Available thesis topics

Data Ductus is a global, full-service software engineering consultancy company. We plan, build and manage service automation and orchestration in areas where disruptive change is taking place.

We offer a dynamic and open working environment with interesting and challenging projects where you get the opportunity to expand your skill-set, explore the latest IT innovations, and work on real-life solutions that transform companies. We believe in giving our consultants the freedom and the tools they need to succeed and develop.

We welcome all nationalities, ages and backgrounds and operate from ten offices on three continents.

A great way of getting to know us is by doing your degree project with us. Have a look at what we have to offer - or talk to us about your own idea!

Quick Facts About Data Ductus

Number of employees:	Number of offices:	Year founded:	Ownership:
230	10	1989	Privately held

EMEA

Data Ductus
931 31 Skellefteå
Sweden
+46 910-73 23 00

US

Data Ductus
201 Terry Street Unit 2A
Longmont, CO 80501
USA
+1 303 731 2197

APAC

Data Ductus
111 Somerset Road #16-06
Tripleone Somerset
SINGAPORE (238164)
REPUBLIC OF SINGAPORE
+ 65 8296 2834

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.

Submit your application on our website: www.dataductus.com/careers

List of contents

1 DOCKER WITH KUBERNETES IN A HYBRID CLOUD

Master Thesis - Skellefteå or Malmö

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.

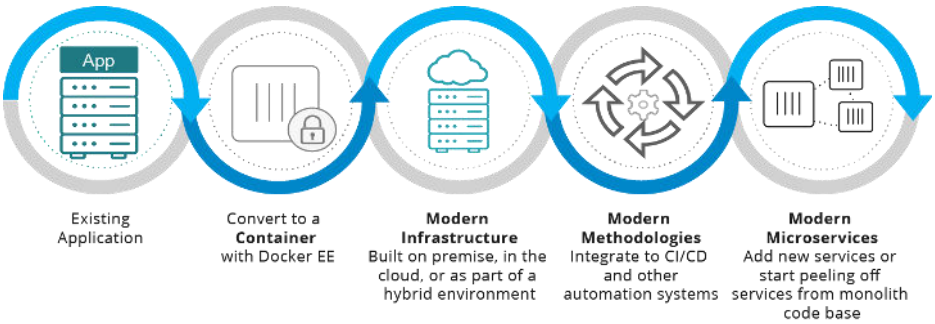


Figure 1 - Docker MTA Program

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

2 SECURE DELIVERY SYSTEM USING BLOCKCHAIN TECHNOLOGY

Master Thesis - Skellefteå or Malmö

Blockchain technology has exploded in 2017. Especially now when Etherium (www.entethalliance.org) is stabilizing and improving such as privacy and speed. Etherium 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).

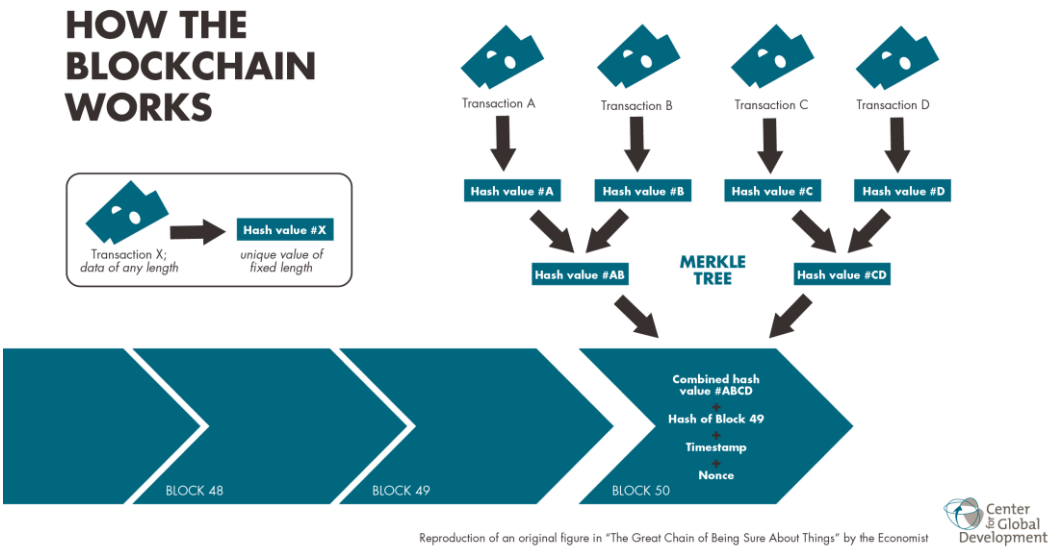
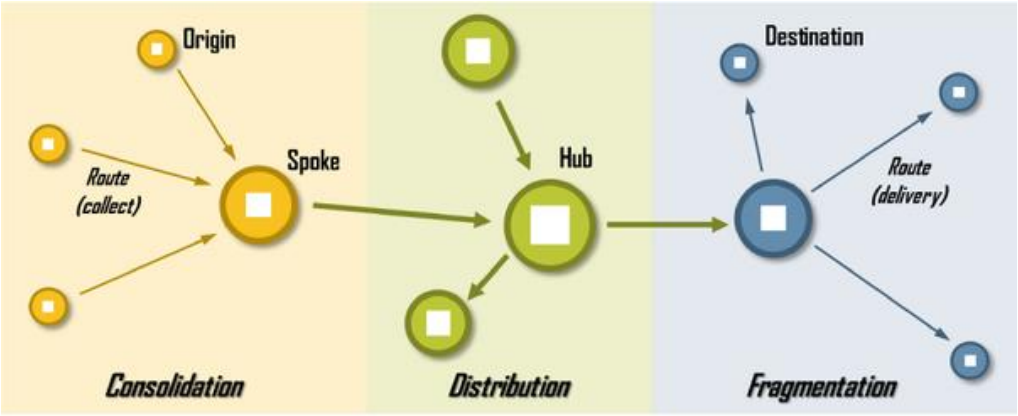


