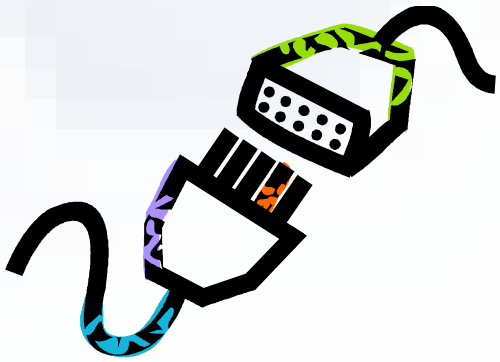
boostcon
BOOST LIBRARIES CONFERENCE 2011

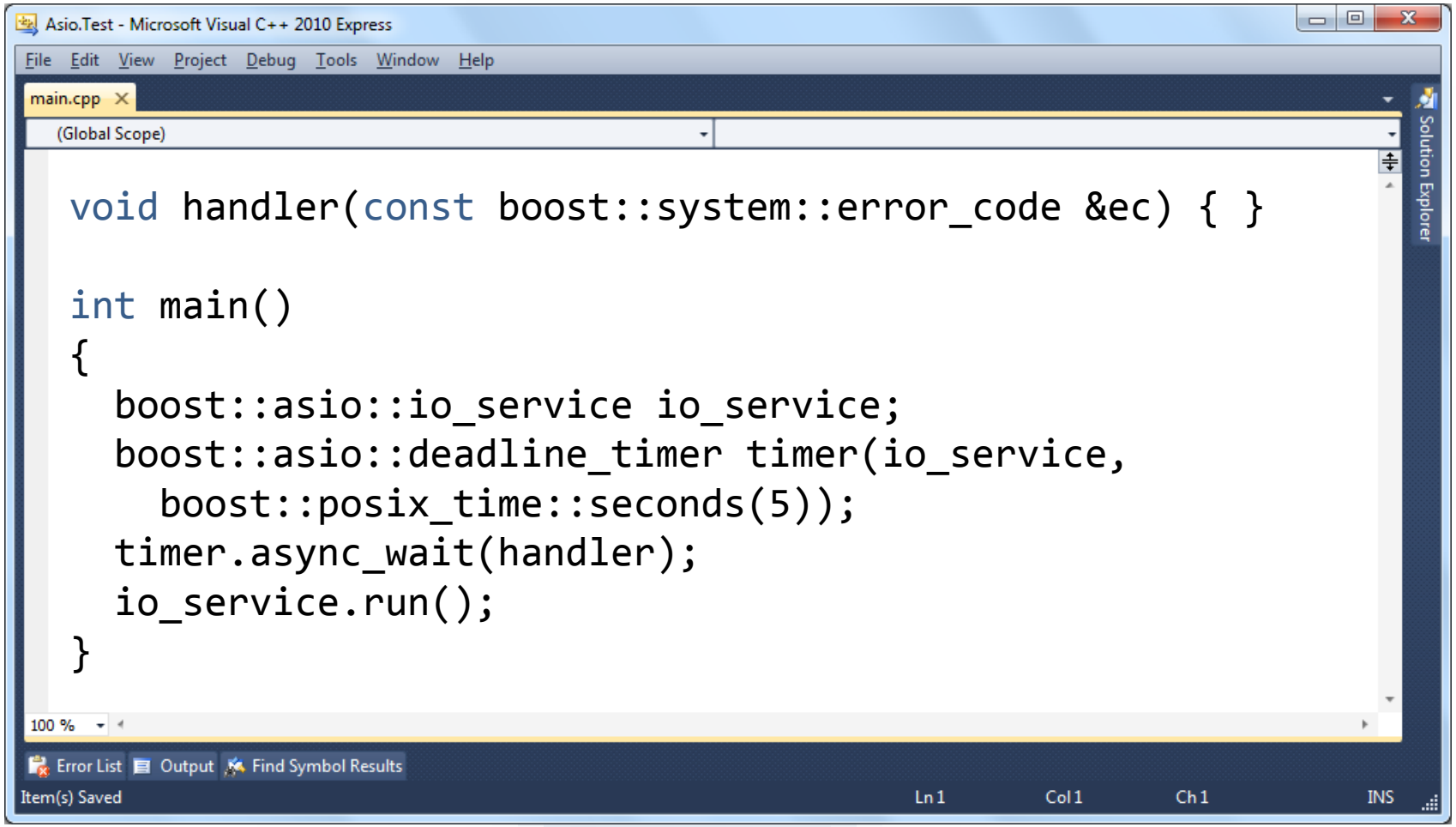
Creating Boost.Asio extensions

Boris Schäling, May 2011, www.highscore.de

- How does Boost.Asio look like internally?
- What are I/O service objects, I/O services and I/O objects?
- How do I access platform-specific I/O sevices?



