

Available thesis topics

At Data Ductus we are always looking for bright minds to help us explore IT. These are the areas we would like to focus on right now. If you can help us and would like to do your thesis at one of Sweden's leading IT consultants, then get in touch. If you have amazing idea or project that's not listed, we are open to suggestion, but you need to wow our experts.

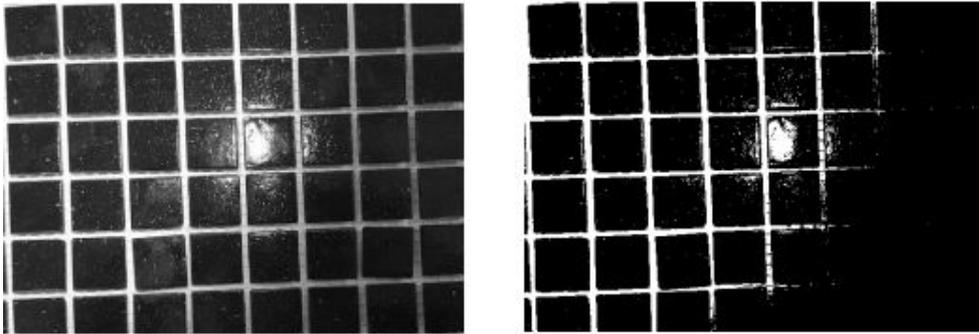
Send your application to degree@dataductus.se.

List of contents

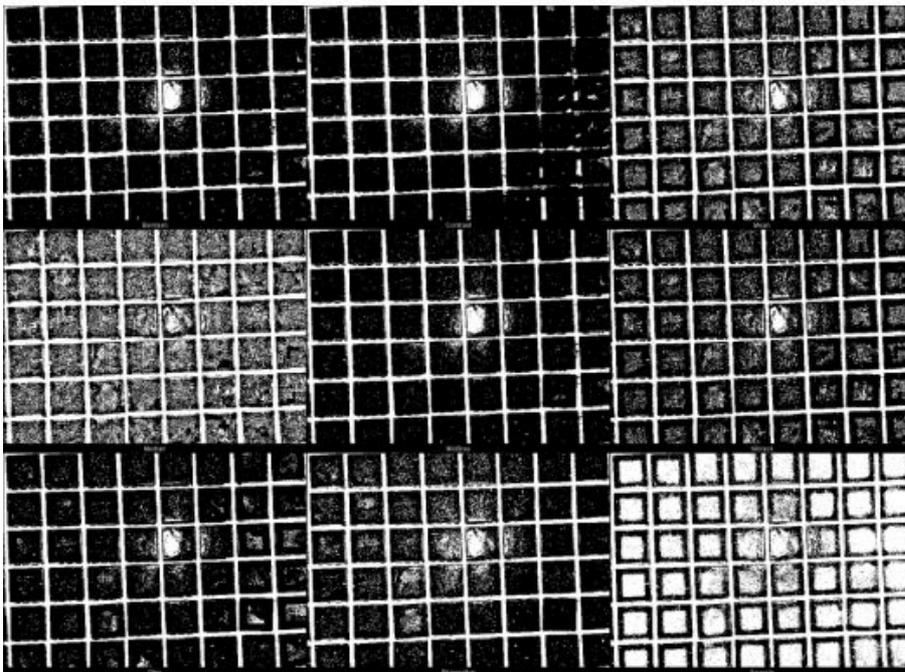
1	Implementation and evaluation of adaptive thresholding methods for robust object segmentation	2
2	Implementation and evaluation of registration methods for multimodal image data	3
3	Docker with Kubernetes in a Hybrid Cloud	4
4	Secure Delivery System using Blockchain Technology	6
5	Near real-time analytics platform	8
6	Mesos Karaf OSGi Framework	9
7	Eclipse YANG Design Surface	10
8	Web of Information REST Protocol – OData v4	11
9	Sharepoint Contest Site – Mobile	12
10	SharePoint Security Module	13
11	Network device configuration simulator	14

1 IMPLEMENTATION AND EVALUATION OF ADAPTIVE THRESHOLDING METHODS FOR ROBUST OBJECT SEGMENTATION

Uneven illumination is a challenge in most machine vision application and poses a problem for robust object segmentation.



Several methods based on adaptive thresholding have been developed. However, some are very sensitive to parameter tuning and other only perform well for certain types of images.

