XML DOCUMENT MANAGEMENT SERVER (XDMS) pdf

1: Using Simulators and Other Testing Tools

XDMS - XML Document Management Server. An XDMS is responsible for hosting several Application Usages (essentially an Application Usage is an instance of an XML configuration document relative to a particular aspect of a service eg Converged Address Book, Social Presence).

Provisional Patent Application No. The contents of these applications are expressly incorporated herein by reference. These devices tend to be autonomous entities that employ wired or wireless access technologies to communicate information back to a server, or otherwise into a processing network. These devices are typically deployed to devices such as gas or water meters and other such devices, as well as to devices that are remote and cause difficulty in obtaining readings from. The advantages of deploying numerous M2M devices are well known and understood in the field. In the telecommunications field, a number of services are made available to devices through the wireless and wired networks. These services are often not utilized by M2M communications as the devices are often designed to interact with a data structure designed by people not familiar with the functionality of the existing services, and without an understanding of how to modify the existing services to properly make use of them. As a result a large amount of existing functionality is duplicated. This not only increases development lead times, but also often results in inefficient use of limited bandwidth to perform functions that can be delegated to the network. Therefore, it would be desirable to provide a system and method that obviate or mitigate problems associated with allowing the interaction of a plurality of M2M devices that are not under the management or control of a carrier, with carrier infrastructure. In accordance with a first aspect of the present invention, there is provided a method of facilitating communication between an unmanaged machine-to-machine device and a network resource in a managed network. The method comprises the steps of receiving a request for the resource from the unmanaged device; generating an Extensible Markup Language Configuration Access Protocol XCAP request in accordance with the received request; and transmitting the generated request towards an Extensible Markup Language Document Management Server XDMS using an asserted identity determined in accordance with identities of both the resource and the unmanaged device. In an embodiment of the first aspect of the present invention, the step of receiving the request includes receiving a request in a format other than XCAP, and optionally the step of generating includes translating the request from a non-XCAP format to an XCAP format. In another embodiment, the step of transmitting further includes selecting an XDMS in accordance with the resource associated with the received request. In a further embodiment, the step of transmitting includes transmitting the generated request to an aggregation proxy associated with the XDMS. In another embodiment, the method further includes the step of receiving a response from the XDMS. In a further embodiment, the received response is formatted as an XCAP response. In another embodiment the method includes the step of transmitting the received response to the unmanaged device, and can optionally include translating the response from an XCAP compliant format to a format selected in accordance with characteristics of the unmanaged device. In another embodiment, the step of receiving the response includes receiving the response from the XDMS via an aggregation proxy. In accordance with a second aspect of the present invention, there is provided a Machine-to-Machine Service Capability Platform M2M SC for facilitating communication between an unmanaged machine-to-machine M2M device and a network resource in a managed network. The M2M device interface receives messages from and sends messages to the unmanaged M2M device. The proxy identity assertion engine determines an identity associated with the request received in accordance with an identity of the unmanaged device. In an embodiment of the second aspect of the present invention, the XDMS interface includes an Aggregation Proxy interface for transmitting the generated XCAP compliant request to an aggregation proxy when the selected XDMS is in a different network domain. In a further embodiment, the M2M SC further comprises a response handling engine for receiving a response from the selected XDMS over the XDMS interface, and for transmitting a response to the unmanaged device in accordance with the received response. Optionally, the response handling engine receives the response as an XCAP response and transmits the response to the unmanaged device in a non-XCAP format. Other aspects and features of the present

