Federated Control of Distributed Multi-Partner Cloud Resources for Adaptive C2 in Disadvantaged Networks

Harrie Bastiaansen, Johan van der Geest, Casper van den Broek, Thomas Kudla, Anthony Isenor, Sean Webb, Niranjan Suri, Mattia Fogli, Bruno Canessa, Andrea Masini, Robert Goniacz, and Joanna Sliwa

Abstract

In military mission contexts with limited network connectivity, availability and utilization of information may be improved through adaptive information processing over a federated cloud infrastructure. Orchestration mechanisms to dynamically distribute data and computing tasks over the available partner cloud infrastructures enable improved exploitation of the information processing, storage, and communication means that are ever more available and powerful in battlefield situations.

Introduction

Military vehicles and personnel have ever more data sensing, processing, storage, and communication devices available. This provides major opportunities for improving the information position in military mission contexts. However, degraded and disadvantaged networks that are typical in military battlefield situations make moving relevant information and intelligence a challenge. It may prevent (potential mission-critical) information from being available in time.

In our previous work [1], an adaptive and federated approach was proposed to address this challenge. It builds upon the concept of federation in the sense that mission partners mutually expose their computing and storage resources to other mission partners through well-defined application programming interfaces (APIs). A cloud approach is adopted within each individual partner's domain to enable flexibility in distributing computation tasks over multiple resources, referred to as "partner" clouds. Jointly, they provide adaptivity to users as to where (which cloud) to execute information processing tasks.

The basic challenge for a federated cloud infrastructure may be defined as follows. Given a heterogenous set of partner clouds, some having insufficient resources to do all information processing locally, the objective is to adaptively schedule information processing tasks on a partner's cloud, while taking into account the status of the connectivity to that cloud in a disadvantaged mission network context and access control and resource sharing policies. In this context, a cloud is defined as a system of computing resources (processing, storage, sensors, data) available on demand.

The demarcation of the individual clouds should be based on the (anticipated) balance between on one hand the ownership and clustering options of available computing/storage resources in combination with adequate "intra cloud" bandwidth (e.g., a cloud spanning the resources and network within a vehicle) and on the other hand the restricted availability of "inter cloud" bandwidth in the disadvantaged mission context to connect to such a cluster (e.g., between vehicles or to compounds). There may even be cases in which the resources of an individual dismounted soldier will be configured as part of a cloud.

In military mission contexts with limited network connectivity, a federated cloud approach can enable adaptive execution of processing intensive tasks for cases where individual nodes lack sufficient computing resources (e.g., processing or storage capacity and battery power) [2]. Additionally, computations may be offloaded in cases where one nation's cloud provides a unique computational capability not available in another nation's cloud; or different communications conduits may be used when a dismounted soldier (or a vehicle) has better connectivity to a partner nation's cloud as compared to his/her own nation's cloud.

Potential military benefits of a federated cloud infrastructure are improved information availability including resource-constrained nodes (e.g., mobile or dismounted soldiers) and communication-constrained nodes (e.g., underwater platforms), prevention from data overload [3], and resiliency by being able to use partner cloud services and resources when your own cloud services suffer from (cyber) attacks. Potential IT operations benefits are improved utilization of available computing resources, reduced pressure on the limited network connectivity, improved protection options for sensitive data by only allowing this data to be processed within your own cloud while providing mission partners with (less sensitive) processed data, the ability to select

Orchestration mechanisms to dynamically distribute data and computing tasks over the available partner cloud infrastructures enable improved exploitation of the information processing, storage, and communication means that are ever more available and powerful in battlefield situations.

Harrie Bastiaansen, Johan van der Geest, and Casper van den Broek are with TNO; Thomas Kudla is with Fraunhofer FKIE; Anthony Isenor and Sean Webb are with Atlantic Research Centre of DRDC Canada; Niranjan Suri and Mattia Fogli are with the Florida Institute for Human and Machine Cognition; Bruno Canessa and Andrea Masini are with Flyby; Robert Goniacz and Joanna Sliwa are with the Military Communication Institute

Digital Object Identifier: 10.1109/MCOM.001.2000246

As nations will have different requirements for their hardware and vendors, it is each nation's own choice as to the type of hardware and vendor to use. Thus, for the construction of the federated cloud infrastructure, a vendor agnostic approach is important. This may also prevent vendor lock-in.