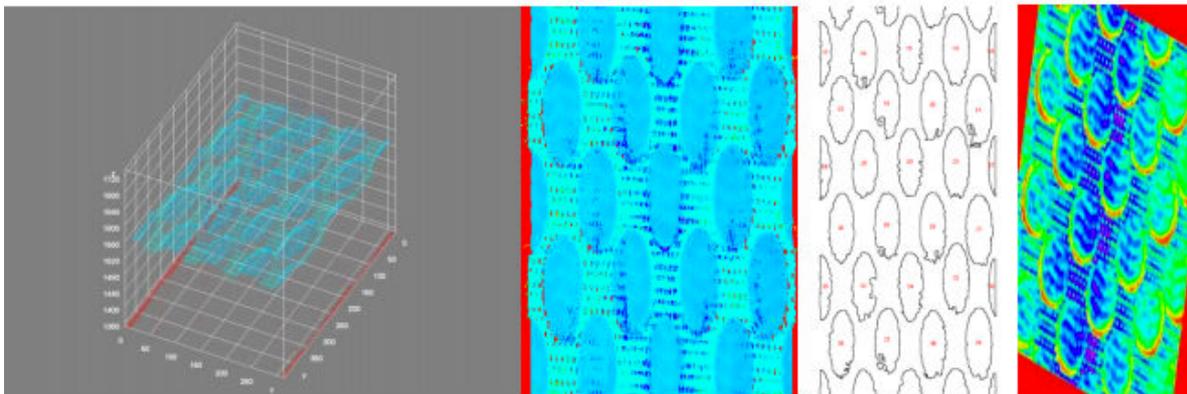


In this project some of these methods will be implemented and evaluated on relevant image data from industrial applications. Implementation of algorithms will be done primarily in Python and C# and use third party libraries such as OpenCV.

Reference: martin.simonsson@dataductus.se

2 IMPLEMENTATION AND EVALUATION OF REGISTRATION METHODS FOR MULTIMODAL IMAGE DATA

With an increasing number of cost effective sensor types such as thermal, TOF (Time of Flight) and laser scanners the industry request more advanced measurement setups with multimodal image data. Since several sensors are used to measure the same object there is a need to register the data, so distance, thermal and intensity data can be superimposed for the same point on the object.



The challenge is to accommodate for 2D and 3D-data from several sensors with different geometries and resolution. In this project several registration methods for 2D and 3D-data will be implemented and evaluated on relevant image data from industrial applications. Implementation of algorithms will be done primarily in Python and C# and use third party libraries such as OpenCV.

Reference: martin.simonsson@dataductus.se

3 DOCKER WITH KUBERNETES IN A HYBRID CLOUD

At Data Ductus we have a set of “Legacy” applications alongside with new applications running on virtual machines. We also use a cloud infra structure for other applications and services. This is a common scenario that many companies find themselves in. A virtual environment gives some sense of control and manageability. However, it often leads to waste CPU, Memory, and disk overhead. It has no means of splitting up existing applications into more granular services and fully utilize the machines.

Here’s where container technology comes into play and the most democratized is Docker (www.docker.com). This allows to produce services and place those into containers and stich those together into applications. It is also, in many cases, possible to break apart legacy applications and dockerize those. This gives the operations a finer granular control of how the application is deployed in order to improve performance, CPU and memory utilization. In addition, it enables moving the services around, from backend to cloud, and cloud to cloud.

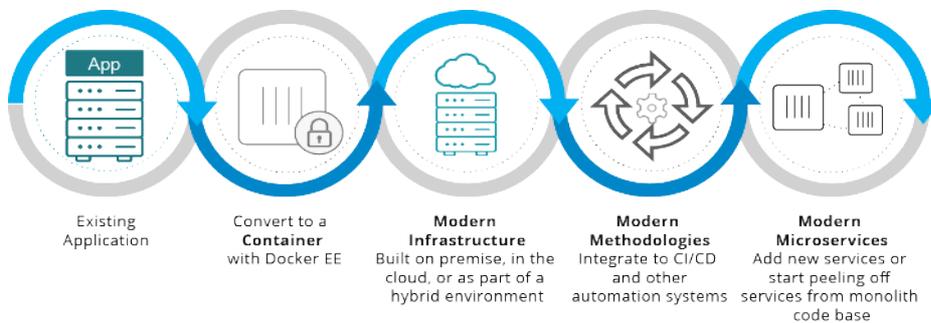


Figure 1 - Docker MTA Program

Thus, it is possible to setup a virtual network that spans from backend to different cloud providers, enabling a hybrid cloud solution. This opens up a wide range of possibilities to use services in the cloud to augment the existing services such as image recognition, AI based analysis on economy system data etc.

Docker now comes, out of the box (in beta), a full integration with Kubernetes (kubernetes.io) and thus allows for management automatic, policy driven scheduling of containers both backend and cloud environment as it would be one data center.



