lunatic

Erlang-inspired runtime for Rust

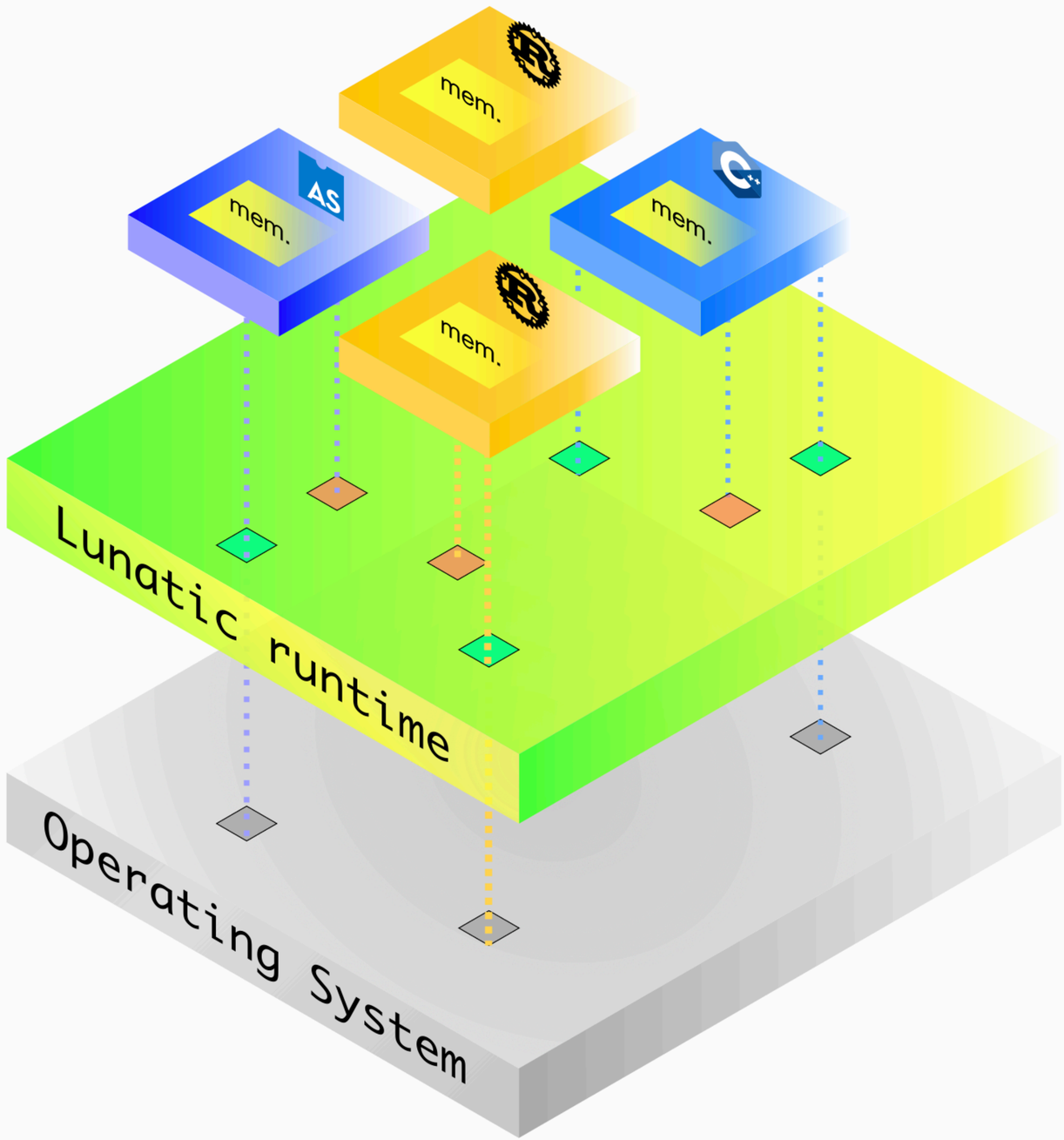
Grain

AssemblyScript

Bernard / 25.05.2023

Properties
of lunatic

- Concurrency
- Fault tolerance
- Sandboxing
- Distributed lunatic



Concurrency

(how to wait on stuff)

```
line = [0; 512];
let result = stream.read(&mut line);
```

