For example, an EUS "Sales for region X" and a related EUS "Competition info for region X" may share at least one same input parameter.

[32] Looking at regional sales performance for a particular period and particular product; need to look at performance for previous period for same product or get the sales people performance for same period and product line. When the output for an EUS provides the input for another related EUS, the two EUS' have a relationship that shares output details. For example, an EUS such as "Show all messages" may deliver messages from X number of people. Now a related EUS "Payment history for selected user" depends on an input, which may be selected from the output of the EUS "Show all messages", i.e., one of the X numbers of people. Two relationships may not have an input or output sharing and may still be linked. For example, a relationship may be stored such that whenever the EUS "List financial portfolios" is executed, a link service may be triggered that provides a link to the stock market to the user. A relationship may also be based on a shared information context – Represent personalized relationships; service relationships uniquely seen by individuals .

[33] While there is a preferred accepted way of working in a domain, individuals may have particular response patterns (contextual service selection preferences) that are unique. Preference on related service selections in context may be based on specific individual's hypotheses on correlations; if x happens, there is a possibility that y is a reason; proceed with standard investigation if y has been eliminated as a source of problem.

[34] Today relationships are explicitly defined using configurator. In the future "dynamic" relationships are believed to be possible. Relationships also could "dynamically"

evolve over time based on user transactions. The current state of data definitions already supports such a capability. Only our interfaces need to be modified for the purpose. What this means is that, in iAllWays, user has access to any end user service in catalogue, in the context of a particular EUS being viewed. If system captures this detail, there is the possibility of storing this as a service relationship; the fact that EUS B was selected in the context of EUS A by user X may have potential as a useful contextual relationship for a whole set of their users.

[35] The Following us a comparison of features of an exemplary embodiment of the disclosed system with related art.

FEATURE COMPARISON

[36] IBM, SAP and Oracle were chosen as the 3 premium vendors that share IPath's direction vis-à-vis "Enterprise Mobilization"

- The following highlighting convention has been followed through this document:
    - Italics for bulleted text are used to indicate descriptive text for the bold marked preceding level bulleted items. For example, "*semantic wrapper*" is a descriptive for "Unified Information Access / **Unified Information Representation**"
    - Vendor names marked in bold (as in, **IBM / SAP / Oracle / IPath**) without any other font styles applied indicate availability of the feature as described by previous level item
    - Vendor names marked in bold and strikethrough (as in, ~~**Oracle**~~) indicate non-availability of feature (as per vendor documents previewed)
    - Vendor names marked in bold and strikethrough + italics (as in, ~~***Oracle***~~) indicate non-availability of feature (as per vendor documents previewed); see

possibility, however, to roadmap support based on general direction of product roadmap

- **IPath** indicates roadmap support for feature

- ***Addressing Information & Communication Experiences ("Enterprise Mobilization") completely – Demands on Technology:***

  - Unified Information Access

    - **Unified Information Representation**

      - *Semantic wrapper!*

      - **IBM / SAP / ~~Oracle~~ / IPath**

    - Heterogeneous data source support

      - Standards-based interfaces

        - WSDL Web Services

          - **IBM / SAP / Oracle / IPath**

        - RDBMS

          - **IBM / SAP / Oracle / IPath**

        - IMAP/POP3

          - **IBM / SAP / Oracle / IPath**

      - Enterprise Software Packages

        - **IBM / SAP / Oracle / IPath**

      - Custom application interfaces

        - **IBM / SAP / Oracle / IPath**

    - Incremental Support for an increasing range of data sources

      - Independent integration modules (connectors or adaptors)

        - **IBM / SAP / Oracle / IPath**

- Business Domain view of back-end
  - Unified information representation that is business end user focused
    - ***IBM / SAP / ~~Oracle~~ / IPath***

- Unified Experience Delivery
  - Multiple channel/device experiences around information and communication services
    - Support for an increasing range of devices
      - PCs
        - **IBM / SAP / Oracle / IPath**
      - Mobile Handsets
        - **IBM / SAP / Oracle / IPath**
      - Television
        - ***~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath***
      - Personal Audio/Video Devices like iPods
        - ***~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath***
      - Alternate devices like GPS Navigator
        - ***~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath***
    - Support for an increasing range of communication channels
      - Voice
        - **IBM / SAP / Oracle / IPath**
      - Email
        - **IBM / SAP / Oracle / IPath**
      - SMS