Figure 2 - Slide from DockerCon EU 2017 CPH

The work is to evolve a method to convert, break apart, legacy applications into docker container services. Make decision plans what goes where; backend or cloud. Define how new applications shall be brought into the containerized data center from third party and internal developed ones. Define a method to bring in those services into the scheduler / management system (Kubernetes). Also define with policies shall be applied on which type of applications / services.

The end goal of the method is to be able to in a fast manner, consistent time, bring in new or old applications into the dockerized data center. The result should be automated as much as possible so it would not need many persons to run the complete datacenter since almost everything shall be automated.

In order to prove the method, convert two legacy applications, use and extend existing CI / CD chain for inhouse developed solution into the environment. Setup and define (develop if necessary) policies to automate scheduling of containers. Test upgrades with zero downtime. The solution shall be as a hybrid cloud data center. Main KPIs are:

Reduced Infrastructure

Reduced Personnel

Reduced amount of Failures

Increase in agility

Increase in performance (when needed)

Increase in business opportunities

The project shall be open source, including documentation and setup under data ductus GitHub organization repository. We prefer that the project is released using docker and especially docker compose so it is possible to run the system by simply doing a docker-compose up.

Reference: mario.toffia@dataductus.se

4 SECURE DELIVERY SYSTEM USING BLOCKCHAIN TECHNOLOGY

Blockchain technology has exploded in 2017. Especially now when Ethereum (www.entethalliance.org) is stabilizing and improving such as privacy and speed. Ethereum is the vNext of bitcoin (www.bitcoin.org) whereas the latter is a currency (at the time of writing; 1 BTC costs around 5.000 EUR).

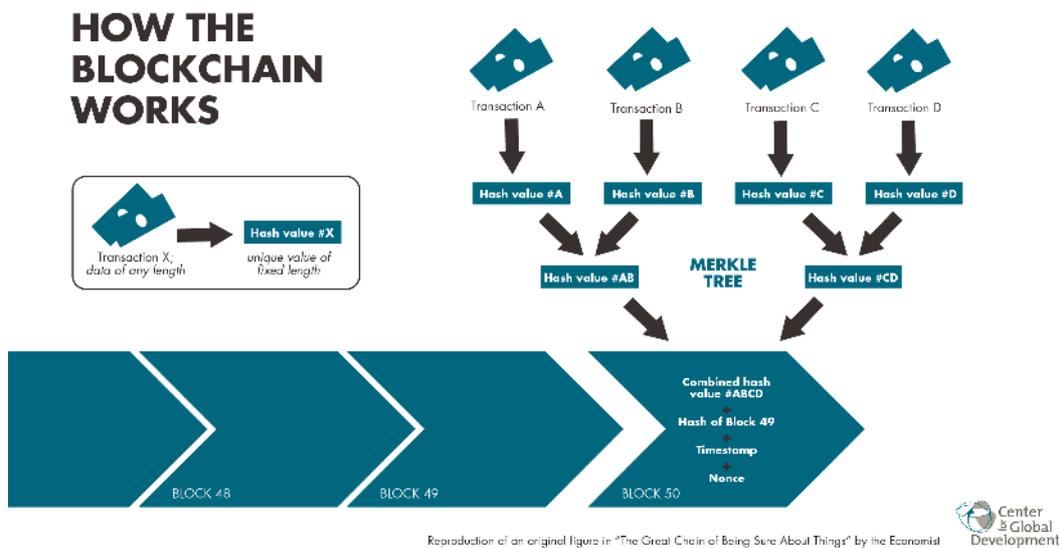
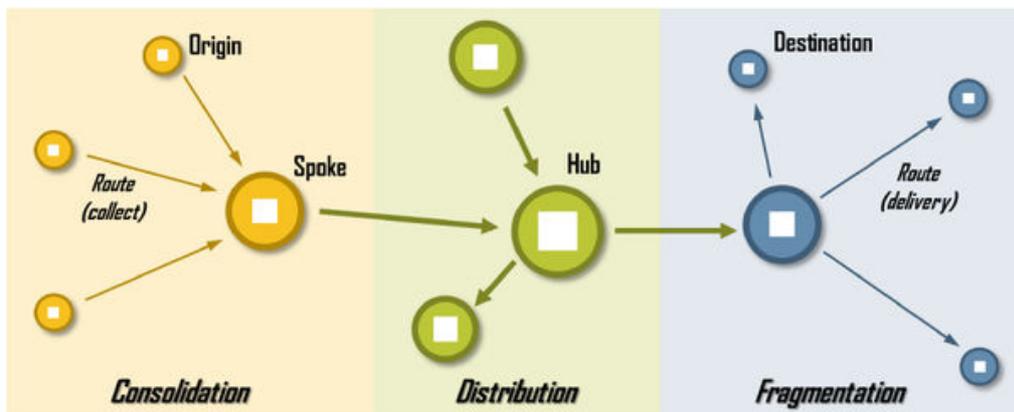