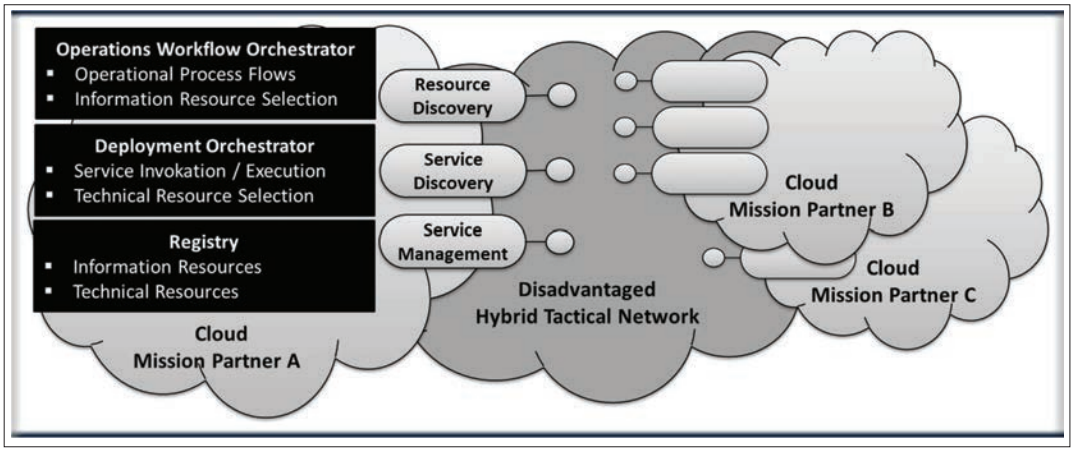

Figure 1. Artifacts for the federated cloud: service APIs and capabilities.

more secure services and resources for processing sensitive data, and proactively setting priorities on processing tasks [4]. On the other hand, new security threats arise by exposing the resources of the individual mission partners' clouds, which require stringent countermeasures.

This article presents the current state of research from the NATO IST-168 Research Task Group (RTG) on "Adaptive information processing and distribution to support Command and Control." This article extends the RTG's previous work [1] by elaborating the architecture, describing the scenarios, use cases, and tests, and providing initial experimentation results and discussion about issues.

Adaptive Federated Cloud Architecture

Architectural Principles

The federated cloud architecture is based on a number of architectural principles in order to work in a military context.

For military systems, it is crucial for mission partners to have sovereignty over their infrastructure and over the data and services hosted on their infrastructure. This is due to individual security and classification policies. If a partner wants to access another partner's resources, this can only be through APIs, which are strictly managed with individual policies on how their clouds expose services and capabilities to partners. This enables full control of their own infrastructure.

As nations will have different requirements for their hardware and vendors, it is each nation's own choice as to the type of hardware and vendor to use. Thus, for the construction of the federated cloud infrastructure, a vendor-agnostic approach is important. This may also prevent vendor lock-in.

The architecture must ensure functional interoperability and compatibility of partner clouds. Therefore, standards-based cloud container technology is adopted, such as the Open Container Initiative (OCI) [5], together with Kubernetes as the orchestration provider. Additional measures must be taken to ensure information interoperability, for example, by using standardized NATO services.

Finally, the federated cloud infrastructure must be able to operate in mission contexts with disadvantaged tactical networks. This applies to the interactions and communications for "inter-cloud connectivity" between nations' clouds. With the cloud demarcation approach described in the introduction, disadvantaged network conditions should not apply to intra-cloud connectivity within an individual cloud spanning multiple nodes.

Architectural Artifacts

Figure 1 depicts the artifacts of the federated cloud architecture: the cloud services, their APIs, and the cloud capabilities.

The cloud service API definitions are the same for all partner clouds. Ultimately, they should be standardized for optimal interoperability. When sending requests to the API, the response values may differ (e.g., depending on the origin of the request). The response values will adhere to the specific policy of the nation's cloud exposing the API.

In the basic cloud service architecture, three APIs are identified and defined:

- The Service Discovery Service API for getting a list of all application services running in a cloud and information on a specific service
- The Resource Discovery Service API for getting metrics of all resource types (CPU, GPU, RAM, connectivity) or of specific resource types.
- The Service Management Service API for invocation and managing service instances

These cloud service APIs as exposed by the individual clouds are the artifacts that form the basis for adaptive information processing in the federated infrastructure. A primary goal is the functional correctness and the network performance impact of these cloud service APIs on a disadvantaged mission network.

The *cloud capabilities* define the internal functions of a partner cloud for realizing adaptive information processing. The capabilities fulfill a cloud internal, partner-specific role. They may differ per cloud instance and mission context. In the basic design, a minimal set of three cloud capabilities are defined and developed:

- *Operations Workflow Orchestrator* manages the overarching handling and control over the information flows according to the requirements of the mission operations model. As described later, initially two information flow models will be distinguished, "Information Demand" (demand/supply) and "Information Chain" (trigger/execute).