Figure 3 - How the Blockchain is Constituted

Etherium supports something called smart contracts that are written in a language called solidity. There are good tooling around implementing smart contracts and run a Etherium 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 Etherium 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

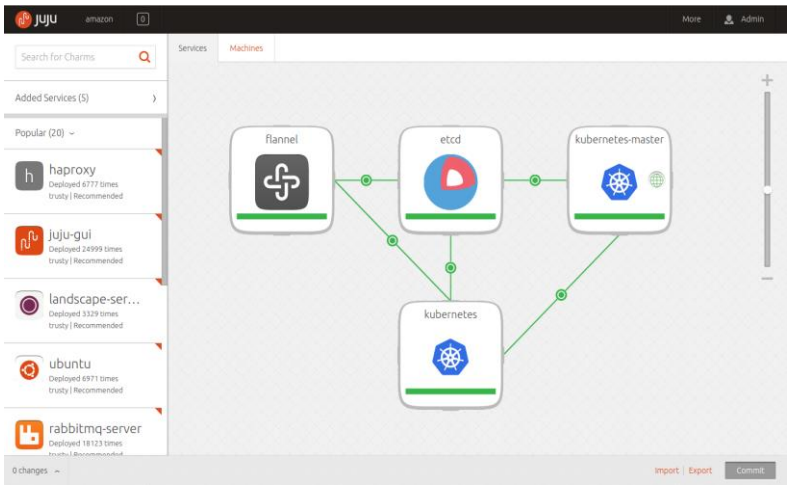
3 REAL-TIME ANALYTICS PLATFORM

Bachelor Thesis - Skellefteå or Malmö

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.

The system is first of all a real-time analytics system, secondary can be done from micro batches to batch analytics (e.g. during the night). 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 Flink (flink.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 different load of analytics, either apache Mesos (mesos.apache.org) can be used or use kubernetes with custom profiles (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 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!

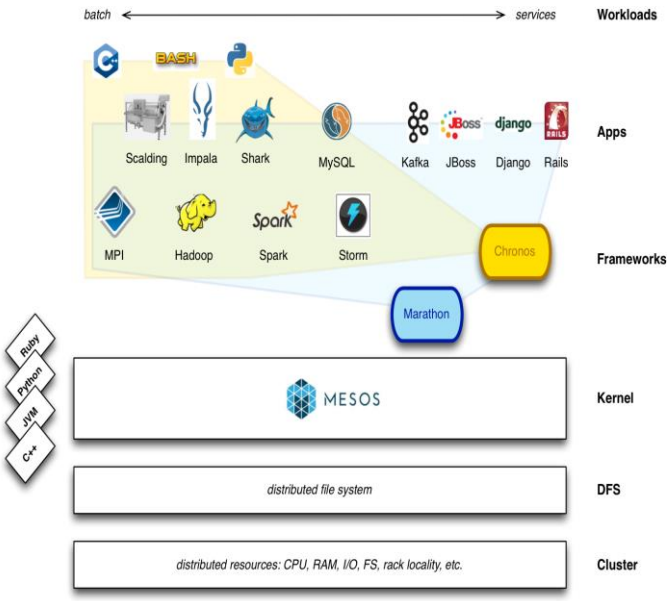
Reference: mario.toffia@dataductus.se

4 MESOS KARAF OSGI FRAMEWORK

Master Thesis - Skellefteå or Malmö

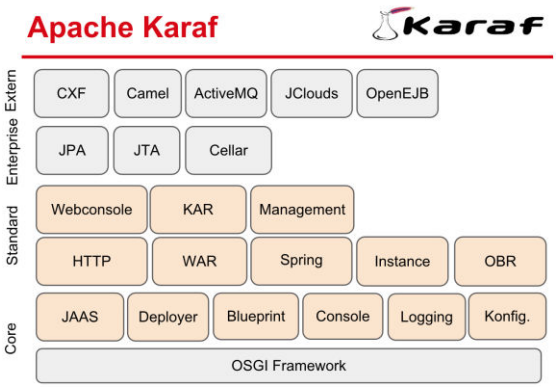
Apache Mesos is a way of seeing the datacentre as a resource layer where the Linux kernel can be seen as laid on top of all machines. It handles task scheduling and execution by resource allocations. Several frameworks (scheduler, executor) exists for e.g. MySQL, Spark, Hadoop to name a few.

