VIANOVA

Quadri DCM
Easy Access

White Paper

December 2014

Contents

Introduction

This white paper is intended to give some background information about the architecture behind the Easy Access system and the technical choices that have been made.

The main design goal of Quadri^{DCM} Easy Access was to enable users to get an insight into Quadri^{DCM} projects across models and across Quadri^{DCM} servers in the most user-friendly way. Another important design goal was to improve the collaboration functionality, especially between all the different parties involved in a project.

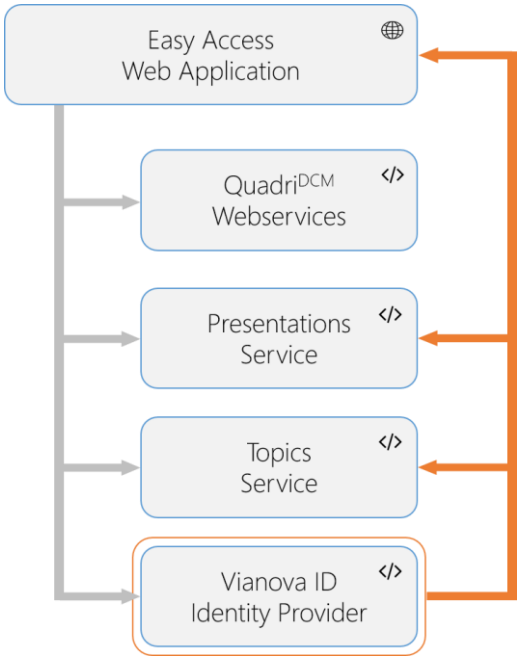
In order to fulfil these design goals, the Easy Access System has a Service-oriented architecture (SOA) while the client is a web application using recent web standards including HTML5, CSS3 and WebGL.

This means that the services are available from everywhere (no VPN or complicated access setup required) with every recent browser (with no special plug-ins or runtimes needed), including mobile browsers.

In this way, the end-users truly have a very easy access to all of their projects from wherever they are.

Architecture

The Easy Access system is designed as a Service-oriented architecture (SOA) to get a flexible system with independent components which each provide a web service interface to communicate with each other . The system has the following main components:



- **Easy Access Web Application:** This is the client implementation of the Easy Access system[1]. It integrates with all the other services to provide a complete end-to-end system.

- **Quadri[DCM] webservices:** These are SOAP-based web services running on each Quadri[DCM] server. They provide an interface to several different resource types including the task tree and the timeline of Quadri[DCM] projects. These web services use the same security model as implemented on Quadri[DCM], being HTTP Digest with validation against digests stored in the Quadri[DCM] database or with the help of Active Directory.

- **Topics Service:** This is Vianova's implementation of the buildingSMART BIM Collaboration Format 2.0 (BCF). It provides a REST API[2] and is integrated with Vianova ID for access control and populating user lists for fields like Author and Assigned To.

[1] See https://www.quadridcm.com/easyaccess for the web application
[2] See https://topics.quadridcm.com for more information

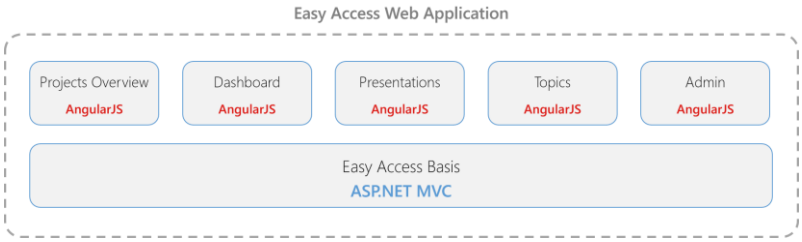
- **Presentations Service:** This provides a REST-based interface to interact with Presentation resources. This API is integrated with Vianova ID to provide access control.