Boost.Asio: Asynchronous functions



Boost.Asio: Asynchronous functions

Asio.Test - Microsoft Visual C++ 2010 Express

File Edit View Project Debug Tools Window Help

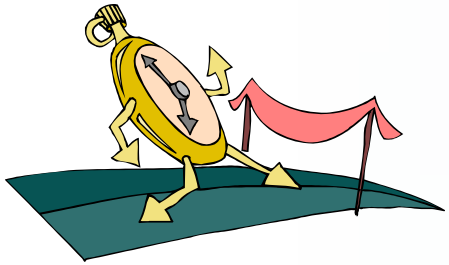
main.cpp

(Global Scope)

```
boost::asio::deadline_timer timer(io_service,
  boost::posix_time::seconds(5));
timer.async_wait(handler);
```

Solution Explorer

deadline_timer is one of many classes in Boost.Asio which make it possible to call functions asynchronously



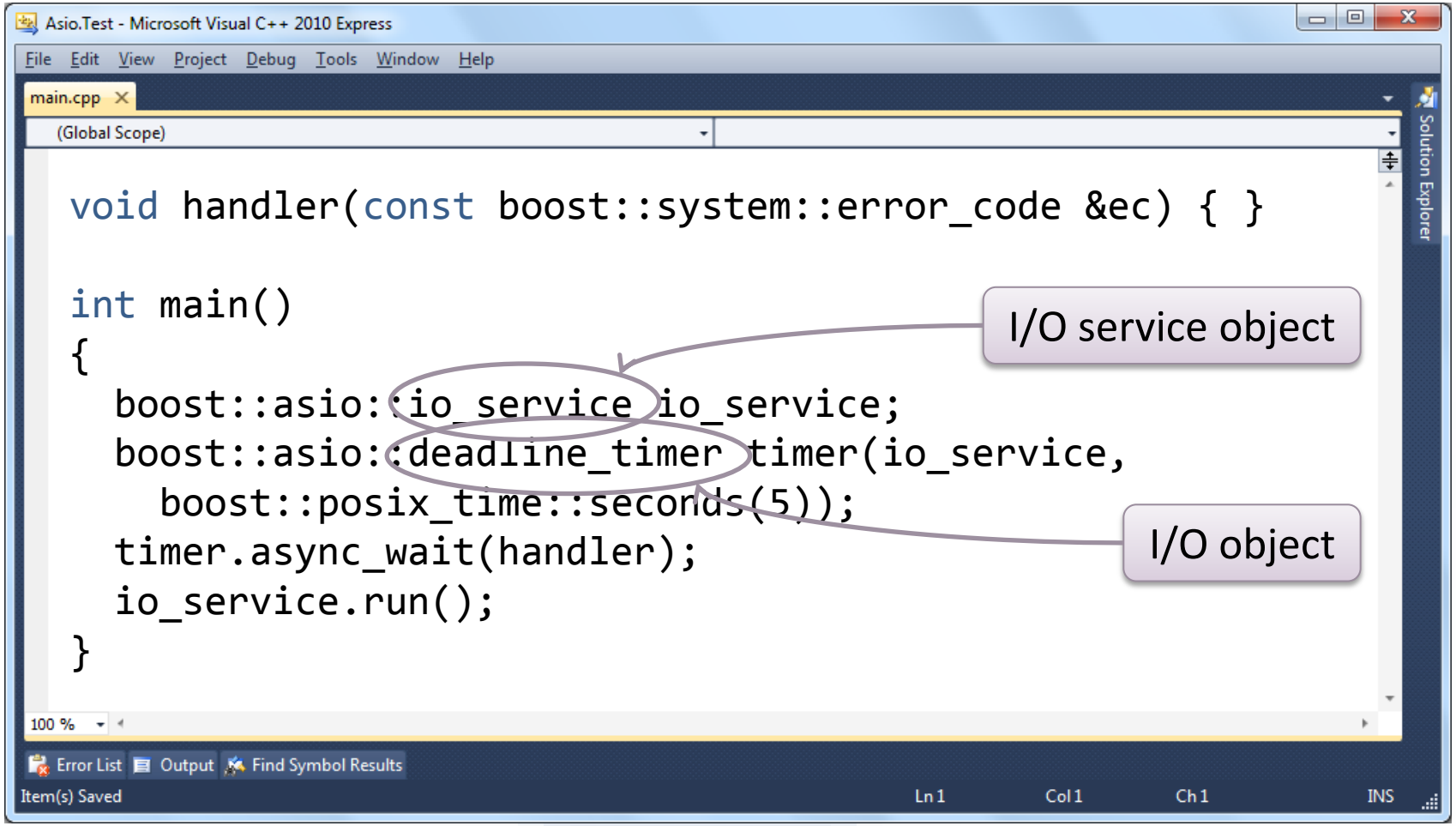
Calling asynchronously means that async_wait() doesn‘t block – after the time expired the function is called which is passed as a parameter (here handler)

