

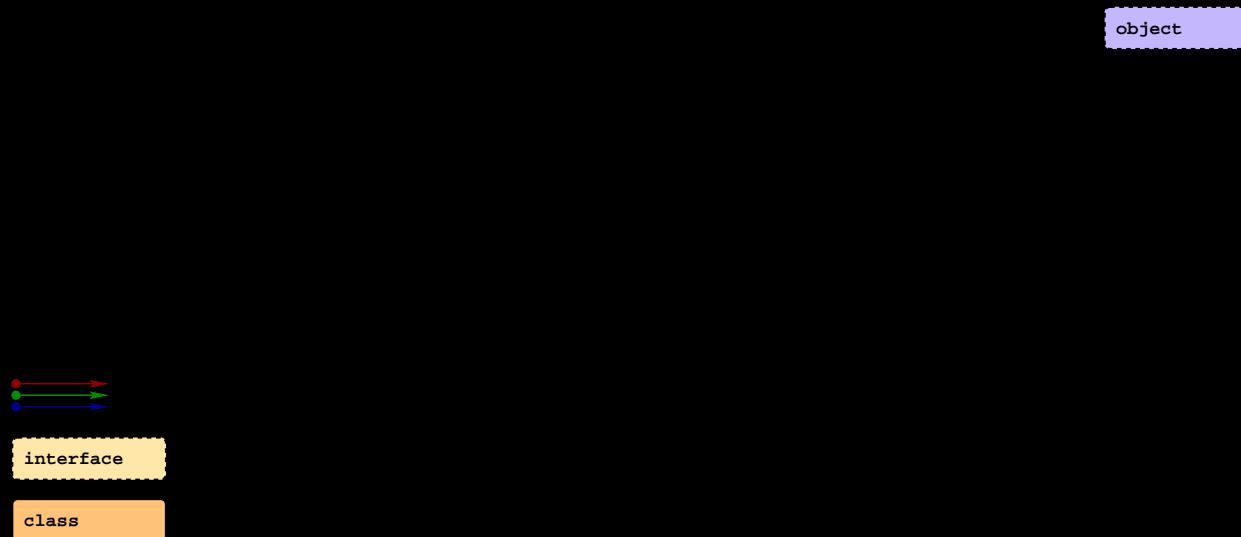
# Outline

---

- Motivation & Overview
- Look & Feel
  - ▶ Error Management
  - ▶ Session Management
  - ▶ Monitoring and Notification
  - ▶ Asynchronous Operations
- Scope
  - ▶ Job Management
  - ▶ File Management
  - ▶ Replica Management
  - ▶ Streaming