invention will become apparent to those ordinarily skilled in the art upon review of the following description of specific embodiments of the invention in conjunction with the accompanying figures. Reference may be made below to specific elements, numbered in accordance with the attached figures. The discussion below should be taken to be exemplary in nature, and not as limiting of the scope of the present invention. The scope of the present invention is defined in the claims, and should not be considered as limited by the implementation details described below, which as one skilled in the art will appreciate, can be modified by replacing elements with equivalent functional elements. In a carrier grade environment, the use of an XDMS allows nodes trusted by the carrier to access data in a secure and trusted fashion. In an environment where there are a limited number of applications, and a high degree of carrier control over the nodes on which the applications are executed, this is a near ideal architecture. XDMS resources are stored in a manner that allows for a per-application access level control, and there are a limited number of nodes that can access the XDMS. As a shift to M2M devices has occurred, it has become clear that the current architecture has problems when there are a large number of nodes that are running applications. In an M2M environment, each M2M device is capable of running at least one application. To provide the carrier with security, it is preferable that the M2M devices and their respective applications not be able to directly access the XDMS. Further problems are raised when an application access the network through a first carrier and needs access to the XDMS services hosted by a second carrier. However, M2M devices are manufactured by a variety of different entities, and not all of the devices will make use of IMS access procedures. Instead, many existing M2M devices are simply provided with an IP stack and a set of commands and instructions that they are responsive to. This creates a mismatch both in the ability of the devices to connect to the network, and the ability of the network to allow connections from the many devices. In the following discussion, an architecture for an enhanced XDMS framework is discussed, which makes use of a new directory structure for access rights. Additionally, a mechanism for a M2M Service Capability Platform to act as a proxy between the devices and the XDMS is introduced to allow the M2M devices to access the XDMS resources without breaking the security and reliability of the carrier networks Given that an M2M service provider has several independent clients legal entities each of which may require privacy, or even complete privacy, of their data from other clients, the following directory scheme shown below will preferably be applied for this exemplary embodiment. In this scheme, several principles enumerated here can be observed. Each legal entity will preferably have a different tree under the XCAP root. This can allow for complete privacy and simplification of the entire XDMS operation. Each legal entity tree will preferably have one common application usage Entity Access applicable to the all application usages under the legal entity tree. Application usages that are common to all legal entities will preferably be defined immediately under the XCAP-root. AUIDn in the figure below. The same preferably applies to the oma. Eleven Application Usages are identified herein for managing M2M resource. This list should not be considered to be exhaustive, but instead viewed as being sufficient for explanatory purposes. Access Right Application Usage defined under the legal entry tree. Although Access rights resources apply to all legal entities, permissions are specific to every legal entity and as such it is defined under the legal entity tree. This Application Usage handles the management of M2M access rights resources created under the legal entity. Group Application Usage defined under the legal entity tree. The Group resource is specific to a legal entity and as such it is contained within the legal entity tree. This Application Usage handles the management of M2M group resources created under the legal entity. Container Application Usage defined under the legal entity tree. The Container resource is specific to a legal entity and as such it is contained within the legal entity tree. This Application Usage handles the management of M2M container resources created under the legal entity. This Application Usage can address the management of M2M applications created under the legal entity. This Application Usage handles the management of M2M announced applications resources created under the legal entity. Container Collection Application Usage defined under the legal entity tree. This Application Usage handles the management of M2M container collection resources, including announced containers, created under the legal entity. Group Collection Application Usage defined under the legal entity tree. This Application Usage handles the management of M2M group collection resources, including announced groups, created under the legal entity. Access Rights Collection Application Usage defined under

the legal entity tree. This Application Usage handles the management of M2M access rights collection resources, including announced access rights, created under the legal entity. Application Collection Application Usage defined under the legal entity tree. This Application Usage handles the management of M2M application collection resources, including announced applications, created under the legal entity. M2M Legal Entity Application usage defined under the legal entity tree. This Application Usage handles the management of all M2M resources created under the legal tree. Even though the directory structure in XDMS is not completely identical to the resource structure, the M2M Service Capability AS is able to recreate an identical match for the resource structure to be able to respond to requests over the mIa and mId interfaces. The Application Usages for the various resources will elaborate these cases in more details. The above enumerated usage cases will now be discussed in more detail. These details should be understood to be explanatory in nature, and should not be considered as limiting of the scope of the present invention. The M2M Legal Entity Application Usage is an application that controls access to other resources under the legal entity tree and also stores information pertinent to the legal entity tree. The XDMS server will preferably authorize requests for all operations on all XDM documents located under the entire Legal Entity tree using the accessPermission document located under the global tree for that purpose. This Application Usage defines seven global documents. The well-known name of the document is attributes. The well-known name of the document is accessPermission. The third well-known name of the document is applications. There preferably is only a single instance for all these documents. The fourth well-known name of the document is groups. The fifth well-known name of the document is containers. The sixth well-known name of the document is accessRights. The seventh well-known name of the document is subscriptions. The binding will preferably be achieved by referencing the XCAP document matching the unique identity for the selected Access Right resource. In option 1, illustrated in FIG. The index document encompasses all the information in option 2 below except accessPermission which is identical in both options In option two shown in FIG. For either option there is preferably only a single instance for every document The XDMS server will preferably authorize requests for all operations on XDM documents located under the XUI tree using the accessPermission document located under the same XUI tree for that purpose. The XDMS will also preferably ensure that members allocated to a group are uniquely identified. The application unique identifier for the case illustrated in FIG. In addition to conforming to the XML schema, the XDMS will preferably enforce the syntax allocated to member ids and will preferably ensure the uniqueness of member ids in the group. There are two naming convention options for storing Group XDM documents. In option 1, as illustrated in FIG. The index document encompasses all the information in option 2 below except the accessPermission which is identical in both options In option two shown in FIG. For either option there is preferably only a single instance for every document. The index document encompasses all the information in option 2 below except the accessPermission which is identical in both options In option two shown in the FIG. The XDMS server will preferably authorize requests for all operations on XDM documents located under the global tree using the accessPermission document located under the global tree for that purpose.

XML DOCUMENT MANAGEMENT SERVER (XDMS) pdf

2: Virtual Office - Document Mangament

The JBoss Communications Platform XML Document Management Server (XDMS) is part of the JBoss Communications Platform SIP Presence www.enganchecubano.com is the first free and open source implementation of an XML Document Management Server as defined in the Open Mobile Alliance (OMA) XML Document Management v specification.

