Enabling a Big Data and AI Infrastructure with a Data Centric and Microservice Approach: Challenges and Developments

J. (Johan) van der Geest, MSc

TNO
Eemsgolaan 3
9727 DW Groningen
THE NETHERLANDS

johan.vandergeest@tno.nl

C. (Casper) van den Broek, MSc

TNO
Anna van Buerenplein 1
2509 JE The Hague
THE NETHERLANDS

casper.vandenbroek@tno.nl

H.J.M. (Harrie) Bastiaansen, PhD

TNO
Eemsgolaan 3
9727 DW Groningen
THE NETHERLANDS

harrie.bastiaansen@tno.nl

M.R. (Mike) Schenk, MSc

TNO
Anna van Buerenplein 1
2509 JE The Hague
THE NETHERLANDS

mike.schenk@tno.nl

ABSTRACT

The evolution towards Big Data and AI for Military Decision Making in the NATO context should be firmly grounded on a sound enabling infrastructure. The design of this enabling infrastructure can be based on the assumption that data is the really valuable asset. Therefore, focussing on a data centric approach for the enabling infrastructure may provide a better and future-proof basis than a traditional application centric architecture, which has been a major focal point for IT-design and development over the last decades. In combination with flexible development and deployment concepts (based on containerized microservice architectures), this potentially provides a very promising approach as enabling infrastructure for Big Data and AI for Military Decision Making.

In this paper, we will describe the main challenges, developments and solution/research directions on the combination of a data-centric and containerized microservice approach for Big Data and AI for Decision Making. We focus on a NATO military environment with largely distributed data sources (over the various military operational situations, e.g. for in-vehicle usage or for the dismounted soldier) and in a federated organizational constellation of cooperating countries and organizations. The data centric and containerized microservice approach enables data to be made more easily accessible and suitable for AI applications and allows these applications to make optimal use of available resources.

1.0 INTRODUCTION

In a complex NATO environment with largely distributed data sources, a data centric and containerized microservice infrastructure approach may yield major potential benefits in the support of the first three steps of the OODA (Observe, Orient, Decide, Act) loop from data acquisition to decision making. It especially plays a major role in the 'Observe' and 'Orient' phases in which (big) data is to be harvested from sensor information and to be analysed and combined to provide useful information, The term ‘sensor’ is used here in its broadest form, hence including social media analysis and other forms of (un)structured data collection.

Data-centricity can improve the ease of development and maintainability of the infrastructure and simplify the design leading to a high degree of reliability, performance, and capacity. In combination with the loosely coupled, containerized microservices it enables applications to be more easily made suitable for various military operational contexts. This may be done on the basis of the latest agile development and deployment concepts and technologies, especially Continuous Integration/Deployment (CI/CD) and DevOps.

The next chapter describes these opportunities in more detail. Then, we discuss the challenges of implementing these opportunities. Finally, we present a high-level architecture applicable to a NATO context that combines these opportunities and ultimately enables a Big Data and AI infrastructure.

2.0 DEVELOPMENTS

Each year, new technologies emerge in the IT landscape. It is sometimes hard to differentiate short-living hypes from technologies and concepts that are more mature and will stay around for a while. In this chapter we focus on opportunities that enable a Big Data and AI infrastructure to a NATO context, for which we believe are beyond the hype phase and have proven themselves at enterprise-scale.

2.1 Data Centric Approach

The majority of the applications developed for use in a military environment use the application centric paradigm where the application is the one and only master of the data used and produced by the application. It is typically the application that takes care of storing this data in an application specific data store. The only access to this data is via the API's offered by the application.

The data centric paradigm puts the data central and an application is just an entity that performs an operation on some data and produces some new data. All data is stored in a generically accessible data store via a relatively small number of API's. No longer does an application requiring data from various sources, need to interface with a number of other applications, it simply interfaces to a single logical data store.

This switch from an application centric to a data centric approach is illustrated in Figure 2-1. On the left side of the figure the Army, Navy and Airforce of a single or multiple nation have their own applications each applications having its own data. On the right side of the figure there is a common data store (data cloud) from which all applications (the application cloud) from the Army, Navy and Airforce will receive and store their data.