Boost.Asio: Asynchronous functions

deadline_timer	async_wait() to wait until some time is expired
ip::tcp::acceptor	async_accept() to accept TCP/IP connections
ip::tcp::resolver	async_resolve() to resolve hostnames
ip::tcp::socket	async_read_some() and async_write_some() to send and receive data

Boost.Asio provides different classes which turn different blocking functions into asynchronous functions

Boost.Asio internals



Boost.Asio internals

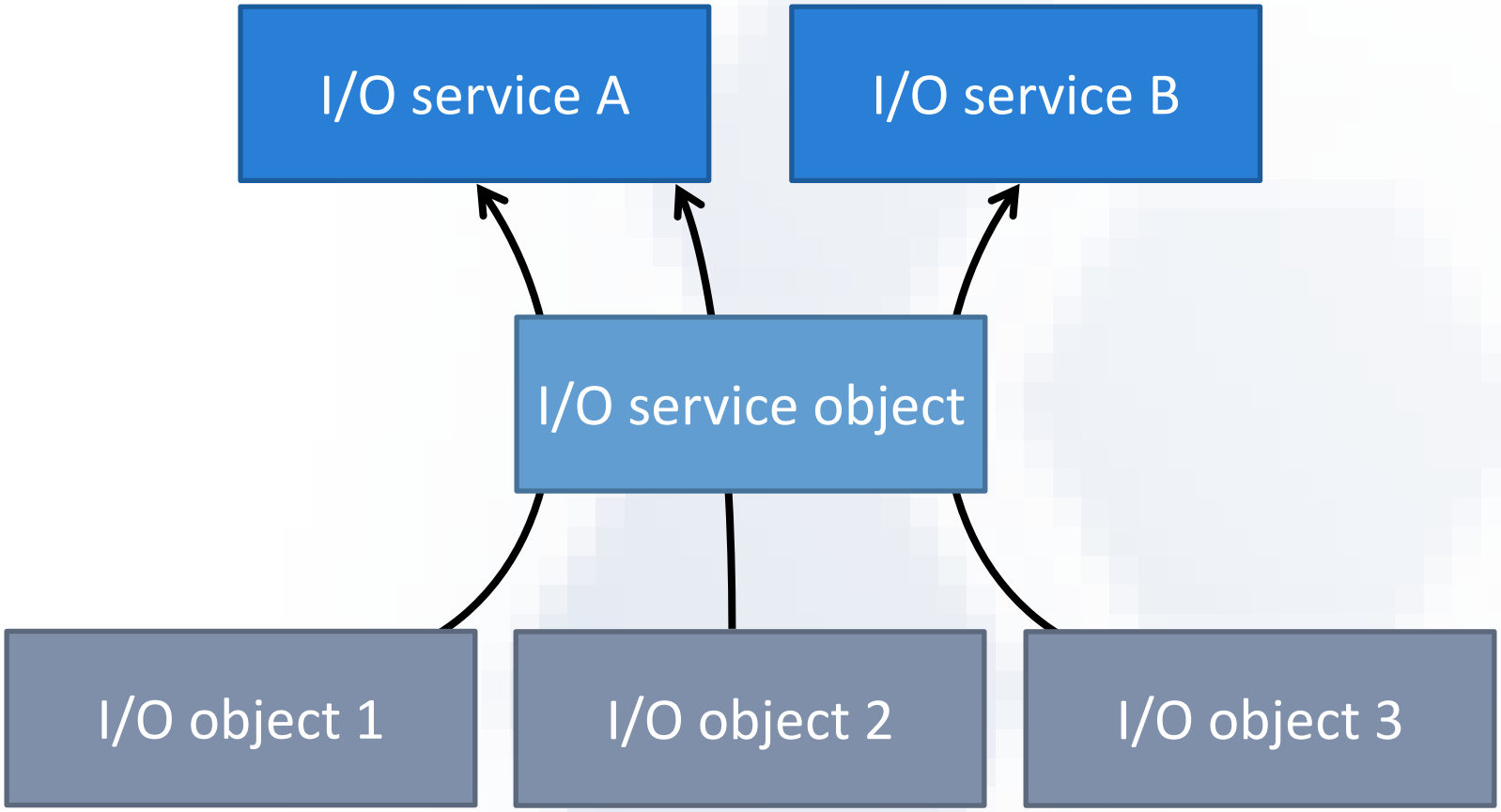
deadline_timer ⟷ io_service

An I/O object is initialized with an I/O service object. It doesn‘t use the I/O service object though – it uses services provided by the I/O service object.

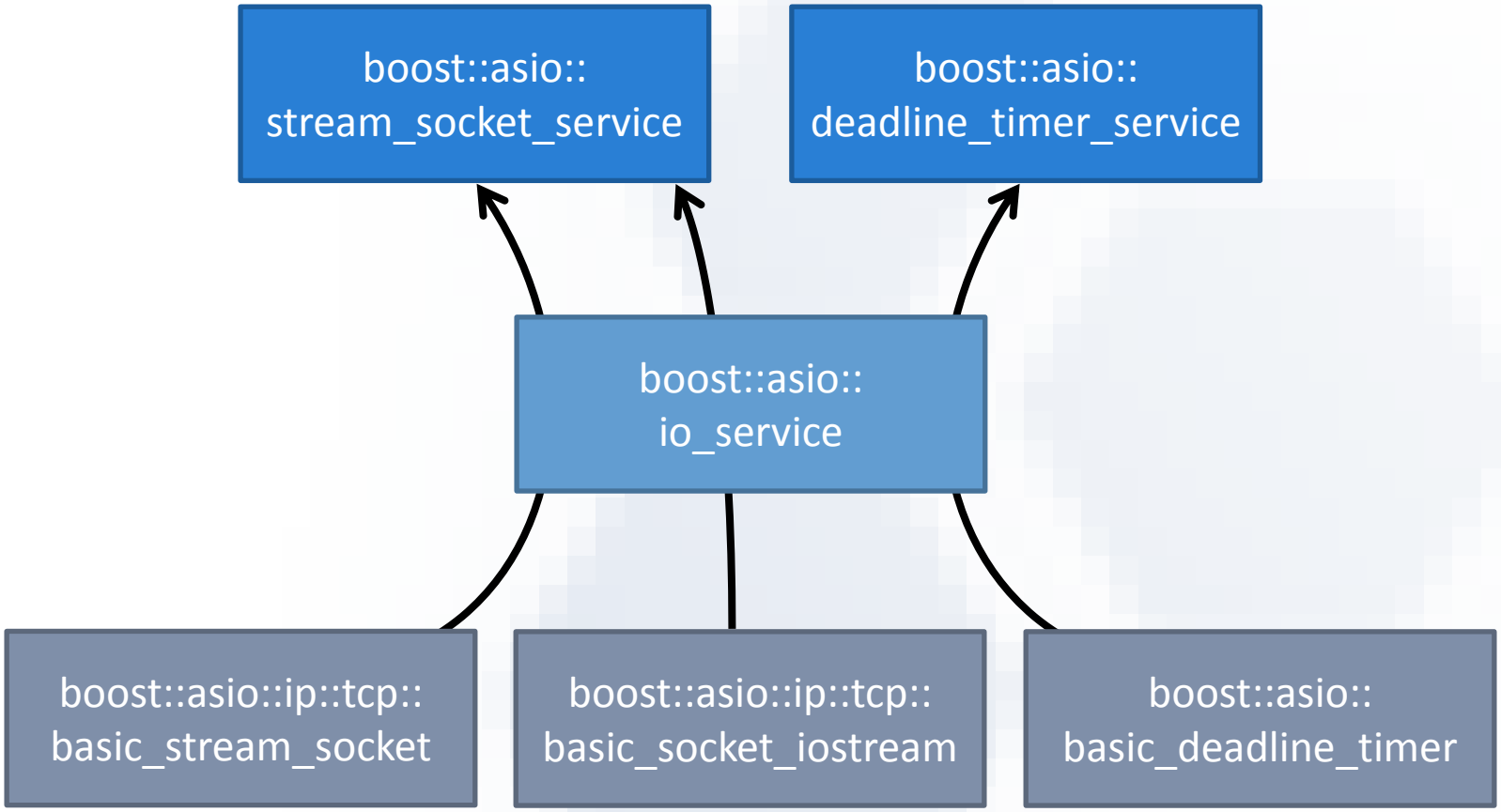
An I/O service object provides services to I/O objects. Think of it as a set of services – set because there is maximum one instance of each and every service.

