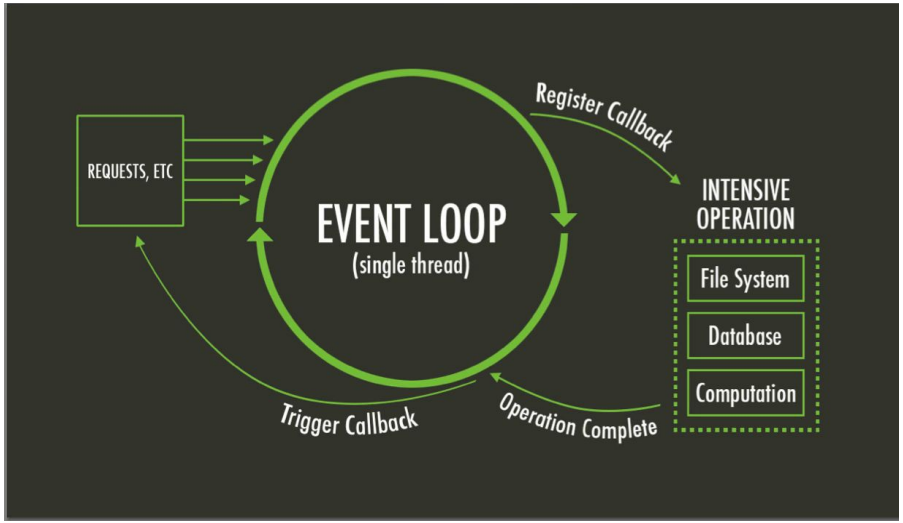
Node Isolates

Claire Furtick, Linnea Dierksheide, Genevieve Flynn

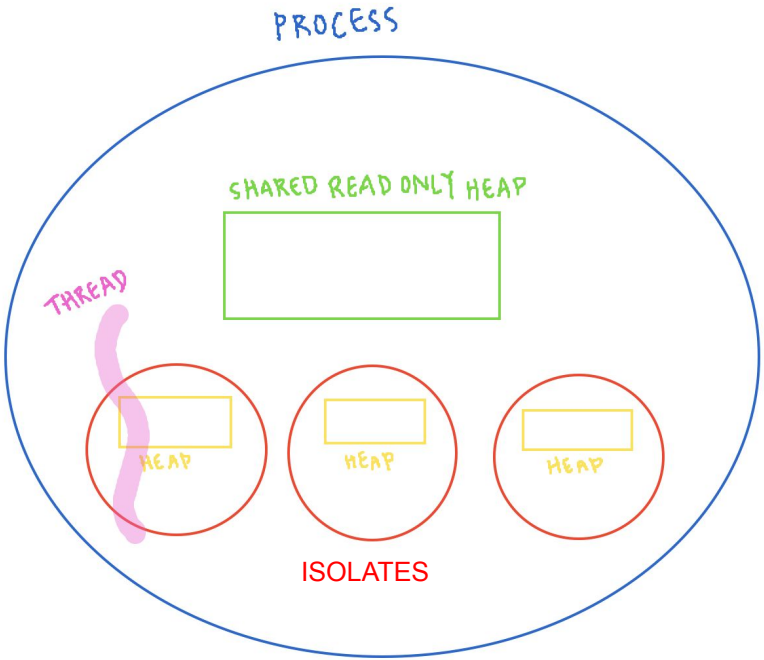
Background

- v8: Google Javascript Engine
- Provides memory management, compiler, etc
- Node.js: Javascript Runtime
- Node.js enabled server-side JS; easier for web developers
- Single-threaded execution model



v8 Isolates

- Each isolate is an instance of v8
- Memory isolation within a process
- 1:1 isolates to threads
- 1:n isolates to contexts
- Fast startup/context switching



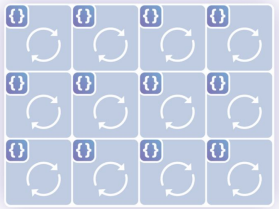
Isolated-vm: JS Library

```
// Create a new isolate limited to 128MB
const ivm = require('isolated-vm');
const isolate = new ivm.Isolate({ memoryLimit: 128 });

// Create a new context within this isolate. Each context has its own copy of all the builtin
// Objects. So for instance if one context does Object.prototype.foo = 1 this would not affec
// other contexts.
const context = isolate.createContextSync();

// Get a Reference{} to the global object within the context.
const jail = context.global;

// This makes the global object available in the context as `global`. We use `derefInto()` he
// because otherwise `global` would actually be a Reference{} object in the new isolate.
jail.setSync('global', jail.derefInto());

// We will create a basic `log` function for the new isolate to use.
jail.setSync('log', function(...args) {
        console.log(...args);
});

// And let's test it out:
context.evalSync('log("hello world")');
// > hello world
```