Figure 3 - How the Blockchain is Constituted

Ethereum supports something called smart contracts that are written in a language called solidity. There are good tooling around implementing smart contracts and run a Ethereum network (consortium). Smart contracts are executed during the mining process and when consensus in the network is reached the outcome is agreed upon. This can be used for anything such as commit the withdrawal of money, a position and a temperature is embedded into the contract etc.

The key thing is that this system is a distributed ledger and is therefore hard to compromise. Thus, applications that want to achieve mutual trust but not need to trust the individual parties is a good target.

This work is about emulating the Blocket secure package. Where two parties have decided to trade FIAT money for some goods. When the agreed amount of money and an insurance of the goods quality from the seller side the seller sends this package. The buyer picks up the package and have 24 hours to confirm that the goods in the shape agreed upon. Blocket will act as a broker and keep the money that the buyer has paid and transfer to seller after 24 hours. If buyer disagrees, s/he returns the package within 24 hours and the Blocket will transfer the money to the buyer.



In this work, we wish to eliminate the third party, instead make use of smart contracts that is a currency itself that will be delivered to seller's wallet when buyer do not return the package and automatically reverts the transaction when buyer do return the package. The package delivery company do get paid in each case, but after the result of the transaction outcome.

Since package management from, in transit, reception may be subject to damage. The package shall be instrumented accordingly, e.g. temperature sensor, shake / acceleration sensor. All shall deliver data into the running contract transaction. This allows e.g. the buyer to check if this package has been mistreated of the quality parameter set when buying the goods (e.g. temperature over a certain value for goods needs to be cooled). All data should be combined with location tracking such as GPS or RFID based system.

There are different actors in this system all of which needs their information, but others should be obfuscated. Therefore, encryption of the payload is needed where different, and overlapping, data needs to be concealed and recovered by correct actor.

The task is to write a whitepaper on how such system shall work and be implemented using A ERC20 compatible Ethereum token and smart contract. Including the necessary equipment and actors / roles in the system. This token is both the currency and the contract. Realize the system as a simulation to prove the whitepaper. Hardware sensor may be simulated such as a temperature sensor, distribution hub, truck or seller etc.

The project shall be open source, including documentation and setup under data ductus GitHub organization repository. We prefer that the project is released using docker and especially docker compose so it is possible to run the system by simply doing a docker-compose up.

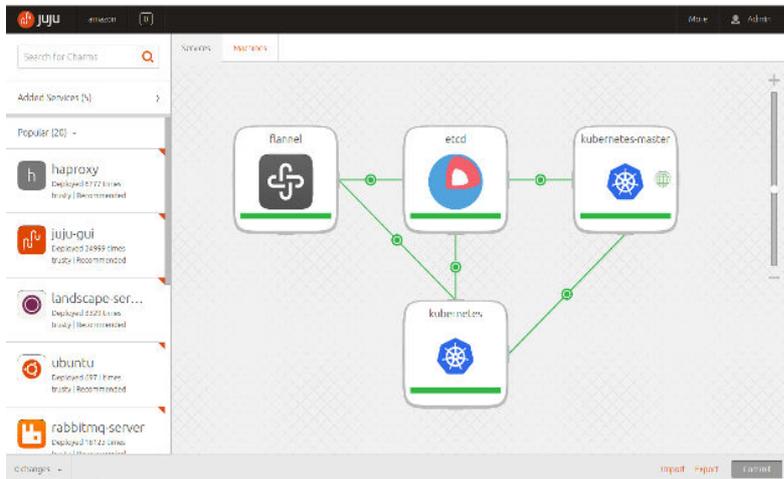
Reference: mario.toffia@dataductus.se

5 NEAR REAL-TIME ANALYTICS PLATFORM

Today data is collected from a diversified set of systems ranging from platforms, databases, http logs to small devices out in the field. This needs to be analysed both in batch analysis (heavy analytics during e.g. the nights) but parts of it needs to be addressed near real-time. In extreme cases real-time is also needed in order for systems to perform its work.

In order to handle varying loads on the system it needs to be adaptive and schedule more analytics tasks during peak hours and reduce the amount of analytics tasks during off-peak hours. The analysed data shall be stored for reporting and querying as well as a stream of data.