- *Deployment Orchestrator* selects the appropriate instance of information resources (i.e. services and data), reserves the technical resources required (locally or in a partner cloud), and manages and monitors the execution of the services over their life cycle.
- *Registry* discovers and administers both information resources and technical resources.

Solutions and Technologies

The deployment and management of containerized applications in a cloud environment typically needs an orchestrator. Here, Kubernetes [6] is used as the orchestrator. It offers a portable, extensible, open source platform for managing workloads and services, facilitating both declarative configuration and automation. It has a large, rapidly growing ecosystem.

There are many Kubernetes distributions available to satisfy a wide range of needs and environments, ranging from cloud to edge computing. This allows different mission partners to select their own distribution, as this will likely happen in realistic mission scenarios. However, reference implementations for both cloud and edge computing environments have also been created, allowing partners involved in the RTG project to quickly get a Kubernetes cluster up and running. The cloud computing reference implementation uses the following technologies:

- *Ansible*: An open source "infrastructure as code" configuration management tool, used to configure Debian or Ubuntu virtual machines (VMs) and bootstrap these machines with a default user account, base packages, Docker, and Rancher
- *Rancher*: A tool for the management of national Kubernetes clusters that also aids in the easy installation of Kubernetes
- *Vagrant*: A tool for building and managing VMs in a single workflow that is used only for development purposes

The combined use of Ansible and Vagrant represents a key benefit for standardizing the installation across partners and speeding up the development and test process.

The edge computing reference implementation used the following technologies [6, 7]:

- *k3OS*: A lightweight operating system is purpose-built for running Kubernetes in low-resource computing environments.
- *Cloud-init*: The configuration of the k3OS device(s) is done with a cloud-init configuration file, allowing the definition of default user accounts and network parameters.
- *k3s*: A lightweight and certified Kubernetes distribution has been built for IoT and edge computing that runs on x86, ARM64, and ARMv7 architectures.

A highly dynamic and changing environment of running services is expected in the federated cloud architecture. Services must be able to connect to other services, which can run on different clouds, or even switch between clouds during runtime. Therefore, on top of Kubernetes, Istio was selected as the service mesh implementation to dynamically change the interconnection between services, even across clouds. The following Istio and service mesh features are particularly interesting to the RTG:

- *Connectivity*: The traffic flow between services can be controlled dynamically with Istio, for both ingress (traffic flowing into the cloud) and egress (traffic flowing out of the cloud).
- *Security*: Traffic between services flows through proxies and is automatically encrypted, preventing man-in-the-middle and other types of attacks.
- *Controllability*: Policies for access control system, billing systems, and quota enforcement systems (like CPU and memory resources) can be applied to manage partner access.
- *Observability*: Istio provides traceability, monitoring, and logging of all services, since traffic flows through the proxies, allowing Istio to capture telemetry data.

In addition, the deployments use the freely available Extendable Mobile Ad-hoc Networking Emulator (EMANE) [8] to emulate multiple types of radios and networks that would link partner clouds (inter-cloud) and Kubernetes nodes (intra-cloud). It enables the evaluation of the network performance of the federated cloud infrastructure when discovering, disseminating, and providing services in tactical network environments.

For demonstration purposes, a dockerized Kubernetes-capable Super Resolution Service module provided by the project partner FlySight [9] is used. It allows the exploitation of information contained in several low-resolution video frames of a specific target to obtain a high-resolution image.

Scenarios, Use Cases, and Tests

To demonstrate and assess the potential military and IT-operations benefits, the federated cloud architecture is considered from three perspectives: scenarios, use cases, and tests (Fig. 2).

Scenarios illustrate how the federated infrastructure concept can be used in representative maritime and land scenarios for adaptive information processing. A maritime and land scenario is elaborated in the following section.

Use cases describe the functional interactions between partner clouds for realizing the operational workflows of the scenarios within the mission context. For instance, they specify the required workflows for either moving the services to the data or moving the data to the services as enabled by the three-cloud service APIs. Hence, the functional use cases are aimed at assessing the main functions of each of the three-cloud service APIs.

Tests assess the performance of a federated infrastructure on various technical aspects in a mission environments, distinguishing:

- *Network performance testing*. This addresses two issues. First, calling upon the service APIs in a partner cloud to support the workflows in the functional use cases introduces bandwidth overhead on the inter-cloud connectivity link. Moreover, it may have minimal quality requirements on data loss and latency. Second, it is expected that the initial Kubernetes systems managing the individual partner clouds perform badly over limited intra-cloud connectivity links. Hence, lightweight Kubernetes variants that are currently emerging will be assessed. For instance, this includes K3S Kubernetes for edge and IoT

Services must be able to connect to other services, which can run on different clouds, or even switch between clouds during runtime. Therefore, on top of Kubernetes, Istio was selected as the service mesh implementation to dynamically change the interconnection between services, even across clouds.

