OPEN-SOURCE COMPLIANCE: A TACTICAL APPROACH

Ashish Bakshi[1], Andreas Kotulla[2], Oldřich Faldík[1], Oldřich Trenz[1]

[1]Department of Economics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic

[2]Bitsea GmbH, Schloßstraße 7, 53757 Sankt Augustin, Germany

ABSTRACT

As open source becomes increasingly prevalent, understanding the intricacies of various license types, including permissive and copyleft licenses, becomes essential for developers and organizations alike (Tourani, Adams and Serebrenik, 2017). This paper not only explores these license types but also examines the implications of copyright laws and Export Control Compliance (ECC) on open-source software.

A significant portion of the paper is dedicated to evaluating key tools used in open-source compliance, such as SW360, FOSSology, OSS Review Toolkit (ORT), and Software Bill of Materials (SBOM).

In this paper, a comprehensive analysis of open-source license compliance offers practical insights and recommendations for developers and organizations navigating the complexities of open-source software adoption. The specific contribution of this paper lies in providing a detailed comparative analysis of these tools, alongside a case study on their application in real-time audits.

Keywords: open-source compliance, sw360, fossology, ORT, SBOM

JEL: L8, O3

1 INTRODUCTION

In the evolving landscape of software development, open-source software (OSS) has emerged as a cornerstone, driving innovation and collaboration across industries. However, with the widespread adoption of OSS (Sherae, 2016), the complexity and importance of license compliance have become more pronounced. This paper aims to provide a comprehensive overview of the multifaceted aspects of open-source license compliance, Export Control Compliance (ECC), and the security vulnerabilities and risks associated with open-source software, which remain crucial yet challenging domains for many organizations and developers (He, Peters, Menzies and Yang, 2013).

Various tools have been developed to aid in navigating these complexities, such as SW360, FOSSology, OSS Review Toolkit (ORT), SPDX, and Software Bill of Materials (SBOM). This paper

https://doi.org/10.11118/978-80-7509-990-7-0008

will evaluate the efficacy of these tools in facilitating compliance with open-source licenses, alongside a practical case study of a real-time audit. It will aim to investigate the impact of complexities in open-source licensing, export controls, and security vulnerabilities on the sustainable development and utilization of open-source software. Additionally, it will seek to explore how specialized tools such as SW360, FOSSology, ORT, SPDX, and SBOM can contribute to the effective management of legal and security risks in open-source software projects.

2 UNDERSTANDING OPEN-SOURCE LICENSES

Copyleft licenses, in contrast to permissive licenses, are designed to ensure that derivative works of the software remain open source. They require that modifications and extensions of the original software be distributed under the same license terms (Lindman, Paajanen and Rossi, 2010). Copyleft licenses ensure that derivative works of the software remain open source, requiring any modifications to be distributed under the same license terms. GNU General Public License (GPL), GNU Lesser General Public License (LGPL), Mozilla Public License (MPL) are some examples of copyleft licenses (González-Barahona, 2009).

2.1 Permissive licenses

Permissive licenses are a category of open-source licenses that impose minimal restrictions on how software can be used, modified, and distributed (Maryka, 2015). These licenses are often preferred for their flexibility and ease of integration into proprietary projects. Due to their minimal restrictions, permissive licensed software is often used in a variety of applications, from open-source projects to commercial products. MIT License, Apache License 2.0, and BSD Licenses (2-clause and 3-clause) are some examples of permissive Licenses (Coleman, 2014).

2.2 Copyleft licenses

Copyleft licenses, in contrast to permissive licenses, are designed to ensure that derivative works of the software remain open source. They require that modifications and extensions of the original software be distributed under the same license terms. Copyleft licenses ensure that derivative works of the software remain open source, requiring any modifications to be distributed under the same license terms. GNU General Public License (GPL), GNU Lesser General Public License (LGPL), and Mozilla Public License (MPL) are some examples of copyleft licenses (Mathur, Choudhary, Vashist, Thies and Thilagam, 2012).

3 EXPORT CONTROL COMPLIANCE (ECC)

Export Control Compliance (ECC) is an integral part of distributing and developing open-source software in a global context. Developers and organizations need to be vigilant and proactive in understanding and adhering to relevant export control laws to avoid legal repercussions and ensure responsible software distribution (Kumar, 2022). This is especially crucial in the context of open-source software, which often transcends international borders. ECC refers to a set of laws and regulations imposed by countries to control the export of certain technologies, including software, for reasons of national security or foreign policy (Choi, 2008).

