TRACER: A Platform for Securing Legacy Code

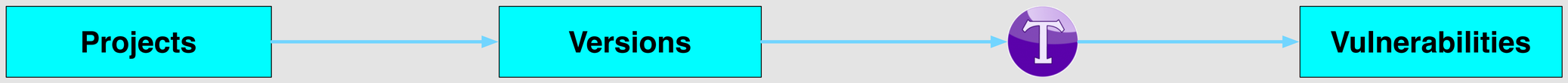
Kostantinos Stroggylos*, Dimitris Mitropoulos*, Zacharias Tzermias^, Panagiotis Papadopoulos^, Fotios Rafailidis', Sotiris Ioannidis^, Diomidis Spinellis* and Panagiotis Katsaros'

*Athens University of Economics and Business
Department of Management Science and Technology

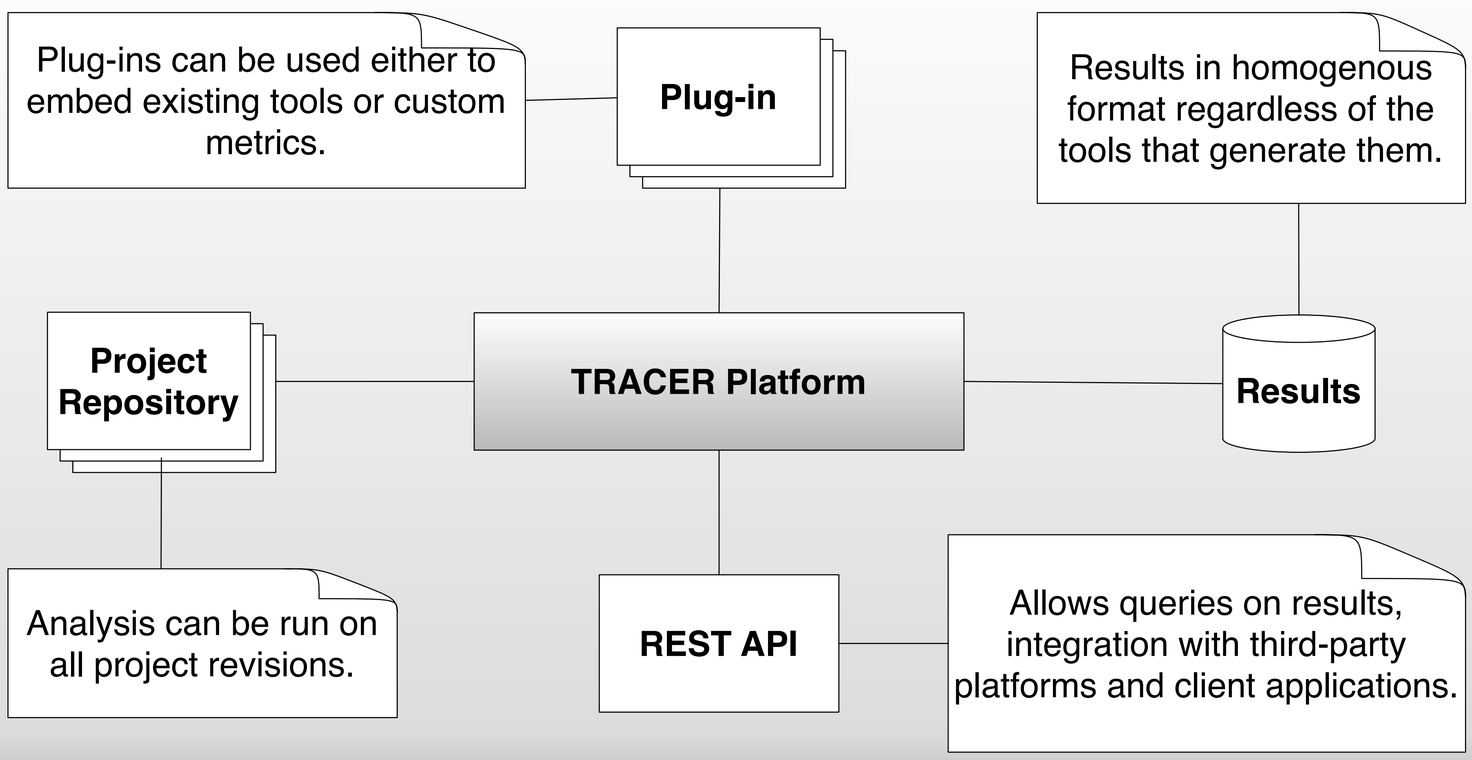
^Institute of Computer Science
Foundation for Research and Technology - Hellas

'Department of Informatics
Aristotle University of Thessaloniki

TRACER is a framework to support the development of secure applications by constantly monitoring software projects for vulnerabilities. TRACER simplifies the integration of existing tools that detect software vulnerabilities and promotes their use during development and maintenance.



TRACER High Level Architecture.

