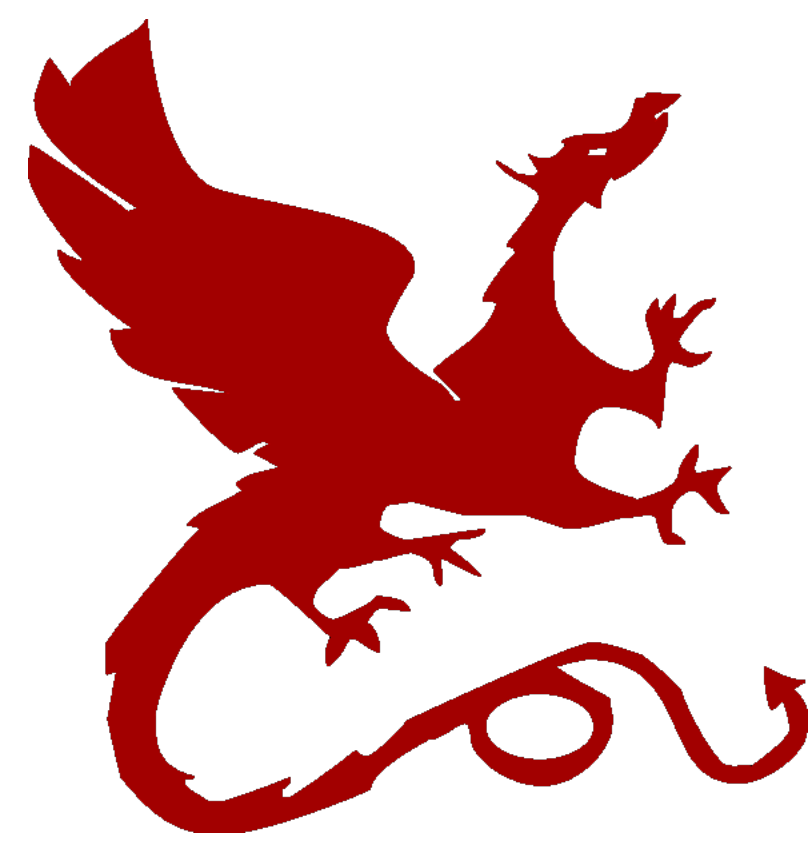


Motivation

- Obsidian is a new programming language for blockchain smart contracts which includes a sophisticated type system designed to catch common classes of smart contract bugs at compile time.
- Editor integration can dramatically speed up the process of learning a new type system. It can present errors while the programmer is actively thinking about the code being written



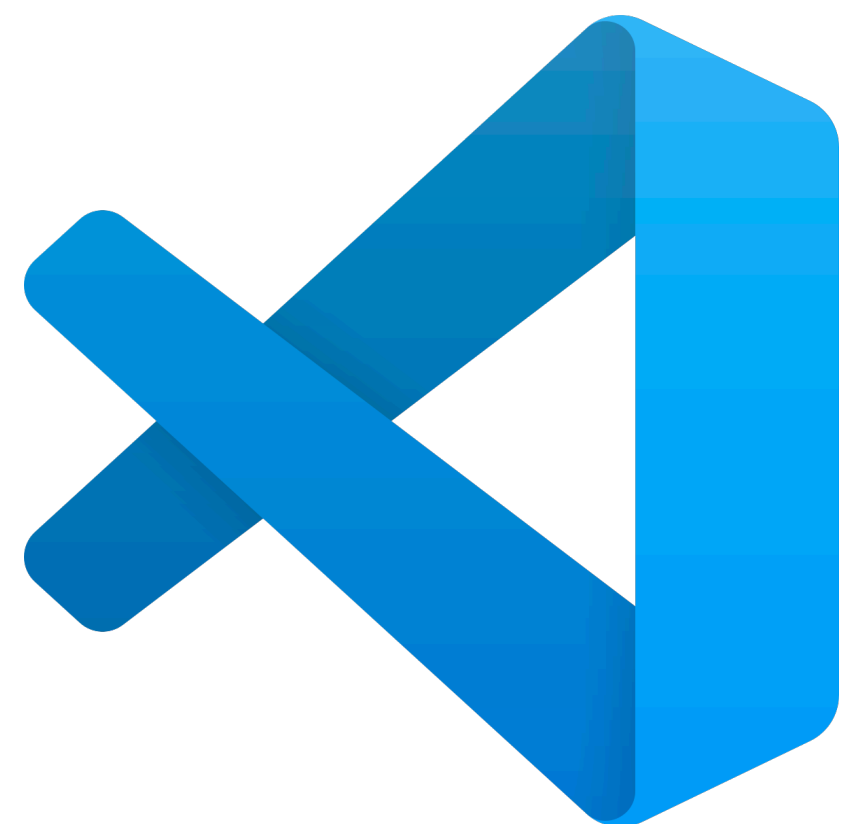
Carnegie Mellon

Ishan Bhargava
(ibhargav)
Michael Coblenz
(mcoblenz)

Abstract

Traditionally, programmers had to write hundreds of lines of code, compile it, and then go back and fix the inevitable compilation errors. This hampers developer productivity, as the programmer may end up working on disjoint sections of the code before running the compiler. This project aims to improve the Obsidian developer experience through editor integration

Obsidian Language Server



Editor integration is implemented for Microsoft's Visual Studio Code, but other editors which support the Language Server Protocol are also compatible.

As the Language Server Protocol works best with Node.js, a Node.js server marshals the source code data from the editor to the compiler, and the diagnostics reported from the compiler back to the editor.

The compiler itself had to be modified to support parsing/typechecking programs which may not have been saved to disk, or which may have dependencies in a different directory than the file's location.

Examples

```
tr Cannot assign a value of contract  
Option[Car@Unowned] to a variable that requires a  
value of contract Option[Car@Owned].  
Peek Problem (⌘F8) No quick fixes available  
Option[Car@Owned] prevCar = cars.peek(key);  
switch prevCar {  
case Some {
```

```
transaction addCar(int id, string color, string make, st  
Variable 'prevCar' is incompatibly typed as both  
'Option[Car@Some]' and 'Option[Car@Shared]' after  
branch.  
Peek Problem (⌘F8) No quick fixes available  
switch prevCar {  
case Some {  
revert("Car with that ID already exists");  
}
```

```
Car@Owned() {  
// Add some sample data  
Car tom Missing specification for a value for the permission  
Car bra variable s in asset ValueType@s@Owned where s is Owned  
Car jin in type parameter Car.  
Peek Problem (⌘F8) No quick fixes available  
cars = new Dict[Integer, Car](new IntegerComparator());  
insertCar(0, tomoko);  
insertCar(1, brad);  
insertCar(2, jinsoo);
```