While I/O objects and I/O service objects are visible in user code, I/O services do the hard work in the background.

Boost.Asio internals

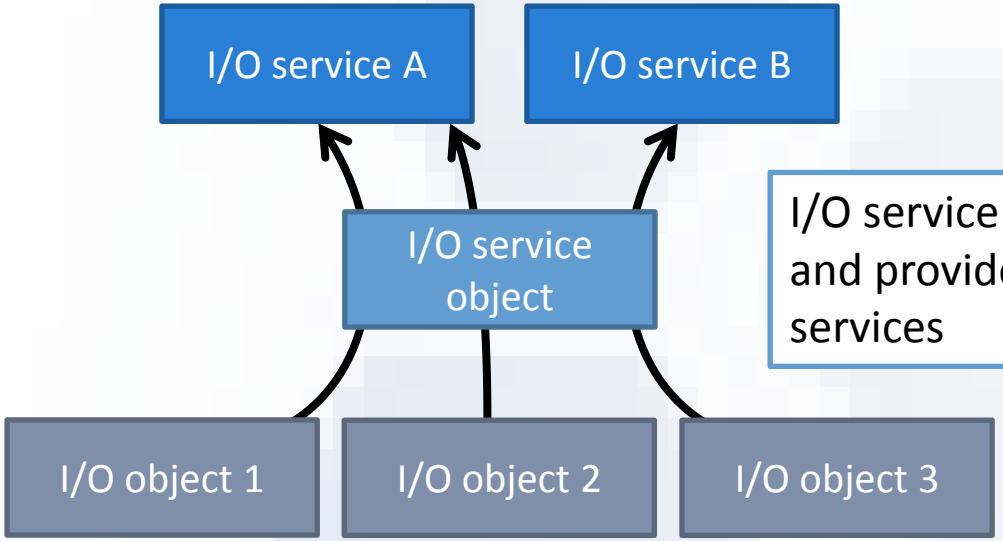


Boost.Asio internals



Boost.Asio internals

I/O services are based on system functions to provide a service to one or several I/O objects