- **IBM / SAP / Oracle / IPath**

- Blogs
  - ~~***IBM***~~ / ~~***SAP***~~ / **Oracle / IPath**
- IM
  - **IBM / SAP / Oracle / IPath**
- Video Streaming
  - **IBM / ~~SAP~~ / ~~Oracle~~ / IPath**
- Audio Streaming
  - **IBM / ~~SAP~~ / ~~*Oracle*~~ / IPath**

- Multi-modal delivery of experiences
  - Based on interactive requests
    - **IBM / SAP / Oracle / IPath**
  - Based on non-interactive requests
    - ~~**IBM**~~ / ~~**SAP**~~ / ~~**Oracle**~~ **/ IPath**
  - Triggered by Schedulers and/or System-Side Events
    - ~~**IBM**~~ / ~~**SAP**~~ / ~~**Oracle**~~ **/ IPath**
  - Off-line availability of experiences with periodic on-line synchronizations
    - **IBM / SAP / Oracle / IPath**
- Delivery on standards based interfaces
  - As defined by data browsers (PC and Mobile)
    - **IBM / SAP / Oracle / IPath**
  - Or voice browsers
    - **IBM / SAP / Oracle / IPath**
- Delivery specific to devices

- Say, mobile handset-specific applications
  - **IBM / SAP / Oracle / IPath**
- Delivery within the context of third-party applications
  - Embedded access from within, say, Microsoft Outlook
    - **~~IBM~~ / SAP / Oracle / IPath**

- Individual-centric Experiences
  - *For Maximal Availability, Minimal overload*
  - Highly personalizable
    - Availability
      - **IBM / SAP / Oracle / IPath**
    - Detail Exposure
      - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**
    - Experienced
      - access aspects
        - personalized shortcuts (voice commands, SMS keywords..)
          - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**
      - presentation aspects
        - **IBM / ~~SAP~~ / ~~Oracle~~ / IPath**
      - relationship/navigation aspects
        - **~~IBM~~ / *~~SAP~~* / *~~Oracle~~* / IPath**
  - Context-centric Actionability
    - Access to related actions in context
      - Actioning around primary experience

- **~~IBM~~ / *SAP* / *Oracle* / IPath**

- Consistent experience across devices/channels
  - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**

- Seamless interplay of information and communication
  - *Leverage information/communication convergence*
  - Information services in the context of communications
    - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**
  - Communication services in the context of information experiences
    - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**

- Loose coupling for configurable orchestrations
  - Definition of server driven sequenced execution of services
    - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**
  - *Enhanced actioning in contexts*

- Responsive to external events and information captures
  - Integration interfaces for external devices
    - Image scanners
      - **IBM / SAP / Oracle / IPath**
    - Security devices
      - **IBM / SAP / Oracle / IPath**
    - Location devices
      - **IBM / SAP / Oracle / IPath**
  - Integration interfaces for External Applications
    - **~~IBM~~ / SAP / ~~Oracle~~ / IPath**

- Service Experience triggering based on external events
  - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**
- Dynamic context control based on external tracking mechanisms
  - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**

- Support multiple convergence component options
  - Multi-vendor Web Servers
    - **~~IBM~~ / SAP / ~~Oracle~~ / IPath**
  - Multi-vendor Voice Termination Servers
    - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**
  - SMS Gateways
    - **IBM / SAP / Oracle / IPath**
  - Audio/Video Streaming Gateways
    - **IBM / ~~SAP~~ / Oracle / IPath**

- Support dynamic experience management
  - *Wide audience exposure demands continuous change in experience*
  - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**

- End-to-End Manageability
  - *Multiple users, multiple devices, channels can mean multiple life-cycles*
  - Need for simplified life-cycles
    - *Traditional approach to multi- application building means multiple life-cycles. Challenges in managing and evolving versions*
    - Single point from which to manage every experience detail
      - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**

