Untangling spaghetti of platform/scheduler/

haraken@chromium.org, altimin@chromium.org

2018 Jan 10

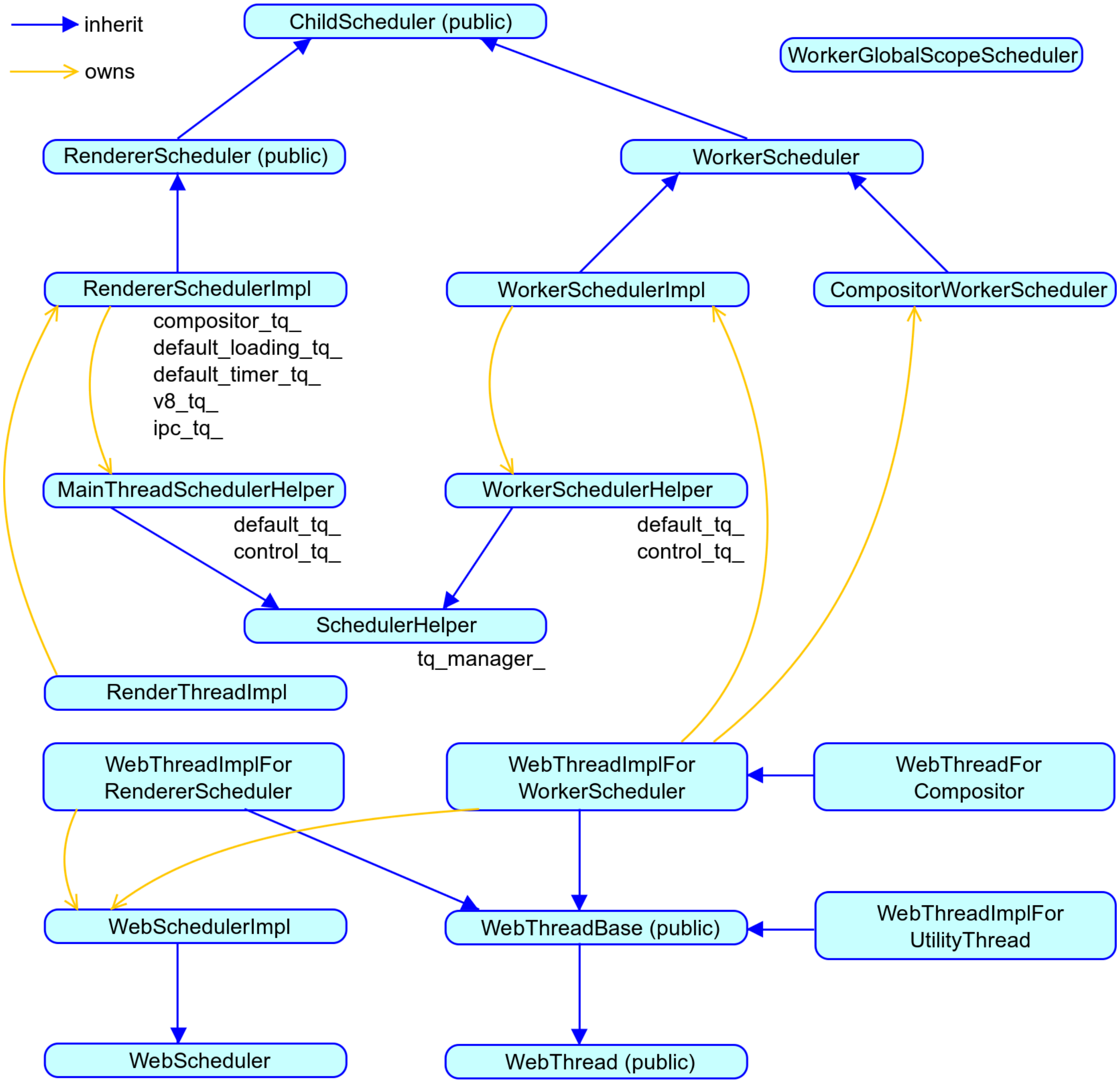
Status: PUBLIC

Parent document: Better layering for platform/scheduler. This document was written with a lot of insights from Alexander :D

platform/scheduler/ has a bunch of incoherent abstraction layers (partly because historically platform/scheduler/ was living in //content/ and moved to Blink). The class hierarchy is really complex and has been slowing down engineering velocity. This document proposes a rearchitecture plan of platform/scheduler/.