Threads

- OS
- Usually 8 MB stack
- Managed by the kernel

Async Tasks

- Library
- Usually few k(bytes)
- Managed in user space

Lunatic processes

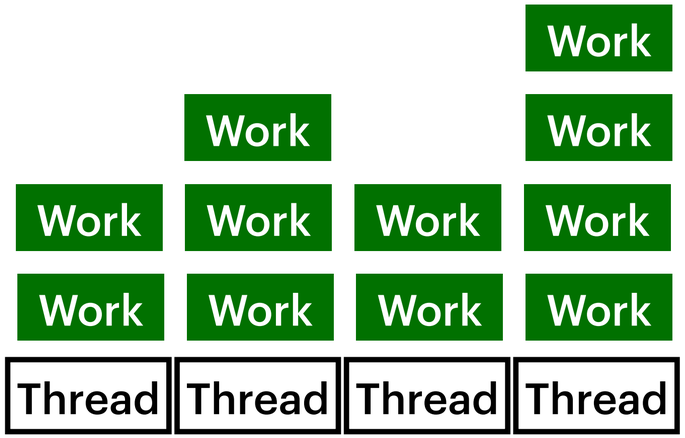
- **Runtime**
- **1 MB virtual stack**
- **Managed in user space**

Concurrency

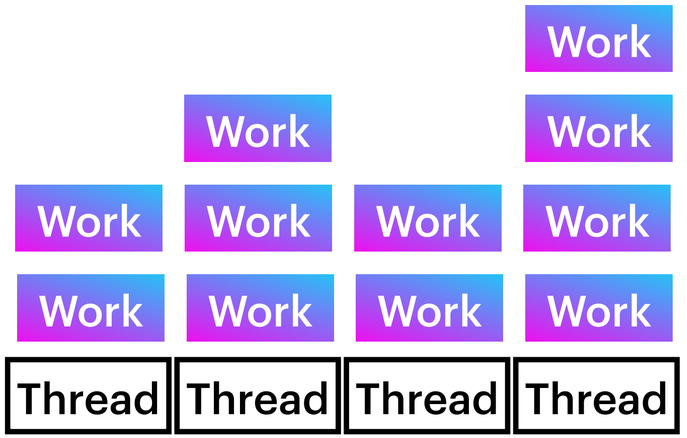
(how to wait on stuff)