- Abstraction of every aspect of information delivery, source to experience
  - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**
- Complete configurability of every aspect of experience
  - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**
- *Simple enough to visualize such a widely impacting system from creation standpoint*
- *Unified thinking about what is being delivered irrespective of what mode, device or channel of delivery*
- Service thinking all the way to edge
  - **~~IBM~~ / ~~SAP~~ / ~~Oracle~~ / IPath**
- *End-to-end information flow managed*
  - *Increased exposure means increased challenges*
    - *Security*
    - *Differential experiences*
    - *Life-cycle dynamics*
  - *Multiple life-cycle through creation of multiple applications to meet demands of increased exposure best avoided*
- *Every aspect of consumptions captured as information flow happens from source to destination*
  - *La data packets across a network (OSI)*
  - *Need to address*
    - *Manageability*
    - *Usability*
      - *Personalizability*
      - *No overload*

- *Relevance*
- *Completeness*

- *Deliverability*
  - *Rapid time to market*
- *Continuous Customizability*

What is claimed is:

1. An enterprise mobilization system comprising:

EUS which receives user requirement and translates the requirement into a content component **and platform independent** delivery component; and

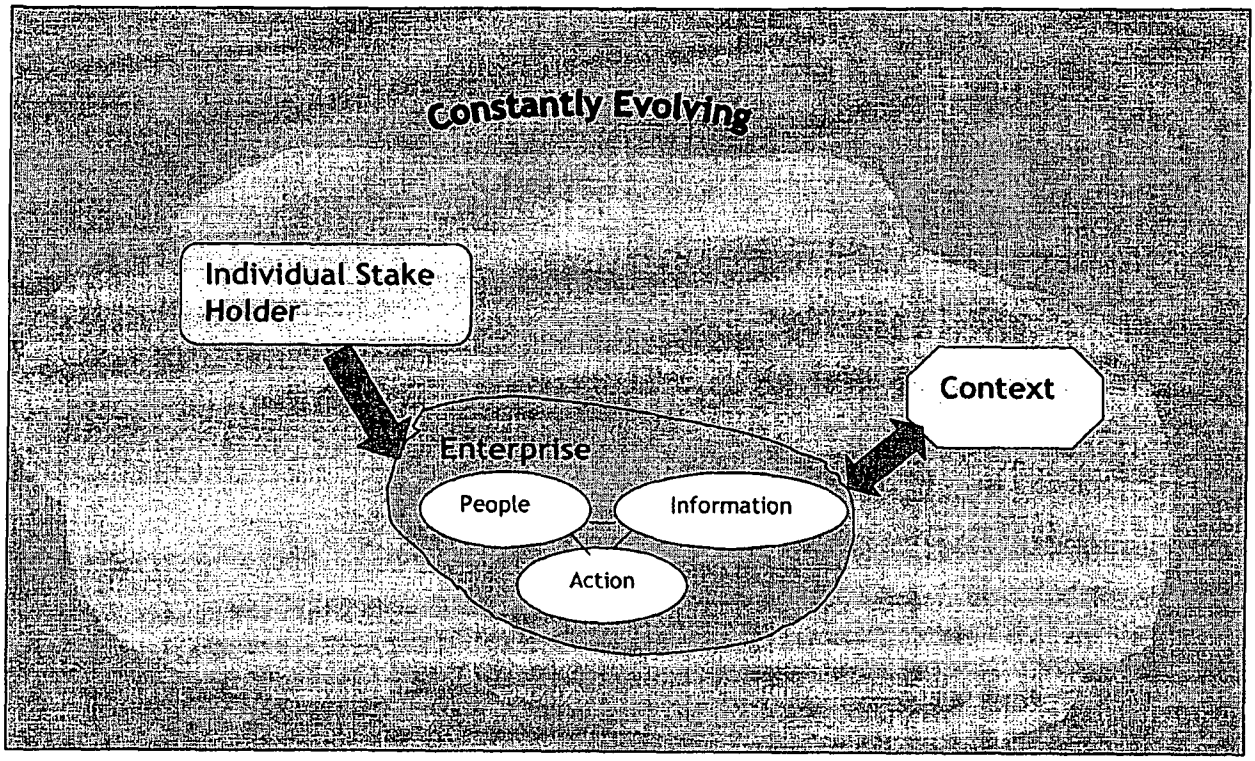
DSIM which receives an **information tree** based on the content component and translates it into requests for data from at least one data source,

wherein the information is contextualized

wherein the system provides an ability to take actions based on the context

wherein all user experiences related to the mobilization are achieved through the concept of end-user services.