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.



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.

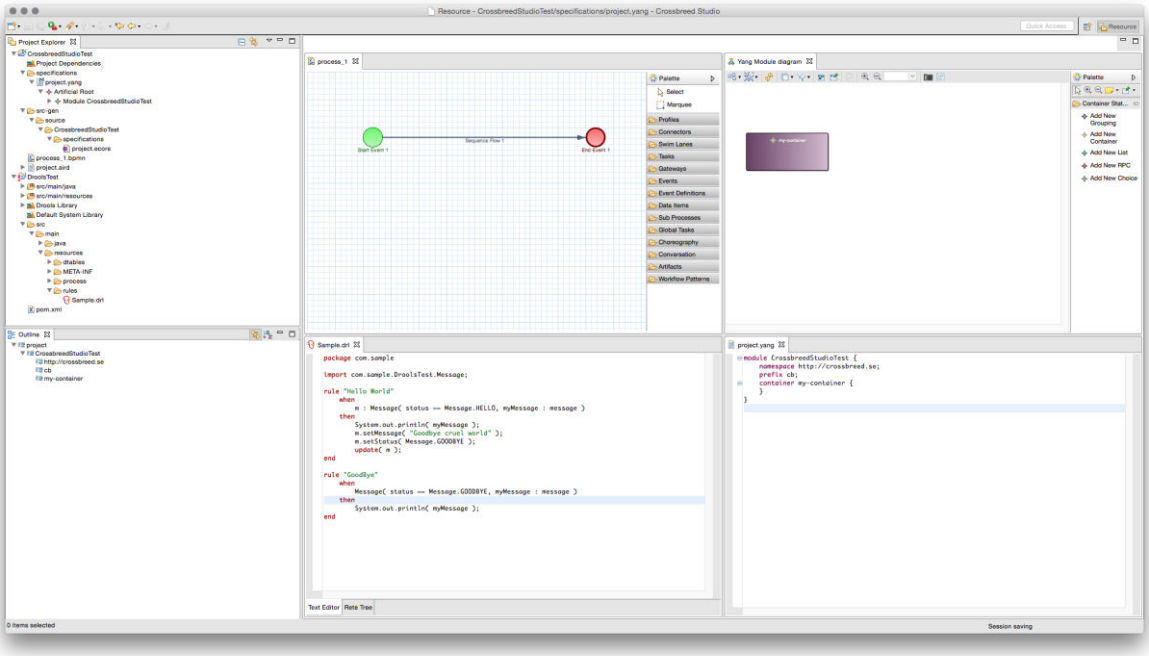
We also suggest to open source this project on github as a framework for the Mesos apache project.

Reference: mario.toffia@dataductus.se

5 ECLIPSE YANG DESIGN SURFACE

Bachelor Thesis - Skellefteå or Malmö

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

6 WEB OF INFORMATION REST PROTOCOL – ODATA V4

Bachelor Thesis - Skellefteå or Malmö

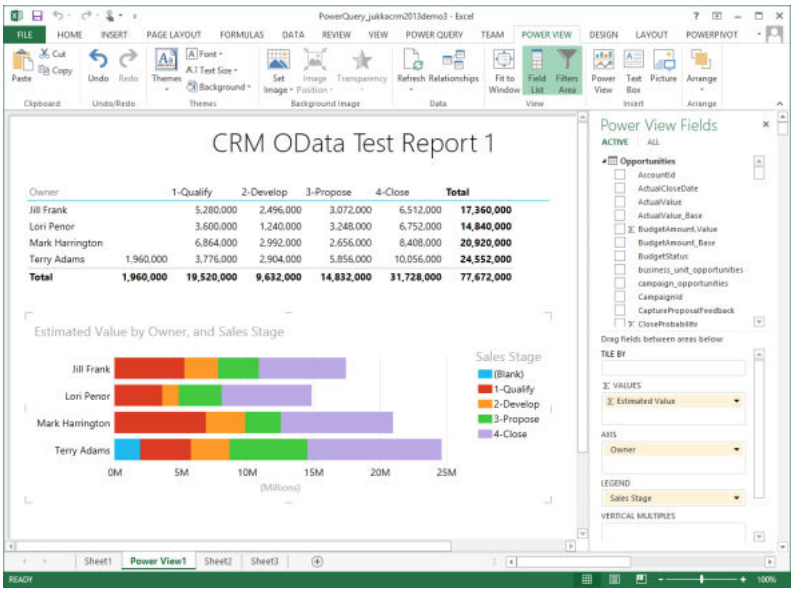
OData v4 is the latest version of the now Oasis standard for the OData protocol. This protocol is a REST protocol for data CRUD operations along with aggregation, and other "database" operations. However, it is much more flexible than that. The neat thing with OData is that many vendors have support 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.