Threads

Async Tasks



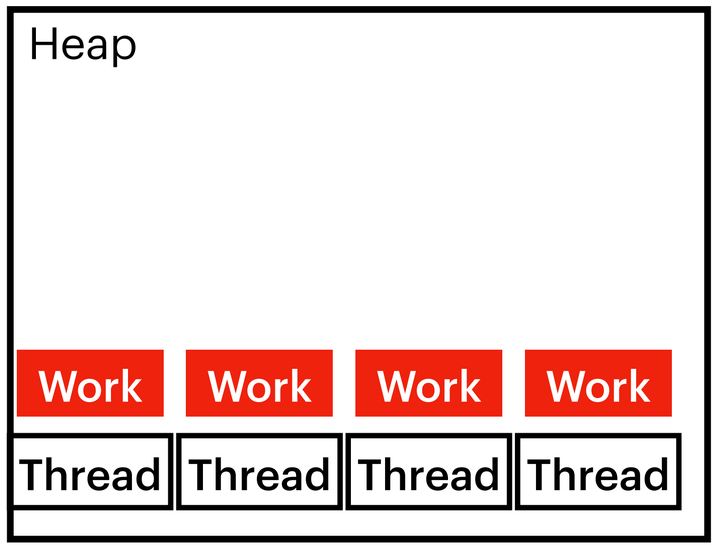
Lunatic processes



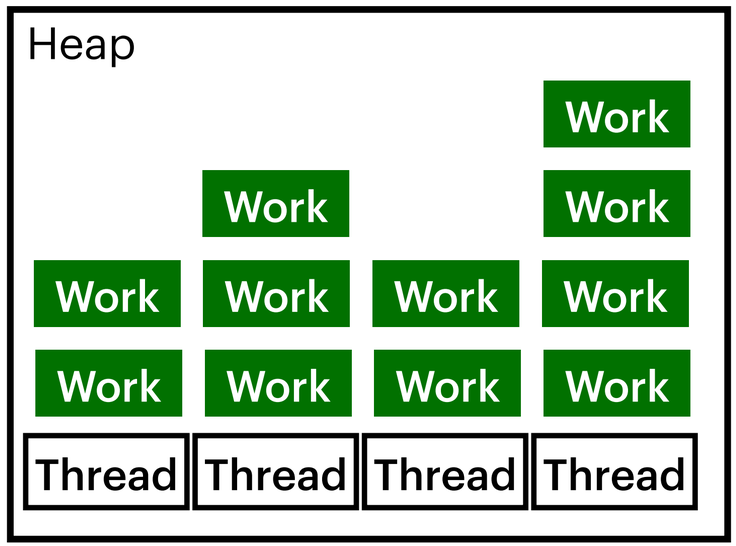
Concurrency

(how to wait on stuff)