Class design (around ChildScheduler)

Today we have the following class relationship:

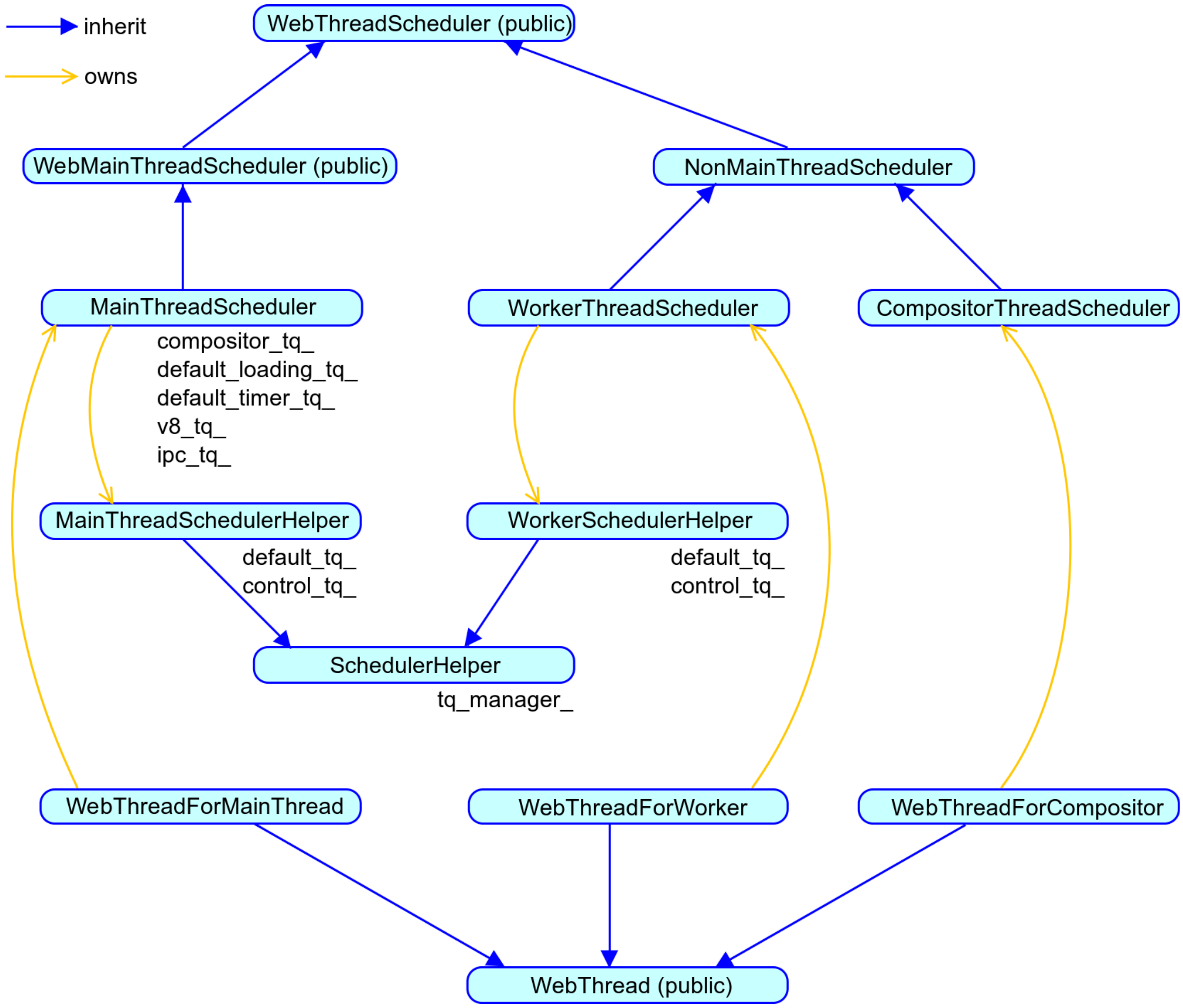


There are a couple of points that may confuse developers:

- Public classes are not prefixed with "Web". Not-public classes are prefixed with "Web".
- WebScheduler is used to expose ChildScheduler and RendererScheduler to Blink. WebScheduler should be merged into RendererScheduler and WorkerScheduler.
- WorkerGlobalScopeScheduler is isolated. It should be merged into WorkerScheduler.
- WebThreadForCompositor does not inherit from WebThreadBase. WebThreadForCompositor does not own CompositorWorkerScheduler.
- It's strange that WebThread"Base" inherits from WebThread. WebThreadBase should be removed.
- WebThreadImplForRendererScheduler does not own RendererSchedulerImpl.

