Pedram Hadjian

Software Architect

Sonnenblumenweg 1a,
91080 Spardorf

+49 151 6522 5832

www.linkedin.com/in/hadjian

pedram.hadjian@gmail.com

13 years of experience as a Software Developer / Manager. Recently developed and lead customer facing IoT solutions in different industrial fields (building automation, manufacturing, energy). Maintained a competitive time frame of 1 year from concept to fully operational solution across projects. Responsible for building IoT portfolio with cross-functional teams of product owners, software architects and developers. Responsible for acc. R&D budget of 6 M within 3 years. Comfortable engaging accounts on CxO-level (CEO, CIO etc.). And I like to write understandable documents.

Work Experience

Oct/2018 — today
1 year, 5 Months

Product Owner / Software Architect

evosoft GmbH

Tasks: Lead Customer Cocreation with world's largest HVAC OEM
Developed cloud-based Expert and Reporting System for Chillers
Integrated Expert and Reporting System into Siemens BP Cloud Infra
Pitched W3C Web of Things (WoT) Implementation to Siemens Management
Designed Architecture / created first Prototype of WoT Protocol Adapter
Lead Development of WoT Building Blocks for cross-domain Solutions

Tools: *Knowledge Base: gitlab, sphinx-doc, reStructuredText, graphviz*
Programming: golang, Docker, Linux
Network Programming: HTTP/S, WebSocket, TCP/IP, OPC-UA, CoAP

Achievements: Responsible for R&D Budget of 2.5 M
Operationalized Expert and Reporting System across 400 Chillers
Received Funding from Siemens Management for WoT Development
Delivered first WoT Building Block as Inner Source

Oct/2016 — Sept/2018
2 years

Portfolio Manager / Software Architect

evosoft GmbH

Tasks: Created IoT Portfolio with Focus on Solutions and Building Blocks
Managed Knowledge Ramp-up for Building, Energy and Manufacturing
Architect for IoT Integration Framework within Building Automation
Defined and contributed to UX process
Delivered 12 projects across Portfolio

Tools: *Knowledge Base: gitlab, sphinx-doc, reStructuredText, graphviz*
Programming: golang, Docker, Linux
Network Programming: HTTP/S, WebSocket, TCP/IP, OPC-UA, CoAP

Achievements: Grew Portfolio to Staff of 15 (developers, product owners)
Responsible for R&D Budget of 4 M
Defined Architecture / developed first IoT Building Block

Mar/2012 — Sept/2016
4 years, 7 months

Sales Director / Program Manager

Digital Medics GmbH

Tasks: Lead international Projects (Germany, Saudi Arabia)
Deployment of a cloud-based teleradiology platform
Personnel responsibility for staff of 6 (2 Project Leads, 2 DevOps, 2 Sales)
First point of contact for stakeholder at CXO-level

Achievements: Shipped 12 projects (~ 400,000 € /project)

Jan/2011 — Feb/2012
1 year, 2 months

Project Lead: Evangelisches Klinikum Niederrhein

Digital Medics GmbH

Tasks: Personnel responsibility for staff of 2 (1 Project Lead, 1 Back-Office)
Project delivery to large hospital (1,274 beds) across 4 sites
Lead cooperation with IBM for private cloud project
Implementation of a SOA for an image streaming platform
First point-of-contact for C-level stakeholder

Tools: *Image Streaming Service: TCP/HTTP, C++, Qt Framework, LDAP*
Data Migration Service: TCP/HTTP, Londiste, proprietary caching system
DevOps Tools: Jira, Icinga, OpenVPN, key-based SSH, Debian Apt, CFEngine

Achievements: Implemented private cloud (~100 TB) across 4 sites
Implemented an agile deployment pipeline
On-Budget agile implementation of customization projects

Nov/2008 — Dec/2010
2 year, 2 months

Sales Director

Digital Medics GmbH

Tasks: Project Lead for new accounts
Negotiation on C-level with large hospital (1,274 beds)
Host reference site visits for qualified leads
Showcase at international exhibitions (Dubai, South-Korea)

Achievements: Closed 6 new projects (~120.000 €/project)
Closed contract with large hospital (1,274 beds, 700,000 €)

Nov/2007 — Oct/2008
1 year

Software Certification, MDD (93/42/EWG)