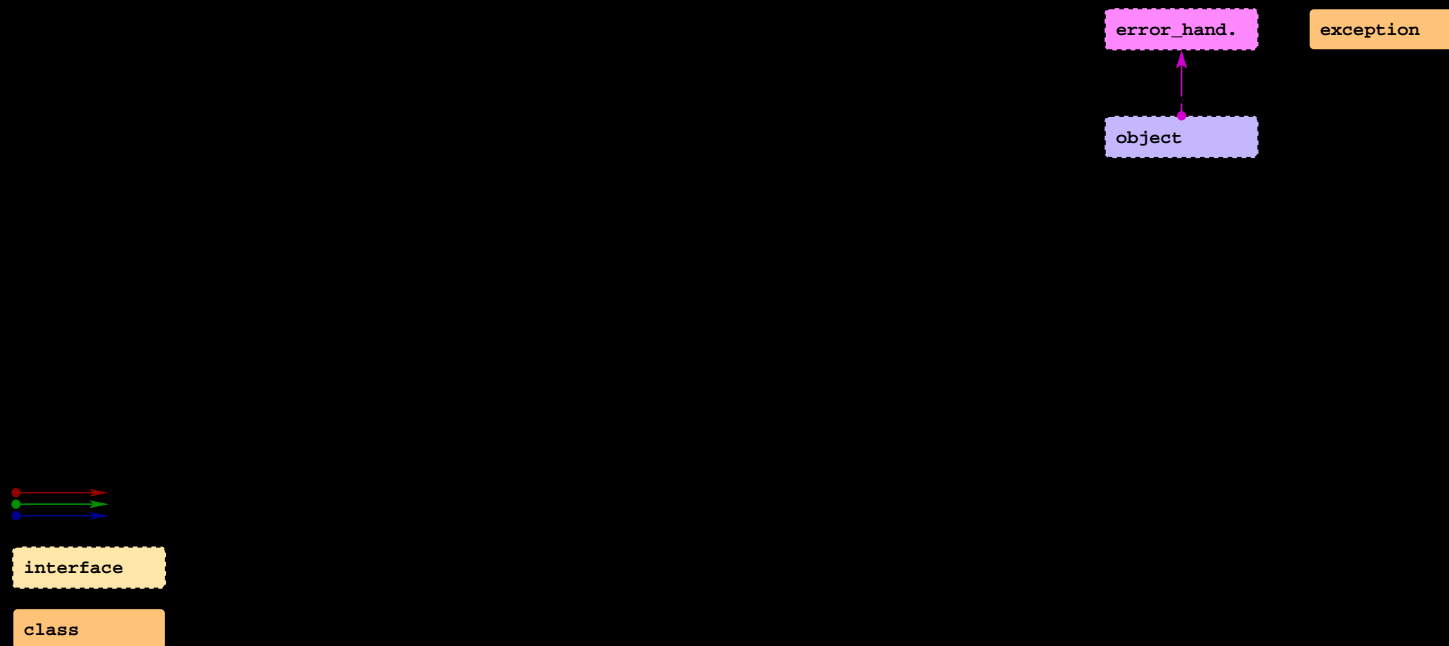
# Interface/Class hierarchy

---



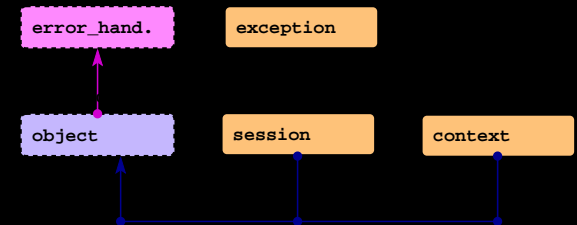
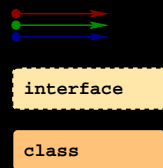
SAGA Look & Feel:  
saga::object allows for object uuids, clone() etc.

# Interface/Class hierarchy



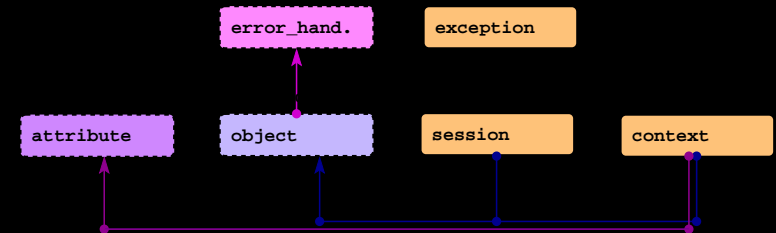
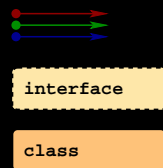
SAGA Look & Feel:  
errors are based on exceptions or error codes,  
depending on the language binding.