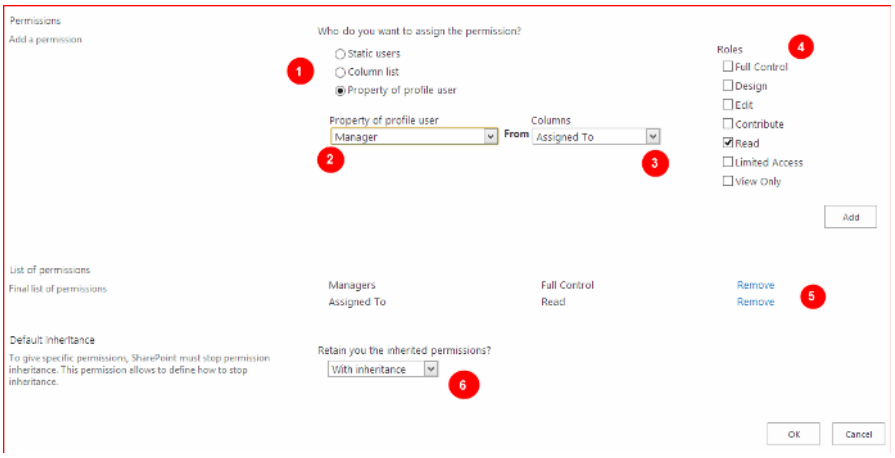
Reference: mario.toffia@dataductus.se

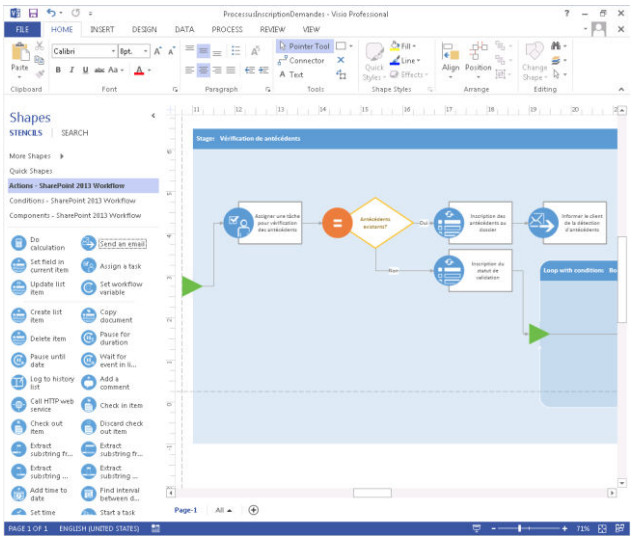


7 OFFICE 365 - SHAREPOINT SECURITY MODULE

Bachelor Thesis - Skellefteå or Malmö

In Office 365 (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 create such component for office 365 and publish it as open source. 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/logic apps/flows 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

8 EDGE CLOUD OPTIMIZATION

Master Thesis work - Luleå

Short Summary

This thesis work targets to find concepts for optimizing edge cloud workload distribution that can improve service characteristics and reduce cost of operating a distributed edge cloud network. The thesis work aims at defining a service model framework whereby different services can be modelled thus enabling a total workload optimization.

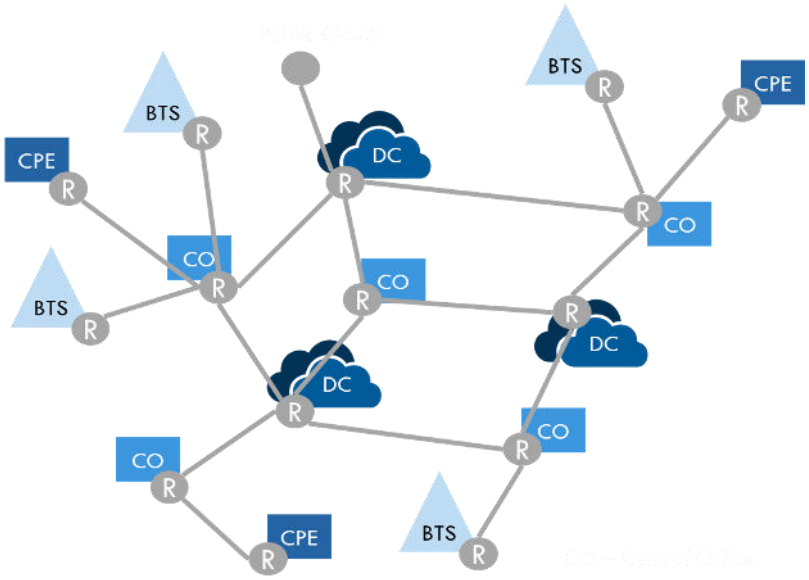
Introduction

Data Centers have for some years been a clear trend that address producing compute more cost efficient as a result of economies of scale.

But with emerging trends in communication and new applications such as Internet of Things, VR/AR, Industry Automation over Cloud etc the need for what is called Edge Computing emerge.

This essentially means that SW workloads need to be moved out in the network closer to the final consumers / companies.

Those concepts imply there is an optimization challenge in how to distribute various workloads between a set of compute nodes that are connected with each other.



Edge compute locations might be CPE (Customer Premise Equipment), Central Offices (i.e. operator switching sites), Mobile base stations or more regular data center locations.

Typical optimization challenges relate to the balance between cost and constraints like service latency, bandwidth and other similar QoS characteristics. The cost aspect is driven from the huge amount of data expected to be collected by various IoT devices as well as increasing video traffic over the network, which causes bottlenecks that can be avoided by distributing the workload in a more optimal way. Furthermore, we can expect that small data centers closer to network edge will be costlier to operate.