Digital Medics GmbH

Tasks: First point-of-contact for certified body
Certification in compliance with the Medical Device Directive, class 2b
Implemented documented risk management in compliance with ISO 13485
Implemented software development life cycle in compliance with DIN EN 62304

Jan/2006 — Oct/2008
2 years, 10 months

Software Developer

Digital Medics GmbH

Tasks: Requirement analysis with clinical partners
Development of a high-performance radiology reading software
GPU-acceleration of compute-intense modules
Flexible visualization of database query results

Tools: *Medical Imaging Front-End: C++, Python, Qt, OpenGL*

Achievements: First software to market, integrating complex visualization functionality in radiology

Education

Dec/2005
6 years

Diplom Informatiker *(Grade: "Sehr Gut")*

(approx. equiv. to M.Sc. in Computer Science (Grade: 4.0/4.0))
Technical University of Dortmund

Jan/2005 — Dec/2005

1 year

Thesis: "Voxelerator - an FPGA-based volume visualization pipeline", *Grade: "Sehr Gut" (best grade)*
Technical University of Dortmund

Tasks: Implementation of a logic design for volume visualization
Design on register-transfer level
Verification of synthesis results on gate-level
Implement software-based pipeline simulation for verification

Oct/2002 — Sept/2003
1 year

Project Group "RoCK - Robot Construction Kit"
Technical University of Dortmund

Tasks: Development and implementation of a robot control unit
Development of a custom OS and a simple ANSI-C compiler
Circuit board design and implementation
Low-level driver programming to interface with sensors
Full-stack debugging of application-, compiler- and logic design errors

Further Qualifications

IT-Skills

Programming/OS: golang, C++, Python, Bash, CMake, Make, qmake, PostgreSQL, Linux (server/clients)

VCS/CI/CD: git, subversion, gitlab, OpenShift, Kubernetes, Docker

IDE: emacs, vim

Network Programming: HTTP/S, WebSocket, TCP/IP, OPC-UA, CoAP)

Languages

German (native)

English (business fluent)

Awards

2009

1st place, "Gründerwettbewerb" business plan competition
Organized by: Federal Ministry of Economic Affairs and Technology
Prize money: 25,000 €

2006

Selected for EXIST-SEED support program
Organized by: Federal Ministry of Economic Affairs and Technology
Grant: 40,000 €

2005

1st place, "Start2Grow" business plan competition
Organized by: The Economic Development Agency of Dortmund
Prize Money: 65,000 €

UNIVERSITÄT DORTMUND

FACHBEREICH INFORMATIK

DIPLOM

Ali Pedram Hadjian

geboren am 18. Mai 1980 in Teheran
hat am 27. Oktober 2005 die Diplomprüfung im Studiengang

Angewandte Informatik
mit dem Schwerpunkt Ingenieurwissenschaften
(Ingenieurinformatik)

gemäß der Prüfungsordnung bestanden
mit dem Gesamturteil

SEHR GUT

Aufgrund der Prüfung wird ihm hiermit der akademische Grad

Diplom-Informatiker

(Dipl.-Inform.)

verliehen.

Dortmund, 31. Januar 2006

Der Dekan
des Fachbereichs Informatik

Universität Dortmund - Fachbereich Informatik
1
Der Dekan

Die Vorsitzende
des Prüfungsausschusses

Introduction to MindSphere

Disclaimer

"Unless specific license terms are provided or expressly agreed with you, this documentation is intended solely to enable you to evaluate our developer services in relation to applications for the *MindSphere* Platform, and you are not entitled to use this documentation for or in connection with the development of any such applications. No license to any software, know-how or other intellectual property rights is granted, conveyed or implied, and all rights not conferred are expressly reserved by Siemens.

UNLESS EXPRESSLY AGREED OTHERWISE WITH YOU, WE MAKE AND YOU RECEIVE NO EXPRESS OR IMPLIED WARRANTIES OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY EXPRESS OR IMPLIED WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, COURSE OF DEALING AND USAGE OF TRADE OR AGAINST INFRINGEMENT, WITH RESPECT TO THIS DOCUMENTATION AND ANY SOFTWARE, KNOW-HOW AND OTHER INTELLECTUAL PROPERTY CONTAINED HEREIN. IN NO EVENT WILL SIEMENS AND/OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO YOU FOR ANY CLAIMS OR DAMAGES WHATSOEVER ARISING OUT OF OR RELATED TO THIS DOCUMENTATION AND ANY SOFTWARE, KNOW-HOW AND OTHER INTELLECTUAL PROPERTY CONTAINED HEREIN, REGARDLESS OF THE FORM OF ACTION, WHETHER IN CONTRACT, TORT OR OTHERWISE."

