C++ Reflection to the Rescue?

- Recursive Sync/Send Type Trait Checking

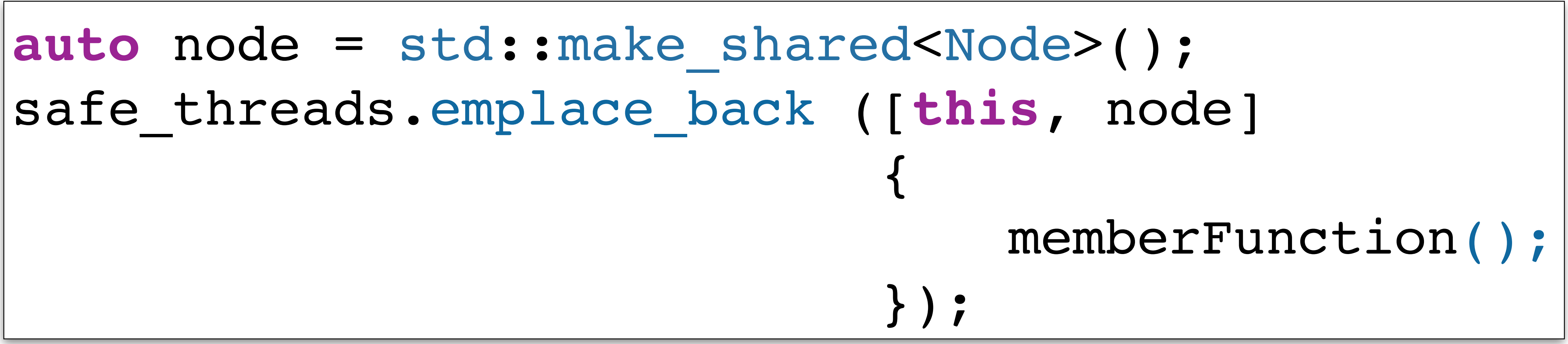
- Check members of types are all sendable

- Check members of lambdas are all sendable

103

```
template<typename T>
struct Node
{
    Node* next;
    Node* prev;
};

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



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

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

```
template<typename T>
struct Node
{
    Node* next;
    Node* prev;
};

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

❌

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

❌

Wrapping with Reflection