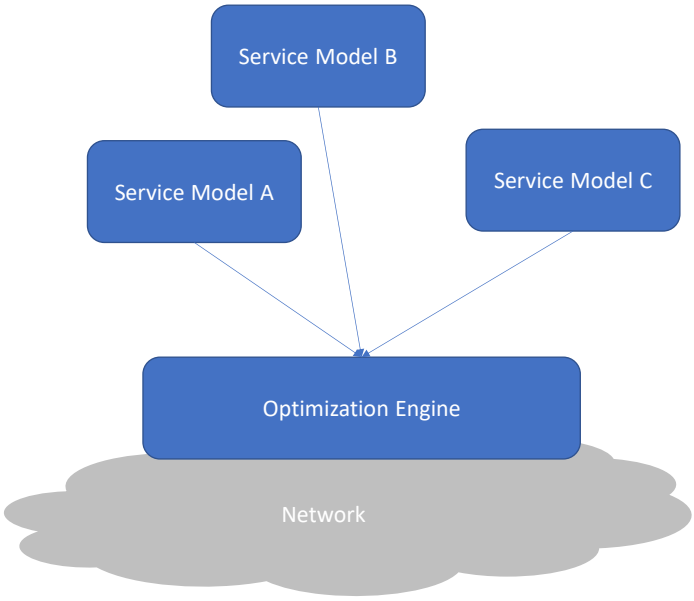
Intended work

Data Ductus want to explore ways by which to model different services that could run on a distributed cloud network. Such model needs to capture latency requirements, bandwidth requirements etc.

The model needs also to capture that the service is consisting of a flexible number of SW components that is supposed to be deployed within the network.

This implies that there is firstly a meta-level of the modelling concept as well as specific service models of different types.

A set of these models should be able to be fed into an optimization engine (e.g. MiniZinc), that can chew all models and arrive at an optimal solution.



Breif on Data Ductus

Data Ductus is a Swedish company with offices in the US and Asia. We specialize in Network Automation, IT Service Management and Internet of Things.

In our Network Automation business unit, we have over 40 customer references with leading operators, global enterprises in all parts of the world.

Reference: mats.eriksson@dataductus.se

9 CLOSED LOOP ORCHESTRATION

Bachelor thesis work - Stockholm

Automation of service provisioning is one of the hottest topics in the industry today. In this thesis you will get the opportunity to build a closed loop orchestration solution using one of the leading network service orchestration products on the market, Cisco's Network Service Orchestrator (NSO).

Network service orchestration is all about pushing configuration out into the network. The assumption is that the devices are reachable and that the configuration is valid. In reality there can be any number of issues that causes the orchestration to fail. In many cases manual operations must be done to recover from the error which is very contradicting since the goal of orchestration is that it should be fully automated.

Closed-loop systems use the system output and feed it back into the system again to adjust the result in case of errors. This could be used for orchestration to handle unexpected errors and misconfigurations.

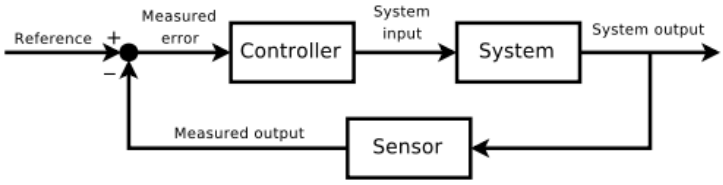


Figure 4: Feedback loop
Source: Wikipedia

Design and create a generic framework integrated to NSO for creating closed-loop solutions. The framework must be able to support some example scenarios. The solution should be designed so that AI and Machine Learning could be used to automatically identify and solve problems in future work.

Reference: mikael.mollberg@dataductus.se

10 VISUALIZATION AND AUTOMATIC FAILURE IDENTIFICATION OF NETWORK DEVICE CONFIGURATIONS

Bachelor thesis work - Stockholm

In the industry today, large Enterprises and Service Providers are struggling in finding the root-cause of a network problem. In this thesis you will get the opportunity to build a solution that can visualize the root-cause of the failure.

Network devices are continuously subject to change, especially in an orchestrated network solution. There are many alternatives out there when it comes to service assurance and ways to track the health of the network. It is fairly easy to notice when something is wrong.
Determining the cause of the problem, however, is a different story.

