JIT Knowledge Centralization and Sharing for JavaScript and Python

- Supervisor: Teabe Boris (boris.teabedjomgwe@enseeiht.fr)
- Location: remotely

Modern runtimes equipped with a Just-In-Time (JIT) compiler make optimization decisions based on information collected locally during execution: observed types, execution frequency, hot paths, compilation decisions, specializations, or deoptimizations. In Cloud and Serverless environments, multiple instances of the same application can independently collect similar information and repeat the same learning and optimization work. This project proposes studying an architecture to centralize, aggregate, and share the knowledge produced by JIT compilers across different instances of the same application. The study will focus on two runtimes, JavaScript, specifically with V8 (Node.js), and Python (PyPy), to identify which information can be shared despite differing optimization strategies. An instance could thus benefit from observations made by others, for example to avoid an optimization that regularly led to a deoptimization, anticipate frequently observed types or execution paths, or select an appropriate compilation strategy more quickly. The goal will be to determine which JIT knowledge can be shared, how to evaluate its stability and transferability between instances, and to what extent this approach reduces warm-up, unnecessary compilations, and the cost of deoptimizations in Cloud and Serverless environments.