

Hands-on Training
Software Development Tools
Winter Semester 21/22

TuCan-No: 20-00-0673-pr
Course Type: 4SWS / 6 CPs
Workload: ~**180hours**



Prof. Dr.-Ing. Mira Mezini

Process

- **Today:** Send an e-mail with your three preferred topics and your knowledge about the topics to: **glanz@cs.tu-darmstadt.de**
- **Tomorrow:** Assignment of topics
- **Next:** Contact your supervisor to discuss details of your topic
- **During Practicum:** Bi-weekly meetings with supervisor in an agile process
 - Discuss the current state and the next steps
- **End of March:** Final submission of artifacts



Dummy Library / App Usage

For a data flow analysis, the usage of libraries / apps needs to be simulated by performing dummy usages of the methods and classes.

- **Task:** Generate code that constructs objects of classes and method usages to support data flow analysis
- **Languages & Frameworks:**  **Scala**  **OPAL**
- **Suitable for:** 2 – 3 people
- **Contact:** glanz@cs.tu-darmstadt.de



Loophole Analyses

In programs, information is often written to files or other destinations and is therefore overlooked by data flow analyses.

- **Task:** Develop an analysis that identifies the targets writes and reads of files and other loopholes.
- **Languages & Frameworks:**  **Scala**  **OPAL**
- **Suitable for:** 1 – 2 people
- **Contact:** glanz@cs.tu-darmstadt.de

Integrating & Developing Small Static Analyses

Many apps contain information that indicates vulnerabilities or privacy violations.

- **Task:** Develop & integrate multiple analyses that identify indications of vulnerabilities or privacy violation like poor crypto usage.
- **Languages & Frameworks:**  **Scala**  **OPAL**
- **Suitable for:** 3 – 4 people
- **Contact:** glanz@cs.tu-darmstadt.de

Analysis of the App Development Landscape

These days, apps are no longer completely customized for different platforms by one developer. Frameworks exist that allow development across platforms.

- **Task:** Develop a script that identifies different (cross-) compilers
- **Languages & Frameworks:**  **Scala**  **OPAL**
- **Suitable for:** 2 – 3 people
- **Contact:** glanz@cs.tu-darmstadt.de

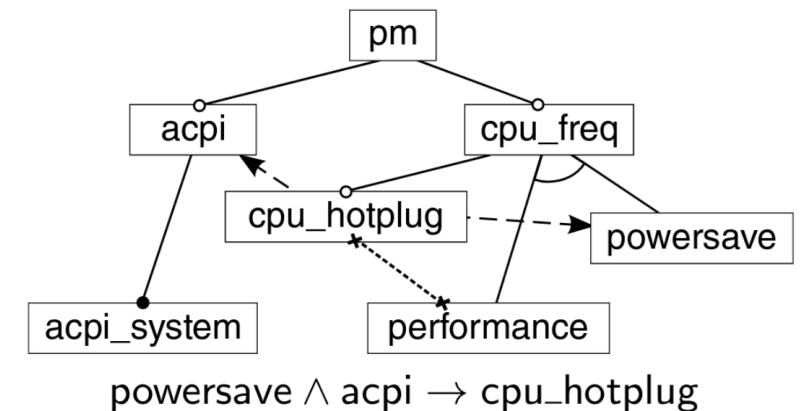
Java 17 Support for OPAL

OPAL, STG's static analysis framework for Java Bytecode should support Java 17 features, in particular sealed classes.

- **Task:** Implement support for Java 17(+) bytecode features, including parsing, generation and tests.
- **Languages & Frameworks:**  **Scala**  **OPAL**
- **Suitable for:** 1 person
- **Contact:** helm@cs.tu-darmstadt.de

Extraction of Feature Models

- **Task:** Implement a framework that extracts Software Product Line feature models from C source code.
- **Languages & Frameworks:**  **Scala**
- **Suitable for:** 2 – 3 people
- **Contact:** mueller@cs.tu-darmstadt.de



Android APK Handling

To facilitate static analyses of Android applications in OPAL, we need to apply several converters.

- **Task:** Implement a framework that applies several converters (native, dalvik) on a given Android apk.

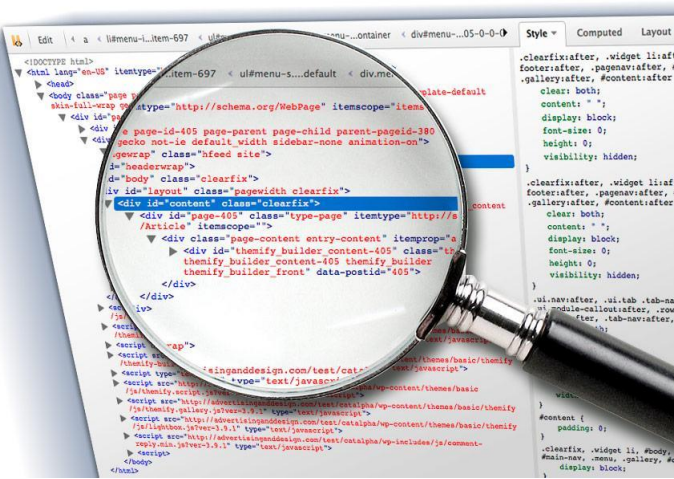
- **Languages & Frameworks:**  **Scala**  **OPAL**  **python** 

- **Suitable for:** 1 – 2 people

- **Contact:** mueller@cs.tu-darmstadt.de

C Source Code Transformation

- **Task:** Write a transformation for C code, based on static analysis results, using LLVM and clang.
- **Languages & Frameworks:**  
- **Suitable for:** 1 – 2 people
- **Contact:** mueller@cs.tu-darmstadt.de



Web Crawler – Analyzing the State of the Art of the Web

For creating appropriate analyses of web applications we need informations about the state-of-the-art applied technology.

- **Task:** Create a program that crawls a list of given webpages and stores the following information in a database:
 - Used languages and frameworks
 - Embedded links and libraries
 - Is WebAssembly used and if so, how?
 - ...

• **Languages & Frameworks:**



Selenium



• **Suitable for:** 1 – 2 people

• **Contact:** roth@cs.tu-darmstadt.de

breitfelder@cs.tu-darmstadt.de

Go Unsafe Toolkit

The usage of the *unsafe* library in Go allows developers to circumvent its memory protection and can introduce security vulnerabilities. *go-geiger* helps developers to spot usages of *unsafe* in their code. Machine-learning can be used to classify the reason and context of this usage.

- **Task:** Automate unsafe usage identification, classification and reporting
- **Languages & Frameworks:** Go & Shell & Docker & Python (?)
- **Suitable for:** 1 people
- **Contact:** wickert@cs.tu-darmstadt.de



Positions & Theses

If you are interested in **HiWi Positions** or **Bachelor- or Master theses** contact:

glanz@cs.tu-darmstadt.de