

XXX title

Aliaksandr Adamenko
University of Vienna
Vienna, Austria
alexadamenko@gmail.com

Andrii Fedorenko
University of Vienna
Vienna, Austria
andriifedorenko@gmail.com

Erich Schikuta
University of Vienna
Vienna, Austria
erich.schikuta@univie.ac.at

Abstract—XXXXXX

Keywords—xxx; xxxx;

I. INTRODUCTION

XXX

II. RELATED WORK AND BASELINE RESEARCH

XXX

III. N2SKY ARCHITECTURE

N2Sky is a simulation platform, which allows all involved stockholders to use it for computational purposes. In order to achieve high performance and scalability, it was decided to use the microservices architecture. This approach is used not only on backend services, but also with frontend and its services.

IV. USER-CENTERED INTERFACE WITHIN MODULAR FRONTEND

The basis of N2Sky is the user-centered design. Moving from complex monolithic application to easily understandable one we were concentrated on past user experiances. To gain the functionality and increase the performance of the application were a fundamental requirement.

A. User Role as an Independent User Interface

N2Sky web, as well as the mobile portal, has an intuitive user interface. Since N2Sky is an expert for different knowledge level users it was decided to separate the user roles. Every user namely user role has its own user interface, which is deeply integrated with user needs. The professionals will get detailed information about processes with a technical jargon, simple arbitrary users will get more simplified information.

- *Arbitrary User.* The Arbitrary User a user, who does not have a deep knowledge of the neural network field or know any programming language. His main purpose is not a contribution, but the usage already existing neural networks and trained models. The main goal of the arbitrary user is to study neural networks within N2Sky platform. The arbitrary user can also evaluate the trained models or execute training against existing neural networks. This kind of user does not have to

use his own training data, he just wants to see the behavior of neural networks. The arbitrary user can be converted to Neural Network Engineer user if he has enough knowledge of it.

- *Neural Network Engineer.* The Neural Network Engineer is an arbitrary user. The Neural Network Engineer has an access only to his own dashboard and publicly available resources on the main application module. He can perform the semantic search for available neural network paradigms and use them. This user can create own neural network instance from existing neural network paradigm. He can also train the running neural network instances and evaluate their trained models. This user can share his trained neural network by making it public.
- *Contributor.* The Contributor is an expert user, which has enough knowledge and experience to create his own neural network. This user can create neural network paradigms using the ViNNSL template schema [1] and publish them on N2Sky. This user can deploy neural networks on the N2Sky environment as well as on his own environment by providing training and testing endpoints. The goal of the contributor is the study how his networks will behave with different network structures, input parameters and training data that is provided by other users.
- *System Administrator.* System Administrator is a user who has a full access to the application including environment management, monitoring and alerting features. The administrator can manage OpenStack and Cloudify instances. He also can shadow any N2Sky user to observe the application from other perspectives. The administrator has access to all dashboards in every module.

B. Modular frontend application design

Maintain the large application like N2Sky with the monolithic approach is unwieldy. Since N2Sky supports microservices approach in the backend it was decided to apply the same solution on the frontend.

Microservices in frontend are small independent web applications, which are consolidated into one application. The main benefits of this approach are:

- *Maintainability.* It is possible to divide application between different teams. Developers do not even need to have some knowledge about other parts of the application.
- *Diversity of technologies.* Monolithic approach makes the whole application stick to one framework. Microservices allowing to use any technology without the need to rewriting the application.
- *Independent deployment.* Every application has some releases periods, every release accompanies with redeployment procedure. There is no need to redeploy the whole application, but just only required components.

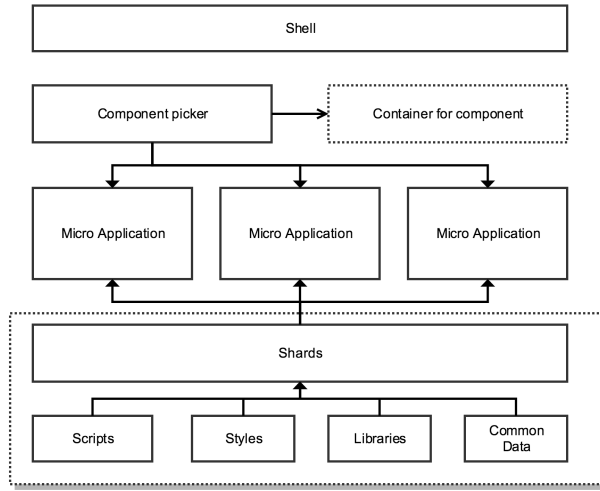


Figure 1. Microservices approach in frontend application

Microservices approach in frontend application is shown in “Fig. 1”. This breaks the entire application into small micro applications [2].