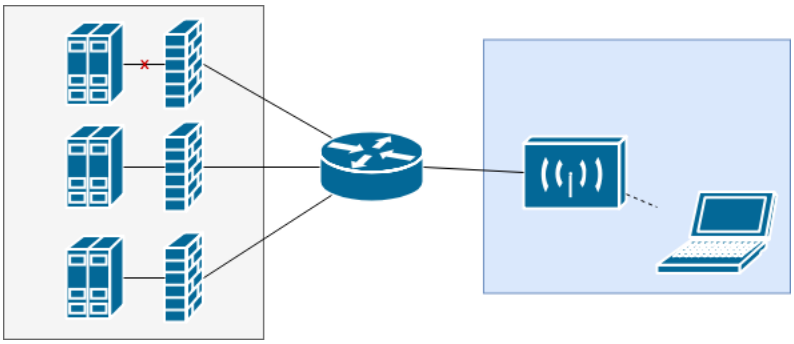


Figure 5: A misconfigured firewall caused the server to be unreachable

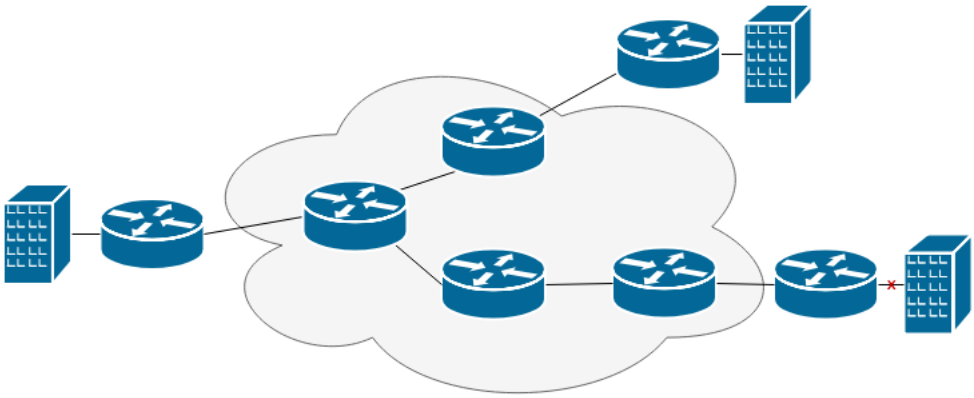


Figure 6: A branch office lost connection to the corporate VPN

Given the device configuration over time, identify means of visualizing the differences in the configuration to simplify the identification of the possible root cause of the problem.

Reference: mikael.mollberg@dataductus.se

11 DEVELOPING A CLOUD GOVERNANCE MODEL & TOOL FOR CIO:S

Bachelor thesis work - Stockholm

One thing that keeps Chief Information Officers (CIO:s) awake at night is the questions when and how to evolve from their existing on-premise data centres to hybrid/public clouds. The imminent introduction of Amazon Web Services and Google Cloud Platform services in Sweden will speed-up the migration of applications from on-premise data centres to public clouds.

In this thesis work you will develop a Model that supports the decision making process to decide where specific enterprise applications is best run. At least the following parameters should be supported in the Model:

- Price
- Speed of innovation
- Compliance to regulations (e.g. GDPR)
- Cyber-security
- Availability
- Latency
- Environmental impact

In the second step of this thesis work a Tool should be developed that implements the theoretical Model.

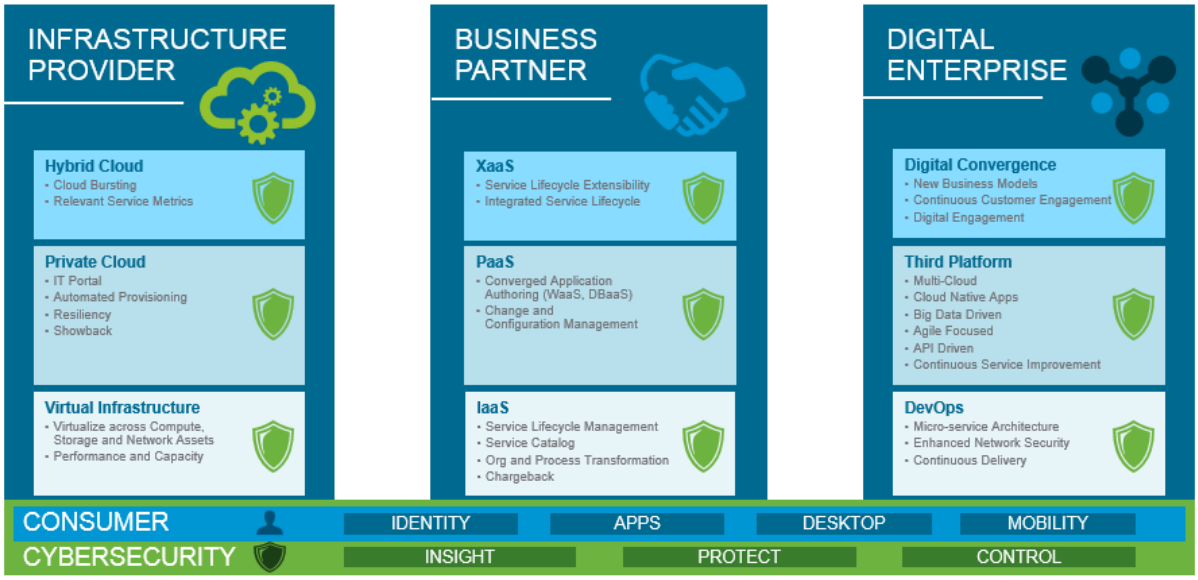


Figure 7: IT Value Model is a framework that support Enterprises in their Tailored Journey into the future. Source: VMware

The Tool should support the flexibility for a CIO to adjust the weight of the parameter to support their specific characteristics and business needs.

A few use cases supporting the CIO decision making process should also be developed in the Tool. For example, how the price change of public cloud service will impact the decision making.