Different countries have varying regulations on software exports. For instance, the United States' Export Administration Regulations (EAR) and the International Traffic in Arms Regulations (ITAR) govern the export of software that could be used in military or strategic contexts. Developers and organizations must be aware of these regulations to ensure compliance.

Software is classified under different categories based on its potential application in sensitive areas, such as encryption technology. Open-source software that falls into certain categories may require specific export licenses or be subject to restrictions.

ECC can impact how open-source software is developed and shared. For example, developers might need to restrict access to certain code repositories or implement measures to prevent the transfer of controlled technology (Shim, 2011).

4 VULNERABILITIES IN OPEN-SOURCE SOFTWARE

Open-source software, known for its numerous benefits like accessibility and collaborative development, is not immune to security vulnerabilities and risks. These vulnerabilities can significantly impact not only the security of the software but also compliance with open-source licenses. The widespread use of open-source software introduces a range of vulnerabilities that can affect both security and compliance with licensing terms. Understanding these vulnerabilities and their implications is crucial for maintaining the integrity and legal standing of software projects.

4.1 Identifying Common Vulnerabilities

4.1.1 Code Quality and Complexity

Open-source projects, particularly large ones, can face issues related to code quality and complexity, making them susceptible to security vulnerabilities. For example, Apache Struts and OpenSSL have encountered significant vulnerabilities, such as the Struts remote code execution vulnerability and the infamous Heartbleed bug in OpenSSL.

4.1.2 Dependency Management

Many open-source projects depend on other libraries and frameworks, where vulnerabilities in these dependencies can compromise the entire project's security. An instance of this was seen in the event-stream incident, posing serious threats to projects like Node.js and other JavaScript frameworks.

4.1.3 Lack of Sustained Maintenance

Inconsistencies in maintaining and updating some open-source projects can lead to outdated code and unaddressed vulnerabilities. Older versions of WordPress plugins, for example, have been known to contain security flaws exploited in various attacks.

4.2 Interplay Between Vulnerabilities and Licensing Compliance

4.2.1 Security Updates and Licensing Terms

Licenses such as the GNU GPL mandate public disclosure of all modifications, including security patches. Failure to comply can result in legal disputes, evident in instances where organizations did not release modified source code back to the community.

4.2.2 Enforceability and Security Flaws

Vulnerabilities can challenge the enforceability of licenses. A security patch that alters the software's original functionality might inadvertently violate the original license terms.

4.2.3 Compliance Challenges in Addressing Vulnerabilities

Effectively managing vulnerabilities, as demonstrated by the Linux kernel (GPL-licensed), involves a delicate balance between regular security updates and adherence to licensing terms, a complex task for many organizations.

5 TOOLSETS FOR COMPREHENSIVE COMPLIANCE

5.1 SW360

SW360 excels in managing and documenting the licenses of all software components within a project. It facilitates an organized approach to maintaining compliance with various open--source licenses, thereby mitigating legal risks. The tool serves as a comprehensive catalog for software components, simplifying the process for organizations to track and manage the use of open-source software in their projects. This functionality is crucial for maintaining a clear overview of all software dependencies and their corresponding licenses.

SW360 can be seamlessly integrated into existing software development processes. This integration ensures that license compliance becomes a continuous and integral part of the development lifecycle, rather than an afterthought.

By providing a central platform for managing open-source components, SW360 fosters collaboration among development teams and enhances transparency in the usage and management of open-source software within an organization.

5.2 Open-Source Review Toolkit (ORT)

ORT offers an extensive suite of capabilities designed to streamline the review and analysis of open-source licenses. Its primary aim is to assist organizations in understanding and complying with open-source licenses, thereby reducing legal risks, and enhancing project integrity. ORT analyzes the licenses of software components within a project, providing a detailed overview of compliance requirements and potential risks.

The process begins with ORT scanning the files of a software project. It identifies and lists all the open-source components and dependencies used within the project.

To detect licenses associated with each component or dependency, ORT employs various scanners, such as *ScanCode*. It searches for license files, headers in source code, and other metadata to accurately identify the licenses.

