```cpp
void set_global_string (std::string*);

void entry_point (std::shared_ptr<synchronized_value<std::string>> sync_s, int tid)
{
    apply ([tid] (auto& s) {
        set_global_string (&s);
        //...
        return s;
    },
    *sync_s);
}
```

```rust
fn entry_point(data: &Mutex<String>, thread_id: i32) {
    let mut guard = data.lock().unwrap();
    guard.push_str("🔥");
    println!("Thread {}: {}", thread_id, *guard);
}
```

Global pointers

```cpp
void set_global_string (std::string*);

void entry_point (std::shared_ptr<synchronized_value<std::string>> sync_s, int tid)
{
    apply ([tid] (auto& s) {
        set_global_string (&s);
        //...
        return s;
    },
    *sync_s);
}
```

```rust
fn entry_point(data: &Mutex<String>, thread_id: i32) {
    let mut guard = data.lock().unwrap();
    guard.push_str("🔥");
    println!("Thread {}: {}", thread_id, *guard);
}
```

Wrapping with Reflection