# Interface/Class hierarchy



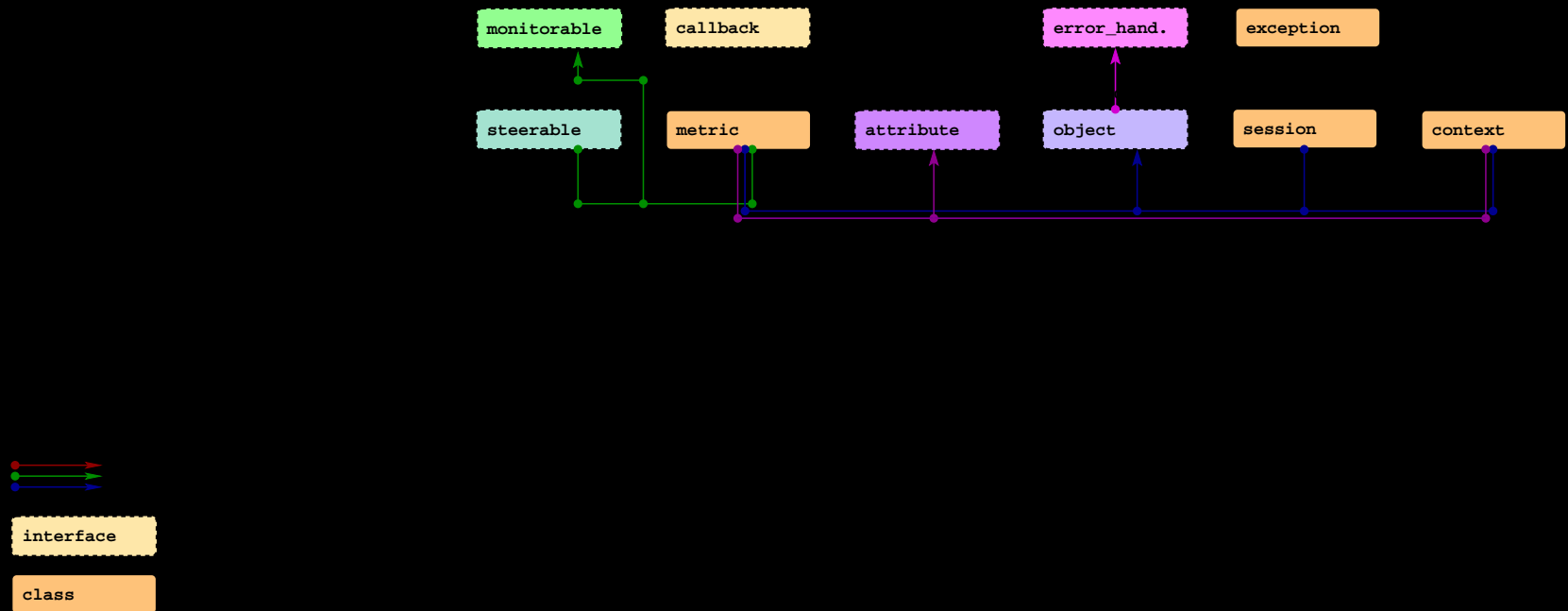
SAGA Look & Feel:  
session and Credential management is, by  
default,