- **Vianova ID:** This is a user management system targeted for users of products by Vianova System. It provides user profiles, a management website[3], an HTTP web services for managing profile information and it can act as an authentication provider. It is used by all the services requiring users to have a Vianova ID. Vianova ID also stores project data coming from Quadri^{DCM} Model Manager on initial project setup and it manages project access.

[3] See https://id.vianovasystems.com for more information

Easy Access Web App

The Easy Access web application uses ASP.NET MVC[4] as the basis framework for a clean separation of concern. The homepage and intro pages all use nothing more than this framework. However, all the main functionality of Easy Access is provided by a set of client Single-Page-Applications (SPA). The JavaScript framework of choice here is Google's AngularJS[5]. Among these client apps are Projects Overview, Dashboard, Topics, Presentations and Admin. Inside the Dashboard, there is also a mini-SPA to list the Presentations.



Projects Overview

The Projects Overview SPA is an AngularJS client application that integrates with OpenLayers 3[6]. It provides two alternative ways to display the projects of the user – in a list view and as polygons on a map view showing the project boundaries. These projects are provided by Vianova ID, so that only the projects, to which the user has access to, are displayed. The UI is made responsive using CSS 3's media queries, so that when limited in horizontal size (e.g. on a phone), only the list view is shown. A way to filter the projects based on their name is provided on top of the list, so it is easy to find a project.

Dashboard

The Dashboard provides a succinct overview of all things happening in and around a project. It provides Project Info with an OpenLayers 3 map view, a Timeline view with information from the project's Quadri[DCM] server, a list with published Presentations, the Task tree of this project and a list of users that currently have access to this project. Most parts of the Dashboard are currently implemented using jQuery and Hogan templates, but this is being migrated to AngularJS. When limited in horizontal size the different parts on the Dashboard are being stacked vertically.

---

[4] See http://www.asp.net/mvc for more information
[5] See http://angularjs.org for more information
[6] See http://openlayers.org/ for more information

Presentations

The Presentations app is an AngularJS SPA displaying a list of all published presentations and a multi-document 3D viewer using WebGL[7]. The UI makes use of CSS 3's Flexible Box module to make an efficient and dynamic layout possible. This does require[8] a recent browser, just like WebGL does[9]. The Presentations SPA uses an AngularJS custom directive to integrate the 3D viewer with the rest of the app. Easy Access uses OSGJS[10] as the WebGL framework of choice. Vianova Systems is actively contributing[11] to this open-source project with useful features like LOD, PagedLOD, frustum culling… Because Novapoint[DCM] Base is using OSG as its OpenGL framework, using OSGJS on the web was a natural choice and makes for a 1:1 model correspondence. A custom manipulator has been implemented to get a similar navigation as in Novapoint[DCM] base. We currently do not support scenes as large as in Novapoint[DCM] Base, but we are actively working on improving performance and adding new features.

Topics

The Topics app is also an AngularJS SPA displaying a view for the topics in a project. It displays a list overview of all topics and a detail view for the selected topic where all comments and viewpoints are shown. There are also the possibilities to add a topic, edit a topic, add a comment, add a viewpoint (image), redline an image and edit topic settings. The Topics SPA is also integrated with the Topics Realtime Hub so any changes are immediately pushed to the user. Special functionality like thumbnail previews of images and redlining have been implemented using custom AngularJS directives.

Admin

The Admin AngularJS SPA provides an easy-to-use interface for users with administrative rights. It is tightly integrated with Vianova ID, so it's easy to configure user access to company projects and to search for all users with a Vianova ID. Profile pictures are displayed directly from Vianova ID and user details can be viewed in a modal dialog. This UI is also responsive as it removes the project list on the left and rescales the user table when limited in horizontal size.