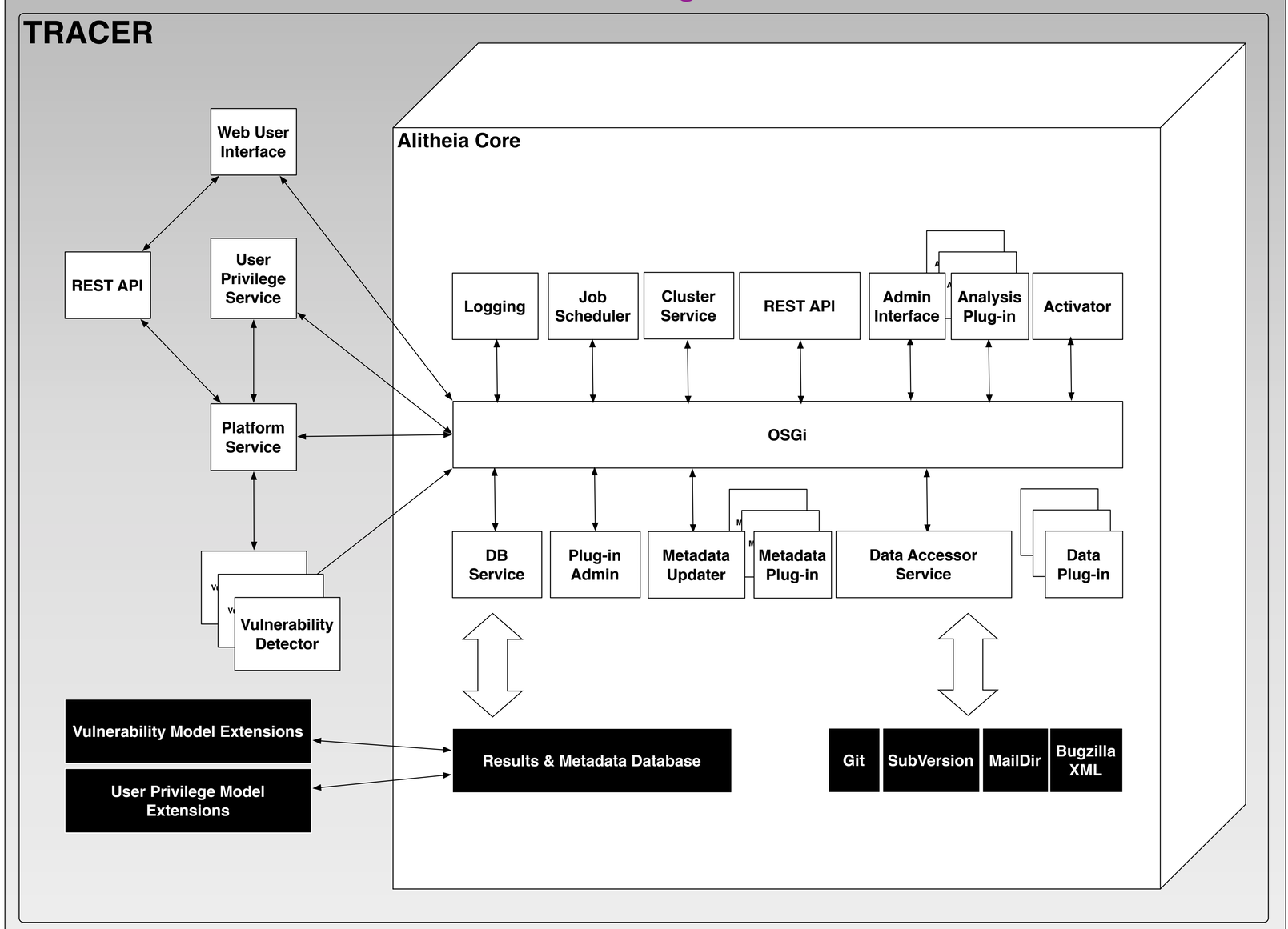


TRACER can examine applications written in **different programming languages** and can secure them against the various, constantly evolving threats. This is done by either **e**mbedding **existing static analysis tools** or calculating **specific security metrics** written by individual developers.

Since the formats in which such tools store and present their results **varies** wildly, it is inherently difficult to utilize a number of them effectively on a software project. tracer simplifies this process by providing a platform to run such tools in an automated manner. Moreover, by using a common representation for metrics and results regardless of the tool that was used to generate them, it enables their analysis, presentation and visualization in a **homogenous way**.

To evaluate our platform, we have created plug-ins to integrate two different tools for vulnerability detection, namely: **FindBugs**, and **Frama-C**.