This job is about setting up an environment that is capable of collecting data from apache Kafka (kafka.apache.org) and process a configurable amount of topics data in apache Spark stream (spark.apache.org). It shall store the resulting data onto HDFS clustered file-system (hadoop.apache.org) and re-publish the analysed data onto a configurable Kafka topic.



In order to handle scheduling of analytics tasks apache Mesos (mesos.apache.org) should be used and optionally be configured and deployed using Kubernetes (kubernetes.io).

We also would like to have a simple web running e.g. node that presents graphs from the resulting analytics on the HDFS or in (near) real-time subscribing to a Kafka Topic.

Suggested rendering lib is d3s.js.

All technologies are up to suggestion and thus may be replaced if other is preferred!

Reference: mario.toffia@dataductus.se



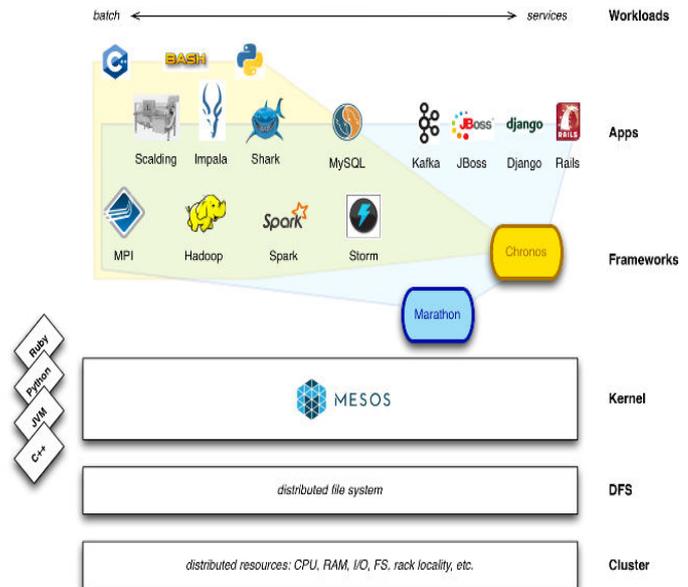
6 MESOS KARAF OSGI FRAMEWORK

Apache Mesos is a way of seeing the datacentre as a resource layer where the Linux kernel can be seen as laid on top of all machines. It handles task scheduling and execution by resource allocations. Several frameworks (scheduler, executor) exists for e.g. MySQL, Spark, Hadoop to name a few.

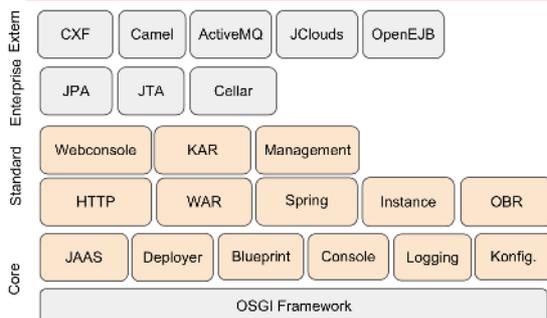
When running services under a OSGi system such as Felix those are isolated into the OSGi container and is not schedulable by default. We wish to be able to e.g. load balance, schedule more services under peak hour and reduce during off-peak hours. Since we use Karaf as OSGi container we would like to use apache Karaf (karaf.apache.org) as the service container. Karaf can in it's turn use apache Felix or equinox as the real OSGi container.

The scheduling of services should be done by e.g. creating new instances of Karaf and deploy services on those or reuse services that have been un-deployed from Karaf to re-deploy them again and so on.

The scheduler shall use OSGi service as the scheduled task but it will need to schedule Karaf containers in order to un-/schedule services (bundles) on Karaf. In order to simplify the task, the services dependencies to other services needs to be calculated but a definition of a core framework (including base services) is always deployed when a instance of Karaf is executed.