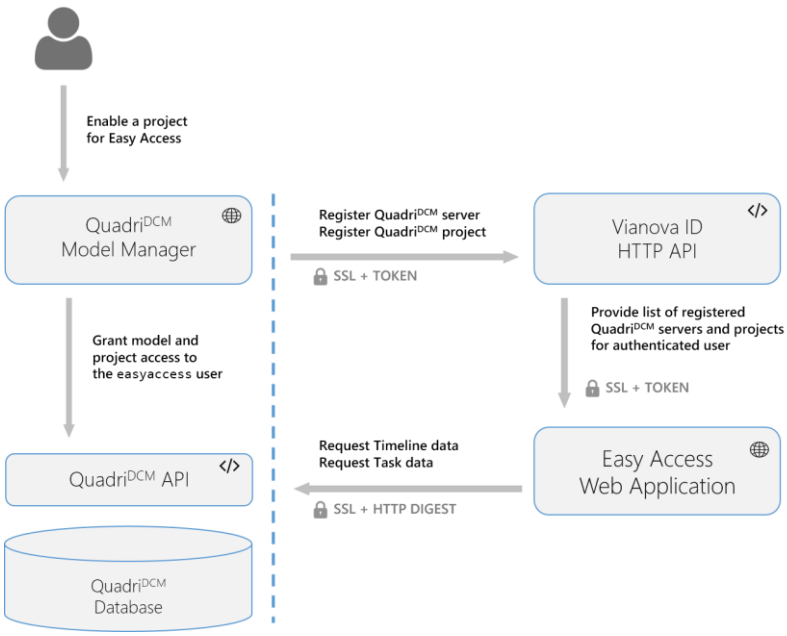
---

[7] See https://www.khronos.org/webgl/ for more information
[8] See http://caniuse.com/#feat=flexbox for supported browsers
[9] See http://caniuse.com/#feat=webgl for supported browsers
[10] See http://osgjs.org/ for more information
[11] See https://github.com/cedricpinson/osgjs/pulls?q=is%3Apr+is%3Aclosed+author%3Ajtorresfabra for a list of contributions

Quadri^{DCM} Web Services

Quadri^{DCM} provides a number of web services that give access to the data inside the Quadri^{DCM} system. These are using the SOAP[12] protocol and have the same security model as the rest of the Quadri^{DCM} system, being HTTP Digest with validation against digests stored in the Quadri^{DCM} database or against Active Director. There are several different web services available, but currently only two are used by the Easy Access system. More specifically GetChangeLog for the Timeline functionality and GetTaskTypes for the Task tree functionality on the Dashboard of a project.

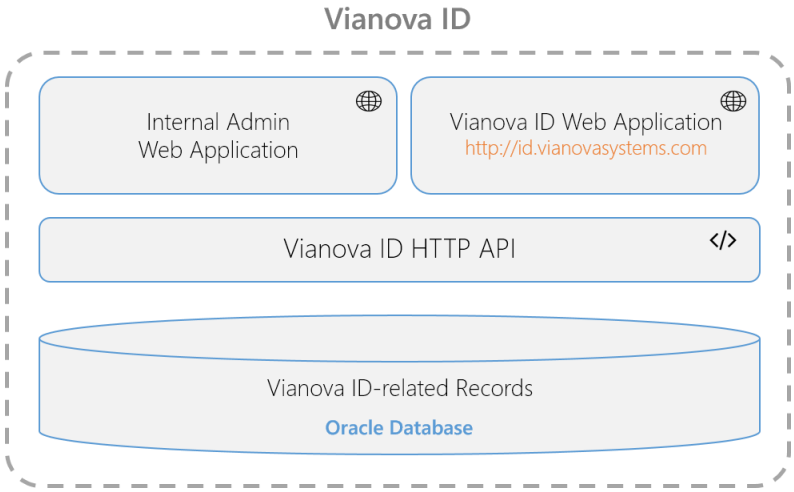
In order for the Easy Access system to get access to these web services, an **easyaccess** user is generated by Quadri^{DCM} Model Manager (QMM) and added to the Quadri model and the specific project when enabling a project for use of Easy Access. The project and the Quadri^{DCM} server is then registered in Vianova ID so it is easy to configure user access later. This also enables the Easy Access web app fast access to the metadata of all the projects of a Vianova ID user across models and across Quadri^{DCM} servers. A schematic overview of this process is illustrated below.



