# A Rustic JavaScript Interpreter

**Terry Sun, Sam Rossi**
**Prof. Steve Zdancewic**

**Abstract**

- Js.rs is a prototype server-side JavaScript interpreter, implementing a core subset of language features
- JavaScript is ubiquitous—runs on every computer and browser
- Written entirely in Rust, Js.rs provides memory safety guarantees, unlike existing C/C++ interpreters, with low performance overhead

**Language Features**

- Functions & function closures
  ```javascript
  function add() {
    var x = 0;
    return function() { x += 1; };  
  }
  ```
- Control Flow
  ```javascript
  if {...} else if {...} else {...}
  for {...} (i = 0; i < 10; i++) {...}
  ```
- Error Handling
  ```javascript
  try {...} catch {...} finally {...}
  ```
- Unary, Binary Operators
  ```javascript
  x <<= (2 + y);
  ```
- Objects, Arrays, and Prototypes
  ```javascript
  var a = [1, 2, 3];
  var b = { a: "hello, world!" };
  ```

**Design**

1. The parser builds an Abstract Syntax Tree from the source code
   - The AST encodes the structure of the program

2. The runtime traverses the AST:
   - Executes the code in each node
   - Maintains a state of local variables

**Results**

- Sputnik is Google’s ECMAScript 3 conformance test suite
- Js.rs passes tests in 73 of 111 (66%) selected categories from Sputnik
- Rust is a robust language to work in, but available tools are fragile