Apache Karaf



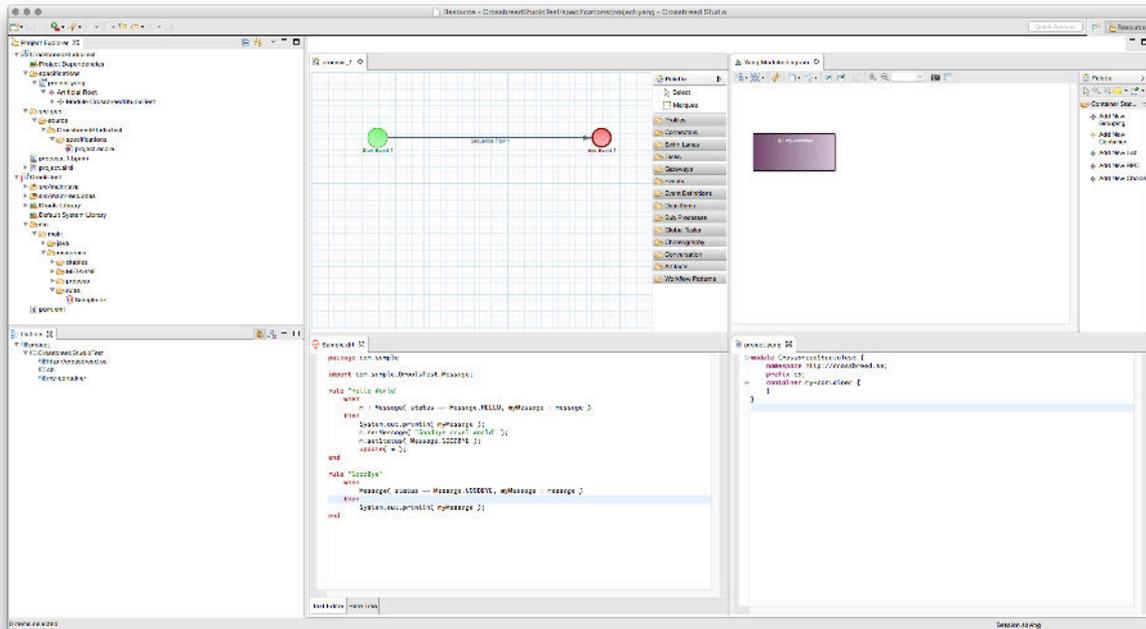
Two types of services exist, one that is exposed at frontend using rest using java standard JAXB annotations. The other type is "backend" service that only needs to be deployed. The frontend service needs to be exposed towards a Ngix proxy. This Ngix proxy is the tyk.io gateway and shall dynamically be provisioned (using the tyk.io rest API). The result should be shown in a web-console that uses two services one REST service and it shall use one backend service where it shall perform its tasks while scheduling more and less services without interruption.

We also suggest to open source this project on github as a framework for the Mesos apache project.

Reference: mario.toffia@dataductus.se

7 ECLIPSE YANG DESIGN SURFACE

YANG is widely used in the telecom and IT industry and is rapidly spreading to the Internet of Things world. Yang is not alike UML or other modelling language, however similarities exists, instead it is much more dynamic in it's nature since it has grown out of how telecom and IT manages their network of devices.



Today there exist a simple eclipse design surface for YANG modelling language. This surface is not optimal for "standard" users. The task is to create a new design surface of which is much more standard user centric (not a hard-core technician). It shall be implemented in the latest eclipse platform, Eclipse Mars. The current implementation is in tandem with a ASCII editor that generates ecore of which we use to do code generation among other things. This is used within a IoT platform; crossbreed.

Our suggestion is to use the eclipse Sirius project to realize the design surface, but it is up to the implementer to choose the framework. However, the current data is exposed through the eclipse standard ecore model of which the YANG editor exposes. What you design shall be visible in the ASCII editor and vice versa. As long as the design surface uses ecore, this process is automatic.

It shall support:

- Definitions of containers (about analogue to class)
 - Inline editing of properties and its metadata
 - Inline editing of functions and its arguments
- References
- Inheritance and Ad-hoc (augments) semantics
- Documentation
- etc.

Reference: mario.toffia@dataductus.se

8 WEB OF INFORMATION REST PROTOCOL – ODATA V4

OData v4 is the latest version of the now Oasis standard for the OData protocol. This protocol is a REST protocol for data CRUD operations along with aggregation, and other “database” operations. However, it is much more flexible than that. The neat thing with OData is that many vendors have support out of the box for OData such as Excel, Jasper Reports, Linq Pad etc.