# Interface/Class hierarchy



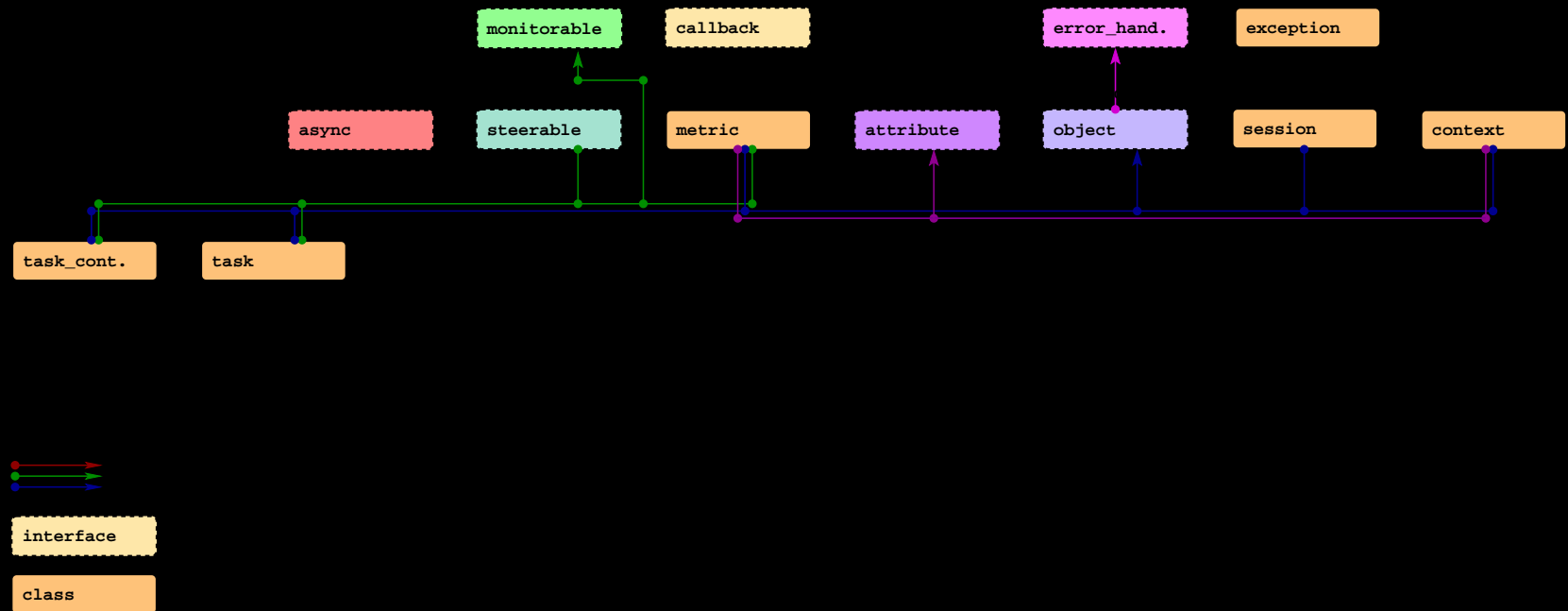
SAGA Look & Feel:  
the Attribute interface allows for meta data on  
several classes.

# Interface/Class hierarchy



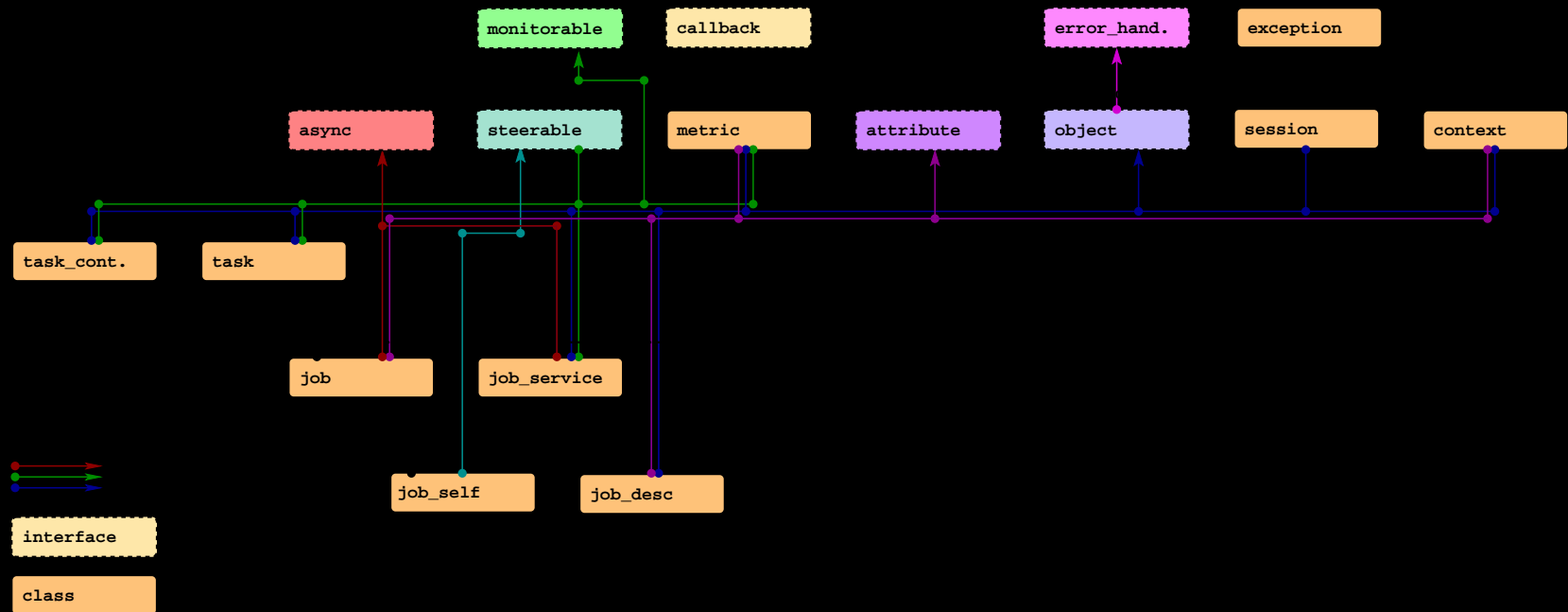
SAGA Look & Feel:  
monitoring does also include asynchronous  
notifications.

