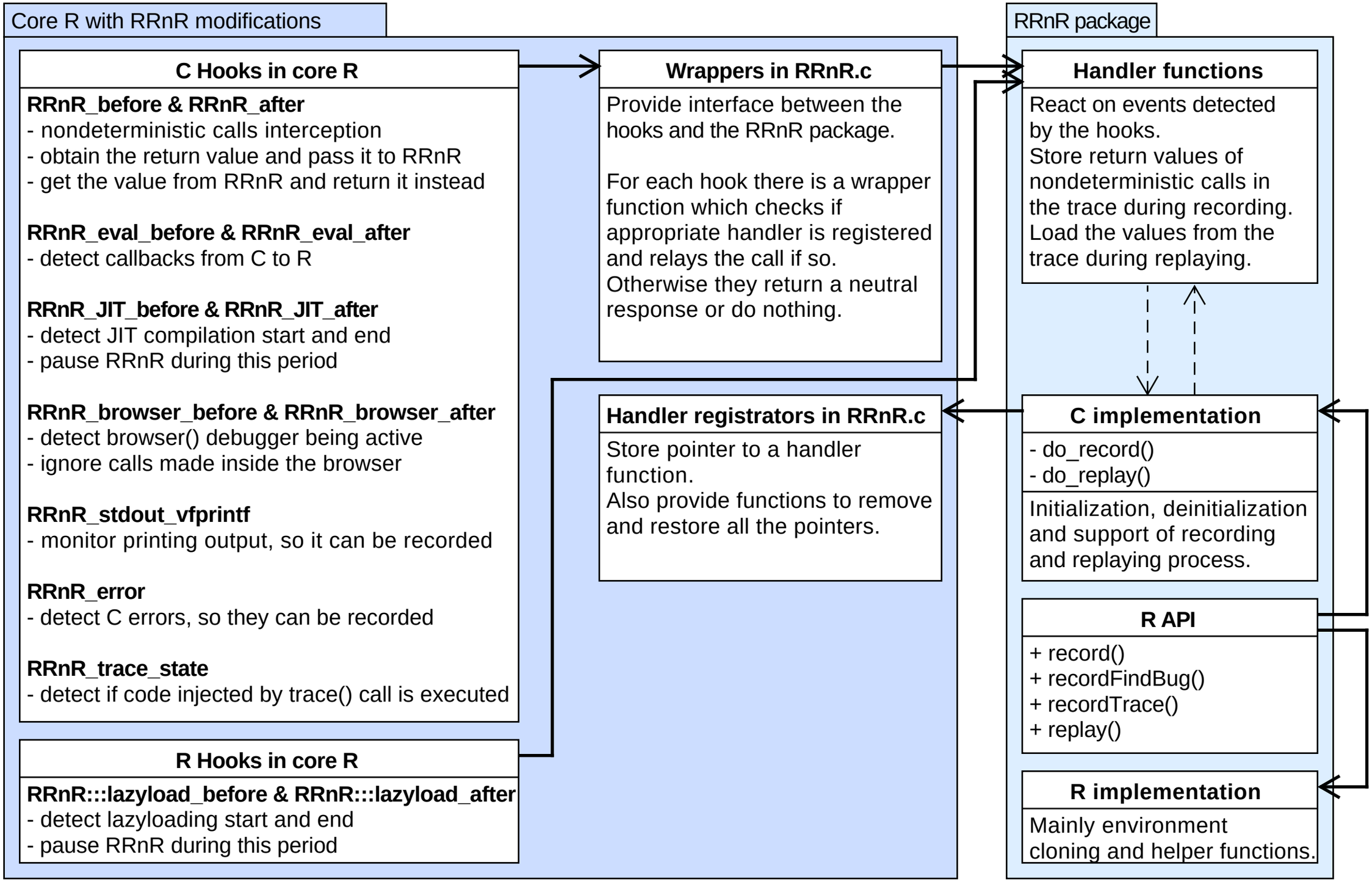
Core R with RRnR modifications
C Hooks in core R
RRnR_before & RRnR_after
- nondeterministic calls interception
- obtain the return value and pass it to RRnR
- get the value from RRnR and return it instead
RRnR_eval_before & RRnR_eval_after
- detect callbacks from C to R
RRnR_JIT_before & RRnR_JIT_after
- detect JIT compilation start and end
- pause RRnR during this period
RRnR_browser_before & RRnR_browser_after
- detect browser() debugger being active
- ignore calls made inside the browser
RRnR_stdout_vfprintf
- monitor printing output, so it can be recorded
RRnR_error
- detect C errors, so they can be recorded
RRnR_trace_state
- detect if code injected by trace() call is executed
R Hooks in core R
RRnR:::lazyload_before & RRnR:::lazyload_after
- detect lazyloading start and end
- pause RRnR during this period
Wrappers in RRnR.c
Provide interface between the hooks and the RRnR package.
For each hook there is a wrapper function which checks if appropriate handler is registered and relays the call if so.
Otherwise they return a neutral response or do nothing.
Handler registrators in RRnR.c
Store pointer to a handler function.
Also provide functions to remove and restore all the pointers.
RRnR package
Handler functions
React on events detected by the hooks.
Store return values of nondeterministic calls in the trace during recording.
Load the values from the trace during replaying.
C implementation
- do_record()
- do_replay()
Initialization, deinitialization and support of recording and replaying process.
R API
+ record()
+ recordFindBug()
+ recordTrace()
+ replay()
R implementation
Mainly environment cloning and helper functions.