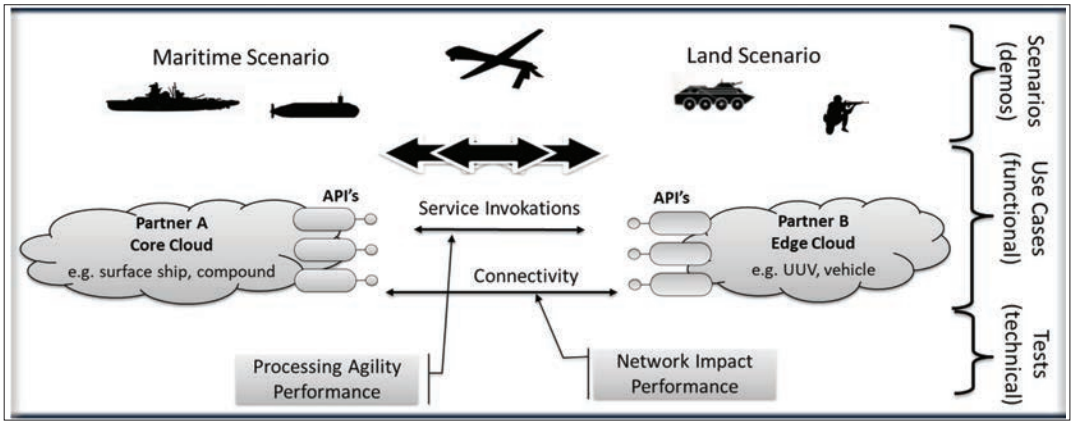

Figure 2. Three perspectives on the federated infrastructure: scenarios, use cases, and tests.

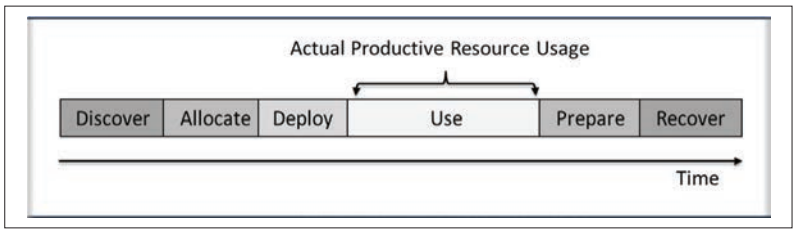

Figure 3. Resource utilization phases.

and the promising concepts of the Kubernetes Federation v2 project.

- *Information processing agility performance testing*. This entails the responsiveness and agility characteristics for the various resource utilization phases of the infrastructure as depicted in Fig. 3. It assesses aspects on how quickly a resource can be discovered, deployed, and utilized, and how quickly a resource can be released [10]. In addition, it addresses the effect of using varying granularity levels of cloud-native services (e.g., on failed remote service executions and on its network performance impact) and the responsiveness of the federated infrastructure in a changing environment (e.g. changing network connectivity conditions and [un] availability of partner clouds).

Scenario Elaboration

To demonstrate the potential of the federated infrastructure in a mission context, the following paragraphs describe an illustrative and representative maritime and land scenario. They have several distinguishing features, as depicted in Fig. 4. Nevertheless, both may be implemented using the federated cloud architecture for adaptive information processing in a similar manner.

The Maritime Scenario

The maritime scenario illustrates an "Information Demand" operational model, as depicted in Fig. 4. In the scenario, surface platforms from two nations conduct a coordinated antisubmarine warfare (ASW) activity in detecting and classifying submarines transiting a narrow strait between an open ocean and an inland sea. The mission context is complex as fishing activity, ferries, and commercial air traffic are present in the operating environment. Moreover, the platforms have varying levels of environmental knowledge.

With the cloud demarcation approach described in the introduction, the computing resources from a particular nation are likely to operate as a single cloud for each of the maritime platforms. Hence, the maritime scenario entails only a small number of clouds. As is typical of an ASW operation, the operation is slow paced as compared to the land scenario. This scenario is also contrived such that the information resources needed at a particular platform (i.e., cloud) need to be found and acquired from a second platform. The two platforms are not from the same nation. A third platform (i.e., a shore site) is also present. Other supporting platforms (e.g., helicopter) introduce diversity in the available communication pathways and computational resources.

All of this introduces a variable supply and demand environment. Here, variations in the information products that are available across the clouds, and the distribution mechanisms that exist between clouds, provide the diversity required for the experimentation.