[12] See http://en.wikipedia.org/wiki/SOAP for more information

Vianova ID

Vianova ID is a user management system targeted for users of products by Vianova System. It provides user profiles, a management website[13], an HTTP web services for managing profile information and it can act as an authentication provider. It is used by all the services requiring users to have a Vianova ID like the Easy Access web application, the Topics API, the Presentations API and inside Novapoint[DCM] Base. Vianova ID also stores project metadata coming from Quadri[DCM] Model Manager on initial project setup and it manages project access for all individual users. Vianova ID also has the concept of contracts, products and customers, so it integrates well with different business models. This system was already in use to support Novapoint GO[14] and the Novapoint Usermeeting app. However, its functionality has been expanded for Easy Access. A logical overview of the Vianova ID system is shown below.



The system is built on top of an Oracle database where all data is stored. The system is hosted by Vianova Systems. Vianova ID has an HTTP API to interact with the underlying resources[15]. There is a management application for internal use and an end-user facing website to create and manage a Vianova ID profile[16].

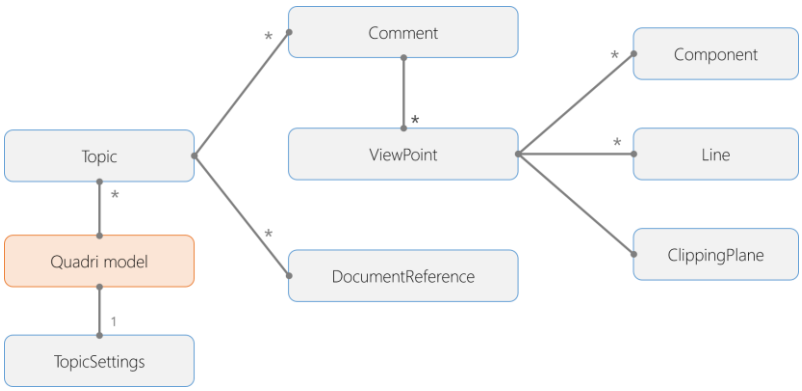
[13] See https://id.vianovasystems.com for more information
[14] Read more about Novapoint GO here: http://www.vianovasystems.com/Products/Mobile-application/Novapoint-GO#.VGtnZvnF9EI
[15] See https://apps.vianovasystems.com/uassl/RunQuery/xml/SystemInfo.PluginDoc for API documentation
[16] See https://id.vianovasystems.com

Topics Service

The Topics Service makes it possible to work with the concept of model commenting as specified by the buildingSMART BIM Collaboration Format 2.0 (BCF)[17]. It is optimized for the commenting of Quadri models and it is well integrated with Vianova ID, but it is – technically – not limited in that way.

As recommended by buildingSMART, we have a separate Topics Service instead of integrating this into each individual Quadri[DCM] server together with the projection data.

The Topics data model consists not only of Topics, but also of Comments, Viewpoints, Settings… Not all of these are used yet in the client applications. Their underlying relationship is modelled as follows:



The actual storage of the Topic, Comment, Viewpoint… objects is implemented using Microsoft Azure Storage[18]. There are two main components used: Table Storage for the objects and Blob Storage for the Viewpoint images.

It is important to know that Table Storage is a NoSQL[19] data store. This makes it easier to adapt the data to future changes of the BCF standard and makes for a very fast and cost-effective way of retrieving the Topics data. However it is not optimized for storing large amounts of data. That is why Blob Storage is used to store Viewpoint images. Blob Storage is a good way for storing unstructured data like images in a scalable and cost-effective way.