5.3 FOSSology

FOSSology operates by scanning software code to detect open-source licenses, enabling users to review, curate, and report on the license information. Its support for SPDX and customizable scanning options make it a versatile tool for managing license compliance in diverse software projects. FOSSology scans the uploaded files to detect and identify open-source licenses, using advanced scanning techniques to examine file contents, including comments, headers, and documentation, for license information. Utilizing its comprehensive database of open-source licenses, FOSSology can identify a wide range of licenses in the scanned files. The tool highlights the exact text snippets where license terms are found, making it easier for users to review and confirm the license findings. Users can manually review the scan results to verify or correct the identified licenses. This step is essential, particularly in cases where the software contains custom licensing terms or dual-licensed files.

6 METHODOLOGY

The OSS Review Toolkit (ORT) is used for analyzing and reviewing dependencies in open-source software. It scans a project to identify all open-source components and their respective licenses. When an Ansible, *https://github.com/ansible/ansible*, project is run through ORT, it generates a detailed report of all the open-source components used in the project, along with their respective licenses.

After the ORT process, the same Ansible project is then fed into Fossology for a more thorough license analysis. This step is likely to provide a more granular view of the licenses and any potential issues or conflicts.

Following the analyses conducted by ORT and Fossology, the results are integrated into SW360. This tool serves as a centralized platform for managing these components, tracking license compliance, and documenting findings. SW360 allows for systematic management of open-source components, ensuring that all data is consistently updated and easily accessible for review. The aggregated data in SW360 can then be reviewed by audit and legal teams. These teams assess the compliance of the open-source components with the organization's policies and legal requirements.

The compliance of these open-source components is then rigorously evaluated by the organization's audit and legal teams. They assess each component against specific review criteria, including adherence to licensing terms, compatibility with internal policies, and legal risk management. This review ensures that all components meet the stringent standards required for organizational use and legal compliance, Finally, once the open-source components have been cleared by the audit and legal teams, they are approved for deployment in production environments. This approach effectively combines automated tooling with manual review, ensuring that open-source components are used responsibly and in compliance with legal requirements. This process not only aids in license compliance but also helps in managing security vulnerabilities that might be present in open-source components.

7 RESULT

The application of FOSSology to the Ansible codebase revealed intriguing results. The initial scan identified a significant number of files with GPL 3.0 licenses.

The report generated by FOSSology initially indicated that there were 616 files licensed under GPL 3.0 and another 377 files with GPL 2.0 licenses. Following the initial scan, a meticulous file-by-file review was conducted to pinpoint the presence of strict copyleft licenses and to identify the copyright holders for each component.

This granular examination was crucial to understand the extent and implications of copyleft licensing within the Ansible project.

The Ansible project was subjected to ORT's analysis, utilizing its analyzer and downloader modules. This process was aimed at meticulously scanning each dependency and transitive dependency within the project.

The analyzer module was particularly instrumental in breaking down and identifying the specifics of each dependency, offering a detailed look into the project's composition. ORT's comprehensive scanning covered not only the direct dependencies of Ansible but also the transitive dependencies, which are often overlooked yet crucial for a complete compliance picture.

This extensive scanning provided a deeper insight into the software stack, revealing the intricate network of dependencies within Ansible.

Upon completion of the ORT analysis, the SPDX (Software Package Data Exchange) report was seamlessly integrated into SW360 through its import functionality. This integration marked a pivotal step in centralizing and streamlining the compliance management process.

The imported SPDX report within SW360 became readily accessible to various teams, including the audit and legal teams. This accessibility facilitated a more efficient and transparent review process.

Such visibility is crucial for these teams to perform thorough compliance checks and to make informed decisions regarding the legal aspects of the project.

With the SPDX report in SW360, collaboration among different departments was significantly improved. The legal team, for instance, could easily cross-reference the report for any licensing issues, while the audit team could use the data for compliance verification.