The scenario outlines a maritime situation that grows in complexity. The evolving situation is used to describe specific use cases that deal with information products, varying communication pathways, and varying compute capability at the clouds. The simplest use case involves the requirement for a particular information product, this being a temperature product in map form. The acquisition of this product begins in the Service Discovery API in Fig. 1, where the cloud requiring the information product obtains a list of available services that are currently executing at the other clouds. The executing service provides sufficient descriptive keyword information to allow the requesting cloud to make a decision on the specific service to provide the product. Once the decision is made as to which service to use, the requesting cloud accesses the service endpoint to acquire the digital product. This effectively completes the loop (Fig. 4) which depicts a demand for an information product, and the supply of that product to the requesting cloud.

Other use cases in the maritime scenario involve simple decision making for similar products. More complicated use cases then involve multiple communication pathways and varying compute resources in different clouds.

The Land Scenario

The land scenario illustrates an "Information Chain" operational model as depicted in Fig. 4. The scenario is based on the third vignette of the Anglova scenario [11], describing neutralization of insurgents in a city environment in the fictitious country of Anglova.

Vehicles from the various mission partners use video cameras to observe their surroundings. These video feeds are analyzed and indexed on the vehicle, for example, registering certain types of activities, vehicles, and objects. The video footage and index are locally stored in the vehicles to save on scarce mission network bandwidth, but still within the originating mission partner's cloud. The video footage library and its index are exposed externally from the cloud to other mission partners.

In the mission, a dismounted soldier detects a vehicle speeding away. The soldier makes video footage of this event with their head-mounted camera, which is stored on their mobile device. The soldier generates a formal intelligence report (SPOT report) and pushes it to the soldiers' own operations center. As it was only vaguely recognizable, the SPOT report did not include the license plate number of the vehicle. The video footage is (still) on the soldier's mobile device. The mobile device is a client that can access either their nation's local cloud or a mission partner's cloud. In the latter option, APIs of the mission partner's cloud are used to trigger the processing and transfer of the video and index to the partner's cloud.

The mission HQ picks up the SPOT report and decides to trigger a follow-up video analysis and intelligence process flow. Its operations workflow orchestrator (OWO) initiates a Super Resolution Service (SRS) to improve the distinctiveness of the video images and to recognize the vehicle's license plate. If the SRS is not available in the cloud where the dismounted soldier's footage is stored, a partner cloud is queried for the service. If found, the service is copied from the partner cloud using the three APIs, taking into account the availability of sufficient resources, including connectivity.

Subsequently, the OWO starts an Object Recognition Service that executes a video matching process to identify relevant video footage available in the various mission partner clouds that should be analyzed on specific aspects. The Service Discovery APIs are used to find available video analysis modules in the various partner clouds. The Resource Discovery APIs provide the current and short-term availability of resources over those partner clouds. On its outcome, the Deployment Orchestrator capability creates a plan for the distribution of how and where to move the data, the code, and the results between the various mission partner clouds. It invokes and manages the execution of the selected services.

Under control of the OWO, the results may be further distributed along the information chain to mission partners, possibly initiating follow-up processes such as vehicle tracking.

Experimentation

The initial experimentation is to illustrate that a federated infrastructure can reduce tactical bandwidth requirements significantly by enabling adaptivity for (local) processing of data near the source. It runs an Object Recognition Service on a live video stream as part of the land scenario. Bandwidth utilization and CPU load on the client devices were measured as indicators for potential network performance improvement when processing a large data resource (video) near its source, as enabled by the federated infrastructure.

	Maritime Scenario	Land Scenario
Number of Mission Partners	Few	Many
Number of Clouds	Few	Many
Adaptivity	Not strictly real-time	(Near) real-time
Operational Model	Information Demand	Information Chain

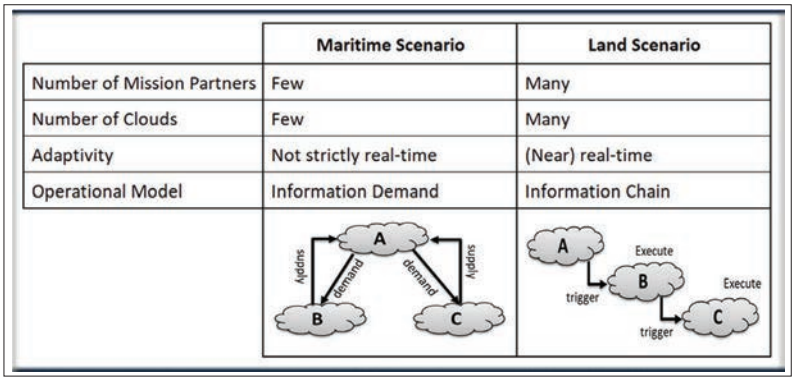

Figure 4. The maritime and land scenario.