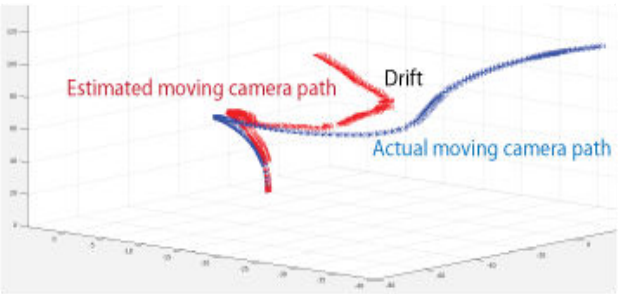
As the basis for the Tool we suggest that the product **VMware vRealize Business for Cloud** is used in order get access to market data, such as price and other parameters for public cloud services and on-premise alternatives.

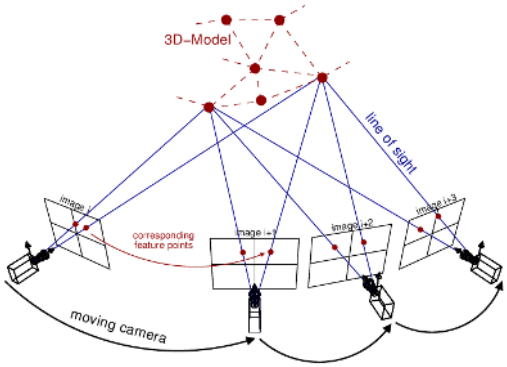
Reference: ingemar.haggstrom@dataductus.se

12 COMBINING POSE ESTIMATION FROM VIDEO AND IMU DATA TO IMPROVE SLAM

Master thesis work – Luleå/Skellefteå

Simultaneous localization and mapping (SLAM), is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. SLAM will always use several different types of sensors, and the powers and limits of various sensor types have been a major driver of new algorithms.





This proposal includes extracting pose estimation from streaming video and combining with data from inertial measurement unit (IMU). Implementation of algorithms will be done primarily in Python and C# and use third party libraries such as OpenCV.

Reference: martin.simonsson@dataductus.se

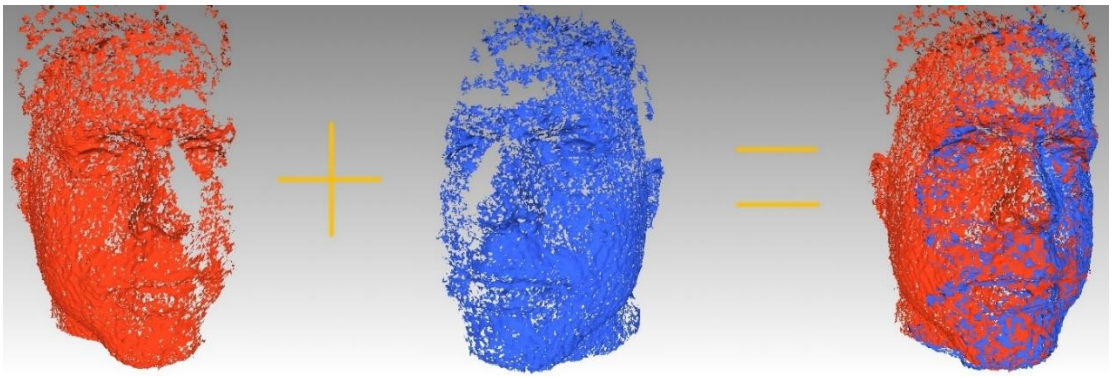
13 POINT CLOUD DATA REGISTRATION, MATCHING AND COMPARISON

Master thesis work – Luleå/Skellefteå

A point cloud is a set of data points in space. Point clouds are generally produced by 3D scanners, which measure many points on the external surfaces of objects around them. Point clouds are often aligned with 3D models or with other point clouds, a process known as point set registration.

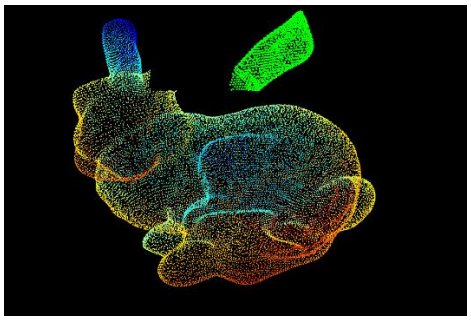
This proposal includes several aspects of working with point clouds and can be split into projects with different focus depending on interest.

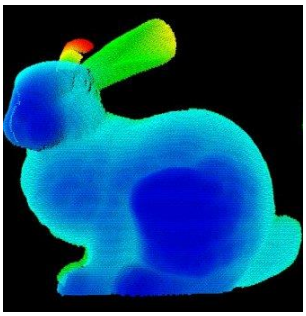
13.1 Automatic registration of scans to produce a complete model.



Techniques for generating point clouds often rely on a 2D optical system to gather data from a 3D object. This results in occlusion, where only a part of the object is visible. For a complete model several scans need to be combined. Methods for registration needs to be implemented and evaluated. Implementation of algorithms will be done primarily in Python and C# and use third party libraries such as OpenCV and Point Cloud Library.