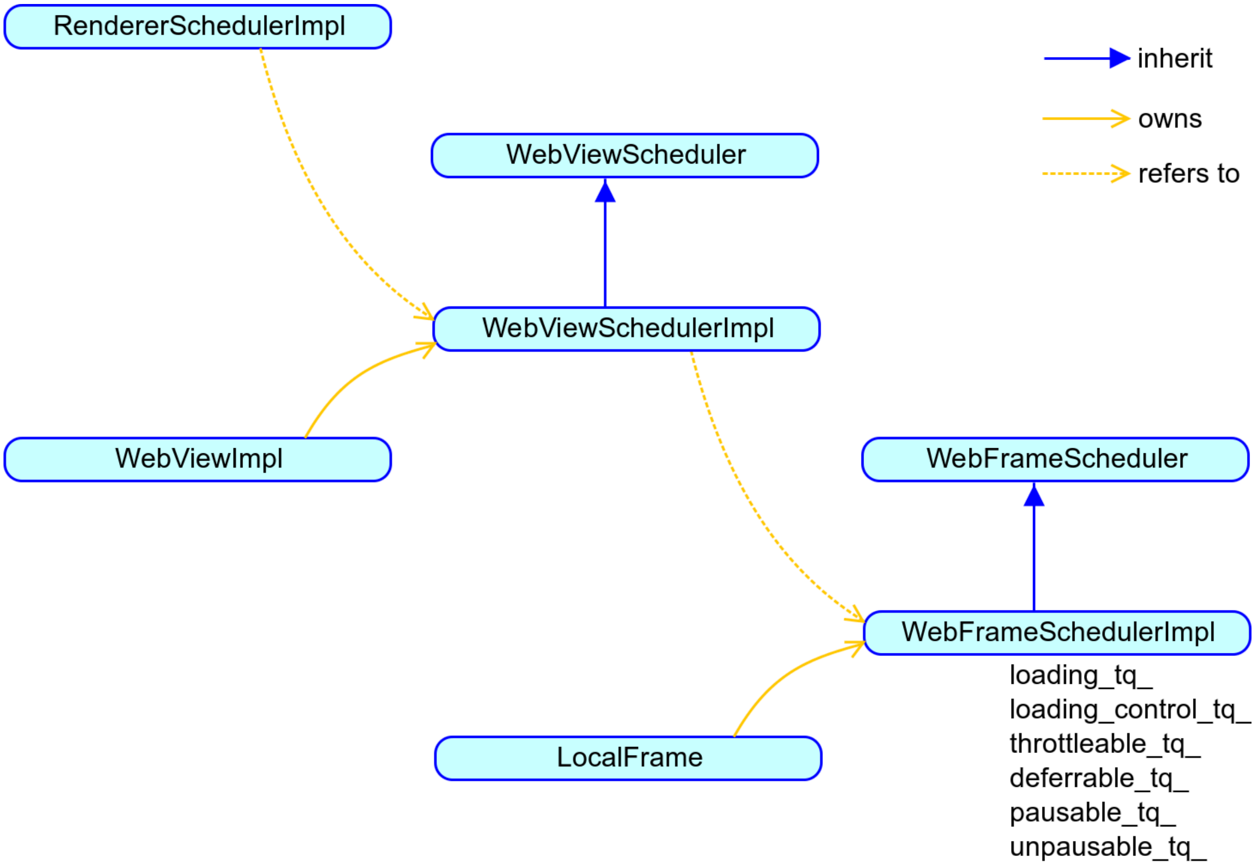
- The utility thread is not a component of Blink. WebThreadImplForUtilityThread should move back to //content/.

In summary, I'd propose reorganizing the class relationship as follows:



Class design (around WebViewScheduler)