The baseline experiment consists of a user with a client device (e.g., laptop) locally running a YOLO [12] object recognition on a video from an IP camera. The enhanced experiment involves containerizing the YOLO service and dynamically deploying it on a (Kubernetes) cloud cluster. For the enhanced case, an assumption was made that the cluster is close/well connected to the IP camera (from a networking perspective). In reality, there may be multiple clusters in the network, and the best one (with good connectivity and adequate resources) would have been selected. However, for this experiment, only a single cluster was used. The baseline and enhanced cases are shown in Fig. 5.

As Fig. 5 shows, both the baseline and enhanced case use a VLC media player to take the stream from the IP camera, a YOLO-based object detection program providing jpeg images as output, and FFmpeg to encode those images and send the result to the NGINX RTMP module. For the enhanced case, additional workflow components are needed: a web server that works as a front-end for users by receiving requests and activating processes and NGINX RTMP for serving the live content with different quality (e.g., bit rate) levels in order to fit within the network limitations from the cloud cluster to the client.

For the baseline case, the client device is a desktop computer equipped with an Intel® Core™ i7-6700 CPU running Ubuntu 18.04.4 LTS. The Kubernetes cloud cluster consists of two servers each equipped with an Intel® Xeon™ E3-1270 v3 CPU also running Ubuntu 18.04.3 LTS and Kubernetes (K8s) 1.17.0 to manage containerized applications with Docker 18.06.2 as the container environment. These are not specific military computing platforms. They are adequate for experimentation as the federated infrastructure is expected to be heterogeneous with a multitude of different computing platforms. Moreover, their measured CPU load will be representative for comparison of the process impact on the system. To measure bandwidth utilization, Tcpdump was used. Top was used to measure CPU load.

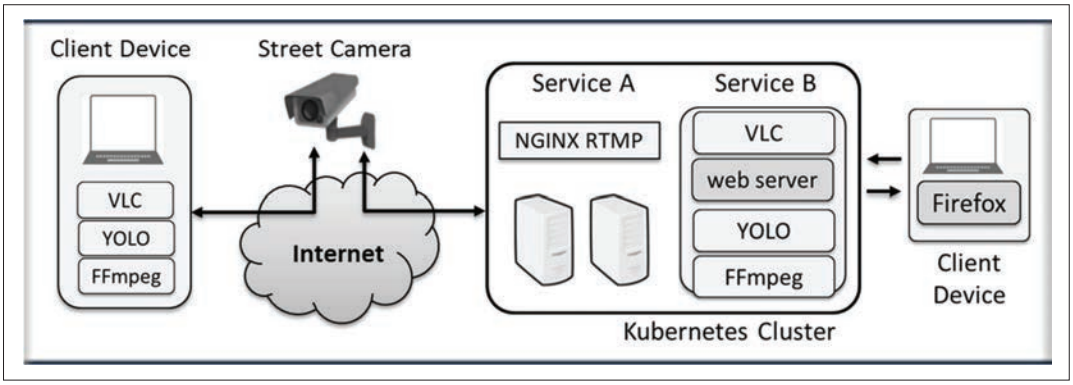

Figure 5. Baseline case (left) and enhanced case (right).

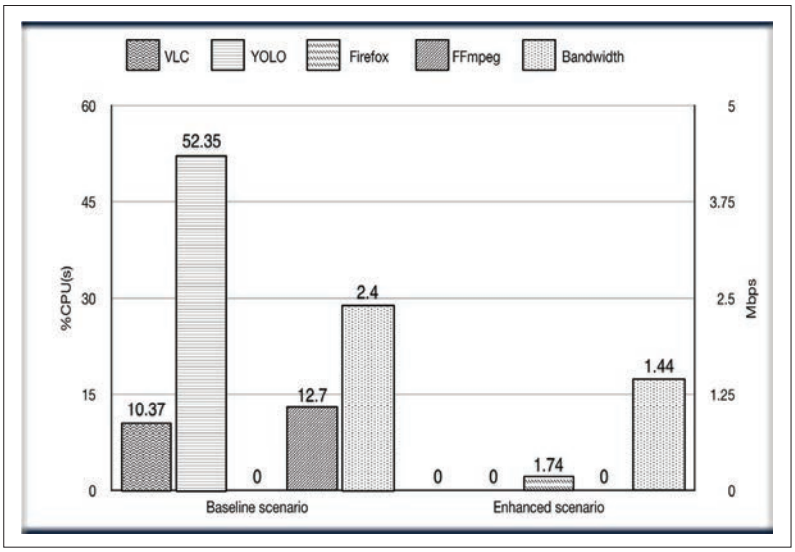

Figure 6. Comparison between baseline and enhanced experimentation cases.