Overview

This introduction was written to give developers an overview of the MindSphere platform. We will briefly describe the underlying technology, basic components and interaction among these components as well as provide pointers to documentation that can help you to dig deeper.

What is MindSphere?

Siemens MindSphere is an operating system for the internet of things. This means that MindSphere provides a complete soft- and hardware infrastructure, which enables organizations to develop and run data- and compute-heavy applications, without requiring deep knowledge of how to build or run such infrastructure.

The Technology behind MindSphere

The following section introduces the different underlying technologies that help MindSphere achieve the ambitious goals mentioned above.

Cloud Computing

Cloud Computing was first made popular by Amazon Web Services (AWS) and for the first time allowed to scale resources on-the-fly according to instantaneous app demands. The software components necessary to provide an environment for cloud applications can be divided into three layers:

- **Infrastructure as a Service (IAAS):** This term is used to describe the possibility to adjust the use of hardware resources, like network, computing and storage, to meet instantaneous application demands. Due to IAAS, the process of manually ordering virtual servers transformed into the simple invocation of a command. Now it is possible to let applications automatically increase resources in times of high load and to quickly loose those additional resources, as soon as the load drops. Such a service is provided by AWS or Google Cloud Compute.
- **Platform as a Service (PAAS):** This layer runs on top of the IAAS layer and instead of creating basic virtual servers, it facilitates easy creation of fully-fledged services and connections among those services. Equipped with a PAAS layer, a developer writes simple configuration files to automatically add storage, caching, authentication and many other services to custom web applications. Configuring each and every service at the OS level is a thing of the past. The **MindSphere** platform offers such a PAAS layer (based on Pivotal's CloudFoundry) and extends the basic IT services with additional ones that are specific to an industrial environment (e.g. asset management, data point acquisition, time series processing, predictive maintenance etc.).
- **Software as a Service:** Applications based upon aforementioned layers fall into this category. As resource scaling is inherent to this type of application, it is common to pay per usage as opposed to paying per license. The **MindSphere** platform makes it easy to develop and run such applications, but also offers ready-to-use cloud applications.

Further Reading

- Google Cloud Compute: https://cloud.google.com/docs/overview
- CloudFoundry: https://www.cloudfoundry.org
- Netflix OSS Blog: http://techblog.netflix.com/

CloudFoundry

CloudFoundry (CF) is a PAAS layer, which was open-sourced by Pivotal Software and is now being developed by the CloudFoundry Foundation, a non-profit organization. Members of this organization are companies like IBM, Google, Dell and VMWare.

By being able to run on different IAAS platforms, CF offers a cross vendor interface for cloud computing. A notable feature of the CF layer is its extendability to include vendor-specific services. The Google Cloud Compute platform for example uses this feature, to expose several proprietary services (BigQuery, Google Machine Learning etc.) through the CF interface.

It will be useful to know two of the terms surrounding the CF interface:

- **CF Org:** An organization within a CF environment, which defines users, access restrictions and spaces and is managed by one or more Org admins
- **Spaces:** Environments that isolate the operation of services and provide a logical grouping. Typically a deployment pipeline for developers makes use of spaces to separate dev from test and these from production

With the objective of becoming the leading open IoT platform, **MindSphere** is based on and extends CF to be portable and independent of specific IAAS vendors.

Further Reading

- CloudFoundry Foundation: https://www.cloudfoundry.org

Microservices

The term "microservices" describes an implementation approach for applications, which are composed of smaller self-contained services. These services use technology agnostic protocols to communicate among each other. In real-world applications, the HTTP-protocol is used almost exclusively for this communication. The goal of this approach is threefold:

- Self-contained services may be updated and replaced independently of each other
- As communication protocols are technology agnostic, the technology for service implementation may also be chosen independently of choices made for other services
- The modular design of such systems lends itself to being continuously updated as opposed to large lock-step updates of many interconnected modules

The **MindSphere** platform itself is composed of microservices and, by extending the CF platform, offers interfaces to enable developers to quickly and easily create custom services and connect them to others.