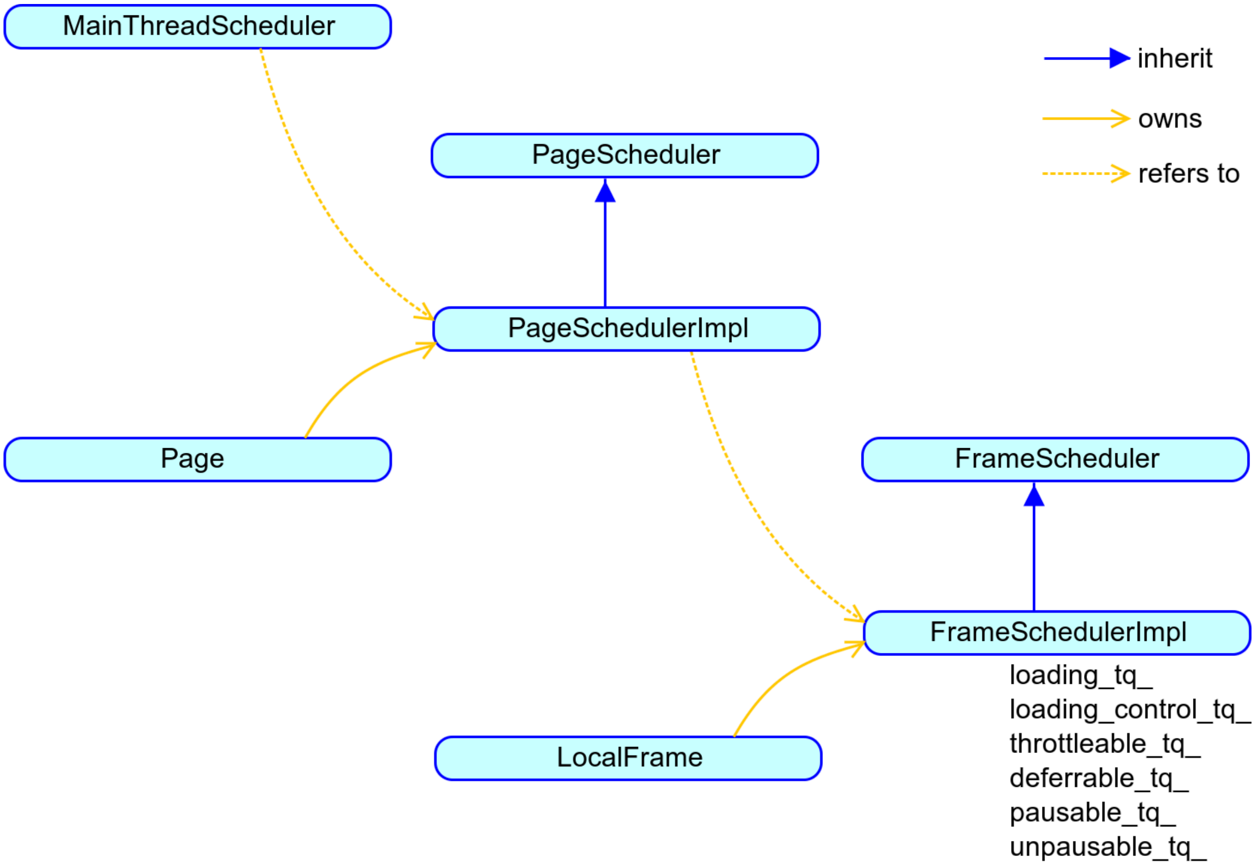
Today we have the following class relationship:



RendererScheduler : WebViewScheduler : WebFrameScheduler = 1 : N : M

RendererScheduler is per main thread. WebViewScheduler is per Page. WebFrameScheduler is per Frame.

I'd propose reorganizing the class relationship as follows:



MainThreadScheduler : PageScheduler : FrameScheduler = 1 : N : M

Directory structure

Currently platform/scheduler/ has base/, child/, renderer/, util/, utility/ and test/.

- platform/scheduler/base/ should move to //base/.
- The distinction between child/ and renderer/ don't make sense because platform/scheduler/ is now in Blink.
- utility/ does not make sense because the utility thread is not a component of Blink. utility/ should move back to //content/.

The remaining files should be reorganized with the following directory structure:

- **public/** : Classes that should be exposed to the rest of Blink. public/ should contain only MainThreadScheduler, WorkerThreadScheduler, Timer and ThreadCPUThrottler.
- **main_thread/** : Classes about main thread scheduling.

- **worker/**: Classes about worker scheduling. Note that workers mean non-main threads, not web workers.
- **common/** : Classes shared between the main thread and workers.
- **util/**: Helper utility classes used by scheduler (mostly for tracing and metrics). They should not depend on the rest of the scheduler. (Note from altimin@: loading/ code uses some metric helpers. It may make sense to expose util/ directory to the rest of the blink. ThreadCPUThrottler could go here too. public/util/ is possible too).
- **test/**: Provides mocks and dummy implementations of scheduling primitives for testing. Test-only, can be referenced from outside the scheduler.

Blink public APIs

Blink public APIs should be prefixed with "Web". WebThreadScheduler and WebMainThreadScheduler would be the only classes we should expose from Blink. The following classes should be removed:

- WebThreadBase
- SingleThreadIdleTaskRunner : This is used by Blink only, so it should be easy to move it to Blink.
- RenderWidgetSchedulingState : This should be merged into WebMainThreadScheduler.

Others

- WebTaskRunner should be replaced with base::SingleThreadTaskRunner. base::SingleThreadTaskRunner should be used all over the Blink code base.