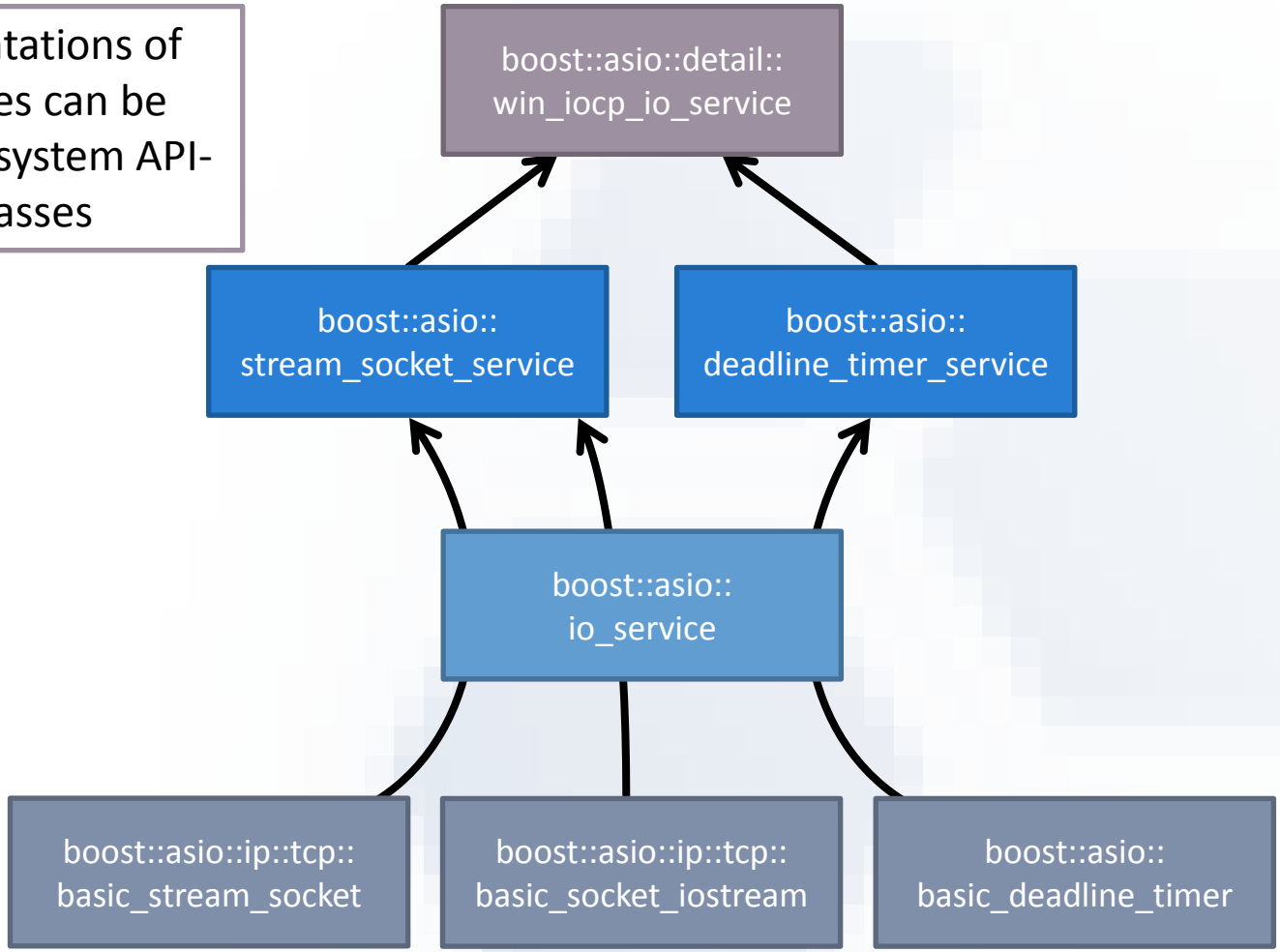


I/O service objects manage and provide access to I/O services

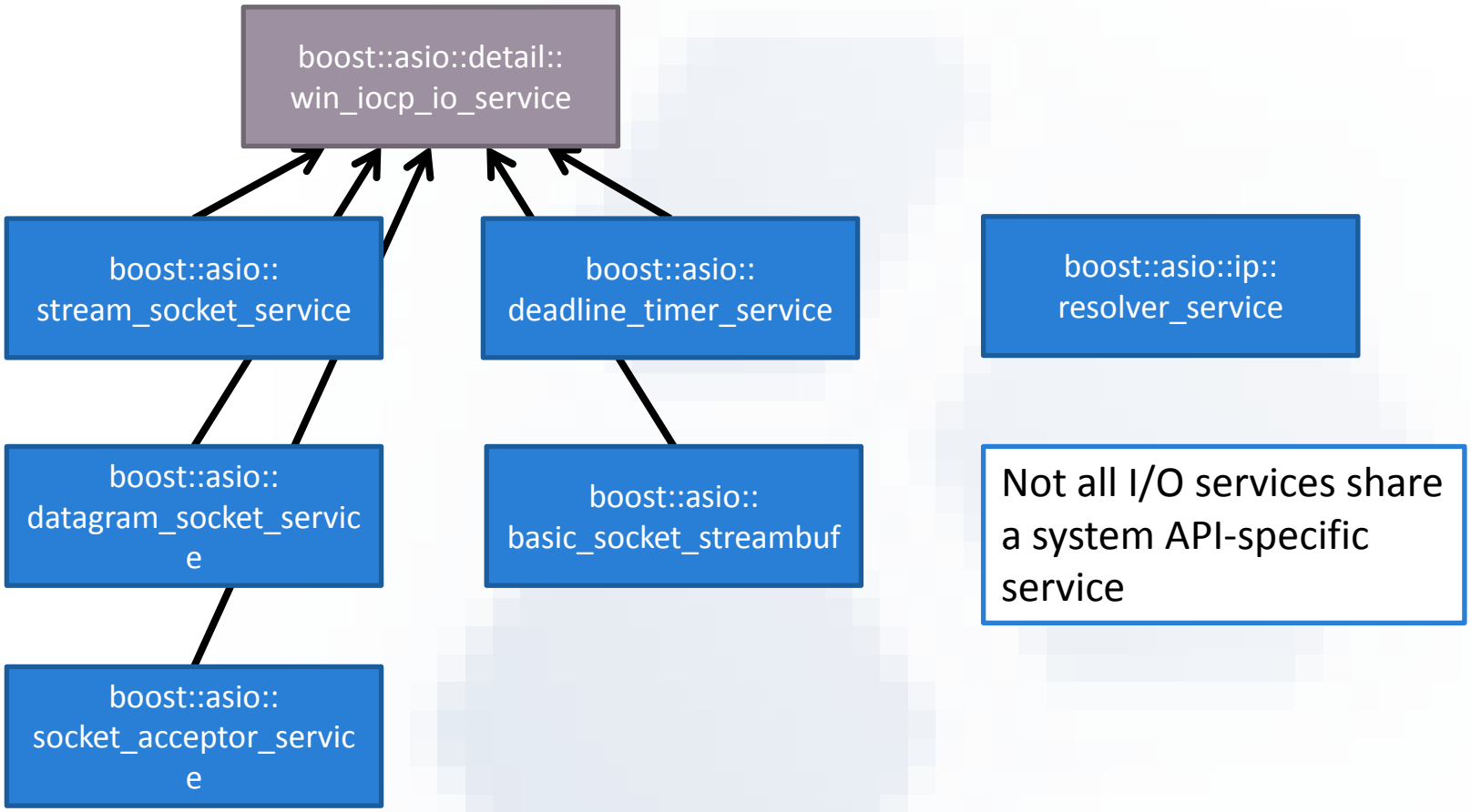
I/O objects get a reference to an I/O service object when instantiated