Further Reading

- Wikipedia article: https://en.wikipedia.org/wiki/Microservices
- Pointers by Martin Fowler: https://martinfowler.com/microservices/

Representational State Transfer (REST)

The term REST is closely related to the microservices approach with an emphasis on the communication among services. This emphasis is not surprising, as the term was coined by one of the main developers of the HTTP protocol. The discussion about what REST is a very broad field, so the reader is advised to combine hands-on implementation and reading on the subject in tandem, until the concept becomes clear enough to serve the development goal. Don't get lost in the rabbit hole.

To sum up: the idea takes the notion of the World Wide Web further and regards not only documents and files as resources, which are addressable by means of an uniform resource identifier (URI, commonly known as the web address). It extends the meaning of resource to many other things that can be represented or manipulated textually. Think of a storage space, which can be increased, decreased or inquired for current size by invoking certain URIs and parsing the response given in XML, JSON or other structured document formats.

With this kind of communication, additional properties are imposed on implementations to qualify as RESTful:

- **Client / Server:** Services should be concerned with one specialized task and offer this service to other entities
- **Stateless:** Services should handle requests from other services w/o relying on the context of the client. This enables e.g. a request to be redirected to any service of the same type

for load balancing

- **Cachable:** Responses must be defined as cachable to decrease unnecessary communication
- **Layered:** Requested information may be gathered and aggregated by an intermediary service. An example being a reverse proxy
- **Uniform Interface:** This property goes deep, but in short it requires the interface to be independent of the internal implementation and its mechanisms to be uniform across services. Think of database entries returned as XML regardless of its internal implementation being a key-value store, a relational database or just a bunch of files.

Use this approach as a guideline, but be aware not to take the concept too far. Too many cascading REST calls quickly amount to unwanted latency hits. The **MindSphere** services are exposed through such an interface and thusly allows for applications to scale gracefully and for developers to conduct a modern approach to software development.

Further Reading

- Explanation by Roy Fielding, the inventor of REST: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

Backing Services

In addition to industry specific services (introduced later), **MindSphere** offers developers to easily create so called backing services from a number of popular open-source products to quickly get their web apps up and running. These services are described in the following sections.

Rabbit MQ

RabbitMQ is an open-source message broker software (sometimes called message-oriented middleware) that implements the Advanced Message Queuing Protocol (AMQP). The RabbitMQ server is written in the Erlang programming language and is built on the Open Telecom Platform framework for clustering and failover. Client libraries to interface with the broker are available for all major programming languages. The source code is released under the Mozilla Public License.

Main usage in MindSphere:

- Publish / Subscribe based messaging between micro-services within CloudFoundry
- Relatively high message throughput

PostgreSQL

PostgreSQL is one of the most widely used relational databases. As a relational database PostgreSQL is ACID-compliant and transactional. PostgreSQL is open-source and is distributed under the permissive PostgreSQL license.

Main usage in MindSphere:

- Store persistent data, which requires ACID-compliance and transactionality

NoSQL Databases

A group of services fall under a type of database named NoSQL, which stands for "non SQL", "non relational" or "not only SQL", so a short introduction of this type is in order.

NoSQL databases do away with the classical notion of storing information in relational tables and by doing so, they relax the ACID guarantees of relational databases.

The need for this new approach arose from the development of web applications, which quickly confront developers with scalability and performance-at-scale problems. Databases at a certain size must scale horizontally instead of vertically, which means you add servers instead of increasing the performance of one. Making a relational database scale across different machines is hard, because one query is likely to span multiple machines. Also keeping data consistent is made harder due to the relational nature of the database.

But if data doesn't require this consistency guarantee, e.g. if each data element is self-contained anyway, then a NoSQL database can store all elements in one long table with no relations. This long table is easily cut into vertical pieces to be distributed across multiple machines.

Also, by stripping away complexity within the database code itself, NoSQL databases are usually orders of magnitude faster than their relational counterparts.

MindSphere currently provides two NoSQL databases to be used by developers: MongoDB and Redis. In the following, the two databases and their intended usage are introduced.

MongoDB

MongoDB is an open-source document-oriented database. A document-oriented database is one type of NoSQL database, which stores documents in a structured format (similar to JSON) and corresponding keys as a way to retrieve the documents.

This type of storage relaxes some constraints of relational databases (ACID) in favor of performance, scalability and low management overhead.

MongoDB is distributed under a combination of the GNU Affero General Public License and the Apache License.

