Wrapping with Reflection

- **Thread-safe wrappers**
  - `synchronized_value`
  - `std::mutex/shared_mutex/spin_lock`
  - `crill::seqlock_object`
- **Value wrappers around shared objects**
  - Automatic reference counting (arc)
  - Copy-on-write objects (cow)
- **Async classes**

62

- P2996 - Reflection for C++26
  Accepted ✅
  - P3294 - Code Injection with Token Sequences
    Hopeful for C++26 🔜
  - P3096 - Function Parameter Reflection in Reflection for C++26
    Proposed C++29 🙏
  - P3394 - Annotations for Reflection
    Proposed C++29 🙏
  - P0707 - Metaclasses
    Proposed C++29 🙏

C++29 😔

- P2996 - Reflection for C++26
  Accepted ✅
    - P3294 - Code Injection with Token Sequences
      Proposed C++29 🙏
- P3096 - Function Parameter Reflection in Reflection for C++26
  Accepted C++26 ✅
- P3394 - Annotations for Reflection
  Accepted C++26 ✅
    - P0707 - Metaclasses
      Proposed C++29 🙏

Wrapping with Reflection

Implicit `synchronized_value`

C++

- synchronized_value

- std::mutex/shared_mutex/spin_lock

- Copy-on-write objects (cow)

- **Async classes**

- Automatic reference counting (arc)

- `crill::seqlock_object`

- **Thread-safe wrappers**

- **Value wrappers around shared objects**

- P3096 - Function Parameter Reflection in Reflection for C++26

- P3294 - Code Injection with Token Sequences

- P2996 - Reflection for C++26

Proposed C++29 🙏

Accepted ✅

Proposed C++29 🙏

- P3394 - Annotations for Reflection

Proposed C++29 🙏

- P0707 - Metaclasses

Hopeful for C++26 🔜

C++29 😔

Proposed C++29 🙏

Accepted C++26 ✅

Accepted C++26 ✅

```
person() = default;
```

```
std::string get_first_name() const
```

```
class(synchronized) person
```

}

{

```
private:
```

```
std::string first_name, last_name;
```

```
void set_first_name (std::string_view new_first)
```

```
};
```

```
first_name = new_first;
```

```
// Repeat for last_name
```

```
public:
```

{

```
return first_name;
```