C++ Reflection to the Rescue?

- **Recursive Sync/Send Type Trait Checking**

- Check members of types are all sendable

- Check members of lambdas are all sendable

57

```
struct node
{
    node* next;
    node* prev;
};

std::shared_ptr<syncronized_value<node>>();
```

```
auto node = std::make_shared<node>();
safe_threads.emplace_back ([this, node]
                           {
                               member_function();
                           });
```

C++ Reflection to the Rescue?

- **Recursive Sync/Send Type Trait Checking**
  - Check members of types are all sendable
  - Check members of lambdas are all sendable

```
struct node
{
    node* next;
    node* prev;
};

std::shared_ptr<syncronized_value<node>>();
```

❌

```
auto node = std::make_shared<node>();
safe_threads.emplace_back ([this, node]
                           {
                               member_function();
                           });
```

❌

```
consteval auto is_send_type (std::meta::info type) -> bool
{
    type = remove_cv (type);

    // Non-member function pointers
    if (is_pointer_type (type)
        && is_function_type (remove_pointer (type))
        && ! is_member_function_pointer_type (type))
       return true;

    // lvalue refs and pointers
    if (is_lvalue_reference_type (type)
        || is_pointer_type (remove_extent (type)))
       return false;

    // POD built-in types
    if (is_arithmetic_type (type))
        return true;

    // Recursive class/struct/lambda members
    if (is_class_type (type))
        return std::ranges::all_of(nonstatic_data_members_of(type),
                                   [](std::meta::info d)
                                   {
                                       return is_send_type (type_of(d));
                                   });

    // Construct from rvalue ref
    if (is_rvalue_reference_type (type)
        && is_constructible_type (type, { remove_reference (type) }))
       return true;

    return false;
}
```