As groups of M2M applications are formed in a network, the group information comprising identities of the groups and identities of M2M applications belonging to each group is relayed to the M2M-AS. The later further sends such information to the XDMS server for storage. Requests for group membership related to M2M applications can then be handled. When a request for group membership related to an application is received at an M2M gateway or at the M2M-AS, the request is further sent to the XMDS server, which replies back to the requestor with identities of the group s that comprise the given M2M application, thus enabling, for example, the requestor to communicate not only with the application, but with entire groups comprising the application. In the third and fourth generations of wireless communications, data is added to the digital voice communications and allows the wireless network users to communicate not only voice signals but also to have data access to the Internet and to a wide variety of services associated therewith. As of today, there are already more wireless subscribers than fixed line subscribers. The next wave of growth for wireless telecommunications over the next 10 years will most probably not come from adding regular users to the network, as it has been the case in the past. Therefore, it is the so-called machine-to-machine M2M type of communications that will ensure the growth both in the data traffic and in the revenues for the operators of wireless networks, while providing services as never seen before for the users. Houses and home appliances, sensors, and other devices will therefore be wirelessly connected, accessible, and controlled via local networks or the Internet. More particularly, M2M communications also means that an application on a given device can communicate with another application on another device, or on a server, without necessarily involving a human being. Examples of connected devices include a pool of electrical counters of a local electrical company that can include millions of electrical counters connected over a wireless interface such as for example a short range wireless interface to the local electricity company gateway, which gathers data from all the counters and relays it for example via a 3G wireless connection to the financial database of the electrical company, or a group of connected devices including a series of meteorological sensors or probes that register the wind and the temperature in various locations within a province or a country, and relay the data gathered to the gateway responsible for assembling all this metrological data before providing it to a processing center of the local meteorological company. It is easily anticipated that many of the networks of connected devices, also referred herein simply as devices, will be significant in size, and will possibly include millions of devices e. Therefore, groups of devices will need to be constituted in order to render the communications more efficient. This, in turn, will engender new issues associated with the management of such large groups of devices. Moreover, in some instances the groups of devices and applications running thereon may be visible to the outside Internet world, e. Finally, yet another issue associated with the management of large groups of devices is how an external party such as for example a subscriber, a network administrator, or another device of a different network can access and communicate with devices that are member of a certain group. In such a circumstance, it would be useful for the external party to easily contact groups of devices and applications, in order to be able to communicate with all those devices at the same time instead of, for example, having to communicate the same information e. However, with the current implementations, where groups are dynamically created, modified, and deleted in various gateways of a wireless network in an ad-hoc fashion it is not possible to know what groups a device is member of since, first, not all the groups are visible to the outside Internet world and second, because there is no standardized way to obtain such information. Reference is now made to FIG. For this purpose, both visible devices and gateways can register with the M2M AS. Applications resident on visible devices register with the device on which they are resident, and the application registration can be further announced to the M2M AS The same applies to applications that are

resident on gateways and , i. On the other hand, applications resident on non-visible devices connected to the gateways or register with the gateways and the application registration can be further announced to the M2M AS A home subscriber server HSS may be present in the core network B for providing access to a subscriber database which allows authorizing access to M2M services in case the access network credentials are reused for M2M access as well. With reference to the exemplary scenario shown in FIG. On the other side, the applications running on the visible devices register locally with their respective device , and can chose to announce their registration to the M2M AS when the device registers with the M2M AS These later applications are fully visible and can be contacted directly by external parties by contacting the device. Conclusively, there are both great numbers and various types of M2M application and devices in many networks, and therefore contacting each device at a time can be proven inefficient, while contacting groups of devices requires prior knowledge of each group membership and identification in an environment where such information cannot be easily obtained or managed, or is simply inexistent. Accordingly, it should be readily appreciated that in order to overcome the deficiencies and shortcomings of the existing solutions, it would be advantageous to have a method and system for efficiently enabling communications with groups of devices. As groups of M2M applications are formed in an M2M network, the group information comprising identities of the groups and identities of M2M applications belonging to each group is relayed to the M2M-AS. The later further sends such information to the XDMS server for more persistent storage. IN this manner, requests for group membership related to M2M applications can be efficiently and rapidly handled. When a request for group membership related to an M2M application is received at an M2M gateway or at the M2M-AS, the request is further sent to the XMDS server, which replies back to the requestor with identities of the group s that comprise the given M2M application, thus enabling, for example, the requestor to communicate not only with the application, but with entire groups comprising the application. In another embodiment, the present invention provides a M2M-AS for use in M2M group management, the M2M-AS comprising a communications interface communicating with a network, and a processor operationally connected to the communications interface. In another embodiment, the present invention provides a method for the management of information related to groups of M2M applications, the method comprising the steps of receiving at an XDMS server from an M2M-AS an identity of a group of M2M applications and a plurality of identities of M2M applications that belong to the group of M2M applications, and storing in a data repository of the XDMS the identity of the group and the identities of the plurality of M2M applications that belong to the group. In another embodiment, the present invention provides an XDMS server for use in M2M group management, the XDMS server comprising a data repository storing information related to groups of M2M applications, and a communications interface communicating with a network. The XDMS server further comprises a processor operationally connected to the communications interface and to the data repository, and an instructions repository storing instructions that when executed by the processor cause the later, when the communication interface receives from an M2M-AS an identity of a group of applications and a plurality of identities of M2M applications that belong to the group, to store on the data repository the identity of the group and the identities of the plurality of M2M applications that belong to the group. However, it should be understood that this class of embodiments provides only a few examples of the many advantageous uses of the innovative teachings of the invention. In general, statements made in the specification of the present application do not necessarily limit any of the various claimed aspects of the present invention. Moreover, some statements may apply to some inventive features but not to others. In the drawings, like or similar elements are designated with identical reference numerals throughout the several views. In one preferred embodiment, the invention provides for an easy manner of managing a group of devices and their M2M applications, and for handling an external request about which group a certain device, or application running thereon is part of. When a request for group membership is received from any requestor, e. Then, for example, a proper answer can be sent back to the requestor including the group identifier s of the group s the device application is member of, in order to enable the requestor to efficiently communicate with the group of devices, e. In the context of the present description it is to be understood that an M2M application refers to an application e. A group of applications refers to a group having an identifier also called herein identity , the