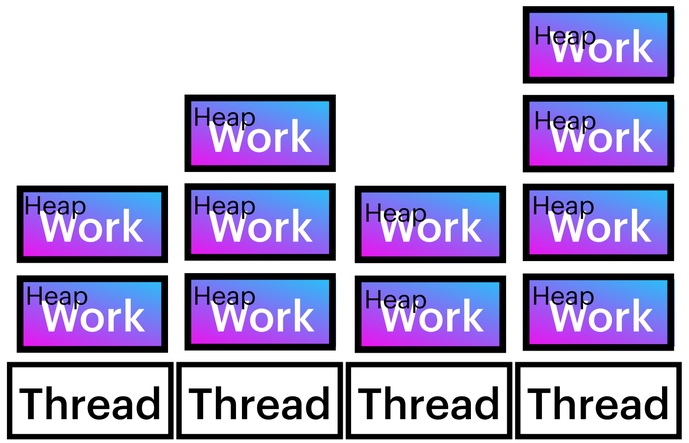
Threads



Async Tasks

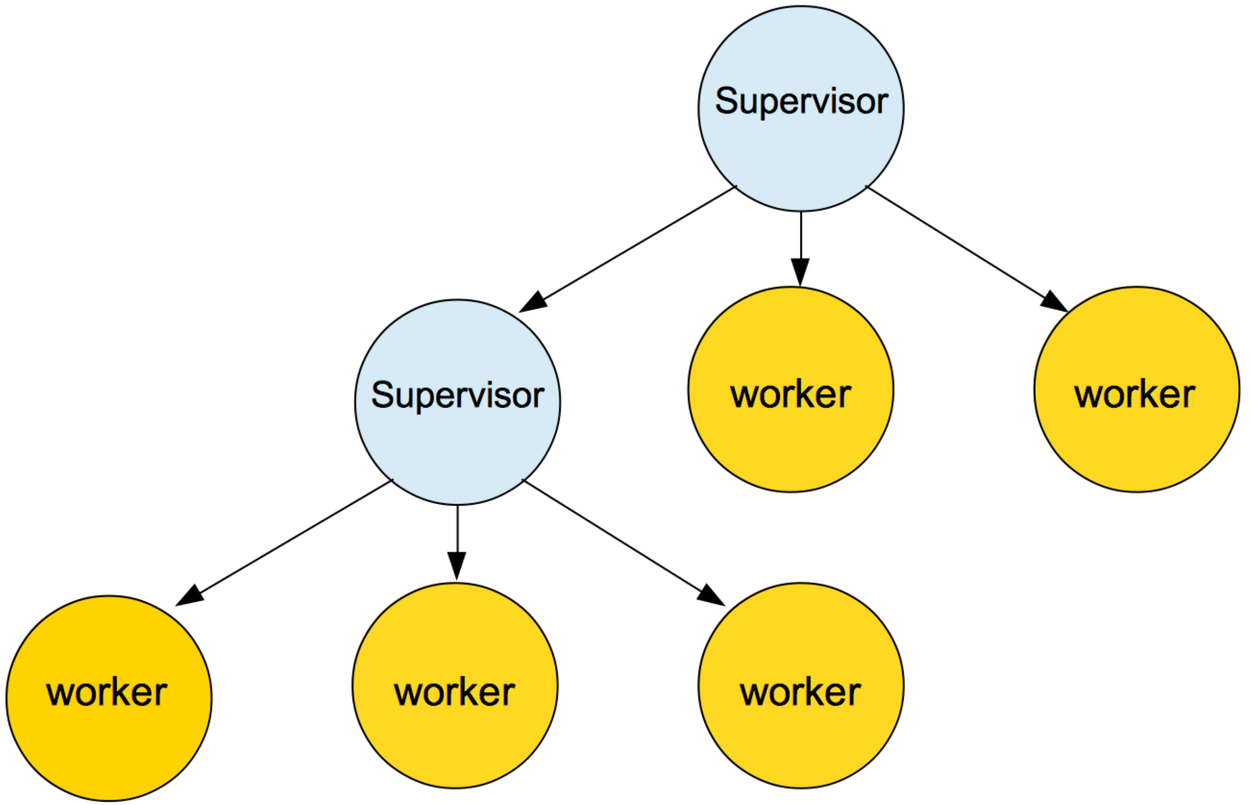


Lunatic processes



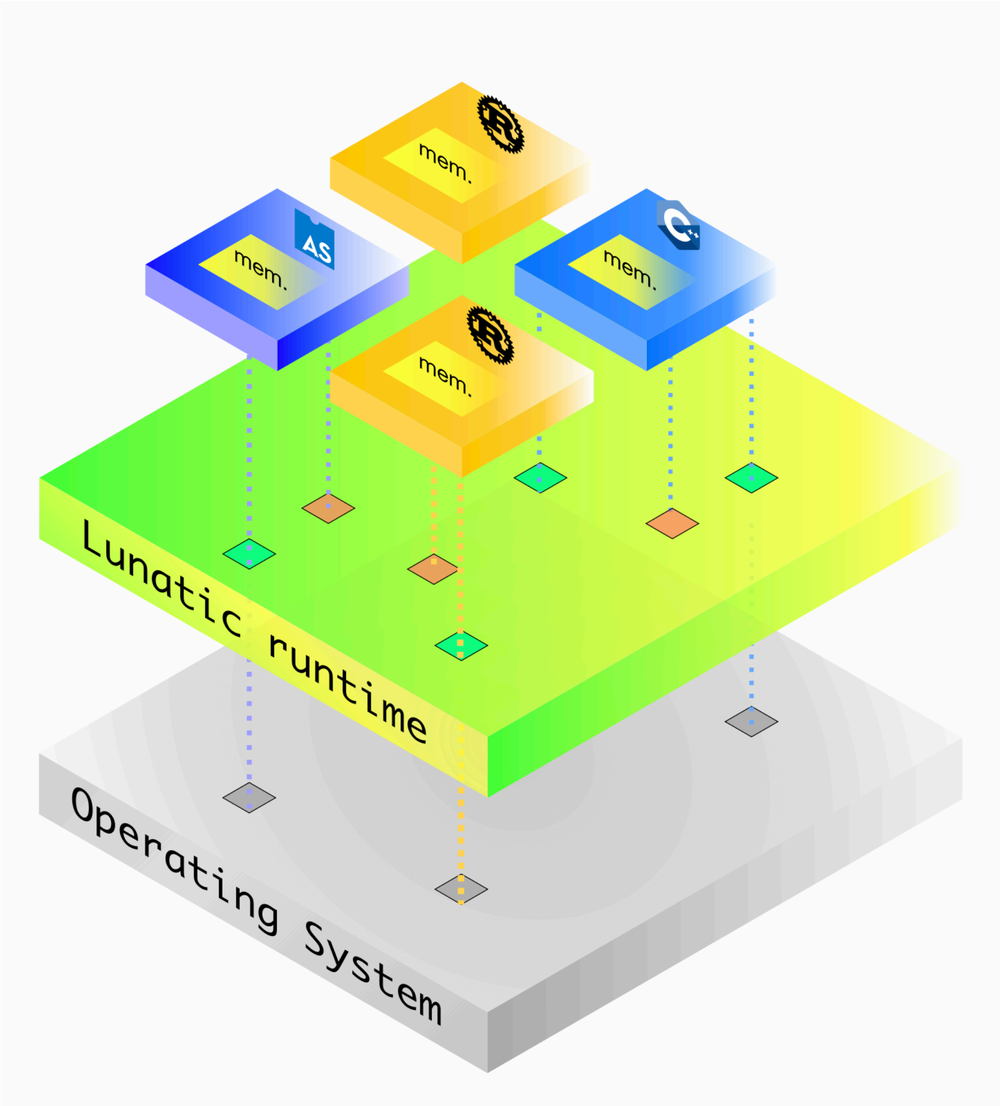
Fault tolerance

Supervisors