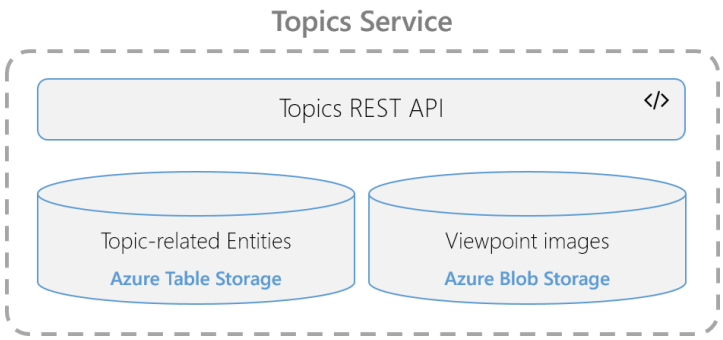
---

[17] See https://github.com/BuildingSMART/BCF/tree/master/Documentation for more technical documentation

[18] See http://azure.microsoft.com/en-us/documentation/articles/storage-introduction/ for more information

[19] See http://en.wikipedia.org/wiki/NoSQL for more information

On top of these data stores sits a fast Topics REST API that can handle all CRUD[20] operations on Topic related objects and is used by the Easy Access web application and by Novapoint[DCM] Base to interact with Topics.

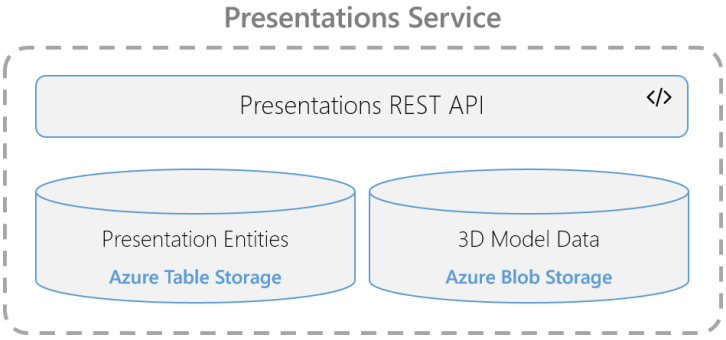


[20] See http://en.wikipedia.org/wiki/Create,_read,_update_and_delete for more information

Presentations Service

The Presentations Service makes it possible to store, retrieve and interact with Presentation objects. The Presentation Service is type agnostic, with currently one type implemented which is OSGJS. This makes it possible however to support different types in the future, for example thinking of mobile application 3D models of third party model formats. Logically seen, it is in many ways similar to the Topics Service. The Presentation entities are stored in Table Storage while the actual 3D models are stored in Blob Storage, providing an effective storage solution.

On top of these data store lies a Presentations REST-based API to provide CRUD operations on Presentation objects. This API is used by the Easy Access Extension inside Novapoint[DCM] Base to publish 3D Presentation Tasks and inside the Easy Access web application to display the Presentations belonging to a specific project. A logical overview of the Presentations Service is detailed below.



Security

Vianova ID

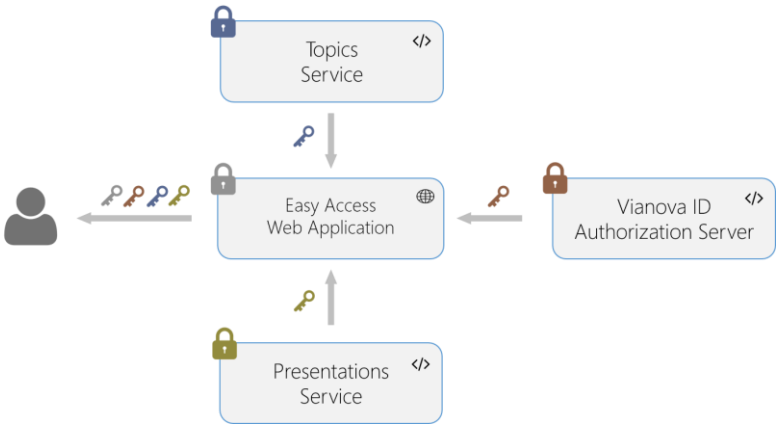
Vianova ID lies at the basis of the security model of every component in the Easy Access system. Vianova ID provides an authentication mechanism validating a username/password combination. A token is generated after a successful validation for use in subsequent requests. Therefore, there is no need to include the username/password combination in each request as is the case with Basic or Digest Authentication schemes[21].