group comprising a plurality of applications, which may, for example, share common characteristics. References is now made to FIG. The exemplary network comprises a first device running first M2M application also called D1A1 second device running a second M2M application also called D2A2. The applications and are part of a local network of devices managed by the M2M gateway An M2M-AS is responsible for gathering information about a plurality of M2M gateways and their associated local networks, including the M2M gateway , and an XDMS server further functions in the network to store and manage resources related to all devices and applications of the wireless telecommunication network , including the applications and tagged as D1A1 and D1A2 respectively since application 1 runs on device 1 while application 2 runs on device 2. It is understood that each application and has a unique identifier, which may be for example in the form of a URL that may identify either the application itself in a unique manner for the entire network, or to uniquely identify the application in relation with the device on which it is running. The exemplary scenario of FIG. The XDMS serve may store such information for many groups and many applications. Thereafter, for example, when a request for a group membership of a given application is received, the M2M-AS or the gateway interrogates the XDMS server for obtaining identities of groups the given application is part of, obtains from the XDMS server the identities of the groups, and respond to the request with the identities of the one or more groups the M2M applications is part of. In particular, in action , every known application of the network is created a record or other data structure saving its application identifier, and optionally any known group the application belongs to. Further, in action , a group of M2M applications is created. This may be achieved, for example, by the application that creates the group of applications in the gateway , by sending a create group message, along with a proposed group identifier, and a series of applications identifiers D1A1 through D4A4 of applications that are to be part of the new group wherein e. In action , the M2M gateway stores the group information including the group identifier and the applications identifiers, and in action sends back a response to the application for confirming the group creation in the gateway In a variant of the preferred embodiment of the invention, other methods for creating a group of applications may further be contemplated, including, for example, a network administrator who creates a group directly in the gateway , via an administrative user interface to the gateway Further in action , the gateway may determine if it is registered with an M2M-AS and in the affirmative, it sends a group announcement message including the group identifier in order to announce to the M2M-AS of the creation of the new group. Otherwise, if in action it is concluded that the gateway is not yet registered with the M2M-AS but would still like to make a group announcement to the M2M-AS for the purpose of communicating the group creation to the XDMS, then the gateway sends a group announcement message including the group identifier and an indication that the gateway is not yet registered with the M2M-AS Optionally, action may trigger a sub-registration procedure , which allows the gateway to register with the M2M-AS In action , having completed the registration process and having been provided with the information about the creation of the new group, the M2M-AS responds back to the gateway to confirm the receipt of the information about the group. In action , the M2M-AS subscribes for receiving updates about the group , and in action , responsive to the subscription, it is provided back with the identities of the applications D1A1 through D4A4 that are member of the new group. In one embodiment, the M2M-AS will continuously refresh the subscription with the gateway for the group as long as the group exists. Other manners can also be contemplated for providing the M2M-AS with information about a group of applications. In a non-limitative example of variant of the preferred embodiment of the invention, messages through may be replaced with other types of messages. In a first of such an example, the Gateway may use one single message for sending out to the M2M AS the group information, including the group identity and the identities of the applications D1A1 to D4A4. Such message may comprise a modified group announcement message that includes not only the group identity but also the group membership information, i. In action , once the M2M-AS is provisioned with the group information, the M2M-AS communicates with the server XDMS in order to provide the later with information about the new group, including identity of the group of M2M applications and a plurality of identities of M2M applications that belong to the group, for allowing the XDMS server to store this information in its data repository. In a first variant of the preferred embodiment of the invention, action may include a transmission from the