Boost.Asio internals

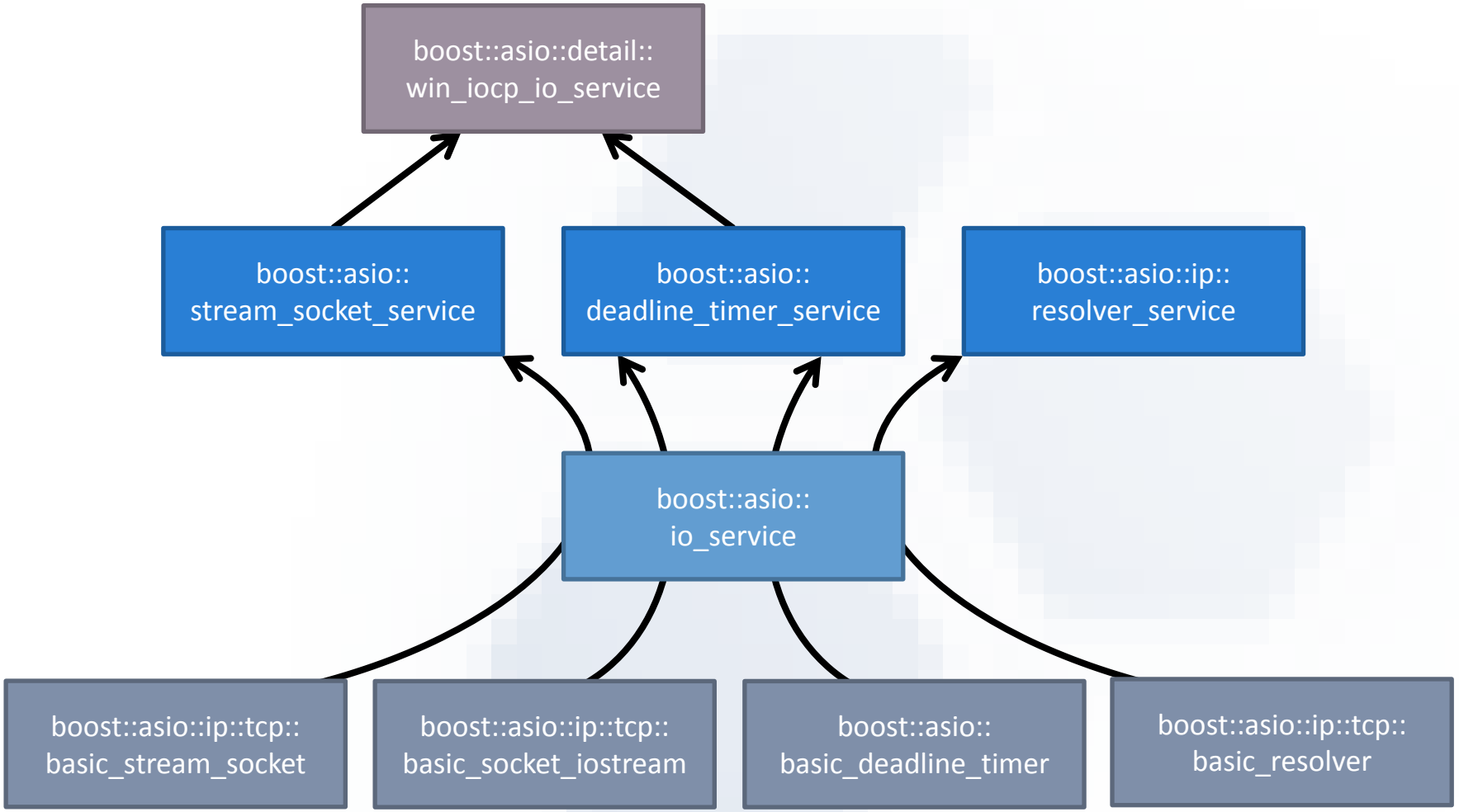
Implementations of I/O services can be based on system API-specific classes