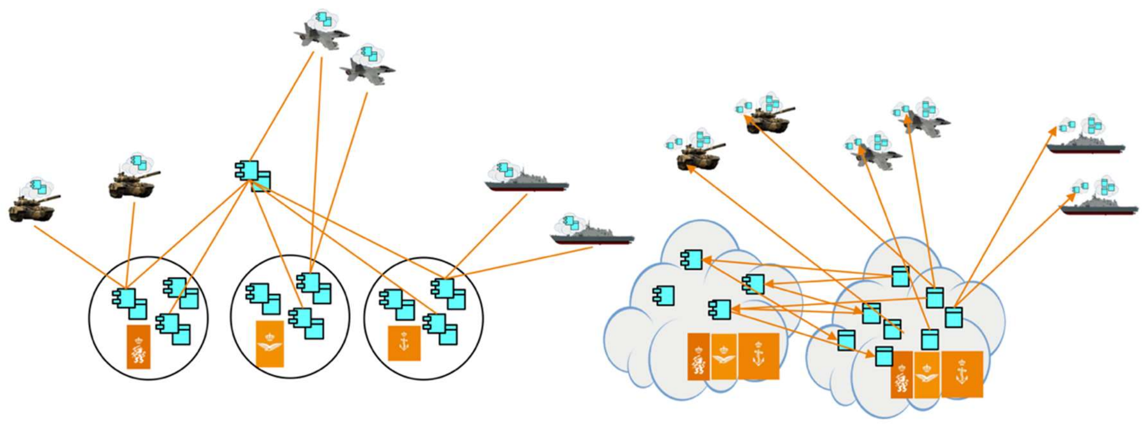

Figure 2-1: Comparison between an application centric (left) and data centric (right) approach.

2.2 Microservice Architecture

A microservice architecture enforces loosely coupled components in the design of a software application. It can be seen as counterpart of monolith applications, where you typically have a single, big code base. In a microservice architecture, each microservice implements a set of narrowly, related functions. Services communicate to each other using either synchronous protocols such as REpresentational State Transfer (REST) and Google Remote Procedure Call (gRPC) or asynchronous protocols such as Advanced Message Queueing Protocol (AMQP) [1]. An example of a microservice architecture implemented by TNO in a military context is shown in Figure 2-2.

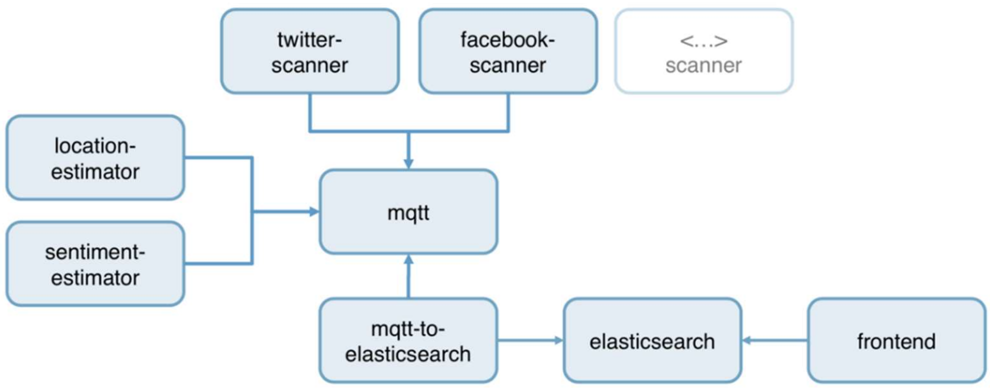

Figure 2-2: Example of an application using a microservice architecture.

In this example, messages with specific keywords from Twitter and Facebook are posted on a MQTT (Message Queueing Telemetry Transport) message bus. These messages are enriched with an estimated GPS location (in case no GPS coordinates are given in a tweet, and the tweet contains for example a street name) and a sentiment (whether the message is positive or negative). The messages on the bus are written to an Elastic Search database and are visualized in a web application.