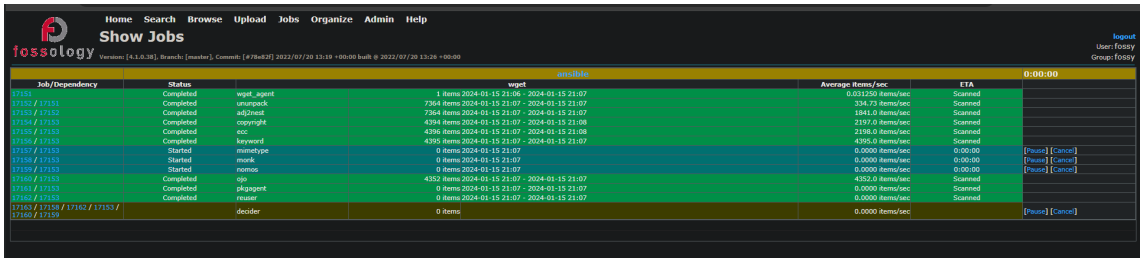

Fig. 1: Fossology scan Ansible

NUMBER	AUDITED	LICENSE
3	0	Apache-2.0
4	0	BSD
65	0	BSD-2-Clause
4	0	BSD-3-Clause
5	0	GPL
1	0	GPL-2.0
9	0	GPL-2.0+
616	0	GPL-3.0
377	0	GPL-3.0+
4	0	MIT
58	0	No_license_found
3	0	PSF-2.0
1	0	Public-domain
5	0	Python
3	0	See-URL
1	0	See-doc.OTHER

Tab. 1 Report of License Ansible

COPYRIGHT STATEMENT	FILE LOCATION
Copyright: Contributors to the Ansible project GNU General Public License v3.0+ (see COPYING or https://www.gnu.org/licenses/gpl-3.0.txt)	lib/ansible/compat/importlib_resources.py lib/ansible/modules/deb822_repository.py test/integration/targets/result_pickle_error/action_plugins/ result_pickle_error.py test/units/module_utils/urls/test_fetch_file.py test/units/module_utils/urls/test_split.py
Copyright: Contributors to the Ansible project	lib/ansible/module_utils/urls.py
Copyright: Ansible Team GNU General Public License v3.0+ (see COPYING or https://www.gnu.org/licenses/gpl-3.0.txt)	lib/ansible/modules/add_host.py lib/ansible/modules/group_by.py

Tab. 2 Report of Copyright Ansible

```
server-6 user (e) base … ort-native mime-types reporter-output-dir ls
bom.spdx.yml  gl-license-scanning-report.json  NOTICE_DEFAULT  scan-report.html  scan-report-web-app.html
server-6 user (e) base … ort-native mime-types reporter-output-dir cat gl-license-scanning-report.json
{
  "version": "2.1",
  "licenses": [
    {
      "id": "Apache-2.0",
      "name": "Apache License 2.0",
      "url": "https://spdx.org/licenses/Apache-2.0"
    },
    {
      "id": "BSD-2-Clause",
      "name": "BSD 2-Clause \"Simplified\" License",
      "url": "https://spdx.org/licenses/BSD-2-Clause"
    },
    {
      "id": "BSD-3-Clause",
      "name": "BSD 3-Clause \"New\" or \"Revised\" License",
      "url": "https://spdx.org/licenses/BSD-3-Clause"
    },
    {
      "id": "CC-BY-3.0",
      "name": "Creative Commons Attribution 3.0 Unported",
      "url": "https://spdx.org/licenses/CC-BY-3.0"
    },
    {
      "id": "CC0-1.0",
      "name": "Creative Commons Zero v1.0 Universal",
      "url": "https://spdx.org/licenses/CC0-1.0"
    },
    {
      "id": "ISC",
      "name": "ISC License",
      "url": "https://spdx.org/licenses/ISC"
    },
    {
      "id": "LicenseRef-scancode-public-domain-disclaimer",
      "name": "",
      "url": ""
    },
    {
      "id": "MIT",
      "name": "MIT License",
      "url": "https://spdx.org/licenses/MIT"
    },
    {
      "id": "WTFPL",
      "name": "Do What The F*ck You Want To Public License",
      "url": "https://spdx.org/licenses/WTFPL"
    },
    {
      "id": "X11",
      "name": "X11 License",
      "url": "https://spdx.org/licenses/X11"
    }
  ],
  "dependencies": [
    {
```