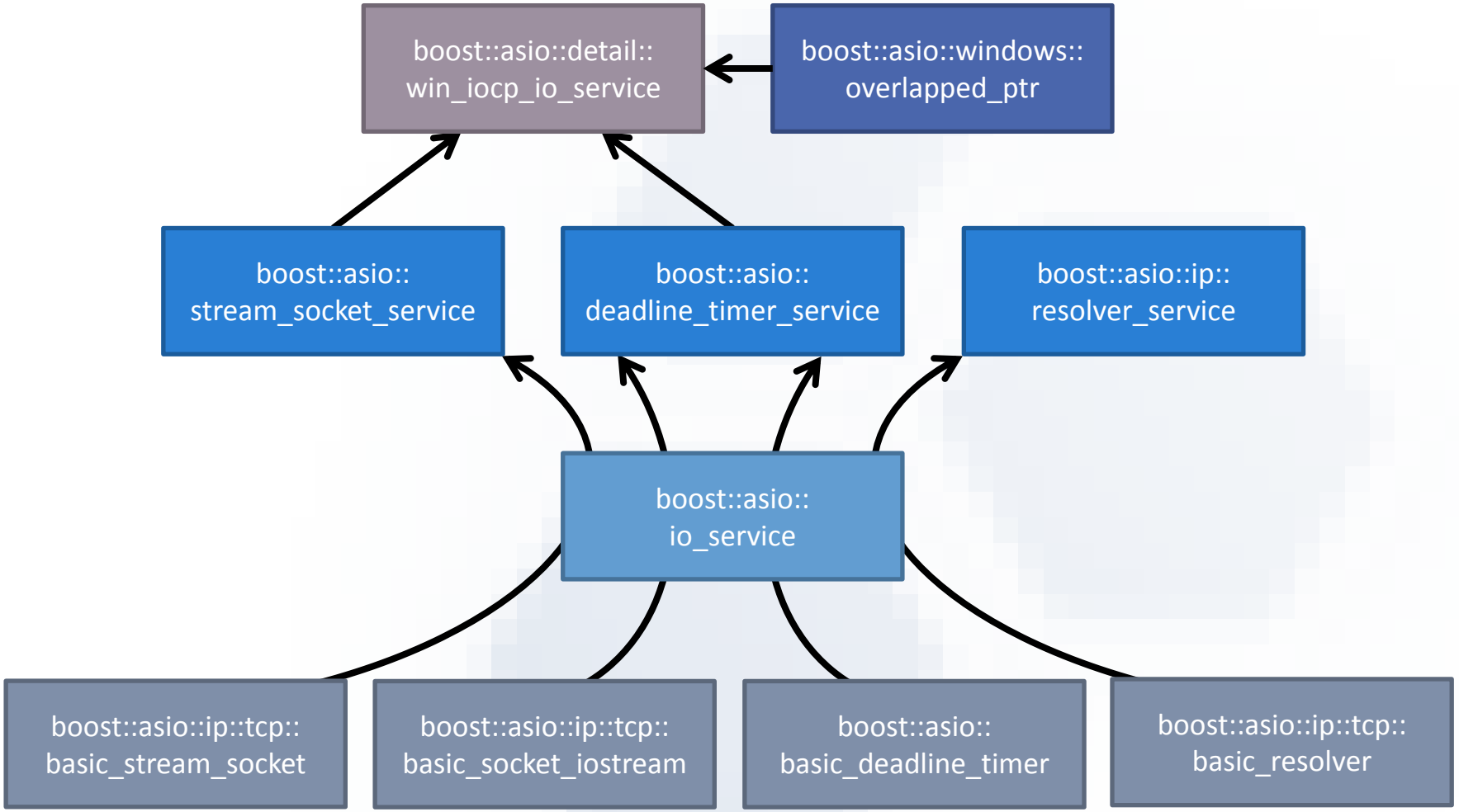
Boost.Asio internals



Boost.Asio internals



Boost.Asio internals



Boost.Asio internals