The advantages of a microservice architecture are clearly demonstrated in this example. First, it is easy to extend the social media scanners with new ones, given that you agree on the message format that is published on the message bus. The same applies to the information enrichment services (like the location and sentiment estimators). It is also easy to swap out the Elastic Search database with another technology, like MySQL. Another advantage is that separate development teams can work on the microservices simultaneously. Testing and deploying the software becomes easier as well, which will be further discussed in paragraph 2.4.

2.3 Software Containerization

Software containerization was formally introduced in 2002, when namespaces were added to the Linux kernel [2]. Namespaces allow you to run software in an isolated environment, while sharing the same operating system. This differs from virtual machines, where each virtual machine requires a guest operating system and adds significant overhead in terms of memory and storage. This makes software containers a very lightweight alternative to virtual machines and a great fit for microservice architectures.

The technology got really popular when Docker was introduced in 2013. Before, setting up containerized environments was a tedious task. With Docker, it is just a matter of running a single command line. Sharing software also became a lot easier, with Docker images. These are basically compressed files containing the software and can be shared with others through a public registry (the Docker Hub [3]) or private registries.

The eco-system around software containerization is continuously growing. Especially around orchestration, logging, traceability and security a lot of tools and best practices can be found, with Kubernetes being the most popular container orchestrator at the moment. It originates from Google, where it was internally known as Borg [4].

3.0 CHALLENGES

In the following paragraphs, the challenges of implementing the data centric approach using containerized microservices and the DevOps methodology in a NATO context are discussed. Figure 3-1 helps in understanding these challenges, as it plots the deployed systems in different NATO domains.

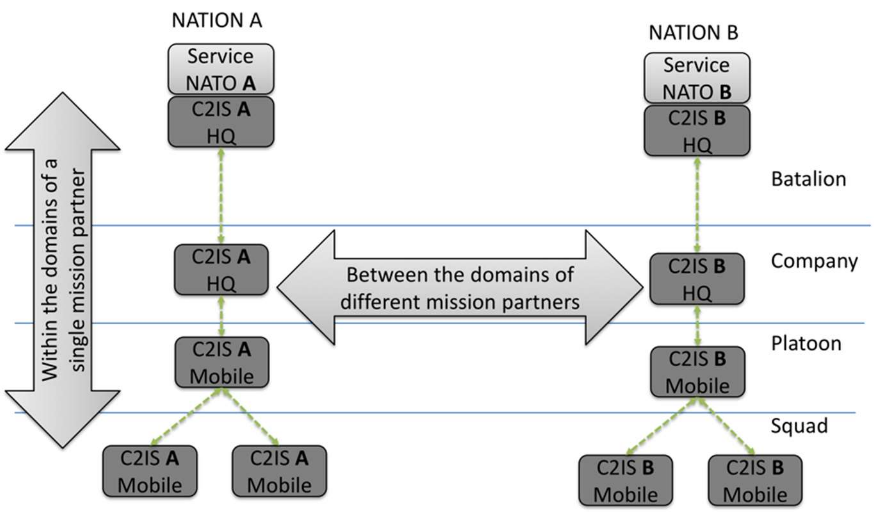

Figure 3-1: The deployed systems and information exchange within the domains of a single mission partner and between the domains of different mission partners. The figure is slightly customized from the *Shared Information Space* paper from Fraunhofer [7].

3.1 Applicable over central (static) and decentral (tactical) mission domains

The (large-scale) acquisition and processing of data in the military context is not only restricted to central infrastructures (headquarter or compound), but can in an ever higher degree been done at the more 'decentralized' levels of the mission infrastructures. In military tactical (operational and decentralized) mission contexts, vehicles and military personnel have ever more IT and communication devices for acquiring, processing and communication of information. These improved sensing, mobile computing and on-board IT systems present great possibilities for the acquisition and processing of (big) data for AI and decision making.

Enabling big data and AI in the tactical (vehicle and dismounted) decentralized mission domain without guaranteed data backhauling to, and processing in the central infrastructures (headquarter or compound), sets new challenges to the design of the enabling infrastructure.