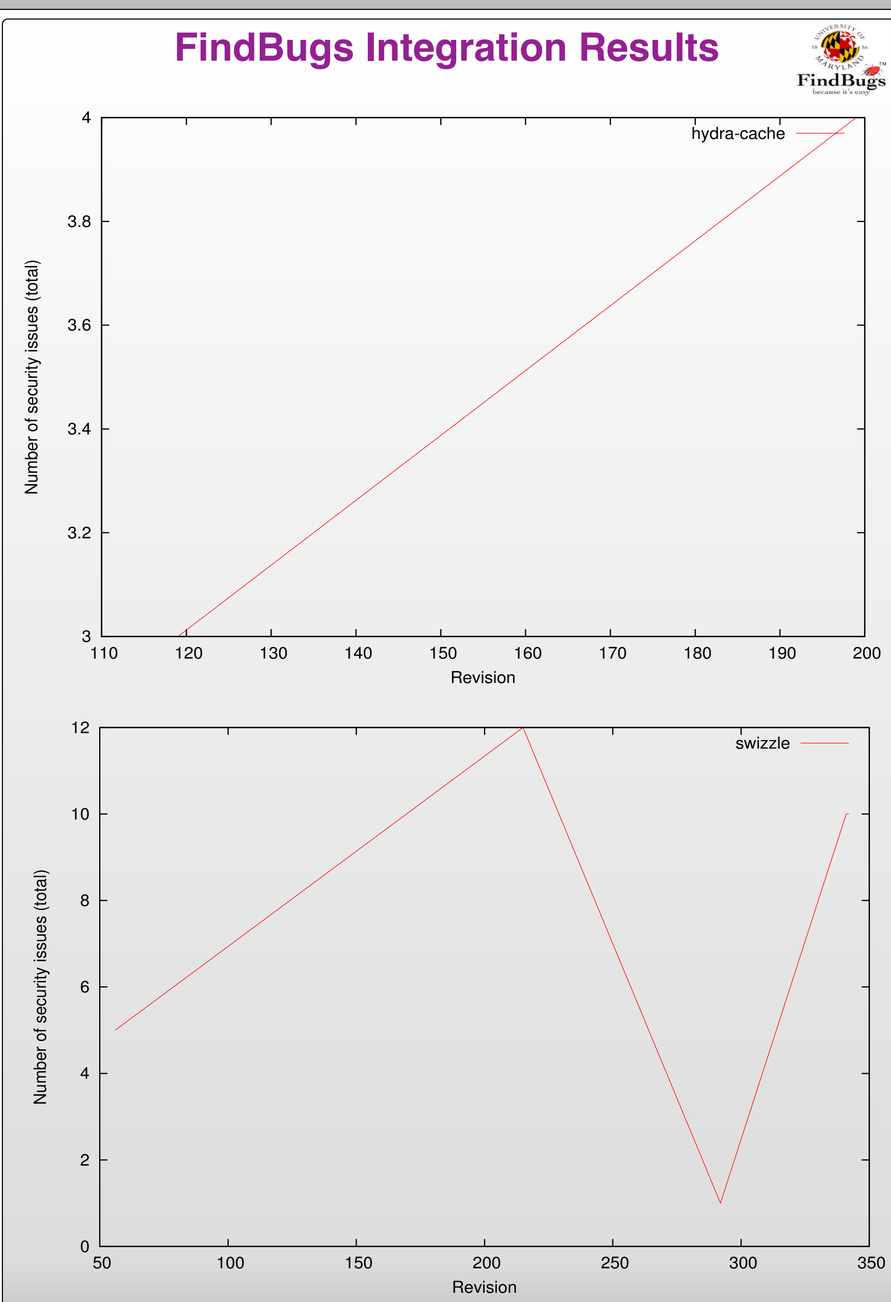
Alitheia Core Integration



Instead of designing and implementing TRACER from the ground up, we built it on top of the open source **Alitheia Core** platform, which is designed for facilitating large scale quantitative software engineering studies. To support the specific objectives of TRACER, a set of new components was added at each level of the Alitheia Core architecture. These include a model for representing software vulnerabilities, a mechanism for automatic vulnerability detection triggering, a REST API for accessing the analysis results, and an archetype for **plug-ins** to integrate new vulnerability detection tools in the platform. Like Alitheia Core, TRACER monitors multiple data sources associated with the development of a software project, such as the source code repository and bug tracking system, and automatically analyzes each revision. Therefore it can be used to track security defects throughout the **evolution** of a project.

On the right you can see that we have integrated and run FindBugs for every revision of two open source projects. Interestingly, security bugs are increasing as projects evolve, contrary to what one would hope.

FindBugs Integration Results



FRAMA-C Integration Results

Project	Vulnerability	Untainted
Clearsilver-0.10.5	Format string	✔
	Double free	✔
	User kernel trust error	✔
	SQL injection attack	✔
	Cross-site-scripting	✔
MCrypt-2.6.8	Format string	✔
	Double free	✔
	User kernel trust error	✔
	SQL injection attack	✔
	Cross-site-scripting	✔

The project is being co-financed by the European Regional Development Fund (ERDF) and national funds and is a part of the Operational Programme "Competitiveness & Entrepreneurship" (OPCE ii), Measure "COOPERATION" (Action i).