M2M-AS to the XDMS server of the group identifier and of the series of application identifiers of applications D1A1 through D4A4 that are part of the new group, and the storage by the XDMS server of the group identifier under a tree or record of each application, so that each application that is part of the group is connected to the group identifier. In another variant of the preferred embodiment of the invention, action may include only the addition of the group identifier information in the record of each application identifier that may have been already present in the XDMS server as created in action In both variants, action results in a data structure to be created or updated in the XDMS , where application identifiers of applications that are part of the new group are linked to their corresponding group identifier, such as for example the application D1A1 being linked to the group identifier One of the purposes of sending group information to the XDMS is, for example, for the XDMS to store such information and be able to answer requests for group membership received from various requestors. Thus, the XDMS server receives a request for a group membership for one of the plurality of M2M application, the request comprising an identity of the one of the plurality of M2M applications. The XDMS server determines the identity of at least one group of applications comprising the one of the plurality of M2M applications based on the identity of the one of the plurality of M2M applications, and responds to the request with the identity of the at least one group of applications that comprises the one of the plurality of M2M applications. In more particular details, in action , it is assumed that a requestor e. For this purpose, the requestor issues a request for the group membership of the application D1A1 , which is received at the M2M gateway The request may comprise the identity of the application D1A1 and an interrogation of the group membership of the application, i. The request is forwarded from the M2M gateway to the XDMS server possibly and optionally via the M2M-AS and in action , the XDMS server determines the group membership of the application D1A1 by interrogating its local data structure that stores the group membership information. In so doing, in action , the XDMS determines that the application D1A1 is member of the group identified by the group identifier , and in action it responds back to the M2M gateway with the group identifier so that M2M gateway can further answer back to the request If application is part of multiple groups, then the identity of each such group is returned. From there, the M2M-AS can further reply back to the requestor with the group membership information. While the scenario illustrated in FIG. For example, gateways for M2M applications may need to form various groups of M2M applications executing on the devices connected to the gateway for the purpose of software update or collective communication with groups of applications to report a reading such as for example in the context of smart meters or water meters , or for configuration management purposes. This allows network applications, servers, and other parties connected to M2M-AS-es to query the M2M-AS about group membership so that it can target the correct group for the intended application resident on a device connected to a gateway. In one embodiment, for example, if an application is not a member of any group and it must be contacted for a given purpose by an M2M application, then the target application must either join the correct group so that the M2M application can just address the group or the M2M application will be forced to address the group and the applications that are not members in the group. As it can be contemplated from the exemplary preferred embodiment shown in FIG. Such group information may be received at the XDMS server from a plurality of M2M gateways alike the M2M gateway , or a plurality of M2M devices not connected to gateways so that the XDMS server ends up in storing and making accessible group information related to many, if not all, of the M2M applications of the network The communications interface is operationally connected to a processor , which in turn is operationally connected via proper internal interfaces to an instructions repository The later includes instructions, that when transmitted to and executed by the processor cause the later to perform the actions shown in FIG. The instructions repository may comprise further instructions, that when executed by the processor cause the later, when receiving a request for a group membership for one of the plurality of M2M applications, to interrogate, via the communications interface, the XDMS server for obtaining one or more identities of groups the one of the plurality of M2M applications is part of and to obtain from the XDMS server the one or more identities of groups the one of the plurality of M2M applications is part of, and further to respond to the request with the one or more group identities. The XDMS server is provided with an M2M applications management database that stores M2M application information comprising group information including the group identifiers and all

the M2M applications that are members of each M2M group. The XDMS server further includes a processor and an instruction repository storing instructions that when executed by the processor cause the later to perform, alone or in combination with the communication interface , the actions related to the XDMS server as shown in FIG. For example, when such instructions are executed by the processor , they cause the processor to perform action or at least the part of action related to the XDMS server for creating a record for known M2M applications of the network , the action where M2M application records are updated with group information, and the action where the group membership of a given M2M application is retrieved from the database , or the creation of the responses and and their transmission via communications interface In particular, the instructions repository may store at least instructions that when executed by the processor cause the later, when the communication interface receives from an M2M application server M2M-AS an identity of a group of applications and a plurality of identities of M2M applications that belong to the group, to store on the data repository the identity of the group and the identities of the plurality of M2M applications that belong to the group. The instructions repository may further store instructions, that when executed by the processor cause the later, when the communications interface receives a request for a group membership for one of the plurality of M2M applications, to retrieve from the data repository the identity of at least one group of applications comprising the one of the plurality of M2M applications using the identity of the one of the plurality of M2M applications, and to respond via the communications interface to the request with the identity of the at least one group of applications comprising the one of the plurality of M2M applications As it can be contemplated, the present invention provides various embodiments that allow an external party to easily request and obtain group membership information of various M2M applications of a network. This provides an extremely valuable capability to requestors such as for example the network administrators, other M2M applications, or other subscribers, for determining the group membership of a given application and then using that information for sending group communications, such as for example one-to-many messages or software updates. Provided with the group information, parties can use the group identifier in order to then communicate with a plurality of applications at a time, often times with a significant number of applications at the same time, thus reducing the signaling associated with, for example, the update of some applications.

XML DOCUMENT MANAGEMENT SERVER (XDMS) pdf

3: USA1 - Xdms for resource management in m2m - Google Patents

The Azetti Network's XML Document Management Server (XDMS) is a key IMS (IP Multimedia Subsystem) element that allows operators to maintain their subscribers' profile information in a central, secure and easy to access repository.