One of those challenges is that the amount of data that can be acquired at a local/decentralized level can be much higher than the amount of data that can be processed at that level. Because connectivity over the disadvantaged tactical networks cannot be guaranteed, processing in a central infrastructure can also not be guaranteed. Hence, a dichotomy exists. On the one hand improved sensing presents great possibility for the acquisition and processing of (big) data for intelligence. On the other hand, the disadvantaged battlefield connectivity and limited local processing power prevent these opportunities from being fully exploited. As a result, the use of data suffers.

A more adaptive infrastructure may be required that allows either the data and/or the application to be moved, thereby allowing processing and information generation at the optimal location. This takes into account the local and current availability of data storage, processing power and connectivity between platforms. Simply said, the consideration becomes to either 'move the data to the code' or to 'move the code to the data'. Therefore, new control strategies for matching data storage, processing and connectivity availability are needed, that orchestrate the distribution and execution of data and processing resources to generate information.

Key for success is an advanced data exchange and service execution orchestration mechanism. Such a mechanism must include the functions for discovery and matching of resources and the function for distribution of data or code. The mechanism creates an optimized, adaptive and executable processing and information distribution strategy based on the combined knowledge of:

- Where data is needed (demand) and where it resides (supply)
- Which resources for data storage and processing are available where
- What connectivity is available between resources

3.2 Interoperability in a NATO mission context

In a NATO mission context, both military and civilian organizations from multiple (NATO and non-NATO) partners will be involved, implying that an adequate interoperability architecture is required. As part of the interoperability architecture the following aspects have to be included for information sharing between the domains of multiple mission partners

- the security architecture for information sharing between (security) domains,
- interoperability at various levels of the information processing stack, e.g. data, application and process level
- interoperability at both the central infrastructure (headquarter or compound) level and at the decentral infrastructure (vehicle and dismounted) levels of the mission context.

3.3 Exploiting the possibilities of data centric and new deployment technologies

The challenges and requirements as identified for big data for AI, may be enabled by various emerging trends and technologies including: the emergence of data centric infrastructures and technologies, the support through advanced data and service discovery and orchestration technologies based on extensive exchange of metadata and the introduction of rapid deployment technologies based on containerisation and (micro)service concepts.

Hence, the information exploitation improvements should be assessed that result from the combined data and service resource matching approach, the data-centric infrastructure approach, and the containerised (microservice) deployment approach. The overarching vision is to make the 'ocean-of-data' readily and seamlessly available in a theatre-independent military NATO tactical mission context.

3.5 Complying to NATO implementation standards, guidelines and policies

For the architecture and interoperability of the data sharing and ICT infrastructures for NATO mission networking contexts, various NATO implementation standards, guidelines and policies have been and are being developed. These include for instance the Federated Mission Networking and the C3 Taxonomy initiatives. An assessment is needed on the compliance and applicability of these initiatives with respect to the challenges and requirements as identified for big data for AI, possibly in resulting in suggestions for further development and alignment.

4.0 HIGH-LEVEL ARCHITECTURE

To develop a data centric, microservice architecture that is able to process big-data using AI is a relatively easy task in the static environment where architectures developed for civil purposes can be transformed for application in a military environment. However, in restricted tactical environments (in vehicles, dismounted and on foot), the amount of data that can be processed is bound by all kinds of limitations, such as storage space, processing power, power usage and physical space. An architecture that can be deployed in these kinds of environments needs to take these limitations into account.

A concept that can help in this respect is the raindrop-to-cloud concept. In this concept, the basic premise is that an individual vehicle, platform or soldier is in possession (either inside the vehicle or man-packed) of all required IT capabilities, but he will also use all available IT capabilities from its surroundings when available. The minimal available IT capabilities form the raindrop on which a soldier can always rely. IT capabilities in its direct surroundings (accompanying vehicles or soldiers) can be used in a fog or edge computing like manner [5]. And if a good connection is possible even cloud-based IT capabilities can be used. If the fog and cloud are constructed in a similar way as the basic raindrop, using additional available IT capabilities involves not much more than distribution of computing tasks.

A first draft of how a high-level data centric, microservice oriented architecture using the raindrop-to-cloud concept could look like is sketched in Figure 4-1.