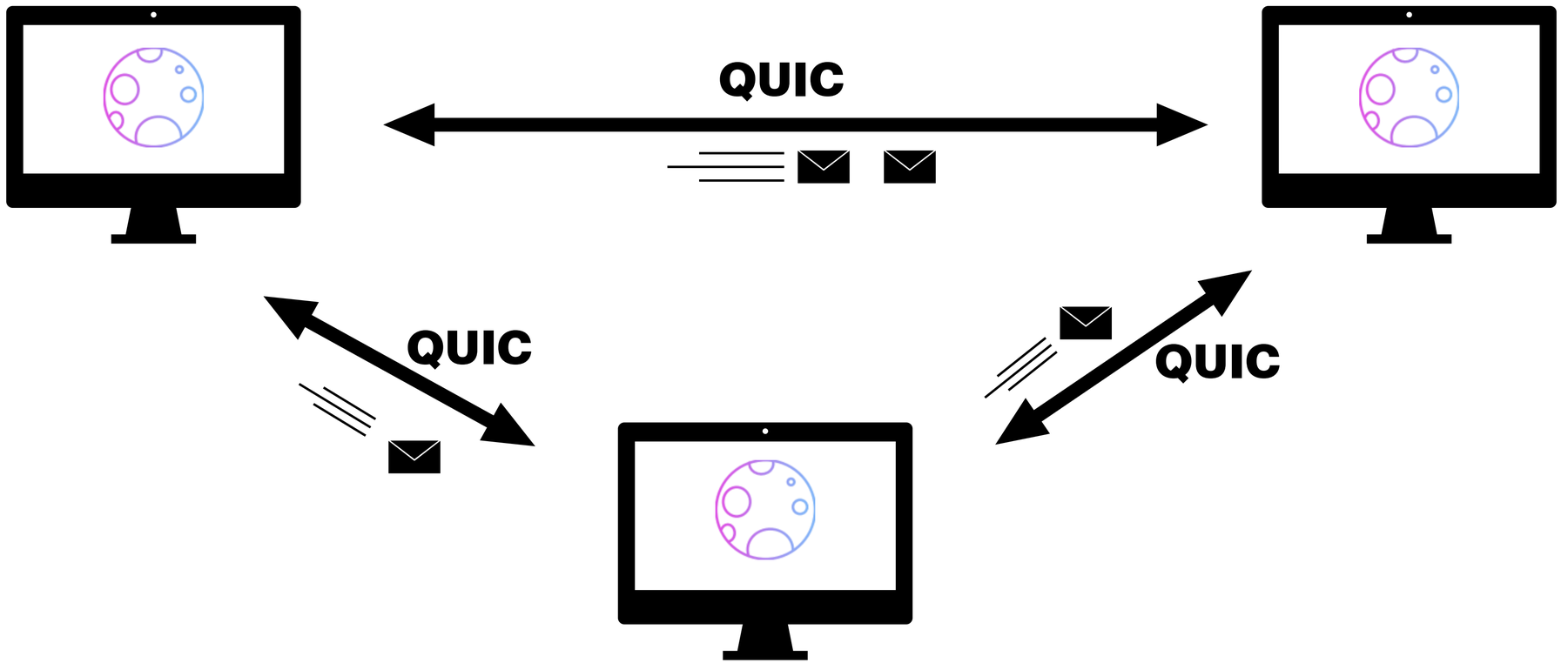


Sandboxing

- WebAssembly is deterministic
- Every “syscall” goes through the VM
- Compute & memory limits per process



Distributed lunatic



Submillisecond live-view demo