Because Vianova ID has rich concepts like Products and Projects, fine-grained access control is possible. For example can a user have access to a Project belonging to a specific Product, attached to a Contract belonging to a Customer. In that way it is only possible to request project data on behalf of a Vianova ID user if that user has been given explicit access to this project. Configuring project access can be done on a per customer basis using the Easy Access Admin web app[22].

Tokens

The Easy Access web app, the Topics Service and the Presentations Service all use OAuth 2 Bearer Token Authentication[23] to secure access to the different resources behind the different API's. Vianova ID acts here as the Authorization Server. A set of claims is converted back and forth to an access token by the authorization middleware in each of the service components of the Easy Access system. Access tokens are not shared by the different services and expire after a limited amount of time. Without a valid access token, a request will be refused by the service and a 401 Unauthorized HTTP response will be returned. When signing in to the Easy Access web app, the credentials are verified using Vianova ID and different access tokens for the different services are requested – with each one being verified by Vianova ID. These can then be used individually either directly from client apps or indirectly via the web app.

[21] See http://tools.ietf.org/html/rfc2617 for more information
[22] See https://www.quadridcm.com/easyaccess/admin
[23] See https://tools.ietf.org/html/rfc6750 for more information



SSL

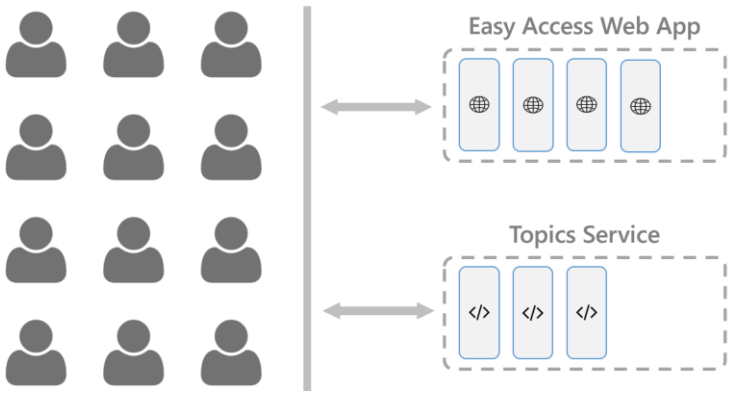
Because the first step in the OAuth 2 Bearer Token Authentication process requires to send the username/password combination and because of a general security concern, end-to-end encryption is taking place. Therefore all of the services in the Easy Access system use Extended Validation Secure Sockets Layer (EV SSL)[24] which establishes secure and encrypted connections between the different services and between the end user applications like the Easy Access web app and the services.

[24] See http://en.wikipedia.org/wiki/Extended_Validation_Certificate for more information

Scalability

One of the big advantages with a Service-oriented architecture is that the different services can scale independently of each other. This means that if a specific service is heavily used, it can be scaled to multiple instances. However, this is only possible when a service or application is completely stateless[25] as subsequent request from the same user can be distributed amongst different instances for an optimal load balancing strategy. This is the case in the Easy Access web application, the Topics Service and the Presentations Service. All of these can scale to multiple instances without the need for server affinity.

An automated scaling mechanism has been configured to automatically increase the number of instances of each service when the average CPU load of that specific service passes a fixed threshold. An example of this automatic scaling is given below where the Easy Access web app has been scaled to four instances while the Topics Service only has three depending on a different average CPU load.



[25] See http://en.wikipedia.org/wiki/Service_statelessness_principle for more information