Main usage in MindSphere:

- Permanent storage of e.g. JSON based configuration data.
- Document and File storage for unstructured data that does not relate to assets (e.g. FW update files, scripts etc.)

Redis

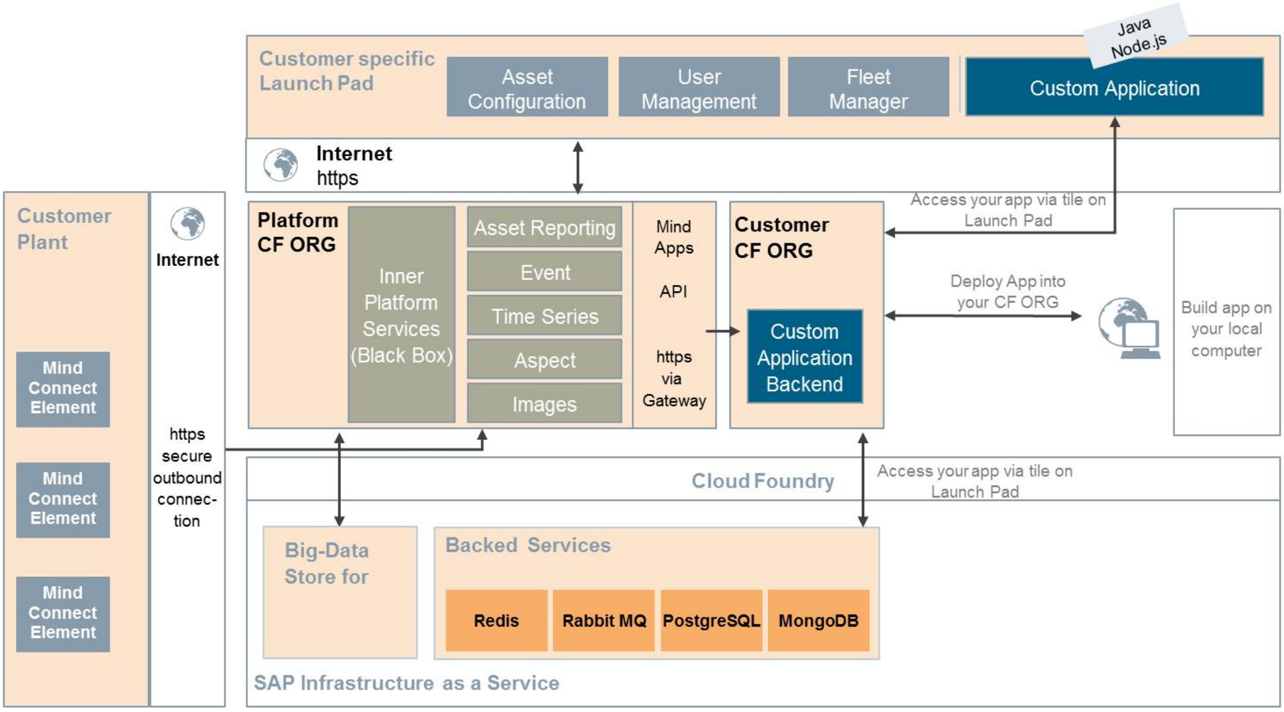
Redis is a data structure server, which means, that applications can use it as if they were using normal data structures for caching in-flight objects like user sessions or page statistics. Redis offers basic structures and operations to execute on these structures.

Storing these objects on a separate server has some advantages, like sharing between processes, built-in persistence and again easy scaling. As an application cache for in-flight structures, the system is tuned for high-write throughput. The source code is released under the BSD license.

Main usage in MindSphere:

- Application cache and temporary data store for volatile data
- Data cache for data in transit (e.g. for chunks coming from the agents)

MindSphere Architecture



1 Overview of the MindSphere landscape

The following section will introduce the overall landscape of **MindSphere** and make you familiar with the commonly used features and how the different parts work together.

The platform itself can roughly be divided into four main parts (as shown in fig. 1):

1. **Platform CF Org:** Remember Orgs in CF? This management domain encapsulates all the core services of the platform, like data ingestion, persistent storage and services that will do useful work on that data. A developer can access this Org through these services, which are exposed through RESTful APIs.

2. **Backing Services:** These services provide basic functionality useful for any web app like databases, logging frameworks, communication busses etc. These backing services are exposed through the CF interface and can be easily configured to connect to your web app. A list of currently available services is given in the section “Backing Services“.