It also enables you to extend and configure a Converged Application Server domain with Diameter capabilities. Figure shows the general components of the graphical user interface for configuring the SCE simulators: Figure Service Creation Environment Simulator Interface As shown, the simulator list provides access to the various configuration page for each type of simulator. After setting the configuration, you can deploy the simulator to the selected target servers by clicking the start icon. The stop icon terminates the simulator process, and also removes the simulator as a deployed application from the target server. Any target server you have added to the SCE project appears in the target server list. You can click the refresh icon to update the list of servers that appear in the list. Refreshing the server list adds newly created servers and removes deleted servers from the list. The following sections provide more information on how to use each type of simulator, as well as the SIPp and Diameter domain configuration tools. The type of information hosted by the XDMS presence authorization policies, contact and group lists, and presence status. The XDMS simulator is suitable for testing purposes only. It supports only those document management capabilities applicable to call functions that are available through the SFT APIs, including call barring, call forwarding, and OIP. Since it lacks data persistence, when you shut down the XDMS simulator, its hosted data is lost. It can be deployed to any server you have added as a target server environment in the SCE. Using an XDMS system involves both loading and querying information. The following steps describe how to deploy and start the simulator. Before starting, configure the target server environment to which you want to deploy the simulator in the SCE. Notice that the target servers appear at the bottom of the pane. To have the XDMS simulator enforce digest authentication requirements, select the Enable digest filter. Select the check box of the server to which you want to deploy the XDMS software. You should deploy the simulator to a Converged Application Server administration server only; it does not operate on engine tier servers. XDMS simulators on different servers cannot share information. For most development testing scenarios, deploying the XDMS simulator to a single server is sufficient. Click the start icon. For example, given an XCAP client instantiated on the communication service, the following code fragment creates the connection to the XDMS simulator on the local host: Because the XDMS simulator stores its contents in memory only, stopping the simulator application clears its contents and returns it to its initial state. Configuring the Media Server Driver Converged applications rely on media servers to enable rich media services, such as conferencing, audio prompting, and speech detection. The SCE provides a JSR adapter that allows you to test interactions between converged applications and an external media server. The SCE includes a media server simulator that you can use to test applications that rely on a media server. To configure the media server connectivity in the SCE, follow these steps: In the simulators list, choose Media Server Simulator. The media server configuration settings appear in the right pane. Configure the following settings: Enter the host name or IP address of the media server in your environment. Enter the port number on which the media server listens for client requests. By default, this is Select the Converged Application Server on which to deploy the media server driver. The page shows the servers that have been configured as target servers for the project. Choose the server on which to deploy the media server driver. Click the start icon to deploy and start the simulator. The media server starts. You can now test the applications that rely on the media server. When finished, click the stop icon to terminate the media server process and remove the deployment from the Converged Application Server. You can now test your converged applications that use the media server simulator. Configuring the Diameter Simulator Settings The Diameter charging and HSS simulators enable you to test operations related to charging and authentication in your converged applications. The Diameter and HSS simulators can operate as standalone servers only. To configure connectivity to the Diameter server from the SCE: In the simulators list, choose one of the following nodes: Diameter Ro Simulator to have the simulator perform online charging functions. Diameter Rf Simulator if the simulator performs offline charging server functions. The configurations settings

for the simulator integration appear in the right pane. Enter the realm name for which the Diameter node simulator has responsibility. Enter the identity of the Diameter simulator. Enter the listen address on which the Diameter simulator listens for Diameter traffic. Enter the listen port on which this node listens for Diameter traffic with the simulator. Select to have debug output printed to the SCE log screen. Select to have log output printed to the SCE log screen. Click the start icon to start the simulator. You can now test run the applications that rely on charging functions of the simulator. When finished, click the stop icon to terminate the simulator process. You can now test the Diameter interactions of your converged applications with the Diameter simulator. Extending Domains with Diameter Capabilities Diameter-enabled converged applications can run only on domains that have been extended with the Diameter domain template. The template provides the container framework for enabling Diameter capabilities. The interface also enables you to specify the initial configuration for the domain from the SCE. The SCE can extend domains on a local server only; it cannot extend domains on remote servers. The Diameter domain extension configuration page has settings that populate the diameter. The configuration page has several types of settings, including: Application settings apply to Diameter applications that run on the node. Peer configuration settings define the other Diameter nodes with which this node operates. You can define peer connection information for each Diameter node. Alternatively, you can use the allow-dynamic-peers functionality in combination with TLS transport to allow peers to be recognized automatically. Routes configuration settings define realm-based routes that the node can use when resolving messages. The target servers for the domains to be extended. Any target server configured in the SCE appears in the list. To extend a domain with Diameter capabilities from the SCE, follow these steps: The Diameter domain extension configuration page appears. Click the Add button next to the Diameter application settings list. In the Diameter Application dialog, specify the following settings: Enter a name for the application configuration. Enter the class name of the application to deploy to this node, from the following options: RoApplication for the Diameter Ro online charging application com. RfApplication for the Diameter Rf offline charging application. The Diameter application ID for this application. If one of the predefined Diameter classes is selected, this field is populated automatically. Enter optional parameters to pass to the application upon startup. For example, the Rf application accepts the parameters cdf. Click Add to save the settings. Configure peer nodes by clicking the Add button next to the peer node configuration list. In the Diameter Peer dialog, specify the peer Diameter node using the following settings: Select this option to import Charging or HSS simulator parameters from the simulator view. Enter the listen port number of the peer node. Select this check box to specify that the peer supports the Diameter Tw watchdog timer interval. Optionally, configure settings the Converged Application Server container will use to resolve routes to the peer nodes by clicking the Add button next to the realm-based routes configuration list. In the Diameter Route dialog, specify the following settings: A unique, identifying name for this route configuration. The peer for which this route applies. The peer nodes you have configured appear as menu options for this item. The target Diameter realm associated with this route. Select an action that this node performs when using the configured route.