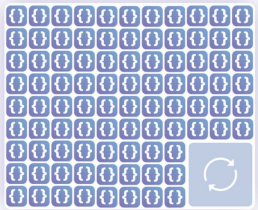
```
int main(int argc, char* argv[]) {
  // Initialize V8.
  v8::V8::InitializeICUDefaultLocation(argv[0]);
  v8::V8::InitializeExternalStartupData(argv[0]);
  std::unique_ptr<v8::Platform> platform = v8::platform::NewDefaultPlatform();
  v8::V8::InitializePlatform(platform.get());
  v8::V8::Initialize();

  // Create a new Isolate and make it the current one.
  v8::Isolate::CreateParams create_params;
  create_params.array_buffer_allocator =
      v8::ArrayBuffer::Allocator::NewDefaultAllocator();
  v8::Isolate* isolate = v8::Isolate::New(create_params);
  {
    v8::Isolate::Scope isolate_scope(isolate);

    // Create a stack-allocated handle scope.
    v8::HandleScope handle_scope(isolate);

    // Create a new context.
    v8::Local<v8::Context> context = v8::Context::New(isolate);

    // Enter the context for compiling and running the hello world script.
    v8::Context::Scope context_scope(context);

    // Create a string containing the JavaScript source code.
    v8::Local<v8::String> source =
        v8::String::NewFromUtf8(isolate, "'Hello' + ', World!'",
                                v8::NewStringType::kNormal)
            .ToLocalChecked();

    // Compile the source code.
    v8::Local<v8::Script> script =
        v8::Script::Compile(context, source).ToLocalChecked();

    // Run the script to get the result.
    v8::Local<v8::Value> result = script->Run(context).ToLocalChecked();

    // Convert the result to an UTF8 string and print it.
    v8::String::Utf8Value utf8(isolate, result);
    printf("%s\n", *utf8);
  }

  // Dispose the isolate and tear down V8.
  isolate->Dispose();
  v8::V8::Dispose();
  v8::V8::ShutdownPlatform();
  delete create_params.array_buffer_allocator;
  return 0;
}
```

Why Isolates?

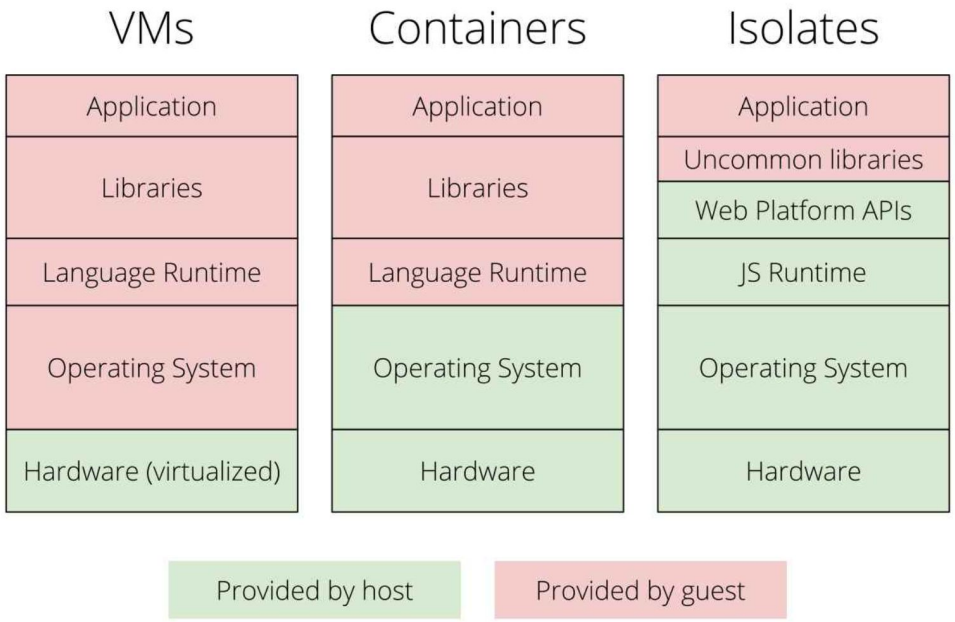
- Cloudflare uses them as a serverless technique instead of the traditional VMs and containers



Virtual machine



Isolate model



Why Isolates part 2

- Using isolated-vm you can create your own lightweight sandboxes
- Screeps, an MMO sandbox game for programmers -- running many players' code at once requires isolation between them