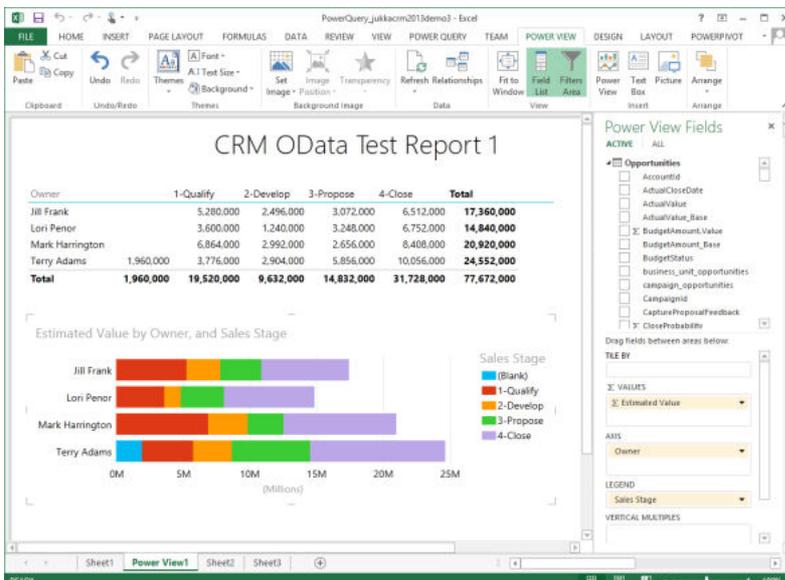
In the crossbreed platform we have OData v2 support and wishes to do a proof of concept of going “all-in” into OData v4. This means that we need to add framework capabilities to support all the

new features in OData v4. We need to extend the support to aid the integration (apache Camel), Workflows/Rules engines, NoSQL data sources such as Couchbase (based on apache CouchDB and memcached) and SQL, data sources as Mariah DB. Since we use a mix of technologies we need to have built in functionality to be able for one OData request use multiple technologies to produce the response.

When a framework implementation is done, a test shot on either a existing home automation service or a custom built to provide proof of concept should be done. A simple web or adapt a existing native mobile app to call the service(s) needs to be done to prove the point.

We wish this job to be performed in our upcoming v2 of the crossbreed platform, but not a requirement. However, if in v2, it needs to be OSGi friendly as well since it is a pure OSGi implementation.

Reference: mario.toffia@dataductus.se



9 SHAREPOINT CONTEST SITE – MOBILE

The Ductus workspace is at core a SharePoint 2013 solution. We like to engage personnel at Data Ductus and possibly include customers in campaigns and other activities. Therefore, we need a SharePoint based system that we define e.g. contests that can be executed during a specified set of time. We also need to provide access to either data, in custom iOS/Android app, or mobile version of the site where the participation occurs. In the mobile application it shall be possible to integrate e.g. step counters or other plug-ins to deliver “data” needed to fulfil the contest.

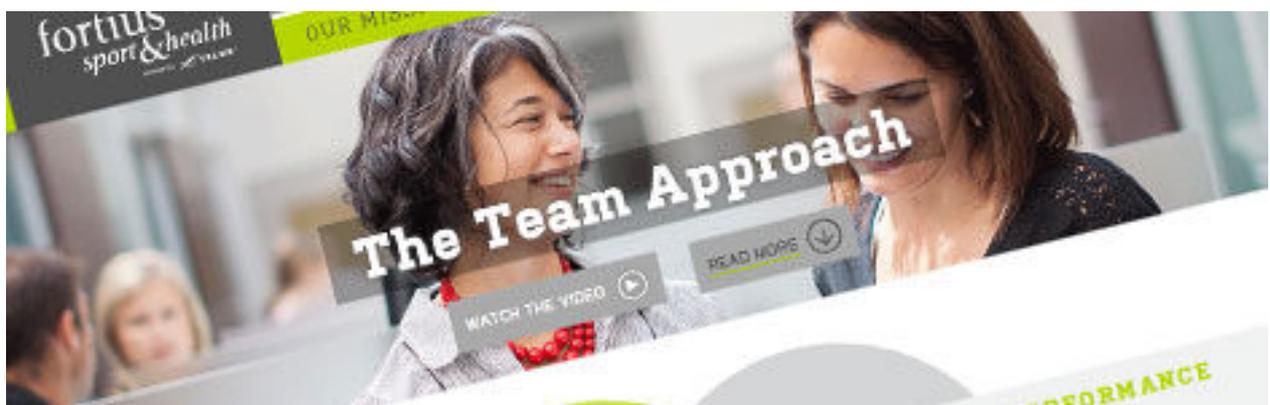


The assignment is to define such a site and a solution for the mobile apps and how they shall communicate. As proof of concept implement a health initiative contest where the participants may fill in data about their activities where they get a point for each day they fulfil the minimum requirements for a single day. At some “checkpoints” the points are calculated. If certain number of points is achieved and a specified price is won. At the end of the contest, a lottery takes place where the number of points increases the chance to win a price.

We’d like during the contest have a news feed to the mobile clients manually edited for communication with the participants – such as Ductus News Flash, news etc. in a html as layout.