3. **Customer CF Org:** This is where developers login, deploy applications and generally make use of services in the Platform Org or of IAAS services exposed through CF. An Org is administered by an Org admin, who can for example create spaces or configure user access rights.

4. **Launch Pad:** This area is where users of apps login and select the app to use. When selecting an app, access rights are checked and the user is redirected to the so called App Router, which is a reverse proxy that handles all requests on behalf of the MindSphere apps running on the platform.

Internal and External Interfaces

There are different interfaces for different roles, which shall be described in the following section.

Data Ingest Interface

To do anything useful, an organization must be able to securely upload data to the MindSphere platform. For this to happen, you should have received a MindConnect agent from your friendly Siemens dealer. An agent can be a connector appliance like the MindConnect Nano or a custom application making use of the MindConnect libraries.

CF CLI Developer Interface

This interface provides developers with methods to develop and operate custom applications, which can be bound to additional backing services and access platform services. It is best to try out the examples in the MindSphere "Getting Started" Guide to get familiar with the workflow of CF app development.

Basically you will require a MindSphere developer account as well as the CF command line client (CLI). With a simple "cf login" you can specify the official MindSphere API endpoint and authenticate with your account credentials. Once authenticated you will be able to push local apps to the platform and run them.

To deploy your app to production, you will have to ask the Org administrator to add certain routing rules to the so called App Router. The App router is a reverse proxy, which receives all requests from the Launch Pad and routes these to the correct app to be handled.

MindApps API

This is the official interface to all the internal services provided by the Platform CF Org. Please refer to the API documentation for details. A more detailed look at available services will be given in the following section.

MindApps Services

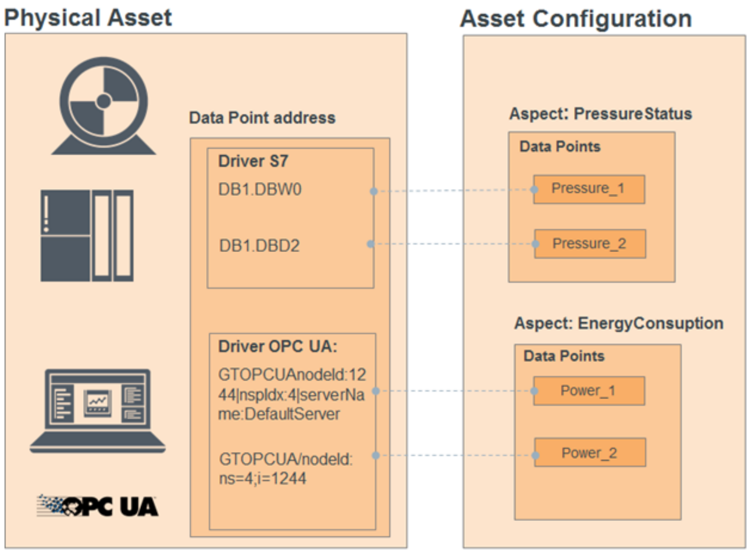
The first services available for public use are described at a high level in this section. All services are accessible through a RESTful API, so you might want to refer to the API reference for details.

Time Series Service

By interacting with the API of the time series interface, it is possible to retrieve collected data for an asset as an array of time stamp and value tuples. Parsing of the raw data was already executed on arrival of the data, such that applications can access it in a structured way.

The interface allows specifying a time interval for requests and supports paging of the returned data for e.g. easy display in a browser.

Aspects Service



The term Aspects is taken from the context of asset configuration and comes to mean groups of data points of one asset with a contextual relationship like the data points for “power”, “current” and “voltage” might belong to the aspect “energy_management” (as depicted in the illustration above).

The aspect service lets you programmatically inquire the name of the aspect, configured variables, the reading cycle, transfer frequency as well as additional meta-data.

Events Service

The events service lets you retrieve a list of notifications of an asset, which requires the attention of maintenance personnel or service engineers. The asset might notify of empty stacking magazines, resource contentions or other error conditions that might require attention.

Every event is comprised of a tuple of properties and their corresponding values. The properties are timestamp, severity, status and description.

Asset Reporting

This service simply returns a list of assets in a structured way. The request may specify filter, ordering and grouping criteria by property.