Both experimental cases were executed 10 times, showing only minor deviation in results. Therefore, the averaged results for both cases are shown as outcome in Fig. 6. The left vertical axis shows the CPU load on the client device. The right vertical axis shows bandwidth utilization between the IP camera and the client device in the baseline experiment and between the cloud cluster and the client device in the enhanced one. The bars with names of workflow components show their average CPU loads. They refer to the left vertical axis (%CPU). The "Bandwidth" bars refer to the right vertical axis (megabits per second).

In the baseline case, the most computationally expensive process is YOLO. Its real-time object detection is not trivial in terms of resources, requiring an average of 52.35 %CPU(s). For the overall application stack, the average aggregated CPU load is 75.42 %CPU(s): 10.37 percent from VLC, 52.35 percent from YOLO, and 12.7 percent from FFmpeg. This heavy CPU load on client devices may be a major drawback in a mission context, or even not possible if a smaller, portable device (e.g., an Android phone) is used. As the figure shows, the bandwidth needed to send the video from the IP camera directly to the client device is 2.4 Mb/s.

In the enhanced case, the computation is offloaded to the cloud cluster away from the client device. Kubernetes manages the containerized applications. The client device only runs a browser to play the stream, resulting in a significantly reduced CPU load of 1.74 percent, as shown on the right side of Fig. 6. As the processing is mainly done in the cloud cluster "closer" to the camera, less network bandwidth is consumed over the link to the client. Moreover, NGINX RTMP using FFmpeg behind the scenes may further reduce bandwidth when necessary. For example, in this scenario, the stream's bandwidth is reduced to 1.44 Mb/s, a clear reduction from the bandwidth required by the baseline scenario. Nevertheless, this may still be beyond what most tactical networks can handle.

Status and Discussion

The status of the research is that various instances of partner clouds have been created internally using different Kubernetes (edge and core) cloud implementations. Each cloud exposes the three cloud service APIs. Jointly, they constitute the federated infrastructure providing users with adaptive information processing capabilities. Tooling for emulating disadvantaged mission network conditions between the clouds has been installed.

This forms the basis for testing the network impact and information processing agility performance of the proposed architecture under actual disadvantaged mission network conditions to validate the claims on military and IT operations benefits as stated in the introduction, using the military (demo) scenarios, (functional) use cases, and (technical) tests as identified in this article.

As part of the use case assessment and testing, various issues will get special attention and may need additional exploration:

- The initial experimentation indicated significant potential for the infrastructure to minimize bandwidth in disadvantaged network conditions. Nevertheless, the overhead induced by invoking the partner cloud service APIs will have to be assessed as well.
- Cyber security threats arise by externally exposing the resources of the individual mission partners' clouds. Stringent countermeasures are required on access control, authorization, and enforcement of the individual partners' cloud security policies.
- Deployment orchestration to monitor and manage the execution of services in partner clouds over their life cycle is complex, especially when availability of cloud resources and connectivity links appears to be very

dynamic for actual mission contexts. Information processing agility performance testing may assess the effect of using varying granularity levels of cloud-native services and the responsiveness of the federated infrastructure on changing connectivity and availability conditions.

- In the introduction it was noted that individual clouds are demarcated such that adequate connectivity within each cloud is presumed, with stable network conditions within each cloud. However, when such stable connectivity conditions within a cloud are breached in mission contexts, issues may arise for running Kubernetes as these work on top of the TCP protocol.

These issues will be addressed as part of the performance testing within the RTG.

Conclusion and Future Work

This article has presented a federated cloud-based architecture, design and technology for adaptive information processing, with the potential of providing military and IT operations benefits in mission contexts with conditions of limited network connectivity. It has demonstrated how it may be used in representative (maritime and land) mission scenarios. Initial results showing potential benefits in reducing bandwidth in mission contexts have been presented. Its full potential for improved exploitation of available data will be further assessed by performance testing for mission context conditions. In addition, future work includes the exploration of new cloud-based technologies on their relevance and applicability to the problem domain, including:

- Microsoft Distributed Application Runtime (DAPR), a runtime environment for building more resilient, microservice stateless and stateful applications for the cloud and edge [13]
- Open Application Model (OAM), a specification (aimed at Kubernetes and other [cloud] platforms) to describe applications decoupled from their implementation, providing a separation of concerns between application developer, application operator, and infrastructure operator [14].

Furthermore, the assessment of the functional and performance aspects of the constructed cloud infrastructure will be done using the military scenarios, use cases, and tests described in this article. This will include assessing the discussion topics identified in the previous section.

References

[1] H. Bastiaansen *et al.*, "Adaptive Information Processing and Distribution to Support Command and Control in Situations of Disadvantaged Battlefield Network Connectivity," *2019 Int'l. Conf. Military Commun. and Info. Systems*, Budva, Montenegro, 2019, pp. 1–7. DOI: 10.1109/ICMCIS.2019.8842794.