FIG. 1


Constantly Evolving
Individual Stake Holder
Context
Enterprise
People
Information
Action

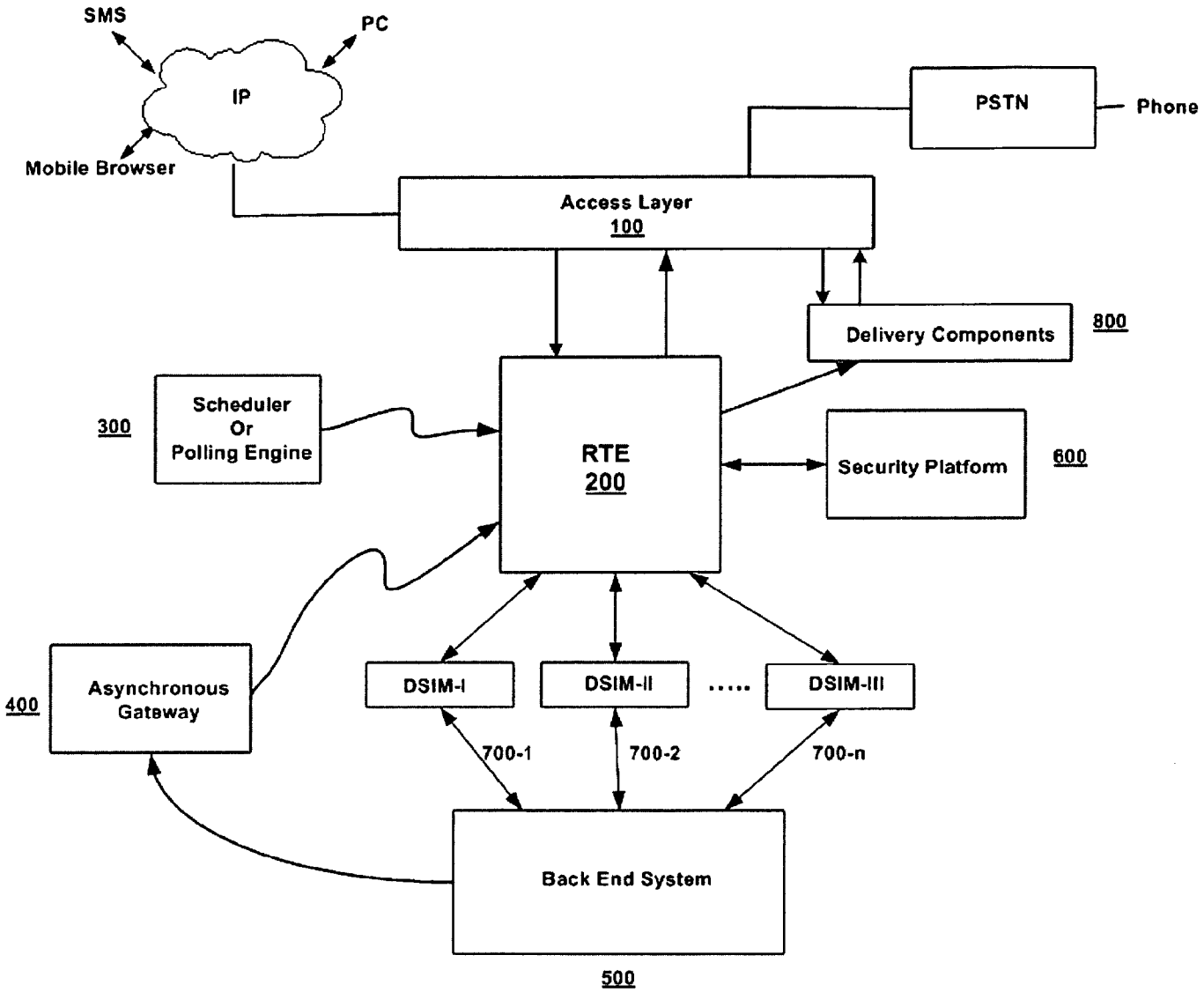

SMS
PC
IP
Mobile Browser
PSTN
Phone
Access Layer
100
Delivery Components
800
300
Scheduler
Or
Polling Engine
RTE
200
Security Platform
600
400
Asynchronous
Gateway
DSIM-I
DSIM-II
.....
DSIM-III
700-1
700-2
700-n
Back End System
500