Image Service

This service is able to return the following images:

- The originally uploaded asset floor plan image
- Original floor image stored for the given asset in Base64 encoded format
- The generated thumbnail image of an asset

Digital Medics GmbH answers the demands of its distributed medical imaging solution with IBM technology

Overview

The need

Digital Medics created an end-to-end high-resolution medical imaging system, which is a major advancement on current solutions that can only handle scaled-down versions of large data sets. The new solution called for a scalable IT infrastructure that could cope with the increased demand in storage and processing power.

The solution

Digital Medics and IBM® created an elastic private cloud, which can be easily scaled to handle very high peaks in workload. The cloud runs in a virtualized Linux® environment supported by IBM BladeCenter® technology. IBM System Storage® DS8300 and TS3100 systems are used for data storage, configured by IBM Tivoli® Storage Manager.

The benefit

By providing the ideal support for Digital Medics' medical imaging solution, the IBM infrastructure contributes to impressive cost savings for clinicians, with one hospital client estimating savings of up to 300,000 EURO a year. Advanced data compression and automated storage management cut data storage costs by more than 50 percent.

Headquartered in Dortmund, Germany, Digital Medics GmbH develops innovative solutions in the field of medical image processing and visualization for medical imaging. VivoLab is the company's revolutionary new diagnostic tool designed to enable radiological specialists to interact with images directly using a pen-enabled display. Innovative solutions such as these help healthcare professionals to rapidly deliver high-quality diagnoses and reduce overheads. Every solution implemented by Digital Medics GmbH is bundled with monitoring and maintenance, ensuring clients are fully supported in their journey towards an end-to-end high-resolution imaging workflow.

Dealing with the data explosion

Image data sets produced by the current generation of medical imaging devices increased up to a factor of 16 to 320 when compared to data sets produced by older generation devices. The increase can be attributed to higher-resolution images, e.g. for multi-slice-CTs, on the one hand and to the addition of functional data, e.g. for PET/CT, on the other. These advancements – also called the data explosion, come with disadvantages as well as benefits.

Physicians in hospitals and practices are flooded with images and are forced to adapt the everyday diagnostic routine to this rapid change in technology. Being confronted with low performance image viewing software at the frontend and a backend originally developed for handling a fraction of the images produced nowadays, physicians take the fastest way out of this dilemma by reducing the size of the produced data sets, before they are transferred to the image archiving and distribution system.

Digital Medics identified three areas that needed improvement to enable an end-to-end high-resolution workflow. These were enhancing image transfer through networks with varying characteristics, performance at the backend and user frontend, and tools at the user frontend for advanced analysis of large-scale data.

Designed for Data

- Images are stored as 'thin slices' in the archive, downloadable at high speed by clinicians as high-resolution medical images from anywhere onsite. IBM Tivoli Storage Manager configures data storage, handling tens of terabytes to optimize use of resources and ensure very high system responsiveness.

Tuned to the Task

- The IBM BladeCenter HS22, IBM Tivoli Storage Manager and Linux operating system together comprise a data retention stack, which provides the ideal platform to meet the expected peaks and troughs, offering unparalleled versatility and performance. Exceptional scalability enables Digital Medics to quickly and easily expand capacity or processing power by simply plugging in an additional storage bay or BladeCenter blade server.

Managed with Cloud Technologies

- Digital Medics has created a private cloud solution for one of its largest clients. Rather than invest in fixed hardware capacity, when demand for diagnostic imagery rises, the IBM Tivoli Storage Manager software and the Red Hat virtualization layer automatically deploy greater resources to ensure continuous high-performance service delivery. The storage location of each study in this multi-site environment is hidden from the user, as migration logic is handled by the cloud itself. Leveraging cloud computing techniques masks the complexity of the underlying components, dynamically optimizing both storage and data processing.

Driving Innovation

- Digital Medics has developed a groundbreaking medical imaging solution. By selecting IBM hardware as the platform of choice, Digital Medics offers clients superb reliability and flexibility, replacing physical systems with high-speed visual diagnostics and visualization dedicated to improving patient outcomes.

A key to succeeding in its goal was to find a suitable hardware infrastructure that could keep pace with the challenging demands posed by this high-resolution workflow. The huge volumes of data involved in processing, storing and distributing medical images meant that an exceptionally high-performance platform was required. Moreover, the selected IT solution needed to offer excellent levels of stability and availability, as any interruption or delays to a medical imaging service could have serious consequences for patients' health. With the demand for medical images impossible to predict and constantly fluctuating, high levels of flexibility and scalability were also called for.