[2] L. Yin *et al.*, "Joint Scheduling of Data and Computation in Geo-Distributed Cloud Systems," *Proc. 2015 15th IEEE/ACM Int'l. Symp. Cluster, Cloud and Grid Computing*, pp. 657–66.

[3] J. G. Hollands, T. Spivak, and E. W. Kramkowski, "Cognitive Load and Situation Awareness for Soldiers: Effects of Message Presentation Rate and Sensory Modality," *Human Factors*, vol. 61, no. 5, pp. 763–73.

[4] F. T. Johnsen and I. M. Frøseth, "SMART II: Android Apps, Cloud Computing and Mobile Device Management as Enablers for Efficient Operations," *24th ICCRTS Symp.*, Laurel, MD, Oct. 29–31 2019.

[5] The Linux Foundation Project, "The Open Container Initiative"; https://www.opencontainers.org/, accessed Dec. 14, 2019.

[6] Kubernetes (K8s), "Production-Grade Container Orchestration"; https://kubernetes.io/, accessed June 15, 2020.

[7] K3S, "Light Weight Kubernetes — K3S Distribution"; https://k3s.io/, accessed Dec. 14, 2019.

[8] U.S. Naval Research Lab, "Extendable Mobile Ad-hoc Network Emulator (EMANE)"; https://www.nrl.navy.mil/itd/ncs/products/emane, accessed Dec. 14, 2019.

[9] FlySight Srl, "FlySight Remote Sensing Intelligence Solutions"; https://www.flysight.it/, accessed Jan. 20, 2020.

[10] N. Suri and G. Cabri, "Agile Computing," *Adaptive, Dynamic, and Resilient Systems*, Auerbach Publications, 2014, pp. 130–63.

[11] N. Suri *et al.*, "The Anglova Tactical Military Scenario and Experimentation Environment," *2018 Int'l. Conf. Military Commun. and Info. Systems*, Warsaw, Poland, 2018, pp. 1–8.

[12] YOLO, "YOLO: Real-time Object Detection"; https://pjreddie.com/darknet/yolo/, accessed Mar. 18th, 2020.

[13] DAPR, "Distributed Application Runtime"; .https://dapr.io/, accessed Mar. 18, 2020.

[14] OAM, "Open Application Model"; https://oam.dev/. accessed Mar. 18, 2020.

Biographies

Harrie Bastiaansen (harrie.bastiaansen@tno.nl) is a business consultant at TNO. His background is in telecommunications and large-scale ICT-architectures. He is project lead of NATO IST-168.

Johan van der Geest (johan.vandergeest@tno.nl) is a research scientist at TNO. He is specialized in cloud native computing, event driven architectures, and DevOps.

Casper van den Broek (casper.vandenbroek@tno.nl) is a senior technical consultant at TNO. His interest is in future C4I(SR) architectures on which he leads a research program.

Thomas Kudla (thomas.kudla@fkie.fraunhofer.de) is a researcher and consultant at Fraunhofer FKIE. His background is computer science with focus on enterprise architectures and cloud computing.

Anthony Isenor (anthony.isenor@forces.gc.ca) is an information specialist and leads the Maritime Information Support Group at the Atlantic Research Centre of DRDC Canada.

Sean Webb (sean.webb3@forces.gc.ca) is a computer scientist in the Maritime Information Support Group at the Atlantic Research Centre of DRDC Canada.

Niranjan Suri (niranjan.suri.civ@mail.mil) is an associate for research in the Information Sciences Division of the U.S. Army Research Laboratory and a senior research scientist at the Florida Institute for Human and Machine Cognition (IHMC).

Mattia Fogli (mfogli@ihmc.us) is a research intern at IHMC. His background is in telecommunications and operating systems.

Bruno Canessa (bruno.canessa@flyby.it) has a background in computer science and GIS. He has worked on Web GIS services and spatial databases for processing of EO satellite data since 2004.

Andrea Masini (andrea.masini@flyby.it) has a background in telecommunications and in remote sensing. He has worked on electro-optical sensors signal processing and data/video fusion since 2004.

Robert Goniacz (r.goniacz@wil.waw.pl) is head of C4 IT-lab of the Military Communication Institute, with interest in standardization and interoperability of secure information exchange.

Joanna Sliwa (j.sliwa@wil.waw.pl) leads R&D projects for the Military Communication Institute on efficient and secure information distribution for lower command levels.

The status of the research is that various instances of partner clouds have been created, internally using different Kubernetes (edge and core) cloud implementations. Each cloud exposes the three cloud service APIs. Jointly they constitute the federated infrastructure providing users with adaptive information processing capabilities.