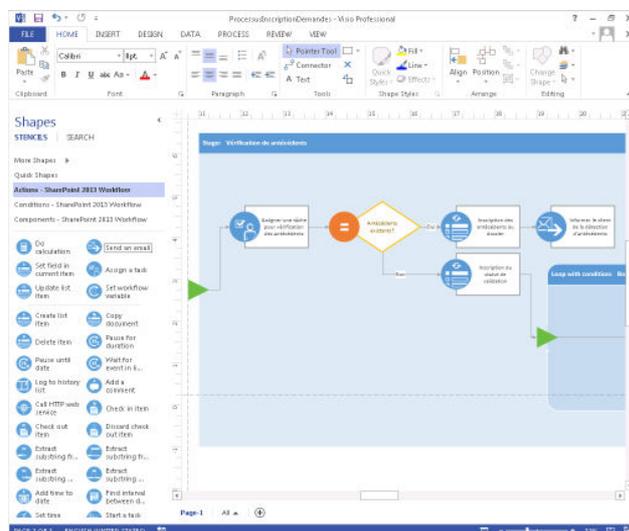
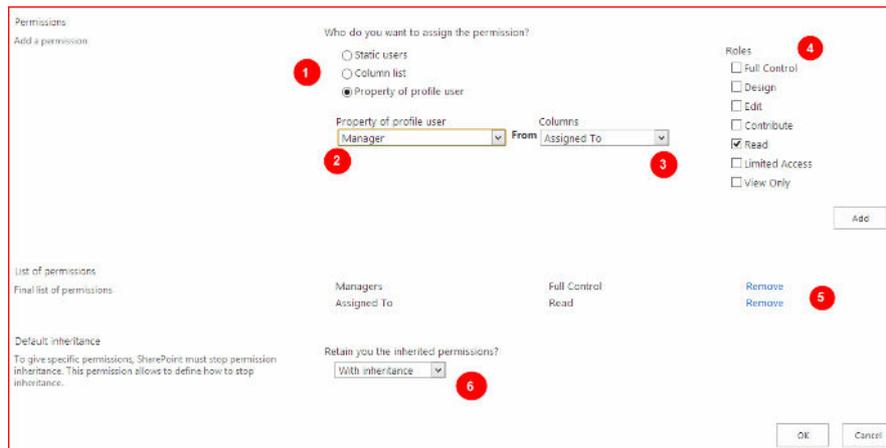
The solution shall be formed as a SharePoint app that can be installed on any SharePoint installation through the App gallery.

Reference: mario.toffia@dataductus.se



10 SHAREPOINT SECURITY MODULE

In SharePoint it is possible to set security on a granular level e.g. a document or a task. However, this is quite cumbersome and error prone. There are efforts to do this in a rules based manner or simplified way. An open source project "SharePoint Rule Permissions" (<http://permissionmanagement.codeplex.com/>) to do such.



We would like to commit to the project, or clone it to GitHub, and finish it to a release. In addition, add more advanced rules capabilities to set rules. This includes document properties of the document, the session user etc.

It also shall have the ability to query underlying data sources such as Active Directory, Databases or a REST service to provide with data to the rule evaluation.

Support to run custom workflows in the rule evaluation shall also be supported. The design of the rules shall be within the web interface and do not require anything else than a standard web browser, except for the creation of workflows.

Everything shall be released as open source in order to evolve this over time. It shall function for SharePoint 2013 and coming 2016.

Reference: mario.toffia@dataductus.se

11 NETWORK DEVICE CONFIGURATION SIMULATOR

When developing network configuration/orchestration applications it is often convenient to run basic and continuous integration tests towards simulated rather than actual hardware devices. In this case we consider devices with a configuration model specified in the YANG data modelling language [1].

Goal:

A simulator for the configuration interface of network devices, implemented in any language, with the following properties (in order of importance):

1. Runs as a process on linux and mac os x (optionally on ms windows).
2. Has a light-weight in-memory database corresponding to a selectable YANG-model [1].
3. Exposes an interface over SSH to update and read the corresponding database model. Choose to implement the first or both of the interface types:
 - 3.1 NETCONF [2]
 - 3.2 (Optional) Command line interface with (e.g. "router-vendor"-like) commands corresponding to the chosen YANG-model.
4. Should be "quick" to start/stop.
5. (Optional) Ability to save/restore the database to/from disk.
6. (Optional) Scriptable responses, e.g: bind a user-defined script (either external executable or embedded scripting language) to generate the output of a particular command.

[1]. <https://tools.ietf.org/html/rfc6020>

[2]. <https://tools.ietf.org/html/rfc6241>

Reference: johan.ehnmark@dataductus.se