13.2 Object matching and comparison to CAD-model





For quality control objects need to be identified in a point cloud and compared to a CAD-model. Deviations need to be measured and visualised. Implementation of algorithms will be done primarily in Python and C# and use third party libraries such as OpenCV and Point Cloud Library.

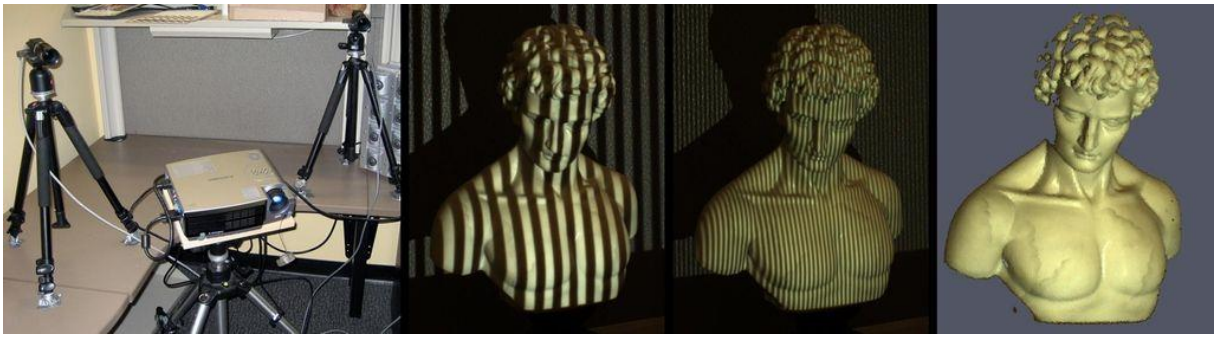
Reference: martin.simonsson@dataductus.se

14 POINT CLOUD DATA GENERATION WITH STEREO CAMERAS AND STRUCTURED LIGHT

Master thesis work – Luleå/Skellefteå

Stereovision is a technique that estimates 3D information of a scene from two or more cameras. Structured light can be projected to the scene to improve the geometric reconstruction.

This proposal includes setting up and calibrate stereo cameras to generate point clouds. Resolution and accuracy of the technique will be evaluated on several objects and in different environments. Structured light will then be added to the setup and the improvement in resolution and accuracy will be evaluated.



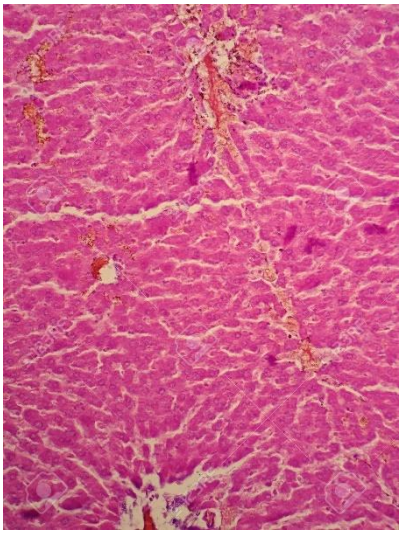
Implementation of algorithms will be done primarily in Python and C# and use third party libraries such as OpenCV, Point Cloud Library and 3DUnderworld.

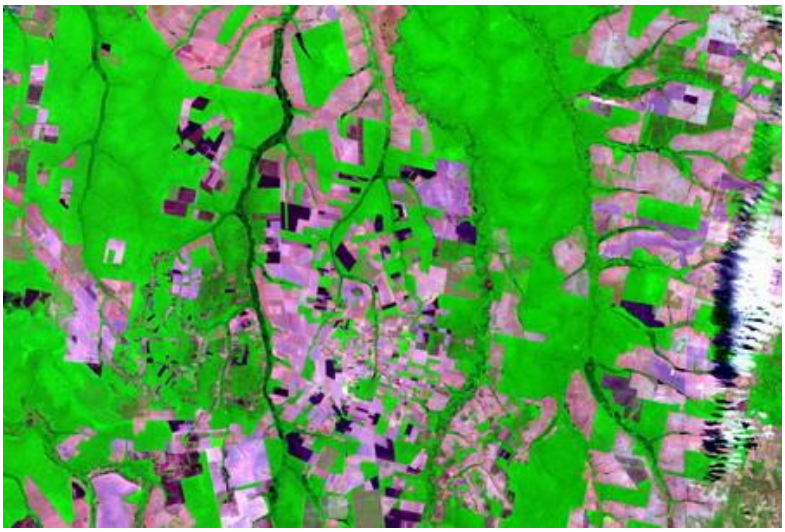
Reference: martin.simonsson@dataductus.se

15 MACHINE LEARNING FOR CLASSIFICATION OF IMAGES

Master thesis work – Luleå/Skellefteå

Data Ductus is involved in several complex vision projects, where we take advantage of machine and deep learning to improve product quality, production rates and perform predictive maintenance. We are working with a wide variety of images from microscope to satellites.





Implementation of algorithms will be done primarily in Python and use third party libraries such as scikit-learn TensorFlow, Keras and OpenCV.

Reference: martin.simonsson@dataductus.se