XML DOCUMENT MANAGEMENT SERVER (XDMS) pdf

4: sp_xml_preparedocument (Transact-SQL) | Microsoft Docs

The JBCP XML Document Management Server (XDM Server) is part of the JBCP SIP Presence Service; it is the first open source implementation of an XML Document Management Server as defined in the Open Mobile Alliance (OMA) XML Document Management v specification.

While the RCS r4. On top of CPM architecture, the RCS has specified additional value added service features like Contents Sharing, Location Information sharing, Social Presence Information sharing reflecting recent market needs that have been proven by many social network services. The IM Server breaks down into three functional components, i. The Participating Function provides User Network Interface UNI towards the UE and the Controlling Function takes the role of conference focus, which is a centralized function that controls signaling procedures for a group communication service. NOTE Many interfaces and representations have been omitted and tweaked in this diagram for simplicity. CPM Architecture Fig 2. The CPM users do not have to select any specific type of messaging service, rather they just do communicate with a contact. Once it is determined, the selected respective Interworking Function e. The Conversation History can also be synchronized across multiple devices of a CPM user thereby the CPM user can be provided with the consistent user experience using any device. The authentication of the RCS enabled device is subject to the security mechanism of underlying network in principle. Each application server may need to support its own authorization mechanism unless the underlying network provides the proper functionalities. This back-end integration is not a scope of RCS specification, rather it is going to be an implementation issue. The Social Presence Information is a sharing object among RCS users which includes portrait icon, favorite link, free text, availability, willingness and geolocation information. Operators may further define their own items to provide a differentiated RCS service. If there is no xml document stored, which is typical when it is the first time access to RCS service, the RCS client shall create new xml documents i. All the RCS messages targeting to multiple recipients i. The video sharing session is available only when there is an existing voice session and if the voice session closes, the video sharing session shall also be closed along with the voice session. If the voice session is affected somehow by e. In the meantime, from RCS r4. That is, the RCS user can establish video sharing session regardless of whether there is an ongoing voice session or not. The Service Capability Discovery can occur periodically based on polling time provisioned by the Provisioning Server or it can also take place per user request. In case the recipient i. The operators should determine what standards to be adopted and what standards can be omitted or replaced. Some traffic optimization methodologies has been already suggested in the GSMA RCS specifications though, operators may need to find more ways reduce the possible traffic overloads. The architectural and functional simplification and traffic optimization are keys to provide robust and stable RCS service and it is only the start of providing qualified RCS service that might be competitive to existing OTT services. Service and Client specification", v1. Service and Client specification", v4.

XML DOCUMENT MANAGEMENT SERVER (XDMS) pdf

5: XML Editor (SQL Server Management Studio) | Microsoft Docs

The Mobicents XML Document Management Server (XDMS) is part of the Mobicents SIP Presence www.enganchecubano.com is the first free and open source implementation of an XML Document Management Server as defined in the Open Mobile Alliance (OMA) XML Document Management v specification.

This functional element of next-generation IP communication networks is responsible for handling the management of user XML documents stored on the network side, such as presence authorization rules, contact and group lists also known as resource lists , static presence information, and much more. The XDM Server comprises the following functional elements: The data source also handles subscriptions to updates on specific documents, or complete XCAP application usages. Authentication Proxy The authentication proxy is responsible for authentication of the user related with each XCAP request handled. Its functions include the authentication and authorization of a subscription, attachment to update events on specific documents or collections, and the sending of notifications when documents change. Configuring the XDM Server 3. It is possible to change the host, the port and the last path segment: Uncomment and set custom servlet name again to the desired last path segment in the XCAP root. Also note that for the Integrated Server the path segment mobicents-xdms is mobicents-sip-presence. The configuration changes through JMX are not persistent. Note that for the Integrated Server the path segment mobicents-xdms is mobicents-sip-presence. The configuratian changes through JMX are not persistent. XDM Server User Profile Provisioning The XCAP interface is public, used by users to manage their information such as buddy lists, presence authorization rules, etc, and needs to enforce user authentication. To do this, the server relies on the User Profile Enabler managed data, such as the users passwords, and this information must be provisioned. This can be done in two ways, both requiring the server to be running: This can also be disabled through configuration of the XCAP interface. Xcap Diff Super Users It is possible to define a set of super users, through the configuration of the XCAP interface, which are authorized to subscribe any document or collection. Any other user may only subscribe documents in its own user directory, for each XCAP app usage deployed. By default there are no super users configured. In order for an application to use those resources, application specific conventions must be specified. Those conventions include the XML schema that defines the structure and constraints of the data, well-known URIs to bootstrap access to the data, and so on. All of those application specific conventions are defined by the application usage. Data Semantics Semantic definition on documents content, used by applications filling data, not validated by servers. Naming Conventions What is the document name for each user? Are there global documents under a specific name? XCAP Clients usually forget to follow these! Resource Interdependencies One request may update other documents as well, e. Authorization Policies What each user can read or write?