# Interface/Class hierarchy



SAGA Look & Feel:  
the task model adds asynchronous operations.

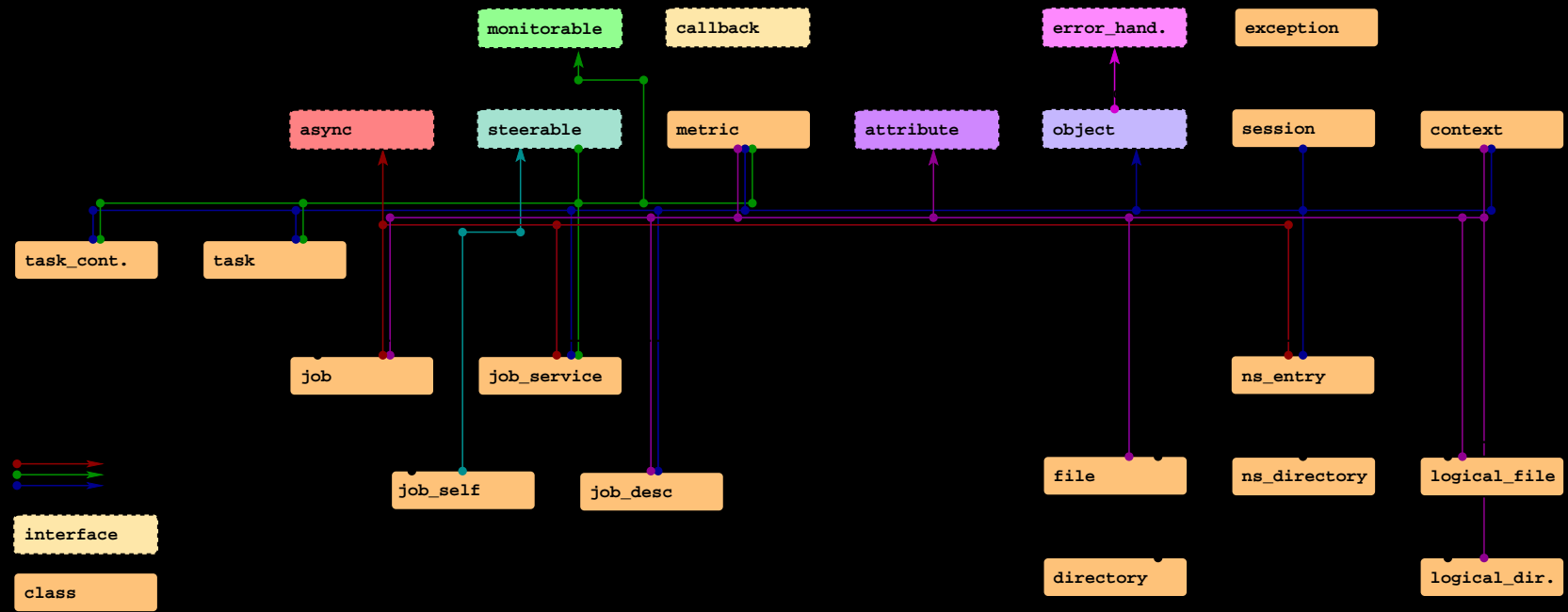
# Interface/Class hierarchy



SAGA Scope:  
job submission and management (DRMAA  
based).