- *Shell.* is a top-level component, which wraps Components picker and Container for the component. It contains application configuration.
- *Component picker.* Is a router, which manages the micro applications.
- *Container for Component.* Container, where the component will be injected.
- *Micro Application.* Independent application, which can be written in any programming language but has to use one of the shards.
- *Shards.* Is a code base, which is shared between Micro Applications. Shards can have multiple levels.

C. N2Sky modular frontend

The central concept of the application is to support the Software as a Service (SaaS) and Platform as a Service (PaaS) distributions [3]. N2Sky consists of two modules:

administration module, main application module as it shown in “Fig. 2”.

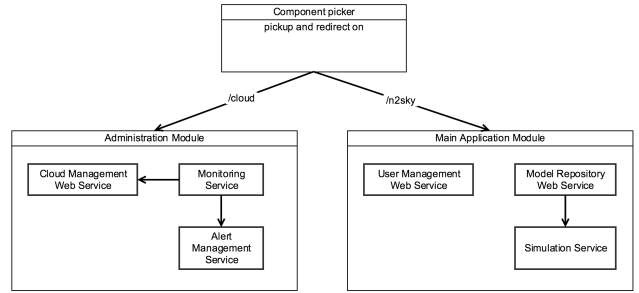


Figure 2. N2Sky frontend application and its services

- *N2Sky component picker.* When the user goes to N2Sky web portal, first he will be dealing with a component picker. The component picker is a small service, which redirects the user depending on URL path.
 - */cloud* redirects to “Administration module”
 - */n2sky* redirects to “Main application module”
- *Administration module* The administration module allows the system administrator to control the environment. The module supports OpenStack and Cloudify monitoring. Managing is possible through the application dashboard. It also contains custom monitoring and an alerting management system, which can be installed on any server within the N2Sky user interface. The administration module implements PaaS. It is fully configurable and wrapped into the open source project in order to make the module accessible to the third-party applications.
- *Main application module* The main application module is the central module of N2Sky. Within this module, users can use, train and test existing neural networks. It is possible to reuse the neural network paradigms and create own neural network. N2Sky allows deploying own network and store data in the cloud. Module services are supporting the SaaS distribution. Experts can use an application directly through the N2Sky API or they can integrate N2Sky services into their own application.

V. THE SAMPLE WORKFLOW

To support Software as a Service (SaaS) distribution every web service can work independently. It means that the stakeholders can use N2Sky via web portal as well as use N2Sky API. N2Sky API allows stakeholders:

- Authorise in the System
- Create new neural from existing paradigm
- Deploy own neural network on N2Sky environment
- Perform training against own as well published neural network

- Perform testing against trained models

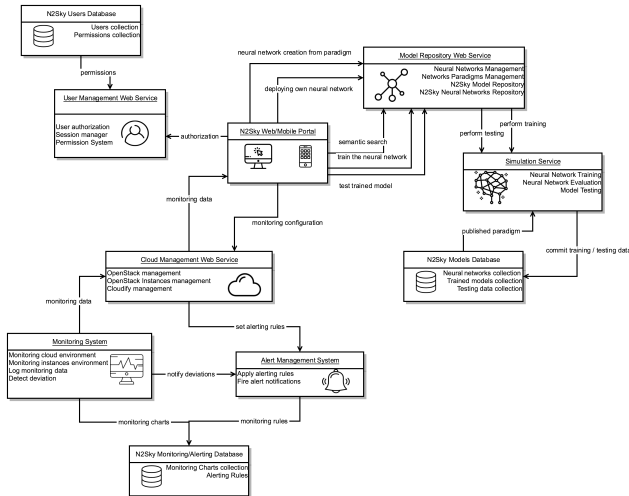


Figure 3. The sample workflow

The sample workflow overview, which shown in figure 3 represents microservices architecture in action:

1) Contributor User

- The Contributor user authorize in the N2Sky portal using his browser on desktop PC or mobile device. He will be redirected to his own dashboard according to permissions, which will be received from User Management Web Service.
- The user described his own neural network paradigm using the ViNNsL template and deploying it in the N2Sky cloud as it shown in 4.

Figure 4. The user uploads the neural network paradigm description in ViNNsL format

- The Contributor user perform training of his neural network using the N2Sky platform. Since the user is an expert he can perform this operation using Simulation Service via Model Repository Web Service API.
- The user publishes his paradigm via N2Sky UI or available API.
- The Contributor user awaiting until other N2Sky users will use his neural network paradigm in order to monitor the behavior of the neural network.
- The user modify, redeploy and retrain his neural network after first results.

2) Neural Network Engineer User

- The neural network engineer user authorize in the N2Sky portal using his browser on desktop PC or mobile device. He will be redirected to his own dashboard according to permissions, which will be received from User Management Web Service.
- From the dashboard, the user creates a neural network from existing paradigm using Model Repository Web Service. The user specify the neural network structure as it shown in 5.