XML DOCUMENT MANAGEMENT SERVER (XDMS) pdf

6: Using the XCAP Interfaces

Login to the WebSphere Application Server Admin Console. From the main menu select Applications > WebSphere enterprise applications. Click the XDMSAggregationProxy link.

Describes a ruleset and no-reply timer for communication diversion. CommunicationWaiting Describes a ruleset for the communication waiting service. IncomingCommunicationBarring Describes a ruleset for barring of incoming communication. OriginatingIdentityPresentation Describes a ruleset for barring of originating requests. OriginatingIdentityPresentationRestriction Describes a ruleset for restriction of originating identity presentation. OutgoingCommunicationBarring Describes a ruleset for barring of outgoing communication. TerminatingIdentityPresentation Describes a ruleset for terminating identity presentation. TerminatingIdentityPresentationRestriction Describes a ruleset for restriction of terminating identity presentation. Example disables the Communication Waiting service. In this example the simServs. The object is assigned to the cw variable. Communication Waiting is set to false, disabling the service. Originating Identification Presentation is enabled. You can perform get, fetch, and delete operations on XML elements containing the specified attributes. Attributable interface identifies an element as having attributes. If the predicate is TRUE, the element is selected. The result of the predicate is only valid if the result is unique, and the uniqueness must be enforced. Element selection without predicates returns a list. The code shown in Example creates a Communication Diversion rule using the rule66 identifier, which it stores in the r1 variable. The r1 variable is then used to call the setByAttrName method, to set the attribute using the rule66 identifier. An XCAP server must not allow any modification that breaks any data constraint, and an XCAP client must not make any modification that will lead to issues with the application-defined schema. While data validation sent and received using XCAP is optional, it is strongly recommended that you use data validation to ensure that the resulting XCAP documents retain their integrity. Validation is only performed using related schemas. When validating data, the entire XML document is validated even if only a single element is being fetched or updated. The code below illustrates the use of an exception which, when validation fails, prevents the creation or modification of the XML file in the XDMS. XCAP Authentication and Authorization Authentication consists of determining whether a user wishing to perform certain operations is who he or she claims to be. Authorization consists of determining whether an authenticated user is allowed to perform the requested operation. User authorization is performed by the relevant XDMS the processes requests coming from authenticated users. Digest authentication uses a simple challenge-response mechanism to verify the identity of a user over SIP or HTTP, and requires a user name and password. Clients must implement digest authentication, assuring interoperability with servers that challenge the client.

7: TigerLogic XDMS vs. Versant ODBMS Comparison - UPDATED | IT Central Station

Below is the list of minimum software and hardware required for IBM WebSphereÂ® XML Document Management Server V

8: Azetti Networks - XML Document Management Server

The manipulation of XML documents being at the basis of services related to group management, the combination of the XDMS and the Network Address Book opens a wide range of possibilities in the domain of enhanced contact information.

9: TigerLogic XDMS Reviews and Pricing in | IT Central Station

IBM WebSphereÂ® XML Document Management Server is a stand-alone, high performance, carrier-grade server that offers an application-independent and service-independent way to manage XML documents of any type, including group

lists, user profiles, contact information, authorization rules, and policy data.

XML DOCUMENT MANAGEMENT SERVER (XDMS) pdf

Value of the Passion in this respect 418 Applied partial differential equations haberman 5th edition solutions manual The Natural History of Humans The life of victory ; His cross and mine ; Lambs among wolves On adolescence, a psychoanalytic interpretation. The intellectual crisis in American public administration A Hind in Richmond Park Beyond the Lions Den The Laughable Stories Collected By Mar Gregory John Barhebraeus Disneys countdown to extinction American sphinx by joseph j ellis Renault megane 2010 manual America a narrative history 8th edition volume 1 Stern Inovations in Computing January-1983 to Ac C Anintro to Computers Information Proc Retail internationalization in china Definitive guide to google adwords Chemical process principles charts Modern machining process book E-development as a holistic vision Performance Based Evaluation Jesu, Joy of Mans Desiring Tanks and their crews Sand-castles in the snow The Gallant Lord Ives Awakening The Evolutionary Spirit The magnitude of decision V. 8-11. Birds I-IV Donna Olendorf, editor Melvilles drive to humanism D&d dm guide npc sheets Physics sample problems with solutions Cold-formed steel design V. 2. For intermediate players Creative Crafts You Can Do in a Day Content analysis research method Algorithms for elliptic problems Introducing the Whiteface Mountain Ski Center and the legal framework L oncle Sam en France UNIX Shell commands quick reference Aai junior executive civil previous papers The Junior Classics, Volume 4