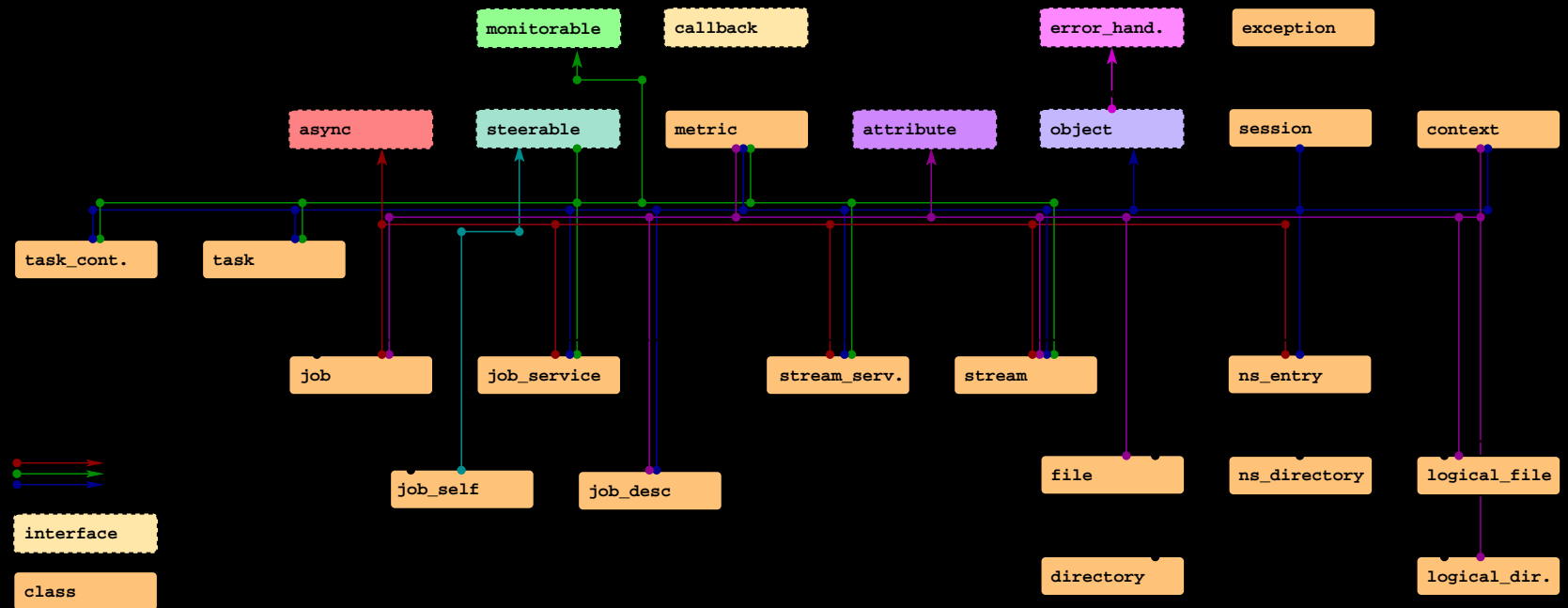


# Interface/Class hierarchy



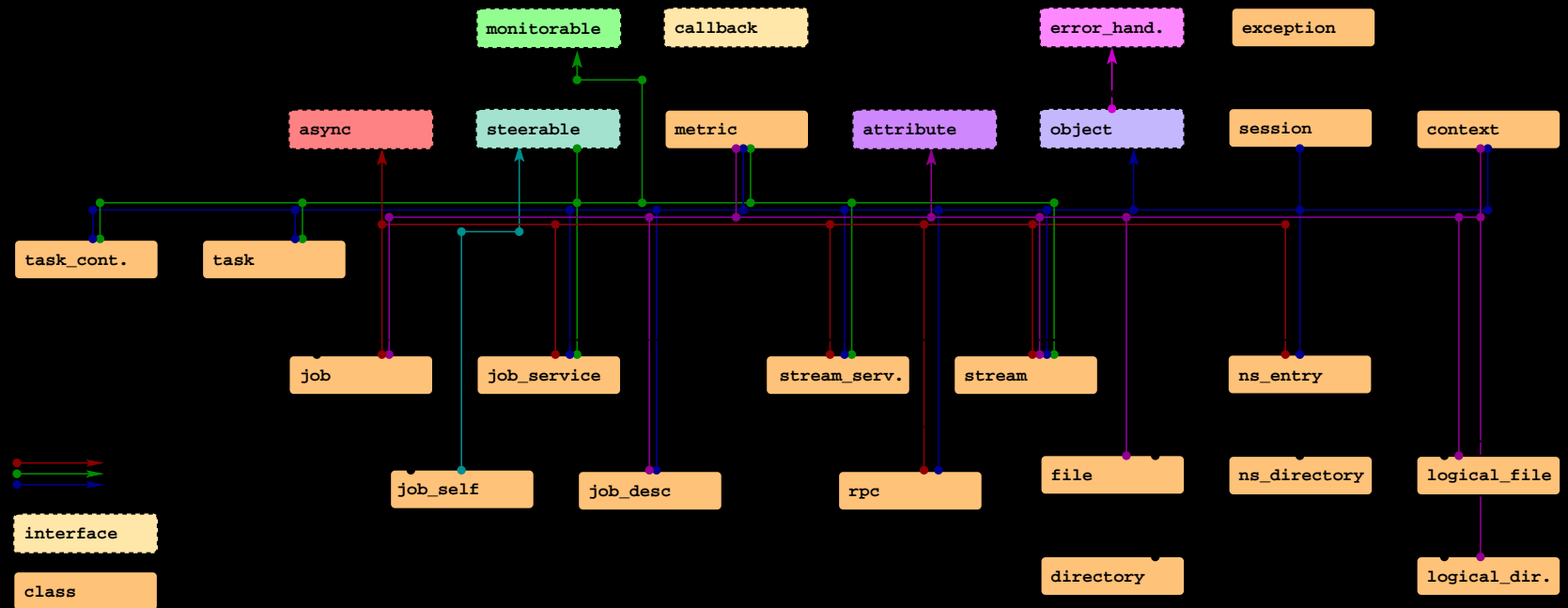
SAGA Scope:  
namespaces for files and replica management  
and remote file access (GFS and O/REP based)

# Interface/Class hierarchy



SAGA Scope:  
authorized data streams

# Interface/Class hierarchy



SAGA Scope:  
remote procedure calls (GridRPC based).

# Implementation Status

---

## ● C++

- ▶ complete implementation by CCT/VU
- ▶ in sync with spec, but work in progress
- ▶ other language bindings planned on top

## ● Java (out of sync with the spec)

- ▶ partial implementation (jobs, files) by DEISA/EPCC
- ▶ complete implementation by OMII-UK

## ● MiddleWare Bindings

- ▶ DEISA/EPCC binds to DEISA
- ▶ OMII-UK binds to OMII stack, but is flexible