Fig. 2: ORT scan Ansible SPDX report

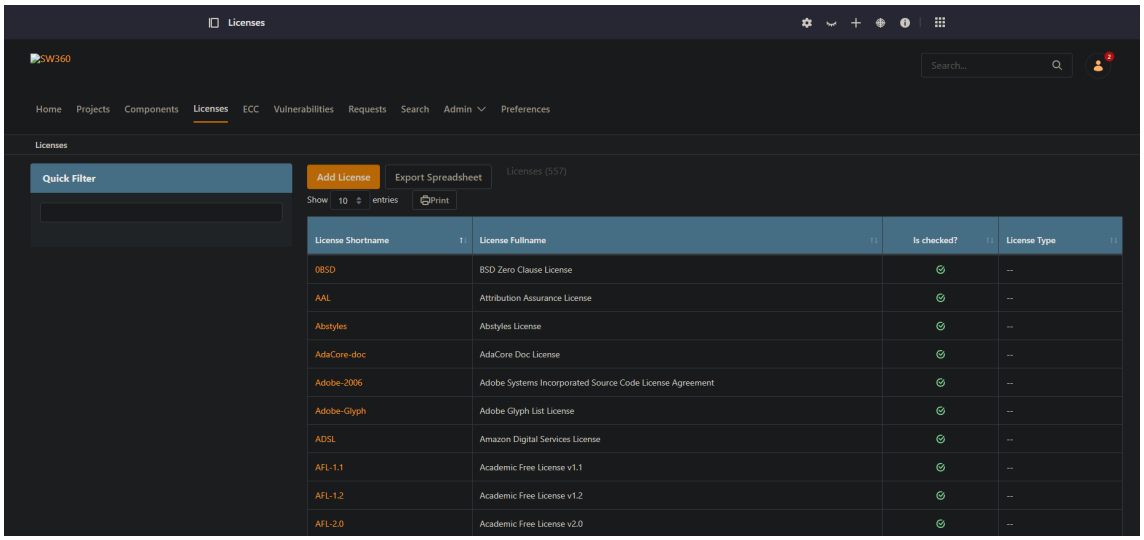

Fig. 3: SW360 Report imported

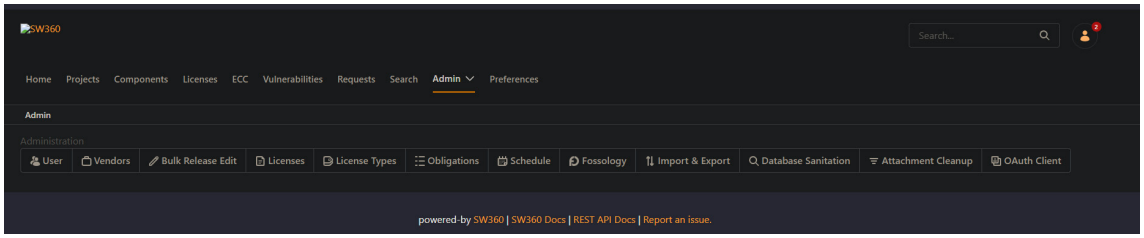

Fig. 4: SW360 ECC, Obligation, Vulnerability

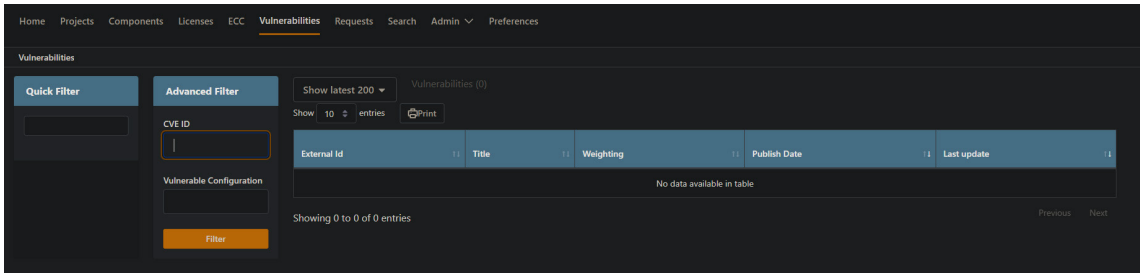

Fig. 5: SW360 CVE id

8 DISCUSSION AND CONCLUSIONS

The prevalence of GPL 3.0 licenses poses certain compliance requirements, especially considering the strict copyleft nature of this license. This finding necessitates a careful approach to how the Ansible software is used, modified, and distributed.