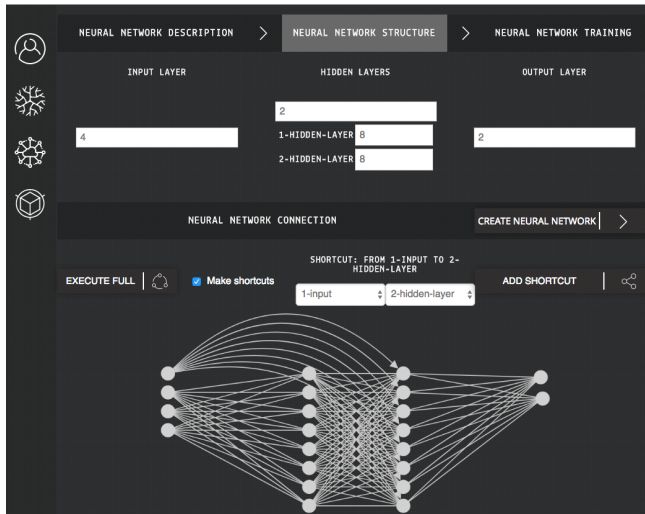


Figure 5. The user specifies the neural network structure

- c) The user perform training against his newly created neural network using the N2Sky platform as it shown in 6.

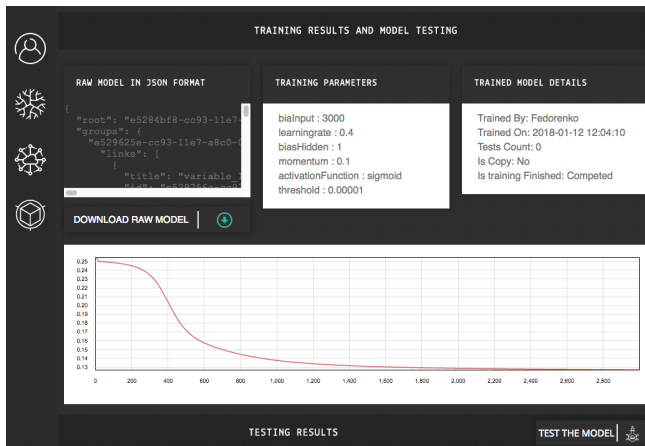


Figure 6. The user performs neural network training

- d) If the user satisfied with a trained model he can perform testing using the N2Sky platform.
- e) The neural network engineer user publish his neural network and trained model in order to make it available for other N2Sky users.

3) Arbitrary User

- a) The arbitrary user authorize in the N2Sky portal using his browser on desktop PC or mobile device. He will be redirected to his own dashboard according to permissions, which will be received from User Management Web Service.
- b) Since the user does not much knowledge in neural network field, he performs a semantic search in order to find some neural network as

well as trained models according to his needs as it shown in 7.

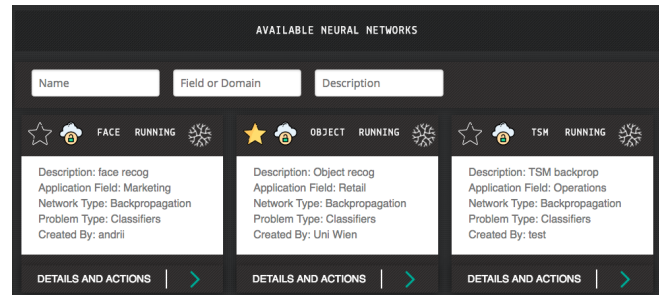


Figure 7. The user performs semantic search in Neural Network Repository

- c) The user copy existing neural network and some trained models into his project.
 - d) The user perform training from N2Sky platform against copied neural network with the default input parameters data.
 - e) The user evaluate trained neural network model with the default parameters.
- ### 4) System Administrator
- a) The system administrator authorize in the N2Sky portal using his browser on desktop PC or mobile device. He will be redirected to administration dashboard.
 - b) The user observes cloud environment.
 - c) The system administrator creates the new monitoring chart with a specific metrics and add it to the administration dashboard.
 - d) The user creates alert against newly created monitoring
 - e) The user is notified by Alert Management System, that some event occurs.

VI. CONCLUSION

The conclusion goes here. this is more of the conclusion

ACKNOWLEDGMENT

The authors would like to thank... more thanks here

REFERENCES

- [1] Peter Paul Beran, Elisabeth Vinek, Erich Schikuta, and Thomas Weishaupt. Vinnsl - the Vienna neural network specification language. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1872–1879. IEEE, 2008.
- [2] Mikki Kobvel. Front-end microservices with web components. [online], <https://hackernoon.com/front-end-microservices-with-web-components-597759313393>, last visited November 2017, 2017.
- [3] Stefan Walraven, Eddy Truyen, and Wouter Joosen. Comparing paas offerings in light of saas development. *Computing*, 96(8):669–724, Aug 2014.