Specifically, the company needed to select the correct infrastructure to meet the requirements of a key client: the German clinical center Evangelische und Johanniter Klinikum Niederrhein (EJK), which has four locations in Duisburg, Oberhausen and Dinslaken. Aiming to boost efficiency and continuity of service, EJK wanted to consolidate its different radiological systems into a single, fully digitized solution. Digital Medics planned to create a private medical imaging cloud, accessible by all of EJK's healthcare professionals from anywhere onsite.

Choosing IBM solutions

Digital Medics selected the IBM BladeCenter HS22 server as its platform of choice due to its unparalleled flexibility and performance. At EJK, the company deployed two IBM BladeCenter HS22 blade servers running the Red Hat Linux Enterprise operating system at the clinic's computing center located at the Heart Center in Duisburg. Acting as a 'nerve center' for Digital Medics' innovative radiological imaging system, the unique design and in-built versatility of the IBM BladeCenter technology optimizes performance of the software.

An IBM System Storage DS8300 disk system was implemented, with data archived to IBM Tape Storage TS3100, introducing multiple layers of failover for added resiliency. The IBM System Storage DS8300 offers integrated virtualization capabilities based on leveraging logical storage volumes. As a result, Digital Medics' clients can benefit from the efficiencies of consolidation, without sacrificing the ability to tailor resources to the workload.

Exceptional scalability

Digital Medics recommends that all clients who run the company's solution select IBM technology. Since demand for medical images varies unpredictably along with the number of diagnoses per year, it is impossible to predict the capacity required, making the easy scalability of IBM BladeCenter technology a major selling point. For example, during the project with EJK, Digital Medics was able to quickly and easily expand capacity three times by simply plugging in an additional blade server or an additional storage bay.

Solution Components

Software

- IBM® Tivoli® Storage Manager
- Red Hat® Enterprise Linux®

Servers

- IBM BladeCenter® HS22
- IBM System Storage® DS8300
- IBM System Storage TS3100

The blade servers are connected to cost-effective Lenovo x86 servers in each of the hospital sites, housing a local cache for images. The complexity of migrating data from one site to another is hidden from the user and triggered on-demand, whenever a user requests the data at the frontend. Each site works mainly on its local cache, where copies of recently archived or accessed studies are stored in Digital Medics' proprietary compression format.

Data that cannot be found by the local cache is transparently requested from the main server at the computing center. This request is then locally forwarded to the IBM Tivoli Storage Manger middleware, which automatically handles local migration from tape archive to local discs, if necessary.

IBM Tivoli Storage Manager supports advanced service management capabilities, masking the complexity of the underlying infrastructure and automating resource configuration. This ensures easy management for solution providers like Digital Medics, enabling them to focus on core competencies with the tools to work most effectively at their fingerprints.

By offering healthcare professionals at EJK access to a private medical imaging cloud, the clinical center benefits from increased efficiency and consistency of operations across all locations.

Contributing to higher-quality patient care

The IBM solution drives rapid access to high-resolution medical images, helping healthcare professionals increase the quality of diagnoses. By providing access to a greater range of medical images at a higher resolution, physicians are given the tools they need to make more accurate diagnoses and best select a treatment plan for patients.

Diagnosing patients faster and more accurately leads to shorter waiting times and faster recovery. Both these factors contribute to shorter patient stays, enabling doctors to treat more people, cutting down on the cost of treating each patient while enhancing the quality of treatment. As a result, patients are more satisfied and the hospital can do more with limited resources.

Hospitals which are replacing printer-based systems with the Digital Medics visualization system make even larger savings – with EJK estimating it will save up to 300,000 EURO a year as a result of the project. Specifically, advanced data compression and automated storage management cut data storage costs at EJK by more than 50 percent.

For more information

To learn more about smarter computing from IBM and how we can help you integrate, automate, protect and transform your IT, contact your IBM sales representative or IBM business partner, or visit: **ibm.com**/smartercomputing

IBM Deutschland GmbH
71137 Ehningen
Deutschland
ibm.com/de

IBM Österreich
Obere Donaustrasse 95
1020 Wien
ibm.com/at

IBM Schweiz
Vulkanstrasse 106
8010 Zürich
ibm.com/ch

Produced in Germany
January 2012

BLC03083-DEEN-00