FIG. 2

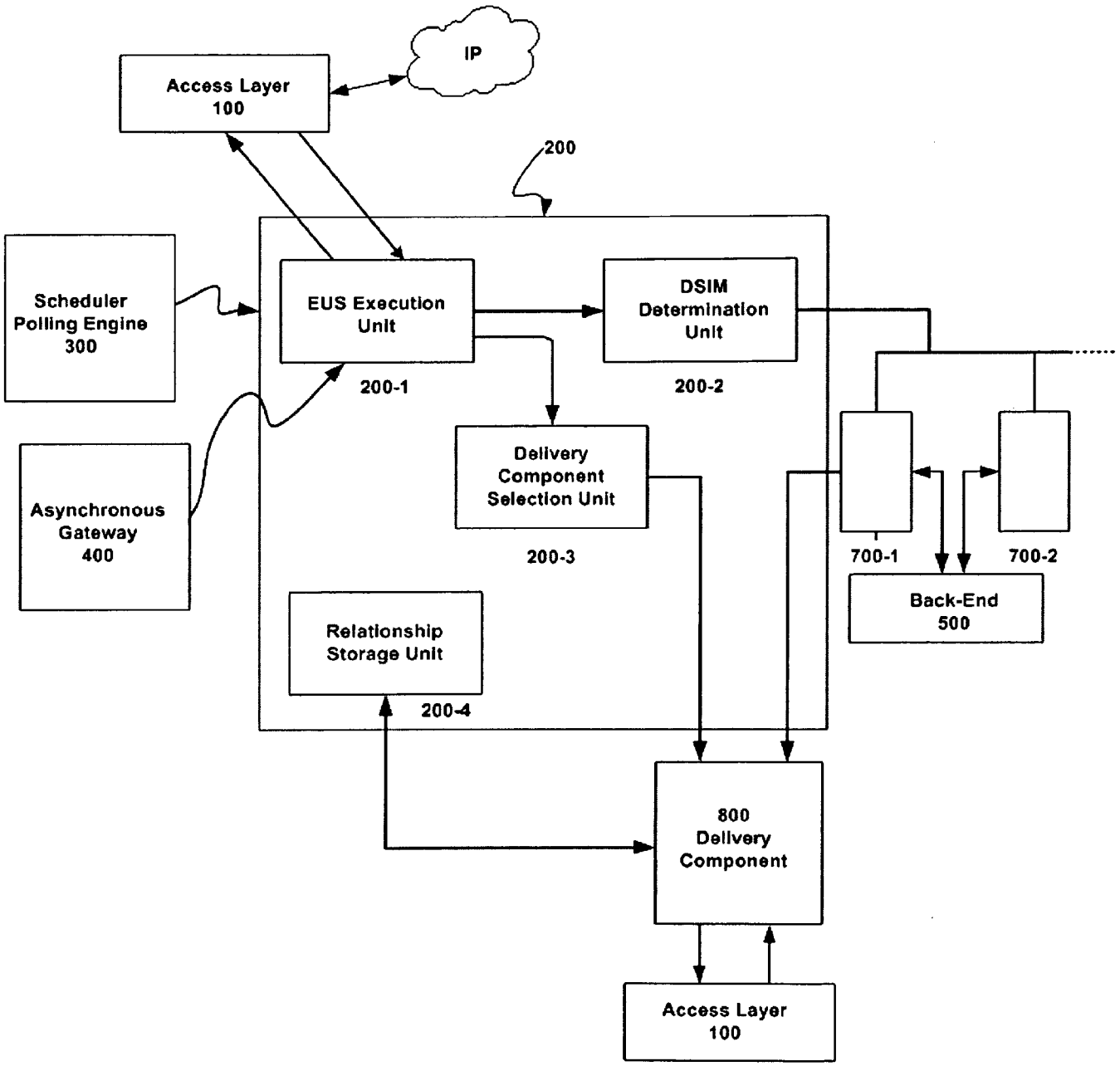

IP
Access Layer
100
200
Scheduler
Polling Engine
300
EUS Execution
Unit
200-1
DSIM
Determination
Unit
200-2
Asynchronous
Gateway
400
Delivery
Component
Selection Unit
200-3
700-1
700-2
Back-End
500
Relationship
Storage Unit
200-4
800
Delivery
Component
Access Layer
100

FIG. 3