The identification of GPL 2.0 licenses also has significant implications, particularly in terms of compatibility with other licenses and the obligations it imposes on derivative works. The use of ORT and SW360 in this manner promotes responsible open-source software development practices. By ensuring thorough compliance, organizations can avoid legal pitfalls and maintain ethical standards.

These results highlight the complexity of managing open-source licenses in large projects. The mix of GPL 3.0 and GPL 2.0 licenses within Ansible underscores the need for thorough compliance checks and an understanding of license obligations.

Accurate license compliance management, facilitated by these tools, is essential for the long-term sustainability of open-source projects. It ensures that projects adhere to legal requirements, thereby securing their viability and reputation in the open-source community.

Acknowledgements

This paper was supported by the project CZ.02.1.01/0.0/0.0/16_017/0002334 Research Infrastructure for Young Scientists, this is co-financed from Operational Programme Research, Development and Education.

REFERENCES

CHOI, C.-H. 2008. A Study on Global Compliance of Global Companies under the Circumstance of Export Control. *The Korean Research Institute of International Commerce and Law.*

COLEMAN, M. A. 2014. *Freedom From Restriction, Freedom Of A Restriction: A Comparison Of Some Open Source Software Licenses*. https://arxiv.org/abs/1402.2079

GONZÁLEZ-BARAHONA, D. M. 2009. An Empirical Study of the Reuse of Software Licensed under the GNU General Public License. In: BOLDYREFF, C., CROWSTON, K., LUNDELL, B., WASSERMAN, A. I. (eds.). *Open Source Ecosystems: Diverse Communities Interacting. OSS 2009. IFIP Advances in Information and Communication Technology*. Vol 299. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-02032-2_17

HE, Z., PETERS, F., MENZIES, T. and YANG, Y. 2013. Learning from Open-Source Projects: An Empirical Study on Defect Prediction. In: *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement.* Baltimore, MD, USA, 2013, pp. 45-54, doi: 10.1109/ESEM.2013.20

KUMAR, N. 2022. *Export Compliance as a Response To Export Control*. Rel. Guido Sassi. Politecnico di Torino, Corso di laurea magistrale in Ingegneria Gestionale (Engineering And Management).

LINDMAN, J., PAAJANEN, A. and ROSSI, M. 2010. Choosing an Open Source Software License in Commercial Context: A Managerial Perspective. In: *36th EUROMICRO Conference on Software Engineering and Advanced Applications. L*ille, France, 2010, pp. 237-244, doi: 10.1109/SEAA.2010.26

MATHUR, A., CHOUDHARY, H., VASHIST, P., THIES, W. and THILAGAM, S. 2012. An Empirical Study of License Violations in Open Source Projects. In: *35th Annual IEEE Software Engineering Workshop.* Heraklion, Greece, 2012, pp. 168-176, doi: 10.1109/SEW.2012.24

SHERAE, D. and STEWART, K. 2016. Open source project success: Resource access, flow, and integration. *The Journal of Strategic Information Systems*, 25(3), 159-176. https://doi.org/10.1016/j.jsis.2016.02.006

SHIM, S.-R. 2011. A Comparative Study on the Compliance Program(CP) of Strategic Export Control System between Korea and Japan. *International Commerce and Information Review*.

TOURANI, P., ADAMS, B. and SEREBRENIK, A. 2017. Code of conduct in open source projects. In: *IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER).* Klagenfurt, Austria, 2017, pp. 24-33, doi: 10.1109/SANER.2017.7884606

MARYKA, T., GERMAN, D. M. and POO-CAAMAÑO, G. 2015. On the Variability of the BSD and MIT Licenses. In: DAMIANI, E., FRATI, F., RIEHLE, D., WASSERMAN, A. (eds.). *Open Source Systems: Adoption and Impact. OSS 2015. IFIP Advances in Information and Communication Technology*. Vol 451. Springer, Cham. https://doi.org/10.1007/978-3-319-17837-0_14

Contact information
Ashish Bakshi: e-mail: xbakshi@mendelu.cz
Andreas Kotulla: e-mail: andreas.kotulla@bitsea.de
Oldřich Faldík: e-mail: oldrich.faldik@mendelu.cz
Oldřich Trenz: e-mail: oldrich.trenz@mendelu.cz