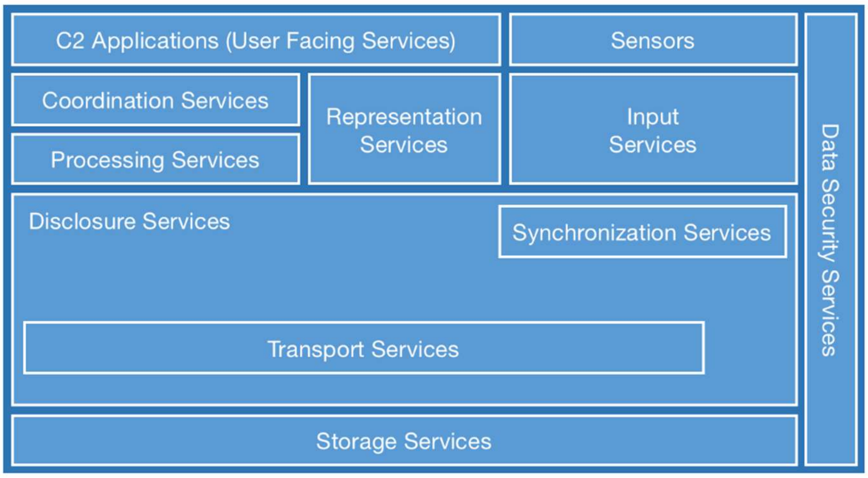

Figure 4-1: High-level architecture using the data-centric approach and microservices as described in chapter 2.

In this high-level architecture, the data storage services form the foundation. Data can be disclosed to either be transported, processed or represented. These services form the core functionalities with which C2 applications can be built. C2 applications as well as sensors can create input data which is synchronized to those storage locations where the data is optimally available for the use by the C2 applications. When a C2 application needs to perform some kind of processing, the coordination services ensure that the processing task or tasks are distributed towards the available IT capabilities (according to the raindrop-to-cloud concept).

5.0 CONCLUSION

As described in this paper, a large number of challenges are to be tackled for enabling the infrastructure for AI in a NATO mission context. As these challenges are NATO cross-panel in nature, a "Thematic Approach" (as launched by the NATO Strategic Sub Group (SSG) Meeting in January 2017, seems to be a viable and conditional pre-requisite. This requires that themes are defined as multi-disciplinary topics described as a military capability that nations need to fulfil. Due to the envisaged (partial) overlap between the NATO theme 'Big data and AI for military decision making' and the theme 'Autonomy from a System Perspective', it is recommended to scope these challenges in relation to the focal point of the Big Data theme, prior to defining and initiating activity on these challenges.

REFERENCES

[1] C. Richardson, "Pattern: Microservice Architecture," [Online]. Available: http://microservices.io/patterns/microservices.html. [Accessed 26 April 2018].

[2] “patch-2.4.19 linux-2.4.19/kernel/kmod.c,” 2 Augustus 2002. [Online]. Available: ftp://ftp.csc.fi/pub/Linux/kernel/v2.4/patch-html/patch-2.4.19/linux-2.4.19_kernel_kmod.c.html. [Accessed 26 April 2018].

[3] “Docker Hub,” [Online]. Available: https://hub.docker.com/. [Accessed 26 April 2018].

[4] A. Verma, L. Pedrosa, M. Korupolu, D. Oppenheimer, E. Tune and J. Wilkes, “Large-scale cluster management at Google with Borg,” in *Proceedings of the European Conference on Computer Systems (EuroSys)*, Bordeaux, France, 2015.

[5] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, Sateesh Addepalli , "Fog Computing and Its Role in the Internet of Things", in proceedings MCC'12 Helsinki, Aug 2012

[6] G. Kim, J. Humble, P. Debois and J. Willis, in *The DevOps Handbook - How to Create World-Class Agility, Reliability, & Security in Technology Organizations*, Portland, IT Revolution Press, LLC, 2016, pp. 3-5.

[7] F. Angelstorf, S. Apelt, N. Bau, N. Jansen and S. Kathner, “Shared Information Space - A solution outline for a global information grid,” in *2017 International Conference on Military Communications and Information Systems (ICMCIS)*, Jun, 2017.